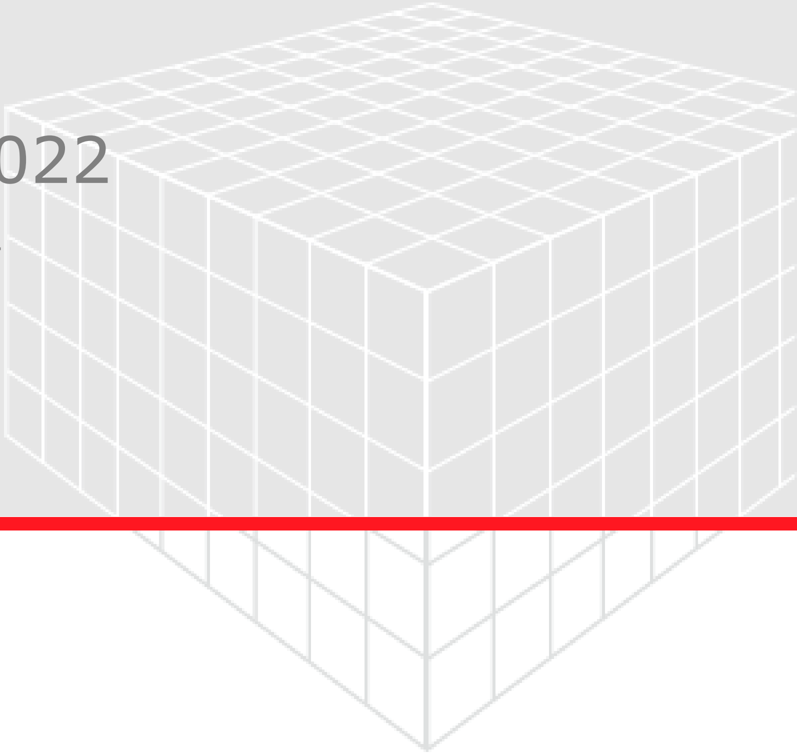
FilterDict

User Guide

GeoDict release 2022

Published: November 25, 2021



GeoDict

https://doi.org/10.30423/userguide.geodict2022-filterdict

SIMULATING FILTRATION WITH FILTERDICT

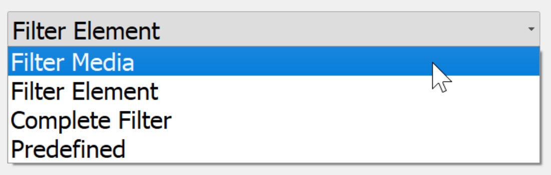
The FilterDict module uses the simulation of fluid flow and a simulation of the particle movement within that flow to predict filter efficiency, filter life time and dust holding capacity of a filter.

With FilterDict, the user 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. It is not possible yet to model liquid-liquid separation processes with the standard FilterDict. Methods to simulate coalescence of liquid droplets are currently under development, please contact support@math2market for more details 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. In previous versions of GeoDict, it was divided into two sub-modules FilterDict-Media and FilterDict-Element, but those features are combined into a single module FilterDict since GeoDict 2021.


Filter Element
Filter Media
Filter Element
Complete Filter
Predefined

- **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 a voxel size.
- **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** simulations are suited for the simulation of a complete filter including the filter housing. Here, different inlet and outlet geometries can be simulated.
- **Predefined** scripts are available to simulate cross-flow filtration.

Depending on the GeoDict license purchased, not all of those options are available.

Theoretical Basis

The simulation of filter processes with the FilterDict module consists of three main processes:

1. **Fluid flow**

   The flow through the (partially clogged) filter is computed with a flow solver.

2. **Particle Tracking**

   Particles are tracked through the calculated flow field. Tracking of particles is based on solving an ordinary differential equation, and includes effects like friction with fluid, 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 flowing 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. **Clogging and Resistivity models**

   When particles are touching the filter surface, different adhesion models can be selected to decide whether the particle sticks to the surface or continues travelling.

For filter life-time simulation, the flow field is recomputed by considering the deposited particles and the three steps are repeated. In this case, one simulation step consisting of the Fluid Flow simulation, Particle Tracking and the Filter Clogging simulation is called a **Batch**.

1. Fluid Flow

FilterDict assumes that the flow is stationary, i.e. does not change over time. In this case, 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.

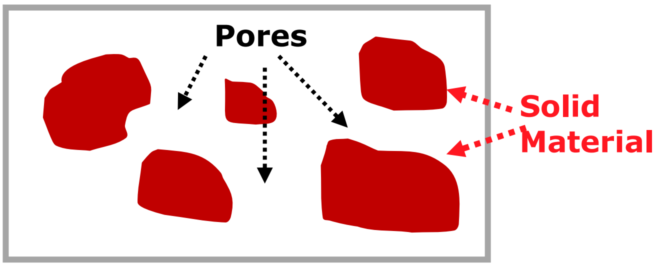
Navier-Stokes Equations

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:

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

$$\nabla\cdot\vec{u} = 0 \qquad (2)$$

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

$$-\mu\,\Delta\vec{u} + \nabla p = \vec{f} \qquad (3)$$

$$\nabla\cdot\vec{u} = 0 \qquad (4)$$

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

Navier-Stokes-Brinkman Equations

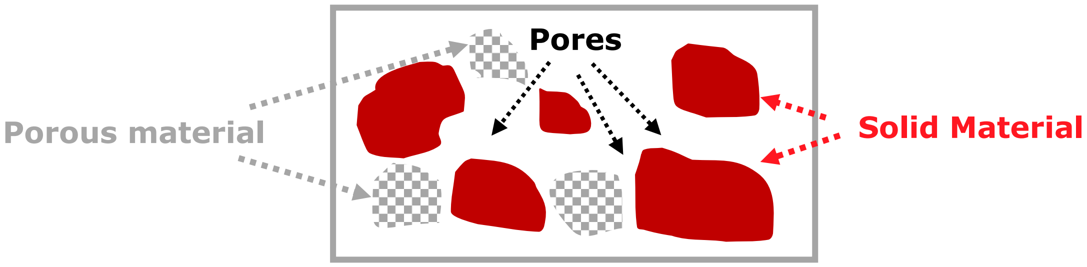
If the geometry of the pores is not fully resolved by the computational grid, a voxel may describe pore space, solid material, or porous material. The Brinkman term allows to describe flow in an unresolved porous medium. In filter media simulations, this is the case when the filter becomes clogged with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

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

$$\nabla\cdot\vec{u} = 0 \qquad (6)$$

Here, $\vec{u}$ is the velocity vector, p is the pressure, $\vec{f}$ is the force, and μ is the effective viscosity. The local viscous flow resistivity σ of a voxel is related to the local permeability by $\sigma = \frac{\mu}{\kappa}$, 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** 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** or **Complete Filter** simulations, however, the pores and solids in the initial structure might be unresolved, and therefore a porous domain initially exists. Here, the Brinkman term describes porous voxels representing the porous media, as well as porous voxels containing deposited particles.



Stokes-Brinkman Equations

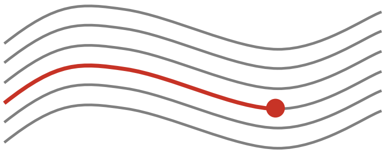
Again, under the assumption of a slow and laminar flow, the equations can be simplified and given as:

$$-\mu\,\Delta\vec{u} + \sigma\vec{u} + \nabla p = \vec{f} \qquad (7)$$

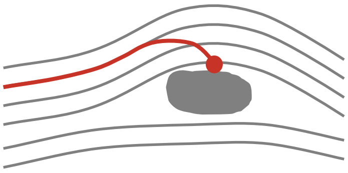
$$\nabla \cdot \vec{u} = 0 \qquad (8)$$

2. Particle tracking

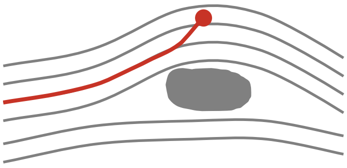
FilterDict simulates filter clogging and tracks how particles are caught in the filter. The movement of particles in the fluid is influenced by the following factors:

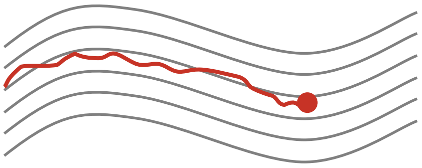


Drag forces are the dominant forces in most cases, and particle trajectories are close to the streamlines of the flow field.



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

Particle Momentum = Stokes Drag + External Forces

or, as formula:

$$m\frac{d\vec{v}}{dt} = 6\pi\mu\frac{R}{C_c}\left(\vec{u} - \vec{v} + \sqrt{2D}\frac{d\vec{W}(t)}{dt}\right) + Q\vec{E} + \vec{F} \qquad (9)$$

In the friction coefficient

$$\gamma = 6\pi\mu\frac{R}{C_c} \qquad (10)$$

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

$$C_c = 1 + \frac{\lambda}{R}\left(1.17 + 0.525e^{-0.78\frac{R}{\lambda}}\right) \qquad (11)$$

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

$$D = \frac{k_B T}{\gamma} \tag{12}$$

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

$$\vec{F} = 0 \tag{13}$$

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

Equations (10), (11), (12) and (13) can be changed through user defined functions, e.g., to include gravity or buoyancy forces in $\vec{F}$. See the Appendix (page 91) for details.

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 $E = -\nabla\Phi$ is determined by solving the Poisson equation for the potential Φ (unit:V)

$$\Delta\Phi = -\frac{\xi}{\epsilon_0}\int_{\partial G} \delta \tag{14}$$

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.

Here, 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 (9) 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.

3. Clogging and Resistivity Models

Choose the filtration model based on the resolution of particles and pore space.

Resolved and unresolved Simulations

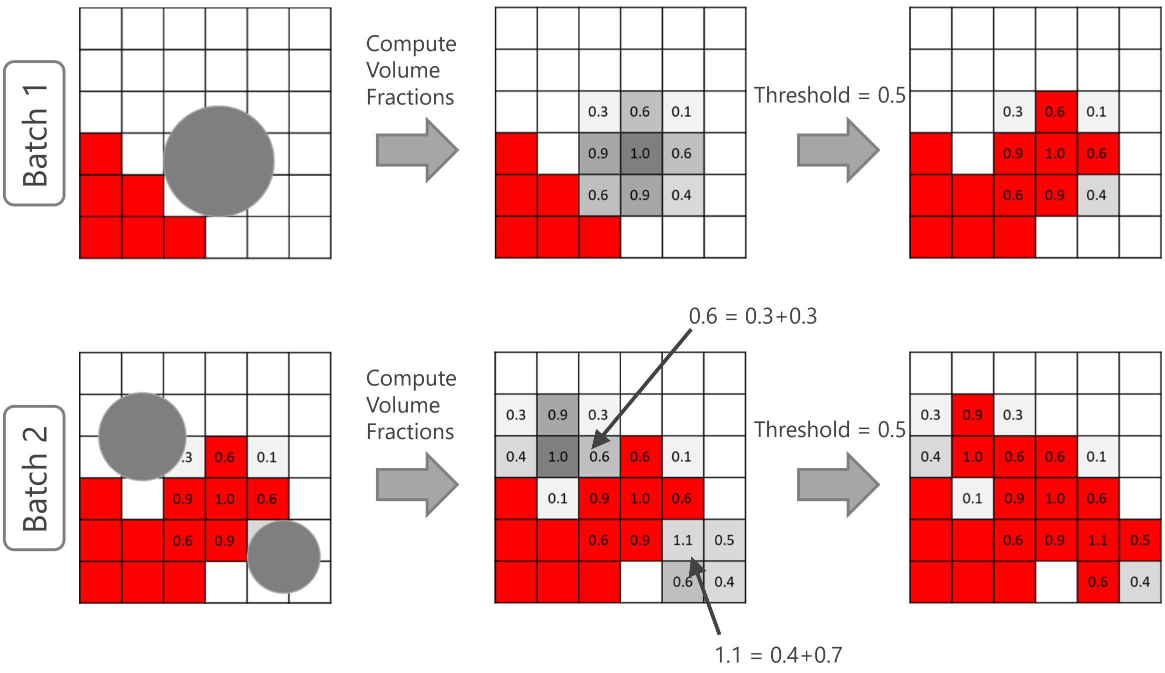
In the examples below, the initial structure consists of empty pores (white in the illustrations below) and full solid (red) voxels. During the filtration process, particles (grey) are caught in the filter. In Filter Element simulations, the original structure can already contain porous voxels.

Resolved Simulations

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

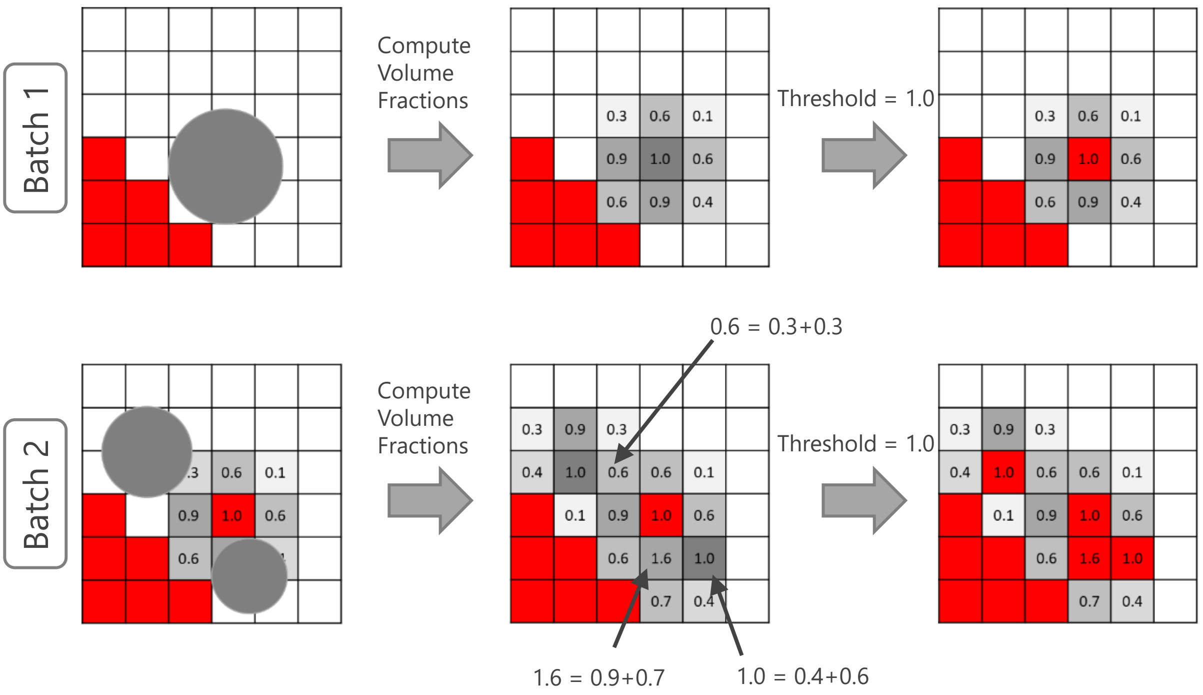
The volume fractions from succeeding batches are added. Under certain circumstances, the volume fraction of a voxel might exceed 1 (e.g., 1.1 as in the example below). In the following two figures, the effect of the choice of the **Volume Fraction Threshold** is shown. A value of 0.5 will usually lead to more realistic results than a value of 1.0.

Resolved Case: Red voxels are solid, all other voxels are treated as empty



Example: Volume Fraction Threshold = 0.5

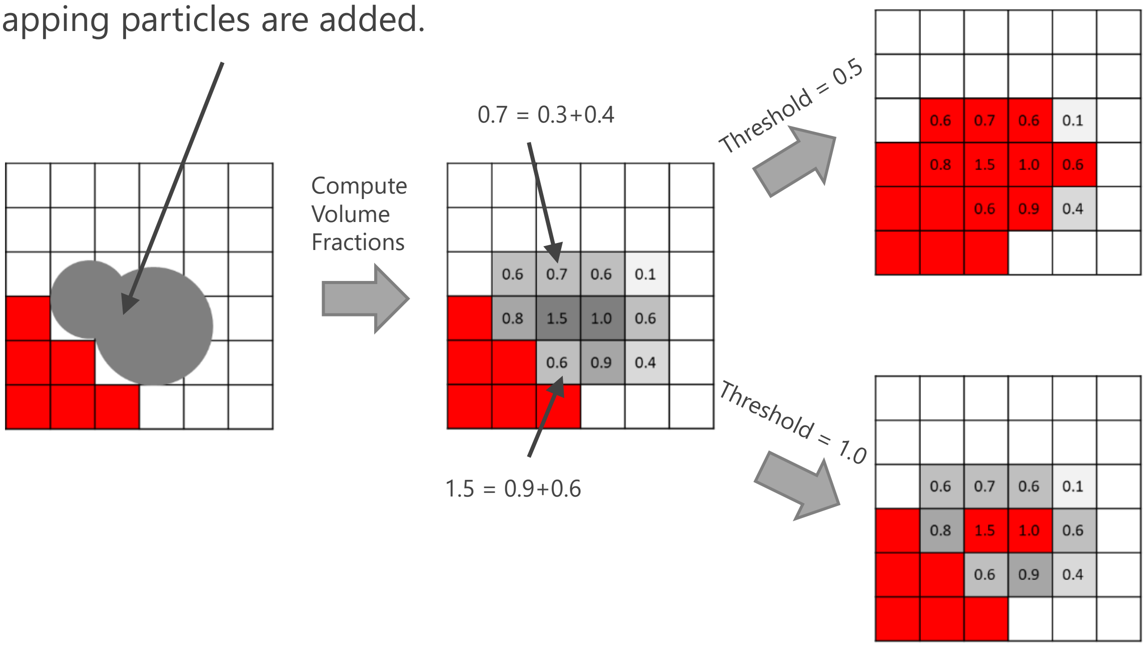
Resolved Case: Red voxels are solid, all other voxels are treated as empty



Example: Volume Fraction Threshold = 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.

Particles in one batch “don’t see” each other, so they might overlap. The volume fractions of overlapping particles are added.

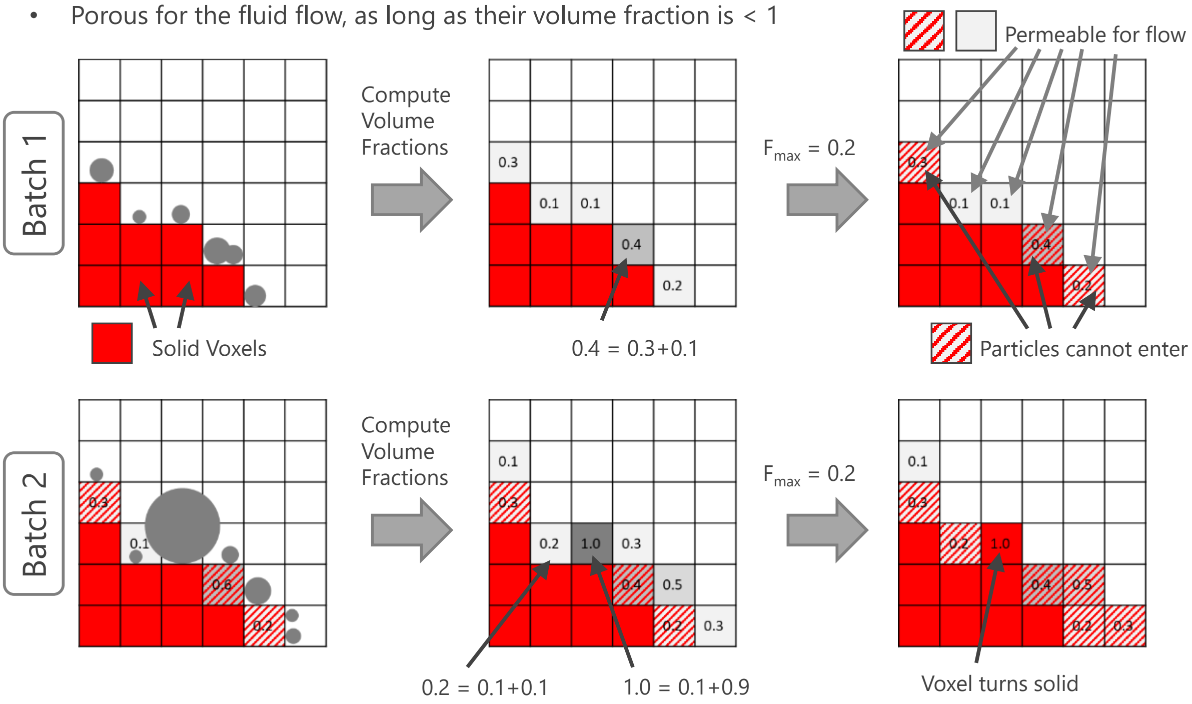


Unresolved Simulations

In unresolved simulations, voxels can be porous. These can be voxels with caught particles or a 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_{max}**.

Unresolved Case: Red voxels are solid, other voxels are treated as:

- Impermeable for particles, if their volume fraction is > F_{max} (**Max. Particle Packing Density**)
- Porous for the fluid flow, as long as their volume fraction is < 1



Therefore, the Navier-Stokes-Brinkman equations (5) and (6) have to be used to determine the flow through the resulting structure.

The flow resistivity $\sigma = \frac{\mu}{\kappa}$ 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, see page 41.

Collision Models for resolved simulations

When a particle touches a filter structure, it can be deposited or continue moving. FilterDict offers three different collision 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

$$v^2 < \frac{H}{4\pi\rho a_0 R^2} \tag{15}$$

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 4·10^{-10}m = 0.4 nm = 4 Ångstrom), and R is the particle radius. See reference [8].

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.

$$\text{Restitution Coefficient } = \frac{v_2}{v_1} \tag{16}$$

Sieving

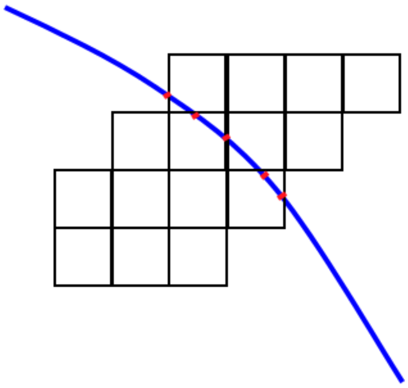
In the **Sieving** model, the particles never stick to the filter, but a particle is caught by the filter if it does not move anymore and touches the filter at two different points. The restitution parameter is used in the same way as in the Hamaker model.

User-Defined Collision Models

Additional collision models can be defined in user defined functions. See the Appendix I (page 91) for details.

Absorption Models for unresolved simulations

In 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. If a particle is filtered depends on the pass-through probability of the passed voxels, and the distance covered in these voxels. 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

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

$$p(l) = \beta^{-\frac{l}{L}} \tag{17}$$

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

$$p(L) = \frac{1}{\beta} \tag{18}$$

and the passing probability of passing the media n times is

$$p(nL) = p(L)^n \tag{19}$$

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

$$E = 1 - \frac{1}{\beta} \tag{20}$$

Thus, the pass though probability is given through

$$p(l) = \beta^{-\frac{l}{L}} = (1 - E)^{\frac{l}{L}} \tag{21}$$

Velocity Dependent efficiency

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

$$p(l) = \left(1 - E(v)\right)^{\frac{l}{L}} \tag{22}$$

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

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 (References [5],[6]).

$$\frac{\partial C}{\partial t} + \vec{u}\nabla C - D\Delta C = -\frac{\partial M}{\partial t} \tag{23}$$

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 $\frac{\partial M}{\partial t}$ 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 (23) can be simplified to

$$u\,\frac{\partial C}{\partial x} = -\frac{\partial M}{\partial t} \tag{24}$$

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 Reference [7]]

$$\frac{\partial M}{\partial t} = \alpha C \tag{25}$$

Where α is the constant absorption rate. The analytic solution can then be derived for Equations (24) and (25):

$$C(x,t) = C_{in}\, e^{-\frac{\alpha}{U}x} \tag{26}$$

$$M(x,t) = \alpha t C_{in}\, e^{-\frac{\alpha}{U}x} \tag{27}$$

Then the β-rating is given as

$$\beta(t) = \frac{C_{in}}{C(L,t)} = e^{\frac{\alpha}{U}L} \tag{28}$$

and the pass-through probability is

$$p(l) = e^{-l\frac{\alpha}{U}} \tag{29}$$

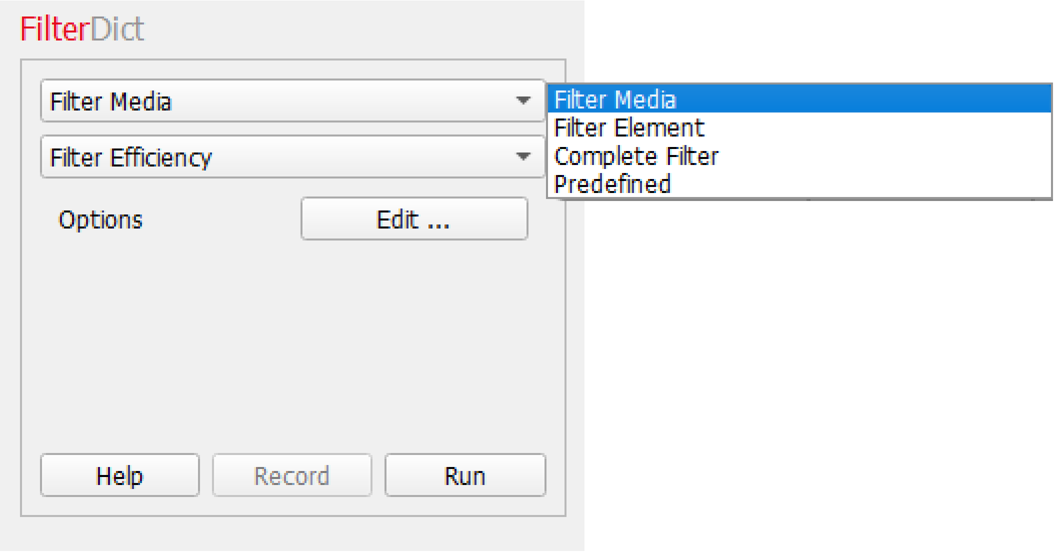
depending on the constant absorption rate α and the velocity.

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 (see page 96).

FilterDict section

FilterDict starts when selecting **Predict → FilterDict** from the menu bar. The module section, at the left of the GeoDict GUI, changes to FilterDict.



A filtration process can be simulated on different scales. Depending on your FilterDict license, up to 4 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.

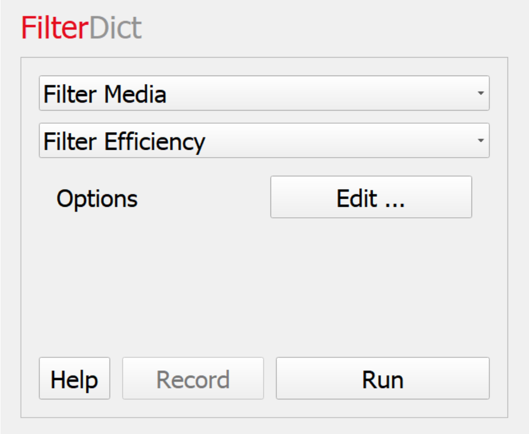
At the **Filter Element** scale, simulate the filtration process on single or multiple pleats generated with PleatGeo, on filter elements, or on DPF (Diesel Particulate Filter) structures.

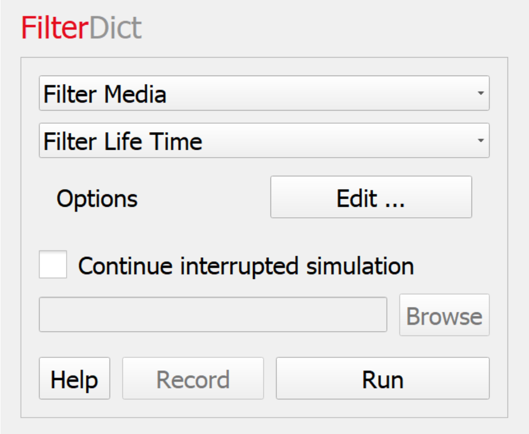
At the **Complete Filter** scale, simulate the filtration process on a CAD model of the complete filter including the filter housing. The CAD model must be imported using ImportGeo-CAD.

When **Predefined** is selected from the upper pull-down menu, the following predefined simulations can be selected: **Complete Filter Flow**, **Cross-Flow Filtration**, and **Cross-Flow**. During the run of these predefined simulations, the corresponding GeoDict macros are called and executed with preset parameters. These parameters and other options can be edited through dialogs (Edit…) or re-scripted (Open Macro/Script File) through a text editor.

Filter Media

Two types of simulations can be selected from the pull-down menu below **Filter Media**: **Filter Efficiency** and **Filter Life Time**:





A **Filter Efficiency** simulation determines the theoretical efficiency of the clean filter and can be used to calculate the most penetrating particle size (MPPS).

A **Filter Life Time** simulation includes clogging of the filter and computes pressure drop and deposited dust over time.

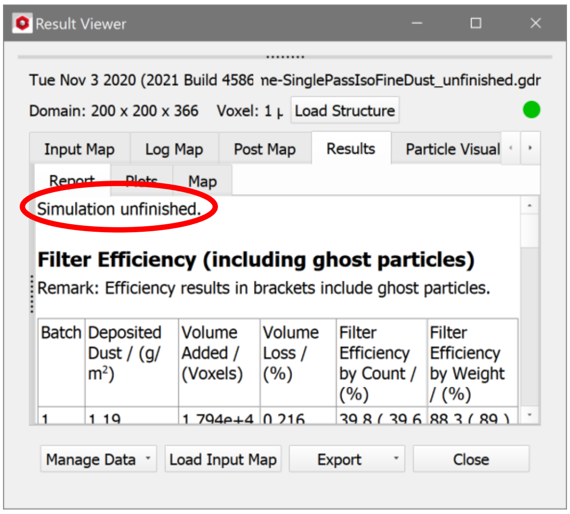
- In a Single Pass simulation, fluids pass through the filter only once and are not recirculated.
- In a Multi Pass simulation, fluids move in a circuit through the system, and particle size distribution and concentration in front of the filter change over time.

For all commands, the input parameters for the selected FilterDict command can be modified through the **Options' Edit...** button. For each type of simulation, a different dialog opens to enter or edit parameters.

When the input parameters for the selected FilterDict simulation have been entered, clicking the **Run** button in the FilterDict section starts the computations. When recording a macro (Macro → Start Macro Recording…) to later automate the simulation run, the **Record** button becomes active and the **Run** button changes to **Run & Record**.

For **Filter Life Time** simulations, a previously interrupted FilterDict simulation can be restarted by checking **Continue interrupted simulation**. For this, at least the computation of the first flow field must be finished. When the simulation was interrupted earlier, it must be started anew.

A cancelled simulation produces an incomplete (*.gdr) result file which is not automatically opened at the end of the computations. When opened through **File → Open *.gdr File...**, the first line under the **Results - Report** subtab indicates that the simulation was not finished.



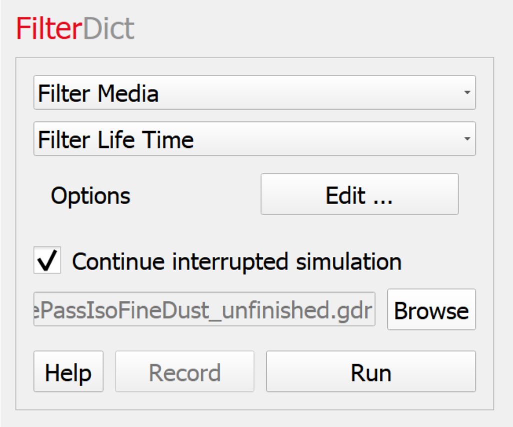
Simulation unfinished.

Filter Efficiency (including ghost particles)

Remark: Efficiency results in brackets include ghost particles.

Batch	Deposited Dust / (g/m²)	Volume Added / (Voxels)	Volume Loss / (%)	Filter Efficiency by Count / (%)	Filter Efficiency by Weight / (%)
1	1.19	1.794e+4	0.216	39.8 (39.6	88.3 (89)

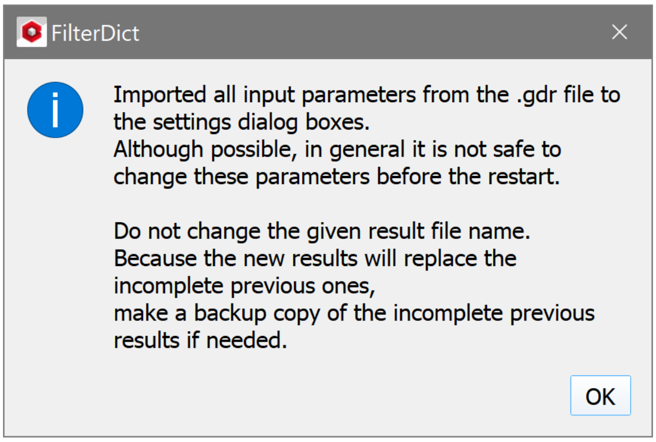
To continue the unfinished simulation, check **Continue interrupted simulation** and click **Browse** to select the name of the result file. The result file from the unfinished simulation must be located in the project folder and the structure on which the simulation is to be finished must be loaded in memory (displayed in the Visualization area).



☑ Continue interrupted simulation

ePassIsoFineDust_unfinished.gdr

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



Imported all input parameters from the .gdr file to the settings dialog boxes.
Although possible, in general it is not safe to change these parameters before the restart.

Do not change the given result file name.
Because the new results will replace the incomplete previous ones,
make a backup copy of the incomplete previous results if needed.

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

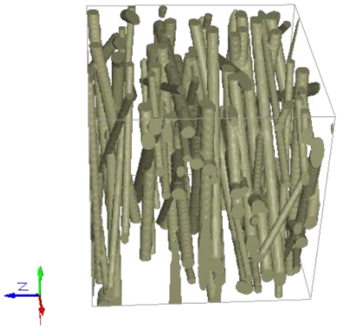
For all **Filter Media** simulations, it is essential that the structure model contains enough free space as inflow region (Z- direction) and outflow region (Z+ direction).

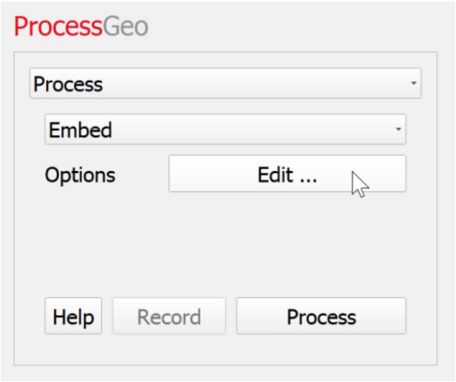
For efficiency simulations, the inflow region should be larger than the largest particle diameter. For filter lifetime simulations, the inflow region should be larger than the largest particle diameter plus the expected height of the filter cake.

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, select **Model** → **ProcessGeo**. In the ProcessGeo section, check **Embed**, enter the appropriate number of voxels, in the Z- and Z+ boxes, and click **Embed**.

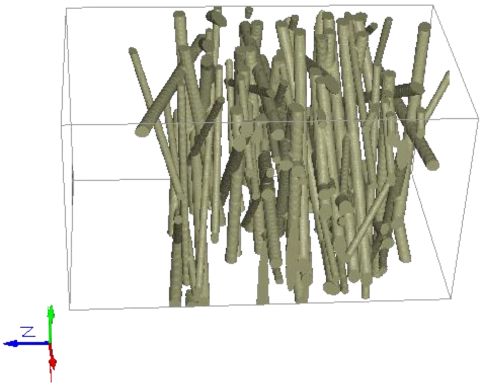
Then, switch back to FilterDict.

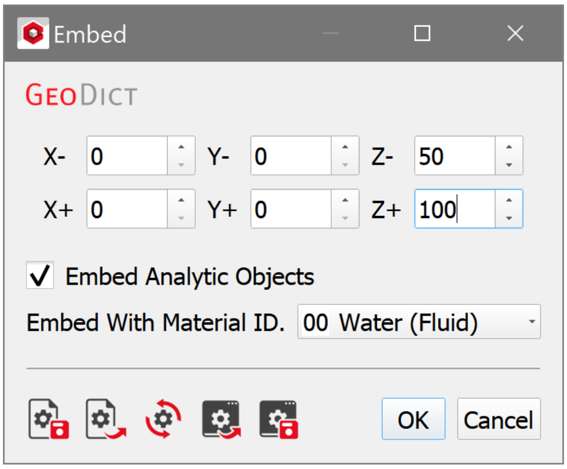



ProcessGeo

Process: Embed

Options: Edit ...




Embed

GeoDict

X-	0	Y-	0	Z-	50
X+	0	Y+	0	Z+	100

☑ Embed Analytic Objects

Embed With Material ID. 00 Water (Fluid)

Filter Efficiency

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

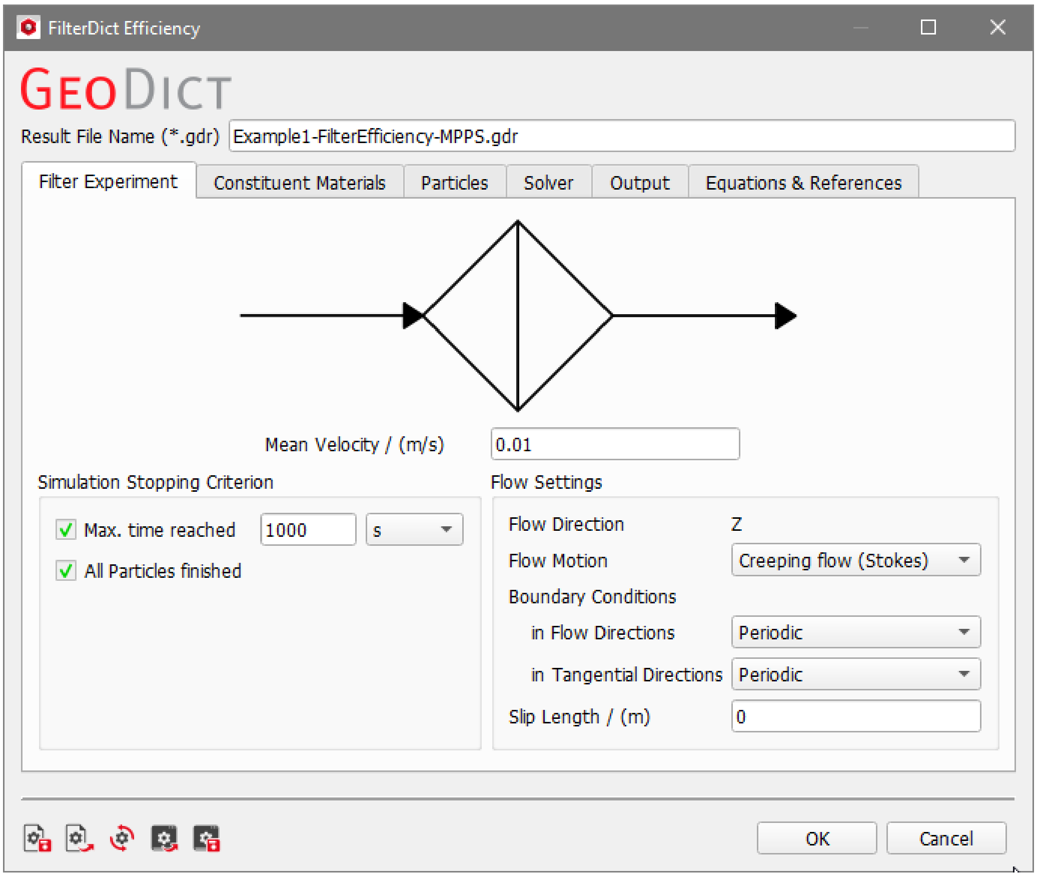
As mentioned above on page 18, it is essential to add inflow and outflow regions to the filter media model before starting a filter efficiency simulation.

After choosing **Filter Efficiency** from the pull-down menu and clicking the **Option**'s **Edit…** button, a dialog opens to enter or edit simulation options. A **Result File Name** for the result file (*.gdr) must be entered, fitting the 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.

Filter Experiment

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



Result File Name (*.gdr) Example1-FilterEfficiency-MPPS.gdr

Mean Velocity / (m/s) 0.01

Simulation Stopping Criterion

☑ Max. time reached 1000 s

☑ All Particles finished

Flow Settings

Flow Direction Z

Flow Motion Creeping flow (Stokes)

Boundary Conditions

in Flow Directions Periodic

in Tangential Directions Periodic

Slip Length / (m) 0

Each particle trajectory is calculated by solving the equations (9) to (13). 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.

In the **Flow Settings** panel, based on the flow velocity and the Reynolds number, the **Flow Motion** regime can be chosen as **Creeping flow (Stokes)** or **Fast flow (Navier-Stokes)**.

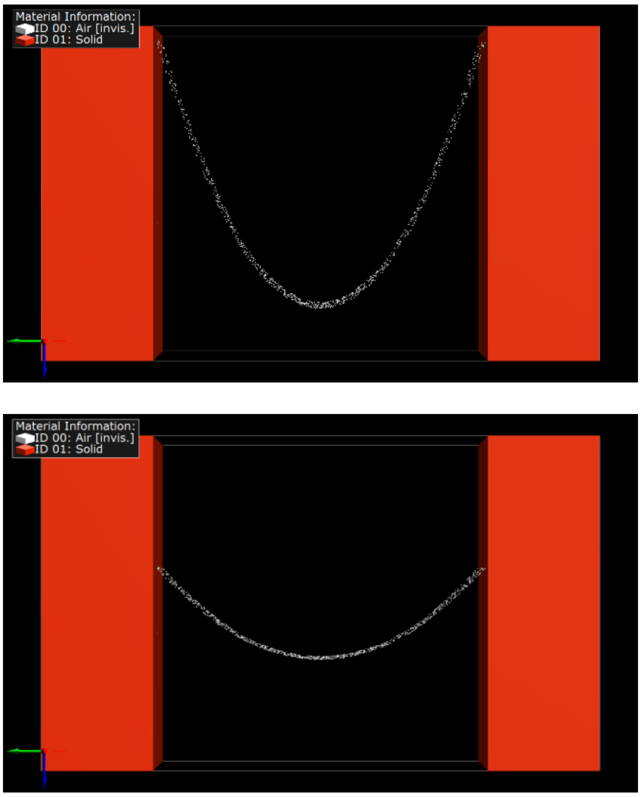
The flow is always directed in Z-direction in FilterDict.

The **Boundary Conditions in Flow Directions** can be selected as **Periodic** or **VinPout**, which refers to velocity at inlet (Vin) and pressure at outlet (Pout). In **Tangential Directions**, **Periodic** or **Symmetric** boundary conditions can be selected. For more information on these boundary conditions, please consult the FlowDict handbook of this 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.

By default, the pore-solid boundary conditions are set to a **Slip Length** of 0, which corresponds to the standard no-slip boundary conditions of the flow. Setting a non-zero **Slip Length** allows to include sliding effects in the flow simulation. Please consult the FlowDict handbook of this User Guide for more information how the slip length is used to compute the flow field.

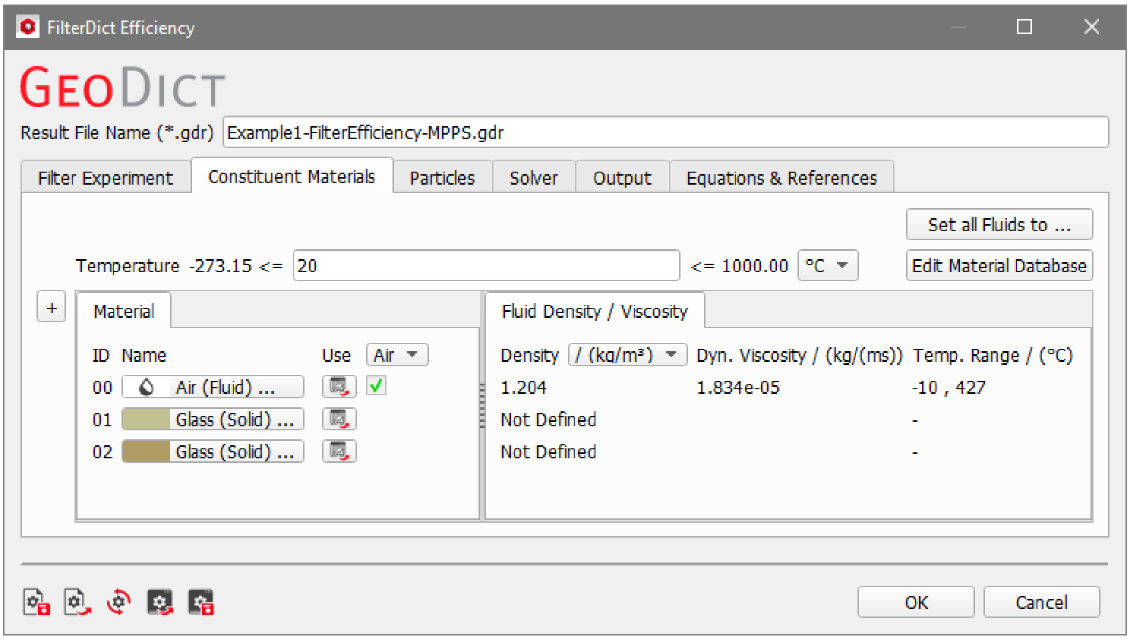
Typically, the slip length is of the same order of magnitude as the mean free path of the fluid (which is e.g. 68 nm for air at ambient conditions) and can be neglected if this is much smaller than the voxel length. It becomes relevant when simulating air flow around nanofibers.



When particles follow the flow, they visualize the Poiseuille profile that appears when using no-slip conditions (left). Using slip conditions (right), also the particles that start close to the wall have moved in the flow direction.

Constituent Materials

The **Temperature** for the filtration process is selectable in Kelvin (K), Celsius [°C], and Fahrenheit (F). The chosen temperature changes the density and viscosity values of the fluid during the simulation, if those are defined to be temperature-dependent in the material database. Furthermore, it determines the strength of the Brownian motion through equation (12).


FilterDict Efficiency
GeoDict
Filter Experiment | Constituent Materials | Particles | Solver | Output | Equations & References
Set all Fluids to ...
Edit Material Database
OK
Cancel

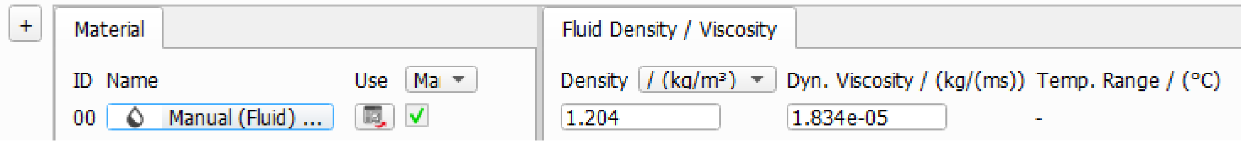
Result File Name (*.gdr) Example1-FilterEfficiency-MPPS.gdr

Temperature -273.15 <= 20 <= 1000.00 °C

ID	Name	Use	Density / (kg/m³)	Dyn. Viscosity / (kg/(ms))	Temp. Range / (°C)
00	Air (Fluid) ...	✓ (Air)	1.204	1.834e-05	-10 , 427
01	Glass (Solid) ...		Not Defined		-
02	Glass (Solid) ...		Not Defined		-

FilterDict uses single-phase flow simulations. It is not possible to run a filtration 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 the fluid for the simulation.

If a fluid from the GeoDict Material Database is used, the values for **Density**, **Dynamic Viscosity**, and **Kinematic Viscosity** are taken from the database and can be dependent on the given **Temperature** value. If **Manual** is chosen, these values can be entered manually:



ID	Name	Use	Density / (kg/m³)	Dyn. Viscosity / (kg/(ms))	Temp. Range / (°C)
00	Manual (Fluid) ...	✓ (Ma)	1.204	1.834e-05	-

To add materials to the database, use the **Edit Material Database** button. More information on editing, expanding, and using the **GeoDict Material Database** is available in the GeoDict Material Database handbook of this User Guide.

The selected fluid is shown under the **Material** subtab. The density and the dynamic viscosity of this fluid, at the selected temperature range, is shown under the **Fluid Density/Viscosity** subtab.

Also, under the **Material** subtab, the material **ID**s of the materials in the structure are shown (00, 01, 02, etc.). Choose materials by clicking on the material buttons.

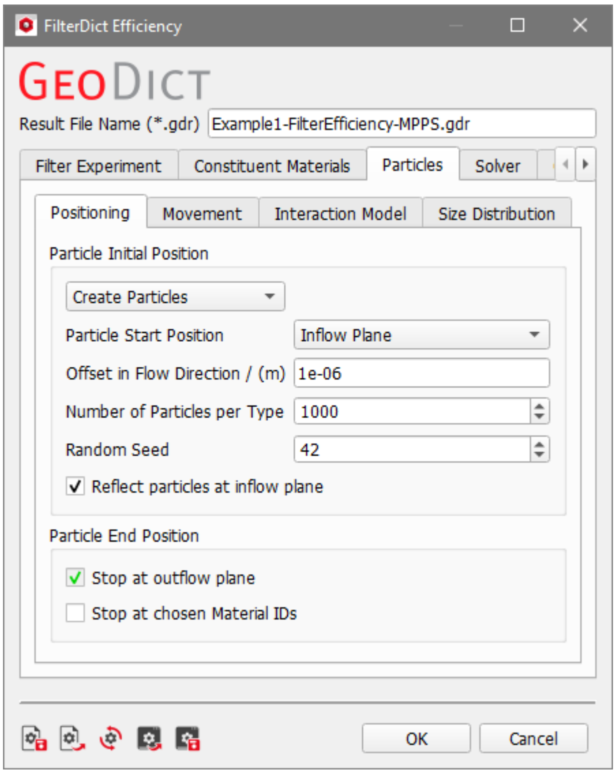
Particles

Under the **Particles** tab, four subtabs are available for **Positioning**, **Movement**, **Interaction Model** and **Size Distribution**.

Positioning

Particle Initial Position

By default, particles are created in **FilterDict** according to a given size distribution. In Filter Efficiency simulations, particles can also be loaded from a file. To do so, choose Load from File from the pull-down menu below Particle Initial Position. If particles are loaded from a file, the input parameters concerning the number and size of particles are not used.


FilterDict Efficiency

GeoDict

Result File Name (*.gdr) Example1-FilterEfficiency-MPPS.gdr

Particle Initial Position

Create Particles

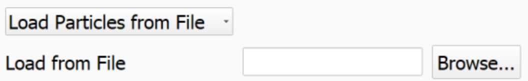
Particle Start Position	Inflow Plane
Offset in Flow Direction / (m)	1e-06
Number of Particles per Type	1000
Random Seed	42

☑ Reflect particles at inflow plane

Particle End Position

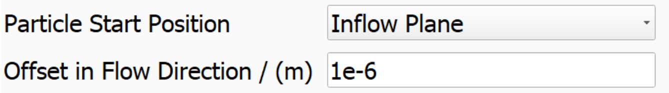
☑ Stop at outflow plane

☐ Stop at chosen Material IDs


Load Particles from File

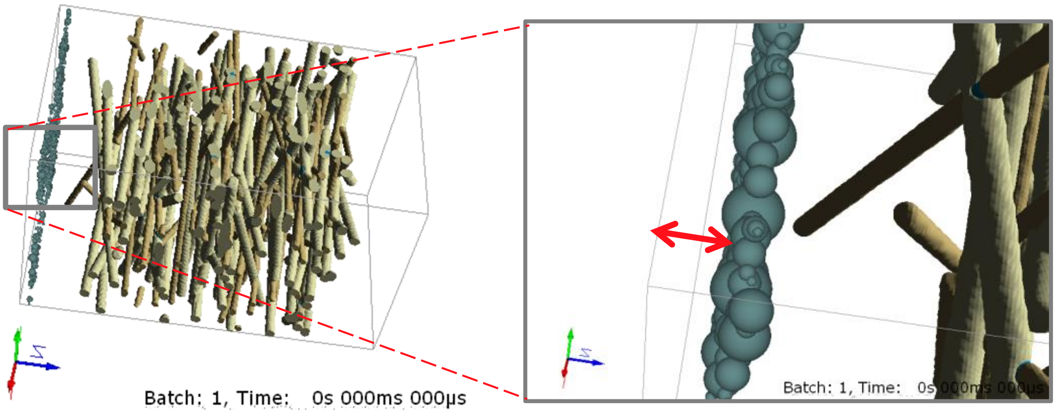
Load from File

The **Particle Start Position** can be defined if the particles are created during the FilterDict simulation. The default start position is the Inflow Plane, but also a **Box**, a **Sphere**, a **Chosen Material IDs** or **Everywhere** in the domain can be selected.

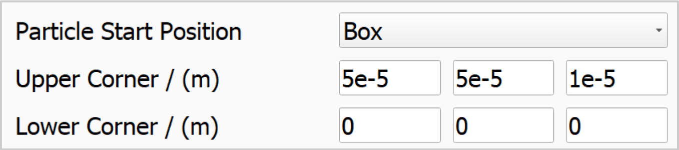


Particle Start Position	Inflow Plane
Offset in Flow Direction / (m)	1e-6

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.

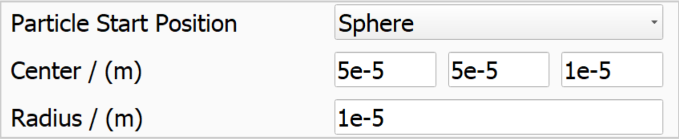

Batch: 1, Time: 0s 000ms 000µs
Batch: 1, Time: 0s 000ms 000µs

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 this box.



Particle Start Position	Box		
Upper Corner / (m)	5e-5	5e-5	1e-5
Lower Corner / (m)	0	0	0

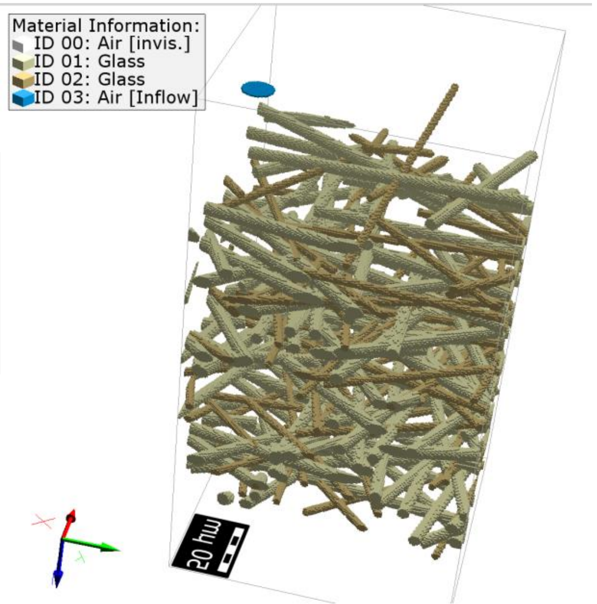
Similarly, if **Sphere** is chosen as **Particle Start Position**, the center and radius of the sphere must be defined.

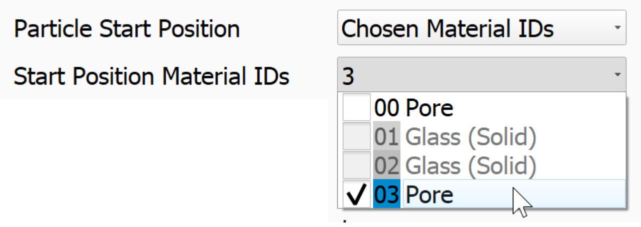


Particle Start Position	Sphere		
Center / (m)	5e-5	5e-5	1e-5
Radius / (m)	1e-5		

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 on the right).



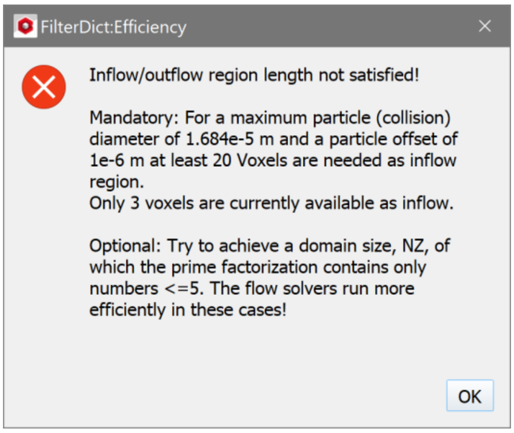


Particle Start Position	Chosen Material IDs
Start Position Material IDs	3

- □ 00 Pore
- □ 01 Glass (Solid)
- □ 02 Glass (Solid)
- ✓ 03 Pore

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.

If the inflow region is too small and 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.



FilterDict:Efficiency

Inflow/outflow region length not satisfied!

Mandatory: For a maximum particle (collision) diameter of 1.684e-5 m and a particle offset of 1e-6 m at least 20 Voxels are needed as inflow region.
Only 3 voxels are currently available as inflow.

Optional: Try to achieve a domain size, NZ, of which the prime factorization contains only numbers <=5. The flow solvers run more efficiently in these cases!

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.

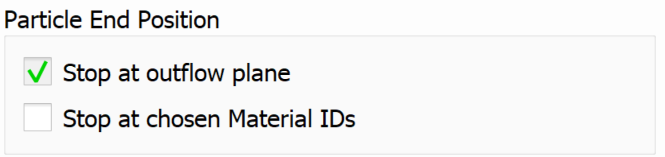
To determine the overall efficiency from the fractional efficiencies, the percentages entered in the particle size distribution table (see page 28) are taken into account.

Particles to be filtered are placed randomly in the area defined above. 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**.

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.



Particle End Position

☑ Stop at outflow plane

□ Stop at chosen Material IDs

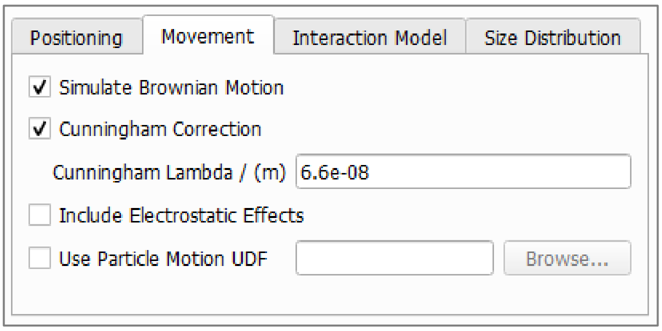
Additionally, and similar to the definition of the particle start position, an additional area can be marked and particles which arrive in those area will also stop moving and are counted as unfiltered.

Movement

The parameters in the **Movement** panel define the simulation variables for the particle motion. See the theory section on page 5 for details.

Checking **Simulate Brownian Motion** activates the simulation of random motion in particle movement.

If **Cunningham Correction** is checked, equation (11) is used to calculate C_c. Otherwise, $C_c = 1$ is used. The **Cunningham Lambda** value is the mean free path λ used in equation (11).



☑ Simulate Brownian Motion

☑ Cunningham Correction

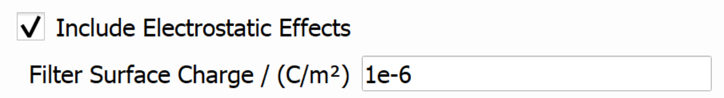
Cunningham Lambda / (m) 6.6e-08

☐ Include Electrostatic Effects

☐ Use Particle Motion UDF

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.

If **Include Electrostatic Effects** is checked, electrostatic effects between filter media and particles in the fluid are simulated. The **Filter Surface Charge** (C/m^2) corresponds to ξ in equation (14)



☑ Include Electrostatic Effects

Filter Surface Charge / (C/m^2) 1e-6

To model electrostatic effects, it is also necessary to set a particle surface charge under the **Interaction Model** subtab as described below.

Particles and filter are assumed to have opposite charges. Entering positive surface charge values for filter and particles leads to the filter attracting the particles. In this case, the filter efficiency is enhanced. Entering opposite charge values (5e^{-7} and -5e^{-7} C/m^2) leads to the filter repulsing the particles and a lower filtration efficiency.

Use Particle Motion UDF

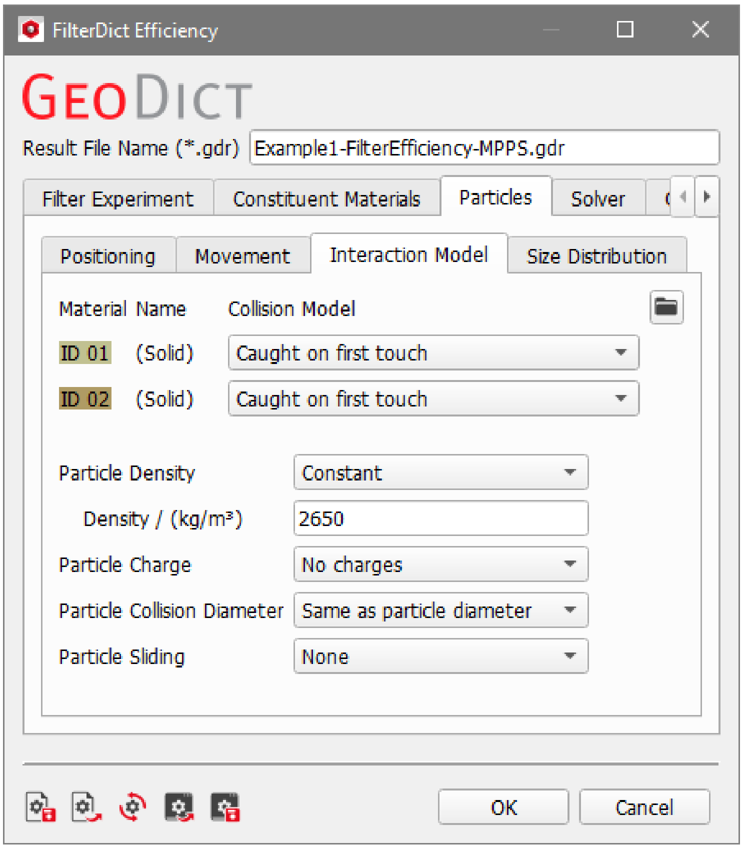
☑ Use Particle Motion UDF

With this option, replace equations (10), (11), and (13) with your own definitions. When **Use Particle Motion UDF** is checked, the values in the (modified) UDF file are used in the FilterDict simulation instead of the built-in solver options.

Refer to the Appendix I for details on how to modify and compile user-defined functions.

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.



Result File Name (*.gdr) Example1-FilterEfficiency-MPPS.gdr

Material Name		Collision Model
ID 01	(Solid)	Caught on first touch
ID 02	(Solid)	Caught on first touch

Particle Density: Constant

Density / (kg/m³): 2650

Particle Charge: No charges

Particle Collision Diameter: Same as particle diameter

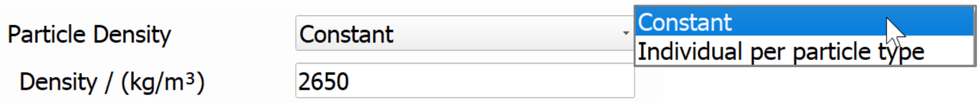
Particle Sliding: None

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 on page 10). 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. See also the Appendix I (page 91) for information on how to compile UDFs.

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

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** subtab, 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

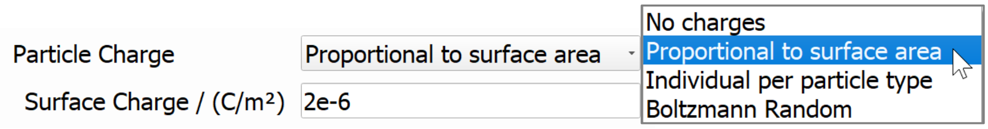


Particle Density: Constant

Density / (kg/m³): 2650

Particle Charge

If electrostatic effects are negligible, the **Particle Surface Charge** can be set to **No charges**.



Particle Charge Proportional to surface area

Surface Charge / (C/m²) 2e-6

Otherwise, if **Proportional to surface area** is chosen, a **Surface Charge** (C/m^2) has to be entered, which defines the value of Q in equation (9) by

$$Q = \text{Particle Surface Charge} \times \text{Particle surface area}$$

If **Individual per particle type** is chosen, a new column called **Particle Charge [C]** is added to the table under the **Size Distribution** subtab, and an individual particle charge can be entered for each particle type or size. In this case

$$Q = \text{Particle Charge}$$

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

$$\sigma = \frac{\sqrt{2\pi d \epsilon_0 kT}}{e} \tag{30}$$

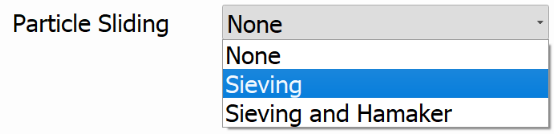
where d denotes the particle diameter, ϵ_0 the dielectric constant, k the Boltzmann contant, T the temperature and e the electron charge.

Particle Collision Diameter

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**. The **Diameter**, as given in the table under the **Size Distribution** subtab, is used when tracking the particle in the flow field (R as used in equations (9) to (14) to determine the mass and the friction.

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 it may happen that the flow moves a particle very close along a pore surface, causing many consecutive hits and thus the repeated application of the restitution factor will cause the particle to stop moving at all. This behavior is often undesired, and it may overestimate the filter efficiency when combining a **Sieving** collision model with a relatively low restitution value.



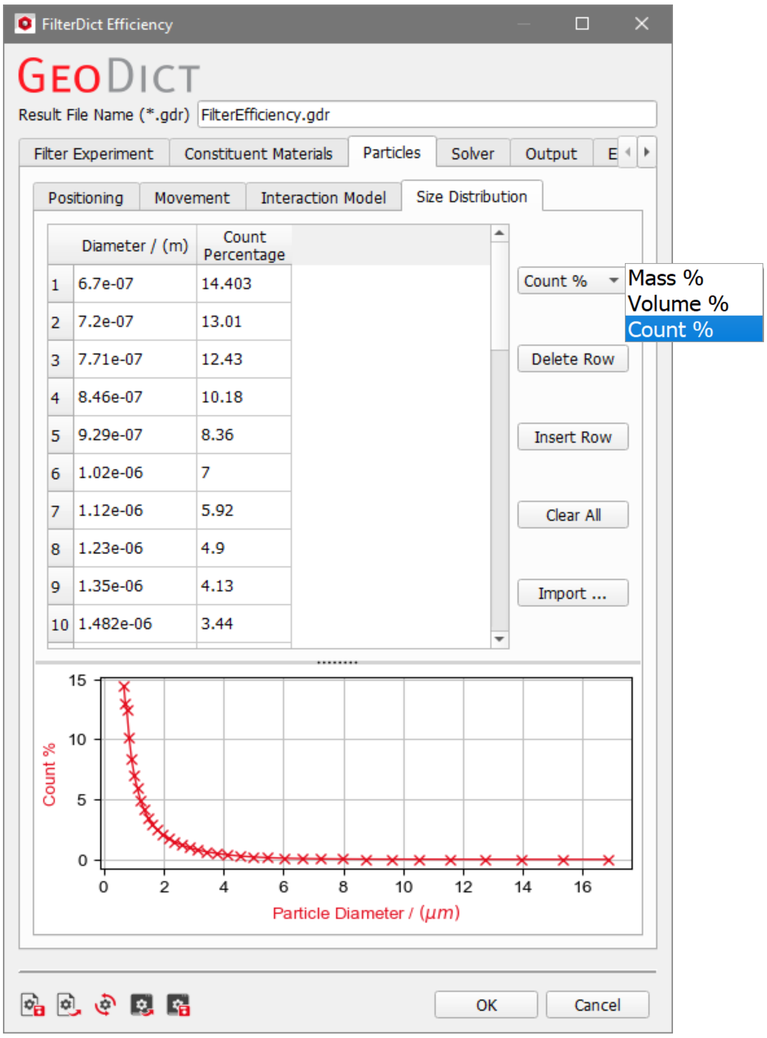
Particle Sliding None

Enabling **Particle Sliding** for **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 normal restitution remains unchanged. In effect, the particle will slide along the surface.

Size Distribution

The **Size Distribution** tab contains a table with the particle size distribution. With the default values selected under the **Interaction Model** subtab, the table contains only two columns.

The left column lists the **Diameter** of particles. The right column contains the particle percentages. The choice of **Mass %**, **Volume %**, or **Count %** is done through the upper right pull-down menu.



Result File Name (*.gdr) FilterEfficiency.gdr

	Diameter / (m)	Count Percentage
1	6.7e-07	14.403
2	7.2e-07	13.01
3	7.71e-07	12.43
4	8.46e-07	10.18
5	9.29e-07	8.36
6	1.02e-06	7
7	1.12e-06	5.92
8	1.23e-06	4.9
9	1.35e-06	4.13
10	1.482e-06	3.44

Additional columns appear depending on the choices made under the **Particle 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.

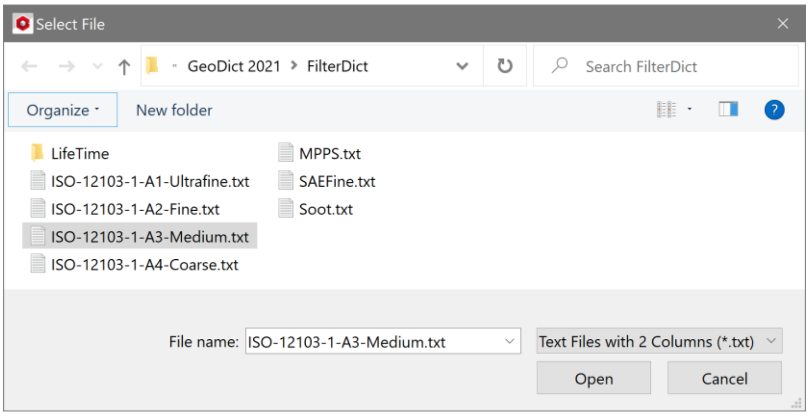
For user-defined collision models, the number of material parameters to be entered in the table is specified in the UDF.

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.

Clicking **Import...** allows 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/GeoDict2022/FilterDict).



File name: ISO-12103-1-A3-Medium.txt

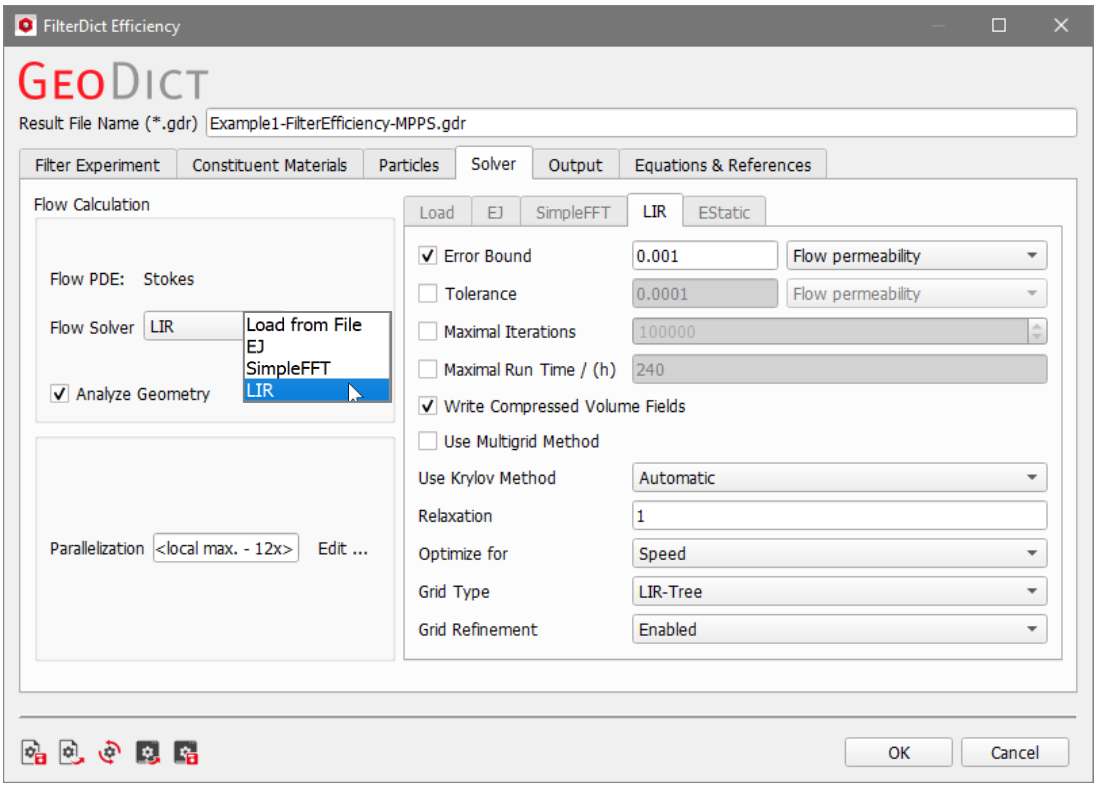
Text Files with 2 Columns (*.txt)

These particle distributions are generated according to the ISO 12103 standard but are not based on measured data. The standard leaves room for interpretation, so these particle size distributions might not match those of real test dusts.

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.

After editing the distribution table, if the new count percentages do not add up to 100%, a warning message appears when clicking **OK**. The values can be automatically normalized by clicking **Yes**, or manually by clicking **Cancel** and returning to the particle distribution table.

Solver



Result File Name (*.gdr) Example1-FilterEfficiency-MPPS.gdr

Flow Calculation

Flow PDE: Stokes

Flow Solver LIR

- Load from File
- EJ
- SimpleFFT
- LIR

☑ Analyze Geometry

Parallelization <local max. - 12x> Edit ...

☑ Error Bound 0.001 Flow permeability

☐ Tolerance 0.0001 Flow permeability

☐ Maximal Iterations 100000

☐ Maximal Run Time / (h) 240

☑ Write Compressed Volume Fields

☐ Use Multigrid Method

Use Krylov Method Automatic

Relaxation 1

Optimize for Speed

Grid Type LIR-Tree

Grid Refinement Enabled

The flow field, which is used to track the particles, can be loaded from a previous flow simulation (**Load from File**) or be computed directly using one of the GeoDict's flow solvers. Depending on the **Flow Motion** type chosen on the Filter Experiment tab, 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.

Load from File

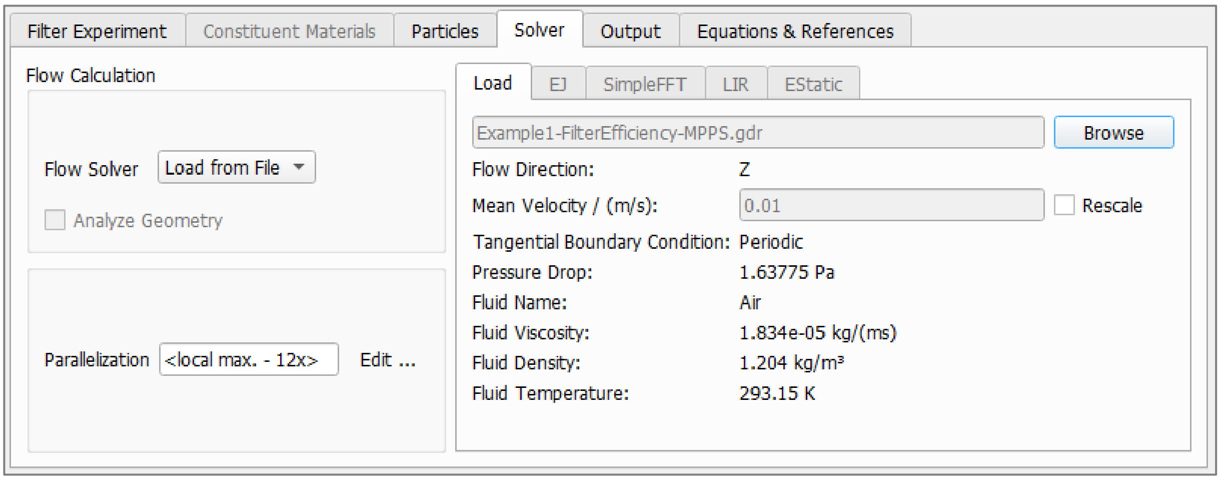
With **Load From File**, a flow field from a previously run flow simulation is used. The flow field may originate from a FlowDict simulation or from a FilterDict simulation.

When calculated with FlowDict, the user needs 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
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.

As mentioned above, depending on the velocity of the flow, the user may have decided to solve the Stokes or the Navier-Stokes equations with the flow solvers when running the simulation (that will be loaded now) in FlowDict.

Under the **Load** tab, click **Browse** and choose a flow field 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.



Flow Solver Load from File

□ Analyze Geometry

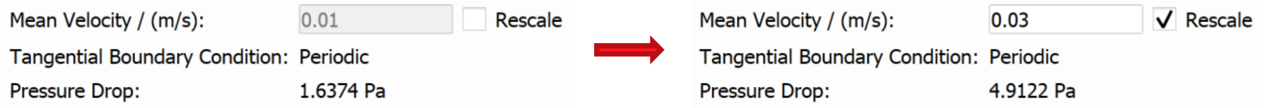
Parallelization <local max. - 12x>

Example1-FilterEfficiency-MPPS.gdr

Flow Direction:	Z
Mean Velocity / (m/s):	0.01 □ Rescale
Tangential Boundary Condition:	Periodic
Pressure Drop:	1.63775 Pa
Fluid Name:	Air
Fluid Viscosity:	1.834e-05 kg/(ms)
Fluid Density:	1.204 kg/m³
Fluid Temperature:	293.15 K

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.



Mean Velocity / (m/s):	0.01 □ Rescale
Tangential Boundary Condition:	Periodic
Pressure Drop:	1.6374 Pa

Mean Velocity / (m/s):	0.03 ☑ Rescale
Tangential Boundary Condition:	Periodic
Pressure Drop:	4.9122 Pa

- The fluid parameters **Fluid Viscosity** [Pa·s], **Fluid Density** [kg/m³], 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.

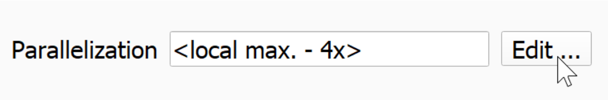
EJ, SimpleFFT, and LIR

In FilterDict, different solvers are available for the computation of the flow field.

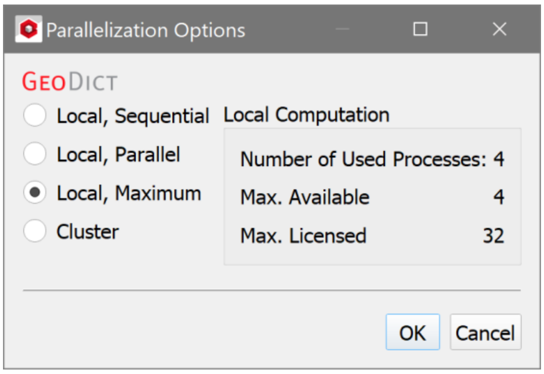
These options are explained in detail in the FlowDict handbook of this User Guide. Be aware that the default accuracy in FilterDict is one order of magnitude higher than in FlowDict (Error Bound, Tolerance and Residual). The stopping criteria depend on global values, whereas the particle movement depends on the local flow field which is subjected to larger deviations.

Parallelization

The chosen parallelization settings apply for all steps of the simulation. Per default, the maximum number of available processes is used, depending on the number of cores on the current computer and the number of licensed processes.



Parallelization <local max. - 4x> Edit ...



○ Local, Sequential
○ Local, Parallel
● Local, Maximum
○ Cluster

Local Computation

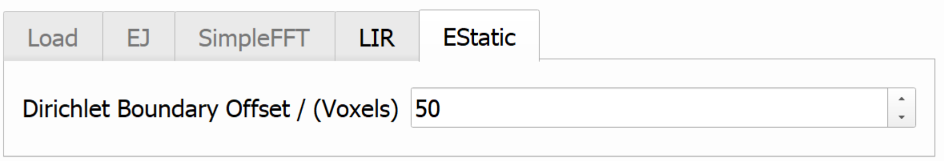
Number of Used Processes: 4
Max. Available 4
Max. Licensed 32

For details on how to set up und run parallel computations, consult the High Performance Computation handbook of this User Guide.

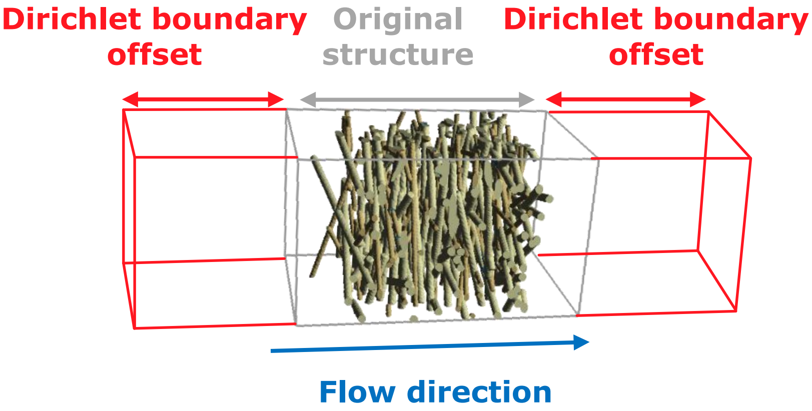
EStatic Solver and Dirichlet Boundary Offset

If **Include Electrostatic Effects** has been checked under the **Particles->Movement** tab, equation (14) 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.

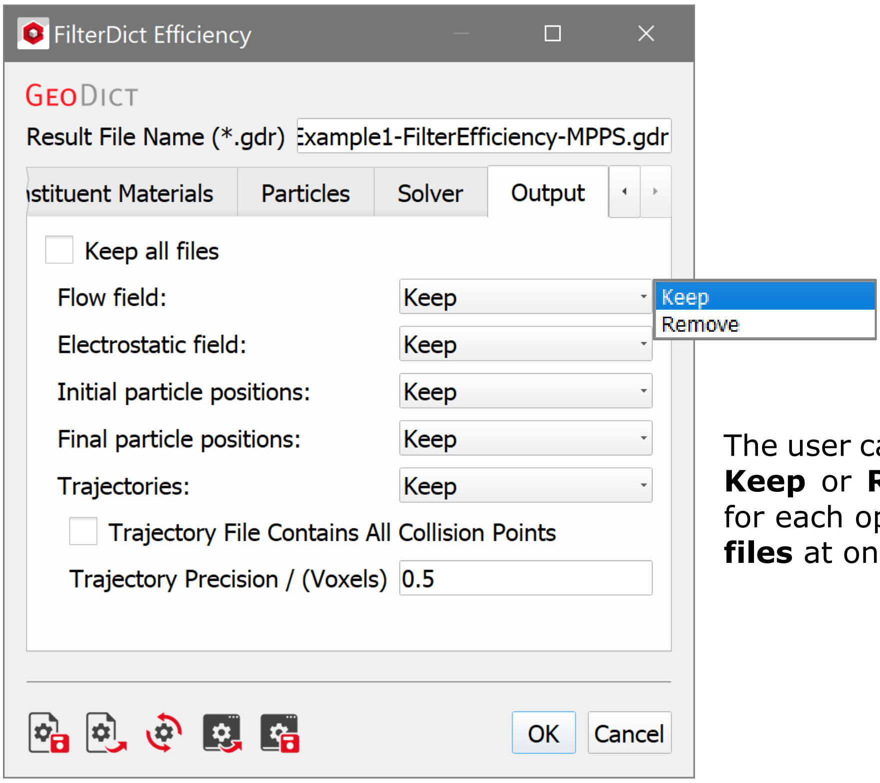


Dirichlet Boundary Offset / (Voxels) 50



Output

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



Result File Name (*.gdr) :xample1-FilterEfficiency-MPPS.gdr

□ Keep all files

Flow field: Keep (Keep / Remove)

Electrostatic field: Keep

Initial particle positions: Keep

Final particle positions: Keep

Trajectories: Keep

□ Trajectory File Contains All Collision Points

Trajectory Precision / (Voxels) 0.5

The user can choose to individually **Keep** or **Remove** the result files for each option, or check **Keep all files** at once.

By selecting to keep **Trajectories**, it is possible to control if particle trajectories are stored and how they are stored**.** The particle trajectories file is needed solely for visualization of the particle movement and may become very large if many particles are tracked.

Additionally, if **Trajectory File Contains All Collision Points** is checked, all points where a particle has touched the surface of a solid object are saved.

The value entered as **Trajectory Precision [Voxels]** determines how accurately the trajectories are stored. The default value of 0.5 voxels is usually adequate. For large structures and/or for large amounts of particles, keeping this value may result in large particle trajectory files (*.gpt). It is then suggested to increase the **Trajectory Precision** value to reduce the trajectory resolution.

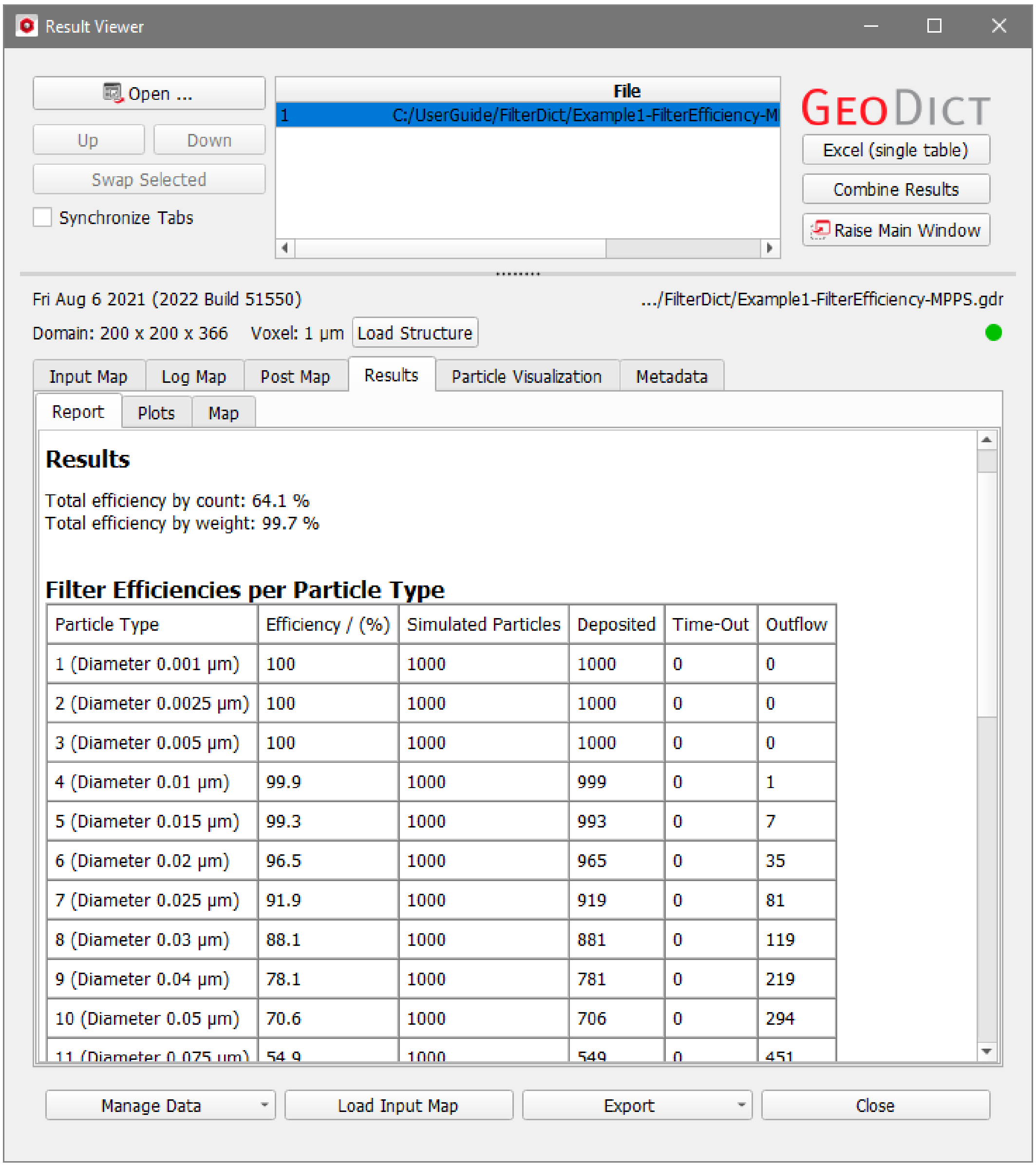
Equations & References

The **Equations & References** tab shows the relevant equations, i.e the Navier-Stokes equations (1)-(2) for fast flows or the Stokes equations (3)-(4) for slow flows. Also, the equations (9)) to (12) governing the particle motion are shown.

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.



1 C:/UserGuide/FilterDict/Example1-FilterEfficiency-M

Fri Aug 6 2021 (2022 Build 51550) .../FilterDict/Example1-FilterEfficiency-MPPS.gdr

Domain: 200 x 200 x 366 Voxel: 1 µm

Results

Total efficiency by count: 64.1 %
Total efficiency by weight: 99.7 %

Filter Efficiencies per Particle Type

Particle Type	Efficiency / (%)	Simulated Particles	Deposited	Time-Out	Outflow
1 (Diameter 0.001 µm)	100	1000	1000	0	0
2 (Diameter 0.0025 µm)	100	1000	1000	0	0
3 (Diameter 0.005 µm)	100	1000	1000	0	0
4 (Diameter 0.01 µm)	99.9	1000	999	0	1
5 (Diameter 0.015 µm)	99.3	1000	993	0	7
6 (Diameter 0.02 µm)	96.5	1000	965	0	35
7 (Diameter 0.025 µm)	91.9	1000	919	0	81
8 (Diameter 0.03 µm)	88.1	1000	881	0	119
9 (Diameter 0.04 µm)	78.1	1000	781	0	219
10 (Diameter 0.05 µm)	70.6	1000	706	0	294
11 (Diameter 0.075 µm)	54.9	1000	549	0	451

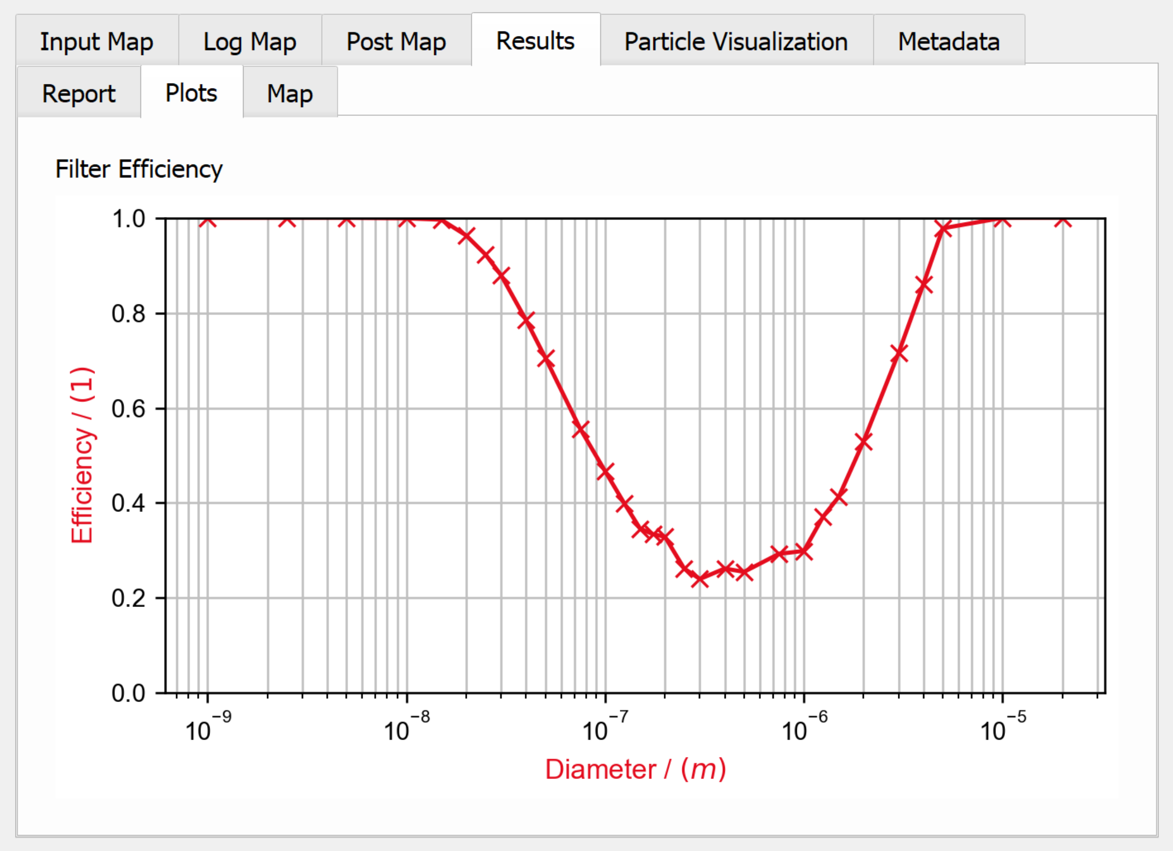
Under the **Report** tab, the following results are reported:

- The **Total efficiency by count**, computed as $\sum_i E_i * Pc_i$, where E_i is the fractional filtration efficiency of a particle type and Pc_i the count probability (**Count %** entered on the **Particles->Size Distribution** tab) of this particle type.
- The **Total efficiency by weight**, computed as $\sum_i E_i * Pm_i$, where E_i is the fractional filtration efficiency of a particle type and Pm_i the mass probability (**Mass %** entered on the **Particles->Size Distribution** tab) of this particle type.

In the table below, for each particle type, the following values are given:

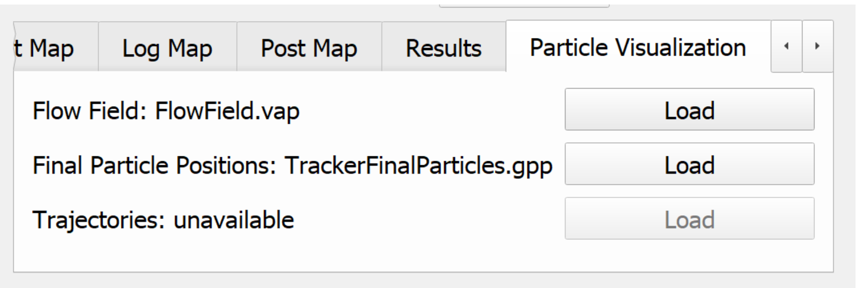
- The fractional filtration **Efficiency.**
- The overall number of **Simulated Particles** of this type. Be aware that those numbers do not follow the size distribution entered on the **Particles->Size Distribution** tab, each particle type is simulated **Number of particles per type** times as entered 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).
- The number of particles that have arrived unfiltered in the outflow area.

On the **Plots** tab, the computed fractional filtration efficiencies are depicted:



Data Visualization

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



Flow Field: FlowField.vap

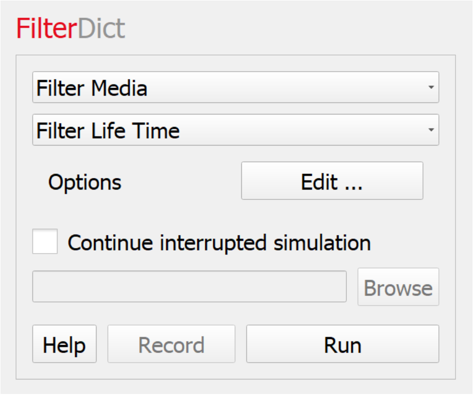
Final Particle Positions: TrackerFinalParticles.gpp

Trajectories: unavailable

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. The options available for the particle visualization are explained in detail in the following section (see page 61).

Filter Life Time

The **Filter Life Time** command simulates either a Single-Pass or a Multi-Pass test.



FilterDict

Filter Media

Filter Life Time

Options: Edit ...

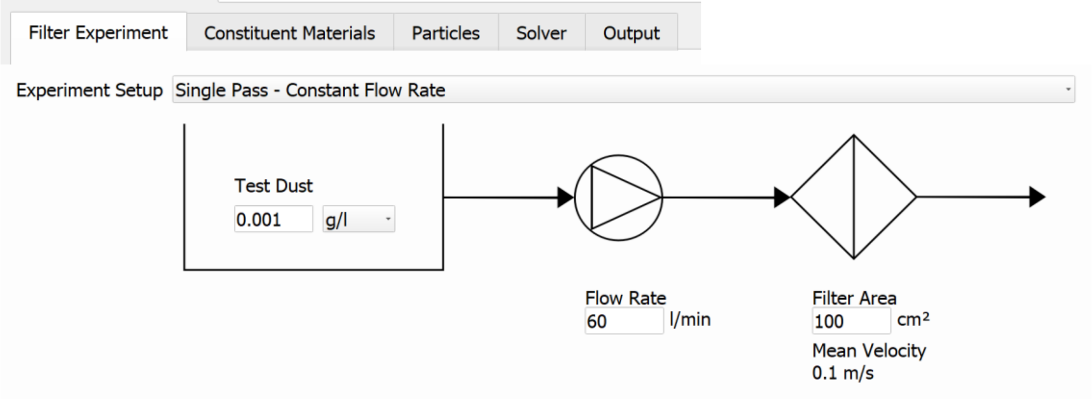
□ Continue interrupted simulation

After clicking the **Options' Edit...** button, the Filter Life Time dialog opens. 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.

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.



Experiment Setup: Single Pass - Constant Flow Rate

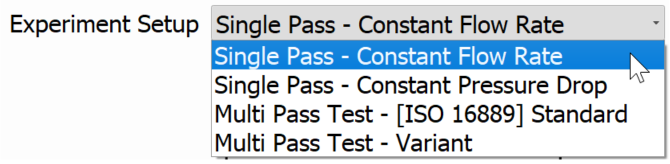
Test Dust: 0.001 g/l

Flow Rate: 60 l/min

Filter Area: 100 cm²

Mean Velocity: 0.1 m/s

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 Life Time dialog.



Experiment Setup: Single Pass - Constant Flow Rate

Single Pass - Constant Flow Rate
Single Pass - Constant Pressure Drop
Multi Pass Test - [ISO 16889] Standard
Multi Pass Test - Variant

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), the **Flow Rate** (in l/min), and the test **Filter Area** (in cm^2).

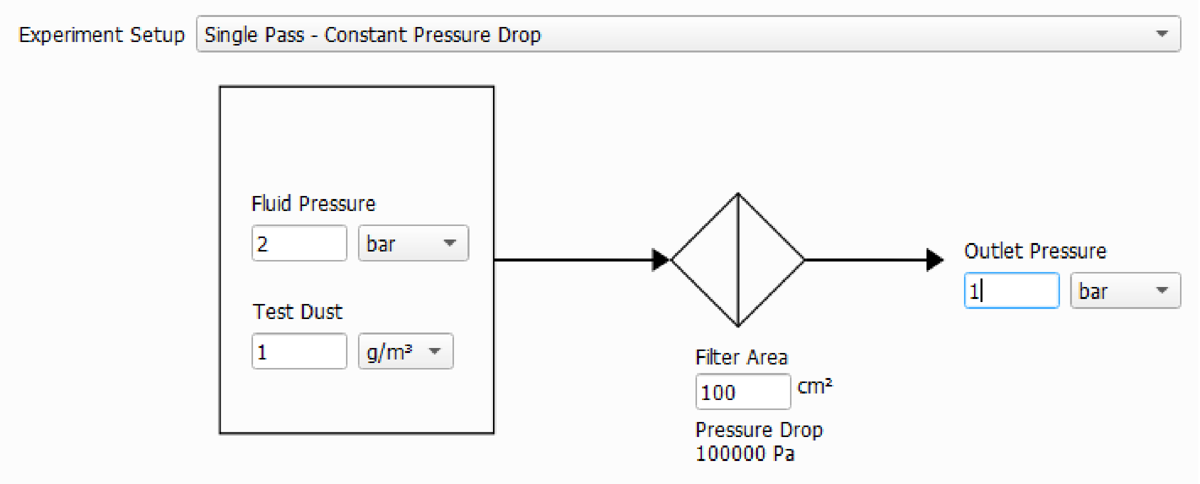
The **Filter Area** is the surface area of the filter in the experiment (and not the surface of the structure in GeoDict). The mean velocity for the simulation is computed from the entered filter area and the flow rate.

The combination of the three parameters (Test Dust concentration, Flow Rate and Filter Area) controls 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**, the **Test Dust** concentration in the fluid, and the test **Filter Area**. As seen above for the constant flow rate, the **Filter Area** is the surface area of the filter in the experiment (and not the surface of the structure in GeoDict).



Experiment Setup: Single Pass - Constant Pressure Drop

Fluid Pressure: 2 bar

Test Dust: 1 g/m³

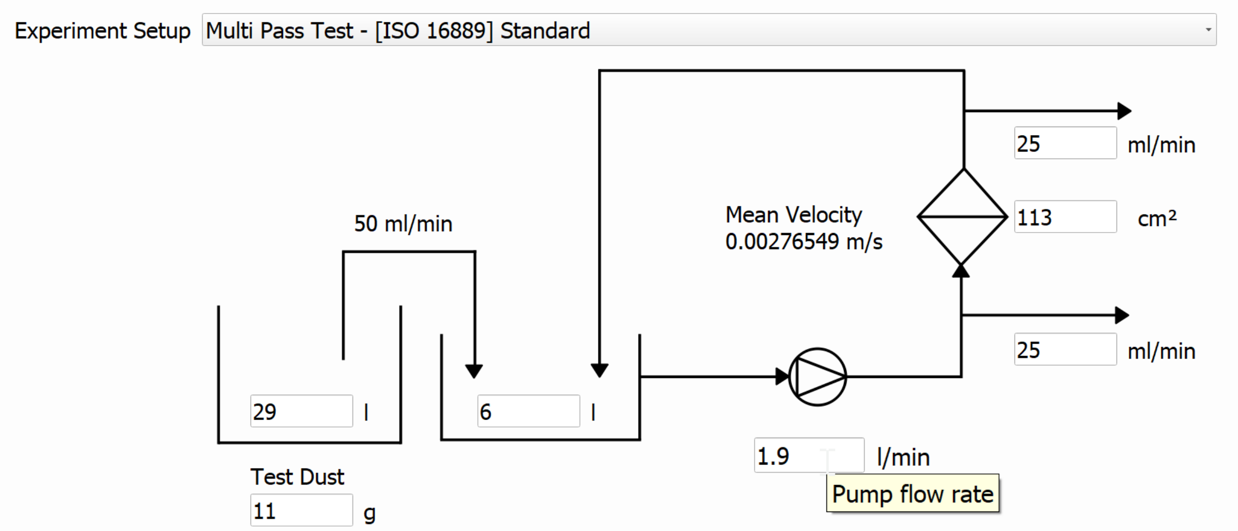
Outlet Pressure: 1 bar

Filter Area: 100 cm²

Pressure Drop
100000 Pa

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.



Experiment Setup: Multi Pass Test - [ISO 16889] Standard

50 ml/min

29 l

6 l

Test Dust: 11 g

Mean Velocity
0.00276549 m/s

25 ml/min

113 cm²

25 ml/min

1.9 l/min

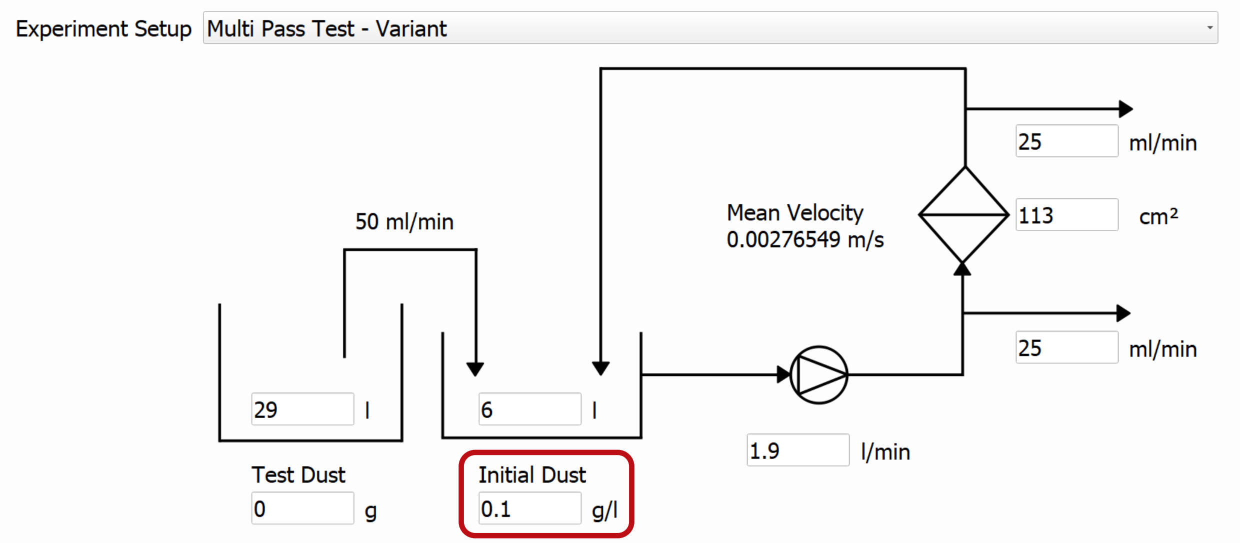
Pump flow rate

Besides the **Test Dust** (in g), the **Pump flow rate** (in l/min), and the **Test filter area** (in cm^2), here 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).

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.



Experiment Setup: Multi Pass Test - Variant

50 ml/min

Mean Velocity 0.00276549 m/s

25 ml/min

113 cm²

25 ml/min

29 l

6 l

1.9 l/min

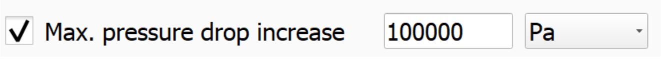
Test Dust: 0 g

Initial Dust: 0.1 g/l

Simulation Stopping Criterion

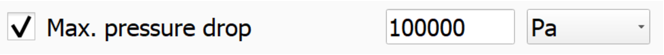
The simulation continues to run until one of the following criteria is reached:

- 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.



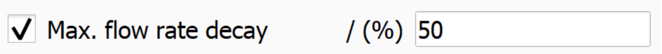
☑ Max. pressure drop increase 100000 Pa

- 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.



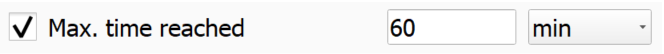
☑ Max. pressure drop 100000 Pa

- 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.



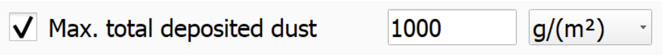
☑ Max. flow rate decay / (%) 50

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



☑ Max. time reached 60 min

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



☑ Max. total deposited dust 1000 g/(m²)

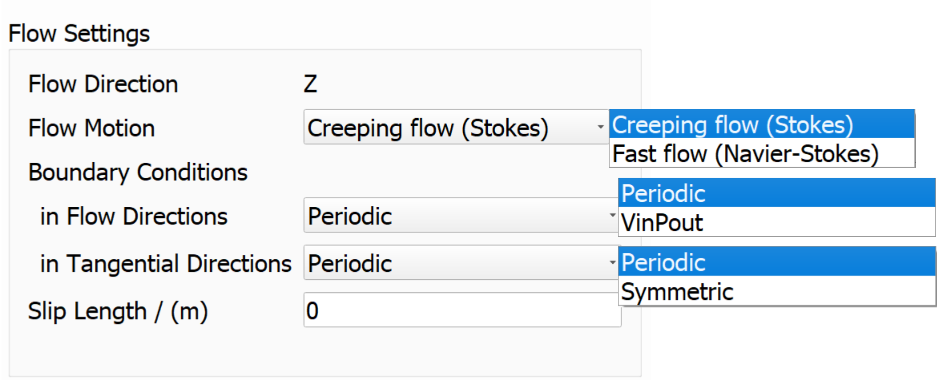
- The simulation always automatically stops when the **Inflow region** is **filled** with particles.

✓ Inflow region filled

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

✓ Injection reservoir empty

Flow Settings



Flow Settings

Flow Direction: Z

Flow Motion: Creeping flow (Stokes)

Boundary Conditions

in Flow Directions: Periodic

in Tangential Directions: Periodic

Slip Length / (m): 0

In FilterDict, the **Flow direction** and, hence the filtration direction, is always the Z-direction, where the particles enter on the Z= 0 plane.

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.

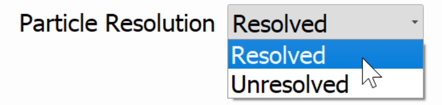
For **Fast Flow (Navier-Stokes)**, the boundary conditions in flow direction are set to Velocity in, Pressure out (**VinPout**) if **Constant Flow Rate** was chosen under the Filter Experiment tab, and set to **Periodic** if **Constant Pressure Drop** was chosen under the Filter Experiment tab.

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

Constituent Materials

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



Particle Resolution: Resolved

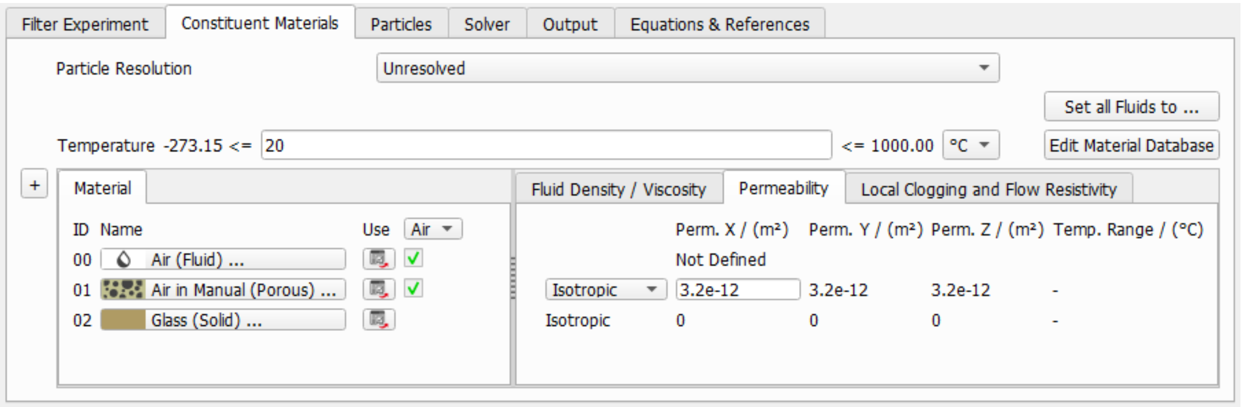
Choose **Resolved** if all particles are larger than the current voxel length, and **Unresolved** if some particles are smaller than the current voxel length. For further explanation, see Resolved and unresolved Simulations, page 8

Define **Temperature** and **Fluid Density/Viscosity** as described for the **Constituent Materials** tab of the Filter Efficiency command on pages 21ff.

Compared to Filter Efficiency, two additional tabs are available for the material properties: **Permeability** and, if the unresolved particle model is chosen, **Local Clogging and Flow Resistivity**.

Permeability

If no porous materials are present in the current 3D structure, no parameters have to be set on this tab. For all porous materials, the flow permeability must be set here. It can either be **Isotropic** or **Anisotropic**, in which case the permeabilities in the three space directions X,Y,Z can be entered.



Particle Resolution: Unresolved

Temperature -273.15 <= 20 <= 1000.00 °C

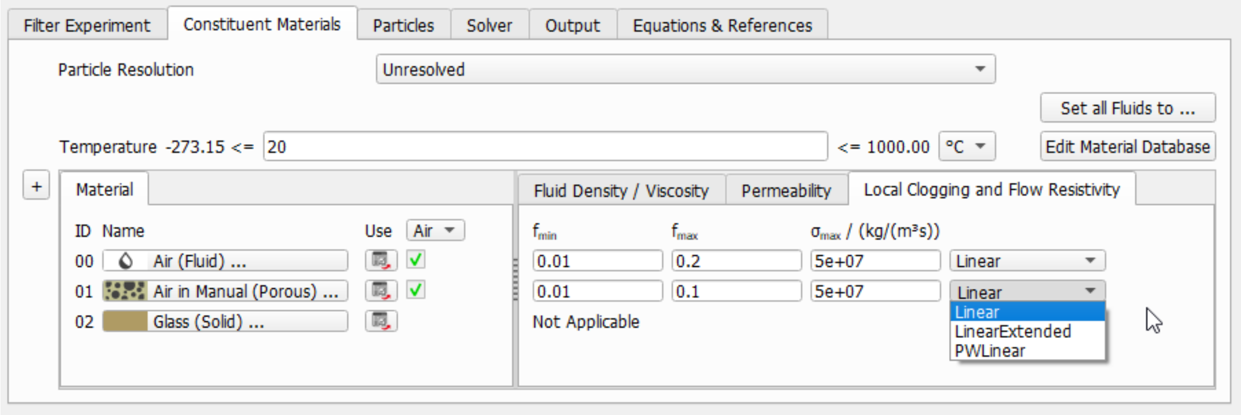
ID	Name	Use	Air
00	Air (Fluid) ...		✓
01	Air in Manual (Porous) ...		✓
02	Glass (Solid) ...		

	Perm. X / (m²)	Perm. Y / (m²)	Perm. Z / (m²)	Temp. Range / (°C)
	Not Defined			
Isotropic	3.2e-12	3.2e-12	3.2e-12	-
Isotropic	0	0	0	-

If porous materials are present in the current 3D structure, the **Use Pass-Through Model** checkbox on the **Particles->Movement** tab must be switched on, too (see page 46). This option enables particle movement through porous layers and particle deposition in such layers.

Local Clogging and Flow Resistivity

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.



Particle Resolution: Unresolved

Temperature -273.15 <= 20 <= 1000.00 °C

ID	Name	Use: Air
00	Air (Fluid) ...	✓
01	Air in Manual (Porous) ...	✓
02	Glass (Solid) ...	

f_{min}	f_{max}	σ_{max} / (kg/(m³s))	
0.01	0.2	5e+07	Linear
0.01	0.1	5e+07	Linear
Not Applicable			

The flow resistivity σ as described on page 8 depends on the values for f_{min}, f_{max} and σ_{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.

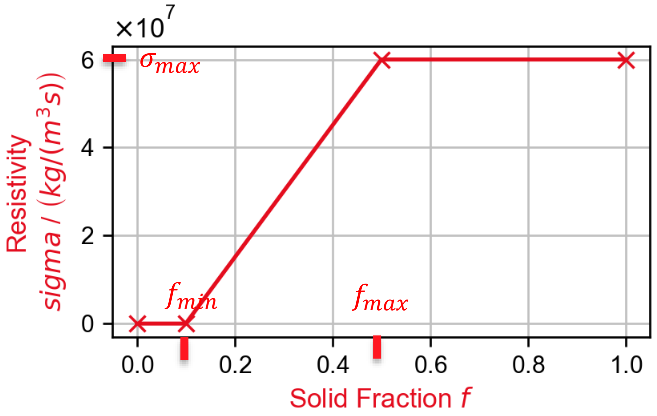
At first glance, setting these parameters for the fluid might seem unintuitive. However, these parameters are actually set for the filter cake because the filter cake forms in the fluid domain.

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 (PWLinear).

Linear

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

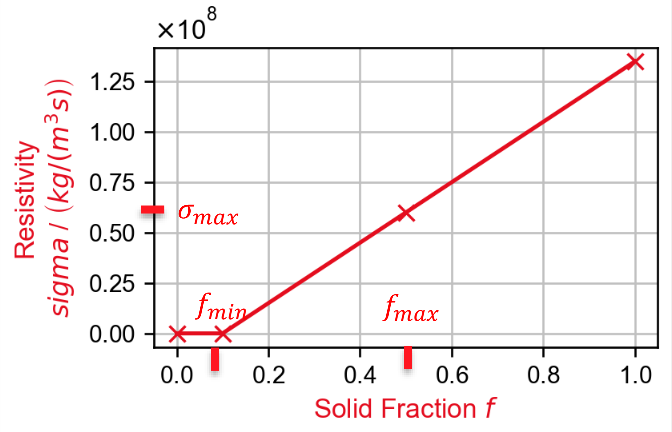


The function $\sigma(f)$ is defined through:

$$\sigma = \begin{cases} 0 & \text{for } f < f_{min} \\ \dfrac{f - f_{min}}{f_{max} - f_{min}}\,\sigma_{max} & \text{for } f_{min} \le f \le f_{max} \\ \sigma_{max} & \text{for } f_{max} < f < 1 \end{cases} \tag{31}$$

Linear Extended

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

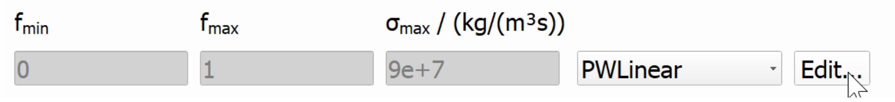


The function $\sigma(f)$ is defined through:

$$\sigma = \begin{cases} 0 & \text{for } f < f_{min} \\ \dfrac{f - f_{min}}{f_{max} - f_{min}} \sigma_{max} & \text{for } f_{min} \le f < 1 \end{cases} \qquad (32)$$

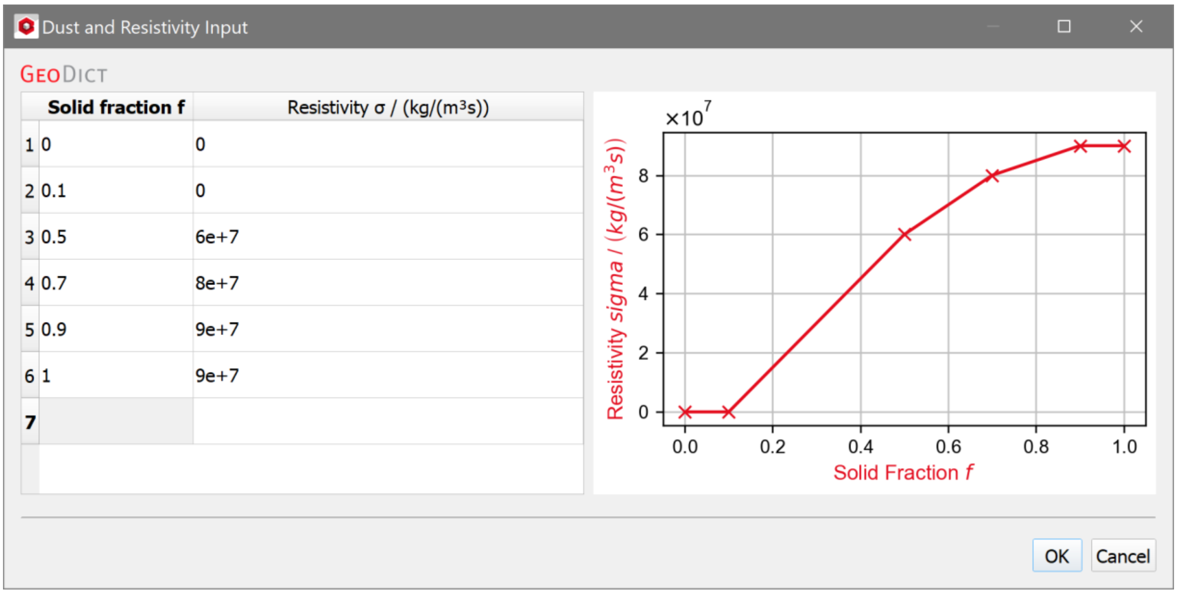
Piecewise Linear (PWLinear)

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



f_{min}	f_{max}	σ_{max} / (kg/(m³s))		
0	1	9e+7	PWLinear	Edit...

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



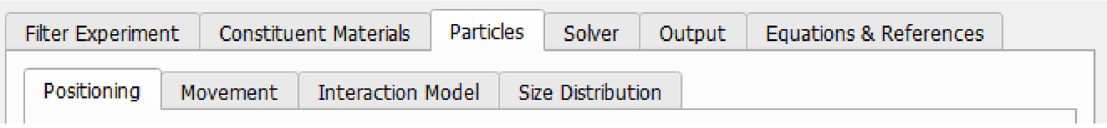
	Solid fraction f	Resistivity σ / (kg/(m³s))
1	0	0
2	0.1	0
3	0.5	6e+7
4	0.7	8e+7
5	0.9	9e+7
6	1	9e+7
7		

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 $f < 1$).

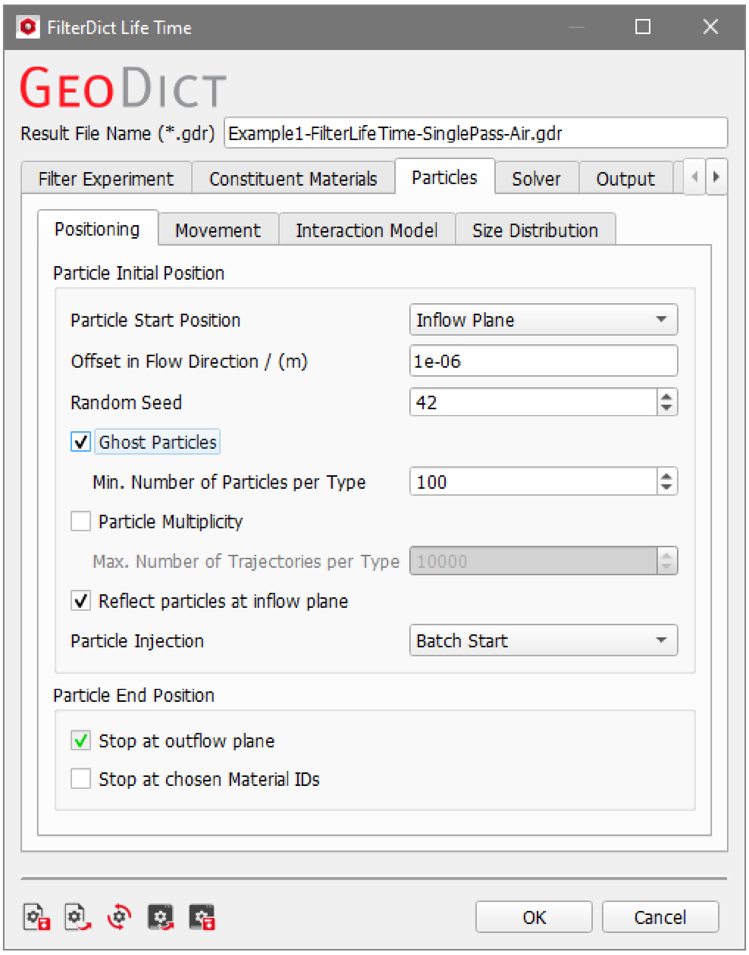
PARTICLES

Under the **Particles** tab, four subtabs are available to define **Positioning**, **Movement**, **Interaction Model** and **Size Distribution**.



POSITIONING

For **Particle Start Position**, **Offset in Flow Direction** and **Random Seed** see pages 22ff.



Result File Name (*.gdr) Example1-FilterLifeTime-SinglePass-Air.gdr

Particle Initial Position

Particle Start Position: Inflow Plane

Offset in Flow Direction / (m): 1e-06

Random Seed: 42

☑ Ghost Particles

Min. Number of Particles per Type: 100

□ Particle Multiplicity

Max. Number of Trajectories per Type: 10000

☑ Reflect particles at inflow plane

Particle Injection: Batch Start

Particle End Position

☑ Stop at outflow plane

□ Stop at chosen Material IDs

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 **Unresolved** 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** (see page 50) 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 the user checks **Particle Multiplicity** and sets **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.

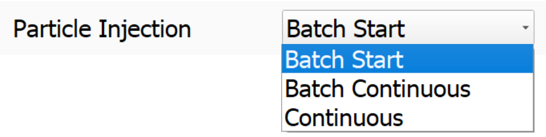
To make sure that the choice of **Particle Multiplicity** does not introduce large numerical errors, FilterDict ensures that the particle volume multiplied by its multiplicity is less than 5% of the single voxel volume.

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**).

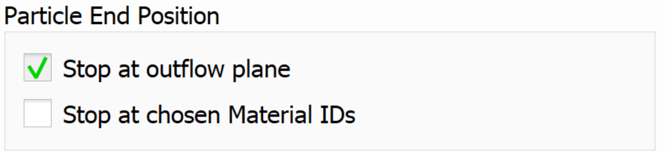

Particle Injection
Batch Start

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.

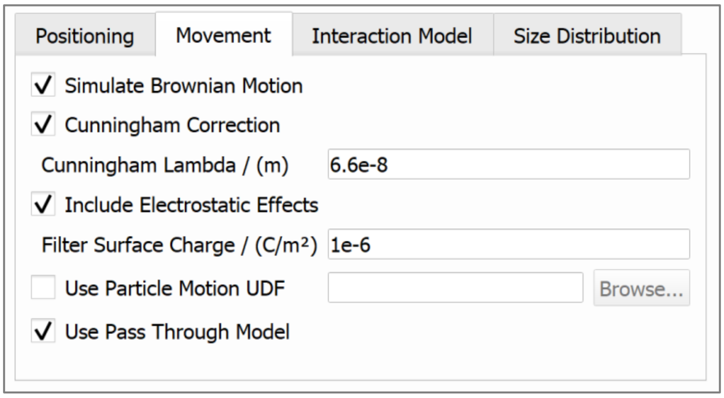


Particle End Position

☑ Stop at outflow plane
☐ Stop at chosen Material IDs

Additionally, and similar to the definition of the particle start position, an additional area can be marked and particles which arrive in those area will also stop moving and are counted as unfiltered.

Movement



☑ Simulate Brownian Motion
☑ Cunningham Correction
Cunningham Lambda / (m) 6.6e-8
☑ Include Electrostatic Effects
Filter Surface Charge / (C/m²) 1e-6
☐ Use Particle Motion UDF
☑ Use Pass Through Model

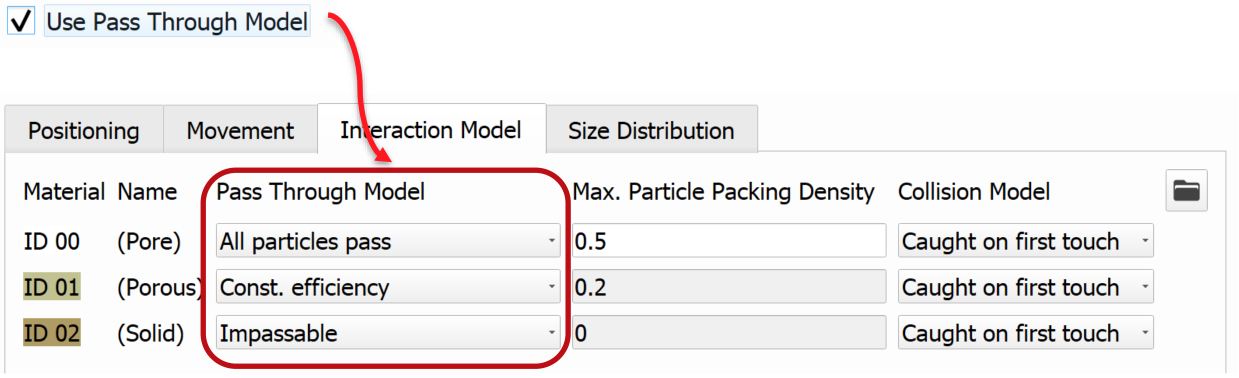
For **Simulate Brownian Motion**, **Cunningham Correction**, and usage of the **Particle Motion UDF** see pages 25ff.

As described on pages 25ff., electrostatic forces can be switched on or off and the parameter **Filter Surface Charge** can be given.

Use Pass Through Model

If porous materials are present in the current 3D structure, the **Use Pass-Through Model** checkbox on the **Particles->Movement** tab must be switched on. 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** subtab.



☑ Use Pass Through Model

Material	Name	Pass Through Model	Max. Particle Packing Density	Collision Model
ID 00	(Pore)	All particles pass	0.5	Caught on first touch
ID 01	(Porous)	Const. efficiency	0.2	Caught on first touch
ID 02	(Solid)	Impassable	0	Caught on first touch

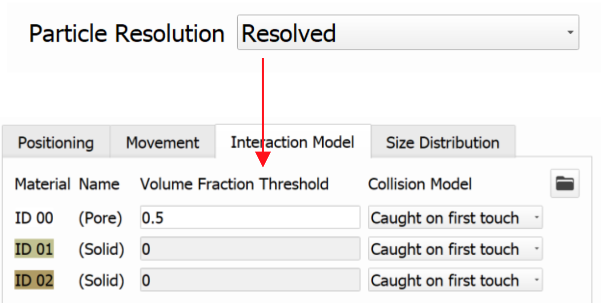
Interaction Model

For Filter Life Time simulations, a **Collision Model** must be chosen for the collision with solid materials, but also for the collision with particles that have been deposited in previous batches (the filter cake). Those particles are deposited in the pore space, therefore the **Collison 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.

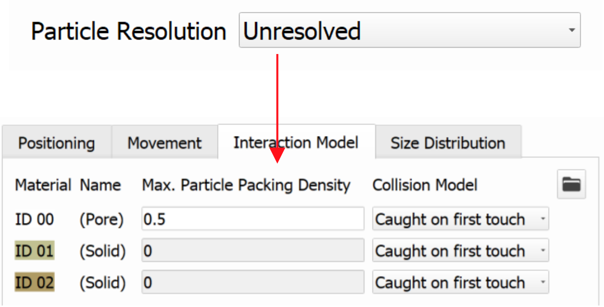
In **Unresolved** simulations, every voxel that exceeds the **Max. Particle Packing Density** is treated as impassable and arriving particles will collide with it using the defined collision model.

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



Particle Resolution: Resolved

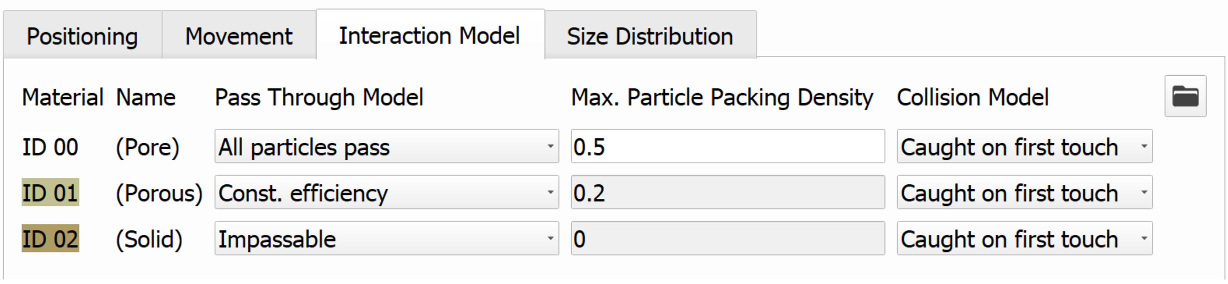
Material	Name	Volume Fraction Threshold	Collision Model
ID 00	(Pore)	0.5	Caught on first touch
ID 01	(Solid)	0	Caught on first touch
ID 02	(Solid)	0	Caught on first touch



Particle Resolution: Unresolved

Material	Name	Max. Particle Packing Density	Collision Model
ID 00	(Pore)	0.5	Caught on first touch
ID 01	(Solid)	0	Caught on first touch
ID 02	(Solid)	0	Caught on first touch

If **Use Pass Through Model** is selected, a **Pass Through Model** must be selected for the pore space and all porous materials:



Material	Name	Pass Through Model	Max. Particle Packing Density	Collision Model
ID 00	(Pore)	All particles pass	0.5	Caught on first touch
ID 01	(Porous)	Const. efficiency	0.2	Caught on first touch
ID 02	(Solid)	Impassable	0	Caught on first touch

See the theoretical background on page 12 for more information on the selectable absorption (or pass through) models **All Particles Pass, Constant Efficiency, Velocity-Dependent Efficiency,** and **Constant Absorption Rate**.

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.

As explained in the paragraph about the constituent materials (page 40), multiple pore materials can be defined for a structure. This allows to use different packing densities models in different zones of the structure.

Particle Density, Particle Charge, Particle Collision Diameter and Particle Sliding

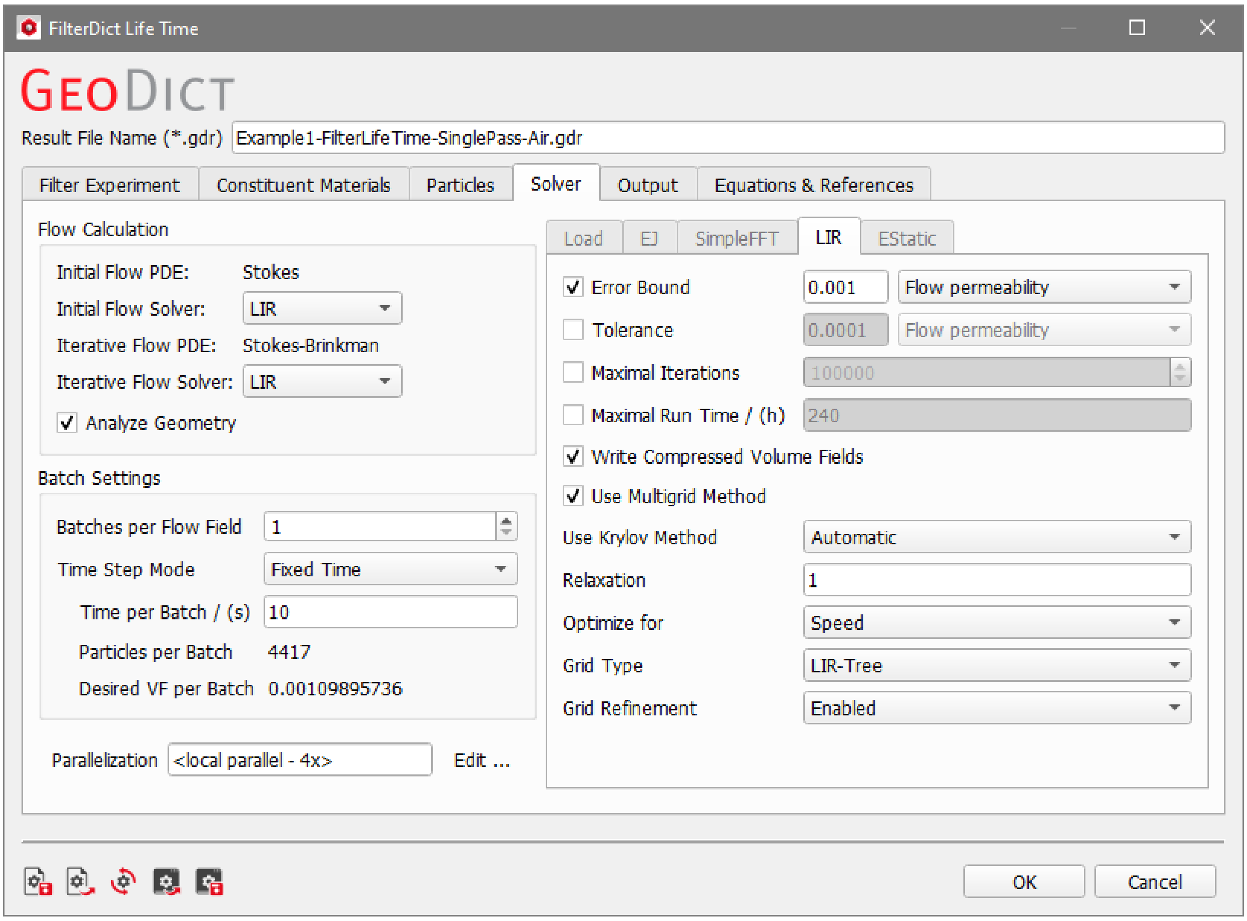
These parameters have the same meaning as in the Filter Efficiency command and are explained on page 26.

Size Distribution

The options under the **Size Distribution** tab are the same as seen on pages 28ff.

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 greyed out and not selectable.



Result File Name (*.gdr): Example1-FilterLifeTime-SinglePass-Air.gdr

Flow Calculation

Initial Flow PDE: Stokes

Initial Flow Solver: LIR

Iterative Flow PDE: Stokes-Brinkman

Iterative Flow Solver: LIR

☑ Analyze Geometry

Batch Settings

Batches per Flow Field: 1

Time Step Mode: Fixed Time

Time per Batch / (s): 10

Particles per Batch: 4417

Desired VF per Batch: 0.00109895736

Parallelization: <local parallel - 4x> Edit ...

☑ Error Bound: 0.001 Flow permeability

☐ Tolerance: 0.0001 Flow permeability

☐ Maximal Iterations: 100000

☐ Maximal Run Time / (h): 240

☑ Write Compressed Volume Fields

☑ Use Multigrid Method

Use Krylov Method: Automatic

Relaxation: 1

Optimize for: Speed

Grid Type: LIR-Tree

Grid Refinement: Enabled

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 / Unresolved particles in the **Constituent Materials** tab and the choice of Creeping / Fast Flow in **Flow Motion** (page 40) under the **Filter Experiment** tab.

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 alternative also be loaded from a previous FilterDict or FlowDict computation.

Tooltips helps 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.

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 2nd or 3rd batch.

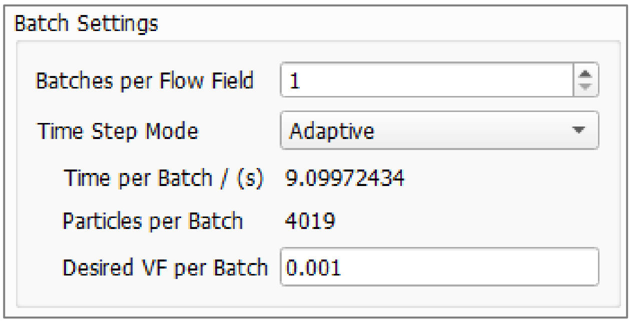
However, when more than one batch per flow field is chosen, the accuracy of the result decreases. In this case, numerical artifacts might be introduced. The most accurate results are obtained by computing one batch per flow field.

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 one of the **Time Step Mode**s 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.



Batch Settings

Batches per Flow Field: 1

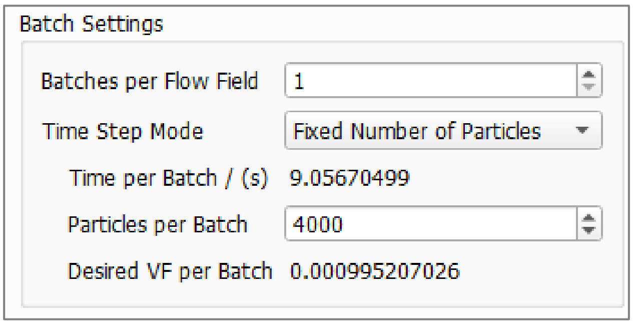
Time Step Mode: Adaptive

Time per Batch / (s): 9.09972434

Particles per Batch: 4019

Desired VF per Batch: 0.001

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



Batch Settings

Batches per Flow Field: 1

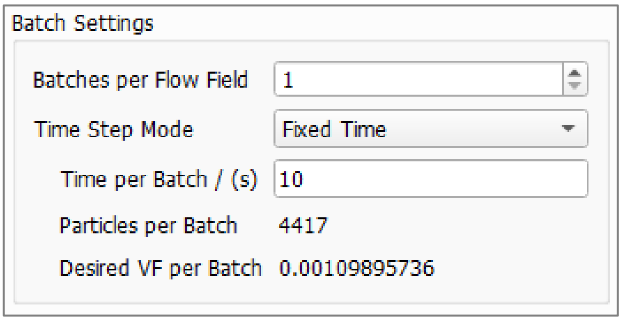
Time Step Mode: Fixed Number of Particles

Time per Batch / (s): 9.05670499

Particles per Batch: 4000

Desired VF per Batch: 0.000995207026

- **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.



Batch Settings

Batches per Flow Field: 1

Time Step Mode: Fixed Time

Time per Batch / (s): 10

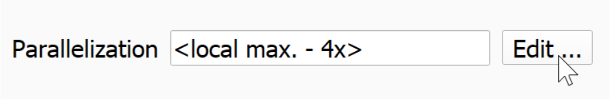
Particles per Batch: 4417

Desired VF per Batch: 0.00109895736

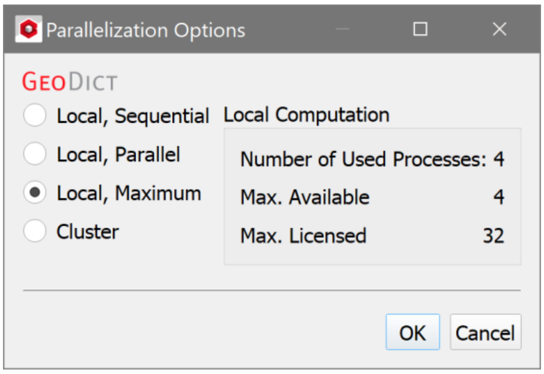
For multi pass simulations, **Fixed Number** or **Adaptive** should be preferred over **Fixed Time**. The particle concentration in the fluid changes over time. Therefore, also the number of particles could change significantly with **Fixed Time** steps and this might lead to a less stable simulation.

PARALLELIZATION

The chosen parallelization settings apply for all steps of the simulation. Per default, the maximum number of available processes is used, depending on the number of cores on the current computer and the number of licensed processes.



Parallelization: <local max. - 4x>



○ Local, Sequential

○ Local, Parallel

● Local, Maximum

○ Cluster

Local Computation

Number of Used Processes: 4

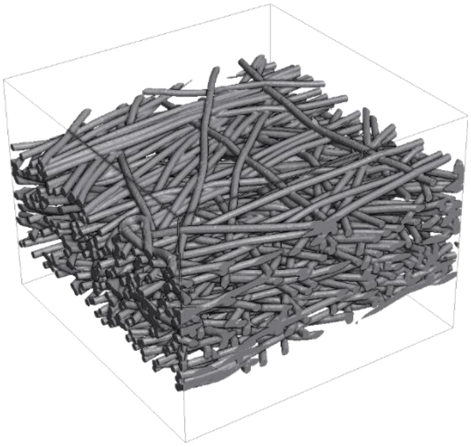
Max. Available: 4

Max. Licensed: 32

For details on how to set up und run parallel computations, refer to the High Performance Computations handbook of this User Guide.

Speed-up through parallelization

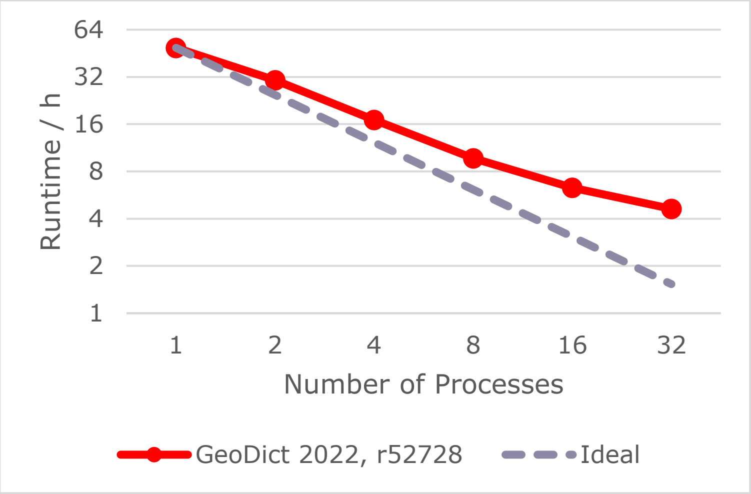
An example of a parallelization benchmark for FilterDict is shown here.



The 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 life time 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 is computed with the LIR solver.

The following figure 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

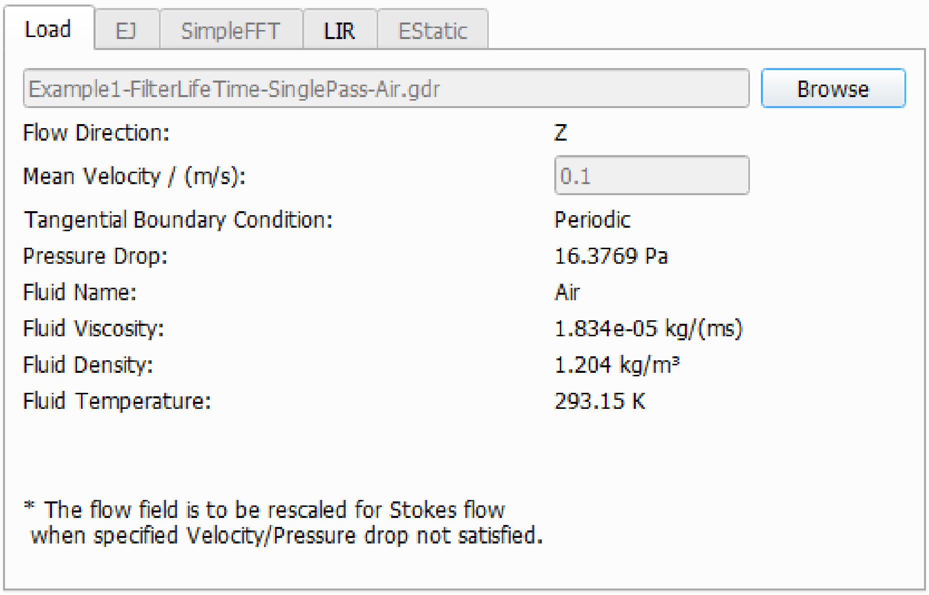
With **Load from File**, a flow field from a previously run flow simulation is used. The flow field may originate from a FlowDict simulation or from a FilterDict simulation.

When calculated with FlowDict, the user needs 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
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.

As mentioned above, depending on the velocity of the flow, the user may have decided to solve the Stokes or the Navier-Stokes equations with the flow solvers when running the simulation (that will be loaded now) in FlowDict.

Under the **Load** tab, click **Browse** and choose a flow field 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.



Example1-FilterLifeTime-SinglePass-Air.gdr

Flow Direction:	Z
Mean Velocity / (m/s):	0.1
Tangential Boundary Condition:	Periodic
Pressure Drop:	16.3769 Pa
Fluid Name:	Air
Fluid Viscosity:	1.834e-05 kg/(ms)
Fluid Density:	1.204 kg/m³
Fluid Temperature:	293.15 K

* The flow field is to be rescaled for Stokes flow when specified Velocity/Pressure drop not satisfied.

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** [Pa·s], **Fluid Density** [kg/m³], 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** tab 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, SimpleFFT and LIR

The settings for the chosen solver are explained in detail in the FlowDict handbook of this User Guide.

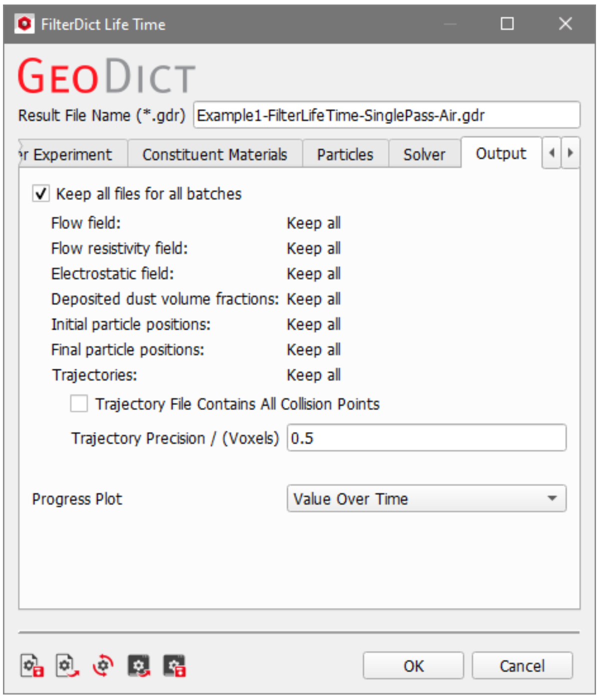
EStatic

The **EStatic** tab is only selectable when **Include Electrostatic Effects** has been checked previously under the **Model** tab. See page 32 for a description of the **Dirichlet Boundary Offset**.

Output

When **Keep all files for all batches** 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 produced.

If **Keep all files for all batches** is unchecked, individual settings can be selected.



Result File Name (*.gdr) Example1-FilterLifeTime-SinglePass-Air.gdr

☑ Keep all files for all batches

Flow field: Keep all
Flow resistivity field: Keep all
Electrostatic field: Keep all
Deposited dust volume fractions: Keep all
Initial particle positions: Keep all
Final particle positions: Keep all
Trajectories: Keep all

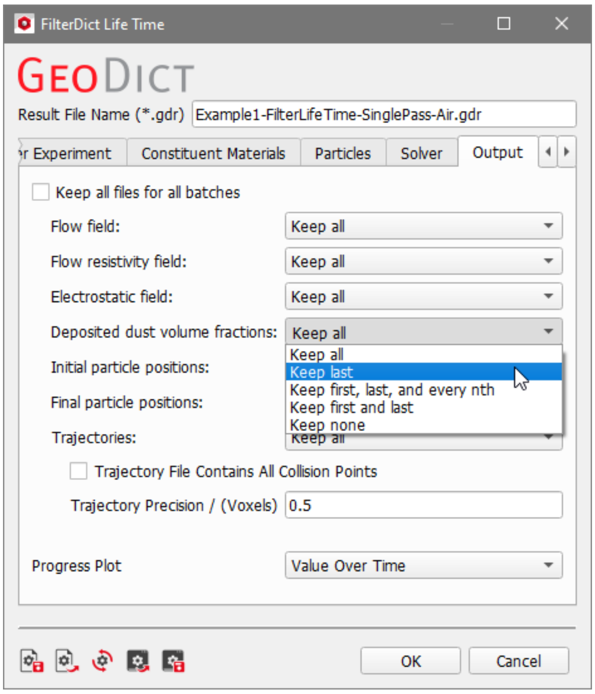
□ Trajectory File Contains All Collision Points

Trajectory Precision / (Voxels) 0.5

Progress Plot: Value Over Time

In all cases, the user has the choice between **Keep all**, **Keep last**, **Keep first, last and every nth**, **Keep first and last** and **Keep none.**.

The **Save interval n** must be selected in case where **Keep first, last and every nth** was chosen.



Result File Name (*.gdr) Example1-FilterLifeTime-SinglePass-Air.gdr

□ Keep all files for all batches

Flow field: Keep all
Flow resistivity field: Keep all
Electrostatic field: Keep all
Deposited dust volume fractions: Keep all
- Keep all
- Keep last
- Keep first, last, and every nth
- Keep first and last
- Keep none

Initial particle positions:
Final particle positions:
Trajectories: Keep all

□ Trajectory File Contains All Collision Points

Trajectory Precision / (Voxels) 0.5

Progress Plot: Value Over Time

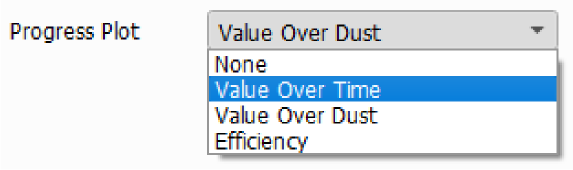
All these files are needed only for visualization of the results. The choices made here do not influence the filter lifetime results contained in the GDR (result) file but may disable some of the visualization options which require to load additional data. In any case, GeoDict keeps all files necessary to continue a simulation that was interrupted.

Trajectory files are needed solely for visualization of the particle movement and may become very large if many particles are tracked.

If **Trajectory File Contains All Collision Points** is checked, the information of all points where a particle has touched the pore surface to a solid object are saved as part of the particle trajectory information.

The value entered as **Trajectory Precision / (Voxels)** determines how accurate the trajectories are stored. The default value of 0.5 voxels is usually adequate, but for large structures and/or for large amount of filtering particles, keeping this value may result in large particle trajectory files (*.gpt). In this case, the suggestion is to increase the **Trajectory Precision** value to reduce the trajectory resolution.

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.



Progress Plot: Value Over Dust

EQUATIONS & REFERENCES

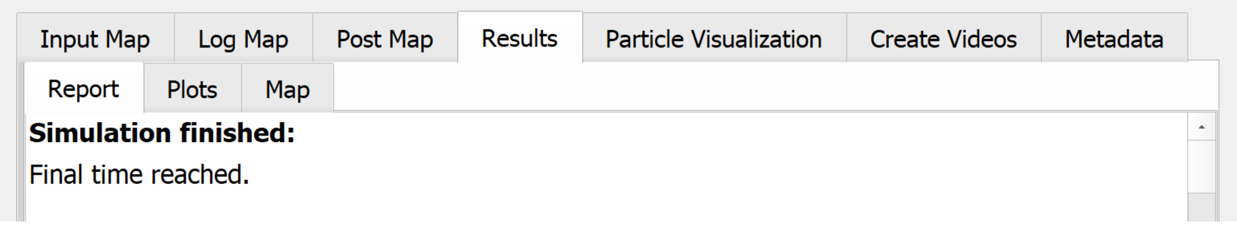
The **Equations & References** tab shows the relevant equations, i.e the Navier-Stokes equations (1)-(2) for fast flows or the Stokes equations (3)-(4) for slow flows. Also, the equations (9)) to (12) governing the particle motion are Furthermore, equation (17) used to model the movement of particles through porous media is shown here, too.

Results

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

Report

At the top of the **Report**, the stopping criteria reached by the simulation is shown.



Simulation finished:

Final time reached.

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

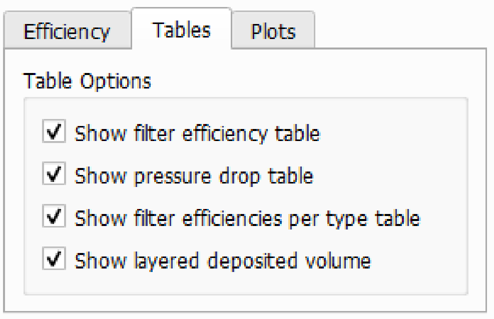


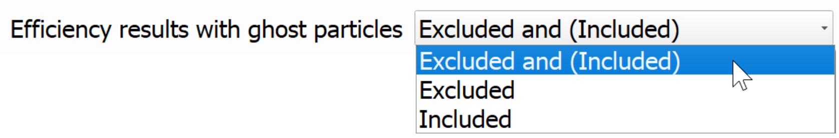
Table Options

- ☑ Show filter efficiency table
- ☑ Show pressure drop table
- ☑ Show filter efficiencies per type table
- ☑ Show layered deposited volume

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**.

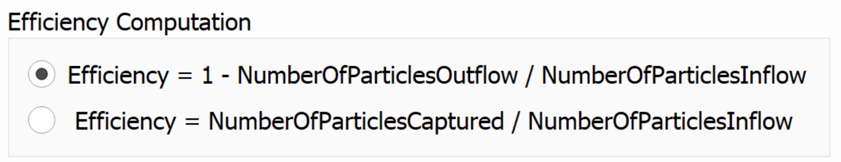
For the computation of the efficiency, **Efficiency results with ghost particles**, controls how the results with and without ghost particles are displayed in the tables. Selecting **Excluded and (Included)**, the tables contain the efficiency without ghost particles and additionally the efficiency with ghost particles in brackets. The other two options, **Excluded** or **Included**, show either the result with or without ghost particles.



Efficiency results with ghost particles: Excluded and (Included)

This option is only available if the user selected to run the simulation with ghost particles.

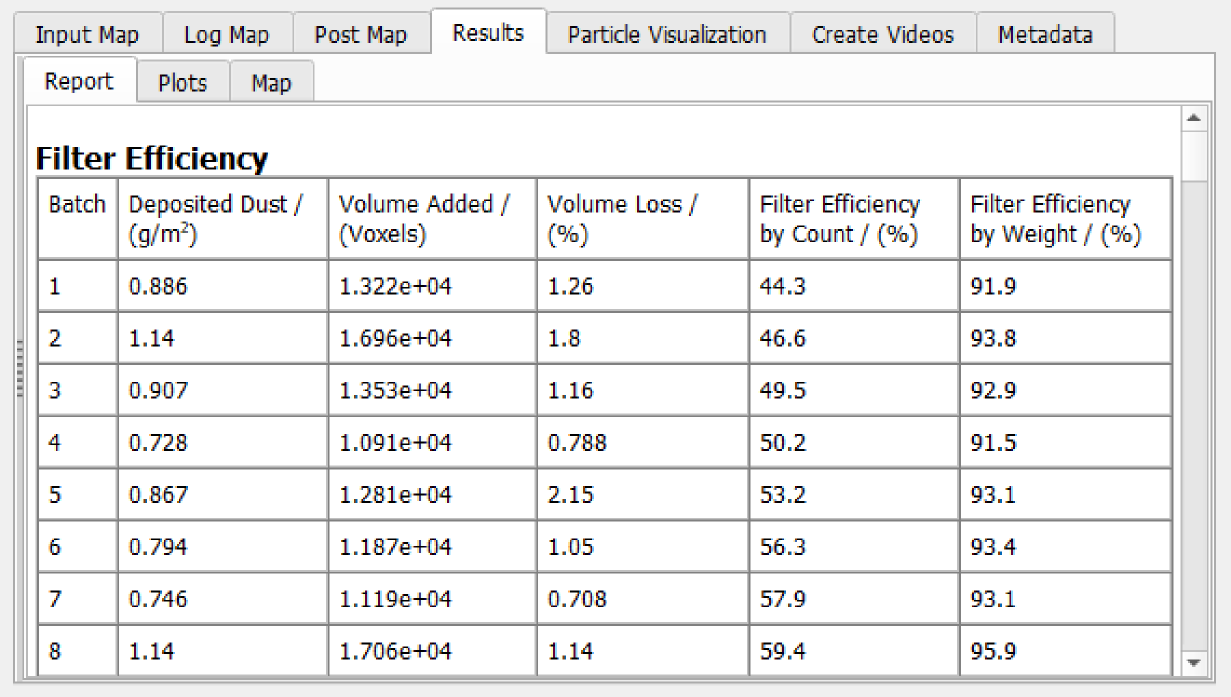
The **Efficiency Computation** panel allows to select the definition of the filtration efficiency. It can either be based on the number of particles which leave the domain via the outflow region (**Efficiency = 1 – NumberOfParticlesOutflow / NumberOfParticlesInflow**), or on the number of particles which are captured (**Efficiency = NumberOfParticlesCaptured / NumberOfParticlesInflow**). The difference between both definitions is the handling of particles which are still in the domain but are not (yet) marked as captured.



Efficiency Computation

● Efficiency = 1 - NumberOfParticlesOutflow / NumberOfParticlesInflow

○ Efficiency = NumberOfParticlesCaptured / NumberOfParticlesInflow

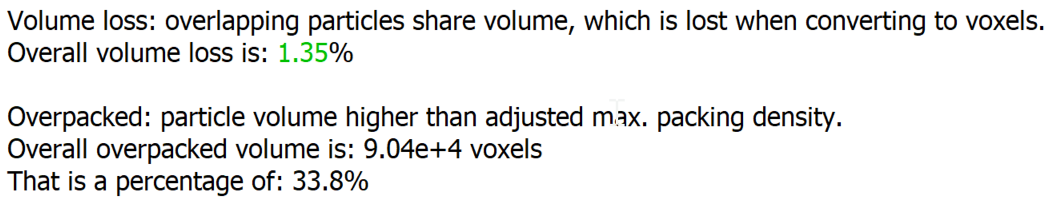


Filter Efficiency

Batch	Deposited Dust / (g/m^2)	Volume Added / (Voxels)	Volume Loss / (%)	Filter Efficiency by Count / (%)	Filter Efficiency by Weight / (%)
1	0.886	1.322e+04	1.26	44.3	91.9
2	1.14	1.696e+04	1.8	46.6	93.8
3	0.907	1.353e+04	1.16	49.5	92.9
4	0.728	1.091e+04	0.788	50.2	91.5
5	0.867	1.281e+04	2.15	53.2	93.1
6	0.794	1.187e+04	1.05	56.3	93.4
7	0.746	1.119e+04	0.708	57.9	93.1
8	1.14	1.706e+04	1.14	59.4	95.9

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:



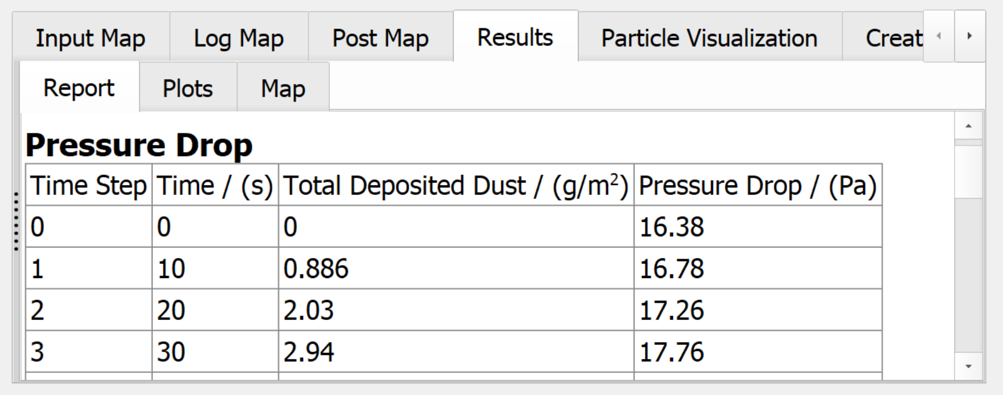
Volume loss: overlapping particles share volume, which is lost when converting to voxels.
Overall volume loss is: 1.35%

Overpacked: particle volume higher than adjusted max. packing density.
Overall overpacked volume is: 9.04e+4 voxels
That is a percentage of: 33.8%

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, 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 and the Pressure Drop for every batch.

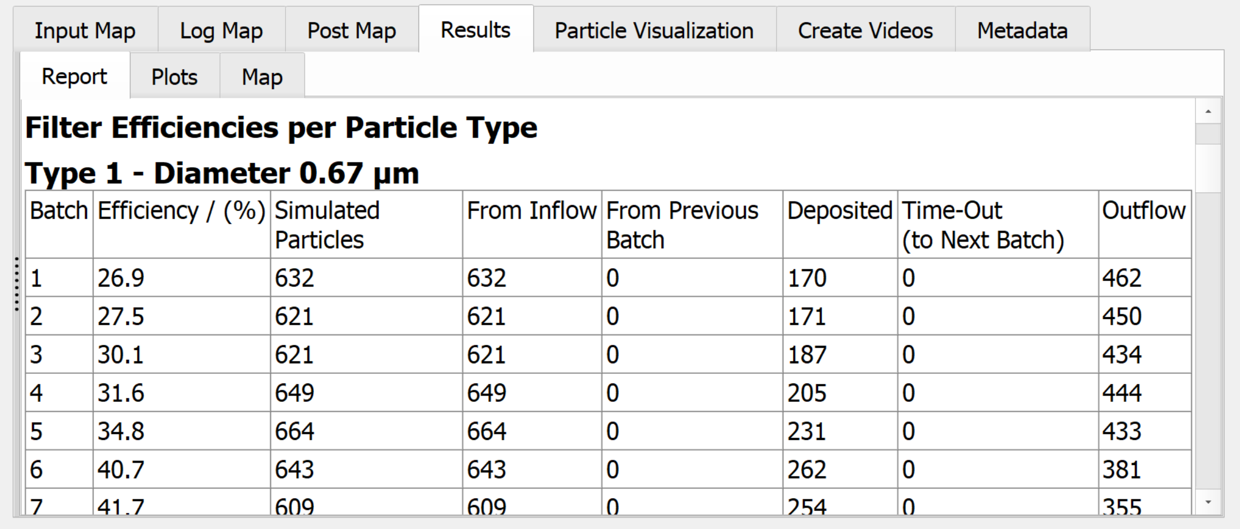


Pressure Drop

Time Step	Time / (s)	Total Deposited Dust / (g/m^2)	Pressure Drop / (Pa)
0	0	0	16.38
1	10	0.886	16.78
2	20	2.03	17.26
3	30	2.94	17.76

Filter Efficiencies per Particle 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**. 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

Type 1 - Diameter 0.67 µm

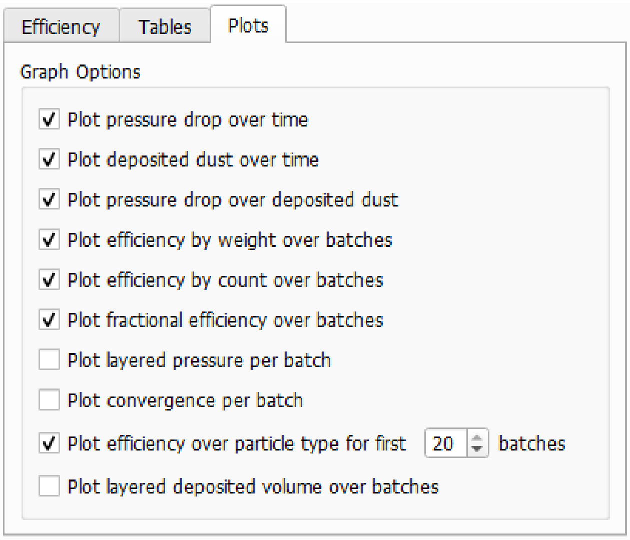
Batch	Efficiency / (%)	Simulated Particles	From Inflow	From Previous Batch	Deposited	Time-Out (to Next Batch)	Outflow
1	26.9	632	632	0	170	0	462
2	27.5	621	621	0	171	0	450
3	30.1	621	621	0	187	0	434
4	31.6	649	649	0	205	0	444
5	34.8	664	664	0	231	0	433
6	40.7	643	643	0	262	0	381
7	41.7	609	609	0	254	0	355

Layered Deposited Volume

The table shows the amount of deposited dust per z-layer for each batch. For large structures and multiple batches, the table becomes very large and hardly readable. It is recommended to visualize these results through the corresponding plot which is selectable in the Plots tab instead.

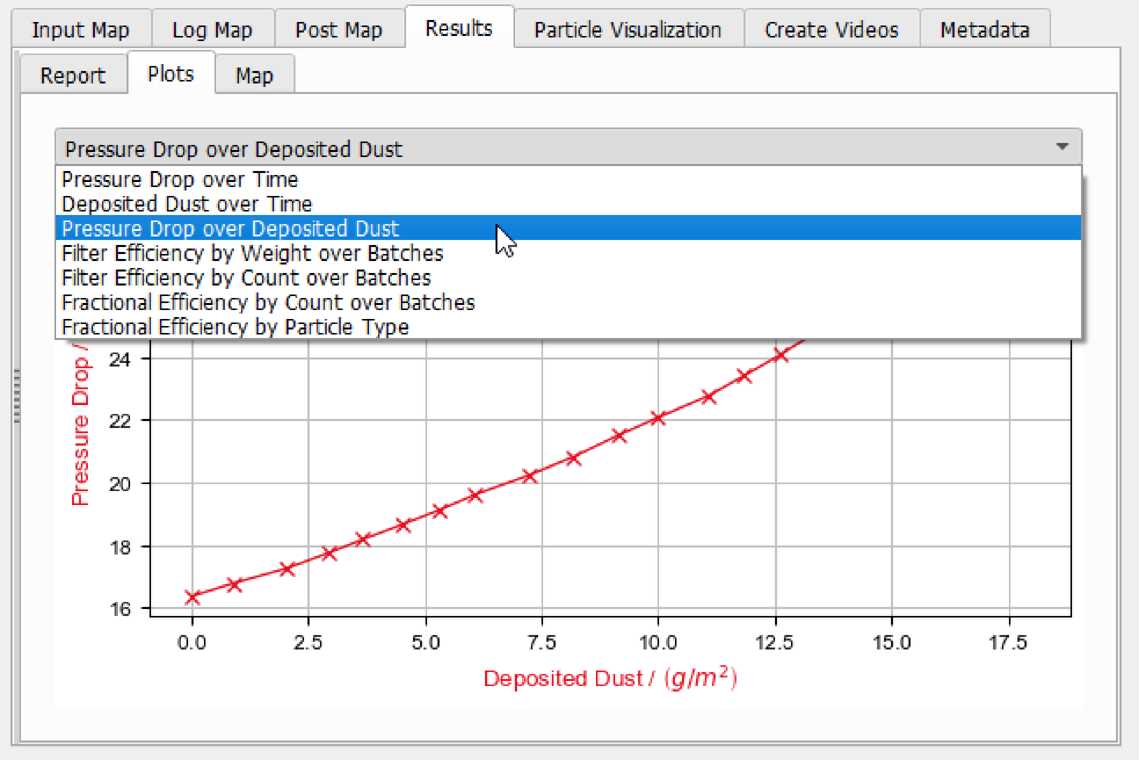
PLOTS

The **Plots** tab contains as many plots as selected under the **Graph Options** in the post-processing section. Up to two plots are shown above each other. If more plots are generated, a dropdown-menu is shown to select the plot to be shown.



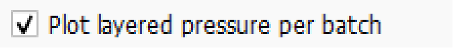
Graph Options

☑ Plot pressure drop over time
☑ Plot deposited dust over time
☑ Plot pressure drop over deposited dust
☑ Plot efficiency by weight over batches
☑ Plot efficiency by count over batches
☑ Plot fractional efficiency over batches
□ Plot layered pressure per batch
□ Plot convergence per batch
☑ Plot efficiency over particle type for first 20 batches
□ Plot layered deposited volume over batches



Input Map
Log Map
Post Map
Results
Particle Visualization
Create Videos
Metadata
Report
Plots
Map
Pressure Drop over Deposited Dust
Pressure Drop over Time
Deposited Dust over Time
Pressure Drop over Deposited Dust
Filter Efficiency by Weight over Batches
Filter Efficiency by Count over Batches
Fractional Efficiency by Count over Batches
Fractional Efficiency by Particle Type
Pressure Drop /
24
22
20
18
16
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Deposited Dust / (g/m^2)

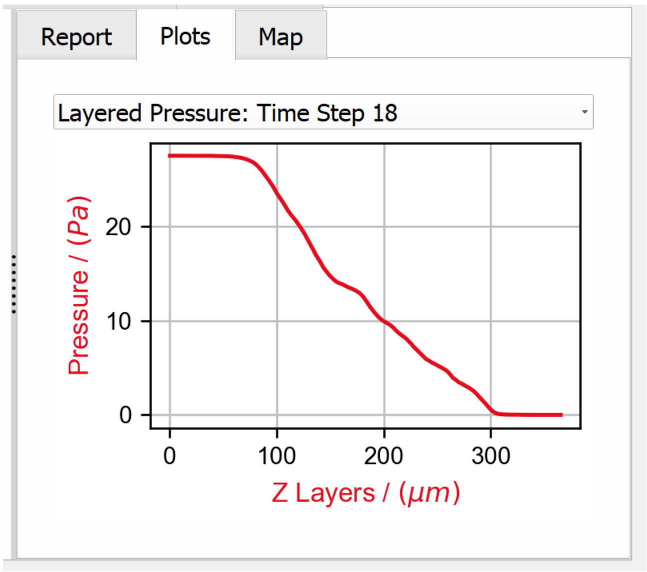
Plot Layered Pressure



☑ Plot layered pressure per batch

These plots show 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.

Select the time step with the **Layered Pressure: Time Step** pull-down menu.

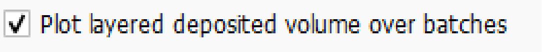


Layered Pressure: Time Step 18

Plot Convergence

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.

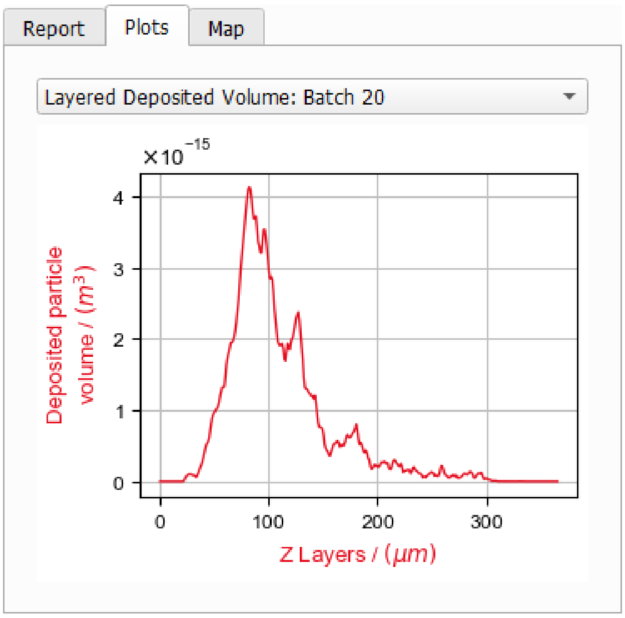
Plot Layered Deposited Volume over Batches



☑ Plot layered deposited volume over batches

These plots show the distribution of the deposited mass over the height of the filter media (z-direction) for each batch.

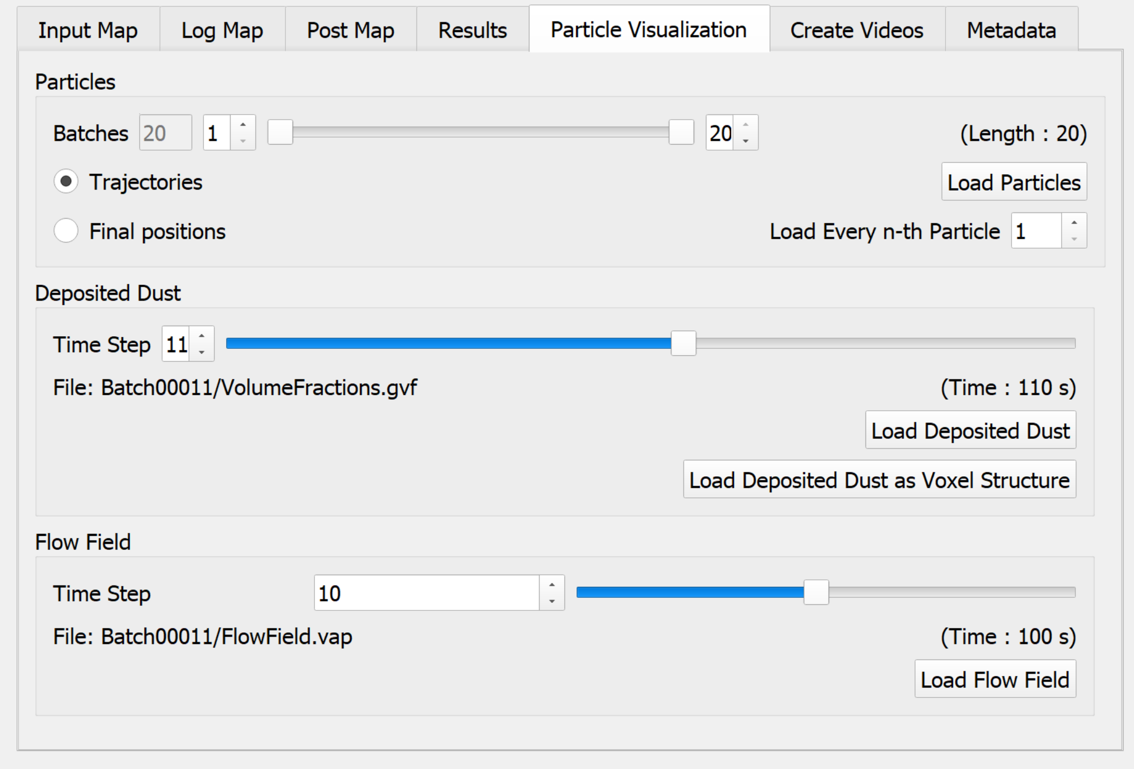
Select the time step (batch) with the **Layered Deposited Volume: Batch** pull-down menu.



Layered Deposited Volume: Batch 20

Data 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

Batches 20 1 20 (Length : 20)

● Trajectories

○ Final positions

Load Every n-th Particle 1

Deposited Dust

Time Step 11

File: Batch00011/VolumeFractions.gvf (Time : 110 s)

Flow Field

Time Step 10

File: Batch00011/FlowField.vap (Time : 100 s)

Besides the option to **Load Particles** and visualize their trajectories or final positions at each batch, the deposited dust particles can be also loaded as volume fraction data (.gvf file for each batch) by clicking **Load Deposited Dust**. The deposited particles can also be added to the voxel structure (**Load Deposited Dust as Voxel Structure**).

Additionally, the flow field at different time steps can be visualized by loading the corresponding FlowField.vap file at each time step.

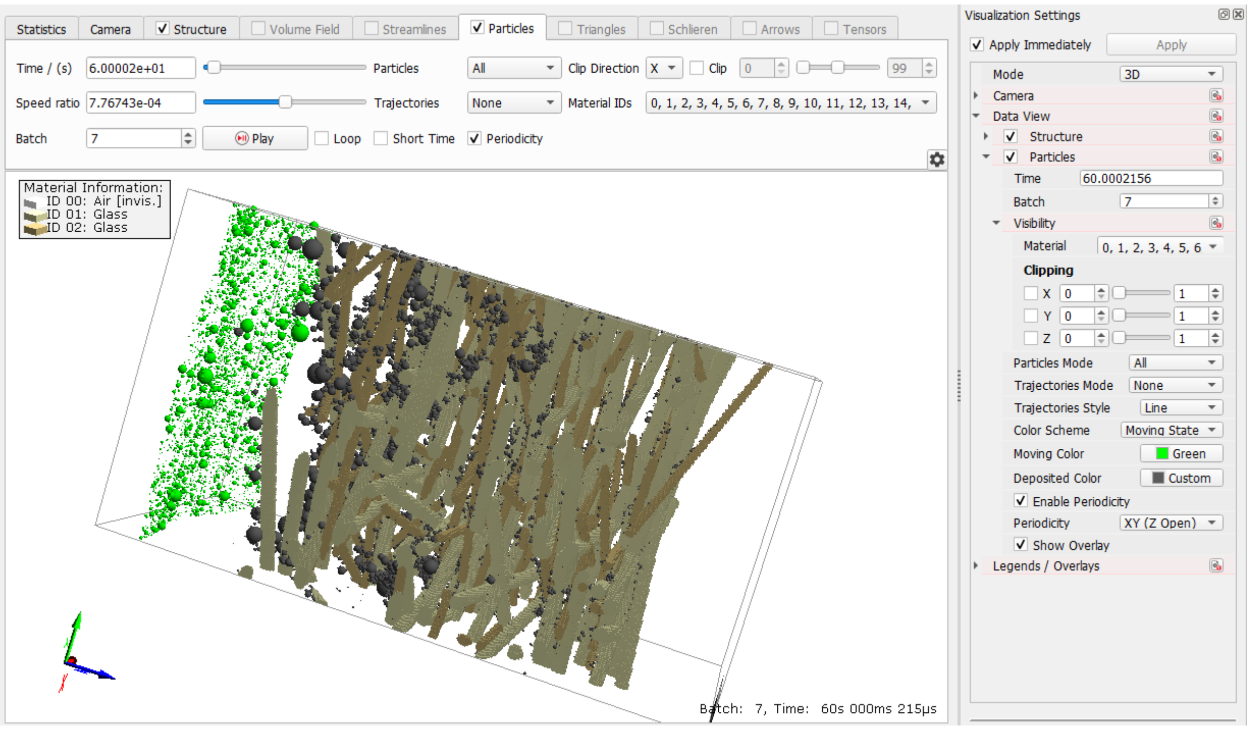
Visualization of Particles

In the **Particles** panel, one or several succeeding batches can be selected with the sliders. Clicking **Load Particles**, only the particles from the selected batches are loaded. Furthermore, either particle **Trajectories** or **Final Positions** can be selected. Loading the trajectories allows to animate the particle motion, while loading only the final positions is less memory consuming. If a very large number of particles was simulated, and visualization of all particles is not possible, the number of visualized particles can be reduced by loading only **Every n-th Particle**.

In the example below, **Trajectories** was checked. At the bottom-right of the visualization area, observe the current **Batch** number and the point in **Time** of the filtration process.

The most important settings for the particle visualization can be accessed through the visualization panel, under the **Particles** tab.

All available visualization settings are opened by clicking the gear icon ⚙ on the bottom right of visualization panel in the GeoDict GUI. Here, some additional parameters can be changed, e.g., the default green color of the moving particles can be changed in **Visualization Settings** → **Data View** → **Particles** → **Moving Color**. The color of the deposited particles can also be changed from the default grey.



Time / (s) 6.00002e+01

Speed ratio 7.76743e-04

Batch 7

Particles: All

Trajectories: None

Clip Direction: X

□ Clip 0 99

Material IDs: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14,

□ Loop □ Short Time ☑ Periodicity

Batch: 7, Time: 60s 000ms 215µs

Visualization Settings

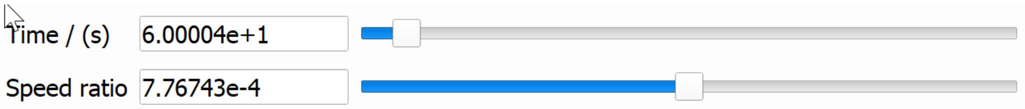
☑ Apply Immediately

- Mode: 3D
- Camera
- Data View
  - ☑ Structure
  - ☑ Particles
    - Time: 60.0002156
    - Batch: 7
    - Visibility
      - Material: 0, 1, 2, 3, 4, 5, 6
      - **Clipping**
      - □ X 0 1
      - □ Y 0 1
      - □ Z 0 1
    - Particles Mode: All
    - Trajectories Mode: None
    - Trajectories Style: Line
    - Color Scheme: Moving State
    - Moving Color: 🟩 Green
    - Deposited Color: ■ Custom
    - ☑ Enable Periodicity
    - Periodicity: XY (Z Open)
    - ☑ Show Overlay
- Legends / Overlays

The color of the particles can be set to reflect their size. To do so, change the **Color Scheme** from the default **Moving State** to **Size.**

Sometimes it is useful to switch off the visualization of the structure by unchecking □ Structure.

In the visualization panel, above the Visualization area, the sliders for **Time** and **Speed ratio** control the visualization of the particle movement. Moving the slider or directly entering a **Time** allows displaying the particles at a given moment in time.



Time / (s) 6.00004e+1

Speed ratio 7.76743e-4

The **Speed ratio** (Simulated Time (ms)/Real Time (s)) controls the visualization speed by entering values directly or by moving the slider.

The animation of the particles starts by clicking **Play**. The **Batch** number changes from the first loaded batch to the last loaded batch.

The **Time** slider does not return to zero when the animation has finished and must be set to the initial value to restart the animation. Checking **Loop** makes the slider return to zero and the animation is played endlessly.

If the simulation was done with periodic boundary conditions, **Periodicity** should be checked to animate the particles with periodic boundary conditions, too. This way, particles which leave the domain at one side re-enter at the opposite side.

Checking **Short Time** adapts the visualization **Time** to the period when most particles move. In each batch, this option cuts off the final part of the simulation where only a few particles move at very low velocities and virtually no changes are visible.

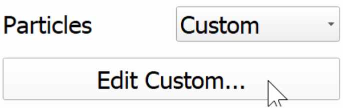
Custom visualization of particles

The visualization of particles can be customized with the **Particles** and **Trajectories** pulldown menus.

The **Particles** pull-down menu can be set to:

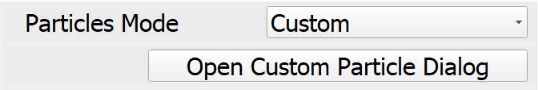
- **None** to show no particles,
- **Filtered** to show only the filtered particles,
- **Not Filtered** to only show the particles that are not filtered,
- **Time Out** to display only particles whose movement could not be simulated completely within the given simulation time,
- **All** to show all particles, or
- **Custom**.

If **Custom** is chosen for the visualization of the particles, settings for a user-defined, customized visualization can be defined by clicking the **Edit Custom…** button. The Custom Particle Selection dialog opens.



Particles: Custom

Alternatively, the same dialog can be accessed by clicking **Open Custom Particle Dialog** in the Visualization Settings panel.



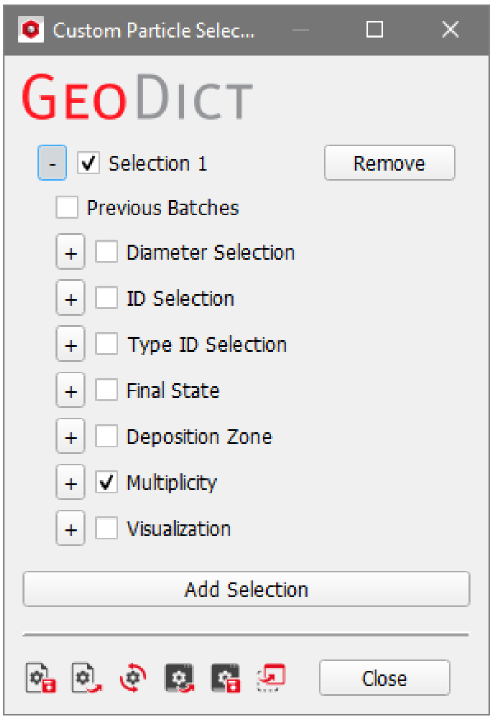
Particles Mode: Custom

Clicking **Add Selection** generates **Selection 1**.

Clicking + at the left of **Selection 1** opens the options: **Previous Batches**, **Diameter Selection**, **ID Selection**, **Type ID Selection Final State**, **Deposition Zone**, **Multiplicity**, and **Visualization**. Parameters for all these options are accessed by clicking on their + symbol.

Clicking **Add Selection** again adds a new selection (e.g., Selection 2), which allows combining different selection models and is useful to distinguish e.g., particles with two different diameter ranges (**Diameter Selection**).

Clicking **Remove** deletes the selection.



Custom Particle Selec...

GeoDict

- ☑ Selection 1
- ☐ Previous Batches
- ☐ Diameter Selection
- ☐ ID Selection
- ☐ Type ID Selection
- ☐ Final State
- ☐ Deposition Zone
- ☑ Multiplicity
- ☐ Visualization

The **Custom Particle Selection** dialog may remain open to observe the changes that the selections have on the visualization of particles moving through the structure. The custom particle options can be changed while the particle animation is playing, so that the effect of the selection can be observed immediately.

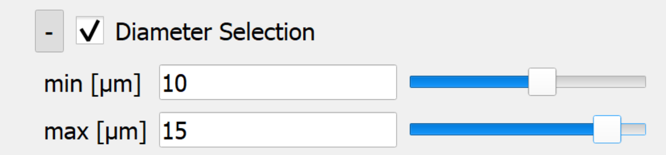
Previous Batches

If **Previous Batches** is unchecked, the selection applies only to the current batch.

When **Previous Batches** is checked, the selection only applies to the particles deposited in previous batches. The selection does not apply for the current batch (the particles currently in motion in the visualization).

Diameter Selection

Checking **Diameter Selection** limits the visualized particles to those with a particle radius in the range specified by min and max.



☑ Diameter Selection

min [µm] 10

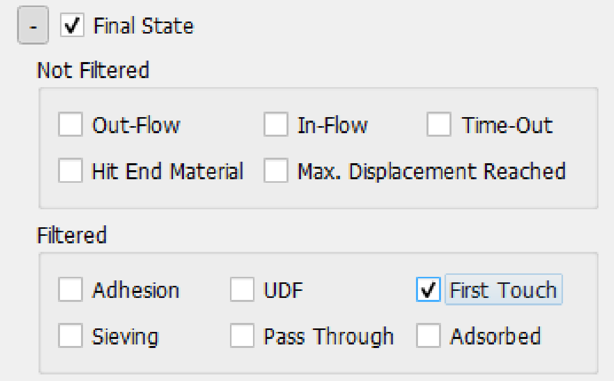
max [µm] 15

ID Selection and Type ID Selection

Every particle is counted and has its own specific **ID**. Furthermore, every particle has a **Type ID**, that corresponds to the particle types entered in the Particles tab (see page 28). If you have e.g., 30 different radii, there are 30 different Type IDs. The **ID Selection** and **Type ID Selection** allow to select particles based on their individual particle number and specific **Type ID**.

Final State

Final State enables the visualization of particles with the given state at the end of the simulation.



☑ Final State

Not Filtered

□ Out-Flow □ In-Flow □ Time-Out

□ Hit End Material □ Max. Displacement Reached

Filtered

□ Adhesion □ UDF ☑ First Touch

□ Sieving □ Pass Through □ Adsorbed

In the **Not Filtered** panel, particles that were not filtered can be selected as follows:

- **Out-Flow** denotes particles that left the computational domain through the fluid outflow region.
- **In-Flow** designates particles that left the computational domain through the fluid inflow region.
- **Time-Out** identifies all particles which are still moving at the end of the time interval.
- **Hit End Material** only shows particles which hit the end material. This only applies if the **Particle End Position** in the simulation is set to material ID and not to the outflow plane.

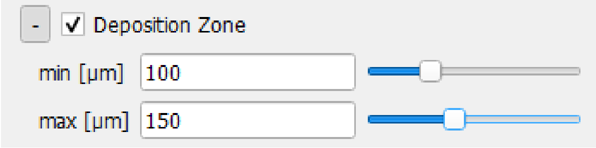
- The **Max. Displacement Reached** state can only be achieved in AddiDict, therefore is not relevant for FilterDict results.

The **Filtered** particles can be distinguished by the collision model (see page 10):

- **Adhesion** selects all particles that are filtered by an adhesion model.
- **UDF** selects all particles that are filtered by a model from a user defined function (UDF)
- **First Touch** selects particles that are filtered by the caught on first touch model.
- **Sieving** selects particles are filtered by the sieving model.
- **Pass Through** selects particles that are filtered by the pass through model.
- **Adsorbed** selects particles that were adsorbed in a porous medium. The adsorption model is only available in AddiDict, therefore is not relevant for FilterDict results.

Deposition Zone

The **Deposition Zone** allows to select particles depending on their final position in the structure.



- ☑ Deposition Zone

min [µm] 100

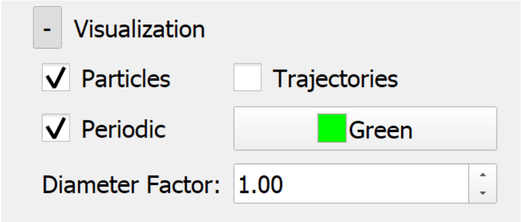
max [µm] 150

Multiplicity

Particles can be selected depending on their multiplicity. Particle Multiplicity is a simulation option for unresolved particles which helps reducing simulation time for very small particles (see page 45 for further information).

Visualization

The visualization settings can be set for each particle selection separately. This allows to assign different colors to different selections of particles.



- Visualization

☑ Particles □ Trajectories

☑ Periodic 🟩 Green

Diameter Factor: 1.00

Checking **Particles** displays all particles that conform to the options previously set in the dialog.

Checking **Trajectories** displays the trajectories which correspond to the selected particles. With this option, it is e.g., possible to show only the trajectories which correspond to a certain particle type.

Checking **Periodic** enables periodic boundaries for the display of the particle movements. Particles which leave the domain at one side re-enter at the opposite side.

Green, the default color for the particles, can be changed to another color (e.g., Dark red) through the button.

The **Diameter Factor** scales the particles sizes relatively to their original size for visualization purposes and allows to visualize very small particles.

Visualization of Deposited Dust

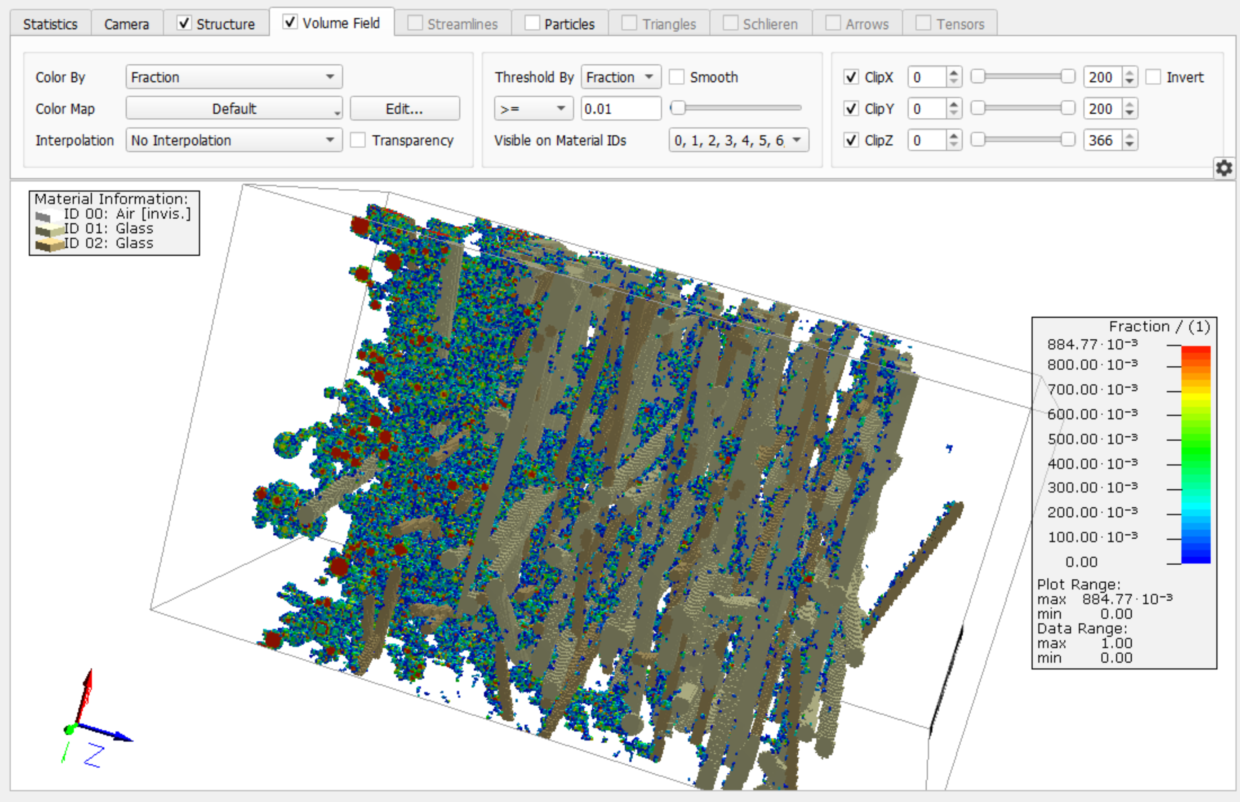
There are two options to visualize the deposited dust from the simulation: **Load Deposited Dust** and **Load Deposited Dust as Voxel Structure**.

With **Load Deposited Dust**, the result field (*.gvf) is loaded. It 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.

Load Deposited Dust

In the Result Viewer, under the **Particle Visualization** tab, in the **Deposited Dust** panel, choose the Time Step of interest, and click **Load Deposited Dust**.



Color By: Fraction
Color Map: Default
Interpolation: No Interpolation
□ Transparency

Threshold By: Fraction
□ Smooth
>= 0.01
Visible on Material IDs: 0, 1, 2, 3, 4, 5, 6

☑ ClipX 0 – 200
☑ ClipY 0 – 200
☑ ClipZ 0 – 366
□ Invert

Material Information:
ID 00: Air [invis.]
ID 01: Glass
ID 02: Glass

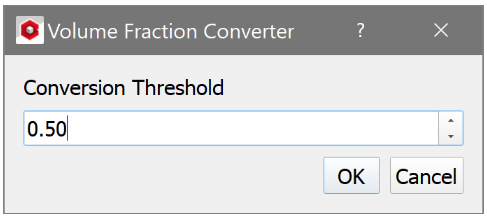
Plot Range:
max 884.77 · 10^{-3}
min 0.00
Data Range:
max 1.00
min 0.00

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 it's individual color without any interpolation and surface smoothing.

Load Deposited Dust as Voxel Structure

In the Result Viewer, under the **Particle Visualization** tab, in the **Deposited Dust** panel, choose the Time Step of interest, and click **Load Deposited Dust as Voxel Structure**.

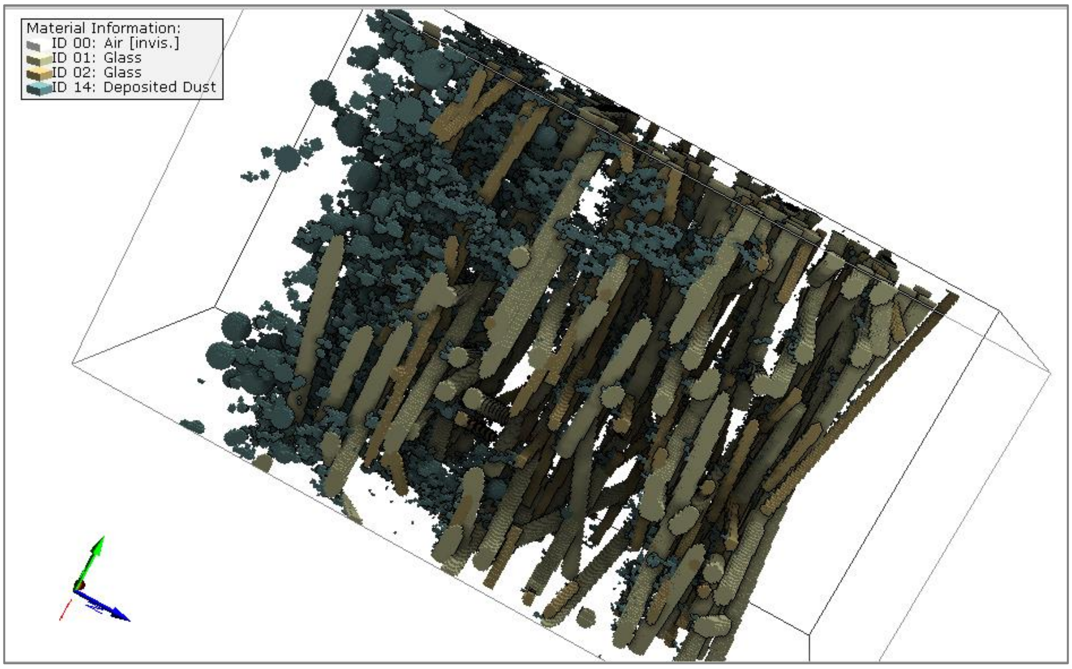
To convert the double valued 3D field stored inside the .gvf file into a structure, a conversion threshold is needed, and must be entered by the user. Enter a value between 0 and 1 here.



Conversion Threshold

0.50

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



Visualization of Flow Field

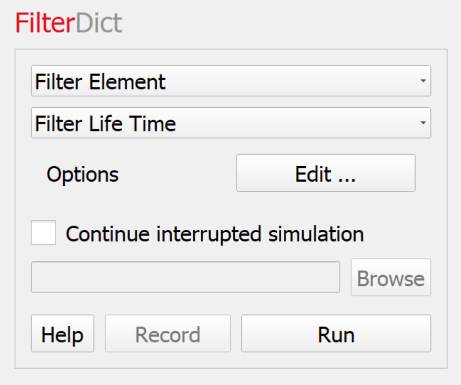
For the visualization of the flow field, please refer to the FlowDict and Visualization handbooks of this User Guide.

Filter Element

Select **Filter Element** from the drop-down menu. A **Filter Life Time** 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.

As for filter media simulations, an interrupted simulation can be restarted with **Continue interrupted simulation**. This is possible when the simulation was stopped after the calculation of the first flow field or later. When the simulation was stopped earlier, it must be started anew. Details for restarting an interrupted simulation can be found on page 16.


FilterDict

Filter Element

Filter Life Time

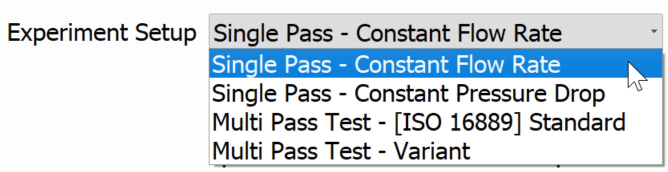
Options

□ Continue interrupted simulation

The **Filter Element Life Time** dialog opens when clicking the **Edit...** button. As for filter media simulations, it is organized into the tabs: **Filter Experiment**, **Constituent Materials**, **Particles**, **Solver**, **Output,** and **Equations and References**.

Filter Experiment

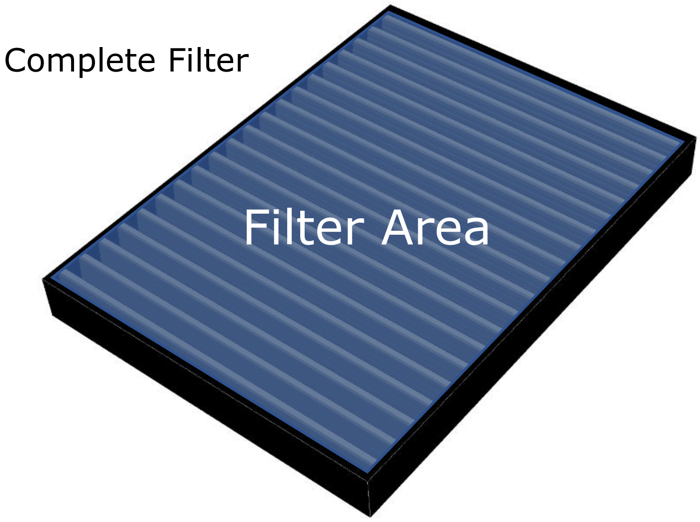
Under the **Filter Experiment** tab, choose between a **Single Pass** or a **Multi Pass** experiment. For **Single Pass**, the experiment can be run at a **Constant Flow Rate** or at a **Constant Pressure Drop** (often used for membranes). For **Multi Pass**, choose between the standard test setup or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

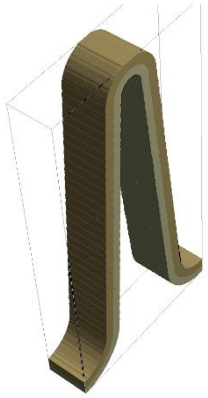

Experiment Setup: Single Pass - Constant Flow Rate

Single Pass - Constant Flow Rate
Single Pass - Constant Pressure Drop
Multi Pass Test - [ISO 16889] Standard
Multi Pass Test - Variant

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 Life Time dialog.

See the explanations on page 37 for a description of the simulated experimental setups. 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.

Simulation Stopping Criterion

See the explanation on page 39.

Flow Settings

See the explanation on page 40.

Constituent Materials

See the explanation on page 40.

Guidelines to set material parameters for filter elements

In most of the filter element and complete filter simulations, the filter media material cannot be resolved, as this would require immense computational resources. For this reason, it is possible to model the filter media layer as porous layer and parameterize the properties of the depth and cake filtration regime.

In general, these input parameters can either be taken from measurements, preferably on the flat sheet material, or from simulations on a more detailed scale. These parameters could be determined by running a resolved filtration simulation on the media scale until the filter cake has built up. (Multiscale simulation approach). From this filter media simulation, the parameters necessary to model the filter media as porous layer can be calculated.

In the following, we will explain the most straightforward and simple way to determine and set these material parameters in FilterDict-Filter Element and FilterDict-Complete Filter. These choices will not lead to correct results in all situations. Depending on the filtration process and geometry of the pleat or filter structure, it might be necessary to choose a more complex model (see also the remarks below).

To use an unresolved simulation (see page 8) in GeoDict the local clogging (f_{max}) and flow resistivity (σ_{max}) parameters for both the depth and cake regime need to be determined on the media scale according to the formulas:

$$f_{max,depth} = f_{clogged} - f_{clean} \tag{33}$$

$$\sigma_{max,depth} = \sigma_{clogged} - \sigma_{clean} \tag{34}$$

$$f_{max,cake} = f_{cake} \tag{35}$$

$$\sigma_{max,cake} = \sigma_{cake} \tag{36}$$

where the parameters mean:

- f_{clean}= Solid volume fraction of the clean filter media layer
- $f_{clogged}$ = Solid volume fraction of the clogged filter media, considering only depth filtration
- f_{cake} = Solid volume fraction of the filter cake
- σ_{clean}= flow resistivity of the clean filter media, obtained from flow simulation
- $\sigma_{clogged}$= flow resistivity of the clogged filter media, considering only depth filtration
- σ_{cake}= flow resistivity of the filter cake

This multi scale approach is furthermore useful to simulate very fine (nano-sized) fibrous layers in combination with coarse (micro-sized) filter layers. To resolve both layers a very small voxel length need to be selected leading to large number of voxels. Instead, the fine layer can be modeled as porous layer. In this case a larger voxel length can be chosen for the coarse layer, leading to a significant smaller number of voxels and therefore computational time and resources.

FLUID PHASE:

- **Permeability**: Select Isotropic and enter the through-plane permeability of the clean flat sheet, obtained by a flow simulation on the media scale or experimental measurement. Isotropic is also the best option for anisotropic flat sheet materials, because GeoDict does not take the orientation of the material into account yet (Anisotropic is always aligned with the axes of the coordinate system)
- $\boldsymbol{f_{min}}$: Select 0, or a very small value. Computation times may be slightly larger for 0 than for a small value.
- $\boldsymbol{f_{max,cake}}$: Enter the computed solidity of the filter cake here. (See remark 1 and 5 below)
- $\boldsymbol{\sigma_{max,cake}}$: Enter the calculated flow resistivity of the filter cake here. (See remark 1 and 5)
- **Pass Through Model** Select 'All particles pass'. This means that all particles will pass through a voxel until the max. particle packing density is reached inside of a voxel.
- **Max. Particle Packing Density**: Enter the computed solidity of the filter cake here. (See remark 1)
- **Collision Model**: Select Caught on first touch, Sieving or Hamaker. (See remark 2)

Porous material:

- f_{min}: Select 0, or a very small value.
- $f_{max,depth}$: (See remark 4)
- $\sigma_{max,depth}$: (See remark 4)
- **Pass Through Model**: Select 'Const. efficiency' (See remark 3)
- **Max. Particle Packing Density**: set equal to f_{max} (See remark 5)
- **Collision Model**: Select Caught on first touch, Sieving or Hamaker. (See remark 2)

Solid material, e.g., a support mesh or housing:

- **Collision model**: Select Sieving. This will prevent dust particles from sticking solely to the support structure or housing of the filter.

Particles:

- **Restitution for Sieving** (with solid material): This is an a-priori unknown parameter, which has very small influence on the results if not chosen too small.
- **Media Thickness for Const. efficiency**: Enter the thickness of the flat sheet material or height of the filter geometry here.
- **Efficiency for Const. efficiency**: Enter the filtration efficiency of the flat sheet material for this particle size or the average filtration efficiency per particle type over all computed batches from the media scale simulation. (See remark 3)

Remark 1: Choice of flow resistivity model for the filter 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 or a pleat geometry), using the solidity of the filter cake for f_{max} and the Maximal Particle Packing Density and using the flow resistivity as σ_{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 a voxel length, the filter cake that emerges during the simulation will have a different solidity than f_{max} and a different flow resistivity as σ_{max}. This has been described in [10], and the paper also describes a way to estimate those parameters in this case.

Remark 2: Choice of Collision Model

The collision model 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 dust 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.

Remark 3: Choice of Pass Through Model

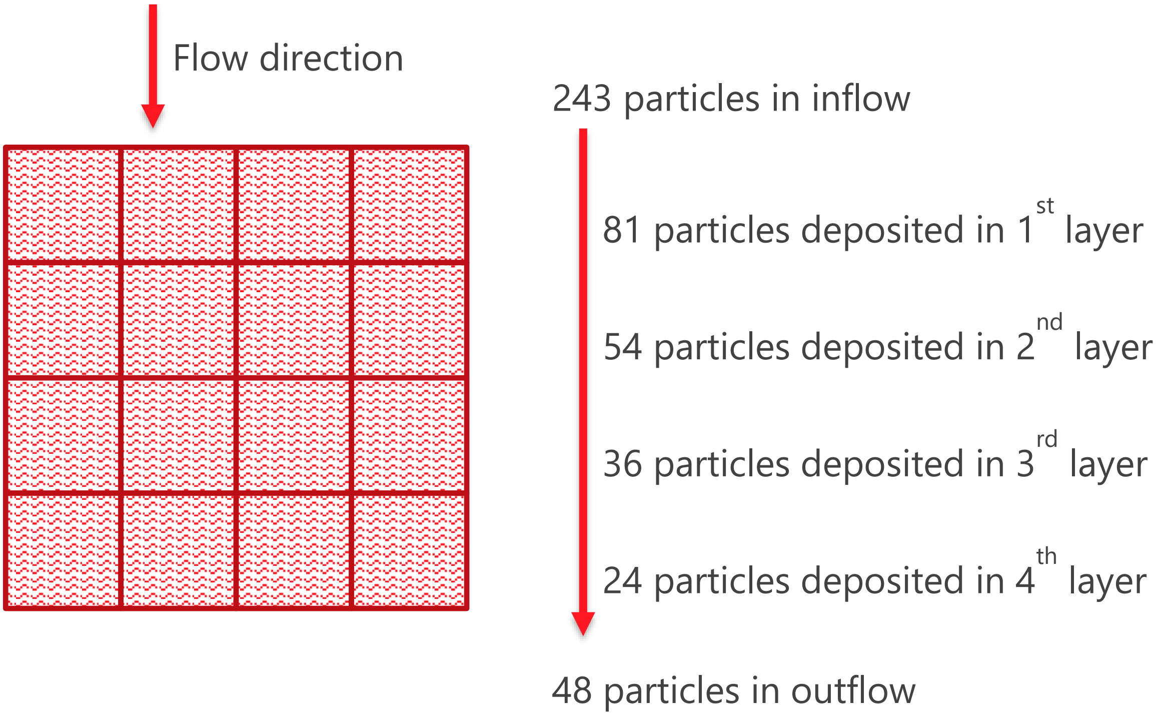
Assuming a constant efficiency is a severe simplification, which surely does not describe the behavior of the real material in detail. However, as soon as the Max. Particle Packing Density of a voxel is reached, the voxel will not let any other particles pass through, and therefore the overall efficiency of the filter will change over time even when using a Const. Efficiency model. When using this model, it is recommended to use an efficiency value that represents an average efficiency value over the depth filtration phase.

Pass through models which modify the efficiency value based on the local dust clogging are currently not available through the user interface but can be implemented through user defined functions (UDF).

Remark 4: Choice of Flow Resistivity Model for the porous material

The difficulty in choosing the flow resistivity model lies in the dependence of this model on the chosen resolution. It is dependent on the resolution, because the entered Max. Particle Packing Density will only be achieved in the first voxels that lie on the upstream side of the filter – all voxels below will not become completely filled. Therefore, when deriving these parameters from a micro scale simulation of the flat sheet material, the resolution of the macro-scale model has to be taken into account: if the voxel length on the macro scale is 50 µm, an analysis of those (50 µm)³ cubes of the micro-scale result that lie on the upstream side of the filter would be needed to estimate the parameters of the flow resistivity model.

The image below illustrates how the deposited particles will be distributed in a porous layer, where every porous voxel has the same filtration efficiency (Here: a third of the particles are filtered when passing through a voxel length).


Flow direction
243 particles in inflow
81 particles deposited in 1st layer
54 particles deposited in 2nd layer
36 particles deposited in 3rd layer
24 particles deposited in 4th layer
48 particles in outflow

For any given efficiency, the first layer will always collect more particles than any layer below, therefore it will reach the Max. Particle Packing Density first, and not let further particles through. At that moment, cake filtration starts on the top of the first layer.

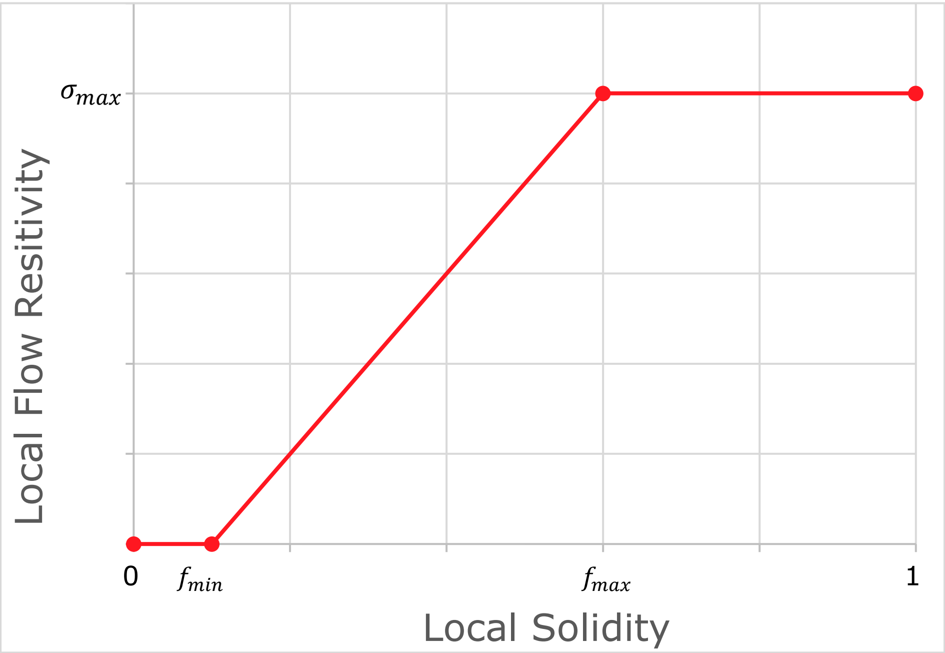
Remark 5: Difference between f_{max} and Maximum Particle Packing Density

In the standard setup as described above, both values are set to the same value, and GeoDict versions earlier than 2019 did not allow to set those values differently.

However, both values model different things, and can therefore now be set independently of each other.

The Maximum Particle Packing Density is used to decide if additional particles can move into a voxel. If this value is exceeded, no more particles can move into this voxel, and an oncoming particle will collide on the voxel boundary (the chosen **Collision Model** will be applied).

The value of f_{max} is used to define equation (31), which computes the local flow resistivity in a voxel based on the local solidity. In principle, an arbitrary function could be used to describe this behavior. By default, GeoDict defines this function as a piecewise linear function as follows:



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, the pair (f_{max}, σ_{max}) can be chosen arbitrarily, and the given numbers are not used for any other purpose than to define the relationship between local solidity and local flow resistivity.

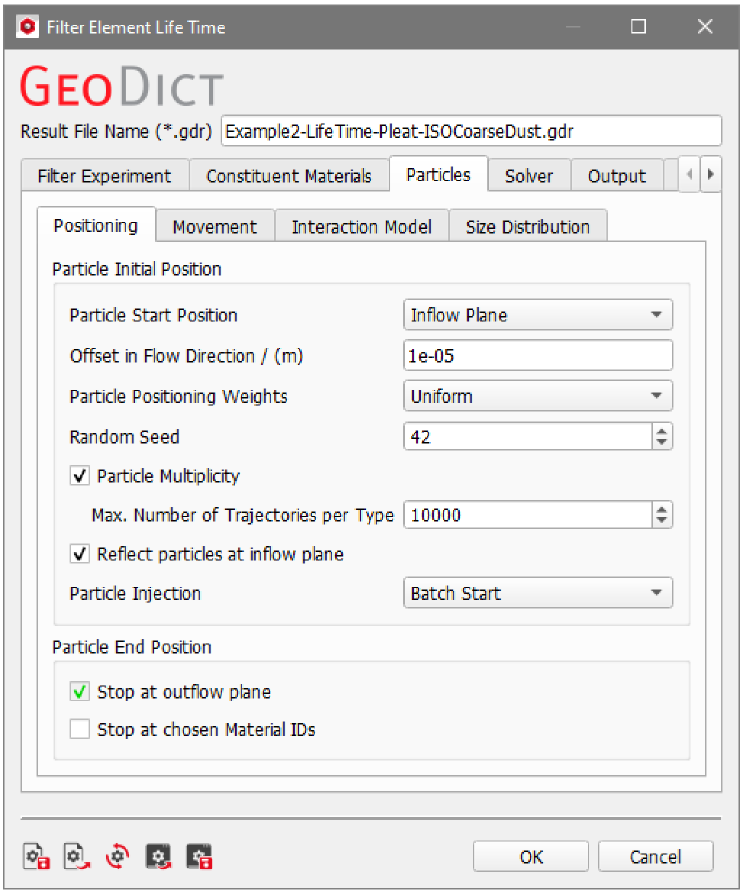
Therefore, f_{max} is not the highest solidity that may appear in a voxel and σ_{max} is not automatically the flow resistivity of the filter cake!

PARTICLES

Under the **Particles** tab, four subtabs are available to define **Positioning**, **Movement**, **Interaction Model**, and **Size Distribution**.

POSITIONING

The parameters here are mostly the same as for the **Particles Positioning** tab for the FilterDict-Filter Media simulation, starting on page 44.



Result File Name (*.gdr) Example2-LifeTime-Pleat-ISOCoarseDust.gdr

Particle Initial Position

Particle Start Position: Inflow Plane

Offset in Flow Direction / (m): 1e-05

Particle Positioning Weights: Uniform

Random Seed: 42

☑ Particle Multiplicity

Max. Number of Trajectories per Type: 10000

☑ Reflect particles at inflow plane

Particle Injection: Batch Start

Particle End Position

☑ Stop at outflow plane

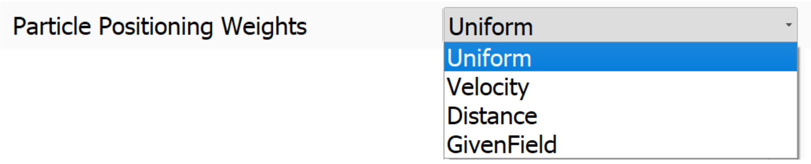
□ Stop at chosen Material IDs

Additionally, it is possible to change the **Particle Positioning Weights**, which enables non-uniformly distributed start positions inside the defined volume.

It is not possible to choose the **Ghost Particles** option for Filter Element simulations.

Particle Positioning Weights

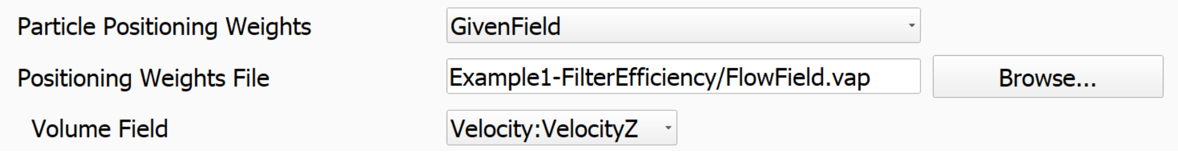
Particle start positions are scattered randomly inside the defined **Particle Start Position** area or volume. By default, those positions are uniformly distributed.



Particle Positioning Weights: Uniform

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.



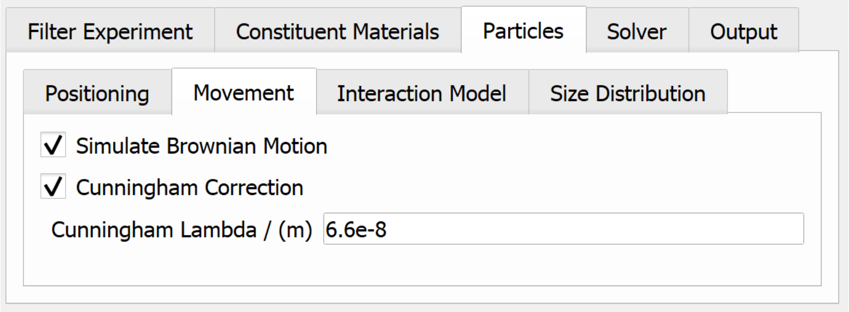
Particle Positioning Weights: GivenField

Positioning Weights File: Example1-FilterEfficiency/FlowField.vap

Volume Field: Velocity:VelocityZ

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

MOVEMENT



☑ Simulate Brownian Motion

☑ Cunningham Correction

Cunningham Lambda / (m): 6.6e-8

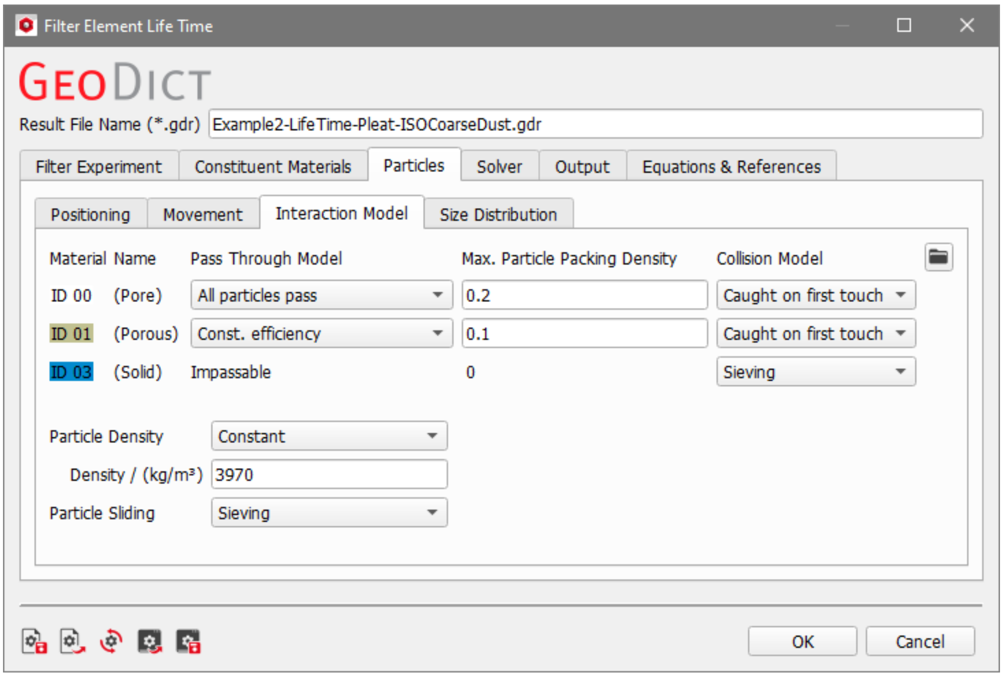
For **Simulate Brownian Motion** and **Cunningham Correction**, see pages 25ff.

It is not possible to model electrostatic effects if parts of the filter structure are unresolved, therefore this option is not selectable here.

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 in the **Interaction Model** subtab.

INTERACTION MODEL

For all materials in the model, a **Pass Through Model**, the **Max. Particle Packing Density** and a **Collision Model** must be set.



Result File Name (*.gdr): Example2-LifeTime-Pleat-ISOCoarseDust.gdr

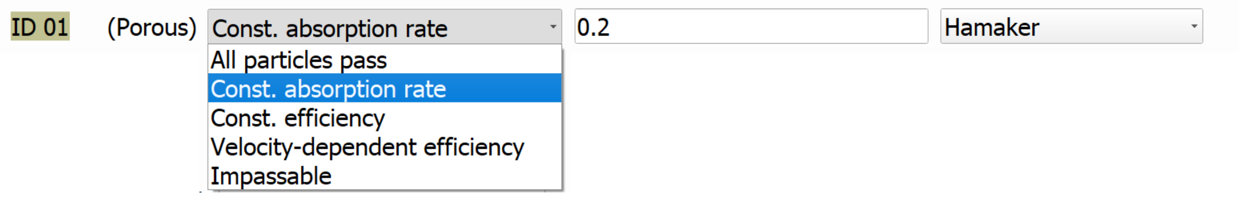
Material Name		Pass Through Model	Max. Particle Packing Density	Collision Model
ID 00	(Pore)	All particles pass	0.2	Caught on first touch
ID 01	(Porous)	Const. efficiency	0.1	Caught on first touch
ID 03	(Solid)	Impassable	0	Sieving

Particle Density: Constant

Density / (kg/m³): 3970

Particle Sliding: Sieving

For porous materials the Pass Through Model can be chosen as **All Particles pass**, **Constant absorption rate**, **Constant efficiency**, **Velocity-dependent efficiency**, or **Impassable**. If set to **Impassable**, the material is treated as a solid material and particles cannot enter. The models are described on pages 12ff. in detail.



ID 01 (Porous) Const. absorption rate | 0.2 | Hamaker

If **All Particles Pass**, **Constant absorption rate**, **Constant efficiency**, or **Velocity-dependent efficiency** is chosen, 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. For this case, also a collision model must be chosen among **Caught on first touch**, **Hamaker**, or **Sieving**. If Velocity-dependent efficiency is selected, it must be defined for each particle type in the **Size Distribution** 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 **Collison 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.

Particle Density and Particle Sliding

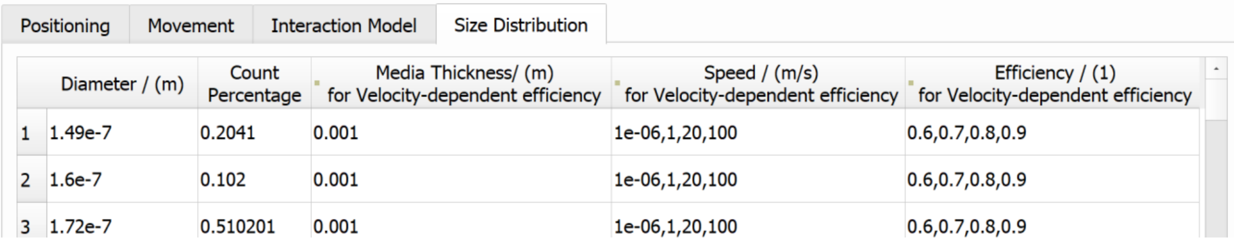
These parameters have the same meaning as in the Filter Efficiency command and are explained on page 26.

Size Distribution

For the options under the **Size Distribution** tab, see pages 28ff. Here, the size distribution table may contain additional columns dependent on the chosen **Pass Through Model**.

If **Velocity-dependent efficiency** is chosen as pass through model, efficiency values depend on the velocity as shown in the following screenshot (see equations on pages 12ff.).

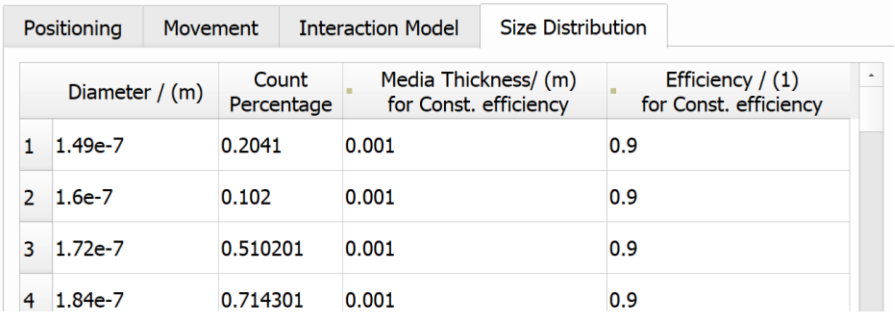
For example, with the definitions in the table below, at a particle velocity of 1 m/s, 70% of the particles are filtered when passing through 1 mm of the media.



	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	1.49e-7	0.2041	0.001	1e-06,1,20,100	0.6,0.7,0.8,0.9
2	1.6e-7	0.102	0.001	1e-06,1,20,100	0.6,0.7,0.8,0.9
3	1.72e-7	0.510201	0.001	1e-06,1,20,100	0.6,0.7,0.8,0.9

If **Constant efficiency** is chosen as pass through model, fractional filtration efficiency values obtained from the flat sheet experiments or from the micro-structural simulation must be manually inserted.

The fractional filtration efficiencies inserted here are the average efficiency values over the duration of the filtration simulation on the microstructure model of the media.



	Diameter / (m)	Count Percentage	Media Thickness/ (m) for Const. efficiency	Efficiency / (1) for Const. efficiency
1	1.49e-7	0.2041	0.001	0.9
2	1.6e-7	0.102	0.001	0.9
3	1.72e-7	0.510201	0.001	0.9
4	1.84e-7	0.714301	0.001	0.9

Solver

The **EJ solver** is not available as **Initial** or **Iterative Flow Solver** because it cannot solve Stokes-Brinkman or Navier-Stokes-Brinkman equations, needed for filtration simulations for material models with porous voxels.

Otherwise, the options under the **Solver** tab are the same as explained in pages 49ff.

Output

The parameters here are the same as for the **Output** tab for the Filter Media simulation, starting on page 54.

Equations & References

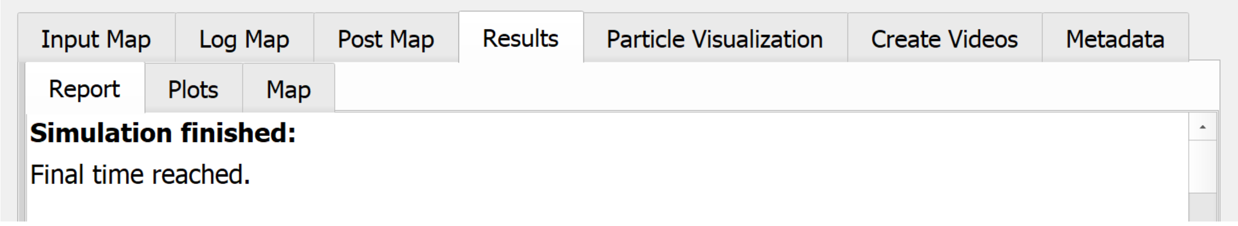
The **Equations & References** tab shows the relevant equations, i.e the Navier-Stokes equations (1)-(2) for fast flows or the Stokes equations (3)-(4) for slow flows. Also, the equations (9)) to (12) governing the particle motion are shown. Furthermore, equation (17) used to model the movement of particles through porous media is shown here, too.

Results

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

Report

At the top of the **Report**, the stopping criteria reached by the simulation is shown.



Simulation finished:

Final time reached.

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

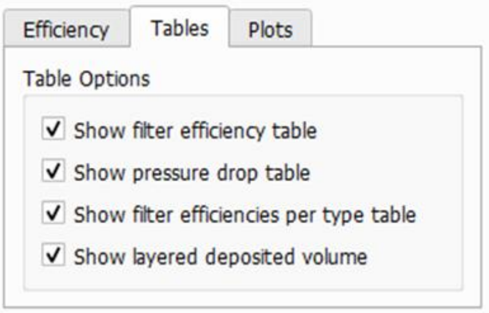


Table Options

☑ Show filter efficiency table

☑ Show pressure drop table

☑ Show filter efficiencies per type table

☑ Show layered deposited volume

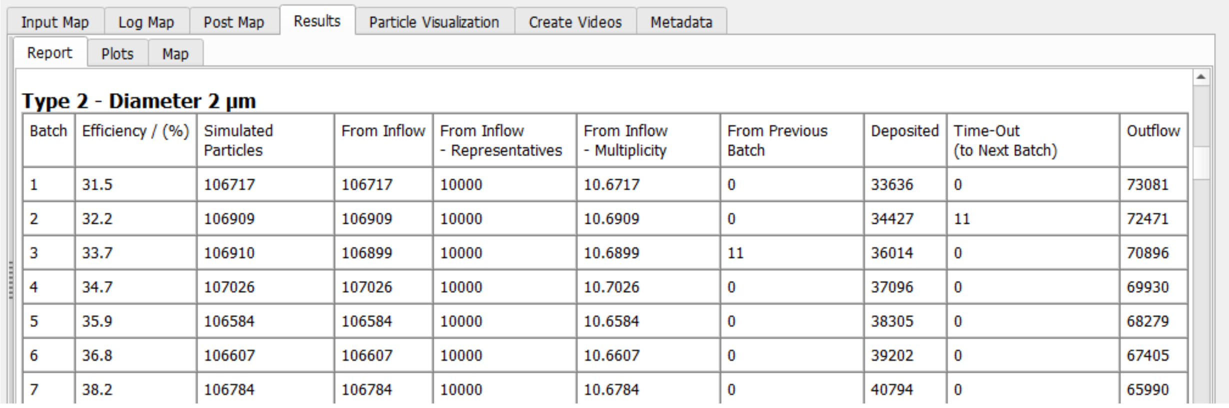
Filter Efficiency and Pressure Drop

The tables and options are identical to that of a filter media simulation (see page 56ff).

Filter Efficiencies per Particle 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 (see page 45), 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 number, so a value of 10.67 indicates that there are some representatives with a multiplicity of 10, and others with a multiplicity of 11.



Type 2 - Diameter 2 µm

Batch	Efficiency / (%)	Simulated Particles	From Inflow	From Inflow - Representatives	From Inflow - Multiplicity	From Previous Batch	Deposited	Time-Out (to Next Batch)	Outflow
1	31.5	106717	106717	10000	10.6717	0	33636	0	73081
2	32.2	106909	106909	10000	10.6909	0	34427	11	72471
3	33.7	106910	106899	10000	10.6899	11	36014	0	70896
4	34.7	107026	107026	10000	10.7026	0	37096	0	69930
5	35.9	106584	106584	10000	10.6584	0	38305	0	68279
6	36.8	106607	106607	10000	10.6607	0	39202	0	67405
7	38.2	106784	106784	10000	10.6784	0	40794	0	65990

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**.

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.

Layered Deposited Volume

The table shows the amount of deposited dust per z-layer for each batch. For large structures and multiple batches, the table becomes very large and hardly readable. It is recommended to visualize these results through the corresponding plot which is selectable in the Plots tab instead.

Plots

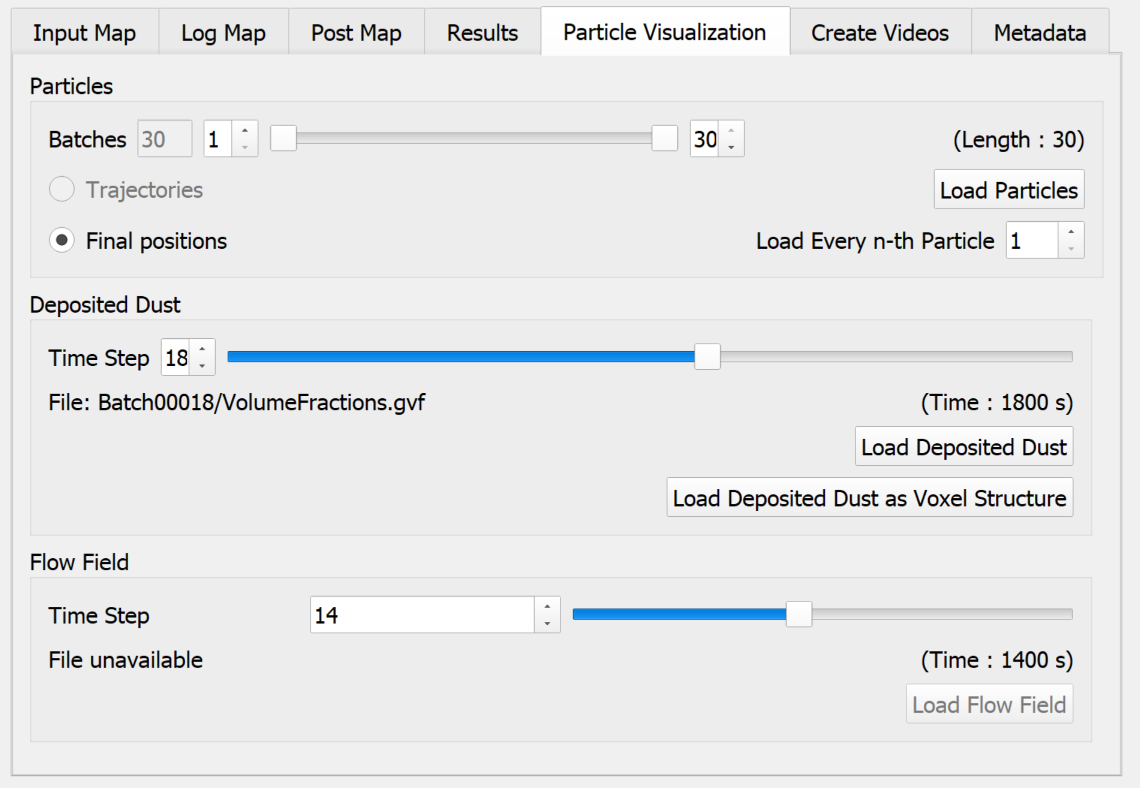
The available plots are identical to that of a filter media simulation (see page 59f).

Data 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.

Besides the option to **Load Particles** and visualize their trajectories or final positions at each batch, the deposited dust particles can be also loaded as volume fraction data (.gvf file for each batch) by clicking **Load Deposited Dust**. The deposited particles can also be added to the voxel structure (**Load Deposited Dust as Voxel Structure**).

Additionally, the flow field at different time steps can be visualized by loading the corresponding FlowField.vap file at each time step.



Particles

Batches 30 1 30 (Length : 30)

○ Trajectories

● Final positions

Load Every n-th Particle 1

Deposited Dust

Time Step 18

File: Batch00018/VolumeFractions.gvf (Time : 1800 s)

Flow Field

Time Step 14

File unavailable (Time : 1400 s)

Visualization of Particles

The particle visualization is described in detail on pages 61ff. When visualizing results of filtration simulations on the pleat scale or filter element scale, it is often necessary to enlarge the particles for better visibility (see page 66 how set the **Diameter Factor**), 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 buildup 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.

Visualization of Deposited Dust

Two options exists to visualize the simulation of deposited dust: **Load Deposited Dust** and **Load Deposited Dust as Voxel Structure**.

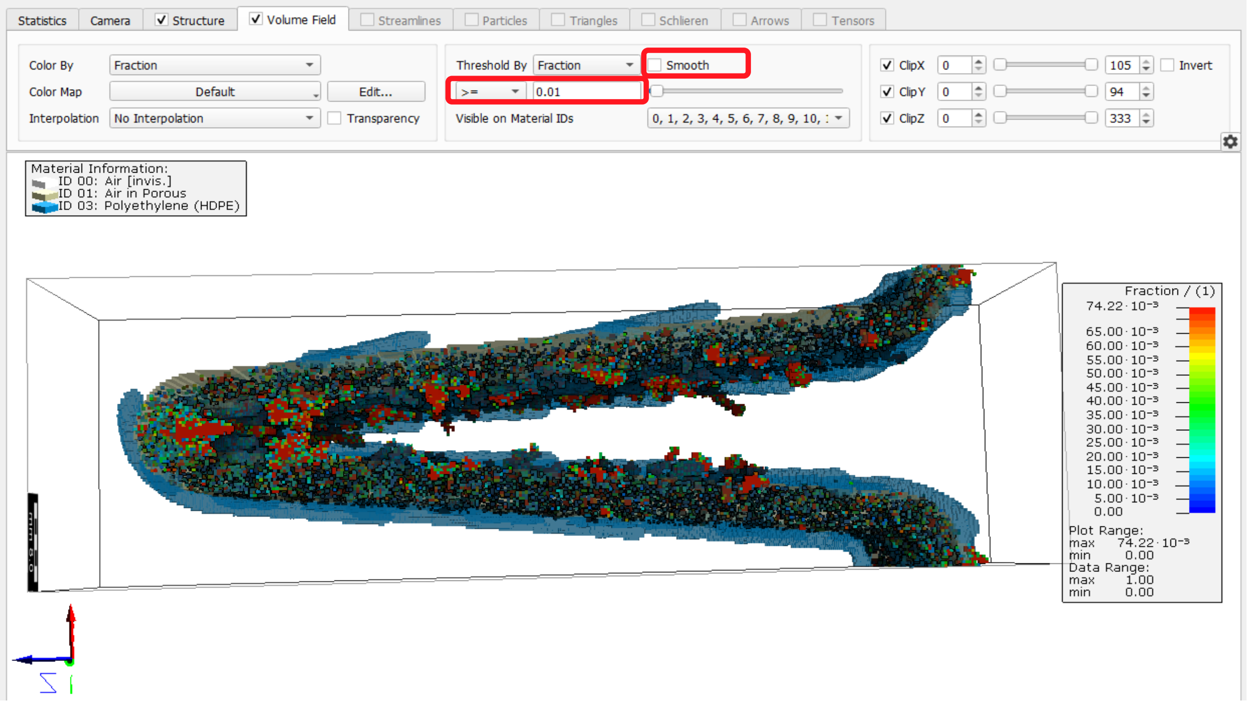
Load Deposited Dust is recommended to visualize the filter cake on the filter element scale. This loads the result field (*.gvf) containing, for every voxel, a single double value in the range [0,1], that describes the volume fraction of the voxel filled with dust particles.

With **Load Deposited Dust as Voxel Structure**, the volume field is converted to a voxel structure, and so turned into an empty/solid model. Thus, it does not allow a detailed visualization of a filter cake that mainly consists of partially filled voxels.

Load Deposited Dust

In the Result Viewer, under the **Particle Visualization** tab, in the **Deposited Dust** panel, choose the Time Step of interest, and click **Load Deposited Dust**.

For a good visualization, clip away all the unfilled voxels to see e.g., only the voxels filled by more than 1%. Also, switch off **Smooth** to see every individual voxel with its corresponding filling.

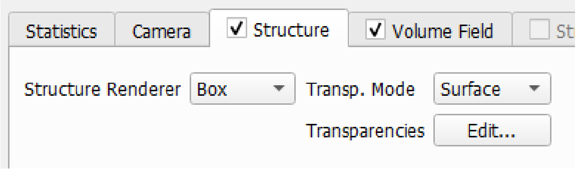


Color By: Fraction
Color Map: Default
Interpolation: No Interpolation
□ Transparency
Threshold By: Fraction
□ Smooth
>= 0.01
Visible on Material IDs: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
☑ ClipX 0 105
☑ ClipY 0 94
☑ ClipZ 0 333
□ Invert

Material Information:
ID 00: Air [invis.]
ID 01: Air in Porous
ID 03: Polyethylene (HDPE)

Plot Range:
max 74.22·10⁻³
min 0.00
Data Range:
max 1.00
min 0.00

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



Structure Renderer Box

Transp. Mode Surface

Transparencies

For more information about the visualization option, see the Visualization handbook of this User Guide.

Visualization of Flow Field

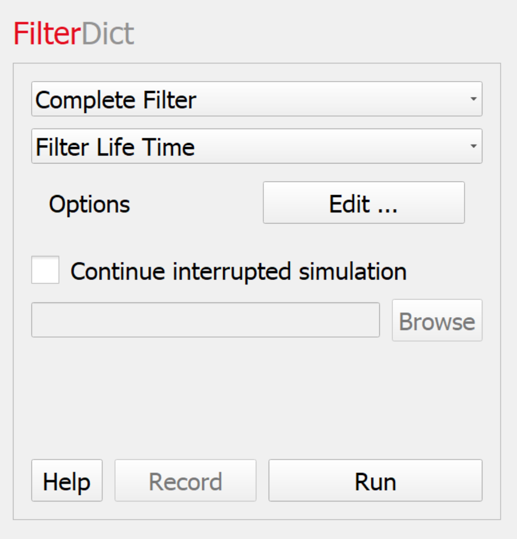
For the visualization of the flow field, please refer to the FlowDict and Visualization handbooks of this User Guide.

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** and **Filter Element** 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.



FilterDict

Complete Filter

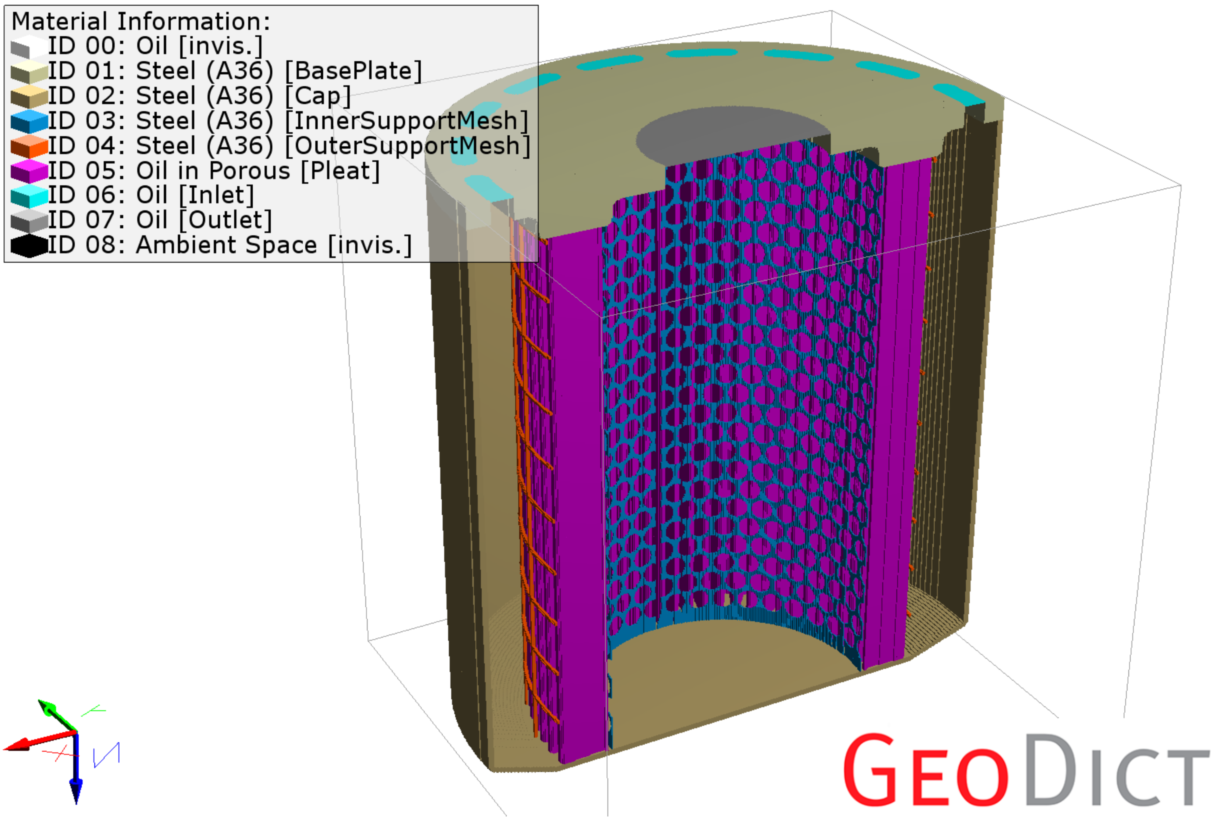
Filter Life Time

Options Edit ...

□ Continue interrupted simulation

Browse

Help Record Run

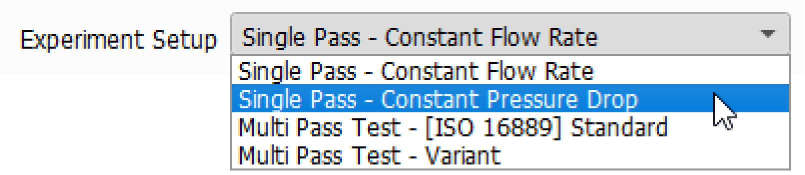

Material Information:
ID 00: Oil [invis.]
ID 01: Steel (A36) [BasePlate]
ID 02: Steel (A36) [Cap]
ID 03: Steel (A36) [InnerSupportMesh]
ID 04: Steel (A36) [OuterSupportMesh]
ID 05: Oil in Porous [Pleat]
ID 06: Oil [Inlet]
ID 07: Oil [Outlet]
ID 08: Ambient Space [invis.]
GeoDict

As for filter media simulations, an interrupted simulation can be restarted with **Continue interrupted simulation**. This is possible when the simulation was stopped after the calculation of the first flow field or later. When the simulation was stopped earlier, it must be started anew. Details for restarting an interrupted simulation can be found on page 16.

The **Complete Filter Simulation** dialog opens when clicking the **Edit...** button. As for filter media simulations, it is organized into the tabs: **Filter Experiment**, **Constituent Materials**, **Particles**, **Solver**, **Output** and **Equations & References.**.

Filter Experiment

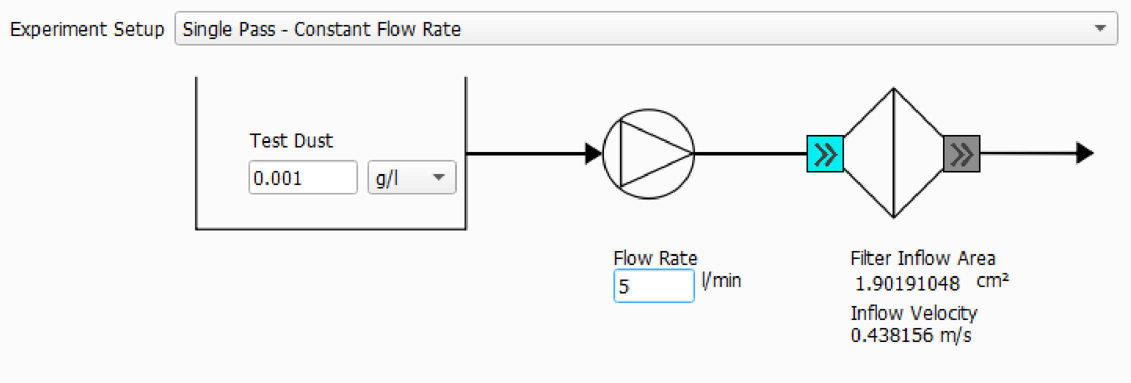
Under the **Filter Experiment** tab, choose between a **Single Pass** or a **Multi Pass** experiment. For **Single Pass**, the experiment can be run at a **Constant Flow Rate** or at a **Constant Pressure Drop** (often used for membranes).


Experiment Setup
Single Pass - Constant Flow Rate
Single Pass - Constant Flow Rate
Single Pass - Constant Pressure Drop
Multi Pass Test - [ISO 16889] Standard
Multi Pass Test - Variant

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 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) and the **Flow Rate** (in l/min).



Experiment Setup: Single Pass - Constant Flow Rate

Test Dust: 0.001 g/l

Flow Rate: 5 l/min

Filter Inflow Area: 1.90191048 cm²

Inflow Velocity: 0.438156 m/s

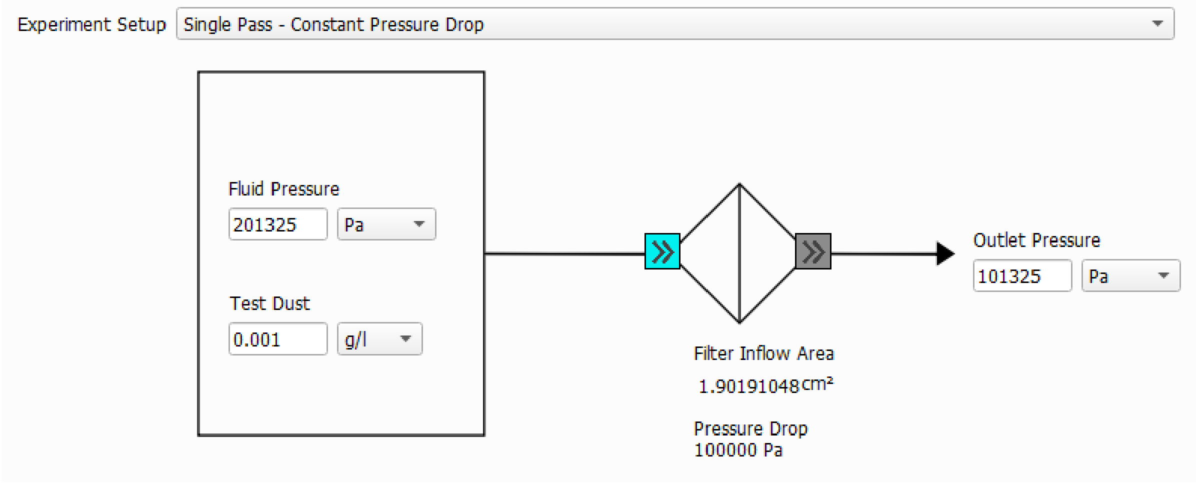
The arrow symbols before and after the filter show the color of the inflow material (here cyan ») and the outflow material (here dark grey »). **Filter Inflow Area** and **Inflow Velocity** are computed automatically.

The combination of the two parameters Test Dust concentration and Flow Rate controls 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**, the **Test Dust** concentration in the fluid. As described above, the arrow symbols before and after the filter show the color of the inflow material (here cyan ») and the outflow material (here dark grey ») and the **Filter Inflow Area** is computed automatically.



Experiment Setup: Single Pass - Constant Pressure Drop

Fluid Pressure: 201325 Pa

Test Dust: 0.001 g/l

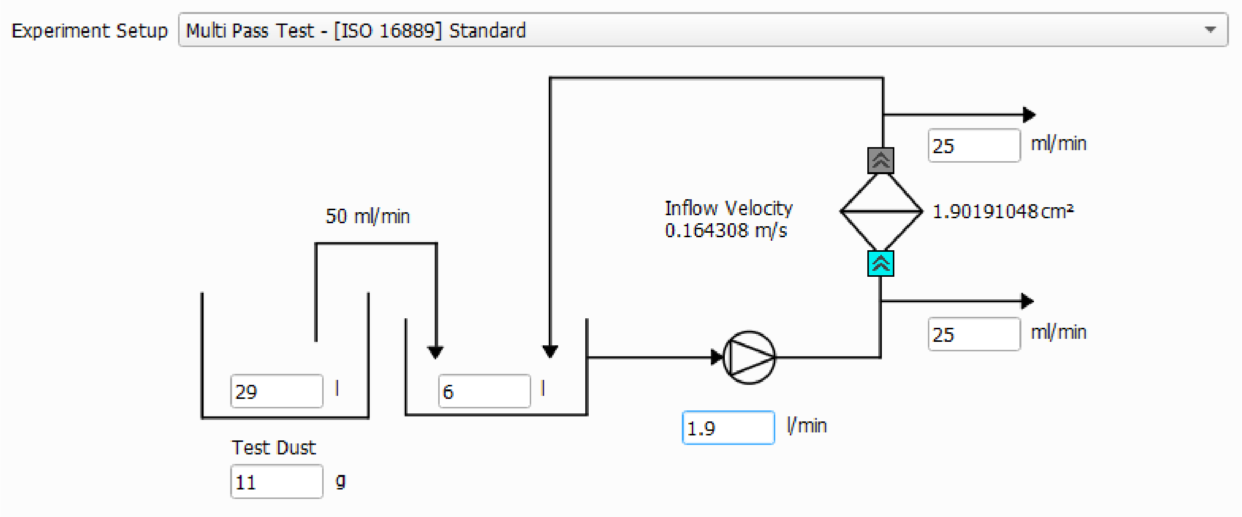
Outlet Pressure: 101325 Pa

Filter Inflow Area: 1.90191048 cm²

Pressure Drop: 100000 Pa

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.



Experiment Setup: Multi Pass Test - [ISO 16889] Standard

50 ml/min

29 l

6 l

Test Dust: 11 g

1.9 l/min

Inflow Velocity: 0.164308 m/s

1.90191048 cm²

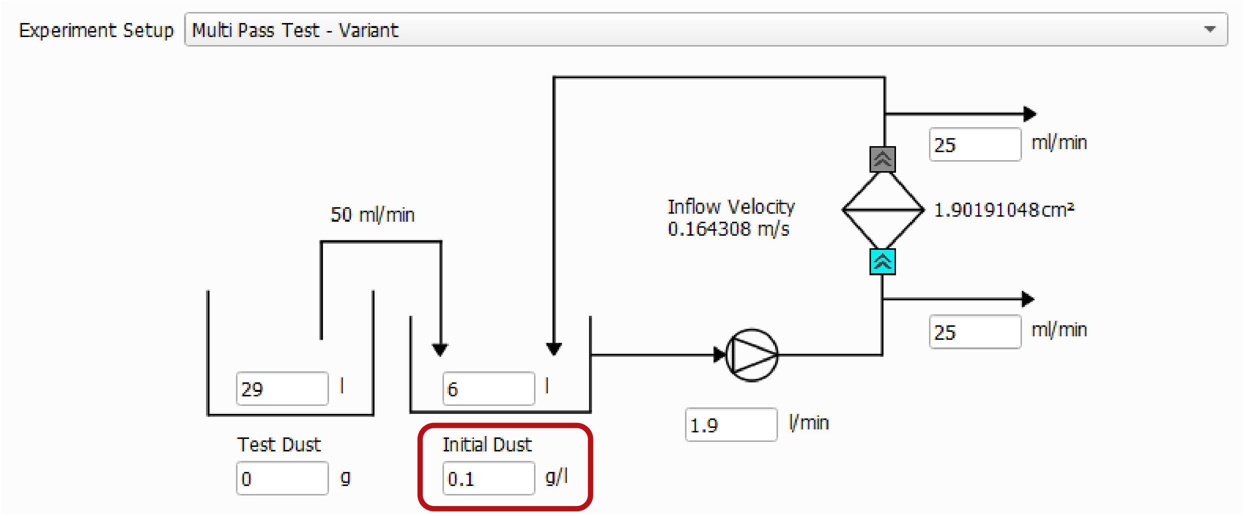
25 ml/min

25 ml/min

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). Again, the arrow symbols before and after the filter show the color of the inflow material (here cyan ») and the outflow material (here dark grey ») and **Filter Inflow Area** (here 1.90191 cm²) 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. Again, the arrow symbols before and after the filter show the color of the inflow material (here cyan ») and the outflow material (here dark grey ») and **Filter Inflow Area** (here 1.90191 cm²) and **Inflow Velocity** are computed automatically.



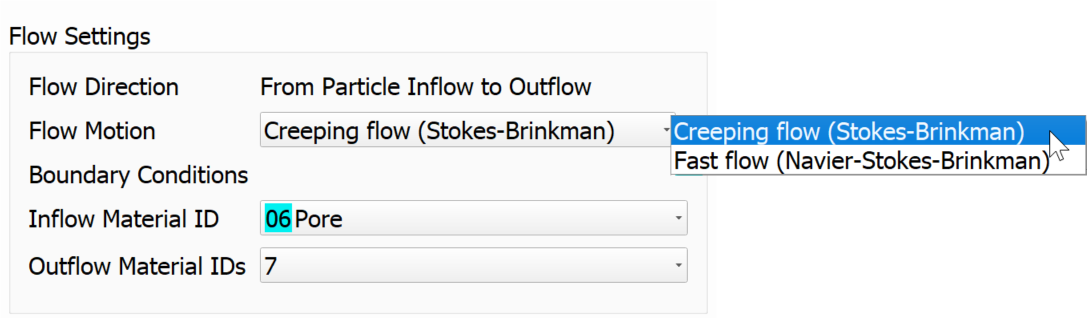
Experiment Setup: Multi Pass Test - Variant

Simulation Stopping Criterion

See the explanation on page 39.

Flow Settings

In this command, the **Flow Direction** is fixed to **From Particle Inflow to Outflow**.



Flow Settings

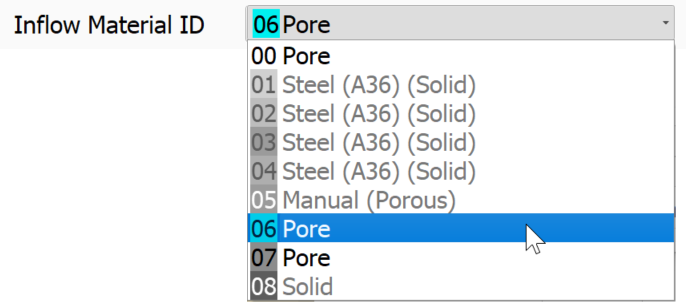
Flow Direction	From Particle Inflow to Outflow
Flow Motion	Creeping flow (Stokes-Brinkman)
Boundary Conditions	
Inflow Material ID	06 Pore
Outflow Material IDs	7

Creeping flow (Stokes-Brinkman)
Fast flow (Navier-Stokes-Brinkman)

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.

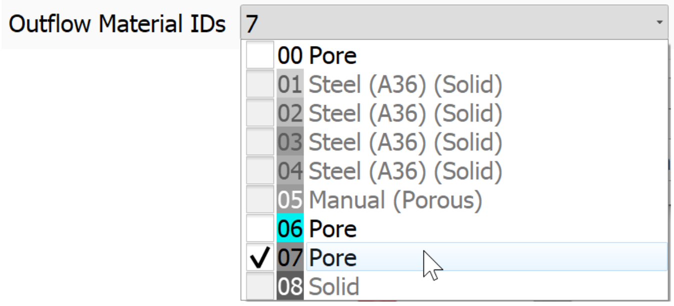
As the Stokes equation is a simplification of the Navier-Stokes equation, choosing **Fast flow (Navier-Stokes)** will give correct results also in the case of slow, creeping flow. Solving the Navier-Stokes equations is computationally more expensive than solving the Stokes equations, so it is advisable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

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.



Inflow Material ID: 06 Pore

As **Outflow Material**, one or several material IDs can be selected. After selecting a material ID here, the » symbol changes to the selected colors

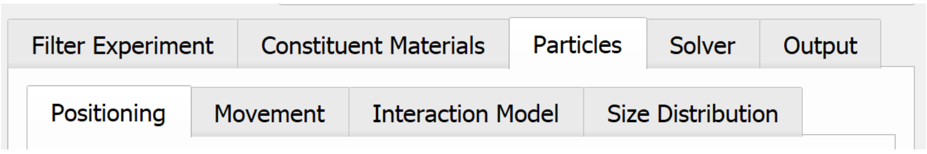

Outflow Material IDs 7

Constituent Materials

See the explanation on page 40.

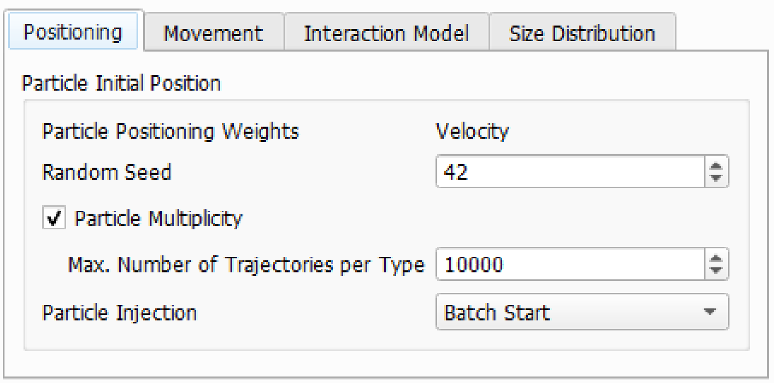
Particles

Under the **Particles** tab, four subtabs are available to define **Positioning**, **Movement**, **Interaction Model** and **Size Distribution**.



Positioning

The particles are placed randomly in the inflow area defined in the **Filter Experiment** 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. (see the description on **Particle Positioning Weights** on page 74)



Particle Initial Position

Particle Positioning Weights Velocity

Random Seed 42

☑ Particle Multiplicity

Max. Number of Trajectories per Type 10000

Particle Injection Batch Start

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.

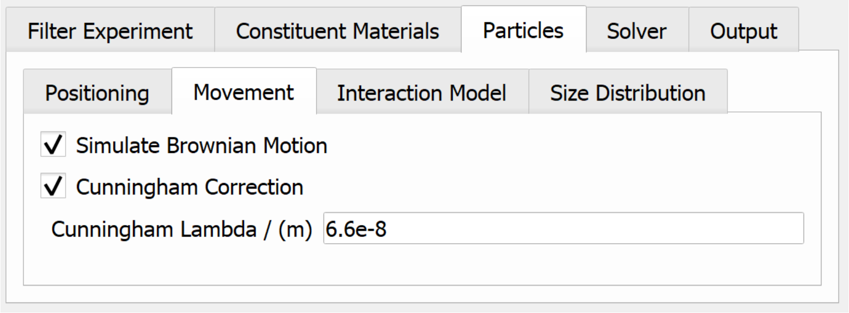
Particle Multiplicity

See the description on page 45.

Particle Injection

See the description on page 45.

Movement



☑ Simulate Brownian Motion

☑ Cunningham Correction

Cunningham Lambda / (m) 6.6e-8

For **Simulate Brownian Motion**, **Cunningham Correction**, see pages 25ff.

It is not possible to model electrostatic effects if parts of the filter structure are unresolved, therefore this option is not selectable here.

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.

Interaction Model and Size Distribution

See the descriptions on page 75ff.

Solver

The **EJ solver** is not available as **Initial** or **Iterative Flow Solver** because it cannot solve Stokes-Brinkman or Navier-Stokes-Brinkman equations, needed for filtration simulations for material models with porous voxels.

Otherwise, the options under the **Solver** tab are the same as explained on pages 49ff.

Output

The parameters here are the same as for the **Output** tab for the Filter Media simulation, starting on page 54.

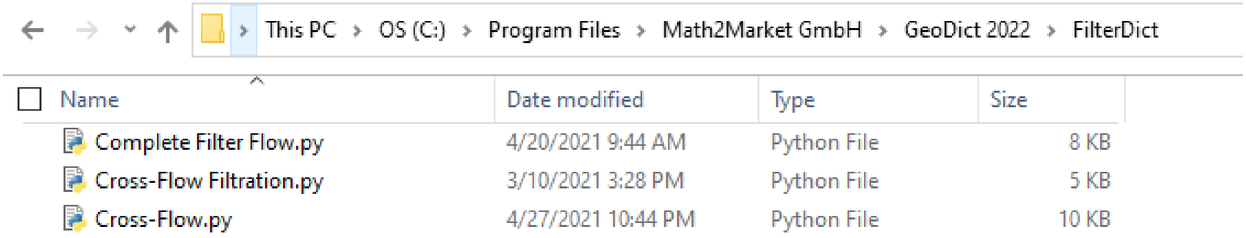
Results

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

The result report and the visualization options are almost identical to those of the Filter Element simulations (see page 78ff for details). Because of the arbitrary positions of inflow and outflow, it does not make sense to plot average pressure values per z-layer, so the **Plot layered pressure per batch** option is not available here. An example can be found in the Digital modelling and simulation of a complete filter tutorial.

PREDEFINED

When **Predefined** is selected, all GeoDict python macros stored in the GeoDict installation folder C:\Program Files\Math2Market GmbH\GeoDict 2022\FilterDict are shown.



This PC › OS (C:) › Program Files › Math2Market GmbH › GeoDict 2022 › FilterDict

Name	Date modified	Type	Size
Complete Filter Flow.py	4/20/2021 9:44 AM	Python File	8 KB
Cross-Flow Filtration.py	3/10/2021 3:28 PM	Python File	5 KB
Cross-Flow.py	4/27/2021 10:44 PM	Python File	10 KB

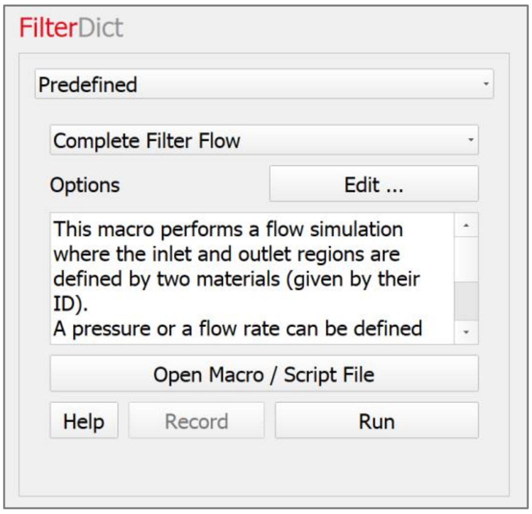
Three macros are included in the standard installation package, but additional macros could be placed there by the system administrator and would then be available for all users.

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.

COMPLETE FILTER FLOW

The **Complete Filter Flow** script allows to simulate the flow through a complete filter. A filter life time simulation is not included, but it could be considered as a simplified version of the **Complete Filter** command.



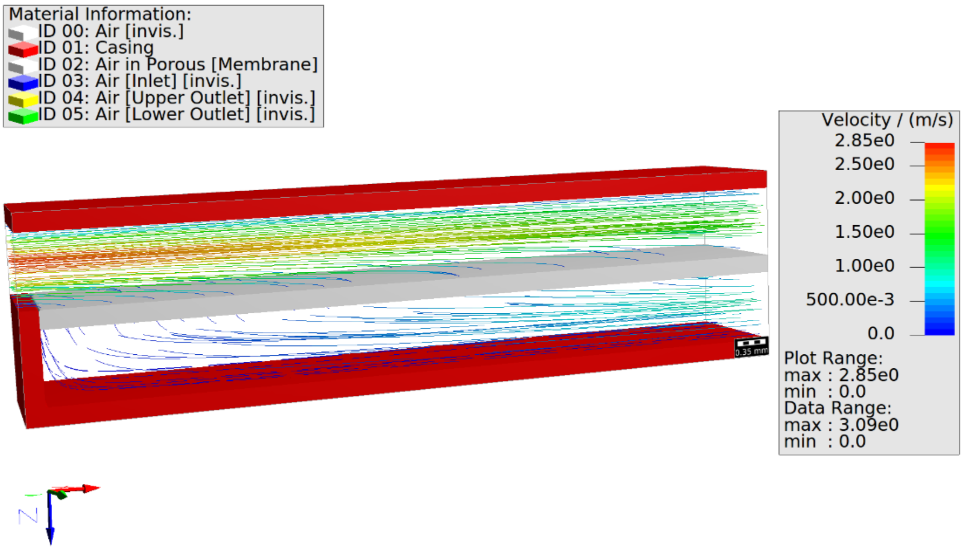
This macro performs a flow simulation where the inlet and outlet regions are defined by two materials (given by their ID).
A pressure or a flow rate can be defined

The settings for the complete filter flow simulation can be set by clicking the **Edit ...** button.

The Digital modelling and simulation of a complete filter tutorial shows how to prepare a structure model and use this macro to simulate the flow through a complete 3D filter model including housing, pleats, support structures, inflow and outflow.

Cross-Flow

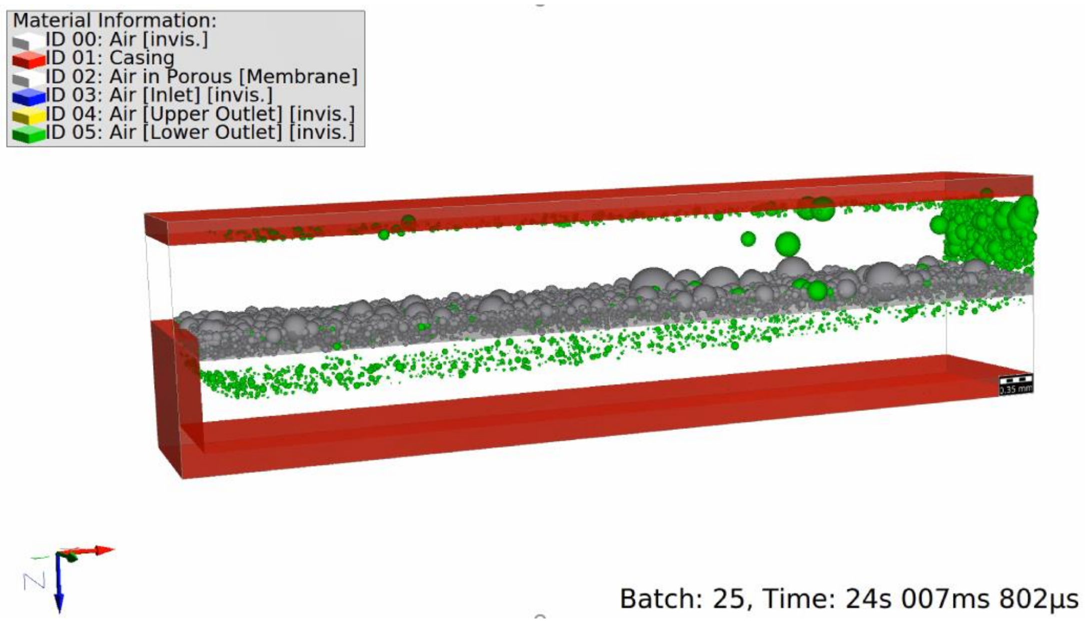
The **Cross-Flow** predefined script runs a flow simulation similar to the **Complete Filter Flow** script. To enable a cross-flow simulation, one inlet and two outlets with different outlet pressures have to be defined by the user. Defining the positions of the inlet and outlets works analogous to the **Complete Filter Flow** predefined script.



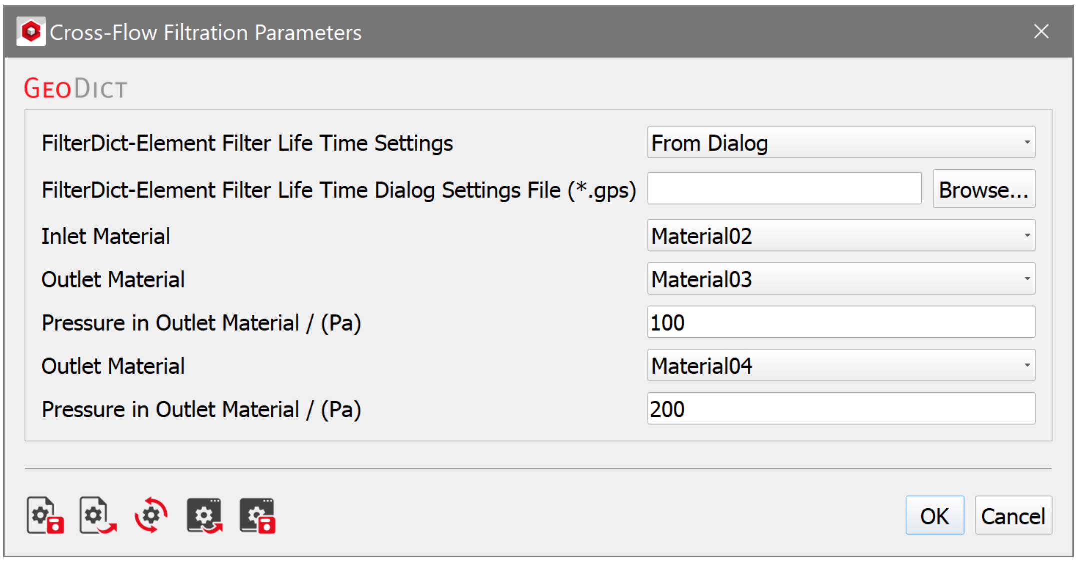
Please contact support@math2market.de if you are interested in running simulations of cross flow filtration.

Cross-Flow Filtration

The **Cross-Flow Filtration** predefined script allows to set up a filtration simulation with one inlet, but with two outlets which have different outlet pressures. Defining the positions of the inlet and outlets works analogous to the Complete Filter Flow predefined script.


Material Information:
ID 00: Air [invis.]
ID 01: Casing
ID 02: Air in Porous [Membrane]
ID 03: Air [Inlet] [invis.]
ID 04: Air [Upper Outlet] [invis.]
ID 05: Air [Lower Outlet] [invis.]
0.35 mm
Z
Batch: 25, Time: 24s 007ms 802µs

The settings for the cross-flow filtration simulation can be defined by clicking the **Edit ...** button. Here, most of the parameters used are directly loaded from the Filter Element – Filter Life Time dialog or a corresponding settings file (*.gps).



FilterDict-Element Filter Life Time Settings	From Dialog
FilterDict-Element Filter Life Time Dialog Settings File (*.gps)	
Inlet Material	Material02
Outlet Material	Material03
Pressure in Outlet Material / (Pa)	100
Outlet Material	Material04
Pressure in Outlet Material / (Pa)	200

Please contact support@math2market.de if you are interested in running simulations of cross flow filtration.

Appendix I: User-Defined Functions (UDFs)

Four different types of User-Defined Functions (UDF) are used in FilterDict:

- The **Particle motion UDF**, to change the equations that describe the particle motion.
- The **Collision UDF**, to define collision models in addition to the in-built models (Caught on First Touch, Sieving, and Hamaker)
- The **Pass Through UDF**, to define alternative pass-through models.
- The **User Data UDF** can be used in combination with one of the other UDF types. It allows to attach additional data to a particle, which can be used and modified during the simulation.

During the first startup, GeoDict copies a number of sample UDFs into the user's directory at:

- /home/*username*/.geodict2022/UDF/ (Linux)
- C:\Users\username\GeoDict2022\UDF\ (Windows).

These files contain Particle Motion UDFs:

- ParticleMotionUDF-Standard.cpp
- ParticleMotionUDF-Gravity.cpp
- ParticleMotionUDF-GravityPlusBuoyancy.cpp

These files contain Collision UDFs:

- CollisionUDF-Count.cpp
- CollisionUDF-Hamaker.cpp

This file contains a Pass Through UDF:

- PassThroughUDF.cpp

The files UDFUserData.h and UDFUserData.cpp can be used to define and initialize the user defined struct.

GeoDict does not compile the source code (*.cpp) automatically. Therefore, the user must compile the UDF before using it in GeoDict, as described below.

Particle motion UDF

The Particle Motion UDF allows changing the equations (10), (11), (12), and (13) of pages 5ff.:

$$\gamma = 6\pi\mu\frac{R}{C_c} \tag{10}$$

$$C_c = 1 + \frac{\lambda}{R}\left(1.17 + 0.525e^{-0.78\frac{R}{\lambda}}\right) \tag{11}$$

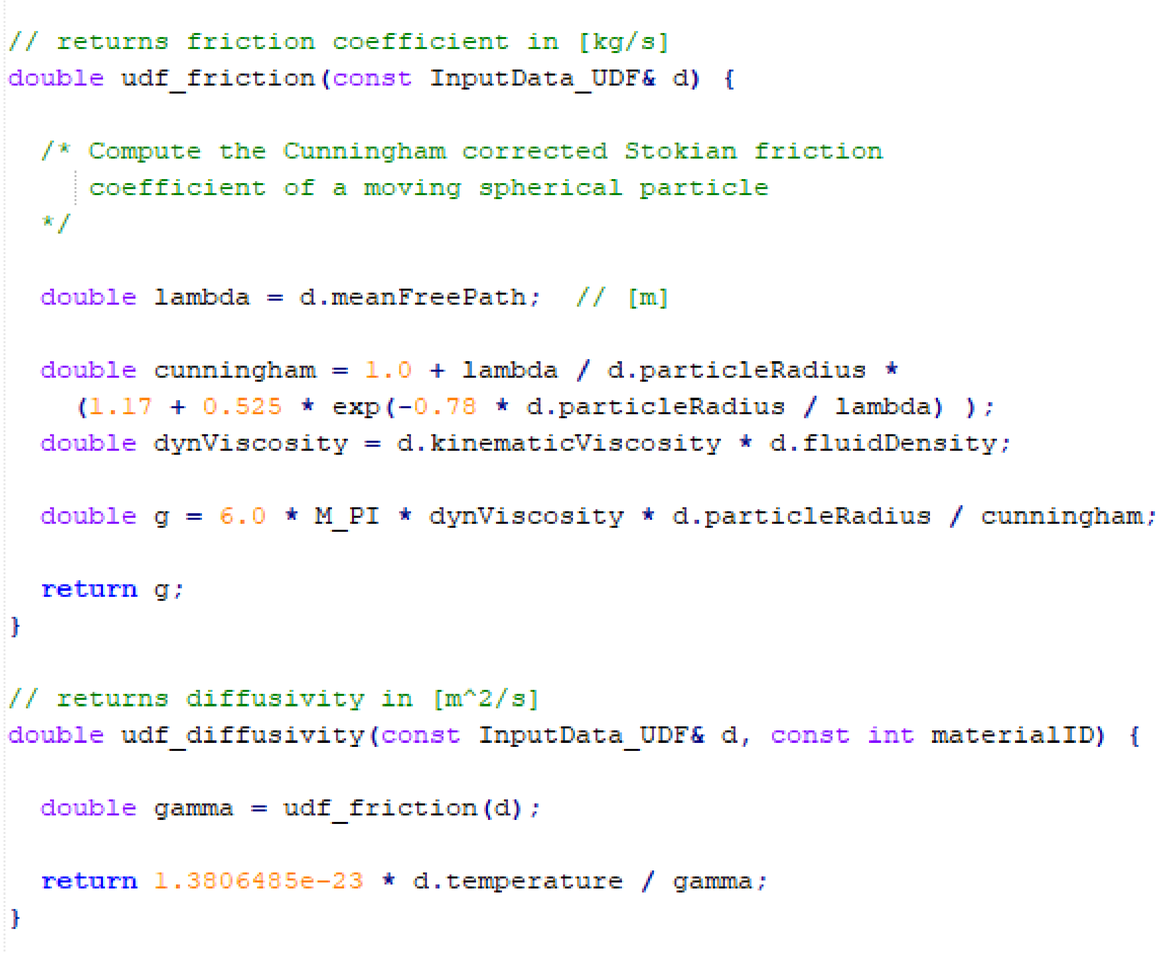
$$D = \frac{k_B T}{\gamma} \tag{12}$$

$$\vec{F} = 0 \tag{13}$$

This UDF contains three functions: `udf_friction()`, `udf_diffusivity()`, and `udf_force()`.

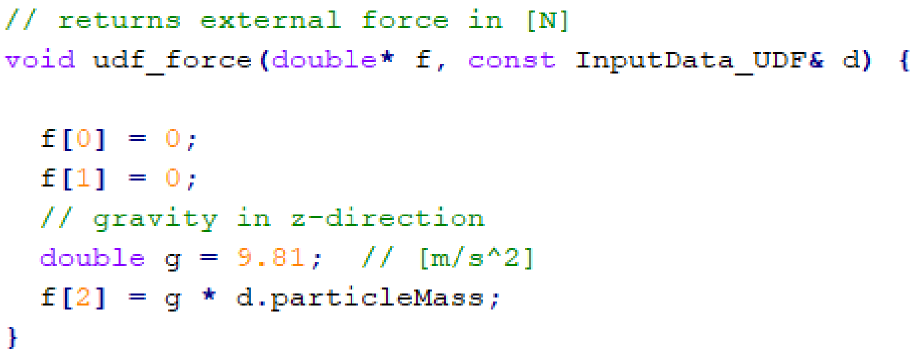
The first function `udf_friction()` contains the definition of the Cunningham correction and returns γ. The second function returns D. The last function returns the external force $\vec{F}$.

As an example, the file **ParticleMotionUDF-Standard.cpp** shows the formulas as implemented in GeoDict:



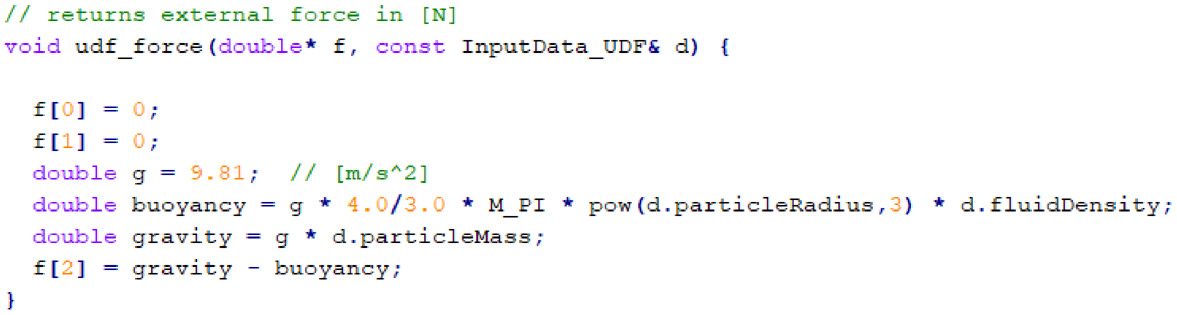
```
// returns friction coefficient in [kg/s]
double udf_friction(const InputData_UDF& d) {

  /* Compute the Cunningham corrected Stokian friction
     coefficient of a moving spherical particle
  */

  double lambda = d.meanFreePath;  // [m]

  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 diffusivity in [m^2/s]
double udf_diffusivity(const InputData_UDF& d, const int materialID) {

  double gamma = udf_friction(d);

  return 1.3806485e-23 * d.temperature / gamma;
}
```

Another example is the file **ParticleMotionUDF-Gravity.cpp**, where f is used to model gravity:



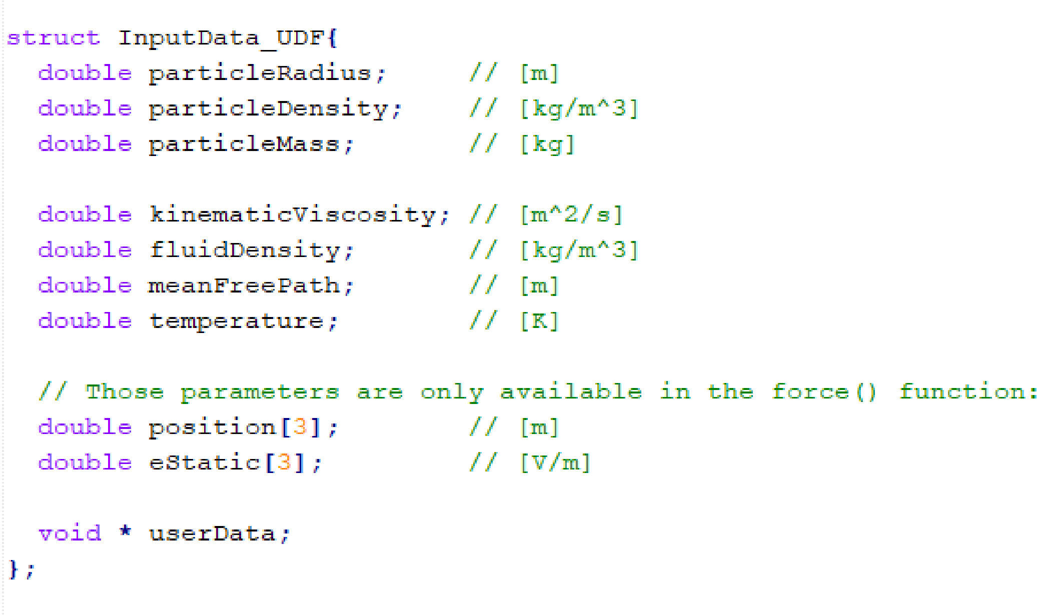
```
// returns external force in [N]
void udf_force(double* f, const InputData_UDF& d) {

  f[0] = 0;
  f[1] = 0;
  // gravity in z-direction
  double g = 9.81;  // [m/s^2]
  f[2] = g * d.particleMass;
}
```

To include e.g., gravity and buoyancy, the function can be changed to:



```
// returns external force in [N]
void udf_force(double* f, const InputData_UDF& d) {

  f[0] = 0;
  f[1] = 0;
  double g = 9.81;  // [m/s^2]
  double buoyancy = g * 4.0/3.0 * M_PI * pow(d.particleRadius,3) * d.fluidDensity;
  double gravity = g * d.particleMass;
  f[2] = gravity - buoyancy;
}
```

The input data struct `InputData_UDF` is defined in the ParticleMotionStructs.h header file which can be found in the `include` folder and contains the following data:



```
struct InputData_UDF{
  double particleRadius;     // [m]
  double particleDensity;    // [kg/m^3]
  double particleMass;       // [kg]

  double kinematicViscosity; // [m^2/s]
  double fluidDensity;       // [kg/m^3]
  double meanFreePath;       // [m]
  double temperature;        // [K]

  // Those parameters are only available in the force() function:
  double position[3];        // [m]
  double eStatic[3];         // [V/m]

  void * userData;
};
```

Collision UDF

A collision UDF contains three functions.

The first function

```
int udf_numberOfMaterialParameters();
```

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, and the second function

```
const char* udf_nameOfMaterialParameters(int i);
```

allows to set the headings that will appear in the Particles tab table.

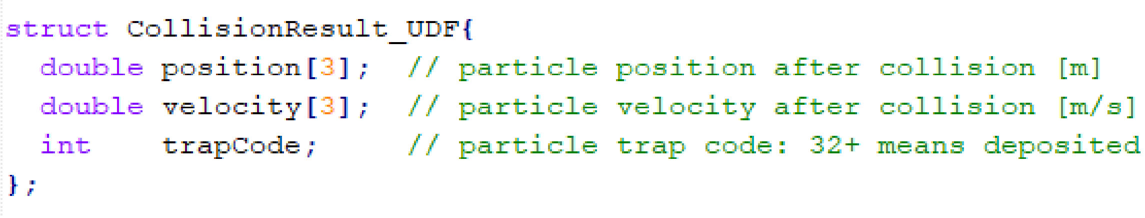
The third function describes the collision of a particle with a solid.

```
CollisionResult_UDF udf_collision(ParticleData_UDF& p,
                                  CollisionData_UDF& c,
                                  const double* materialParameters);
```

Input data are

```
struct ParticleData_UDF{
  double diameter;              // [m]
  double collisionDiameter;     // [m]
  double depositionDiameter;    // [m]
  double density;               // [g/l] = [kg/m^3]
  double prevPosition[3];       // [m]
  double position[3];           // [m]
  double velocity[3];           // [m/s]
  double travelTime;            // [s]
  double diffusivity;           // [m^2/s]
  int    collisionCounter[16]; // total number of collisions with material[i]
  int    multiplicity;
  int    status;                // see comment below
  double residenceTime[16];     // total residence time of particle in material[i]
  int64_t particleID;

  void * userData;
};

struct CollisionData_UDF{
  double normal[3];
  int    material;
};
```

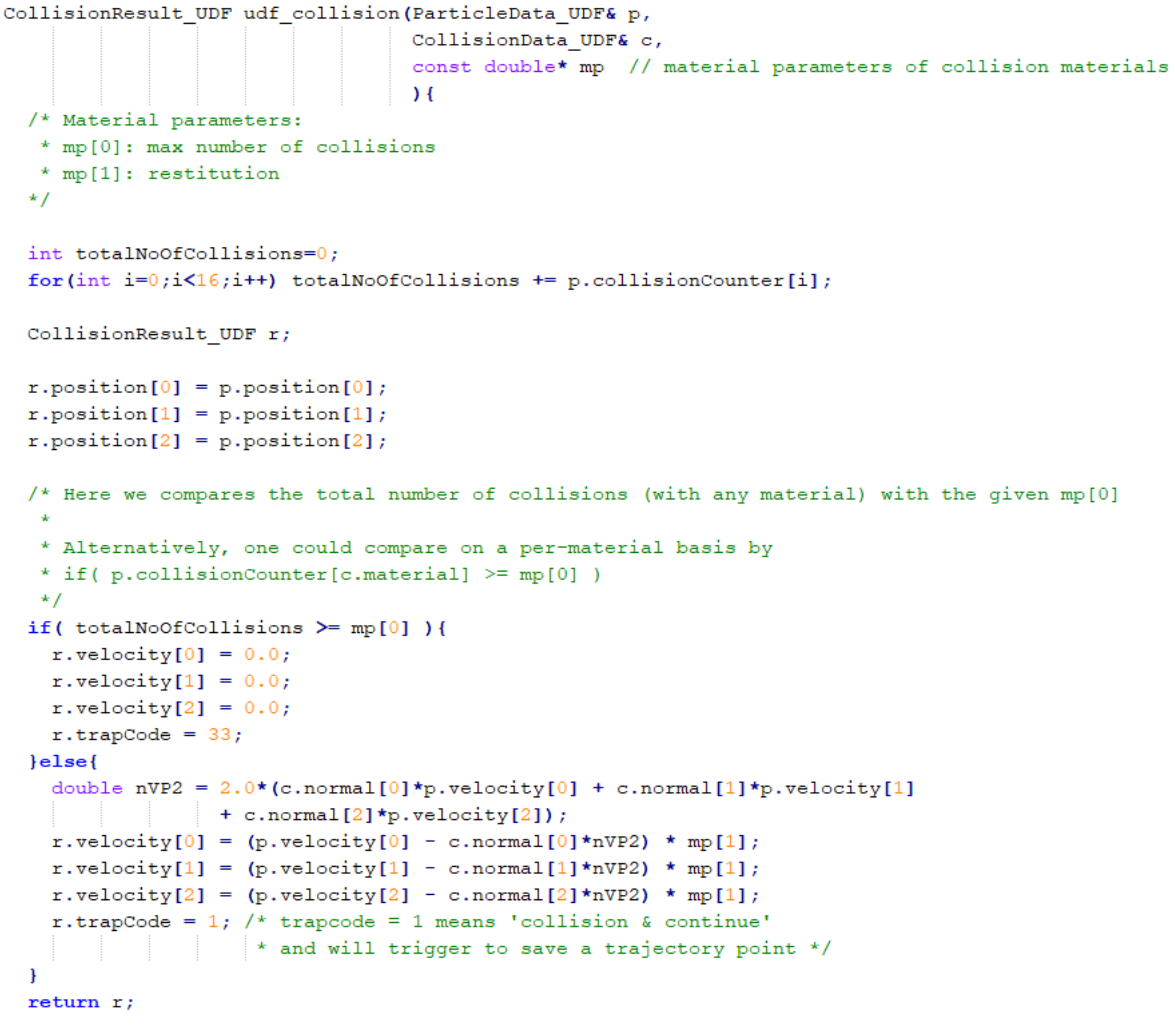
where the `*materialParameters` pointer contains the material parameters of the filter material at the collision point defined through the FilterDict dialog. This array has as many entries as defined in the `udf_numberOfMaterialParameters` function. The return value of the function is a struct that contains particle position and velocity after the collision and a trap code.



```
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
};
```

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.

As an example, look at the functions defined in CollisionUDF-Count.cpp. The CollisionUDF-Count.cpp needs two material parameters: the max number of collisions and the restitution:



```
CollisionResult_UDF udf_collision(ParticleData_UDF& p,
                                  CollisionData_UDF& c,
                                  const double* mp  // material parameters of collision materials
                                  ){
  /* Material parameters:
   * mp[0]: max number of collisions
   * mp[1]: restitution
  */

  int totalNoOfCollisions=0;
  for(int i=0;i<16;i++) totalNoOfCollisions += p.collisionCounter[i];

  CollisionResult_UDF r;

  r.position[0] = p.position[0];
  r.position[1] = p.position[1];
  r.position[2] = p.position[2];

  /* Here we compares the total number of collisions (with any material) with the given mp[0]
   *
   * Alternatively, one could compare on a per-material basis by
   * if( p.collisionCounter[c.material] >= mp[0] )
   */
  if( totalNoOfCollisions >= mp[0] ){
    r.velocity[0] = 0.0;
    r.velocity[1] = 0.0;
    r.velocity[2] = 0.0;
    r.trapCode = 33;
  }else{
    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] = (p.velocity[0] - 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;
```

1. The total number of previous collisions is determined.
2. The new particle position is set as the current particle position.
3. The total number of collisions is compared to the allowed number.
   a. If the number of collisions is higher, the particle is trapped, i.e. the velocity is set to 0 and the trapCode to 33.
   b. If the number of collisions is lower, the particle bounces off, but the new velocity is reduced by a restitution factor.

Pass Through UDF

Similar to a Collision UDF, a Pass Through UDF contains three functions.

The first function

```
int udf_numberOfMaterialParameters();
```

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, and again a second function

```
const char* udf_nameOfMaterialParameters(int i);
```

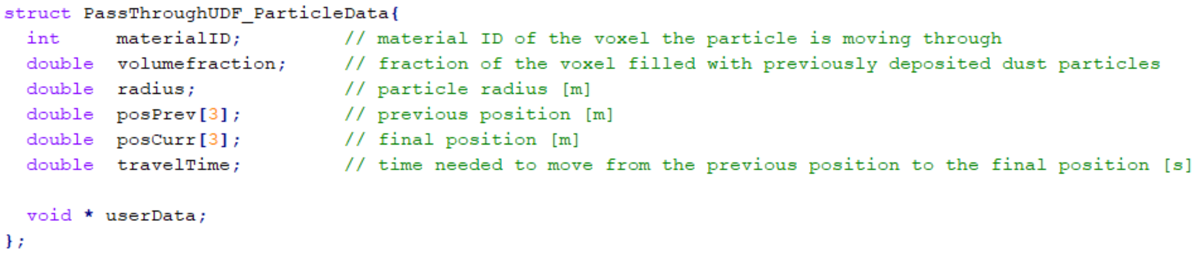
allows to set the headings that will appear in the Particles tab table.

The third function computes the passing probability of a particles moving through a porous material:

```
double_UDF udf_passThrough(PassThroughUDF_ParticleData& p,
                           const double* materialParameters);
```

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.

Input data are



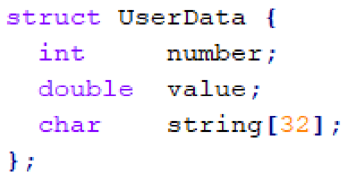
```
struct PassThroughUDF_ParticleData{
  int     materialID;          // material ID of the voxel the particle is moving through
  double  volumefraction;      // fraction of the voxel filled with previously deposited dust particles
  double  radius;              // particle radius [m]
  double  posPrev[3];          // 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;
};
```

and a *materialParameters pointer containing the material parameters of the filter material defined through the FilterDict dialog. This array has as many entries as defined in the udf_numberOfMaterialParameters function.

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 the user to attach arbitrary data to a particle, possibly modify this data at any particle-wall collision and use it during particle motion.

An example struct is defined in the file UDFUserData.h



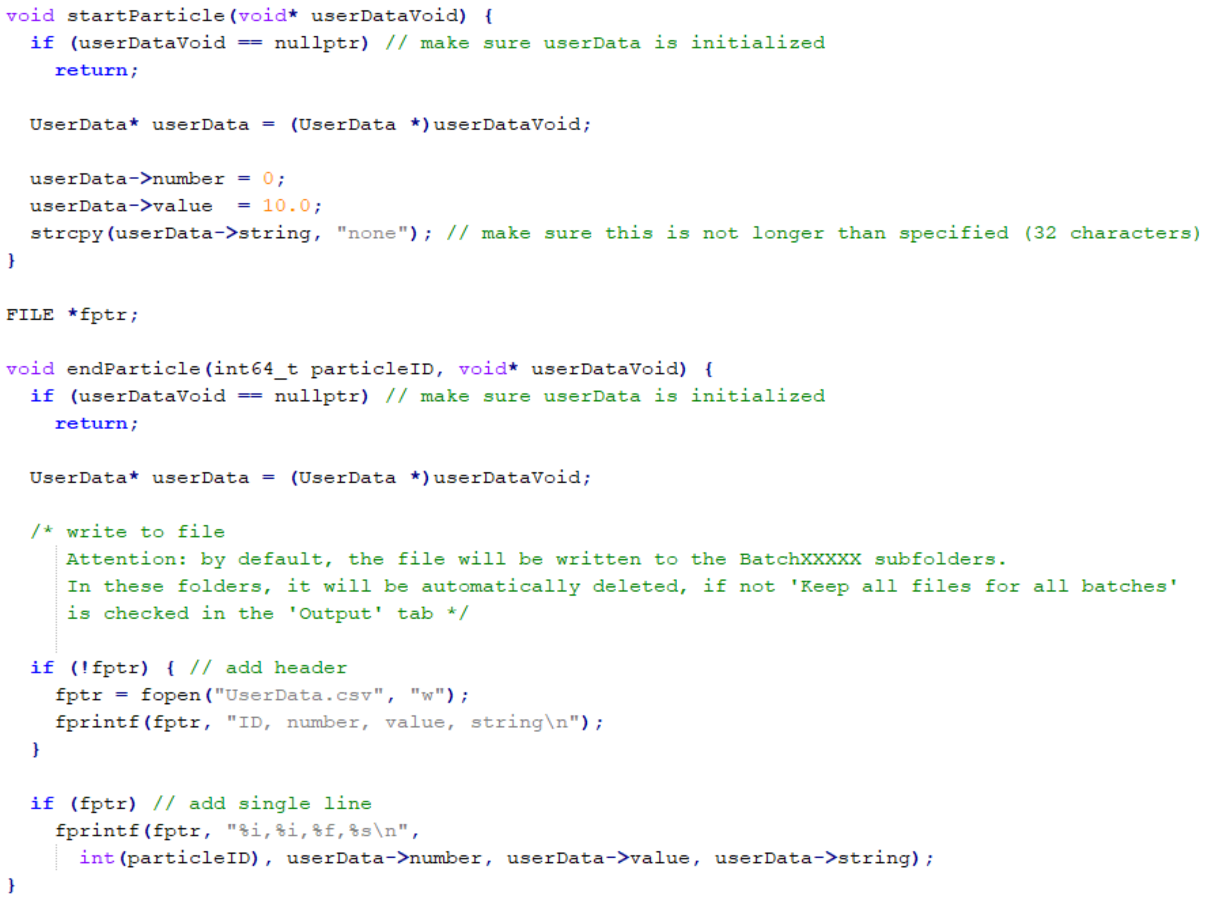
```
struct UserData {
  int     number;
  double  value;
  char    string[32];
};
```

The user 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 excluded 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. In the default case, all UserData is initialized with 0.

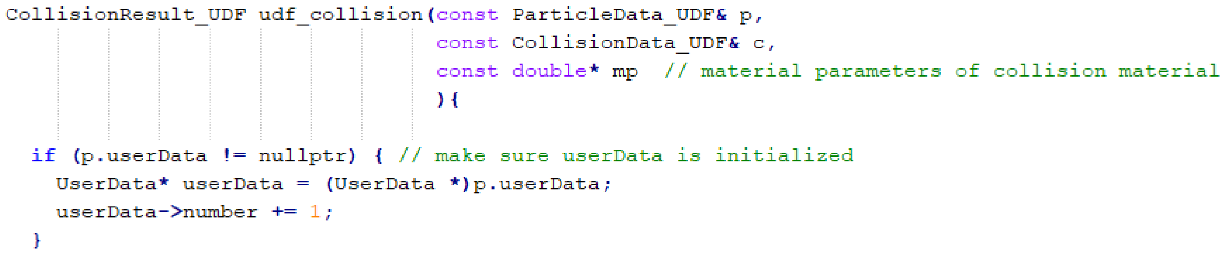
The `endParticle` function will be called at the end of a particle run, such that UserData can be printed or saved to a file. This function will always be called on the main thread (not parallelized). Be aware that this function is not be called in the order of the particleID's.

An example can be found in the file UDFUserData.cpp:



```
void startParticle(void* userDataVoid) {
  if (userDataVoid == nullptr) // make sure userData is initialized
    return;

  UserData* userData = (UserData *)userDataVoid;

  userData->number = 0;
  userData->value  = 10.0;
  strcpy(userData->string, "none"); // make sure this is not longer than specified (32 characters)
}

FILE *fptr;

void endParticle(int64_t particleID, void* userDataVoid) {
  if (userDataVoid == nullptr) // make sure userData is initialized
    return;

  UserData* userData = (UserData *)userDataVoid;

  /* write to file
     Attention: by default, the file will be written to the BatchXXXXX subfolders.
     In these folders, it will be automatically deleted, if not 'Keep all files for all batches'
     is checked in the 'Output' tab */

  if (!fptr) { // add header
    fptr = fopen("UserData.csv", "w");
    fprintf(fptr, "ID, number, value, string\n");
  }

  if (fptr) // add single line
    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:



```
CollisionResult_UDF udf_collision(const ParticleData_UDF& p,
                                  const CollisionData_UDF& c,
                                  const double* mp  // material parameters of collision material
                                  ){

  if (p.userData != nullptr) { // make sure userData is initialized
    UserData* userData = (UserData *)p.userData;
    userData->number += 1;
  }
```

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.

Modifying and Compiling User Defined Functions

in Linux

The simplest way to work with UDFs is as follows:

1. change into the UDF directory with:

   ```
   cd ./geodict2022/UDF
   ```

2. copy one of the sample 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.

Requirements	gcc-c++ compiler and libraries installed

in Windows

To compile the three 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: ParticleMotionUDF-Standard.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 **ParticleMotionUDF-Standard.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.

Requirements	Microsoft Visual Studio 2013 (or later) installed

Appendix II: Particle status codes in *.gpp files

When opening *.gpp particle files (e.g., TrackerFinalParticles.gpp) in Matlab (with GeoLab) or in Python, the table below can be used to understand the particle status. When loading particles directly in GeoDict, the status is automatically converted.

```
-1 : initialize velocity from .vap file

0  : running & continue
1  : collision & continue

8  : particle left through z+ (outlet)
9  : particle left through z- (inlet)

10 : particle has reached end material ID
11 : particle reached maximum displacement (AddiDict only)
12 : time-out, particle will continue to move in next batch

13 : particle left through x+ (AddiDict only)
14 : particle left through x- (AddiDict only)
15 : particle left through y+ (AddiDict only)
16 : particle left through y- (AddiDict only)

32 : trapped by adhesion (Hamaker model)
33 : trapped by collision count (UDF)
34 : trapped by sieving
35 : trapped by caught on first touch
36 : trapped while passing through porous layers
37 : trapped by adsorption (AddiDict only)

40 : initial particle position overlaps with structure
```

REFERENCES

[1] D.J. Tritton, Physical fluid dynamics (2nd ed.). Oxford: Claredon Press.p. 58. ISBN 978 – 0198544937, 2006.

[2] 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.

[3] C.T. Crowe, Multiphase flow handbook, 1st edition, Taylor and Francis, 2006.

[4] C.T. Crowe, Multiphase Flows with Droplets and Particles, Second Edition-CRC Press, 2011.

[5] O. Iliev, V. Laptev, D. Vasileva, Algorithms and software for flow through oil filters. Filtech Europa, Volume I, pp. I-327-I334, October 2003.

[6] 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.

[7] R.C, Brown, D. Wake, Loading filters with monodisperse aerosols: Macroscopic treatment, Journal of Aerosol Science, Vol. 30 (2), pp. 227-234, 1999.

[8] H. Krupp, Particle adhesion, theory and experiment, Advances in Colloid and Interface Sci., 1967, 1 (2), 111.

[9] 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 (2018) 1-9. DOI: 10.1002/ceat.201700585

[10] J. Becker, L. Cheng, C. Kronsbein, A. Wiegmann, *Simulation of cake filtration for polydisperse particles*, Chemical Engineering & Technology 39 (2016) 559–566. DOI: 10.1002/ceat.201500350

https://doi.org/10.30423/userguide.geodict2022-filterdict

Technical documentation:

Jürgen Becker

Philipp Eichheimer

Barbara Planas

Math2Market GmbH

Richard-Wagner-Str. 1, 67655 Kaiserslautern, Germany
www.geodict.com