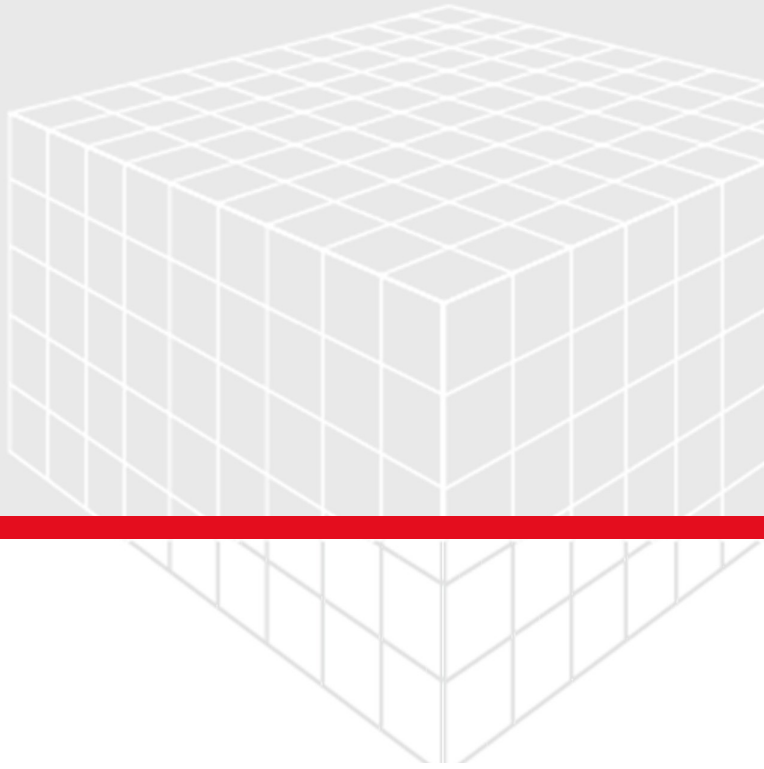
FilterDict

User Guide



GEODICT Release 2026

Published: June 2026

GeoDict

https://doi.org/10.30423/userguide.geodict

FilterDict

1 FilterDict

Predict filter efficiency, filter lifetime, and dust holding capacity of a filter with the FilterDict module.

With FilterDict, you can predict the outcome of single pass or multi pass tests in various setups. Single pass tests can be performed with constant flow rate or constant pressure drop. Multi-pass tests can be performed starting from an initially clean or initially polluted test reservoir.

The fluid to be filtered can be air, water, or oil, and test dusts may be SAE standard test dusts, soot, or any user defined particle size distribution. In fact, any liquid-solid separation can be simulated for as long as the concentration of dust particles is relatively low. Sludges or slurries are not possible. Since the 2023 release FilterDict includes methods to simulate coalescence of liquid droplets. The coalescence models are separately licensed, so please contact filtration@math2market.de if you are interested in this feature.

FilterDict models depth and cake filtration, either in a crossflow or the usual dead-end filtration setup. Not yet included in FilterDict is the possibility to simulate filter cleaning by pulse jets or other methods.

FilterDict allows simulations on different length scales. Depending on your FilterDict license, different options are selectable.

At the **Filter Media** scale, simulate the filtration process on a porous media generated with GeoDict's structure generation modules, or on models obtained from 3D-image data (µCT, FIB-SEM, etc.) imported and segmented using ImportGeo-Vol.

At the **Filter Element** scale, simulate the filtration process on single or multiple pleats generated with PleatGeo, on filter elements, or on DPF (Diesel Particulate Filter) structures.

At the **Complete Filter** scale, simulate the filtration process on a CAD model of the complete filter including the filter housing. The CAD model must be imported using ImportGeo-CAD.

New in GeoDict 2026

- Particles can start with an initial velocity 73 that is different from the velocity of the surrounding fluid.

Learn to use FilterDict

- Theoretical Background 5
  How FilterDict simulates fluid flow and the particle movement within that flow to predict filter efficiency, filter lifetime and dust holding capacity.
- Filter Media 18
  Filter Media simulations are suited for the analysis of filter media. Micro-scale 3D structure models resolve every fiber or grain of the filter material. Often, the analyzed particles are resolved in the simulation domain: This means that they are larger than the size of one voxel.
- Filter Element 125

Learn to use FilterDict

Filter Element simulations are suited for the simulation of pleats, filter parts or filter elements. At this scale, the details of the inner structure of the filter media are usually no longer resolved.

- Complete Filter 169
  Complete Filter simulations are suited for the simulation of a complete filter including the filter housing. Here, different inlet and outlet geometries can be simulated
- Cross Flow 217
  Cross Flow simulations are suited for the simulation of a filter with multiple outlets, where one outflow contains the unfiltered bulk fluid and the other the permeate
- Filter GeoApps 231
  Pre-defined macros included in GeoDict that are useful for filtration simulations.
- User-Defined Functions (UDFs) 238
  With user defined functions you may change the governing equations and introduce additional forces.
- References 257
  Journal articles cited in this section.

Join the conversation

Visit the Community on GeoDict Forum to be inspired and get answers to top questions.
Find illustrative application examples on the website.
Our tutorials and seminar videos give you a first hands-on experience. They are collected in the Learning Center.
Send your Support Request to our technical Support.

1.1 Theoretical Background

The simulation of filter processes with the FilterDict module consists of the following steps:

1. Fluid flow: The flow through the (clean or partially clogged) filter is computed by solving the appropriate equations within the fluid phase.
2. Particle Tracking: Particles are tracked in the calculated flow field. Tracking of particles is based on solving an ordinary differential equation, and includes drag forces, electrostatic attraction between particles, and filter surface, as well as diffusive motion. Within one batch, the particles are treated as independent objects. It is always assumed that the particle density is low and that the moving particles have no effect on the flow. Therefore, FilterDict can only be used in applications where these assumptions hold. This is the case in most areas of air and liquid filtration. It does not hold for sludge filtration, where the particles and their interaction determine the properties of the fluid.
3. Particle Collisions (at Surfaces): When a moving particle collides with a solid surface, the collision model decides whether the particle sticks to the surface or bounces off and continues traveling.
4. Particle Capture (in Porous Materials): When a particle moves through a porous material, the particle may be deposited inside of the material. The pass through model decides how probable it is for the particle to move through the material layer. This is only applicable if both the particle and the porous material are unresolved by the voxel grid, i.e. particles and pores are smaller than the voxel length.
5. Filter Clogging: Particles that stick to the surface after a collision and particles that are captured inside a porous material do not continue their movement. Rather, they are considered as *deposited* particles. These particles clog the filter and increase the flow resistivity locally. The *clogging and flow resistivity* models describe the relation between the volume of the deposited particle and the increase of the local flow resistivity. Different models are used for particles that are resolved by the voxel grid or particles that are unresolved in the voxel grid.

In filter lifetime simulation, the simulation spans the whole filter life. Therefore, the steps flow computation, particle tracking, particle collision or absorption, and filter clogging are iterated to model the changing filter properties over time.

In this case, one iteration corresponds to a (physical) time interval in an experiment. The amount of particles that is simulated in one iteration is called a *batch*. The batch size corresponds to the challenged mass of the filter experiment in the simulated time interval.

1.1.1 Fluid Flow

FilterDict assumes that the flow is stationary throughout a batch, i.e. no turbulences or oscillating behavior occurs. Therefore, the partial differential equations that need to be solved do not contain derivatives regarding time. Other assumptions are that the fluid is a Newtonian fluid and it is incompressible.

Navier Stokes Equations

If the geometry of the pores is resolved by the computational grid, all voxels are either pore space or solid material. The fluid flow in the pores is described by the stationary Navier-Stokes equations:

$$-\mu\Delta\vec{u} + \rho(\vec{u}\cdot\nabla)\vec{u} + \nabla p = \vec{f} \tag{1}$$

$$\nabla\cdot\vec{u} = 0 \tag{2}$$

Here, ρ is the fluid density, μ is the dynamic viscosity, $\vec{u}$ is the velocity vector, p is the pressure and $\vec{f}$ is the force.

Stokes Equations

In a filter, the flow is often slow and laminar. In such cases, the Navier-Stokes equations can be simplified to the Stokes equations, which read as follows:

$$-\mu\Delta\vec{u} + \nabla p = \vec{f} \tag{3}$$

$$\nabla\cdot\vec{u} = 0 \tag{4}$$

Here, μ is the dynamic viscosity, $\vec{u}$ is the velocity vector, p is the pressure and $\vec{f}$ is the force.

Navier-Stokes-Brinkman Equations

If the geometry of the pores is not fully resolved by the computational grid, a voxel may describe pore space, solid material, or porous material. The Brinkman term allows to describe flow in an unresolved porous medium. In filter media simulations, this is the case when the filter becomes clogged with very small particles, i.e. the dust particle size is smaller than the voxel size. The Navier-Stokes-Brinkman equations are given as follows:

$$-\mu\Delta\vec{u} + \rho(\vec{u}\cdot\nabla)\vec{u} + \sigma\vec{u} + \nabla p = \vec{f} \tag{5}$$

$$\nabla\cdot\vec{u} = 0 \tag{6}$$

Here, $\vec{u}$ is the velocity vector, p is the pressure, $\vec{f}$ is the force, and μ is the dynamic viscosity. The local viscous flow resistivity σ of a voxel is related to the local permeability by

$$\sigma = \frac{\mu}{\kappa}, \tag{7}$$

where κ is the isotropic permeability of the voxel. In the pore space, the flow resistivity is zero and the Brinkman term disappears.

The initial structure as input for Filter Media 18 simulations contains only solid material and pores. The porous domains appear when dust particles smaller than the voxel size are caught in the filter. Then, the number of regions where the Brinkman term is not zero becomes larger with each time step. In Filter Element 125 or Complete Filter 169 simulations, however, the pores and solids in the initial structure might be unresolved, and therefore a porous domain exists initially. Here, the Brinkman term describes porous voxels representing the porous media, as well as porous voxels containing deposited particles.

Stokes-Brinkman Equations

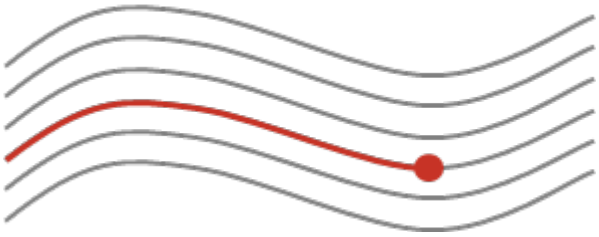
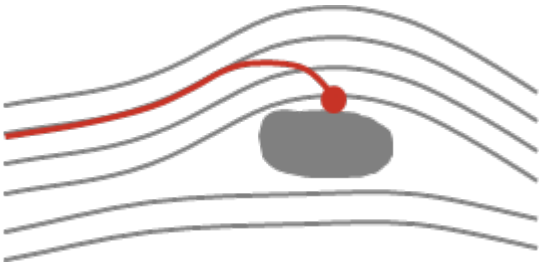
Again, under the assumption of a slow and laminar flow, the equations can be simplified and given as:

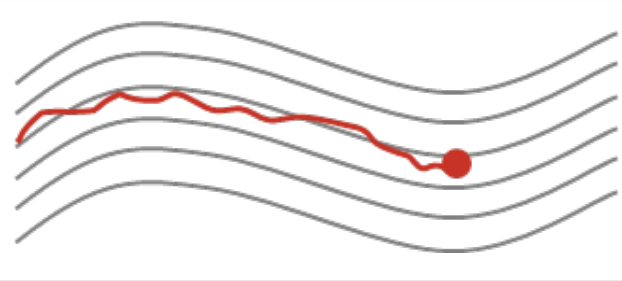
$$-\mu\Delta\vec{u} + \sigma\vec{u} + \nabla p = \vec{f} \tag{8}$$

$$\nabla \cdot \vec{u} = 0 \tag{9}$$

1.1.2 Particle Tracking

FilterDict simulates filter clogging and tracks how particles are caught in the filter. The movement of particles in the fluid is influenced by the following factors:

	**Drag forces** are the dominant forces in most cases, and particle trajectories are close to the streamlines of the flow field.
	**Electrostatic effects**. The particles and pore surface can be electrostatically charged. This leads to attraction or repulsion effects.



Diffusive (or Brownian) motion. Particle collisions with fluid molecules lead to small random direction changes.

FilterDict simulates the filtration process considering these three effects. Electrostatic attraction and/or diffusive motion can be switched off if is known a-priori that these effects have no influence on the filtration process. Overall, the particle movement is governed by

Particle Momentum = Stokes Drag + External Forces

or, as formula:

$$m\frac{d\vec{v}}{dt} = \gamma\left(\vec{u} - \vec{v} + \sqrt{2D}\frac{d\vec{W}(t)}{dt}\right) + Q\vec{E} + \vec{F} \tag{10}$$

In the friction coefficient

$$\gamma = 6\pi\mu\frac{R}{C_c} \tag{11}$$

the particle radius R is optionally corrected by the Cunningham correction factor,

$$C_c = 1 + \frac{\lambda}{R}\left(1.17 + 0.525 \cdot e^{-0.78\frac{R}{\lambda}}\right) \tag{12}$$

to account for the reduced drag of very tiny particles that may easily pass between the molecules of the surrounding fluid. The particle diffusivity D computes Brownian motion through

$$D = \frac{k_B T}{\gamma} \tag{13}$$

The simulation of Brownian motion can be disabled by the user. In that case the diffusivity is set to zero. External forces are the electrostatic force $Q\vec{E}$ and an additional force field $\vec{F}$ which is

$$\vec{F} = 0 \tag{14}$$

by default. When the user disables electrostatic effects, the electrostatic field $\vec{E}$ is set to zero.

Equations (11), (12), (13), and (14) can be changed through user defined functions, e.g., to include gravity or buoyancy forces in $\vec{F}$.

The used variables and their units are:

Symbol	Unit	Meaning
$\vec{v}$	m/s	particle velocity
$\vec{u}$	m/s	fluid velocity
γ	kg/s	friction coefficient
μ	kg/m·s	dynamic viscosity
R	m	particle radius
C_c	1	Cunningham correction factor
m	kg	particle mass
λ	m	mean free path

Symbol	Unit	Meaning
Q	C	particle charge
$\vec{E}$	V/m	electric field
k_B	J/K	Boltzmann constant
dW	$\sqrt{s}$	3D Wiener measure
T	K	temperature
D	m²/s	diffusivity
$\vec{F}$	N	external force

The electrostatic charges are assumed as constant given forces on the filter surface. A constant charge density ξ is assigned on all voxel walls. The electric field

$$E = -\nabla\Phi \tag{15}$$

is determined by solving the Poisson equation for the potential Φ (unit: V)

$$\Delta\Phi = -\frac{\xi}{\varepsilon_0}\int_{\partial G}\delta \tag{16}$$

where ξ is the surface charge density (unit: C/m²) and ε_0 = 8.854188E-12 F/m is the permittivity. Here, ∂G denotes the filter surface and δ is the Dirac distribution.

ξ is periodic in the tangential directions and should satisfy zero Dirichlet boundary conditions at $-\infty$ and $+\infty$ in flow direction. Numerically, infinity is replaced by $-Z_0$ and $nz + Z_0$ in the z-direction.

By construction, these boundaries lay away from the filter material and there is no conflict between singular forces on filter surfaces and these Dirichlet conditions. Due to the periodic boundary conditions, the potential feels a non-integrable amount of charges and tends to infinity in the filter as the Dirichlet boundary is moved away from the filter.

Thus, the potential ξ depends on the position where the Dirichlet condition is located. However, only the electric field E is needed to determine the movement of the particles in equation (10) and this remains almost unchanged from the location of the Dirichlet boundary as soon as this boundary is sufficiently far away from the filter material.

1.1.3 Particle Collisions

Particles moving through the fluid phase may collide with the filter material. In the simulation, such a collision happens when the particle surface (modeled as spherical particle with a given diameter) touches the surface of a grid cell (voxel) that does not allow particles to enter. Such a grid cell may belong to

a. the solid filter material,

b. solid particles deposited in previous batches,

c. or porous grid cells which do not allow additional particles to enter.

When a particle collides with the surface of such a grid cell, it can stick to the current position and become a deposited particle or it can bounce off and continue moving in another direction. This behavior is defined in the **Collision Model**. FilterDict offers three different built-in collision models and the possibility to define your own models.

Caught on first touch

Caught on first touch is the simplest collision model in FilterDict. With this model, the particle sticks to the filter media as soon as it touches the filter surface.

Hamaker model

In the **Hamaker** model, the velocity of the particle is compared to the adhesive forces. The particle is caught by the filter if its velocity when touching the structure is sufficiently small. The condition on the velocity is

$$v^2 < \frac{H}{4\pi\rho a_0 R^2} \tag{17}$$

where H is the **adhesion** (Hamaker constant), ρ is the particle density, a_0 is the adhesion distance or equilibrium spacing between the particle and the surface (assumed to be the typical value of $4 \cdot 10^{-10}\ \mathrm{m} = 0.4\ \mathrm{nm}$, and R is the particle radius. See Krupp, 1967.

Adhesion (Hamaker constant) and, another parameter, **restitution** need to be fitted in FilterDict.

The restitution parameter determines the amount of kinetic energy conserved during the collision. The restitution value ranges from 0 to 1. If the restitution parameter is set to 1, the

collision is perfectly elastic. Then, the particle is reflected with the same velocity it had before the collision.

For restitution values smaller than one, energy is absorbed by the collision and the particle slows down. For example, a 0.5 restitution value means that a particle loses half of its velocity in the collision and gets 50% slower.

$$\text{Restitution Coefficient} = \frac{v_2}{v_1} \tag{18}$$

Sieving

In the **Sieving** model, the particles never stick to the filter, but a particle is caught by the filter if it does not move anymore and touches the filter at two different points. The restitution parameter is used in the same way as in the Hamaker model.

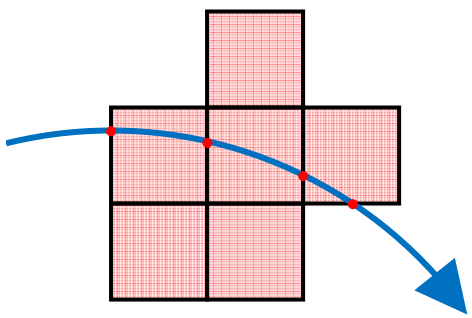
User-Defined Collision Models

Additional collision models can be defined in user defined functions[242].

1.1.4 Particle Capture

Often, the grid resolution is not able to resolve all features of the filter material. In this case, not only fluid voxels or solid voxels exist, but some voxels model a porous media themselves. Small particles (with diameter much smaller than the voxel length) may pass through the voxel, but they may also be captured inside of the voxel.

In such unresolved simulations, the filter media is modeled as homogeneous material with a defined permeability. The interaction between particles and the surface cannot be modeled explicitly. Instead, the pass-through model describes the probability that a particle passes through the filter media. Whether or not a particle is filtered depends on the pass-through probability of the voxels it passes through and the distance it covers. Particles can be trapped anywhere on their way through the medium.



Travel path of a particle inside the filter media and its crossings through the grid cells (voxels).

Assume that a filter media with a thickness L has a rating β (at a given flow velocity). Then, 1 in β particles pass through. Thus, the probability for a particle to travel a distance l without getting captured is

$$p(l) = \beta^{-\frac{l}{L}} \tag{19}$$

So, the probability that the particle travels through the whole media is

$$p(L) = \frac{1}{\beta} \tag{20}$$

and the passing probability of passing the media n times is

$$p(nL) = p(L)^n \tag{21}$$

The following models are available in GeoDict:

All Particles Pass

In this model, the pass-through probability is 100% for all particles.

Impassable

In this model, particles cannot enter the porous material. The material behaves like a solid material for the particles, but it is permeable for the fluid.

Constant Efficiency

In this model, a fixed efficiency E is given for each particle type. The β-rating is computed from the given efficiency value through

$$E = 1 - \frac{1}{\beta} \tag{22}$$

Thus, the pass though probability is given through

$$p(l) = \beta^{-\frac{l}{L}} = (1 - E)^{\frac{l}{L}} \tag{23}$$

Velocity Dependent Efficiency

In this model, the efficiency depends on the particle velocity v. Thus,

$$p(l) = (1 - E(v))^{\frac{l}{L}} \tag{24}$$

The velocity dependent efficiency must be given in form of a table.

Clogging

In the clogging model, it is assumed that the efficiency E depends on the amount of deposited dust. A voxel representing a porous filter medium has in the clean state an initial fractional filtration efficiency of E_0. During the simulation, dust may deposit inside the voxel, increasing the deposited dust volume fraction inside of the voxel from an initial fraction $f = 0$ to an assumed maximal particle packing density $f_{\max}$. When $f_{\max}$ is reached, the filtration efficiency reaches 100%:

$$E(f) = \begin{cases} E_0 + \dfrac{f}{f_{\max}}(1 - E_0) & \text{for } f < f_{\max} \\ 1 & \text{for } f \geq f_{\max} \end{cases} \tag{25}$$

The implemented **Clogging** model assumes a linear dependency between efficiency and local dust volume fraction f. The passing probability is then again computed as

$$p(l) = (1 - E(f))^{\frac{l}{L}} \tag{26}$$

Constant Absorption Rate

On the macro scale, a macroscopic equation for the concentration of particles, the Convection Diffusion-Reaction equation, can be adopted for particle transport (see Iliev et al., 2003[257], Dedering et al., 2008[257]):

$$\frac{\partial C}{\partial t} + \vec{u}\nabla C - D\Delta C = -\frac{\partial M}{\partial t} \tag{27}$$

where C is the concentration of particles, $\vec{u}$ is the velocity, D is the diffusivity coefficient, M is the mass of the captured particles in the filter medium, and $\partial M/\partial t$ means the rate of deposition. When diffusion is negligible, the time variation of the concentration is solely governed by the absorption rate, and the 1D case is considered. Then, Equation (27)[13] can be simplified to

$$u\frac{\partial C}{\partial x} = -\frac{\partial M}{\partial t} \tag{28}$$

Assuming that the amount of the deposited particles is small compared to the pore space of the filter medium, the mass of the deposited particles is considered to be proportional to the concentration of particles (see Brown et al., 1999) [257].

$$\frac{\partial M}{\partial t} = \alpha C \tag{29}$$

where α is the constant absorption rate. The analytic solution can then be derived for Equations (28) [14] and (29) [14]:

$$C(x,t) = C_{\text{in}} e^{-\frac{\alpha}{U}x} \tag{30}$$

$$M(x,t) = \alpha t \cdot C_{\text{in}} e^{-\frac{\alpha}{U}x} \tag{31}$$

Then the β-rating is given as

$$\beta(t) = \frac{C_{\text{in}}}{C(L,t)} = e^{\frac{\alpha}{U}L} \tag{32}$$

and the pass-through probability is

$$p(l) = e^{-l\frac{\alpha}{U}} \tag{33}$$

depending on the constant absorption rate α and the velocity.

User Defined

Other models to define the passing probability $p(l)$ of a particle can be implemented by the user and added as a user defined function [247].

1.1.5 Filter Clogging

During the filtration process, particles are deposited on the structure. This is reflected in the simulation by changing the 3D structure model after each batch. The new structure consists of the filter material plus the deposited particles. Large particles, that are resolved in the voxel

grid, will simply add solid grid cells to the structure. However, many particles are comparable in size to the voxel length, or even smaller. In this case, they will not completely fill a grid cell. Therefore, the simulation has to track the (dust filled) **Volume Fractions** of each voxel and model flow and filtration properties according to the state of the voxel.

FilterDict offers different models to handle this:

- In **Resolved** simulations, all voxels are treated as either empty or solid for flow and filtration, and the volume fractions are simply used to decide whether a voxel is empty or solid.
- In **Unresolved** or **Mixed** resolution simulations, partially filled voxels are modeled as porous voxels for flow and filtration. A **Flow Resistivity** model determines how the flow resistivity depends on the locally deposited volume fraction.

In the examples below, the initial structure consists of empty pores (white in the illustrations below) and full solid (red) voxels. During the filtration process, particles (gray) are caught in the filter. In Filter Element 125 simulations, the original structure can already contain porous voxels.

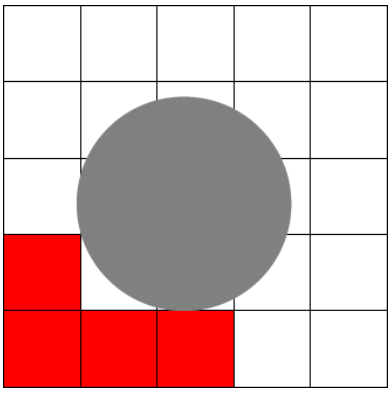
Resolved Particles Simulation Mode

This model is only suitable if all particles diameters are larger than the voxel length.

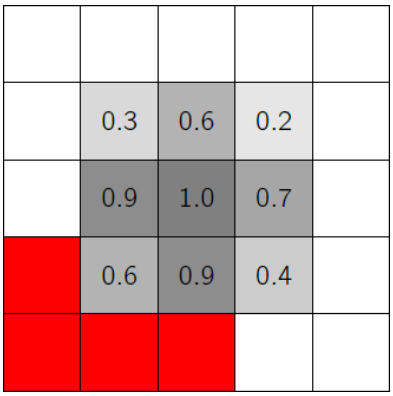
After each batch, the voxels with caught particles are analyzed. If the volume fraction in a voxel is higher than the **Volume Fraction Threshold** f_{solid} (default is 0.5), the voxel is set to solid. Otherwise, the voxel is considered as pore.

The volume fractions from succeeding batches are added. Under certain circumstances, the volume fraction of a voxel might exceed 1.

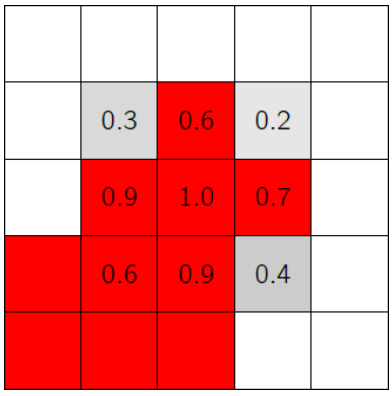
First Batch



1. A large particle deposits on some solid voxels shown in red.

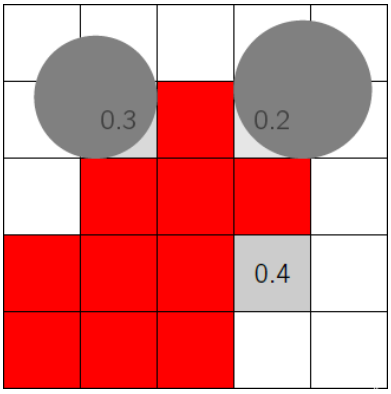


2. Volume fractions f shown for the corresponding voxels.

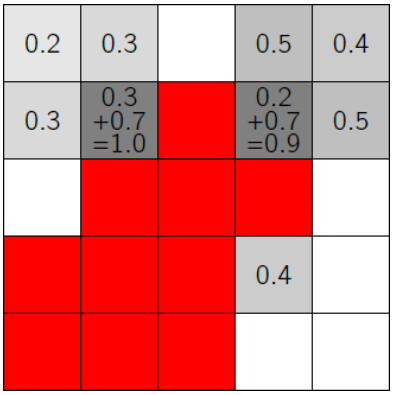


3. New solid voxels appear when f is larger than the Volume Fraction Threshold (here 0.5).

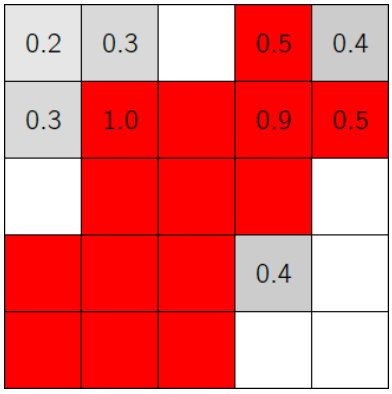
Subsequent Batch



4. Additional large particles deposit in the following batch.



5. Volume fractions f shown for the corresponding voxels.



6. New solid voxels appear when f is larger than the Volume Fraction threshold (here 0.5).

Mixed/Unresolved Particles Simulation Mode

This model is suitable if there are at least some particle diameters smaller than the voxel length, or if all particle diameters are smaller than the voxel length.

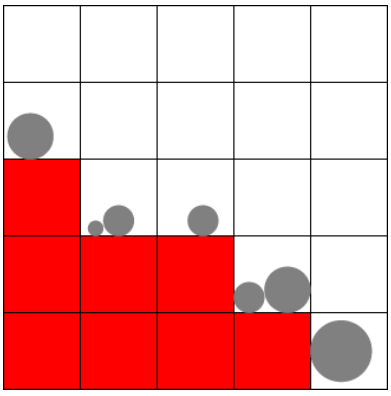
In unresolved simulations, voxels may be porous. These voxels may contain captured particles or be made of porous filter material. Porous voxels have a flow resistivity depending on their volume fraction. Particles can only enter a voxel when its volume fraction is smaller than the **Maximal Particle Packing Density** f_{packing}.

Therefore, the Navier-Stokes-Brinkman equations (5) 6 and (6) 6 have to be used to determine the flow through the resulting structure. The flow resistivity $\sigma = \mu/\kappa$ is given as a function $\sigma(f)$, where f denotes the local solid volume fraction inside a voxel. Different options are available for the user to describe this function in the Constituent Materials tab 66.

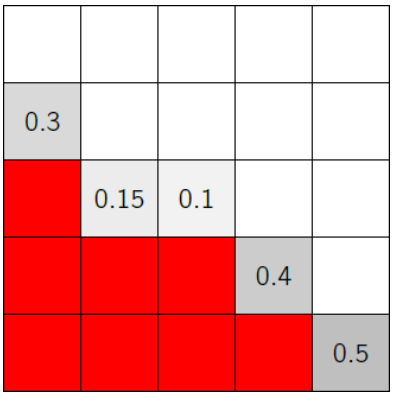
The figure below shows the process of small particles clogging the structure in mixed/unresolved simulation mode. Solid voxels (marked in red) will block flow and particles, as in the resolved mode. Partially filled voxels (shown in gray) will have an increased flow resistivity, and voxels with $f \geq f_{\text{packing}}$ (shown in striped red) will block particles from entering.

Also in the mixed/unresolved simulation mode, some voxels may turn into solids, if $f \geq f_{solid}$. f_{solid} is always larger than f_{packing} and by default set to $f_{\text{solid}} = 1.0$. Thus, when a voxel is completely filled with dust particles, no flow is possible any more.

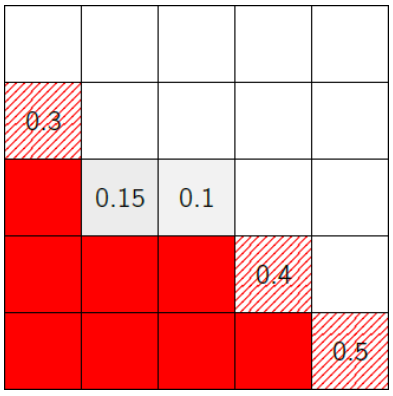
First Batch



1. Particles deposit on some solid voxels shown in red.

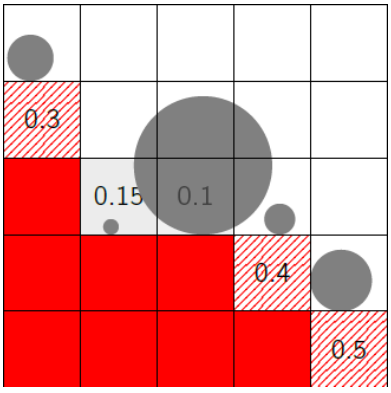


2. Volume fractions f shown for the corresponding voxels.

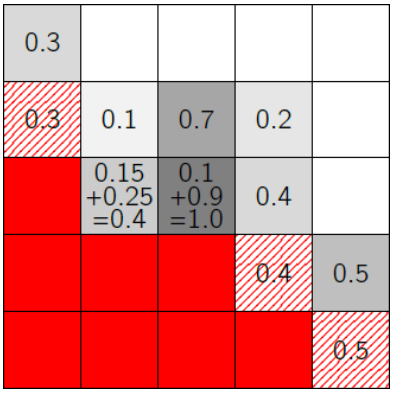


3. Voxels where f is larger than the Maximal Particle Packing Density (here 0.3) become impassable for particles (shown in striped red).

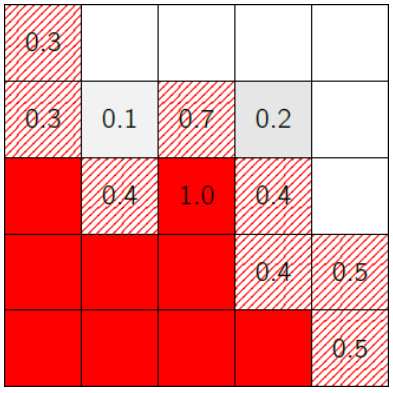
Subsequent Batch



4. Additional particles deposit in the following batch.



5. Volume fractions f shown for the corresponding voxels.



3. Voxels where f is larger than the Maximal Particle Packing Density (here 0.3) become impassable for particles (shown in striped red). A new solid voxel appears where f is larger than f_{solid} (here 1.0).

During one batch, interaction between particles is not simulated. Thus, it is possible that particles from one batch overlap each other. A measure for this effect is the **Volume Loss**. A high value indicates that the number of overlapping particles is high. To avoid this, decrease the number of particles per batch.

1.2 Filter Media

On the **Filter Media** scale, simulate the filtration process using a 3D model of a micro-scale structure. Micro-scale 3D structure models resolve every fiber or grain of the filter material. You may generate them with GeoDict's structure generation modules, or import and segment 3D-image data with ImportGeo-Vol.

You can select two types of simulations from the pull-down menu below **Filter Media**: **Filter Efficiency** and **Filter Lifetime**.

Commands available for filter media analysis

- Filter Efficiency 18
  Determine the initial efficiency of the filter and calculate the most penetrating particle size.
- Filter Lifetime 57
  A Filter Lifetime simulation includes clogging of the filter and computes pressure drop and deposited dust over time. In a Single Pass simulation, fluids pass through the filter only once and are not recirculated. In a Multi Pass simulation, fluids move in a circuit through the system, and particle size distribution and concentration in front of the filter change over time.

Note! For all **Filter Media** simulations, it is essential that the structure model contains enough free space as inflow region (Z- direction) and outflow region (Z+ direction).

For efficiency simulations, the inflow region should be larger than the largest particle diameter. For filter lifetime simulations, the inflow region should be larger than the largest particle diameter plus the expected height of the filter cake.

The outflow region should be long enough for the flow to adjust to the boundary conditions. For faster flows, larger outflow regions are required.

To add inflow and outflow regions, use the ProcessGeo-Embed command. Enter the appropriate number of voxels, in the Z- and Z+ boxes, and click Embed. Then, switch back to FilterDict.

1.2.1 Filter Efficiency

Filter Efficiency determines the theoretical efficiency of the clean filter, and it is used to calculate the most penetrating particle size (MPPS) of the filter or to determine the filter class. In contrast to modeling a Filter Lifetime experiment, only one step (batch) is simulated, and every particle is tracked on the clean filter.

The Filter Efficiency command for filter media

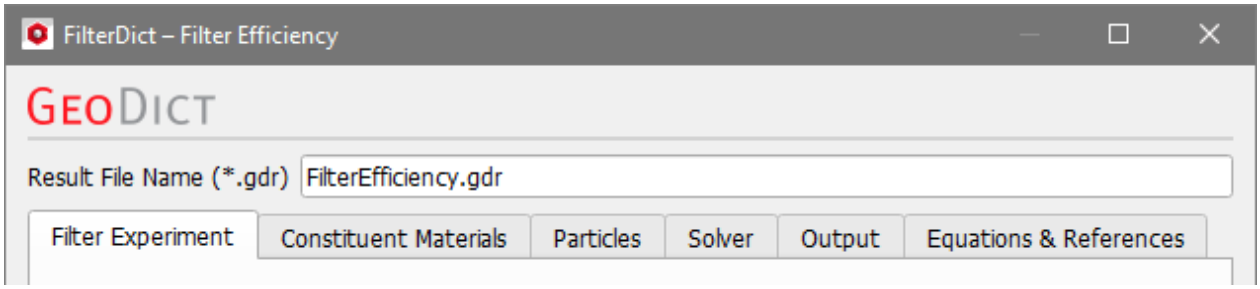
- Options 19
  How to set the input parameters for the Filter Efficiency command.
- Results 52
  How to understand the computed results.

The Filter Efficiency command for filter media

- Particle Visualization 57
  How to visualize the computed results.

Options

Select **Filter Efficiency** from the pull-down menu and click the **Option**'s **Edit...** button to open a dialog to enter or edit the simulation options. Choose a **Result File Name** for the result file (*.gdr) fitting to your current project.



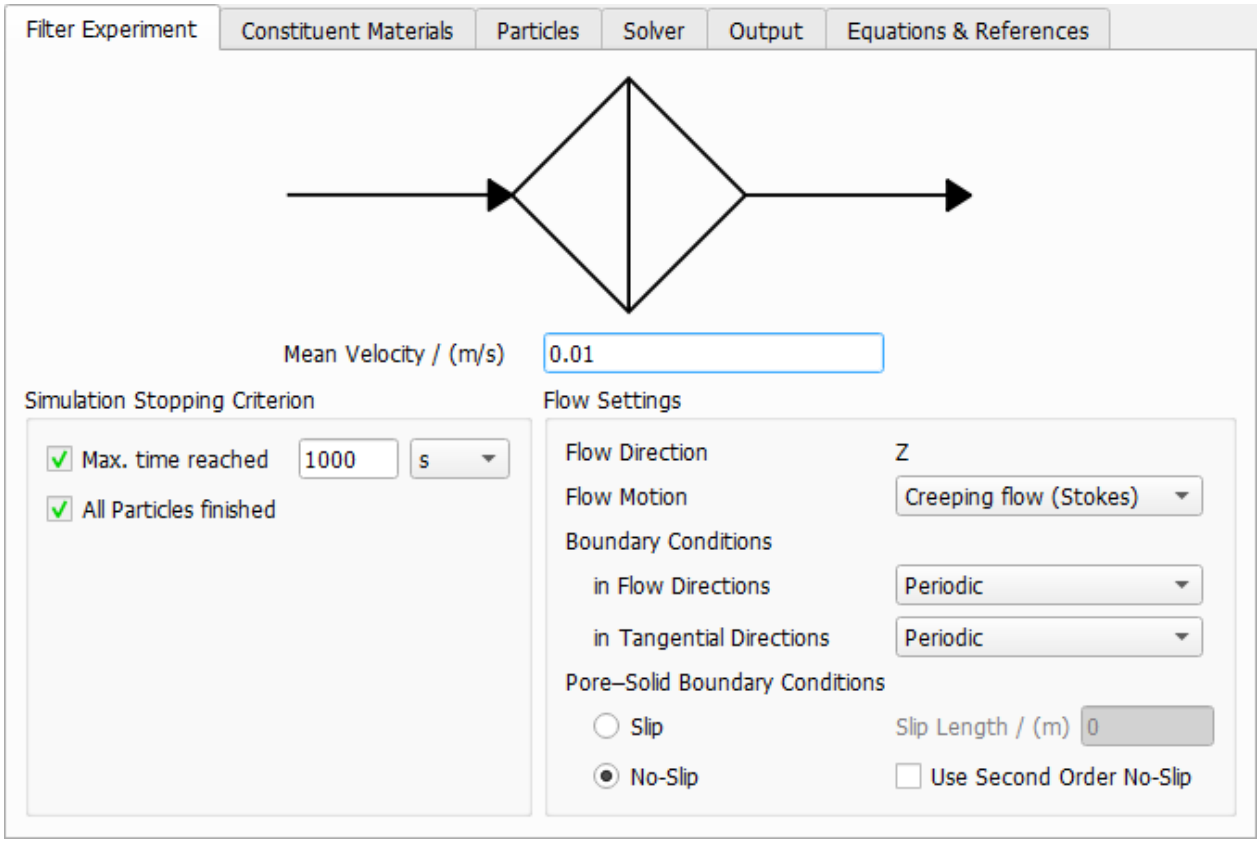
The filter efficiency simulation parameters are organized under the tabs: **Filter Experiment, Constituent Materials**, **Particles**, **Solver**, and **Output**. The **Equations & References** tab gives a summary of the used equations and symbols which are explained in more detail in this user guide.

Tabs of the Filter Efficiency dialog

- Filter Experiment 19
  Set the mean flow velocity and the boundary conditions.
- Constituent Materials 21
  Define fluid type and temperature. Set material IDs and parameters for porous layers.
- Particles 23
  Select particle sizes and their start positions. Set all parameters that govern the particle motion and define collision and adhesion models.
- Solver 37
  Select the flow solver, set the stopping criteria and define the computational resources.
- Output 51
  Choose which data will be stored for visualization.

Filter Experiment

Under the **Filter Experiment** tab, enter the **Mean Velocity** of the fluid passing through the filter.



Simulation Stopping Criterion

Each particle trajectory is calculated by solving the equations (10) to (14). Particle movement starts at $t = 0$ and ends when the particle leaves the domain unfiltered, is filtered by the structure, or t reaches the end of the given **Max. time reached**. This value is used to prevent exceptionally slow particles from increasing the simulation time and does not have a direct connection to the time of a measurement.

Particles which are still moving when the maximal time is reached are marked as **time out** particles.

Flow Settings

Important! The flow is always directed in Z-direction in FilterDict.

Select the **Flow Motion** regime to be **Creeping flow (Stokes)** or **Fast flow (Navier-Stokes)** based on the flow velocity and the Reynolds number.

Select the **Boundary Conditions in Flow Directions** as **Periodic** or **VinPout**, which refers to velocity at inlet (Vin) and pressure at outlet (Pout). In **Tangential Directions**, select **Periodic** or **Symmetric** boundary conditions. For more information on these boundary conditions, please consult the FlowDict user guide.

The choices made for the flow boundary conditions in tangential directions also determine what happens when a particle reaches the domain boundary in one of the tangential directions. With **Periodic** boundary conditions, the particle will leave the domain and reappear on the opposite site. With **Symmetric** boundary conditions, a particle will be reflected at the domain boundary.

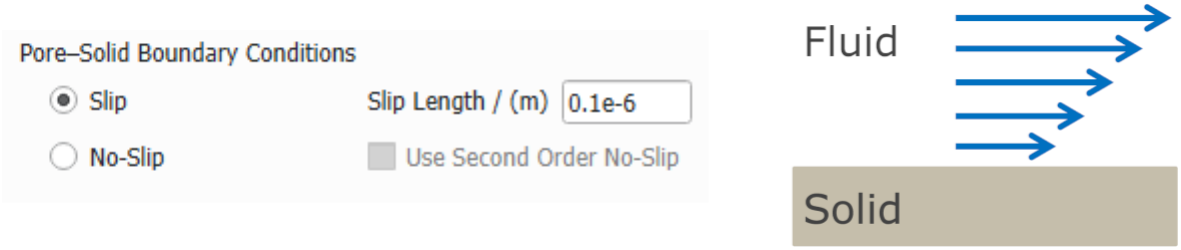
Pore-Solid Boundary Conditions

The **Slip Length** allows to include sliding effects in the flow simulation. Typically, the slip length is of the same order of magnitude as the mean free path of the fluid (which is, e.g., 68 nm for air at ambient conditions) and can be neglected if this is much smaller than the voxel length. It becomes relevant when simulating air flow around nanofibers.

The default setting **No-Slip** corresponds to a flow velocity of zero along the structure.

Use Second Order No-Slip sets the tangential velocity to zero at the center of the voxel surfaces and uses a second order approximation of the velocity field. The first order setting, which is used when the box is left unchecked uses a linear approximation. **Second Order No-Slip** will increase precision when there are narrow pores in the structure.

Slip with a non-zero **Slip Length** simulates the sliding of the fluid along the solid, increasing the fluid mean flux and thus, the permeability of the structure. This option might be used when it is realistic for a given physical material. Currently, the same slip length value must be set for all materials in the structure.



Constituent Materials

Temperature

The **Temperature** for the filtration process is selectable in Kelvin (K), Celsius (°C), and Fahrenheit (°F). The chosen temperature may change the density and viscosity values of the fluid during the simulation, if those are defined to be temperature-dependent in the material database. Furthermore, if the Brownian motion of the particles is included in the simulation, it

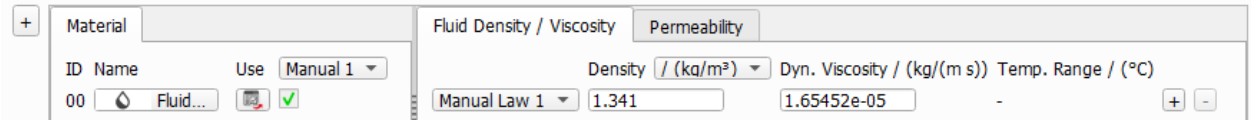
determines the strength of the Brownian motion through equation (13) 8.

Fluid Density and Viscosity

FilterDict uses single-phase flow simulations. It is not possible to run a flow simulation if two fluids are present in the structure. If multiple fluids are present in the structure, you may click **Set all Fluids to...** and select one fluid for the simulation.

The selected fluid is also displayed in the drop-down menu in the **Material** subtab behind **Use**. The density and the dynamic viscosity of this fluid are shown under the **Fluid Density/Viscosity** subtab.

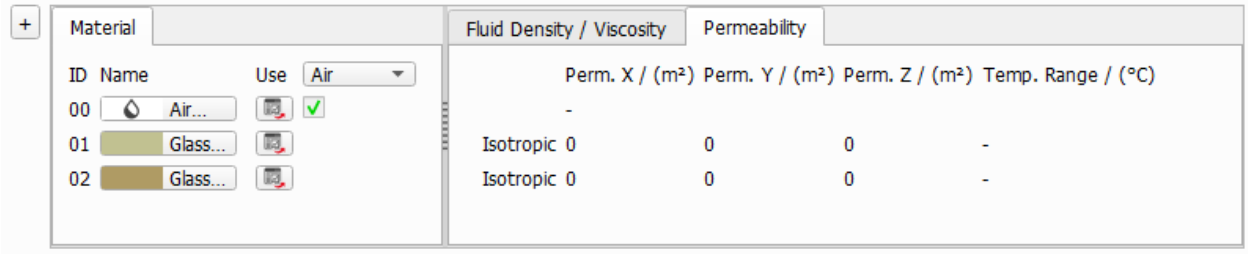
If a fluid from the GeoDict Material Database is used, the values for **Density** and **Dynamic Viscosity** are taken from the database and may depend on the given **Temperature** value. Select **Manual** as fluid material to enter parameter values manually.



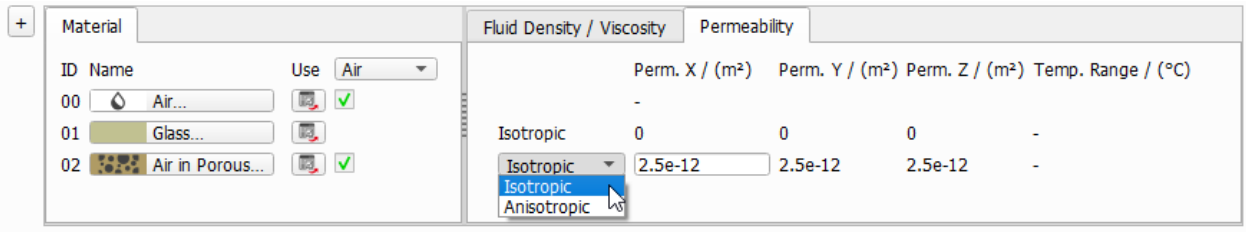
To add materials to the database, use the **Open & Edit Material Database** button. More information on editing, expanding, and using the **GeoDict Material Database** is available in the Material Database user guide.

Filter Media Permeability

Under the **Permeability** subtab, enter the permeability of porous materials present in the structure. As long as there are only fluid and solid materials present in the structure, no parameters are needed here:



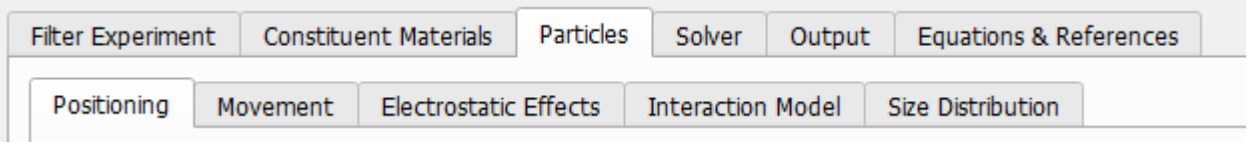
For porous materials, choose if the material is **Isotropic** or **Anisotropic**. In the isotropic case, the permeability is the same in all space directions. In the anisotropic case, enter a different permeability for each spacial direction.



You can select and change a material by clicking on the material button.

Particles

Under the **Particles** tab, five sub-tabs are available for **Positioning**, **Movement**, **Electrostatic Effects**, **Interaction Model**, and **Size Distribution**.



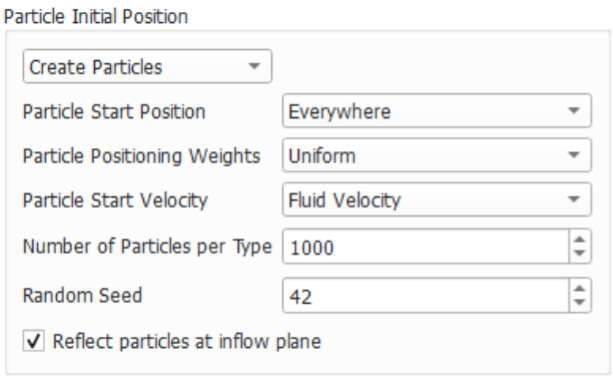
Tabs of the Particles tab

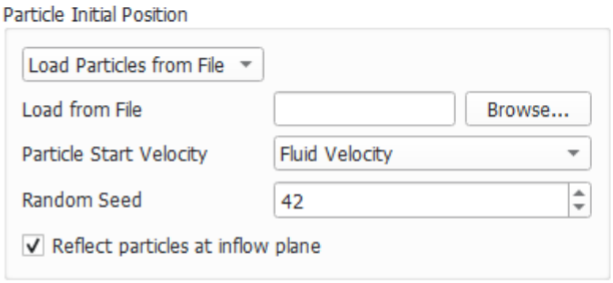
- Positioning 23
  Define the start positions of the simulated particles.
- Movement 28
  Select the parameters modelling the motion of the particle.
- Electrostatic Effects 29
  Switch electrostatic effects on or off.
- Interaction Model 32
  Define collision and adhesion models for particles-material collisons.
- Size Distribution 36
  Define the particle size distribution and all particle size dependent parameters.

Positioning

Particle Initial Position

First, select whether the particles should be created according to the given size distribution or loaded from a file. If particles are loaded from a file, number and size of particles are as given in the file.





Select **Create Particles** and determine the **Particle Start Position**, the **Particle Positioning Weights**, the **Particle Start Velocity**, and the **Number of Particles per Type** to define the initial set of particles.

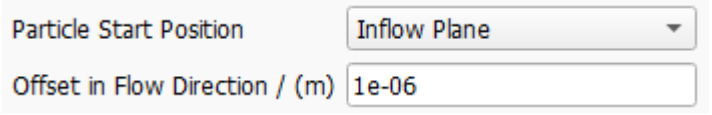
Select **Load Particles from File** and **Browse...** for a .gpp file to load.

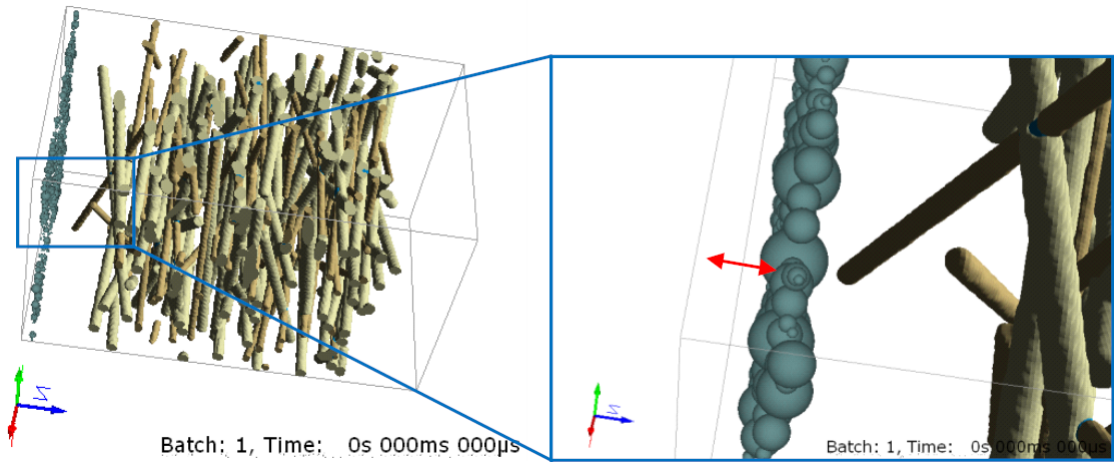
Particle Start Position

Select one of the following **Particle Start Position** types. The default start position is the Inflow Plane.

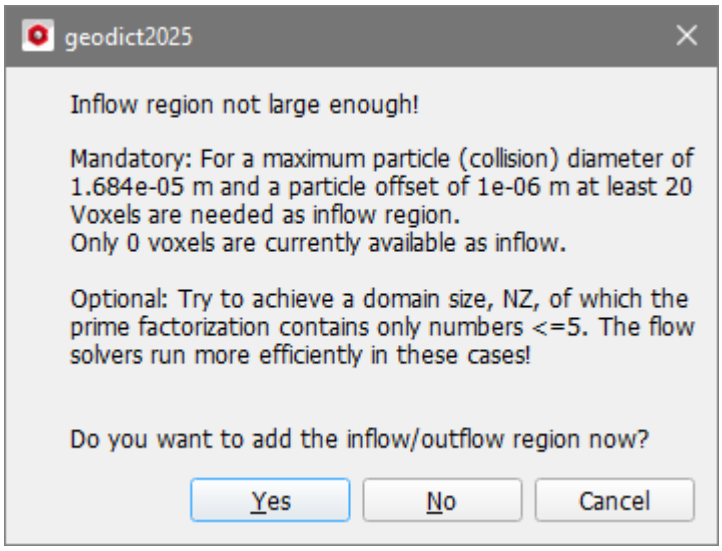
Inflow Plane

If **Inflow Plane** is chosen as **Particle Start Position**, the **Offset in Flow Direction** defines the initial distance of the particles to the boundary of the inflow region.



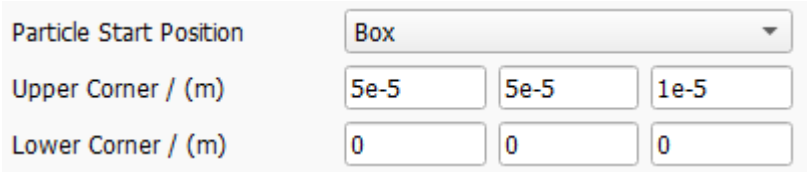


If the inflow region is too small and the particles cannot be placed in the inlet without intersecting the solid material, a warning appears after clicking **Run** to start the simulation in the **FilterDict** section.



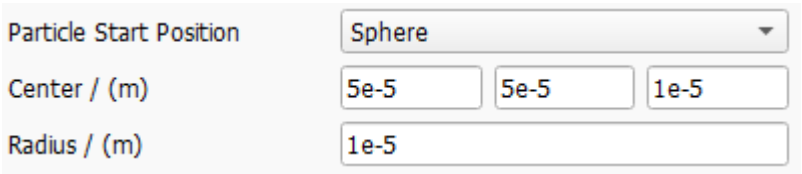
Box

If **Box** is chosen as **Particle Start Position**, two opposite corners of a rectangular box must be defined. All particle start positions will then be inside of this box.



Sphere

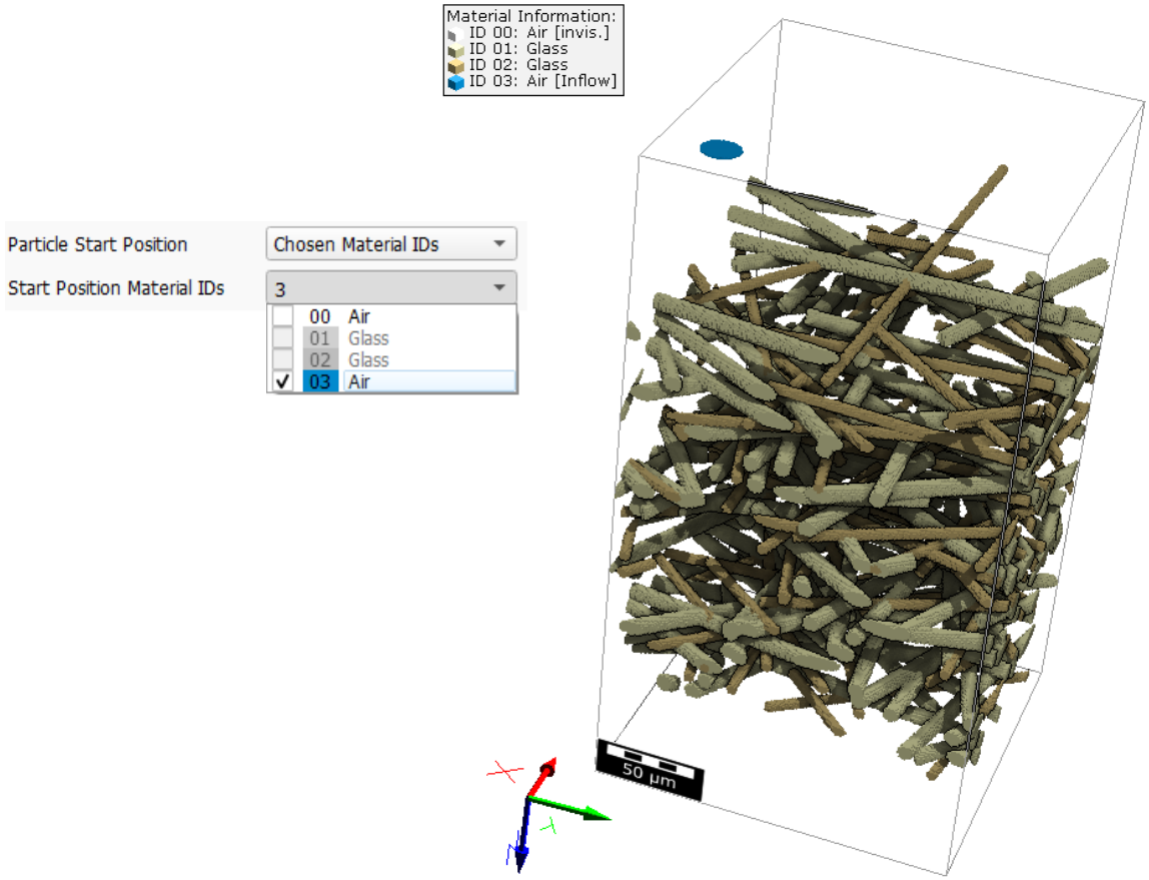
If **Sphere** is chosen as **Particle Start Position**, the center and radius of the sphere must be defined. All particle start positions will then be inside of this sphere.



Chosen Material IDs

An arbitrary location can be set by selecting **Chosen Material IDs** as **Particle Start Position**.

For example, by using GadGeo, various objects can be added into the inflow area. Those objects can be assigned to the same fluid material as the surrounding pore space, so they only serve as markers of a certain area (see picture below).



If this area ID is now chosen as **Start Position Material ID**, all particles start in this area. It is possible to mark several areas in this way and/or to choose several IDs as starting areas.

Everywhere

If **Everywhere** is chosen as **Particle Start Position**, particles may start everywhere in the pore space of the structure.

Particle Positioning Weights

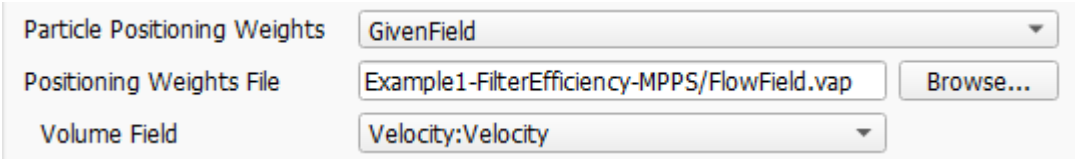
Particle start positions are scattered randomly inside the defined **Particle Start Position** area or volume.

By default, those positions are uniformly distributed. Be aware that a spatially **Uniform** placement in the inlet will lead to an increased particle concentration in areas with low fluid velocity, e.g., in proximity to a surface or a walls.

If **Velocity** is selected, the starting probability is based on the local fluid velocity. Selecting this option will lead to a uniform particle concentration in the inflow.

If **Distance** is selected, the starting probabiliy is based on the distance from the next solid object. This will lead to higher particle numbers in the core flow, while there will be less particles starting next to the surface.

With **GivenField**, it is possible to load an arbitrary 3D field of scalar values to define the starting probability. For example, it is possible to load the flow field of a previous computation and select the velocity or one of its components.



Particle Start Velocity

Select either **Fluid Velocity** (the default option, which was used until GeoDict 2025) or **Given Velocity** as the **Particle Start Velocity**.

With the first option, particles begin moving with the fluid velocity at their start position.

With the second option, all particles start with the same user-defined velocity.

Number of Particles per Type

The **Number of Particles per Type** defines how many particles are simulated for each given particle type/particle size. Larger numbers lead to more accurate results, but also to longer runtimes of the simulation. For each particle size, FilterDict tracks the paths of this number of particles and calculates the fractional filter efficiencies from the results.

To determine the overall efficiency from the fractional efficiencies, the percentages entered in the particle size distribution tab 36 are taken into account.

Random Seed

Particles to be filtered are placed randomly in the area according to the selected **Particle Start Position**. The parameter **Random Seed** controls the underlying random number generator. The same random seed produces identical results, whereas results with different random seeds are similar but not identical.

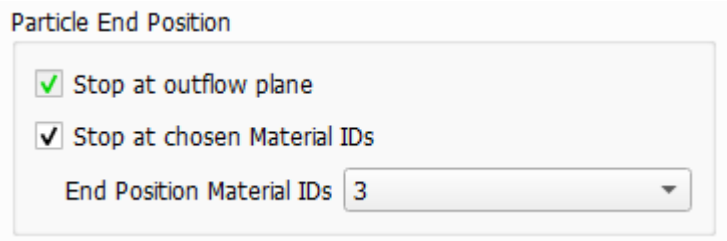
If the results generated by different random seeds are considerably different, increase the **Number of Particles per Type**.

Reflect Particles at Inflow Plane

If diffusion through Brownian motion is simulated, some particles might diffuse against the flow direction, reach the inflow region and exit the domain. Check **Reflect particles at inflow plane** to avoid the exit of such particles.

Particle End Position

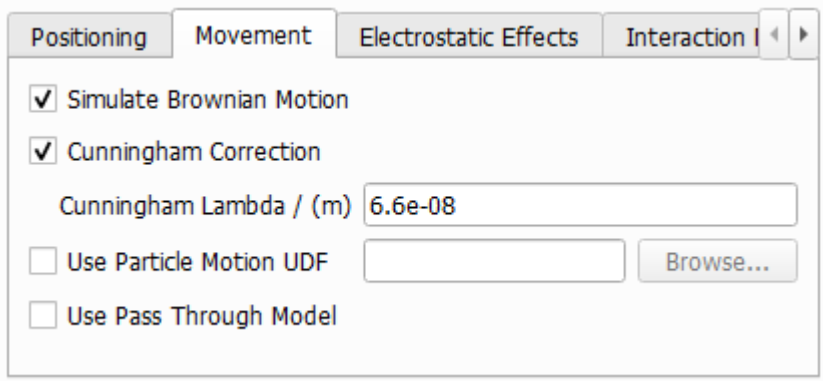
Particles reaching the outflow plane are always considered as unfiltered.



Similarly to the definition of the particle start position, an additional area can be marked. Particles that arrive in this area will also stop moving and be counted as unfiltered.

Movement

The parameters in the **Movement** panel define the simulation variables for the particle tracking.



Brownian Motion and Cunningham Correction

Checking **Simulate Brownian Motion** activates the simulation of random motion in particle movement. If this option is unchecked, the simulation is run with diffusivity $D = 0$.

If **Cunningham Correction** is checked, equation (12) 8 is used to calculate C_c. Otherwise, $C_c = 0$ is used.

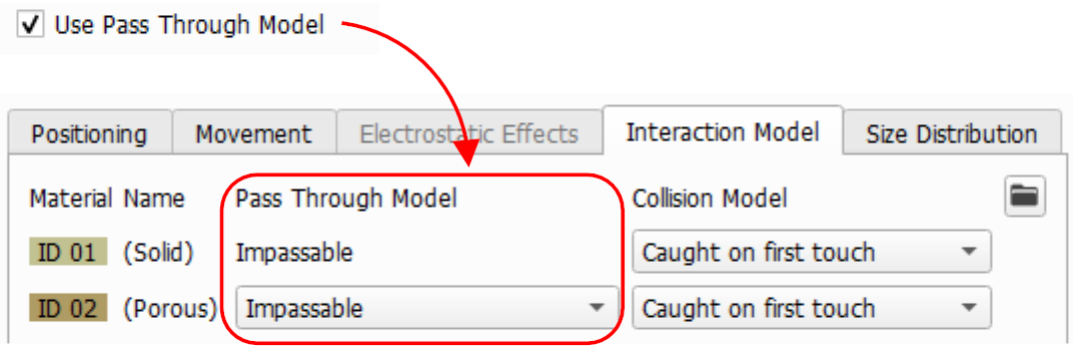
The **Cunningham Lambda** value is the mean free path λ used in equation (12) 8. Make sure, that you use the Cunningham Correction with the default mean free path only in air filtration. Inside of water or oil, the mean free path is much smaller, and there is no need to take the Cunningham Correction into account.

Use Particle Motion UDF

Check **Use Particle Motion UDF** to replace equations (11) 8, (12) 8, (13) 8, and (14) 8 with your own definitions. Refer to the Particle Motion UDF 239 page for details on how to modify and compile user-defined functions.

Use Pass Through Model

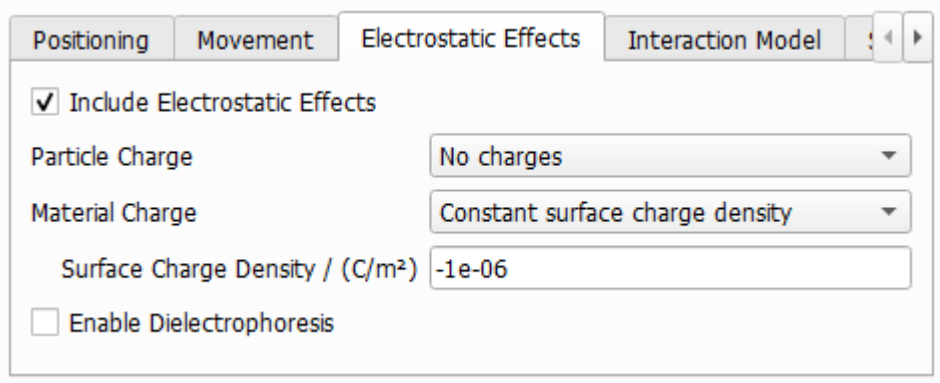
If porous materials are present in the current structure, you must check the **Use Pass-Through Model** box. This option enables particle movement through porous layers and particle deposition in such layers. When this option is switched on, the **Pass Through Model** column becomes available in the Interaction Model 32 subtab.



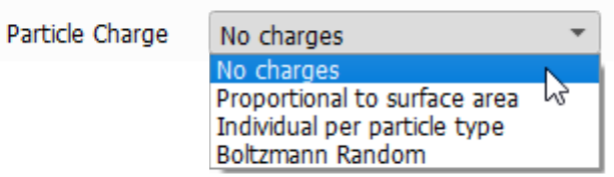
Electrostatic Effects

Important! This tab is only available if there are no porous materials present in the 3D structure.

Check **Include Electrostatic Effects** to simulate electrostatic effects between filter media and particles in the fluid.

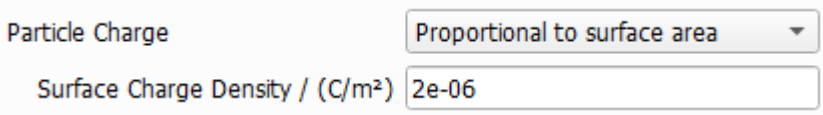


Particle Charge



If electrostatic effects are negligible, set **Particle Charge** to **No charges**.

Select **Proportional to surface area** and enter a **Surface Charge Density** to set the value of Q in equation (10) to the product of the entered **Surface Charge Density** and the particle surface area.



Select **Individual per particle type** to enter an individual particle charge for each particle type or size. In this case Q in equation (10) is the entered particle charge. Enter the desired values in the **Particle Charge [C]** column that is automatically added to the table under the Size Distribution subtab.

If **Boltzmann Random** is chosen, particle charges follow a normal distribution with mean value 0 and standard deviation

$$\sigma = \frac{\sqrt{2\pi d \varepsilon_0 k_B T}}{e} \tag{34}$$

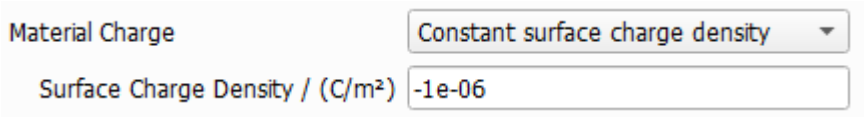
where d denotes the particle diameter, ε_0 the dielectric constant, k_B the Boltzmann constant, T the temperature and e the electron charge.

Note! Electrical charges appear only as multiples of the elementary electric charge. Since GeoDict 2025, the **Boltzmann Random** distribution is therefore discretized into elementary electric charges. For large particles, this approaches a normal distribution of charge values, whereas small particles are mostly uncharged or carry just a single

positive or negative charge.

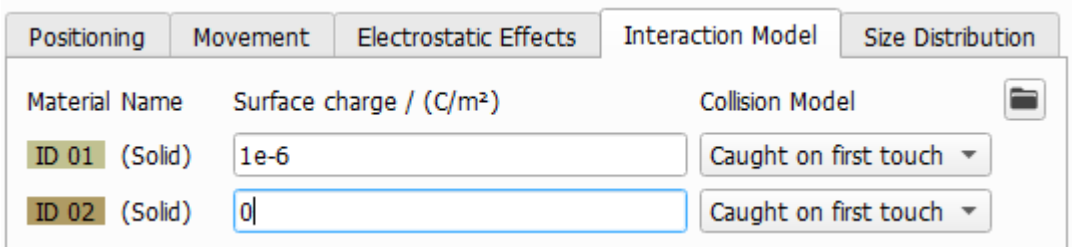
Material Charge

Select **Constant surface charge density** and set the **Surface Charge Density** (C/m^2), which corresponds to ξ in equation (16) [9].



This charge density is applied to all solid material surfaces.

Select **Individual per material ID** to set different surface charge densities for different material IDs. If selected, an additional column **Surface Charge (C/m^2)** appears on the Interaction Model tab. Enter the surface charge density of each material ID there.



Since the charges are given "per surface area," it is especially important to consider how surface area is taken into account in the simulation. GeoDict uses a voxel grid. Simply applying the surface charge to each voxel surface would overestimate the overall charge because the sum of all voxel surfaces overestimates the true surface area of the material. hus, the surface charge density is corrected using the ratio of the material's computed surface area (see the **Estimate Surface Area** command in the MatDict user guide) to the sum of the voxel surfaces.

Note! Until GeoDict 2022, particles and filter were assumed to have opposite charges. Entering positive surface charge values for filter and particles lead to the filter attracting the particles. This was unintuitive and did not reflect the correct physics.

Therefore, since GeoDict 2023, it is no longer assumed that particles and filter have opposite charges. Entering opposite sign charge values (e.g., 5e-7 C/m^2 as surface charge and -6e-7 C/m^2 as material charge) for filter and particles leads to the filter attracting the particles. In this case, the filter efficiency is enhanced. Entering same sign charge values leads to the filter repulsing the particles and a lower filtration efficiency.

Enable Dielectrophoresis

Dielectrophoresis is a phenomenon in which a force $\vec{F}$ is exerted on a particle when it is

subjected to a non-uniform electric field $\vec{E}$. This force does not require the particle to be charged. All particles exhibit dielectrophoretic activity in the presence of electric fields.

When dielectrophoresis is enabled, the force $\vec{F}$ in equation (10) is computed as

$$\vec{F} = 2\pi R^3 \varepsilon_m \left(\frac{\varepsilon_p - \varepsilon_m}{\varepsilon_p + 2\varepsilon_m} \right) \nabla |\vec{E}|^2 \quad (35)$$

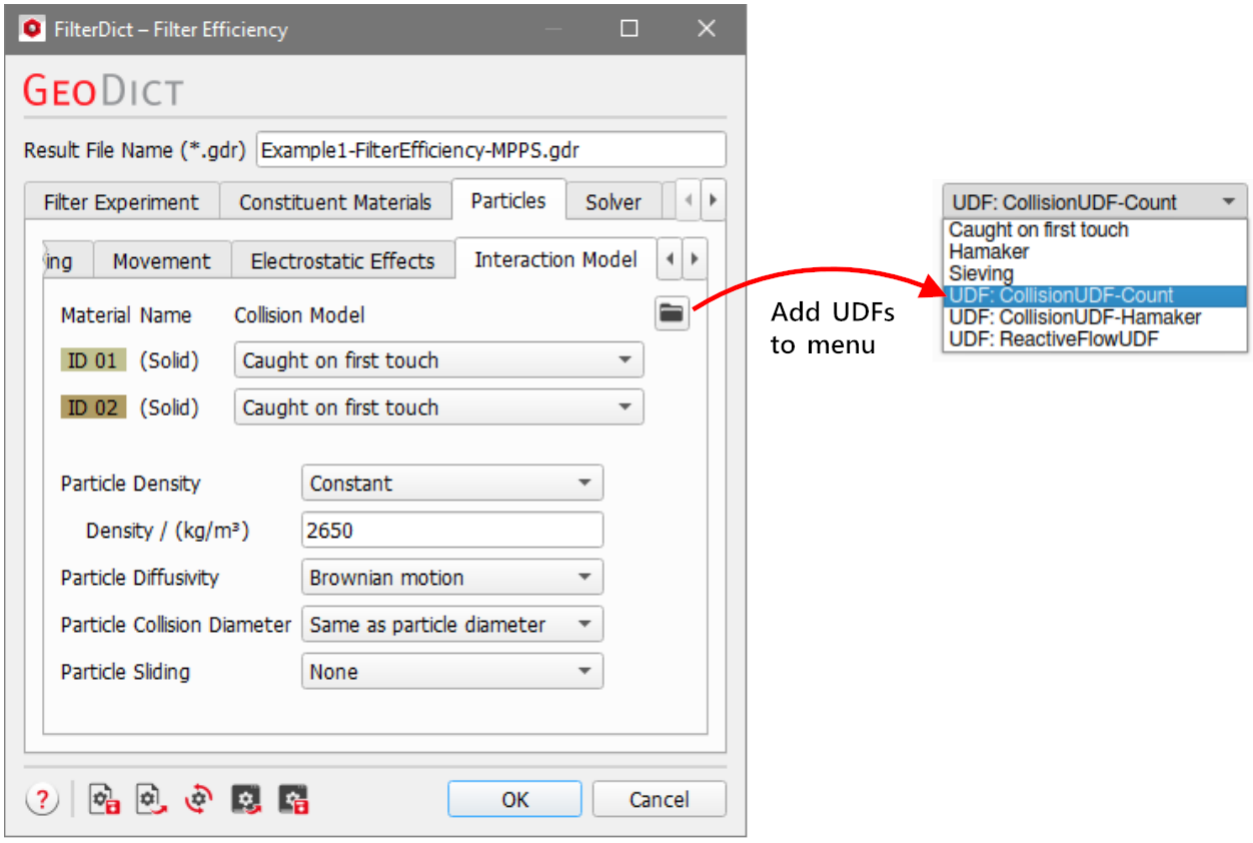
Here, ε_p is the relative permittivity of the particles, ε_m is the relative permittivity of the fluid, and R is the particle radius. $\nabla|\vec{E}|$ denotes the gradient of the electric field and dictates the direction of the dielectric force.

Switch on this effect by marking the check box, and enter the relative permittivity values ε_m and ε_p in the dialog:

Interaction Model

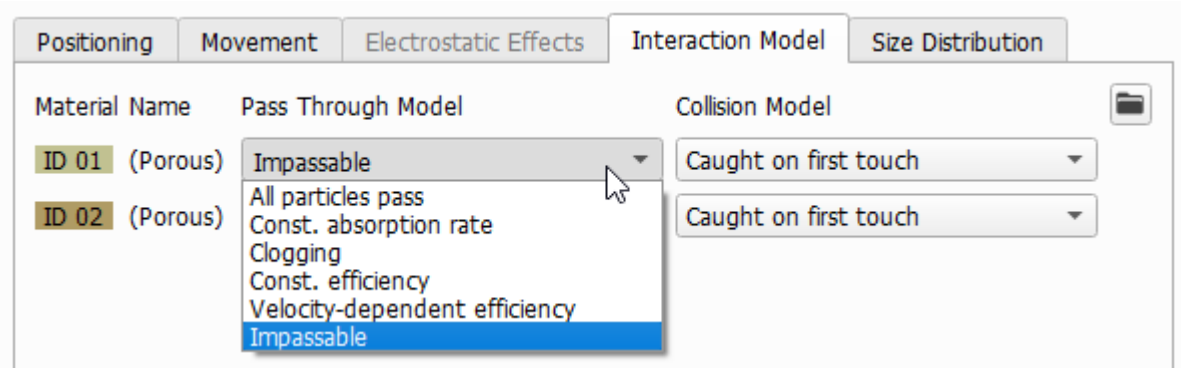
The choice of parameters under the **Interaction Model** subtab affects the entries and columns shown in the table under the Size Distribution subtab.

All materials of the current structure are listed here. For all solid materials in the structure, a collision model must be chosen among **Caught on first touch**, **Hamaker**, or **Sieving** (see the theoretical background for particle collision models). Additionally, all compiled Collision UDFs that are stored in the users UDF folder, appear here as additional choices in the pull-down menus. Additional UDF folders can be selected with the button.



New columns appear in the table under the **Size Distribution** subtab when **Hamaker** (columns Restitution and Adhesion) or **Sieving** (column Restitution) are chosen.

If **Use Pass Through Model** is selected on the Movement tab 28, a **Pass Through Model** must be selected for all porous materials.

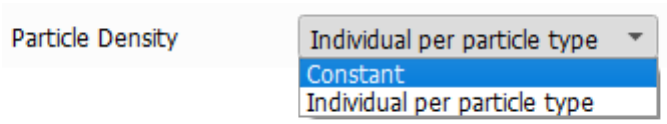


See the Theoretical Background 11 for more information on the selectable pass through models **All Particles Pass, Constant Absorption Rate, Clogging, Constant Efficiency**, and **Velocity-Dependent Efficiency**.

Particle Density

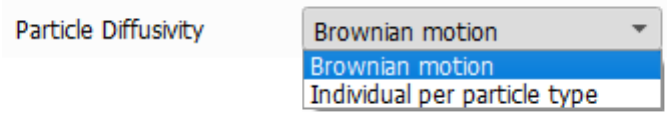
If **Individual per particle type** densities are chosen, a new column called **Particle Density** is added to the table under the Size Distribution tab, and an individual density can be entered for each particle type or size. Otherwise, the **Particle Density** entered here is used for all

particle sizes and types.



Particle Diffusivity

If **Individual per particle type** is chosen, a new column called **Difusivity in Pore** is added to the table under the Size Distribution tab, and an individual diffusivity can be entered for each particle type or size.



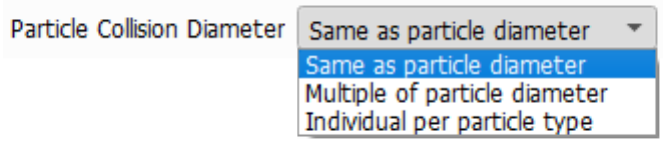
For **Brownian Motion**, the particle diffusivity is computed for each particle type according to equation (13) 8.

If **Simulate Brownian Motion** is switched off on the Movement tab, only **None** is selectable here.

If **Use Particle Motion UDF** is selected on the Movement tab, the choices made here have no effect, as the formula implemented in the UDF will be used.

Particle Collision Diameter

The particle **Diameter**, as given in the table under the Size Distribution tab, is used when tracking the particle in the flow field. It determines the particle radius R used in equations (10) 8, (11) 8 and (12) 8 to determine the mass and the friction.

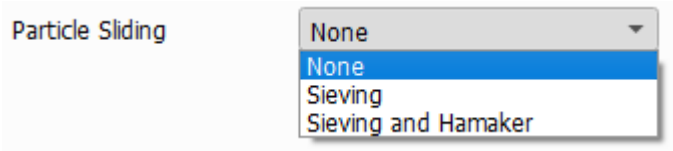


The **Particle Collision Diameter** is used to check if the particle collides with the filter structure and should usually be set to **Same as particle diameter**. With this selection, the particle is consistently modeled as a sphere.

However, real dust particles may exhibit different, possibly less compact, shapes. Thus, they may have a higher probability of colliding with the structure than a spherical particles of the same mass. By setting the **Particle Collision Diameter** to a different value than the **Diameter**, you can model this effect. It is possible to set a fixed **Diameter factor** for all particles sizes by choosing **Multiple of particle diameter** or to select **Individual per particle type** and enter the collision diameter in the Size Distribution tab for all particle types.

Particle Sliding

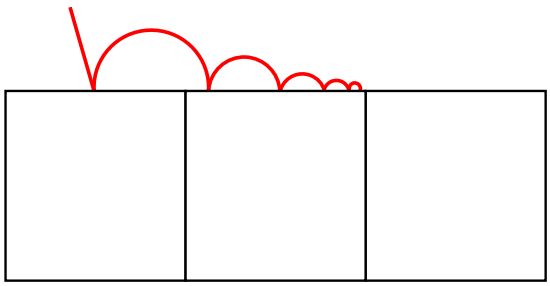
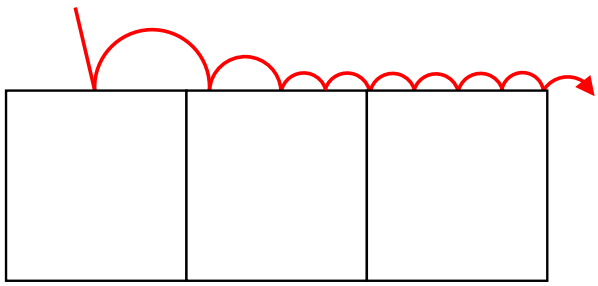
Particles that collide with the surface loose some of their energy if the restitution factor is smaller than 1. In certain pore geometries, the flow may move a particle very close to a pore surface. This causes many consecutive hits, and repeatedly applying the restitution factor will stop the particle's motion. This behavior is often undesired, and it may overestimate the filter efficiency when combining a **Sieving** collision model with a relatively low restitution value.



When **Sieving** is selected in the drop-down menu, all **Sieving** particle-wall collisions are modified, when **Sieving and Hamaker** is selected, all **Sieving** particle-wall collisions are modified and all **Hamaker** particle-wall collisions are modified. **Caught on First Touch** particle-wall collisions are not modified.

Enabling **Particle Sliding** for the **Sieving** or **Hamaker** collision models enlarges the tangential restitution factor when a particle hits a surface in a location that lies close to the last surface collision.

The first bounce on the wall is always treated the same, independent whether particle sliding is active or not: A particle arrives with a certain velocity, loses energy on impact, and bounces back again. The strength of the bounce-back is controlled with the given **Restitution**. When particle sliding is active, further impacts near the first hit are treated differently: Only the momentum perpendicular to the wall is reduced. The velocity along the wall is not reduced further, so in this direction the restitution is equal to 1. This means that particle movement along the wall is no longer slowed down. The normal restitution remains unchanged. In effect, the particle will slide along the surface.

Without particle sliding:	With particle sliding
• energy lost at every collision • particle caught on surface	• energy conserved after some hits • particle moves along surface
	

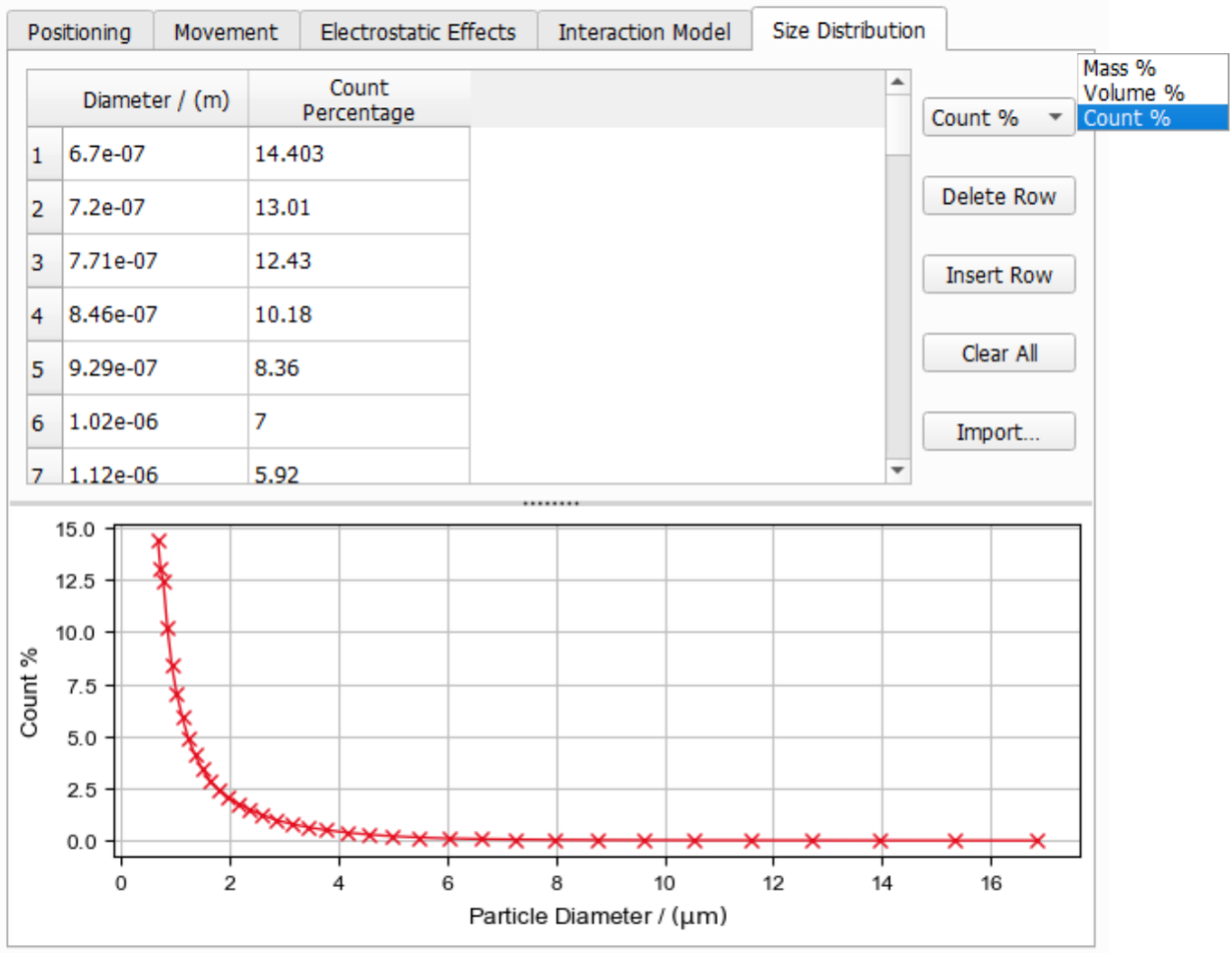
In setups where particles are sieved by filter materials having pores of well-defined sizes (e.g., in meshes, nets or weaves), selecting particle sliding has a great influence on the computed filter efficiency and pressure drop and it is recommended to select this option.

In setups where particles are mainly caught by adhesion or where the filter material consists of irregular pores of many different sizes (e.g., nonwovens), the choice of the particle sliding

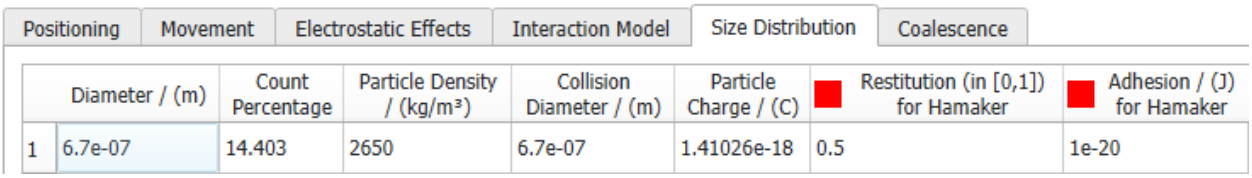
model has little to no effect on the simulation results.

Size Distribution

The **Size Distribution** tab contains a table with the particle size distribution. The table consists of at least two columns. The first column lists the **Diameter** of the particles. The second column contains the particle distribution. You can select if the percentages describe **Mass %**, **Volume %**, or **Count %** through the upper right pull-down menu.



Additional columns appear depending on the choices made under the Interaction Model subtab:



These columns are:

- **Restitution** for materials with Sieving collision model.
- **Restitution** and **Adhesion** for Hamaker material.
- **Particle Density** column for Individual per particle type densities.

- **Particle Charge** column for Individual per particle type charges.
- **Collision Diameter** column for Individual per particle type collision diameters.

For user-defined collision models, the number of material parameters to be entered in the table is specified in the UDF.

By clicking on a column heading, all values in the selected column can be set to the value in the first row. The entries in the table do not need to be sorted.

Click **Delete Row** to delete the currently selected row. Click **Insert Row** to insert a row below the currently selected row. Click **Clear All** to remove all rows.

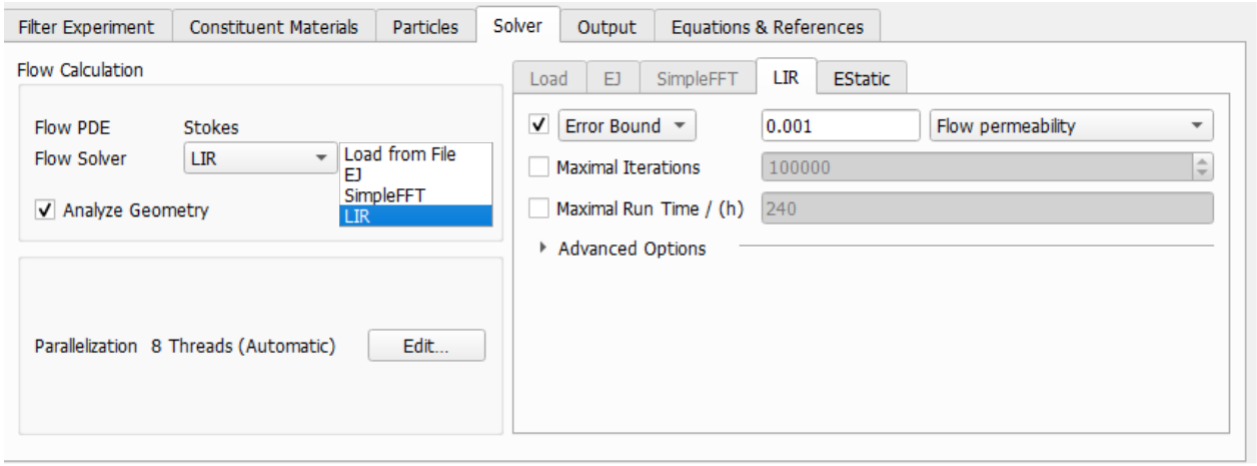
Click **Import...** to import data from an ASCII text file. Some example particle distributions are provided with GeoDict and can be found in the installation folder (Program Files/Math2Market GmbH/GeoDict2026/FilterDict).

More convenient than importing is often to simply copy & paste entries from an Excel table into GeoDict, which is supported by the size distribution table. To copy a table from Excel, first clear the size distribution table by clicking **Clear All**. Then, select the cells in Excel and press Ctrl+C. Then, click into the upper left cell in the size distribution table and press Ctrl+V. This works also if more than two columns are present.

If the entered percentages do not add up to 100%, a warning message pops up when closing the dialog. You can select to normalize the values automatically or to return to the dialog and adjust the percentages manually.

Solver

The **Solver** tab contains on the left the **Flow Calculation** and the **Parallelization** panel and on the right individual tabs for every solver. Tabs corresponding to unused solvers are grayed out and not selectable.



In the **Flow Calculation** panel, select the flow solver used to compute the flow field or choose to load a flow field already precomputed in a previous computation. Depending on the **Flow Motion** type chosen on the Filter Experiment tab 19, either the Stokes or Navier-Stokes equations are solved. Three solvers **EJ** (only for Stokes flow), **SimpleFFT** and **LIR** are available.

Analyze Geometry

If this option is chosen, a geometrical analysis at first determines whether a through path exists and removes unconnected pore components from the computational grid. This may speed up the flow computations but requires time for the geometrical analysis.

Parallelization

Depending on the purchased license, the simulation process can be parallelized. The **Parallelization Options** dialog opens when clicking the **Edit...** button, to choose between **Sequential**, **Parallel (Shared Memory)**, **Automatic Maximum of Threads** and **Cluster**. For details on how to set up and run parallel computations, refer to the High Performance Computations user guide

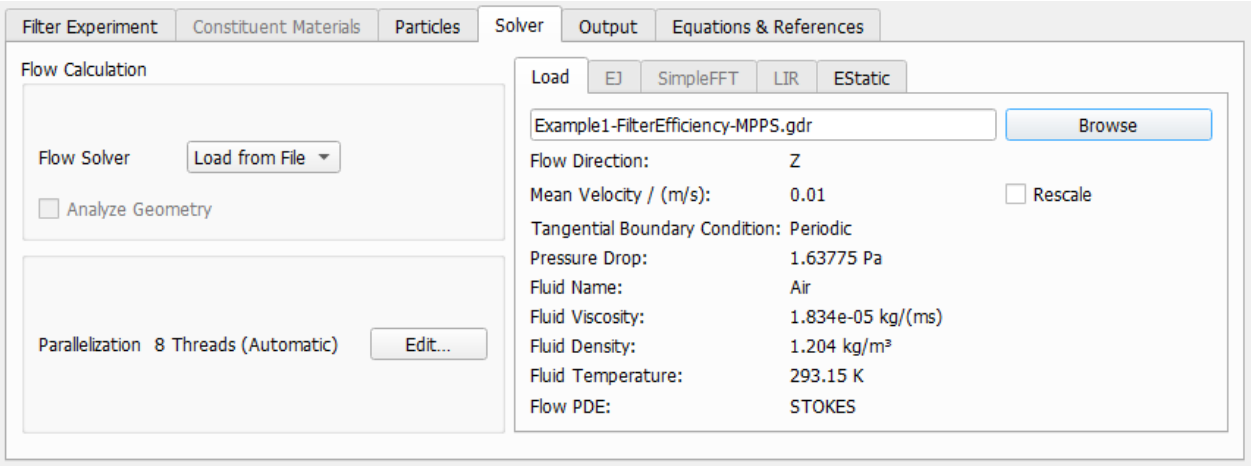
The chosen parallelization settings apply for all steps of the simulation. Per default, the **Automatic Maximum of Threads** option is used, and the exact number of threads depends on the number of cores on the current computer and the number of licensed processes.

Load from File

Select **Load From File** in the **Flow Calculation** panel to use a precomputed flow field. The flow field may originate from a FlowDict simulation or from a FilterDict simulation. When calculated with FlowDict, you have to make sure to:

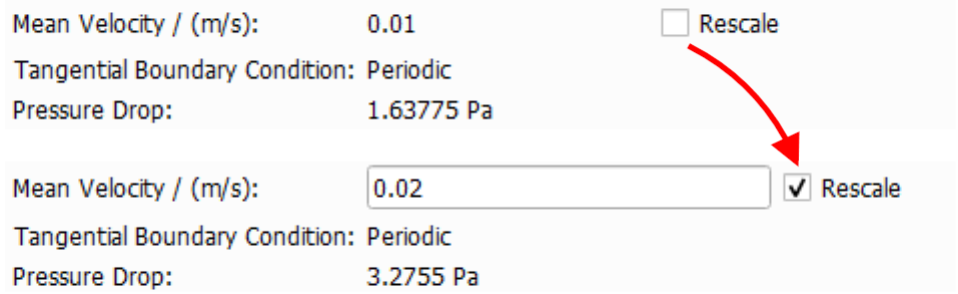
1. Add inflow and outflow regions to the media model before running FilterDict.
2. Compute the flow in the Z-direction. You may either use the Stokes or Navier-Stokes equations depending on the flow velocity for this.
3. Set the accuracy at least one order of magnitude higher than the default in FlowDict (e.g., Error Bound = 0.001 instead of 0.01). The stopping criteria depend on global values, whereas the particle movement depends on the local flow field, which is subjected to larger deviations.

Under the **Load** tab, click **Browse** and choose a result file (GDR) to open. If no flow field is selected, a warning message appears (No flow field chosen) when trying to run the simulation.



The physical properties of the fluid used in the flow simulation are entered automatically when loading the flow GDR file and appear listed under the **Load** tab.

- The **Flow direction: Z** is the main direction of the flow.
- The **Mean Velocity** (m/s) of the flow field, the used **Tangential Boundary Conditions** and the **Pressure Drop** are taken from the loaded flow simulation. If the flow result was obtained by solving the Stokes equation, the solution is linear and can be rescaled. When checking **Rescale**, the value that was entered automatically from the flow result file can be manually changed. Changing the mean velocity automatically rescales the pressure drop, too.



- The fluid parameters **Fluid Viscosity** (kg/m s) , **Fluid Density** (kg/m^3), and **Fluid Temperature** (K) are the physical values of the fluid used by the solver for the calculation of the flow field.

The fluid settings must be the same as the fluid that was used in the previously run flow simulation, loaded through the GDR file. Therefore, the **Constituent Materials** tab is inactive when **Load from File** is selected. Also, the **Tangential Boundary Conditions** selected in the **Filter Experiment** tab must match with those used in the flow simulation.

EJ

The **EJ** tab becomes editable when you select the EJ solver in the **Flow Calculation** panel.

Simulation Stopping Criterion: Tolerance or Residual

The **Tolerance** stopping criterion looks for stagnation of the method when the process becomes stationary, i.e., the improvement in the pressure drop value becomes

extremely small from iteration to iteration.

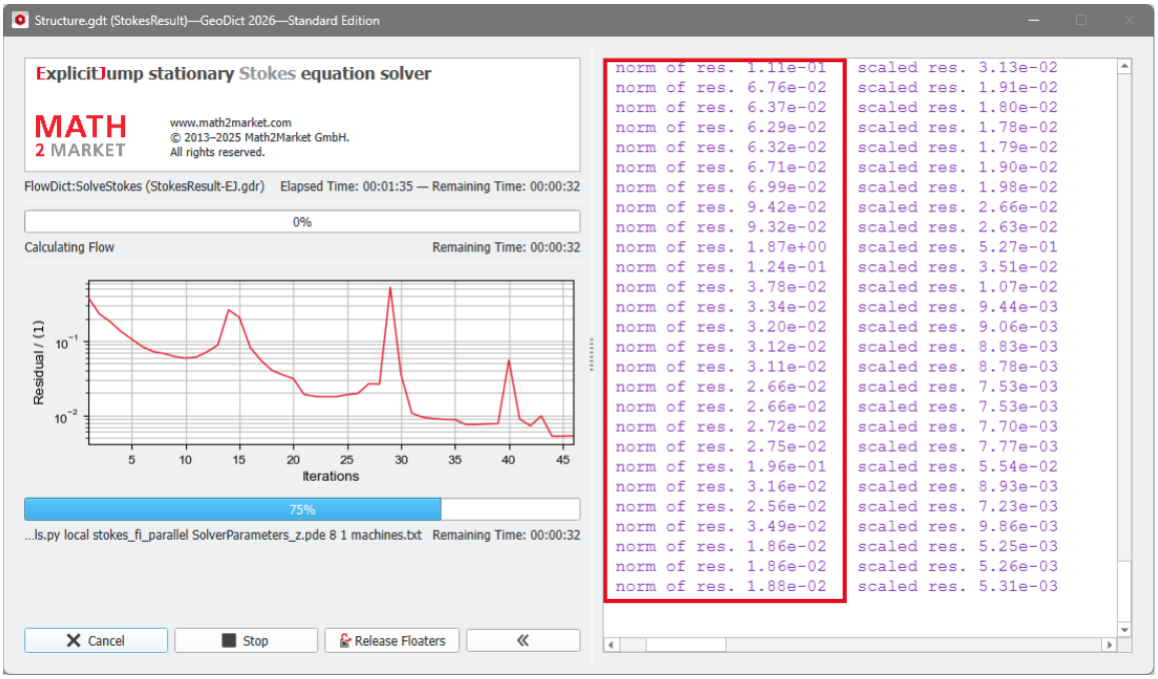
In each iteration, the solver checks for the current computed value x_c against the values x_{c-i} of the last 100 iterations if there are any changes.

$$\max_{i\in[1,100]} \frac{x_c - x_{c-i}}{x_{c-i}} < \text{tolerance} \tag{36}$$

The computation is stopped if the maximal relative change is smaller than the value entered for **Tolerance**. When there is doubt about the quality of the solution, decrease the **Tolerance** value by a factor of ten. The drawback of this criterion is that the solver sometimes might stop too early in case of slow convergence rate or at local extrema of oscillatory convergence curves.

When the **Residual** stopping criterion is used, the iteration is stopped if the solution satisfies the equation up to the required accuracy.

By setting the stopping criterion to **Residual**, the computations terminate as soon as the relative norm drops below the selected residual threshold. The relative norm of the Schur Complement residual is computed and displayed in the console window during the calculations. The console is visible by clicking the double arrow button on the lower right corner in the progress dialog.



The recommendation to choose **Tolerance** or **Residual** for the **EJ** solver is based on the structure's porosity. Both give similar results for highly porous structures. For dense structures, if using the Schur Complement **Residual**, the relative norm of the residual may be small even though the correct pressure drop has not been reached. So, when in doubt, use the **Tolerance** criteria – the default option.

Be aware that the default accuracy in FilterDict is one order of magnitude higher than in FlowDict. The stopping criteria depend on global values, whereas the particle movement depends on the local flow field. It is therefore necessary to make sure that not only the overall pressure drop is computed up to the given accuracy, but that also the local values are sufficiently accurate. This typically achieved by setting a higher accuracy as stopping criterion.

Simulation Stopping Criterion: Max. Iterations and Max. Run Time

Use the **Maximum Iterations** value or the **Maximum Run Time (h)** stopping criteria causes the solver to stop if the maximal number of iterations and/or the maximal runtime (in hours) is exceeded.

When the solver stops because one of these criteria has been reached, no guarantee on the quality of solution can be given. In this case, a warning is printed into the report. The following possibilities might help:

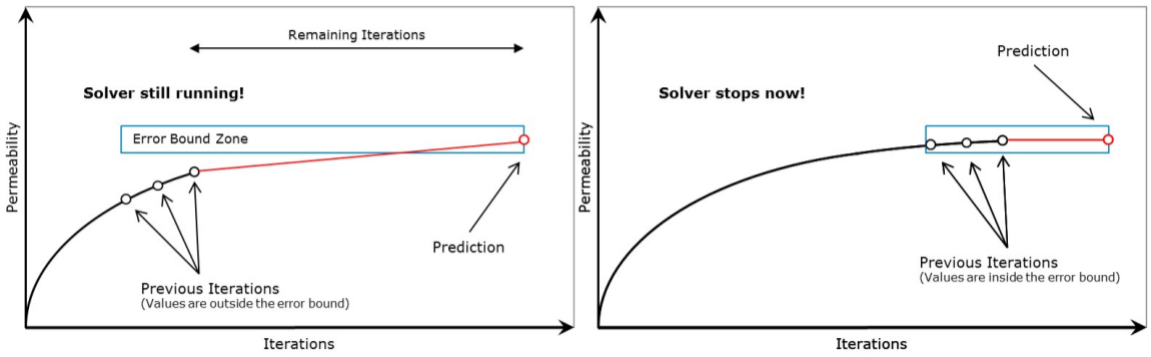
- Check the corresponding .log file to see how large the residual values and the pressure drop values are. If the pressure drop values are in reasonable range and they do not have changed during the last iterations, you may decide to use the current result.
- Double-check the structure and parameter values. Unphysical parameters or too rough resolution of the structure (leading, e.g., to artificial unconnected components) can cause an iterative solver to fail.

SimpleFFT

The **SimpleFFT** tab becomes editable when you select the SimpleFFT solver in the **Flow Calculation** panel.

Simulation Stopping Criterion: Error Bound, Tolerance and Residual

The stopping criterion **Error Bound** uses the result of previous iterations, and predicts the final solution based on linear and quadratic extrapolation. The solver stops if the relative difference between computed and predicted solution is smaller than the specified error bound. The stopping criterion recognizes oscillations in the convergence behavior and prevents premature stopping at local minima or maxima. A damped convergence curve is fit through the oscillating curve and the solver stops then regarding the damped convergence curve.



If the Krylov method (under LIR - Advanced Options) is activated, the definition of **Error Bound** is somewhat different. Here, no prediction is made, instead the continuity between neighboring cells and the conservation of mass is checked. The maximal value is normalized by the mean flow and if this value is smaller than the **Error Bound** the simulation stops.

The **Tolerance** stopping criterion looks for stagnation of the method when the process becomes stationary, i.e., the improvement in the pressure drop value becomes extremely small from iteration to iteration.

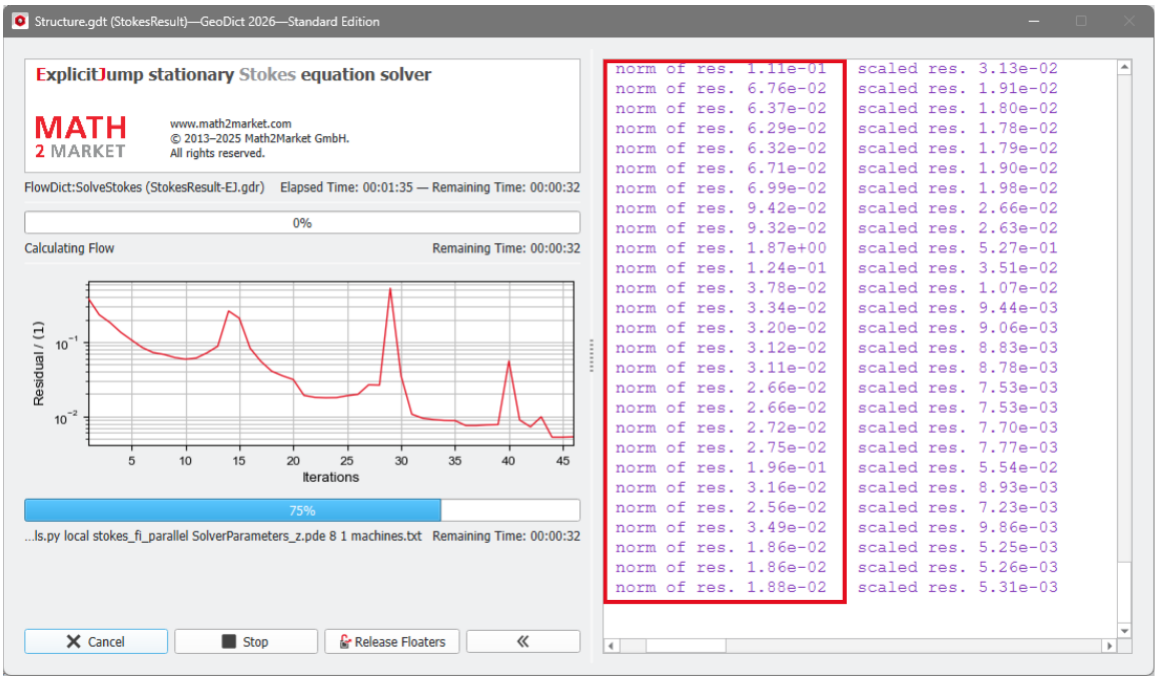
In each iteration, the solver checks for the current computed value x_c against the values x_{c-i} of the last 100 iterations if there are any changes.

$$\max_{i\in[1,100]} \frac{x_c - x_{c-i}}{x_{c-i}} < \text{tolerance} \tag{36}$$

The computation is stopped if the maximal relative change is smaller than the value entered for **Tolerance**. When there is doubt about the quality of the solution, decrease the **Tolerance** value by a factor of ten. The drawback of this criterion is that the solver sometimes might stop too early in case of slow convergence rate or at local extrema of oscillatory convergence curves.

When the **Residual** stopping criterion is used, the iteration is stopped if the solution satisfies the equation up to the required accuracy.

By setting the stopping criterion to **Residual**, the computations terminate as soon as the relative norm drops below the selected residual threshold. The relative norm of the Schur Complement residual is computed and displayed in the console window during the calculations. The console is visible by clicking the double arrow button on the lower right corner in the progress dialog.



The recommendation to choose **Tolerance** or **Residual** for the **EJ** solver is based on the structure's porosity. Both give similar results for highly porous structures. For dense structures, if using the Schur Complement **Residual**, the relative norm of the residual may be small even though the correct pressure drop has not been reached. So, when in doubt, use the **Tolerance** criteria – the default option.

Be aware that the default accuracy in FilterDict is one order of magnitude higher than in FlowDict. The stopping criteria depend on global values, whereas the particle movement depends on the local flow field. It is therefore necessary to make sure that not only the overall pressure drop is computed up to the given accuracy, but that also the local values are sufficiently accurate. This typically achieved by setting a higher accuracy as stopping criterion.

Simulation Stopping Criterion: Max. Iterations and Max. Run Time

Use the **Maximum Iterations** value or the **Maximum Run Time (h)** stopping criteria causes the solver to stop if the maximal number of iterations and/or the maximal runtime (in hours) is exceeded.

When the solver stops because one of these criteria has been reached, no guarantee on the quality of solution can be given. In this case, a warning is printed into the report. The following possibilities might help:

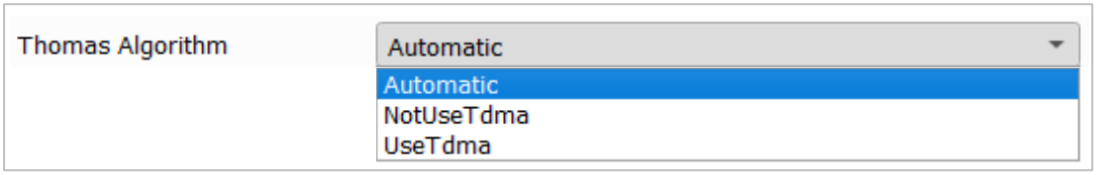
- Check the corresponding .log file to see how large the residual values and the pressure drop values are. If the pressure drop values are in reasonable range and they do not have changed during the last iterations, you may decide to use the current result.
- Double-check the structure and parameter values. Unphysical parameters or too

rough resolution of the structure (leading, e.g., to artificial unconnected components) can cause an iterative solver to fail.

Thomas Algorithm

The tridiagonal matrix algorithm (Tdma), also known as the **Thomas Algorithm**, may help to improve the convergence for high porosity structure, especially for Navier-Stokes. The algorithm transforms the 3D flow problem to 2D. Thus, it leads to a shorter runtime and increases the convergence speed.

Three options can be chosen from the pull-down menu: **Automatic**, **NotUse Tdma** or **Use Tdma**.



If **Automatic** is selected, the SimpleFFT solver decides if **Tdma** will be used, regarding the porosity of the structure. Thus, **Tdma** is used if the porosity is high. For most cases it is recommended to choose **Automatic**.

If a message dialog appears, displaying that NaN is detected in iterations, **NotUse Tdma** should be tried.

If the structures porosity is very high, it might decrease the runtime a little bit to explicitly choose **Use Tdma**. But for most cases **Automatic** will work well.

For more information about the Thomas algorithm see here.

Velocity Relaxation and Pressure Relaxation

Depending on the material parameters and geometrical structure of the structure, the underlying mathematical problem can vary in complexity, thus influencing the behavior of the solver. This is directly related to the Reynolds number, an indication of the complexity of the flow solver computations. The higher the Reynolds number, the more **Stable** the flow solver settings should be, resulting in higher number of iterations, slower time stepping, and longer flow solver run times. However, making the solver run less iterations and, thus, faster (**Fast**), implies the risk that the solver does not converge.

For the SimpleFFT solver, the management of this balance is done through the **Velocity relaxation: Stable ↔ Fast** and **Pressure relaxation: Stable ↔ Fast** slide bars.

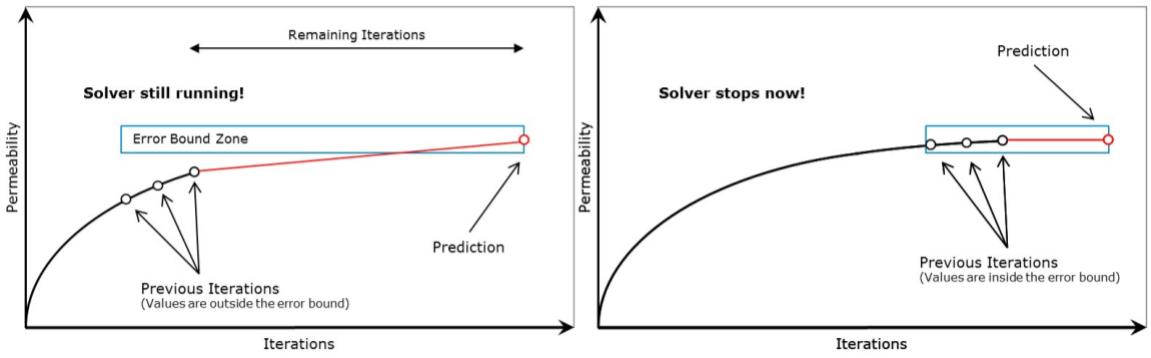
Setting the balance of **Stable** versus **Fast** is a trial-and-error process. Although there is no general rule to optimize it, the log files and the visualization of the structure might help finding the balance. Non-sense structure visualization results indicate that a more stable, slower solver is needed.

LIR

The LIR tab becomes editable when you select the LIR solver in the **Flow Calculation** panel.

Simulation Stopping Criterion: Error Bound or Tolerance

The stopping criterion **Error Bound** uses the result of previous iterations, and predicts the final solution based on linear and quadratic extrapolation. The solver stops if the relative difference between computed and predicted solution is smaller than the specified error bound. The stopping criterion recognizes oscillations in the convergence behavior and prevents premature stopping at local minima or maxima. A damped convergence curve is fit through the oscillating curve and the solver stops then regarding the damped convergence curve.



If the Krylov method (under LIR - Advanced Options) is activated, the definition of **Error Bound** is somewhat different. Here, no prediction is made, instead the continuity between neighboring cells and the conservation of mass is checked. The maximal value is normalized by the mean flow and if this value is smaller than the **Error Bound** the simulation stops.

The **Tolerance** stopping criterion looks for stagnation of the method when the process becomes stationary, i.e., the improvement in the pressure drop value becomes extremely small from iteration to iteration.

In each iteration, the solver checks for the current computed value x_c against the values x_{c-i} of the last 100 iterations if there are any changes.

$$\max_{i \in [1,100]} \frac{x_c - x_{c-i}}{x_{c-i}} < \text{tolerance} \tag{36}$$

The computation is stopped if the maximal relative change is smaller than the value

entered for **Tolerance**. When there is doubt about the quality of the solution, decrease the **Tolerance** value by a factor of ten. The drawback of this criterion is that the solver sometimes might stop too early in case of slow convergence rate or at local extrema of oscillatory convergence curves.

Be aware that the default accuracy in FilterDict is one order of magnitude higher than in FlowDict. The stopping criteria depend on global values, whereas the particle movement depends on the local flow field. It is therefore necessary to make sure that not only the overall pressure drop is computed up to the given accuracy, but that also the local values are sufficiently accurate. This typically achieved by setting a higher accuracy as stopping criterion.

Simulation Stopping Criterion: Max. Iterations and Max. Run Time

Use the **Maximum Iterations** value or the **Maximum Run Time (h)** stopping criteria causes the solver to stop if the maximal number of iterations and/or the maximal runtime (in hours) is exceeded.

When the solver stops because one of these criteria has been reached, no guarantee on the quality of solution can be given. In this case, a warning is printed into the report. The following possibilities might help:

- Check the corresponding .log file to see how large the residual values and the pressure drop values are. If the pressure drop values are in reasonable range and they do not have changed during the last iterations, you may decide to use the current result.
- Double-check the structure and parameter values. Unphysical parameters or too rough resolution of the structure (leading, e.g., to artificial unconnected components) can cause an iterative solver to fail.

Advanced Options: Write Compressed Volume Field

The LIR solver uses a very memory efficient adaptive grid structure for the simulations.

If the option **Write Compressed Volume Fields** is checked, then the adaptive grid is used as compression method for writing out *.vap files. This option allows to save 80-90% space on hard drive. The runtime for writing *.vap files is also reduced significantly. But the runtime for loading and uncompressing of compressed *.vap is increased by the amount of runtime that was saved for writing out compressed *.vap files.

If the option **Write Compressed Volume Fields** is not checked, then a usual regular grid is used for writing out *.vap files.

Advanced Options: Optimization Options

The **Multigrid Method** (see, e.g., Wesseling, 2004) was introduced to speed-up the computation and reduce the runtime significantly. The main idea of **Multigrid** is the usage of multiple coarser adaptive grids to speed up convergence behavior but requires only little more memory.

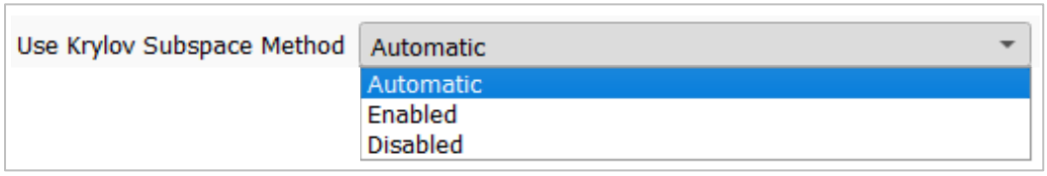
The method is available to solve the Stokes and Stokes–Brinkman equations as well as for solving mechanics, diffusion, thermal, and electrical conduction and is enabled by default.

Another speed-up option to accelerate the convergence behavior of the LIR solver is called **Krylov Subspace Method**.

The runtime of the LIR solver depends on many different properties of the structure and the simulation parameters. The BiCGStab algorithm is used, which can reduce the runtime for challenging simulation very drastically.

Note! Using the BiCGStab method approximately doubles the amount of RAM needed for the computation.

Unfortunately, the Krylov method is not always faster than a simulation without the Krylov method and therefore we introduced an **Automatic** mode which uses some heuristics to choose the most efficient method based on structure, material parameters, and boundary conditions automatically. Of course, it is possible to explicitly enable (**Enabled**) or disable (**Disabled**) the method.



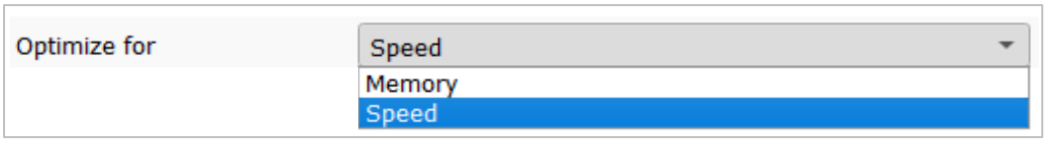
Note! In case that the Krylov subspace method (BICGStab) is used, the **Relaxation** may also be adapted automatically.

Depending on the material parameters and geometry of the structure, the underlying mathematical problem can vary in complexity, thus influencing the behavior of the solver. The more complex the problem is, the more stable the solver settings should be.

With the **Relaxation** number, the solver is adjusted from **Stable** (which results in higher number of iterations, slower time stepping, and longer solver run times), to **Fast**, which makes the solver run less iterations but implies the risk that the solver does not converge. The **Relaxation** is a parameter of the SOR method and must be between 0 and 2 to ensure convergence. For relaxation values smaller than one (<1.0), the

simulation is more stable. For relaxation values larger than one (>1.0), the simulation converges faster.

The **LIR** solver can **Optimize for Speed** or **Memory**.



- If **Speed** is chosen, the solver constructs additional optimization structures. The runtime is decreased by up to 30% but requires up to 50% more memory compared to the other option.
- If **Memory** is chosen, the runtime is increased by up to 40% but the solver requires up to 50% less memory.

Advanced Options: Grid Options

The **Grid Type** decides what kind of tree structure is used for the simulation.

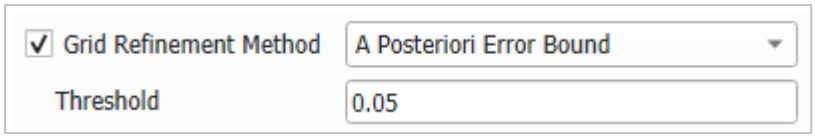
The default option is **LIR-Tree** and should always be used. The solver uses an adaptive grid structure called LIR-tree and needs up to 10 times less runtime and memory compared to the **Regular Grid** option.

The solver can analyze the result field during the computation and improves the adaptive grid in places where more accuracy is needed. The LIR solver splits cells where a high gradient occurs.

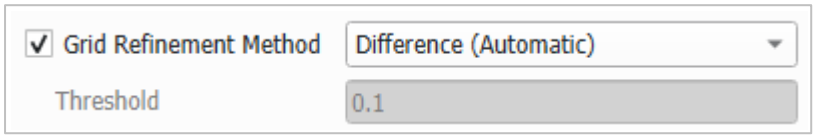
The solver can analyze the computed field and refines the adaptive grid during the computation at locations where more accuracy is needed. The LIR solver splits cells where a high gradients occur when the **Grid Refinement Method** is enabled. In this case, the additional parameters **Number of Grid Refinements**, **Allow Sub-Voxel Resolution**, and **Allow Grid Re-Coarsening** become available.

Select one of the three available refinement methods from the drop-down menu.

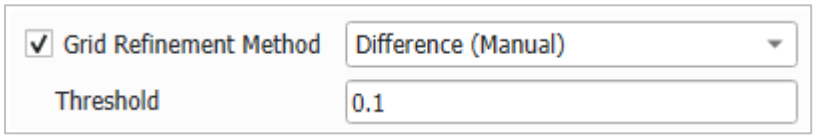
When **A Posteriori Error Bound** is selected, the solver targets the specified accuracy **Threshold**. While the **Error Bound** (set as Simulation Stopping Criterion) determines the relative error in the solution of the linear system, **A Posteriori Error Bound** refers to the relative error estimated by comparing high-order and low-order discrete solutions. The accuracy **Threshold** refers to the relative error to the analytical solution and the value must be between 0.0 and 1.0.



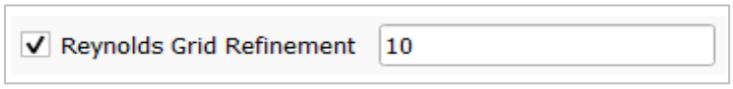
Choosing **Difference (Automatic)** leads to computational cells being split when the difference in values between neighboring cells exceeds a certain **Threshold**. The solver automatically chooses all internal parameters based on the structure and simulation settings. For this option, cells are split where the current gradient is greater than the **Threshold** multiplied by the maximum gradient, where the threshold is determined automatically.



Selecting **Difference (Manual)** also causes the computational cells to be split when the difference in values between neighboring cells exceeds the specified **Threshold**. You can specify all parameters (Threshold and Number of Grid Refinements) for the grid refinement manually. For this option, cells are split where the current gradient is greater than the **Threshold** multiplied by the maximum gradient. The **Threshold** value must be between 0.0 and 1.0. The recommended value range is between 0.05 and 0.1.



Activate **Reynolds Grid Refinement** to use the local Reynolds number (Re) for adaptive grid refinement.



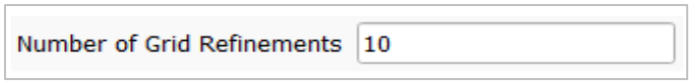
A grid cell is split if the local Reynolds number, i.e., the ratio between inertial and viscous forces exceeds the given threshold. The Reynolds number is defined as:

$$\mathrm{Re} = \frac{\rho \mathrm{u} \mathrm{L}}{\mu} \quad (37)$$

Reynolds Number

where ρ is the density of the fluid (SI units: kg/m^3), u is the velocity of the fluid (m/s), μ is the dynamic viscosity of the fluid (Pa·s or N·s/m^2 or kg/m·s), L is a characteristic length (m). For the local Reynolds number, the size of the grid cell is used for the characteristic length.

The **Number of Grid Refinements** controls the maximum number of grid refinements that the solver can perform during the simulation. The value should be set between 0 and 10, where a value of 0 means that no grid refinements will be made. Grid refinements may increase the number of iterations, runtime, and memory requirements.



When **Allow Sub-Voxel Resolution** is enabled, the solver is allowed to split computational cells to sizes smaller than the voxel length.

This feature is beneficial for low-porosity structures for which the pore throats require finer resolution or for modeling fast Navier-Stokes flows with strong vortices.

Enabling this feature may increase the number of iterations, runtime, and memory requirements.

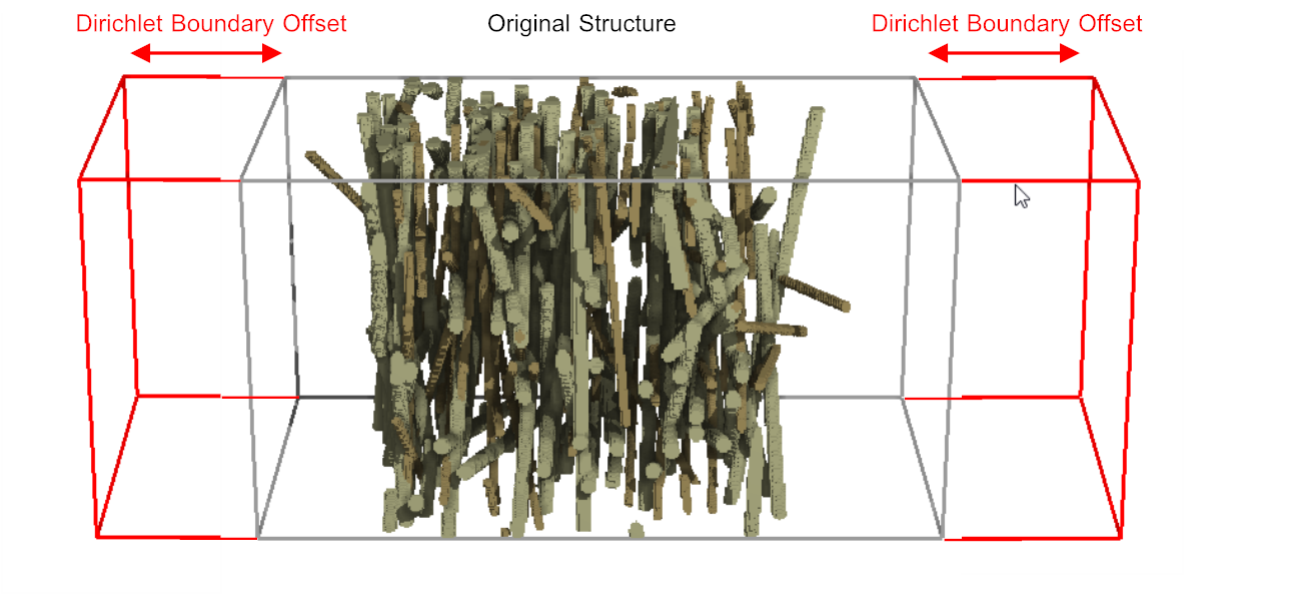
Check **Allow Grid Re-Coarsening** to allow the solver to automatically revert the grid refinement and coarsen the computational cells.

This means, grid refinement is done temporarily. Afterwards, the cells are merged as soon as accuracy is not needed anymore. This reduces the memory and runtime requirements.

Estatic

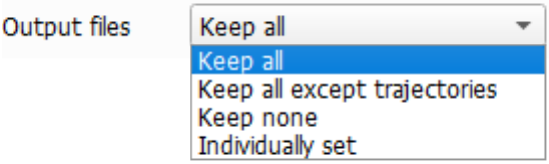
The **EStatic** tab is becomes editable when **Include Electrostatic Effects** has been checked on the Electrostatic Effects tab. Equation (15) 9 is solved to calculate the electrostatic potential.

The **Dirichlet Boundary Offset** is the offset Z_0(in Voxels), where the zero Dirichlet boundary conditions for the potential Φ apply. For large numbers Z_0, the computation is more accurate but also requires more numerical resources. This increment of the computational domain occurs internally for the electrostatic solver and does not influence the structure seen by the user in the visualization area.

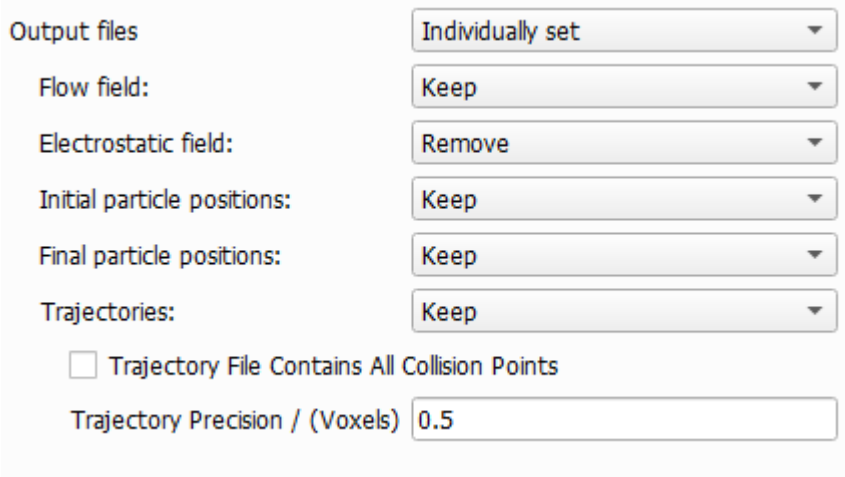


Output

The **Output** tab allows to select the files stored by the simulation. They can be used afterwards to visualize the corresponding results.



Select one of the predefined options: **Keep all**, **Keep all except trajectories**, or **Keep none**. Alternatively, select **Individually set** to control which file types you wish to keep.



For each file type, you can select to **Keep** or **Remove** the files. You have to keep the flow field if you want to visualize the flow (e.g., with streamlines) after the simulation. You have to keep the final particle positions if you want to visualize the deposited particles.

To visualize the movement of particles, you need to keep the **Trajectories**. Be aware, that these files may become very large if many particles are simulated. You can influence the file

size and the precision of the stored trajectories (e.g., the number of points on the particle paths saved in the file) by two additional parameters:

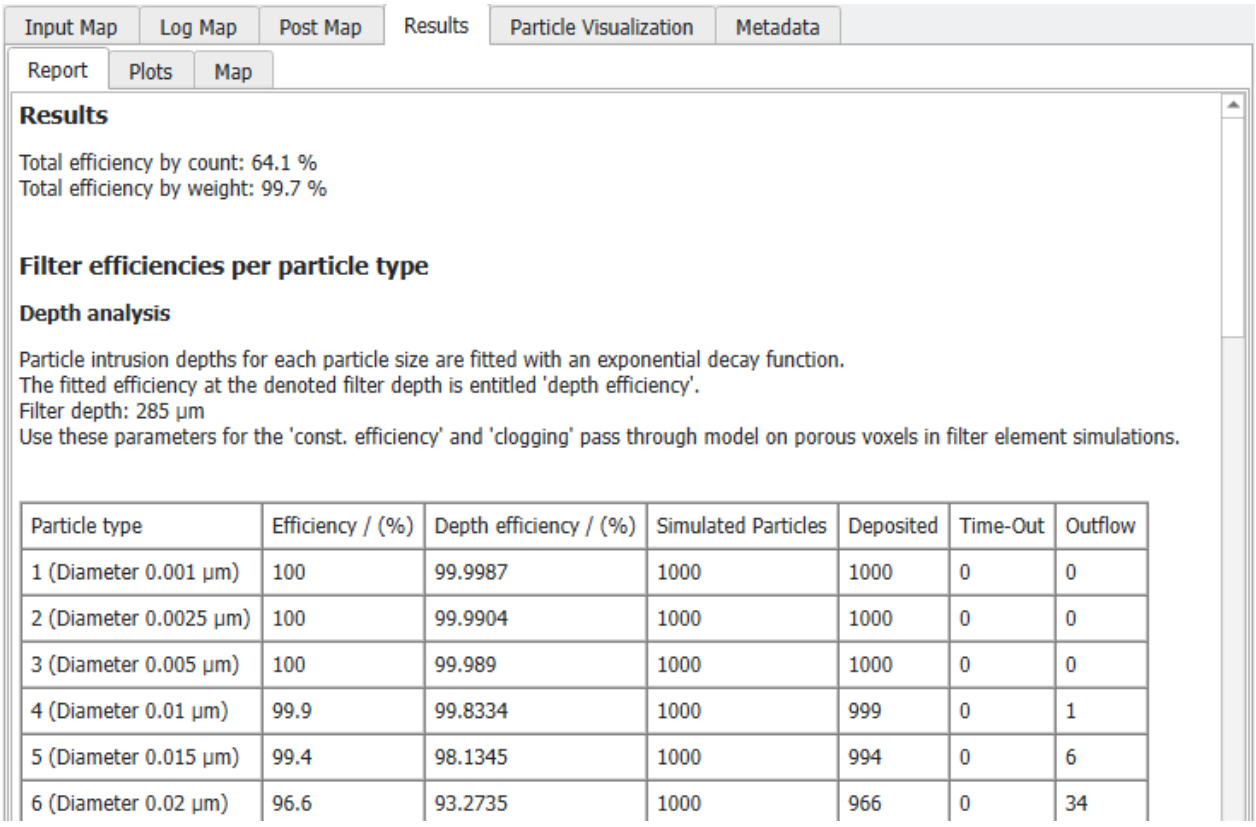
- Check **Trajectory File Contains All Collision Points** to add all points where a particle touches the surface of a solid object to the trajectory file.
- The value entered as **Trajectory Precision** determines how accurately the trajectories are stored. The given value determines the traveling distance between two points stored in the trajectory file.

Results

Click **OK** to input the entered parameters, and then click **Run** in the FilterDict section to start the Filter Efficiency simulation.

The results are immediately shown in the opening Result Viewer after the simulation is finished.

Report



Particle type	Efficiency / (%)	Depth efficiency / (%)	Simulated Particles	Deposited	Time-Out	Outflow
1 (Diameter 0.001 µm)	100	99.9987	1000	1000	0	0
2 (Diameter 0.0025 µm)	100	99.9904	1000	1000	0	0
3 (Diameter 0.005 µm)	100	99.989	1000	1000	0	0
4 (Diameter 0.01 µm)	99.9	99.8334	1000	999	0	1
5 (Diameter 0.015 µm)	99.4	98.1345	1000	994	0	6
6 (Diameter 0.02 µm)	96.6	93.2735	1000	966	0	34

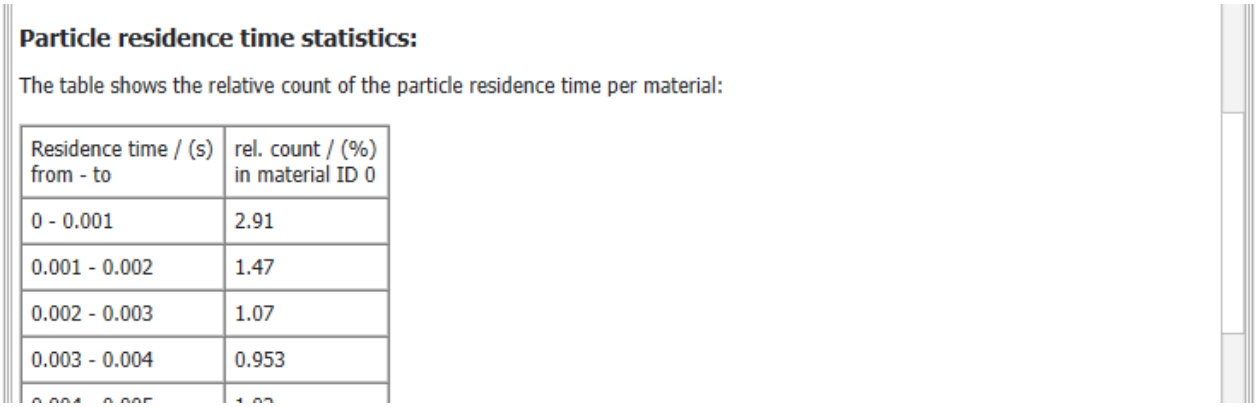
Under the **Report** tab, the following results are reported:

- The **Total efficiency by count**, computed as $\sum_i E_i \cdot P_{c,i}$, where E_i is the fractional filtration efficiency of a particle type and $P_{c,i}$ the count probability (**Count %** entered on the **Particles→Size Distribution** tab) of this particle type.

- The **Total efficiency by weight**, computed as $\sum_i E_i \cdot P_{\mathrm{m},i}$, where E_i is the fractional filtration efficiency of a particle type and $P_{\mathrm{m},i}$ the mass probability (**Mass %** entered on the **Particles→Size Distribution** tab) of this particle type.

In the **Filter Efficiencies per Particle Type** table below, for each particle type, the following values are given:

- The fractional filtration **Efficiency**.
- An estimated fractional filtration efficiency, called **Depth Efficiency**, determined by fitting an exponential decay function to the particle intrusion depth distribution (see the description of the Plots 54 sub-tab below for details).
- The overall number of **Simulated Particles** of this type. Be aware that those numbers do not follow the size distribution entered on the **Particles→Size Distribution** tab, each particle type is simulated **Number of particles per type** times as entered on the **Particles→Positioning** tab.
- The number of those particles **Deposited** on the filter.
- The number of **Time-Out** particles, which are still in motion when the maximal simulation time is reached (**Max time reached** on the Filter Experiment tab 19).
- The number of particles that have arrived unfiltered in the **Outflow** area.



Particle residence time statistics:

The table shows the relative count of the particle residence time per material:

Residence time / (s) from - to	rel. count / (%) in material ID 0
0 - 0.001	2.91
0.001 - 0.002	1.47
0.002 - 0.003	1.07
0.003 - 0.004	0.953

The **Particle residence time statistics** table shows how long a particle stays inside of a medium. For filter media simulations, particles can only reside inside of the fluid, which by default has material ID 0. So therefore, this table shows how long the particles moved inside of the fluid until they either deposited on the filter material or left the simulation area through the outlet.

The Residence time is subdivided into steps whose size is defined by the Max. time reached 20 entered in the **Filter Experiment** tab divided by 1000.

Particle collision statistics:

The table shows the mean values for a single particle:

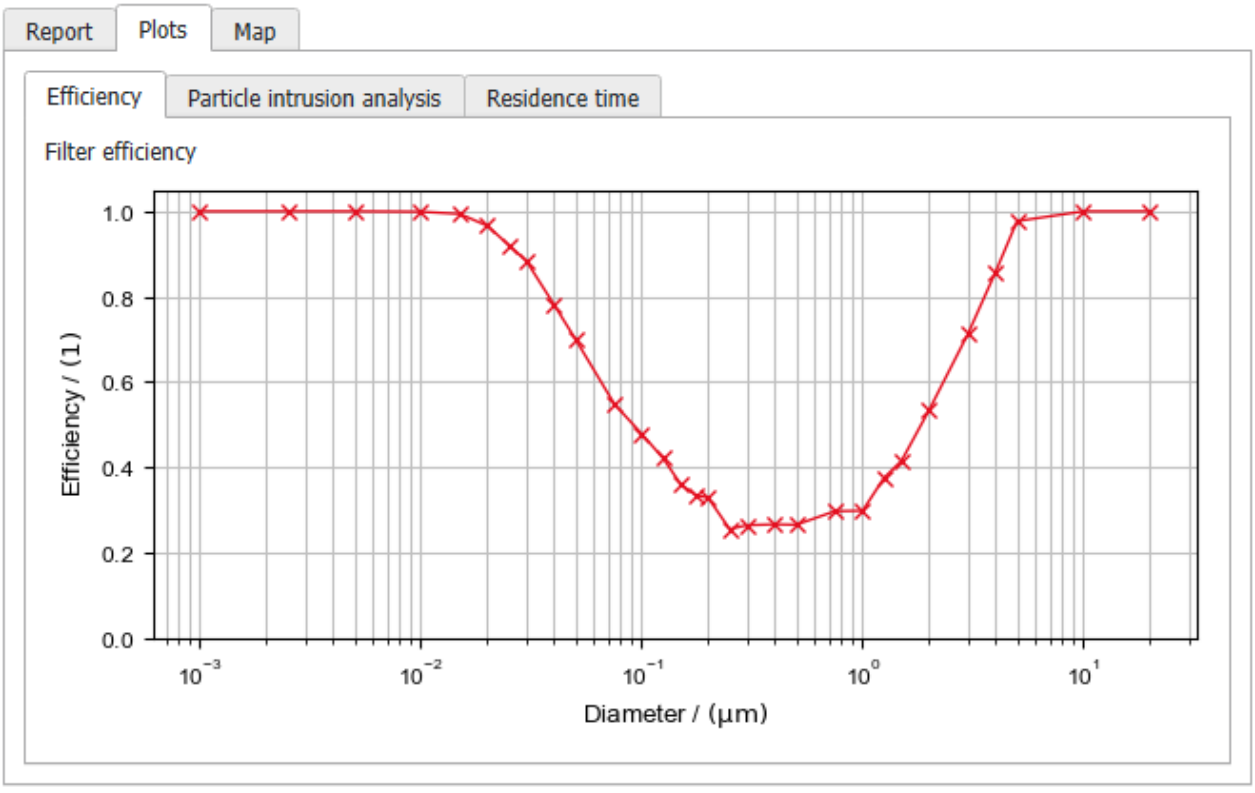
Material ID	Residence time / (s)	Collisions	Entries	Trapped / (%)
0	0.0199052	0	0	0
1	0	0.3841	0	38.41
2	0	0.257067	0	25.7067

Total runtime: 00:03:27, Total memory usage: 4.767 GiB

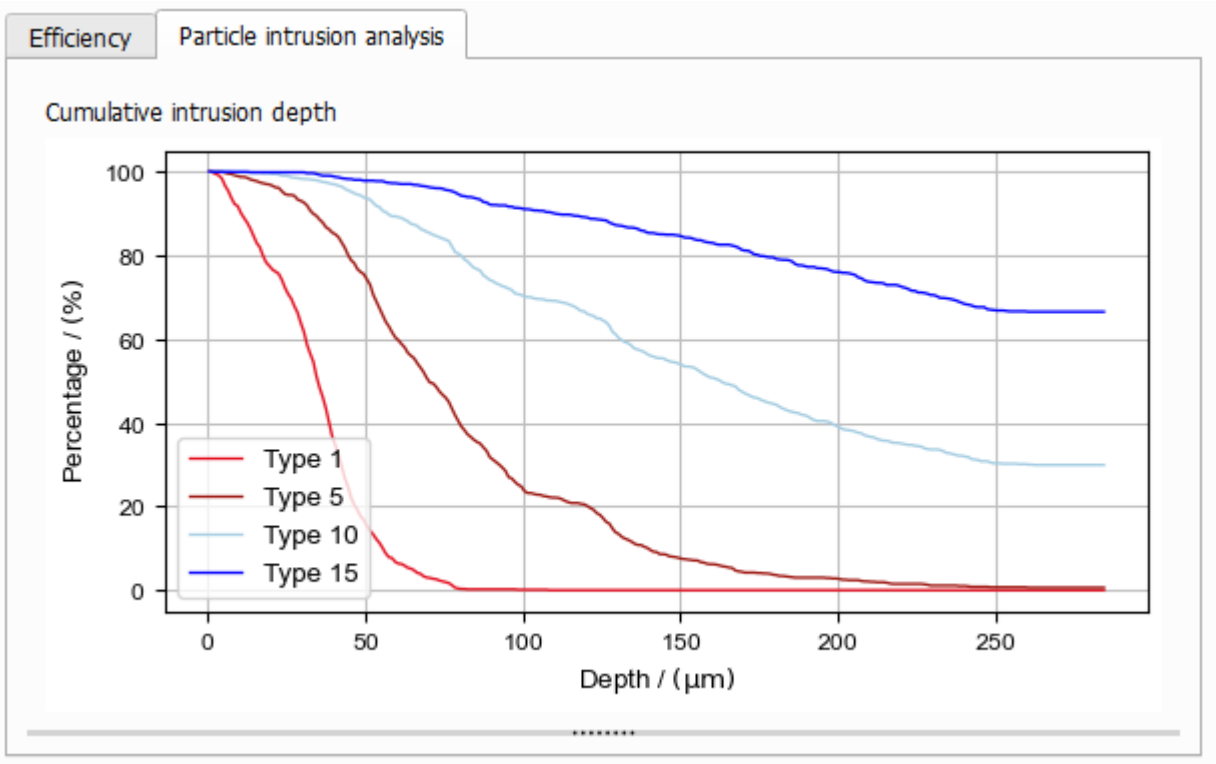
The **Particle collision statistics** table shows the mean value of the residence times in each material. In solid materials that the particle cannot enter, the residence time is 0. The **Entries** column shows how often particles enter into the corresponding material. In the above case, where material ID 0 is the fluid material, and material ID 1 and 2 are solid materials that the particles cannot enter, the number of entries is 0. The **Trapped** column shows the percentage of particles deposited on the surface of the material. The **Collisions** column shows the average number of collisions a particle has with the material surface. With the **Caught On First Touch** collision model, the number of collisions corresponds to the percentage of trapped particles, because every collision leads to a trapped particle.

Plots

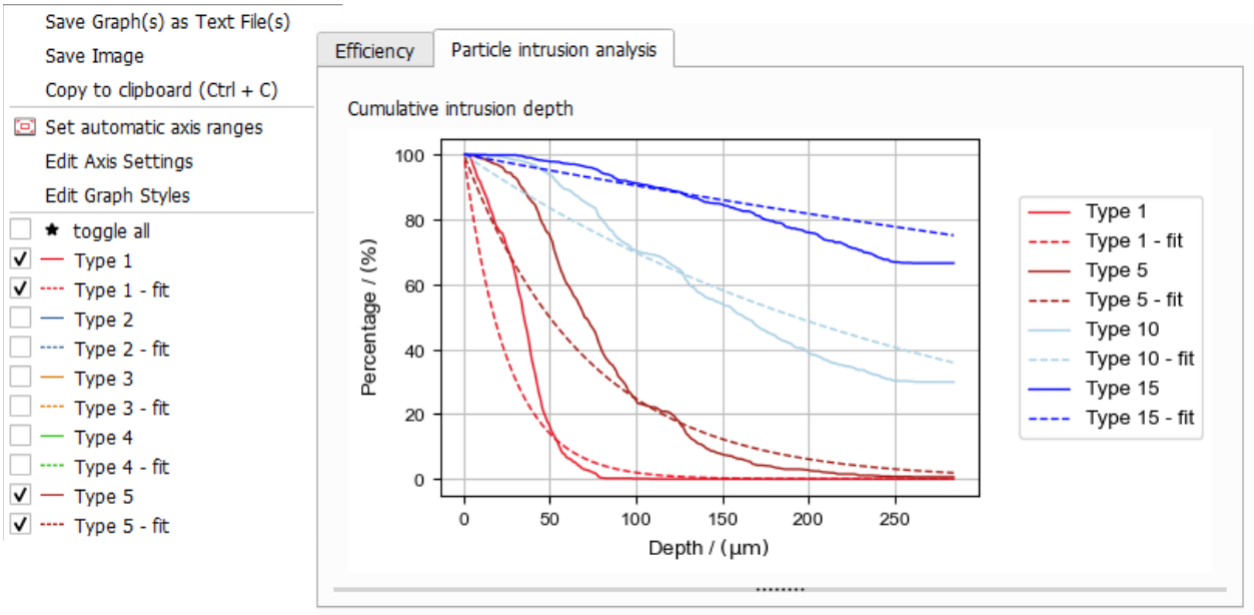
On the **Plots** tab, three sub-tabs are available. The **Efficiency** tab depicts the computed fractional filtration efficiencies:



Two plots are shown on the **Particle Intrusion Analysis** plot. The **Cumulative Intrusion Depth** shows for each particle type (i.e. particle size), how far particles of this size move into the filter before they are captured. Starting on the left, at Depth 0 still 100% of the particles are unfiltered. When moving into the filter, the percentage of unfiltered particles goes down, until at the downstream side the particles reach the outflow.



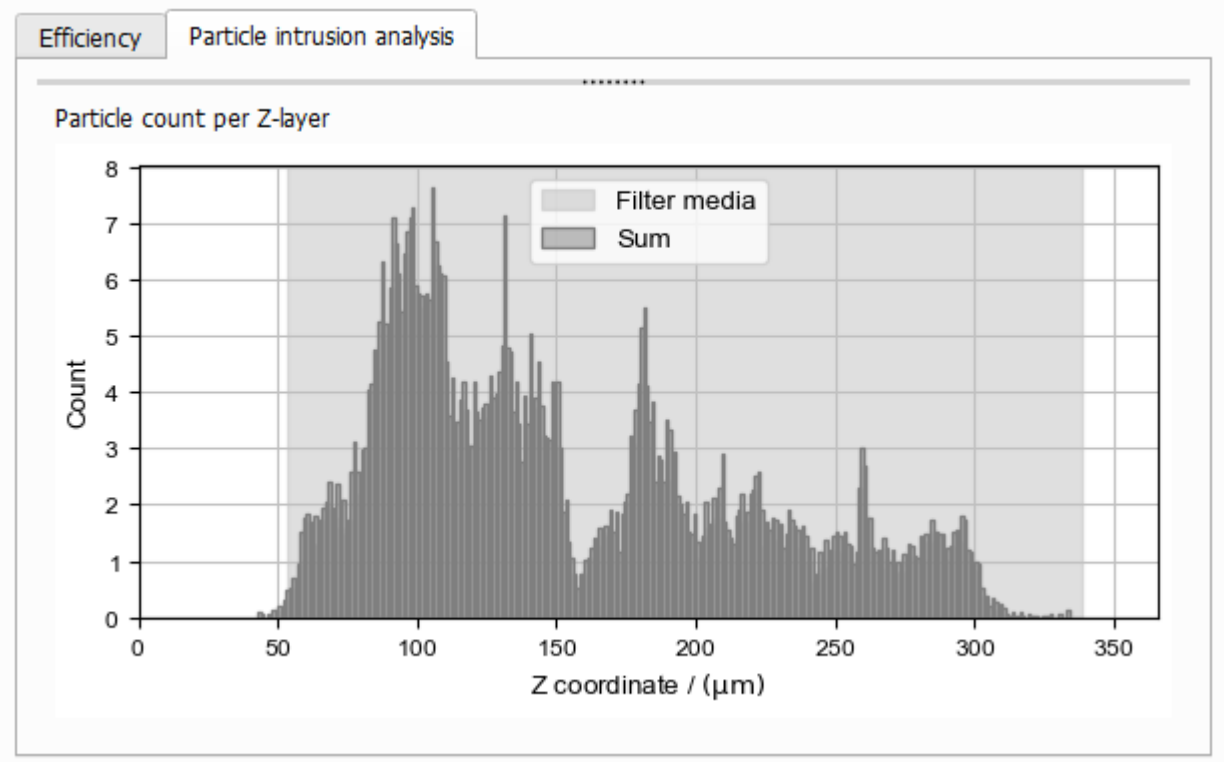
Right-click into the plot area to open a dialog box that lets you select which graphs are shown. For each particle type, you can additionally select to plot the fitted exponential decay function:



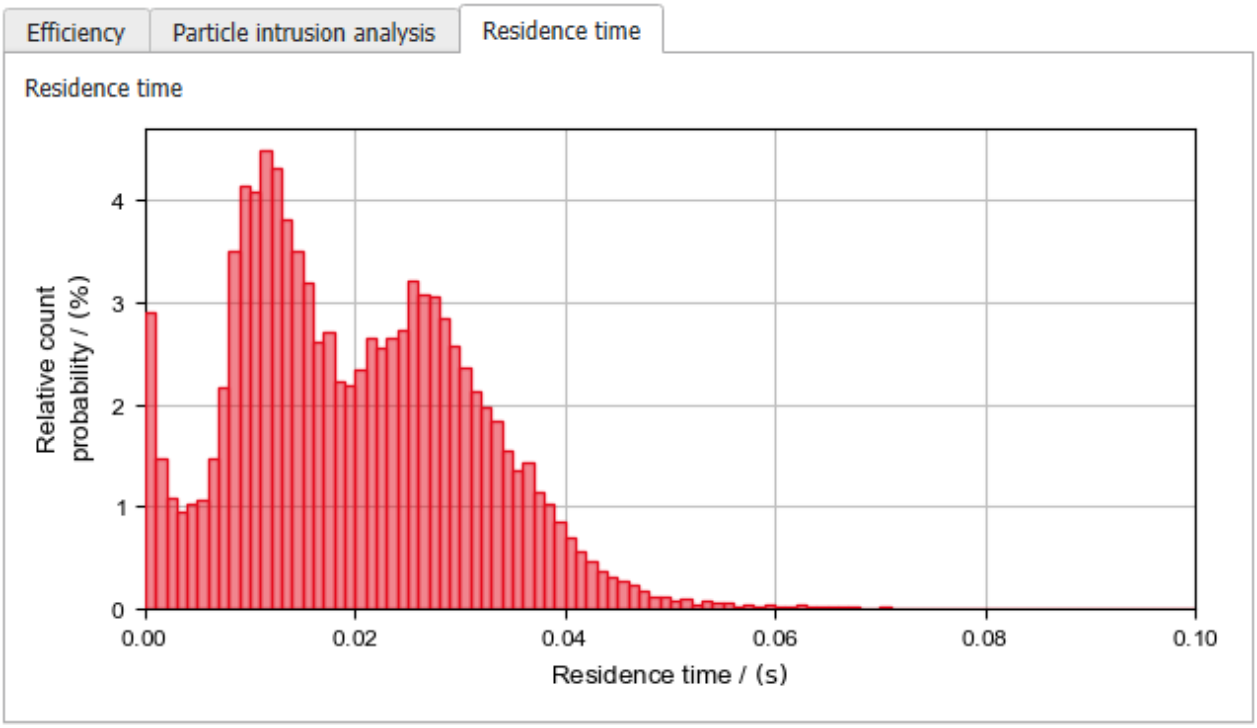
If the filter material is homogeneous over the depth of the filter, the probability that a moving particle hits a fiber is constant during the movement through the filter. In that case, the particle

intrusion depth should follow an exponential decay, e.g., 50% are captured after the first 50 µm, 75% after 100 µm, 87.5% after 150 µm and so on. Therefore it is sensible to fit an exponential decay function through the computed intrusion depths as a model for the filter efficiency. The reported **Depth Efficiency** is the filtration efficiency predicted by this model.

On the **Particle count per Z-layer** plot, the distribution of the collected particles over the filter height is plotted. For each particle type, the number of particles deposited on each Z-layer is counted. To reflect the given particles size distribution, this value is scaled with the particle types **Count Percentage** (recall that for Filter Efficiency simulations the same number of particles is simulated for each particle type independently of the given physical particles size distribution).

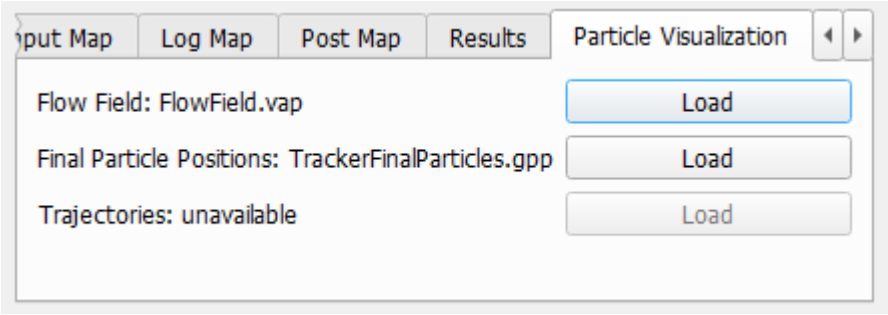


The **Residence time** plot plots the values shown in the **Particle residence time statistics** table.



Particle Visualization

On the **Particle Visualization** tab, it is possible to load and visualize the computed flow field, the computed particle trajectories, and particle end positions. The options available here depend on the choices made on the Output tab 51.



During filter efficiency simulations, no filter cake is created, and all particles move independently from each other. Nevertheless, a large number of possible trajectories is computed for all particle sizes to get statistically meaningful efficiency results. Therefore, visualization of the particle end positions and trajectories will not give a realistic impression of a filtration process. Rather, the trajectories show all the possible pathways a dust particle may take.

Clicking **Load** imports the particle data and displays the particles in the viewport.

1.2.2 Filter Lifetime

A **Filter Lifetime** simulation includes clogging of the filter and computes pressure drop and deposited dust over time. In a Single Pass simulation, fluids pass through the filter only once

and are not recirculated. In a Multi Pass simulation, fluids move in a circuit through the system, and particle size distribution and concentration in front of the filter change over time.

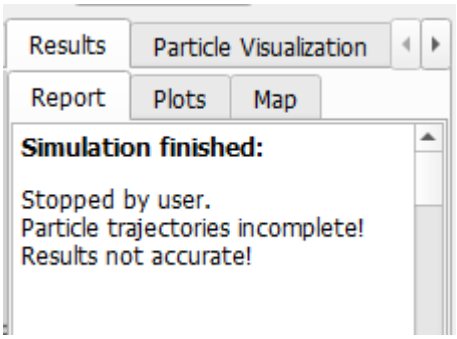
The Filter Lifetime command for filter media

- Options 59
  How to set the input parameters for the Filter Lifetime command.
- Results 105
  How to understand the computed results.
- Particle Visualization 116
  How to visualize the computed results.
- Create Videos 120
  How to create videos from the computed results.

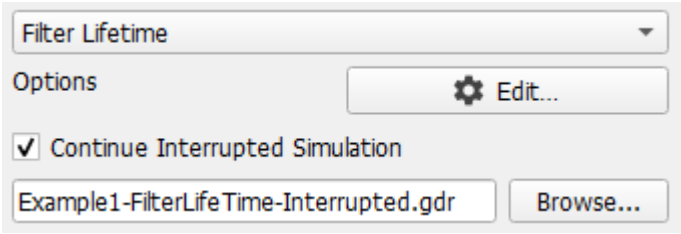
Continue Interrupted Simulation

An interrupted filter life time simulation can be restarted by checking **Continue Interrupted Simulation**. For this, at least the computation of the first flow field must have been finished. When the simulation was interrupted earlier, it must be started anew.

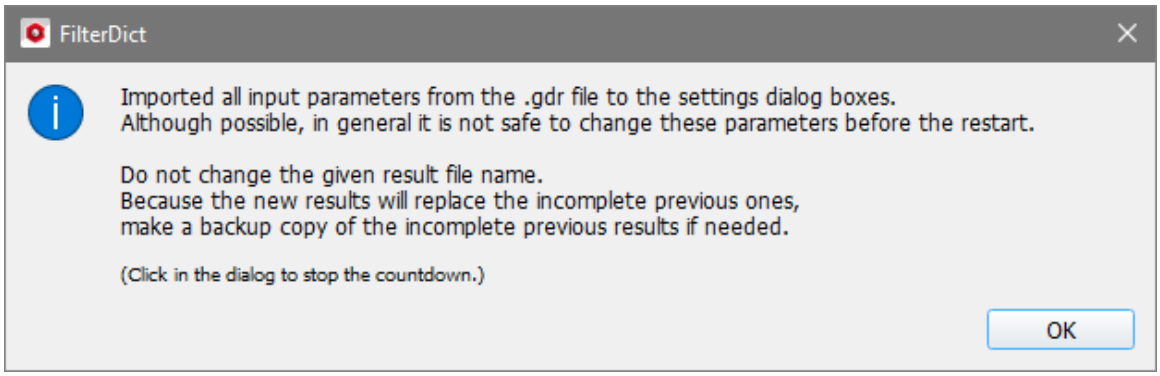
A stopped or interrupted simulation produces an incomplete (*.gdr) result file which may not automatically open at the end of the computations. When opened through **File → Open *.gdr File...**, you can find the information that the simulation was not finished under the **Results - Report** subtab:



To continue the unfinished simulation, check **Continue Interrupted Simulation** and click **Browse** to select the name of the result file. To continue the interrupted simulation, the result file from the unfinished simulation must be located in the project folder and the structure on which the simulation is to be finished must be already loaded.



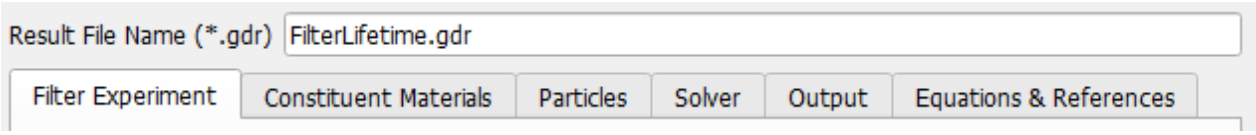
A message appears giving information and instructions on how to handle the simulation re-run.



Click **OK** in the message to return to the FilterDict section and click **Run** to continue the interrupted simulation..

Options

Select **Filter Lifetime** from the pull-down menu and click the **Option**'s **Edit...** button to open a dialog to enter or edit the simulation options. Choose a **Result File Name** for the result file (*.gdr) fitting to your current project.



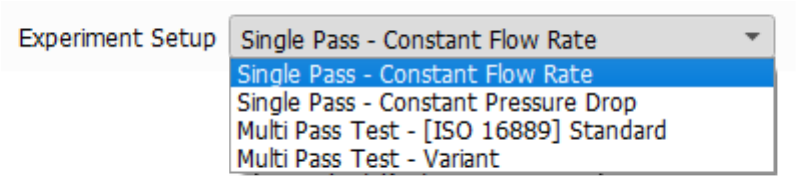
The parameters are organized under the tabs: **Filter Experiment**, **Constituent Materials**, **Particles**, **Solver**, and **Output**. The **Equations & References** tab gives a summary of the used equations and symbols which are explained in more detail in this user guide.

Tabs of the Filter Lifetime dialog

- Filter Experiment 60
  Select Single Pass or Multi Pass experiment, and set the boundary conditions.
- Constituent Materials 64
  Define fluid type and temperature. Set material IDs and parameters for porous layers.
- Particles 69
  Select particle sizes and their start positions. Set all parameters that govern the particle motion and define collision and adhesion models.
- Solver 87
  Select the flow solver, set the stopping criteria and define the computational resources.
- Output 104
  Choose which data will be stored for visualization.

Filter Experiment

Under the **Filter Experiment** tab, choose between a **Single Pass** or a **Multi Pass** experiment. For **Single Pass**, the experiment can be run at a **Constant Flow Rate** or at a **Constant Pressure Drop** (often used for membranes). For **Multi Pass**, choose between the standard test setup or a variant with an initially contaminated test reservoir.



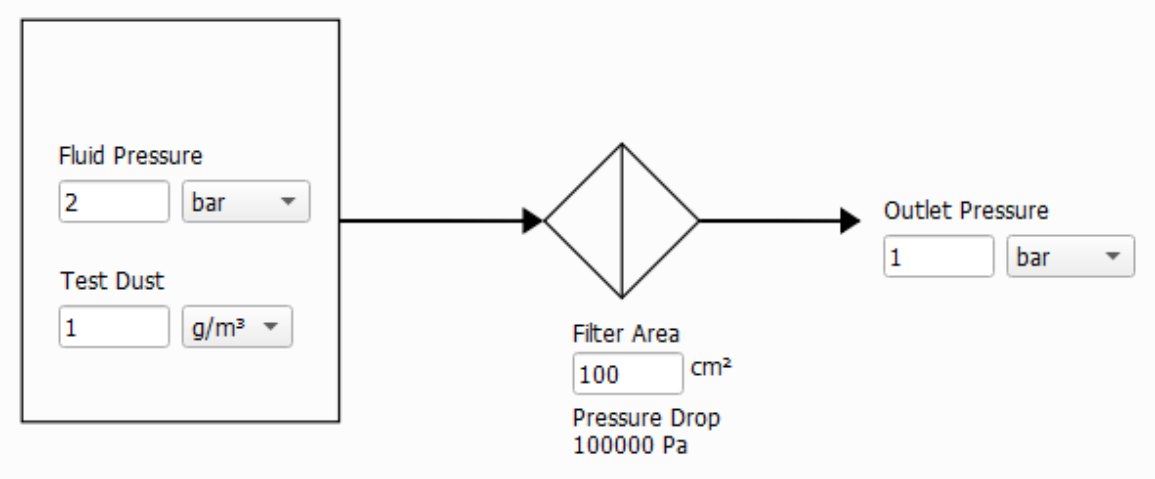
When a method is selected, the setup is shown schematically in the dialog below the pull-down menu. Furthermore, the selected method will influence which options are available elsewhere in the Filter Lifetime dialog.

Single Pass - Constant Flow Rate

Single Pass - Constant Pressure Drop

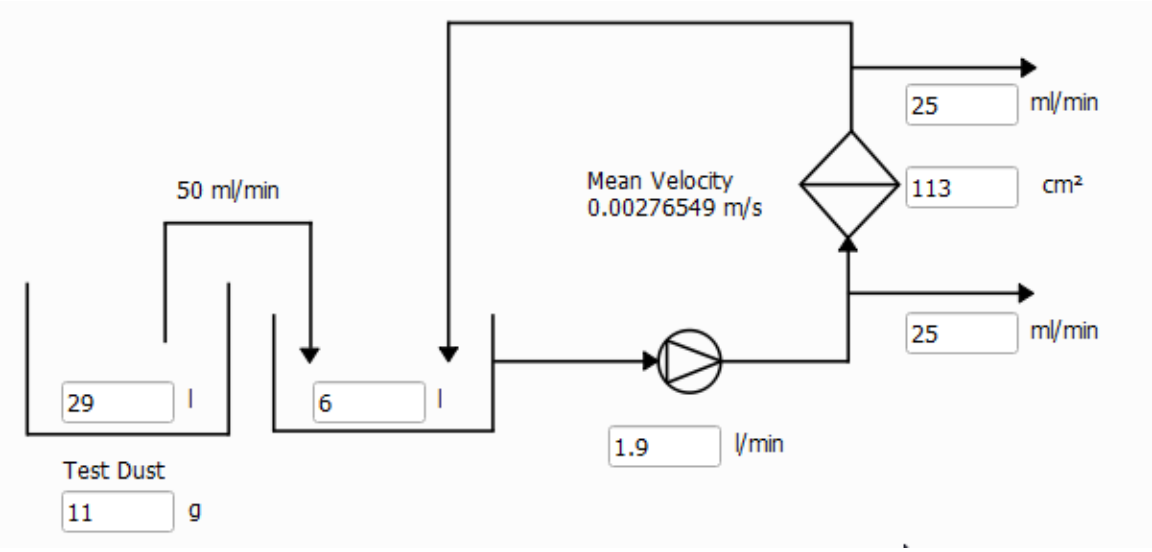
Here, the pressure drop at the filter is kept constant and the mean velocity / flow rate decreases over time.

For this choice, enter the **Fluid Pressure** in the containment, the **Outlet Pressure**, the **Test Dust** concentration in the fluid, and the test **Filter Area**. The **Filter Area** is the surface area of the filter in the experiment (and not the surface of the structure in GeoDict). The **Pressure Drop** used in the simulation is the difference between the inlet and the outlet pressure.



Multi Pass Test - [ISO 16889] Standard

In the standard multi pass test, the test reservoir is initially clean, and the test dust is injected into the system during the experiment.

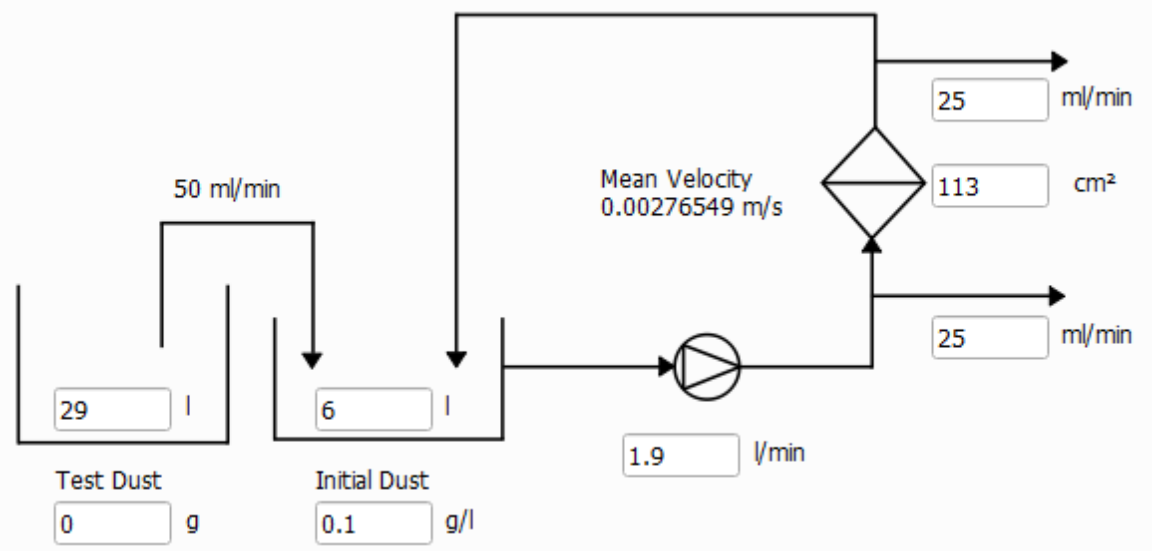


In the sketch, define the **Test Dust** (in g), the **Pump flow rate** (in l/min), and the **Test filter area** (in cm^2). Enter also the **Injection reservoir volume** and the **Test reservoir volume** (recirculation). The **Outflow to upstream sampler** and the **Outflow to downstream sampler** (in ml/min) can also be set. The **Injection flow rate** matches the sum of the upstream and the downstream outflows (in ml/min).

Hovering with the mouse over the edit boxes shows additional information in a tooltip.

Multi Pass Test - Variant

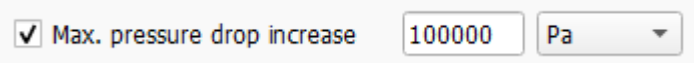
With this option, also the amount of dust in the **Test Reservoir** can be set in the beginning of the simulation, i.e. you can use this to model an initially contaminated test reservoir. Otherwise, this option and its parameters are the same as those of the ISO 16889 standard multi-pass test.



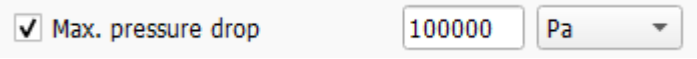
Simulation Stopping Criterion

The simulation continues to run until one of the following criteria is reached:

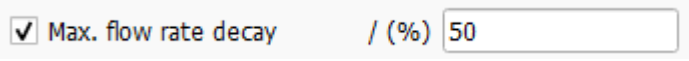
- If **Max. pressure drop increase** is selected, the simulation stops if the initial pressure drop is increased by the given value. This option is not available if **Single Pass - Constant Pressure Drop** was selected.



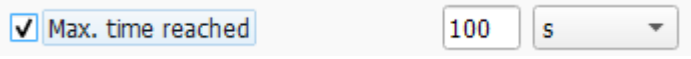
- If **Max. pressure drop** is selected, the simulation stops if the given pressure drop is reached. This option is not available if **Single Pass - Constant Pressure Drop** was selected.



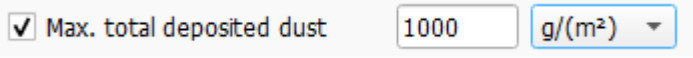
- If **Max. flow rate decay** is selected, the simulation stops when the flow rate falls below the given percentage (%) of the initial flow rate. This option is only available if **Single Pass - Constant Pressure Drop** was selected.



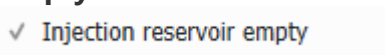
- If **Max. time reached** is selected, the simulation stops when the defined simulation time is reached.



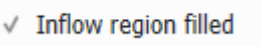
- If **Max. total deposited dust** is selected, the simulation stops when the defined amount of deposited dust is reached.



- **Multi Pass Test** simulations always automatically stop when the **Injection reservoir** is **empty**.



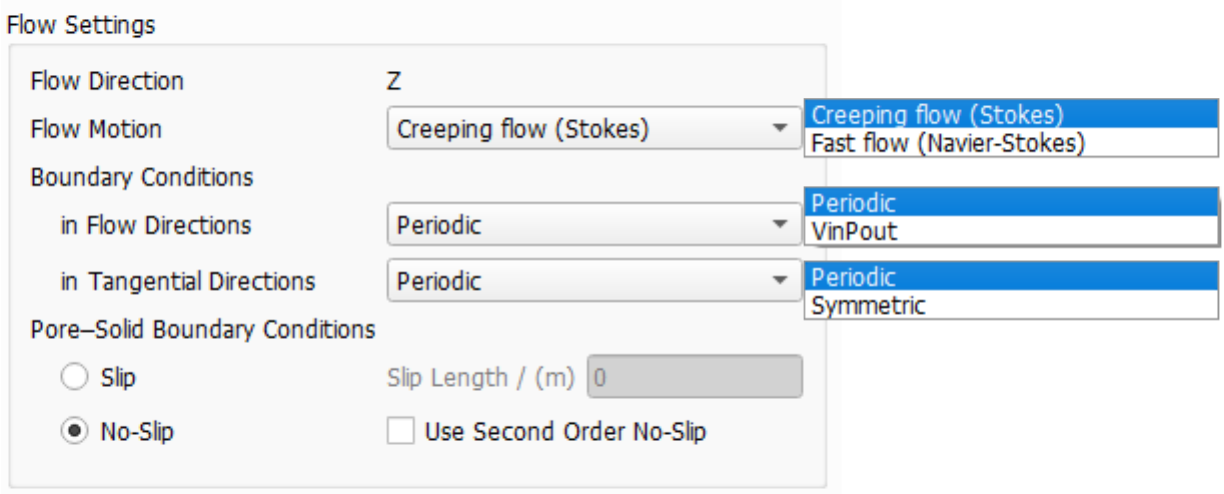
- The simulation always automatically stops when the **Inflow region** is **filled** with particles. This is triggered when the first deposited particle reaches the inflow plane.



Note! Before GeoDict 2024 SP1, the simulation ran until an entering particle could not be placed anywhere in the inflow area. Thus, the inflow region got completely filled before the **Inflow region filled** state was reached. As a result, if the simulation ran until this criterion was reached, it showed a dense and compact filter cake.

Since GeoDict 2024 SP1, the simulation stops when the first deposited particle reaches the inflow plane. In case of dendritic growth of the filter cake, this means that the simulation will stop earlier now, but also that the dendritic shape of the cake will be visible up to the end of the simulation.

Flow Settings



In FilterDict, the **Flow direction** and, hence the filtration direction, is always the Z-direction.

The **Flow Motion** can either be creeping or fast flow. In the creeping flow case, the Stokes equations are solved to determine the flow field. If fast flow is chosen, the Navier-Stokes equations are solved to determine the flow field.

As the Stokes equation is a simplification of the Navier-Stokes equation, choosing **Fast flow (Navier-Stokes)** will give correct results also in the case of slow, creeping flow. Solving the Navier-Stokes equations is computationally more expensive than solving the Stokes equations, so it is advisable to choose **Creeping flow (Stokes)** whenever justified.

For **Creeping flow (Stokes)**, **Periodic,** and Velocity in - Pressure out (**VinPout**) can be chosen as **Boundary Condition** in flow direction. VinPout cannot be chosen for a **Single Pass – Constant Pressure Drop** experiment.

For **Fast Flow (Navier-Stokes)**, the boundary conditions in flow direction are set to Velocity in, Pressure out (**VinPout**) if **Constant Flow Rate** was chosen under the Filter Experiment tab, and set to **Periodic** if **Constant Pressure Drop** was chosen under the Filter Experiment tab.

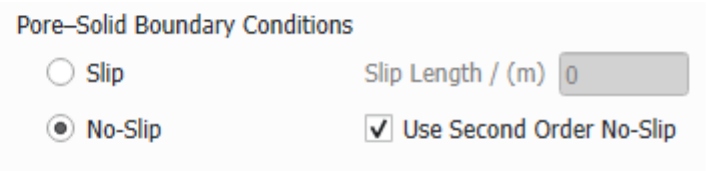
In **Tangential Directions**, **Periodic** or **Symmetric** boundary conditions can be selected. For more information about the flow computation, please refer to the FlowDict user guide.

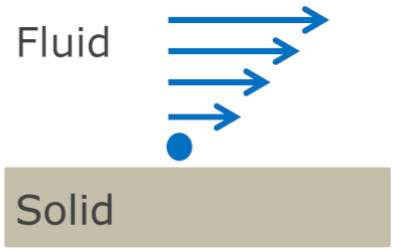
The choices made for the flow boundary conditions in tangential directions also determine what happens when a particle reaches the domain boundary in one of the tangential directions. With **Periodic** boundary conditions, the particle will leave the domain and reappear on the opposite site. With **Symmetric** boundary conditions, a particle will be reflected at the domain boundary.

Pore-Solid Boundary Conditions

The **Slip Length** allows to include sliding effects in the flow simulation. Typically, the slip length is of the same order of magnitude as the mean free path of the fluid (which is, e.g., 68 nm for air at ambient conditions) and can be neglected if this is much smaller than the voxel length. It becomes relevant when simulating air flow around nanofibers.

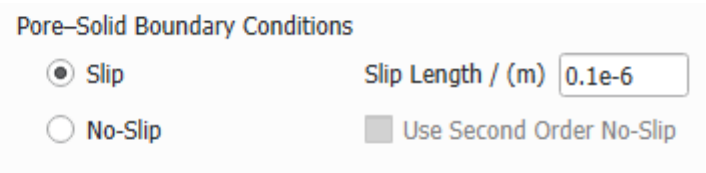
The default setting **No-Slip** corresponds to a flow velocity of zero along the structure.

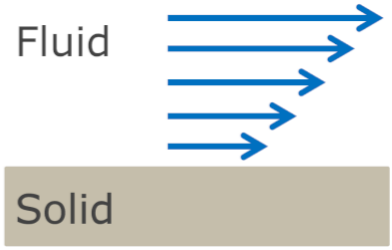




Use Second Order No-Slip sets the tangential velocity to zero at the center of the voxel surfaces and uses a second order approximation of the velocity field. The first order setting, which is used when the box is left unchecked uses a linear approximation. **Second Order No-Slip** will increase precision when there are narrow pores in the structure.

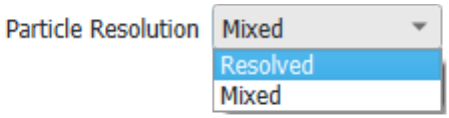
Slip with a non-zero **Slip Length** simulates the sliding of the fluid along the solid, increasing the fluid mean flux and thus, the permeability of the structure. This option might be used when it is realistic for a given physical material. Currently, the same slip length value must be set for all materials in the structure.





Constituent Materials

On the top of this tab, select if the **Particle Resolution** is **Resolved** or **Mixed**.



Choose **Resolved** if all particles are larger than the current voxel length, and **Mixed** if some particles are smaller than the current voxel length. For further explanation, see Filter Clogging 14 topic page.

Temperature

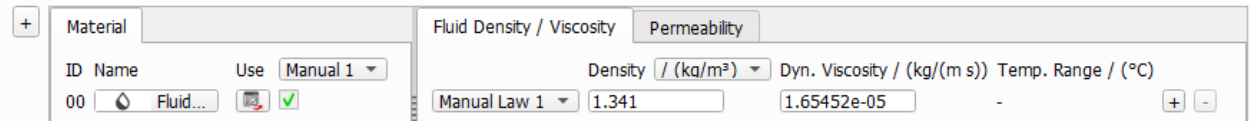
The **Temperature** for the filtration process is selectable in Kelvin (K), Celsius (°C), and Fahrenheit (°F). The chosen temperature may change the density and viscosity values of the fluid during the simulation, if those are defined to be temperature-dependent in the material database. Furthermore, if the Brownian motion of the particles is included in the simulation, it determines the strength of the Brownian motion through equation (13) 8.

Fluid Density and Viscosity

FilterDict uses single-phase flow simulations. It is not possible to run a flow simulation if two fluids are present in the structure. If multiple fluids are present in the structure, you may click **Set all Fluids to...** and select one fluid for the simulation.

The selected fluid is also displayed in the drop-down menu in the **Material** subtab behind **Use**. The density and the dynamic viscosity of this fluid are shown under the **Fluid Density/Viscosity** subtab.

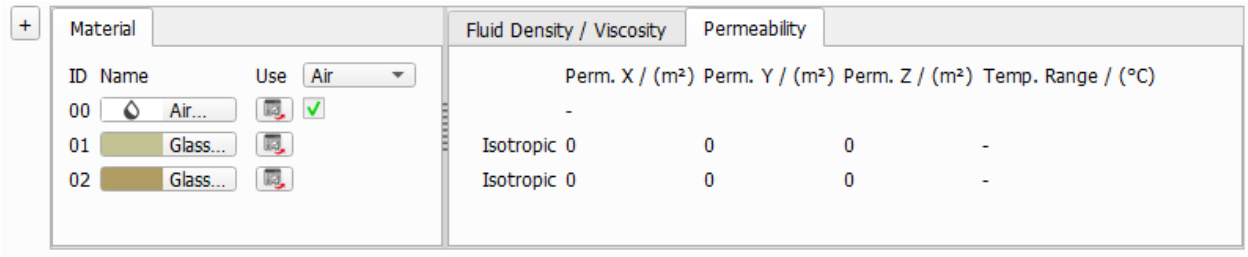
If a fluid from the GeoDict Material Database is used, the values for **Density** and **Dynamic Viscosity** are taken from the database and may depend on the given **Temperature** value. Select **Manual** as fluid material to enter parameter values manually.



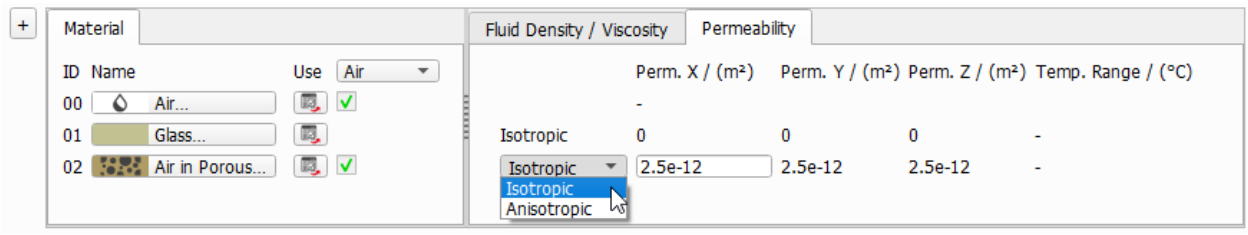
To add materials to the database, use the **Open & Edit Material Database** button. More information on editing, expanding, and using the **GeoDict Material Database** is available in the Material Database user guide.

Filter Media Permeability

Under the **Permeability** subtab, enter the permeability of porous materials present in the structure. As long as there are only fluid and solid materials present in the structure, no parameters are needed here:



For porous materials, choose if the material is **Isotropic** or **Anisotropic**. In the isotropic case, the permeability is the same in all space directions. In the anisotropic case, enter a different permeability for each spacial direction.



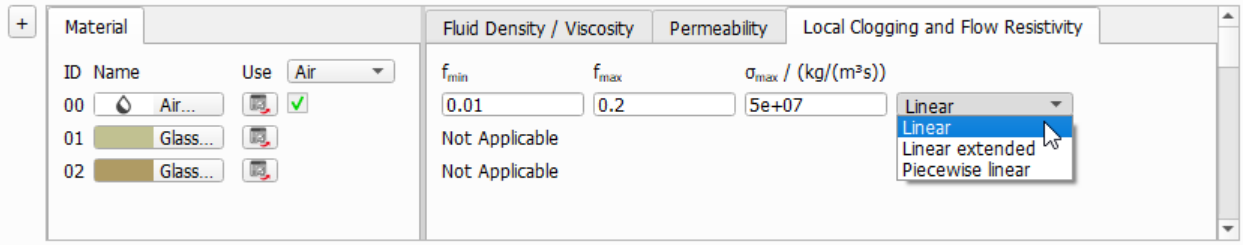
You can select and change a material by clicking on the material button.

If porous materials are present in the current 3D structure, the **Use Pass-Through Model** check box on the Particles Movement tab 28 must be switched on, too. This option enables particle movement through porous layers and particle deposition in such layers.

Local Clogging and Flow Resistivity

If the unresolved particle model is selected, the **Local Clogging and Flow Resistivity** tab will also be available.

Unresolved particles will create porous voxels when deposited into the 3D structure. In those voxels, flow is still possible, but experiences an increased flow resistivity. In the **Local Clogging and Flow Resistivity** tab, the relation between the dust volume fraction f inside of a voxel and the flow resistivity σ must be defined. This tab is only available for unresolved simulations, because in the **Resolved** case all voxels are treated as either empty or solid.



The flow resistivity σ depends on the values for $f_{\min}$, $f_{\max}$, and $\sigma_{\max}$, which must be defined for every material of the structure which can be filled by dust particles during the simulation, i.e., inside of the fluid (where the filter cake will form) and in all porous layers.

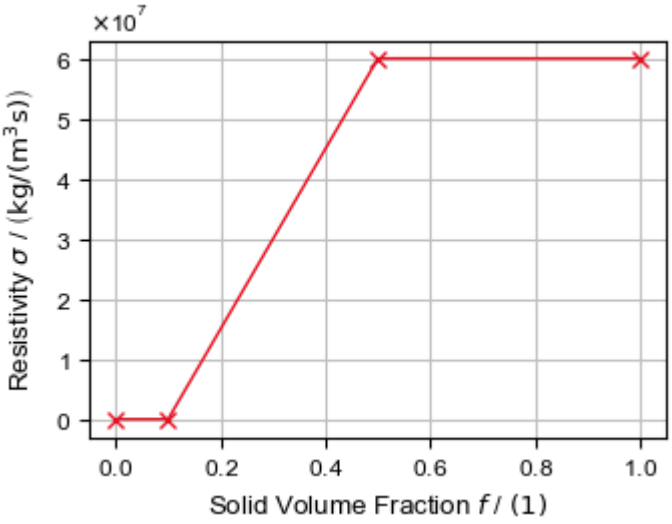
Note! It is necessary to set these parameters for the fluid material, although this might be unintuitive at first glance. Be aware that the parameters set for the fluid material describe the behavior of the filter cake because the filter cake consists of partially dust-filled voxels of the fluid domain.

Note! It is not possible to use different fluids in a structure, but it is possible to use different material IDs with the same fluid. This option is useful, when the filter cake should have different properties depending on its location in the structure.

Three different models are available to define the function $\sigma(f)$: **Linear**, **Linear extended**, and **Piecewise linear**.

Linear

In this model, the flow resistivity depends linearly in a certain interval on the local solidity.



The function $\sigma(f)$ is defined through:

$$\sigma = \begin{cases} 0 & \text{for } f < f_{\min} \\ \dfrac{f - f_{\min}}{f_{\max} - f_{\min}} \cdot \sigma_{\max} & \text{for } f_{\min} < f < f_{\max} \\ \sigma_{\max} & \text{for } f_{\max} < f < 1 \end{cases} \tag{38}$$

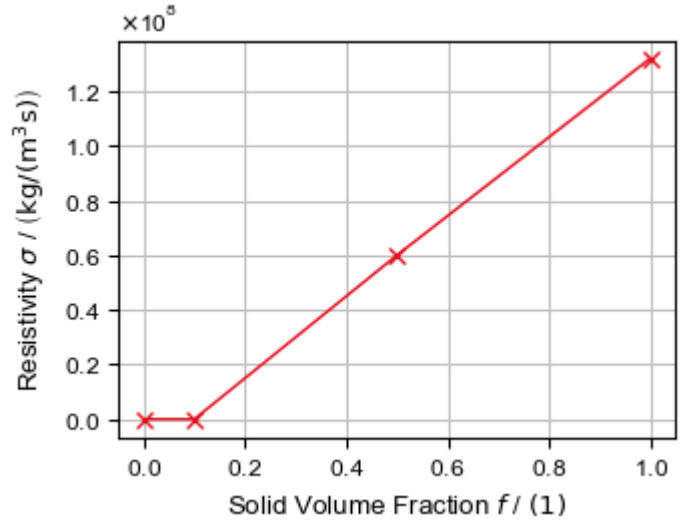
Know how! The general idea of this model is that $f_{\max}$ describes the maximal solidity (or Maximum Particle Packing Density) of the filter cake and $\sigma_{\max}$ the flow resistivity of the cake.

Under the assumption that all particle sizes are smaller than the voxel length (which holds true in many settings that consider a complete filter), using the solidity of the filter cake for $f_{\max}$ and the Maximal Particle Packing Density and using the flow resistivity as $\sigma_{\max}$ will lead to the expected results: The filter cake that emerges in the simulation will again have this solidity and flow resistivity.

If a significant number of particles are larger than the voxel length, the filter cake that emerges during the simulation will have a different solidity than $f_{\max}$ and a different flow resistivity as $\sigma_{\max}$. This has been described by Becker et al. 2016[257], and the paper also describes a way to estimate those parameters in this case.

Linear Extended

This model is basically the same as the **Linear** model, but the function is not cut off at the value of $\sigma_{\max}$, but extended linearly until $f = 1$.



The function $\sigma(f)$ is defined through:

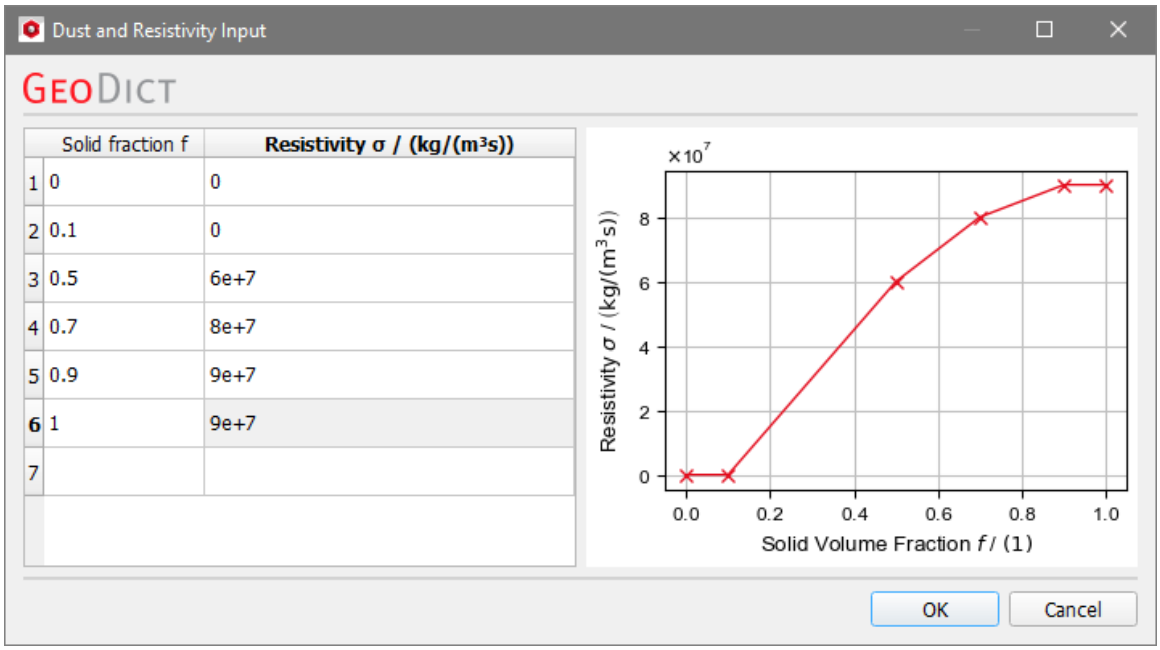
$$\sigma = \begin{cases} 0 & \text{for } f < f_{\min} \\ \dfrac{f - f_{\min}}{f_{\max} - f_{\min}} \cdot \sigma_{\max} & \text{for } f_{\min} \le f < 1 \end{cases} \quad (39)$$

Know how! The local solidity inside of a voxel can exceed the Maximum Particle Packing Density after the volume of the last particles deposited in this voxel is added. In fact, any value between 0.0 and 1.0 may appear during the course of a simulation. Therefore, it makes sense to extend the linear model beyond $\sigma_{\max}$.

Piecewise Linear

If **Piecewise linear** is selected, an arbitrary piecewise linear function can be defined by clicking the **Edit...** button.

The **Dust and Resistivity** Input dialog open and allows to define a piecewise linear function:

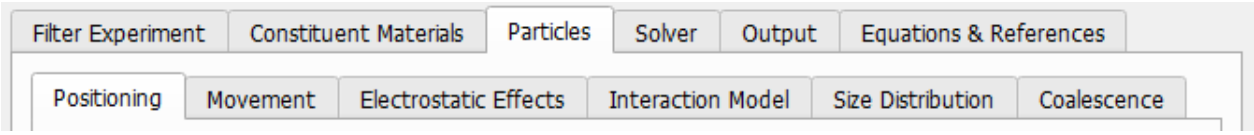


It is recommended to start the list with a solid fraction of 0, and end with a solid fraction of 1, to define the function for the whole interval [0,1).

Be aware that for a local solidity of 1.0, the voxel is always treated as solid, and no flow is possible. This is independent from the definition of the function $\sigma(f)$, which only applies to partially filled voxels (local solidity $f < 1$).

Particles

Under the **Particles** tab, six sub-tabs are available to define **Positioning**, **Movement**, **Electrostatic Effects**, **Interaction Model**, **Size Distribution**, and **Coalescence**.



The **Electrostatic Effects** tab is not available if porous materials are present in the structure. In this case it is grayed out.

The tab **Coalescence** appears only if this feature is separately licensed.

Tabs of the Particles tab

- Positioning 70
  Define the start positions of the simulated particles.
- Movement 75
  Select the parameters modelling the motion of the particle.
- Electrostatic Effects 77
  Switch electrostatic effects on or off and select a decay model.
- Interaction Model 80

Tabs of the Particles tab

Define collision and adhesion models for particles-material collisons.

- Size Distribution 83
  Define the particle size distribution and all particle size dependent parameters.
- Coalescence 85
  Simulate liquid-liquid filtration with a coalescence model.

Positioning

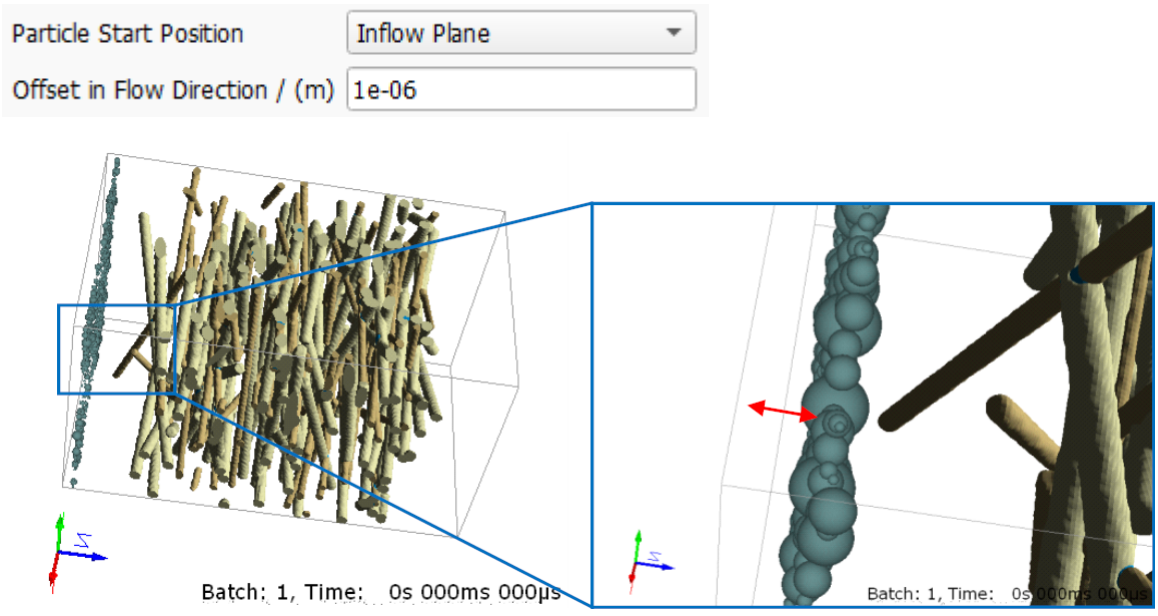
Particle Initial Position

Particle Start Position

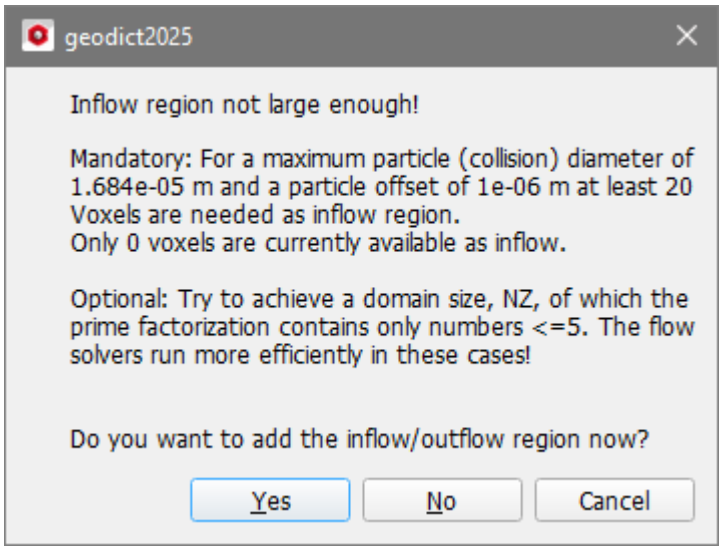
Select one of the following **Particle Start Position** types. The default start position is the Inflow Plane.

Inflow Plane

If **Inflow Plane** is chosen as **Particle Start Position**, the **Offset in Flow Direction** defines the initial distance of the particles to the boundary of the inflow region.

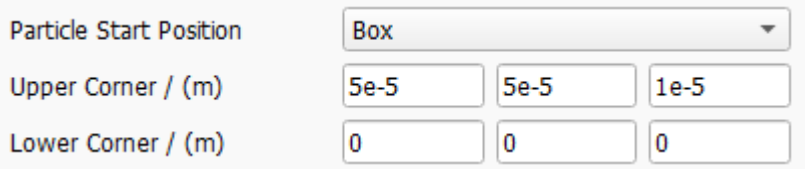


If the inflow region is too small and the particles cannot be placed in the inlet without intersecting the solid material, a warning appears after clicking **Run** to start the simulation in the **FilterDict** section.



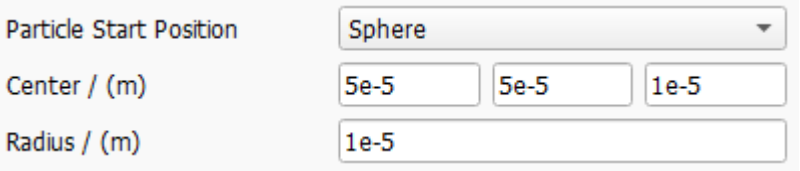
Box

If **Box** is chosen as **Particle Start Position**, two opposite corners of a rectangular box must be defined. All particle start positions will then be inside of this box.



Sphere

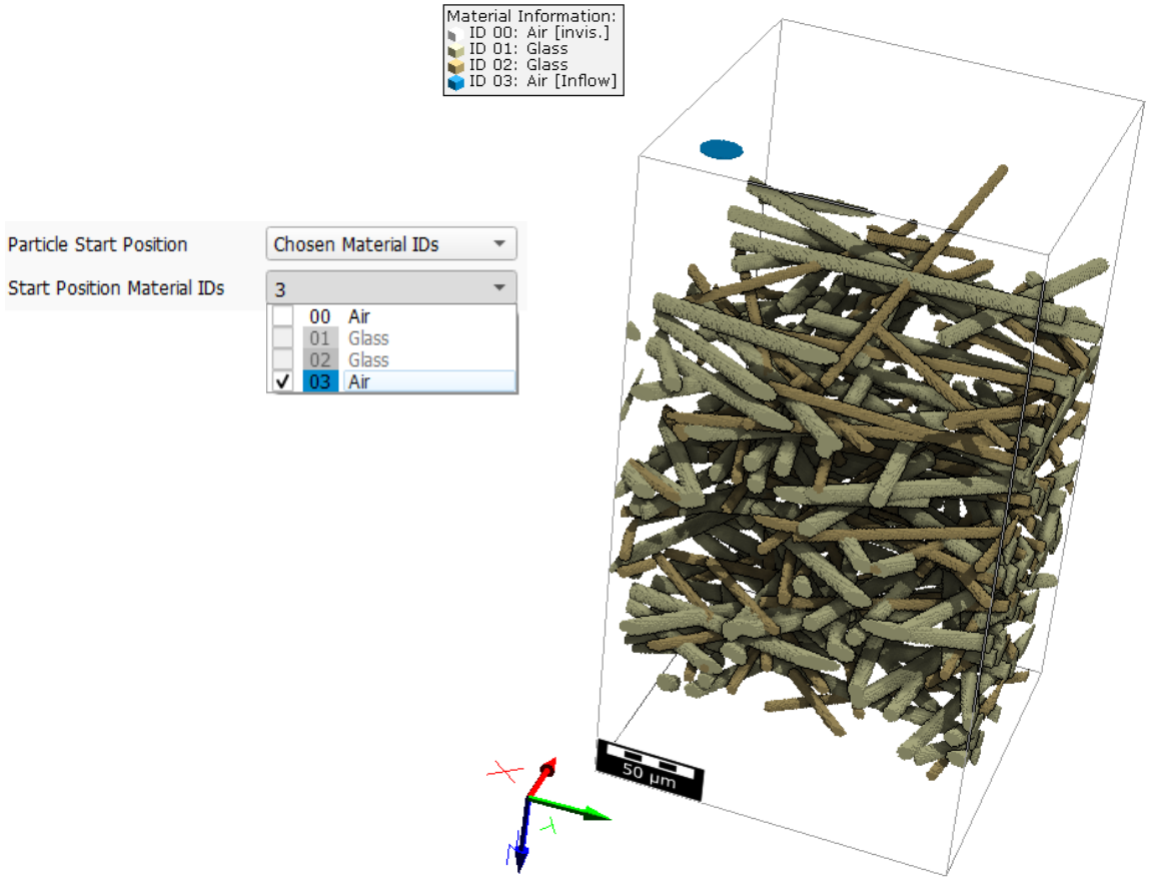
If **Sphere** is chosen as **Particle Start Position**, the center and radius of the sphere must be defined. All particle start positions will then be inside of this sphere.



Chosen Material IDs

An arbitrary location can be set by selecting **Chosen Material IDs** as **Particle Start Position**.

For example, by using GadGeo, various objects can be added into the inflow area. Those objects can be assigned to the same fluid material as the surrounding pore space, so they only serve as markers of a certain area (see picture below).



If this area ID is now chosen as **Start Position Material ID**, all particles start in this area. It is possible to mark several areas in this way and/or to choose several IDs as starting areas.

Everywhere

If **Everywhere** is chosen as **Particle Start Position**, particles may start everywhere in the pore space of the structure.

Particle Positioning Weights

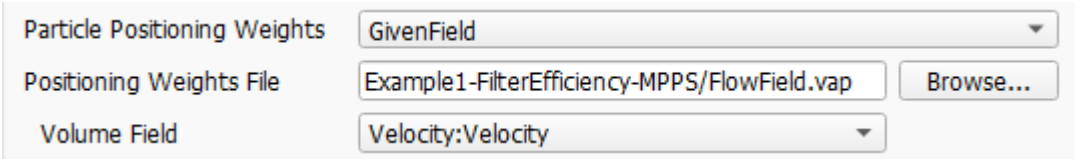
Particle start positions are scattered randomly inside the defined **Particle Start Position** area or volume.

By default, those positions are uniformly distributed. Be aware that a spatially **Uniform** placement in the inlet will lead to an increased particle concentration in areas with low fluid velocity, e.g., in proximity to a surface or a walls.

If **Velocity** is selected, the starting probability is based on the local fluid velocity. Selecting this option will lead to a uniform particle concentration in the inflow.

If **Distance** is selected, the starting probabiliy is based on the distance from the next solid object. This will lead to higher particle numbers in the core flow, while there will be less particles starting next to the surface.

With **GivenField**, it is possible to load an arbitrary 3D field of scalar values to define the starting probability. For example, it is possible to load the flow field of a previous computation and select the velocity or one of its components.



Particle Start Velocity

Select either **Fluid Velocity** (the default option, which was used until GeoDict 2025) or **Given Velocity** as the **Particle Start Velocity**.

With the first option, particles begin moving with the fluid velocity at their start position.

With the second option, all particles start with the same user-defined velocity.

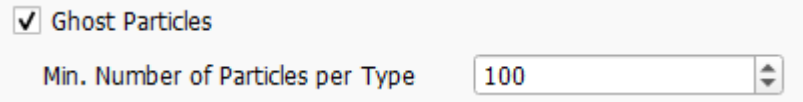
Random Seed

Particles to be filtered are placed randomly in the area according to the selected **Particle Start Position**. The parameter **Random Seed** controls the underlying random number generator. The same random seed produces identical results, whereas results with different random seeds are similar but not identical.

Ghost Particles

FilterDict simulations usually consider only a small cutout of the filter media and typical test dusts contain only a very small number of large particles. Together, this leads to the effect that

the arrival of a large particle is not very likely and does not happen in every batch, which makes it impossible to compute a fractional filtration efficiency per batch with statistical relevance.



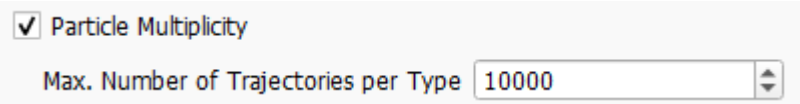
Ghost Particles can be added to help to increase the statistical reliability of the efficiency results in each batch. They are tracked like ordinary particles and counted as filtered or unfiltered. Afterwards, they disappear, so that they do not clog the filter or influence the results of later time steps (batches) in any other way.

After checking **Ghost Particles**, choose the **Min. Number of Particles per Type** and batch.

For example, in the first batch, 300 real particles of 1 µm and 30 real particles of 5 µm are simulated. Enough real particles of size 1 µm are already present, so that when entering 100 as the minimum number of particles per type, 70 ghost particles of size 5 µm are added to the simulation. The fractional filtration efficiency of particles of size 5 µm is then computed using all 100 simulated particles (30 real particles + 70 ghost particles).

Particle Multiplicity

When the Particle Resolution is set to **Mixed** in the Constituent Materials tab, the parameter **Particle Multiplicity** becomes available and a value can be entered in the **Max. Number of Trajectories per Type** box.



Unresolved particles may be much smaller than the used voxel length and thousands of dust particles are needed to fill a single voxel. Therefore, the number of **Particles per Batch** (on the **Solver tab**) may grow very large, leading to increased simulation times. With **Particle Multiplicity**, the number of particles that are tracked can be restricted.

When, for example, 1,000,000 particles of diameter 0.05 µm are to be simulated in the next batch, and you check **Particle Multiplicity** and set the **Max. Number of Trajectories per Type** to 10,000, only the movement of 10,000 particles is tracked. However, each of these particles represents 100 particles, and the multiplicity is 100 (100 x 10,000 = 1,000,000). If one of these particles is filtered, the mass and volume fractions of 100 particles are added to the deposited dust.

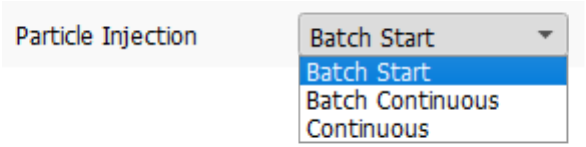
To make sure that the choice of **Particle Multiplicity** does not introduce large numerical errors, FilterDict ensures that the particle volume multiplied by its multiplicity is less than 5% of the single voxel volume.

Reflect particles at inflow plane

If diffusion through Brownian motion is simulated, some particles might diffuse against the flow direction, reach the inflow region and exit the domain. Check **Reflect particles at inflow plane** to avoid the exit of such particles.

Particle Injection

All particles simulated in a time inteval (a batch) move independently from each other. Select, if all particles should start moving at the beginning of the time interval (**Batch Start**) or if starting times should be distributed uniformly over the batch interval (**Continuous**).

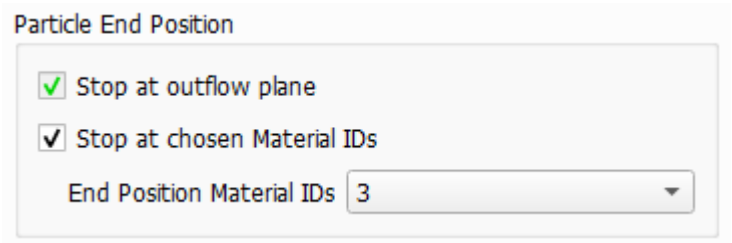


While **Continuous** offers a physically correct description of what happens in a real filter experiment, it may make a later visualization of the particle movement almost impossible. In a typical air filter media simulation, a batch interval may last minutes, while particles move with a velocity of several meters per second through a filter material of less than a millimeter thickness, which means that at most observation points in time, no moving particles can be seen. This can be overcome by letting all particles start at **Batch Start**, or distributed over a short time interval at the beginning of the batch (**Batch Continuous**).

The selection made here has very little influence of the computed results, because in any case all particles of the same batch do not interact with each other. They cannot collide in flight, and they cannot deposit on top of each other. They can only interact with the particles deposited in previous batches.

Particle End Position

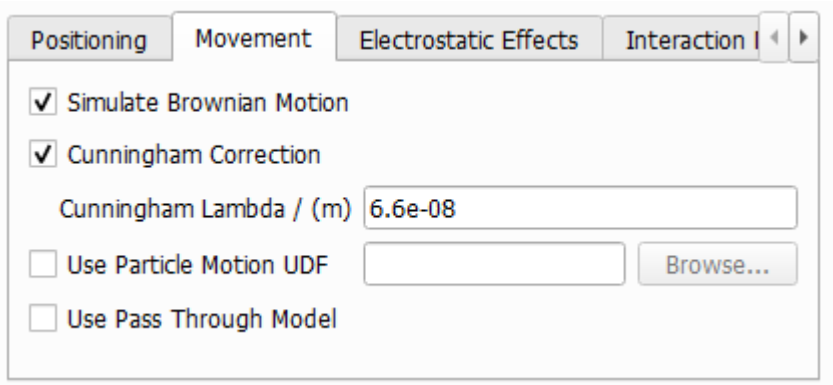
Particles reaching the outflow plane are always considered as unfiltered.



Similarly to the definition of the particle start position, an additional area can be marked. Particles that arrive in this area will also stop moving and be counted as unfiltered.

Movement

The parameters in the **Movement** panel define the simulation variables for the particle tracking.



Brownian Motion and Cunningham Correction

Checking **Simulate Brownian Motion** activates the simulation of random motion in particle movement. If this option is unchecked, the simulation is run with diffusivity $D = 0$.

If **Cunningham Correction** is checked, equation (12) 8 is used to calculate C_c. Otherwise, $C_c = 0$ is used.

The **Cunningham Lambda** value is the mean free path λ used in equation (12) 8. Make sure, that you use the Cunningham Correction with the default mean free path only in air filtration. Inside of water or oil, the mean free path is much smaller, and there is no need to take the Cunningham Correction into account.

Use Particle Motion UDF

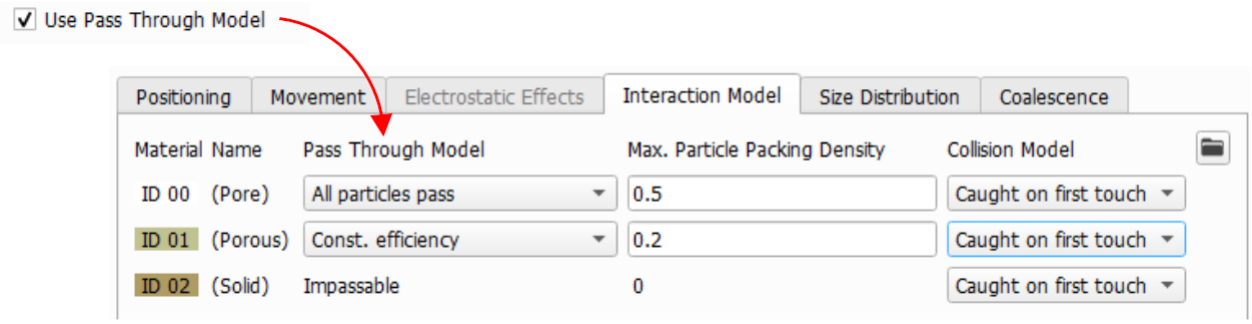
Check **Use Particle Motion UDF** to replace equations (11) 8, (12) 8, (13) 8, and (14) 8 with your own definitions. Refer to the Particle Motion UDF 239 page for details on how to modify and compile user-defined functions.

Use Pass Through Model

If porous materials are present in the current 3D structure, you must check the **Use Pass-Through Model** box. This option enables particle movement through porous layers and particle deposition in such layers.

If no porous materials are present in the current 3D structure, select the **Use Pass-Through Model** option only if you want to define a pass-through model for the filter cake.

When this option is switched on, the **Pass Through Model** column becomes available in the Interaction Model 80 subtab.



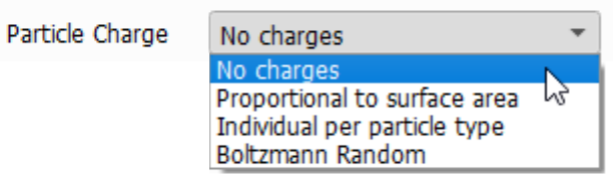
Important! Be aware that you cannot use a **Pass Through Model** and simulate **Electrostatic Effects** at the same time. Simulation of electrostatic effects requires that all media are resolved!

Electrostatic Effects

Important! This tab is only available if there are no porous materials present in the 3D structure.

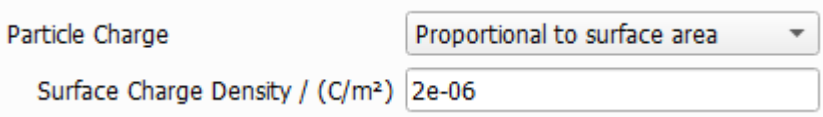
Check **Include Electrostatic Effects** to simulate electrostatic effects between filter media and particles in the fluid.

Particle Charge



If electrostatic effects are negligible, set **Particle Charge** to **No charges**.

Select **Proportional to surface area** and enter a **Surface Charge Density** to set the value of Q in equation (10) to the product of the entered **Surface Charge Density** and the particle surface area.



Select **Individual per particle type** to enter an individual particle charge for each particle type or size. In this case Q in equation (10) is the entered particle charge. Enter the desired values in the **Particle Charge [C]** column that is automatically added to the table under the Size Distribution subtab.

If **Boltzmann Random** is chosen, particle charges follow a normal distribution with mean value 0 and standard deviation

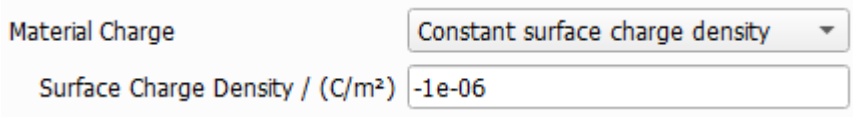
$$\sigma = \frac{\sqrt{2\pi d \varepsilon_0 k_B T}}{e} \tag{40}$$

where d denotes the particle diameter, ε_0 the dielectric constant, k_B the Boltzmann constant, T the temperature and e the electron charge.

Note! Electrical charges appear only as multiples of the elementary electric charge. Since GeoDict 2025, the **Boltzmann Random** distribution is therefore discretized into elementary electric charges. For large particles, this approaches a normal distribution of charge values, whereas small particles are mostly uncharged or carry just a single positive or negative charge.

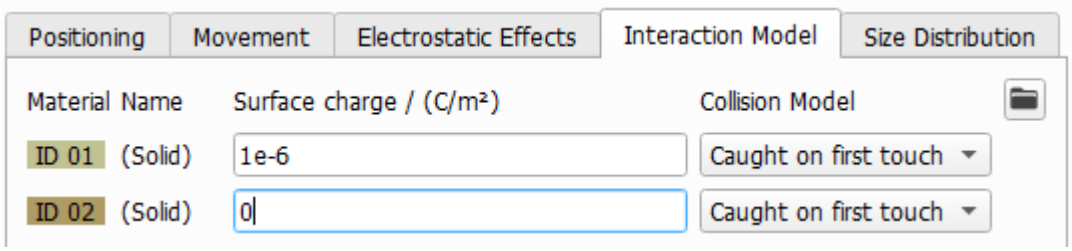
Material Charge

Select **Constant surface charge density** and set the **Surface Charge Density** (C/m²), which corresponds to ξ in equation (16).



This charge density is applied to all solid material surfaces.

Select **Individual per material ID** to set different surface charge densities for different material IDs. If selected, an additional column **Surface Charge (C/m²)** appears on the Interaction Model tab. Enter the surface charge density of each material ID there.



Since the charges are given "per surface area," it is especially important to consider how surface area is taken into account in the simulation. GeoDict uses a voxel grid. Simply applying the surface charge to each voxel surface would overestimate the overall charge because the sum of all voxel surfaces overestimates the true surface area of the material. hus, the surface charge density is corrected using the ratio of the material's computed surface area (see the **Estimate Surface Area** command in the MatDict user guide) to the sum of the voxel surfaces.

Note! Until GeoDict 2022, particles and filter were assumed to have opposite charges. Entering positive surface charge values for filter and particles lead to the filter attracting the particles. This was unintuitive and did not reflect the correct physics.

Therefore, since GeoDict 2023, it is no longer assumed that particles and filter have opposite charges. Entering opposite sign charge values (e.g., 5e-7 C/m^2 as surface charge and -6e-7 C/m^2 as material charge) for filter and particles leads to the filter attracting the particles. In this case, the filter efficiency is enhanced. Entering same sign charge values leads to the filter repulsing the particles and a lower filtration efficiency.

Enable Dielectrophoresis

Dielectrophoresis is a phenomenon in which a force $\vec{F}$ is exerted on a particle when it is subjected to a non-uniform electric field $\vec{E}$. This force does not require the particle to be charged. All particles exhibit dielectrophoretic activity in the presence of electric fields.

When dielectrophoresis is enabled, the force $\vec{F}$ in equation (10) is computed as

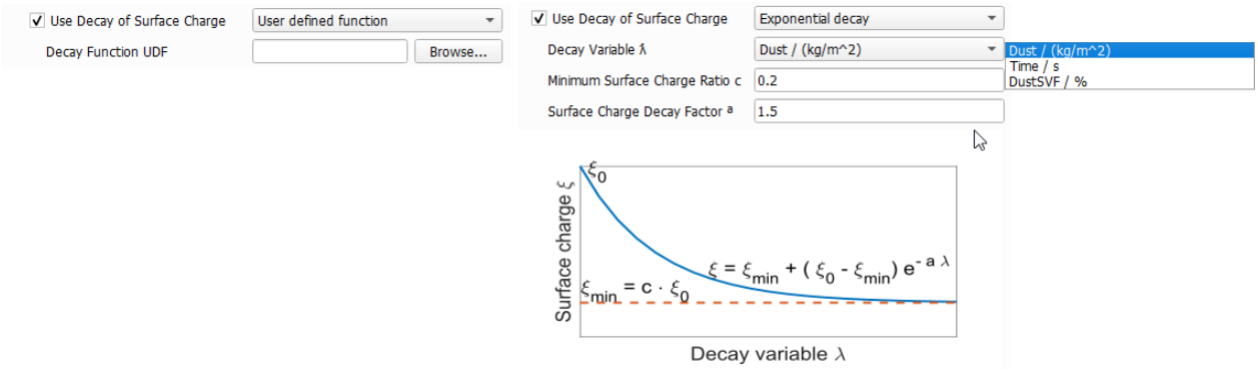
$$\vec{F} = 2\pi R^3 \varepsilon_m \left(\frac{\varepsilon_p - \varepsilon_m}{\varepsilon_p + 2\varepsilon_m} \right) \nabla |\vec{E}|^2 \tag{41}$$

Here, ε_p is the relative permittivity of the particles, ε_m is the relative permittivity of the fluid, and R is the particle radius. $\nabla|\vec{E}|$ denotes the gradient of the electric field and dictates the direction of the dielectric force.

Switch on this effect by marking the check box, and enter the relative permittivity values ε_m and ε_p in the dialog:

Decay of Surface Charge

The initial material charge may decay during the filter lifetime. It is possible to use a simple decay model in GeoDict to model the reduction of the surface charge over time.



You may either select to use a [user-defined function for the decay](253) or use a built-in simple decay formula.

The decay of the charge ξ follows the equation

$$\xi = \xi_{\min} + (\xi_0 - \xi_{\min}) \cdot e^{-\alpha\lambda} \tag{42}$$

Thus, the initial charge ξ_0 is eventually reduced to

$$\xi_{\min} = c \cdot \xi_0 \tag{43}$$

To fit the decay model, select the **Minimum Surface Charge Ratio** c, which should be a number between 0 and 1. Select further, if the **Decay Variable** λ is the amount of deposited **Dust** (in kg/m^2), the **Time** (in s), or the **Dust Solid Volume Fraction** (in %). The **Surface Charge Decay Factor** determines how quickly the charge decays.

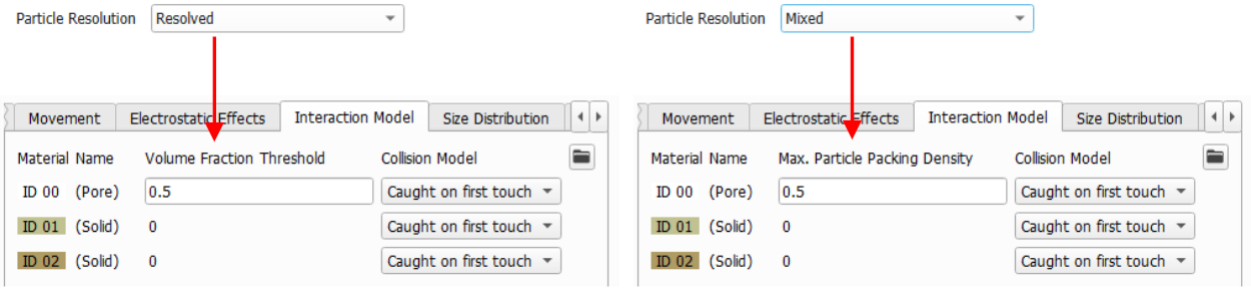
Interaction Model

For Filter Lifetime simulations, a **Collision Model** must be chosen for collisions with solid materials and with particles that have been deposited in previous batches (the filter cake). Those particles are deposited in the pore space, therefore the **Collision Model** defined in the pore space is the model that applies for the filter cake.

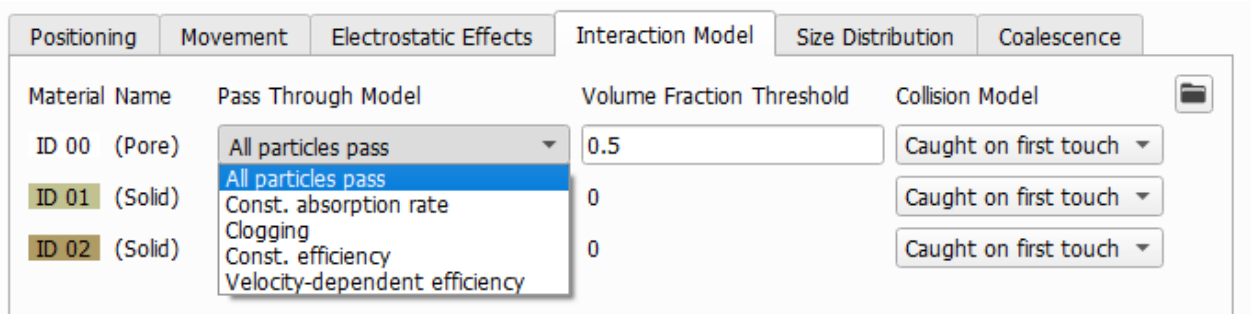
In **Resolved** simulations, every voxel that exceeds the **Volume Fraction Threshold** is treated as solid material and arriving particles will collide with it using the defined collision model.

In **Mixed** (unresolved) simulations, every voxel that exceeds the **Max. Particle Packing Density** is treated as impassable and arriving particles will collide with it using the defined collision model.

See the [theoretical background](10) for more information on the selectable collision models **Caught on first touch**, **Hamaker**, and **Sieving**.



If **Use Pass Through Model** is selected on the Movement tab 75, a **Pass Through Model** must be selected for the pore space and all porous materials.



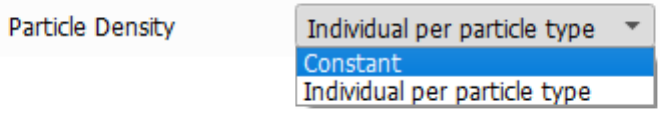
See the theoretical background 11 for more information on the selectable absorption (or pass through) models **All particles pass**, **Const. absorption rate**, **Clogging**, **Const. efficiency**, and **Velocity-dependent efficiency**.

The user interface can be understood as follows: The **Collision Model** on the right applies only after the given **Volume Fraction Threshold** or **Max. Particle Packing Density** has been reached. If the material is solid, the **Max. Particle Packing Density** or **Volume Fraction Threshold** is fixed to 0, and the **Collision Model** applies directly.

On the Constituent Materials tab 64, you may select multiple pore materials for a structure. This allows to use different packing densities models in different zones of the structure.

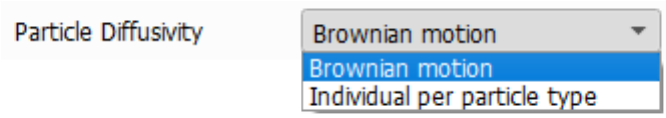
Particle Density

If **Individual per particle type** densities are chosen, a new column called **Particle Density** is added to the table under the Size Distribution tab, and an individual density can be entered for each particle type or size. Otherwise, the **Particle Density** entered here is used for all particle sizes and types.



Particle Diffusivity

If **Individual per particle type** is chosen, a new column called **Difusivity in Pore** is added to the table under the Size Distribution tab, and an individual diffusivity can be entered for each particle type or size.



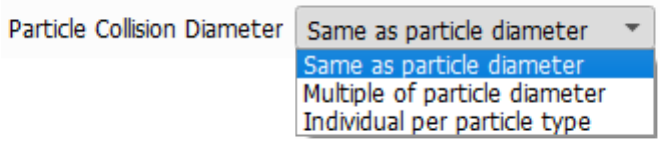
For **Brownian Motion**, the particle diffusivity is computed for each particle type according to equation (13).

If **Simulate Brownian Motion** is switched off on the Movement tab, only **None** is selectable here.

If **Use Particle Motion UDF** is selected on the Movement tab, the choices made here have no effect, as the formula implemented in the UDF will be used.

Particle Collision Diameter

The particle **Diameter**, as given in the table under the Size Distribution tab, is used when tracking the particle in the flow field. It determines the particle radius R used in equations (10), (11) and (12) to determine the mass and the friction.

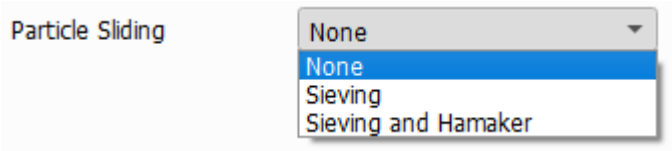


The **Particle Collision Diameter** is used to check if the particle collides with the filter structure and should usually be set to **Same as particle diameter**. With this selection, the particle is consistently modeled as a sphere.

However, real dust particles may exhibit different, possibly less compact, shapes. Thus, they may have a higher probability of colliding with the structure than a spherical particles of the same mass. By setting the **Particle Collision Diameter** to a different value than the **Diameter**, you can model this effect. It is possible to set a fixed **Diameter factor** for all particles sizes by choosing **Multiple of particle diameter** or to select **Individual per particle type** and enter the collision diameter in the Size Distribution tab for all particle types.

Particle Sliding

Particles that collide with the surface loose some of their energy if the restitution factor is smaller than 1. In certain pore geometries, the flow may move a particle very close to a pore surface. This causes many consecutive hits, and repeatedly applying the restitution factor will stop the particle's motion. This behavior is often undesired, and it may overestimate the filter efficiency when combining a **Sieving** collision model with a relatively low restitution value.

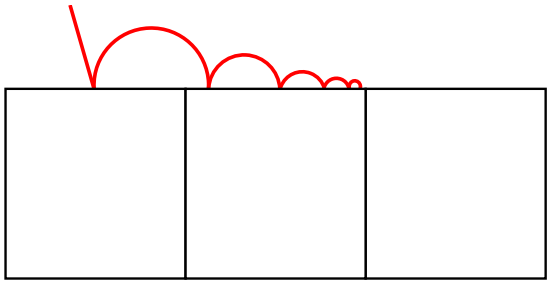
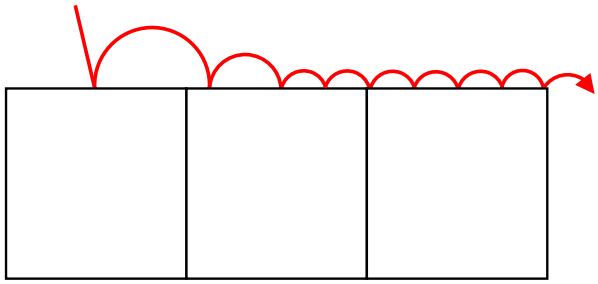


When **Sieving** is selected in the drop-down menu, all **Sieving** particle-wall collisions are

modified, when **Sieving and Hamaker** is selected, all **Sieving** particle-wall collisions are modified and all **Hamaker** particle-wall collisions are modified. **Caught on First Touch** particle-wall collisions are not modified.

Enabling **Particle Sliding** for the **Sieving** or **Hamaker** collision models enlarges the tangential restitution factor when a particle hits a surface in a location that lies close to the last surface collision.

The first bounce on the wall is always treated the same, independent whether particle sliding is active or not: A particle arrives with a certain velocity, loses energy on impact, and bounces back again. The strength of the bounce-back is controlled with the given **Restitution**. When particle sliding is active, further impacts near the first hit are treated differently: Only the momentum perpendicular to the wall is reduced. The velocity along the wall is not reduced further, so in this direction the restitution is equal to 1. This means that particle movement along the wall is no longer slowed down. The normal restitution remains unchanged. In effect, the particle will slide along the surface.

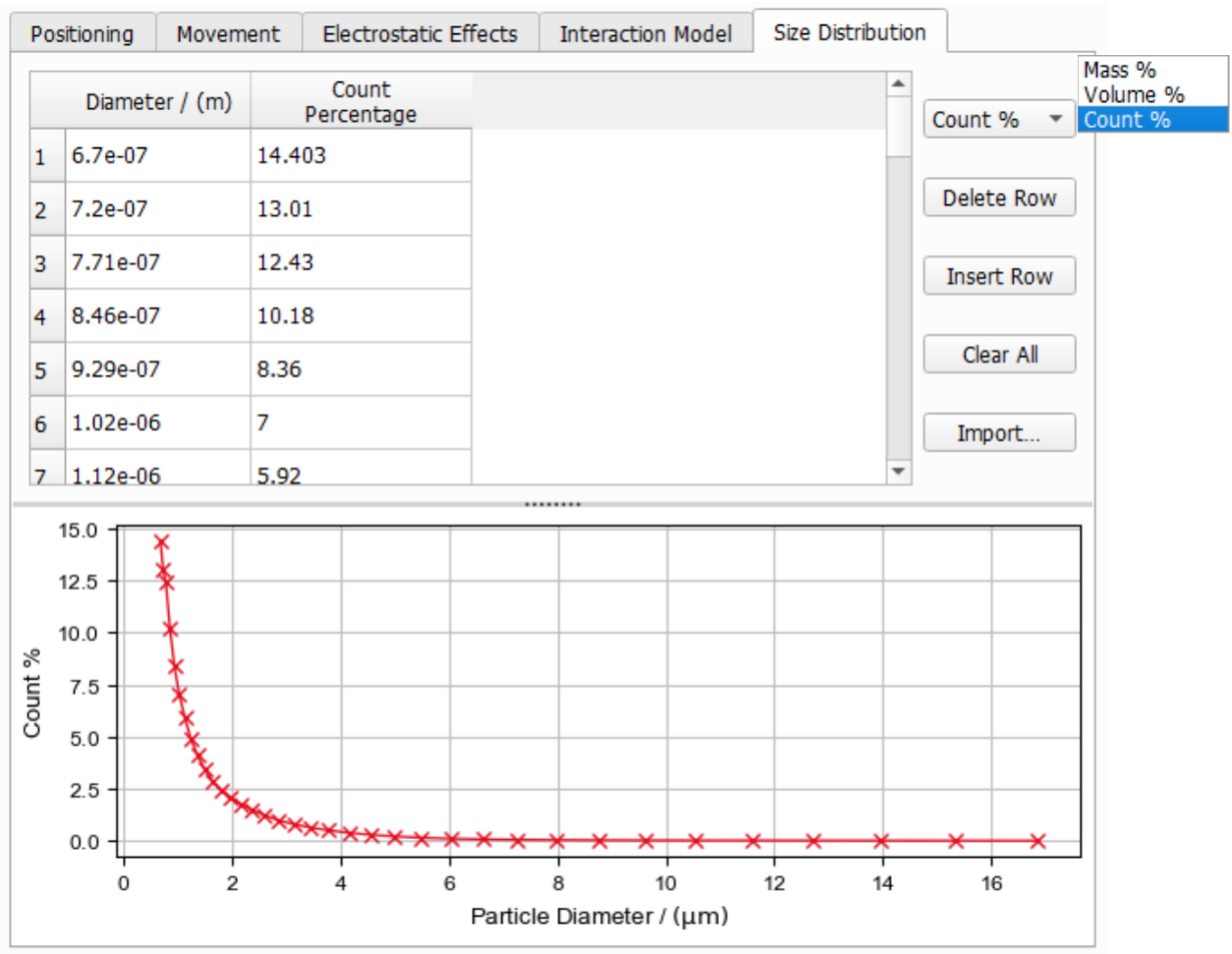
Without particle sliding:	With particle sliding
• energy lost at every collision • particle caught on surface	• energy conserved after some hits • particle moves along surface
	

In setups where particles are sieved by filter materials having pores of well-defined sizes (e.g., in meshes, nets or weaves), selecting particle sliding has a great influence on the computed filter efficiency and pressure drop and it is recommended to select this option.

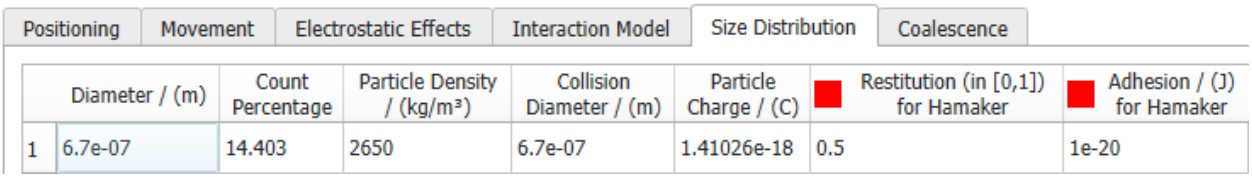
In setups where particles are mainly caught by adhesion or where the filter material consists of irregular pores of many different sizes (e.g., nonwovens), the choice of the particle sliding model has little to no effect on the simulation results.

Size Distribution

The **Size Distribution** tab contains a table with the particle size distribution. The table consists of at least two columns. The first column lists the **Diameter** of the particles. The second column contains the particle distribution. You can select if the percentages describe **Mass %**, **Volume %**, or **Count %** through the upper right pull-down menu.



Additional columns appear depending on the choices made under the Interaction Model subtab:



These columns are:

- **Restitution** for materials with Sieving collision model.
- **Restitution** and **Adhesion** for Hamaker material.
- **Particle Density** column for Individual per particle type densities.
- **Particle Charge** column for Individual per particle type charges.
- **Collision Diameter** column for Individual per particle type collision diameters.

For user-defined collision models, the number of material parameters to be entered in the table is specified in the UDF.

By clicking on a column heading, all values in the selected column can be set to the value in the first row. The entries in the table do not need to be sorted.

Click **Delete Row** to delete the currently selected row. Click **Insert Row** to insert a row below the currently selected row. Click **Clear All** to remove all rows.

Click **Import...** to import data from an ASCII text file. Some example particle distributions are provided with GeoDict and can be found in the installation folder (Program Files/Math2Market GmbH/GeoDict2026/FilterDict).

More convenient than importing is often to simply copy & paste entries from an Excel table into GeoDict, which is supported by the size distribution table. To copy a table from Excel, first clear the size distribution table by clicking **Clear All**. Then, select the cells in Excel and press Ctrl+C. Then, click into the upper left cell in the size distribution table and press Ctrl+V. This works also if more than two columns are present.

If the entered percentages do not add up to 100%, a warning message pops up when closing the dialog. You can select to normalize the values automatically or to return to the dialog and adjust the percentages manually.

Coalescence

This feature is only available with a separate license. Ask support(at)math2market.de if you are interested in obtaining a license.

If the deposited particles are not solid, but liquid aerosols, they will not form a porous filter cake. Oncoming particles will coalesce with already deposited particles and form new and growing liquid droplets, that are attached to the filter media. Since GeoDict 2023 it is possible to model liquid aerosols and the subsequent coalescence process.

During flight, liquid droplets will be treated similar to the solid particles. Particle movement is modeled by GeoDict as described for solid dust particles. Also, the collisions are modeled as described for the solid particles.

After the movement and deposition of a batch of particles is simulated, the coalescence of the particles is considered: two particles touching each other will join and form a new particle that consists of the mass and center of gravity of both particles. The new particle is placed on the filter media to respect the prescribed contact angle. This creates a flat film for small contact angles and nearly spherical droplets for large contact angles.

After the coalescence of the particles is computed, the next batch of particles is simulated. Thus, the simulation of liquid aerosols follows the same scheme as the simulation of solid particles, but consists of one additional step: namely the modelling of the coalescence.

This approach is described in detail by Hoch, Weber, and Niessner[257]. In this article, a distance-map approach is proposed to describe droplet coalescence in fibrous filter media. The distance-map based approach is validated by comparing droplet shapes to those predicted by the VOF method. The distance-map approach is fast and easy since no conservation equations are solved for the liquid phase.

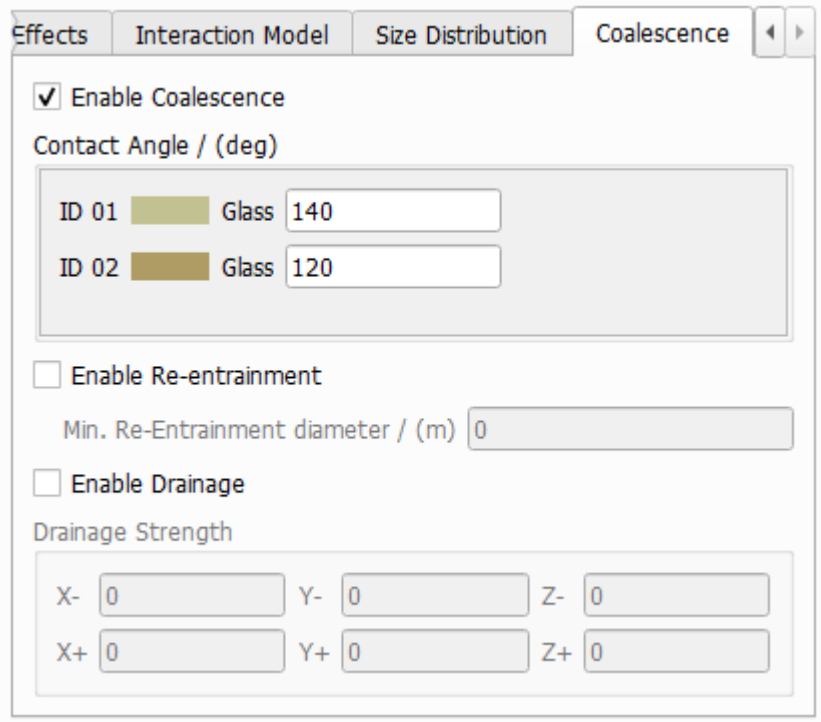
Enable Coalescence

In the Coalescence tab, check **Enable Coalescence** to use this feature.

During the simulation, GeoDict assumes that all particles are resolved by the computational grid. Therefore, you cannot use this option if you selected **Mixed** as **Particle Resolution** in the Constituent Materials tab 64.

For all material IDs representing solid filter material, define a **Contact Angle**. Contact angles above 90 degrees correspond to a wetting, i.e. hydrophilic (if the particles are water) or oleophilic (if the particles are oil) behavior. Contact angles below 90 degrees correspond to a non-wetting, i.e. hydrophobic or oleophobic behavior.

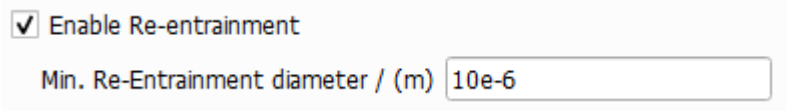
When using this feature, small particles will coalesce into large droplets which are attached to the filter material. Be aware, that in the coalescence step, the complete deposited mass may be redistributed, which is a major difference to the behavior of solid particles in a filter cake. During a filter lifetime simulation, the liquid droplets on the filter material will continue to grow, and eventually block the filter completely (infinite pressure drop). This differs from the behavior of filter cakes made of solid particles, which are porous materials that always allow flow.



In industrial applications, coalescing filters typically achieve a steady state, where the pressure drop stays constant. This is achieved when the mass of small particles captured in a time interval matches the mass of large droplets that detach from the filter, e.g., by gravitational effects.

To achieve a steady state also in the simulations, it is necessary to model also the detachment of droplets. Two mechanism have been implemented in GeoDict that allow to model the detachment of particles: **Re-Entrainment** and **Drainage**. Both mechanisms can be used to achieve a steady state. However, no validation has been done so far, so both features have to be considered as preliminary or experimental, and will be subject to change in future versions of GeoDict.

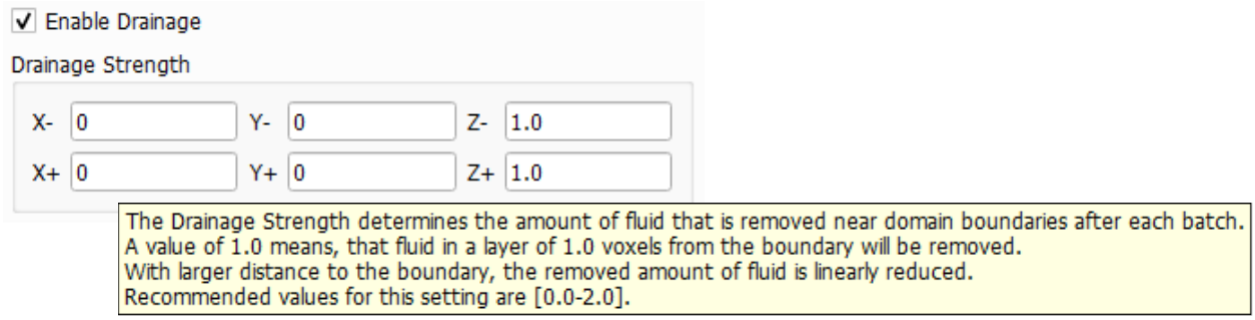
Enable Re-Entrainment (preliminary model)



Re-Entrainment enables the detachment of large liquid droplets from the filter media. Enter the minimal particle diameter required for particles to detach.

Enable Drainage (preliminary model)

Drainage enables the removal of liquid droplets adjacent to the side of the domain. The tooltip gives further explanations on the parameters.



Visualization of Coalescence Results

Be aware that due to the fluidization and redistribution of the deposited particles after each batch, some of the classically used visualization methods in GeoDict will not show the simulation results correctly. Read here 119 how to visualize the results.

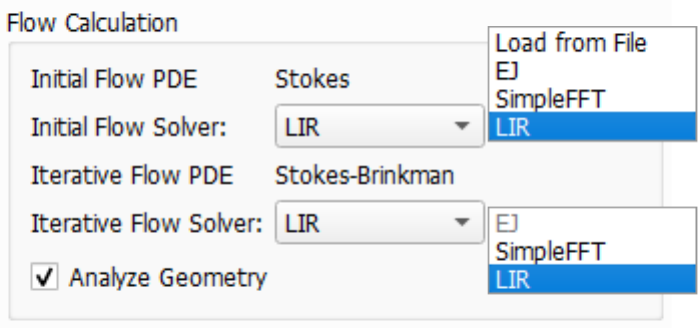
Solver

The **Solver** tab contains on the left the **Flow Calculation**, **Batch Settings**, and the **Parallelization** setup and on the right individual tabs for every solver. Tabs corresponding to unused solvers are grayed out and not selectable.

Flow Calculation

FilterDict treats the flow for the first (initial) batch and all subsequent (iterative) batches differently for two main reasons:

- In the initial batch, the filter is still clean, while in all iterative batches, previously deposited particles are present.
- In all iterative steps, the flow solver can use the solution of the previous step as initial guess, which is not possible for the initial batch.



The names of the equations to solve the **Initial Flow PDE** and **Iterative Flow PDE** are shown for information. These entries cannot be changed directly because they depend on the selection of Resolved / Mixed particles in the Constituent Materials tab 64 and the choice of Creeping / Fast Flow in **Flow Motion** under the Filter Experiment tab 60.

To solve the Flow PDEs, different flow solvers are available as **Initial Flow Solver** and as **Iterative Flow Solver** (EJ, SimpleFFT or LIR). The initial flow field may alternatively also be loaded from a previous FilterDict or FlowDict computation.

Tooltips help choosing the appropriate Initial and Iterative Flow Solvers. The default choice is the LIR solver for both the initial and the iterative flow solver. In most cases, LIR is computationally more efficient than EJ or SimpleFFT. The EJ solver can only solve the Stokes equations, so you cannot select this solver for the Navier-Stokes or Stokes-Brinkman equation.

Analyze Geometry

If this option is chosen, a geometrical analysis at first determines whether a through path exists and removes unconnected pore components from the computational grid. This may speed up the flow computations but requires time for the geometrical analysis.

Batch Settings

The number of **Batches per Flow Field** determines how often the flow field is re-computed. Usually, the flow field should be recomputed for every batch (time interval), but if this is numerically too costly, it can be changed to re-compute the flow field only after every second or third batch. However, when more than one batch per flow field is chosen, the accuracy of the result decreases. In this case, numerical artifacts might be introduced. The most accurate results are obtained by computing one batch per flow field.

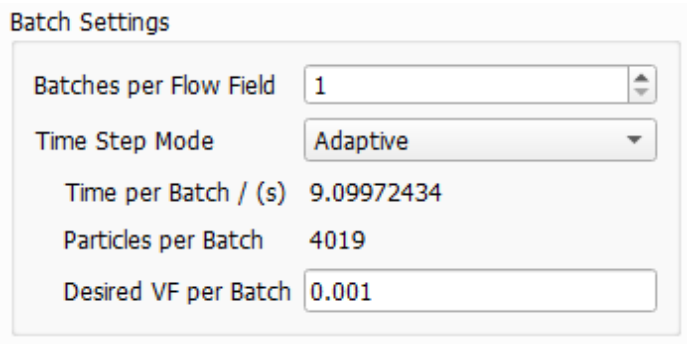
A batch of particles corresponds to a certain time interval in the experiment. The particles simulated in a batch do not interact with each other, but they do interact with the particles deposited in the previous batches. Decreasing the number of particles per batch leads to a higher accuracy, but also to longer simulation times.

The length of the time intervals per batch and the number of particles can be chosen by selecting a **Time Step Mode** from the pull-down menu:

Adaptive

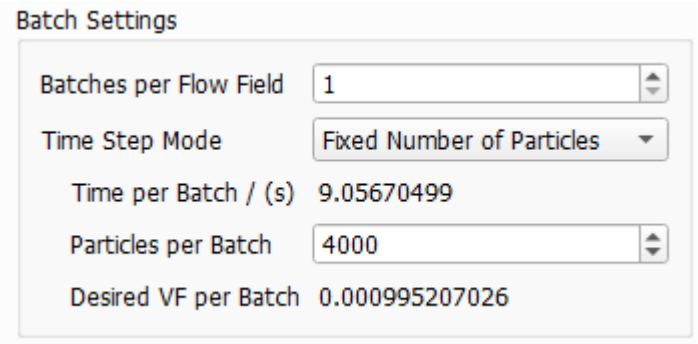
The number of particles is determined adaptively in each step. It depends on the desired volume fraction of deposited particles (**Desired VF per Batch**).

Additionally, FilterDict considers information from the previous steps like, e.g., the particle deposition zone and filter efficiency. Small **Desired VF per Batch** values correspond to shorter time steps and less particles per batch. Larger values lead to longer time steps and more particles per batch.



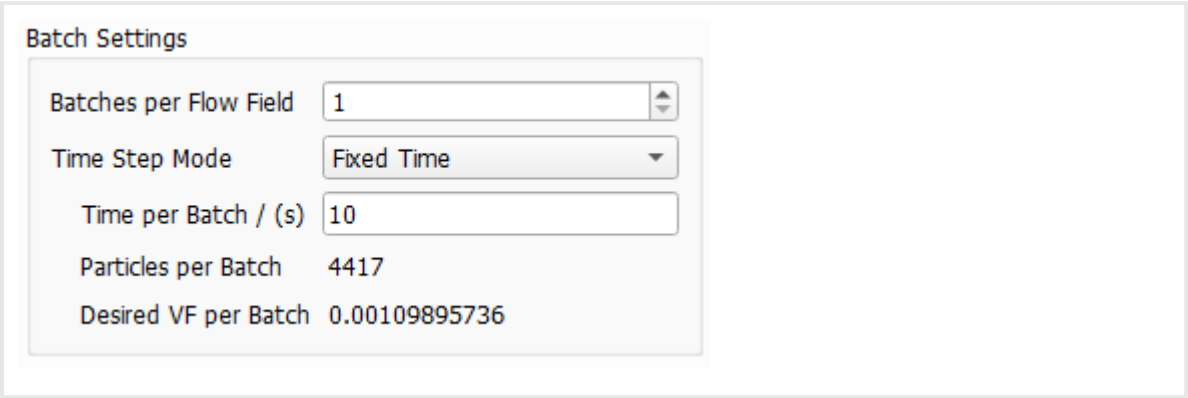
Fixed Number of Particles

Set the number of **Particles per Batch**. The time per batch is computed based on the number of particles.



Fixed Time

The number of particles per batch is determined by the **Time per Batch / (s)**. With a **Max. time reached** of 100 s chosen as simulation stopping criterion and 10 s chosen as **Time per Batch** here, the simulation stops after 10 batches of particles.



For multi pass simulations, **Fixed Number** or **Adaptive** should be preferred over **Fixed Time**. The particle concentration in the fluid changes over time. Therefore, also the number of particles could change significantly with **Fixed Time** steps and this might lead to a less stable simulation.

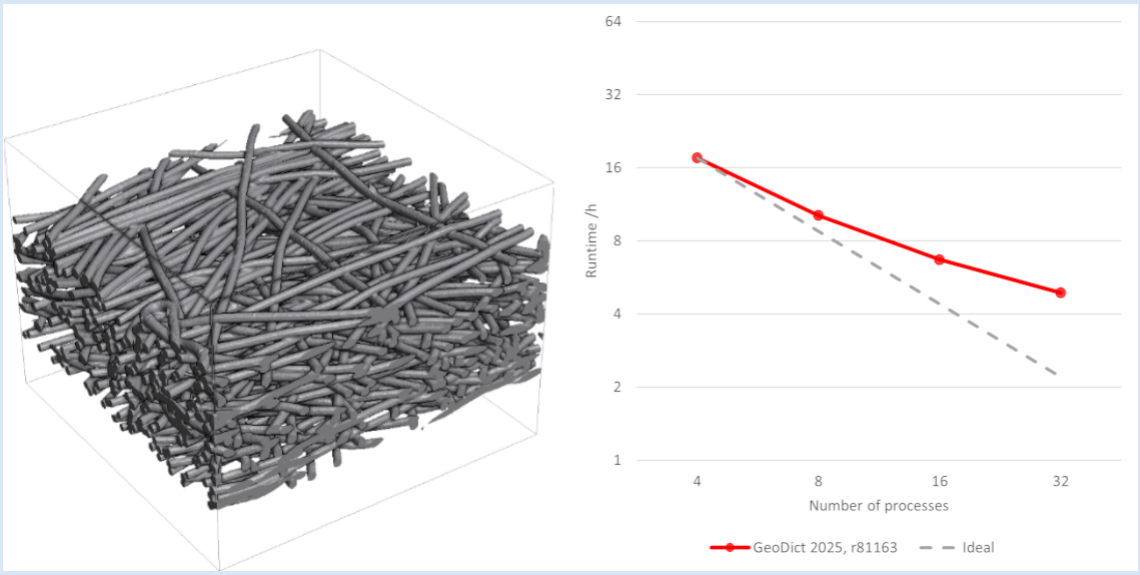
Parallelization

Depending on the purchased license, the simulation process can be parallelized. The **Parallelization Options** dialog opens when clicking the **Edit...** button, to choose between **Sequential**, **Parallel (Shared Memory)**, **Automatic Maximum of Threads** and **Cluster**. For details on how to set up and run parallel computations, refer to the High Performance Computations user guide

The chosen parallelization settings apply for all steps of the simulation. Per default, the **Automatic Maximum of Threads** option is used, and the exact number of threads depends on the number of cores on the current computer and the number of licensed processes.

Parallelization Benchmark Results

The example computation was run on a server with 2xIntel E5-2697A v4 processors with 16 cores each, running with a maximum of 3.60GHz. The input structure is an air filter media of size 1024x1024x768 voxels. A filter lifetime single pass simulation was run for a particle distribution according to ISO A1 ultrafine particles. 50 batches were computed with a time per batch of 10 s. For each batch approximately 35000 particles were simulated. The flow was computed with the LIR solver, which required approx 30 GB of RAM to run the simulation.



The plot shows the runtime for a different number of processes for the whole simulation. The ideal speedup, i.e. getting half the runtime for twice the number of processes, is also shown as a gray dashed line.

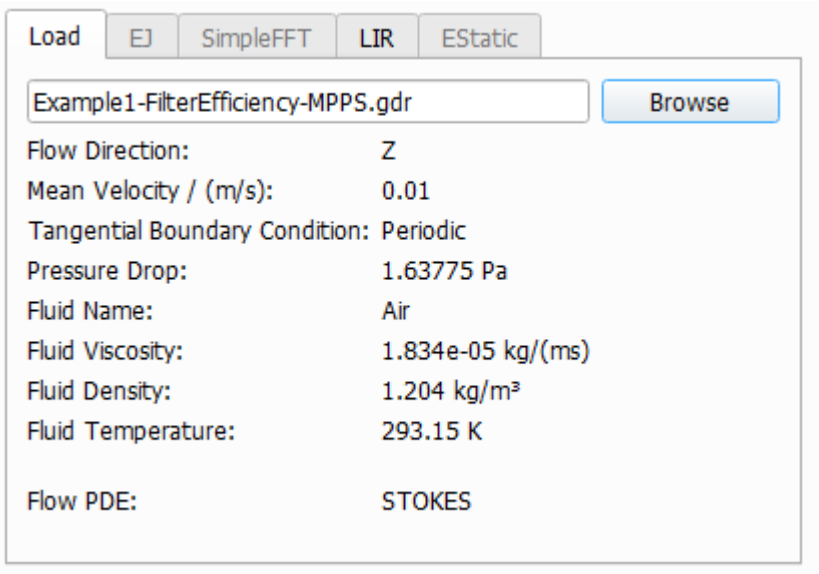
For the computation of 50 batches, a significant part of the runtime is spent for input and output in this example, as is typical for filter simulations. This leads to a reduced speedup when using a large number of processes for a small sized structure size.

Load

Select **Load From File** in the **Flow Calculation** panel as **Initial Flow Solver** to use a precomputed flow field. The flow field may originate from a FlowDict simulation or from a FilterDict simulation. When calculated with FlowDict, you have to make sure to:

1. Add inflow and outflow regions to the media model before running FlowDict.
2. Compute the flow in the Z-direction. You may either use the Stokes or Navier-Stokes equations depending on the flow velocity for this. Make sure that you use the same boundary conditions in FlowDict as you would use in FilterDict.
3. Set the accuracy at least one order of magnitude higher than the default in FlowDict (e.g., Error Bound = 0.001 instead of 0.01). The stopping criteria depend on global values, whereas the particle movement depends on the local flow field which is subjected to larger deviations.

Under the **Load** tab, click **Browse** and choose a result file (GDR) to open. If no flow field is selected, a warning message appears (No flow field chosen) when trying to run the simulation.



The physical properties of the fluid used in the flow simulation are entered automatically when loading the flow GDR file and appear listed under the **Load** tab.

- The **Flow direction: Z** is the main direction of the flow.
- The **Mean Velocity** (m/s) of the flow field, the used **Tangential Boundary Conditions**, and the **Pressure Drop** are taken from the loaded flow simulation. If the flow result was obtained by solving the Stokes equation, the solution is linear and will be rescaled automatically to match the Mean Velocity entered in the **Filter Experiment** tab.
- The fluid parameters **Fluid Viscosity** (kg/m s), **Fluid Density** (kg/m^3), and **Fluid Temperature** (K) are the physical values of the fluid used by the solver for the calculation of the flow field.

The fluid settings chosen under the Constituent Materials must be the same as the fluid that was used in the previously run flow simulation, loaded through the GDR file.

Also, the **Tangential Boundary Conditions** selected in the **Filter Experiment** tab must match with those used in the flow simulation.

EJ

The **EJ** tab becomes editable when you select the EJ solver as **Initial Flow Solver** or **Iterative Flow Solver** in the Flow Calculation 87 panel.

Simulation Stopping Criterion: Tolerance or Residual

The **Tolerance** stopping criterion looks for stagnation of the method when the process becomes stationary, i.e., the improvement in the pressure drop value becomes extremely small from iteration to iteration.

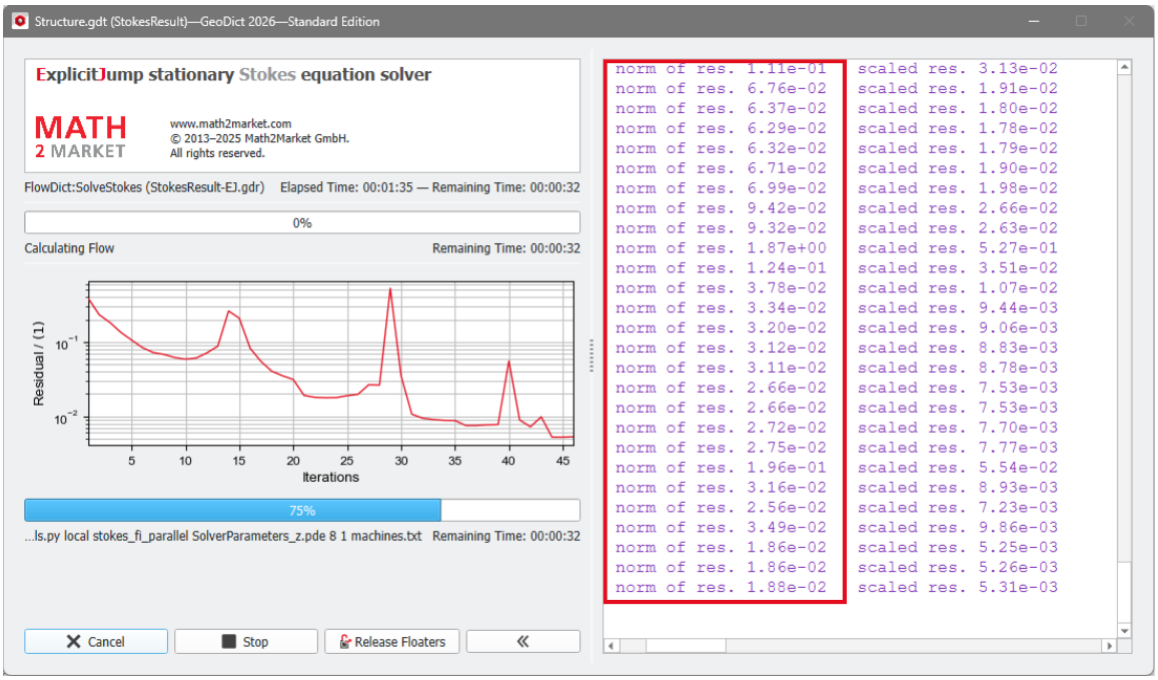
In each iteration, the solver checks for the current computed value x_c against the values x_{c-i} of the last 100 iterations if there are any changes.

$$\max_{i\in[1,100]} \frac{x_c - x_{c-i}}{x_{c-i}} < \text{tolerance} \tag{44}$$

The computation is stopped if the maximal relative change is smaller than the value entered for **Tolerance**. When there is doubt about the quality of the solution, decrease the **Tolerance** value by a factor of ten. The drawback of this criterion is that the solver sometimes might stop too early in case of slow convergence rate or at local extrema of oscillatory convergence curves.

When the **Residual** stopping criterion is used, the iteration is stopped if the solution satisfies the equation up to the required accuracy.

By setting the stopping criterion to **Residual**, the computations terminate as soon as the relative norm drops below the selected residual threshold. The relative norm of the Schur Complement residual is computed and displayed in the console window during the calculations. The console is visible by clicking the double arrow button on the lower right corner in the progress dialog.



The recommendation to choose **Tolerance** or **Residual** for the **EJ** solver is based on the structure's porosity. Both give similar results for highly porous structures. For dense structures, if using the Schur Complement **Residual**, the relative norm of the residual may be small even though the correct pressure drop has not been reached. So, when in doubt, use the **Tolerance** criteria – the default option.

Be aware that the default accuracy in FilterDict is one order of magnitude higher than in FlowDict. The stopping criteria depend on global values, whereas the particle movement depends on the local flow field. It is therefore necessary to make sure that not only the overall pressure drop is computed up to the given accuracy, but that also

the local values are sufficiently accurate. This typically achieved by setting a higher accuracy as stopping criterion.

Simulation Stopping Criterion: Max. Iterations and Max. Run Time

Use the **Maximum Iterations** value or the **Maximum Run Time (h)** stopping criteria causes the solver to stop if the maximal number of iterations and/or the maximal runtime (in hours) is exceeded.

When the solver stops because one of these criteria has been reached, no guarantee on the quality of solution can be given. In this case, a warning is printed into the report. The following possibilities might help:

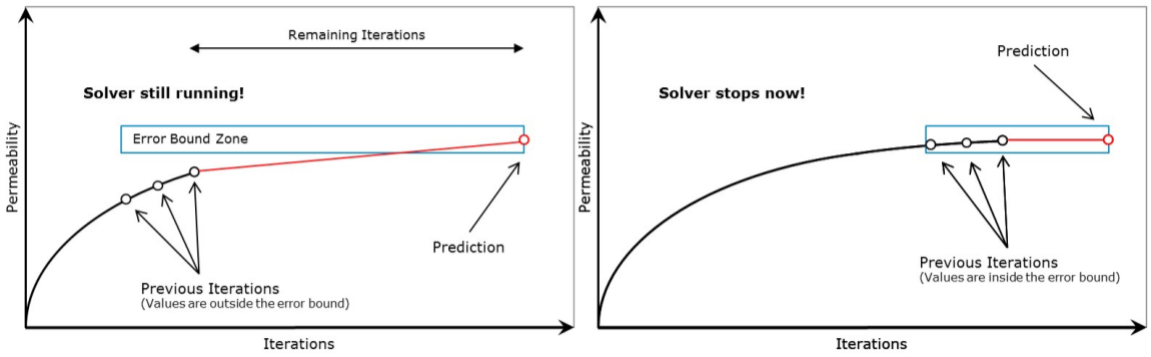
- Check the corresponding .log file to see how large the residual values and the pressure drop values are. If the pressure drop values are in reasonable range and they do not have changed during the last iterations, you may decide to use the current result.
- Double-check the structure and parameter values. Unphysical parameters or too rough resolution of the structure (leading, e.g., to artificial unconnected components) can cause an iterative solver to fail.

SimpleFFT

The **SimpleFFT** tab becomes editable when you select the SimpleFFT solver as **Initial Flow Solver** or **Iterative Flow Solver** in the Flow Calculation 87 panel.

Simulation Stopping Criterion: Error Bound, Tolerance and Residual

The stopping criterion **Error Bound** uses the result of previous iterations, and predicts the final solution based on linear and quadratic extrapolation. The solver stops if the relative difference between computed and predicted solution is smaller than the specified error bound. The stopping criterion recognizes oscillations in the convergence behavior and prevents premature stopping at local minima or maxima. A damped convergence curve is fit through the oscillating curve and the solver stops then regarding the damped convergence curve.



If the Krylov method (under LIR - Advanced Options) is activated, the definition of **Error Bound** is somewhat different. Here, no prediction is made, instead the continuity between neighboring cells and the conservation of mass is checked. The maximal value is normalized by the mean flow and if this value is smaller than the **Error Bound** the simulation stops.

The **Tolerance** stopping criterion looks for stagnation of the method when the process becomes stationary, i.e., the improvement in the pressure drop value becomes extremely small from iteration to iteration.

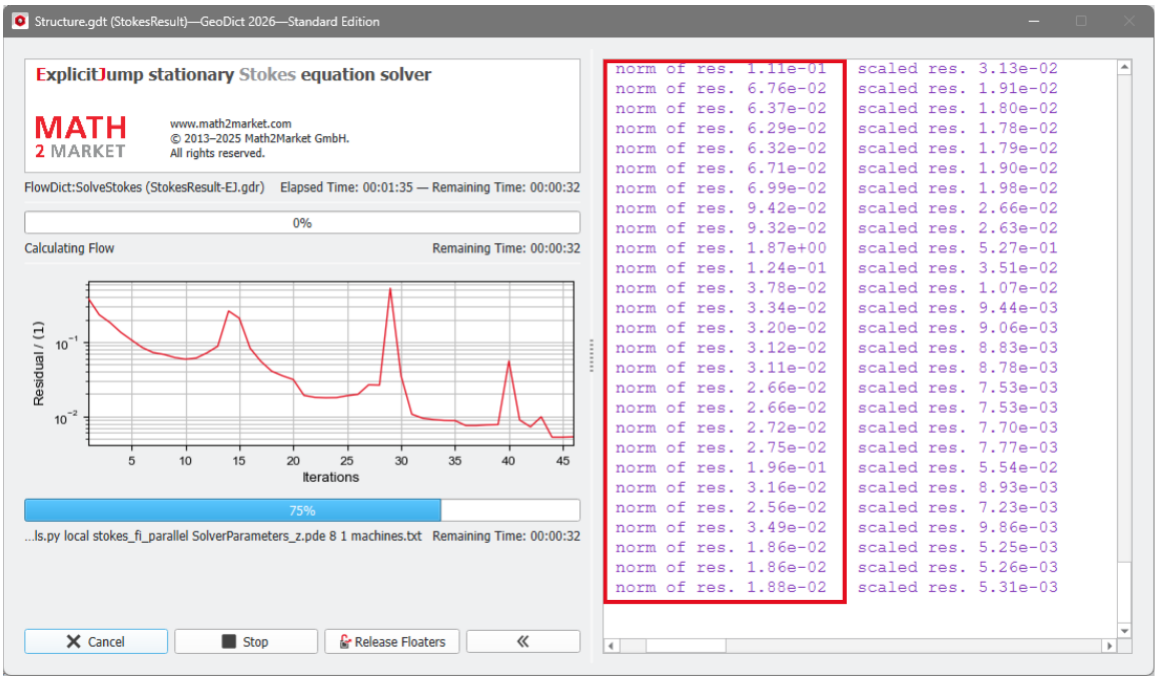
In each iteration, the solver checks for the current computed value x_c against the values x_{c-i} of the last 100 iterations if there are any changes.

$$\max_{i\in[1,100]} \frac{x_c - x_{c-i}}{x_{c-i}} < \text{tolerance} \tag{44}$$

The computation is stopped if the maximal relative change is smaller than the value entered for **Tolerance**. When there is doubt about the quality of the solution, decrease the **Tolerance** value by a factor of ten. The drawback of this criterion is that the solver sometimes might stop too early in case of slow convergence rate or at local extrema of oscillatory convergence curves.

When the **Residual** stopping criterion is used, the iteration is stopped if the solution satisfies the equation up to the required accuracy.

By setting the stopping criterion to **Residual**, the computations terminate as soon as the relative norm drops below the selected residual threshold. The relative norm of the Schur Complement residual is computed and displayed in the console window during the calculations. The console is visible by clicking the double arrow button on the lower right corner in the progress dialog.



The recommendation to choose **Tolerance** or **Residual** for the **EJ** solver is based on the structure's porosity. Both give similar results for highly porous structures. For dense structures, if using the Schur Complement **Residual**, the relative norm of the residual may be small even though the correct pressure drop has not been reached. So, when in doubt, use the **Tolerance** criteria – the default option.

Be aware that the default accuracy in FilterDict is one order of magnitude higher than in FlowDict. The stopping criteria depend on global values, whereas the particle movement depends on the local flow field. It is therefore necessary to make sure that not only the overall pressure drop is computed up to the given accuracy, but that also the local values are sufficiently accurate. This typically achieved by setting a higher accuracy as stopping criterion.

Simulation Stopping Criterion: Max. Iterations and Max. Run Time

Use the **Maximum Iterations** value or the **Maximum Run Time (h)** stopping criteria causes the solver to stop if the maximal number of iterations and/or the maximal runtime (in hours) is exceeded.

When the solver stops because one of these criteria has been reached, no guarantee on the quality of solution can be given. In this case, a warning is printed into the report. The following possibilities might help:

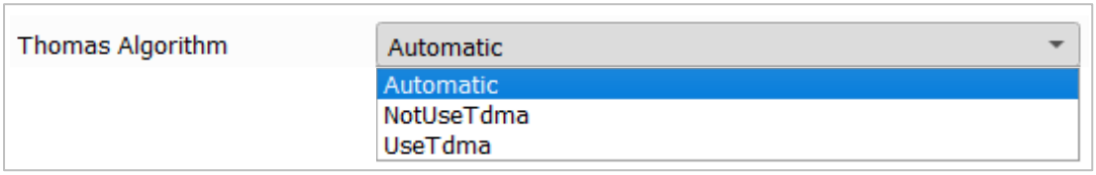
- Check the corresponding .log file to see how large the residual values and the pressure drop values are. If the pressure drop values are in reasonable range and they do not have changed during the last iterations, you may decide to use the current result.
- Double-check the structure and parameter values. Unphysical parameters or too

rough resolution of the structure (leading, e.g., to artificial unconnected components) can cause an iterative solver to fail.

Thomas Algorithm

The tridiagonal matrix algorithm (Tdma), also known as the **Thomas Algorithm**, may help to improve the convergence for high porosity structure, especially for Navier-Stokes. The algorithm transforms the 3D flow problem to 2D. Thus, it leads to a shorter runtime and increases the convergence speed.

Three options can be chosen from the pull-down menu: **Automatic**, **NotUse Tdma** or **Use Tdma**.



If **Automatic** is selected, the SimpleFFT solver decides if **Tdma** will be used, regarding the porosity of the structure. Thus, **Tdma** is used if the porosity is high. For most cases it is recommended to choose **Automatic**.

If a message dialog appears, displaying that NaN is detected in iterations, **NotUse Tdma** should be tried.

If the structures porosity is very high, it might decrease the runtime a little bit to explicitly choose **Use Tdma**. But for most cases **Automatic** will work well.

For more information about the Thomas algorithm see here.

Velocity Relaxation and Pressure Relaxation

Depending on the material parameters and geometrical structure of the structure, the underlying mathematical problem can vary in complexity, thus influencing the behavior of the solver. This is directly related to the Reynolds number, an indication of the complexity of the flow solver computations. The higher the Reynolds number, the more **Stable** the flow solver settings should be, resulting in higher number of iterations, slower time stepping, and longer flow solver run times. However, making the solver run less iterations and, thus, faster (**Fast**), implies the risk that the solver does not converge.

For the SimpleFFT solver, the management of this balance is done through the **Velocity relaxation: Stable ↔ Fast** and **Pressure relaxation: Stable ↔ Fast** slide bars.

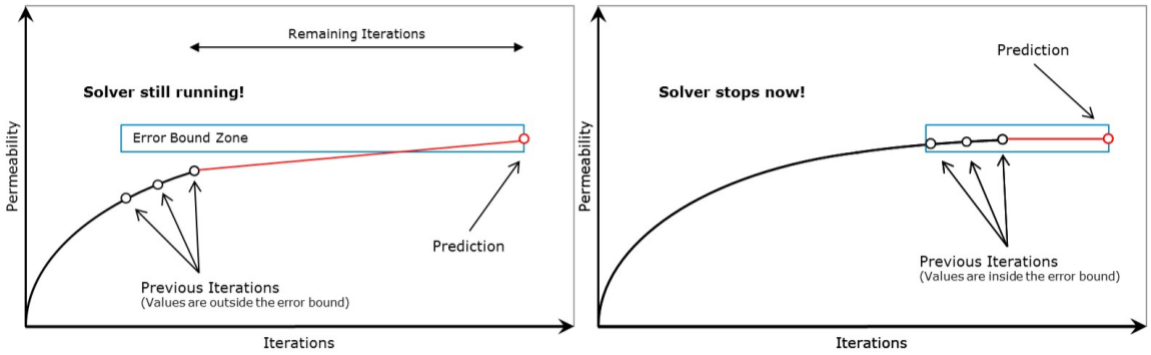
Setting the balance of **Stable** versus **Fast** is a trial-and-error process. Although there is no general rule to optimize it, the log files and the visualization of the structure might help finding the balance. Non-sense structure visualization results indicate that a more stable, slower solver is needed.

LIR

The LIR tab becomes editable when you select the LIR solver as **Initial Flow Solver** or **Iterative Flow Solver** in the Flow Calculation 87 panel.

Simulation Stopping Criterion: Error Bound or Tolerance

The stopping criterion **Error Bound** uses the result of previous iterations, and predicts the final solution based on linear and quadratic extrapolation. The solver stops if the relative difference between computed and predicted solution is smaller than the specified error bound. The stopping criterion recognizes oscillations in the convergence behavior and prevents premature stopping at local minima or maxima. A damped convergence curve is fit through the oscillating curve and the solver stops then regarding the damped convergence curve.



If the Krylov method (under LIR - Advanced Options) is activated, the definition of **Error Bound** is somewhat different. Here, no prediction is made, instead the continuity between neighboring cells and the conservation of mass is checked. The maximal value is normalized by the mean flow and if this value is smaller than the **Error Bound** the simulation stops.

The **Tolerance** stopping criterion looks for stagnation of the method when the process becomes stationary, i.e., the improvement in the pressure drop value becomes extremely small from iteration to iteration.

In each iteration, the solver checks for the current computed value x_c against the values x_{c-i} of the last 100 iterations if there are any changes.

$$\max_{i\in[1,100]} \frac{x_c - x_{c-i}}{x_{c-i}} < \text{tolerance} \tag{44}$$

The computation is stopped if the maximal relative change is smaller than the value entered for **Tolerance**. When there is doubt about the quality of the solution, decrease the **Tolerance** value by a factor of ten. The drawback of this criterion is that the solver sometimes might stop too early in case of slow convergence rate or at local extrema of oscillatory convergence curves.

Be aware that the default accuracy in FilterDict is one order of magnitude higher than in FlowDict. The stopping criteria depend on global values, whereas the particle movement depends on the local flow field. It is therefore necessary to make sure that not only the overall pressure drop is computed up to the given accuracy, but that also the local values are sufficiently accurate. This typically achieved by setting a higher accuracy as stopping criterion.

Simulation Stopping Criterion: Max. Iterations and Max. Run Time

Use the **Maximum Iterations** value or the **Maximum Run Time (h)** stopping criteria causes the solver to stop if the maximal number of iterations and/or the maximal runtime (in hours) is exceeded.

When the solver stops because one of these criteria has been reached, no guarantee on the quality of solution can be given. In this case, a warning is printed into the report. The following possibilities might help:

- Check the corresponding .log file to see how large the residual values and the pressure drop values are. If the pressure drop values are in reasonable range and they do not have changed during the last iterations, you may decide to use the current result.
- Double-check the structure and parameter values. Unphysical parameters or too rough resolution of the structure (leading, e.g., to artificial unconnected components) can cause an iterative solver to fail.

Advanced Options: Write Compressed Volume Field

The LIR solver uses a very memory efficient adaptive grid structure for the simulations.

If the option **Write Compressed Volume Fields** is checked, then the adaptive grid is used as compression method for writing out *.vap files. This option allows to save 80-90% space on hard drive. The runtime for writing *.vap files is also reduced significantly. But the runtime for loading and uncompressing of compressed *.vap is increased by the amount of runtime that was saved for writing out compressed *.vap files.

If the option **Write Compressed Volume Fields** is not checked, then a usual regular grid is used for writing out *.vap files.

Advanced Options: Optimization Options

The **Multigrid Method** (see, e.g., Wesseling, 2004) was introduced to speed-up the computation and reduce the runtime significantly. The main idea of **Multigrid** is the usage of multiple coarser adaptive grids to speed up convergence behavior but requires only little more memory.

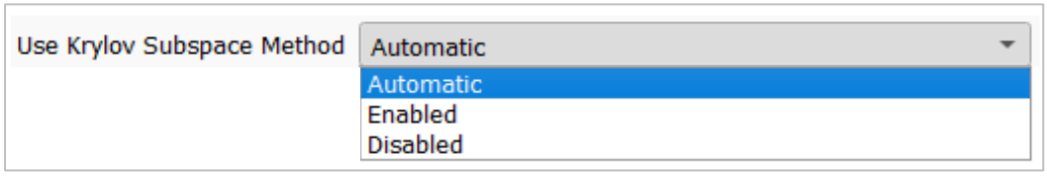
The method is available to solve the Stokes and Stokes–Brinkman equations as well as for solving mechanics, diffusion, thermal, and electrical conduction and is enabled by default.

Another speed-up option to accelerate the convergence behavior of the LIR solver is called **Krylov Subspace Method**.

The runtime of the LIR solver depends on many different properties of the structure and the simulation parameters. The BiCGStab algorithm is used, which can reduce the runtime for challenging simulation very drastically.

Note! Using the BiCGStab method approximately doubles the amount of RAM needed for the computation.

Unfortunately, the Krylov method is not always faster than a simulation without the Krylov method and therefore we introduced an **Automatic** mode which uses some heuristics to choose the most efficient method based on structure, material parameters, and boundary conditions automatically. Of course, it is possible to explicitly enable (**Enabled**) or disable (**Disabled**) the method.



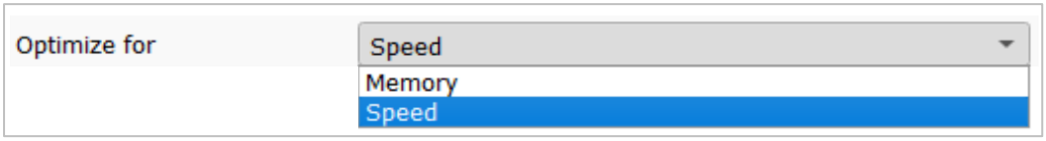
Note! In case that the Krylov subspace method (BICGStab) is used, the **Relaxation** may also be adapted automatically.

Depending on the material parameters and geometry of the structure, the underlying mathematical problem can vary in complexity, thus influencing the behavior of the solver. The more complex the problem is, the more stable the solver settings should be.

With the **Relaxation** number, the solver is adjusted from **Stable** (which results in higher number of iterations, slower time stepping, and longer solver run times), to **Fast**, which makes the solver run less iterations but implies the risk that the solver does not

converge. The **Relaxation** is a parameter of the SOR method and must be between 0 and 2 to ensure convergence. For relaxation values smaller than one (<1.0), the simulation is more stable. For relaxation values larger than one (>1.0), the simulation converges faster.

The **LIR** solver can **Optimize for Speed** or **Memory**.



- If **Speed** is chosen, the solver constructs additional optimization structures. The runtime is decreased by up to 30% but requires up to 50% more memory compared to the other option.
- If **Memory** is chosen, the runtime is increased by up to 40% but the solver requires up to 50% less memory.

Advanced Options: Grid Options

The **Grid Type** decides what kind of tree structure is used for the simulation.

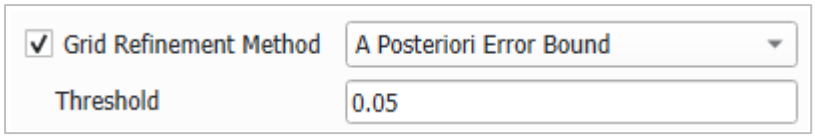
The default option is **LIR-Tree** and should always be used. The solver uses an adaptive grid structure called LIR-tree and needs up to 10 times less runtime and memory compared to the **Regular Grid** option.

The solver can analyze the result field during the computation and improves the adaptive grid in places where more accuracy is needed. The LIR solver splits cells where a high gradient occurs.

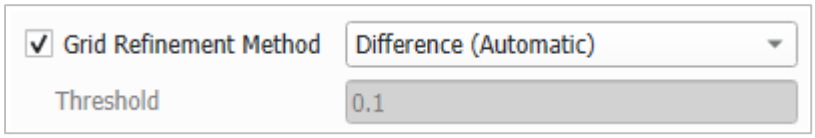
The solver can analyze the computed field and refines the adaptive grid during the computation at locations where more accuracy is needed. The LIR solver splits cells where a high gradients occur when the **Grid Refinement Method** is enabled. In this case, the additional parameters **Number of Grid Refinements**, **Allow Sub-Voxel Resolution**, and **Allow Grid Re-Coarsening** become available.

Select one of the three available refinement methods from the drop-down menu.

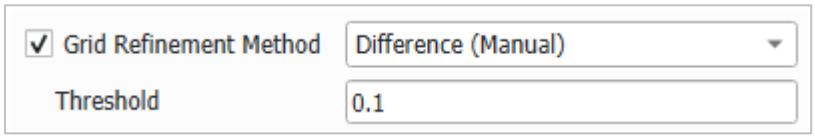
When **A Posteriori Error Bound** is selected, the solver targets the specified accuracy **Threshold**. While the **Error Bound** (set as Simulation Stopping Criterion) determines the relative error in the solution of the linear system, **A Posteriori Error Bound** refers to the relative error estimated by comparing high-order and low-order discrete solutions. The accuracy **Threshold** refers to the relative error to the analytical solution and the value must be between 0.0 and 1.0.



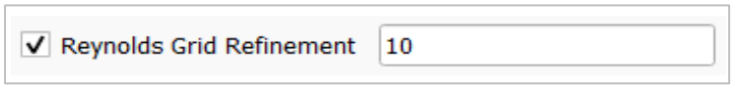
Choosing **Difference (Automatic)** leads to computational cells being split when the difference in values between neighboring cells exceeds a certain **Threshold**. The solver automatically chooses all internal parameters based on the structure and simulation settings. For this option, cells are split where the current gradient is greater than the **Threshold** multiplied by the maximum gradient, where the threshold is determined automatically.



Selecting **Difference (Manual)** also causes the computational cells to be split when the difference in values between neighboring cells exceeds the specified **Threshold**. You can specify all parameters (Threshold and Number of Grid Refinements) for the grid refinement manually. For this option, cells are split where the current gradient is greater than the **Threshold** multiplied by the maximum gradient. The **Threshold** value must be between 0.0 and 1.0. The recommended value range is between 0.05 and 0.1.



Activate **Reynolds Grid Refinement** to use the local Reynolds number (Re) for adaptive grid refinement.



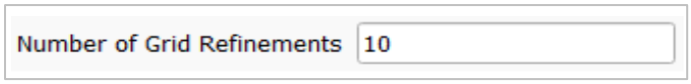
A grid cell is split if the local Reynolds number, i.e., the ratio between inertial and viscous forces exceeds the given threshold. The Reynolds number is defined as:

$$\mathrm{Re} = \frac{\rho \mathrm{u} \mathrm{L}}{\mu} \quad (45)$$

Reynolds Number

where ρ is the density of the fluid (SI units: kg/m^3), u is the velocity of the fluid (m/s), μ is the dynamic viscosity of the fluid (Pa·s or N·s/m^2 or kg/m·s), L is a characteristic length (m). For the local Reynolds number, the size of the grid cell is used for the characteristic length.

The **Number of Grid Refinements** controls the maximum number of grid refinements that the solver can perform during the simulation. The value should be set between 0 and 10, where a value of 0 means that no grid refinements will be made. Grid refinements may increase the number of iterations, runtime, and memory requirements.

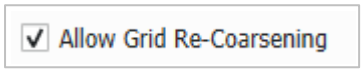


When **Allow Sub-Voxel Resolution** is enabled, the solver is allowed to split computational cells to sizes smaller than the voxel length.

This feature is beneficial for low-porosity structures for which the pore throats require finer resolution or for modeling fast Navier-Stokes flows with strong vortices.

Enabling this feature may increase the number of iterations, runtime, and memory requirements.

Check **Allow Grid Re-Coarsening** to allow the solver to automatically revert the grid refinement and coarsen the computational cells.

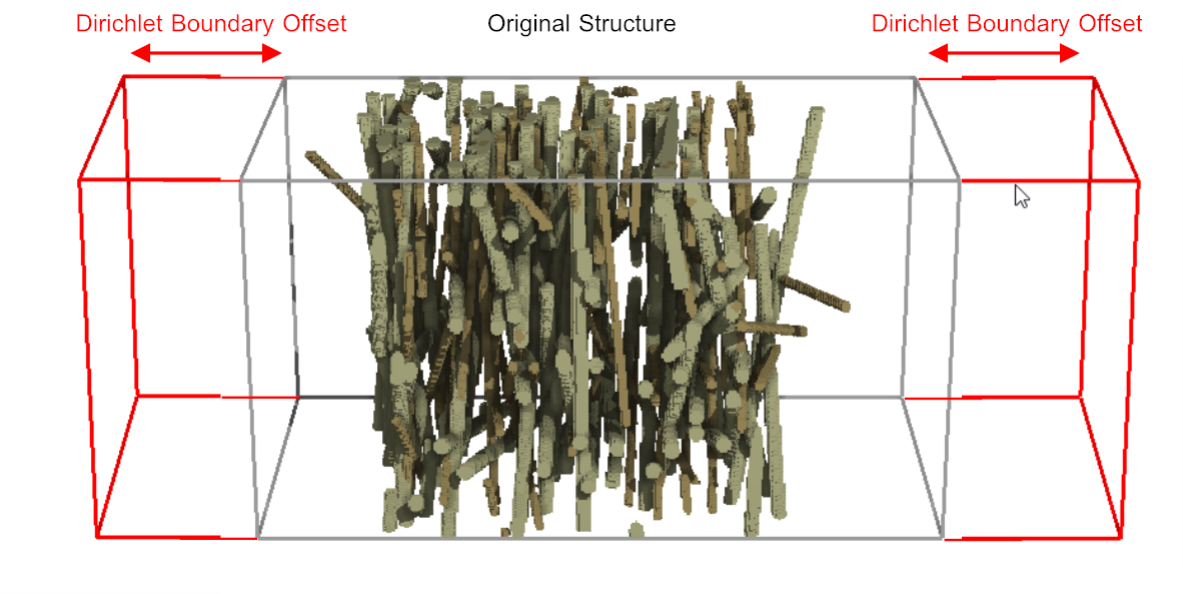


This means, grid refinement is done temporarily. Afterwards, the cells are merged as soon as accuracy is not needed anymore. This reduces the memory and runtime requirements.

Estatic

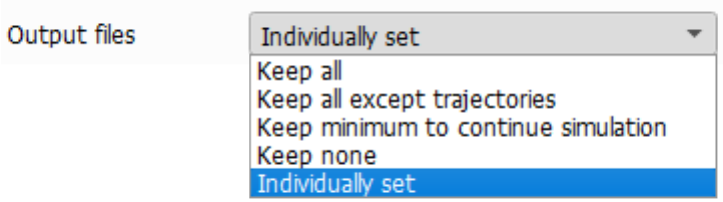
The **EStatic** tab is becomes editable when **Include Electrostatic Effects** has been checked on the Electrostatic Effects tab. Equation (15) 9 is solved to calculate the electrostatic potential.

The **Dirichlet Boundary Offset** is the offset Z_0(in Voxels), where the zero Dirichlet boundary conditions for the potential Φ apply. For large numbers Z_0, the computation is more accurate but also requires more numerical resources. This increment of the computational domain occurs internally for the electrostatic solver and does not influence the structure seen by the user in the visualization area.



Output

First, select which **Output files** should be stored. You may select one of the predefined settings **Keep all**, **Keep all except trajectories**, **Keep minimum to continue simulation**, **Keep none**, or **Individually set**.



All these files are needed only for visualization of the results. The choices made here do not influence the filter lifetime results contained in the GDR (result) file but may disable some of the visualization options which require to load additional data.

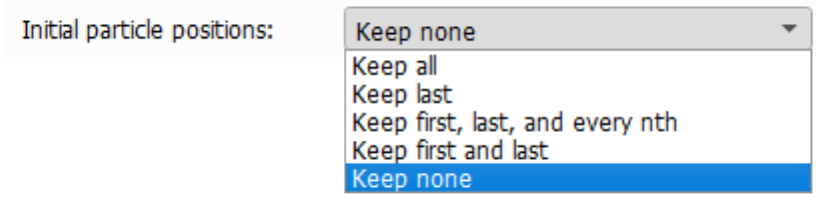
When **Keep all** is checked, all intermediate files written and read by the flow solver, particle tracker and electrostatic solver are kept. If the filter model is large, and/or the simulation contains many batches, a large amount of potentially unnecessary data is stored in that way.

Check **Keep all except trajectories** if you want to keep all data files except of the particle trajectories. Trajectory files are needed solely for visualization of the particle movement and may become very large if many particles are tracked.

Check **Keep minimum to continue simulation** to save only the files need to continue from an interrupted simulation.

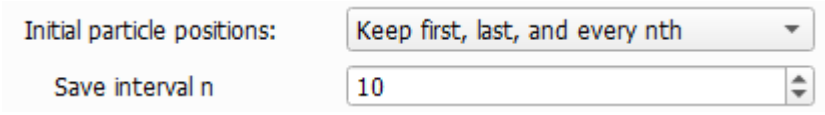
Check **Keep none** if you want to reduce the amount of hard disk space needed as far as possible.

If you select **Individually set**, you may decide in detail which files to keep for visualization. For each file type, choose between **Keep all**, **Keep last**, **Keep first, last and every nth**, **Keep first and last** and **Keep none**.

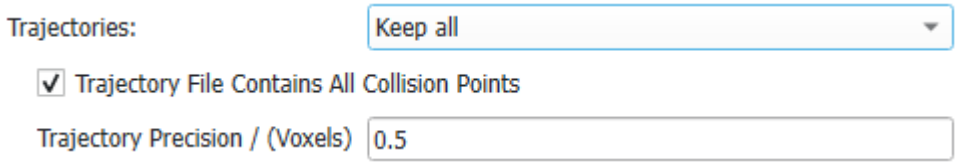


Decide, for which batch this file type is stored: every batch, only the first and/or last batch, or set an interval such that this file type is stored for every nth batch.

In that case, set the **Save interval n**:



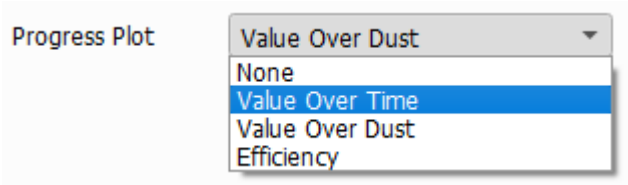
To visualize the movement of particles, you need to keep the **Trajectories**. Be aware, that these files may become very large if many particles are simulated.



You can influence the file size and the precision of the stored trajectories (e.g., the number of points on the particle paths saved in the file) by two additional parameters:

- Check **Trajectory File Contains All Collision Points** to add all points where a particle touches the surface of a solid object to the trajectory file.
- The value entered as **Trajectory Precision** determines how accurately the trajectories are stored. The given value determines the traveling distance between two points stored in the trajectory file.

Select which plot is shown during the computation by setting **Progress Plot** to either **None**, **Value** (i.e. pressure drop or flow rate) **Over Time**, **Value** (i.e. pressure drop or flow rate) **Over Dust**, or the fractional filter **Efficiency** achieved over all batches.



Results

Click **OK** to input the entered parameters, and then click **Run** in the FilterDict section to start the Filter Lifetime command. The results are immediately shown in the opening Result Viewer after the simulation is finished.

Report

Stopping Criterion

At the top of the **Report**, the stopping criteria reached by the simulation is shown. Here, it is also reported if the simulation was canceled (**Stopped by user**). If you open the result file while the simulation is still running, the first line reads **Simulation unfinished**.

Filter Clogging Analysis

Below, the **Filter clogging analysis** analyzes the properties of the filter cake. These parameters can be used as inputs for simulating filter elements or complete filters when resolving filter media and particles is not possible and an effective description of filtration parameters is required.

Filter clogging analysis

	Layer thickness / (mm)	f_{max}	$sigma_{max}$ / (kg / m^3 s)
Depth filtration	0.285	0.207588	2.94984e+06
Cake filtration	0.105	0.230616	8.56599e+06

Clean filter permeability: 3.19892e-11 m^2

Clogging factors - maximum packing density f_{max} and flow resistivity $sigma_{max}$ -
are used for upscaling to the filter element scale and can be entered in the porous material properties.

The **Depth filtration** parameters are computed as follows:

The **Layer thickness** h_{depth} is the thickness of the filter media. All voxel layers containing solid voxels are considered as part of the filter media. Note, that this definition is different from the definition of the thickness used in the MatDict - Material Statistics – Thickness Estimation command.

$f_{\max}$ is the maximum solid density achieved in one z-layer.

The flow resistivity σ_{depth} caused by particles filtered in the depth of the filter media is computed as quotient of the pressure drop increase $\Delta P_{\mathrm{final}} - \Delta P_{\mathrm{clean}}$ over the filter media (here, the same part is considered as for the Layer Thickness) and the average flow velocity $\bar{v}$:

$$\sigma_{\mathrm{depth}} = \frac{\Delta P_{\mathrm{final}} - \Delta P_{\mathrm{clean}}}{\bar{v} h_{\mathrm{depth}}} \quad (46)$$

This resistivity represents the average over all z-layers of the filter media, which does not directly correspond to the maximum solid density $f_{\max}$, but to the average solid density f_{depth} of all layers. $\sigma_{\max}$ is then computed by scaling the resistivity:

$$\sigma_{\max} = \sqrt{\frac{f_{\max}}{f_{\text{depth}}}} \cdot \sigma_{\text{depth}} \quad (47)$$

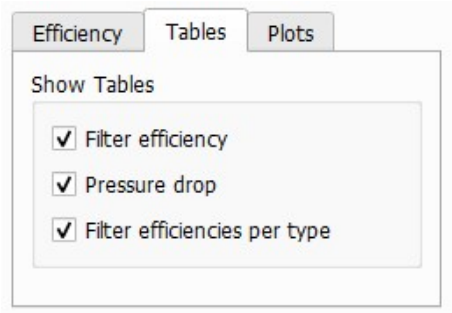
The **Cake filtration** parameters are computed if a sufficiently thick filter cake has formed. They are determined as follows:

- All voxel layers in the inflow region which are at least partially filled with deposited dust are analyzed. $f_{\max}$ is the maximum solid density achieved in a single z-layer in the inflow region.
- The **Layer thickness** is the thickness of the filter cake. Here, only layers that achieve a solid volume fraction of at least 80% of $f_{\max}$ are taken into account. Thus, dendrites forming on top of the cake are not taken into account when determining the thickness.
- $\sigma_{\max}$ is the flow resistivity, computed as quotient of the pressure drop over the filter cake (here, the same part is considered as for the Layer Thickness) and the average flow velocity.

The **Clean filter permeability** is the computed permeability of the clean filter. The value is computed using the **Depth filtration - Layer thickness** as thickness of the filter media.

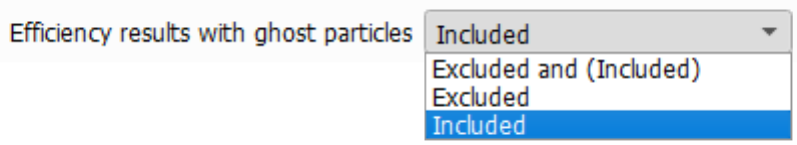
Tables

Below, a number of tables are presented, that can be selected under the **Tables** tab in the post-processing section on the left.



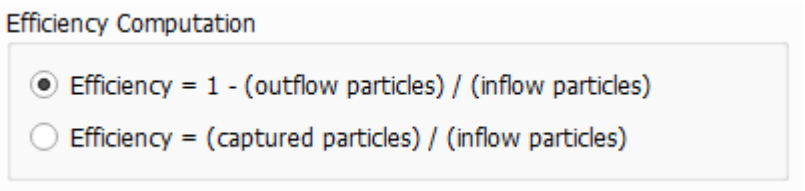
Filter efficiency

The **Filter efficiency** table contains, for every batch, the **Deposited dust**, the **Volume added** to the material by the deposited particles, the **Volume loss**, and the **Filter efficiency** in percentage **by count** and **by weight**.

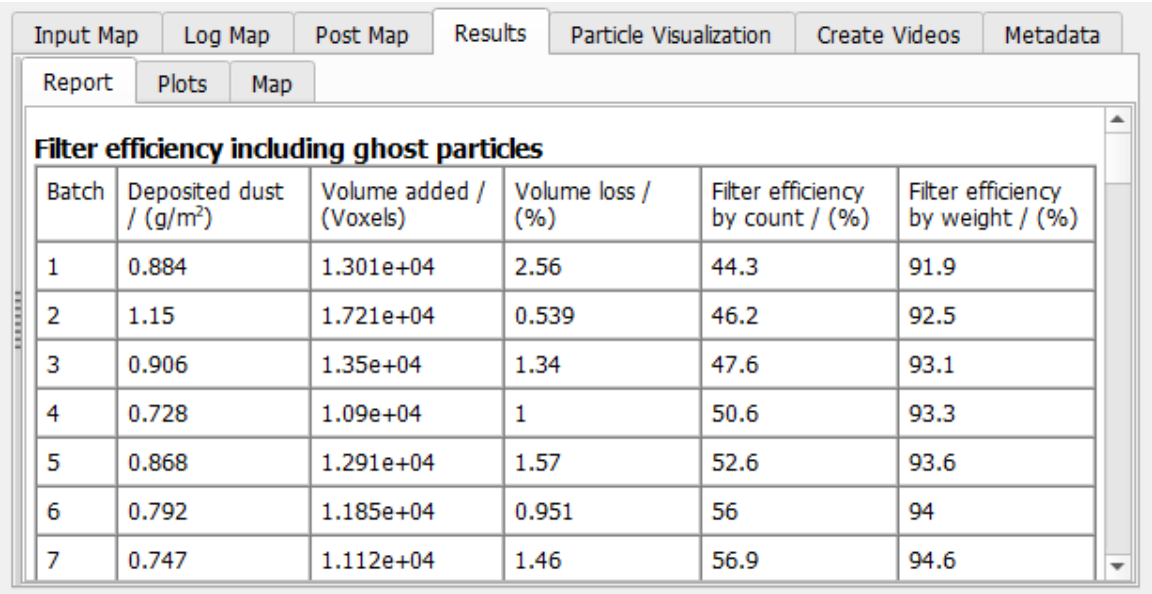


For the computation of the efficiency, **Efficiency results with ghost particles** controls how the results with and without ghost particles are displayed in the tables. Selecting **Excluded and (Included)**, the tables contain the efficiency without ghost particles and

additionally the efficiency with ghost particles in brackets. The other two options, **Excluded** or **Included**, show either the result with or without ghost particles. This option is only available if the user selected to run the simulation with ghost particles.



The **Efficiency Computation** panel allows to select the definition of the filtration efficiency. It can either be based on the number of particles which leave the domain via the outflow region or on the number of particles which are captured. The difference between both definitions is the handling of particles which are still in the domain but are not (yet) marked as captured.



Filter efficiency including ghost particles

Batch	Deposited dust / (g/m²)	Volume added / (Voxels)	Volume loss / (%)	Filter efficiency by count / (%)	Filter efficiency by weight / (%)
1	0.884	1.301e+04	2.56	44.3	91.9
2	1.15	1.721e+04	0.539	46.2	92.5
3	0.906	1.35e+04	1.34	47.6	93.1
4	0.728	1.09e+04	1	50.6	93.3
5	0.868	1.291e+04	1.57	52.6	93.6
6	0.792	1.185e+04	0.951	56	94
7	0.747	1.112e+04	1.46	56.9	94.6

The **Volume loss** is the particle volume lost during each batch due to the overlapping of particles. This value is a good indicator whether the number of particles per batch has been chosen correctly. If the Volume Loss is too high, the number of particles per batch should be reduced to achieve a sufficient accuracy of the computation.

The overall volume loss is reported below the Filter Efficiency table:

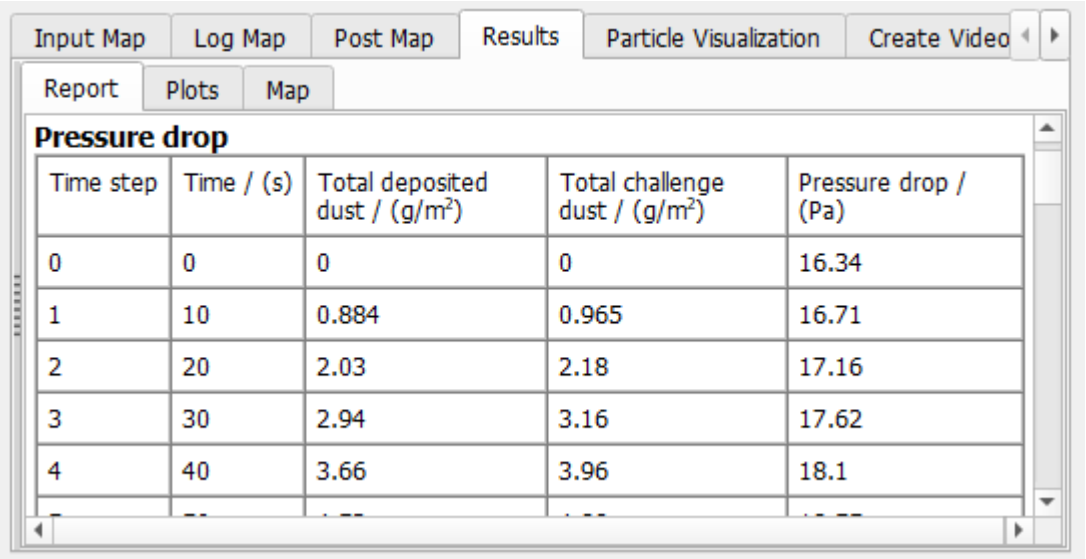
```
Volume loss: overlapping particles share volume, which is lost when converting to voxels.
Overall volume loss is: 1.42%

Overpacked: particle volume higher than adjusted max. packing density.
Overall overpacked volume is: 4.8e+05 voxels
That is a percentage of: 33.5%

Filter efficiency = 1-(NumberOfParticlesOutflow) / (NumberOfParticlesInflow).
Note: there may exist time-out particles in the medium.
```

For simulation with **Mixed** (unresolved) particles, also the number of voxels filled by a higher volume fraction than the defined maximal packing density is reported. If some

particles are smaller and some are larger than the voxel length, the number of overpacked voxels is typically very large. All locations where a large particle is deposited are overpacked, because a large particle fills up several voxels completely.

Pressure drop

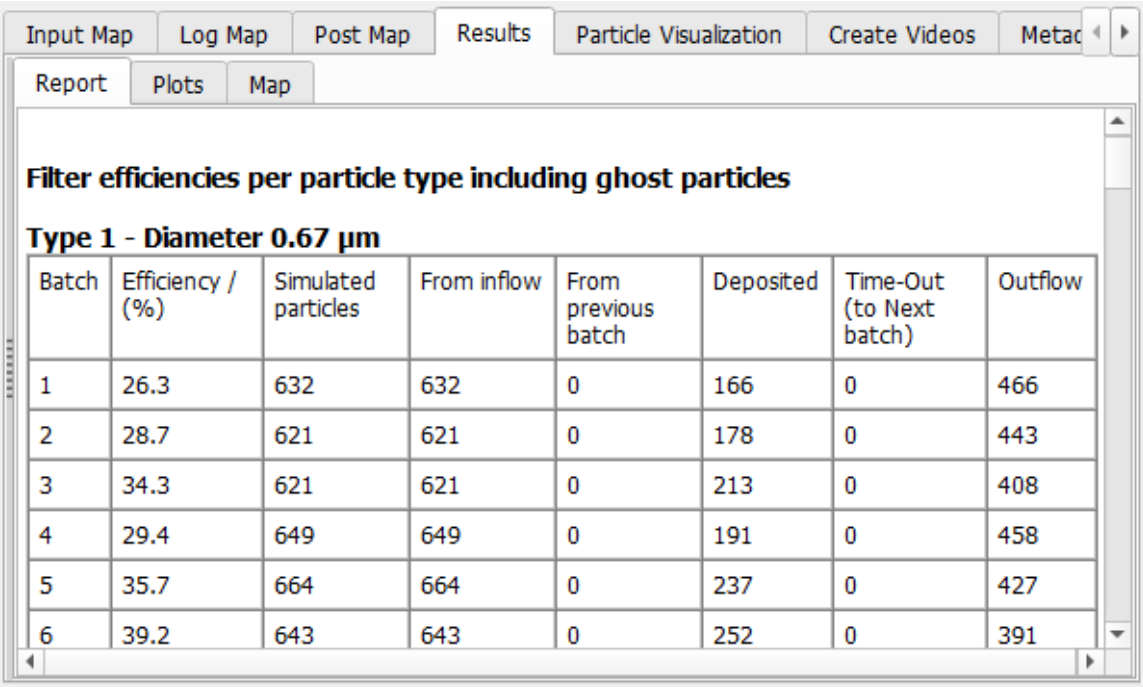
The **Pressure drop** table contains the **Total deposited dust**, the **Total challenge dust**, and the **Pressure drop** for every batch.



Time step	Time / (s)	Total deposited dust / (g/m²)	Total challenge dust / (g/m²)	Pressure drop / (Pa)
0	0	0	0	16.34
1	10	0.884	0.965	16.71
2	20	2.03	2.18	17.16
3	30	2.94	3.16	17.62
4	40	3.66	3.96	18.1

Filter efficiencies per type

The **Filter efficiencies per particle type** tables show the filter **Efficiency** of a certain particle type for every batch. It shows the overall number of **Simulated particles**, consisting of particles entering **From inflow**, or continuing movement **From previous batch**. At the end of the batch, particles are either **Deposited**, are still moving inside of the structure (**Time-Out)**, or have left the domain through the **Outflow**.



Filter efficiencies per particle type including ghost particles

Type 1 - Diameter 0.67 µm

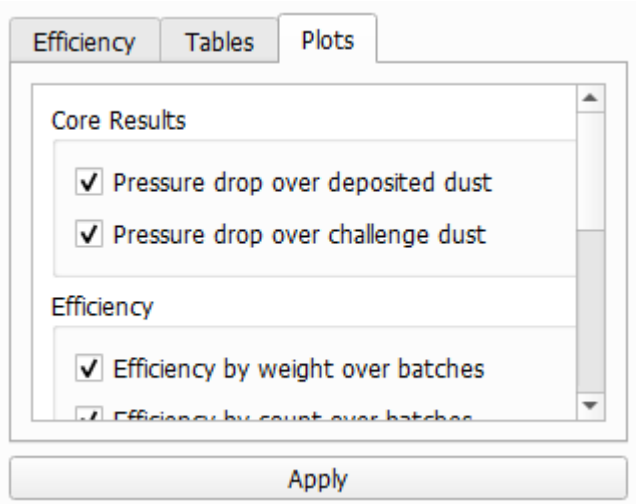
Batch	Efficiency / (%)	Simulated particles	From inflow	From previous batch	Deposited	Time-Out (to Next batch)	Outflow
1	26.3	632	632	0	166	0	466
2	28.7	621	621	0	178	0	443
3	34.3	621	621	0	213	0	408
4	29.4	649	649	0	191	0	458
5	35.7	664	664	0	237	0	427
6	39.2	643	643	0	252	0	391

Depending on the choice of input parameters, some particles may neither count as **Deposited**, **Time-Out**, or **Outflow**, e.g., because they leave the domain through the inflow area. In this case, those particles are removed completely from the reported statistics Those particles are not included in the number of **Simulated Particles**.

If in a batch the number of **Simulated Particles** of one particle type (diameter) is 0, the efficiency of this particle type cannot be computed from the statistics (0/0 particles are filtered). In that case, the efficiency of this type is set to the efficiency of the next particle type (next larger diameter). If no particle with a larger diameter exists, the efficiency is assumed to be 100%. Use **Ghost Particles** to assure that a sufficiently large number of particles of each type is simulated in every batch.

Plots

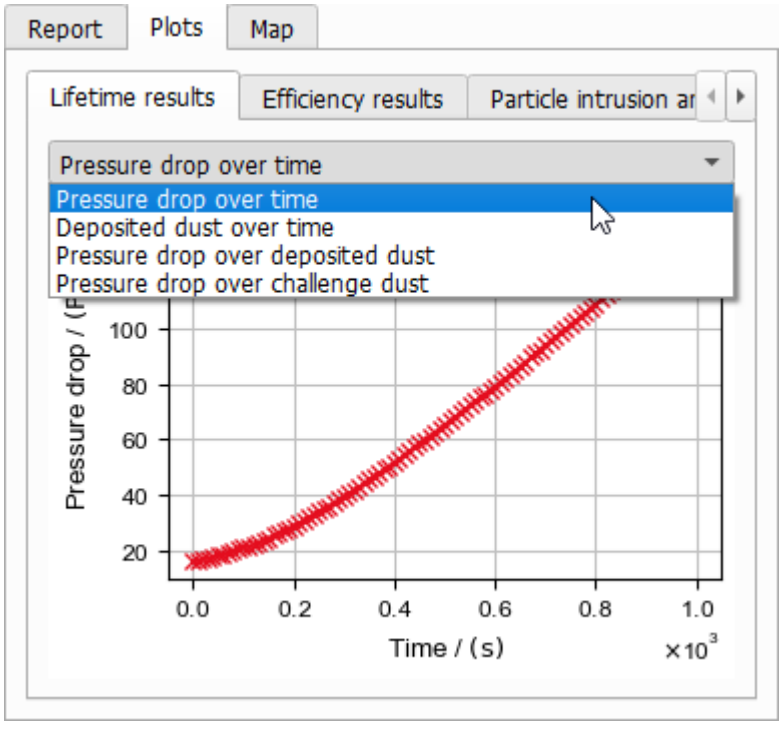
The **Plots** tab always shows the two plots **Pressure drop over time** and **Deposited dust over time**. You can select additional plots in the **Plots** tab in the post-processing section on the left hand side. Simply select the plots to be created and click the **Apply** button.



The plots appear sorted into sub-tabs in the **Plots** area on the right hand side. Up to two plots are shown above each other. If more plots are generated, a dropdown-menu allows to select the plot to be shown. The following plots are available:

Core Results: pressure drop and deposited dust

The **Pressure drop over time** and **Deposited dust over time** are always shown in the **Lifetime results** tab. **Pressure drop over deposited dust** and **Pressure drop over challenge dust** appear additionally if selected.

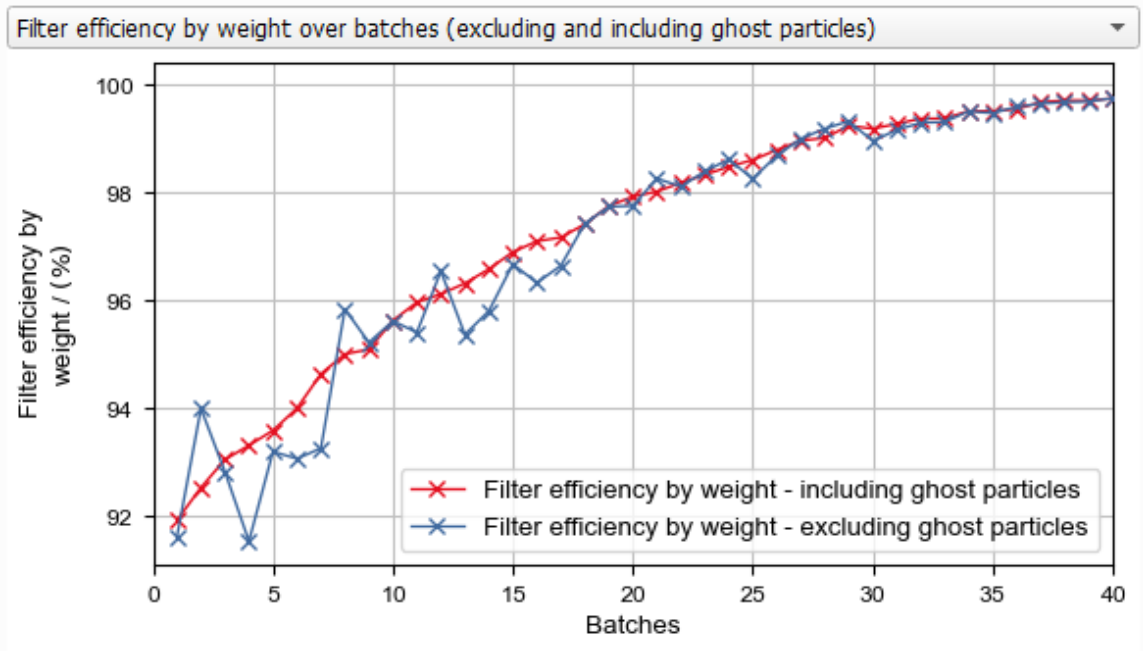


Efficiency

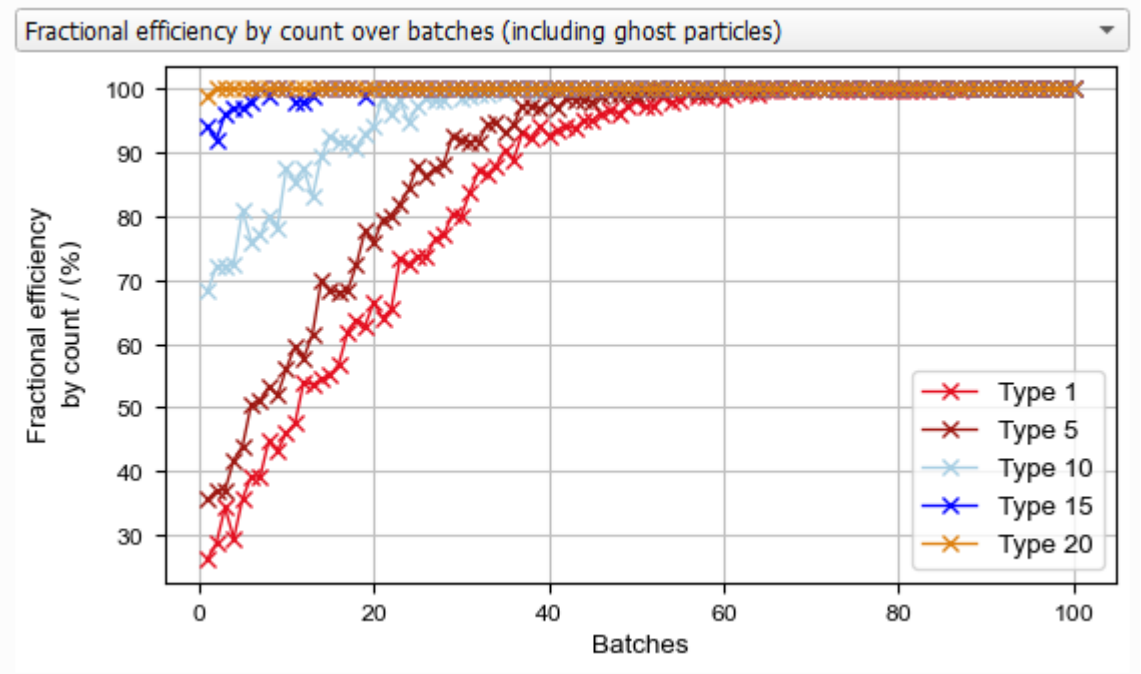
You can plot the filter efficiency development over the computed batches, either

computed by weight or by count. **Efficiency by count** is typically dominated by the small particles, whereas **Efficiency by weight** mostly depends on the large particles.

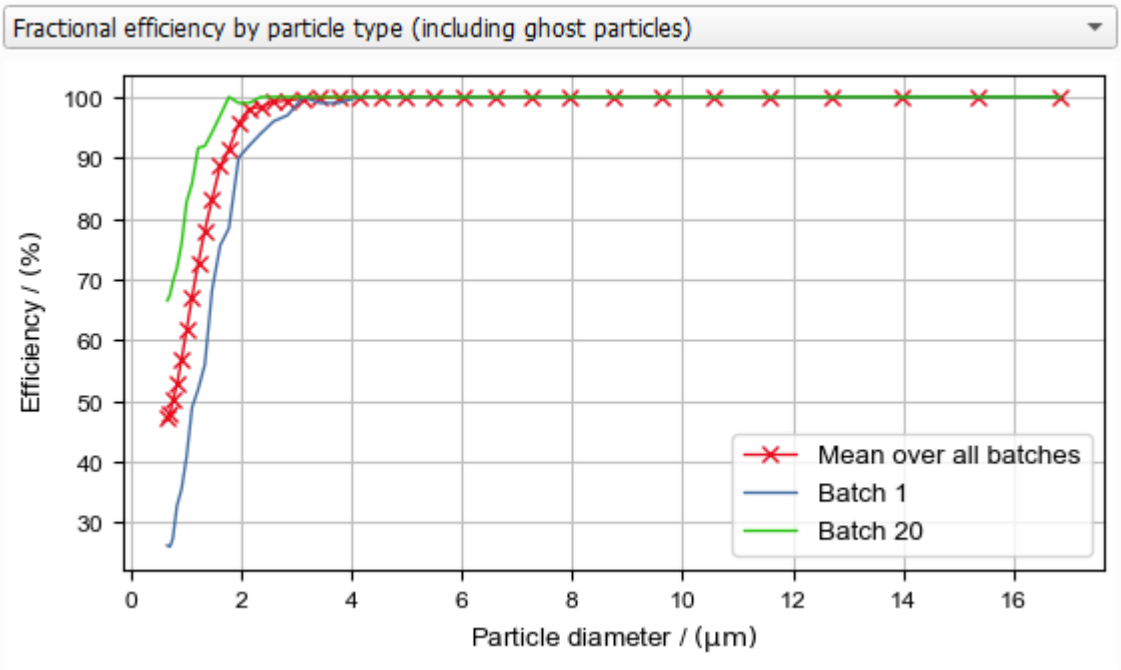
The plots respect the choices made on the **Efficiency** tab regarding the computation of the efficiency and the inclusion of ghost particles. If you select **Excluded and Included** for the ghost particles, you can see the graphs for both options and compare the results.



The **Fractional efficiency over batches** plot shows the evolution of the fractional filter efficiency over time. For each particle type, an individual efficiency curve is plotted.



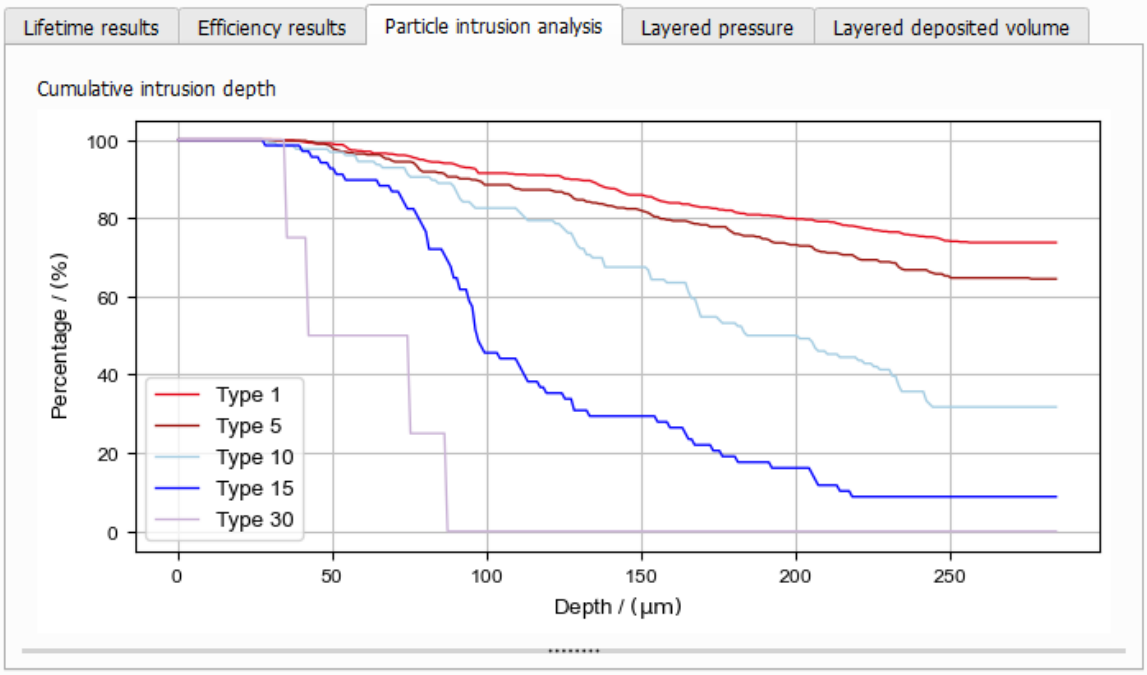
The **Efficiency over particle type for first XX batches** plot shows the fractional efficiencies plotted over the particle sizes on the x-axis. The plot shows the curves for each of the batches and also the mean over the selected range.



Geometric Analysis: Particle intrusion analysis

Two plots are shown on the **Particle Intrusion Analysis** plot. The **Cumulative Intrusion Depth** shows for each particle type (i.e., particle size), how far particles of this size move into the filter before they are captured. Starting on the left, at Depth 0 still 100% of the particles are unfiltered. When moving into the filter, the percentage of unfiltered particles goes down, until at the downstream side the particles reach the outflow.

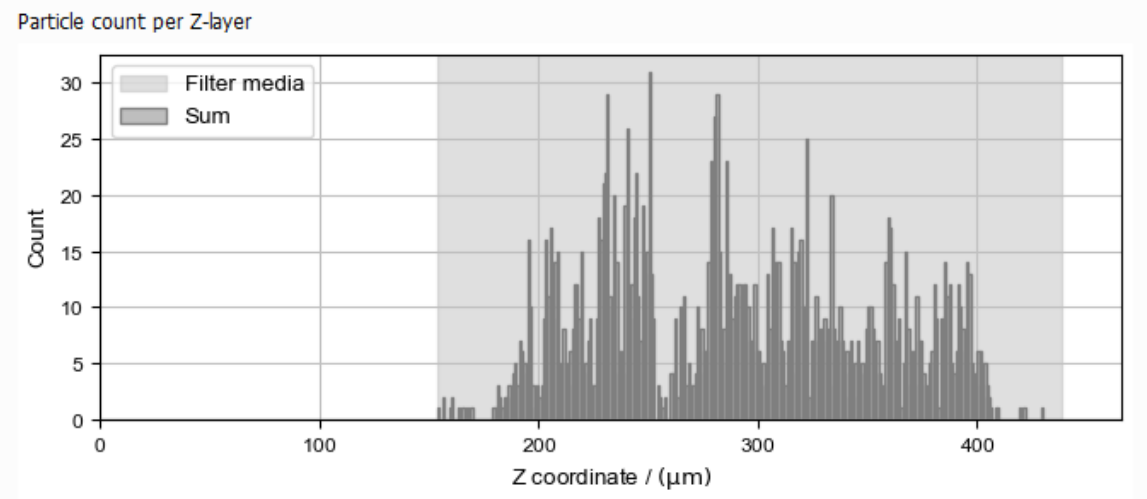
Important! The plot only takes the results of the first batch into account when the filter is still clean.



Right-click into the plot area to open a dialog box that lets you select which graphs are shown. For each particle type, you can additionally select to plot the fitted exponential decay function:

If the filter material is homogeneous over the depth of the filter, the probability that a moving particle hits a fiber is constant during the movement through the filter. In that case, the particle intrusion depth should follow an exponential decay, e.g., 50% are captured after the first 50 µm, 75% after 100 µm, 87.5% after 150 µm and so on. Therefore it is sensible to fit an exponential decay function through the computed intrusion depths as a model for the filter efficiency. The reported **Depth Efficiency** is the filtration efficiency predicted by this model.

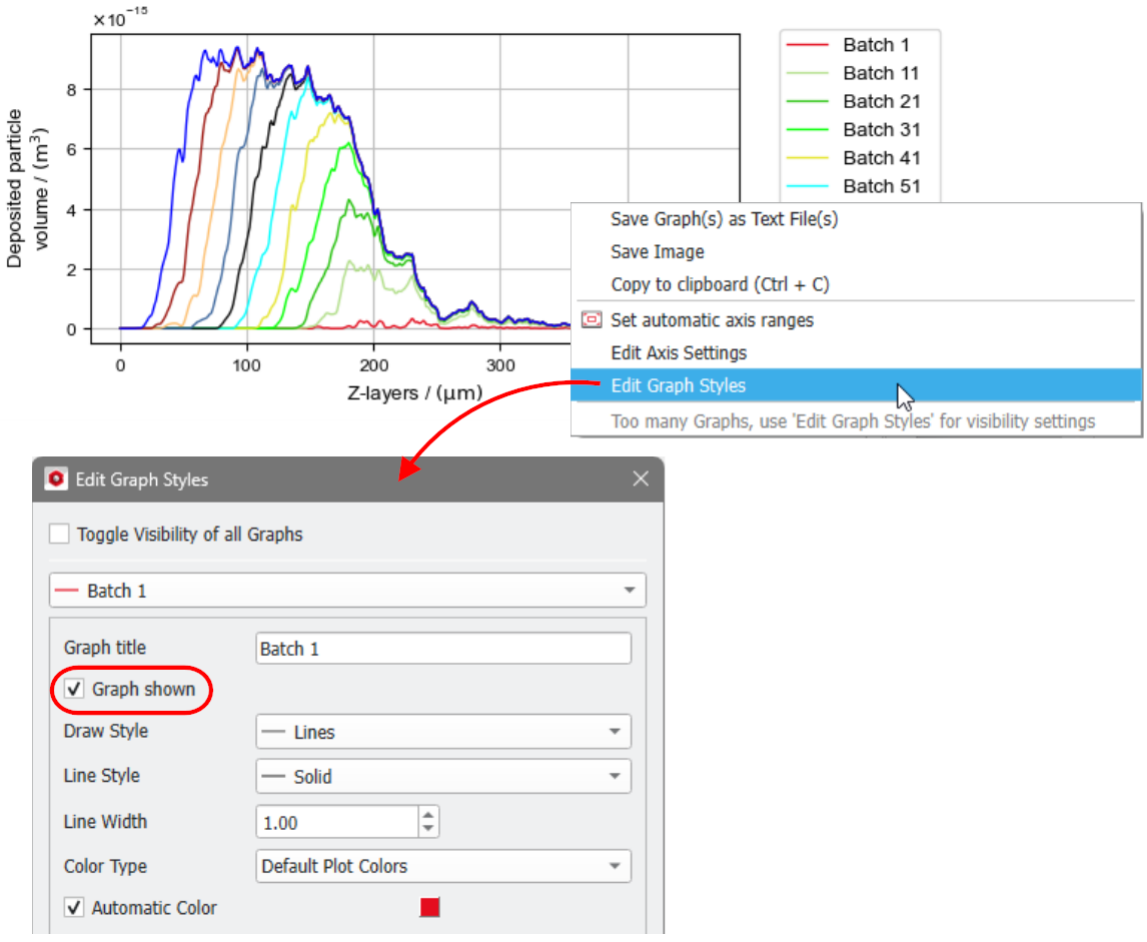
On the **Particle count per Z-layer** plot, the distribution of the collected particles over the filter height is plotted. For each particle type, the number of particles deposited on each Z-layer is counted.



Only the results of the first batch are included in this graph, too. Ghost particles are not taken into account. Be aware that the x-axis of the **Particle count per Z-layer** plot includes inflow and outflow area, while the x-axis of the **Cumulative intrusion depth** plot shows only the filter media (the range marked in gray in the lower plot).

Geometric Analysis: Layered deposited volume over batches

The plot shows the distribution of the deposited mass over the height of the filter media (z-direction) for each batch.



By default, a selection of time steps (batches) is shown. Select which steps are shown by right-clicking into the chart and selecting **Edit Graph Styles**. In the opening dialog, select in the drop-down menu the corresponding graph and switch the visibility on or off by checking the **Graph shown** box.

Geometric Analysis: Layered pressure over batches

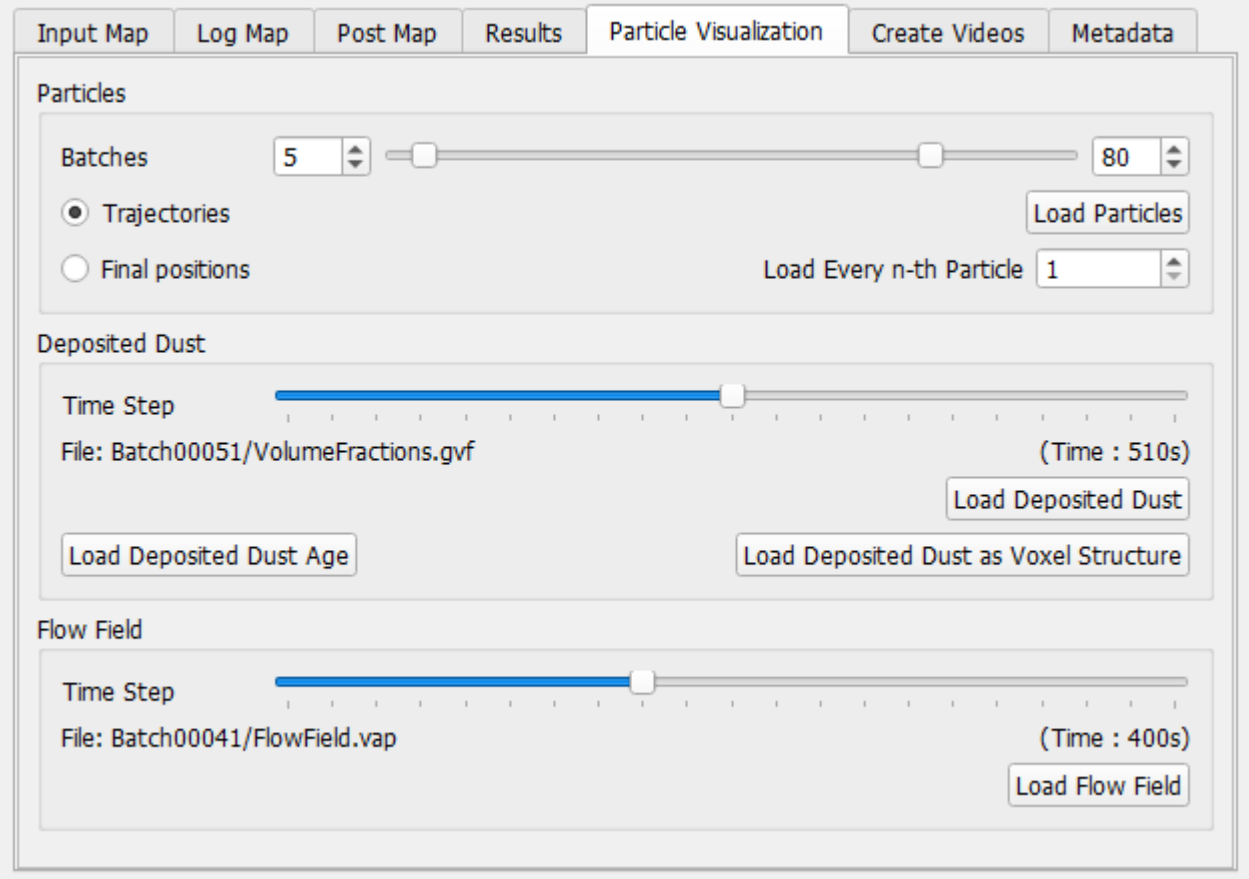
The plot shows the average pressure per voxel layer in the structure for every time step. A steep increase in a certain layer indicates that the filter is blocked in that layer.

Convergence over batches

The **Convergence** tab displays the convergence of the flow solver over the iterations for every time step (which was also shown in the progress dialog while the solver was running). The convergence chart can be a tool to check if the flow solver parameters were chosen correctly.

Particle Visualization

To visualize the initial and final positions of particles, track their trajectories, and study the flow of the fluid through the filter structure, select the **Particle Visualization** tab.



Particles

In the **Particles** panel, select one or several succeeding batches with the sliders. Select **Trajectories** to animate the particle motion from inflow to outflow or deposition place. If a large number of particles was simulated, loading all the trajectories at once is memory consuming and may not be possible. Loading only the **Final positions** does not allow to animate the particle motion, but is less memory consuming. Memory consumption may also be reduced by visualizing only a fraction of the simulated particles per batch by loading only **Every n-th Particle**.

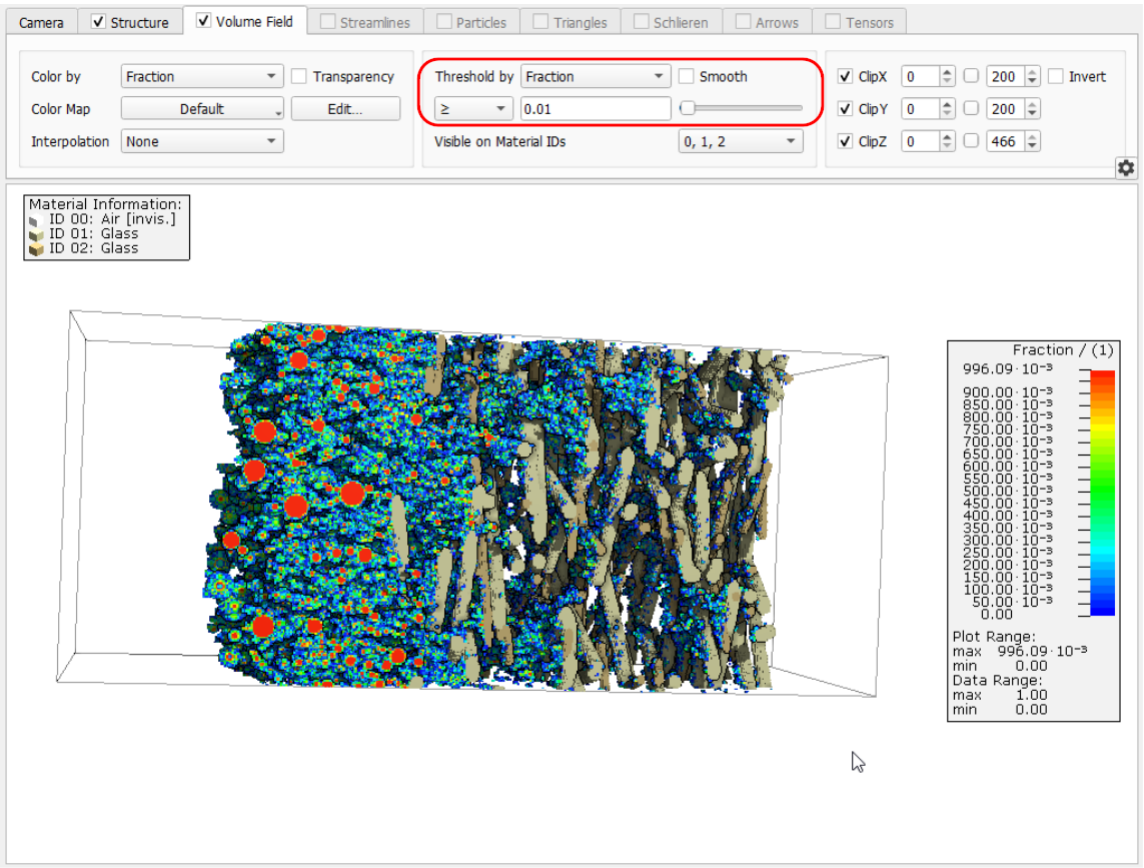
Clicking **Load Particles** loads the selected trajectories or final particle positions and displays the particles in the viewport.

Deposited Dust

Use the slider to select the **Time Step**. There are three options to visualize the deposited dust from the simulation:

Load Deposited Dust

With **Load Deposited Dust**, a result field (*.gvf) is loaded that contains for every voxel a single double value in the range [0,1], describing the volume fraction of the voxel filled with dust particles.



The visualization settings can be chosen under the **Volume Field** tab in the visualization panel. In the figure, **Threshold by Fraction** is selected to show only areas where a voxel is filled by more than 1% (pull-down menu set to ≥ and slider moved to **0.01**). Unchecking **Smooth** shows every voxel with its individual color without any interpolation and surface smoothing.

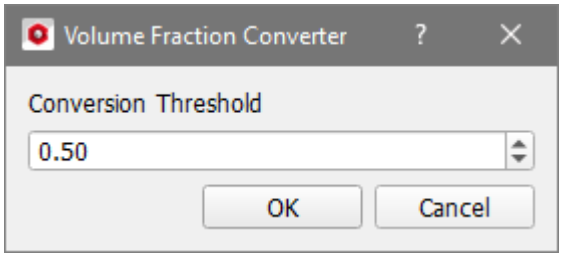
To see also the dust accumulation inside of porous materials, either switch off the visualization of the structure completely, or turn the rendering in the **Structure** tab to transparent.

For more information about the visualization option, see the Visualization user guide.

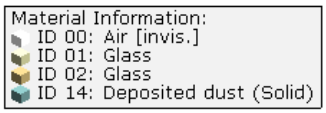
Load Deposited Dust as Voxel Structure

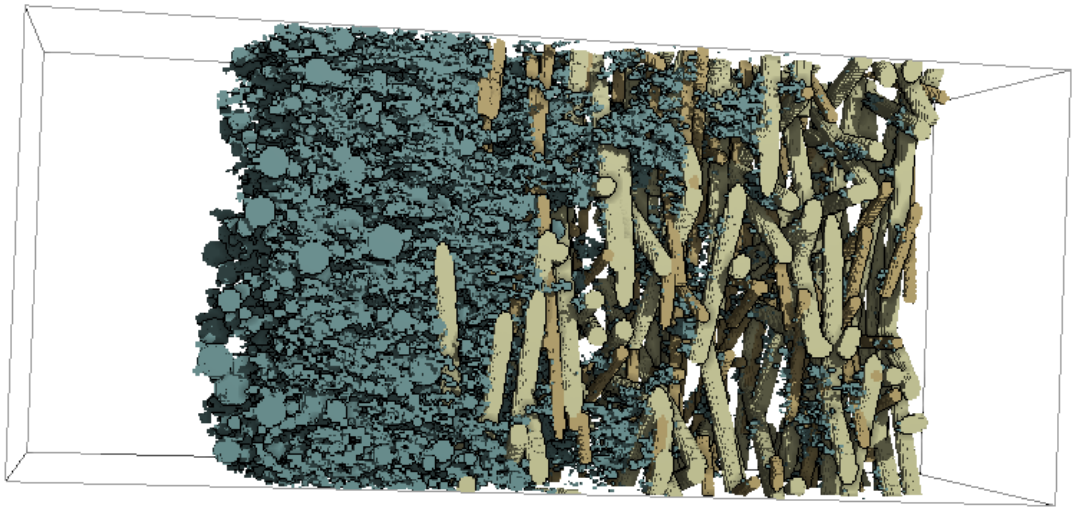
With **Load Deposited Dust as Voxel Structure**, the volume field is converted to a voxel structure.

To convert the double valued 3D field stored inside the *.gvf file into a structure, a conversion threshold is needed, and must be entered by the user. Enter a value between 0 and 1 here.



All voxels filled by at least the entered fraction will be turned into a solid material with Material ID 14.



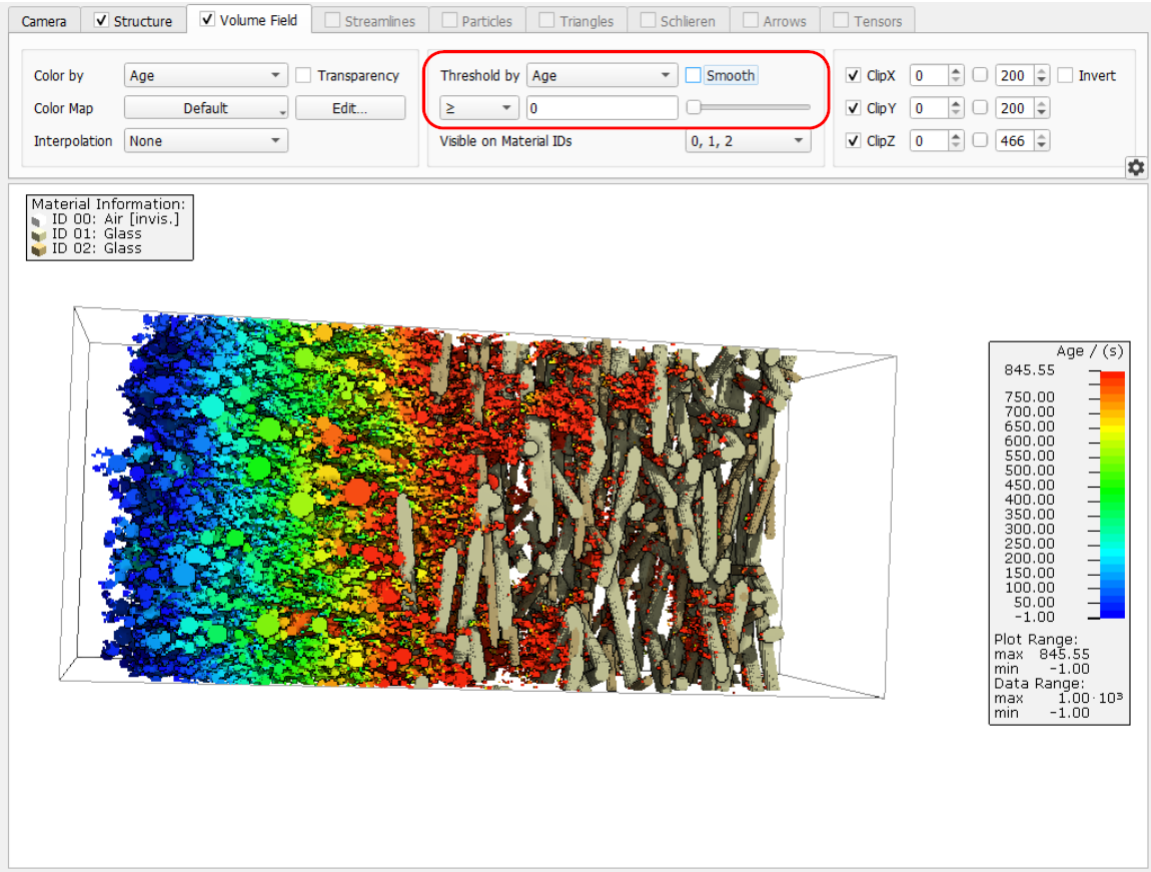


Load Deposited Dust Age

With **Load Deposited Dust Age**, a result field is loaded that contains for every voxel a single integer value describing how many batches ago this particle was deposited. Voxels filed in the last batch contain the value zero, voxels filled in the previous batch the value 1, and so on...

Under the **Volume Field** tab, select **Threshold by Age**, set the pull-down menu set to ≤ and move the slider to **0**. Uncheck **Smooth** to see every voxel with its individual color without any interpolation and surface smoothing.

The imported data contains the deposited dust at the end of the simulation, it is not possible to load intermediate time steps. The value in each voxel denotes the time that has passed since the voxel was filled. In the picture below, particles that entered the filter first are colored red, particles that entered last are colored blue.



Flow Field

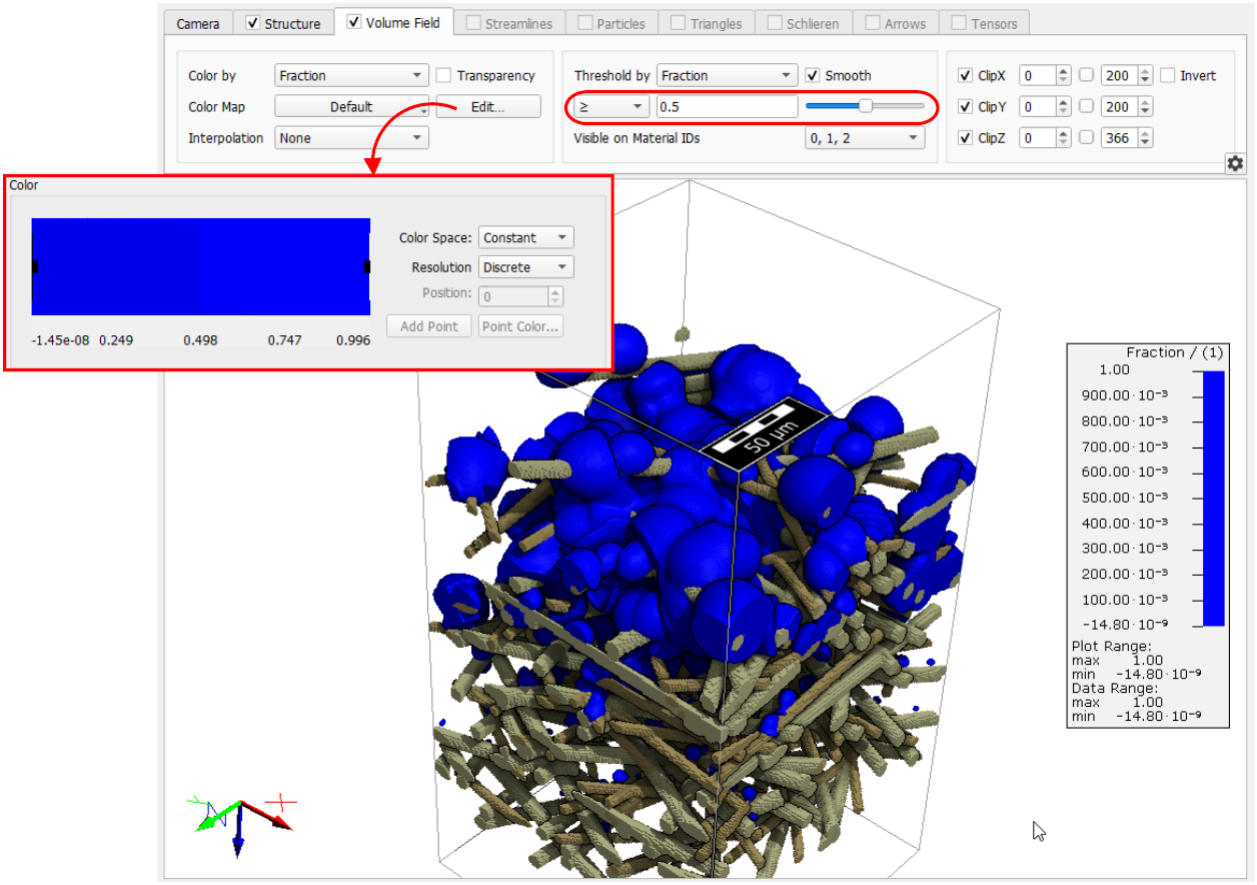
Use the sliders to select the corresponding time step. Click **Load Flow Field** to load the result and refer to the Visualization user guide for details.

Visualization of Coalescence Results

When coalescence is enabled in the simulation, some of the standard visualization options will only show a partial result and give a wrong impression of the simulation. Visualizing particles (either as **Trajectories** or as **Final Positions**) is not recommended. This will show only the movement of the oncoming batch of particles and where they deposit on the filter. It will not show how the particles coalesce with the already deposited particles and how they attach to the filter. Especially, the position of a deposited particle is not shown correctly, as this position changes whenever a new particle coalesces with it.

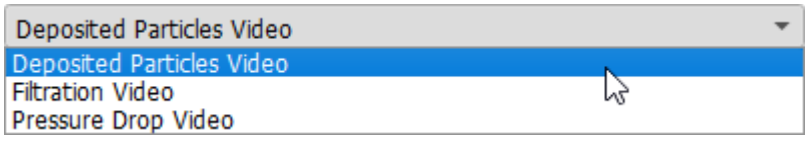
To visualize the position of the deposited particles correctly, use **Load Deposited Dust**. Set the Threshold to ≥ 0.5, and GeoDict will show all voxels that are filled by more than 50% with the deposited fluid.

By default, GeoDict will color each voxel depending on the volume of fluid contained in it. To change this to a single color, click on the Color Map **Edit...** button and select **Constant** as **Color Space**.

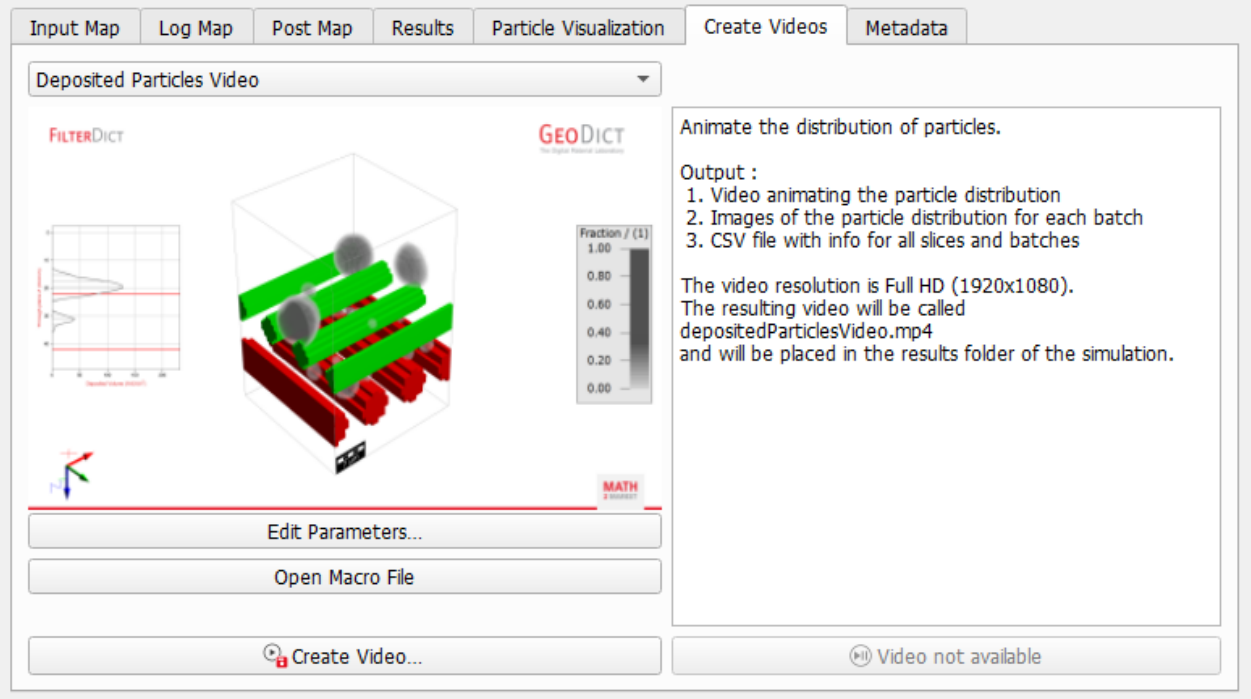


Create Videos

Under the **Create Videos** tab, three predefined videos can be generated. In the drop-down menu, select either the **Deposited Particles Video**, the **Filtration Video**, or the **Pressure Drop Video**.



On the left of the dialog, a simple animation shows the type of movie that is created. On the right, a short description explains the contents of the movie.



Each video is created using a python macro placed in the folder `C:Program Files\Math2Market GmbH\GeoDict 2026\VideoMacros\FilterDict\MediaLifetime`.

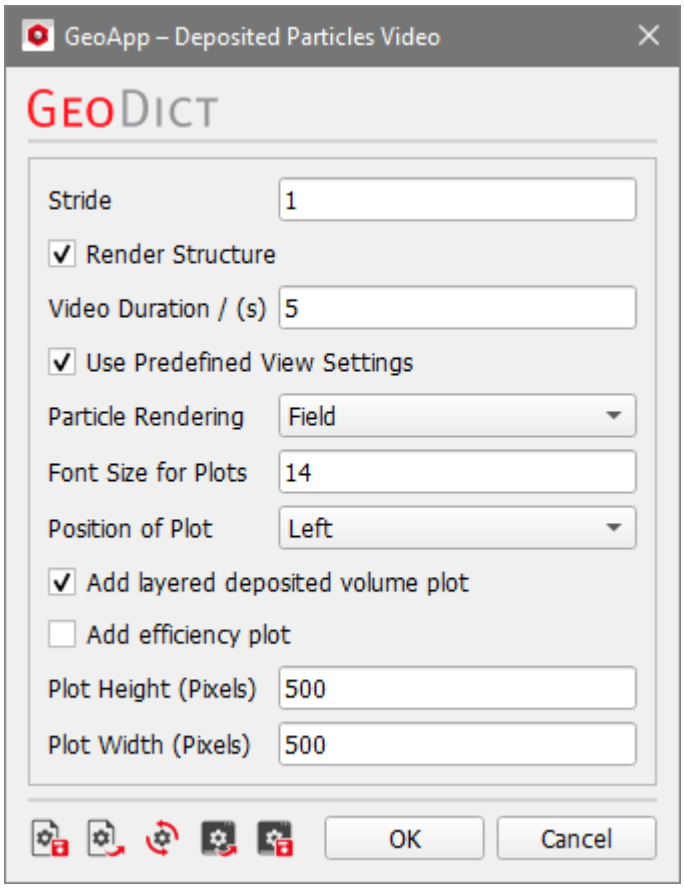
Important! Video generation is not possible if the GeoDict-Tools package has not been installed.

By clicking on **Open Macro File**, the macro is opened in a text editor. It is not necessary to edit the macro to create the standard video, but you can make a copy of this macro and use it as a starting point for your own video generator.

Click on **Edit Parameters...** to open a dialog showing the parameters available in the macro. After selecting the parameters, click on **Create Video...** to start the generation of the movie. The mp4 file is automatically saved to the result folder of the filtration run, and can be copied from there. Alternatively, the video can be started directly from GeoDict through the **Play Video** button which replaces the grayed out **Video not available** button after a successful video generation.

Deposited Particles Video

This video animates the distribution of particles. It requires that either the **Final Particle Positions** or the **Deposited Dust Volume Fractions** were kept for all batches that are to be shown in the video. This must be selected in the Output options 104 before starting the filtration simulation.



The **Stride** defines the batch increment between two frames, i.e, a stride of 1 means that every batch will be rendered, while a stride of 3 means that only the results of every third batch will be rendered in the video.

Uncheck **Render Structure** if the video should only show the deposited particles, but not the filter structure.

The **Video Duration** is the overall length in seconds that the created mp4 file will play.

Check **Use Predefined View Settings** if a predefined camera position should be used, uncheck it if you want to use the current camera position.

Particle Rendering can be done in three different ways:

- **Particles**: corresponds to using **Load Particles** in the Particle Visualization tab
- **Field**: corresponds to using **Load Deposited Dust** in the Particle Visualization tab
- **Voxel**: corresponds to using **Load Deposited Dust as Voxel Structure** in the Particle Visualization tab

The video combines plots with a visualization of the particles. Two different plots may be selected with the **Add layered deposited volume plot** and the **Add efficiency plot** options. The position, height, width and font size of the plots can also be selected.

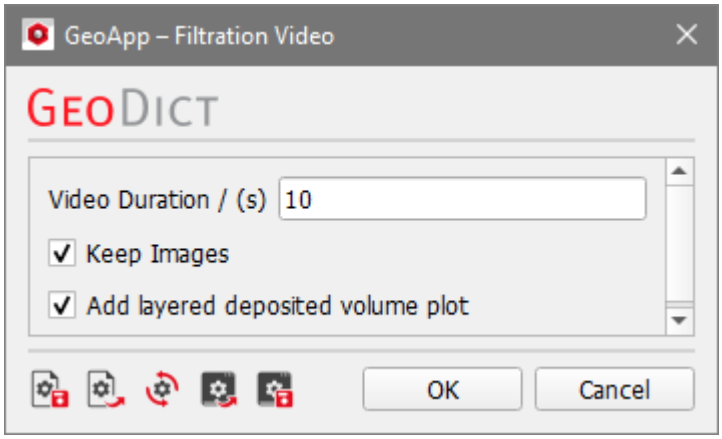
Start the video generation by clicking **Create Video....** A pop-up dialog appears with some information.

Generating the video requires that certain output files have been created. Depending on the selected Output options 104, some files may not have been stored for all batches, but only after every nth batch. In this case, the script automatically changes the **Stride**, such that the necessary files are available for the rendered batches. E.g., if the volume fractions files were only stored every 5th batch, the **Stride** will be changed to 5 if **Particle Rendering: Field** is selected.

When the video generation is finished, a pop-up window appears that shows the location of the generated file.

Filtration Video

Generating this video requires certain output files. Select **Keep all** for **Trajectories** and **Deposited dust volume fractions** on the Output tab 104 prior to running the simulation if you want to create this video afterwards.



The **Video Duration** is the overall length in seconds that the created mp4 file will play.

Uncheck **Keep Images** to remove the created .png image files after the video was generated. In this case, only the .mp4 file fill be saved.

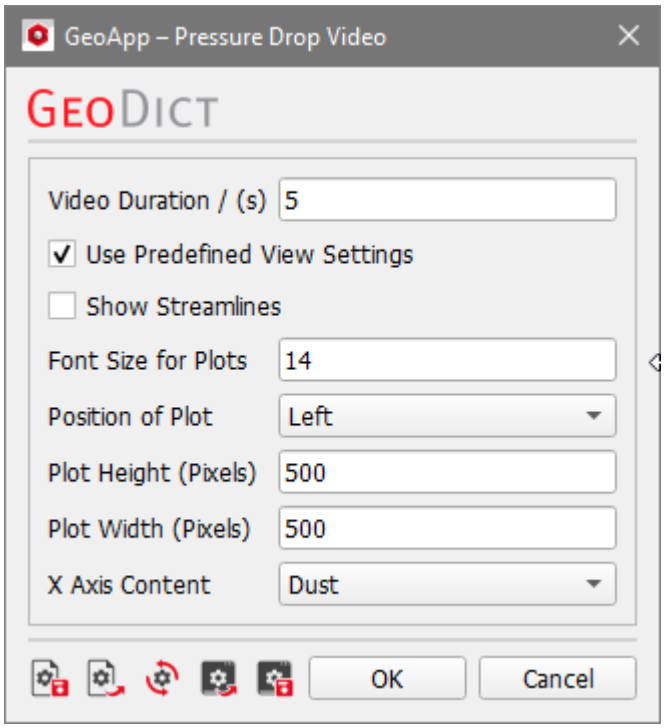
Select **Add layered deposited volume plot** to include a plot on the left that shows the distribution of the deposited mass per Z-layer.

The video combines plots with a visualization of the particles. It is particularly suited to visualize coalescence of particles as the deposited mass will be visualized by the corresponding volume fractions, and not as a collection of solid particles.

The created video will use the camera position currently set in the viewport.

Pressure Drop Video

This video animates the buildup of a pressure drop. It requires that the **Flow field** was kept for all batches that are to be shown in the video. This must be selected in the Output options 104 before starting the filtration simulation.



The **Video Duration** is the overall length in seconds that the created mp4 file will play.

Check **Use Predefined View Settings** if a predefined camera position should be used, uncheck it if you want to use the current camera position.

Check **Show Streamlines** to visualize the flow streamlines, too.

The video combines a plot of the pressure drop with a visualization of the particles. The position, height, width and font size of the plots can also be selected.

The pressure drop may be plotted over **Time** or deposited **Dust** on the **X-axis**.

1.3 Filter Element

Filter Element simulations are suited for the simulation of pleats, filter parts, or filter elements. At this scale, the details of the inner structure of the filter media are usually no longer resolved.

Commands available for filter element analysis

- Filter Lifetime 125
  A Filter Lifetime simulation includes clogging of the filter and computes pressure drop and deposited dust over time. In a Single Pass simulation, fluids pass through the filter only once and are not recirculated. In a Multi Pass simulation, fluids move in a circuit through the system, and particle size distribution and concentration in front of the filter change over time.

1.3.1 Filter Lifetime

A **Filter Lifetime** simulation includes clogging of the filter and computes pressure drop and deposited dust over time. In a Single Pass simulation, fluids pass through the filter only once and are not recirculated. In a Multi Pass simulation, fluids move in a circuit through the system, and particle size distribution and concentration in front of the filter change over time.

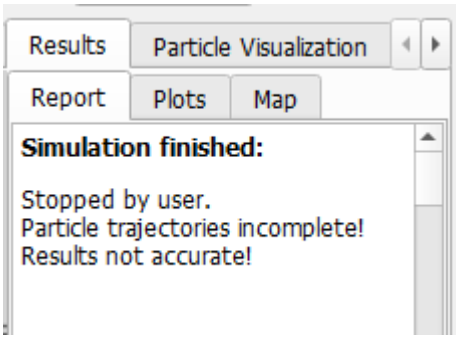
The Filter Lifetime command for filter elements

- Options 126
  How to set the input parameters for the Filter Lifetime command.
- Results 163
  How to understand the computed results.
- Particle Visualization 166
  How to visualize the computed results.
- Create Videos 168
  How to create videos from the computed results.

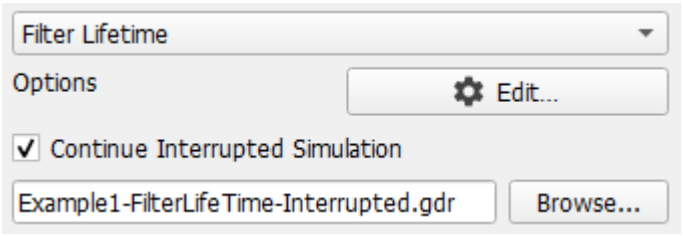
Continue Interrupted Simulation

An interrupted filter life time simulation can be restarted by checking **Continue Interrupted Simulation**. For this, at least the computation of the first flow field must have been finished. When the simulation was interrupted earlier, it must be started anew.

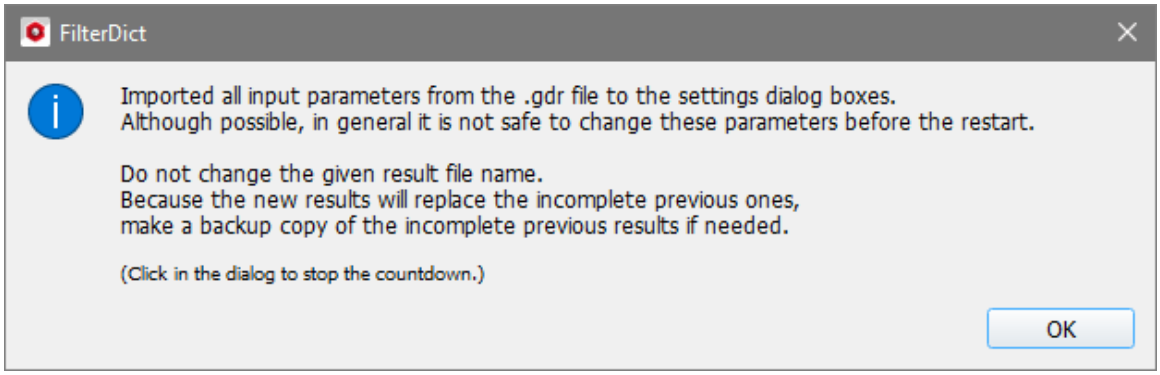
A stopped or interrupted simulation produces an incomplete (*.gdr) result file which may not automatically open at the end of the computations. When opened through **File → Open *.gdr File...**, you can find the information that the simulation was not finished under the **Results - Report** subtab:



To continue the unfinished simulation, check **Continue Interrupted Simulation** and click **Browse** to select the name of the result file. To continue the interrupted simulation, the result file from the unfinished simulation must be located in the project folder and the structure on which the simulation is to be finished must be already loaded.



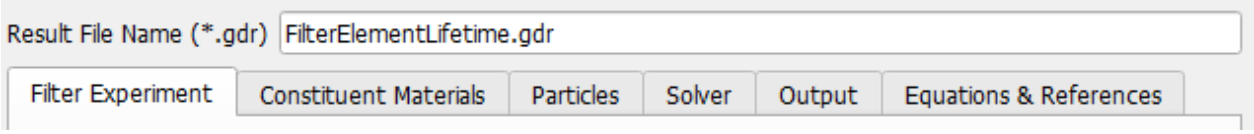
A message appears giving information and instructions on how to handle the simulation re-run.



Click **OK** in the message to return to the FilterDict section and click **Run** to continue the interrupted simulation..

Options

Select **Filter Lifetime** from the pull-down menu and click the **Option**'s **Edit...** button to open a dialog to enter or edit the simulation options. Choose a **Result File Name** for the result file (*.gdr) fitting to your current project.



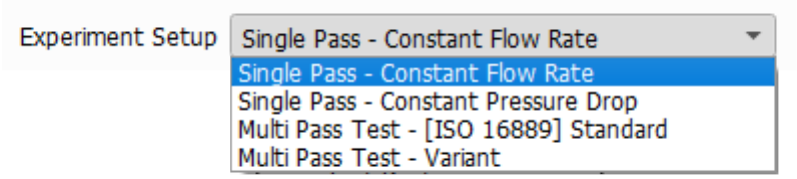
The parameters are organized under the tabs: **Filter Experiment**, **Constituent Materials**, **Particles**, **Solver**, and **Output**. The **Equations & References** tab gives a summary of the used equations and symbols which are explained in more detail in this user guide.

Tabs of the Filter Element Lifetime dialog

- Filter Experiment 127
  Select Single Pass or Multi Pass experiment, and set the boundary conditions.
- Constituent Materials 132
  Define fluid type and temperature. Set material IDs and parameters for porous layers.
- Particles 137
  Select particle sizes and their start positions. Set all parameters that govern the particle motion and define collision and adhesion models.
- Solver 149
  Select the flow solver, set the stopping criteria and define the computational resources.
- Output 161
  Choose which data will be stored for visualization.

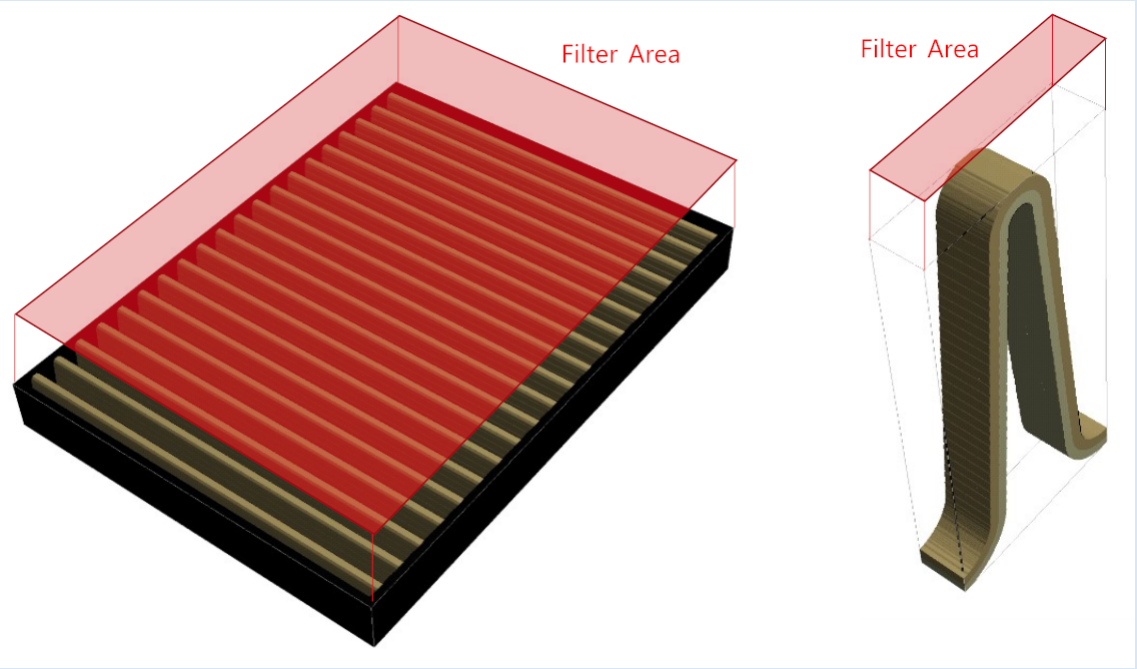
Filter Experiment

Under the **Filter Experiment** tab, choose between a **Single Pass** or a **Multi Pass** experiment. For **Single Pass**, the experiment can be run at a **Constant Flow Rate** or at a **Constant Pressure Drop** (often used for membranes). For **Multi Pass**, choose between the standard test setup or a variant with an initially contaminated test reservoir.



When a method is selected, the setup is shown schematically in the dialog below the pull-down menu. Furthermore, the selected method will influence which options are available elsewhere in the Filter Lifetime dialog.

For filter element scale simulations, the meaning of the Filter Area parameter must be clarified. Here, the **Filter Area** is the rectangular inflow area of the complete filter as illustrated in the figure below, it is not the surface area of the (pleated or otherwise folded) filter medium.



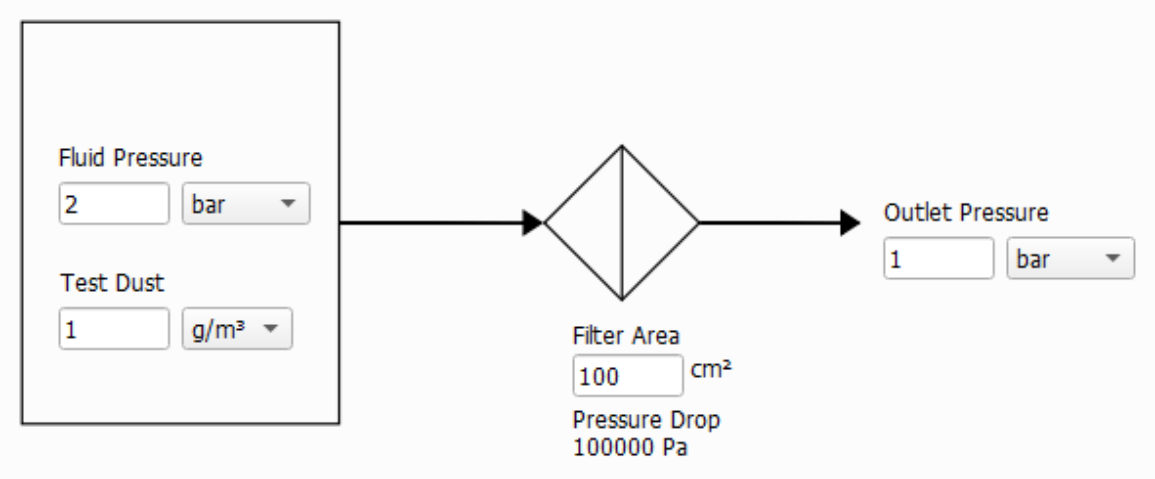
The filter area is needed to compute the flow velocity and the number of particles on the part of the structure that is simulated in GeoDict.

Single Pass - Constant Flow Rate

Single Pass - Constant Pressure Drop

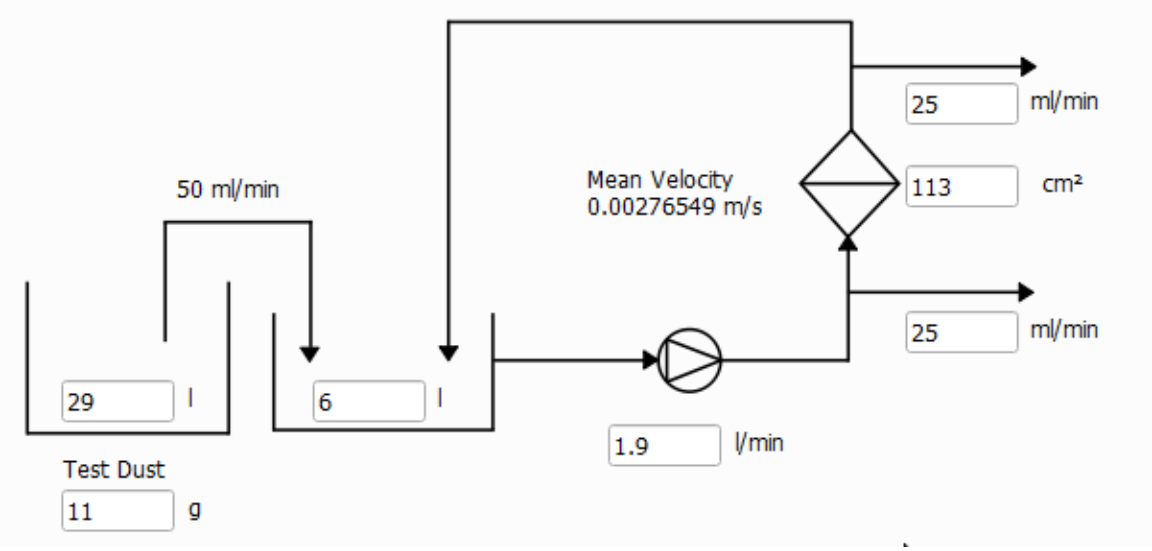
Here, the pressure drop at the filter is kept constant and the mean velocity / flow rate decreases over time.

For this choice, enter the **Fluid Pressure** in the containment, the **Outlet Pressure**, the **Test Dust** concentration in the fluid, and the test **Filter Area**. The **Filter Area** is the surface area of the filter in the experiment (and not the surface of the structure in GeoDict). The **Pressure Drop** used in the simulation is the difference between the inlet and the outlet pressure.



Multi Pass Test - [ISO 16889] Standard

In the standard multi pass test, the test reservoir is initially clean, and the test dust is injected into the system during the experiment.

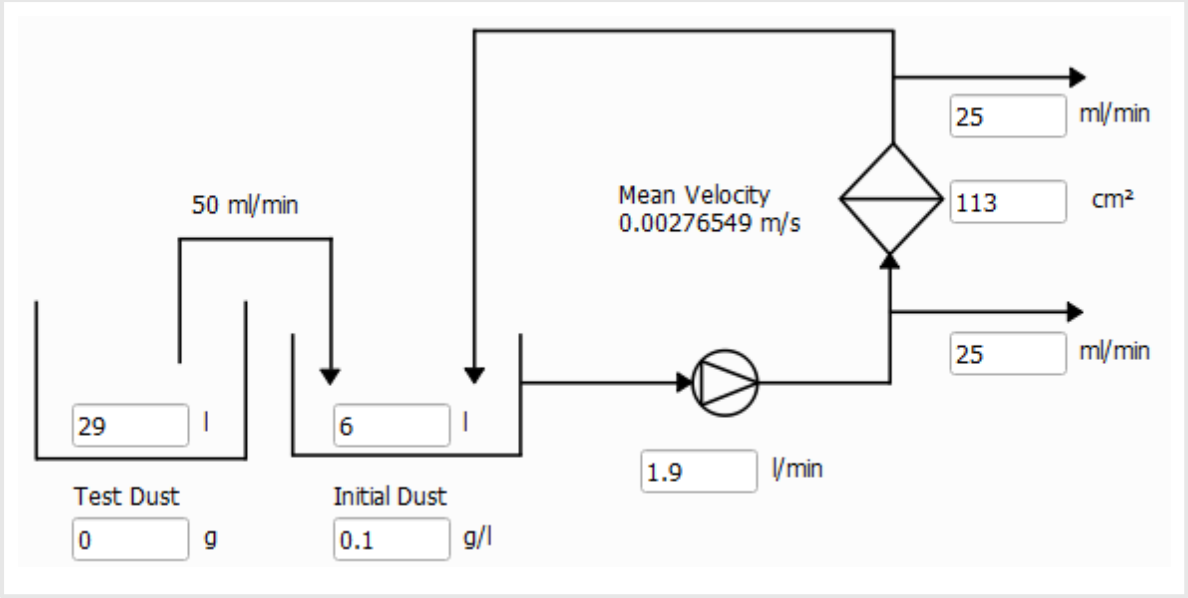


In the sketch, define the **Test Dust** (in g), the **Pump flow rate** (in l/min), and the **Test filter area** (in cm^2). Enter also the **Injection reservoir volume** and the **Test reservoir volume** (recirculation). The **Outflow to upstream sampler** and the **Outflow to downstream sampler** (in ml/min) can also be set. The **Injection flow rate** matches the sum of the upstream and the downstream outflows (in ml/min).

Hovering with the mouse over the edit boxes shows additional information in a tooltip.

Multi Pass Test - Variant

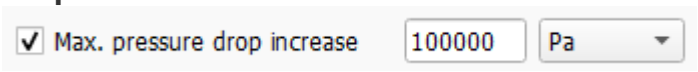
With this option, also the amount of dust in the **Test Reservoir** can be set in the beginning of the simulation, i.e. you can use this to model an initially contaminated test reservoir. Otherwise, this option and its parameters are the same as those of the [ISO 16889 standard multi-pass test](129).



Simulation Stopping Criterion

The simulation continues to run until one of the following criteria is reached:

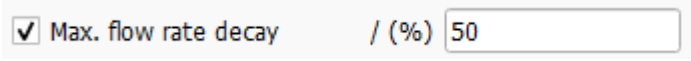
- If **Max. pressure drop increase** is selected, the simulation stops if the initial pressure drop is increased by the given value. This option is not available if **Single Pass - Constant Pressure Drop** was selected.

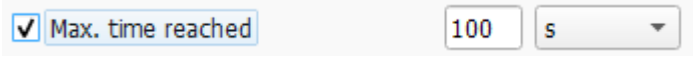
- If **Max. pressure drop** is selected, the simulation stops if the given pressure drop is reached. This option is not available if **Single Pass - Constant Pressure Drop** was selected.

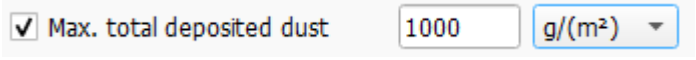
- If **Max. flow rate decay** is selected, the simulation stops when the flow rate falls below the given percentage (%) of the initial flow rate. This option is only available if **Single Pass - Constant Pressure Drop** was selected.

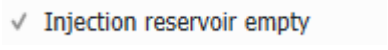
- If **Max. time reached** is selected, the simulation stops when the defined simulation time is reached.

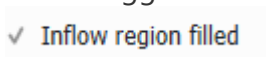
- If **Max. total deposited dust** is selected, the simulation stops when the defined amount of deposited dust is reached.

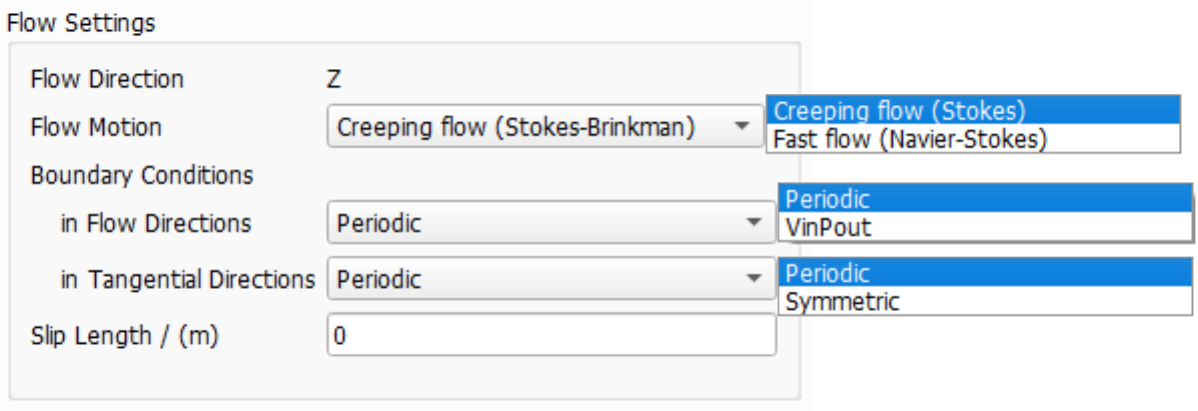
- **Multi Pass Test** simulations always automatically stop when the **Injection reservoir** is **empty**.

- The simulation always automatically stops when the **Inflow region** is **filled** with particles. This is triggered when the first deposited particle reaches the inflow plane.



Flow Settings



In FilterDict, the **Flow Direction** and, hence the filtration direction, is always the **Z**-direction.

The **Flow Motion** can either be creeping or fast flow. In the creeping flow case, the Stokes equations are solved to determine the flow field. If fast flow is chosen, the Navier-Stokes equations are solved to determine the flow field.

As the Stokes equation is a simplification of the Navier-Stokes equation, choosing **Fast flow (Navier-Stokes)** will give correct results also in the case of slow, creeping flow. Solving the Navier-Stokes equations is computationally more expensive than solving the Stokes equations, so it is advisable to choose **Creeping flow (Stokes)** whenever justified.

For **Creeping flow (Stokes)**, **Periodic,** and Velocity in - Pressure out (**VinPout**) can be chosen as **Boundary Condition** in flow direction. VinPout cannot be chosen for a **Single Pass – Constant Pressure Drop** experiment.

For **Fast Flow (Navier-Stokes)**, the boundary conditions in flow direction are set to Velocity in, Pressure out (**VinPout**) if **Constant Flow Rate** was chosen under the Filter Experiment tab, and set to **Periodic** if **Constant Pressure Drop** was chosen under the Filter Experiment tab.

In **Tangential Directions**, **Periodic** or **Symmetric** boundary conditions can be selected. For more information about the flow computation, please refer to the FlowDict user guide.

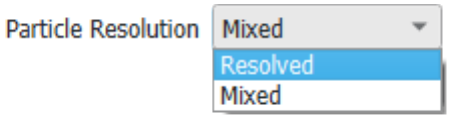
The choices made for the flow boundary conditions in tangential directions also determine what happens when a particle reaches the domain boundary in one of the tangential directions. With **Periodic** boundary conditions, the particle will leave the domain and reappear on the opposite site. With **Symmetric** boundary conditions, a particle will be reflected at the domain boundary.

The **Slip Length** allows to include sliding effects in the flow simulation. Typically, the slip length is of the same order of magnitude as the mean free path of the fluid (which is, e.g., 68 nm for air at ambient conditions) and can be neglected if this is much smaller than the voxel

length, which is typically the case for simulations on the filter element scale. A value of 0 corresponds to no-slip boundary conditions.

Constituent Materials

On the top of this tab, select if the **Particle Resolution** is **Resolved** or **Mixed**.



Choose **Resolved** if all particles are larger than the current voxel length, and **Mixed** (unresolved) if some particles are smaller than the current voxel length. For further explanation, see the Filter Clogging 14 page.

Temperature

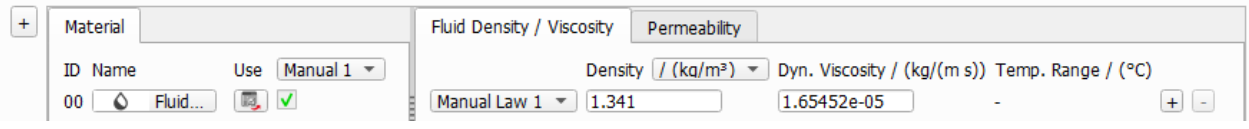
The **Temperature** for the filtration process is selectable in Kelvin (K), Celsius (°C), and Fahrenheit (°F). The chosen temperature may change the density and viscosity values of the fluid during the simulation, if those are defined to be temperature-dependent in the material database. Furthermore, if the Brownian motion of the particles is included in the simulation, it determines the strength of the Brownian motion through equation (13) 8.

Fluid Density and Viscosity

FilterDict uses single-phase flow simulations. It is not possible to run a flow simulation if two fluids are present in the structure. If multiple fluids are present in the structure, you may click **Set all Fluids to...** and select one fluid for the simulation.

The selected fluid is also displayed in the drop-down menu in the **Material** subtab behind **Use**. The density and the dynamic viscosity of this fluid are shown under the **Fluid Density/Viscosity** subtab.

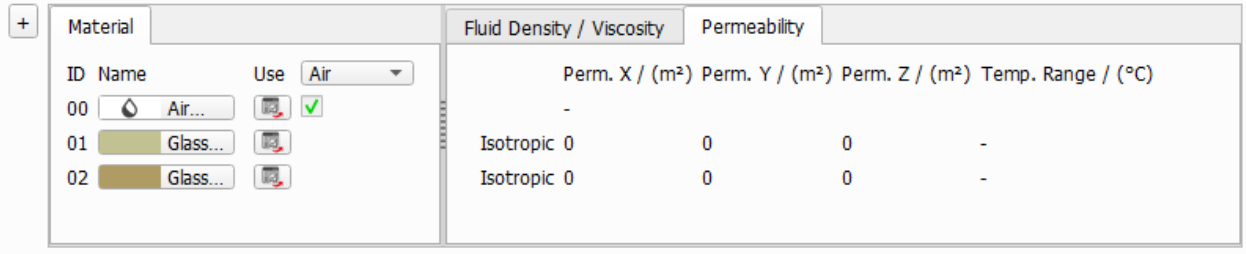
If a fluid from the GeoDict Material Database is used, the values for **Density** and **Dynamic Viscosity** are taken from the database and may depend on the given **Temperature** value. Select **Manual** as fluid material to enter parameter values manually.



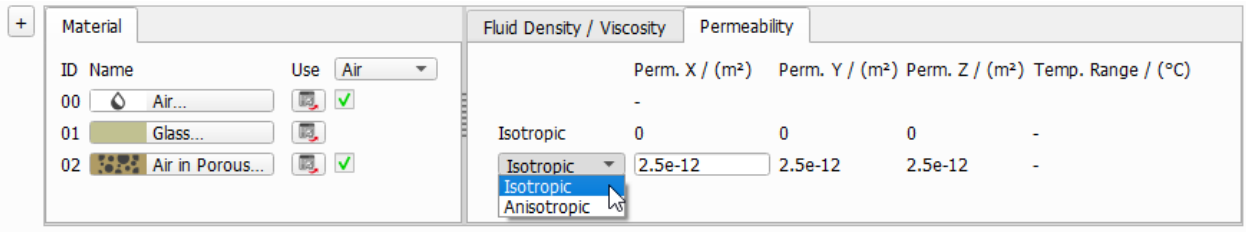
To add materials to the database, use the **Open & Edit Material Database** button. More information on editing, expanding, and using the **GeoDict Material Database** is available in the Material Database user guide.

Filter Media Permeability

Under the **Permeability** subtab, enter the permeability of porous materials present in the structure. As long as there are only fluid and solid materials present in the structure, no parameters are needed here:



For porous materials, choose if the material is **Isotropic** or **Anisotropic**. In the isotropic case, the permeability is the same in all space directions. In the anisotropic case, enter a different permeability for each spacial direction.



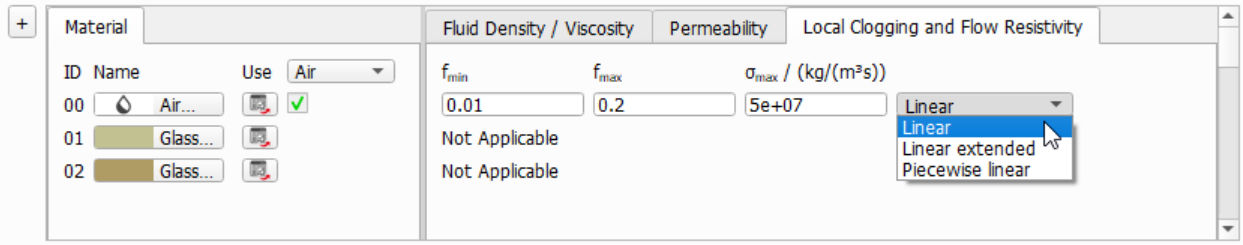
You can select and change a material by clicking on the material button.

Know how! The anisotropic permeability is always aligned with the coordinate system, not with the orientation of the filter media. Therefore, it is often the best choice to select **Isotropic** and use the through-plane permeability of the flat sheet material, as this is the permeability most relevant to the flow direction.

Local Clogging and Flow Resistivity

If the unresolved particle model is selected, the **Local Clogging and Flow Resistivity** tab will also be available.

Unresolved particles will create porous voxels when deposited into the 3D structure. In those voxels, flow is still possible, but experiences an increased flow resistivity. In the **Local Clogging and Flow Resistivity** tab, the relation between the dust volume fraction f inside of a voxel and the flow resistivity σ must be defined. This tab is only available for unresolved simulations, because in the **Resolved** case all voxels are treated as either empty or solid.



The flow resistivity σ depends on the values for $f_{\min}$, $f_{\max}$, and $\sigma_{\max}$, which must be defined for every material of the structure which can be filled by dust particles during the simulation, i.e., inside of the fluid (where the filter cake will form) and in all porous layers.

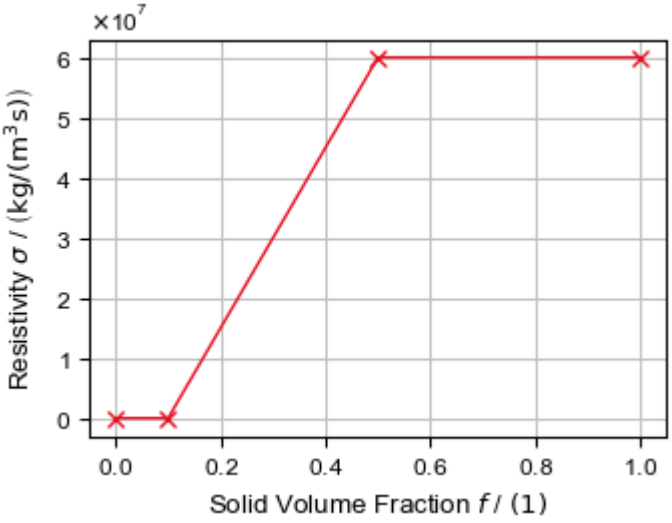
Note! It is necessary to set these parameters for the fluid material, although this might be unintuitive at first glance. Be aware that the parameters set for the fluid material describe the behavior of the filter cake because the filter cake consists of partially dust-filled voxels of the fluid domain.

Note! It is not possible to use different fluids in a structure, but it is possible to use different material IDs with the same fluid. This option is useful, when the filter cake should have different properties depending on its location in the structure.

Three different models are available to define the function $\sigma(f)$: **Linear**, **Linear extended**, and **Piecewise linear**.

Linear

In this model, the flow resistivity depends linearly in a certain interval on the local solidity.



The function $\sigma(f)$ is defined through:

$$\sigma = \begin{cases} 0 & \text{for } f < f_{\min} \\ \dfrac{f - f_{\min}}{f_{\max} - f_{\min}} \cdot \sigma_{\max} & \text{for } f_{\min} < f < f_{\max} \\ \sigma_{\max} & \text{for } f_{\max} < f < 1 \end{cases} \quad (48)$$

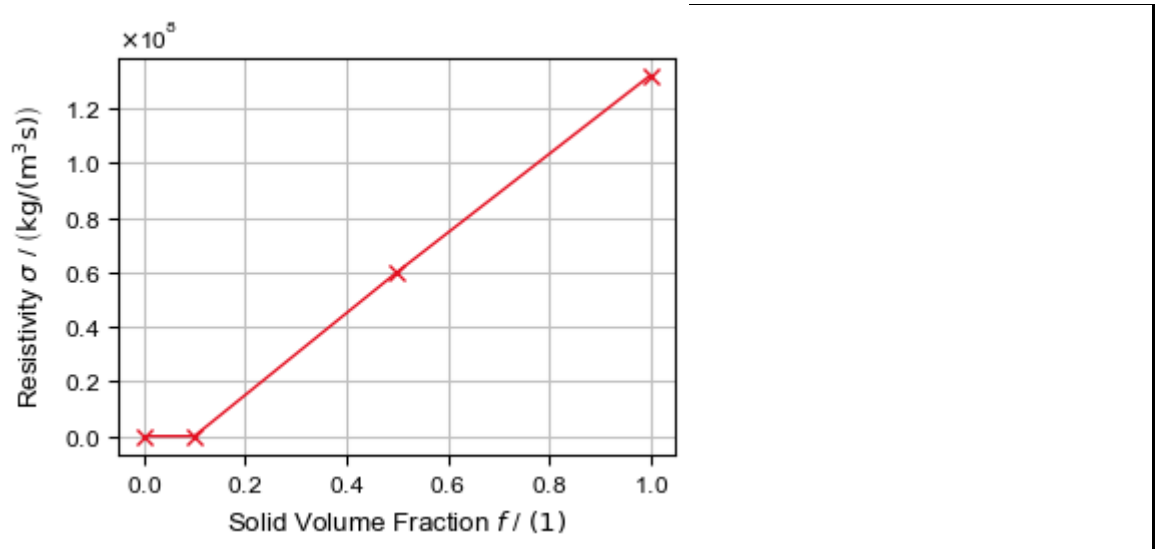
Know how! The general idea of this model is that $f_{\max}$ describes the maximal solidity (or Maximum Particle Packing Density) of the filter cake and $\sigma_{\max}$ the flow resistivity of the cake.

Under the assumption that all particle sizes are smaller than the voxel length (which holds true in many settings that consider a complete filter), using the solidity of the filter cake for $f_{\max}$ and the Maximal Particle Packing Density and using the flow resistivity as $\sigma_{\max}$ will lead to the expected results: The filter cake that emerges in the simulation will again have this solidity and flow resistivity.

If a significant number of particles are larger than the voxel length, the filter cake that emerges during the simulation will have a different solidity than $f_{\max}$ and a different flow resistivity as $\sigma_{\max}$. This has been described by Becker et al, 2016[257], and the paper also describes a way to estimate those parameters in this case.

Linear Extended

This model is basically the same as the **Linear** model, but the function is not cut off at the value of $\sigma_{\max}$, but extended linearly until $f = 1$.



The function $\sigma(f)$ is defined through:

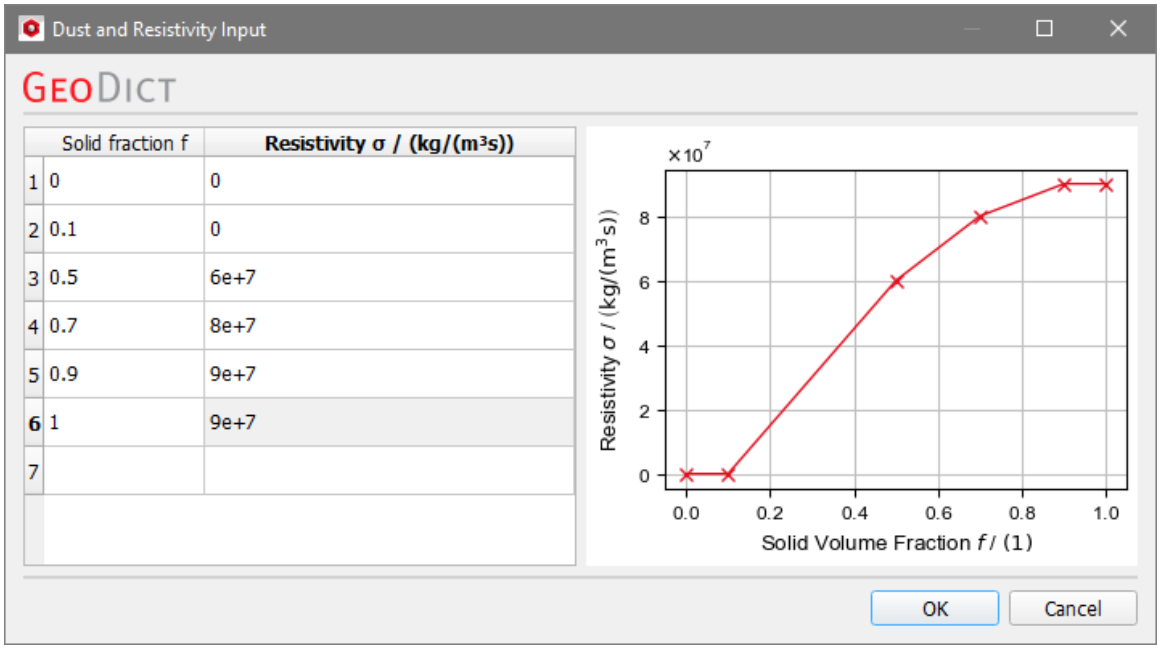
$$\sigma = \begin{cases} 0 & \text{for } f < f_{\min} \\ \dfrac{f - f_{\min}}{f_{\max} - f_{\min}} \cdot \sigma_{\max} & \text{for } f_{\min} \le f < 1 \end{cases} \quad (49)$$

Know how! The local solidity inside of a voxel can exceed the Maximum Particle Packing Density after the volume of the last particles deposited in this voxel is added. In fact, any value between 0.0 and 1.0 may appear during the course of a simulation. Therefore, it makes sense to extend the linear model beyond $\sigma_{\max}$.

Piecewise Linear

If **Piecewise linear** is selected, an arbitrary piecewise linear function can be defined by clicking the **Edit...** button.

The **Dust and Resistivity** Input dialog open and allows to define a piecewise linear function:



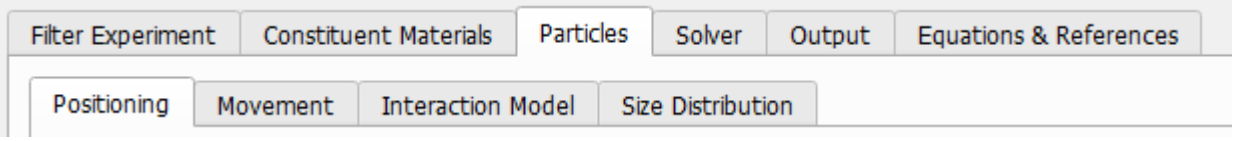
It is recommended to start the list with a solid fraction of 0, and end with a solid fraction of 1, to define the function for the whole interval [0,1).

Be aware that for a local solidity of 1.0, the voxel is always treated as solid, and no flow is possible. This is independent from the definition of the function $\sigma(f)$, which only applies to partially filled voxels (local solidity $f < 1$).

Know how! You can upscale the results of a fully resolved filter media simulation to determine these input parameters. In the Report tab of a **Filter Media - Filter Lifetime** command, the Filter Clogging Analysis 106 computes parameters that can be used as input here. Use the reported **Cake Filtration** values as $f_{\max}$ and $\sigma_{\max}$ of the fluid phase and the reported **Depth Filtration** values as $f_{\max}$ and $\sigma_{\max}$ of the porous filter material.

Particles

Under the **Particles** tab, four subtabs are available to define **Positioning**, **Movement**, **Interaction Model**, and **Size Distribution**.



Tabs of the Particles tab

- Positioning 137
  Define the start positions of the simulated particles.
- Movement 142
  Select the parameters modelling the motion of the particle.
- Interaction Model 143
  Define collision and adhesion models for particles-material collisons.
- Size Distribution 146
  Define the particle size distribution and all particle size dependent parameters.

Positioning

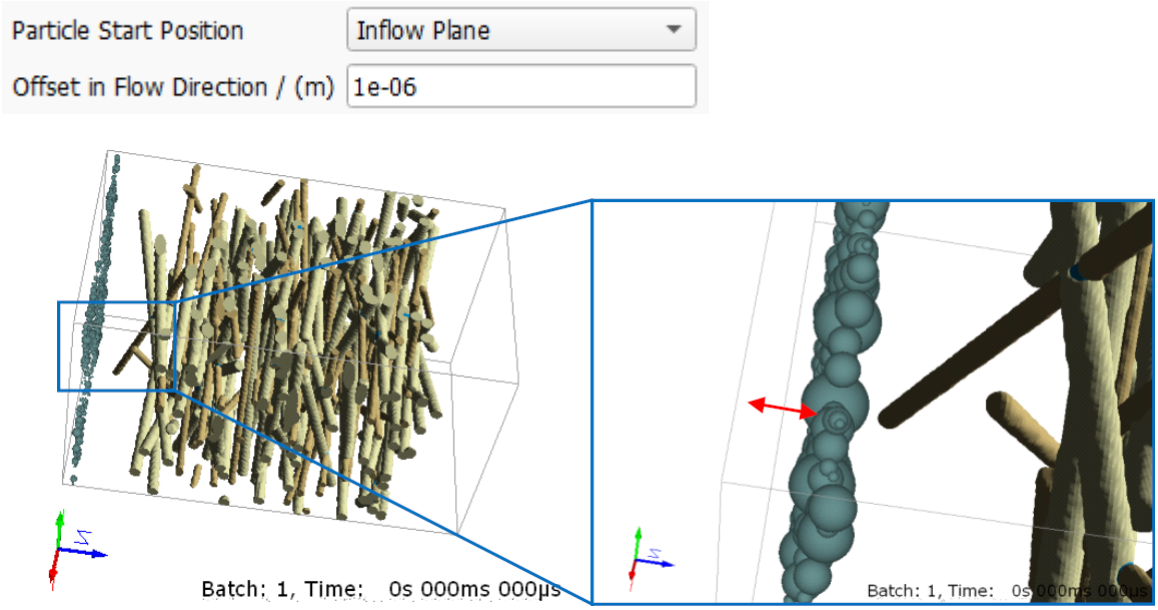
Particle Initial Position

Particle Start Position

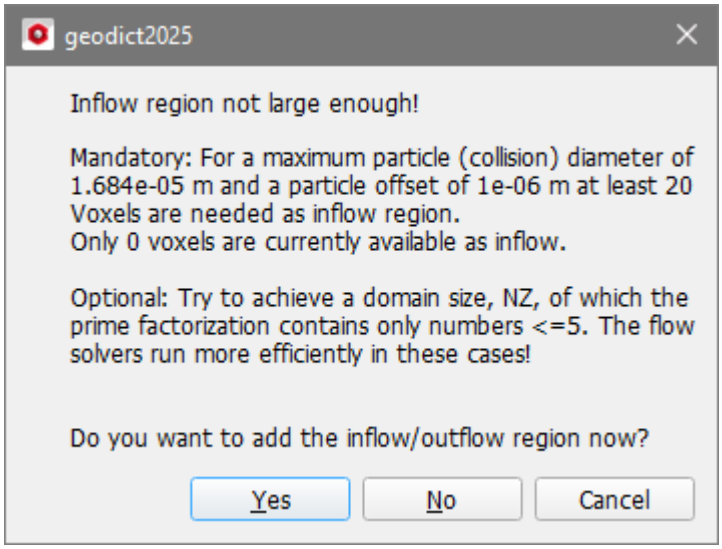
Select one of the following **Particle Start Position** types. The default start position is the Inflow Plane.

Inflow Plane

If **Inflow Plane** is chosen as **Particle Start Position**, the **Offset in Flow Direction** defines the initial distance of the particles to the boundary of the inflow region.

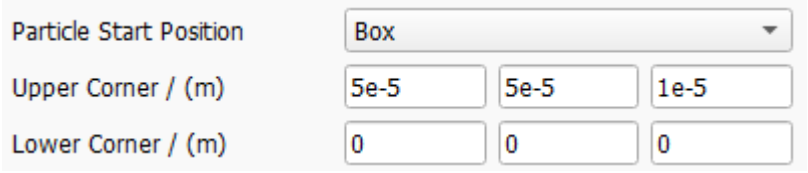


If the inflow region is too small and the particles cannot be placed in the inlet without intersecting the solid material, a warning appears after clicking **Run** to start the simulation in the **FilterDict** section.



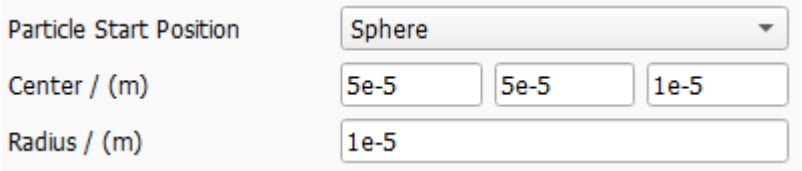
Box

If **Box** is chosen as **Particle Start Position**, two opposite corners of a rectangular box must be defined. All particle start positions will then be inside of this box.



Sphere

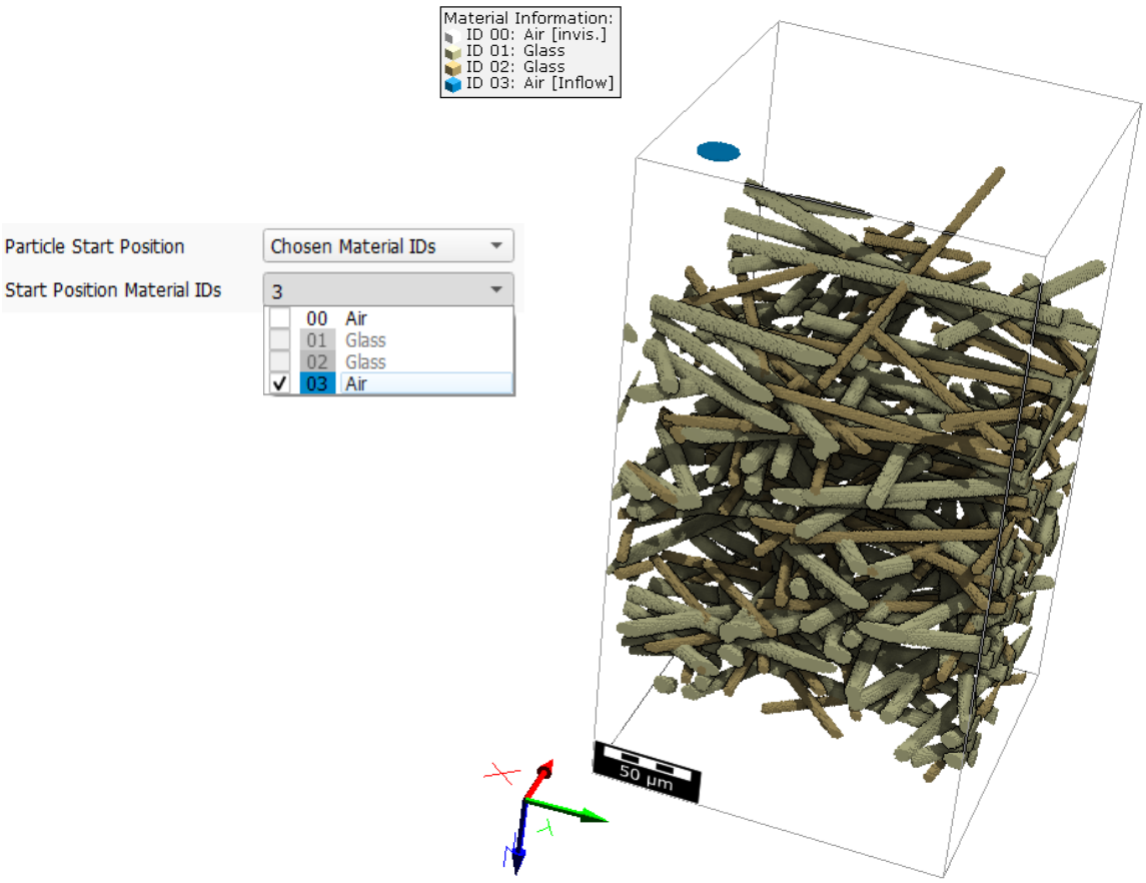
If **Sphere** is chosen as **Particle Start Position**, the center and radius of the sphere must be defined. All particle start positions will then be inside of this sphere.



Chosen Material IDs

An arbitrary location can be set by selecting **Chosen Material IDs** as **Particle Start Position**.

For example, by using GadGeo, various objects can be added into the inflow area. Those objects can be assigned to the same fluid material as the surrounding pore space, so they only serve as markers of a certain area (see picture below).



If this area ID is now chosen as **Start Position Material ID**, all particles start in this area. It is possible to mark several areas in this way and/or to choose several IDs as starting areas.

Everywhere

If **Everywhere** is chosen as **Particle Start Position**, particles may start everywhere in the pore space of the structure.

Particle Positioning Weights

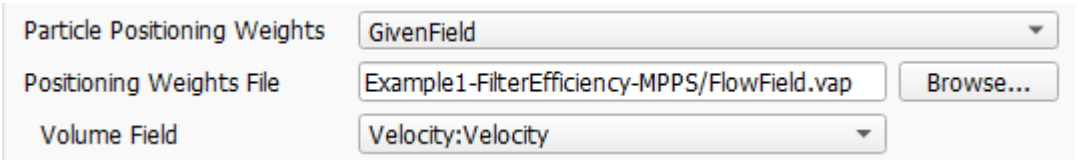
Particle start positions are scattered randomly inside the defined **Particle Start Position** area or volume.

By default, those positions are uniformly distributed. Be aware that a spatially **Uniform** placement in the inlet will lead to an increased particle concentration in areas with low fluid velocity, e.g., in proximity to a surface or a walls.

If **Velocity** is selected, the starting probability is based on the local fluid velocity. Selecting this option will lead to a uniform particle concentration in the inflow.

If **Distance** is selected, the starting probabiliy is based on the distance from the next solid object. This will lead to higher particle numbers in the core flow, while there will be less particles starting next to the surface.

With **GivenField**, it is possible to load an arbitrary 3D field of scalar values to define the starting probability. For example, it is possible to load the flow field of a previous computation and select the velocity or one of its components.



Particle Start Velocity

Select either **Fluid Velocity** (the default option, which was used until GeoDict 2025) or **Given Velocity** as the **Particle Start Velocity**.

With the first option, particles begin moving with the fluid velocity at their start position.

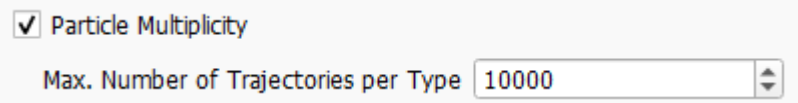
With the second option, all particles start with the same user-defined velocity.

Random Seed

Particles to be filtered are placed randomly in the area according to the selected **Particle Start Position**. The parameter **Random Seed** controls the underlying random number generator. The same random seed produces identical results, whereas results with different random seeds are similar but not identical.

Particle Multiplicity

When the Particle Resolution is set to **Mixed** in the Constituent Materials tab, the parameter **Particle Multiplicity** becomes available and a value can be entered in the **Max. Number of Trajectories per Type** box.



Unresolved particles may be much smaller than the used voxel length and thousands of dust particles are needed to fill a single voxel. Therefore, the number of **Particles per Batch** (on the **Solver tab**) may grow very large, leading to increased simulation times. With **Particle Multiplicity**, the number of particles that are tracked can be restricted.

When, for example, 1,000,000 particles of diameter 0.05 µm are to be simulated in the next batch, and you check **Particle Multiplicity** and set the **Max. Number of Trajectories per Type** to 10,000, only the movement of 10,000 particles is tracked. However, each of these particles represents 100 particles, and the multiplicity is 100 (100 x 10,000 = 1,000,000). If one of these particles is filtered, the mass and volume fractions of 100 particles are added to the deposited dust.

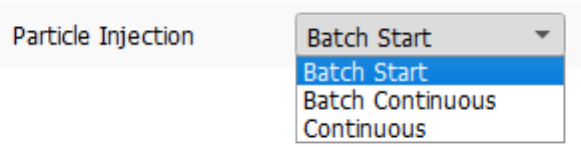
To make sure that the choice of **Particle Multiplicity** does not introduce large numerical errors, FilterDict ensures that the particle volume multiplied by its multiplicity is less than 5% of the single voxel volume.

Reflect particles at inflow plane

If diffusion through Brownian motion is simulated, some particles might diffuse against the flow direction, reach the inflow region and exit the domain. Check **Reflect particles at inflow plane** to avoid the exit of such particles.

Particle Injection

All particles simulated in a time inteval (a batch) move independently from each other. Select, if all particles should start moving at the beginning of the time interval (**Batch Start**) or if starting times should be distributed uniformly over the batch interval (**Continuous**).

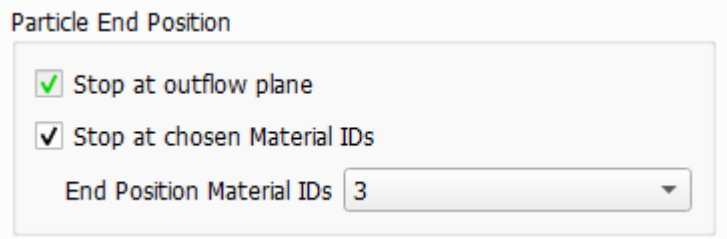


While **Continuous** offers a physically correct description of what happens in a real filter experiment, it may make a later visualization of the particle movement almost impossible. In a typical air filter media simulation, a batch interval may last minutes, while particles move with a velocity of several meters per second through a filter material of less than a millimeter thickness, which means that at most observation points in time, no moving particles can be seen. This can be overcome by letting all particles start at **Batch Start**, or distributed over a short time interval at the beginning of the batch (**Batch Continuous**).

The selection made here has very little influence of the computed results, because in any case all particles of the same batch do not interact with each other. They cannot collide in flight, and they cannot deposit on top of each other. They can only interact with the particles deposited in previous batches.

Particle End Position

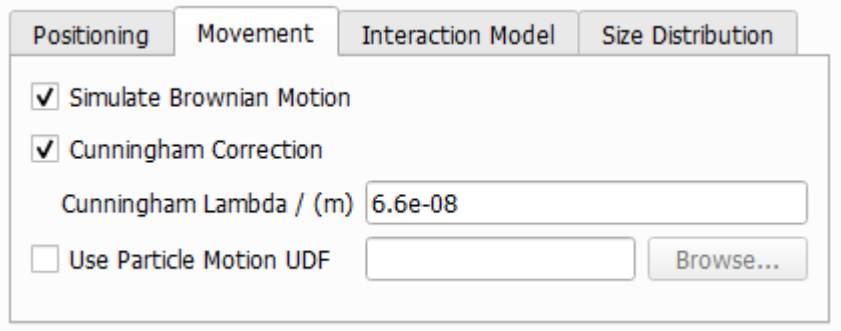
Particles reaching the outflow plane are always considered as unfiltered.



Similarly to the definition of the particle start position, an additional area can be marked. Particles that arrive in this area will also stop moving and be counted as unfiltered.

Movement

The parameters in the **Movement** panel define the simulation variables for the particle tracking.



Brownian Motion and Cunningham Correction

Checking **Simulate Brownian Motion** activates the simulation of random motion in particle movement. If this option is unchecked, the simulation is run with diffusivity $D = 0$.

If **Cunningham Correction** is checked, equation (12) is used to calculate C_c. Otherwise, $C_c = 0$ is used.

The **Cunningham Lambda** value is the mean free path λ used in equation (12). Make sure, that you use the Cunningham Correction with the default mean free path only in air filtration. Inside of water or oil, the mean free path is much smaller, and there is no need to take the Cunningham Correction into account.

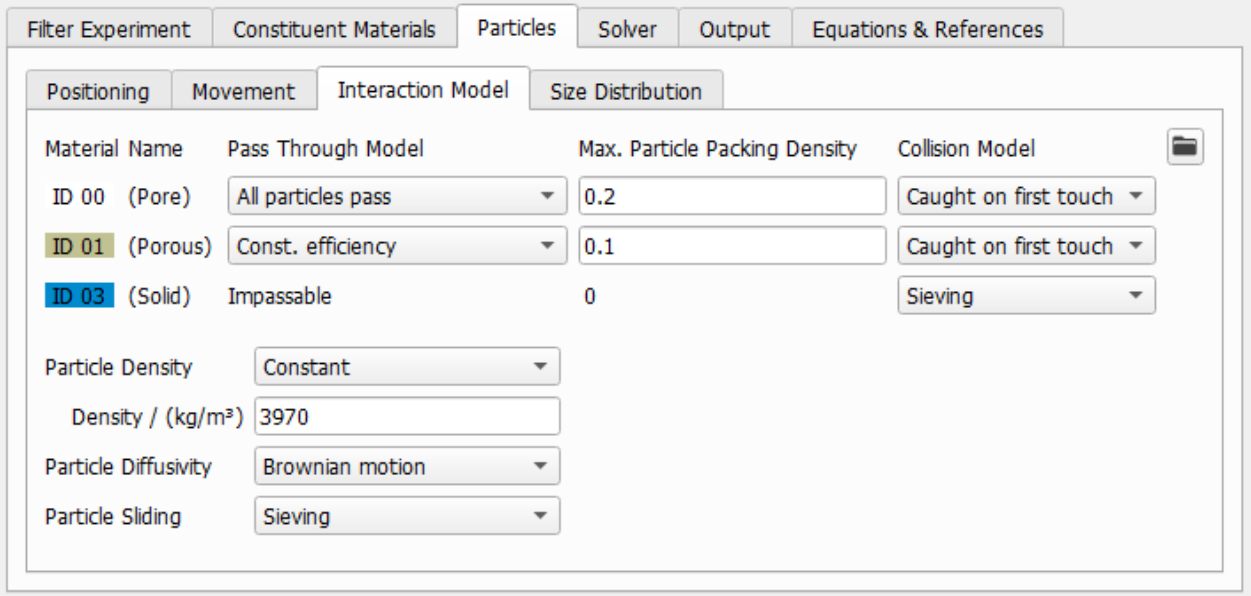
Use Particle Motion UDF

Check **Use Particle Motion UDF** to replace equations (11), (12), (13), and (14) with your own definitions. Refer to the Particle Motion UDF page for details on how to modify and compile user-defined functions.

Interaction Model

For filter element simulations, it is generally assumed that not all parts of the filter media are resolved, and that porous voxels are present in the model. Therefore, the **Pass Through Model** column is always available.

For all materials in the model, a **Pass Through Model**, the **Max. Particle Packing Density**, and a **Collision Model** must be set.



For each porous material, select one of the Pass Through Models **All Particles pass**, **Constant absorption rate**, **Clogging**, **Constant efficiency**, **Velocity-dependent efficiency**, or **Impassable**. If set to **Impassable**, the material is treated as a solid material and particles cannot enter.

Additionally, all compiled Pass Through UDFs[247] that are stored in the users UDF folder, appear here as additional choices in the pull-down menus. Additional UDF folders can be selected with the button.

If you select one of the models **All Particles Pass**, **Const. absorption rate**, **Clogging**, **Const. efficiency**, or **Velocity-dependent efficiency**, a voxel is filled until the **Max. Particle Packing Density** is reached. Once the voxel is filled, it automatically turns impassable for further particles and the approaching particles collide at the voxel surface. Select a **Collision Model** to determine how these collisions are handled. If **Velocity-dependent efficiency** is selected, it must be defined for each particle type in the Size Distribution[146] table

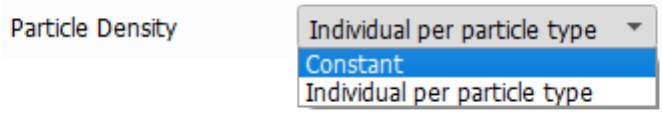
In this sense, the user interface can be understood as follows: The **Pass Through Model** on the left is valid until the **Max. Particle Packing Density** is reached, and afterwards the **Collision Model** on the right applies. If the material is solid, the material is **Impassable**, thus the **Max. Particle Packing Density** is fixed to 0, and the **Collision Model** applies directly.

Know how! You can upscale the results of a fully resolved filter media simulation to determine the **Max. Particle Packing Density**. In the Report tab of a **Filter Media - Filter Lifetime** command, the Filter Clogging Analysis[106] computes parameters that can be used as input here. Use the reported **Cake Filtration** value of $f_{\max}$ as **Max. Particle Packing Density** of the fluid phase and the reported **Depth Filtration** value of $f_{\max}$ as **Max. Particle Packing Density** of the porous filter material.

Know how! The **Collision Model** of the fluid phase describes the behavior of particles when they arrive at the surface of a voxel filled with dust particles. In most setups, it is safe to assume that the arriving particle is deposited somewhere in the voxel on top of the filled voxel. In that case, **Caught on First Touch** is a good choice. Only in cases, where particles may bounce off from the filter cake surface, and get transported to a significantly different position, it is necessary to change this model to the **Hamaker** model. This might be the case for large particles, with high velocities, or with specific angle of attacks of the flow.

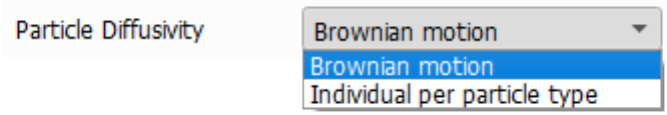
Particle Density

If **Individual per particle type** densities are chosen, a new column called **Particle Density** is added to the table under the Size Distribution tab, and an individual density can be entered for each particle type or size. Otherwise, the **Particle Density** entered here is used for all particle sizes and types.



Particle Diffusivity

If **Individual per particle type** is chosen, a new column called **Difusivity in Pore** is added to the table under the Size Distribution tab, and an individual diffusivity can be entered for each particle type or size.



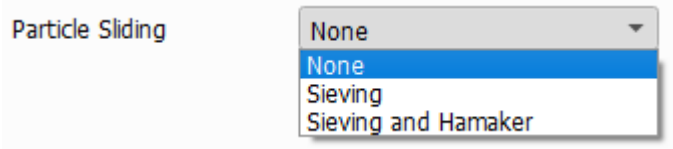
For **Brownian Motion**, the particle diffusivity is computed for each particle type according to equation (13).

If **Simulate Brownian Motion** is switched off on the Movement tab, only **None** is selectable here.

If **Use Particle Motion UDF** is selected on the Movement tab, the choices made here have no effect, as the formula implemented in the UDF will be used.

Particle Sliding

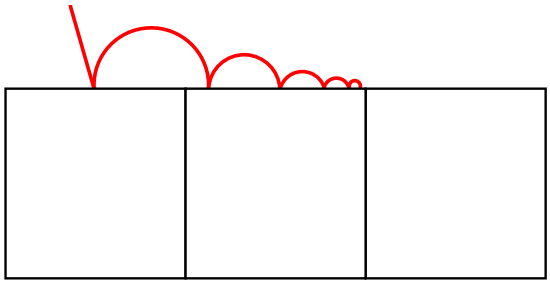
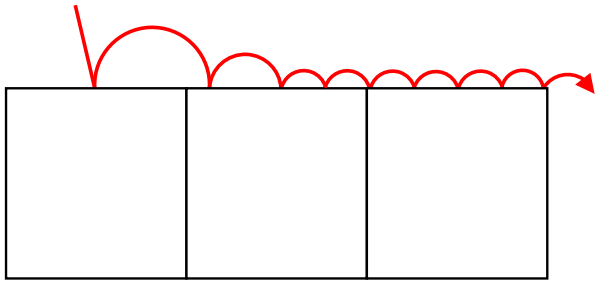
Particles that collide with the surface loose some of their energy if the restitution factor is smaller than 1. In certain pore geometries, the flow may move a particle very close to a pore surface. This causes many consecutive hits, and repeatedly applying the restitution factor will stop the particle's motion. This behavior is often undesired, and it may overestimate the filter efficiency when combining a **Sieving** collision model with a relatively low restitution value.



When **Sieving** is selected in the drop-down menu, all **Sieving** particle-wall collisions are modified, when **Sieving and Hamaker** is selected, all **Sieving** particle-wall collisions are modified and all **Hamaker** particle-wall collisions are modified. **Caught on First Touch** particle-wall collisions are not modified.

Enabling **Particle Sliding** for the **Sieving** or **Hamaker** collision models enlarges the tangential restitution factor when a particle hits a surface in a location that lies close to the last surface collision.

The first bounce on the wall is always treated the same, independent whether particle sliding is active or not: A particle arrives with a certain velocity, loses energy on impact, and bounces back again. The strength of the bounce-back is controlled with the given **Restitution**. When particle sliding is active, further impacts near the first hit are treated differently: Only the momentum perpendicular to the wall is reduced. The velocity along the wall is not reduced further, so in this direction the restitution is equal to 1. This means that particle movement along the wall is no longer slowed down. The normal restitution remains unchanged. In effect, the particle will slide along the surface.

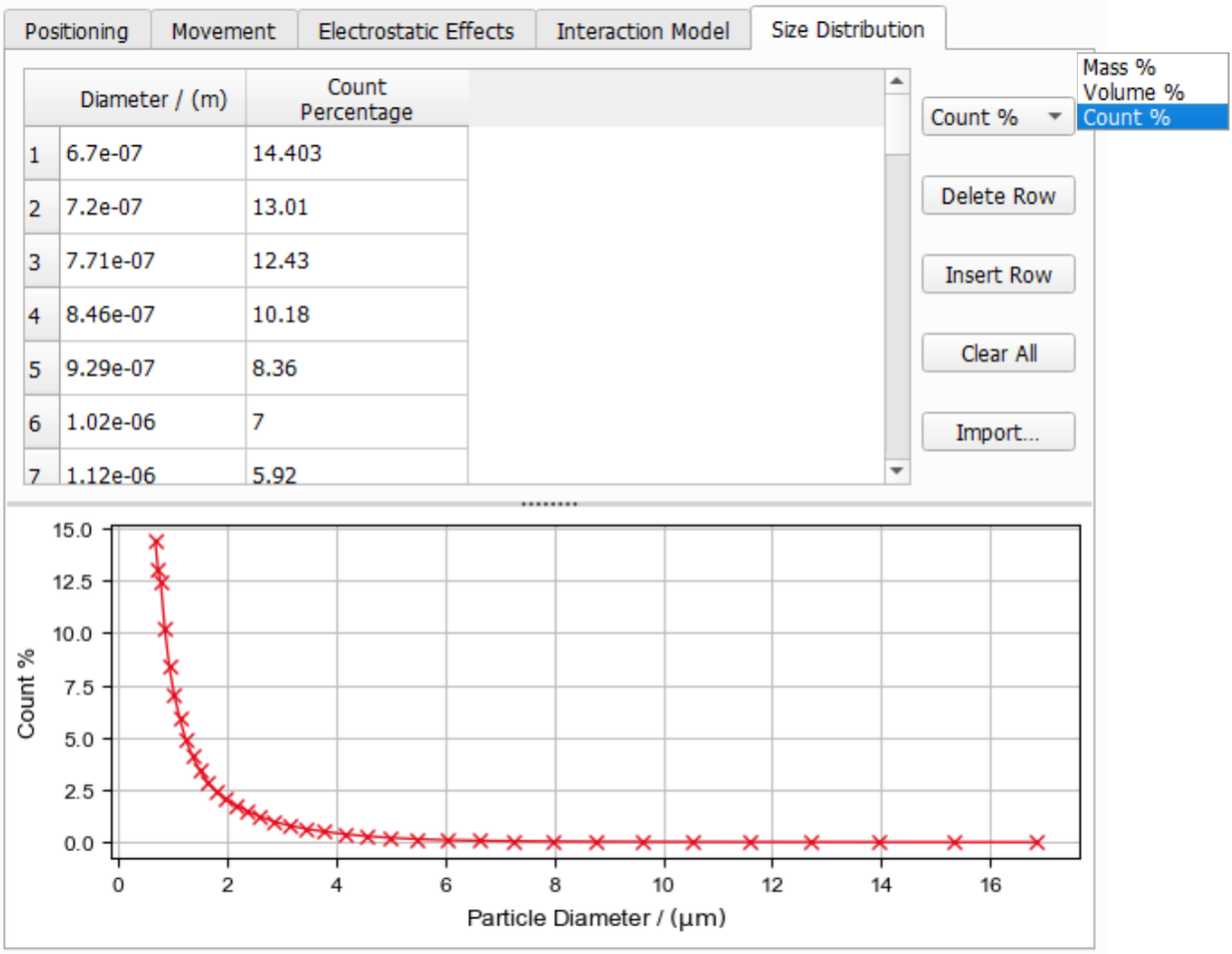
Without particle sliding:	With particle sliding
• energy lost at every collision • particle caught on surface	• energy conserved after some hits • particle moves along surface
	

In setups where particles are sieved by filter materials having pores of well-defined sizes (e.g., in meshes, nets or weaves), selecting particle sliding has a great influence on the computed filter efficiency and pressure drop and it is recommended to select this option.

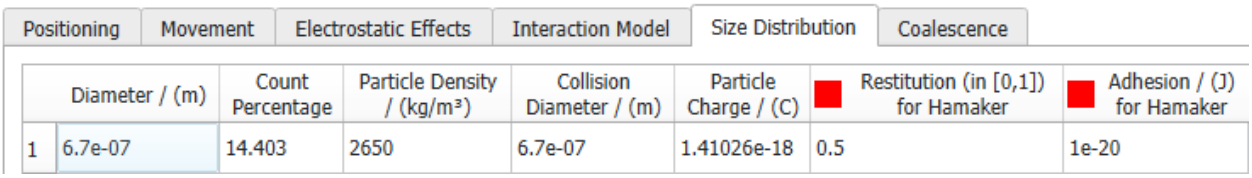
In setups where particles are mainly caught by adhesion or where the filter material consists of irregular pores of many different sizes (e.g., nonwovens), the choice of the particle sliding model has little to no effect on the simulation results.

Size Distribution

The **Size Distribution** tab contains a table with the particle size distribution. The table consists of at least two columns. The first column lists the **Diameter** of the particles. The second column contains the particle distribution. You can select if the percentages describe **Mass %**, **Volume %**, or **Count %** through the upper right pull-down menu.



Additional columns appear depending on the choices made under the Interaction Model subtab:



These columns are:

- **Restitution** for materials with Sieving collision model.
- **Restitution** and **Adhesion** for Hamaker material.
- **Particle Density** column for Individual per particle type densities.
- **Particle Charge** column for Individual per particle type charges.
- **Collision Diameter** column for Individual per particle type collision diameters.

For user-defined collision models, the number of material parameters to be entered in the table is specified in the UDF.

By clicking on a column heading, all values in the selected column can be set to the value in the first row. The entries in the table do not need to be sorted.

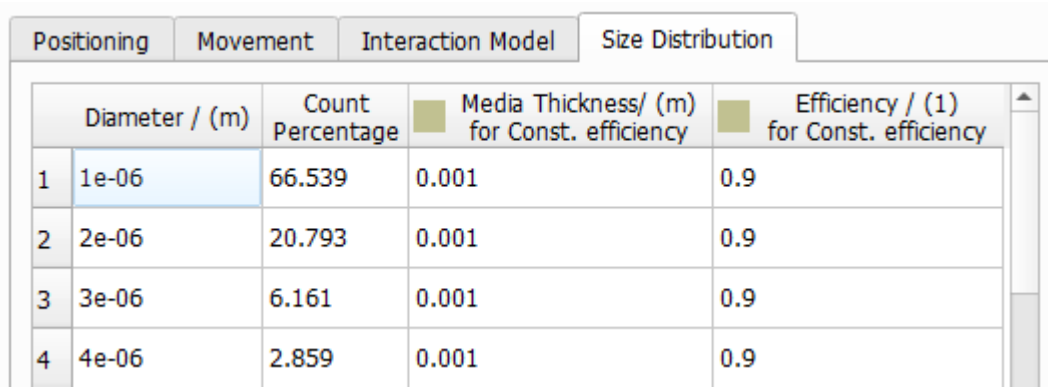
Click **Delete Row** to delete the currently selected row. Click **Insert Row** to insert a row below the currently selected row. Click **Clear All** to remove all rows.

Click **Import...** to import data from an ASCII text file. Some example particle distributions are provided with GeoDict and can be found in the installation folder (Program Files/Math2Market GmbH/GeoDict2026/FilterDict).

More convenient than importing is often to simply copy & paste entries from an Excel table into GeoDict, which is supported by the size distribution table. To copy a table from Excel, first clear the size distribution table by clicking **Clear All**. Then, select the cells in Excel and press Ctrl+C. Then, click into the upper left cell in the size distribution table and press Ctrl+V. This works also if more than two columns are present.

If the entered percentages do not add up to 100%, a warning message pops up when closing the dialog. You can select to normalize the values automatically or to return to the dialog and adjust the percentages manually.

If **Constant efficiency** or **Clogging** is selected as pass through model, a fractional filtration efficiency value and the corresponding media thickness must be entered for each particle size. For example, with the definitions in the table below, 90% of the particles are filtered when passing through 1 mm of the media.

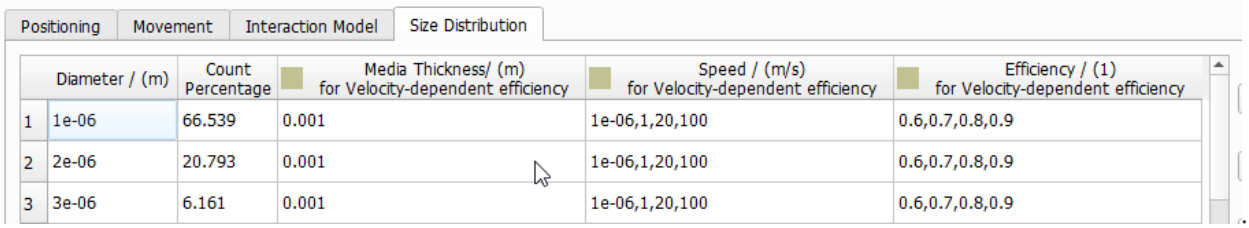


	Diameter / (m)	Count Percentage	Media Thickness/ (m) for Const. efficiency	Efficiency / (1) for Const. efficiency
1	1e-06	66.539	0.001	0.9
2	2e-06	20.793	0.001	0.9
3	3e-06	6.161	0.001	0.9
4	4e-06	2.859	0.001	0.9

Know how! When using the **Clogging** model, enter the filtration efficiency of the clean filter media as efficiency value. Whereas, when using the **Constant Efficiency** model, it is recommended to use an efficiency value that represents an average efficiency value over the depth filtration phase.

Enter the thickness of the flat sheet material or height of the filter geometry as **Media Thickness**, if the entered efficiency value corresponds to the complete thickness of the filter.

If **Velocity-dependent efficiency** is selected as pass through model, efficiency values depend on the velocity as shown in the following screenshot. For example, with the definitions in the table below, at a particle velocity of 1 m/s, 70% of the particles are filtered when passing through 1 mm of the media.



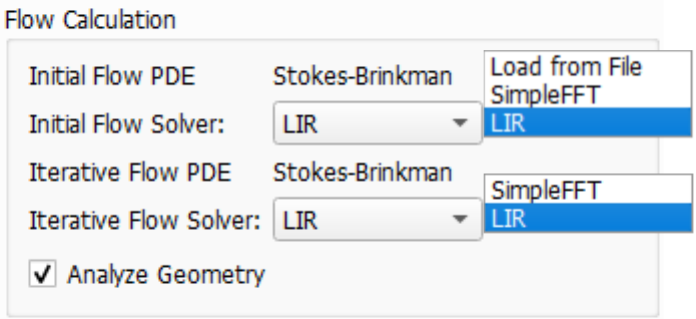
Solver

The **Solver** tab contains on the left the **Flow Calculation**, **Batch Settings**, and the **Parallelization** setup and on the right individual tabs for every solver. Tabs corresponding to unused solvers are grayed out and not selectable.

Flow Calculation

FilterDict treats the flow for the first (initial) batch and all subsequent (iterative) batches differently for two main reasons:

- In the initial batch, the filter is still clean, while in all iterative batches, previously deposited particles are present.
- In all iterative steps, the flow solver can use the solution of the previous step as initial guess, which is not possible for the initial batch.



The names of the equations to solve the **Initial Flow PDE** and **Iterative Flow PDE** are shown for information. These entries cannot be changed directly because they depend on the selection of **Resolved** / **Mixed** particles in the Constituent Materials tab 132 and the choice of **Creeping** / **Fast Flow** in **Flow Motion** under the Filter Experiment tab 127.

To solve the Flow PDEs, different flow solvers are available as **Initial Flow Solver** and as **Iterative Flow Solver** (SimpleFFT or LIR). The initial flow field may alternatively also be loaded from a previous FilterDict or FlowDict computation.

Tooltips help choosing the appropriate Initial and Iterative Flow Solvers. The default choice is the LIR solver for both the initial and the iterative flow solver. In most cases, LIR is computationally more efficient than SimpleFFT.

Analyze Geometry

If this option is chosen, a geometrical analysis at first determines whether a through path exists and removes unconnected pore components from the computational grid. This may speed up the flow computations but requires time for the geometrical analysis.

Batch Settings

The number of **Batches per Flow Field** determines how often the flow field is re-computed. Usually, the flow field should be recomputed for every batch (time interval), but if this is numerically too costly, it can be changed to re-compute the flow field only after every second or third batch. However, when more than one batch per flow field is chosen, the accuracy of the result decreases. In this case, numerical artifacts might be introduced. The most accurate results are obtained by computing one batch per flow field.

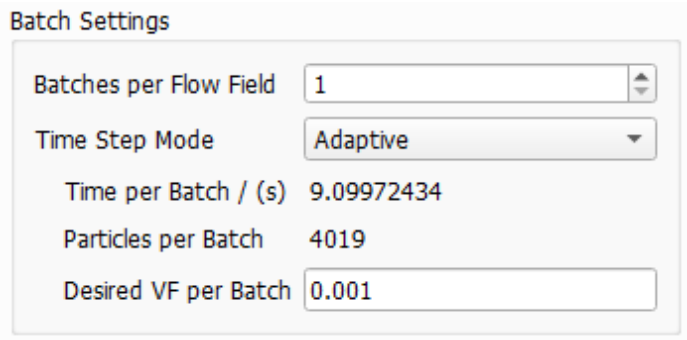
A batch of particles corresponds to a certain time interval in the experiment. The particles simulated in a batch do not interact with each other, but they do interact with the particles deposited in the previous batches. Decreasing the number of particles per batch leads to a higher accuracy, but also to longer simulation times.

The length of the time intervals per batch and the number of particles can be chosen by selecting a **Time Step Mode** from the pull-down menu:

Adaptive

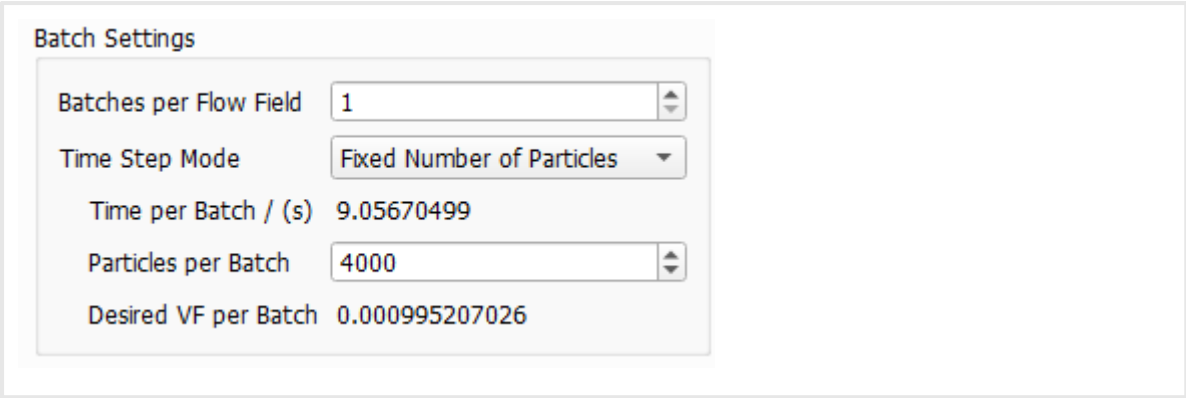
The number of particles is determined adaptively in each step. It depends on the desired volume fraction of deposited particles (**Desired VF per Batch**).

Additionally, FilterDict considers information from the previous steps like, e.g., the particle deposition zone and filter efficiency. Small **Desired VF per Batch** values correspond to shorter time steps and less particles per batch. Larger values lead to longer time steps and more particles per batch.



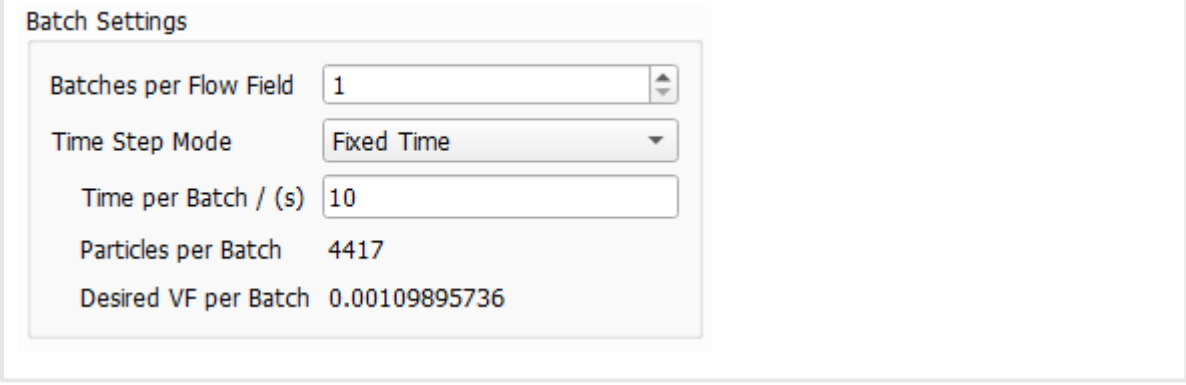
Fixed Number of Particles

Set the number of **Particles per Batch**. The time per batch is computed based on the number of particles.



Fixed Time

The number of particles per batch is determined by the **Time per Batch / (s)**. With a **Max. time reached** of 100 s chosen as simulation stopping criterion and 10 s chosen as **Time per Batch** here, the simulation stops after 10 batches of particles.



For multi pass simulations, **Fixed Number** or **Adaptive** should be preferred over **Fixed Time**. The particle concentration in the fluid changes over time. Therefore, also the number of particles could change significantly with **Fixed Time** steps and this might lead to a less stable simulation.

Parallelization

Depending on the purchased license, the simulation process can be parallelized. The **Parallelization Options** dialog opens when clicking the **Edit...** button, to choose between **Sequential**, **Parallel (Shared Memory)**, **Automatic Maximum of Threads** and **Cluster**. For details on how to set up and run parallel computations, refer to the High Performance Computations user guide

The chosen parallelization settings apply for all steps of the simulation. Per default, the **Automatic Maximum of Threads** option is used, and the exact number of threads depends on the number of cores on the current computer and the number of licensed processes.

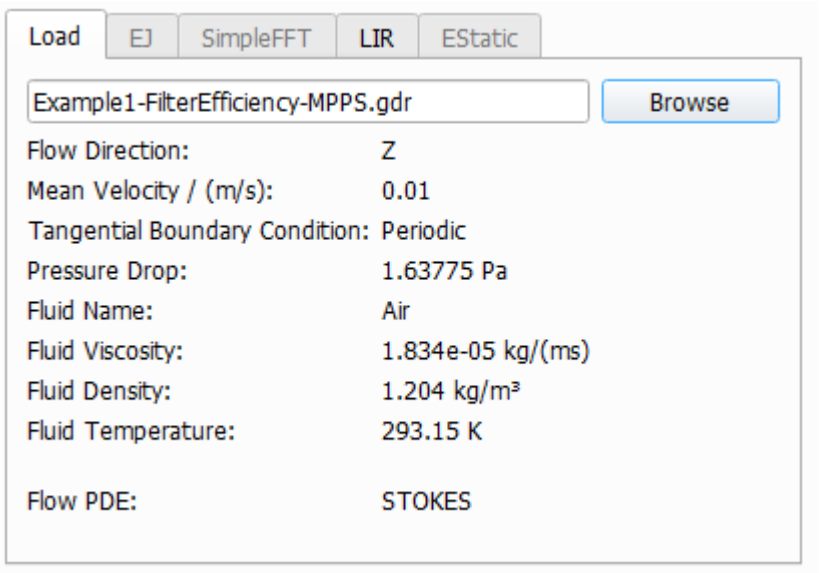
Load

Select **Load From File** in the **Flow Calculation** panel as **Initial Flow Solver** to use a

precomputed flow field. The flow field may originate from a FlowDict simulation or from a FilterDict simulation. When calculated with FlowDict, you have to make sure to:

1. Add inflow and outflow regions to the media model before running FlowDict.
2. Compute the flow in the Z-direction. You may either use the Stokes or Navier-Stokes equations depending on the flow velocity for this. Make sure that you use the same boundary conditions in FlowDict as you would use in FilterDict.
3. Set the accuracy at least one order of magnitude higher than the default in FlowDict (e.g., Error Bound = 0.001 instead of 0.01). The stopping criteria depend on global values, whereas the particle movement depends on the local flow field which is subjected to larger deviations.

Under the **Load** tab, click **Browse** and choose a result file (GDR) to open. If no flow field is selected, a warning message appears (No flow field chosen) when trying to run the simulation.



The physical properties of the fluid used in the flow simulation are entered automatically when loading the flow GDR file and appear listed under the **Load** tab.

- The **Flow direction: Z** is the main direction of the flow.
- The **Mean Velocity** (m/s) of the flow field, the used **Tangential Boundary Conditions**, and the **Pressure Drop** are taken from the loaded flow simulation. If the flow result was obtained by solving the Stokes equation, the solution is linear and will be rescaled automatically to match the Mean Velocity entered in the **Filter Experiment** tab.
- The fluid parameters **Fluid Viscosity** (kg/m s), **Fluid Density** (kg/m^3), and **Fluid Temperature** (K) are the physical values of the fluid used by the solver for the calculation of the flow field.

The fluid settings chosen under the Constituent Materials must be the same as the fluid that was used in the previously run flow simulation, loaded through the GDR file.

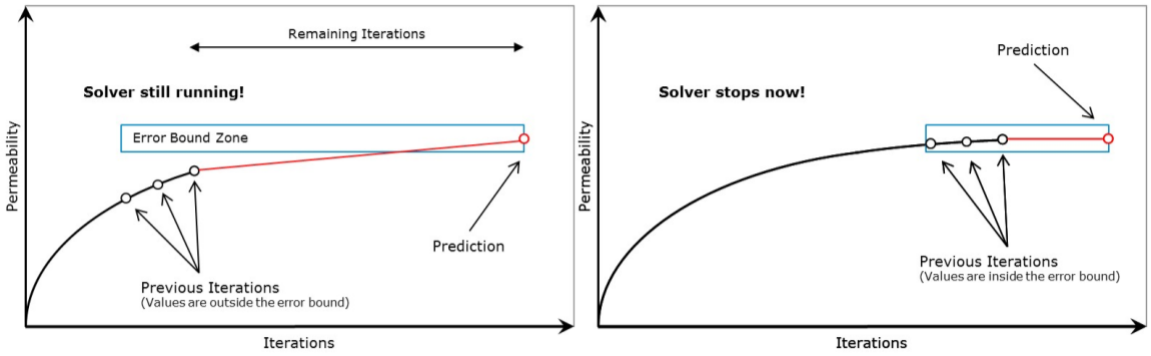
Also, the **Tangential Boundary Conditions** selected in the **Filter Experiment** tab must match with those used in the flow simulation.

SimpleFFT

The **SimpleFFT** tab becomes editable when you select the SimpleFFT solver as **Initial Flow Solver** or **Iterative Flow Solver** in the Flow Calculation 149 panel.

Simulation Stopping Criterion: Error Bound, Tolerance, and Residual

The stopping criterion **Error Bound** uses the result of previous iterations, and predicts the final solution based on linear and quadratic extrapolation. The solver stops if the relative difference between computed and predicted solution is smaller than the specified error bound. The stopping criterion recognizes oscillations in the convergence behavior and prevents premature stopping at local minima or maxima. A damped convergence curve is fit through the oscillating curve and the solver stops then regarding the damped convergence curve.



If the Krylov method (under LIR - Advanced Options) is activated, the definition of **Error Bound** is somewhat different. Here, no prediction is made, instead the continuity between neighboring cells and the conservation of mass is checked. The maximal value is normalized by the mean flow and if this value is smaller than the **Error Bound** the simulation stops.

The **Tolerance** stopping criterion looks for stagnation of the method when the process becomes stationary, i.e., the improvement in the pressure drop value becomes extremely small from iteration to iteration.

In each iteration, the solver checks for the current computed value x_c against the values x_{c-i} of the last 100 iterations if there are any changes.

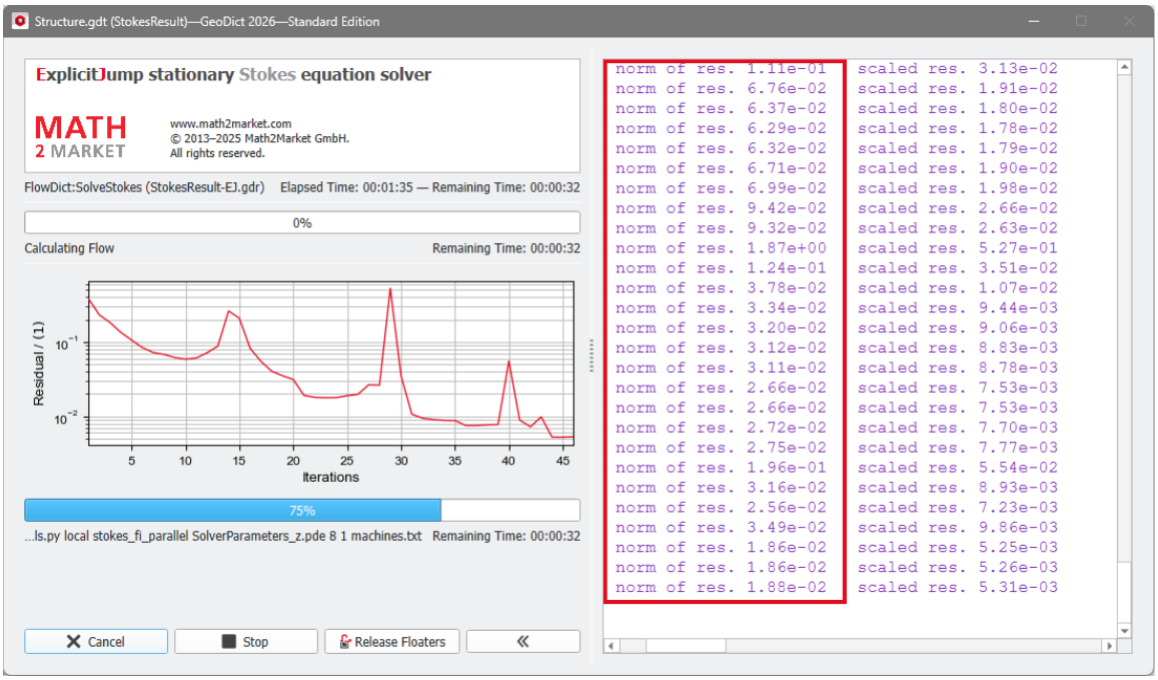
$$\max_{i\in[1,100]} \frac{x_c - x_{c-i}}{x_{c-i}} < \text{tolerance} \tag{50}$$

The computation is stopped if the maximal relative change is smaller than the value entered for **Tolerance**. When there is doubt about the quality of the solution, decrease the **Tolerance** value by a factor of ten. The drawback of this criterion is that the solver sometimes might stop too early in case of slow convergence rate or at local extrema of

oscillatory convergence curves.

When the **Residual** stopping criterion is used, the iteration is stopped if the solution satisfies the equation up to the required accuracy.

By setting the stopping criterion to **Residual**, the computations terminate as soon as the relative norm drops below the selected residual threshold. The relative norm of the Schur Complement residual is computed and displayed in the console window during the calculations. The console is visible by clicking the double arrow button on the lower right corner in the progress dialog.



The recommendation to choose **Tolerance** or **Residual** for the **EJ** solver is based on the structure's porosity. Both give similar results for highly porous structures. For dense structures, if using the Schur Complement **Residual**, the relative norm of the residual may be small even though the correct pressure drop has not been reached. So, when in doubt, use the **Tolerance** criteria – the default option.

Be aware that the default accuracy in FilterDict is one order of magnitude higher than in FlowDict. The stopping criteria depend on global values, whereas the particle movement depends on the local flow field. It is therefore necessary to make sure that not only the overall pressure drop is computed up to the given accuracy, but that also the local values are sufficiently accurate. This typically achieved by setting a higher accuracy as stopping criterion.

Simulation Stopping Criterion: Max. Iterations and Max. Run Time

Use the **Maximum Iterations** value or the **Maximum Run Time (h)** stopping criteria causes the solver to stop if the maximal number of iterations and/or the maximal runtime (in hours) is exceeded.

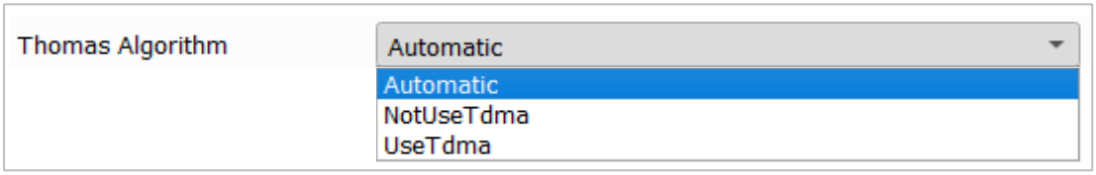
When the solver stops because one of these criteria has been reached, no guarantee on the quality of solution can be given. In this case, a warning is printed into the report. The following possibilities might help:

- Check the corresponding .log file to see how large the residual values and the pressure drop values are. If the pressure drop values are in reasonable range and they do not have changed during the last iterations, you may decide to use the current result.
- Double-check the structure and parameter values. Unphysical parameters or too rough resolution of the structure (leading, e.g., to artificial unconnected components) can cause an iterative solver to fail.

Thomas Algorithm

The tridiagonal matrix algorithm (Tdma), also known as the **Thomas Algorithm**, may help to improve the convergence for high porosity structure, especially for Navier-Stokes. The algorithm transforms the 3D flow problem to 2D. Thus, it leads to a shorter runtime and increases the convergence speed.

Three options can be chosen from the pull-down menu: **Automatic**, **NotUse Tdma** or **Use Tdma**.



If **Automatic** is selected, the SimpleFFT solver decides if **Tdma** will be used, regarding the porosity of the structure. Thus, **Tdma** is used if the porosity is high. For most cases it is recommended to choose **Automatic**.

If a message dialog appears, displaying that NaN is detected in iterations, **NotUse Tdma** should be tried.

If the structures porosity is very high, it might decrease the runtime a little bit to explicitly choose **Use Tdma**. But for most cases **Automatic** will work well.

For more information about the Thomas algorithm see here.

Velocity Relaxation and Pressure Relaxation

Depending on the material parameters and geometrical structure of the structure, the underlying mathematical problem can vary in complexity, thus influencing the behavior of the solver. This is directly related to the Reynolds number, an indication of the complexity of the flow solver computations. The higher the Reynolds number, the more **Stable** the flow solver settings should be, resulting in higher number of iterations, slower time stepping, and longer flow solver run times. However, making the solver run

less iterations and, thus, faster (**Fast**), implies the risk that the solver does not converge.

For the SimpleFFT solver, the management of this balance is done through the **Velocity relaxation: Stable ↔ Fast** and **Pressure relaxation: Stable ↔ Fast** slide bars.

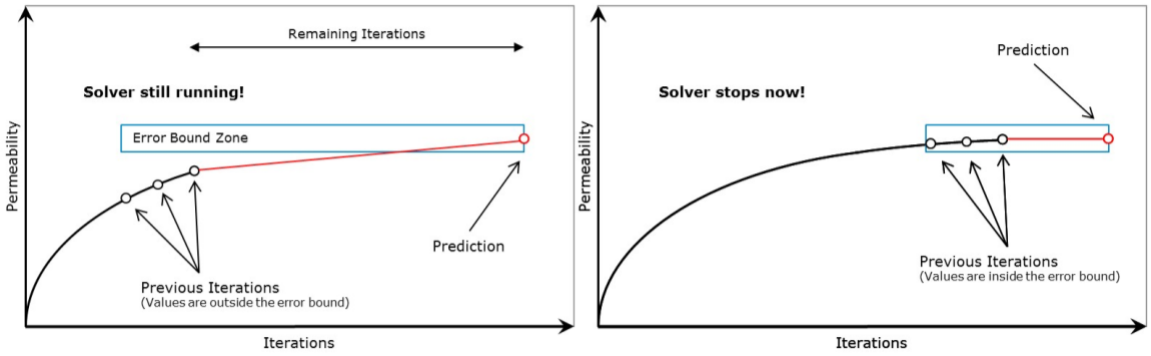
Setting the balance of **Stable** versus **Fast** is a trial-and-error process. Although there is no general rule to optimize it, the log files and the visualization of the structure might help finding the balance. Non-sense structure visualization results indicate that a more stable, slower solver is needed.

LIR

The LIR tab becomes editable when you select the LIR solver as **Initial Flow Solver** or **Iterative Flow Solver** in the Flow Calculation 149 panel.

Simulation Stopping Criterion: Error Bound or Tolerance

The stopping criterion **Error Bound** uses the result of previous iterations, and predicts the final solution based on linear and quadratic extrapolation. The solver stops if the relative difference between computed and predicted solution is smaller than the specified error bound. The stopping criterion recognizes oscillations in the convergence behavior and prevents premature stopping at local minima or maxima. A damped convergence curve is fit through the oscillating curve and the solver stops then regarding the damped convergence curve.



If the Krylov method (under LIR - Advanced Options) is activated, the definition of **Error Bound** is somewhat different. Here, no prediction is made, instead the continuity between neighboring cells and the conservation of mass is checked. The maximal value is normalized by the mean flow and if this value is smaller than the **Error Bound** the simulation stops.

The **Tolerance** stopping criterion looks for stagnation of the method when the process becomes stationary, i.e., the improvement in the pressure drop value becomes extremely small from iteration to iteration.

In each iteration, the solver checks for the current computed value x_c against the values x_{c-i} of the last 100 iterations if there are any changes.

$$\max_{i\in[1,100]} \frac{x_c - x_{c-i}}{x_{c-i}} < \text{tolerance} \tag{50}$$

The computation is stopped if the maximal relative change is smaller than the value entered for **Tolerance**. When there is doubt about the quality of the solution, decrease the **Tolerance** value by a factor of ten. The drawback of this criterion is that the solver sometimes might stop too early in case of slow convergence rate or at local extrema of oscillatory convergence curves.

Be aware that the default accuracy in FilterDict is one order of magnitude higher than in FlowDict. The stopping criteria depend on global values, whereas the particle movement depends on the local flow field. It is therefore necessary to make sure that not only the overall pressure drop is computed up to the given accuracy, but that also the local values are sufficiently accurate. This typically achieved by setting a higher accuracy as stopping criterion.

Simulation Stopping Criterion: Max. Iterations and Max. Run Time

Use the **Maximum Iterations** value or the **Maximum Run Time (h)** stopping criteria causes the solver to stop if the maximal number of iterations and/or the maximal runtime (in hours) is exceeded.

When the solver stops because one of these criteria has been reached, no guarantee on the quality of solution can be given. In this case, a warning is printed into the report. The following possibilities might help:

- Check the corresponding .log file to see how large the residual values and the pressure drop values are. If the pressure drop values are in reasonable range and they do not have changed during the last iterations, you may decide to use the current result.
- Double-check the structure and parameter values. Unphysical parameters or too rough resolution of the structure (leading, e.g., to artificial unconnected components) can cause an iterative solver to fail.

Advanced Options: Write Compressed Volume Field

The LIR solver uses a very memory efficient adaptive grid structure for the simulations.

If the option **Write Compressed Volume Fields** is checked, then the adaptive grid is

used as compression method for writing out *.vap files. This option allows to save 80-90% space on hard drive. The runtime for writing *.vap files is also reduced significantly. But the runtime for loading and uncompressing of compressed *.vap is increased by the amount of runtime that was saved for writing out compressed *.vap files.

If the option **Write Compressed Volume Fields** is not checked, then a usual regular grid is used for writing out *.vap files.

Advanced Options: Optimization Options

The **Multigrid Method** (see, e.g., Wesseling, 2004) was introduced to speed-up the computation and reduce the runtime significantly. The main idea of **Multigrid** is the usage of multiple coarser adaptive grids to speed up convergence behavior but requires only little more memory.

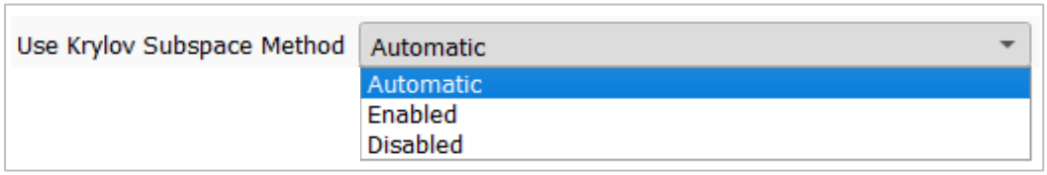
The method is available to solve the Stokes and Stokes–Brinkman equations as well as for solving mechanics, diffusion, thermal, and electrical conduction and is enabled by default.

Another speed-up option to accelerate the convergence behavior of the LIR solver is called **Krylov Subspace Method**.

The runtime of the LIR solver depends on many different properties of the structure and the simulation parameters. The BiCGStab algorithm is used, which can reduce the runtime for challenging simulation very drastically.

Note! Using the BiCGStab method approximately doubles the amount of RAM needed for the computation.

Unfortunately, the Krylov method is not always faster than a simulation without the Krylov method and therefore we introduced an **Automatic** mode which uses some heuristics to choose the most efficient method based on structure, material parameters, and boundary conditions automatically. Of course, it is possible to explicitly enable (**Enabled**) or disable (**Disabled**) the method.



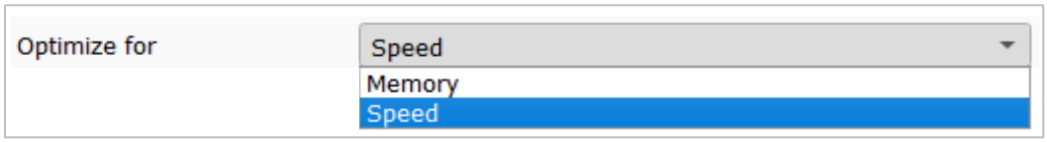
Note! In case that the Krylov subspace method (BICGStab) is used, the

Relaxation may also be adapted automatically.

Depending on the material parameters and geometry of the structure, the underlying mathematical problem can vary in complexity, thus influencing the behavior of the solver. The more complex the problem is, the more stable the solver settings should be.

With the **Relaxation** number, the solver is adjusted from **Stable** (which results in higher number of iterations, slower time stepping, and longer solver run times), to **Fast**, which makes the solver run less iterations but implies the risk that the solver does not converge. The **Relaxation** is a parameter of the SOR method and must be between 0 and 2 to ensure convergence. For relaxation values smaller than one (<1.0), the simulation is more stable. For relaxation values larger than one (>1.0), the simulation converges faster.

The **LIR** solver can **Optimize for Speed** or **Memory**.



- If **Speed** is chosen, the solver constructs additional optimization structures. The runtime is decreased by up to 30% but requires up to 50% more memory compared to the other option.
- If **Memory** is chosen, the runtime is increased by up to 40% but the solver requires up to 50% less memory.

Advanced Options: Grid Options

The **Grid Type** decides what kind of tree structure is used for the simulation.

The default option is **LIR-Tree** and should always be used. The solver uses an adaptive grid structure called LIR-tree and needs up to 10 times less runtime and memory compared to the **Regular Grid** option.

The solver can analyze the result field during the computation and improves the adaptive grid in places where more accuracy is needed. The LIR solver splits cells where a high gradient occurs.

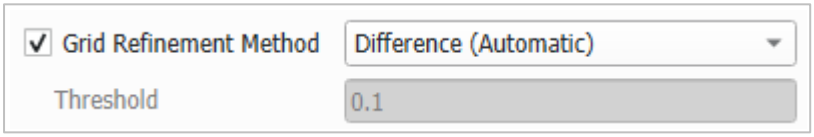
The solver can analyze the computed field and refines the adaptive grid during the computation at locations where more accuracy is needed. The LIR solver splits cells

where a high gradients occur when the **Grid Refinement Method** is enabled. In this case, the additional parameters **Number of Grid Refinements**, **Allow Sub-Voxel Resolution**, and **Allow Grid Re-Coarsening** become available.

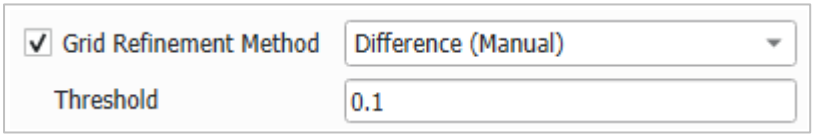
Select one of the three available refinement methods from the drop-down menu.

When **A Posteriori Error Bound** is selected, the solver targets the specified accuracy **Threshold**. While the **Error Bound** (set as Simulation Stopping Criterion) determines the relative error in the solution of the linear system, **A Posteriori Error Bound** refers to the relative error estimated by comparing high-order and low-order discrete solutions. The accuracy **Threshold** refers to the relative error to the analytical solution and the value must be between 0.0 and 1.0.

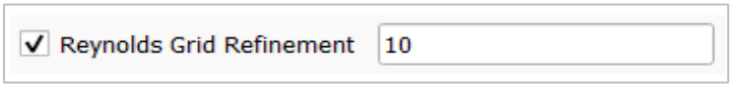
Choosing **Difference (Automatic)** leads to computational cells being split when the difference in values between neighboring cells exceeds a certain **Threshold**. The solver automatically chooses all internal parameters based on the structure and simulation settings. For this option, cells are split where the current gradient is greater than the **Threshold** multiplied by the maximum gradient, where the threshold is determined automatically.



Selecting **Difference (Manual)** also causes the computational cells to be split when the difference in values between neighboring cells exceeds the specified **Threshold**. You can specify all parameters (Threshold and Number of Grid Refinements) for the grid refinement manually. For this option, cells are split where the current gradient is greater than the **Threshold** multiplied by the maximum gradient. The **Threshold** value must be between 0.0 and 1.0. The recommended value range is between 0.05 and 0.1.



Activate **Reynolds Grid Refinement** to use the local Reynolds number (Re) for adaptive grid refinement.



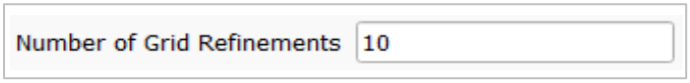
A grid cell is split if the local Reynolds number, i.e., the ratio between inertial and viscous forces exceeds the given threshold. The Reynolds number is defined as:

$$\mathrm{Re} = \frac{\rho \mathrm{u} \mathrm{L}}{\mu} \quad (51)$$

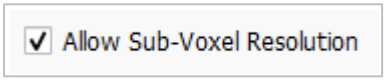
Reynolds Number

where ρ is the density of the fluid (SI units: kg/m^3), u is the velocity of the fluid (m/s), μ is the dynamic viscosity of the fluid (Pa·s or N·s/m^2 or kg/m·s), L is a characteristic length (m). For the local Reynolds number, the size of the grid cell is used for the characteristic length.

The **Number of Grid Refinements** controls the maximum number of grid refinements that the solver can perform during the simulation. The value should be set between 0 and 10, where a value of 0 means that no grid refinements will be made. Grid refinements may increase the number of iterations, runtime, and memory requirements.



When **Allow Sub-Voxel Resolution** is enabled, the solver is allowed to split computational cells to sizes smaller than the voxel length.



This feature is beneficial for low-porosity structures for which the pore throats require finer resolution or for modeling fast Navier-Stokes flows with strong vortices.

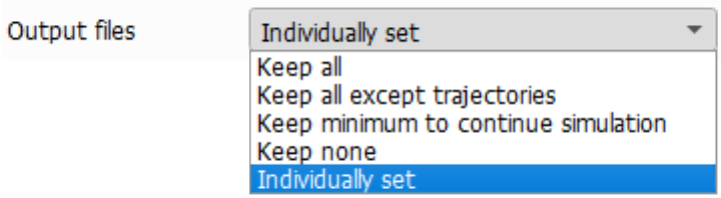
Enabling this feature may increase the number of iterations, runtime, and memory requirements.

Check **Allow Grid Re-Coarsening** to allow the solver to automatically revert the grid refinement and coarsen the computational cells.

This means, grid refinement is done temporarily. Afterwards, the cells are merged as soon as accuracy is not needed anymore. This reduces the memory and runtime requirements.

Output

First, select which **Output files** should be stored. You may select one of the predefined settings **Keep all**, **Keep all except trajectories**, **Keep minimum to continue simulation**, **Keep none**, or **Individually set**.



All these files are needed only for visualization of the results. The choices made here do not influence the filter lifetime results contained in the GDR (result) file but may disable some of the visualization options which require to load additional data.

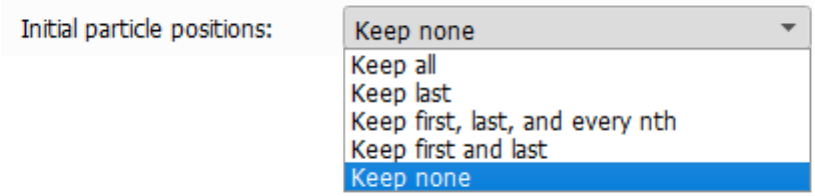
When **Keep all** is checked, all intermediate files written and read by the flow solver, particle tracker and electrostatic solver are kept. If the filter model is large, and/or the simulation contains many batches, a large amount of potentially unnecessary data is stored in that way.

Check **Keep all except trajectories** if you want to keep all data files except of the particle trajectories. Trajectory files are needed solely for visualization of the particle movement and may become very large if many particles are tracked.

Check **Keep minimum to continue simulation** to save only the files need to continue from an interrupted simulation.

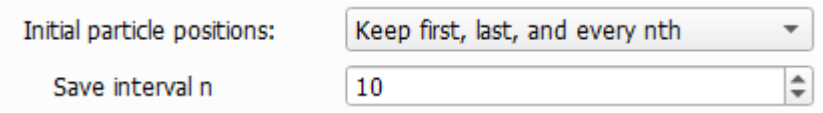
Check **Keep none** if you want to reduce the amount of hard disk space needed as far as possible.

If you select **Individually set**, you may decide in detail which files to keep for visualization. For each file type, choose between **Keep all**, **Keep last**, **Keep first, last and every nth**, **Keep first and last** and **Keep none**.

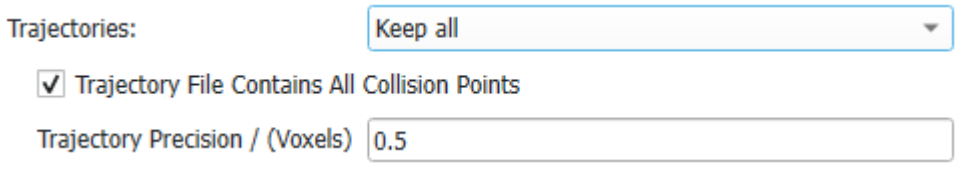


Decide, for which batch this file type is stored: every batch, only the first and/or last batch, or set an interval such that this file type is stored for every nth batch.

In that case, set the **Save interval n**:



To visualize the movement of particles, you need to keep the **Trajectories**. Be aware, that these files may become very large if many particles are simulated.



You can influence the file size and the precision of the stored trajectories (e.g., the number of points on the particle paths saved in the file) by two additional parameters:

- Check **Trajectory File Contains All Collision Points** to add all points where a particle touches the surface of a solid object to the trajectory file.
- The value entered as **Trajectory Precision** determines how accurately the trajectories are stored. The given value determines the traveling distance between two points stored in the trajectory file.

Results

Click **OK** to input the entered parameters, and then click **Run** in the FilterDict section to start the Filter Lifetime command. The results are immediately shown in the opening Result Viewer after the simulation is finished.

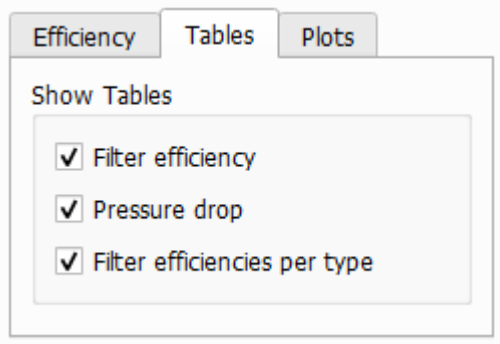
Report

Stopping Criterion

At the top of the **Report**, the stopping criteria reached by the simulation is shown. Here, it is also reported if the simulation was canceled (**Stopped by user**). If you open the result file while the simulation is still running, the first line reads **Simulation unfinished**.

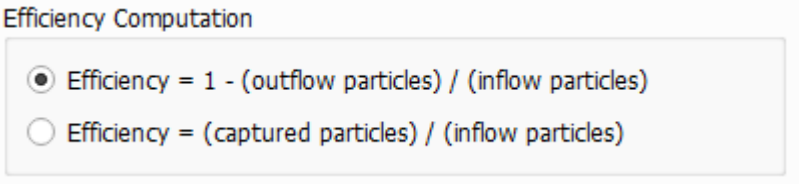
Tables

Below, a number of tables are presented, that can be selected under the **Tables** tab in the post-processing section on the left.



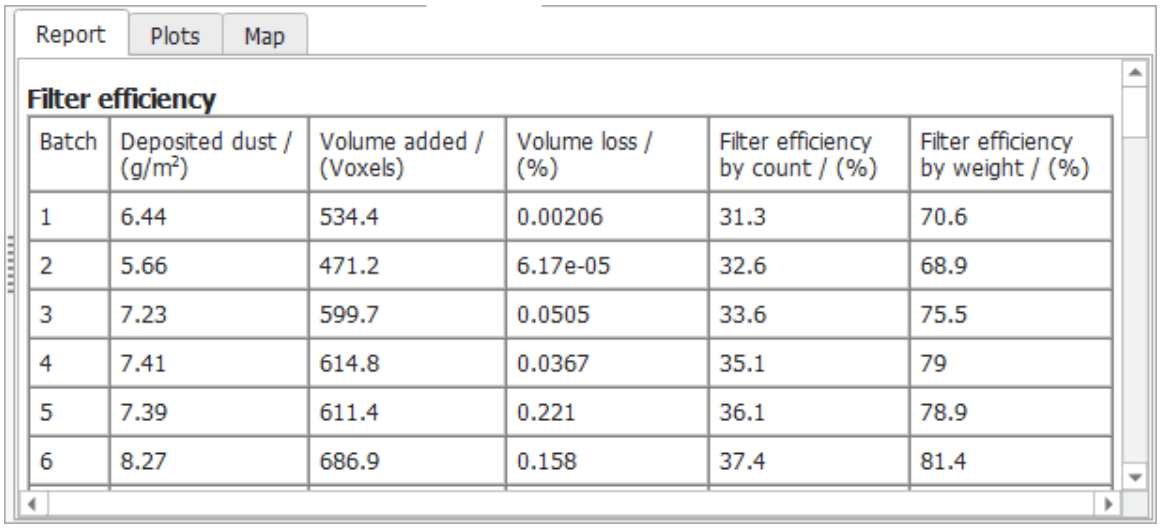
Filter efficiency

The **Filter efficiency** table contains, for every batch, the **Deposited dust**, the **Volume added** to the material by the deposited particles, the **Volume loss**, and the **Filter efficiency** in percentage **by count** and **by weight**.



The **Efficiency Computation** panel allows to select the definition of the filtration efficiency. It can either be based on the number of particles which leave the domain via the outflow region or on the number of particles which are captured. The difference

between both definitions is the handling of particles which are still in the domain but are not (yet) marked as captured.



Report | Plots | Map

Filter efficiency

Batch	Deposited dust / (g/m²)	Volume added / (Voxels)	Volume loss / (%)	Filter efficiency by count / (%)	Filter efficiency by weight / (%)
1	6.44	534.4	0.00206	31.3	70.6
2	5.66	471.2	6.17e-05	32.6	68.9
3	7.23	599.7	0.0505	33.6	75.5
4	7.41	614.8	0.0367	35.1	79
5	7.39	611.4	0.221	36.1	78.9
6	8.27	686.9	0.158	37.4	81.4

The **Volume loss** is the particle volume lost during each batch due to the overlapping of particles. This value is a good indicator whether the number of particles per batch has been chosen correctly. If the volume loss is too high, the number of particles per batch should be reduced to achieve a sufficient accuracy of the computation.

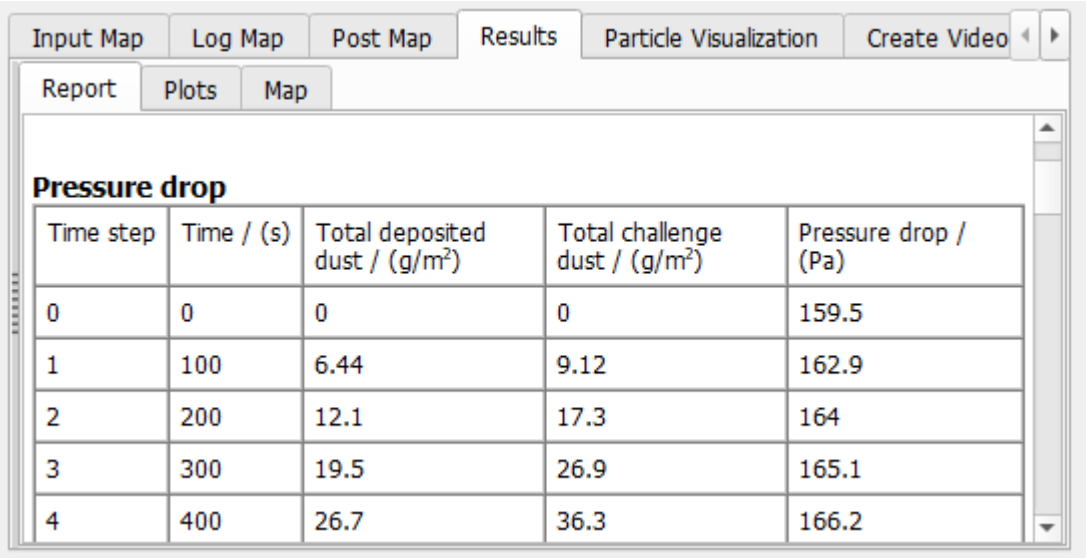
The overall volume loss is reported below the Filter Efficiency table:

```
Volume loss: overlapping particles share volume, which is lost when converting to voxels.
Overall volume loss is: 0.303%

Overpacked: particle volume higher than adjusted max. packing density.
Overall overpacked volume is: 9.01e+03 voxels
That is a percentage of: 42.8%

Filter efficiency = 1-(NumberOfParticlesOutflow) / (NumberOfParticlesInflow).
Note: there may exist time-out particles in the medium.
```

For simulation with unresolved particles, also the number of voxels filled by a higher volume fraction than the defined maximal packing density is reported. If some particles are smaller and some are larger than the voxel length (**Mixed** particle resolution mode), the number of overpacked voxels is typically very large (all locations where a large particle is deposited are overpacked, because a large particle fills up several voxels completely).

Pressure drop

The **Pressure drop** table contains the **Total deposited dust**, the **Total challenge dust**, and the **Pressure drop** for every batch.

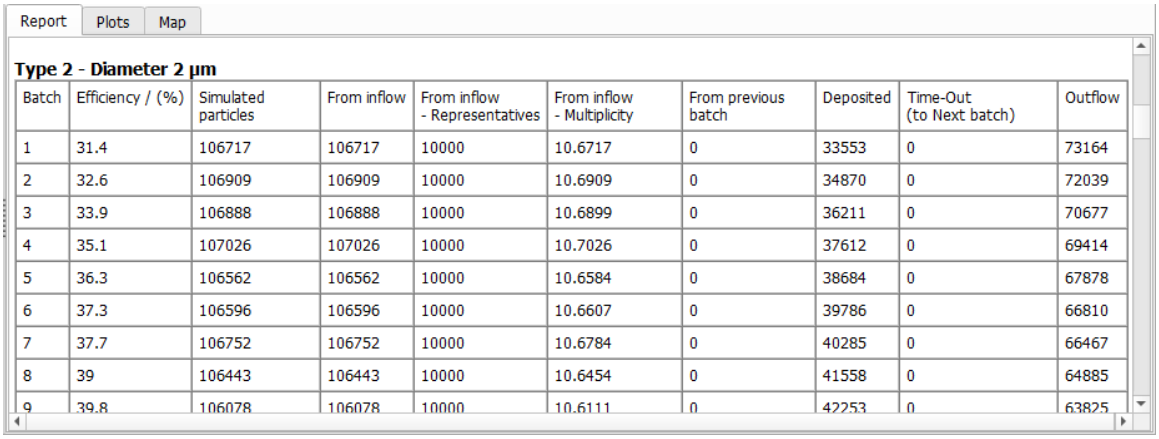


Time step	Time / (s)	Total deposited dust / (g/m²)	Total challenge dust / (g/m²)	Pressure drop / (Pa)
0	0	0	0	159.5
1	100	6.44	9.12	162.9
2	200	12.1	17.3	164
3	300	19.5	26.9	165.1
4	400	26.7	36.3	166.2

Filter efficiencies per type

The **Filter efficiencies per particle type** tables shows the filter **Efficiency** of a certain particle type for every batch. It shows the overall number of **Simulated Particles**, consisting of particles entering **From Inflow**, or continuing movement **From Previous Batch**.

If **Particle Multiplicity** was used, particles entering from the inflow area are simulated as representative trajectories with a given multiplicity. The table reports how many **From Inflow -Representatives** were simulated and the average **From Inflow – Multiplicity** of those particle trajectories. The multiplicity of a representative trajectory is always an integer value, so a value of 10.67 indicates that there are some representatives with a multiplicity of 10, and others with a multiplicity of 11.



Batch	Efficiency / (%)	Simulated particles	From inflow	From inflow - Representatives	From inflow - Multiplicity	From previous batch	Deposited	Time-Out (to Next batch)	Outflow
1	31.4	106717	106717	10000	10.6717	0	33553	0	73164
2	32.6	106909	106909	10000	10.6909	0	34870	0	72039
3	33.9	106888	106888	10000	10.6899	0	36211	0	70677
4	35.1	107026	107026	10000	10.7026	0	37612	0	69414
5	36.3	106562	106562	10000	10.6584	0	38684	0	67878
6	37.3	106596	106596	10000	10.6607	0	39786	0	66810
7	37.7	106752	106752	10000	10.6784	0	40285	0	66467
8	39	106443	106443	10000	10.6454	0	41558	0	64885
9	39.8	106078	106078	10000	10.6111	0	42253	0	63825

At the end of the batch, particles are either **Deposited**, are still moving inside of the structure (**Time-Out**), or have left the domain through the **Outflow**.

Depending on the choice of parameters, some particles may neither count as Deposited, Time-Out, or Outflow, e.g., because they leave the domain through the

inflow area. In this case, those particles are removed completely from the reported statistics. Those particles are not included in the number of **Simulated Particles** and simulated **Representatives**.

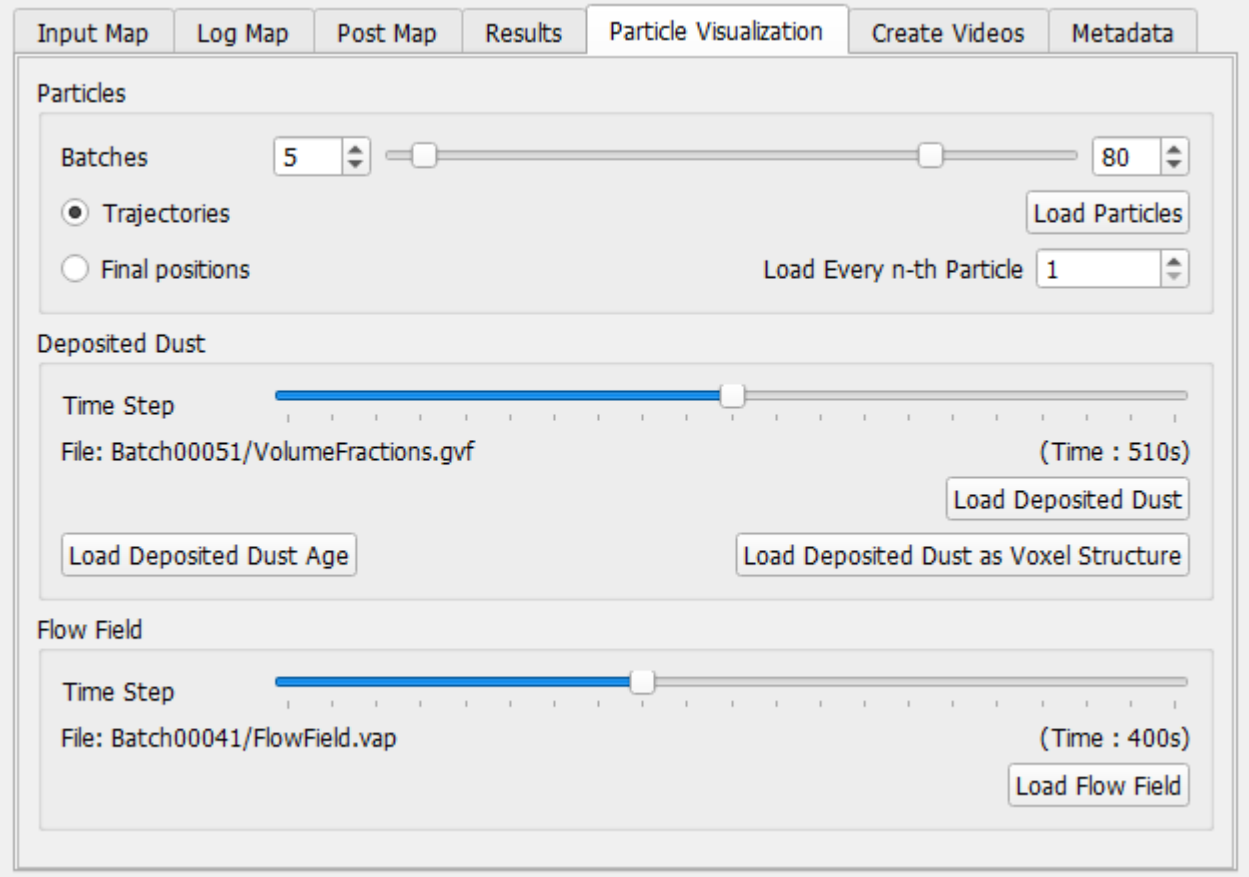
Be aware that a representative particle trajectory starting with a multiplicity of 11, may arrive at the outflow with a multiplicity of 3, and would in this case lead to 8 deposited particles and 3 particles leaving through the outflow.

Plots

The available plots are identical to the filter media result plots.

Particle Visualization

To visualize the initial and final positions of particles, track their trajectories, and study the flow of the fluid through the filter structure, select the **Particle Visualization** tab.



Particles

In the **Particles** panel, select one or several succeeding batches with the sliders. Select **Trajectories** to animate the particle motion from inflow to outflow or deposition place. If a

large number of particles was simulated, loading all the trajectories at once is memory consuming and may not be possible. Loading only the **Final positions** does not allow to animate the particle motion, but is less memory consuming. Memory consumption may also be reduced by visualizing only a fraction of the simulated particles per batch by loading only **Every n-th Particle**.

Clicking **Load Particles** loads the selected trajectories or final particle positions and displays the particles in the viewport.

When visualizing results of filtration simulations on the pleat scale or filter element scale, it is often necessary to enlarge the particles Diameter Factor for better visibility, because particle diameters are small in comparison to the structure model and therefore difficult to depict when shown in their original size.

Because of the need to enlarge particles and because of the fact that those particles show only a representative particle when using **Multiplicity**, the build-up of a filter cake cannot be visualized properly on the filter element scale by visualizing every single particle. Also, the total amount of particles simulated would make this task impossible or very demanding for the computer's graphic card. Therefore, it is recommended to use the **Deposited Dust** result fields to visualize the development of the filter cake, as explained below.

Deposited Dust

Use the slider to select the **Time Step**. There are three options to visualize the deposited dust from the simulation:

- With **Load Deposited Dust**, a result field (*.gvf) is loaded that contains for every voxel a single double value in the range [0,1], describing the volume fraction of the voxel filled with dust particles.
- With **Load Deposited Dust as Voxel Structure**, the volume field is converted to a voxel structure.
- With **Load Deposited Dust Age**, a result field is loaded that contains for every voxel a single integer value describing how many batches ago this particle was deposited. Voxels filed in the last batch contain the value zero, voxels filled in the previous batch the value 1, and so on...

Clicking on the appropriate button loads the data and displays the deposited dust in the viewport 117.

Flow Field

Use the sliders to select the corresponding time step. Click **Load Flow Field** to load the result and refer to the Visualization user guide for details.

Create Videos

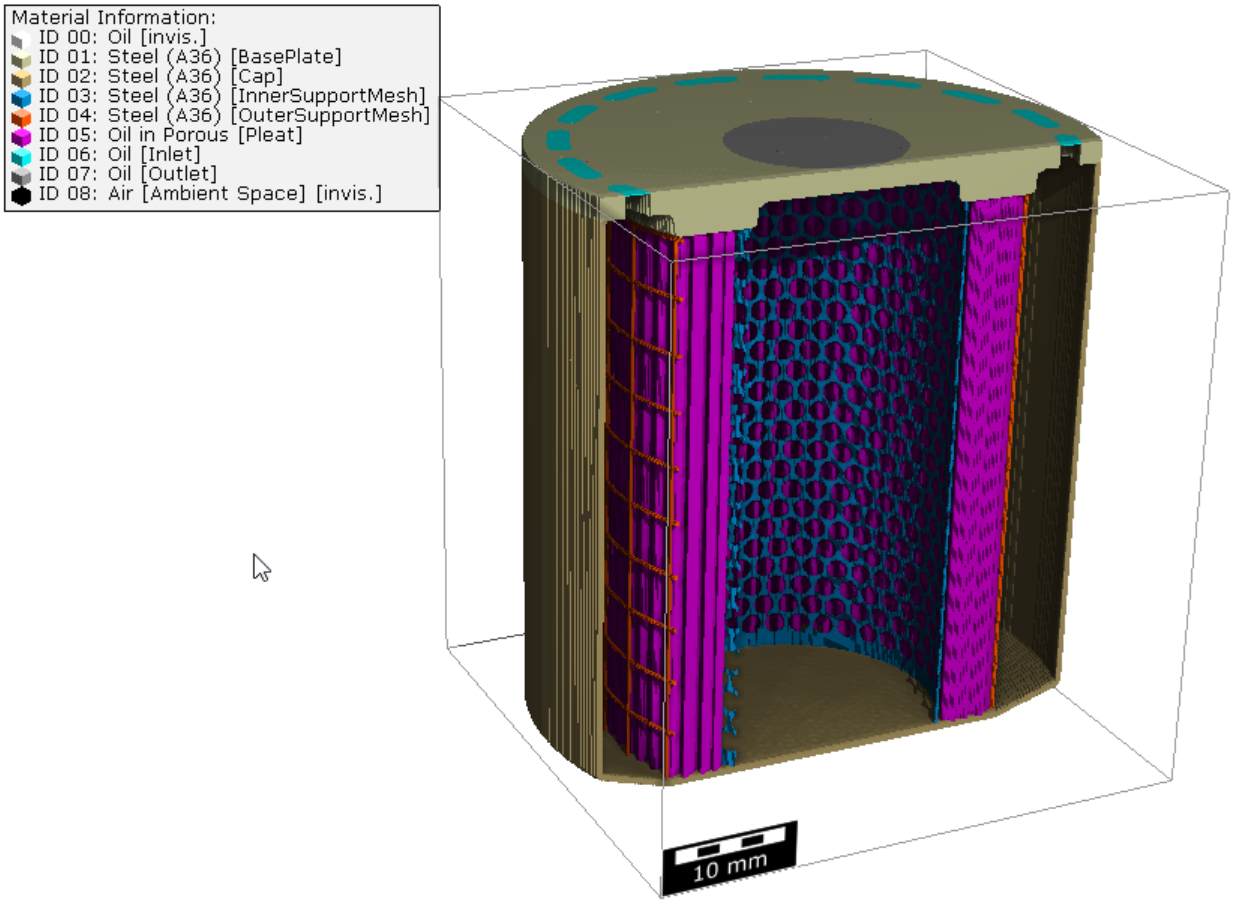
Under the **Create Videos** tab, two predefined videos can be generated. In the drop-down menu, select either the Deposited Particles Video or the Pressure Drop Video.

1.4 Complete Filter

To model particulate flow through a complete filter, select **Complete Filter** from the drop-down menu.

The main difference to the previously described Filter Media 18 and Filter Element 125 options lies in the flow direction. Both **Filter Media** and **Filter Element** simulate a flow from the inflow plane located at Z- to the outflow plane located at Z+. With **Complete Filter**, arbitrary areas can be defined as inflow areas and as outflow areas.

In the example below, multiple inflow areas (ID 06) and one outflow area (ID 07) are all located on the Z- side of the cubic domain.



Commands available for the analysis of complete filters

- Filter Lifetime 170
  A Filter Lifetime simulation includes clogging of the filter and computes pressure drop and deposited dust over time. In a Single Pass simulation, fluids pass through the filter only once and are not recirculated. In a Multi Pass simulation, fluids move in a circuit through the system, and particle size distribution and concentration in front of the filter change over time.
- Filter Flow Experiment 202
  A Filter Flow Experiment simulates the flow through the filter and determines the initial pressure drop.

1.4.1 Filter Lifetime

A **Filter Lifetime** simulation includes clogging of the filter and computes pressure drop and deposited dust over time. In a Single Pass simulation, fluids pass through the filter only once and are not recirculated. In a Multi Pass simulation, fluids move in a circuit through the system, and particle size distribution and concentration in front of the filter change over time.

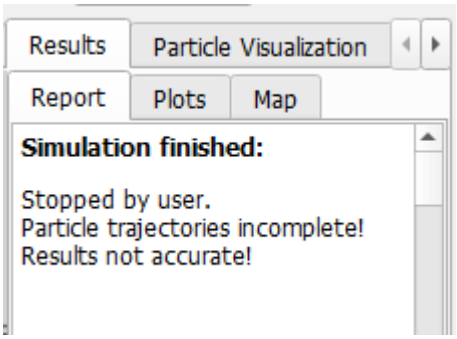
The Filter Lifetime command for complete filter media

- Options 171
  How to set the input parameters for the Filter Lifetime command.
- Results 200
  How to understand the computed results.
- Particle Visualization 201
  How to visualize the computed results.

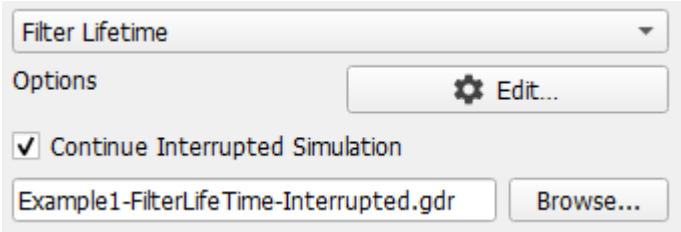
Continue Interrupted Simulation

An interrupted filter life time simulation can be restarted by checking **Continue Interrupted Simulation**. For this, at least the computation of the first flow field must have been finished. When the simulation was interrupted earlier, it must be started anew.

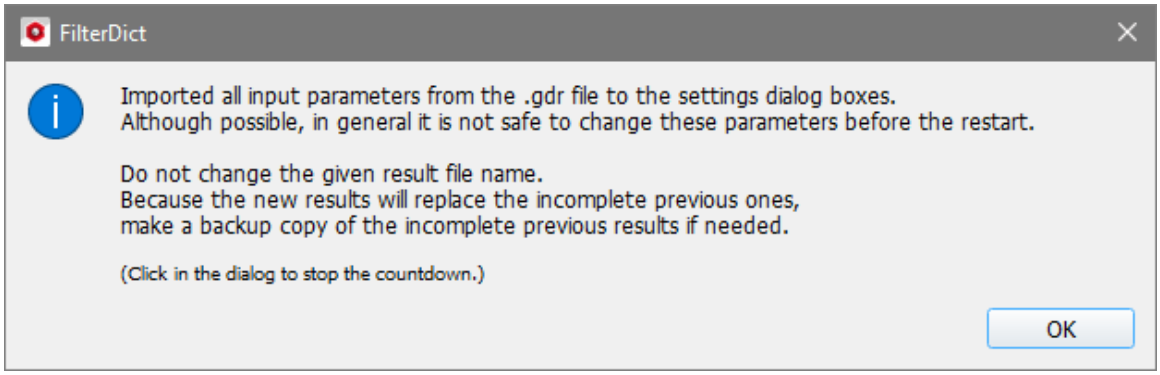
A stopped or interrupted simulation produces an incomplete (*.gdr) result file which may not automatically open at the end of the computations. When opened through **File → Open *.gdr File...**, you can find the information that the simulation was not finished under the **Results - Report** subtab:



To continue the unfinished simulation, check **Continue Interrupted Simulation** and click **Browse** to select the name of the result file. To continue the interrupted simulation, the result file from the unfinished simulation must be located in the project folder and the structure on which the simulation is to be finished must be already loaded.



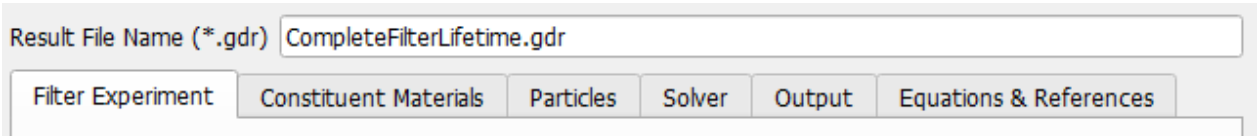
A message appears giving information and instructions on how to handle the simulation re-run.



Click **OK** in the message to return to the FilterDict section and click **Run** to continue the interrupted simulation..

Options

Select **Filter Lifetime** from the pull-down menu and click the **Option**'s **Edit...** button to open a dialog to enter or edit the simulation options. Choose a **Result File Name** for the result file (*.gdr) fitting to your current project.



The parameters are organized under the tabs: **Filter Experiment**, **Constituent Materials**, **Particles**, **Solver**, and **Output**. The **Equations & References** tab gives a summary of the used equations and symbols which are explained in more detail in this user guide.

Tabs of the Complete Filter Lifetime dialog

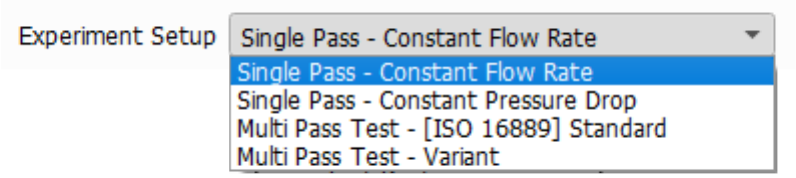
- Filter Experiment 172
  Select Single Pass or Multi Pass experiment, determine onflow and outflow materials and set the boundary conditions.
- Constituent Materials 176
  Define fluid type and temperature. Set material IDs and parameters for porous layers.
- Particles 181
  Select particle sizes and their start positions. Set all parameters that govern the particle motion and define collision and adhesion models.

Tabs of the Complete Filter Lifetime dialog

- Solver 190
  Select the flow solver, set the stopping criteria and define the computational resources.
- Output 199
  Choose which data will be stored for visualization.

Filter Experiment

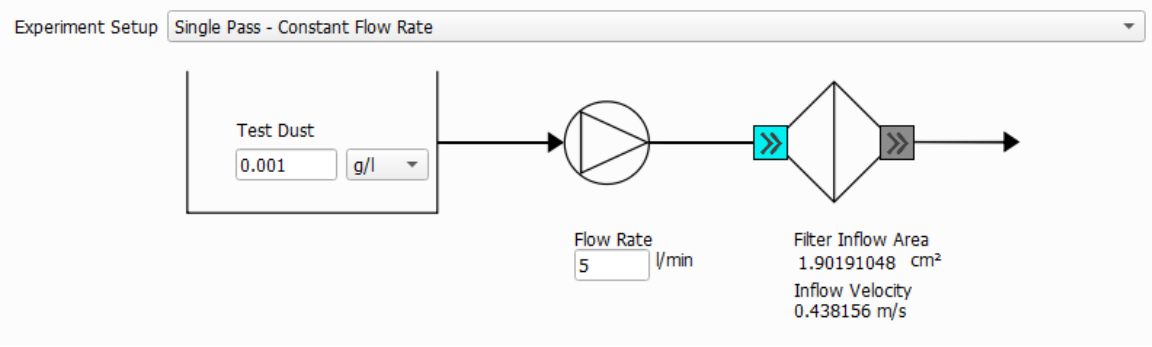
Under the **Filter Experiment** tab, choose between a **Single Pass** or a **Multi Pass** experiment. For **Single Pass**, the experiment can be run at a **Constant Flow Rate** or at a **Constant Pressure Drop** (often used for membranes). For **Multi Pass**, choose between the standard test setup or a variant with an initially contaminated test reservoir.



When a method is selected, the setup is shown schematically in the dialog below the pull-down menu. Furthermore, the selected method will influence which options are available elsewhere in the Filter Lifetime dialog.

Single Pass - Constant Flow Rate

In this case, the pump flow rate is kept constant during the experiment and the pressure drop increases over time. For this experiment, enter the **Test Dust** concentration in the fluid (in g/l or g/m^3) and the **Flow Rate** (in l/min).



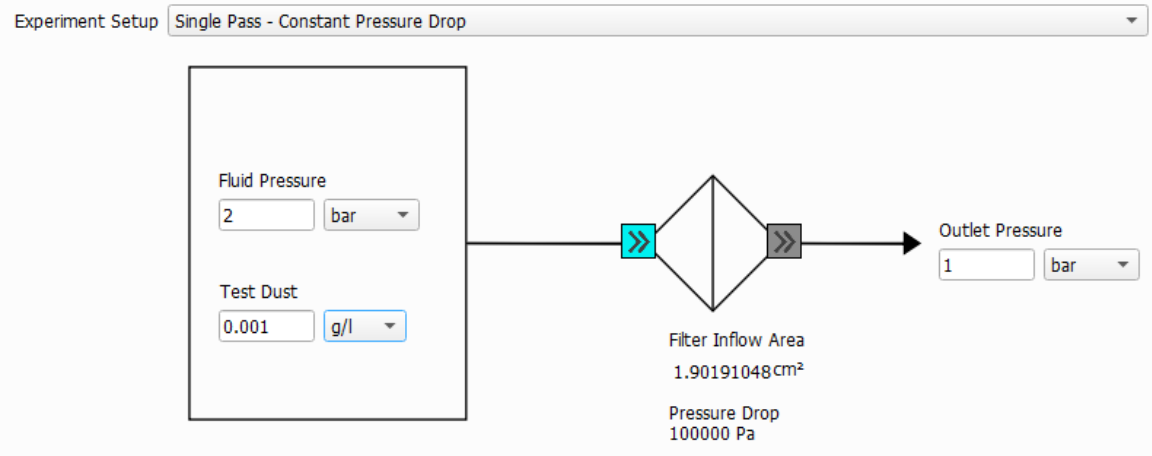
The arrow symbols before and after the filter show the color of the inflow material (here cyan) and the outflow material (here dark gray). **Filter Inflow Area** and **Inflow Velocity** are computed automatically.

The combination of the two parameters Test Dust concentration and Flow Rate control other simulation parameters, as for example, the number of particles per batch (in the **Solver** tab).

Single Pass - Constant Pressure Drop

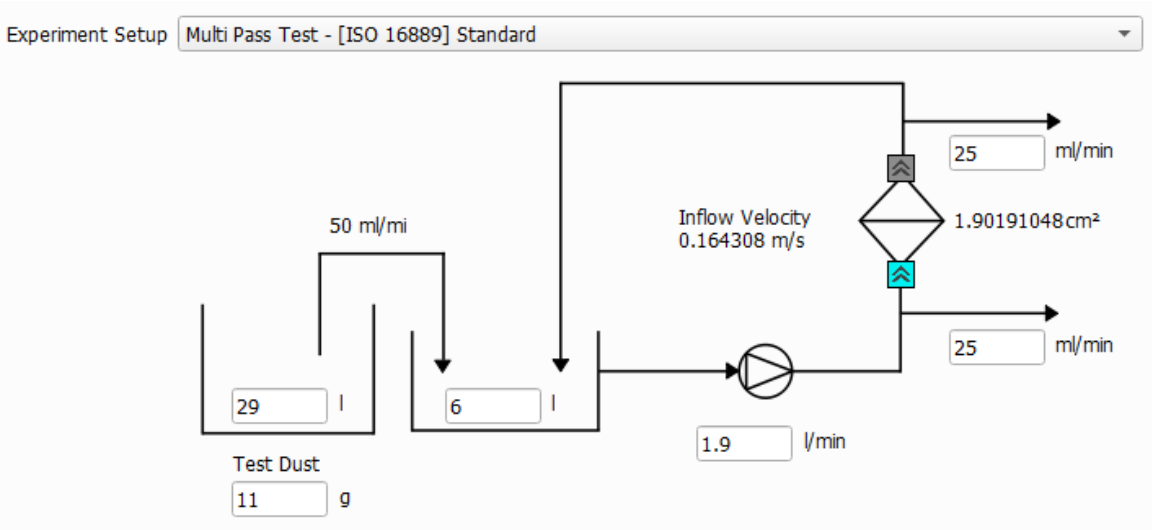
With **Constant Pressure Drop**, the pressure drop at the filter is kept constant and the mean velocity / flow rate decreases over time.

For this choice, enter the **Fluid Pressure** in the containment, the **Outlet Pressure**, and the **Test Dust** concentration in the fluid. The arrow symbols before and after the filter show the color of the inflow material (here cyan) and the outflow material (here dark gray) and the **Filter Inflow Area** is computed automatically.



Multi Pass Test - [ISO 16889] Standard

A schematic of the multi-pass filtration process is shown. Tooltips appear when resting the mouse on the boxes, explaining the meaning of each value.

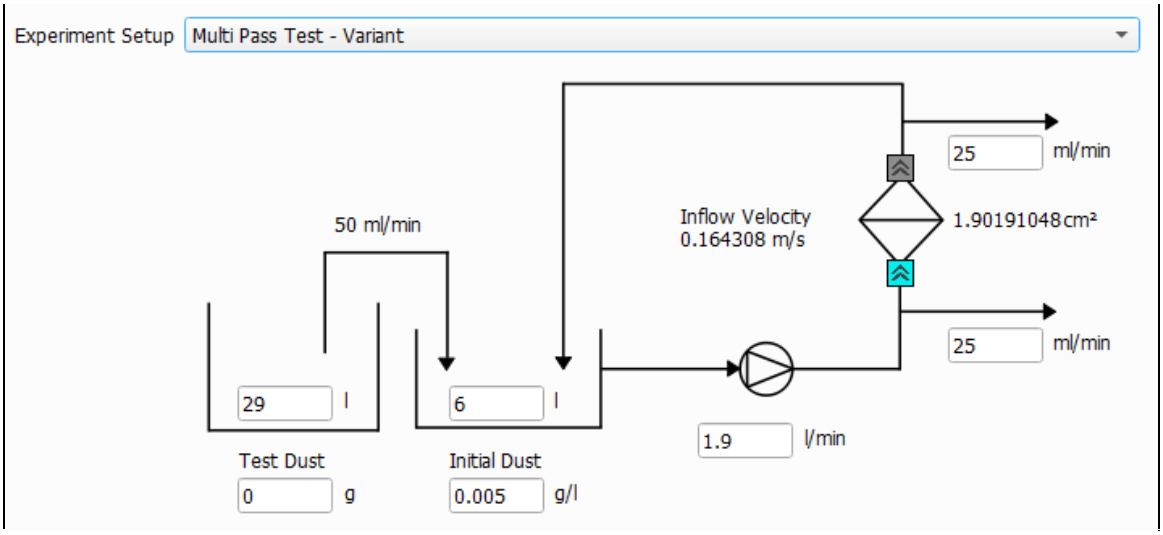


Besides the **Test Dust** (in g), the **Pump flow rate** (in l/min), the **Injection reservoir volume**, and the **Test reservoir volume** (recirculation) can be entered. The **Outflow to upstream sampler** and the **Outflow to downstream sampler** (in ml/min) can also be set. The **Injection flow rate** matches the sum of the upstream and the downstream outflows (in ml/min). The arrow symbols before and after the filter show the color of

the inflow material (here cyan ») and the outflow material (here dark gray »). **Filter Inflow Area** (here 1.90191 cm^2) and **Inflow Velocity** are computed automatically.

Multi Pass Test - Variant

As an alternative, a **Multi Pass Test - Variant** is available. With this option, also the amount of dust in the Test Reservoir can be set in the beginning of the simulation. The arrow symbols before and after the filter show the color of the inflow material (here cyan ») and the outflow material (here dark gray »). **Filter Inflow Area** (here 1.90191 cm^2) and **Inflow Velocity** are computed automatically.

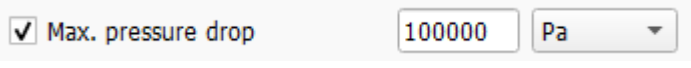


Simulation Stopping Criterion

The simulation continues to run until one of the following criteria is reached:

- If **Max. pressure drop increase** is selected, the simulation stops if the initial pressure drop is increased by the given value. This option is not available if **Single Pass - Constant Pressure Drop** was selected.

- If **Max. pressure drop** is selected, the simulation stops if the given pressure drop is reached. This option is not available if **Single Pass - Constant Pressure Drop** was selected.

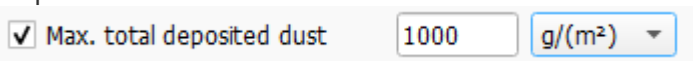


- If **Max. flow rate decay** is selected, the simulation stops when the flow rate falls below the given percentage (%) of the initial flow rate. This option is only available if **Single Pass - Constant Pressure Drop** was selected.

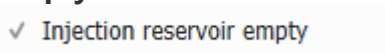
- If **Max. time reached** is selected, the simulation stops when the defined simulation time is

reached.

- If **Max. total deposited dust** is selected, the simulation stops when the defined amount of deposited dust is reached.



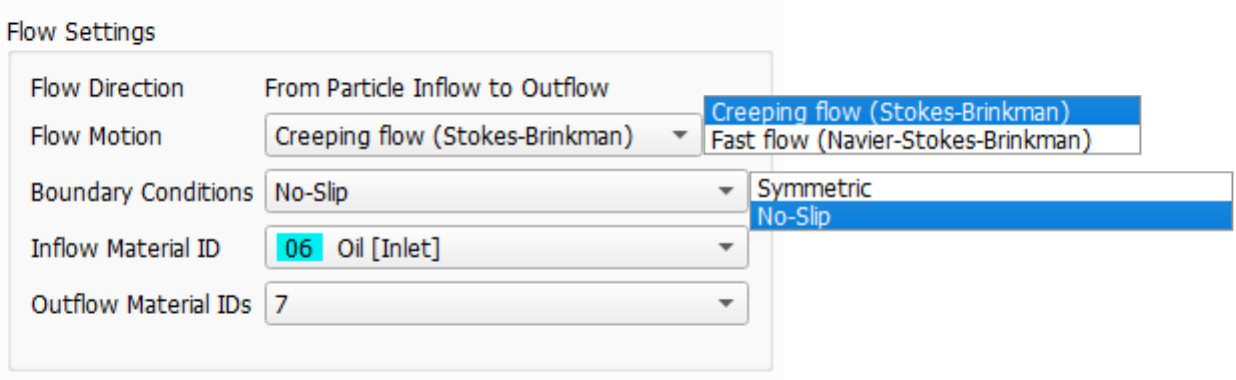
- **Multi Pass Test** simulations always automatically stop when the **Injection reservoir** is **empty**.



- The simulation always automatically stops when the **Inflow region** is **filled** with particles. This is triggered when the first deposited particle reaches the inflow plane.

Flow Settings

In this command, the **Flow Direction** is fixed to **From Particle Inflow to Outflow**.

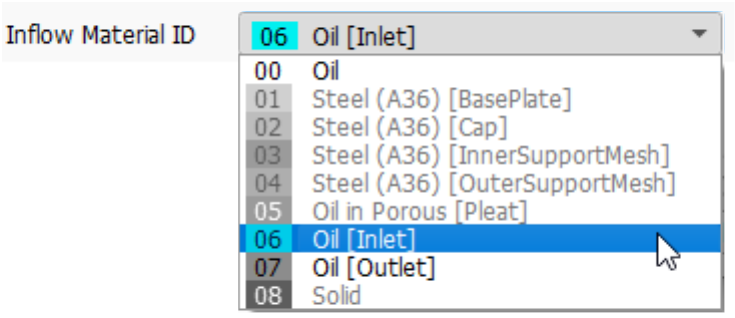


The **Flow Motion** can either be creeping or fast flow. In the creeping flow case, the Stokes equations are solved to determine the flow field. If fast flow is chosen, the Navier-Stokes equations are solved to determine the flow field.

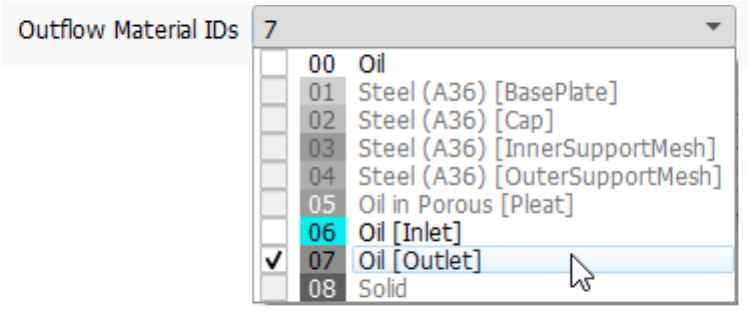
As the Stokes equation is a simplification of the Navier-Stokes equation, choosing **Fast flow (Navier-Stokes)** will give correct results also in the case of slow, creeping flow. Solving the Navier-Stokes equations is computationally more expensive than solving the Stokes equations, so it is advisable to choose **Creeping flow (Stokes)** whenever justified.

The 3D model of the complete filter typically contains also the filter housing. In this case, flow will only occur inside of the filter housing and the domain **Boundary Conditions** do not apply. However, if the filter is symmetric, it is possible to reduce the computational cost by modelling only one half or one quarter of the filter. In these cases, the flow area will be open to one or two sides of the domain and one should select **Symmetric** boundary conditions for the flow at the domain boundary. **No-Slip** means that the structure is encased in a solid wall at the domain boundary. Particles will be reflected at the boundary in this case.

As **Inflow Material** one of the fluid material IDs can be selected. After selecting a material ID here, the » symbol changes to the selected color and the **Filter Inflow Area** is recomputed.



As **Outflow Material**, one or several material IDs can be selected. After selecting a material ID here, the » symbol changes to the selected colors.



Constituent Materials

For simulations of the complete filter, the **Particle Resolution** is always unresolved (**Mixed**).

Temperature

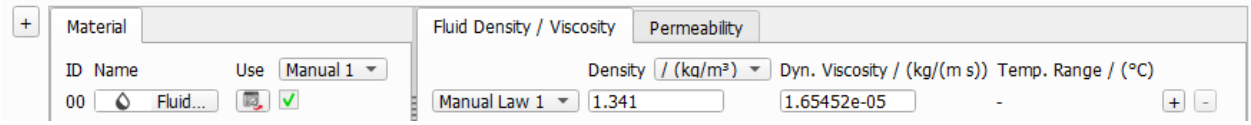
The **Temperature** for the filtration process is selectable in Kelvin (K), Celsius (°C), and Fahrenheit (°F). The chosen temperature may change the density and viscosity values of the fluid during the simulation, if those are defined to be temperature-dependent in the material database. Furthermore, if the Brownian motion of the particles is included in the simulation, it determines the strength of the Brownian motion through equation (13) 8.

Fluid Density and Viscosity

FilterDict uses single-phase flow simulations. It is not possible to run a flow simulation if two fluids are present in the structure. If multiple fluids are present in the structure, you may click **Set all Fluids to...** and select one fluid for the simulation.

The selected fluid is also displayed in the drop-down menu in the **Material** subtab behind **Use**. The density and the dynamic viscosity of this fluid are shown under the **Fluid Density/Viscosity** subtab.

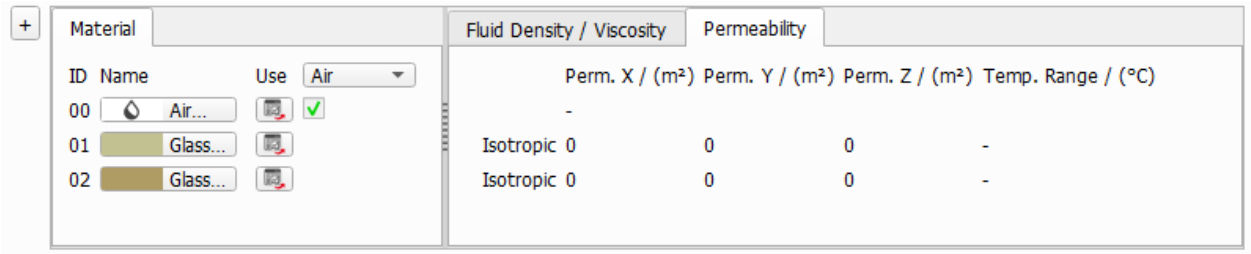
If a fluid from the GeoDict Material Database is used, the values for **Density** and **Dynamic Viscosity** are taken from the database and may depend on the given **Temperature** value. Select **Manual** as fluid material to enter parameter values manually.



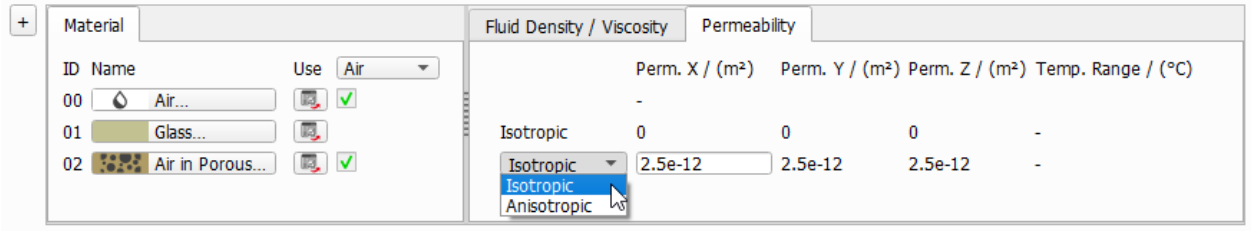
To add materials to the database, use the **Open & Edit Material Database** button. More information on editing, expanding, and using the **GeoDict Material Database** is available in the Material Database user guide.

Filter Media Permeability

Under the **Permeability** subtab, enter the permeability of porous materials present in the structure. As long as there are only fluid and solid materials present in the structure, no parameters are needed here:



For porous materials, choose if the material is **Isotropic** or **Anisotropic**. In the isotropic case, the permeability is the same in all space directions. In the anisotropic case, enter a different permeability for each spacial direction.



You can select and change a material by clicking on the material button.

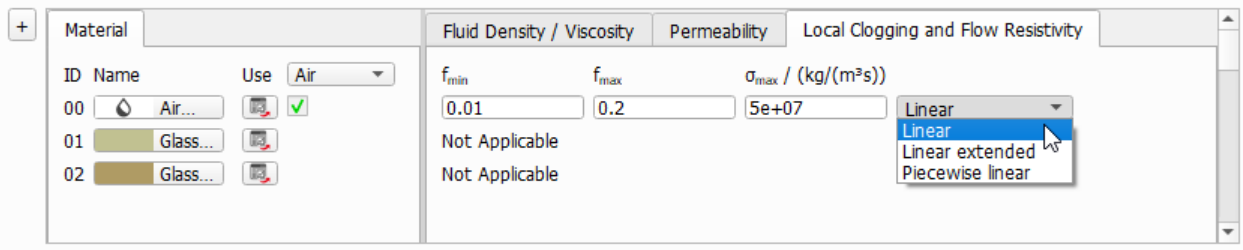
Know how! The anisotropic permeability is always aligned with the coordinate system, not with the orientation of the filter media. Therefore, it is often the best choice to select **Isotropic** and use the through-plane permeability of the flat sheet material, as this is the permeability most relevant to the flow direction.

Local Clogging and Flow Resistivity

If the unresolved particle model is selected, the **Local Clogging and Flow Resistivity** tab will

also be available.

Unresolved particles will create porous voxels when deposited into the 3D structure. In those voxels, flow is still possible, but experiences an increased flow resistivity. In the **Local Clogging and Flow Resistivity** tab, the relation between the dust volume fraction f inside of a voxel and the flow resistivity σ must be defined. This tab is only available for unresolved simulations, because in the **Resolved** case all voxels are treated as either empty or solid.



The flow resistivity σ depends on the values for $f_{\min}$, $f_{\max}$, and $\sigma_{\max}$, which must be defined for every material of the structure which can be filled by dust particles during the simulation, i.e., inside of the fluid (where the filter cake will form) and in all porous layers.

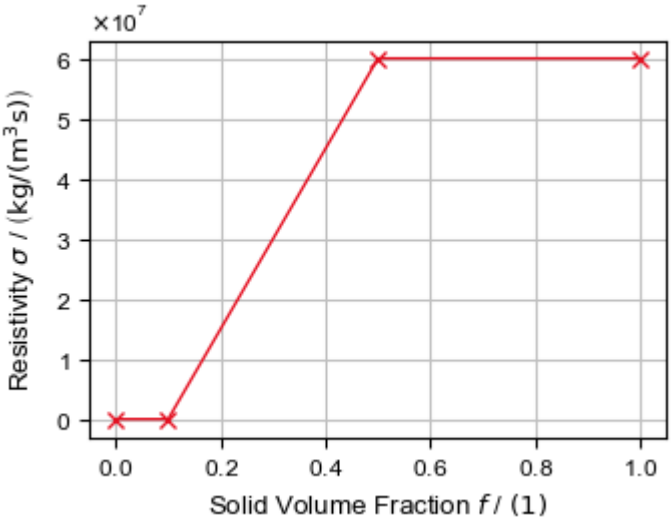
Note! It is necessary to set these parameters for the fluid material, although this might be unintuitive at first glance. Be aware that the parameters set for the fluid material describe the behavior of the filter cake because the filter cake consists of partially dust-filled voxels of the fluid domain.

Note! It is not possible to use different fluids in a structure, but it is possible to use different material IDs with the same fluid. This option is useful, when the filter cake should have different properties depending on its location in the structure.

Three different models are available to define the function $\sigma(f)$: **Linear**, **Linear extended**, and **Piecewise linear**.

Linear

In this model, the flow resistivity depends linearly in a certain interval on the local solidity.



The function $\sigma(f)$ is defined through:

$$\sigma = \begin{cases} 0 & \text{for } f < f_{\min} \\ \dfrac{f - f_{\min}}{f_{\max} - f_{\min}} \cdot \sigma_{\max} & \text{for } f_{\min} < f < f_{\max} \\ \sigma_{\max} & \text{for } f_{\max} < f < 1 \end{cases} \tag{52}$$

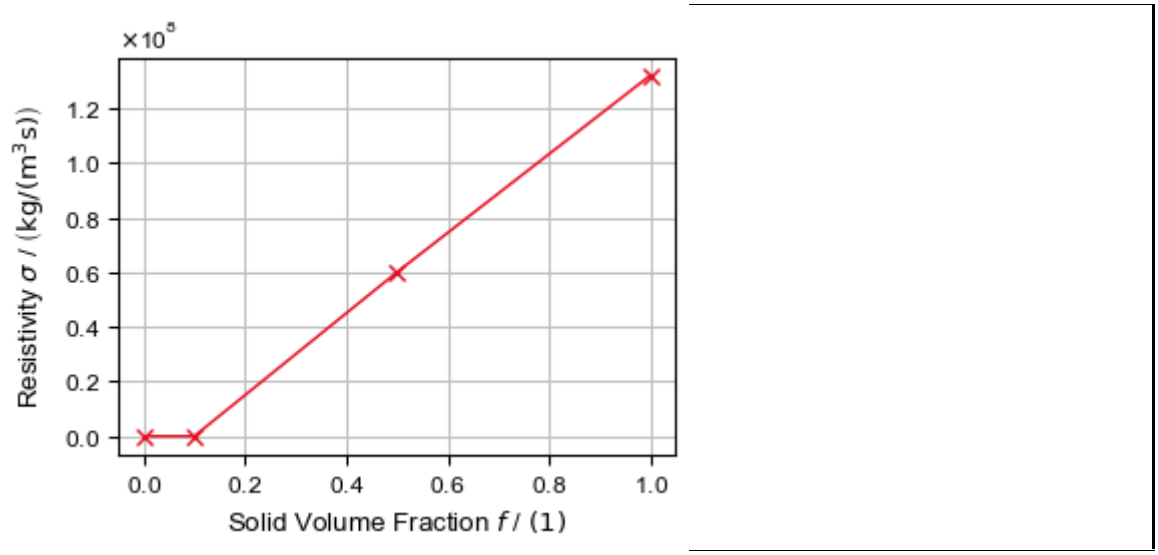
Know how! The general idea of this model is that $f_{\max}$ describes the maximal solidity (or Maximum Particle Packing Density) of the filter cake and $\sigma_{\max}$ the flow resistivity of the cake.

Under the assumption that all particle sizes are smaller than the voxel length (which holds true in many settings that consider a complete filter), using the solidity of the filter cake for $f_{\max}$ and the Maximal Particle Packing Density and using the flow resistivity as $\sigma_{\max}$ will lead to the expected results: The filter cake that emerges in the simulation will again have this solidity and flow resistivity.

If a significant number of particles are larger than the voxel length, the filter cake that emerges during the simulation will have a different solidity than $f_{\max}$ and a different flow resistivity as $\sigma_{\max}$. This has been described by Becker et al. 2016[257], and the paper also describes a way to estimate those parameters in this case.

Linear Extended

This model is basically the same as the **Linear** model, but the function is not cut off at the value of $\sigma_{\max}$, but extended linearly until $f = 1$.



The function $\sigma(f)$ is defined through:

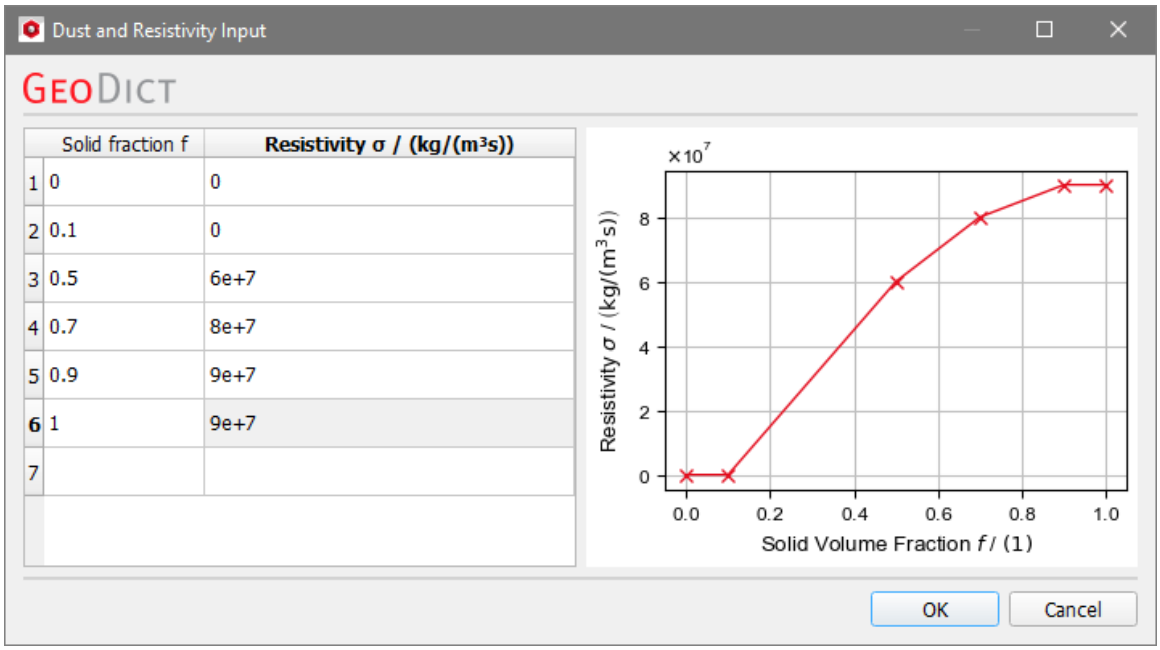
$$\sigma = \begin{cases} 0 & \text{for } f < f_{\min} \\ \dfrac{f - f_{\min}}{f_{\max} - f_{\min}} \cdot \sigma_{\max} & \text{for } f_{\min} \leq f < 1 \end{cases} \quad (53)$$

Know how! The local solidity inside of a voxel can exceed the Maximum Particle Packing Density after the volume of the last particles deposited in this voxel is added. In fact, any value between 0.0 and 1.0 may appear during the course of a simulation. Therefore, it makes sense to extend the linear model beyond $\sigma_{\max}$.

Piecewise Linear

If **Piecewise linear** is selected, an arbitrary piecewise linear function can be defined by clicking the **Edit...** button.

The **Dust and Resistivity** Input dialog open and allows to define a piecewise linear function:



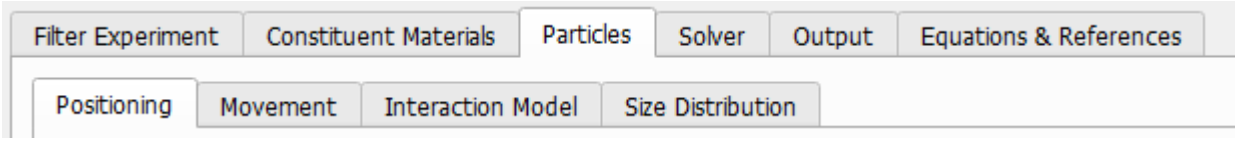
It is recommended to start the list with a solid fraction of 0, and end with a solid fraction of 1, to define the function for the whole interval [0,1).

Be aware that for a local solidity of 1.0, the voxel is always treated as solid, and no flow is possible. This is independent from the definition of the function $\sigma(f)$, which only applies to partially filled voxels (local solidity $f < 1$).

Know how! You can upscale the results of a fully resolved filter media simulation to determine these input parameters. In the Report tab of a **Filter Media - Filter Lifetime** command, the Filter Clogging Analysis 106 computes parameters that can be used as input here. Use the reported **Cake Filtration** values as $f_{\max}$ and $\sigma_{\max}$ of the fluid phase and the reported **Depth Filtration** values as $f_{\max}$ and $\sigma_{\max}$ of the porous filter material.

Particles

Under the **Particles** tab, four subtabs are available to define **Positioning**, **Movement**, **Interaction Model**, and **Size Distribution**.



Tabs of the Particles tab

- Positioning 182
  Define the start positions of the simulated particles.
- Movement 184

Tabs of the Particles tab

Select the parameters modelling the motion of the particle.

- Interaction Model 184
  Define collision and adhesion models for particles-material collisons.
- Size Distribution 188
  Define the particle size distribution and all particle size dependent parameters.

Positioning

Particle Inital Position

Particle Positioning Weights

The particles are placed randomly in the inflow area defined in the Filter Experiment 172 tab. The particles are placed with a probability based on the local fluid velocity, which will lead to a uniform particle concentration in the inflow.

Particle Start Velocity

Select either **Fluid Velocity** (the default option, which was used until GeoDict 2025) or **Given Velocity** as the **Particle Start Velocity**.

With the first option, particles begin moving with the fluid velocity at their start position.

With the second option, all particles start with the same user-defined velocity.

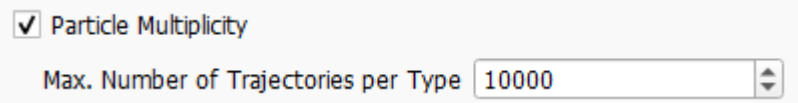
Random Seed

The parameter **Random Seed** controls the underlying random number generator. The same random seed produces identical results, whereas results with different random seeds are similar but not identical.

Particle Multiplicity

When the Particle Resolution is set to **Mixed** in the Constituent Materials tab, the parameter

Particle Multiplicity becomes available and a value can be entered in the **Max. Number of Trajectories per Type** box.



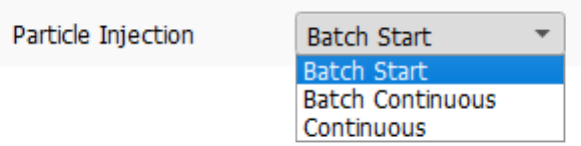
Unresolved particles may be much smaller than the used voxel length and thousands of dust particles are needed to fill a single voxel. Therefore, the number of **Particles per Batch** (on the **Solver tab**) may grow very large, leading to increased simulation times. With **Particle Multiplicity**, the number of particles that are tracked can be restricted.

When, for example, 1,000,000 particles of diameter 0.05 µm are to be simulated in the next batch, and you check **Particle Multiplicity** and set the **Max. Number of Trajectories per Type** to 10,000, only the movement of 10,000 particles is tracked. However, each of these particles represents 100 particles, and the multiplicity is 100 (100 x 10,000 = 1,000,000). If one of these particles is filtered, the mass and volume fractions of 100 particles are added to the deposited dust.

To make sure that the choice of **Particle Multiplicity** does not introduce large numerical errors, FilterDict ensures that the particle volume multiplied by its multiplicity is less than 5% of the single voxel volume.

Particle Injection

All particles simulated in a time inteval (a batch) move independently from each other. Select, if all particles should start moving at the beginning of the time interval (**Batch Start**) or if starting times should be distributed uniformly over the batch interval (**Continuous**).

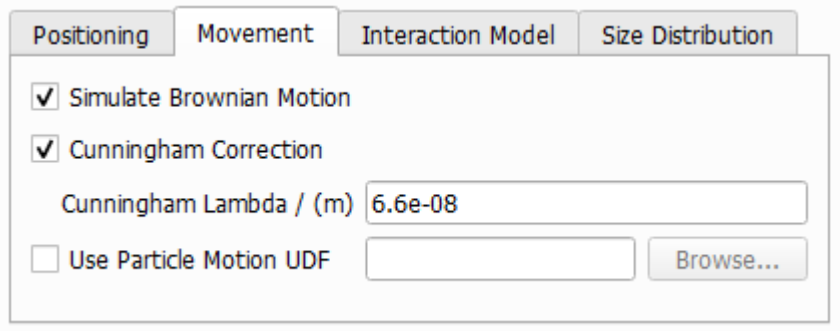


While **Continuous** offers a physically correct description of what happens in a real filter experiment, it may make a later visualization of the particle movement almost impossible. In a typical air filter media simulation, a batch interval may last minutes, while particles move with a velocity of several meters per second through a filter material of less than a millimeter thickness, which means that at most observation points in time, no moving particles can be seen. This can be overcome by letting all particles start at **Batch Start**, or distributed over a short time interval at the beginning of the batch (**Batch Continuous**).

The selection made here has very little influence of the computed results, because in any case all particles of the same batch do not interact with each other. They cannot collide in flight, and they cannot deposit on top of each other. They can only interact with the particles deposited in previous batches.

Movement

The parameters in the **Movement** panel define the simulation variables for the particle tracking 7.



Brownian Motion and Cunningham Correction

Checking **Simulate Brownian Motion** activates the simulation of random motion in particle movement. If this option is unchecked, the simulation is run with diffusivity $D = 0$.

If **Cunningham Correction** is checked, equation (12) 8 is used to calculate C_c. Otherwise, $C_c = 0$ is used.

The **Cunningham Lambda** value is the mean free path λ used in equation (12) 8. Make sure, that you use the Cunningham Correction with the default mean free path only in air filtration. Inside of water or oil, the mean free path is much smaller, and there is no need to take the Cunningham Correction into account.

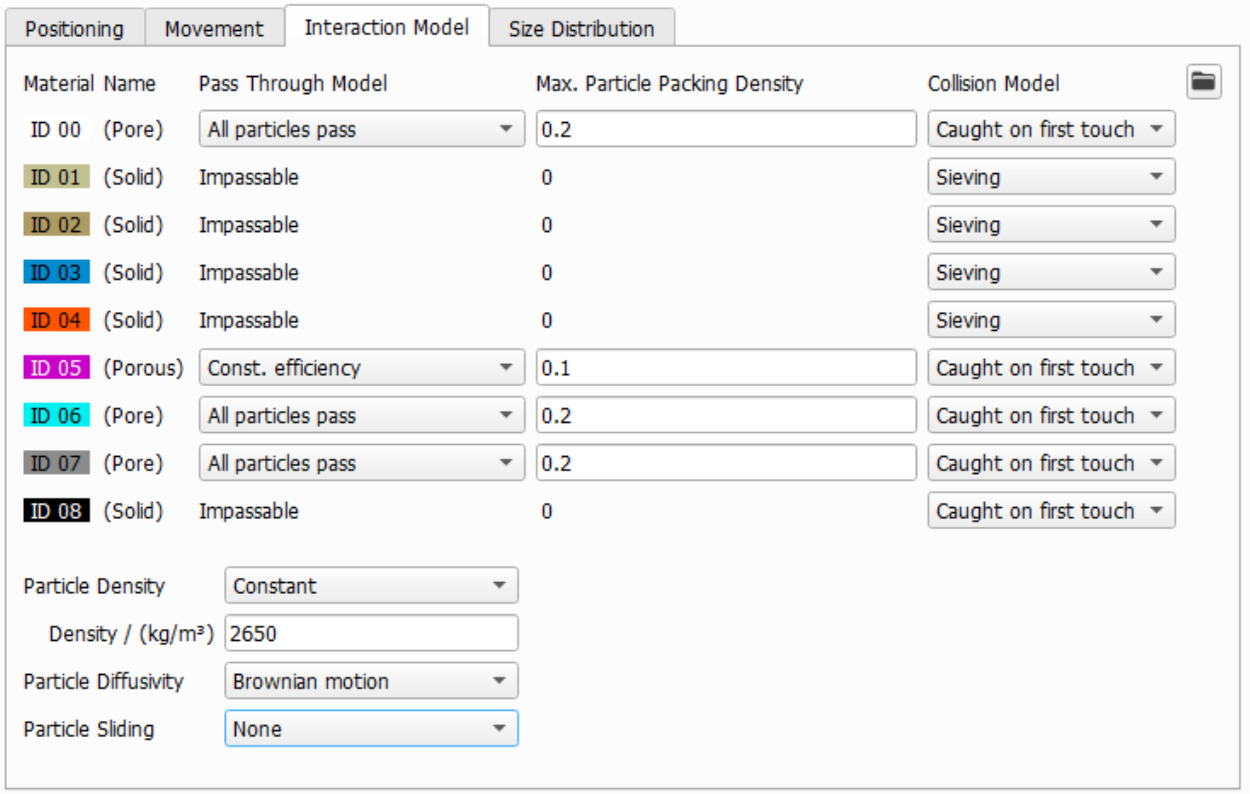
Use Particle Motion UDF

Check **Use Particle Motion UDF** to replace equations (11) 8, (12) 8, (13) 8, and (14) 8 with your own definitions. Refer to the Particle Motion UDF 239 page for details on how to modify and compile user-defined functions.

Interaction Model

For complete filter simulations, it is generally assumed that not all parts of the filter media are resolved, and that porous voxels are present in the model. Therefore, the **Pass Through Model** column is always available in the **Interaction Model** subtab.

For all materials in the model, a **Pass Through Model**, the **Max. Particle Packing Density**, and a **Collision Model** must be set.



For each porous material, select on of the Pass Through Models[11] **All Particles pass**, **Constant absorption rate**, **Clogging**, **Constant efficiency**, **Velocity-dependent efficiency**, or **Impassable**. If set to **Impassable**, the material is treated as a solid material and particles cannot enter.

Additionally, all compiled Pass Through UDFs[247] that are stored in the users UDF folder, appear here as additional choices in the pull-down menus. Additional UDF folders can be selected with the 🗀 button.

If you select one of the models **All Particles Pass**, **Const. absorption rate**, **Clogging**, **Const. efficiency**, or **Velocity-dependent efficiency**, a voxel is filled until the **Max. Particle Packing Density** is reached. Once the voxel is filled, it automatically turns impassable for further particles and the approaching particles collide at the voxel surface. Select a **Collision Model** to determine how these collisions are handled. If **Velocity-dependent efficiency** is selected, it must be defined for each particle type in the Size Distribution[188] table

In this sense, the user interface can be understood as follows: The **Pass Through Model** on the left is valid until the **Max. Particle Packing Density** is reached, and afterwards the **Collision Model** on the right applies. If the material is solid, the material is **Impassable**, thus the **Max. Particle Packing Density** is fixed to 0, and the **Collision Model** applies directly.

Know how! You can upscale the results of a fully resolved filter media simulation to determine the **Max. Particle Packing Density**. In the Report tab of a **Filter Media - Filter Lifetime** command, the Filter Clogging Analysis[106] computes parameters that can be used as input here. Use the reported **Cake Filtration** value of $f_{\max}$ as **Max.**

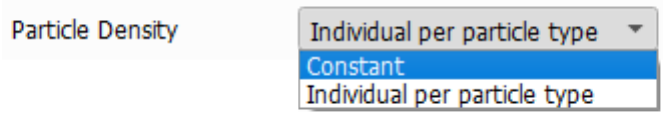
Particle Packing Density of the fluid phase and the reported **Depth Filtration** value of f_{max} as **Max. Particle Packing Density** of the porous filter material.

Know how! The **Collision Model** of the fluid phase describes the behavior of particles when they arrive at the surface of a voxel filled with dust particles. In most setups, it is safe to assume that the arriving particle is deposited somewhere in the voxel on top of the filled voxel. In that case, **Caught on First Touch** is a good choice. Only in cases, where particles may bounce off from the filter cake surface, and get transported to a significantly different position, it is necessary to change this model to the **Hamaker** model. This might be the case for large particles, with high velocities, or with specific angle of attacks of the flow.

Know how! Select **Sieving** as **Collision Model** for all solid materials describing the filter housing, to prevent particles from sticking to these materials.

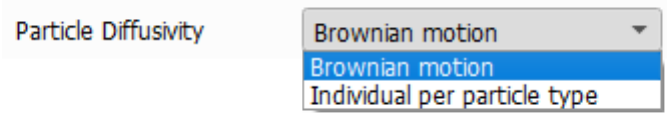
Particle Density

If **Individual per particle type** densities are chosen, a new column called **Particle Density** is added to the table under the Size Distribution tab, and an individual density can be entered for each particle type or size. Otherwise, the **Particle Density** entered here is used for all particle sizes and types.



Particle Diffusivity

If **Individual per particle type** is chosen, a new column called **Difusivity in Pore** is added to the table under the Size Distribution tab, and an individual diffusivity can be entered for each particle type or size.



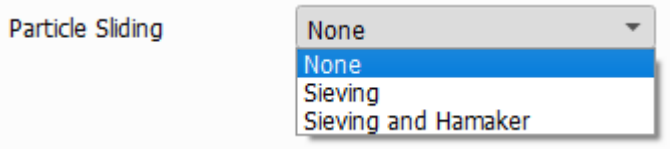
For **Brownian Motion**, the particle diffusivity is computed for each particle type according to equation (13) 8.

If **Simulate Brownian Motion** is switched off on the Movement tab, only **None** is selectable here.

If **Use Particle Motion UDF** is selected on the Movement tab, the choices made here have no effect, as the formula implemented in the UDF will be used.

Particle Sliding

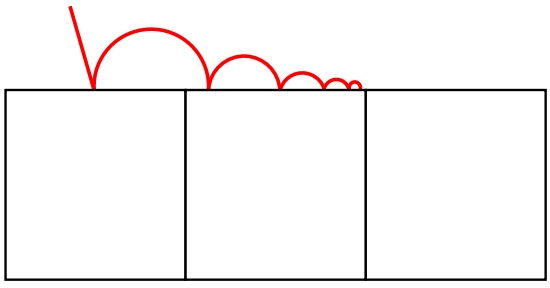
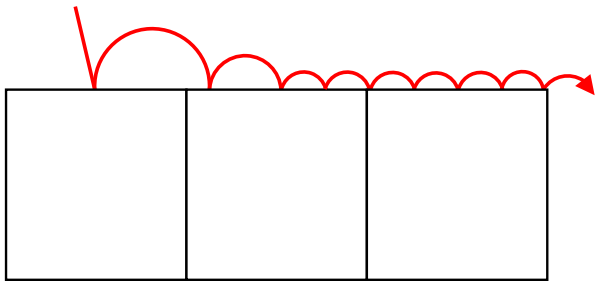
Particles that collide with the surface loose some of their energy if the restitution factor is smaller than 1. In certain pore geometries, the flow may move a particle very close to a pore surface. This causes many consecutive hits, and repeatedly applying the restitution factor will stop the particle's motion. This behavior is often undesired, and it may overestimate the filter efficiency when combining a **Sieving** collision model with a relatively low restitution value.



When **Sieving** is selected in the drop-down menu, all **Sieving** particle-wall collisions are modified, when **Sieving and Hamaker** is selected, all **Sieving** particle-wall collisions are modified and all **Hamaker** particle-wall collisions are modified. **Caught on First Touch** particle-wall collisions are not modified.

Enabling **Particle Sliding** for the **Sieving** or **Hamaker** collision models enlarges the tangential restitution factor when a particle hits a surface in a location that lies close to the last surface collision.

The first bounce on the wall is always treated the same, independent whether particle sliding is active or not: A particle arrives with a certain velocity, loses energy on impact, and bounces back again. The strength of the bounce-back is controlled with the given **Restitution**. When particle sliding is active, further impacts near the first hit are treated differently: Only the momentum perpendicular to the wall is reduced. The velocity along the wall is not reduced further, so in this direction the restitution is equal to 1. This means that particle movement along the wall is no longer slowed down. The normal restitution remains unchanged. In effect, the particle will slide along the surface.

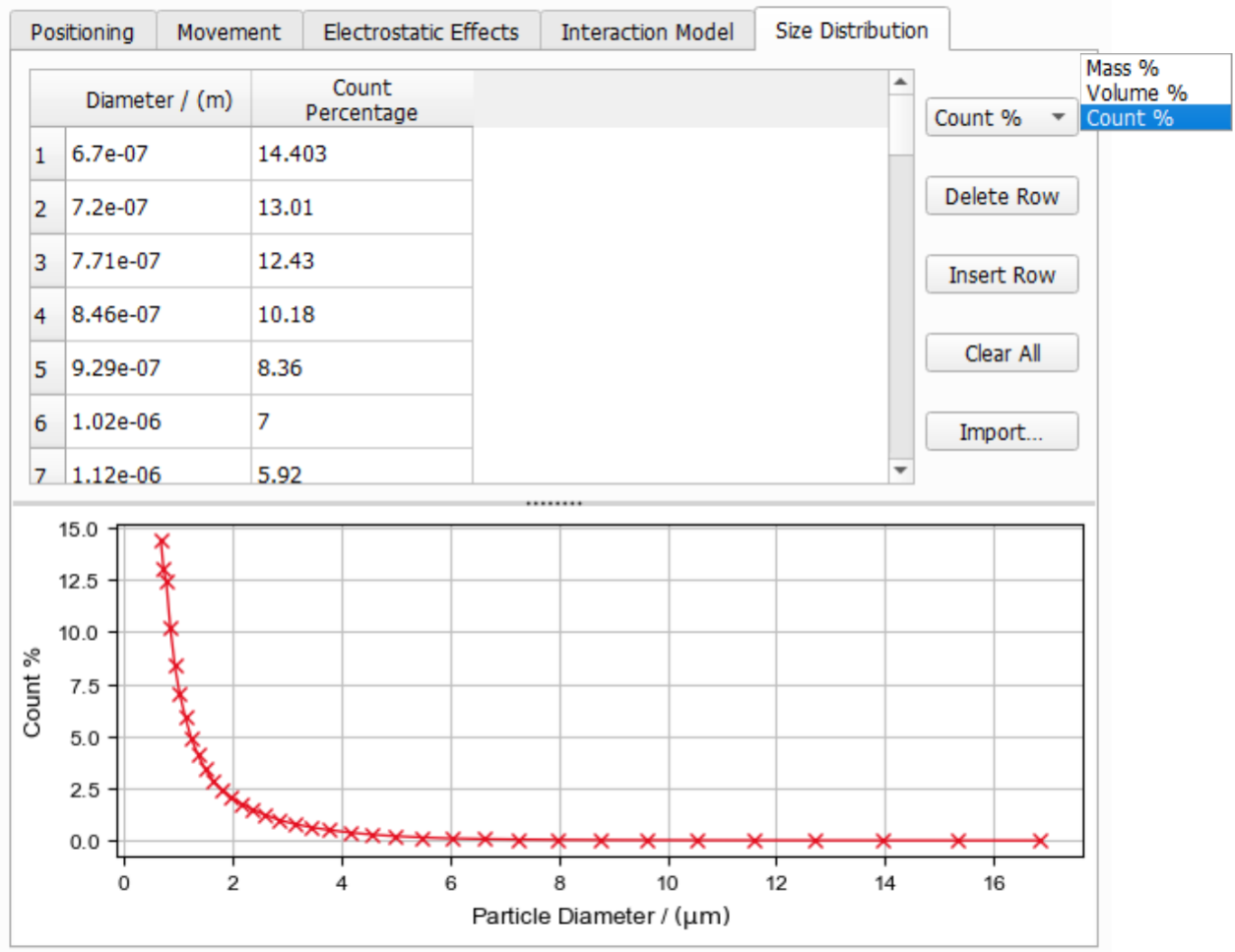
Without particle sliding:	With particle sliding
• energy lost at every collision • particle caught on surface	• energy conserved after some hits • particle moves along surface
	

In setups where particles are sieved by filter materials having pores of well-defined sizes (e.g., in meshes, nets or weaves), selecting particle sliding has a great influence on the computed filter efficiency and pressure drop and it is recommended to select this option.

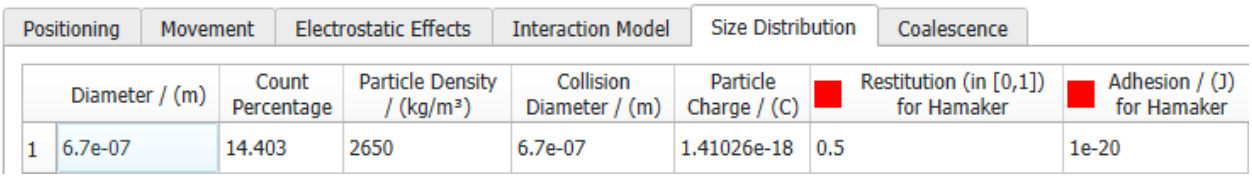
In setups where particles are mainly caught by adhesion or where the filter material consists of irregular pores of many different sizes (e.g., nonwovens), the choice of the particle sliding model has little to no effect on the simulation results.

Size Distribution

The **Size Distribution** tab contains a table with the particle size distribution. The table consists of at least two columns. The first column lists the **Diameter** of the particles. The second column contains the particle distribution. You can select if the percentages describe **Mass %**, **Volume %**, or **Count %** through the upper right pull-down menu.



Additional columns appear depending on the choices made under the Interaction Model subtab:



These columns are:

- **Restitution** for materials with Sieving collision model.
- **Restitution** and **Adhesion** for Hamaker material.

- **Particle Density** column for Individual per particle type densities.
- **Particle Charge** column for Individual per particle type charges.
- **Collision Diameter** column for Individual per particle type collision diameters.

For user-defined collision models, the number of material parameters to be entered in the table is specified in the UDF.

By clicking on a column heading, all values in the selected column can be set to the value in the first row. The entries in the table do not need to be sorted.

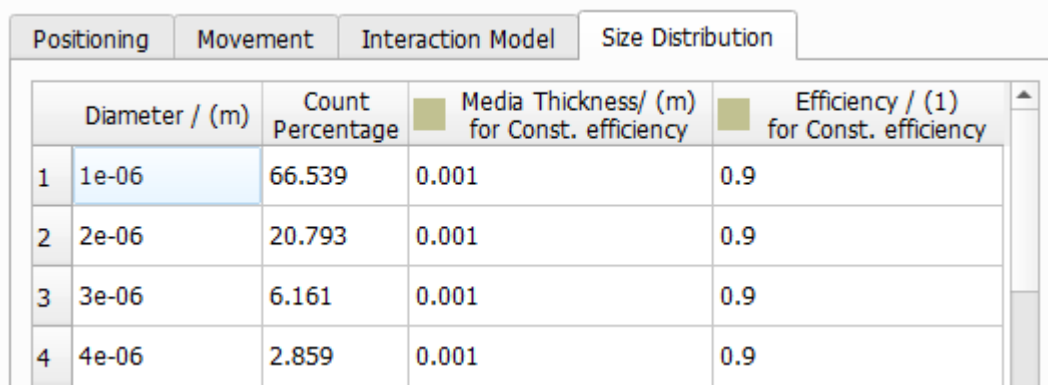
Click **Delete Row** to delete the currently selected row. Click **Insert Row** to insert a row below the currently selected row. Click **Clear All** to remove all rows.

Click **Import...** to import data from an ASCII text file. Some example particle distributions are provided with GeoDict and can be found in the installation folder (Program Files/Math2Market GmbH/GeoDict2026/FilterDict).

More convenient than importing is often to simply copy & paste entries from an Excel table into GeoDict, which is supported by the size distribution table. To copy a table from Excel, first clear the size distribution table by clicking **Clear All**. Then, select the cells in Excel and press Ctrl+C. Then, click into the upper left cell in the size distribution table and press Ctrl+V. This works also if more than two columns are present.

If the entered percentages do not add up to 100%, a warning message pops up when closing the dialog. You can select to normalize the values automatically or to return to the dialog and adjust the percentages manually.

If **Constant efficiency** or **Clogging** is selected as pass through model, a fractional filtration efficiency value and the corresponding media thickness must be entered for each particle size. For example, with the definitions in the table below, 90% of the particles are filtered when passing through 1 mm of the media.



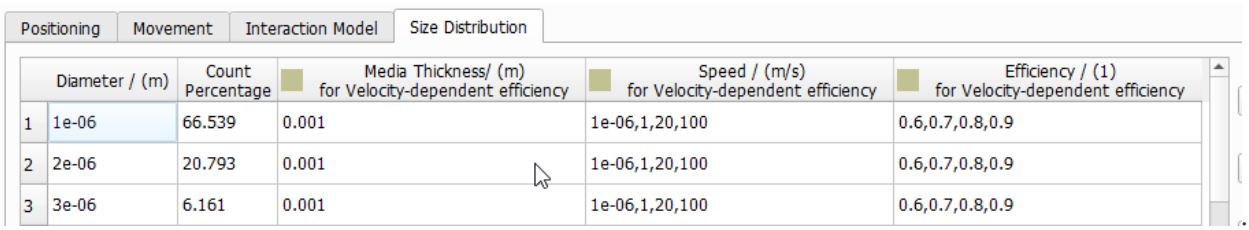
	Diameter / (m)	Count Percentage	Media Thickness/ (m) for Const. efficiency	Efficiency / (1) for Const. efficiency
1	1e-06	66.539	0.001	0.9
2	2e-06	20.793	0.001	0.9
3	3e-06	6.161	0.001	0.9
4	4e-06	2.859	0.001	0.9

Know how! When using the **Clogging** model, enter the filtration efficiency of the clean filter media as efficiency value. Whereas, when using the **Constant Efficiency** model, it is recommended to use an efficiency value that represents an average efficiency value over the depth filtration phase.

Enter the thickness of the flat sheet material or height of the filter geometry as **Media**

Thickness, if the entered efficiency value corresponds to the complete thickness of the filter.

If **Velocity-dependent efficiency** is selected as pass through model, efficiency values depend on the velocity as shown in the following screenshot. For example, with the definitions in the table below, at a particle velocity of 1 m/s, 70% of the particles are filtered when passing through 1 mm of the media.



Positioning | Movement | Interaction Model | Size Distribution

	Diameter / (m)	Count Percentage	Media Thickness/ (m) for Velocity-dependent efficiency	Speed / (m/s) for Velocity-dependent efficiency	Efficiency / (1) for Velocity-dependent efficiency
1	1e-06	66.539	0.001	1e-06,1,20,100	0.6,0.7,0.8,0.9
2	2e-06	20.793	0.001	1e-06,1,20,100	0.6,0.7,0.8,0.9
3	3e-06	6.161	0.001	1e-06,1,20,100	0.6,0.7,0.8,0.9

Solver

Flow Calculation

The names of the equations to solve the **Initial Flow PDE** and **Iterative Flow PDE** are shown for information. These entries cannot be changed directly because they depend on the choice of Creeping / Fast Flow in **Flow Motion** under the Filter Experiment tab 172.

Only the LIR solver can be used for simulations of the **Complete Filter**. The initial flow field may alternatively also be loaded from a previous FilterDict or FlowDict computation.

Analyze Geometry

If this option is chosen, a geometrical analysis at first determines whether a through path exists and removes unconnected pore components from the computational grid. This may speed up the flow computations but requires time for the geometrical analysis.

Batch Settings

The number of **Batches per Flow Field** determines how often the flow field is re-computed. Usually, the flow field should be recomputed for every batch (time interval), but if this is numerically too costly, it can be changed to re-compute the flow field only after every second or third batch. However, when more than one batch per flow field is chosen, the accuracy of the result decreases. In this case, numerical artifacts might be introduced. The most accurate results are obtained by computing one batch per flow field.

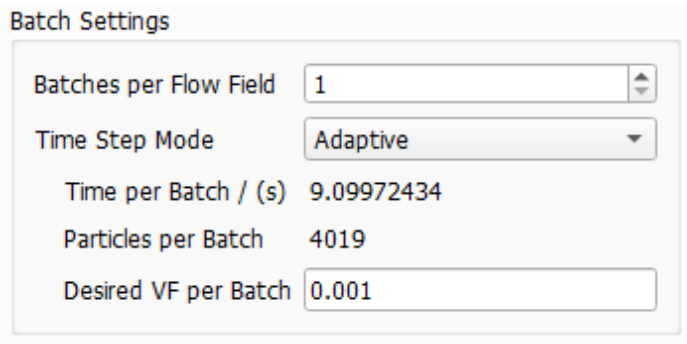
A batch of particles corresponds to a certain time interval in the experiment. The particles simulated in a batch do not interact with each other, but they do interact with the particles deposited in the previous batches. Decreasing the number of particles per batch leads to a higher accuracy, but also to longer simulation times.

The length of the time intervals per batch and the number of particles can be chosen by selecting a **Time Step Mode** from the pull-down menu:

Adaptive

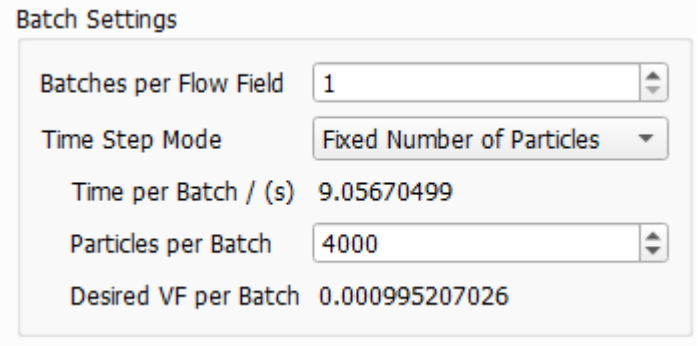
The number of particles is determined adaptively in each step. It depends on the desired volume fraction of deposited particles (**Desired VF per Batch**).

Additionally, FilterDict considers information from the previous steps like, e.g., the particle deposition zone and filter efficiency. Small **Desired VF per Batch** values correspond to shorter time steps and less particles per batch. Larger values lead to longer time steps and more particles per batch.



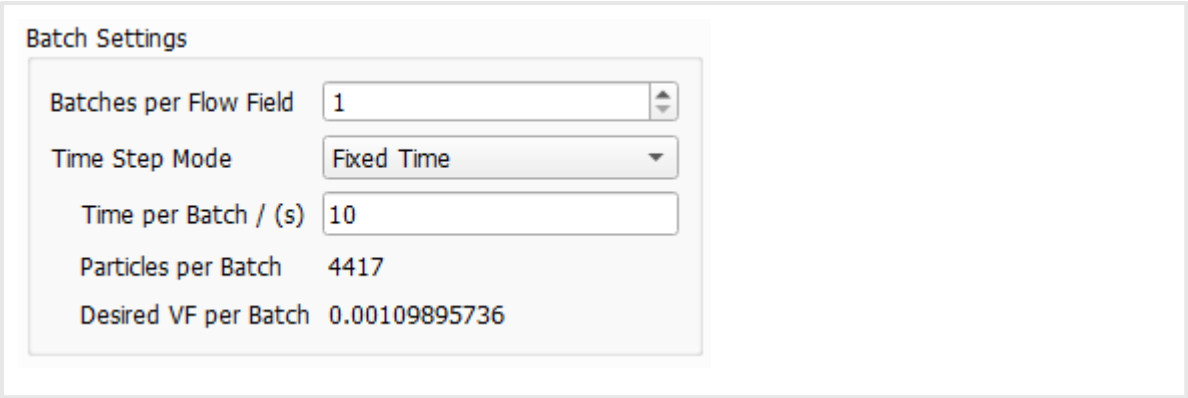
Fixed Number of Particles

Set the number of **Particles per Batch**. The time per batch is computed based on the number of particles.



Fixed Time

The number of particles per batch is determined by the **Time per Batch / (s)**. With a **Max. time reached** of 100 s chosen as simulation stopping criterion and 10 s chosen as **Time per Batch** here, the simulation stops after 10 batches of particles.



For multi pass simulations, **Fixed Number** or **Adaptive** should be preferred over **Fixed Time**. The particle concentration in the fluid changes over time. Therefore, also the number of particles could change significantly with **Fixed Time** steps and this might lead to a less stable simulation.

Parallelization

Depending on the purchased license, the simulation process can be parallelized. The **Parallelization Options** dialog opens when clicking the **Edit...** button, to choose between **Sequential**, **Parallel (Shared Memory)**, **Automatic Maximum of Threads** and **Cluster**. For details on how to set up and run parallel computations, refer to the High Performance Computations user guide

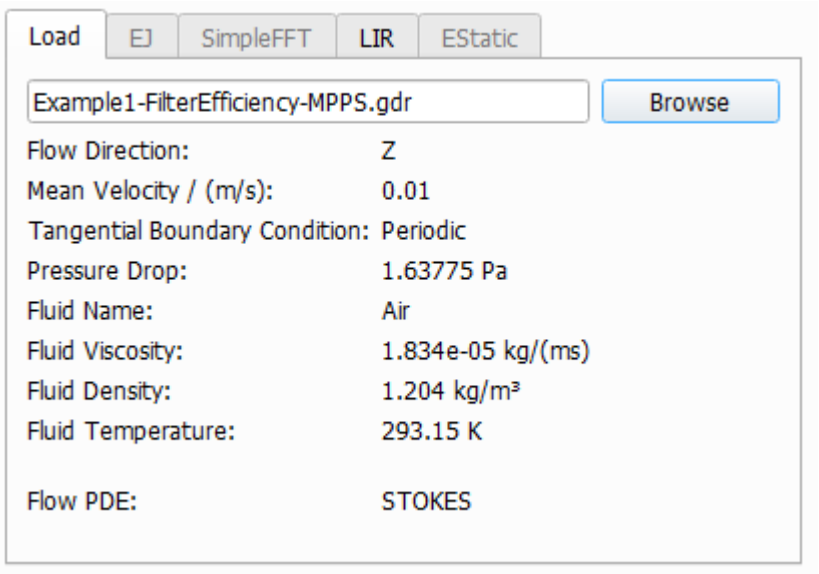
The chosen parallelization settings apply for all steps of the simulation. Per default, the **Automatic Maximum of Threads** option is used, and the exact number of threads depends on the number of cores on the current computer and the number of licensed processes.

Load

Select **Load From File** in the **Flow Calculation** panel as **Initial Flow Solver** to use a precomputed flow field. The flow field may originate from a FlowDict simulation or from a FilterDict simulation. When calculated with FlowDict, you have to make sure to:

1. Add inflow and outflow regions to the media model before running FlowDict.
2. Compute the flow in the Z-direction. You may either use the Stokes or Navier-Stokes equations depending on the flow velocity for this. Make sure that you use the same boundary conditions in FlowDict as you would use in FilterDict.
3. Set the accuracy at least one order of magnitude higher than the default in FlowDict (e.g., Error Bound = 0.001 instead of 0.01). The stopping criteria depend on global values, whereas the particle movement depends on the local flow field which is subjected to larger deviations.

Under the **Load** tab, click **Browse** and choose a result file (GDR) to open. If no flow field is selected, a warning message appears (No flow field chosen) when trying to run the simulation.



The physical properties of the fluid used in the flow simulation are entered automatically when loading the flow GDR file and appear listed under the **Load** tab.

- The **Flow direction: Z** is the main direction of the flow.
- The **Mean Velocity** (m/s) of the flow field, the used **Tangential Boundary Conditions**, and the **Pressure Drop** are taken from the loaded flow simulation. If the flow result was obtained by solving the Stokes equation, the solution is linear and will be rescaled automatically to match the Mean Velocity entered in the **Filter Experiment** tab.
- The fluid parameters **Fluid Viscosity** (kg/m s), **Fluid Density** (kg/m^3), and **Fluid Temperature** (K) are the physical values of the fluid used by the solver for the calculation of the flow field.

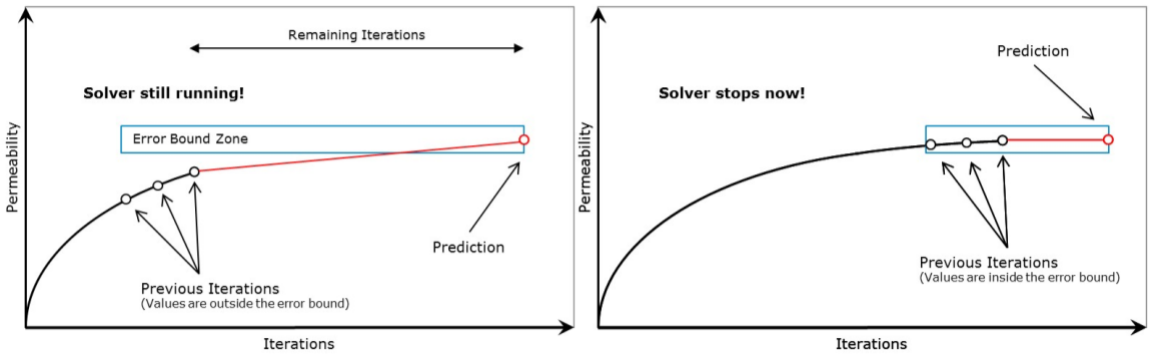
The fluid settings chosen under the Constituent Materials must be the same as the fluid that was used in the previously run flow simulation, loaded through the GDR file.

Also, the **Tangential Boundary Conditions** selected in the **Filter Experiment** tab must match with those used in the flow simulation.

LIR

Simulation Stopping Criterion: Error Bound or Tolerance

The stopping criterion **Error Bound** uses the result of previous iterations, and predicts the final solution based on linear and quadratic extrapolation. The solver stops if the relative difference between computed and predicted solution is smaller than the specified error bound. The stopping criterion recognizes oscillations in the convergence behavior and prevents premature stopping at local minima or maxima. A damped convergence curve is fit through the oscillating curve and the solver stops then regarding the damped convergence curve.



If the Krylov method (under LIR - Advanced Options) is activated, the definition of **Error Bound** is somewhat different. Here, no prediction is made, instead the continuity between neighboring cells and the conservation of mass is checked. The maximal value is normalized by the mean flow and if this value is smaller than the **Error Bound** the simulation stops.

The **Tolerance** stopping criterion looks for stagnation of the method when the process becomes stationary, i.e., the improvement in the pressure drop value becomes extremely small from iteration to iteration.

In each iteration, the solver checks for the current computed value x_c against the values x_{c-i} of the last 100 iterations if there are any changes.

$$\max_{i\in[1,100]} \frac{x_c - x_{c-i}}{x_{c-i}} < \text{tolerance} \tag{54}$$

The computation is stopped if the maximal relative change is smaller than the value entered for **Tolerance**. When there is doubt about the quality of the solution, decrease the **Tolerance** value by a factor of ten. The drawback of this criterion is that the solver sometimes might stop too early in case of slow convergence rate or at local extrema of oscillatory convergence curves.

Be aware that the default accuracy in FilterDict is one order of magnitude higher than in FlowDict. The stopping criteria depend on global values, whereas the particle movement depends on the local flow field. It is therefore necessary to make sure that not only the overall pressure drop is computed up to the given accuracy, but that also the local values are sufficiently accurate. This typically achieved by setting a higher accuracy as stopping criterion.

Simulation Stopping Criterion: Max. Iterations and Max. Run Time

Use the **Maximum Iterations** value or the **Maximum Run Time (h)** stopping criteria causes the solver to stop if the maximal number of iterations and/or the maximal runtime (in hours) is exceeded.

When the solver stops because one of these criteria has been reached, no guarantee on

the quality of solution can be given. In this case, a warning is printed into the report. The following possibilities might help:

- Check the corresponding .log file to see how large the residual values and the pressure drop values are. If the pressure drop values are in reasonable range and they do not have changed during the last iterations, you may decide to use the current result.
- Double-check the structure and parameter values. Unphysical parameters or too rough resolution of the structure (leading, e.g., to artificial unconnected components) can cause an iterative solver to fail.

Advanced Options: Write Compressed Volume Field

The LIR solver uses a very memory efficient adaptive grid structure for the simulations.

If the option **Write Compressed Volume Fields** is checked, then the adaptive grid is used as compression method for writing out *.vap files. This option allows to save 80-90% space on hard drive. The runtime for writing *.vap files is also reduced significantly. But the runtime for loading and uncompressing of compressed *.vap is increased by the amount of runtime that was saved for writing out compressed *.vap files.

If the option **Write Compressed Volume Fields** is not checked, then a usual regular grid is used for writing out *.vap files.

Advanced Options: Optimization Options

The **Multigrid Method** (see, e.g., Wesseling, 2004) was introduced to speed-up the computation and reduce the runtime significantly. The main idea of **Multigrid** is the usage of multiple coarser adaptive grids to speed up convergence behavior but requires only little more memory.

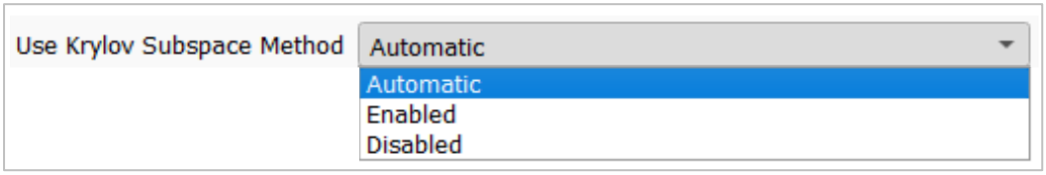
The method is available to solve the Stokes and Stokes–Brinkman equations as well as for solving mechanics, diffusion, thermal, and electrical conduction and is enabled by default.

Another speed-up option to accelerate the convergence behavior of the LIR solver is called **Krylov Subspace Method**.

The runtime of the LIR solver depends on many different properties of the structure and the simulation parameters. The BiCGStab algorithm is used, which can reduce the runtime for challenging simulation very drastically.

Note! Using the BiCGStab method approximately doubles the amount of RAM needed for the computation.

Unfortunately, the Krylov method is not always faster than a simulation without the Krylov method and therefore we introduced an **Automatic** mode which uses some heuristics to choose the most efficient method based on structure, material parameters, and boundary conditions automatically. Of course, it is possible to explicitly enable (**Enabled**) or disable (**Disabled**) the method.

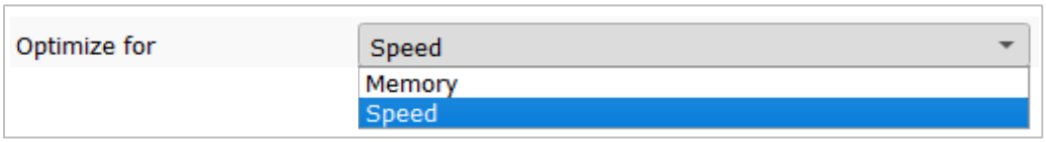


Note! In case that the Krylov subspace method (BICGStab) is used, the **Relaxation** may also be adapted automatically.

Depending on the material parameters and geometry of the structure, the underlying mathematical problem can vary in complexity, thus influencing the behavior of the solver. The more complex the problem is, the more stable the solver settings should be.

With the **Relaxation** number, the solver is adjusted from **Stable** (which results in higher number of iterations, slower time stepping, and longer solver run times), to **Fast**, which makes the solver run less iterations but implies the risk that the solver does not converge. The **Relaxation** is a parameter of the SOR method and must be between 0 and 2 to ensure convergence. For relaxation values smaller than one (<1.0), the simulation is more stable. For relaxation values larger than one (>1.0), the simulation converges faster.

The **LIR** solver can **Optimize for Speed** or **Memory**.



- If **Speed** is chosen, the solver constructs additional optimization structures. The runtime is decreased by up to 30% but requires up to 50% more memory compared to the other option.
- If **Memory** is chosen, the runtime is increased by up to 40% but the solver requires up to 50% less memory.

Advanced Options: Grid Options

The **Grid Type** decides what kind of tree structure is used for the simulation.

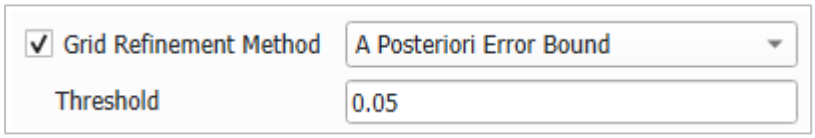
The default option is **LIR-Tree** and should always be used. The solver uses an adaptive grid structure called LIR-tree and needs up to 10 times less runtime and memory compared to the **Regular Grid** option.

The solver can analyze the result field during the computation and improves the adaptive grid in places where more accuracy is needed. The LIR solver splits cells where a high gradient occurs.

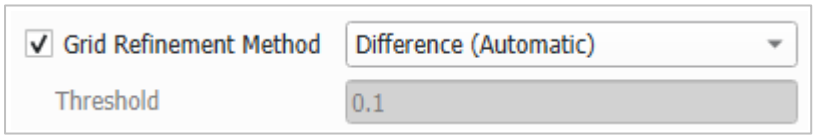
The solver can analyze the computed field and refines the adaptive grid during the computation at locations where more accuracy is needed. The LIR solver splits cells where a high gradients occur when the **Grid Refinement Method** is enabled. In this case, the additional parameters **Number of Grid Refinements**, **Allow Sub-Voxel Resolution**, and **Allow Grid Re-Coarsening** become available.

Select one of the three available refinement methods from the drop-down menu.

When **A Posteriori Error Bound** is selected, the solver targets the specified accuracy **Threshold**. While the **Error Bound** (set as Simulation Stopping Criterion) determines the relative error in the solution of the linear system, **A Posteriori Error Bound** refers to the relative error estimated by comparing high-order and low-order discrete solutions. The accuracy **Threshold** refers to the relative error to the analytical solution and the value must be between 0.0 and 1.0.

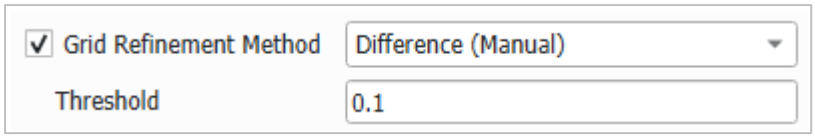


Choosing **Difference (Automatic)** leads to computational cells being split when the difference in values between neighboring cells exceeds a certain **Threshold**. The solver automatically chooses all internal parameters based on the structure and simulation settings. For this option, cells are split where the current gradient is greater than the **Threshold** multiplied by the maximum gradient, where the threshold is determined automatically.

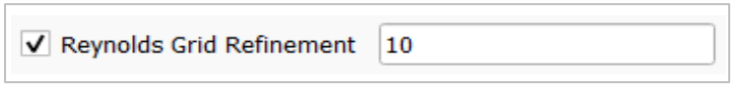


Selecting **Difference (Manual)** also causes the computational cells to be split when the difference in values between neighboring cells exceeds the specified **Threshold**. You can specify all parameters (Threshold and Number of Grid Refinements) for the grid refinement manually. For this option, cells are split where the current gradient is greater than the **Threshold** multiplied by the maximum gradient. The **Threshold** value

must be between 0.0 and 1.0. The recommended value range is between 0.05 and 0.1.



Activate **Reynolds Grid Refinement** to use the local Reynolds number (Re) for adaptive grid refinement.



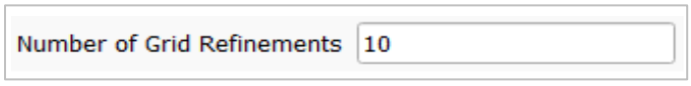
A grid cell is split if the local Reynolds number, i.e., the ratio between inertial and viscous forces exceeds the given threshold. The Reynolds number is defined as:

$$\mathrm{Re} = \frac{\rho \mathrm{u} \mathrm{L}}{\mu} \quad (55)$$

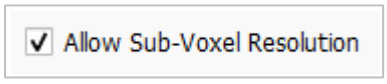
Reynolds Number

where ρ is the density of the fluid (SI units: kg/m^3), u is the velocity of the fluid (m/s), μ is the dynamic viscosity of the fluid (Pa·s or N·s/m^2 or kg/m·s), L is a characteristic length (m). For the local Reynolds number, the size of the grid cell is used for the characteristic length.

The **Number of Grid Refinements** controls the maximum number of grid refinements that the solver can perform during the simulation. The value should be set between 0 and 10, where a value of 0 means that no grid refinements will be made. Grid refinements may increase the number of iterations, runtime, and memory requirements.



When **Allow Sub-Voxel Resolution** is enabled, the solver is allowed to split computational cells to sizes smaller than the voxel length.



This feature is beneficial for low-porosity structures for which the pore throats require finer resolution or for modeling fast Navier-Stokes flows with strong vortices.

Enabling this feature may increase the number of iterations, runtime, and memory requirements.

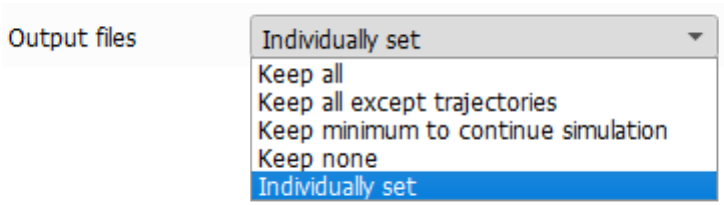
Check **Allow Grid Re-Coarsening** to allow the solver to automatically revert the grid refinement and coarsen the computational cells.

This means, grid refinement is done temporarily. Afterwards, the cells are merged as soon as accuracy is not needed anymore. This reduces the memory and runtime

requirements.

Output

First, select which **Output files** should be stored. You may select one of the predefined settings **Keep all**, **Keep all except trajectories**, **Keep minimum to continue simulation**, **Keep none**, or **Individually set**.



All these files are needed only for visualization of the results. The choices made here do not influence the filter lifetime results contained in the GDR (result) file but may disable some of the visualization options which require to load additional data.

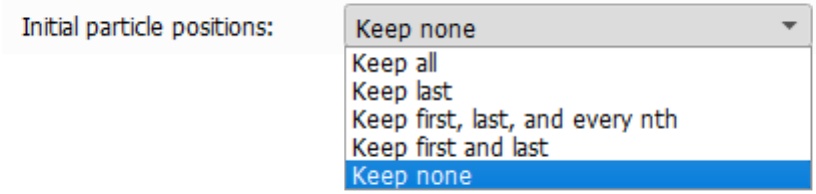
When **Keep all** is checked, all intermediate files written and read by the flow solver, particle tracker and electrostatic solver are kept. If the filter model is large, and/or the simulation contains many batches, a large amount of potentially unnecessary data is stored in that way.

Check **Keep all except trajectories** if you want to keep all data files except of the particle trajectories. Trajectory files are needed solely for visualization of the particle movement and may become very large if many particles are tracked.

Check **Keep minimum to continue simulation** to save only the files need to continue from an interrupted simulation.

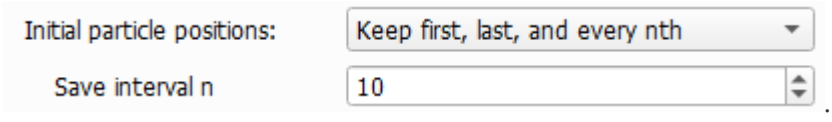
Check **Keep none** if you want to reduce the amount of hard disk space needed as far as possible.

If you select **Individually set**, you may decide in detail which files to keep for visualization. For each file type, choose between **Keep all**, **Keep last**, **Keep first, last and every nth**, **Keep first and last** and **Keep none**.



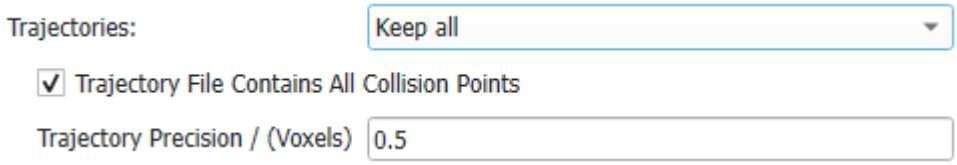
Decide, for which batch this file type is stored: every batch, only the first and/or last batch, or set an interval such that this file type is stored for every nth batch.

In that case, set the **Save interval n**:



To visualize the movement of particles, you need to keep the **Trajectories**. Be aware, that

these files may become very large if many particles are simulated.



You can influence the file size and the precision of the stored trajectories (e.g., the number of points on the particle paths saved in the file) by two additional parameters:

- Check **Trajectory File Contains All Collision Points** to add all points where a particle touches the surface of a solid object to the trajectory file.
- The value entered as **Trajectory Precision** determines how accurately the trajectories are stored. The given value determines the traveling distance between two points stored in the trajectory file.

Results

Click **OK** to input the entered parameters, and then click **Run** in the FilterDict section to start the Complete Filter - Filter Lifetime command. The results are immediately shown in the opening Result Viewer after the simulation is finished.

Report

Stopping Criterion

At the top of the **Report**, the stopping criteria reached by the simulation is shown. Here, it is also reported if the simulation was canceled (**Stopped by user**). If you open the result file while the simulation is still running, the first line reads **Simulation unfinished**.

Tables

The available tables are identical to the filter element result tables 163.

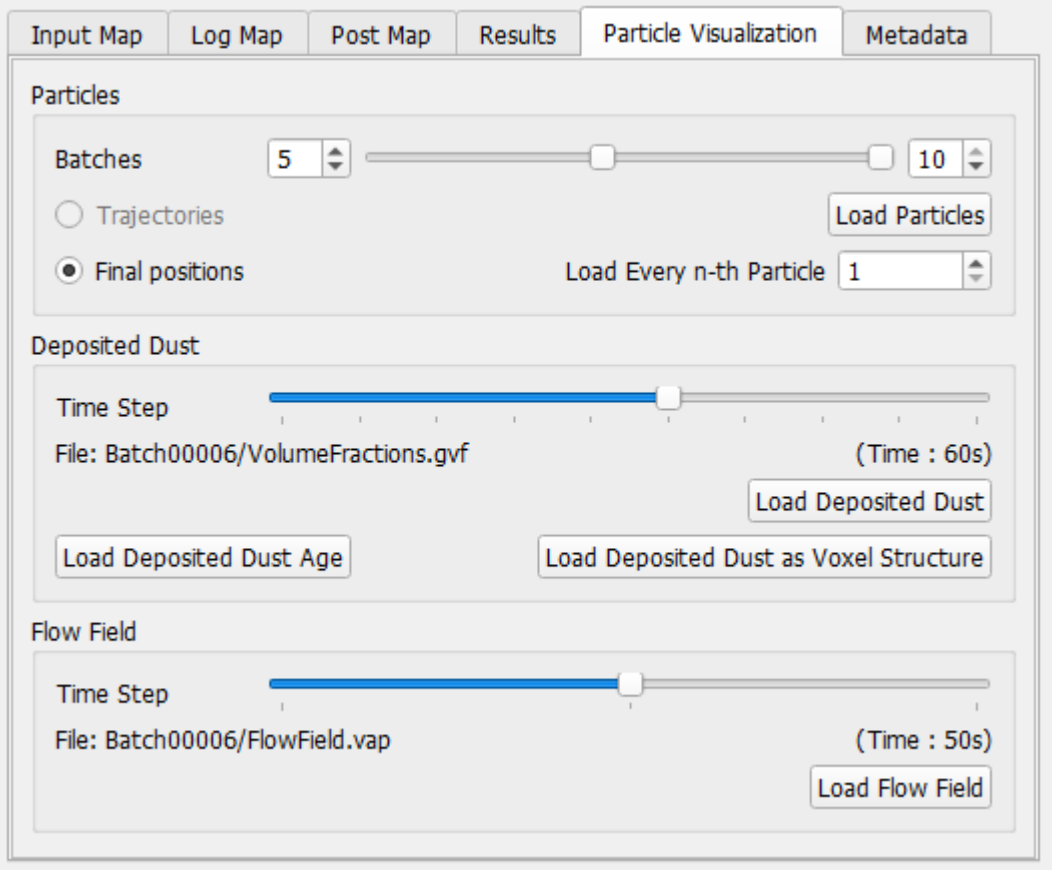
Plots

The available plots are almost identical to the filter media result plots 110.

Because of the arbitrary positions of inflow and outflow, it does not make sense to plot average values per z-layer, so the three **Geometric Analysis** plots **Particle intrusion analysis**, **Layered deposited volume over batches**, and **Layered pressure over batches** are not available here.

Particle Visualization

To visualize the initial and final positions of particles, track their trajectories, and study the flow of the fluid through the filter structure, select the **Particle Visualization** tab.



Particles

In the **Particles** panel, select one or several succeeding batches with the sliders. Select **Trajectories** to animate the particle motion from inflow to outflow or deposition place. If a large number of particles was simulated, loading all the trajectories at once is memory consuming and may not be possible. Loading only the **Final positions** does not allow to animate the particle motion, but is less memory consuming. Memory consumption may also be reduced by visualizing only a fraction of the simulated particles per batch by loading only **Every n-th Particle**.

Clicking **Load Particles** loads the selected trajectories or final particle positions and displays the particles in the viewport.

When visualizing results of filtration simulations on the complete filter scale, it is necessary to enlarge the particles Diameter Factor for better visibility, because particle diameters are small in comparison to the structure model and therefore difficult to depict when shown in their original size.

Because of the need to enlarge particles and because of the fact that those particles show only a representative particle when using **Multiplicity**, the build-up of a filter cake cannot be visualized properly on this scale by visualizing every single particle. Also, the total amount of particles simulated would make this task impossible or very demanding for the computer's graphic card. Therefore, it is recommended to use the **Deposited Dust** result fields to visualize the development of the filter cake, as explained below.

Deposited Dust

Use the slider to select the **Time Step**. There are three options to visualize the deposited dust from the simulation:

- With **Load Deposited Dust**, a result field (*.gvf) is loaded that contains for every voxel a single double value in the range [0,1], describing the volume fraction of the voxel filled with dust particles.
- With **Load Deposited Dust as Voxel Structure**, the volume field is converted to a voxel structure.
- With **Load Deposited Dust Age**, a result field is loaded that contains for every voxel a single integer value describing how many batches ago this particle was deposited. Voxels filed in the last batch contain the value zero, voxels filled in the previous batch the value 1, and so on...

Clicking on the appropriate button loads the data and displays the deposited dust in the viewport 117.

Flow Field

Use the sliders to select the corresponding time step. Click **Load Flow Field** to load the result and refer to the Visualization user guide for details.

1.4.2 Filter Flow Experiment

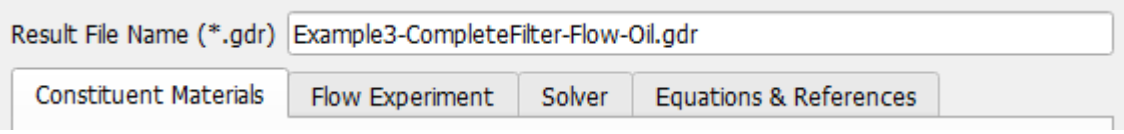
The **Filter Flow Experiment** command allows to simulate the flow through a complete filter. The fluid and the filter material is assumed to be clean. It carries no dust particles and no dust has been deposited in the filter before the flow experiment.

The Filter Flow Experiment command for complete filter models

- Options 203
  How to set the input parameters for the Filter Flow Experiment command.
- Results 215
  How to understand the computed results.
- Flow Visualization 215
  How to visualize the computed results.

Options

Select **Filter Flow Experiment** from the pull-down menu and click the **Option**'s **Edit...** button to open a dialog to enter or edit the simulation options. Choose a **Result File Name** for the result file (*.gdr) fitting to your current project.

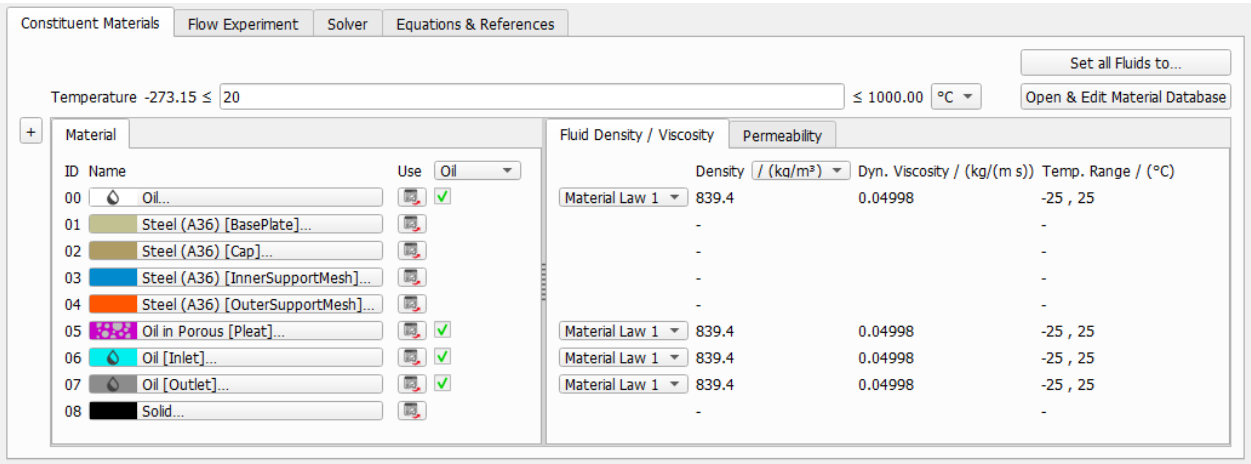


The parameters are organized under the tabs: **Constituent Materials**, **Flow Experiment**, and **Solver**. The **Equations & References** tab gives a summary of the used equations and symbols which are explained in more detail in this user guide.

Tabs of the Complete Filter Flow Experiment dialog

- Constituent Materials 203
  Define fluid type and temperature. Set material IDs and parameters for porous layers.
- Flow Experiment 205
  Define inlet and outlet material IDs, and set the flow boundary conditions.
- Solver 206
  Select the flow solver, set the stopping criteria, and define the computational resources.

Constituent Materials



Temperature

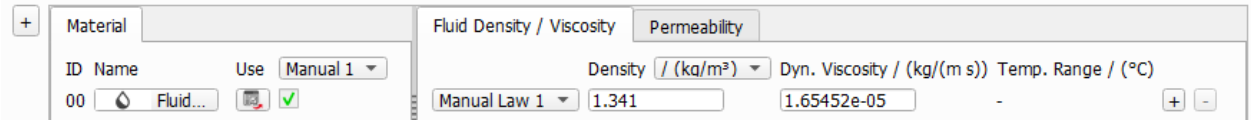
The **Temperature** for the filtration process is selectable in Kelvin (K), Celsius (°C), and Fahrenheit (°F). The chosen temperature may change the density and viscosity values of the fluid during the simulation, if those are defined to be temperature-dependent in the material database. Furthermore, if the Brownian motion of the particles is included in the simulation, it determines the strength of the Brownian motion through equation (13) 8.

Fluid Density and Viscosity

FilterDict uses single-phase flow simulations. It is not possible to run a flow simulation if two fluids are present in the structure. If multiple fluids are present in the structure, you may click **Set all Fluids to...** and select one fluid for the simulation.

The selected fluid is also displayed in the drop-down menu in the **Material** subtab behind **Use**. The density and the dynamic viscosity of this fluid are shown under the **Fluid Density/Viscosity** subtab.

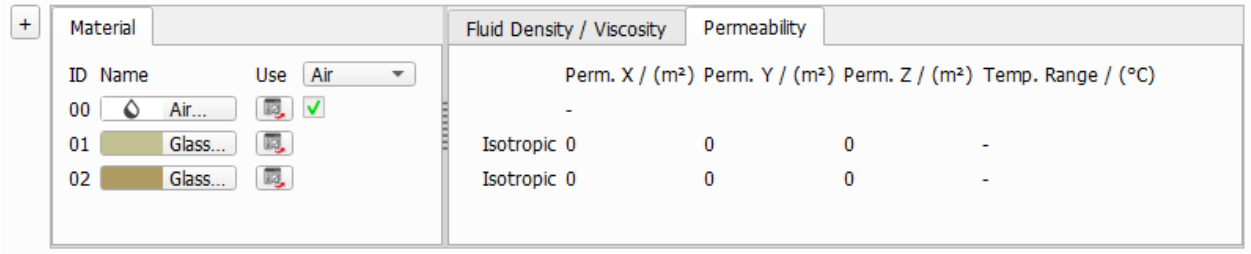
If a fluid from the GeoDict Material Database is used, the values for **Density** and **Dynamic Viscosity** are taken from the database and may depend on the given **Temperature** value. Select **Manual** as fluid material to enter parameter values manually.



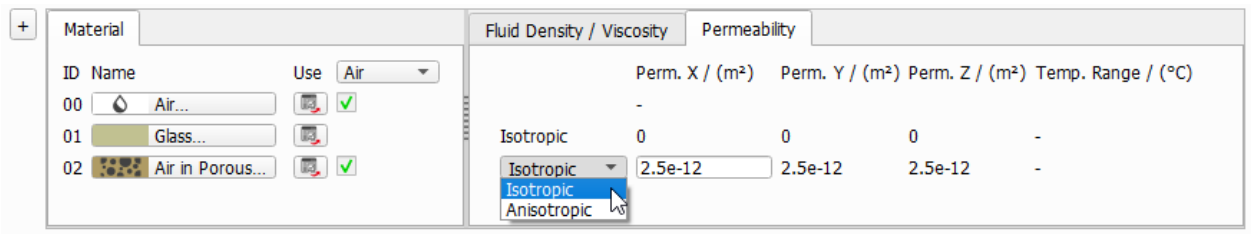
To add materials to the database, use the **Open & Edit Material Database** button. More information on editing, expanding, and using the **GeoDict Material Database** is available in the Material Database user guide.

Filter Media Permeability

Under the **Permeability** subtab, enter the permeability of porous materials present in the structure. As long as there are only fluid and solid materials present in the structure, no parameters are needed here:

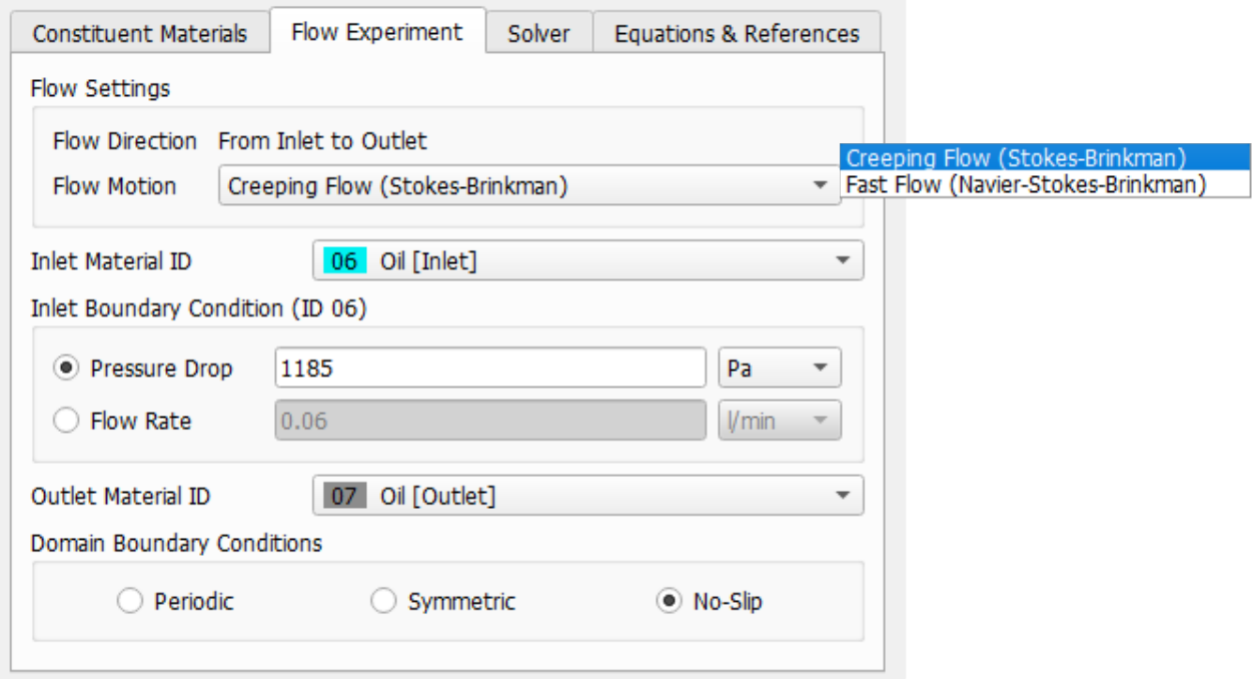


For porous materials, choose if the material is **Isotropic** or **Anisotropic**. In the isotropic case, the permeability is the same in all space directions. In the anisotropic case, enter a different permeability for each spacial direction.



You can select and change a material by clicking on the material button.

Flow Experiment



Flow Settings

The **Flow Direction** is fixed to **From Inlet to Outlet**.

The **Flow Motion** regime can be chosen as **Creeping flow (Stokes)** or **Fast flow (Navier-Stokes)** based on the flow velocity and the Reynolds number.

Inlet

Define the inlet of the flow by selecting the **Inlet Material ID**. For this, it is necessary that the inlet is marked with a different material ID. This can be easily done by selecting a voxel on the inlet area and using the **Flood Fill Component Plane** feature of the **Voxel Selection** tool. In the Modeling of Complete Filter and Simulation of Flow and Particle Filtration tutorial, this workflow is described in detail. For an automatic recognition of Inlet phase it is advised to provide the information “Inlet” to the selected material ID.

The **Inlet Boundary Conditions** can be either set by prescribing the **Pressure Drop** or the **Flow Rate.**

Outlet

Define the outlet of the flow by selecting the **Outlet Material ID**. Analogous to the **Inlet Material ID**, it is necessary that the outlet is marked with a different material ID and for an automatic recognition provide the information “Outlet” to the selected material ID.

Domain Boundary Conditions

The 3D model of the complete filter typically contains also the filter housing. In this case, flow will only occur inside of the filter housing and the domain boundary conditions do not apply. However, if the filter is symmetric, it is possible to reduce the computational cost by modelling only one half or one quarter of the filter. In these cases, the flow area will be open to one or two sides of the domain and one should select **Symmetric** boundary conditions for the flow at the domain boundary.

No-Slip means that the structure is encased in a solid wall at the domain boundary.

Solver

The **Complete Filter - Filter Flow Experiment** command always uses the LIR solver to compute the flow.

Simulation Stopping Criterion

The equations are solved by an iterative approach.

Iterative Solver

The basic idea of an iterative method is to

1. Start with some initial guess for the unknown values
2. Improve the current values in each iterative step
3. Repeat the iterative process until one of the stopping criteria is reached.

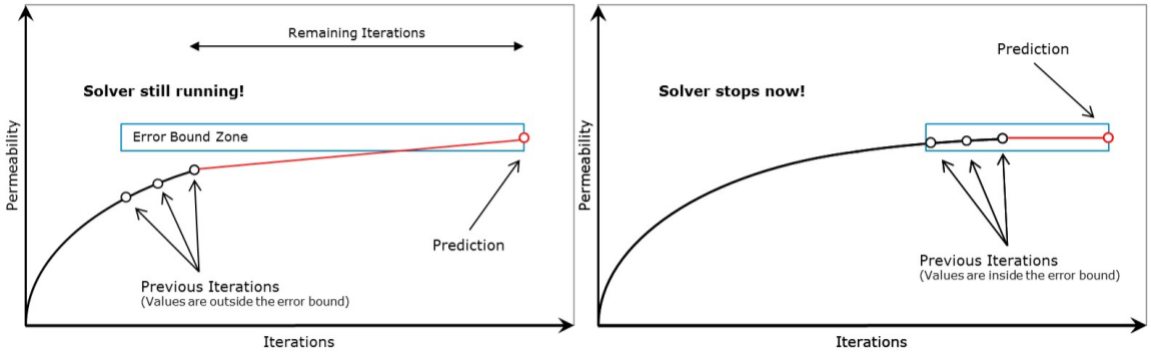
The iterative process is controlled by setting the values and activation for **Error Bound, Tolerance**, **Maximal Iterations**, and **Maximal Run Time (h**). The stopping criteria apply for each computational direction individually.

If multiple stopping criteria are selected, the first criterion that is reached causes the solver to stop.

The stopping criterion that has been reached can be viewed in the Result Viewer of the GeoDict result file (*.gdr) under the **Results Report** tab.

The stopping criterion **Error Bound** uses the result of previous iterations, and predicts the final solution based on linear and quadratic extrapolation. The solver stops if the relative difference between computed and predicted solution is smaller than the specified error bound. The stopping criterion recognizes oscillations in the convergence behavior and prevents premature stopping at local minima or maxima. A damped convergence curve is fit through the oscillating curve and the solver stops then

regarding the damped convergence curve.



If the Krylov method (under LIR - Advanced Options) is activated, the definition of **Error Bound** is somewhat different. Here, no prediction is made, instead the continuity between neighboring cells and the conservation of mass is checked. The maximal value is normalized by the mean flow and if this value is smaller than the **Error Bound** the simulation stops.

The **Tolerance** stopping criterion looks for stagnation of the method when the process becomes stationary, i.e., the improvement in the pressure drop value becomes extremely small from iteration to iteration.

In each iteration, the solver checks for the current computed value x_c against the values x_{c-i} of the last 100 iterations if there are any changes.

$$\max_{i\in[1,100]} \frac{x_c - x_{c-i}}{x_{c-i}} < \text{tolerance} \tag{56}$$

The computation is stopped if the maximal relative change is smaller than the value entered for **Tolerance**. When there is doubt about the quality of the solution, decrease the **Tolerance** value by a factor of ten. The drawback of this criterion is that the solver sometimes might stop too early in case of slow convergence rate or at local extrema of oscillatory convergence curves.

Use the **Maximum Iterations** value or the **Maximum Run Time (h)** stopping criteria causes the solver to stop if the maximal number of iterations and/or the maximal runtime (in hours) is exceeded.

When the solver stops because one of these criteria has been reached, no guarantee on the quality of solution can be given. In this case, a warning is printed into the report. The following possibilities might help:

- Check the corresponding .log file to see how large the residual values and the pressure drop values are. If the pressure drop values are in reasonable range and they do not have changed during the last iterations, you may decide to use the current result.
- Double-check the structure and parameter values. Unphysical parameters or too

rough resolution of the structure (leading, e.g., to artificial unconnected components) can cause an iterative solver to fail.

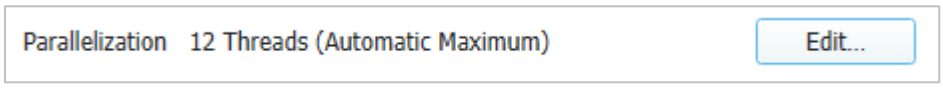
Restart Save Interval

The calculations run by the solvers can be restarted from saved intermediate result files and the interval between auto-saves can be configured from the value entered in **Restart Save Interval (h)**.

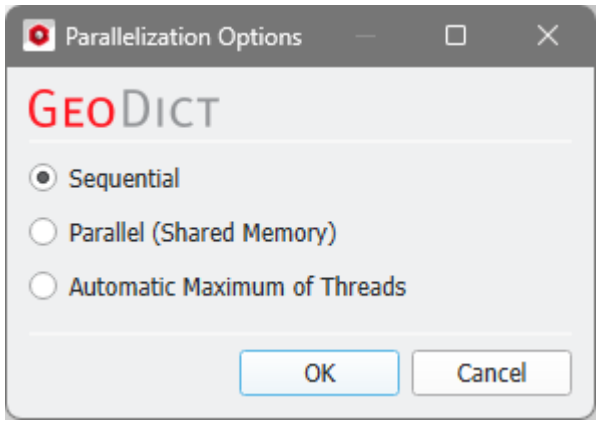
Parallelization

Control how many threads are used for the computation. Parallelization is possible if your license and hardware allow it.

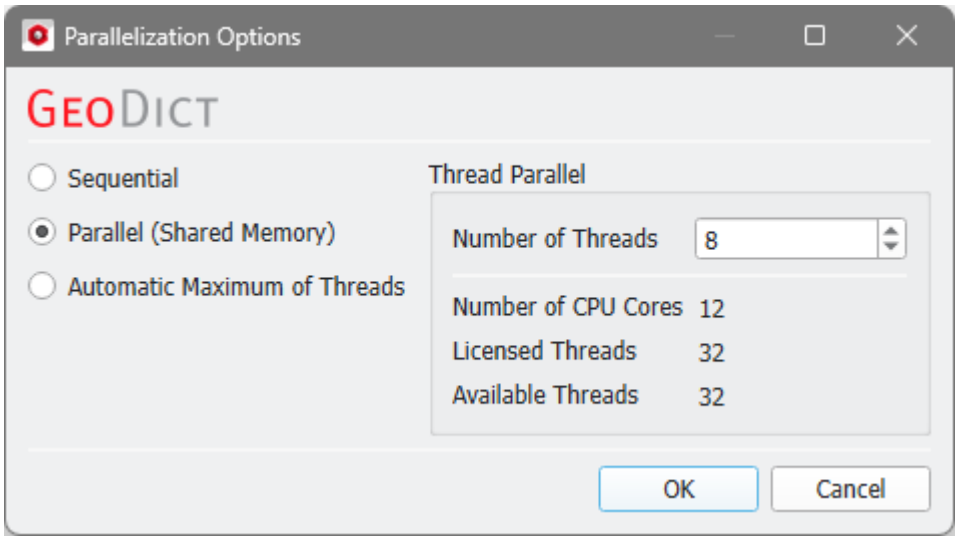
The **Parallelization Options** dialog opens when clicking the **Edit** button and you can choose between **Sequential**, **Parallel (Shared Memory)**, or **Automatic Maximum of Threads**.



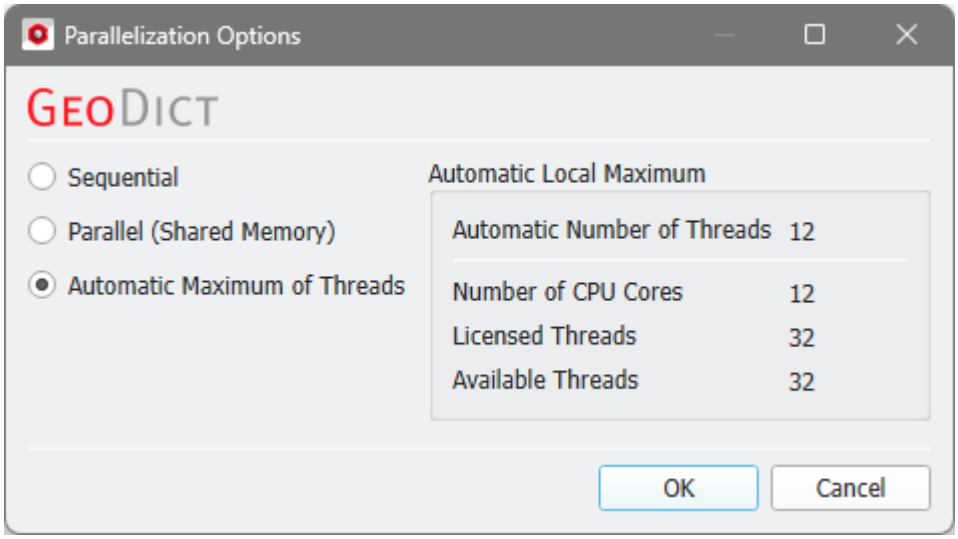
Selecting **Sequential** will not apply parallelization and only one thread is used for the computation.



When **Parallel (Shared Memory)** is selected, the **Number of Threads** can be entered. Below, the **Number of CPU Cores** that the current machine has, the maximum number of **Licensed Threads** and the number of those licensed threads that are available (**Available Threads**) are shown in the dialog. Of course, the maximal number of parallel processes you can use, is the smallest of those three numbers.



If **Automatic Maximum of Threads** is selected, the number of parallel processes is automatically selected for optimal speed, based on the CPU cores and licensed parallel processes.



The **Automatic Local Maximum** of processes is automatically selected, which is the minimum of **Number of CPU Cores**, **Licensed Threads**, and **Available Threads**.

Restart From .gdr File

In some situations, it may be useful to re-use previously computed results and save a great amount of time by restarting the process from an existing GeoDict result file (*.gdr). Typical examples would be

- non-sufficient accuracy of some computation, when it is suspected that more iterations may improve the quality of the result,
- trying another boundary condition (e.g., Dirichlet instead of Periodic or other way around),

- continuing an iteration process,
- or using similar parameter values for the computation.

To restart and initialize the solver from a result file (*.gdr), check **Restart from .gdr File** and enter the name of a file, or **Browse** for it in the project folder.

When using **Restart From .gdr File** only the volume field (*.vap file) associated to the loaded *.gdr file will be used to continue the simulation.

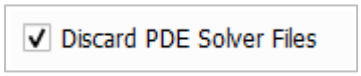
Important! The *.vap file is saved in the result folder of a previous run. Make sure that this file was not deleted from the results folder. If the *.vap file is not available the restart option cannot be used.

No other parameters from the options dialog will be loaded from the *.gdr file, but it is possible to load them by clicking **Load Parameters** in the Result Viewer. This way the simulation can be continued while using the same settings as in the initial simulation.

Important! The structure from the run of which you want to use the result file must be loaded into GeoDict. If this is not the case, an error message is displayed.

Discard PDE Solver Files

Checking **Discard PDE Solver Files** causes the deletion of all intermediate computation files, such as log files and flow field files.



Only the content of the final GeoDict result file (*.gdr) is stored.

While having the benefit of saving hard disk storage place, discarding intermediate solver files has also the effect of disabling the 3D visualization of the results.

Advanced Options: Analyze Geometry

Checking **Analyze Geometry** performs an analysis of the geometry before the solver computations.

If no percolation path through the structure is found, the partial differential equation

does not need to be solved, and a pressure drop of 0.0 is directly reported as the solution.

Furthermore, the solver operates on the whole structure, regardless of which parts are connected or unconnected. However, unconnected components are not transport relevant, thus the effort of solving in these parts is not necessary.

For these reasons, a geometrical analysis is routinely run to determine whether a through path exists. After the analysis is finished, unconnected components are removed from the computational grid.

This may speed up the computations but requires some time for the geometrical analysis, especially for very large structures. So if you know beforehand that there are no unrelevant parts or that the structure has been processed to eliminate them, the geometry analysis can be switched off by unchecking **Analyze Geometry**.

Advanced Options: Write Compressed Volume Field

The LIR solver uses a very memory efficient adaptive grid structure for the simulations.

If the option **Write Compressed Volume Fields** is checked, then the adaptive grid is used as compression method for writing out *.vap files. This option allows to save 80-90% space on hard drive. The runtime for writing *.vap files is also reduced significantly. But the runtime for loading and uncompressing of compressed *.vap is increased by the amount of runtime that was saved for writing out compressed *.vap files.

If the option **Write Compressed Volume Fields** is not checked, then a usual regular grid is used for writing out *.vap files.

Advanced Options: Optimization Options

The **Multigrid Method** (see, e.g., Wesseling, 2004) was introduced to speed-up the computation and reduce the runtime significantly. The main idea of **Multigrid** is the usage of multiple coarser adaptive grids to speed up convergence behavior but requires only little more memory.

The method is available to solve the Stokes and Stokes–Brinkman equations as well as for solving mechanics, diffusion, thermal, and electrical conduction and is enabled by default.

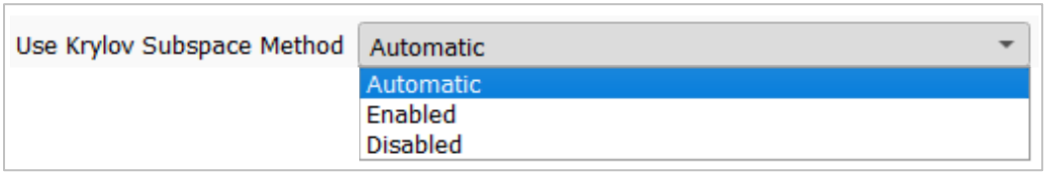
Another speed-up option to accelerate the convergence behavior of the LIR solver is called **Krylov Subspace Method**.

The runtime of the LIR solver depends on many different properties of the structure and

the simulation parameters. The BiCGStab algorithm is used, which can reduce the runtime for challenging simulation very drastically.

Note! Using the BiCGStab method approximately doubles the amount of RAM needed for the computation.

Unfortunately, the Krylov method is not always faster than a simulation without the Krylov method and therefore we introduced an **Automatic** mode which uses some heuristics to choose the most efficient method based on structure, material parameters, and boundary conditions automatically. Of course, it is possible to explicitly enable (**Enabled**) or disable (**Disabled**) the method.

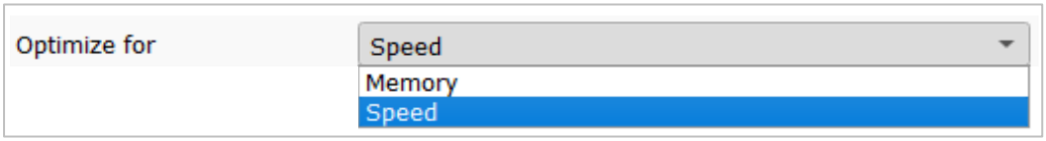


Note! In case that the Krylov subspace method (BICGStab) is used, the **Relaxation** may also be adapted automatically.

Depending on the material parameters and geometry of the structure, the underlying mathematical problem can vary in complexity, thus influencing the behavior of the solver. The more complex the problem is, the more stable the solver settings should be.

With the **Relaxation** number, the solver is adjusted from **Stable** (which results in higher number of iterations, slower time stepping, and longer solver run times), to **Fast**, which makes the solver run less iterations but implies the risk that the solver does not converge. The **Relaxation** is a parameter of the SOR method and must be between 0 and 2 to ensure convergence. For relaxation values smaller than one (<1.0), the simulation is more stable. For relaxation values larger than one (>1.0), the simulation converges faster.

The **LIR** solver can **Optimize for Speed** or **Memory**.



- If **Speed** is chosen, the solver constructs additional optimization structures. The runtime is decreased by up to 30% but requires up to 50% more memory compared to the other option.
- If **Memory** is chosen, the runtime is increased by up to 40% but the solver requires up to 50% less memory.

Advanced Options: Grid Options

The **Grid Type** decides what kind of tree structure is used for the simulation.

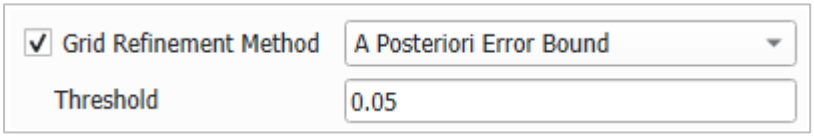
The default option is **LIR-Tree** and should always be used. The solver uses an adaptive grid structure called LIR-tree and needs up to 10 times less runtime and memory compared to the **Regular Grid** option.

The solver can analyze the result field during the computation and improves the adaptive grid in places where more accuracy is needed. The LIR solver splits cells where a high gradient occurs.

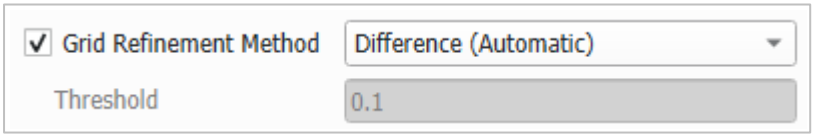
The solver can analyze the computed field and refines the adaptive grid during the computation at locations where more accuracy is needed. The LIR solver splits cells where a high gradients occur when the **Grid Refinement Method** is enabled. In this case, the additional parameters **Number of Grid Refinements**, **Allow Sub-Voxel Resolution**, and **Allow Grid Re-Coarsening** become available.

Select one of the three available refinement methods from the drop-down menu.

When **A Posteriori Error Bound** is selected, the solver targets the specified accuracy **Threshold**. While the **Error Bound** (set as Simulation Stopping Criterion) determines the relative error in the solution of the linear system, **A Posteriori Error Bound** refers to the relative error estimated by comparing high-order and low-order discrete solutions. The accuracy **Threshold** refers to the relative error to the analytical solution and the value must be between 0.0 and 1.0.

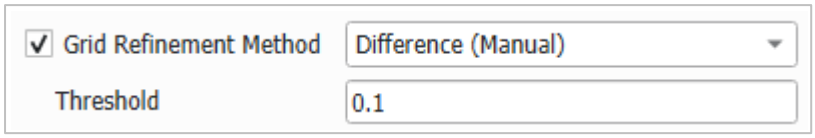


Choosing **Difference (Automatic)** leads to computational cells being split when the difference in values between neighboring cells exceeds a certain **Threshold**. The solver automatically chooses all internal parameters based on the structure and simulation settings. For this option, cells are split where the current gradient is greater than the **Threshold** multiplied by the maximum gradient, where the threshold is determined automatically.

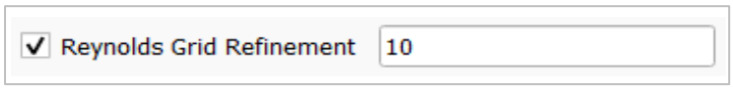


Selecting **Difference (Manual)** also causes the computational cells to be split when the difference in values between neighboring cells exceeds the specified **Threshold**. You can specify all parameters (Threshold and Number of Grid Refinements) for the grid refinement manually. For this option, cells are split where the current gradient is

greater than the **Threshold** multiplied by the maximum gradient. The **Threshold** value must be between 0.0 and 1.0. The recommended value range is between 0.05 and 0.1.



Activate **Reynolds Grid Refinement** to use the local Reynolds number (Re) for adaptive grid refinement.



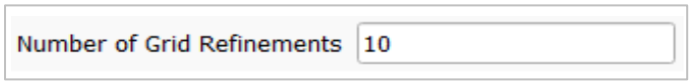
A grid cell is split if the local Reynolds number, i.e., the ratio between inertial and viscous forces exceeds the given threshold. The Reynolds number is defined as:

$$\mathrm{Re} = \frac{\rho \mathrm{u} \mathrm{L}}{\mu} \quad (57)$$

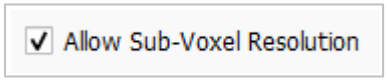
Reynolds Number

where ρ is the density of the fluid (SI units: kg/m^3), u is the velocity of the fluid (m/s), μ is the dynamic viscosity of the fluid (Pa·s or N·s/m^2 or kg/m·s), L is a characteristic length (m). For the local Reynolds number, the size of the grid cell is used for the characteristic length.

The **Number of Grid Refinements** controls the maximum number of grid refinements that the solver can perform during the simulation. The value should be set between 0 and 10, where a value of 0 means that no grid refinements will be made. Grid refinements may increase the number of iterations, runtime, and memory requirements.



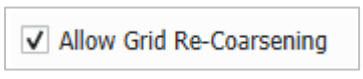
When **Allow Sub-Voxel Resolution** is enabled, the solver is allowed to split computational cells to sizes smaller than the voxel length.



This feature is beneficial for low-porosity structures for which the pore throats require finer resolution or for modeling fast Navier-Stokes flows with strong vortices.

Enabling this feature may increase the number of iterations, runtime, and memory requirements.

Check **Allow Grid Re-Coarsening** to allow the solver to automatically revert the grid refinement and coarsen the computational cells.



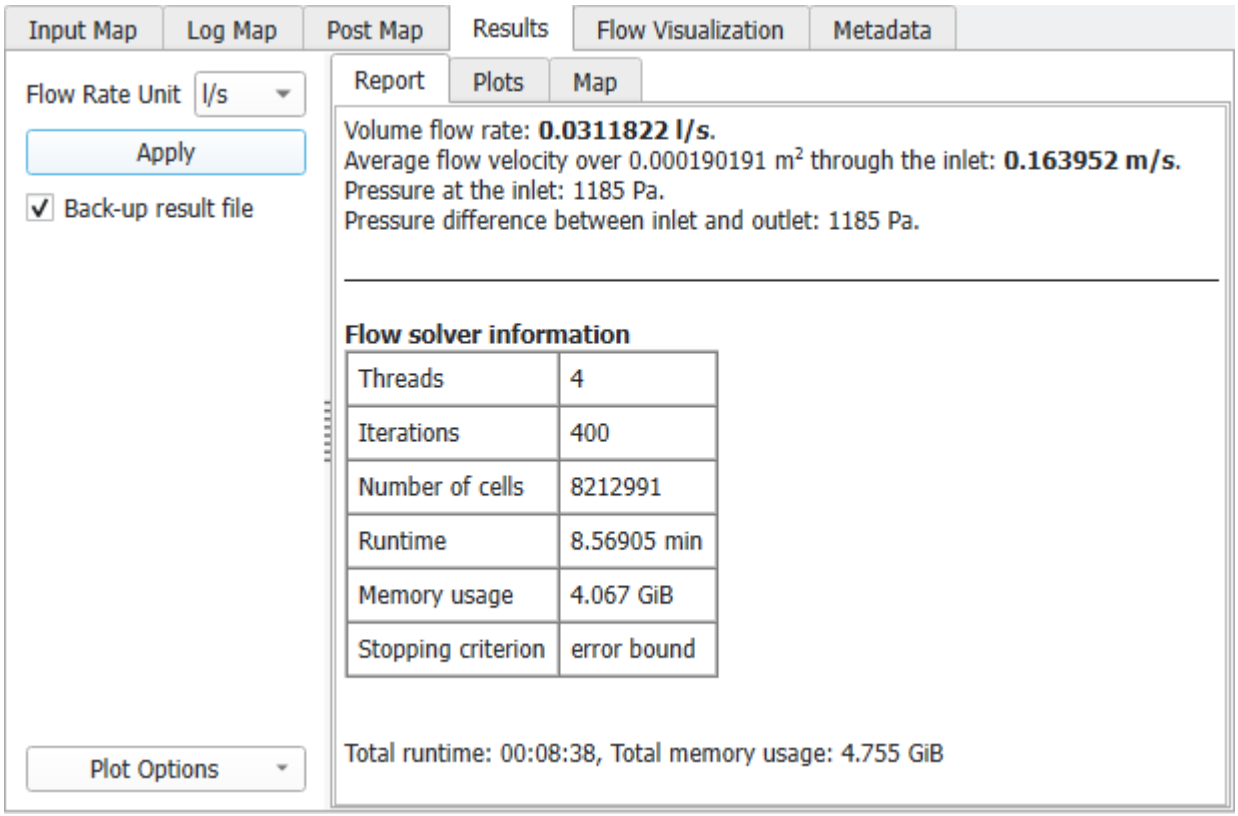
This means, grid refinement is done temporarily. Afterwards, the cells are merged as

soon as accuracy is not needed anymore. This reduces the memory and runtime requirements.

Results

Click **OK** to input the entered parameters, and then click **Run** in the FilterDict section to start the flow simulation.

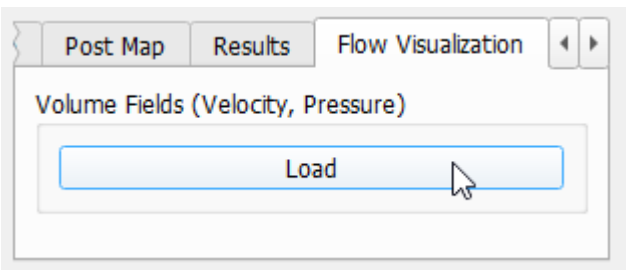
The results are immediately shown in the opening Result Viewer after the simulation is finished.



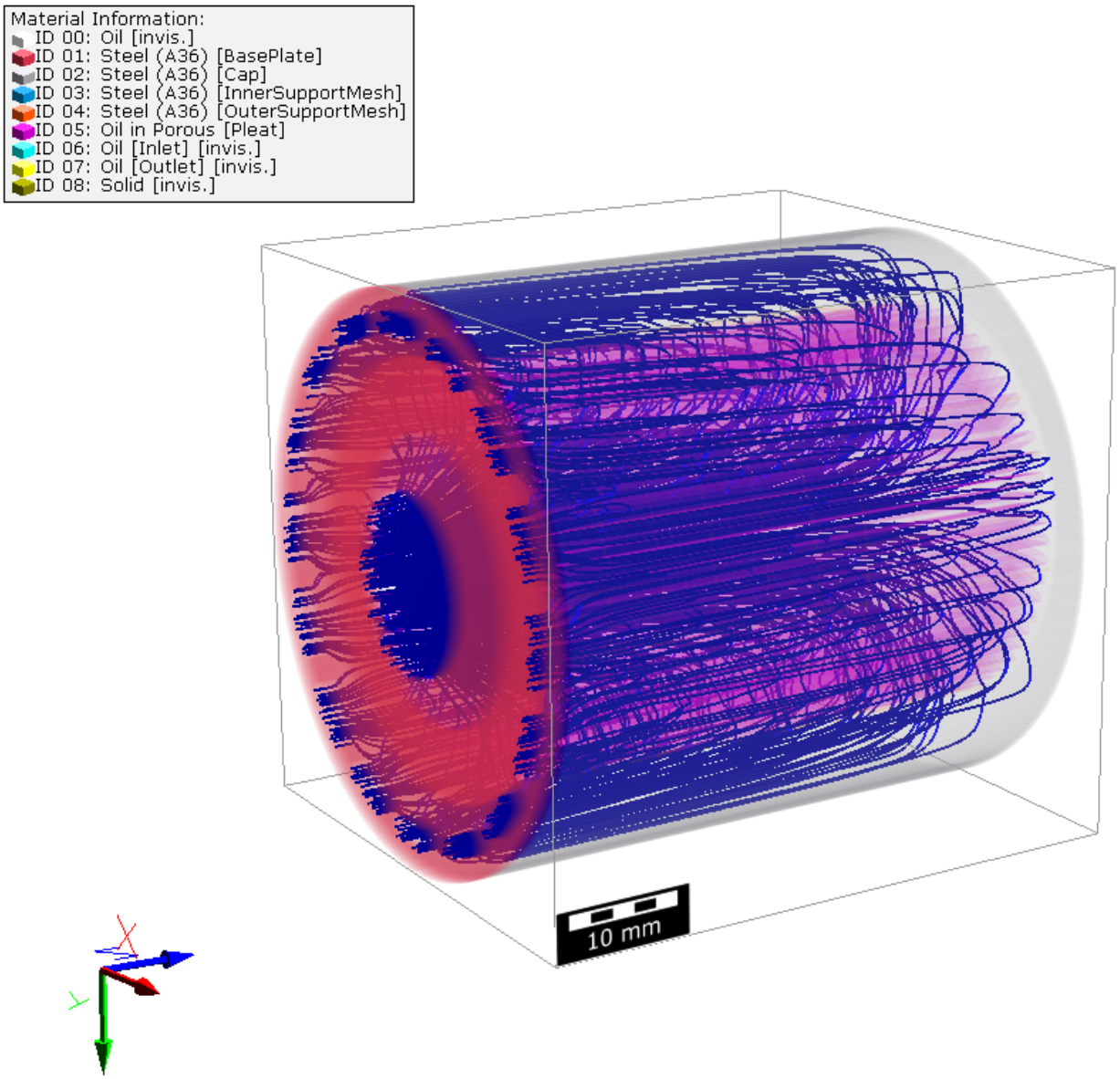
The resulting flow rate can be displayed in various units by changing the **Flow Rate Unit** and clicking **Apply...**

Flow Visualization

To visualize the computed flow and pressure field, click **Load** in the **Flow Visualization** tab.



Refer to the Visualization user guide for details.


Material Information:
ID 00: Oil [invis.]
ID 01: Steel (A36) [BasePlate]
ID 02: Steel (A36) [Cap]
ID 03: Steel (A36) [InnerSupportMesh]
ID 04: Steel (A36) [OuterSupportMesh]
ID 05: Oil in Porous [Pleat]
ID 06: Oil [Inlet] [invis.]
ID 07: Oil [Outlet] [invis.]
ID 08: Solid [invis.]
10 mm

1.5 Cross Flow

Select **Cross Flow** from the drop-down menu. The functionality is identical to the Filter Flow Experiment 202 command with the exception that for **Cross Flow**, multiple outlets can be defined.

The **Cross Flow** command does not simulate particle movement and deposition. For this, use the Cross Flow Filtration GeoApp 232.

The Cross Flow command

- Options 217
  How to set the input parameters for the Cross Flow command.
- Results 229
  How to understand the computed results.
- Flow Visualization 230
  How to visualize the computed results.

1.5.1 Options

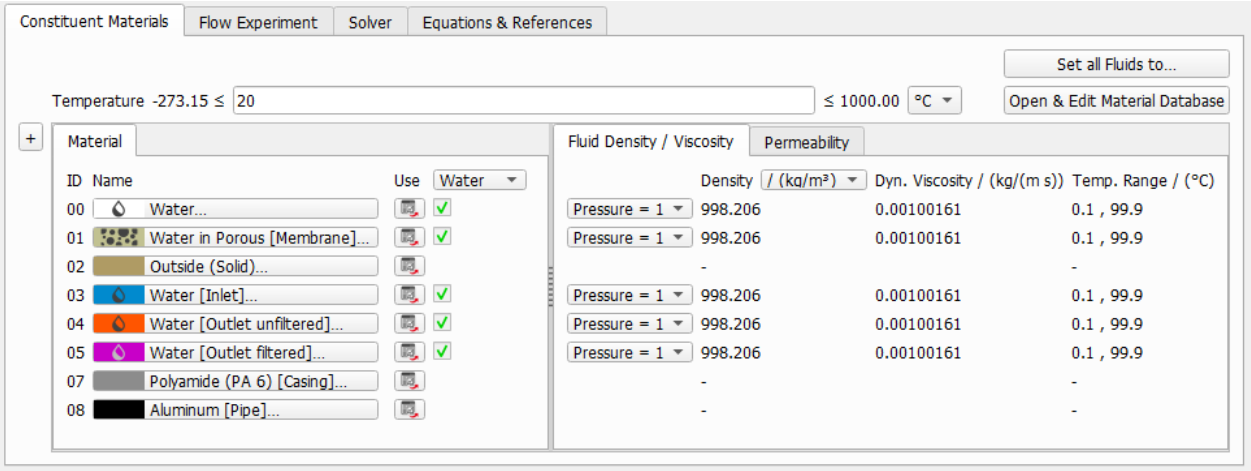
Select **Cross Flow** from the pull-down menu and click the **Option**'s **Edit...** button to open a dialog to enter or edit the simulation options. Choose a **Result File Name** for the result file (*.gdr) fitting to your current project.

The parameters are organized under the tabs: **Constituent Materials**, **Flow Experiment**, and **Solver**. The **Equations & References** tab gives a summary of the used equations and symbols which are explained in more detail in this user guide.

Tabs of the Cross Flow dialog

- Constituent Materials 218
  Define fluid type and temperature. Set material IDs and parameters for porous layers.
- Flow Experiment 219
  Define inlet and outlet material IDs, and set the flow boundary conditions.
- Solver 220
  Select the flow solver, set the stopping criteria and define the computational resources.

Constituent Materials



Temperature

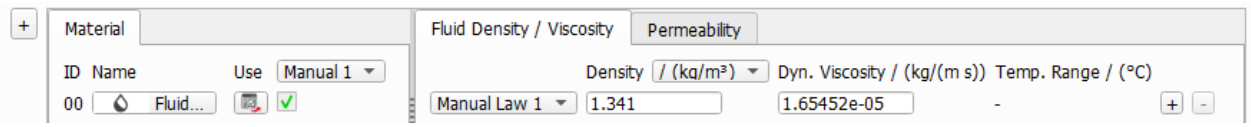
The **Temperature** for the filtration process is selectable in Kelvin (K), Celsius (°C), and Fahrenheit (°F). The chosen temperature may change the density and viscosity values of the fluid during the simulation, if those are defined to be temperature-dependent in the material database. Furthermore, if the Brownian motion of the particles is included in the simulation, it determines the strength of the Brownian motion through equation (13) 8.

Fluid Density and Viscosity

FilterDict uses single-phase flow simulations. It is not possible to run a flow simulation if two fluids are present in the structure. If multiple fluids are present in the structure, you may click **Set all Fluids to...** and select one fluid for the simulation.

The selected fluid is also displayed in the drop-down menu in the **Material** subtab behind **Use**. The density and the dynamic viscosity of this fluid are shown under the **Fluid Density/Viscosity** subtab.

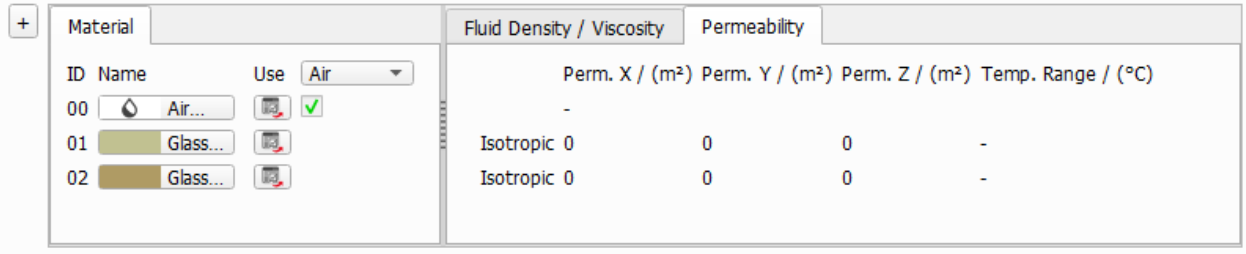
If a fluid from the GeoDict Material Database is used, the values for **Density** and **Dynamic Viscosity** are taken from the database and may depend on the given **Temperature** value. Select **Manual** as fluid material to enter parameter values manually.



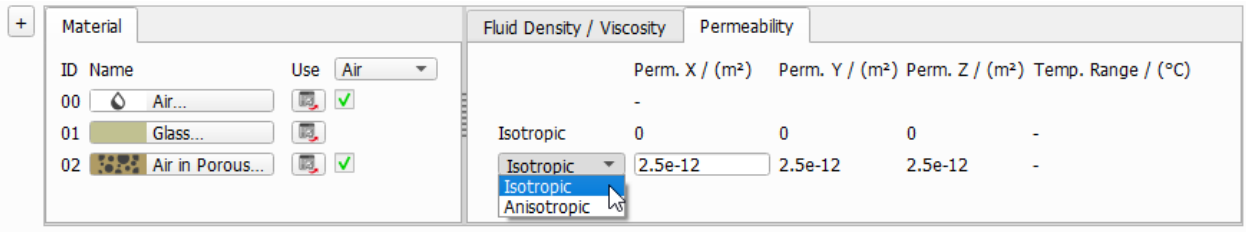
To add materials to the database, use the **Open & Edit Material Database** button. More information on editing, expanding, and using the **GeoDict Material Database** is available in the Material Database user guide.

Filter Media Permeability

Under the **Permeability** subtab, enter the permeability of porous materials present in the structure. As long as there are only fluid and solid materials present in the structure, no parameters are needed here:



For porous materials, choose if the material is **Isotropic** or **Anisotropic**. In the isotropic case, the permeability is the same in all space directions. In the anisotropic case, enter a different permeability for each spacial direction.



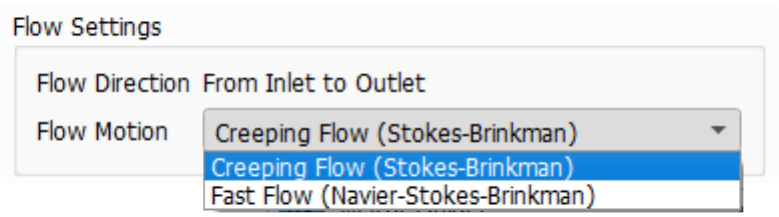
You can select and change a material by clicking on the material button.

Flow Experiment

Flow Settings

The **Flow Direction** is fixed to **From Inlet to Outlet**.

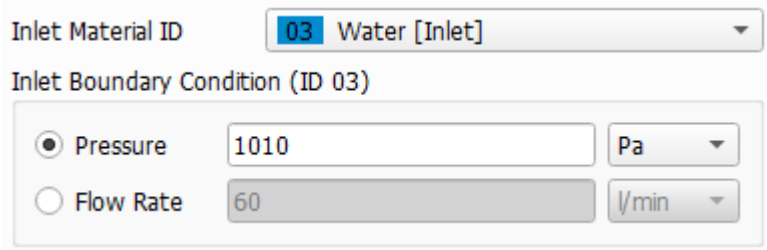
The **Flow Motion** regime can be chosen as **Creeping flow (Stokes)** or **Fast flow (Navier-Stokes)** based on the flow velocity and the Reynolds number.



Inlet Material ID and Inlet Boundary Condition

Define the inlet of the flow by selecting the **Inlet Material ID**. For this, it is necessary that the inlet is marked with a different material ID. This can be easily done by selecting a voxel on the inlet area and using the **Flood Fill Component Plane** feature of the **Voxel Selection** tool. For

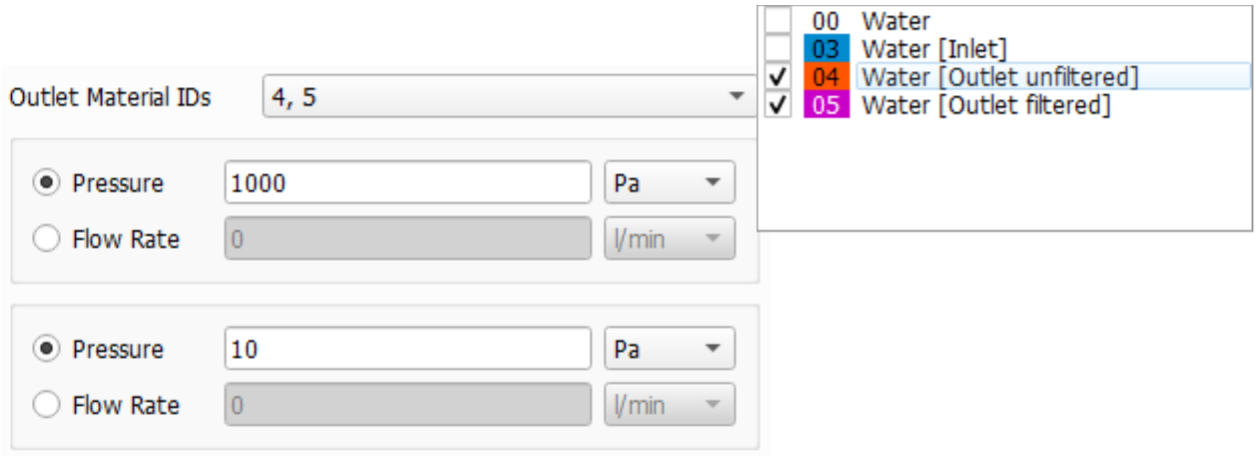
an automatic recognition of Inlet phase it is advised to provide the information “Inlet” to the selected material ID.



The **Inlet Boundary Condition** can be either set by prescribing the **Pressure** or the **Flow Rate**.

Outlet Material IDs

Define the outlet area of the flow by selecting the **Outlet Material IDs**. In cross flow setups multiple outflow areas must be selected. At each outflow area, set the local **Pressure** value.



Domain Boundary Conditions

The 3D model typically contains also the filter housing. In this case, flow will only occur inside of the filter housing and the domain boundary conditions do not apply. However, if the filter is symmetric, it is possible to reduce the computational cost by modelling only one half or one quarter of the filter. In these cases, the flow area will be open to one or two sides of the domain and one should select **Symmetric** boundary conditions for the flow at the domain boundary.

No-Slip means that the structure is encased in a solid wall at the domain boundary.

Solver

The **Complete Filter - Cross Flow** command always uses the LIR solver to compute the flow.

Simulation Stopping Criterion

The equations are solved by an iterative approach.

Iterative Solver

The basic idea of an iterative method is to

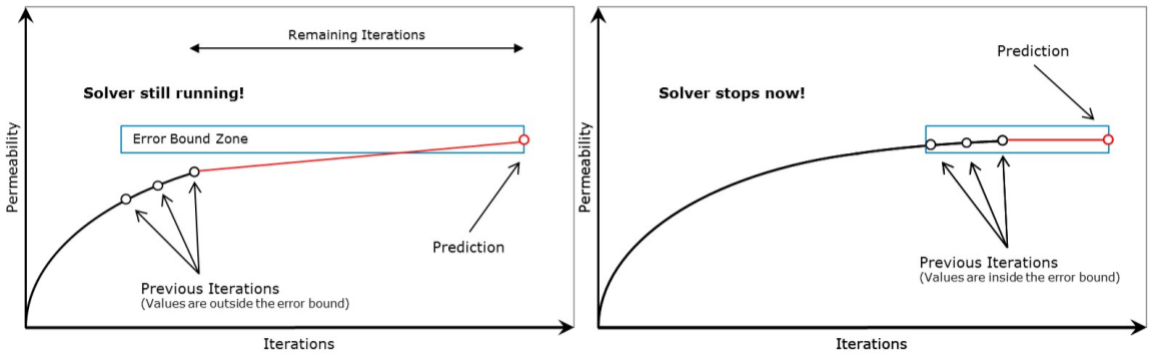
1. Start with some initial guess for the unknown values
2. Improve the current values in each iterative step
3. Repeat the iterative process until one of the stopping criteria is reached.

The iterative process is controlled by setting the values and activation for **Error Bound, Tolerance**, **Maximal Iterations**, and **Maximal Run Time (h**). The stopping criteria apply for each computational direction individually.

If multiple stopping criteria are selected, the first criterion that is reached causes the solver to stop.

The stopping criterion that has been reached can be viewed in the Result Viewer of the GeoDict result file (*.gdr) under the **Results Report** tab.

The stopping criterion **Error Bound** uses the result of previous iterations, and predicts the final solution based on linear and quadratic extrapolation. The solver stops if the relative difference between computed and predicted solution is smaller than the specified error bound. The stopping criterion recognizes oscillations in the convergence behavior and prevents premature stopping at local minima or maxima. A damped convergence curve is fit through the oscillating curve and the solver stops then regarding the damped convergence curve.



If the Krylov method (under LIR - Advanced Options) is activated, the definition of **Error Bound** is somewhat different. Here, no prediction is made, instead the continuity between neighboring cells and the conservation of mass is checked. The maximal value is normalized by the mean flow and if this value is smaller than the **Error Bound** the simulation stops.

The **Tolerance** stopping criterion looks for stagnation of the method when the process becomes stationary, i.e., the improvement in the pressure drop value becomes extremely small from iteration to iteration.

In each iteration, the solver checks for the current computed value x_c against the values

x_{c-i} of the last 100 iterations if there are any changes.

$$\max_{i\in[1,100]} \frac{x_c - x_{c-i}}{x_{c-i}} < \text{tolerance} \quad (58)$$

The computation is stopped if the maximal relative change is smaller than the value entered for **Tolerance**. When there is doubt about the quality of the solution, decrease the **Tolerance** value by a factor of ten. The drawback of this criterion is that the solver sometimes might stop too early in case of slow convergence rate or at local extrema of oscillatory convergence curves.

Use the **Maximum Iterations** value or the **Maximum Run Time (h)** stopping criteria causes the solver to stop if the maximal number of iterations and/or the maximal runtime (in hours) is exceeded.

When the solver stops because one of these criteria has been reached, no guarantee on the quality of solution can be given. In this case, a warning is printed into the report. The following possibilities might help:

- Check the corresponding .log file to see how large the residual values and the pressure drop values are. If the pressure drop values are in reasonable range and they do not have changed during the last iterations, you may decide to use the current result.
- Double-check the structure and parameter values. Unphysical parameters or too rough resolution of the structure (leading, e.g., to artificial unconnected components) can cause an iterative solver to fail.

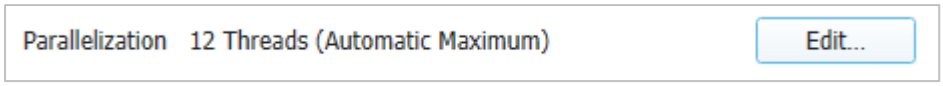
Restart Save Interval

The calculations run by the solvers can be restarted from saved intermediate result files and the interval between auto-saves can be configured from the value entered in **Restart Save Interval (h)**.

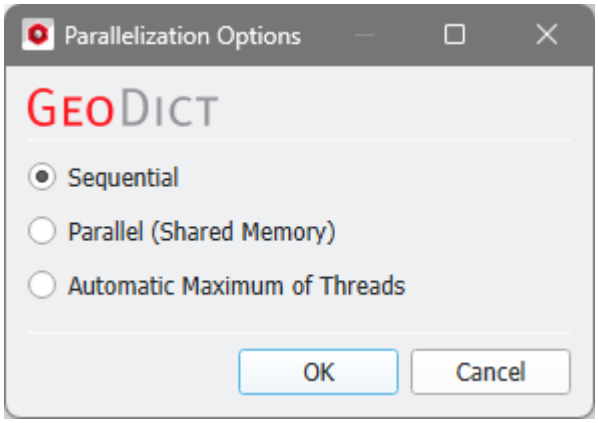
Parallelization

Control how many threads are used for the computation. Parallelization is possible if your license and hardware allow it.

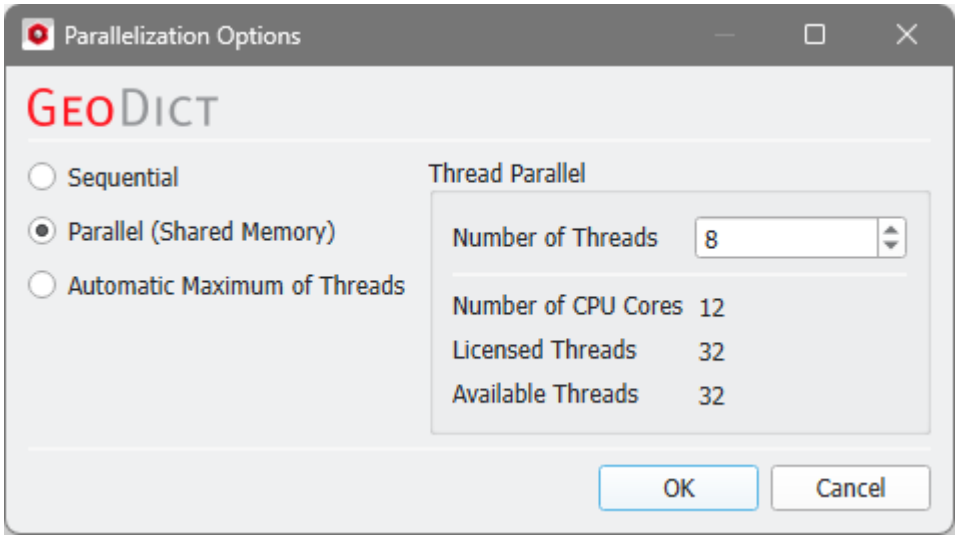
The **Parallelization Options** dialog opens when clicking the **Edit** button and you can choose between **Sequential**, **Parallel (Shared Memory)**, or **Automatic Maximum of Threads**.



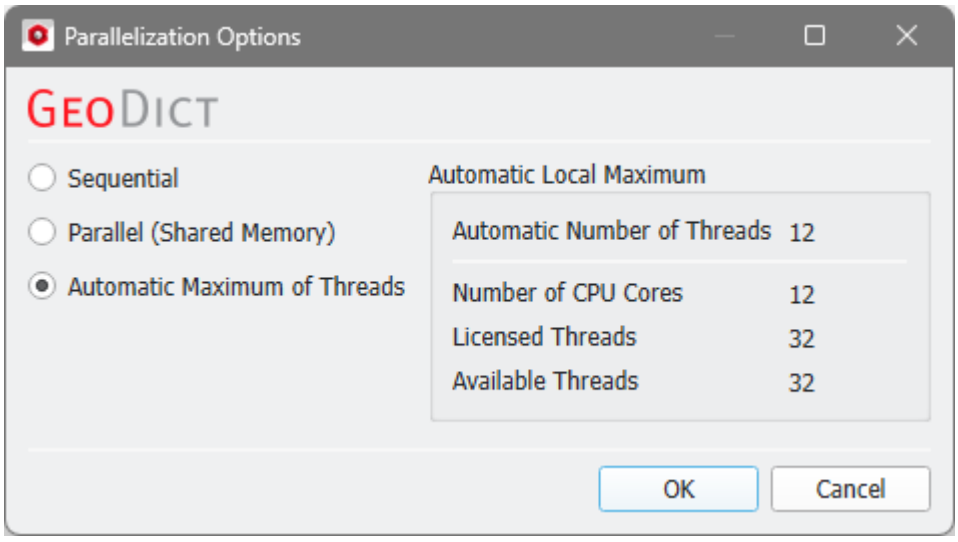
Selecting **Sequential** will not apply parallelization and only one thread is used for the computation.



When **Parallel (Shared Memory)** is selected, the **Number of Threads** can be entered. Below, the **Number of CPU Cores** that the current machine has, the maximum number of **Licensed Threads** and the number of those licensed threads that are available (**Available Threads**) are shown in the dialog. Of course, the maximal number of parallel processes you can use, is the smallest of those three numbers.



If **Automatic Maximum of Threads** is selected, the number of parallel processes is automatically selected for optimal speed, based on the CPU cores and licensed parallel processes.



The **Automatic Local Maximum** of processes is automatically selected, which is the minimum of **Number of CPU Cores**, **Licensed Threads**, and **Available Threads**.

Restart From .gdr File

In some situations, it may be useful to re-use previously computed results and save a great amount of time by restarting the process from an existing GeoDict result file (*.gdr). Typical examples would be

- non-sufficient accuracy of some computation, when it is suspected that more iterations may improve the quality of the result,
- trying another boundary condition (e.g., Dirichlet instead of Periodic or other way around),
- continuing an iteration process,
- or using similar parameter values for the computation.

To restart and initialize the solver from a result file (*.gdr), check **Restart from .gdr File** and enter the name of a file, or **Browse** for it in the project folder.

When using **Restart From .gdr File** only the volume field (*.vap file) associated to the loaded *.gdr file will be used to continue the simulation.

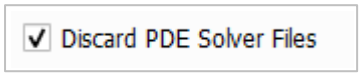
Important! The *.vap file is saved in the result folder of a previous run. Make sure that this file was not deleted from the results folder. If the *.vap file is not available the restart option cannot be used.

No other parameters from the options dialog will be loaded from the *.gdr file, but it is possible to load them by clicking **Load Parameters** in the Result Viewer. This way the simulation can be continued while using the same settings as in the initial simulation.

Important! The structure from the run of which you want to use the result file must be loaded into GeoDict. If this is not the case, an error message is displayed.

Discard PDE Solver Files

Checking **Discard PDE Solver Files** causes the deletion of all intermediate computation files, such as log files and flow field files.



Only the content of the final GeoDict result file (*.gdr) is stored.

While having the benefit of saving hard disk storage place, discarding intermediate solver files has also the effect of disabling the 3D visualization of the results.

Advanced Options: Analyze Geometry

Checking **Analyze Geometry** performs an analysis of the geometry before the solver computations.

If no percolation path through the structure is found, the partial differential equation does not need to be solved, and a pressure drop of 0.0 is directly reported as the solution.

Furthermore, the solver operates on the whole structure, regardless of which parts are connected or unconnected. However, unconnected components are not transport relevant, thus the effort of solving in these parts is not necessary.

For these reasons, a geometrical analysis is routinely run to determine whether a through path exists. After the analysis is finished, unconnected components are removed from the computational grid.

This may speed up the computations but requires some time for the geometrical analysis, especially for very large structures. So if you know beforehand that there are no unrelevant parts or that the structure has been processed to eliminate them, the geometry analysis can be switched off by unchecking **Analyze Geometry**.

Advanced Options: Write Compressed Volume Field

The LIR solver uses a very memory efficient adaptive grid structure for the simulations.

If the option **Write Compressed Volume Fields** is checked, then the adaptive grid is

used as compression method for writing out *.vap files. This option allows to save 80-90% space on hard drive. The runtime for writing *.vap files is also reduced significantly. But the runtime for loading and uncompressing of compressed *.vap is increased by the amount of runtime that was saved for writing out compressed *.vap files.

If the option **Write Compressed Volume Fields** is not checked, then a usual regular grid is used for writing out *.vap files.

Advanced Options: Optimization Options

The **Multigrid Method** (see, e.g., Wesseling, 2004) was introduced to speed-up the computation and reduce the runtime significantly. The main idea of **Multigrid** is the usage of multiple coarser adaptive grids to speed up convergence behavior but requires only little more memory.

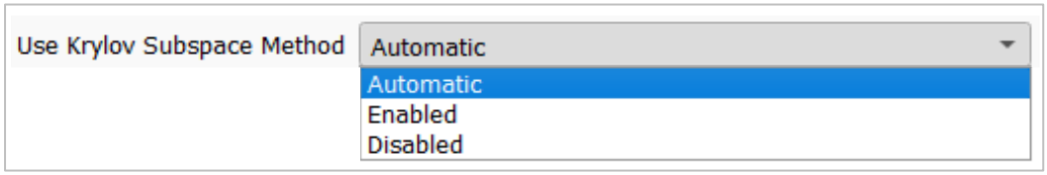
The method is available to solve the Stokes and Stokes–Brinkman equations as well as for solving mechanics, diffusion, thermal, and electrical conduction and is enabled by default.

Another speed-up option to accelerate the convergence behavior of the LIR solver is called **Krylov Subspace Method**.

The runtime of the LIR solver depends on many different properties of the structure and the simulation parameters. The BiCGStab algorithm is used, which can reduce the runtime for challenging simulation very drastically.

Note! Using the BiCGStab method approximately doubles the amount of RAM needed for the computation.

Unfortunately, the Krylov method is not always faster than a simulation without the Krylov method and therefore we introduced an **Automatic** mode which uses some heuristics to choose the most efficient method based on structure, material parameters, and boundary conditions automatically. Of course, it is possible to explicitly enable (**Enabled**) or disable (**Disabled**) the method.



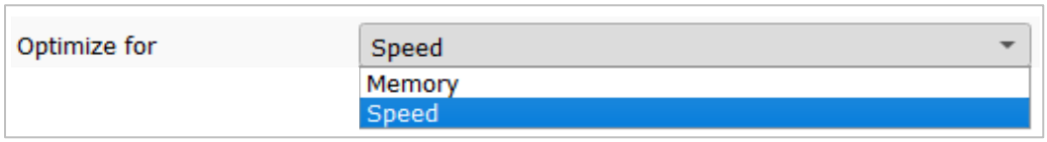
Note! In case that the Krylov subspace method (BICGStab) is used, the

Relaxation may also be adapted automatically.

Depending on the material parameters and geometry of the structure, the underlying mathematical problem can vary in complexity, thus influencing the behavior of the solver. The more complex the problem is, the more stable the solver settings should be.

With the **Relaxation** number, the solver is adjusted from **Stable** (which results in higher number of iterations, slower time stepping, and longer solver run times), to **Fast**, which makes the solver run less iterations but implies the risk that the solver does not converge. The **Relaxation** is a parameter of the SOR method and must be between 0 and 2 to ensure convergence. For relaxation values smaller than one (<1.0), the simulation is more stable. For relaxation values larger than one (>1.0), the simulation converges faster.

The **LIR** solver can **Optimize for Speed** or **Memory**.



- If **Speed** is chosen, the solver constructs additional optimization structures. The runtime is decreased by up to 30% but requires up to 50% more memory compared to the other option.
- If **Memory** is chosen, the runtime is increased by up to 40% but the solver requires up to 50% less memory.

Advanced Options: Grid Options

The **Grid Type** decides what kind of tree structure is used for the simulation.

The default option is **LIR-Tree** and should always be used. The solver uses an adaptive grid structure called LIR-tree and needs up to 10 times less runtime and memory compared to the **Regular Grid** option.

The solver can analyze the result field during the computation and improves the adaptive grid in places where more accuracy is needed. The LIR solver splits cells where a high gradient occurs.

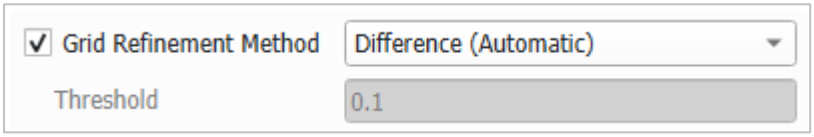
The solver can analyze the computed field and refines the adaptive grid during the computation at locations where more accuracy is needed. The LIR solver splits cells

where a high gradients occur when the **Grid Refinement Method** is enabled. In this case, the additional parameters **Number of Grid Refinements**, **Allow Sub-Voxel Resolution**, and **Allow Grid Re-Coarsening** become available.

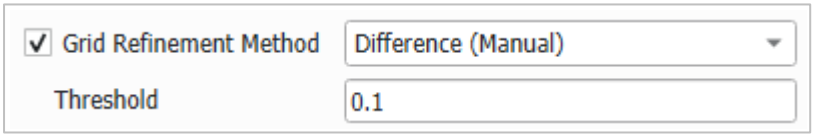
Select one of the three available refinement methods from the drop-down menu.

When **A Posteriori Error Bound** is selected, the solver targets the specified accuracy **Threshold**. While the **Error Bound** (set as Simulation Stopping Criterion) determines the relative error in the solution of the linear system, **A Posteriori Error Bound** refers to the relative error estimated by comparing high-order and low-order discrete solutions. The accuracy **Threshold** refers to the relative error to the analytical solution and the value must be between 0.0 and 1.0.

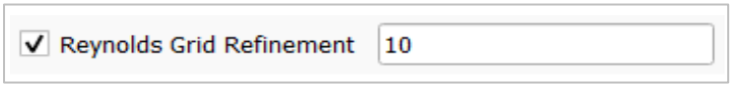
Choosing **Difference (Automatic)** leads to computational cells being split when the difference in values between neighboring cells exceeds a certain **Threshold**. The solver automatically chooses all internal parameters based on the structure and simulation settings. For this option, cells are split where the current gradient is greater than the **Threshold** multiplied by the maximum gradient, where the threshold is determined automatically.



Selecting **Difference (Manual)** also causes the computational cells to be split when the difference in values between neighboring cells exceeds the specified **Threshold**. You can specify all parameters (Threshold and Number of Grid Refinements) for the grid refinement manually. For this option, cells are split where the current gradient is greater than the **Threshold** multiplied by the maximum gradient. The **Threshold** value must be between 0.0 and 1.0. The recommended value range is between 0.05 and 0.1.



Activate **Reynolds Grid Refinement** to use the local Reynolds number (Re) for adaptive grid refinement.

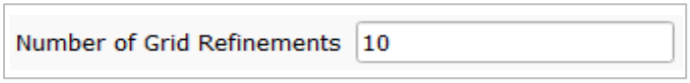


A grid cell is split if the local Reynolds number, i.e., the ratio between inertial and viscous forces exceeds the given threshold. The Reynolds number is defined as:

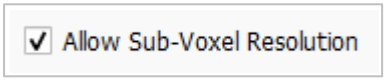
$$\mathrm{Re} = \frac{\rho \mathrm{u} \mathrm{L}}{\mu} \quad (59)$$ Reynolds Number

where ρ is the density of the fluid (SI units: kg/m^3), u is the velocity of the fluid (m/s), μ is the dynamic viscosity of the fluid (Pa·s or N·s/m^2 or kg/m·s), L is a characteristic length (m). For the local Reynolds number, the size of the grid cell is used for the characteristic length.

The **Number of Grid Refinements** controls the maximum number of grid refinements that the solver can perform during the simulation. The value should be set between 0 and 10, where a value of 0 means that no grid refinements will be made. Grid refinements may increase the number of iterations, runtime, and memory requirements.



When **Allow Sub-Voxel Resolution** is enabled, the solver is allowed to split computational cells to sizes smaller than the voxel length.



This feature is beneficial for low-porosity structures for which the pore throats require finer resolution or for modeling fast Navier-Stokes flows with strong vortices.

Enabling this feature may increase the number of iterations, runtime, and memory requirements.

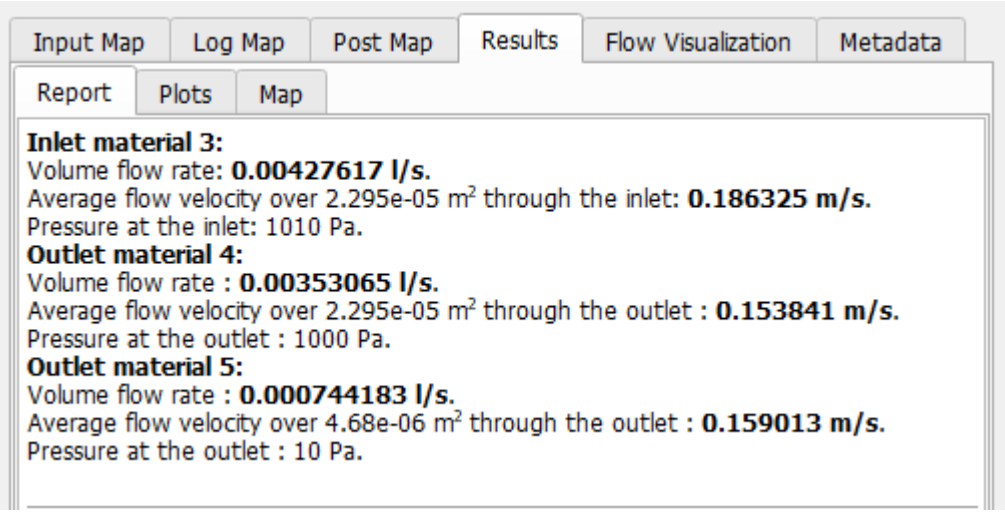
Check **Allow Grid Re-Coarsening** to allow the solver to automatically revert the grid refinement and coarsen the computational cells.

This means, grid refinement is done temporarily. Afterwards, the cells are merged as soon as accuracy is not needed anymore. This reduces the memory and runtime requirements.

1.5.2 Results

Click **OK** to input the entered parameters, and then click **Run** in the FilterDict section to start the cross flow simulation. The results are immediately shown in the opening Result Viewer after the simulation is finished.

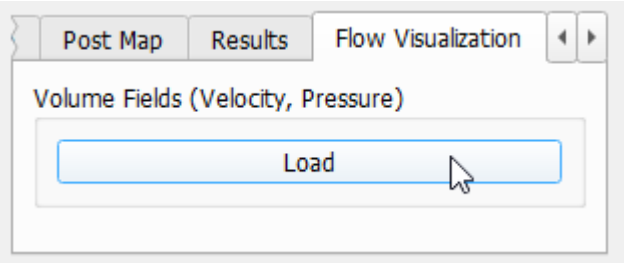
The **Results - Report** subtab reports the computed flow rates and flow velocities at the inlet and at the different outlets. The volume flow rate at the inlet splits up into components leaving through the different outlets.



On the **Plots** tab, the convergence of the computed flow rate and the corresponding residual are shown.

1.5.3 Flow Visualization

To visualize the computed flow and pressure field, click **Load** in the **Flow Visualization** tab.

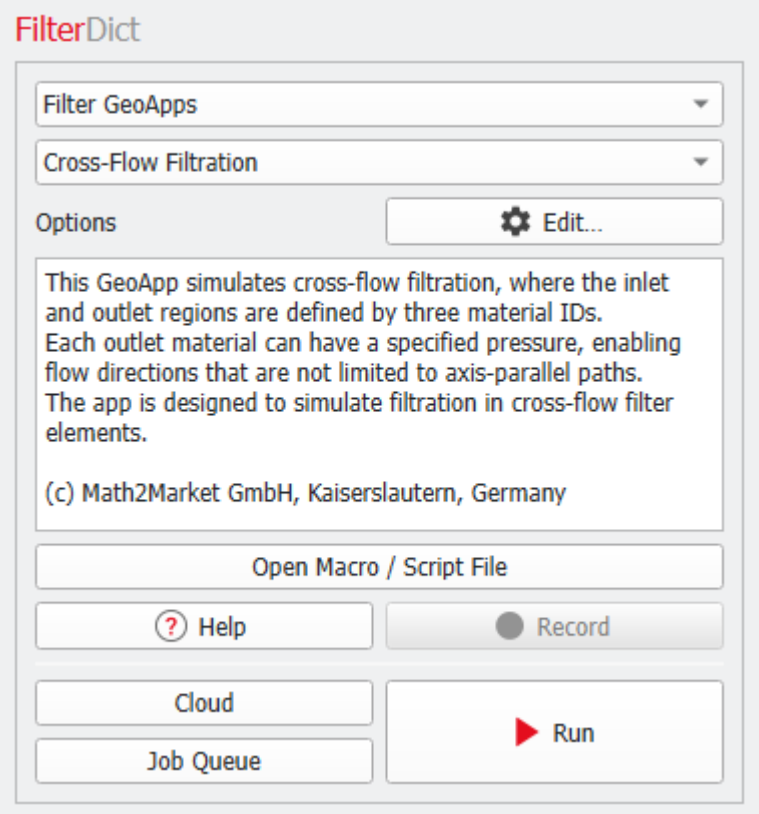


Refer to the Visualization user guide for details.

1.6 Filter GeoApps

Select **Filter GeoApps** to access pre-defined macros included in GeoDict which are useful for filtration simulations.

The macros `Cross-Flow Filtration.py` and `Media2Element.py` are included in the standard installation package. Additional macros can be placed there by the system administrator as described in the GeoApps section and would then be available for all users.



They can be edited with a standard text editor. To open the corresponding *.py file for a predefined macro, click the **Open Macro / Script File** button. After making changes to the script, you can either store the modified macro locally and run it from the main menu **Macro → Execute Macro Script** or as an administrator you may store it in its original place and make the modified version available to all users.

To use a predefined macro, select it from the pull-down menu, click the **Edit...** button to set the parameters, and then the **Run** button to run the simulation.

Apps included in FilterDict

- Cross-Flow Filtration 232
  Run cross-flow filtration simulations.
- Media2Element 233
  Convert Filter Media Lifetime simulation results to the Filter Element scale.

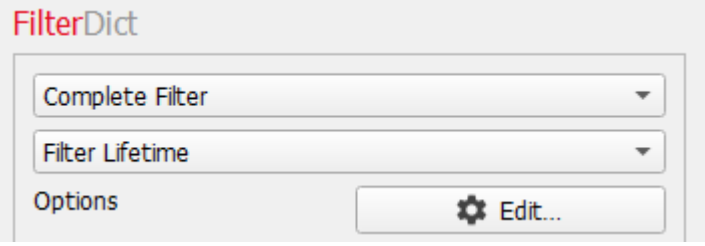
1.6.1 Cross-Flow Filtration

The **Cross-Flow Filtration** GeoApp allows to set up a filtration simulation with one inlet, but with two outlets which have different outlet pressures. In contrast to the Cross Flow 217 command, the app also simulates particle movement and deposition.

Modules needed to run this GeoApp:

FilterDict, FilterDict-Element

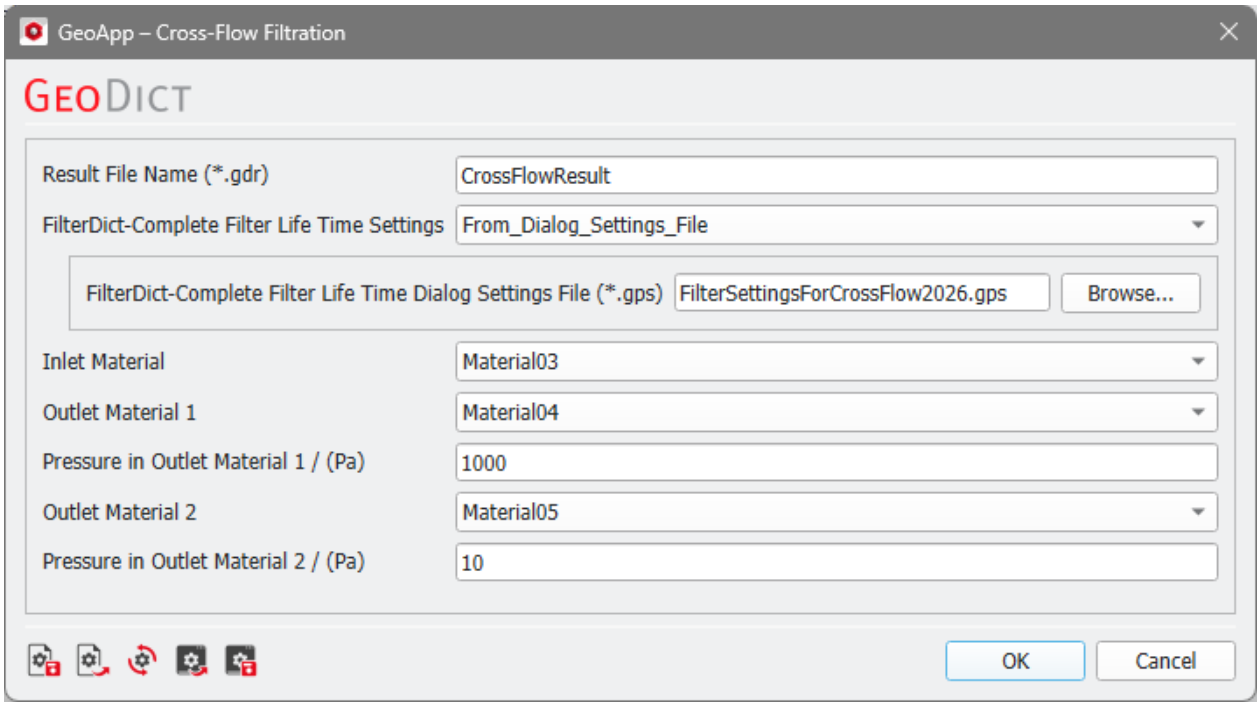
Most of the parameters used are directly loaded from the **Complete Filter - Filter Lifetime** dialog or a corresponding settings file (*.gps). Therefore, in a first step, select **Complete Filter - Filter Lifetime**, open the Complete Filter Lifetime dialog 171 and setup the simulation parameters.



The settings defined in the **Complete Filter - Filter Lifetime** dialog are part of the calculation setup of **Cross-Flow Filtration** GeoApp if you choose to use the current settings. Any changes made during other computations may affect subsequent runs of the **Cross-Flow Filtration** GeoApp. Therefore, unless you are certain that the current configuration is intended, we recommend saving the settings as a *.gps file.

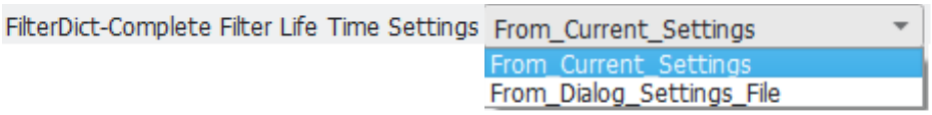
Store the dialog settings in a *.gps file by using the dialog icon. Then close the dialog and return to the Cross-Flow Filtration GeoApp.

The additional settings for the cross-flow filtration simulation can be defined by clicking the **Edit...** button.



At first, choose a **Result File Name**.

Then, select if the parameters should be taken from the **Complete Filter - Filter Lifetime** dialog (select **From_Current_Settings**) or from a stored settings file (select **From_Dialog_Settings_File**).



Now, the only additional selections that must be made are to define the feed **Inlet Material,** the retentate and permeate **Outlet Materials** and the **Pressure in Outlet Materials**. Two outlets, **Outlet Material 1** and **Outlet Material 2**, and their corresponding pressures, **Pressure in Outlet Material 1** and **Pressure in Outlet Material 2**, can be defined. After these selections are made, close the dialog and click **Run** to start the simulation.

1.6.2 Media2Element

This GeoApp allows to convert Filter Media Lifetime 57 simulations to the Filter Element 125 scale.

Modules needed to run this GeoApp:

FilterDict, FilterDict-Element, FilterDict-Media

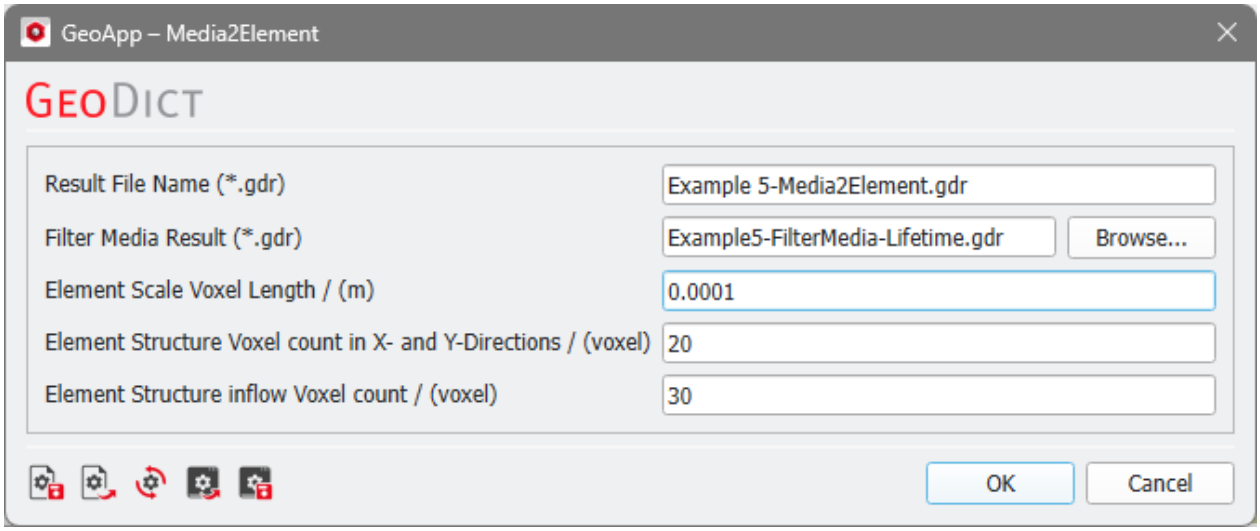
For this, provide a Filter Media Lifetime simulation result (*.gdr file). To be able to create sensible results, the lifetime simulation should

- ... consider a representative part of the filter media (REV),
- ... use a voxel length such that the filter media is well resolved,

- ... run long enough to contain the creation of a filter cake on top of the filter,
- ... simulate a significant number of particles of each size in every batch. This is typically achieved by activating ghost particles.

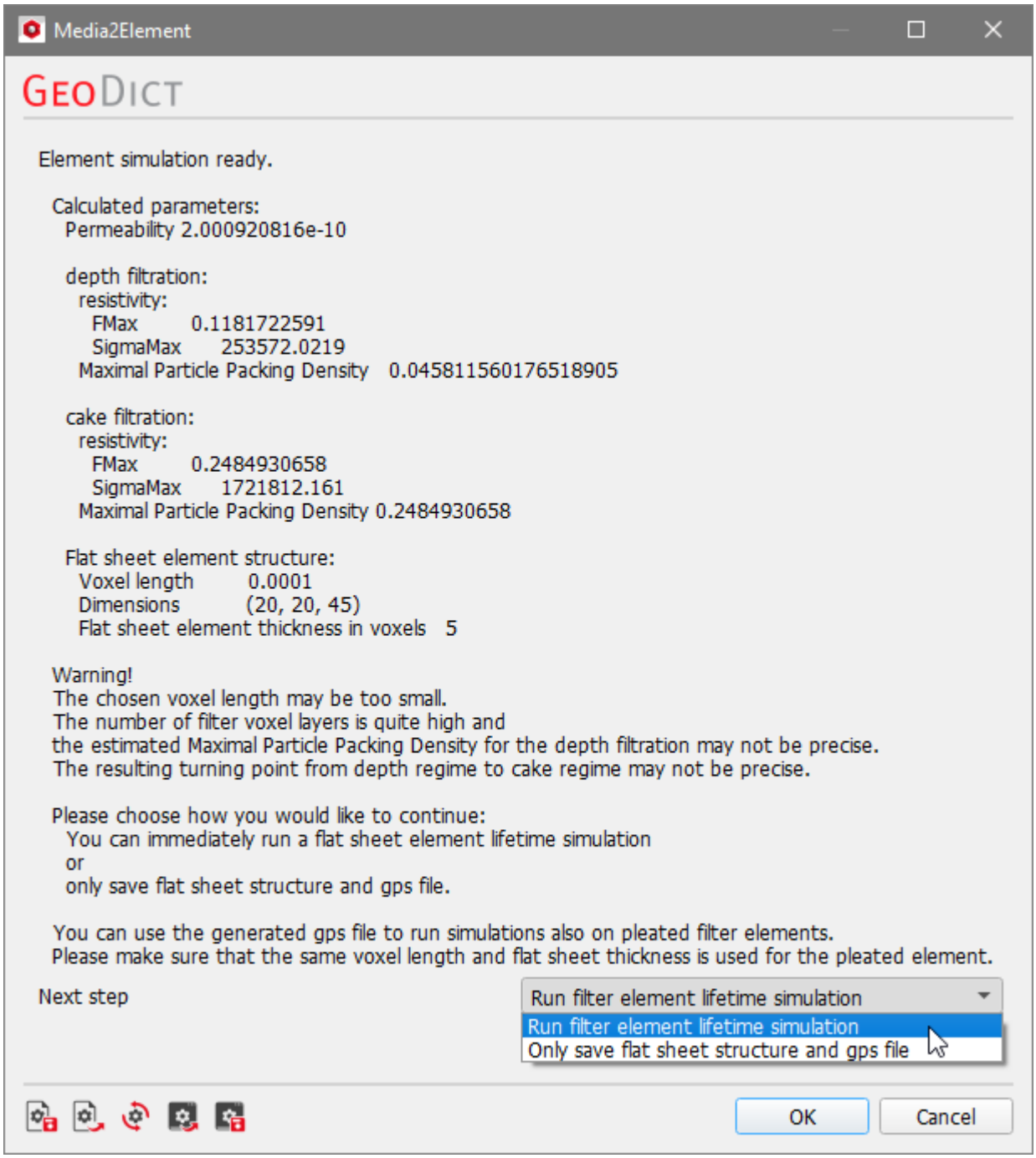
In the app, select a coarser element scale voxel length, such that your filter media thickness is resolved in only a few voxels (the voxel length can even be larger than the media thickness). The app then determines the parameters that describe the filter properties on the coarser scale. It creates a sample structure on the coarser scale and either runs a simulation on it or creates a parameter file to import in Filter Element Lifetime simulations. You can also use the calculated clogging parameters in the Complete Filter 169 scale simulations for the characterization of your pleats.

To run the app, click on the **Options Edit...** button.



In the dialog, select the result file of the high resolution **Filter Media - Filter Lifetime** simulation (**Filter Media Result (*.gdr)**). Select the coarser **Element Scale Voxel Length** that is to be used in the low resolution simulation. The last two parameters describe the size of the element scale sample structure that the app creates as a demo case.

Click **OK** to close the dialog and **Run** to start the app. After a few moments a pop-up dialog appears.



The dialog presents the determined parameters. Clean filter permeability, $f_{\max}$ (FMax) and $\sigma_{\max}$ (SigmaMax) are directly taken from the results of the filter media simulation, i.e., this are the parameters shown in the Filter Clogging Analysis[106] part of the filter media simulation result file:

Filter clogging analysis

	Layer thickness / (mm)	f_{max}	$sigma_{max}$ / (kg / m^3 s)
Depth filtration	0.5	0.118714	253997
Cake filtration	0.118	0.248504	1.71516e+06

Clean filter permeability: 2.00092e-10 m^2

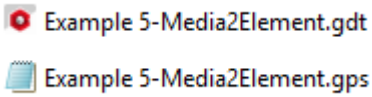
The Maximal Particle Packing Density is often equal to $f_{\max}$, but may differ in some cases depending on the selected resolution. This is the case, if the resulting filter element is only one voxel in thickness, or if the voxel length is even larger than the filter media height.

In cases where the voxel length selected for the coarse scale simulation is too small, a warning appears.

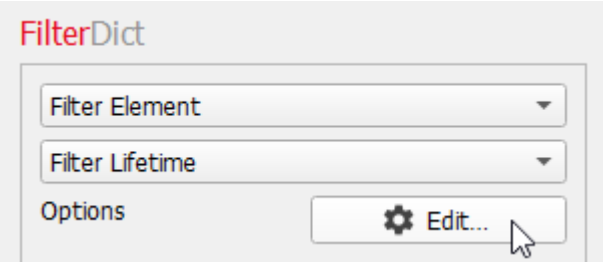
The dialog asks what should be done as next steps. Two choices exist:

Only save flat sheet structure and gps file

This option creates two files in the result folder, named as the selected result file name. A *.gdt file containing the structure and a *.gps file containing the dialog settings:



To run a simulation on the filter element scale, first load the created *.gdt file into GeoDict. Then, select **Filter Element** and **Filter Lifetime** from the drop-down menus and open the **Options** dialog.

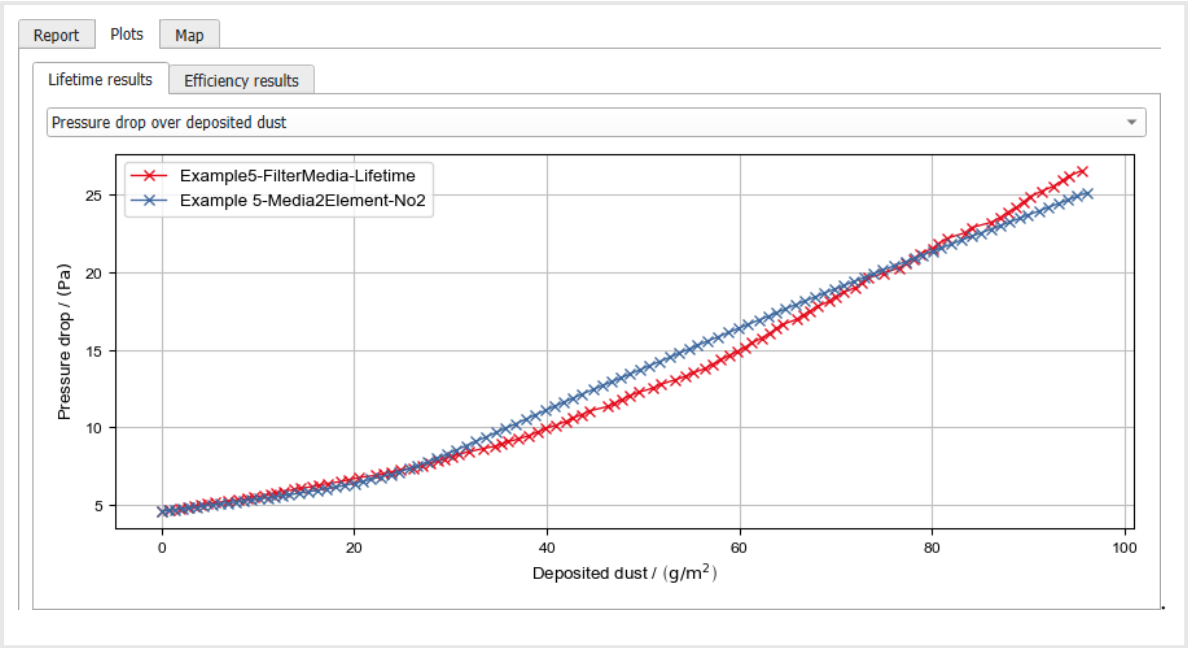


In the dialog, load the setting from the *.gps file. Then run the simulation.

Run filter element lifetime simulation

This option runs a simulation on the coarse scale. The sample geometry is a flat sheet media, where the size of the 3D model (in voxels) is determined by the App's input parameters.

Additionally, a second *.gdr result file is created, beginning with `combined_` followed by the result file name. It contains a comparison of the results obtained on the fine scale (the filter media simulation used as input) and the results obtained on the coarse scale (computed now).


Report
Plots
Map
Lifetime results
Efficiency results
Pressure drop over deposited dust
Example5-FilterMedia-Lifetime
Example 5-Media2Element-No2
Pressure drop / (Pa)
Deposited dust / (g/m²)
0
20
40
60
80
100
5
10
15
20
25

1.7 User-Defined Functions (UDFs)

You may change some of the inbuilt functionality of FilterDict with the help of user defined functions. FilterDict UDFs are written in C++. Five different types of user defined functions (UDF) are possible in FilterDict:

UDFs available in FilterDict

- Particle Motion UDF 239
  Change the equations that describe the particle motion.
- Collision UDF 242
  Define collision models in addition to the in-built models (Caught on First Touch, Sieving, and Hamaker)
- Pass Through UDF 247
  Define alternative pass-through models that calculate the passing probability for particles passing through a porous voxel.
- User Data UDF 251
  Attach additional data to a particle, that can be used and modified during the simulation. Use this UDF combination with one of the other UDF types.
- Electrostatic Decay UDF 253
  Describe the decay of electrostatic charges with your own decay function.
- Particle Status Codes 255
  Table of status codes used for particles.

During the first startup, GeoDict copies a number of sample UDFs into the user's directory at:

- **`/home/`***`username`***`/.geodict2026/UDF/`** (Linux)
- **`C:\Users\`***`username`***`\GeoDict2026\UDF\`** (Windows)

GeoDict does not compile the source code (*.cpp) automatically. Therefore, you must compile the UDF before using it in GeoDict as described below.

Compilation in Linux

The simplest way to work with UDFs is as follows:

1. change into the UDF directory with:

   ```
   cd ./geodict2026/UDF
   ```
2. copy one of the exemplary files. For example:

   ```
   cp ParticleMotionUDF-Standard.cpp MyParticleUDF.cpp
   ```
3. open **`MyParticleUDF.cpp`** in a text editor of your choice (emacs, kwrite, vi...).

The code must be compiled after making all needed changes. For this, the compile.sh script can be used:

```
./compile.sh MyParticleUDF.cpp
```

This script creates the runtime library **`MyParticleUDF.so`**, that can be used by GeoDict.

Compilation in Microsoft Windows

To compile the UDF files delivered with GeoDict, open `FilterDictUDF.sln` with Visual Studio 2013 (or later) and build the solution.

Alternatively, you can run `compilex.bat` from the Visual Studio command prompt (x64). This creates three runtime library files that can be used in FilterDict: `UDFUserData.dll`, `CollisionUDF-Count.dll`, and `CollisionUDF-Hamaker.dll`.

To create additional UDFs, add them as additional projects to your solution. The simplest way to do that is to create a copy of an existing *.cpp and *.vcxproj file first. For example,

1. Create `MyParticleUDF.cpp` as a copy of `ParticleMotionUDF-Standard.cpp` and `MyParticleUDF.vcxproj` as a copy of `ParticleMotionUDF-Standard.vcxproj`.
2. Open the `FilterDictUDF.sln` solution in Visual Studio.
3. Add the existing project `MyParticleUDF.vcxproj`.

This project now still contains the old file `UDFUserData.cpp`. Therefore, remove this file from the project and, instead, add the existing item `MyParticleUDF.cpp` to the project. Then rebuild the project in Visual Studio or run `compilex.bat` to create the runtime library files.

1.7.1 Particle Motion UDF

The Particle Motion UDF allows changing the equations (11), (12), (13), and (14).

This UDF file must contain five functions: `udf_entryMessage()`, `udf_exitMessage()`, `udf_friction()`, `udf_diffusivity()`, and `udf_force()`. Three sample UDF files are delivered with GeoDict: `ParticleMotionUDF-Standard.cpp`, `ParticleMotionUDF-Gravity.cpp`, `ParticleMotionUDF-GravityPlusBuoyancy.cpp`.

udf_entryMessage

This function is called when the UDF is attached and can be used to print a message to the console.

Input: none

Return value: none

Example:

```
void udf_entryMessage(){
  fprintf(stdout,"attaching my new
UDF built %s %
s\n",__DATE__,__TIME__);
}
```

udf_exitMessage

This function is called when the UDF is detached and can be used to print a message to the console.

Input: none

Return value: none

Example:

```
void udf_exitMessage(){
  fprintf(stdout,"detaching my new
UDF built %s %
s\n",__DATE__,__TIME__);
}
```

udf_friction

The function replaces the equation (11) and returns the friction coefficient γ. If the Cunningham correction factor is to be applied, this factor must be defined inside of this function. Therefore, the function also replaces equation (12).

Input:

- InputData_UDF struct

Example: the example file **ParticleMotionUDF-Standard.cpp** contains the same formulas as implemented in GeoDict:

```
double udf_friction(const
InputData_UDF& d) {                  /* Compute the Cunningham
                                     corrected Stokian friction
                                          coefficient of a moving
                                     spherical particle
                                       */


  double lambda = d.meanFreePath;

  double cunningham = 1.0 + lambda
/ d.particleRadius * (1.17 + 0.525
* exp(-0.78 * d.particleRadius /
lambda) );
  double dynViscosity =
d.kinematicViscosity *
d.fluidDensity;
  double g = 6.0 * M_PI *
dynViscosity * d.particleRadius /
cunningham;

  return g;                          // returns friction coefficient in
}                                    [kg/s]
```

udf_diffusivity

The function replaces the equation (13) and return the diffusivity D.

Input:

- InputData_UDF struct
- material ID at the particle position

Example: the example file **ParticleMotionUDF-Standard.cpp** contains the formulas as implemented in GeoDict:

```
double udf_diffusivity(const
InputData_UDF& d, const int
materialID) {

  double gamma = udf_friction(d);   // compute friction coefficient

  return 1.3806485e-23 *            // returns diffusivity in [m^2/s]
d.temperature / gamma;
}
```

udf_force

The function allows to define the external force $\vec{F}$ and replaces equation (14).

Input:

- pointer to double[3] array to store the result
- struct InputData_UDF

Example 1: the file **ParticleMotionUDF-Gravity.cpp** models gravity:

```
void udf_force(double* f, const     // on return, f contains external
InputData_UDF& d) {                 force in [N]

  const double g = 9.81;            // in [m/s^2]

  f[0] = 0;
  f[1] = 0;
  f[2] = g * d.particleDensity *    // gravity in z-direction
(4.18879020479 * d.particleRadius   // 4/3 * Pi = 4.18879020479
* d.particleRadius *
d.particleRadius);

}
```

Example 2: the file **ParticleMotionUDF-GravityPlusBuoyancy.cpp** models gravity and buoyancy forces:

```
void udf_force(double* f, const     // on return, f contains external
InputData_UDF& d) {                 force in [N]

  const double g = 9.81;            // in [m/s^2]
```

```
  f[0] = 0;
  f[1] = 0;

  double buoyancy = g * 4.0/3.0 *
M_PI * pow(d.particleRadius,3) *
d.fluidDensity;
  double gravity = g *
d.particleMass;

  f[2] = gravity - buoyancy;

}
```

GeoDict passes the relevant data to these functions with help of the struct `InputData_UDF`. The struct is defined in the ParticleMotionStructs.h header file which can be found in the `include` folder.

```
struct
InputData_UDF{

  double          // [m]
particleRadius    // [kg/m^3]
;                 // [kg]
  double
particleDensit
y;
  double
particleMass;

  double          // [m²/s]
kinematicVisco    // [kg/m^3]
sity;             // [m]
  double          // [K]
fluidDensity;
  double
meanFreePath;
  double
temperature;

  double          // [m], only available in udf_force function
position[3];      // [V/m], only available in udf_force function
  double
eStatic[3];

  void *          // pointer to user-defined data
userData;

}
```

1.7.2 Collision UDF

The collision UDF may replace the built-in collision models **Caught On First Touch**, **Hamaker**, or **Sieving**. Select it under the Particles - Interaction Model 32 tab.

A collision UDF contains five functions.

Two sample UDF files are delivered with GeoDict: **`CollisionUDF-Count.cpp`**, **`CollisionUDF-Hamaker.cpp`**.

udf_entryMessage

This function is called when the UDF is attached and can be used to print a message to the console.

Input: none

Return value: none

Example:

```
void udf_entryMessage(){
  fprintf(stdout,"attaching my new
UDF built %s %
s\n",__DATE__,__TIME__);
}
```

udf_exitMessage

This function is called when the UDF is detached and can be used to print a message to the console.

Input: none

Return value: none

Example:

```
void udf_exitMessage(){
  fprintf(stdout,"detaching my new
UDF built %s %
s\n",__DATE__,__TIME__);
}
```

udf_numberOfMaterialParameters

The function defines the number of material parameter that the UDF requires as input. This defines the number of material parameters to enter into the **Particles** tab table. These material parameters may be, for example, the Hamaker constant or the restitution.

Input: none

Return value: integer

Example: the example file **CollisionUDF-Count** defines two material parameters

```
int
udf_numberOfMaterialParameters(){
  return 2;
};
```

udf_nameOfMaterialParameters

The function allows to set the headings that will appear in the **Particles** tab table. You must define headings for a number of parameters as defined in the udf_numberOfMaterialParameters 243 function.

Input:

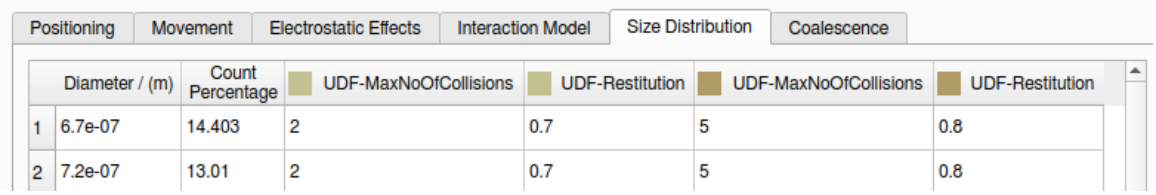
- parameter number (integer)

Return value: const char*

Example: the example file **CollisionUDF-Count** defines two material parameters

```
const char*
udf_nameOfMaterialParameters(int
i){

  if(i==0) return "UDF-                // C++ numbering starts at 0
MaxNoOfCollisions";
  if(i==1) return "UDF-
Restitution";

  return "";                          // return a default value if
}                                     number is out of range
```

For each material using the collision UDF, two columns will now appear in the **Size Distribution** table:



udf_collision

This function describes the collision of a particle with a solid. It is called whenever a particle collides with the solid filter material. The return value of the function is a struct that contains particle position and velocity after the collision and a trap code number. Filtered particles should get a trap code from 32-39; particles that continue to move after the collision should get a trap code of 0.

Input:

- ParticleData_UDF struct

- CollisionData_UDF struct
- double[] array containing the material parameters of the filter material at the collision point entered in the FilterDict dialog. This array has as many entries as defined in the udf_numberOfMaterialParameters 243 function.

Return value: The return value of the function is a struct that contains particle position and velocity after the collision and a trap code 255.

```
struct CollisionResult_UDF{

  double position[3];                    // particle position after
                                         collision [m]

  double velocity[3];                    // particle velocity after
                                         collision [m/s]

  int trapCode;                          // particle trap code: 32+
};                                       means deposited
```

Example: the example file **CollisionUDF-Count**

```
CollisionResult_UDF    /* Material parameters:
udf_collision(Partic      * mp[0]: max number of collisions
leData_UDF& p,            * mp[1]: restitution
CollisionData_UDF&     */
c, const double* mp)
{

  int                  // determine the total number of previous
totalNoOfCollisions=   collisions with all materials
0;
  for(int
i=0;i<nMats;i++)
    totalNoOfCollisi
ons +=
p.collisionCounter[i
];

  CollisionResult_UD
F r;

  r.position[0] =      // set the new particle position to the current
p.position[0];         position
  r.position[1] =
p.position[1];
  r.position[2] =
p.position[2];

  if( totalNoOfColli   // compare the total number of collisions with
sions >= mp[0] ){      the given mp[0]

    r.velocity[0] =    // the particle is trapped:
0.0;                   // the velocity is set to 0
                       // the trapCode to set to 33
```

```
    r.velocity[1] =
0.0;
    r.velocity[2] =
0.0;
    r.trapCode = 33;

  }else{               // particle is not trapped

    double nVP2 =
2.0 * (c.normal[0] *
p.velocity[0] +
c.normal[1] *
p.velocity[1] +
c.normal[2] *
p.velocity[2]);

    r.velocity[0] =    // new velocity is reduced by the restitution
(p.velocity[0] -       factor
c.normal[0]*nVP2) *
mp[1];
    r.velocity[1] =
(p.velocity[1] -
c.normal[1]*nVP2) *
mp[1];
    r.velocity[2] =
(p.velocity[2] -
c.normal[2]*nVP2) *
mp[1];

    r.trapCode = 1;    // trapcode = 1 means 'collision & continue'
                       and will trigger to save a trajectory point

  }
  return r;
}
```

GeoDict passes the relevant data to these functions with help of the structs `ParticleData_UDF` and `CollisionData_UDF`. These structs are defined in the CollisionStructs.h header file which can be found in the `include` folder.

ParticleData_UDF

```
struct ParticleData_UDF{

  double diameter;                 // [m]

  double collisionDiameter;        // [m]

  double depositionDiameter;       // [m]

  double density;                  // [g/l]

  double prevPosition[3];          // [m]
```

```
  double position[3];              // [m]

  double velocity[3];              // [m/s]

  double travelTime;               // [s]

  double diffusivity;              // [m²/s]

  int collisionCounter[nMats];     // total number of collisions with
                                   material i

  int multiplicity;

  int status;                      // the particle status code 255

  double residenceTime[nMats];     // total residence time of particle
                                   in material[i]

  int64_t particleID;

  double velFlow[3];

  void * userData;

};
```

CollisionData_UDF

```
struct CollisionData_UDF{

  double normal[3];

  int material;

};
```

1.7.3 Pass Through UDF

The pass through UDF may replace the built-in pass through models **All Particles pass**, **Constant absorption rate**, **Clogging**, **Constant efficiency**, **Velocity-dependent efficiency**, or **Impassable**. Select it under the Particles - Interaction Model 143 tab.

A pass through UDF contains five functions.

The sample UDF `PassThroughUDF.cpp` is delivered with GeoDict.

udf_entryMessage

This function is called when the UDF is attached and can be used to print a message to the console.

Input: none

Return value: none

Example:

```
void udf_entryMessage(){
  fprintf(stdout,"attaching my new
UDF built %s %
s\n",__DATE__,__TIME__);
}
```

udf_exitMessage

This function is called when the UDF is detached and can be used to print a message to the console.

Input: none

Return value: none

Example:

```
void udf_exitMessage(){
  fprintf(stdout,"detaching my new
UDF built %s %
s\n",__DATE__,__TIME__);
}
```

udf_numberOfMaterialParameters

The function defines the number of material parameter that the UDF requires as input. This defines the number of material parameters to enter into the **Particles** tab table. These material parameters may be, for example, the thickness of the media and the corresponding filter efficiency.

Input: none

Return value: integer

Example: the example file **PassThroughUDF.cpp** defines two material parameters

```
int
udf_numberOfMaterialParameters(){
  return 2;
};
```

udf_nameOfMaterialParameters

The function allows to set the headings that will appear in the **Particles** tab table. You must define headings for a number of parameters as defined in the udf_numberOfMaterialParameters 248 function.

Input:

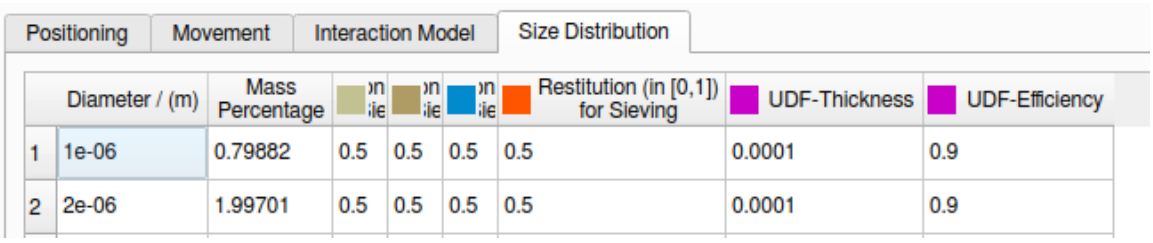
- parameter number (integer)

Return value: const char*

Example: the example file **PassThroughUDF.cpp** defines two material parameters

```
const char*
udf_nameOfMaterialParameters(int
i){

  if(i==0) return "UDF-Thickness"   // C++ numbering starts at 0
;
  if(i==1) return "UDF-
Efficiency";

  return "";                        // return a default value if
}                                   number is out of range
```

For each porous material using the pass through UDF, two columns will now appear in the **Size Distribution** table



udf_passThrough

This function computes the passing probability (19) [12] of a particles moving through a porous material. It is called whenever the particle moves through a porous material.

Input:

- PassThroughUDF_ParticleData struct
- double[] array containing the material parameters of the filter material at the particle position as entered in the FilterDict dialog. This array has as many entries as defined in the udf_numberOfMaterialParameters [248] function

Return value: The probability that the particle moves from the previous position to the current position without being filtered. A result of 1.0 means that a particle will always pass through, and a result of 0.0 means that a particle will always be filtered inside the material.

Example: the example file **PassThroughUDF.cpp**

```
double                                  /* Material parameters:
udf_passThrough( PassThroughUDF_Pa       * mp[0]: l0 (thickness)
rticleData& p, const double* mp )        * mp[1]: filter efficiency
{                                       */

  double disp[3];
  disp[0] = p.posCurr[0]-
p.posPrev[0];
  disp[1] = p.posCurr[1]-
p.posPrev[1];
  disp[2] = p.posCurr[2]-
p.posPrev[2];

  double dist  = norm(disp);

  double speed =
dist/p.travelTime;

  double beta  = 1 /(1-mp[1]);      // compute beta ratio from
                                    efficiency

  double passingProb = pow(beta,-1  // exp(beta,-L/L0)
* dist/mp[0] );

  return passingProb;
};
```

GeoDict passes the relevant data to these functions with help of the struct `PassThroughUDF_ParticleData`. The struct is defined in the CollisionStructs.h header file which can be found in the `include` folder.

PassThroughUDF_ParticleData

```
struct
PassThroughUDF_ParticleData{

  int materialID;                   // material ID of the voxel the
                                    particle is moving through

  double volumefraction;            // fraction of the voxel filled
                                    with previously deposited dust
                                    particles

  double radius;                    // particle radius [m]

  double posPrev[3];                // previous position [m]

  double posCurr[3];                // final position [m]

  double travelTime;                // time needed to move from the
                                    previous position to the final
                                    position [s]
```

```
  void * userData;
}
```

1.7.4 User Data UDF

The input data structs **ParticleData_UDF**, **PassThroughUDF_ParticleData**, and **CollisionData_UDF** contain a **void*** pointer to a user defined data struct. This allows you to attach arbitrary data to a particle. You can modified and use this data in each of the UDFs. For example, a particle-wall collision may change the internal state of the particle, and this may change the particle motion afterwards.

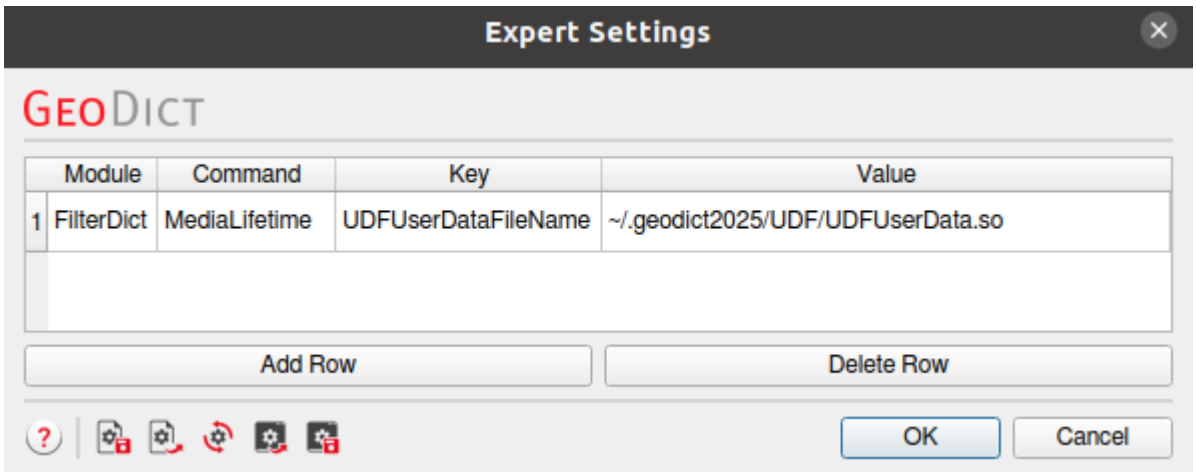
The sample files **UDFUserData.h** and **UDFUserData.cpp** can be used to define and initialize the user defined struct. An example struct is defined in the file **UDFUserData.h**

```
struct UserData {
  int     number;
  double  value;
  char    string[32];
};
```

You may change, add or remove any suitable data to this struct. Each particle will carry its own UserData while its movement is simulated, the data is not shared between different particles. This struct must contain static data and never change in size during the simulation, which excludes the use of **std::vector** or similar classes, that allocate memory dynamically at runtime.

Two functions will be called during the life of a particle. The **startParticle** function will be called at the start of a particle run, such that the UserData may be initialized. The **endParticle** function will be called at the end of a particle run, such that UserData can be printed or saved to a file.

To activate the user data UDF, open **Settings - Edit Expert Settings** in the main menu. There, click **Add Row,** select the appropriate FilterDict command and enter the file path to the .so (Linux) or .dll (Windows) file.



A User Data UDF must contain three functions.

sizeOfUserData

This function is needed to allocate the necessary memory. Do not change!

startParticle

The function will be called at the start of a particle run, such that the UserData may be initialized. In the default case, all UserData is initialized with 0.

Input:

- void* pointer to data

Return value: none

Example: the UDFUserData.cpp file shows how to initialize the example data struct.

```
void startParticle(void* data) {

  if (data == nullptr) return;       // make sure data is initialized

  UserData* userData = (UserData     // cast void* pointer to struct
*) data;

  userData->number = 0;              // initialize values
  userData->value  = 10.0;
  strcpy(userData->string, "none")

}
```

endParticle

The function will be called at the end of a particle run, such that UserData can be printed or saved to a file. This function will always be called on the main thread (not parallelized). Be aware that this function is not necessarily called in the order of the particle ID's.

Input:

- particle ID
- void* pointer to data

Return value: none

Example: the UDFUserData.cpp file saves the final state of the data to a .csv file.

```
FILE *fptr;                          // global file pointer

void endParticle(int64_t
particleID, void* data) {
```

```
  if (data == nullptr) return;       // make sure data is initialized

  UserData* userData = (UserData     // cast void* pointer to struct
*) data;

  if (!fptr) {                       // open file if not already open
    fptr = fopen("UserData.csv",     and add header
"w");
    fprintf(fptr, "ID, number,
value, string\n");
  }

  if (fptr)                          // write single line to file
    fprintf(fptr, "%i,%i,%f,%s\n",
      int(particleID), userData-
>number, userData->value,
userData->string);

}
```

The data can then be used in other UDFs. As example, the file **CollisionUDF-Hamaker.cpp** shows how to access the user data.

At first, the `void*` pointer has to be casted to the actual struct. Afterwards, it can be used to access the variables. This example simply counts the `number` variable up.

```
CollisionResult_UDF udf_collision(
  const ParticleData_UDF& p,
  const CollisionData_UDF& c,
  const double* mp ){

  if (p.userData != nullptr) {                        // make sure userData is
                                                      initialized

    UserData* userData = (UserData *)p.userData;      // cast void* pointer to
                                                      struct

    userData->number += 1;
  }

  // continue with function

}
```

1.7.5 Electrostatic Decay UDF

Replace the decay function (42) 80 of the electrostatic charges with your own decay function. Select it under the Particles - Electrostatic Effects 77 tab.

A electrostatic decay UDF contains three functions.

udf_entryMessage

This function is called when the UDF is attached and can be used to print a message to the console.

Input: none

Return value: none

Example:

```
void udf_entryMessage(){
  fprintf(stdout,"attaching my new
UDF built %s %
s\n",__DATE__,__TIME__);
}
```

udf_exitMessage

This function is called when the UDF is detached and can be used to print a message to the console.

Input: none

Return value: none

Example:

```
void udf_exitMessage(){
  fprintf(stdout,"detaching my new
UDF built %s %
s\n",__DATE__,__TIME__);
}
```

udf_decayFactor

Input:

- surfaceChargeData_UDF struct

Return value: decay factor that describes the current fraction of the electrostatic charge. A value of 1.0 means complete charge, a value of 0.0 means no charges.

Example: The file SurfaceChargeDecayUDF.cpp shows an example:

```
double udf_decayFactor( const
surfaceChargeData_UDF& s ){

  return exp( - 1.0 * s.dust );

}
```

GeoDict passes the relevant data to this function with help of the struct **`surfaceChargeData_UDF`**. The struct is defined in the SurfaceChargeDecayStructs.h header file which can be found in the `include` folder.

surfaceChargeData_UDF

```
struct surfaceChargeData_UDF{

  double dust;                                    // [kg/m²]

  double time;                                    // [s]

  double dustSVF;                                 // [%]
};
```

1.7.6 Particle Status Codes

Each particles carries its own status information during the simulation that describes whether it is moving through the filter or has reached its final position. The status also contains the information, why the particles reached a final state: if may have left the domain, or been captured by the filter material, or some error may have occurred.

When opening *.gpp particle files or *.gpt trajectory files in GeoPy, the table below can be used to understand the particle status. When loading particles directly in GeoDict, the status is automatically converted.

Initial or intermediate status codes	
-1	Initialize velocity from *.vap file
0	Particle continues moving
1	Particle collision point, particle continues moving
Final status codes for unfiltered particles (8-16)	
8	Particle left domain through Z+ side (outlet)
9	Particle left domain through Z- side (inlet)
10	Particle hit end material ID
11	Particle reached maximum displacement (AddiDict only)
12	Time-out: particle continues moving in next batch interval

13	Particle left domain through X+ side (AddiDict only)
14	Particle left domain through X- side (AddiDict only)
15	Particle left domain through Y+ side (AddiDict only)
16	Particle left domain through Y- side (AddiDict only)
Final status codes for filtered particles (32-37)	
32	Particle trapped by adhesion / Hamaker model
33	Status code reserved for Collision UDFs
34	Particle trapped by Sieving model
35	Particle trapped by Caught On First Touch model
36	Particle trapped while passing through porous layers / Pass-Through model
37	Particle trapped by adsorption (AddiDict only)
Error codes	
40	Error: initial particle position overlaps with structure
41	Error: particle stalled by too many bounces on same position
42	Error: particle stalled by continuously having time steps size zero

1.8 References

M. Azimian, C. Kühnle, A. Wiegmann, *Design and optimization of fibrous filter media using life-time multi-pass simulations*, Chemical Engineering & Technology 41, No. 5, 1-9 (2018), https://doi.org/10.1002/ceat.201700585

J. Becker, L. Cheng, C. Kronsbein, A. Wiegmann, *Simulation of cake filtration for polydisperse particles*, Chemical Engineering & Technology 39, 559–566 (2016), https://doi.org/10.1002/ceat.201500350

R.C, Brown, D. Wake, *Loading filters with monodisperse aerosols: Macroscopic treatment*, Journal of Aerosol Science, Vol. 30 (2), pp. 227-234 (1999)

C.T. Crowe, *Multiphase flow handbook*, 1st edition, Taylor and Francis (2006)

C.T. Crowe, *Multiphase Flows with Droplets and Particles*, Second Edition-CRC Press (2011)

M. Dedering, W. Stausberg, O. Iliev, Z. Lakdawala, R. Ciegis, V. Starikovicius, *On new challenges for CFD simulation in filtration*, World Filtration Congress, Leipzig (2008)

D. Hoch, A. Weber, J. Niessner, *Modeling and simulation of coalescence in the context of oil mist filtration*, Transport in Porous Media 148, pp. 47-66 (2023)

O. Iliev, V. Laptev, D. Vasileva, *Algorithms and software for flow through oil filters*, Filtech Europa, Volume I, pp. I-327-I334 (2003)

H. Krupp, *Particle adhesion, theory and experiment*, Advances in Colloid and Interface Sci., 1 (2), 111 (1967)

S. Linden, A. Wiegmann, H. Hagen, *The LIR space partitioning system applied to the Stokes equations*, Graphical Models 82, 58-66 (2015), https://doi.org/10.1016/j.gmod.2015.06.003

F. Lomeland, E. Ebeltoft, W.H. Thomas, *A new versatile relative permeability correlation*, Presented at the International Symposium of the Society of Core Analysts held in Toronto, Canada (2005)

S.V. Partankar, *Numerical Heat Transfer and Fluid Flow*, Hemisphere Publishing Corporation, New York (1980)

S. Rief, A. Latz, A. Wiegmann, *Research Note: Computer Simulations of air Filtration including Electric Surface Charges in 3-dimensional fibrous microstructures*, Filtration Vol 6(2), 169 – 172 (2006)

D.J. Tritton, *Physical fluid dynamics (2nd ed.)*, Oxford: Claredon Press.p. 58. (2006), ISBN 978 – 0198544937

H.K. Versteeg, W. Malalasekera, *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*, Prentice Hall, 2nd Edition (2007)

A. Wiegmann, *Computation of the permeability of porous materials from their microstructure by FFF-Stokes*, Report of the Fraunhofer ITWM 129 (2007)

Citation: Math2Market GmbH, 2026: GeoDict 2026 User Guide, FilterDict, Math2Market GmbH, Germany, doi.org/10.30423/userguide.geodict

MATH 2 MARKET

Math2Market GmbH
Richard-Wagner-Str. 1, 67655 Kaiserslautern / Germany
www.math2market.com