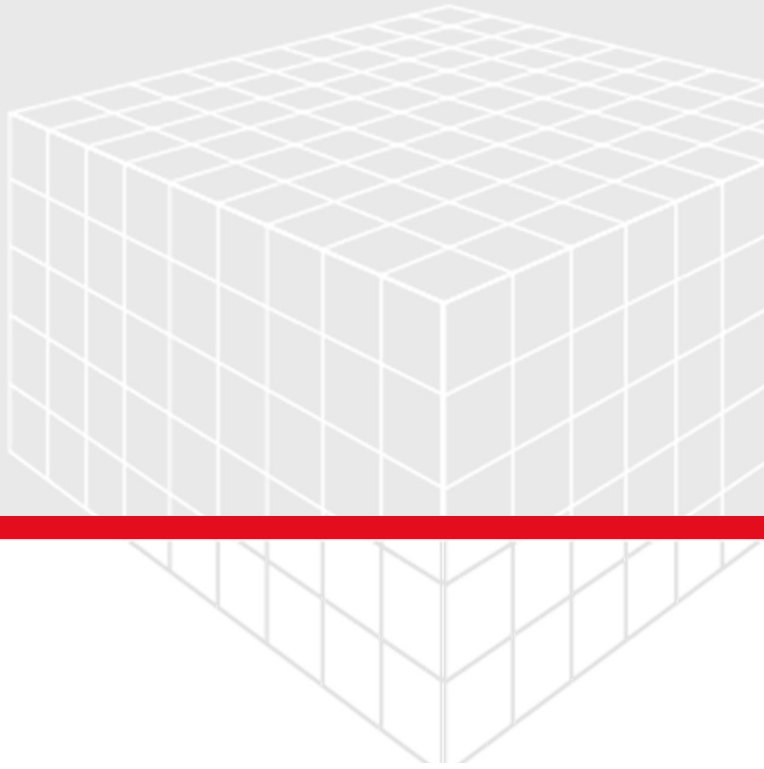
DiffuDict

User Guide



GEODICT Release 2026

Published: June 2026

GeoDict

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

DiffuDict

1 DiffuDict

Simulation of Diffusion Processes

Use DiffuDict to simulate the net movement of molecules or atoms from a region of higher concentration to a region of lower concentration. Diffusion is driven by a gradient in the concentration of the diffusing species. Diffusion depends on particle random walk, and results in mixing or mass transport without requiring directed bulk motion.

Learn to use DiffuDict

Join the conversation

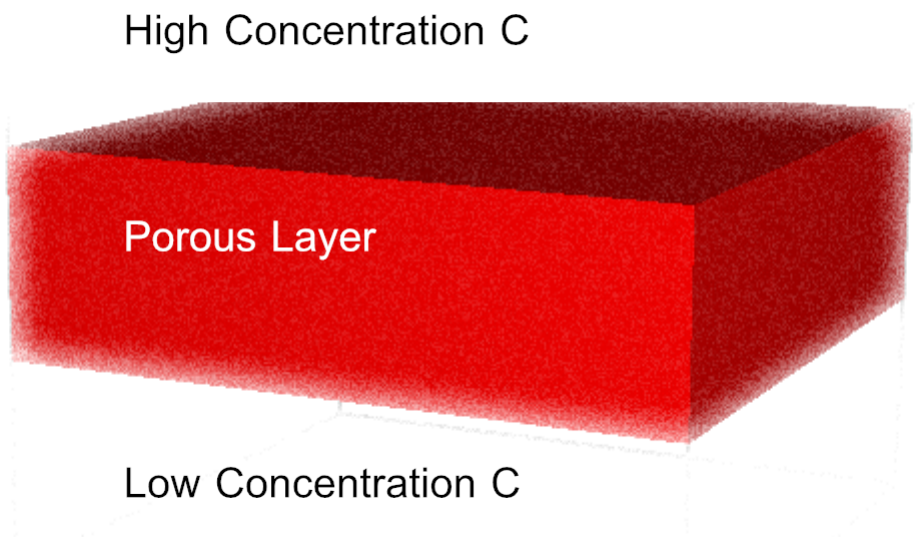
Visit the Community on GeoDict Forum to be inspired and get answers to top questions.
Find illustrative application examples on the website.
Our tutorials and seminar videos give you a first hands-on experience. They are collected in the Learning Center.
Send your Support Request to our technical Support.

1.1 Theoretical Background

Diffusivity or **diffusion coefficient** is a proportionality constant between the molar flux due to molecular diffusion and the gradient in the concentration of the species (or the driving force for diffusion) (see also http://en.wikipedia.org/wiki/Mass_diffusivity). It is described by Fick's first law:

$$j = -d\frac{\Delta C}{L} \tag{1}$$

where j is the diffusive flux, given in mol/m²/s and $\Delta C/L$ is the concentration difference over a distance L. The diffusivity d has the unit m²/s.



Fick's first law also describes diffusion through a porous layer as shown above. In this case, the diffusivity depends on the properties of the diffusing species and on the properties of the porous material. A porous material may offer a different resistance to a diffusing particle depending on the direction that the particle is traveling. Therefore, a more general form of Fick's first law

$$\vec{j} = -D\nabla C \tag{2}$$

can be used to describe the diffusion in three space dimensions. In this form, $\vec{j}$ is a three-dimensional vector, and the diffusivity D is a 3x3 matrix which describes the full directional dependence of the diffusion. In DiffuDict, always the more general form of Fick's first law is considered. For isotropic materials $D = d * \mathrm{Id}$ holds.

In a three-dimensional structure model, the stationary distribution of concentrations can be computed by solving

$$div(d(x)\nabla c) = 0 \tag{3}$$

together with the appropriate boundary conditions for the local concentration c at the boundaries of the domain. Here, the local diffusivity $d(x)$ depends on the local material. When simulating, e.g., the diffusion of oxygen, in the pore space $d(x)$ is the diffusion coefficient of oxygen, in porous materials $d(x)$ is the effective diffusivity of oxygen inside of the porous material, and in solid materials $d(x) = 0$. This equation is solved in DiffuDict's Simulate Diffusion Experiment[13] function.

This leads to the question how the effective diffusivity of a gas inside of a porous material can be determined. In literature, the effective diffusivity is often approximated as

$$d = d_0 \cdot \frac{\eta}{\tau} \tag{4}$$

where d_0 is the intrinsic diffusivity of the gas, η is the porosity of the porous medium, and τ is a tortuosity of the pores. Alternatively, the equation

$$d = d_0 \cdot \frac{\eta}{\tau} \cdot \gamma \tag{5}$$

can be found, where the additional factor γ denotes a so-called constrictivity of the pores. Various geometrical approaches are used to determine τ and γ. However, this approach is not exact because

a. the results depend vastly on the geometrical method used and
b. this approximation is only valid in the continuum and does not include Knudsen diffusion.

Know how! For unary gases, d_0 is the self-diffusion coefficient, that can be computed as

$$d_0 = \frac{1}{3}\lambda\bar{v} \tag{6}$$

where λ is the mean free path and $\bar{v}$ is the mean thermal velocity of the gas.

For binary gases, the intrinsic diffusion coefficient d_0 can be measured.

A mathematically consistent way to determine the effective diffusivity of a porous medium is to use an upscaling approach (see, e.g., Hornung, 1997[61]). For this, a representative 3D model of the pore structure is needed.

Under the assumption that the Knudsen number is small ($\mathrm{Kn} \ll 1$) and that the classical continuum mechanics approach can be used (see Knudsen number topic page[7]), the diffusivity indeed decouples into the species-dependent part d_0 and a porous-media-dependent part D^*:

$$D = d_0 \cdot D^* \tag{7}$$

Here, d_0 is a scalar quantity with unit m²/s and D^* is a dimensionless 3x3 matrix. For isotropic materials, $D^* = d * \cdot Id$ holds with a scalar diffusivity value d^*. To determine D^*, the Laplace equation

$$\Delta c = 0 \tag{8}$$

is solved in the pore space with Neumann boundary conditions on the pore walls, and a concentration drop in one space direction. With the computed concentration flux, it is possible to find the diffusion coefficient in this space direction.

In DiffuDict's Bulk (Laplace) diffusion[35] command, the Laplace equation is solved three times, each time with a concentration gradient in a different space direction, to determine the full tensor D^*. Note, that the relative diffusivity D^* is a dimensionless quantity and a property of the porous media alone that, as such, is independent of the diffusing species and surrounding fluid.

A comparison of the equations (4)[5] and (7)[6] shows that it makes sense to define a tortuosity factor κ through

$$\kappa = \frac{\eta}{d^*} \tag{9}$$

and that this tortuosity factor is also a property of the porous media which is independent of the diffusing species and surrounding fluid. To distinguish this definition from other geometric definitions of the tortuosity, we call it tortuosity factor and denote it with κ. When defined like this, equation (4)[5] holds true exactly, but a direct geometric definition of the tortuosity factor is no longer possible. Also, the constrictivity of the pores is already contained inside the so-defined tortuosity factor κ.

The continuum mechanics approach described above fails, when the number of molecules present in a grid cell becomes too low to define a meaningful concentration value c and therefore the mass transport from one grid cell to another is no longer a diffusive process. The Knudsen number[7], which compares the mean free path of a molecule with the representative pore size of the medium, is a good indicator when this happens.

At ambient conditions, the mean free path of gas molecules typically lies in the range of 50 nm to 200 nm. Thus, Knudsen effects only become significant for pore sizes in the sub-micrometer range, or in case of very dilute gases. For diffusion in liquids, the Knudsen number is always small, and Knudsen diffusion is of no importance.

If the assumption ($\mathrm{Kn} \ll 1$) no longer holds true, the diffusing molecules experience another resistance caused by additional collisions between particles and pore walls. Due to Pollard and Present, 1948[61], the overall diffusivity can be approximated from the bulk diffusivity and the Knudsen diffusivity with the so-called Bosanquet approximation:

$$D = \left(D_{\mathrm{Kn}=0}^{-1} + D_{\mathrm{Kn}=\infty}^{-1}\right)^{-1} \tag{10}$$

The bulk diffusivity is the effective diffusivity at Kn=0, and can be determined as described above from the result of DiffuDict's Bulk (Laplace) diffusion[35] command.

The Knudsen diffusivity is the effective diffusivity at $\mathrm{Kn} = \infty$, and can be determined from the result of DiffuDict's Knudsen Diffusion[45] command.

DiffuDict's Bosanquet Approximation[55] command computes D from the results of the two other commands.

Therefore, to apply Bosanquet's approximation, an algorithm is needed to compute the diffusivity at $\mathrm{Kn} = \infty$, i.e., in a situation where no particle-particle collisions happen, and the diffusion is solely caused by particle-wall interactions. This algorithm is provided in DiffuDict's Knudsen Diffusion[45] command.

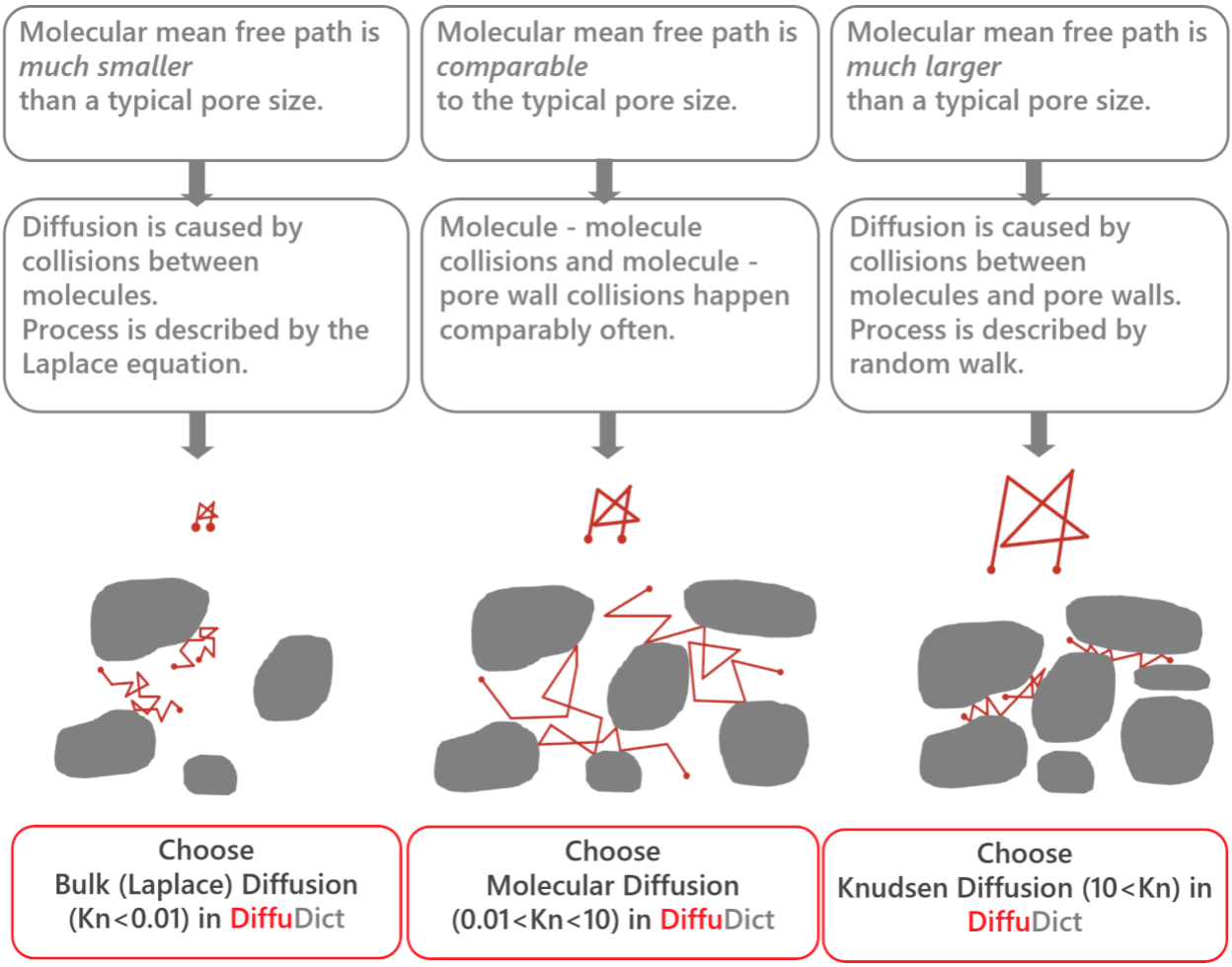
Alternatively, the diffusivity at intermediate Knudsen numbers can be determined by a random walk method that takes particle-wall collisions as well as particle-particle collisions into account. Such an algorithm is provided in the Molecular Diffusion[41] command. Theoretically, this algorithm provides a correct result for all Knudsen numbers, but for small Knudsen numbers the random walk algorithm becomes slow and inaccurate, and it is therefore advisable to determine the diffusivity by solving the Laplace equation in this case.

1.1.1 Knudsen Number

The **Knudsen number** (http://en.wikipedia.org/wiki/Knudsen_number) is a dimensionless parameter, defined as ratio of the molecular mean free path length to a representative physical length scale.

$$\mathrm{Kn} = \frac{\text{molecular mean free path length}}{\text{representative physical length scale}} \tag{11}$$

The physical length scale in case of microstructures typically is the average pore size. Depending on the Knudsen number, three cases are distinguished.



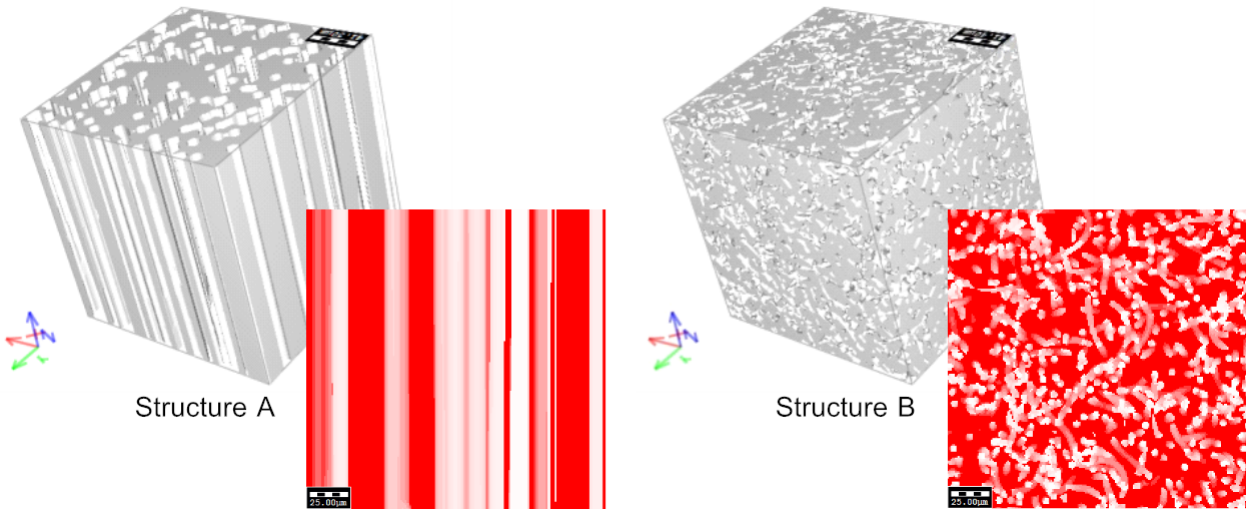
1.1.2 Tortuosity Factor and Geometric Tortuosity

The tortuosity of a porous material quantifies how straight or tortuous paths taken through the porous material are in average. A value of 1 means that all paths are straight, and higher values denote that paths are more tortuous. There have been several attempts to quantify this property, and each definition delivers (slightly) different results.

The **tortuosity factor** describes how strongly the diffusivity is reduced due to the pore shape. The tortuosity factor in a particular direction (X, Y, or Z) is defined through equation (9) [6]. In this expression, the terms "diffusivity" and "tortuosity factor" should be understood as "diffusivity in X (Y or Z) direction" and "tortuosity factor in X (Y or Z) direction". Diffusivities in X, Y, and Z directions are the diagonal entries of the computed diffusivity tensor D^*.

Following the notation of Epstein [61], the **geometric tortuosity** τ is the square root of the tortuosity factor κ.

Consider two structures with the same porosity of 50.01%. In one case, the pores are straight cylinders, and in the other, the pores are strongly curved, and have no preferred orientation. The structures are shown in 3D view and in a 2D cut, parallel to the YZ-plane. Observe the effect of the shape and orientation of the pore space on the calculated diffusivity and tortuosity factor in Z-direction.



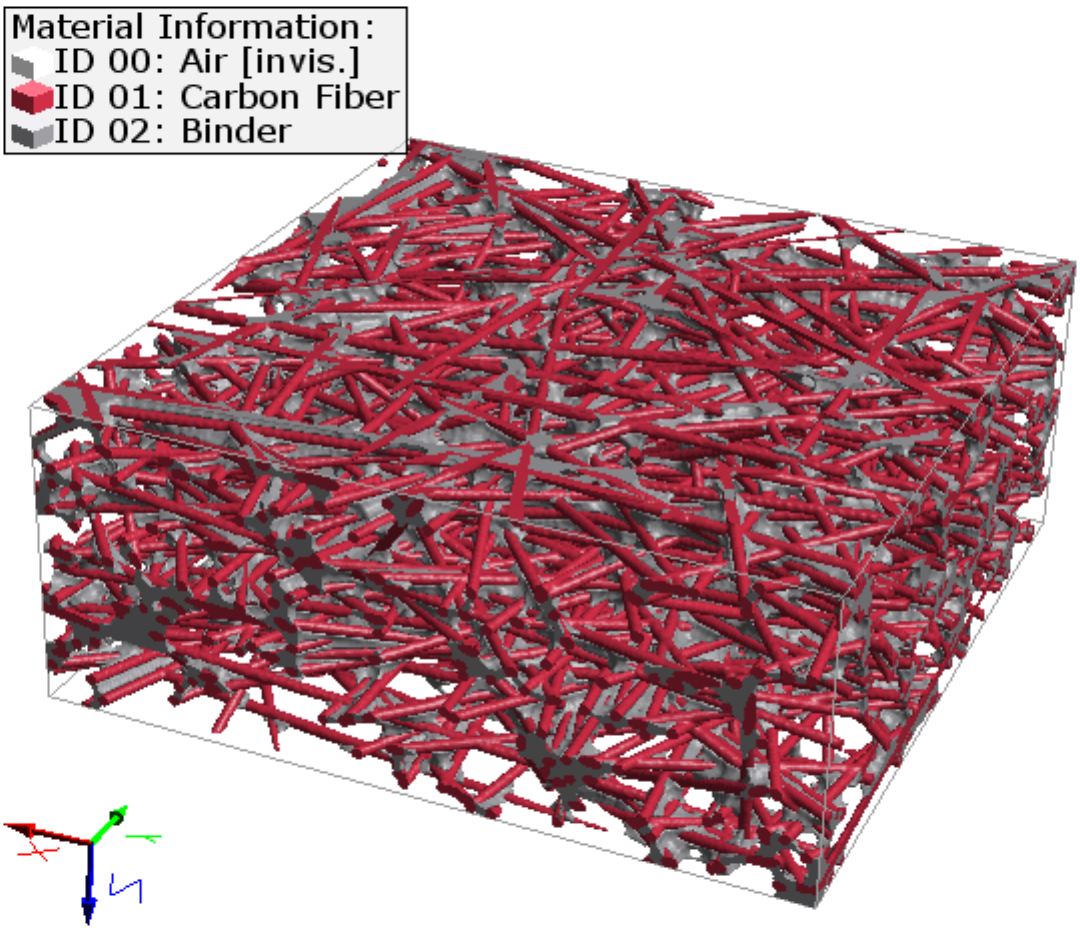
	Structure A	Structure B
Porosity (%)	50.01	50.01
Relative diffusivity (%, in Z)	49.6849	22.835
Tortuosity factor (in Z)	1.006489	2.18999

For the structure on the left (A), the relative diffusivity in Z-direction is very close to the porosity value (50.01%) and the tortuosity factor is close to the minimal possible value of 1.

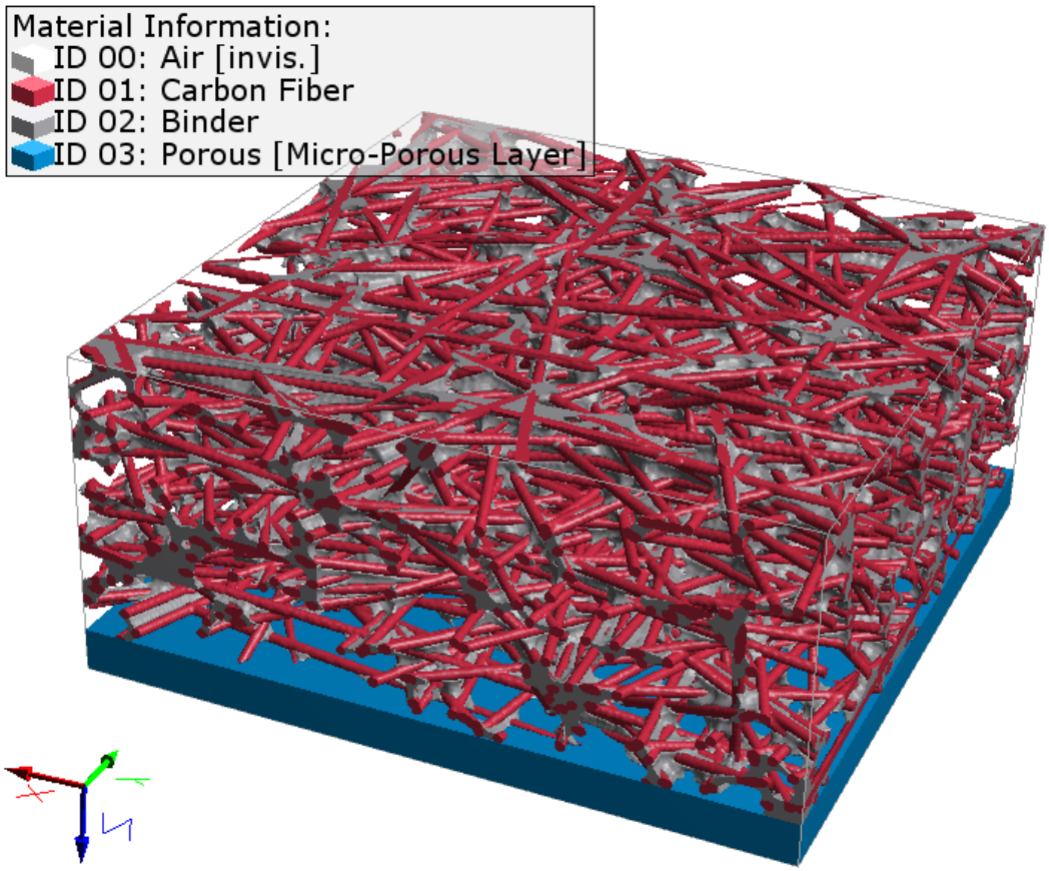
For the structure on the right (B), where the pores are curved, the relative diffusivity in Z-direction is smaller (22.835%) for the same porosity value (50.01%) and the tortuosity factor is two times higher.

1.2 Example Structures

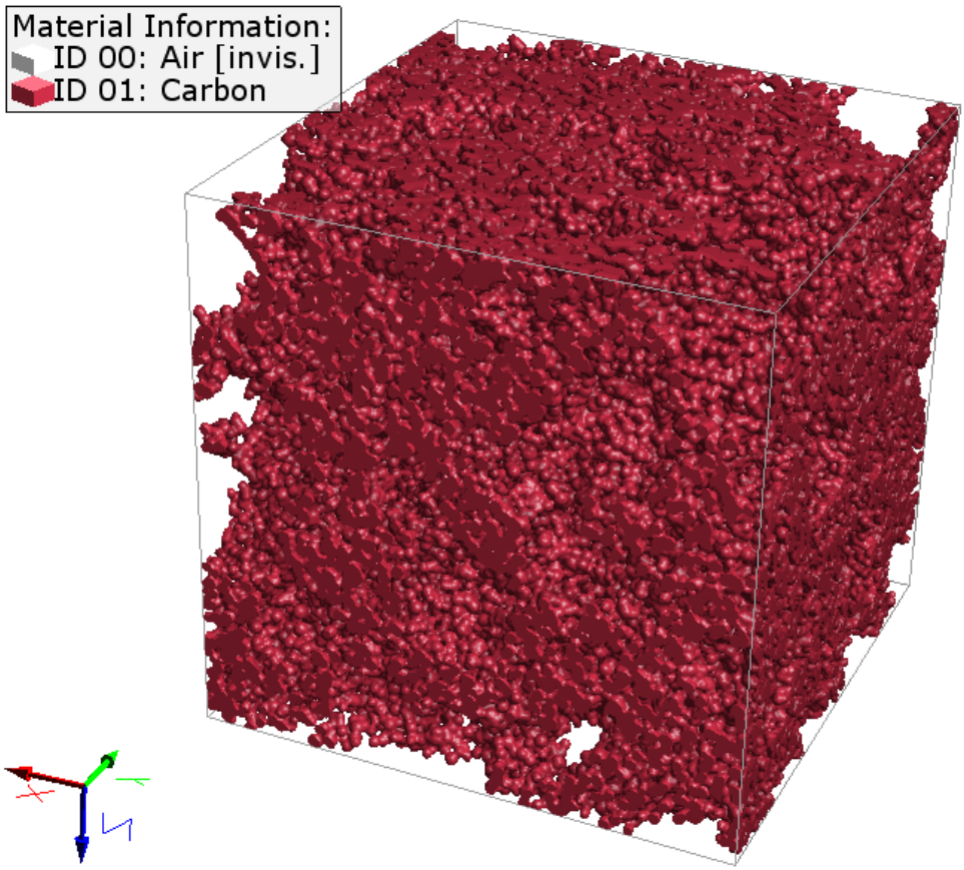
The following example structures are considered within the DiffuDict user guide. Their main differences are the relevant length scales (voxel length and pore sizes) and the presence of a porous constituent material.



Example 1: Model of a fibrous gas diffusion layer (GDL) as used in PEM fuel cells; voxel length 1 µm, model size 500 µm x 500 µm x 200 µm, porosity 79 % with mean pore diameter of 31 µm. (Download .gdt file)



Example 2: The same gas diffusion layer model as in Example 1, but with an attached, unresolved microporous layer; voxel length 1 µm, model size 500 µm x 500 µm x 230 µm. (Download .gdt file)



Example 3: Resolved model of the microporous layer of example 2; voxel length 10 nm, model size 4 µm x 4 µm x 4 µm, porosity 63 % with mean pore diameter of 0.25 µm. (Download .gdt file)

1.3 Simulate Diffusion Experiment

Use **Simulate Diffusion Experiment** to compute a stationary diffusive flux and concentration distribution if the structure contains porous constituent materials.

This command takes the local diffusivity of each Material ID as input. With this input, the Laplace equation is solved to compute the stationary diffusive flux and concentration distribution. Fick's first law is then utilized to derive the homogenized effective diffusivity. To obtain the complete three-dimensional diffusivity tensor, the Laplace equation has to be solved three times using a concentration step in each space direction as boundary condition.

The Simulate Diffusion Experiment command

- Theoretical Background 13
  Equations solved with this command.
- Options 14
  How to set the input parameters of the Simulate Diffusion command.
- Results 29
  How to understand the computed results.
- Data Visualization 32
  How to visualize the computed results.

1.3.1 Theoretical Background

The **Simulate Diffusion Experiment** command allows to compute a stationary diffusive flux and concentration distribution if the structure contains porous constituent materials. As explained in the DiffuDict theoretical background 4, this is done by solving

$$\operatorname{div}(d(x)\nabla c) = 0 \tag{12}$$

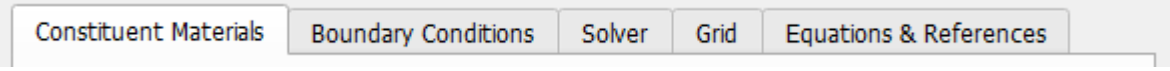
together with the appropriate boundary conditions for the concentration c at the boundaries of the domain. Here, the local diffusivity $d(x)$ depends on the local material. When simulating, e.g., the diffusion of oxygen, in the pore space $d(x)$ is the diffusion coefficient of oxygen, in porous materials $d(x)$ is the effective diffusivity of oxygen inside of the porous material, and in solid materials $d(x) = 0$.

The computed diffusive flux is used to determine the homogenized diffusivity of the 3D structure. For this, Fick's first law is used and the effective diffusivity d is determined from the computed overall diffusive flux j:

$$d = -j\frac{L}{\Delta C} \tag{13}$$

1.3.2 Options

When **Simulate Diffusion Experiment** is chosen from the pull-down menu in the DiffuDict section, clicking the **Options' Edit...** button opens the **Simulate Diffusion Experiment** dialog. The settings are organized under the **Constituent Materials**, **Boundary Conditions**, **Solver**, and **Grid** tabs. The **Equations & References** tab shows the equations (2) 4 and (3) 4 of this DiffuDict chapter for illustration and explains the used variables.



Tabs of the Simulate Diffusion Experiment dialog

- Constituent Materials 14
  Enter the material parameters of the solid material and the diffusing gas species.
- Boundary Conditions 16
  Define the boundary conditions used by the corresponding solvers.
- Solver 19
  Set the simulation stopping criterion and the desired accuracy and optimize the computational resources.
- Grid 28
  If the EJ solver is chosen, choose to discretize the structure by Voxels or Boxels.

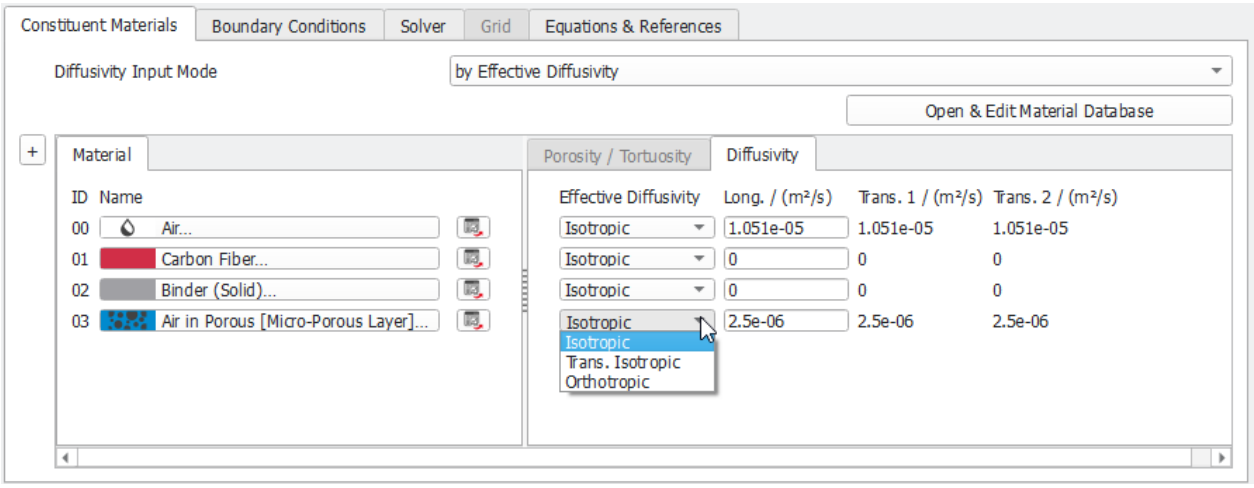
Constituent Materials

Under the **Constituent Materials** tab, the diffusivity of the diffusing species in the constituent materials must be defined. The **Diffusivity Input Mode** determines how the diffusivity is found. It can either be entered directly (**by Effective Diffusivity**) or it is computed **by Porosity / Tortuosity**.

If porous constituent materials are present:

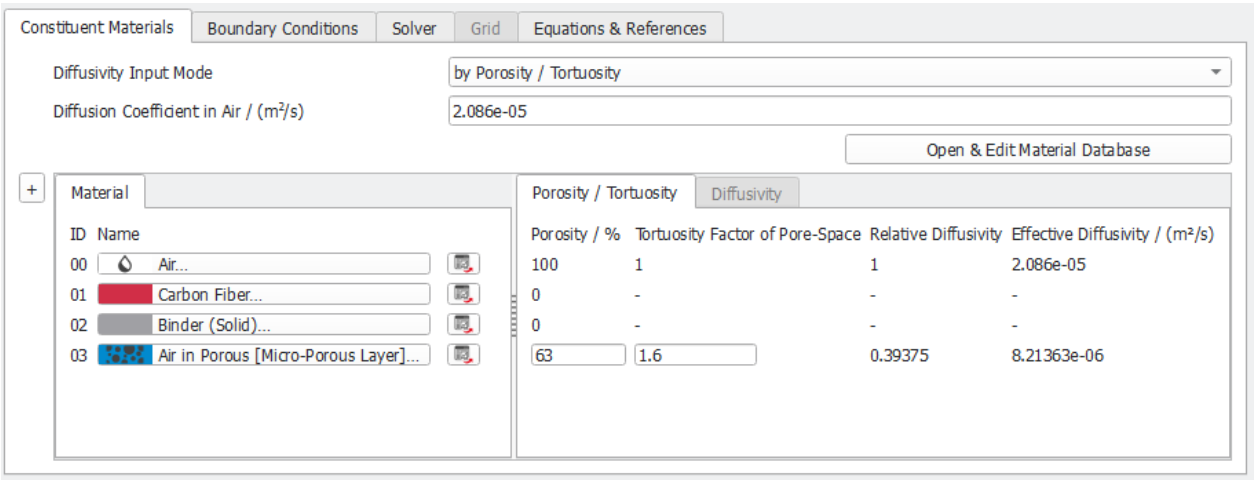
- **by Porosity / Tortuosity** might be chosen if the diffusion inside of the porous constituent material is described by a small Knudsen number.
- **by Effective Diffusivity** must be chosen if the diffusion inside of the porous constituent material is described by a large or intermediate Knudsen number, i.e., the diffusivity inside of this material must be determined with Bosanquets approximation.
- **by Effective Diffusivity** must be chosen, if the diffusion does not happen solely in the fluid phase, but also in the solid materials.

In the example presented here, the Diffusivity Input Mode is set to **by Effective Diffusivity** and the diffusivity of the porous material is set to 2.5e-6 m^2/s, which is the diffusivity computed for the MPL model (Example 3 12) with the help of Bosanquet's approximation 57. The corresponding diffusion coefficient in air, 1.051e-5 m^2/s is computed through equation (6) 5 using a mean free path $\lambda = 68$ nm and a mean thermal velocity $\bar{v} = 464$ m/s.



Inside a porous material, the diffusivity can be independent of the space direction (**Isotropic**) or the internal pore structure might prefer a certain direction of the diffusion. In this case, selecting **Transverse Isotropic** or **Orthotropic** allows to enter different diffusivities for the different directions.

In cases where the diffusion inside of the porous constituent (Material ID 03 in this example) is described by a small Knudsen number, **by Porosity / Tortuosity** can be used as input mode.



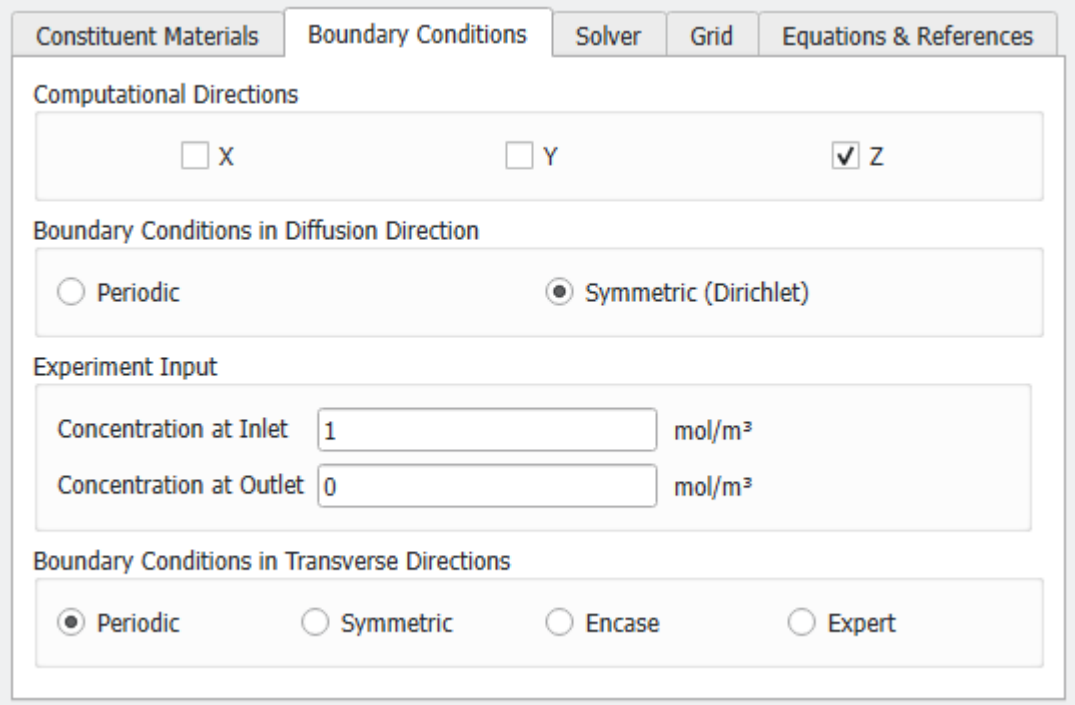
In this case, for all present fluids, the **Diffusion Coefficient in *(Fluid)*** must be entered. This corresponds to the bulk diffusivity inside this fluid constituent material, as can be seen here for Material ID 00, where the porosity is automatically set to 100% and the tortuosity is set to 1. For all porous constituents that contain the fluid, the effective diffusivity d is now computed as

$$d = d_0 \cdot \frac{\eta}{\kappa} \tag{14}$$

Here, d_0 denotes the entered diffusion coefficient, η the porosity and κ the tortuosity factor. For all solid materials, the diffusivity is automatically set to 0. The default value pre-entered in the dialog, 20.86 mm^2/s is the diffusivity of oxygen in nitrogen at 25°C and 1013 mbar (see Becker et al [61]).

Boundary Conditions

Under the **Boundary Conditions** tab, the Computational Direction(s), the Boundary Conditions in Diffusion Direction, the Experiment Input, and the Boundary Conditions in Transverse Direction are to be set.



Computational Directions

For the **Computational Directions**, choose the direction for which the diffusivity should be calculated. To obtain the whole 3x3 diffusivity matrix and a tortuosity value for each direction in the result file, select all three directions.

The directions that are not checked will not be calculated and will appear marked as unknown in the diffusivity matrix in the Result Viewer of the result file.

Boundary Conditions in Diffusion Direction

The choice of boundary conditions depends on the structure and the application.

If **Periodic** is checked, DiffuDict assumes that the structure is periodically repeated in the diffusion direction. If **Symmetric (Dirichlet)** is checked, a constant concentration is ensured along both border planes in the diffusion direction. For non-periodic structures, **Symmetric (Dirichlet)** boundary conditions give more accurate results than **Periodic** boundary conditions.

In previous versions of GeoDict, the computational memory requirements for **Symmetric (Dirichlet)** boundary conditions case were almost two times higher and therefore, the computational time was longer. This is no longer the case since GeoDict 2022.

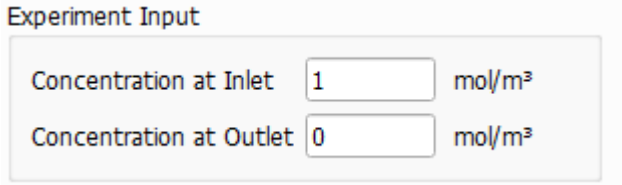
For structures with large porosity (>80%), the difference between these two cases is insignificant and can be compared with a computational error.

In few situations, however, due to the structure's geometry, using Dirichlet boundary conditions is unavoidable. An example, with pores artificially closed by using a periodic boundary condition, is shown below. When repeating the structure periodically it becomes clear, that periodic boundary conditions cannot be used in this case because the periodic repetition artificially closes the pores, reducing the true diffusivity. The use of Dirichlet boundary conditions eliminates this artifact.

Periodic Boundary Conditions | Dirichlet/Symmetric Boundary Conditions

Experiment Input

In the **Experiment Input** panel, the difference in concentration of the diffusing fluid across the material (**Concentration at Inlet**, **Concentration at Outlet**) model for a specific experiment should be entered.



Boundary Conditions in Transverse Directions

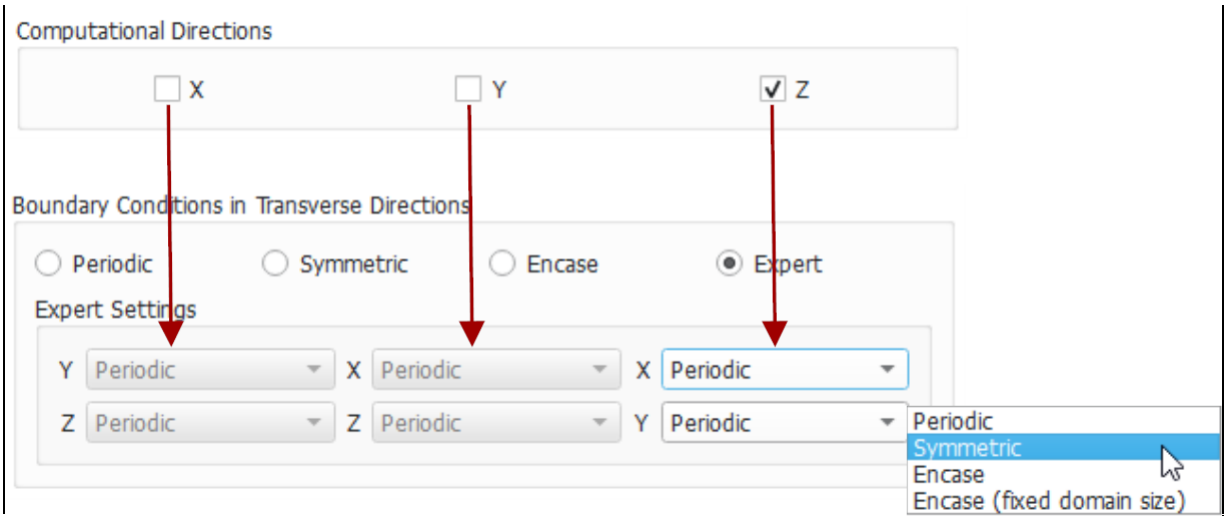
Besides the boundary conditions chosen for the main diffusion direction, the boundary conditions in the transverse (or tangential) directions can be set to be **Periodic**, **Symmetric**, **Encase** or **Expert**.

With **Periodic** boundary conditions, the structure is repeated in the transverse directions.

With **Symmetric** boundary conditions, the structure is mirrored at the domain boundary. The LIR and EJ solver solve the symmetric boundary condition by using Neumann (or zero flux) condition at the domain boundary.

With **Encase,** the domain boundary is treated as closed by solid material.

With **Expert**, it is possible to define different boundary conditions for each side in each computation:



When **Encase** is used, the solver internally adds a one-voxel layer in the required direction and solves with periodic boundary conditions. That effectively is equivalent to solve the structure with casing in two ends in the direction of interest. So, the size of computation in this direction becomes n+1. However, if changing the computational size is not preferable, using **Encase (fixed domain size)** can avoid that, then the first layer of the structure is replaced by a non-diffusing material.

Solver

The Laplace equation can be solved by two different solvers: EJ and LIR. When transverse isotropic or orthotropic effective diffusivities are selected in the Constituent Materials tab 14, the **LIR** must be used.

EJ Solver

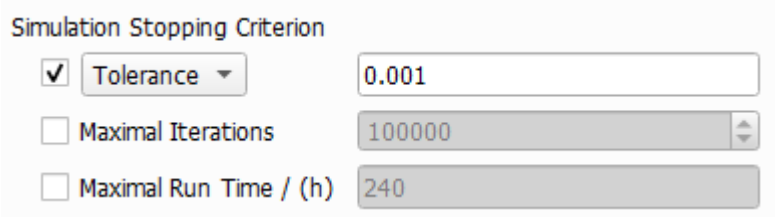
Simulation Stopping Criterion

Iterative Solver

The basic idea of an iterative method is to

1. Start with some initial guess for the unknown values
2. Improve the current values in each iterative step
3. Repeat the iterative process until one of the stopping criteria is reached.

Select one or multiple of the stopping criteria **Tolerance**, **Residual**, **Maximal Iterations** or **Maximal Run Time**.



The default stopping criterion of the EJ solver, **Tolerance**, detects if the iterative process becomes stationary. This occurs when the change in the **Diffusivity** value from iteration to iteration becomes sufficiently small. If the relative change is smaller than the value entered for **Tolerance**, the iteration is stopped:

$$\frac{\text{new diffusivity} - \text{prev. computed diffusivity}}{\text{prev. computed diffusivity}} < \text{tolerance} \tag{15}$$

Alternatively, the **Residual** stopping criterion can be used. In this case, the iteration is stopped, if the solution satisfies the equation up to the required accuracy.

When the solver stops because the **Maximal Iterations** value or **Maximal Run Time** has been reached, no guarantee on the quality of solution can be given. In this case:

- Check the corresponding .log file to see how large the residual values and conductivity increase are. If these values are already very close to the desired result, you may decide to use the current result.
- Double check the structure and parameter values. Unphysical parameters or too rough resolution of the structure (leading, e.g., to artificial unconnected components) can cause the iterative solver to fail.

Which stopping criterion has occurred, can be seen in the result file under the **Results Report** tab.

Restart Save Interval

The calculations run by the solvers can be restarted from saved intermediate result files and the interval between auto-saves can be configured from the value entered in **Restart Save Interval (h)**.

Parallelization

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

Restart from *.gdr File

In some situations, it may be useful to re-use previously computed results and save a great amount of time by restarting the process from an existing GeoDict result file (*.gdr). Typical examples would be

- non-sufficient accuracy of some computation, when it is suspected that more iterations may improve the quality of the result,
- trying another boundary condition (e.g., Dirichlet instead of Periodic or other way around),
- continuing an iteration process,
- or using similar parameter values for the computation.

To restart and initialize the solver from a result file (*.gdr), check **Restart from .gdr File** and enter the name of a file, or **Browse** for it in the project folder.

When using **Restart From .gdr File** only the volume field (*.hht file) associated to the loaded *.gdr file will be used to continue the simulation.

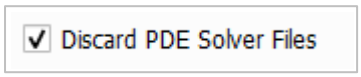
Important! The *.hht file is saved in the result folder of a previous run. Make sure that this file was not deleted from the results folder. If the *.hht file is not available the restart option cannot be used.

No other parameters from the options dialog will be loaded from the *.gdr file, but it is possible to load them by clicking **Load Parameters** in the Result Viewer. This way the simulation can be continued while using the same settings as in the initial simulation.

Important! The structure from the run of which you want to use the result file must be loaded into GeoDict. If this is not the case, an error message is displayed.

Discard PDE Solver Files

Checking **Discard PDE Solver Files** causes the deletion of all intermediate computation files, such as log files and flow field files.



Only the content of the final GeoDict result file (*.gdr) is stored.

While having the benefit of saving hard disk storage place, discarding intermediate solver files has also the effect of disabling the 3D visualization of the results.

Write Diffusion Flux into Solution File

Additional 3D data can be added to the solution *.hht files for visualization or later analysis. With **Write Diffusion Flux into Solution File** the Diffusion Flux in the three coordinate directions is saved, allowing a detailed analysis of the field. The size of the result files and the memory requirements increase when selecting this option.

Advanced Options: Analyze Geometry

Checking **Analyze Geometry** performs an analysis of the geometry before the solver computations.

If no percolation path through the structure is found, the partial differential equation does not need to be solved, and a diffusivity of 0.0 is directly reported as the solution.

Furthermore, the solver operates on the whole structure, regardless of which parts are connected or unconnected. However, unconnected components are not transport relevant, thus the effort of solving in these parts is not necessary.

For these reasons, a geometrical analysis is routinely run to determine whether a through path exists. After the analysis is finished, unconnected components are removed from the computational grid.

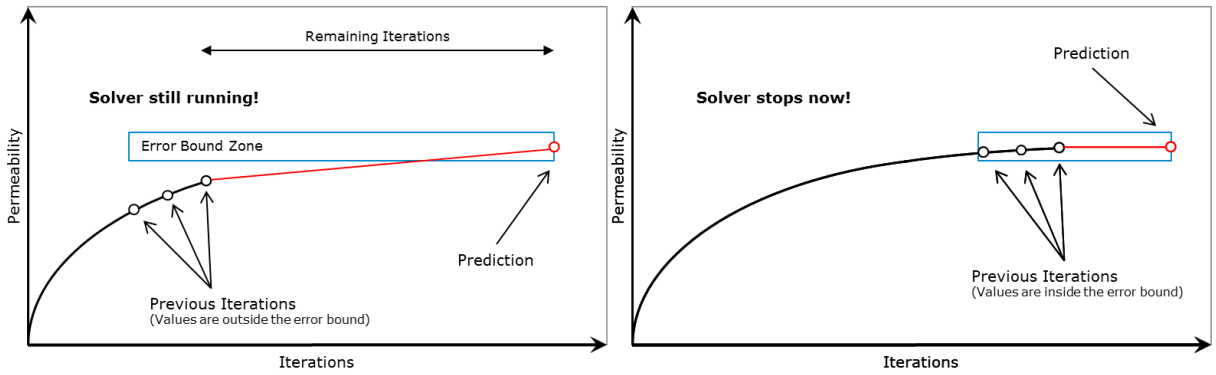
This may speed up the computations but requires some time for the geometrical analysis, especially for very large structures. So if you know beforehand that there are no unrelevant parts or that the structure has been processed to eliminate them, the geometry analysis can be switched off by unchecking **Analyze Geometry**.

LIR Solver

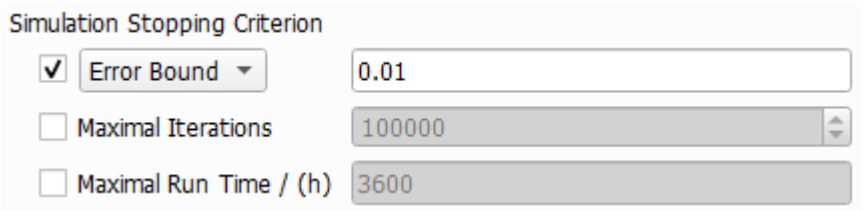
Simulation Stopping Criterion

The default stopping criterion of the LIR solver, **Error Bound,** uses the result of previous iterations and predicts the final solution based on linear and quadratic extrapolation. The solver stops if the relative difference regarding the prediction is smaller than the specified error bound.

The stopping criterion recognizes oscillations in the convergence behavior and prevents premature stopping at local minima or maxima. A damped convergence curve is fit through the oscillating curve and the solver stops then regarding the damped convergence curve.



The stopping criteria **Tolerance**, **Maximal Iterations** and **Maximal Run Time** work as described for the EJ solver 20.



Restart Save Interval, Restart from .gdr file, Discard PDE Solver Files and Write Diffusion Flux into Solution File

The options are the same as for the EJ solver and a description can be found above 21.

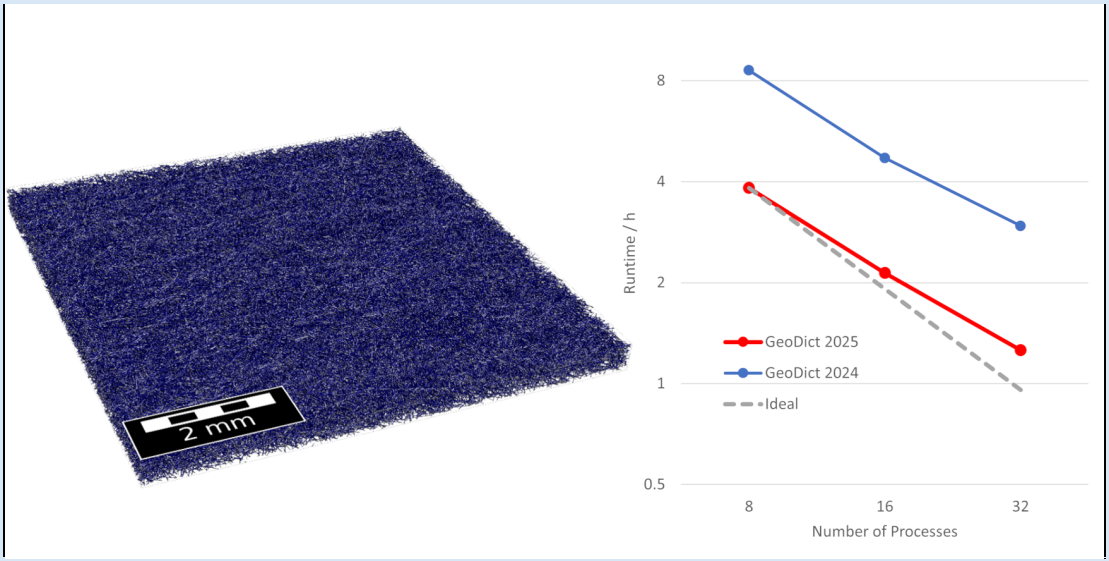
Parallelization

Depending on the purchased license, the simulation process can be parallelized. The Parallelization Options dialog box opens when clicking the **Edit...** button, to choose between **Sequential**, **Parallel (Shared Memory)** and **Automatic Number of Threads**. For

details on how to set up und run parallel computations, consult the High Performance Computations chapter.

Parallelization Benchmark Results

As an example for the parallelization, the diffusivity in through-direction in a Gas Diffusion Layer (GDL) is computed with different number of processes. The computation is run on a server with 2 x Intel E5-2697A v4 processors with 16 cores each, running with a maximum of 3.60 GHz, and 1 TB RAM. The GPL structure has size 8,192 x 8,192 x 512 voxels and is created in GeoDict with the GeoApp Gas Diffusion Layer, available in FiberGeo. The computation for the structure with over 34 billion voxels, requires 302 GB of RAM.

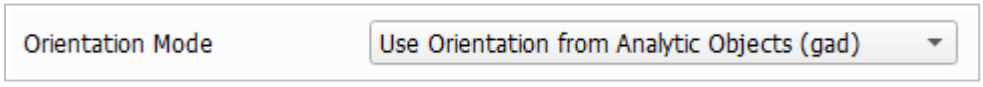


The runtimes for a different number of parallel processes show that even with 8 processes, the simulation for the huge structure is possible in 3:50 h. Due to the very good speed-up for increasing number of processes, the computation with 32 processes is possible in 1:15 h. Runtimes in GeoDict 2025 are improved compared to GeoDict 2024 due to improvements in the cell generation code of the LIR solver.

Orientation Mode

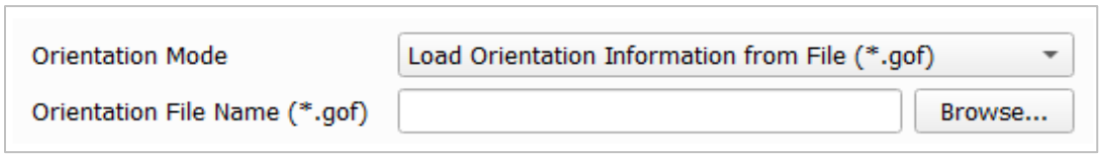
If one of the constituent materials has a non-isotropic material law, a local orientation is needed to compute the diffusivity. There are three different choices how to determine the local orientation.

The standard case is to use the orientation defined by the local orientation of the GAD objects, e.g., the direction of a fiber (**Use Orientation from Analytic Objects (gad)**).

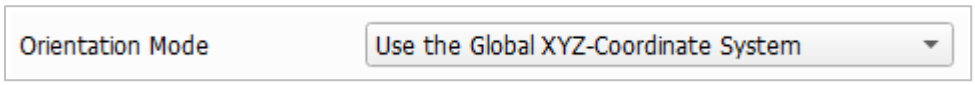


However, if the current structure was not generated using one of the structure generation

modules, but imported from a 3D scan, GAD object information is not available. In such a case, the local orientation must be estimated from the image first, e.g., by using FiberFind or GrainFind. It is then possible to load the local orientation from a file generated by one of those modules, by selecting **Load Orientation Information from File (*.gof)**.



The last option is to simply **Use the Global XYZ-Coordinate System**.



In this case, the diffusivities entered in the Constituent Materials tab are the diffusivities in the X-, Y-, and Z-directions.

For the LIR solver, more settings are hidden under **Advanced Options**. Expand it to make them visible.

Advanced Options: Analyze Geometry

Checking **Analyze Geometry** performs an analysis of the geometry before the solver computations.

If no percolation path through the structure is found, the partial differential equation does not need to be solved, and a diffusivity of 0.0 is directly reported as the solution.

Furthermore, the solver operates on the whole structure, regardless of which parts are connected or unconnected. However, unconnected components are not transport relevant, thus the effort of solving in these parts is not necessary.

For these reasons, a geometrical analysis is routinely run to determine whether a through path exists. After the analysis is finished, unconnected components are removed from the computational grid.

This may speed up the computations but requires some time for the geometrical analysis, especially for very large structures. So if you know beforehand that there are no unrelevant parts or that the structure has been processed to eliminate them, the geometry analysis can be switched off by unchecking **Analyze Geometry**.

Advanced Options: Write Compressed Volume Fields

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

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

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

Advanced Options: Optimization Options

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

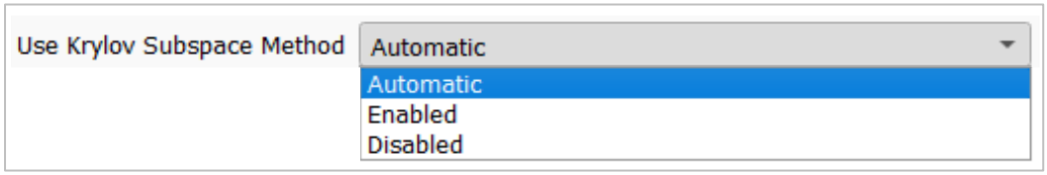
The method is available to solve the Stokes and Stokes–Brinkman equations as well as for solving mechanics, diffusion, thermal, and electrical conduction and is enabled by default.

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

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

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

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

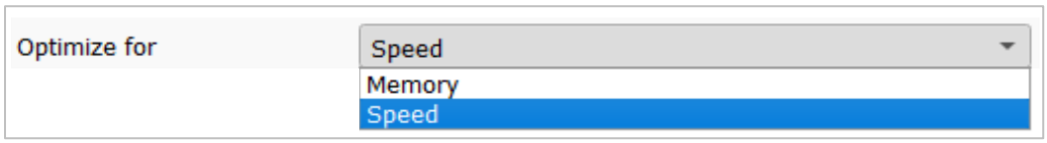


Note! In case that the Krylov subspace method (BICGStab) is used, the **Relaxation** may also be adapted automatically.

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

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

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



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

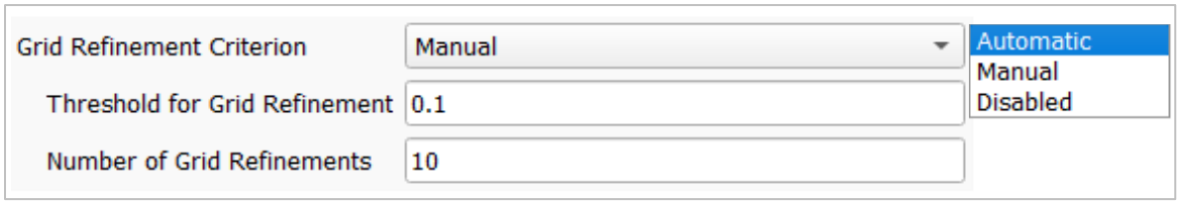
Advanced Options: Grid Options

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

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

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

The solver refines the adaptive grid based on the computed fields. New cells are introduced at locations where higher accuracy is needed. Three settings are available here:



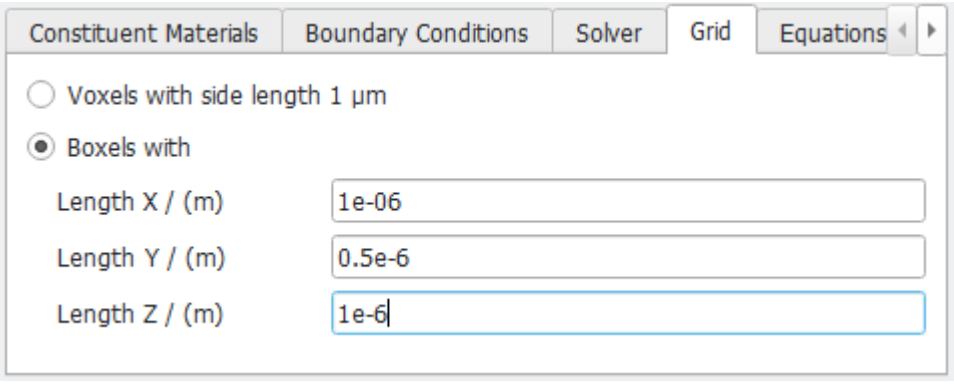
- **Automatic**: The computational cells are split when the difference in values between neighboring cells exceeds an automatically determined threshold. The solver automatically chooses all internal parameters based on the structure and simulation settings.
- **Manual**: The computational cells are split when the difference in values between neighboring cells exceeds the specified threshold. With this option you can specify the parameters **Threshold for Grid Refinement** and the **Number of Grid Refinements** manually. The **Threshold for Grid Refinement** must be between 0.0 and 1.0, while the recommended value range is between 0.05 and 0.1. The solver refines the grid at regions with high gradients and thus, the accuracy of the grid is enhanced in areas where it is necessary. Cells are split where the current gradient exceeds the **Threshold** multiplied by the maximum gradient. The **Number of Grid Refinements** is the maximum number of grid refinements that the solver can perform and should be between 1 and 10. The gradient-based grid refinement can increase the number of iterations, runtime, and memory requirements.
- **Disabled**: The grid refinement is turned off.

Know how! Grid Refinement can be neglected in most of the cases but should be used if a flow simulation is done on a structure with a very long inlet and outlet, for pleated filter structures, or for Navier-Stokes simulations.

Grid

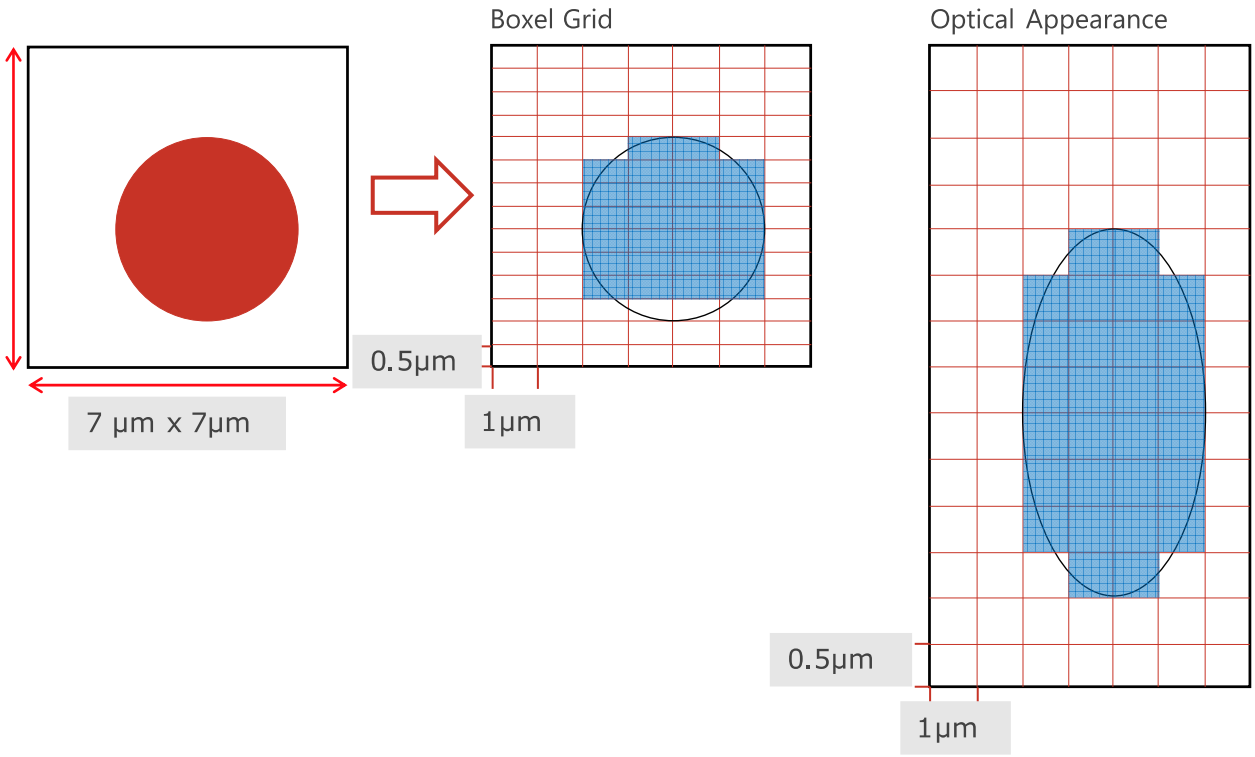
The choices under the **Grid** tab are only possible if you select **EJ** as solver. The tab is deactivated when you select the **LIR** solver.

If **Boxels** (cuboid) instead of **Voxels** (cube) are used to resolve the structure, the boxel lengths can be set in the **Grid** tab.



Sometimes, grid lengths may be different in each direction, e.g., for Focused Ion Beam (FIB) images, the grid size in Z-direction is often different from the size in X- and Y-directions.

For example, a structure is sampled in a computational grid with different lengths in X- and Y-direction (e.g., LX = 2*LY). When the structure is loaded in GeoDict, each grid cell gets assigned 1 voxel and the visual representation of the structure looks stretched. To correct this effect, boxels set as, e.g., LX = 1 µm and LY = 0.5 µm, need to be chosen in the **Grid Size** panel.



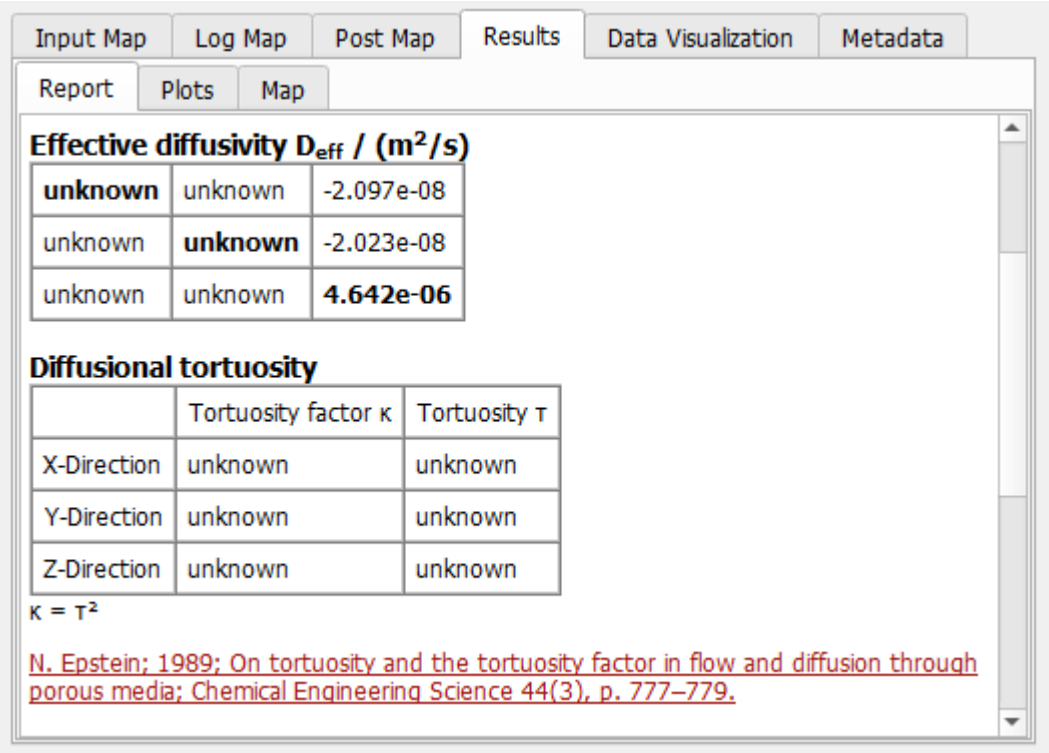
1.3.3 Results

Click **OK** to input the entered parameters, and then click **Run** in the DiffuDict section to start the command.

The results are immediately shown in the opening **Result Viewer** after the process is finished. The screenshot below shows the results obtained with the EJ solver for the GDL with an attached, unresolved MPL (Example 2 11):

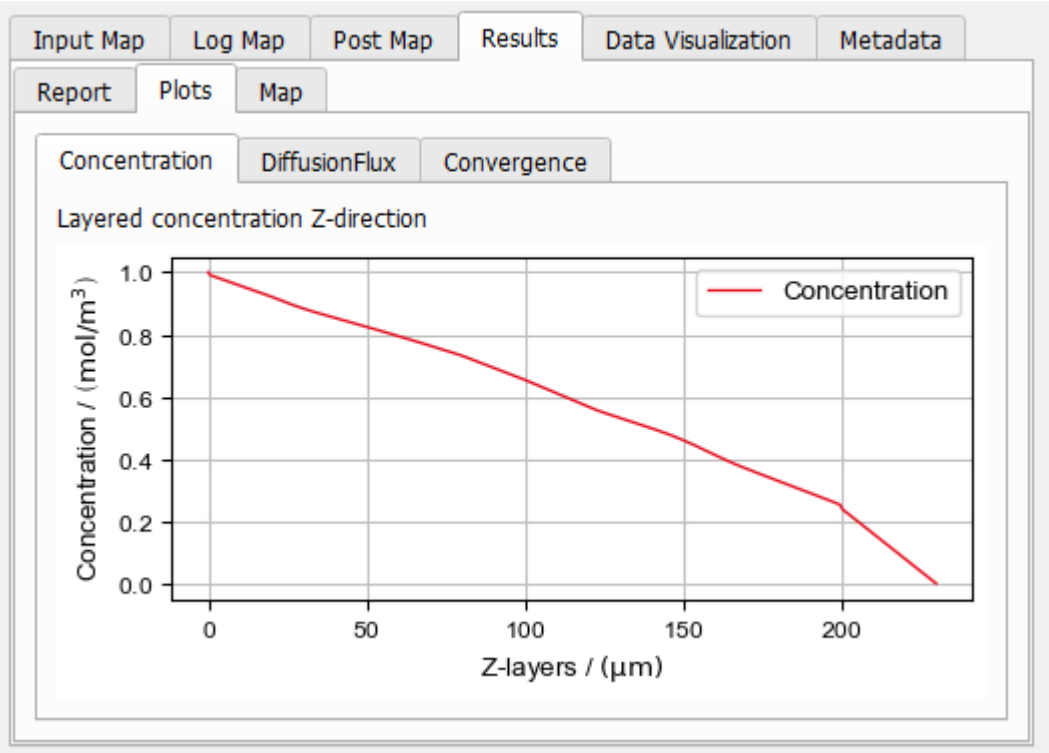
The **Results - Report** subtab shows the structure's computed **Effective diffusivity**.

Additionally, the **Relative diffusivity** and the **Diffusional tortuosity** are reported if the structure contains no porous materials. If the structure contains porous materials or - to be more precise - materials with different effective diffusivities, these values cannot be computed and only the homogenized effective diffusivity is determined.



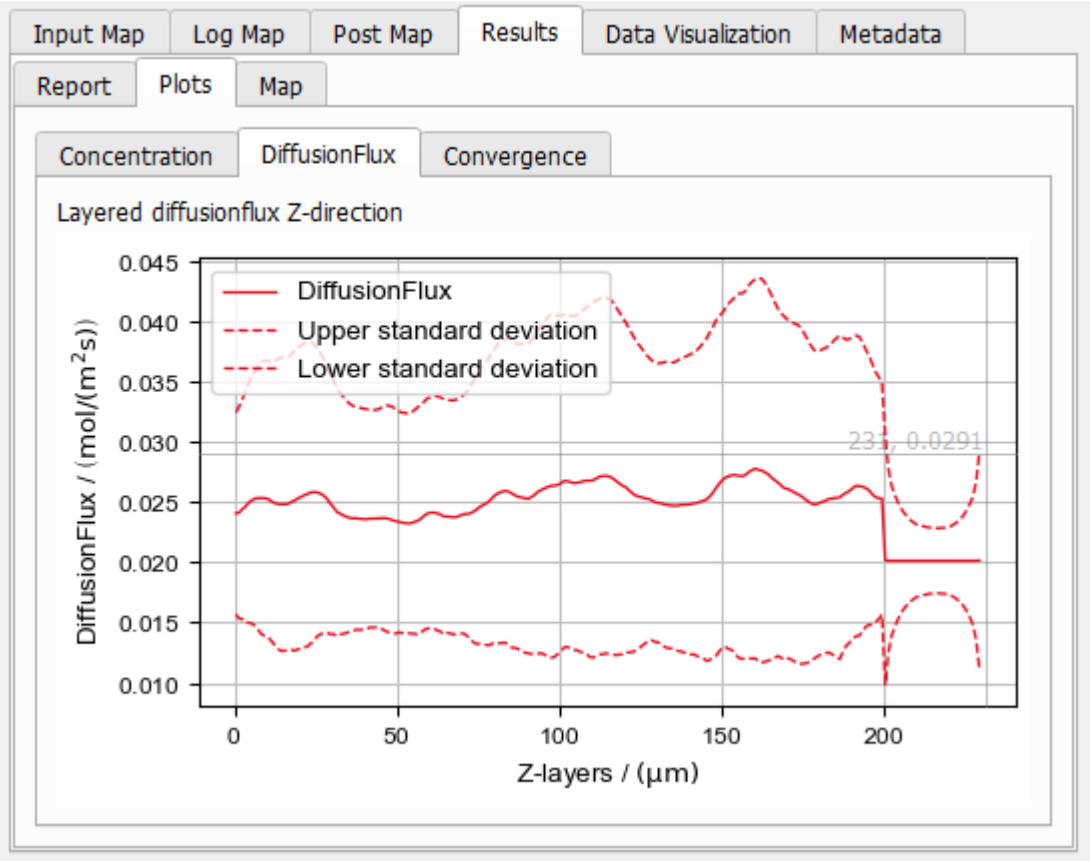
Furthermore, for each computation direction, the used solver, the number of parallel processes, the required iterations, the runtime needed, the memory usage and the applied stopping criterion are reported. At the bottom, find the overall runtime and the total memory usage, which consists of the memory usage of the PDE solver and the GeoDict executable.

Under the **Results – Plots - Concentration** subtab, the average value of the concentration in each voxel layer is plotted, the screenshot shows the drop in Z-direction. At the boundary of the domain, the set boundary conditions of c=0 for Z=200 µm and c=1 for Z=0 µm are reached.



Under the **Results – Plots - DiffusionFlux** subtab, the average value of the diffusive flux and the standard deviation in each voxel layer is plotted. The average is taken over all pore or porous voxels which allow for diffusion.

The screenshot shows, that inside of the porous MPL the solution shows a different characteristics as in the fibrous, resolved part of the 3D model.

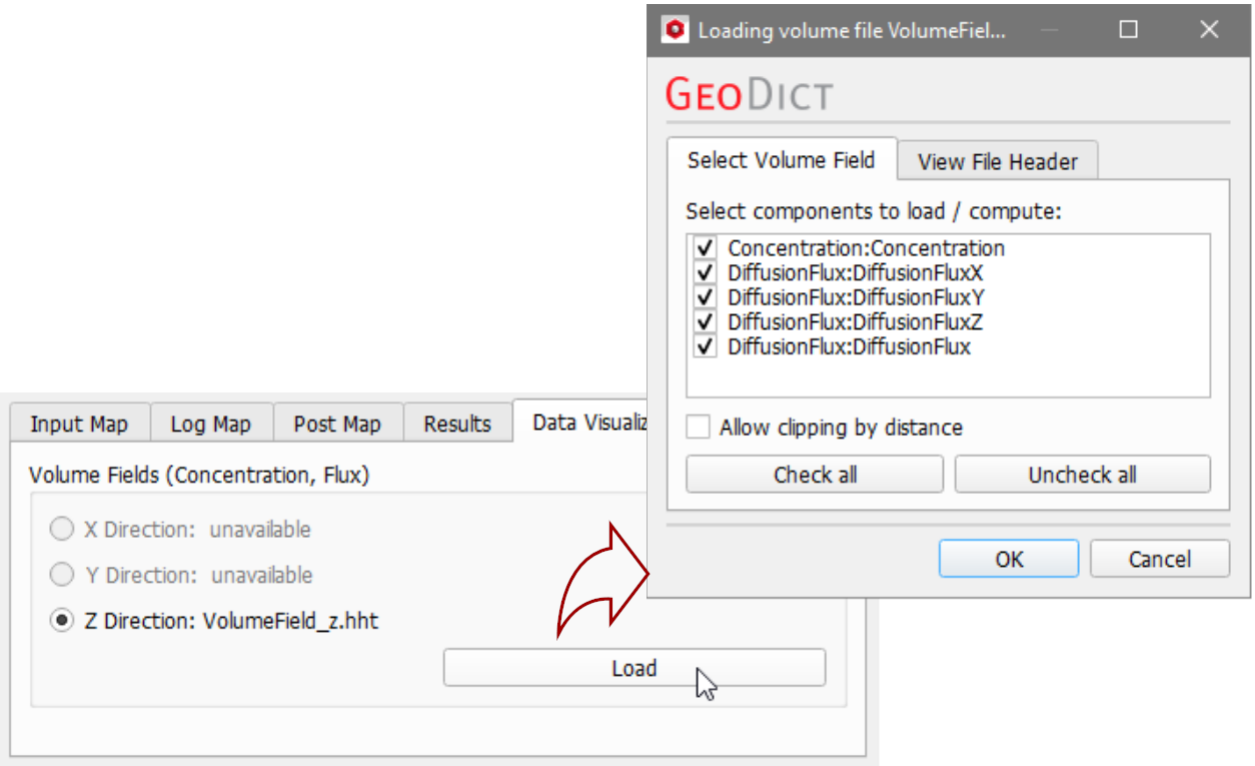


Under the **Results – Plots - Convergence** subtab, the computed diffusivity result is plotted for each iteration of the solver.

1.3.4 Data Visualization

Under the **Data Visualization** tab, the result of the diffusivity calculations can be loaded and graphically visualized. Computation Directions[16] that were not previously selected were not computed, and, thus, are unavailable for visualization.

Select a result file and click **Load**. A new dialog allows to select the result fields contained in the file that should be loaded for visualization.

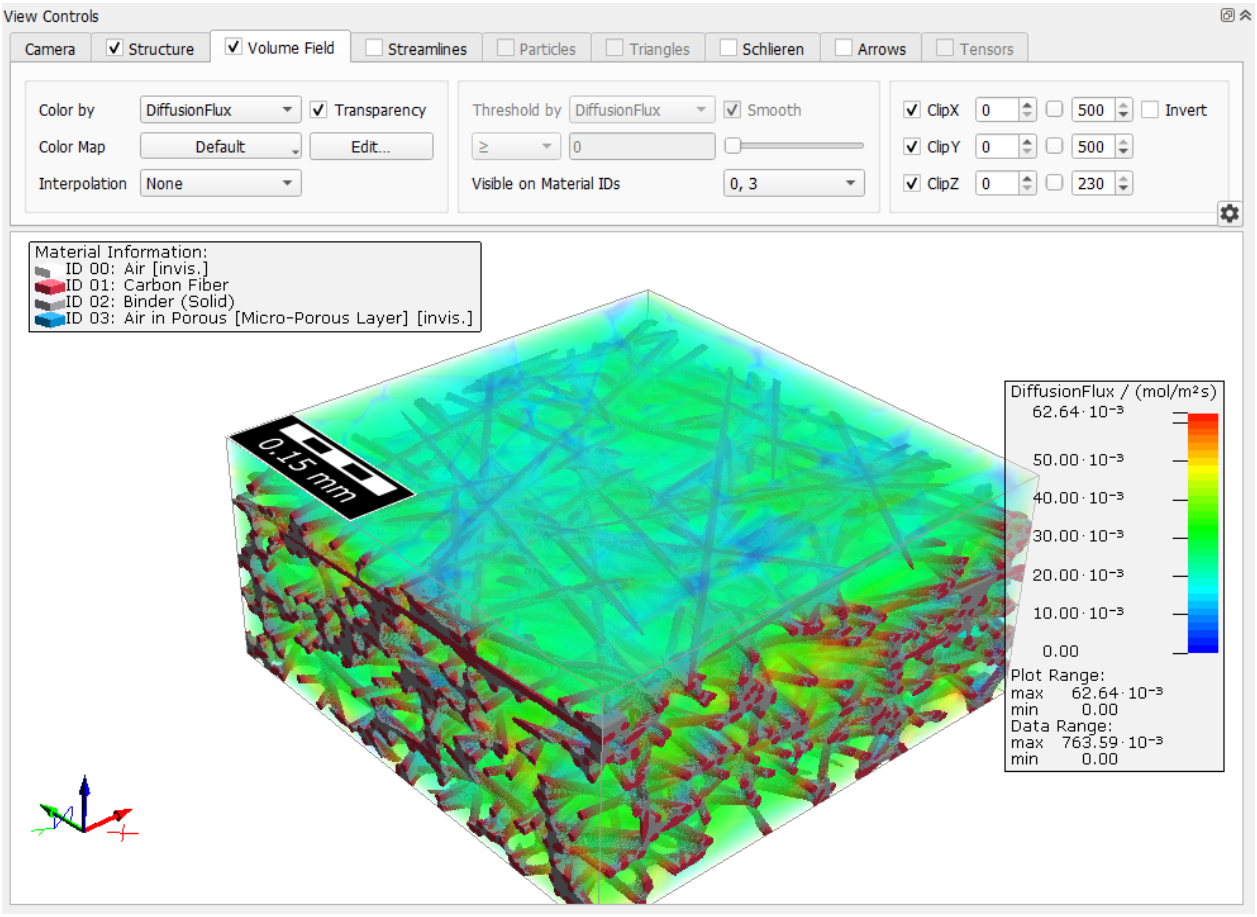


The .hht files contain the local concentration in every voxel. If **Write Diffusion Flux into Solution File** 22 was checked, the result files also contain the diffusion flux.

Different from concentrations and fluxes computed with the Bulk (Laplace) Diffusion 35 command, the concentrations and fluxes have a physical unit here: mol/m^3 for concentrations and $mol/m^2/s$ for the diffusive flux.

Besides the concentration, the diffusion flux can be visualized as result field or, as any other vector valued 3D field, as streamlines or arrow fields.

To better view the concentration or flux also in the porous materials, switch off the visualization of the current 3D structure by unchecking **View** → **Structure** in the menu bar. Alternatively, open the **Color & Visibility Settings** and set all porous materials to invisible. To visualize the result field only inside of the pores or porous materials, uncheck all solids in the **Visible on Material IDs** drop-down menu. The visualization options are explained in detail in the Visualization chapter of this User Guide.


View Controls
Camera
Structure
Volume Field
Streamlines
Particles
Triangles
Schlieren
Arrows
Tensors
Color by
DiffusionFlux
Transparency
Color Map
Default
Edit...
Interpolation
None
Threshold by
DiffusionFlux
Smooth
Visible on Material IDs
0, 3
ClipX
ClipY
ClipZ
Invert
Material Information:
ID 00: Air [invis.]
ID 01: Carbon Fiber
ID 02: Binder (Solid)
ID 03: Air in Porous [Micro-Porous Layer] [invis.]
0.15 mm
DiffusionFlux / (mol/m²s)
62.64·10⁻³
50.00·10⁻³
40.00·10⁻³
30.00·10⁻³
20.00·10⁻³
10.00·10⁻³
0.00
Plot Range:
max 62.64·10⁻³
min 0.00
Data Range:
max 763.59·10⁻³
min 0.00

1.4 Bulk (Laplace) Diffusion

Use **Bulk (Laplace) Diffusion** to determine the tortuosity and the relative diffusivity tensor of the porous media. These quantities are properties of the porous media itself and as such, independent of the diffusing species and surrounding fluid.

This command solves the dimensionless Laplace equation in the pore space to compute a stationary diffusive flux and concentration distribution. Fick's first law is then utilized to derive the diffusivity of the media. To obtain the complete three-dimensional diffusivity tensor, the Laplace equation has to be solved three times using a concentration step in each space direction as boundary condition.

The Bulk (Laplace) Diffusion command

- Theoretical Background 35
  Summary of the computational method used by this command.
- Options 36
  How to set the input parameters of the Laplace Diffusion command.
- Results 37
  How to understand the computed results.
- Data Visualization 39
  How to visualize the computed results.

1.4.1 Theoretical Background

This command solves the Laplace equation (8) 6 in the pore space with Neumann boundary conditions on the pore walls, and a concentration drop in one of the space directions. The equation is solved three times, each time with a concentration gradient in a different space direction, to determine the full relative diffusivity tensor D^*. Recall (see the DiffuDict theoretical background 4) that D^* is a dimensionless quantity and a property of the porous media alone that, as such, is independent of the diffusing species and surrounding fluid.

Two finite volume-based solvers are available to solve this equation: EJ (Wiegmann and Zemitis 61) and LIR (Linden et al 61). Both solvers use harmonic averaging to compute conductivities at voxel faces.

The EJ solver introduces explicit jump variables across material interfaces that represent discontinuities of concentration derivatives. A Schur-complement formulation for the jump variables is derived and solved by using the FFT and BiCGStab methods. The convergence speed of this method is almost independent of the diffusivity contrast which is a very big advantage compared to other approaches, but it depends on the number of material interfaces.

The LIR solver uses an adaptive grid (instead of a regular grid) to reduce the number of grid cells significantly. The adaptive grid basis is a data structure called LIR-tree that is used for spatial partitioning of 3D images. The materials are represented as differently sized rectangular cuboid. The LIR solver is very fast, but the convergence speed depends linearly on the diffusivity contrasts and on the number of material interfaces.

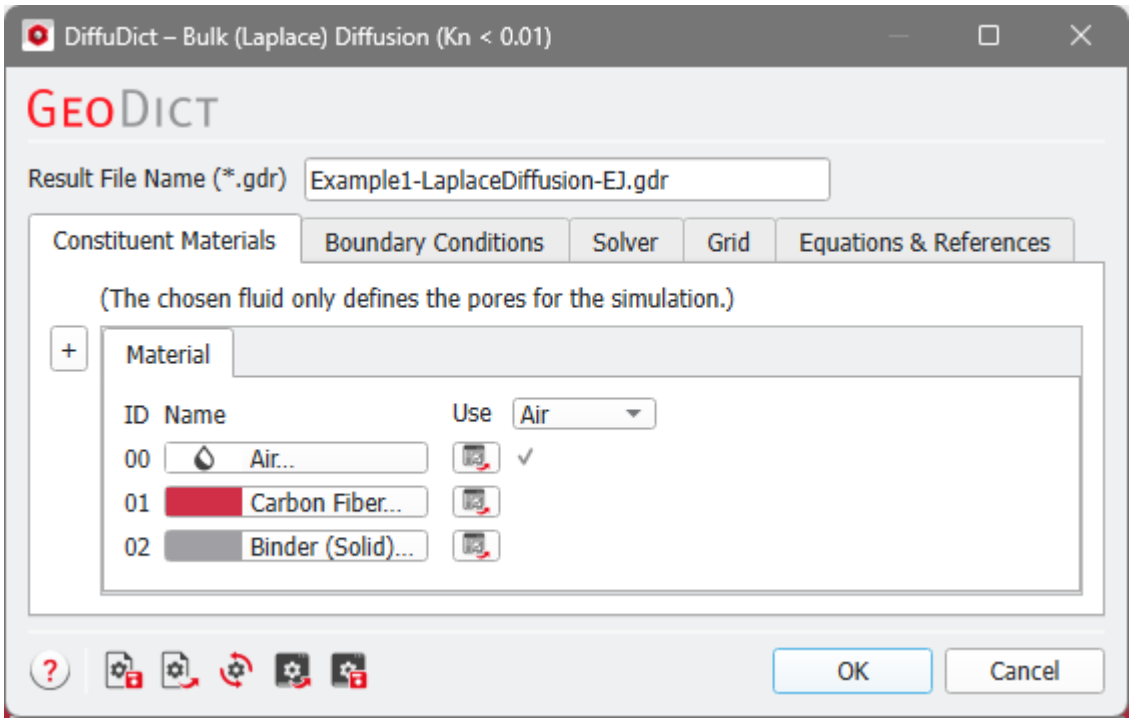
1.4.2 Options

The **Bulk (Laplace) Diffusion** options are organized under five tabs: **Constituent Materials**, **Boundary Conditions**, **Solver**, **Grid,** and **Equations & References**.

Enter a **Result File Name** for the files that will contain the results of the computations.

Constituent Materials

Under the constituent materials tab, set the materials of the Material IDs corresponding to the pore space and those that denote solids. All Material IDs marked with a green checkmark ✅ are open for diffusion. All other Material IDs are treated as solids and no diffusion takes place there.

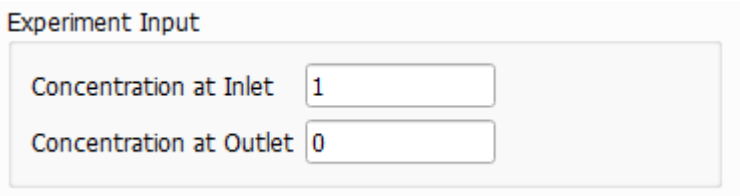


As described in the Theoretical Background 35, the specific choice of fluid and constituent materials does not influence the result of the simulation here.

Boundary Conditions

The options in this tab are the same as in the Constituent Materials tab 14 of the **Simulate Diffusion Experiment** command.

The only difference to the **Simulate Diffusion Experiment** command is that the concentration drop in diffusion direction (**Experimental Input**) has no physical units. As described in the Theoretical Background 35, a dimensionless equation is solved, and therefore the concentration given here has no physical unit.



Solver

The options in this tab are the same as in the Solver tab 19 of the **Simulate Diffusion Experiment** command.

Grid

The options in this tab are the same as in the Grid tab 28 of the **Simulate Diffusion Experiment** command.

Equations & References

The **Equations & References** tab shows the relevant equations for illustration and explains the used variables and constants. Remark, that the concentration and relative diffusivity are dimensionless parameters here.

1.4.3 Results

Click **OK** to input the entered parameters, and then click **Run** in the DiffuDict section to start the command. The results are immediately shown in the opening Result Viewer after the process is finished. The screenshot below shows results obtained for the GDL model (Example 1 10). Here, the **Results - Report** subtab shows:

- The relative diffusivity D^* (here called D_{rel}, as defined through equation (7) 6 in %)
- The tortuosity factor κ as defined through equation (9) 6
- The geometric tortuosity $\tau = \sqrt{\kappa}$

Report | Plots | Map

Laplace diffusion

Relative diffusivity D_{rel} / %

60.19	-0.3838	-0.2318
-0.3838	**58.29**	-0.1626
-0.2317	-0.1626	**51.83**

Effective diffusivity D_{eff} / (m^2/s)

unknown	unknown	unknown
unknown	**unknown**	unknown
unknown	unknown	**unknown**

Diffusional tortuosity

	Tortuosity factor κ	Tortuosity τ
X-Direction	1.319	1.148
Y-Direction	1.362	1.167
Z-Direction	1.532	1.238

$\kappa = \tau^2$

N. Epstein; 1989; On tortuosity and the tortuosity factor in flow and diffusion through porous media; Chemical Engineering Science 44(3), p. 777–779.

Effective diffusivity is not calculated in 'Laplace diffusion',
since no base/bulk diffusivity is given as input.

Furthermore, for each computation direction, the used solver, the number of parallel processes, the required iterations, the runtime needed to solve equation (8), the memory usage and the applied stopping criterion are reported:

Diffusion solver data, Z-direction

Solver	EJ
Processes	4
Iterations	27
Runtime	1.2783 min
Memory usage	1.208 GiB
Stopping criterion	tolerance

Total runtime: 00:03:45, Total memory usage: 2.148 GiB

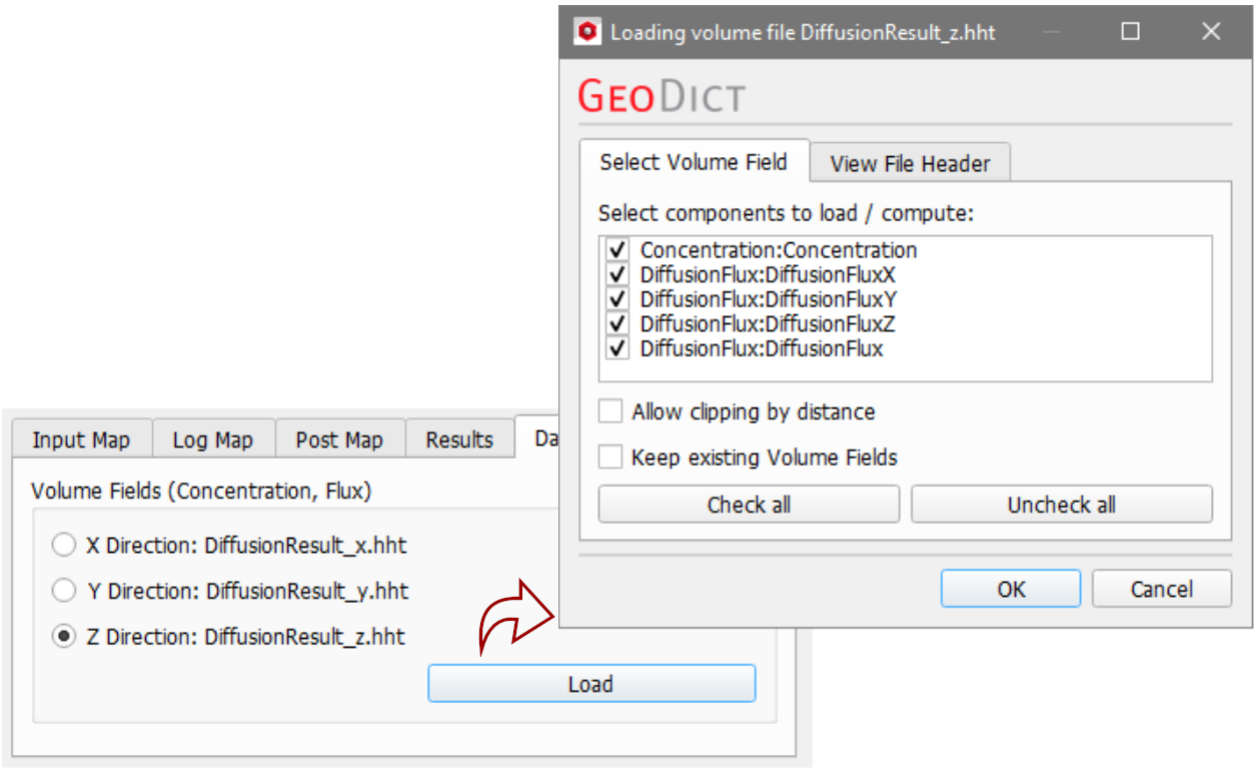
At the bottom, find the overall runtime and the total memory usage, which consists of the memory usage of the PDE solver and the GeoDict executable.

Under the **Plots** sub-tab, the same graphs as for the Simulate Diffusion command[30] are plotted.

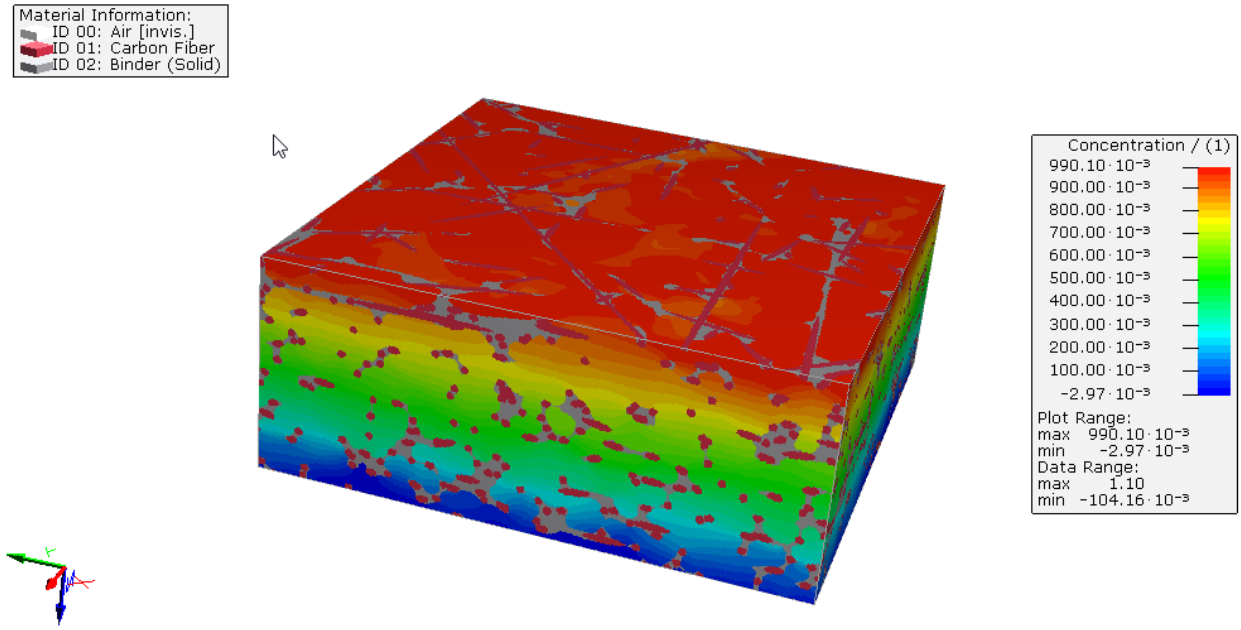
1.4.4 Data Visualization

Under the **Data Visualization** tab, for the Bulk (Laplace) Diffusion command, the result of the diffusivity calculations can be loaded and graphically visualized. Computation Directions[16] that were not previously selected were not computed, and, thus, are unavailable for visualization.

The .hht files contain the local concentration in every voxel. If **Write Diffusion Flux into Solution File** was checked, the result files contain the diffusion flux, too.



After selecting a result file, and clicking **Load Results,** a pop-up dialog allows to select which of the result fields contained in the file should be loaded for visualization. After clicking **OK**, the volume field is displayed over the structure in the Visualization area of the GUI.



Besides the concentration, the diffusion flux can be visualized as result field or, as it is for any other vector valued 3D field, as stream lines or arrow fields. All visualization options are explained in more detail in the Visualization chapter of this User Guide.

1.5 Molecular Diffusion

Use the **Molecular Diffusion** command to model diffusion processes for intermediate Knudsen numbers, i.e., in situations where both molecule-molecule interactions and molecule-wall interactions take place and the diffusion process cannot be described by either one of these processes alone.

This command models the diffusion of particles by a random-walk algorithm. Both molecule-molecule and molecule-wall collisions are modeled. Einstein's formula is used to derive the diffusivity from the computed average particle displacement. The resulting homogenized effective diffusivity depends on gas species, ambient conditions and the pore structure of the material.

The Molecular Diffusion command

- Theoretical Background 41
  Summary of the computational method used by this command.
- Options 42
  How to set the input parameters of the Molecular Diffusion command.
- Results 42
  How to understand the computed results.

1.5.1 Theoretical Background

The Knudsen diffusion algorithm 45 models the movement of molecules in the absence of molecule-molecule collisions. Therefore, it is only valid for high Knudsen numbers. For intermediate Knudsen numbers 7 in the range of 0.01 < Kn < 10, diffusion simulations must also consider molecule-molecule interactions. This can be done by adding molecule-molecule collision to the random walk method.

Again, each single molecule starts with a random, Maxwell-Boltzmann distributed velocity (here, the term “velocity” means the three-dimensional velocity vector, so it includes the direction of the movement). Based on the mean free path length of the gas, an exponentially distributed random distance l is determined, after which the particles will collide with another particle. The particle now moves in a straight line until it either hits a wall or it has traveled the distance l, whatever comes earlier. If it hits a wall, the molecule is reflected diffusely according to Lambert’s cosine law (see Greenwood 61) and leaves the wall with a new random, again Maxwell-Boltzmann distributed velocity and a new random distance l is determined. If it reached the distance l before hitting a wall, it is assumed that the particle collides with another particle. In that case, a new Maxwell-Boltzmann distributed velocity and a new random distance l are drawn and the particle continues its movement with the new velocity.

The molecule continues its way until the desired simulation time is reached. The resulting displacement (distance between the molecules’ start position and end position) is compared to its travel time, which results in the diffusivity for this individual molecule. Thus, to obtain a good estimate for the diffusivity of the whole 3D structure, a high number of simulated molecules is needed.

From the displacement, the diffusivity is determined by using Einstein's formula (16) [45].

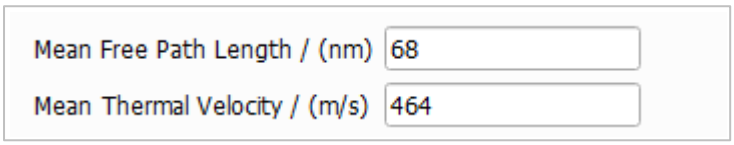
1.5.2 Options

The **Molecular Diffusion** options are organized under the tabs **Constituent Materials**, **Solver Options**, and **Equations & References**.

Constituent Materials

As described for the Bulk (Laplace) Diffusion [36], you set which Material ID corresponds to pore space and which Material IDs denote solids under the **Constituent Materials** tab.

Be aware that the choice of fluid does not influence the results. The relevant parameters describing the fluid are the **Mean Free Path Length** and the **Mean Thermal Velocity**.



Solver Options

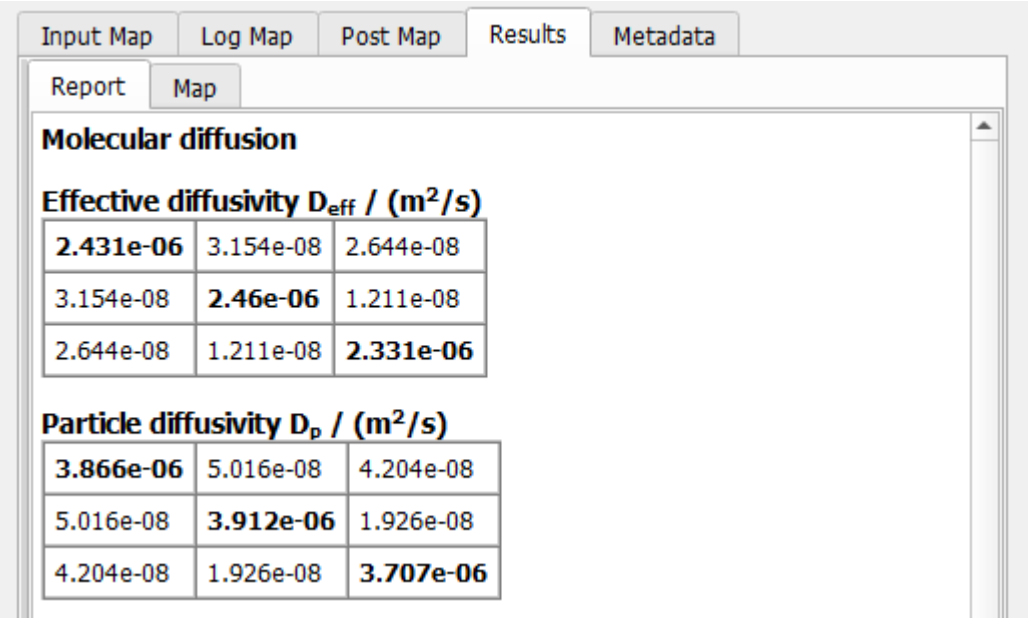
The solver options are identical to those of the Knudsen Diffusion command solver options [47].

Equations & References

The **Equations & References** tab shows the relevant equations for illustration and explains the used variables and constants.

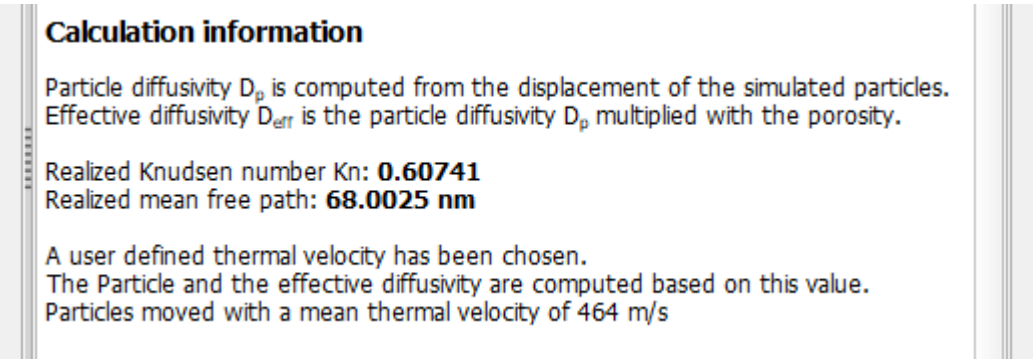
1.5.3 Results

Click **OK** to input the entered parameters, and then click **Run** in the DiffuDict section to start the Molecular Diffusion command. The results are immediately shown in the opening Result Viewer after the process is finished, the screenshots below show the results obtained for the MPL model (Example 3 [12]):



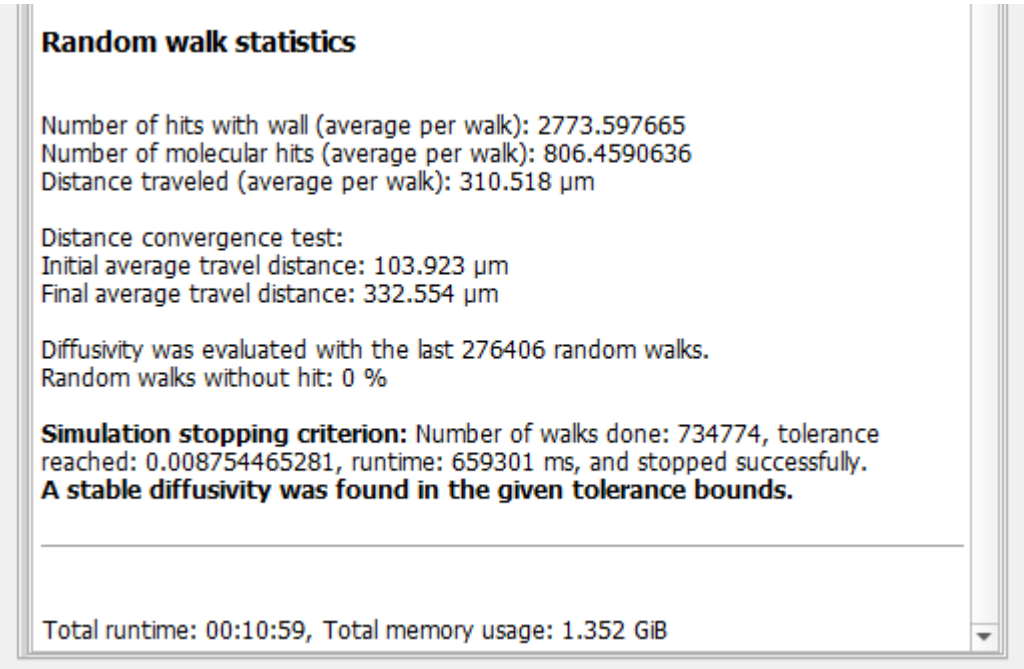
In the **Results - Report** subtab, the following results are reported:

- The **Effective diffusivity** obtained by multiplying the diffusivity of the particles with the porosity.
- The **Particles diffusivity** computed by equation (16) 45 using the user defined mean thermal velocity.



Some calculation information on how to compute the particle diffusivity and effective diffusivity is given as well.

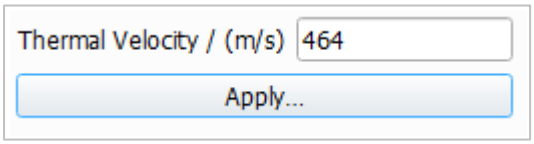
- The Realized Knudsen number is determined as quotient of the realized mean free path and the computed characteristic length.
- The Realized mean free path is computed by dividing the travel distance with the number of molecule-molecule hits. It shows the mean free path realized by the drawn random distribution of free path lengths.



Then, some statistical information about the performed random-walk method is given:

- The average number of collisions with a pore wall a random walker experienced during the simulated time.
- The average number of collisions with another molecule a random walker experienced during the simulation time.
- The average distance that a random walker traveled.
- If **Auto Distance, Check Convergence** was chosen, the initial and final average travel distance used to check the convergence for $t \to \infty$.
- The number of random walks evaluated to achieve the requested tolerance.
- As a check for plausibility of the results: the percentage of random walkers that have not collided with any pore wall during the whole simulation time. If this value is not equal to 0 %, consider increasing the **Average Travel Distance**!
- The criterion that caused the simulation to finish.
- The total runtime and memory usage for this simulation.

The random walk algorithm is done using the **Mean Thermal Velocity** entered in the Constituent Materials input dialog. However, the results can be rescaled to any given mean thermal velocity by entering another **Thermal Velocity** and clicking on the **Apply** button.



1.6 Knudsen Diffusion

Use the **Knudsen Diffusion** command to model dilute gases or diffusion in very small pores, i.e., situations where no molecule-molecule interactions take place and the diffusion process is solely driven by molecule-wall interactions.

This command models the diffusion of particles by a random-walk algorithm. Only molecule-wall collisions are modeled. Einstein's formula is used to derive the diffusivity from the computed average particle displacement. The resulting dimensionless relative diffusivity depends only on the pore structure of the material. As a post-processing step, you may input the mean thermal velocity of the gas species to to derive a dimensional effective Knudsen diffusivity.

The Knudsen Diffusion command

- Theoretical Background 45
  Summary of the computational method used by this command.
- Options 46
  How to set the input parameters of the Knudsen Diffusion command.
- Results 50
  How to understand the computed results.
- Data Visualization 53
  How to visualize the computed results.

1.6.1 Theoretical Background

As described in the DiffuDict theoretical background 4, Knudsen diffusion models the movement of molecules in the absence of molecule-molecule collisions. This means that a molecule will move straight ahead until it encounters a solid obstacle (wall), where it is reflected in a random direction. Numerically, this is solved by a random walk method.

Each single molecule starts with a random, Maxwell-Boltzmann distributed velocity and moves through the pore space until it hits a wall. At the wall, the molecule is reflected diffusively according to Lambert's cosine law (see Greenwood 61) and leaves the wall with a new random, again Maxwell-Boltzmann distributed velocity. The molecule continues its way until the desired simulation time is reached. The resulting displacement (distance between the molecules start position and end position) is compared to its travel time, which results in the diffusivity for this individual molecule. Thus, to obtain a good estimate for the diffusivity of the whole 3D structure, a high number of simulated molecules is needed.

From the displacement, the diffusivity can be determined by Einstein's formula:

$$D = \frac{E[(x_t - x_0)(x_t - x_0)^T]}{2t} \tag{16}$$

where $E[\dots]$ denotes the expectation value, x_0 is the starting position of a particle and x_t the position at time t, and $(.)^T$ denotes the transposed vector.

The resulting 3x3 diffusivity matrix D depends not only on the pore structure of the porous medium, but also on the mean thermal velocity of the diffusing gas. However, if one defines an intrinsic Knudsen diffusivity

$$d_0 = \frac{1}{3} L \bar{v} \tag{17}$$

with the mean thermal velocity $\bar{v}$ of the fluid and the characteristic length L of the porous media, the diffusivity can again be written as

$$D = d_0 D^* \tag{18}$$

Here, D^* again is a dimensionless 3x3 matrix that is independent of the diffusing species. Therefore, the Knudsen diffusion command may determine D^* without requiring the user to define the diffusing species.

Analogously to the approach taken for the bulk diffusivity, it is common in literature (see, e.g., the papers of Tomadakis)[61] to define a **Knudsen tortuosity factor** by writing

$$\kappa = \frac{\eta}{d^*} \tag{19}$$

Unfortunately, this whole definition depends on the definition of the characteristic length L of the porous media, which is by no means a well-defined quantity. Thus, depending on the definition of L, the tortuosity factor may end up being smaller than 1.0, which disagrees with the geometric intuition.

For this reason, this definition of a Knudsen tortuosity factor has been criticized by Zalc et al[61], and they could indeed show, that the Knudsen tortuosity factor matches the tortuosity factor for a specific (and complicated) choice of the characteristic length.

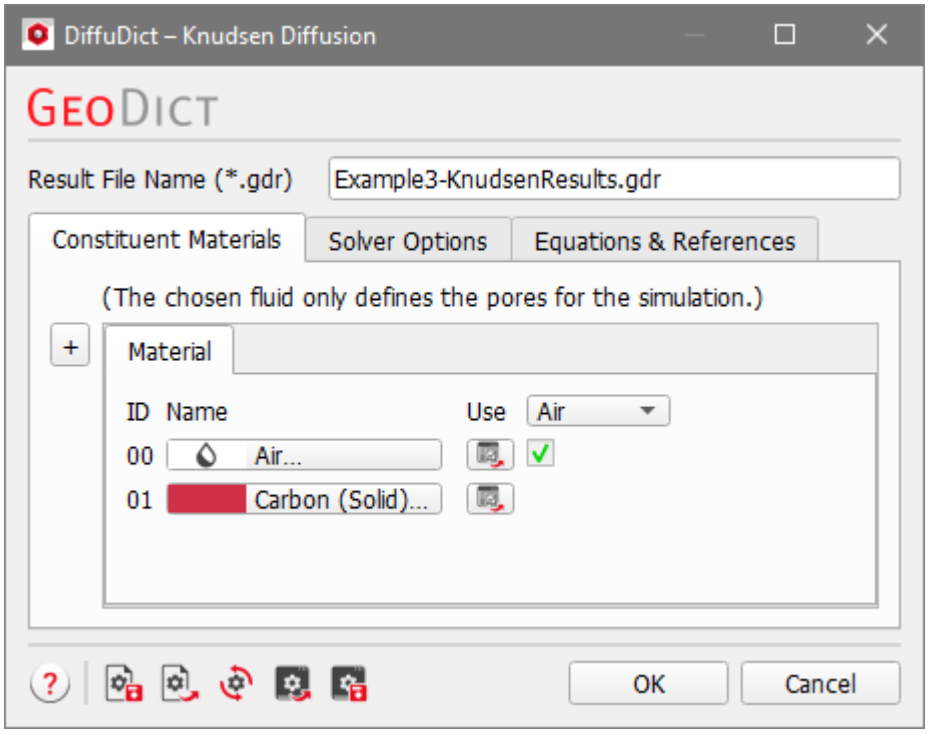
GeoDict does not follow the approach of Zalc et al[61] but computes the ***characteristic length*** L of the structure as the mean path length between two consecutive hits of molecule and wall. Therefore, the reported Knudsen tortuosity factor and the relative diffusivity will differ from the results of the bulk diffusivity simulations, and Knudsen tortuosity factors <1 are possible for some pore geometries.

1.6.2 Options

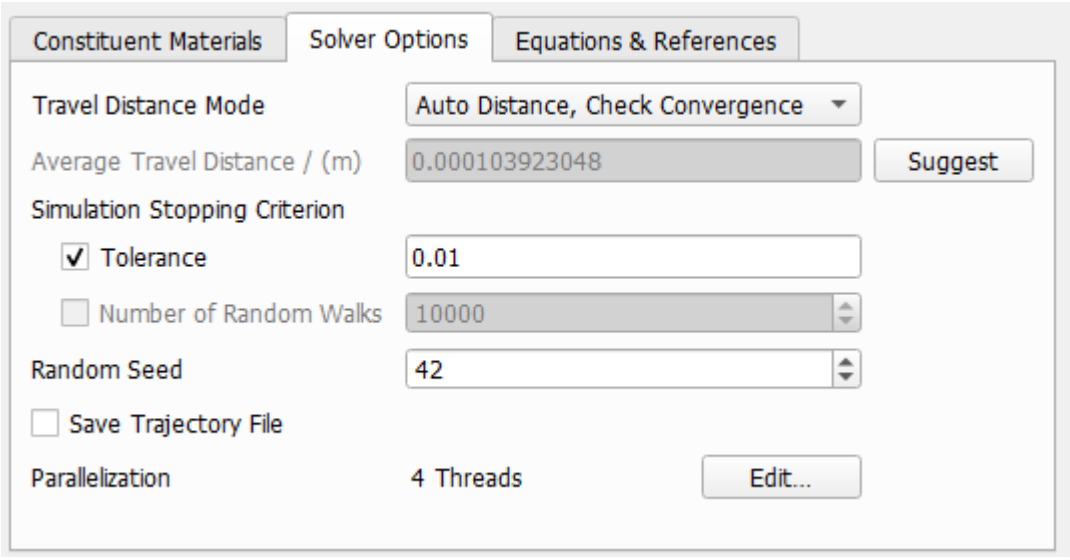
The **Knudsen Diffusion** options are organized under the tabs **Constituent Materials**, **Solver Options**, and **Equations & References**.

Constituent Materials

As described for the Bulk (Laplace) Diffusion, you set the Material ID that corresponds to pore space and those which denote solids under the **Constituent Materials** tab. Be aware, that the choice of fluid does not influence the results. In fact, the Knudsen diffusion simulates the movement of single molecules through empty pores, in which no other molecules are present.



Solver Options



Travel Distance Mode and Average Travel Distance

The computation of the diffusivity D converges when the number of particles $n \to \infty$ and the simulated time $t \to \infty$.

If the default mode **Auto Distance, Check Convergence** is chosen, the simulation time t is iteratively increased until convergence is achieved.

If **Fixed Distance** is chosen, the convergence for $t \to \infty$ is not checked, and a fixed simulation time is chosen. In this case, you have to enter the **Average Travel Distance**, which also fixes the simulation time t. A distance is entered instead of a time value here, because the simulation runs with a dimensionless particle velocity. The mean thermal velocity of the diffusing particles is set during post-processing, and the results can be rescaled without the need to recompute the random walks.

If **Suggest** is clicked, GeoDict suggests a value for the average travel distance.

Simulation Stopping Criterion

Select either **Tolerance** or a fixed **Number of Random Walks**.

With **Tolerance**, the solver stops if the relative standard deviation of the diagonal entries of the resulting diffusivity tensor (computed with equation (16) 45) is smaller than the given threshold.

Alternatively, the parameter **Number of Random Walks** determines the number of molecules that are traced. The higher the number, the higher is the achieved accuracy. Convergence of the result for $n \to \infty$ is not checked in this case and no given tolerance is achieved. This stopping criterion is only selectable if **Fixed Distance** was chosen as **Travel Distance Mode**.

Random Seed

Random Seed sets the seed of the underlying random number generator. The same random seed generates identical results; results with different random seeds should be similar but not the same. If results computed with different random seeds are significantly different, the **Number of Random Walks** and probably the **Average Travel Distance** values should be increased.

Save Trajectory File

By checking **Save Trajectory File**, the particle trajectories are saved and can be visualized from the result viewer. It is not recommended to store the trajectories when the option **Auto Distance, Check Convergence** was chosen. In that case, the number of particles and the length of the trajectories typically becomes too large and the solver will fail due to a lack of RAM or hard disk space available.

Parallelization

Depending on the purchased license, the simulation process can be parallelized.

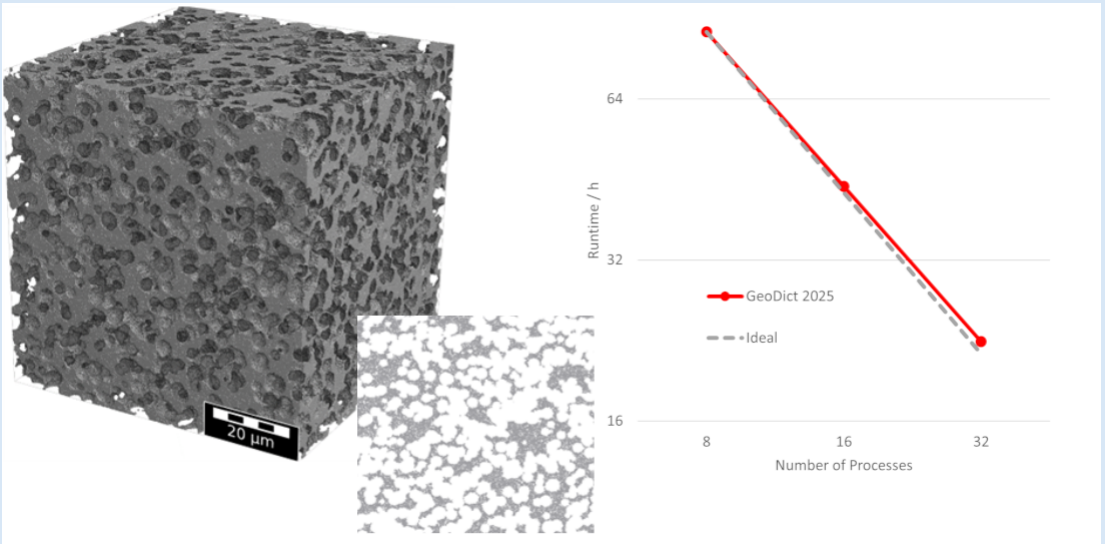
The **Parallelization Options** dialog box opens when clicking the **Edit...** button, to choose between **Sequential**, **Parallel (Shared Memory)** and **Automatic Number Of Threads**. When **Parallel** is chosen, the **Number of Threads** to run can be entered. If the entered number is larger than the one the license supports, an error message appears. For details on how to set up und run parallel computations, consult the High Performance Computations chapter.

Parallelization Benchmarks Result

As an example for the parallelization, Knudsen diffusion in a microporous layer (MPL) is computed with different number of processes. The computation is run on a server with 2 x Intel E5-2697A v4 processors with 16 cores each, running with a maximum of 3.60 GHz, and 128 GB RAM. The MPL structure of size 2,000 x 2,000 x 2,000 was created with GrainGeo and ProcessGeo. It has a porosity of 64%

The computation for the structure with 8 billion voxels requires only 32 GB of RAM. The effective Knudsen diffusivity for oxygen at 20°C in z-direction is 8.82e-05 m^2/s.

Runtimes for different number of parallel processes are shown in the following graph:



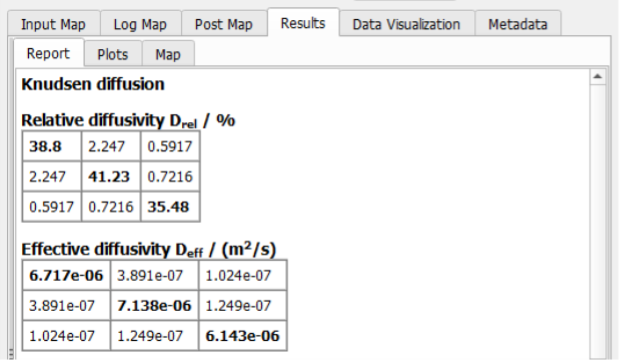
Equations & References

The **Equations & References** tab shows the relevant equations for illustration and explains the used variables and constants.

1.6.3 Results

Click **OK** to input the entered parameters, and then click **Run** in the DiffuDict section to start the Knudsen Diffusion command. The results are immediately shown in the opening Result Viewer after the process is finished, the screenshots below show the results obtained for the MPL model (Example 3 12).

The Report Tab


Input Map | Log Map | Post Map | Results | Data Visualization | Metadata

Report | Plots | Map

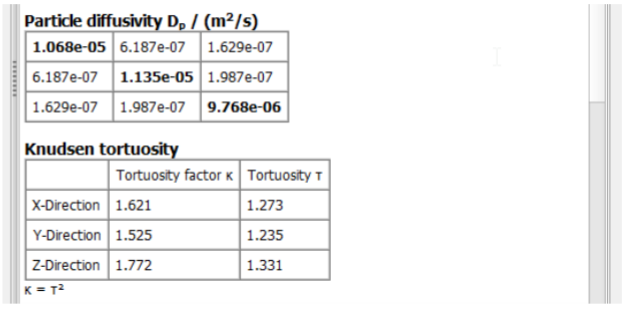
Knudsen diffusion

Relative diffusivity D_{rel} / %

38.8	2.247	0.5917
2.247	**41.23**	0.7216
0.5917	0.7216	**35.48**

Effective diffusivity D_{eff} / (m^2/s)

6.717e-06	3.891e-07	1.024e-07
3.891e-07	**7.138e-06**	1.249e-07
1.024e-07	1.249e-07	**6.143e-06**


Particle diffusivity D_p / (m^2/s)

1.068e-05	6.187e-07	1.629e-07
6.187e-07	**1.135e-05**	1.987e-07
1.629e-07	1.987e-07	**9.768e-06**

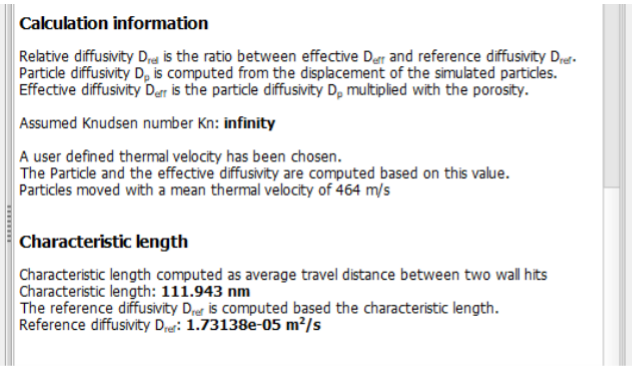
Knudsen tortuosity

	Tortuosity factor κ	Tortuosity τ
X-Direction	1.621	1.273
Y-Direction	1.525	1.235
Z-Direction	1.772	1.331

$\kappa = \tau^2$

In the **Results - Report** subtab, the following results are reported:

- The dimensionless **Relative diffusivity** D^* as defined through equation (18) 46.
- The **Effective diffusivity** in porous media. These values are obtained by multiplying the diffusivity of the particles with the porosity.
- The **Particle diffusivity** computed by equation (16) 45 using either the user defined mean thermal velocity or the default velocity of 1 voxel/s.
- The resulting **Knudsen tortuosity** including both Tortuosity factors κ and Tortuosity τ for each space direction as defined through equation (19) 46.


Calculation information

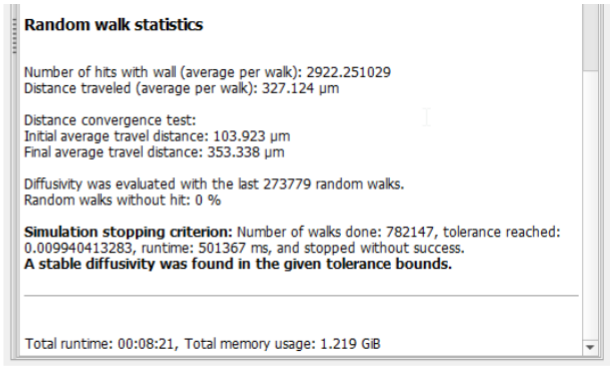
Relative diffusivity D_{rel} is the ratio between effective D_{eff} and reference diffusivity D_{ref}.
Particle diffusivity D_p is computed from the displacement of the simulated particles.
Effective diffusivity D_{eff} is the particle diffusivity D_p multiplied with the porosity.

Assumed Knudsen number Kn: **infinity**

A user defined thermal velocity has been chosen.
The Particle and the effective diffusivity are computed based on this value.
Particles moved with a mean thermal velocity of 464 m/s

Characteristic length

Characteristic length computed as average travel distance between two wall hits
Characteristic length: **111.943 nm**
The reference diffusivity D_{ref} is computed based the characteristic length.
Reference diffusivity D_{ref}: **1.73138e-05 m^2/s**


Random walk statistics

Number of hits with wall (average per walk): 2922.251029
Distance traveled (average per walk): 327.124 µm

Distance convergence test:
Initial average travel distance: 103.923 µm
Final average travel distance: 353.338 µm

Diffusivity was evaluated with the last 273779 random walks.
Random walks without hit: 0 %

Simulation stopping criterion: Number of walks done: 782147, tolerance reached: 0.009940413283, runtime: 501367 ms, and stopped without success.
A stable diffusivity was found in the given tolerance bounds.

Total runtime: 00:08:21, Total memory usage: 1.219 GiB

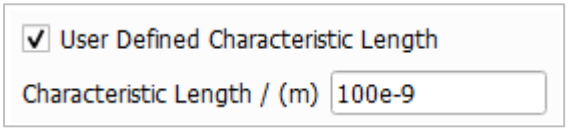
Then, some calculation information is given:

- The computed **Characteristic Length** L.
- The corresponding reference diffusivity computed through (17) 46.
- The overall runtime and memory usage.

Afterwards, some statistical information about the performed random walk method is reported:

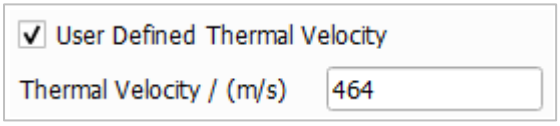
- The average number of collisions with a pore wall a random walker experienced during the simulated time.
- The average distance that a random walker traveled.
- If **Auto Distance, Check Convergence** was chosen, the initial and final average travel distance used to check the convergence for $t \to \infty$.
- The number of random walks evaluated to achieve the requested tolerance.
- As a check for plausibility of the results: the percentage of random walkers that have not collided with any pore wall during the whole simulation time. If this value is not equal to 0 %, consider increasing the **Average Travel Distance**!
- The criterion that caused the simulation to finish.

Instead of the computed characteristic length, a user defined characteristic length can be used to determine the relative diffusivity by entering its value in the left panel and clicking the **Apply...** button.



Changing the characteristic length will also change the computed **Relative diffusivity** and the **Knudsen tortuosity.**

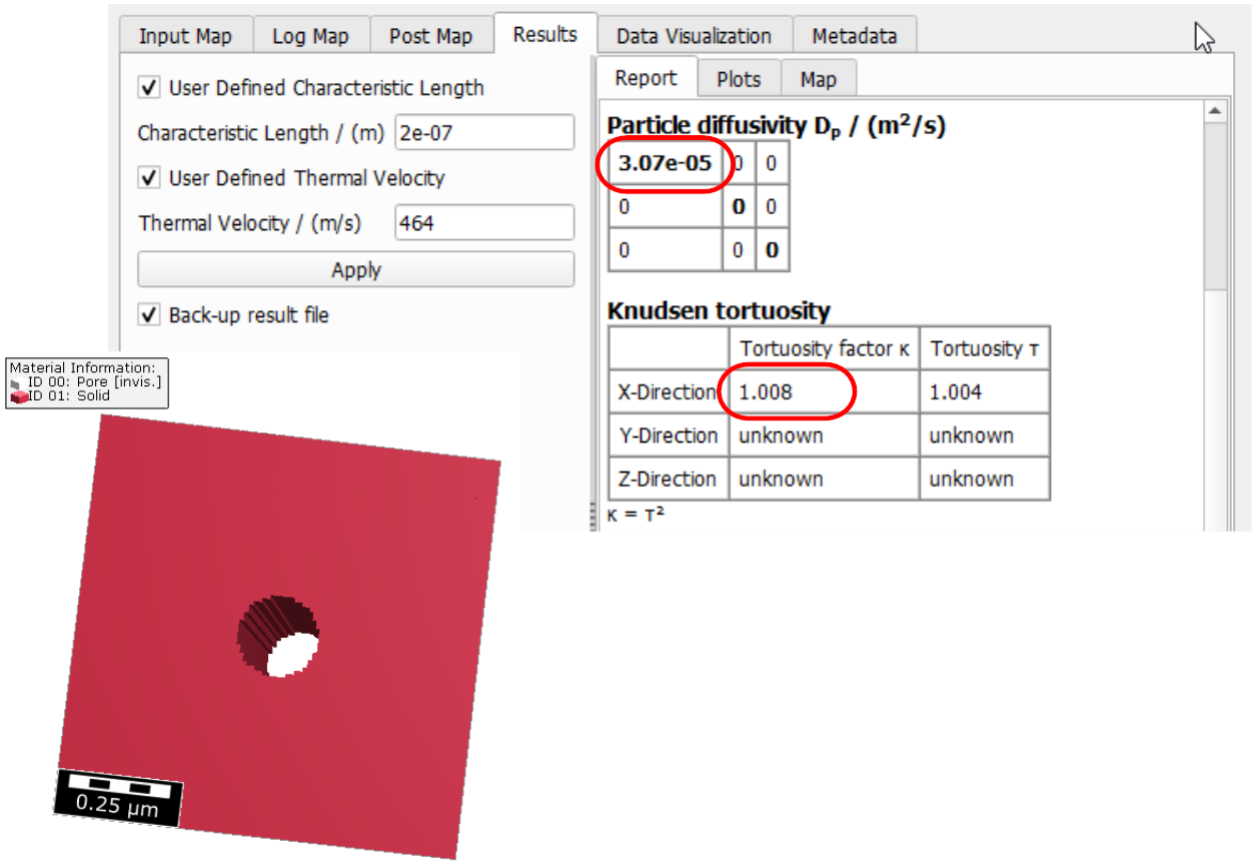
The random walk algorithm is done using a velocity of 1 voxel per second. However, the results can be rescaled to any given mean thermal velocity by entering a **User Defined Thermal Velocity**.



Changing the thermal velocity will also change the computed **Particle diffusivity** and the **Effective diffusivity**. To get the diffusivity in the structure for a specific diffusivity as input (e.g., for oxygen at 20°C), define the corresponding thermal velocity (for oxygen at 20°C: 440.4 m/s). The **Particle diffusivity** describes how quickly an oxygen molecule will travel through the pores of the structure by diffusive reflection at the pore walls. The **Effective diffusivity** describes how quickly a concentration of species will move through the homogenized porous media by diffusion.

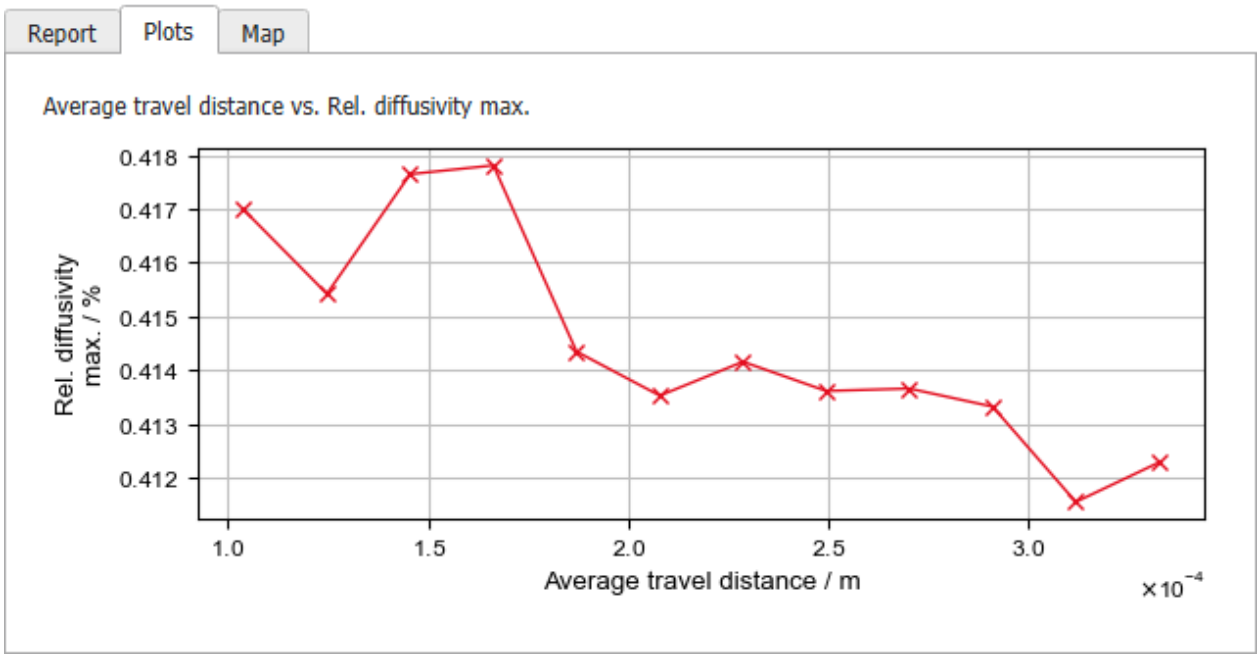
For a cylindrical pore, the **Particle diffusivity** is expected to match $1/3\ d\bar{v}$, where d is the diameter of the pore, and $\bar{v}$ the mean thermal velocity. To achieve this result for a cylindrical pore, the pore diameter has to be entered manually as **User Defined Characteristic Length**.

In this example, for a pore diameter of 200 nm and a mean thermal velocity of 464 m/s, a diffusivity of approx. 3.07e-5 m^2/s is computed, which is very close to the expected value of 3.09e-5 m^2/s. For cylindrical pores, the Knudsen tortuosity factor is 1 in the direction of the pore, when using the pore diameter as characteristic length.



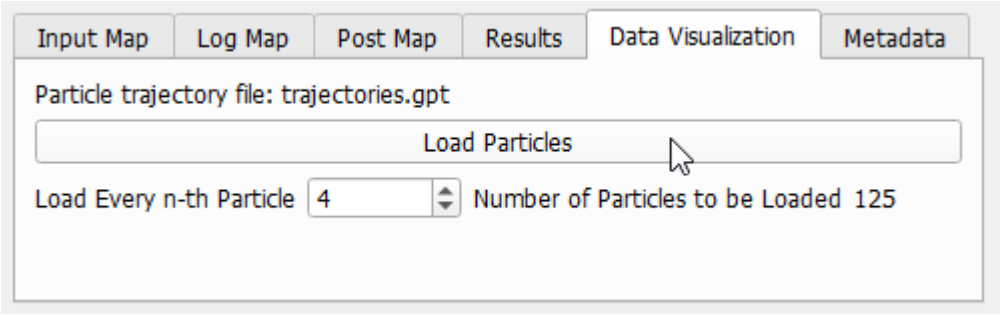
The Plots Tab

In the **Results – Plots** subtab, the determined **Rel. Diffusivity Max**.(which is the largest relative diffusivity value achieved in X-, Y-, or Z-direction) is shown over the average travel distance, starting from the initial average travel distance to the final average travel distance. These values are computed during the computation to check for convergence, and the last point corresponds to the final result. Be aware that this plot does not change when the relative diffusivity changes in the report tab because of a user defined characteristic length, i.e., it always shows the relative diffusivity values corresponding to the computed characteristic length.



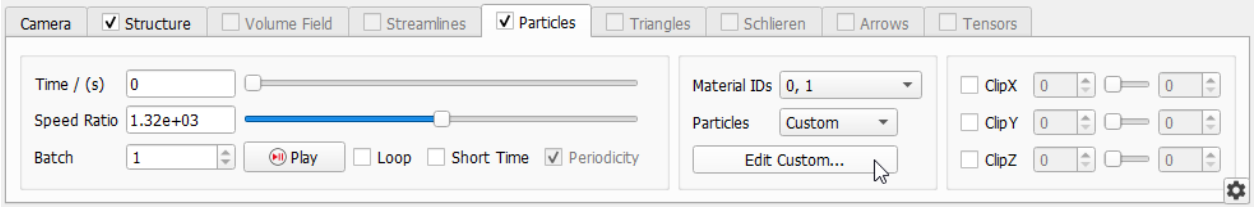
1.6.4 Data Visualization

If **Save Trajectory File** 48 was checked, it is possible to load and visualize the particle trajectories under the **Data Visualization** tab. By clicking **Load Particles**, the GPT file (GPT, GeoDict particle trajectories) is loaded.

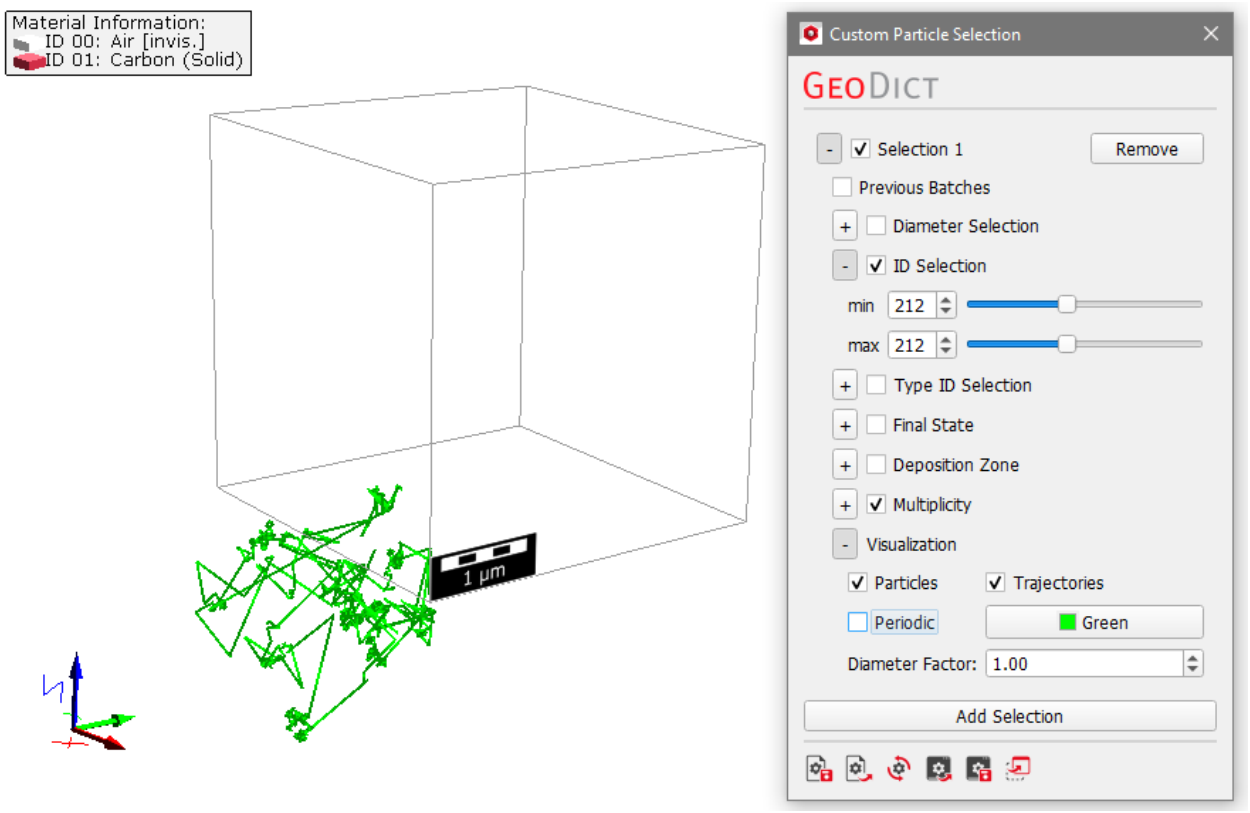


As the number of simulated random walks may be very high, it is possible to load only **Every n-th Particle**. The **Number of Particles to be Loaded** is reported.

The **Particles** tab is activated in the Visualization panel above the Visualization area, where the structure and the particles are displayed. Turn off the structure model by unchecking **View → Structure** in the menu bar. Click **Play** to visualize the movement of the diffusing particles. This will show a large number of particles moving around. To have a closer look at a single particle, choose **Custom** in the **Particles** pull-down menu. Then, click the **Edit Custom...** button (that appears underneath) to open the **Custom Particle Selection**.



With the settings shown below, the trajectory and movement of a single particle (number 212) can be visualized.



Due to the periodic boundary conditions for the computations, a particle trajectory may leave the domain and enter on the opposite side multiple times. If **Periodic** is checked, the trajectory is shown completely inside of the surrounding bounding box, if **Periodic** is unchecked (as in the shown example), the trajectory may be shown outside of the bounding box.

1.7 Bosanquet Approximation

Use the **Bosanquet Approximation** command to approximate the diffusivity for intermediate Knudsen numbers. You must compute the Knudsen diffusion and the Bulk (Laplace) Diffusion before you can apply this command.

This command uses the relative diffusivity at $\mathrm{Kn} = 0$ (computed with the Bulk Diffusion command) and $\mathrm{Kn} = \infty$ (computed with the Knudsen diffusion command) as input. These input parameters describe the geometry of the 3D structure. With this input, Bosanquet's formula allows to estimate the dimensional effective diffusivity for any gas species at arbitrary Knudsen numbers.

Be aware that the result is a quick approximation, and the specific diffusion process is not simulated. To model the specific process, you can use the **Molecular Diffusion** 41 command.

The Bosanquet Approximation command

- Theoretical Background 55
  Equations solved with this command.
- Options 56
  How to set the input parameters of the Bosanquet Approximation command.
- Results 57
  How to understand the computed results.

1.7.1 Theoretical Background

Bosanquet's approximation (Pollard and Present 61) is an estimation of the effective diffusivity at intermediate Knudsen numbers. As stated in the DiffuDict theoretical background 7, the formula is

$$D = \left(D_{\mathrm{Kn}=0}^{-1} + D_{\mathrm{Kn}=\infty}^{-1}\right)^{-1} \tag{20}$$

It computes the effective diffusivity from bulk and Knudsen diffusivity based on the idea that the effect can be described as the sum of two different resistances to the particle movement.

The **Bosanquet Approximation** command uses the results of the Bulk (Laplace) Diffusion 35 and the Knudsen Diffusion 45 commands to compute the resulting diffusivity D. However, both commands have not computed $D_{\mathrm{Kn}=0}^{-1}$ and $D_{\mathrm{Kn}=\infty}^{-1}$, respectively, but the corresponding relative diffusivity.

For unary gases, the diffusivities $D_{\mathrm{Kn}=0}^{-1}$ and $D_{\mathrm{Kn}=\infty}^{-1}$ can be determined from these dimensionless matrices with the help of the mean thermal velocity $\bar{v}$ and the mean free path λ:

$$D_{\mathrm{Kn}=0} = \frac{1}{3}\lambda\bar{v}D_{\mathrm{Kn}=0}^{*} \tag{21}$$

$$D_{\mathrm{Kn}=\infty} = \frac{1}{3} L \bar{v} D^*_{\mathrm{Kn}=\infty} \tag{22}$$

where L is the characteristic length as used in the **Knudsen Diffusion** command. With the definition of the Knudsen number

$$\mathrm{Kn} = \frac{\lambda}{L} \tag{23}$$

it becomes obvious that Bosanquets formula delivers $D = D_{\mathrm{Kn}=0}$ for Kn=0 and $D = D_{\mathrm{Kn}=\infty}$ for $\mathrm{Kn} = \infty$ and in general the Knudsen number describes which of the two diffusivities is dominant.

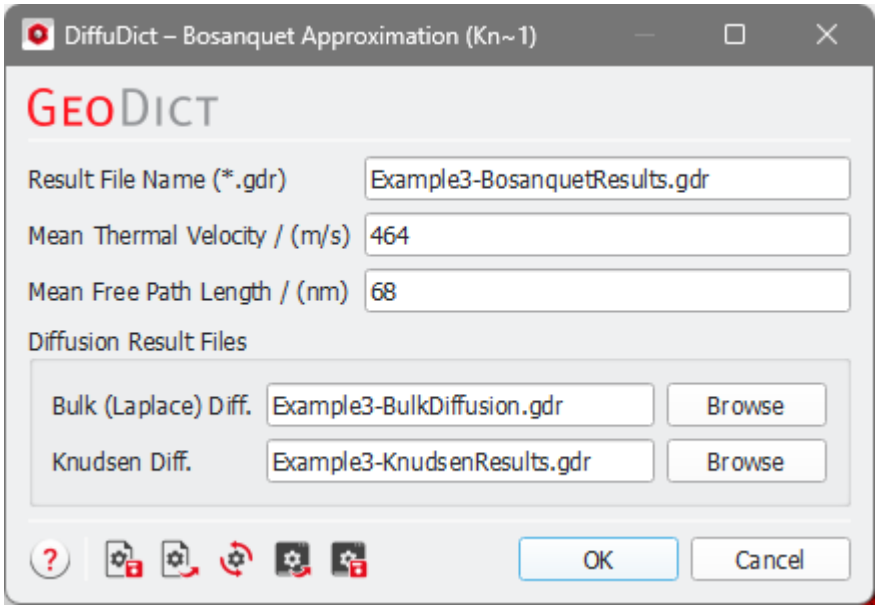
For binary gases, equation (17) 46 no longer holds true, but is possible to use equation (7) 6 where d_0 describes a measurable quantity:

$$D_{\mathrm{Kn}=0} = d_0 D^*_{\mathrm{Kn}=0} \tag{24}$$

In this case, Bosanquets formula still holds true, but D has to be computed manually, and DiffuDict's **Bosanquet Approximation** command cannot be used to compute it.

1.7.2 Options

When **Bosanquet Approximation (Kn ≅ 1)** is chosen from the pull-down menu in the DiffuDict section, clicking the **Options' Edit...** button opens the **Bosanquet Approximation** dialog box.



The **Mean Thermal Velocity** is the average speed of the thermal motion of particles, which make up a gas, liquid, etc. It can be calculated by

$$\bar{v} = \sqrt{\frac{8kT}{\Pi m}} \tag{25}$$

where k is the Boltzmann constant, T the temperature and m the molecular mass.

The **Mean Free Path Length** is the average distance traveled by a moving particle between successive collisions with other molecules. The default **Mean Thermal Velocity** value of 464 m/s and **Mean Free Path Length** value of 68 nm roughly correspond to air/nitrogen at ambient temperature (20° C, 293° K) and pressure (1013 bar, 1 atm).

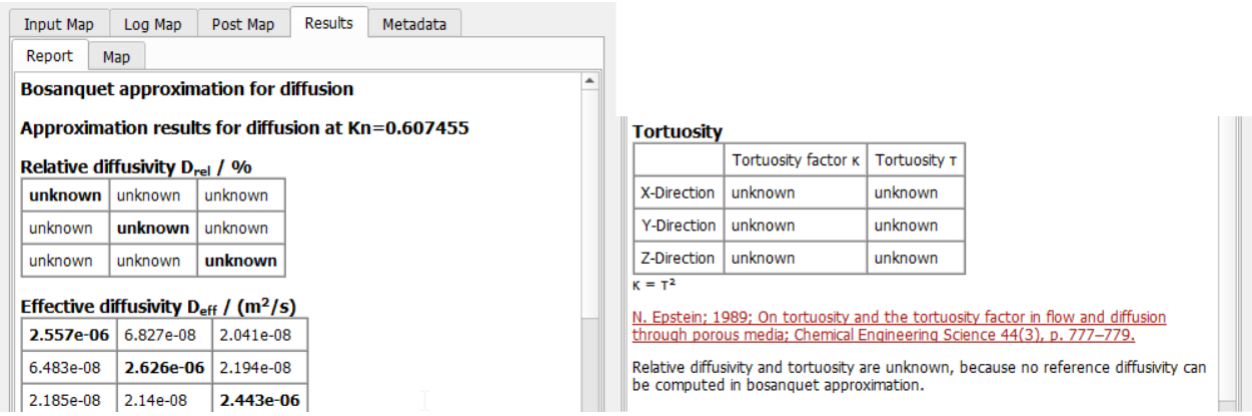
Bulk (Laplace) Diffusion and **Knudsen Diffusion** must be previously computed and the path to the corresponding result files must be loaded using the **Browse** buttons. The loaded result files must correspond to calculations of Bulk (Laplace) Diffusion and Knudsen Diffusion, respectively. Otherwise, a warning appears when clicking **OK** in the Bosanquet Approximation dialog box. The loaded result files must also correspond to the current structure; otherwise, an error message is displayed after clicking Run.

1.7.3 Results

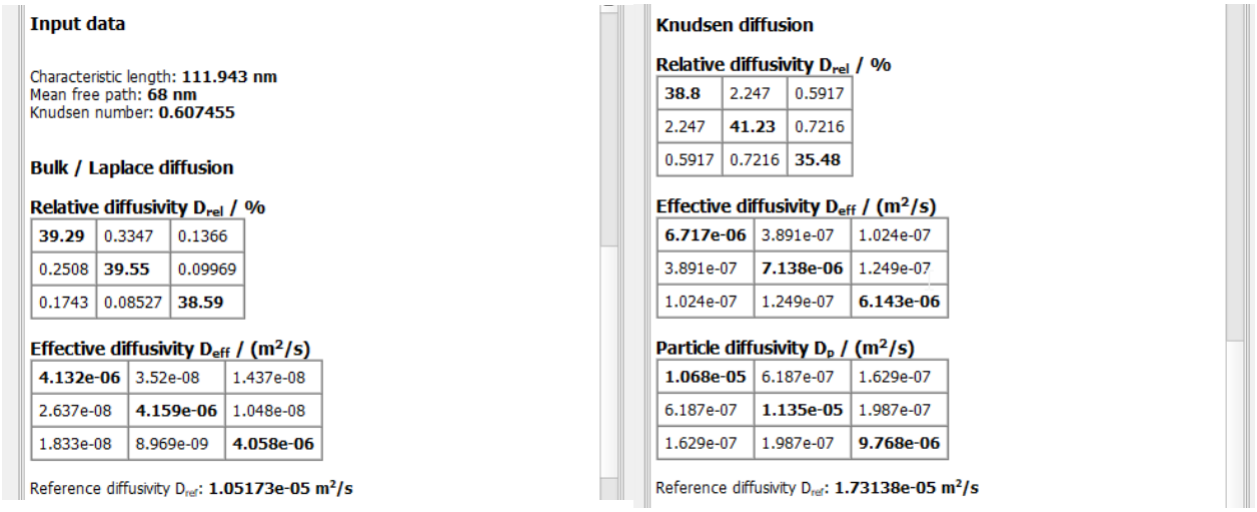
Click **OK** to input the entered parameters, and then click **Run** in the DiffuDict section to start the Bosanquet Approximation command.

The results are instantly shown in the opening Result Viewer after the process is finished, the screenshot below shows the results obtained for the MPL model (Example 3 12):

In the **Results – Report** subtab, the following values are shown:



- The effective diffusivity D (here named D_{eff}), computed with Bosanquet's formula (20) 55.
- The relative diffusivity and the tortuosity are unknown, as there is no common reference diffusivity.



Input data

Characteristic length: **111.943 nm**
Mean free path: **68 nm**
Knudsen number: **0.607455**

Bulk / Laplace diffusion

Relative diffusivity D_{rel} / %

39.29	0.3347	0.1366
0.2508	**39.55**	0.09969
0.1743	0.08527	**38.59**

Effective diffusivity D_{eff} / (m^2/s)

4.132e-06	3.52e-08	1.437e-08
2.637e-08	**4.159e-06**	1.048e-08
1.833e-08	8.969e-09	**4.058e-06**

Reference diffusivity D_{ref}: **1.05173e-05 m^2/s**

Knudsen diffusion

Relative diffusivity D_{rel} / %

38.8	2.247	0.5917
2.247	**41.23**	0.7216
0.5917	0.7216	**35.48**

Effective diffusivity D_{eff} / (m^2/s)

6.717e-06	3.891e-07	1.024e-07
3.891e-07	**7.138e-06**	1.249e-07
1.024e-07	1.249e-07	**6.143e-06**

Particle diffusivity D_p / (m^2/s)

1.068e-05	6.187e-07	1.629e-07
6.187e-07	**1.135e-05**	1.987e-07
1.629e-07	1.987e-07	**9.768e-06**

Reference diffusivity D_{ref}: **1.73138e-05 m^2/s**

- The characteristic length L as used in equation (17) 46 and computed with the Knudsen Diffusion command.
- Mean free path λ, as entered by the user.
- Knudsen number computed as $\mathrm{Kn} = \lambda/L$.
- Relative bulk diffusivity $D^*_{\mathrm{Kn}=0}$ as defined in equation (7) 6 and computed with the Bulk (Laplace) Diffusion command.
- Effective bulk diffusivity $D_{\mathrm{Kn}=0}$, determined by multiplication of the dimensionless bulk diffusivity with the reference diffusivity $1/3\ \lambda\bar{v}$ as in equation (21) 55.
- Relative Knudsen diffusivity $D^*_{\mathrm{Kn}=\infty}$ as defined in equation (18) 46 and computed with the Knudsen Diffusion command.
- Effective Knudsen diffusivity $D_{\mathrm{Kn}=\infty}$, determined by multiplication of the dimensionless Knudsen diffusivity with the reference diffusivity $1/3\ L\bar{v}$ as in equation (22) 56.
- The Particles diffusivity for Knudsen diffusion computed by equation (16) 45 using the user defined mean thermal velocity.
- Total runtime and total memory usage.

1.8 Diffusion GeoApps

A single app is available in the selection box, the **Concentration Dependent Diffusivity** app.

Apps included in DiffuDict

- Concentration Dependent Diffusivity 59
  Approximate a diffusion process where the local diffusivity depends on the concentration of the diffusing species.

1.8.1 Concentration Dependent Diffusivity

The **Concentration Dependent Diffusivity** script approximates a diffusion process where the local diffusivity depends on the concentration of the diffusing species. This is achieved by using the existing constant diffusivity solver in an iterative process, where the local diffusivity in one iteration depends on the local concentration computed in the previous iteration.

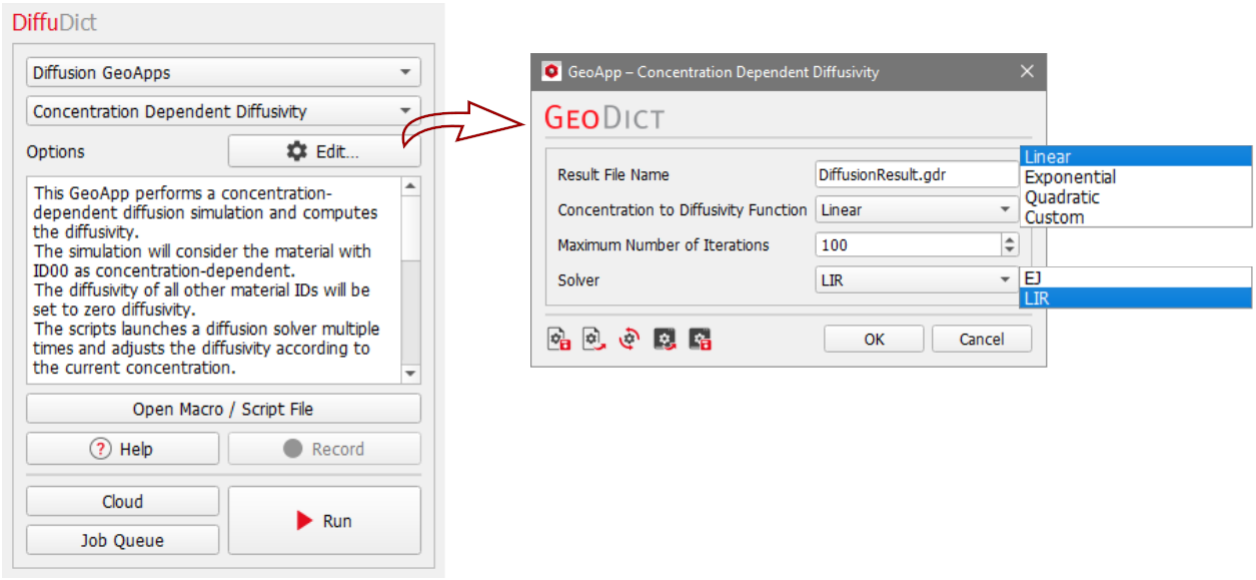
Modules needed to run this GeoApp:

DiffuDict

The window below the selection box shows the description of the script and its input parameters. Click **Open Macro / Script File** to open the underlying Python script in a text editor. Be aware that administrator privileges would be required to change the macro if GeoDict has been installed in the default location for all users.

Click **Run** to start the computation. When the simulation is finished, a DiffuDict result file is opened in the **Result Viewer**.

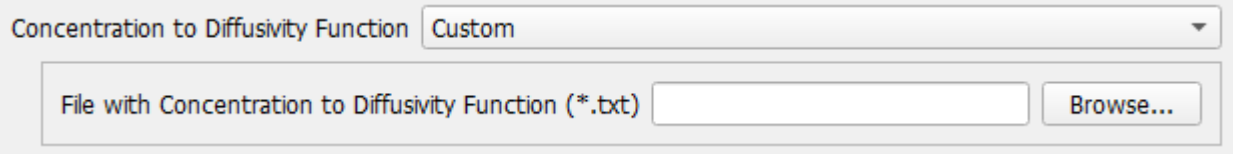
Click **Edit...** to open the dialog to adjust the available parameters.



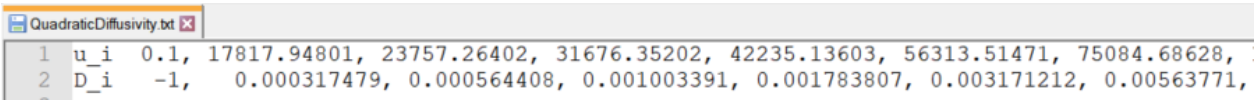
Under **Concentration to Diffusivity Function**, define how the diffusivity depends on concentration. **Linear**, **Exponential**, and **Quadratic** are predefined concentration to diffusivity functions. These predefined functions are stored as text file in the GeoDict installation folder in

the subfolder `C:\Program Files\Math2Market GmbH\GeoDict 2026\GeoApps\General\Physical Properties\DiffusivityFuncs`.

If you want to define your own concentration to diffusivity function select **Custom**.



Create a text file (*.txt) describing the relationship between diffusivity value (`D_i`) and concentration value (`u_i`) as follows:



For example, copy one of the predefined files to a project specific file location and edit it in a text editor. Since the diffusivity is given in bins of a specific range, the number of elements differs by 1 with respect to the number of concentration values. Therefore, the diffusivity value-pairs must start with a sentinel, e.g., (0.1, -1), followed by the desired values.Due to the iterative process described below, after the sentinel 15 different value pairs must be given.

The **Maximum Number of Iterations** can be specified in order to increase the degree of convergence. For the solver choose between the **EJ** and the **LIR** solvers.

The effective diffusivity of the material for the chosen concentration difference is always computed in z-direction according to

$$\operatorname{div}(D(u)\nabla u) = 0 \tag{26}$$

Here, u is the concentration and D the concentration-dependent diffusivity. For the surfaces, no-slip boundary conditions are applied. In Z-direction, constant Dirichlet-boundary conditions are applied where the constants are taken from the given diffusivity file as minimum and maximum values. In the tangential directions, symmetric boundary conditions are used. The solution to the non-linear equation (26) 60 is found in an iterative manner by dividing the pore space into 15 different materials according to the values of u and assigning diffusivity, then solve a linear approximation to (26) 60 for u and repeat until convergence is reached, i.e., when the distribution of the materials does not need to be adjusted anymore for the newly computed u.

Please contact support@math2market.de for more help and information regarding concentration dependent diffusivity simulations, if you are interested in this topic.

1.9 References

H. Babovsky, *On Knudsen flows within thin tubes*, J. Stat. Physics 44, 865-878 (1986)

J. Becker, C. Wieser, S. Fell, K. Steiner, *A multi-scale approach to material modelling of fuel cell diffusion media*, International Journal of Heat and Mass Transfer 54, 1360-1368 (2011)

N. Epstein, *On tortuosity and the tortuosity factor in flow and diffusion through porous media*, Chemical Engineering Science, Vol. 44, No 3, 777-779 (1989)

J. Greenwood, *The correct and incorrect generation of a cosine distribution of scattered particles for Monte-Carlo modelling of vacuum systems*, Vacuum 67, 217-222 (2002)

U. Hornung, *Homogenization and porous media*, Springer (1997)

S. Linden, A. Wiegmann and H. Hagen, *The LIR space partitioning system applied to the Stokes equations*, Graphical Models 82, 58-66 (2015), https://doi.org/10.1016/j.gmod.2015.06.003

W. G. Pollard and R. D. Present, *On gaseous self-diffusion in long capillary tubes*, Phys. Rev. 73, 762 (1948)

M. Tomadakis and S. Sotirchos, *Effective Knudsen diffusivities in structures of randomly overlapping fibers*, AIChE J. 37, 74-86 (1991)

M. Tomadakis and S. Sotirchos, *Ordinary and transition regime diffusion in random fiber structures*, AIChE J. 39, 397-412 (1993)

P. Wesseling, *An Introduction to Multigrid Methods*, (2004)

A. Wiegmann and A. Zemitis, *EJ-HEAT: A fast explicit jump harmonic averaging solver for the effective heat conductivity of composite materials*, Report of the Fraunhofer ITWM, Kaiserslautern, Vol. 94 (2006)

J. M. Zalc, S. C. Reyes and E. Iglesia, *The effects of diffusion mechanism and void structure on transport rates and tortuosity factors in complex porous structures*, Chem. Eng. Sci. 759, 2947-2960 (2004)

Citation: Math2Market GmbH, 2026: GeoDict 2026 User Guide, DiffuDict, Math2Market GmbH, Germany, doi.org/10.30423/userguide.geodict

MATH 2 MARKET

Math2Market GmbH
Richard-Wagner-Str. 1, 67655 Kaiserslautern / Germany
www.math2market.com