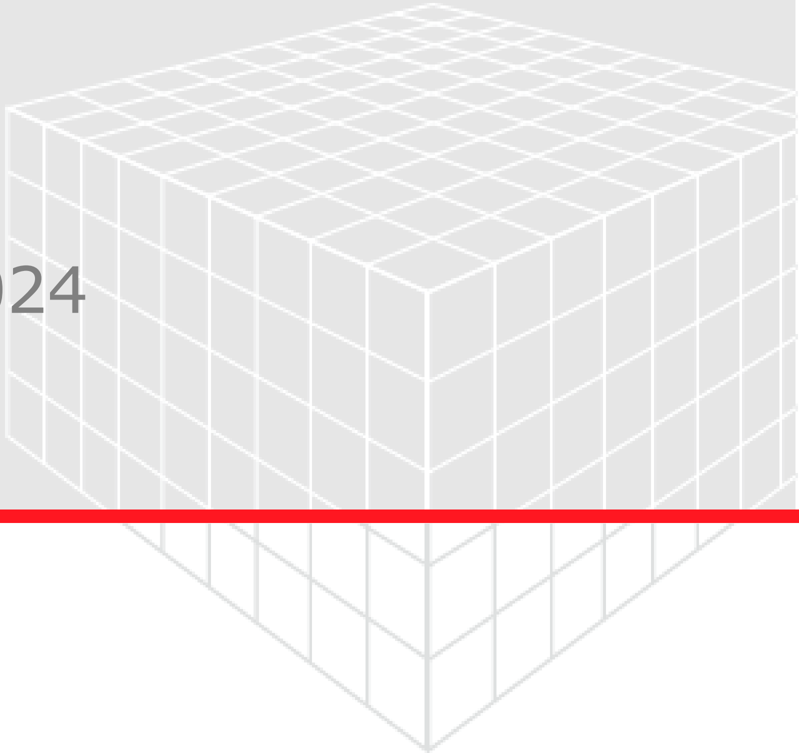
DiffuDict



User Guide

GeoDict release 2024

Published: March 14, 2024

Reviewed: June 14, 2024

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

Citation:
Jürgen Becker, Andreas Wiegmann, Liping Cheng, Barbara Planas. GeoDict 2024 User Guide. DiffuDict handbook. Math2Market GmbH, Germany, doi.org/10.30423/userguide.geodict

The handbooks in the User Guide series of Math2Market GmbH can be obtained from:

Math2Market GmbH
Richard-Wagner-Strasse 1
67655 Kaiserslautern
Germany

Phone: +49 631 205 605 0
Fax: +49 631 205 605 99
Email: info@math2market.de
Web: www.math2market.de

SIMULATION OF DIFFUSION PROCESSES WITH DIFFUDICT

Diffusion is 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.

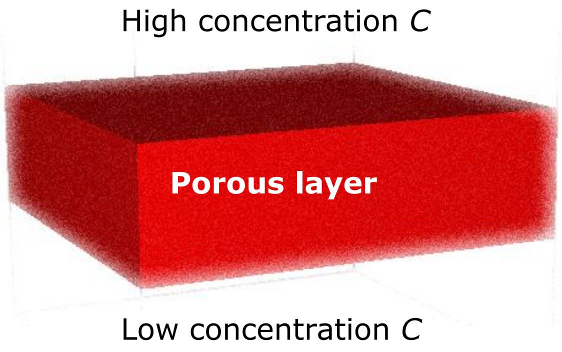
A distinguishing feature of diffusion is that it depends on particle random walk, and results in mixing or mass transport without requiring directed bulk motion.

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 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 $\frac{\Delta C}{L}$ 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 $D = d \cdot Id$ 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** 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.

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 [4]. For this, a representative 3D model of the pore structure is needed.

Under the assumption that the Knudsen number is small ($Kn \ll 1$) and that the classical continuum mechanics approach can be used (see Knudsen number information box below), 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** 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) and (7) 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) 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, 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 ($Kn \ll 1$) no longer holds true, the diffusing molecules experience another resistance caused by additional collisions between particles and pore walls. Due to [6], the overall diffusivity can be approximated from the bulk diffusivity and the Knudsen diffusivity with the so-called Bosanquet approximation:

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

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** command.

The Knudsen diffusivity is the effective diffusivity at $Kn = \infty$, and can be determined from the result of DiffuDict's **Knudsen Diffusion** command.

DiffuDict's **Bosanquet Approximation** 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 $Kn = \infty$, 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** command.

KNUDSEN NUMBER

The **Knudsen number** is a dimensionless parameter, defined as ratio of the molecular mean free path length to a representative physical length scale.

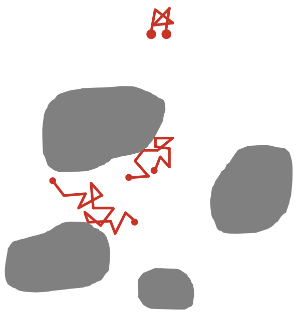
$$Kn = \frac{\text{molecular mean free path length}}{\text{representative physical length scale}} \tag{11}$$

http://en.wikipedia.org/wiki/Knudsen_number

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

Molecular mean free path is *much smaller* than a typical pore size.

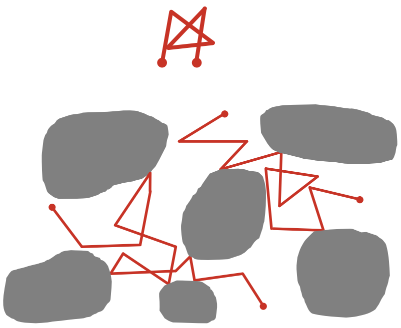
Diffusion is caused by collisions between molecules. Process is described by the Laplace equation.



Choose Bulk (Laplace) Diffusion (Kn<0.01) in DiffuDict

Molecular mean free path is *comparable* to the typical pore size.

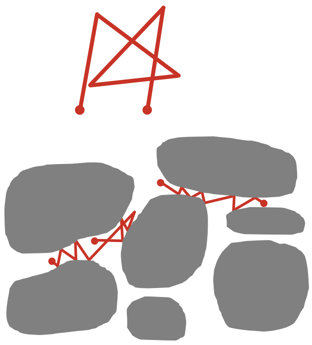
Molecule - molecule collisions and molecule - pore wall collisions happen comparably often.



Choose Molecular Diffusion (0.01<Kn<10) in DiffuDict

Molecular mean free path is *much larger* than a typical pore size.

Diffusion is caused by collisions between molecules and pore walls. Process is described by random walk.



Choose Knudsen Diffusion (10<Kn) in DiffuDict

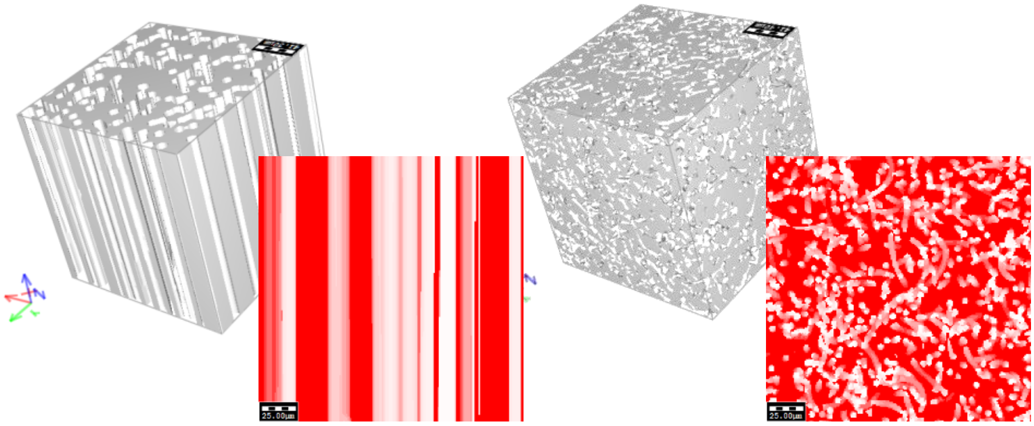
TORTUOSITY FACTOR AND GEOMETRIC TORTUOSITY

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 pore shape. The tortuosity factor in a particular direction (X, Y or Z) is defined through equation (9). 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 [3], 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 YZ-plane. Observe the effect of the shape and orientation of the pore space on the calculated diffusivity and tortuosity factor in Z-direction.



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

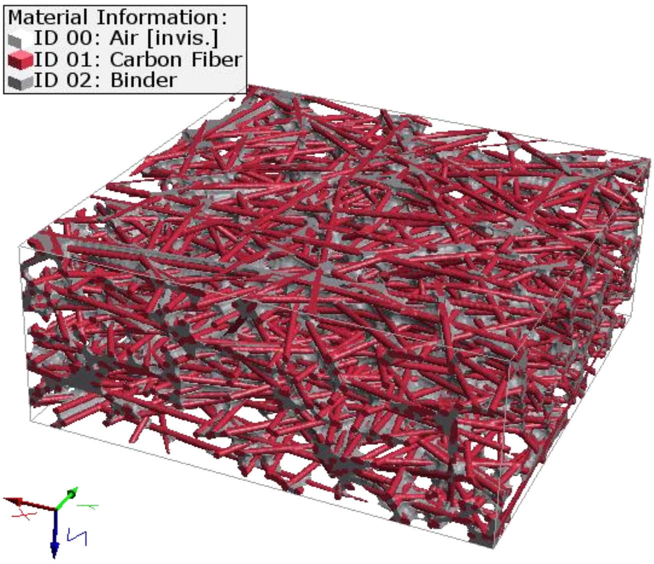
For the structure on the left, the relative diffusivity in Z-direction is very close to the porosity value (50.01%) and the tortuosity factor is close to the minimal possible value of 1.

For the structure on the right, 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.

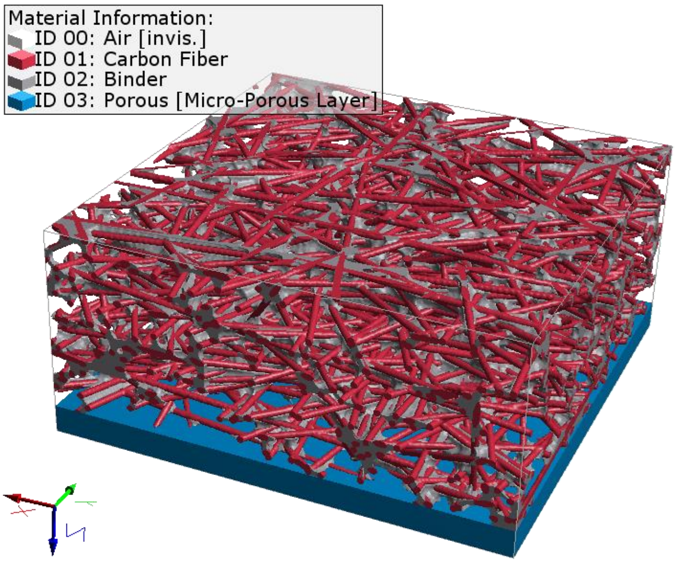
DiffuDict Computations

Material Structures for Computations with DiffuDict

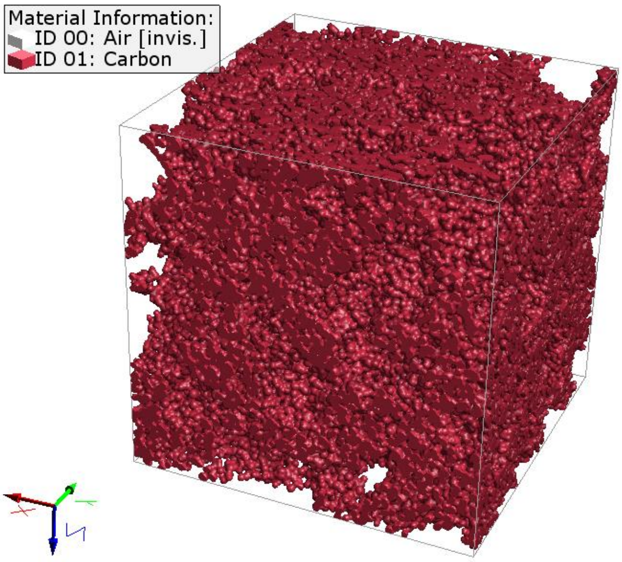
The following example structures are considered within the DiffuDict 2024 handbook. Their main differences are the relevant length scales (voxel length and pore sizes) and the presence of a porous constituent material.



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.



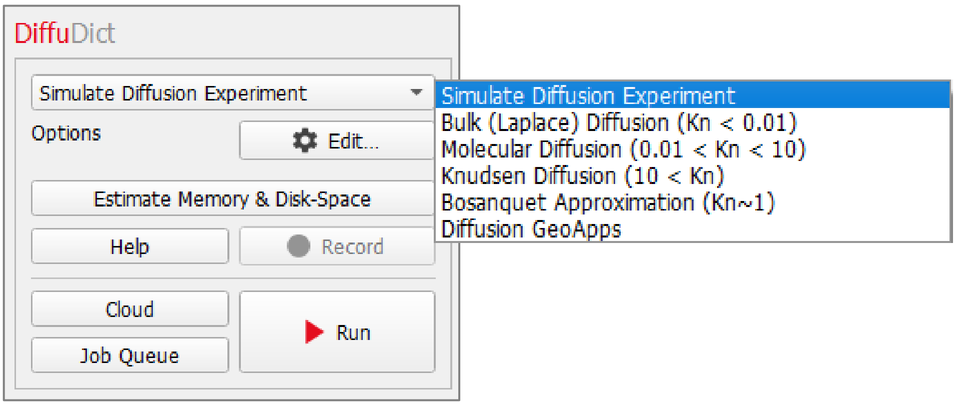
Example 2: The same gas diffusion layer model as Example 1, but with an attached, unresolved microporous layer; voxel length 1 µm, model size 500 µm x 500 µm x 230 µm.



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.

DIFFUDICT GRAPHICAL USER INTERFACE

The heading DiffuDict appears in the module section, on the left side of the GeoDict GUI, when selecting **Predict** → **DiffuDict** in the menu bar. Five commands can be selected from the pull-down menu: **Simulate Diffusion Experiment, Bulk (Laplace) Diffusion**, **Molecular Diffusion**, **Knudsen Diffusion**, and **Bosanquet Approximation**. Additionally, some **Diffusion GeoApps** are available.

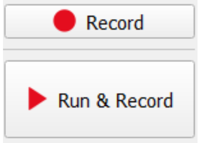


The solver options for each of the algorithms can be modified through the corresponding **Edit...** buttons.

When the options have been entered, click the **Run** button in the DiffuDict section to start the computation.

Click **Cloud** to run it in the Kaleidosim cloud, see the High Performance Computing handbook of the User Guide for details.

When recording a macro, the **Record** button becomes active and the **Run** button changes to **Run & Record**.



After clicking **Run**, the computations start but the diffusivity solver process can be stopped by clicking **Cancel** in the progress dialog box. Depending on the solver's internal processes and actual memory usage, cancelling the computation may not be instantaneous.

For **Simulate Diffusion Experiment** and **Bulk (Laplace) Diffusion**, the hardware requirements of the solver can be estimated a priori by clicking the **Estimate Memory & Disk-Space** button. The estimated computational memory requirements are shown in a **Memory & Disk-Space** message.

Memory:
Estimate 1.279 GiB
Available 15.296 GiB

Disk-Space:
Estimate 226.250 MiB
Available 28.087 GiB

(Click in the dialog to stop the countdown.)

SIMULATE DIFFUSION EXPERIMENT

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 introduction above, this is done by solving

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

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:

$$d = -jL/\Delta C \tag{13}$$

SIMULATE DIFFUSION 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** dialog. The options' settings are organized under the **Constituent Materials**, **Boundary Conditions**, **Solver**, and **Grid** tabs. The **Equations & References** tab gives some background information.

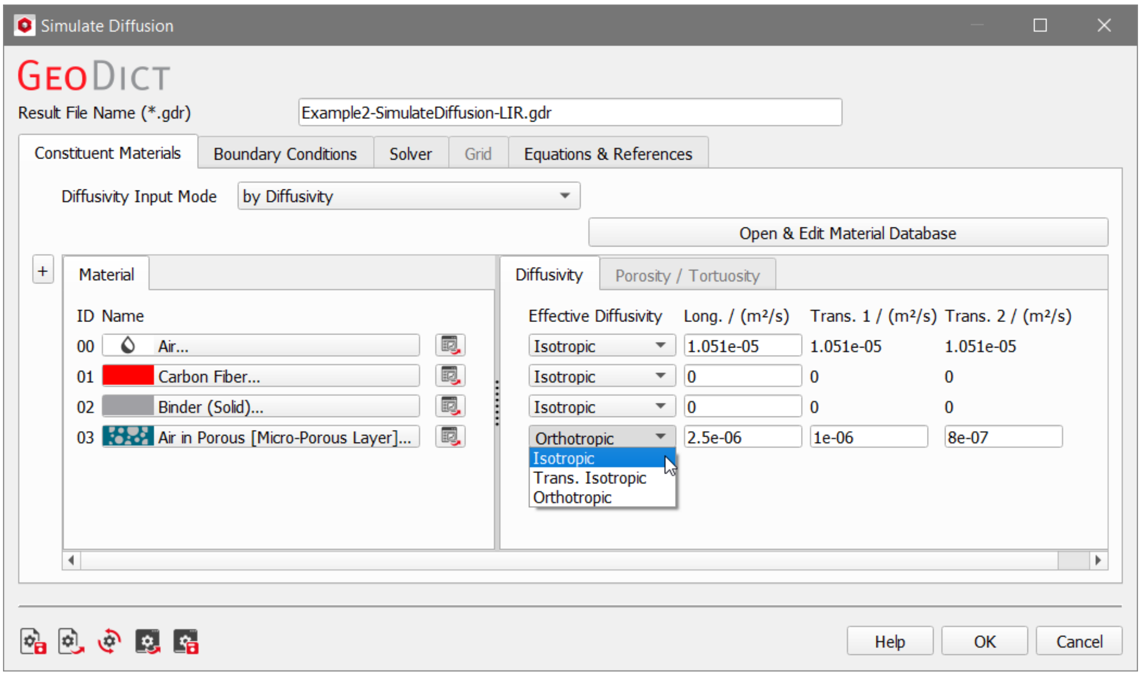
CONSTITUENT MATERIALS

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 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 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 Diffusivity** must be chosen, if the diffusion does not happen solely in the fluid phase, but also in the solid materials.

For the example on which the simulations were run here, the Diffusivity Input Mode is set to **by Diffusivity** and the diffusivity of the porous material is set to 2.5e-6 m²/s, which is the diffusivity computed for the MPL model (Example 3) with the help of Bosanquet's approximation (see section below). The corresponding diffusion coefficient in air, 1.051e-5 m^2/s is computed through equation (6) using a mean free path $\lambda = 68$ nm and a mean thermal velocity $\bar{v} = 464$ m/s.



Result File Name (*.gdr): Example2-SimulateDiffusion-LIR.gdr

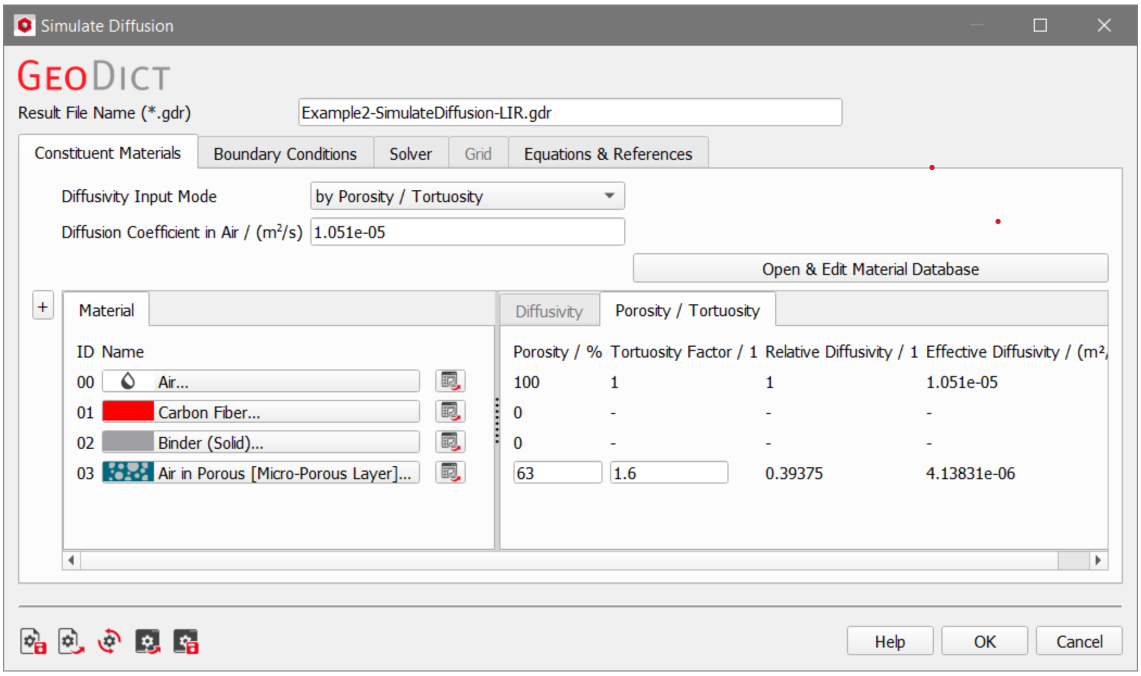
Diffusivity Input Mode: by Diffusivity

ID	Name	Effective Diffusivity	Long. / (m^2/s)	Trans. 1 / (m^2/s)	Trans. 2 / (m^2/s)
00	Air...	Isotropic	1.051e-05	1.051e-05	1.051e-05
01	Carbon Fiber...	Isotropic	0	0	0
02	Binder (Solid)...	Isotropic	0	0	0
03	Air in Porous [Micro-Porous Layer]...	Orthotropic	2.5e-06	1e-06	8e-07

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.

Orthotropic 2.5e-06 1e-06 8e-07

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.



Result File Name (*.gdr): Example2-SimulateDiffusion-LIR.gdr

Diffusivity Input Mode: by Porosity / Tortuosity

Diffusion Coefficient in Air / (m^2/s): 1.051e-05

ID	Name	Porosity / %	Tortuosity Factor / 1	Relative Diffusivity / 1	Effective Diffusivity / (m^2/
00	Air...	100	1	1	1.051e-05
01	Carbon Fiber...	0	-	-	-
02	Binder (Solid)...	0	-	-	-
03	Air in Porous [Micro-Porous Layer]...	63	1.6	0.39375	4.13831e-06

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

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

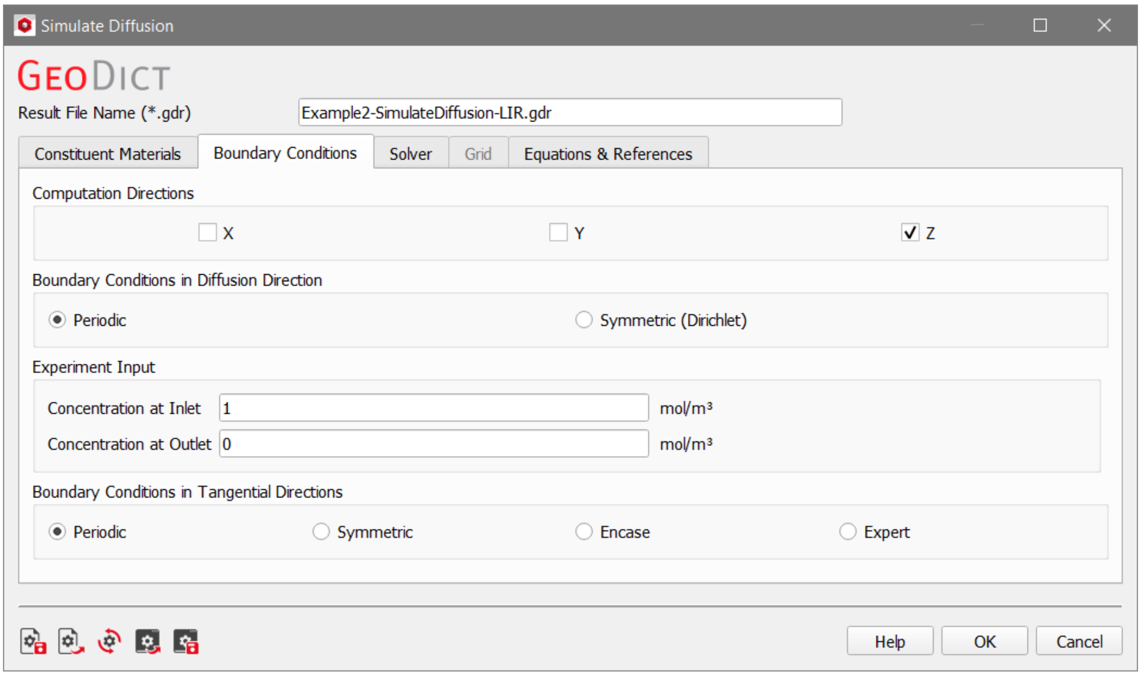
Here, d_0 denotes the entered diffusion coefficient, η the porosity and κ the tortuosity factor. For all solid materials, the diffusivity is automatically set to 0.

The default value pre-entered in the dialog, 20.86 mm^2/s is the diffusivity of oxygen in nitrogen at 25 degrees C and 1013 mbar [2].

General information about constituent materials can be found in the handbook.

BOUNDARY CONDITIONS

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



Result File Name (*.gdr): Example2-SimulateDiffusion-LIR.gdr

Computation Directions

☐ X ☐ Y ☑ Z

Boundary Conditions in Diffusion Direction

● Periodic ○ Symmetric (Dirichlet)

Experiment Input

Concentration at Inlet: 1 mol/m³

Concentration at Outlet: 0 mol/m³

Boundary Conditions in Tangential Directions

● Periodic ○ Symmetric ○ Encase ○ Expert

Computation Directions

For the **Computation 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, it is necessary to choose all three directions.

The directions that are not checked will not be calculated and will appear 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.

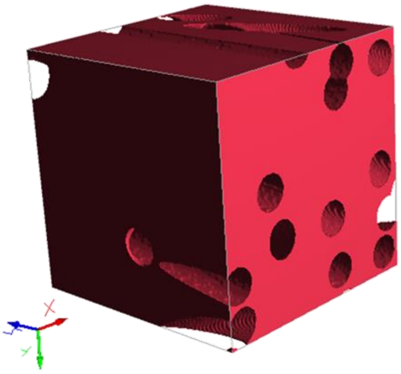
If **Periodic** is checked, DiffuDict assumes that the structure is periodically repeated in the diffusion direction. If **Symmetric (Dirichlet)** is checked, a constant concentration is ensured along both border planes in the diffusion direction. For nonperiodic structures, **Symmetric (Dirichlet)**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

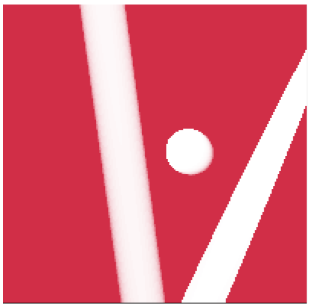
In previous versions of GeoDict, the computational memory requirements for **Symmetric (Dirichlet)** boundary conditions case were almost two times higher and therefore, the computational time was longer. This is no longer the case since GeoDict 2022.

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. In 3D, the structure has straight cylindrical pores. In 2D, in the XZ-plane, the pores appear as channels. 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.

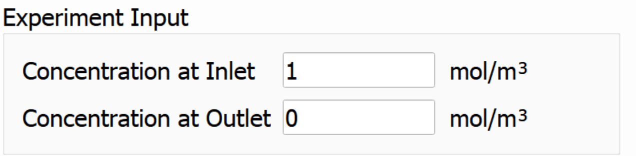
When computing the Bulk Diffusion in Z-direction, considerable differences are observed between Periodic and Dirichlet boundary conditions for this structure.





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.



Experiment Input

Concentration at Inlet	1	mol/m³
Concentration at Outlet	0	mol/m³

Boundary Conditions in Tangential Directions

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

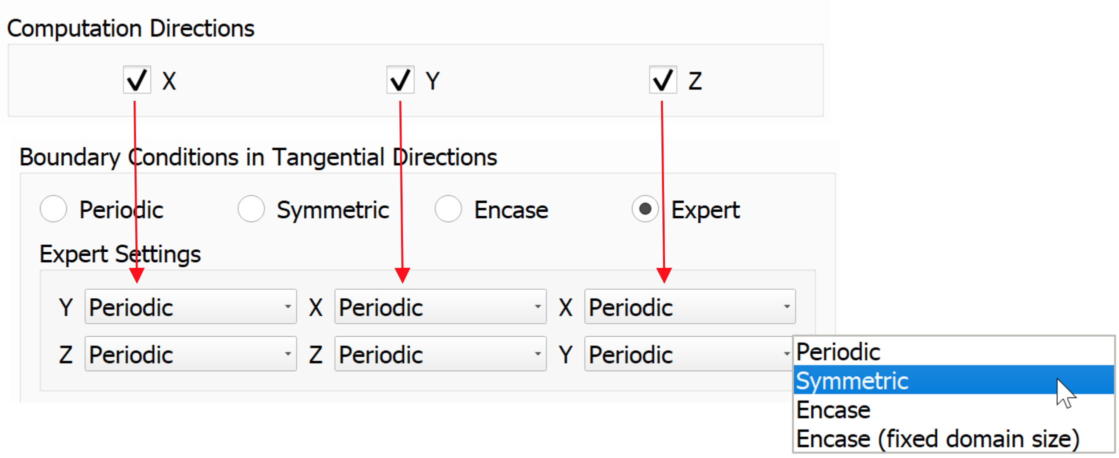
With **Periodic** boundary conditions, the structure is repeated in the tangential directions.

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:



Computation Directions

☑ X ☑ Y ☑ Z

Boundary Conditions in Tangential Directions

○ Periodic ○ Symmetric ○ Encase ● Expert

Expert Settings

Y Periodic | X Periodic | X Periodic

Z Periodic | Z Periodic | Y Periodic

When **Encase** is used, the solver internally adds a one-voxel layer in the required direction and solves with periodic boundary condition. 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 Poisson equation can be solved by two different solvers: EJ and LIR. When transverse isotropic or orthotropic thermal or electrical conductivity materials are chosen, the **LIR** must be used.

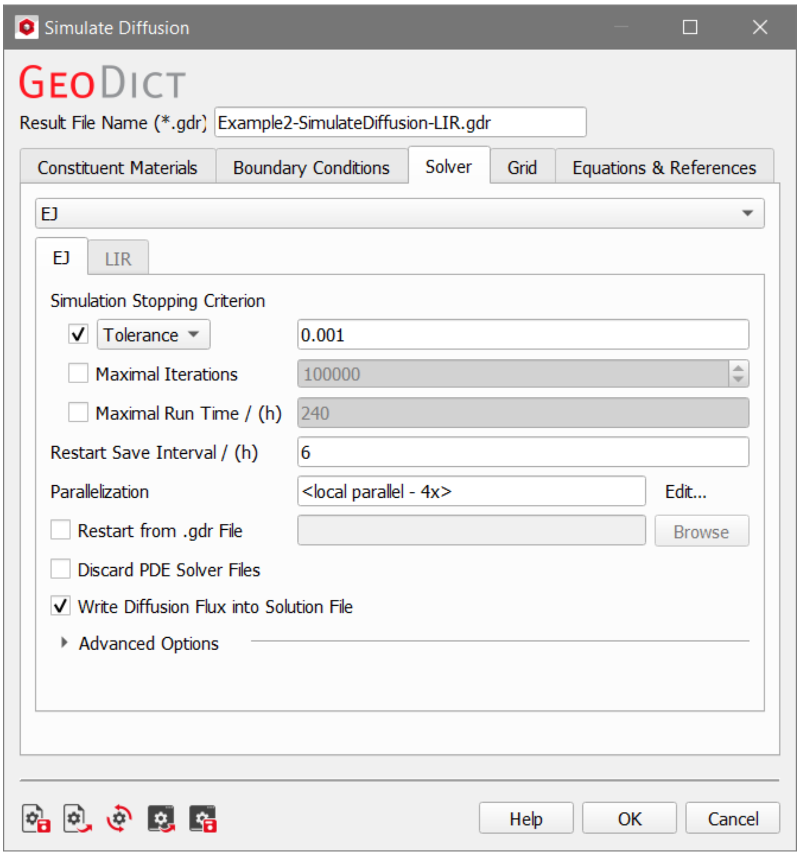
EJ Solver

Simulation Stopping Criterion

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 (**Error Bound**, **Tolerance**, **Residual**, **Maximal Iterations** or **Maximal Run Time**) is reached.

One or multiple of the stopping criteria can be checked.



Simulate Diffusion

GeoDict

Result File Name (*.gdr) Example2-SimulateDiffusion-LIR.gdr

Simulation Stopping Criterion

☑ Tolerance 0.001

☐ Maximal Iterations 100000

☐ Maximal Run Time / (h) 240

Restart Save Interval / (h) 6

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

☐ Restart from .gdr File Browse

☐ Discard PDE Solver Files

☑ Write Diffusion Flux into Solution File

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 extremely small. If the relative change is smaller than the value entered for **Tolerance**, the iteration is stopped:

$$\frac{[\text{new diffusivity} - \text{previously computed diffusivity}]}{\text{previously computed diffusivity}}$$

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.

The following possibilities might help:

- 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 an iterative solver to fail.

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

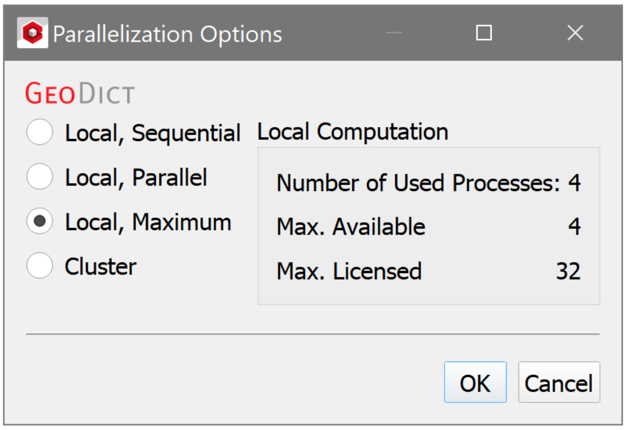
Restart Save Interval

The calculations run by the solvers can be restarted from intermediate results, 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 **Local, Sequential**, **Local, Parallel**, **Local, Maximum**, and **Cluster**.



Number of Used Processes: 4

Max. Available 4

Max. Licensed 32

For details on how to set up und run parallel computations, consult the [High Performance Computations]() handbook of the User Guide.

Restart from *.gdr File

In some situations, it may be useful to re-use previously computed results and, thus, reduce the runtime of the diffusivity computation. Typical examples would be when trying another boundary condition (Dirichlet instead of Periodic or other way around) or non-sufficient accuracy of some computation, when it is suspected that more iterations may improve the quality of the result.

To use some previously computed result, **Restart from .gdr File** can be checked and **Browse** used to search for the file.

Note that the structure used for restarting for both the current and the restart result file, must be the same. If this is not the case, an error message is displayed.

Discard PDE Solver Files.

Checking the **Discard PDE Solver Files** box causes the deletion of all intermediate computation files. While having the benefit of saving storage place, discarding solver files has also the side effect of disabling the 3D visualization of the results.

Of course, the contents of the result file (*.gdr) are not discarded even in this case.

Write Diffusion Flux into Solution File

With **Write Diffusion Flux into Solution File**, the flux in the three coordinate directions is saved, allowing a detailed analysis of the flux field. The size of the result files increases when selecting this option.

Advanced Options: Analyze Geometry

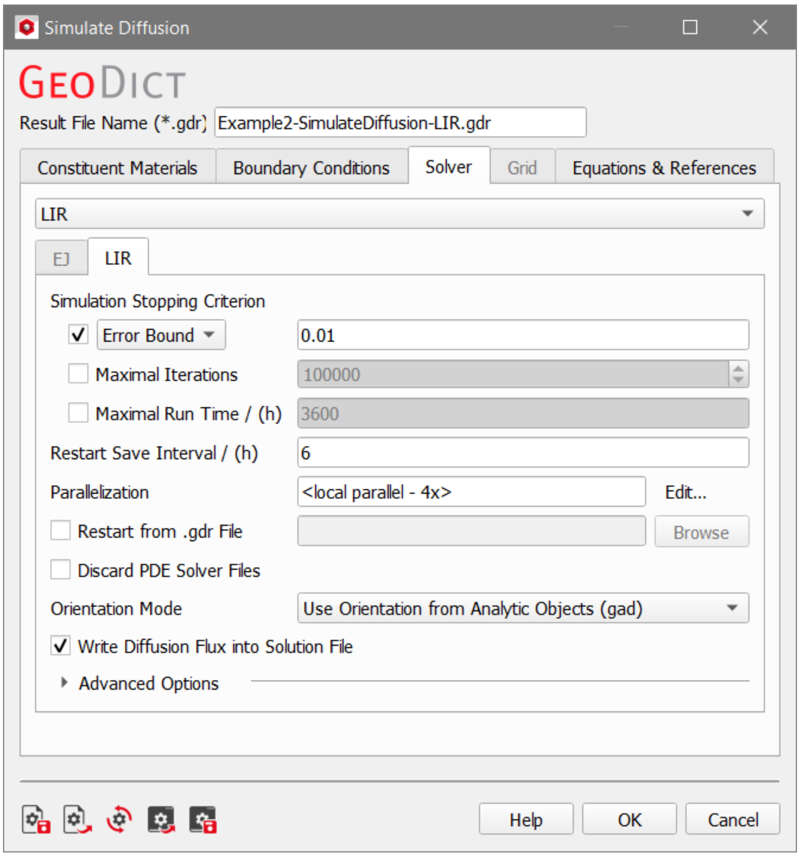
☑ Analyze Geometry

Expand **Advanced Options** to see **Analyze Geometry** for EJ. If **Analyze Geometry** is checked, GeoDict checks if the structure contains a through path before starting the computation. If no path for the diffusing species 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. For LIR, there are more advanced options.

LIR Solver

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.



Simulate Diffusion

GeoDict

Result File Name (*.gdr) Example2-SimulateDiffusion-LIR.gdr

LIR

Simulation Stopping Criterion

☑ Error Bound 0.01

☐ Maximal Iterations 100000

☐ Maximal Run Time / (h) 3600

Restart Save Interval / (h) 6

Parallelization <local parallel - 4x>

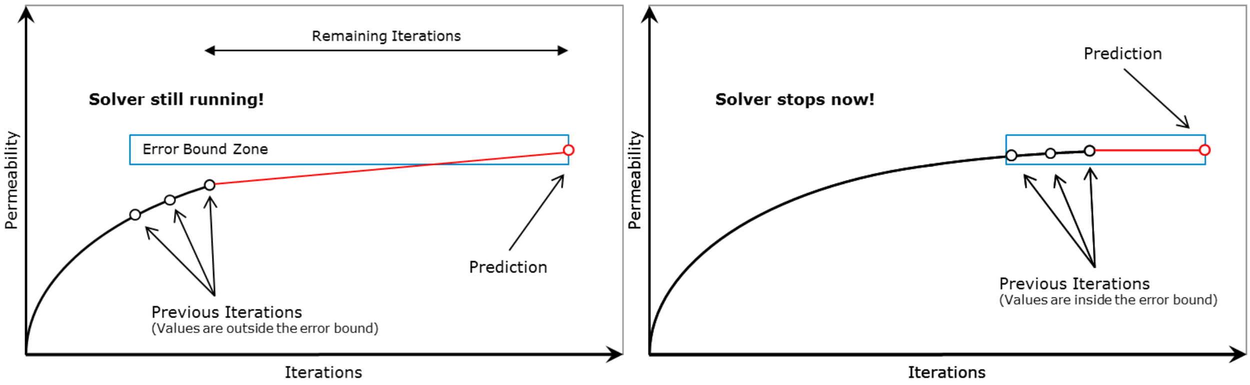
☐ Restart from .gdr File

☐ Discard PDE Solver Files

Orientation Mode Use Orientation from Analytic Objects (gad)

☑ Write Diffusion Flux into Solution File

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.

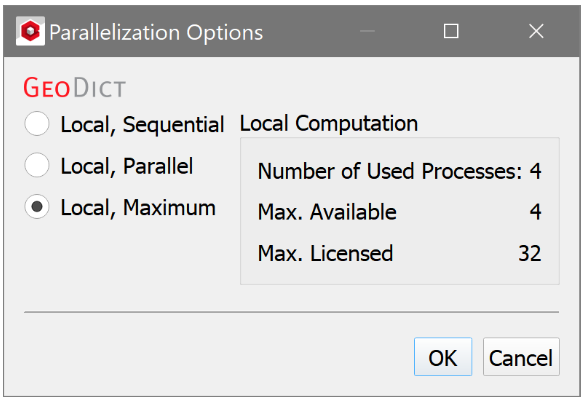
Restart Save Interval, Restart from .gdr file, Discard PDE Solver Files, Analyze Geometry and Write Diffusion Flux into Solution File

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

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 **Local, Sequential**, **Local, Parallel**, **Local, Maximum**.



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

Local Computation

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

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

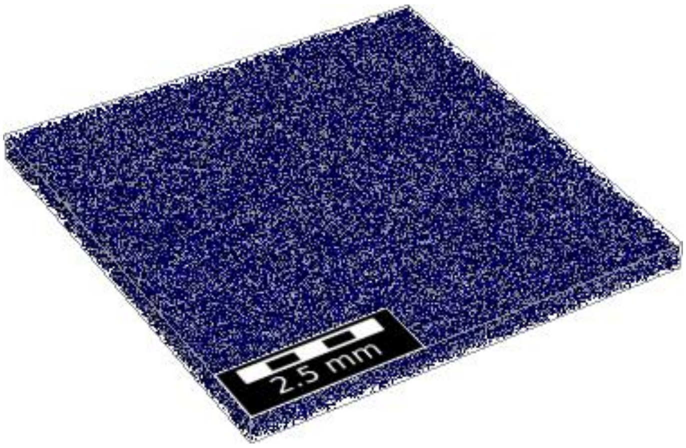
Parallelization Benchmark Results

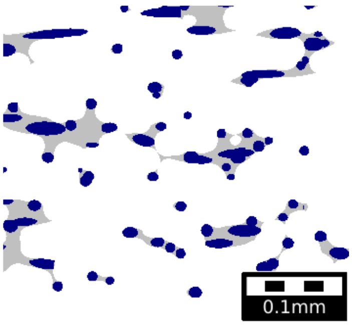
As an example for the parallelization, the diffusivity in through-direction in a Gas Diffusion Layer (GDL) is computed with different number of processes.

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.

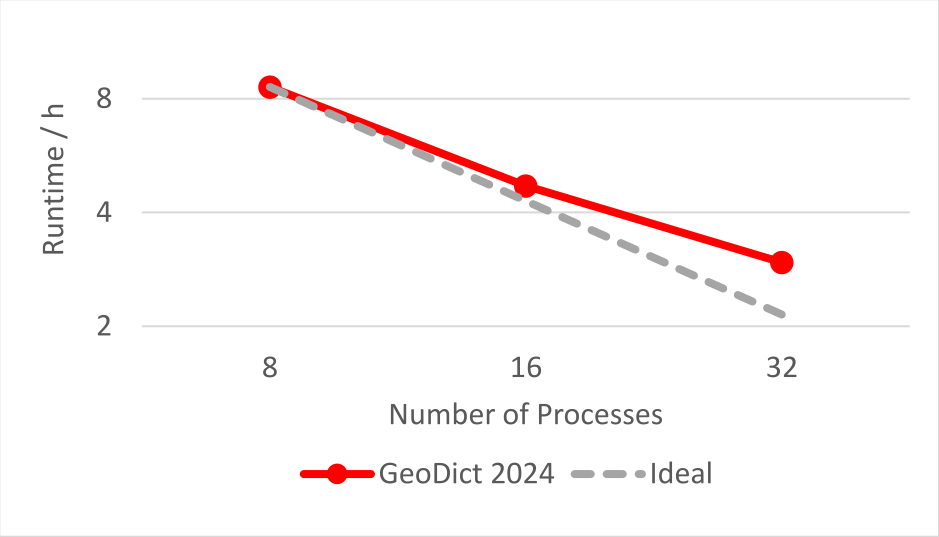
The GPL structure has size 8,192 x 8,192 x 512 voxels and is created in GeoDict with the GeoApp Gas Diffusion Layer, available in FiberGeo.

The computation for the structure with over 34 billion voxels, requires 330 GB of RAM.





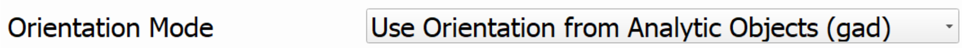
Runtimes for different number of parallel processes are shown on the following figure. Even with 8 processes, the simulation for the huge structure is possible in 8.6 h. Due to the very good speed-up for increasing number of processes, the computation with 32 processes is possible in 3 h.



Orientation Mode

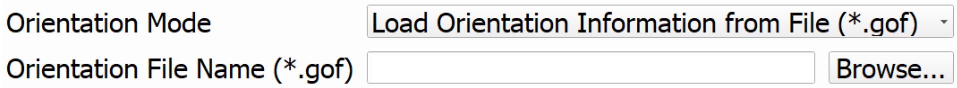
If one of the constituent materials has a transverse isotropic or orthotropic material law, a local orientation is needed to compute the diffusion. There are three different choices how to determine the local orientation.

The standard case is to use the orientation defined by the local orientation of the GAD objects, e.g. the direction of a fiber.



Orientation Mode: 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 image, 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:



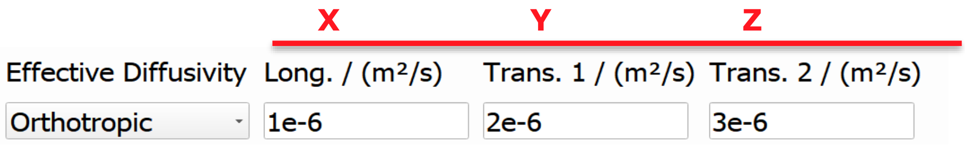
Orientation Mode: Load Orientation Information from File (*.gof)

Orientation File Name (*.gof):

Last, one can simply use the coordinate system:

Orientation Mode: Use the Global XYZ-Coordinate System

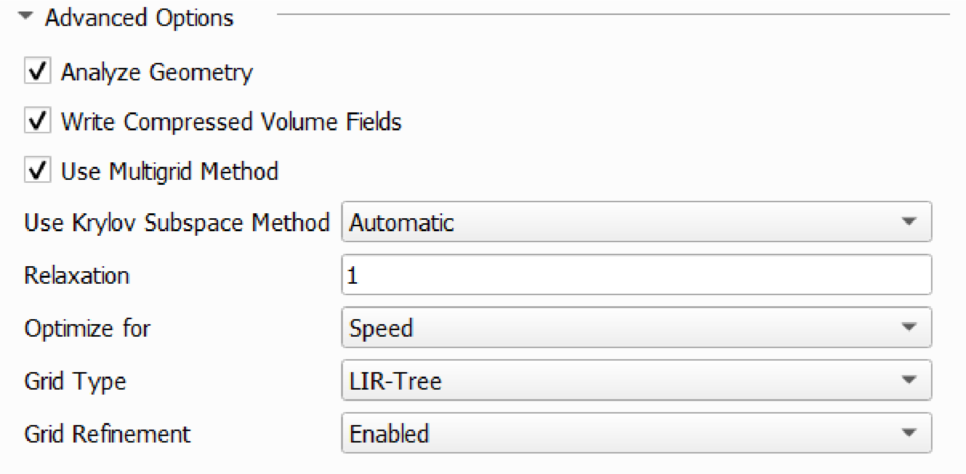
In this case, the entered diffusivities are the diffusivities in the X, Y, Z space directions:



Effective Diffusivity	Long. / (m²/s)	Trans. 1 / (m²/s)	Trans. 2 / (m²/s)
Orthotropic	1e-6	2e-6	3e-6

Advanced Options

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



Advanced Options

☑ Analyze Geometry

☑ Write Compressed Volume Fields

☑ Use Multigrid Method

Use Krylov Subspace Method: Automatic

Relaxation: 1

Optimize for: Speed

Grid Type: LIR-Tree

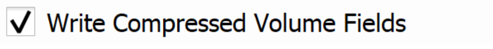
Grid Refinement: Enabled

Analyze Geometry

If **Analyze Geometry** is checked, GeoDict checks if the structure contains a through path before starting the computation. If no path for the diffusing species 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.

Write Compressed Volume Fields

If the option **Write Compressed Volume Fields** is checked for LIR solver then the adaptive grid structure is used as compression method for writing out HHT files. This option allows to save 80-90% space on hard drive.



☑ Write Compressed Volume Fields

The runtime for writing HHT files is also reduced significantly. If the option **Write Compressed Volume Fields** is not checked then a usual regular grid is used for writing out HHT files.

Use Multigrid Method

The **Multigrid** method [9] 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 for solving the Stokes and Stokes-Brinkman equations in FlowDict as well as for solving diffusion, thermal and electrical conduction in DiffuDict and ConductoDict and is enabled by default.

☑ Use Multigrid Method

Use Krylov Subspace Method

Depending on the structure and the corresponding material parameters, a significant speedup of the LIR can be achieved by using the BiCGstab method to compute the solution. Using the BiCGstab method approximately doubles the amount of RAM needed for the computation.

Use Krylov Subspace Method Automatic

When **Use Krylov Subspace Method** is set to **Automatic**, GeoDict decides based on structure, material parameters and boundary condition which method is expected to be faster and uses this method. In case that the Krylov subspace method (BICGstab) is used, the **Relaxation** is also chosen automatically.

Alternatively, the user may also explicitly enable or disable this method.

If the ratio of the largest and smallest diffusivity within the structure (i.e. high contrast) is large (approx. $>10^6$), usage of the Krylov method is recommended.

For structures without a high diffusivity contrast, the usage of this option is not recommended.

Relaxation

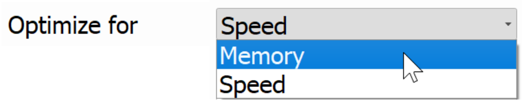
Depending on the material parameters and geometry of the structure, the underlying mathematical problem can vary in complexity, thus influencing the behavior of the solver. The iterative method uses the relaxation number to adjust it from **Stable** (with smaller number chosen, which results in higher number of iterations, slower time stepping, and longer solver run times), to **Fast** with higher number chosen, which makes the solver run less iterations but implies the risk that the solver does not converge.

Relaxation 1.5

For the LIR solver, this balance is managed through the **Relaxation**. The value should be between 0 and 2. For relaxation values smaller than one (<1.0), the simulation is more stable. For relaxation values larger than one (>1.0), the simulation is faster.

Optimize for

The **LIR** solver can **Optimize for** speed or memory.

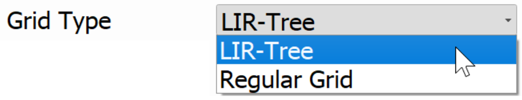


Optimize for Speed

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, then the runtime is increased by up to 40% but the solver requires up to 50% less memory.

Grid Type

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



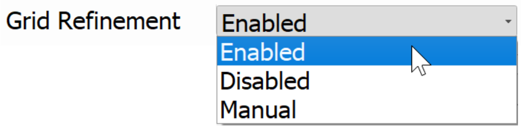
Grid Type LIR-Tree

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

Grid Refinement

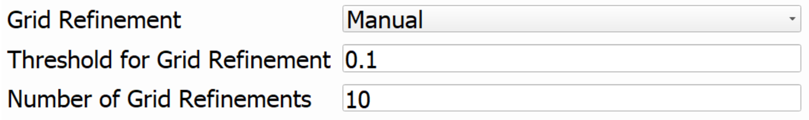
The solver can analyze the velocity and pressure field during the computation and improves the adaptive grid in places where more accuracy is needed. The LIR solver splits cells where a high velocity-gradient or high pressure-gradient occurs. The analysis is enabled if the **Grid Refinement** option is set to **Enabled** or **Manual**.

If the **Grid Refinement** is set to **Enabled**, the solver chooses the **Number of Grid Refinements** and **Threshold for Grid Refinement** automatically.



Grid Refinement: Enabled

If the **Grid Refinement** is set to **Manual**, the user can enter the parameters manually.



Grid Refinement: Manual

Threshold for Grid Refinement: 0.1

Number of Grid Refinements: 10

The **Number of Grid Refinement** controls how many velocity-based and pressure-based grid refinements are allowed during the simulation. The value should be between 0 and 10. Velocity-based and pressure-based grid refinements may increase the number of iterations, runtime and memory requirements.

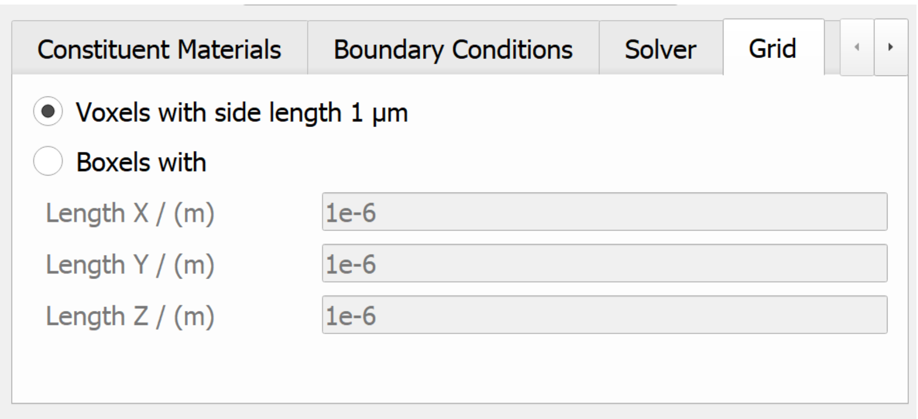
The **Number of Grid Refinement** can be zero in most of the cases and should be greater than zero if a flow simulation is done on a structure with a very long inlet and outlet, for pleated filter structures, or for Navier-Stokes simulations.

Refinement is done at regions with high-velocity gradient or high-pressure gradient. Cells are split where the current velocity gradient (or pressure gradient) is greater than the **Threshold for Grid Refinements** times the maximal velocity gradient (or pressure gradient). This threshold must be between 0.0 and 1.0. The recommended value range is between 0.05 and 0.1.

GRID (FOR EJ)

The choices under the **Grid** tab are only possible if **EJ** was chosen as the solver.

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



● Voxels with side length 1 µm

○ Boxels with

Length X / (m): 1e-6

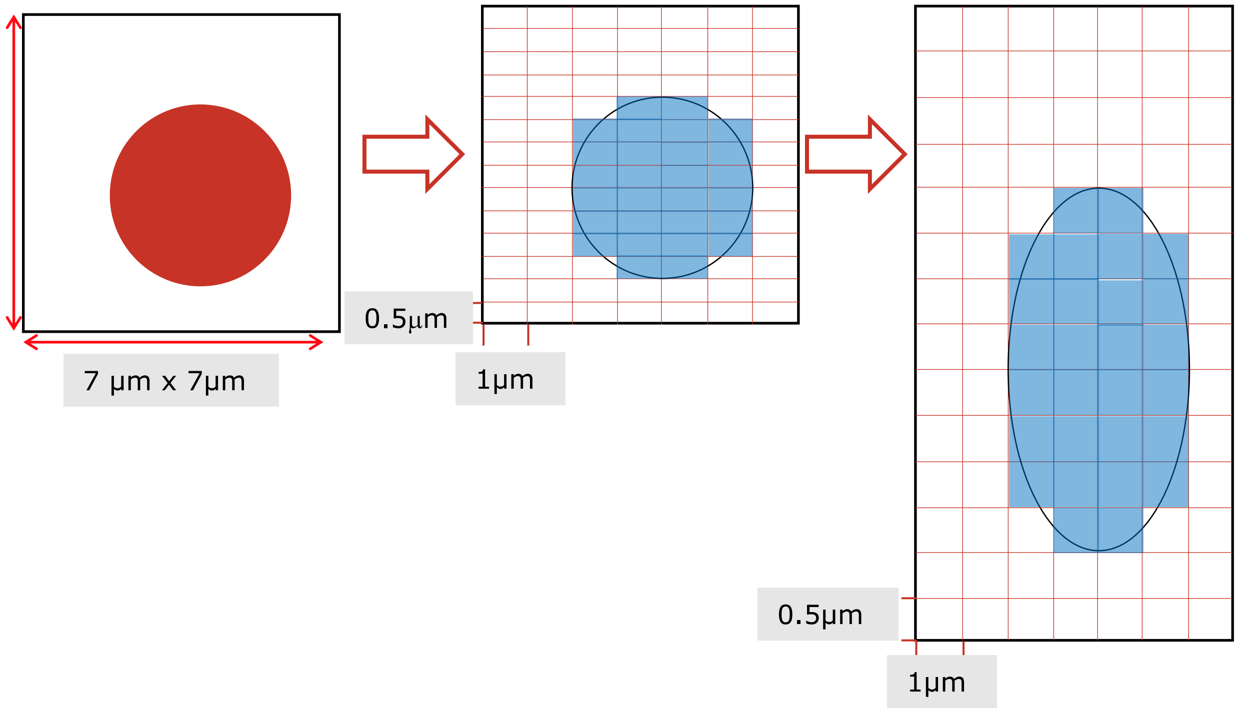
Length Y / (m): 1e-6

Length Z / (m): 1e-6

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.



EQUATIONS & REFERENCES

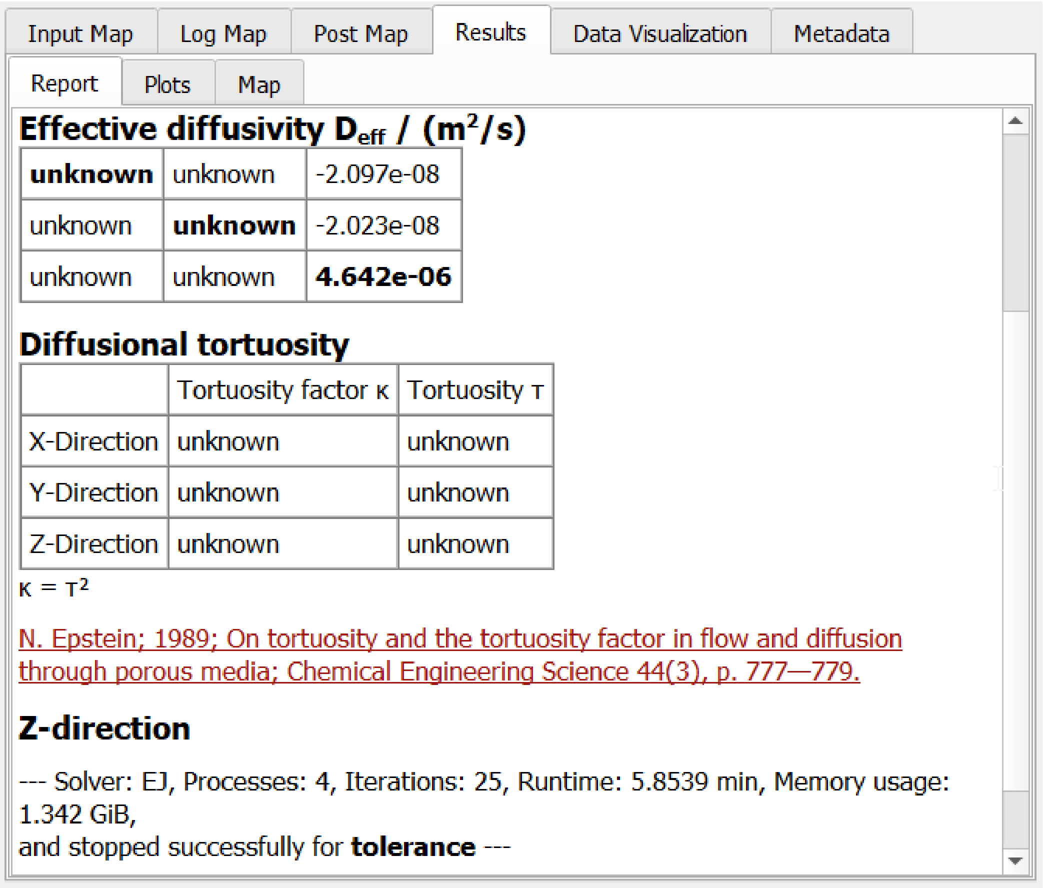
This tab shows the equations (2) and (3) of this DiffuDict handbook for illustration and explains the used variables.

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 EJ solver for the GDL with an attached, unresolved MPL (Example 2):

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



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

unknown	unknown	-2.097e-08
unknown	**unknown**	-2.023e-08
unknown	unknown	**4.642e-06**

Diffusional tortuosity

	Tortuosity factor κ	Tortuosity τ
X-Direction	unknown	unknown
Y-Direction	unknown	unknown
Z-Direction	unknown	unknown

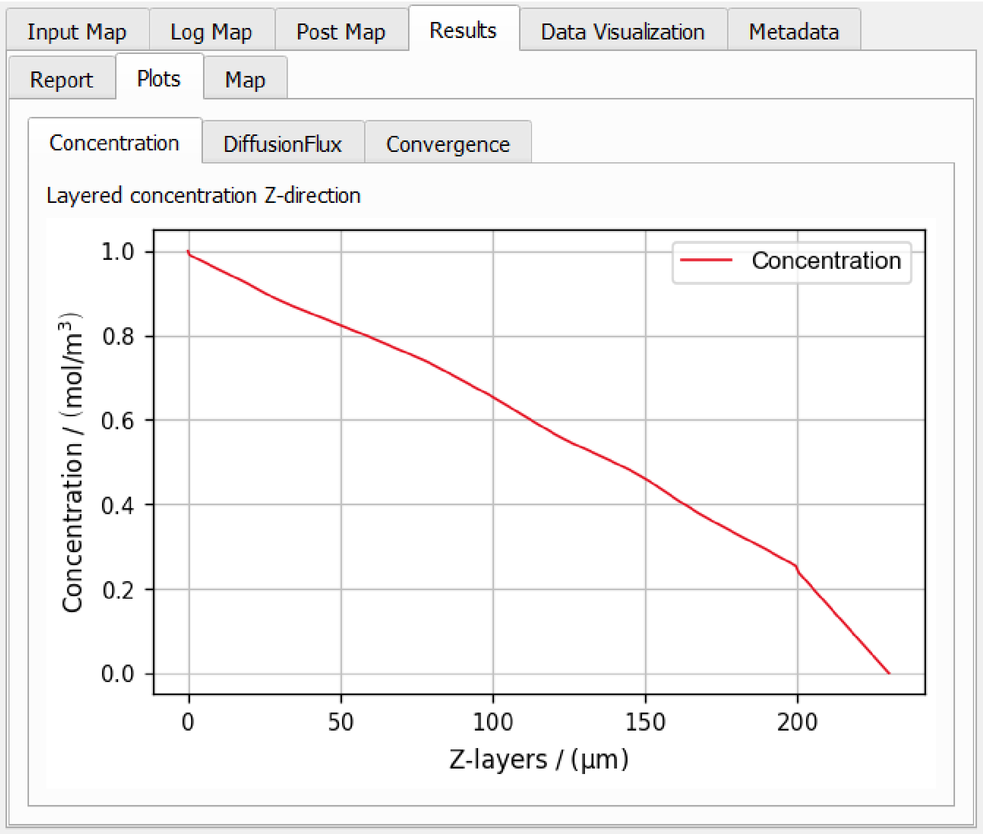
$\kappa = \tau^2$

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

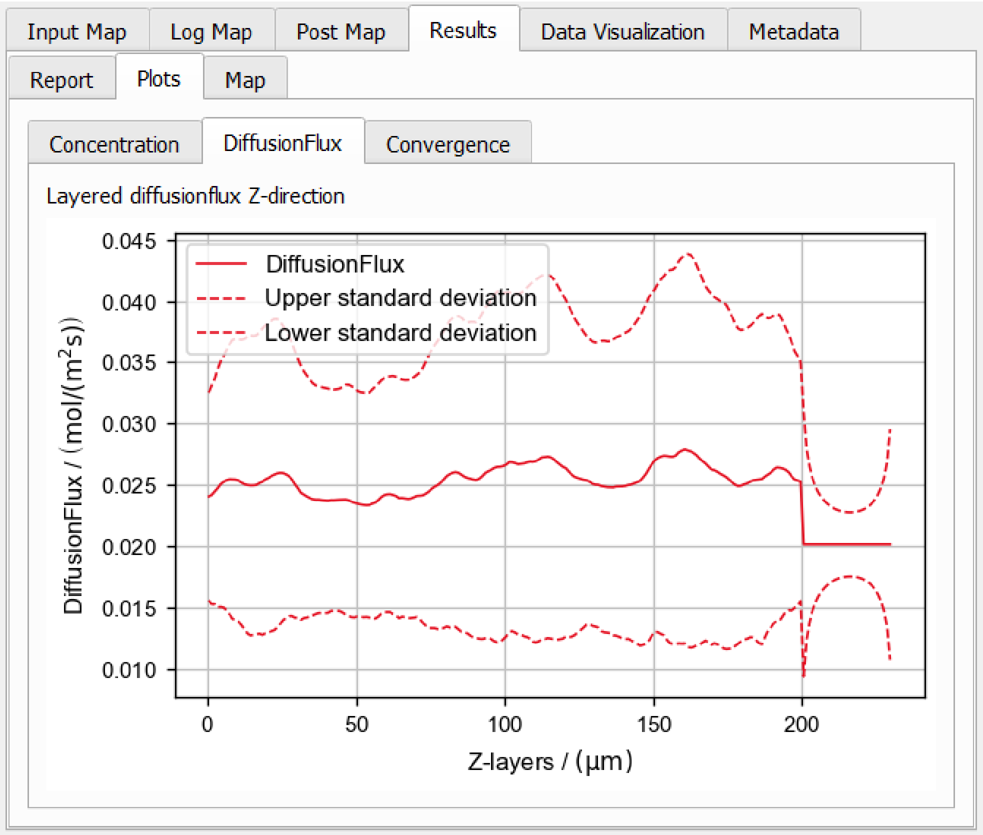
Z-direction

--- Solver: EJ, Processes: 4, Iterations: 25, Runtime: 5.8539 min, Memory usage: 1.342 GiB,
and stopped successfully for **tolerance** ---

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 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.

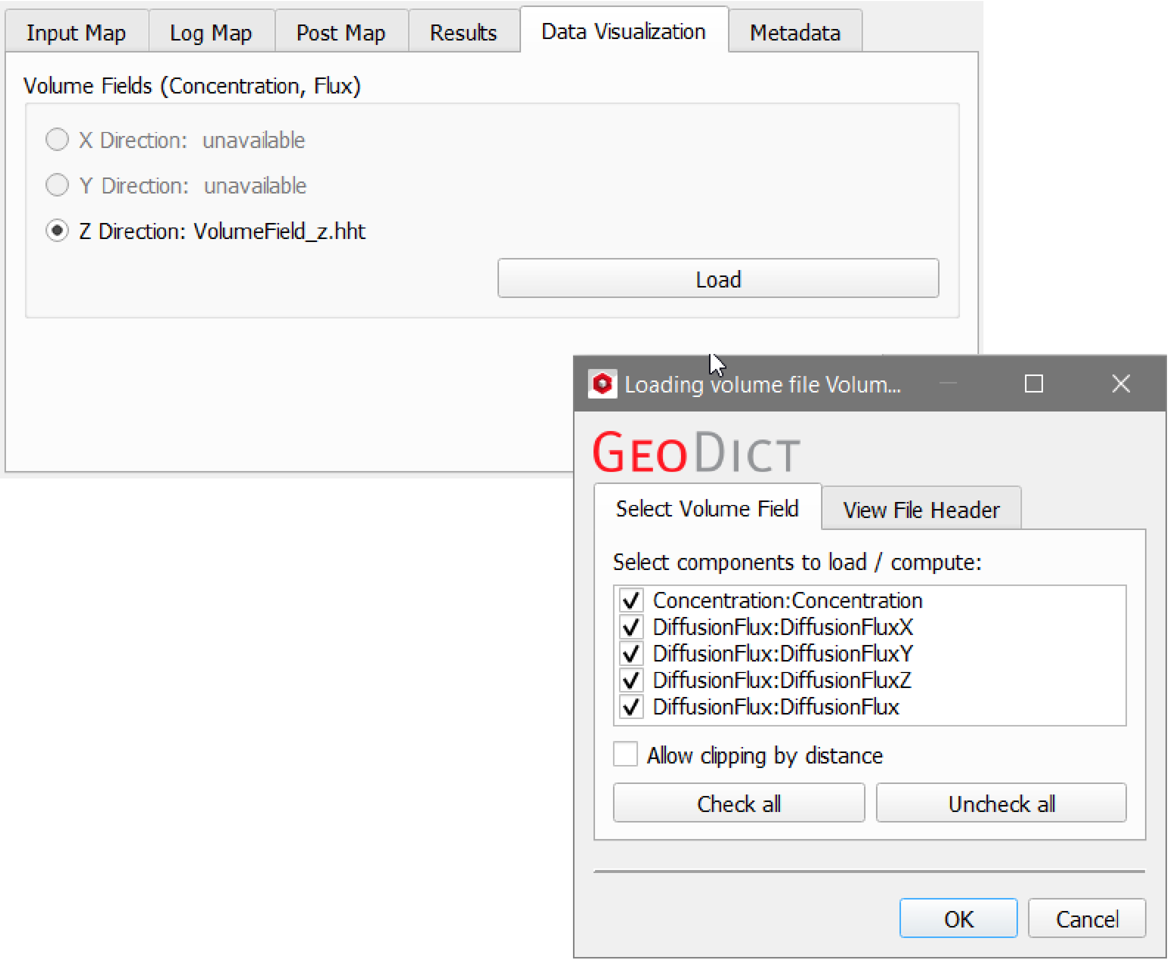


DATA VISUALIZATION

Under the **Data Visualization** tab, the result of the diffusivity calculations can be loaded and graphically visualized.

Computation Directions that were not previously selected (page 11) 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.



Volume Fields (Concentration, Flux)

○ X Direction: unavailable
○ Y Direction: unavailable
● Z Direction: VolumeField_z.hht

Select components to load / compute:

☑ Concentration:Concentration
☑ DiffusionFlux:DiffusionFluxX
☑ DiffusionFlux:DiffusionFluxY
☑ DiffusionFlux:DiffusionFluxZ
☑ DiffusionFlux:DiffusionFlux

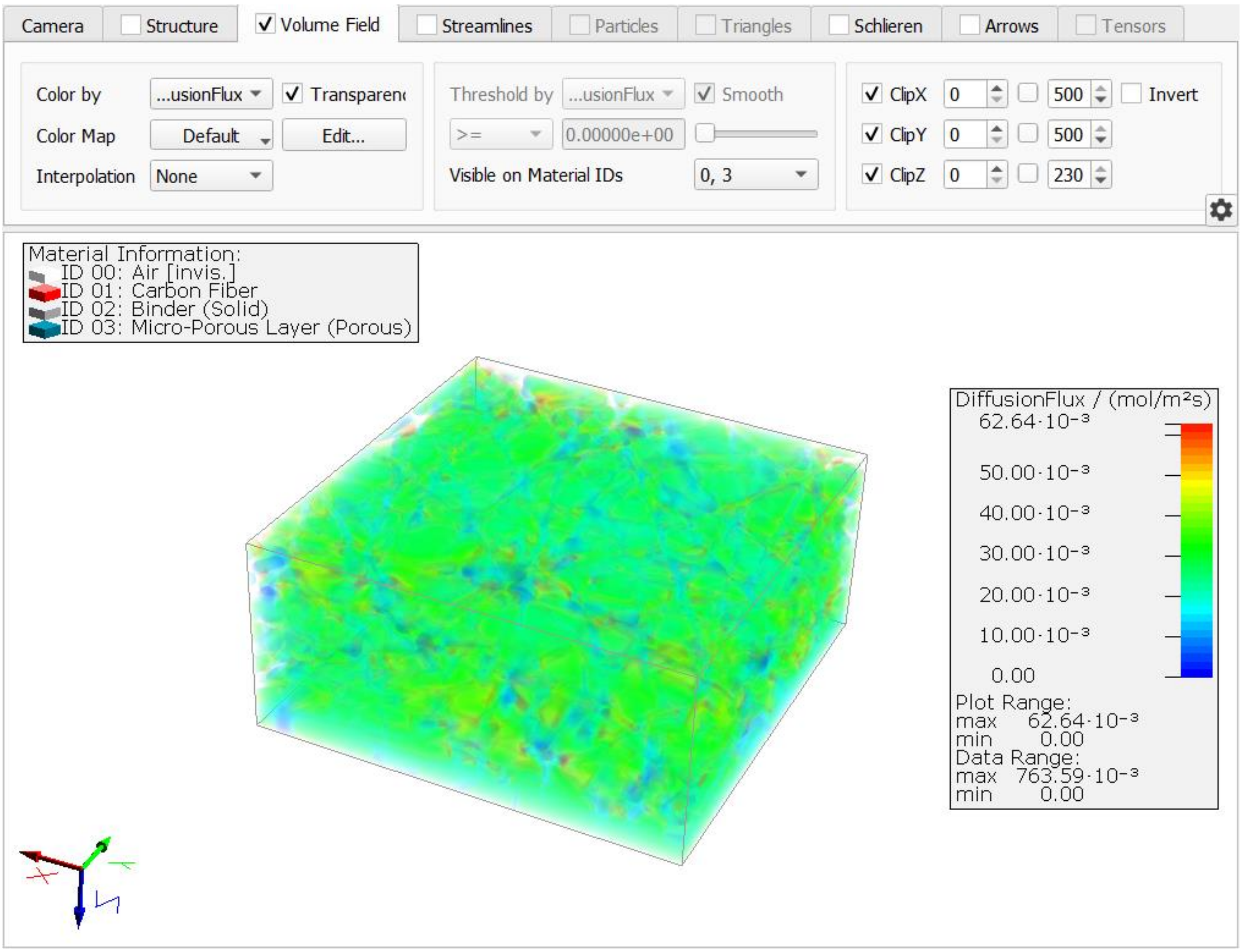
☐ Allow clipping by distance

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

Different from concentrations and fluxes computed with the **Bulk (Laplace) Diffusion** command, the concentrations and fluxes have a physical unit here: mol/m³ for concentrations and mol/m²/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.



Color by ...usionFlux ☑ Transparen

Color Map Default

Interpolation None

Threshold by ...usionFlux ☑ Smooth

>= 0.00000e+00

Visible on Material IDs 0, 3

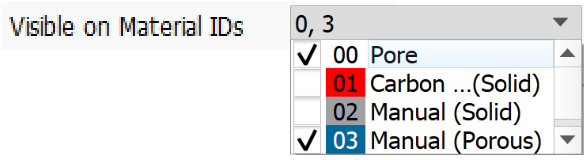
☑ ClipX 0 ☐ 500 ☐ Invert

☑ ClipY 0 ☐ 500

☑ ClipZ 0 ☐ 230

Plot Range:
max 62.64·10⁻³
min 0.00
Data Range:
max 763.59·10⁻³
min 0.00

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.



Visible on Material IDs 0, 3

☑ 00 Pore
☐ 01 Carbon ...(Solid)
☐ 02 Manual (Solid)
☑ 03 Manual (Porous)

The visualization options are explained in detail in the Visualization handbook of this User Guide.

BULK (LAPLACE) DIFFUSION

THEORETICAL BACKGROUND

This command solves the Laplace equation (8)

$$\Delta c = 0 \tag{15}$$

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 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.

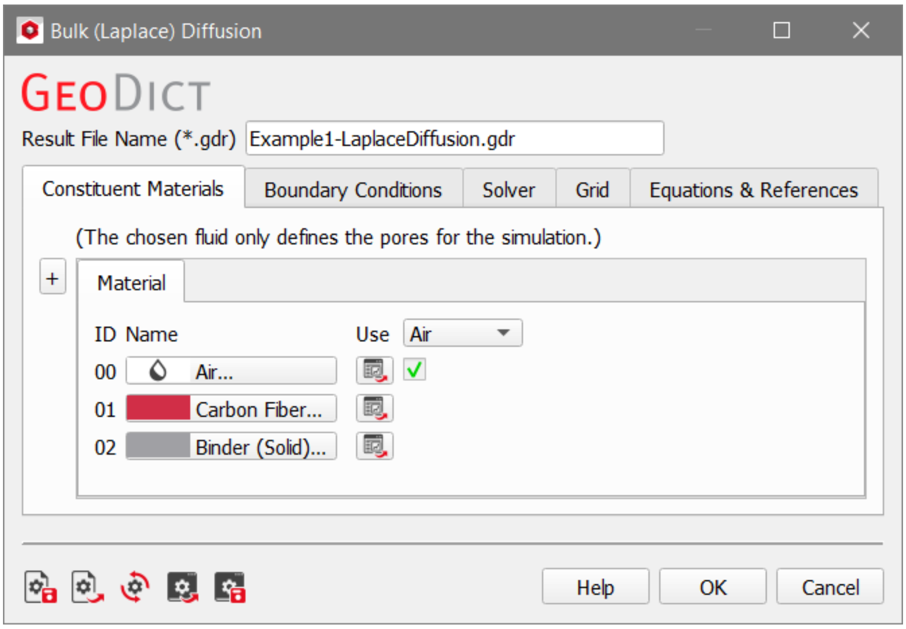
Two finite volume-based solvers are available to solve this equation: EJ [10] and LIR [5]. Both solvers use harmonic averaging to compute conductivities at voxel faces.

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.

BULK (LAPLACE) DIFFUSION OPTIONS

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



Result File Name (*.gdr) Example1-LaplaceDiffusion.gdr

(The chosen fluid only defines the pores for the simulation.)

ID	Name	Use Air
00	Air...	✓
01	Carbon Fiber...	
02	Binder (Solid)...	

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

CONSTITUENT MATERIALS

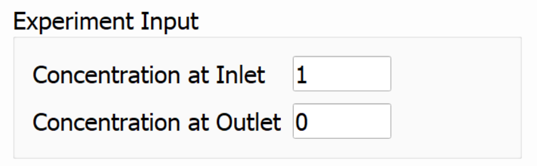
Under the constituent materials tab, the user sets the material IDs corresponding to the pore space and those that denote solids. All material IDs marked with a green checkmark ✓ are open for diffusion. All other material IDs are treated as solids and no diffusion takes place there.

As described above, 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 defined for the **Simulate Diffusion Experiment** command and are described in detail in that section (page 11).

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 above, a dimensionless equation is solved, and therefore the concentration given here has no physical unit.



Experiment Input

Concentration at Inlet 1

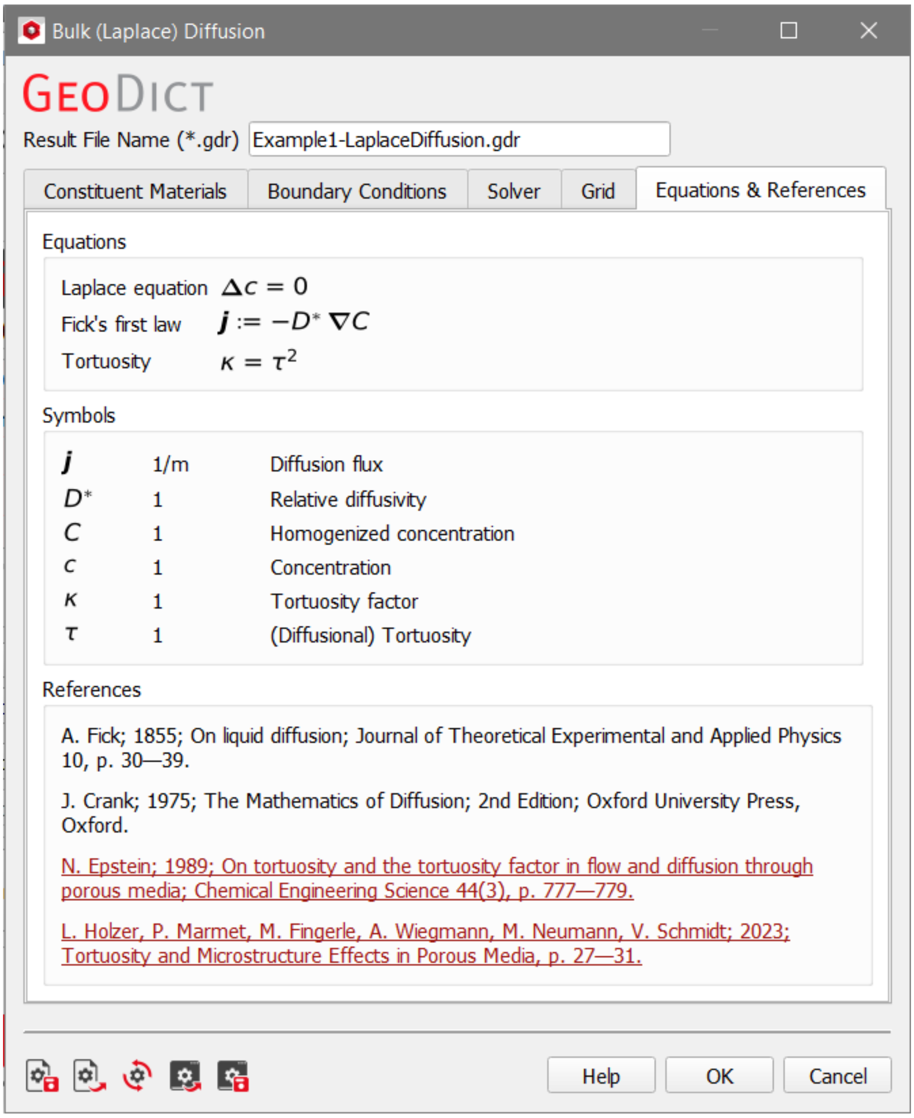
Concentration at Outlet 0

SOLVER AND GRID

The options in these tabs are the same as defined for the **Simulate Diffusion Experiment** command and are described in detail in that section (page 14).

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.



Result File Name (*.gdr) Example1-LaplaceDiffusion.gdr

Equations

Laplace equation $\Delta c = 0$

Fick's first law $\boldsymbol{j} := -D^* \nabla C$

Tortuosity $\kappa = \tau^2$

Symbols

$\boldsymbol{j}$	1/m	Diffusion flux
D^*	1	Relative diffusivity
C	1	Homogenized concentration
c	1	Concentration
κ	1	Tortuosity factor
τ	1	(Diffusional) Tortuosity

References

A. Fick; 1855; On liquid diffusion; Journal of Theoretical Experimental and Applied Physics 10, p. 30—39.

J. Crank; 1975; The Mathematics of Diffusion; 2nd Edition; Oxford University Press, Oxford.

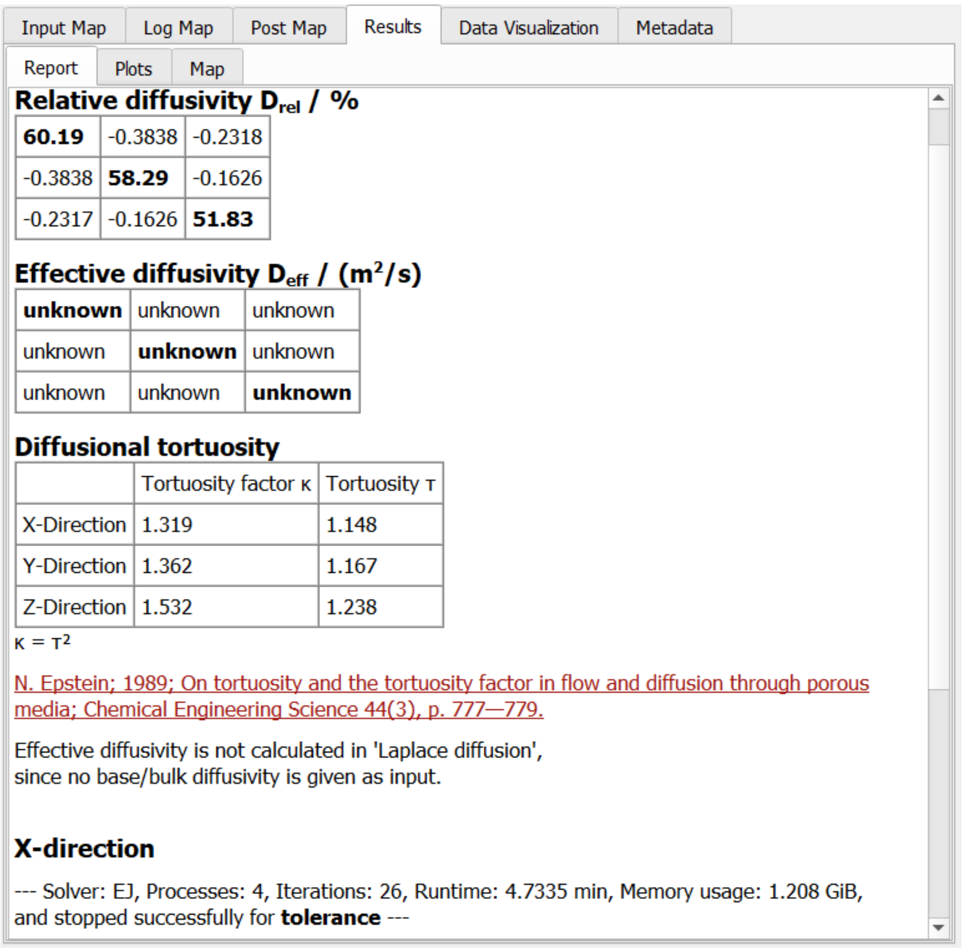
N. Epstein; 1989; On tortuosity and the tortuosity factor in flow and diffusion through porous media; Chemical Engineering Science 44(3), p. 777—779.

L. Holzer, P. Marmet, M. Fingerle, A. Wiegmann, M. Neumann, V. Schmidt; 2023; Tortuosity and Microstructure Effects in Porous Media, p. 27—31.

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). Here, the **Results - Report** subtab shows:

- The relative diffusivity D^* (here called $\mathrm{D}_{\mathrm{rel}}$), as defined through equation (7) in %.
- The tortuosity factor κ as defined through equation (9)
- The geometric tortuosity $\tau = \sqrt{\kappa}$
- The runtime needed to solve equation (8) in each space direction and the stopping criterion that was applied.



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.

X-direction

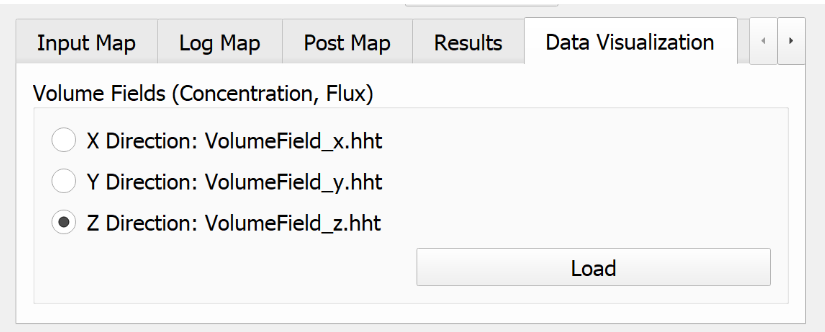
--- Solver: EJ, Processes: 4, Iterations: 26, Runtime: 4.7335 min, Memory usage: 1.208 GiB, and stopped successfully for **tolerance** ---

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 that were not previously selected (page 11) were not computed, and, thus, are unavailable for visualization.

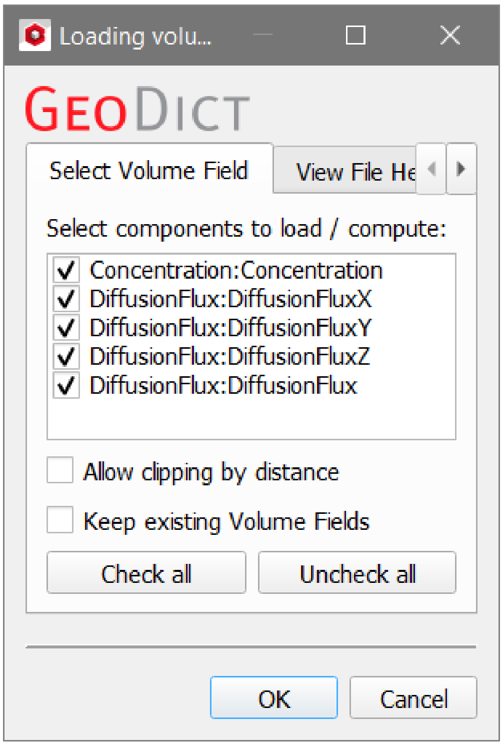
The .hht files contain the local concentration in every voxel. If **Write Diffusion Flux into Solution File** was checked, the result files contain the diffusion flux, too.



Volume Fields (Concentration, Flux)

○ X Direction: VolumeField_x.hht
○ Y Direction: VolumeField_y.hht
● Z Direction: VolumeField_z.hht

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:

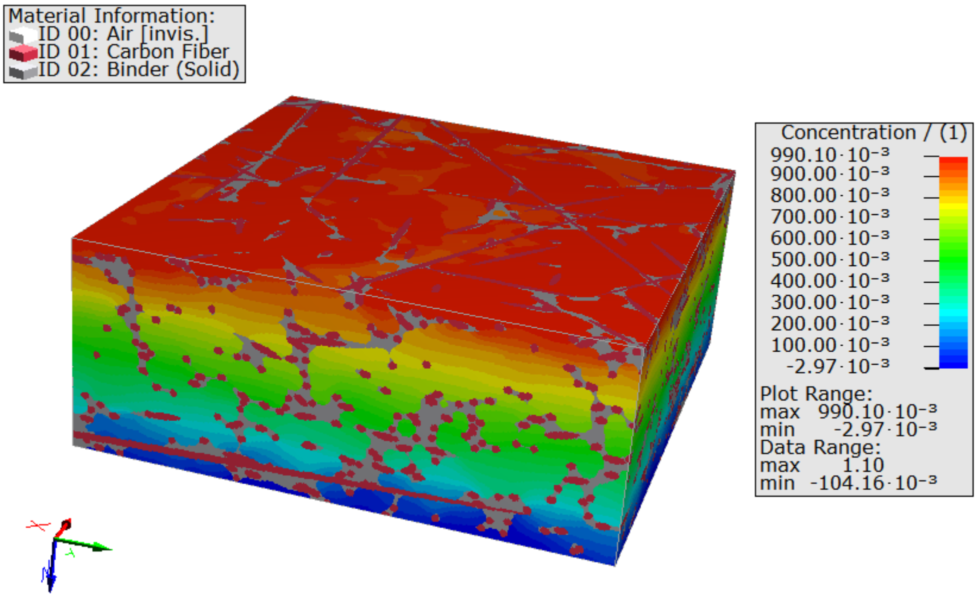


Select components to load / compute:

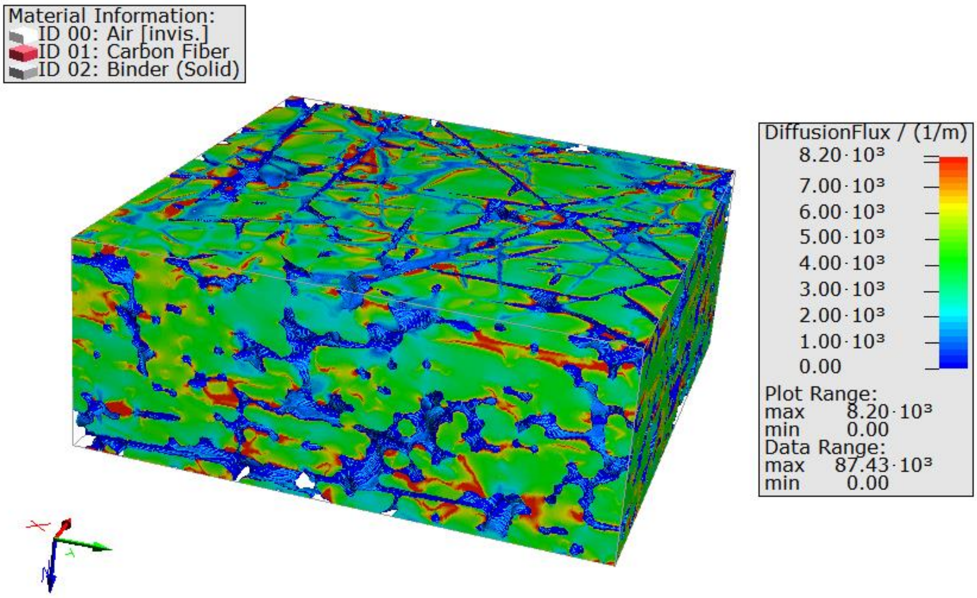
☑ Concentration:Concentration
☑ DiffusionFlux:DiffusionFluxX
☑ DiffusionFlux:DiffusionFluxY
☑ DiffusionFlux:DiffusionFluxZ
☑ DiffusionFlux:DiffusionFlux

☐ Allow clipping by distance
☐ Keep existing Volume Fields

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 handbook of this User Guide.

KNUDSEN DIFFUSION

THEORETICAL BACKGROUND

As described in the introduction, 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 [12] 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:

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

where $E[\ldots]$ denotes the expectation value, x_0 is the starting position of a particle and x_t the position at time t. $(.)^T$ 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 [7], [8] to define a **Knudsen tortuosity factor** by writing

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

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 some authors [11], 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 [11] 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.

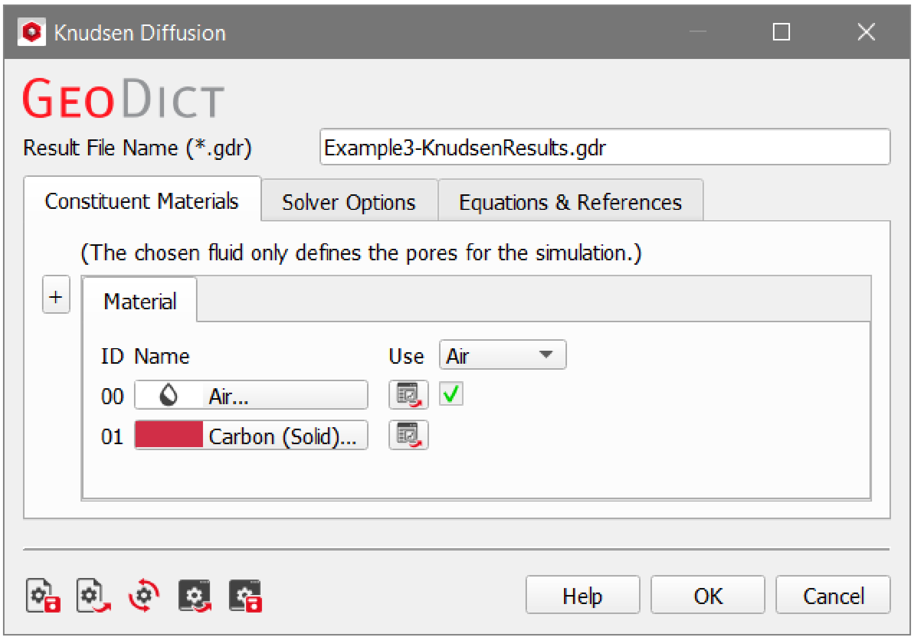
KNUDSEN DIFFUSION OPTIONS

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**, the user sets 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.



Result File Name (*.gdr) Example3-KnudsenResults.gdr

(The chosen fluid only defines the pores for the simulation.)

ID	Name	Use
		Air
00	Air...	✓
01	Carbon (Solid)...	

SOLVER OPTIONS

Travel Distance Mode

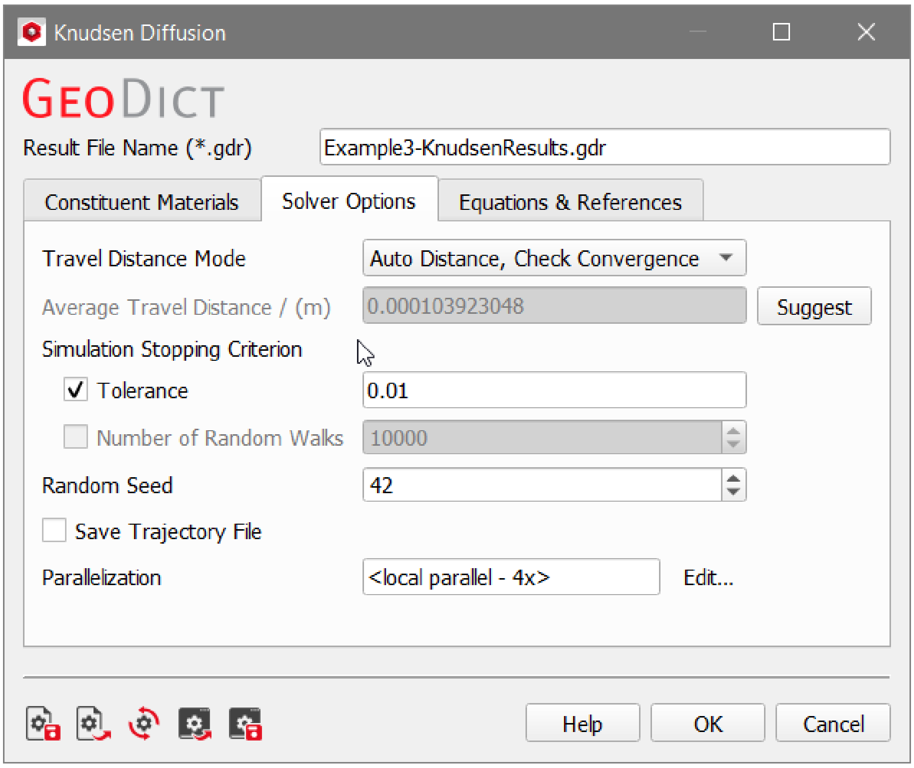
The computation of the diffusivity D converges when the number of particles $n \to \infty$ and the simulated time $t \to \infty$.

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, the user has 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.



Knudsen Diffusion

Result File Name (*.gdr): Example3-KnudsenResults.gdr

Travel Distance Mode: Auto Distance, Check Convergence

Average Travel Distance / (m): 0.000103923048

Simulation Stopping Criterion

☑ Tolerance: 0.01

☐ Number of Random Walks: 10000

Random Seed: 42

☐ Save Trajectory File

Parallelization: <local parallel - 4x>

Simulation Stopping Criterion

The user may choose between **Tolerance** and a fixed **Number of Random Walks**.

The parameter **Tolerance** determines the maximal relative standard deviation of the displacement. The mean displacement μ, and its standard deviation σ will stabilize with a higher number of simulated molecules n and finally, the relative standard deviation will converge to zero as:

$$\sigma/\mu \sim \sqrt{n}/n \qquad (20)$$

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.

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.

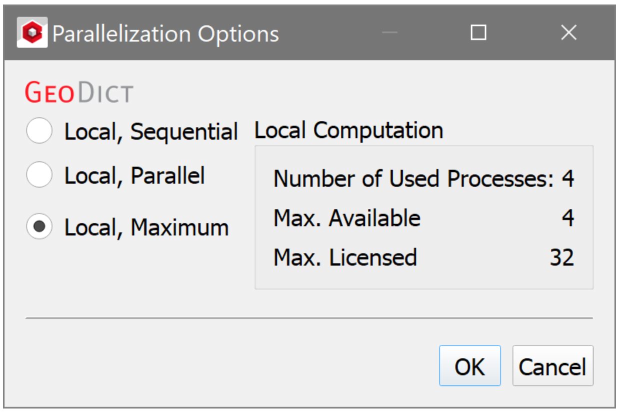
Save Trajectory File

By checking on **Save Trajectory File**, the particle trajectories are saved and can be visualized in 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 **Local, Sequential**, **Local, Parallel**, and **Local, Maximum.**



Parallelization Options

GeoDict

○ Local, Sequential

○ Local, Parallel

● Local, Maximum

Local Computation

Number of Used Processes: 4

Max. Available 4

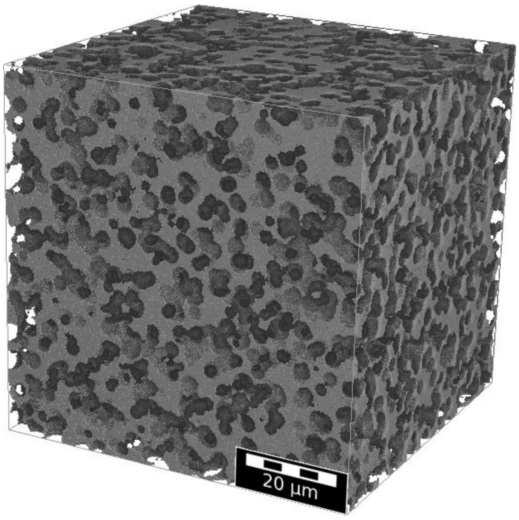
Max. Licensed 32

When **Local, Parallel** is chosen, the **Number of Processes** to run can be entered. If the entered number is larger than the one the license supports, an error message appears.

Parallelization Benchmark Results

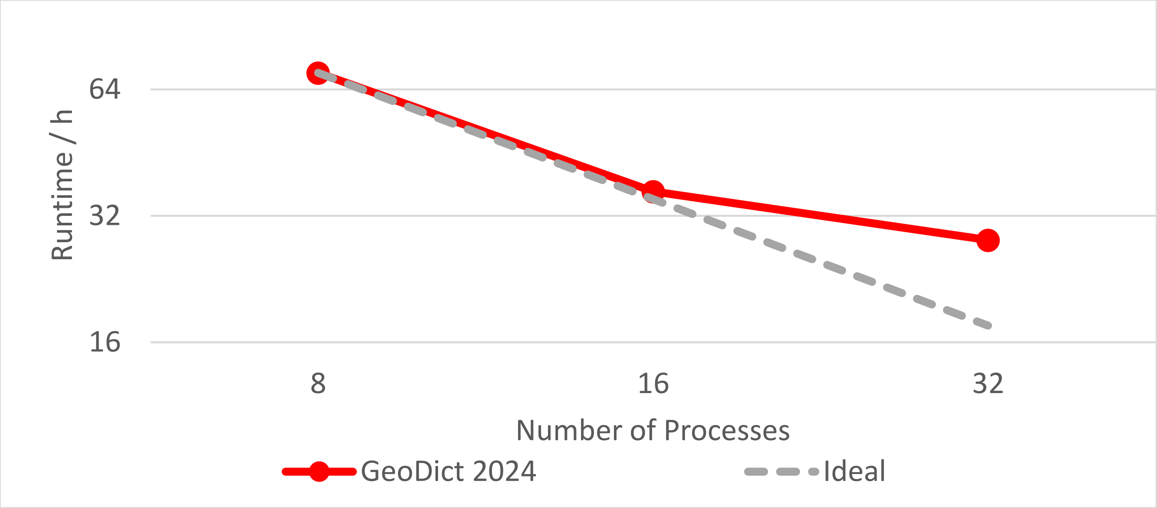
As an example for the parallelization, Knudsen diffusion in a microporous layer (MPL) is computed with different number of processes. The computation is run on a server with 2 x Intel E5-2697A v4 processors with 16 cores each, running with a maximum of 3.60 GHz, and 128 GB RAM.

The MPL structure of size 2,000 x 2,000 x 2,000 is created in GeoDict with GrainGeo and ProcessGeo. It has a porosity of 64%.



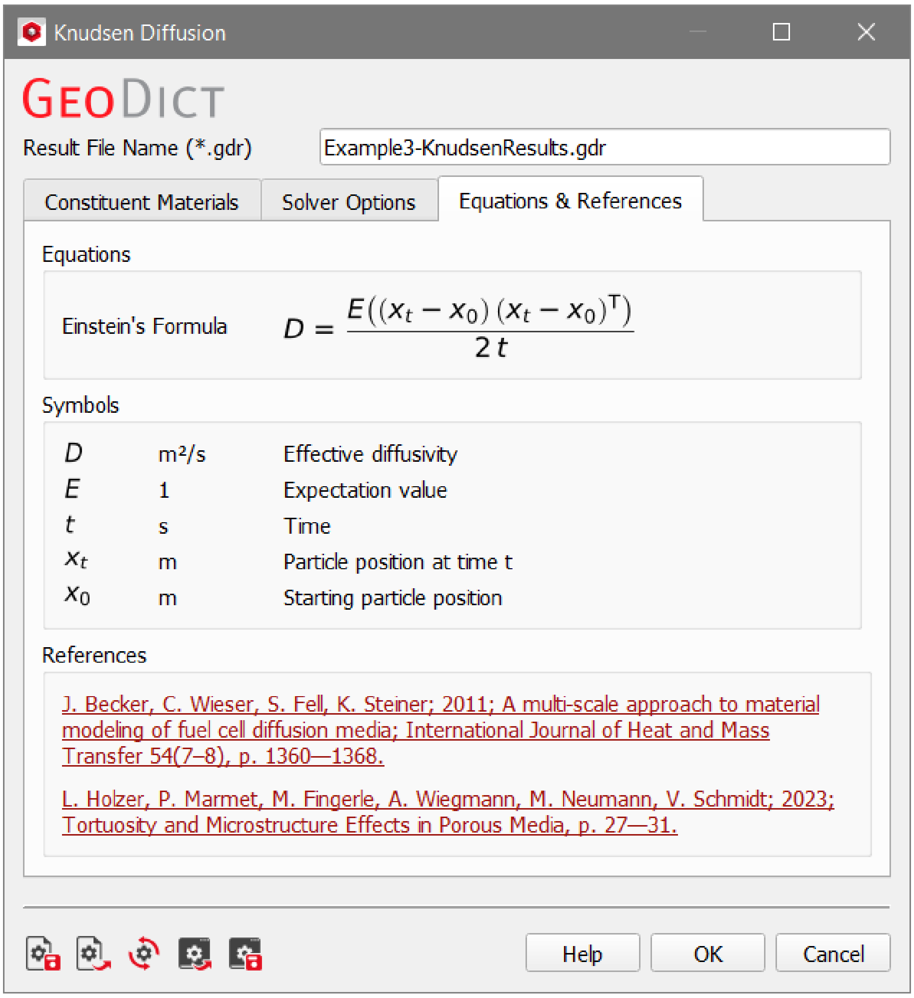
The computation for the structure with 8 billion voxels, requires only 32 GB of RAM. The effective Knudsen diffusivity for oxygen at 20°C in z-direction is 8.82e-05 m²/s.

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



EQUATIONS & REFERENCES

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



Result File Name (*.gdr) Example3-KnudsenResults.gdr

Equations

Einstein's Formula $D = \frac{E((x_t - x_0)(x_t - x_0)^T)}{2t}$

Symbols

D	m²/s	Effective diffusivity
E	1	Expectation value
t	s	Time
x_t	m	Particle position at time t
x_0	m	Starting particle position

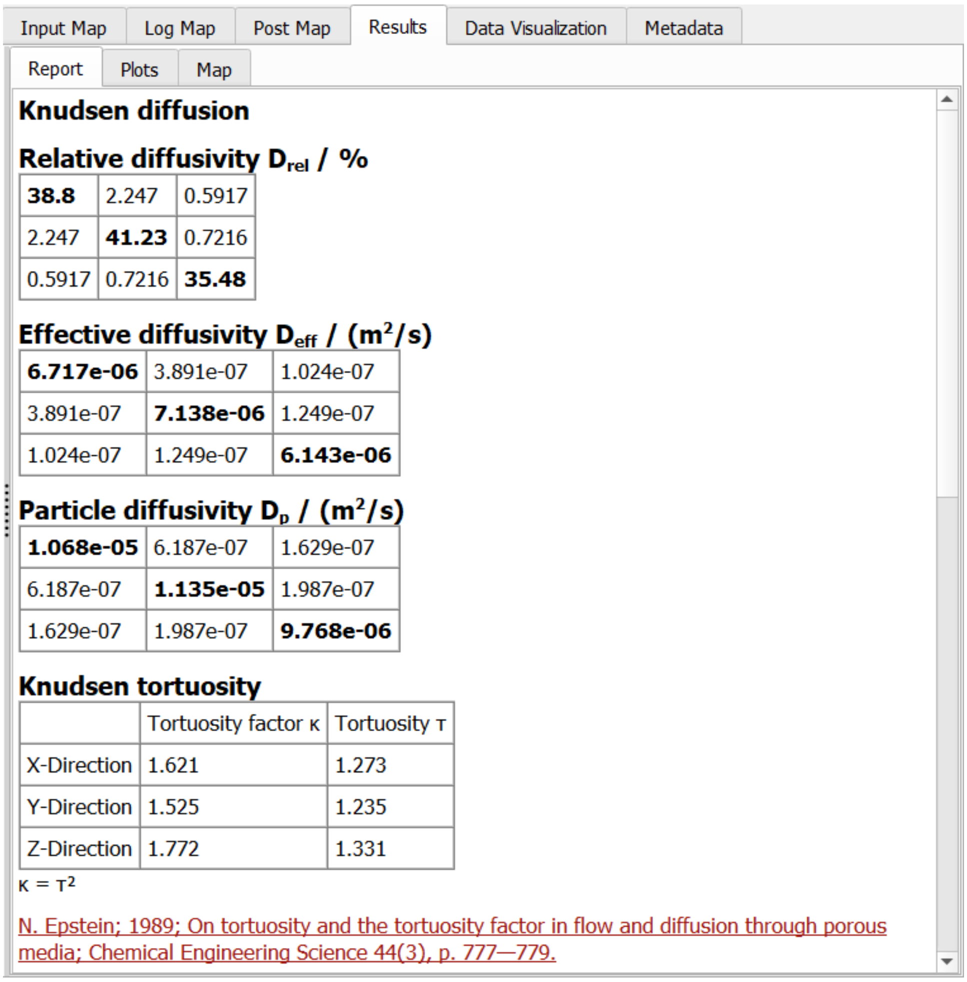
References

J. Becker, C. Wieser, S. Fell, K. Steiner; 2011; A multi-scale approach to material modeling of fuel cell diffusion media; International Journal of Heat and Mass Transfer 54(7–8), p. 1360—1368.

L. Holzer, P. Marmet, M. Fingerle, A. Wiegmann, M. Neumann, V. Schmidt; 2023; Tortuosity and Microstructure Effects in Porous Media, p. 27—31.

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):



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

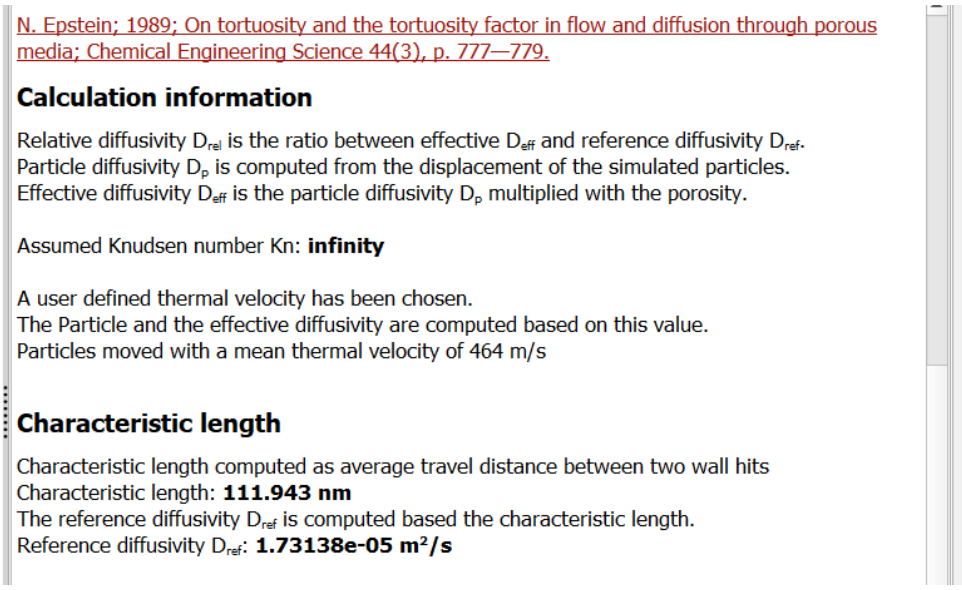
$\kappa = \tau^2$

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

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

- The dimensionless **Relative diffusivity** D^* as defined through equation (18).
- 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) 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).

Then, some calculation information is given:



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

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**

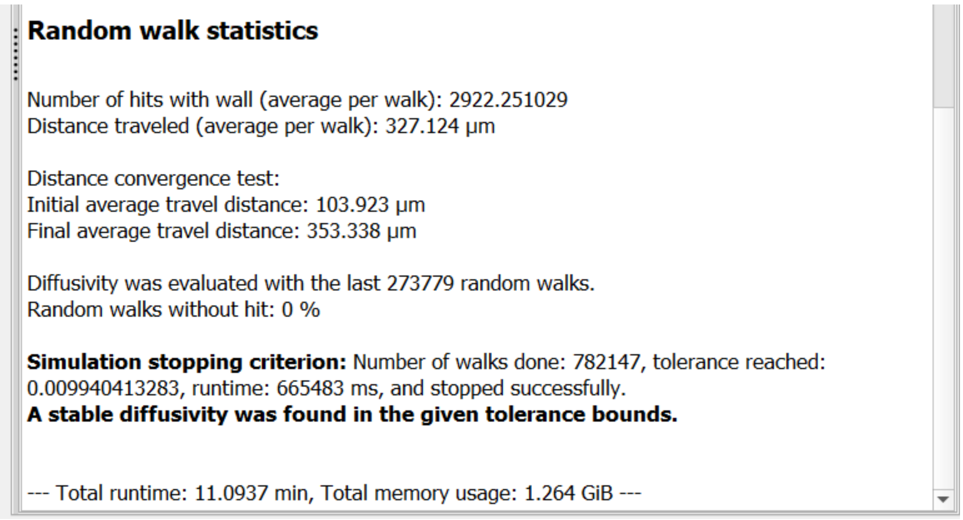
A user defined thermal velocity has been chosen.
The Particle and the effective diffusivity are computed based on this value.
Particles moved with a mean thermal velocity of 464 m/s

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 m²/s**

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

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



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

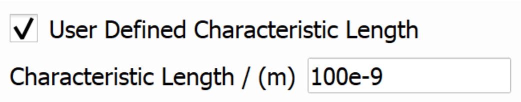
Diffusivity was evaluated with the last 273779 random walks.
Random walks without hit: 0 %

Simulation stopping criterion: Number of walks done: 782147, tolerance reached: 0.009940413283, runtime: 665483 ms, and stopped successfully.
A stable diffusivity was found in the given tolerance bounds.

--- Total runtime: 11.0937 min, Total memory usage: 1.264 GiB ---

- 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.

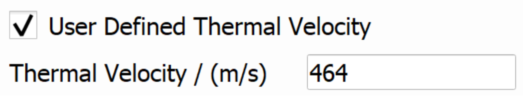


☑ User Defined Characteristic Length

Characteristic Length / (m) 100e-9

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**.



☑ User Defined Thermal Velocity

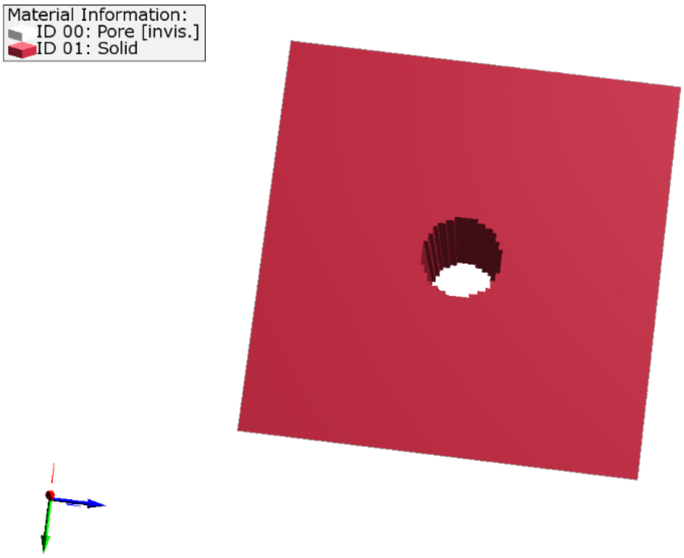
Thermal Velocity / (m/s) 464

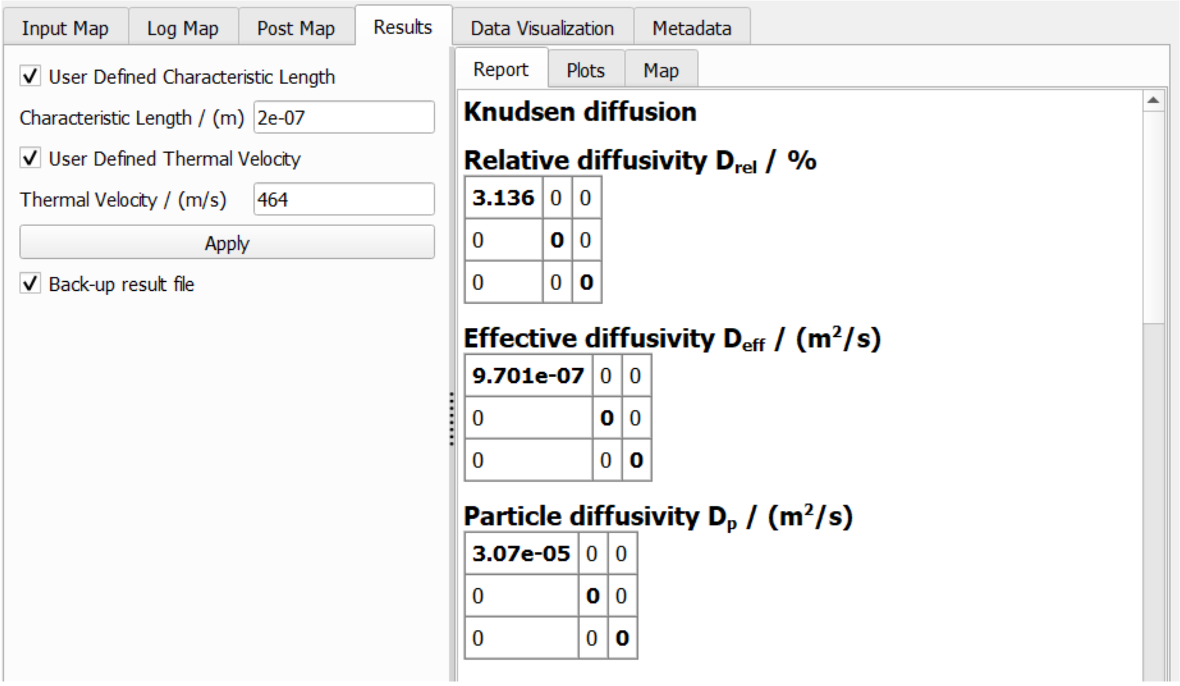
Changing the thermal velocity will also change the computed **Particle diffusivity** and the **Effective diffusivity**. To get the diffusivity in the structure for a specific diffusivity as input (e.g. for oxygen at 20°C), define the corresponding thermal velocity (for oxygen at 20°C: 440.4 m/s). The **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 $\frac{1}{3}d\bar{v}$, 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 m²/s is computed, which is very close to the expected value of 3.09e-5 m²/s:





☑ User Defined Characteristic Length

Characteristic Length / (m) 2e-07

☑ User Defined Thermal Velocity

Thermal Velocity / (m/s) 464

☑ Back-up result file

Knudsen diffusion

Relative diffusivity D_{rel} / %

3.136	0	0
0	**0**	0
0	0	**0**

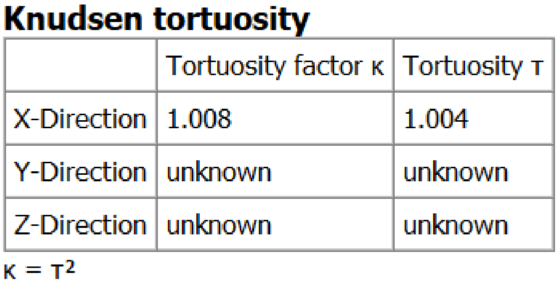
Effective diffusivity D_{eff} / (m^2/s)

9.701e-07	0	0
0	**0**	0
0	0	**0**

Particle diffusivity D_p / (m^2/s)

3.07e-05	0	0
0	**0**	0
0	0	**0**

For cylindrical pores, the Knudsen tortuosity factor is 1 in the direction of the pore, when using the pore diameter as characteristic length.

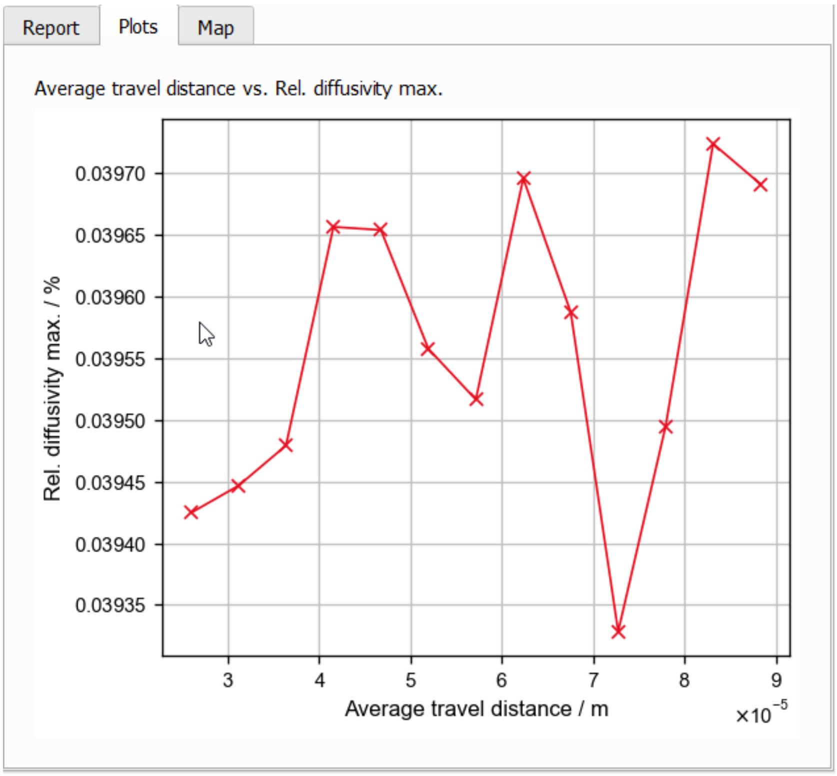


Knudsen tortuosity

	Tortuosity factor κ	Tortuosity τ
X-Direction	1.008	1.004
Y-Direction	unknown	unknown
Z-Direction	unknown	unknown

$\kappa = \tau^2$

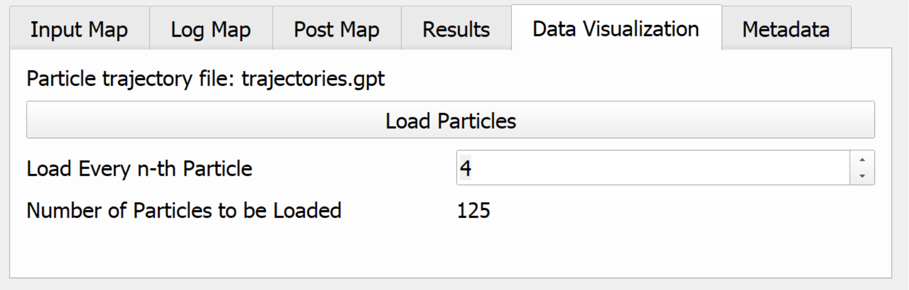
In the **Results – Plots** subtab, the determined **Rel. Diffusivity Max**.(which is the largest relative diffusivity value achieved in x,y or z-direction) is shown over the average travel distance, starting from the initial average travel distance to the final average travel distance. These values are 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.



DATA VISUALIZATION

If **Save Trajectory File** was checked (page 37), it is possible to load and visualize the particle trajectories under the **Data Visualization** tab.

By clicking **Load Particles**, the GPT trajectory file (GPT, GeoDict particle trajectories) is loaded.



Particle trajectory file: trajectories.gpt

Load Every n-th Particle 4

Number of Particles to be Loaded 125

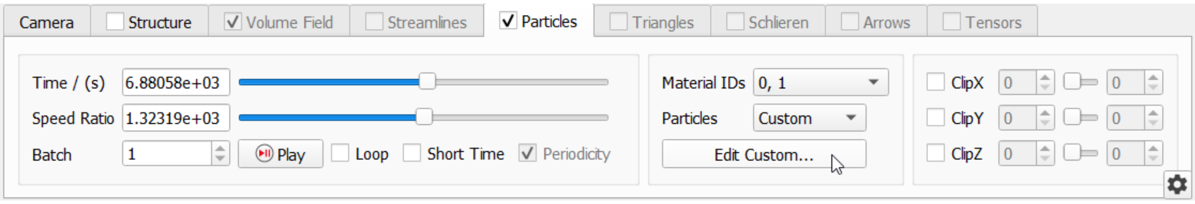
As the number of simulated random walks may be very high, it is possible to load only **Every n-th Particle**. The **Number of Particles to be Loaded** is reported.

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.

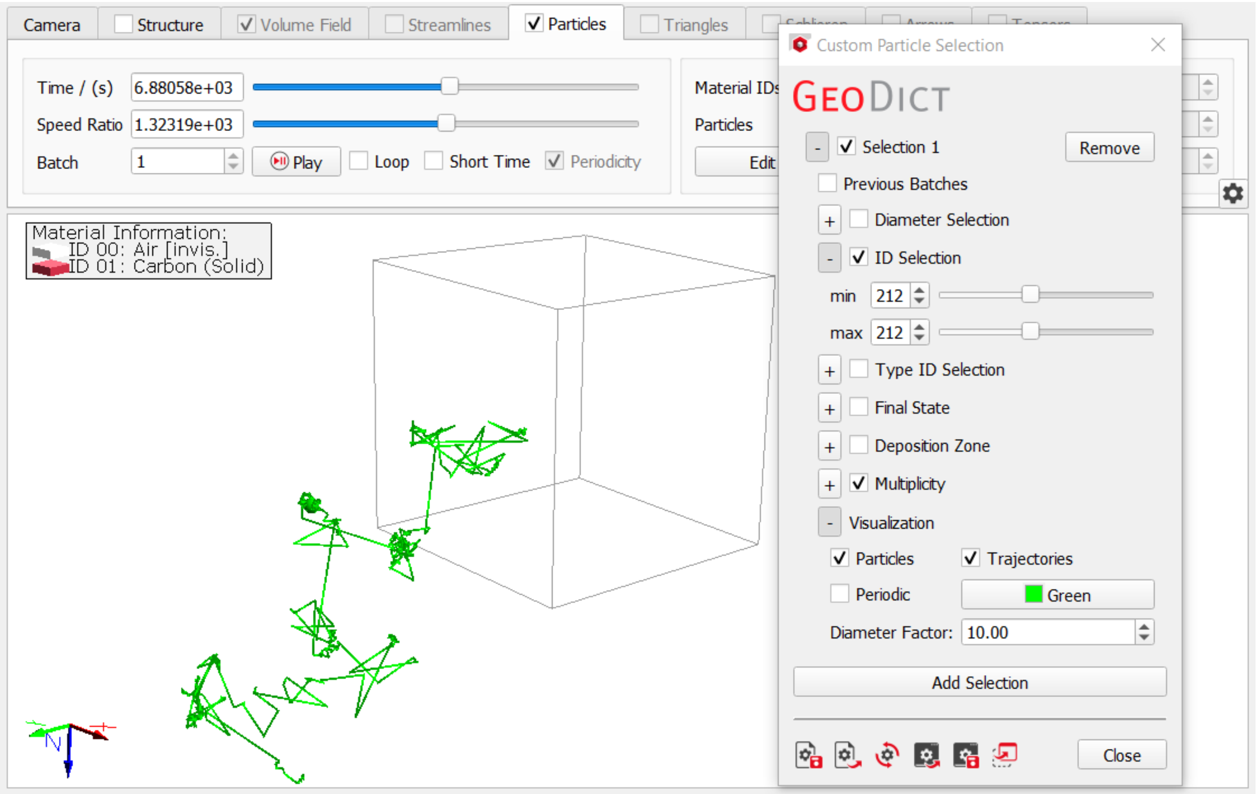
This will show a large number of particles moving around. To have a closer look at a single particle, choose **Custom** in the **Particles** pull-down menu.

Then, click the **Edit Custom...** button (that appears underneath) to open the **Custom Particle Selection**.



Time / (s) 6.88058e+03
Speed Ratio 1.32319e+03
Batch 1
☐ Loop ☐ Short Time ☑ Periodicity
Material IDs 0, 1
Particles Custom
☐ ClipX 0 0
☐ ClipY 0 0
☐ ClipZ 0 0

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



Time / (s) 6.88058e+03
Speed Ratio 1.32319e+03
Batch 1
☐ Loop ☐ Short Time ☑ Periodicity

Material Information:
ID 00: Air [invis.]
ID 01: Carbon (Solid)

☑ Selection 1
☐ Previous Batches
☐ Diameter Selection
☑ ID Selection
min 212
max 212
☐ Type ID Selection
☐ Final State
☐ Deposition Zone
☑ Multiplicity
Visualization
☑ Particles ☑ Trajectories
☐ Periodic 🟩 Green
Diameter Factor: 10.00

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.

MOLECULAR DIFFUSION

THEORETICAL BACKGROUND

The Knudsen diffusion algorithm described in the previous section 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 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 described before.

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 diffusively according to Lambert’s cosine law [12] 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).

MOLECULAR DIFFUSION OPTIONS

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

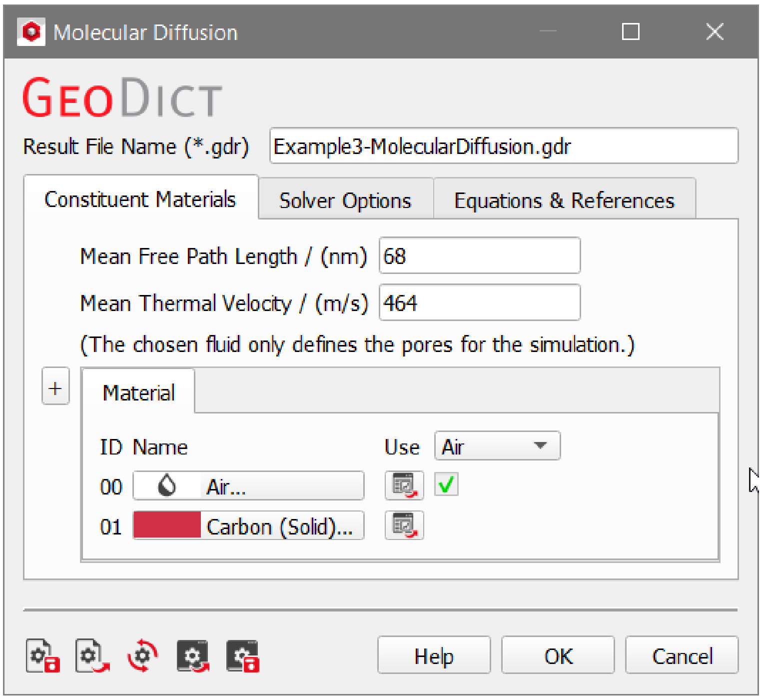
CONSTITUENT MATERIALS

As described for the **Bulk (Laplace) Diffusion**, the user sets 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 described on page 35.



Result File Name (*.gdr) Example3-MolecularDiffusion.gdr

Mean Free Path Length / (nm) 68

Mean Thermal Velocity / (m/s) 464

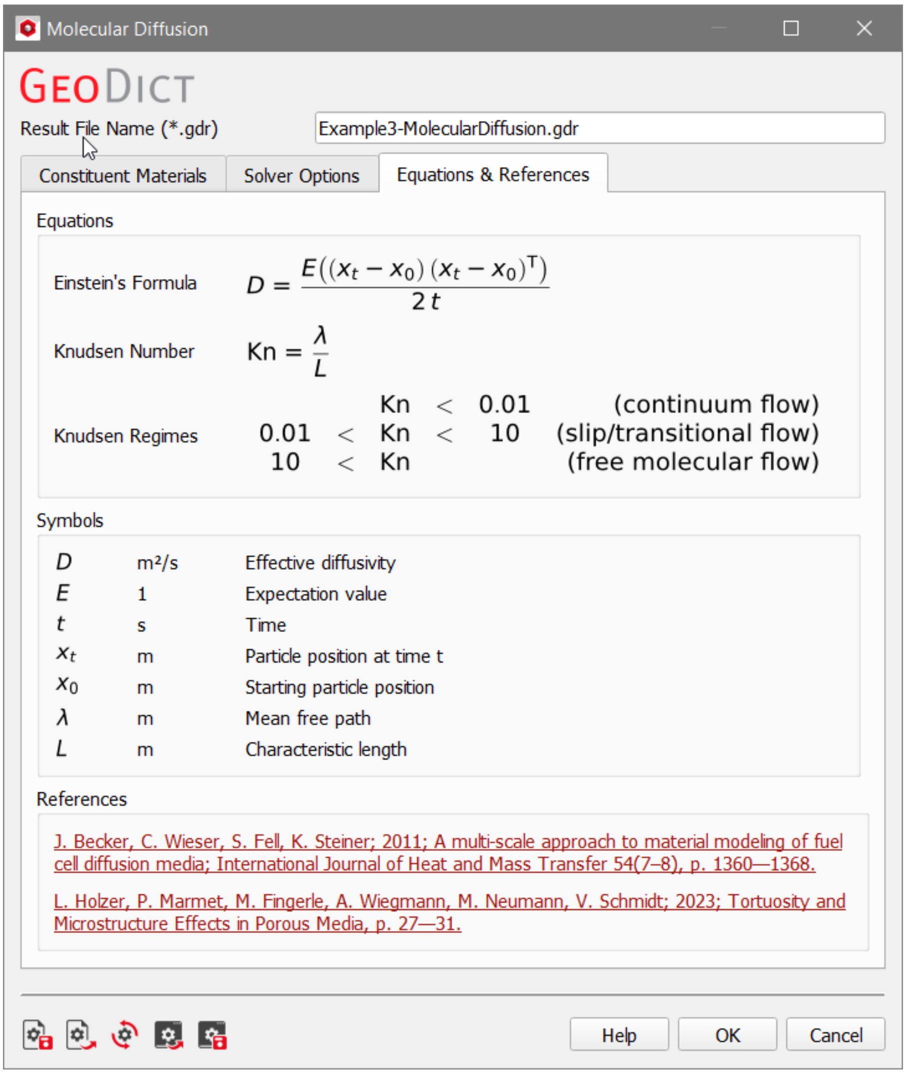
(The chosen fluid only defines the pores for the simulation.)

Material

ID	Name	Use: Air
00	Air...	✓
01	Carbon (Solid)...	

EQUATIONS & REFERENCES

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



Result File Name (*.gdr) Example3-MolecularDiffusion.gdr

Equations

Einstein's Formula $D = \frac{E\left((x_t - x_0)\,(x_t - x_0)^{\mathrm{T}}\right)}{2\,t}$

Knudsen Number $\mathrm{Kn} = \frac{\lambda}{L}$

Knudsen Regimes

$$\begin{array}{rcccrl} & & \mathrm{Kn} & < & 0.01 & \text{(continuum flow)} \\ 0.01 & < & \mathrm{Kn} & < & 10 & \text{(slip/transitional flow)} \\ 10 & < & \mathrm{Kn} & & & \text{(free molecular flow)} \end{array}$$

Symbols

D	m²/s	Effective diffusivity
E	1	Expectation value
t	s	Time
x_t	m	Particle position at time t
x_0	m	Starting particle position
λ	m	Mean free path
L	m	Characteristic length

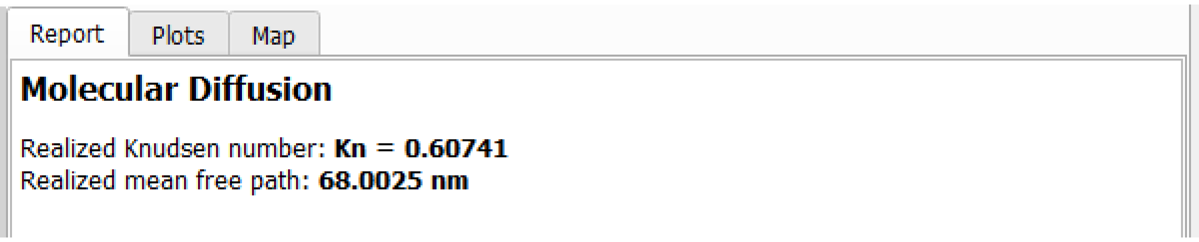
References

J. Becker, C. Wieser, S. Fell, K. Steiner; 2011; A multi-scale approach to material modeling of fuel cell diffusion media; International Journal of Heat and Mass Transfer 54(7–8), p. 1360—1368.

L. Holzer, P. Marmet, M. Fingerle, A. Wiegmann, M. Neumann, V. Schmidt; 2023; Tortuosity and Microstructure Effects in Porous Media, p. 27—31.

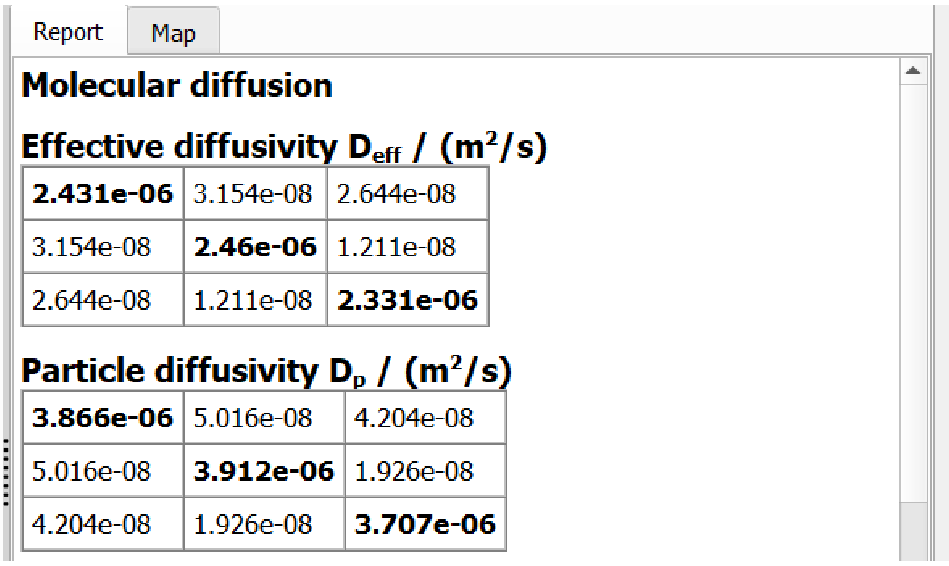
RESULTS

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):



Molecular Diffusion

Realized Knudsen number: **Kn = 0.60741**
Realized mean free path: **68.0025 nm**



Molecular diffusion

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

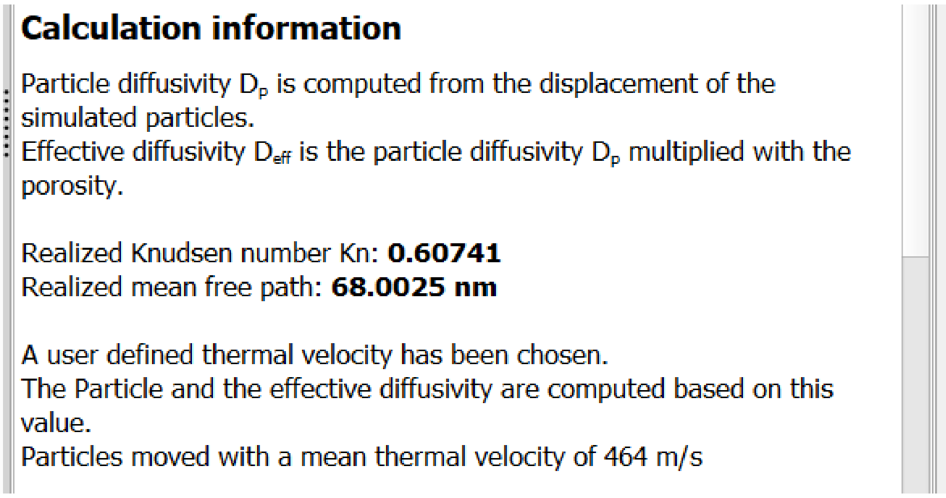
2.431e-06	3.154e-08	2.644e-08
3.154e-08	**2.46e-06**	1.211e-08
2.644e-08	1.211e-08	**2.331e-06**

Particle diffusivity D_p / (m^2/s)

3.866e-06	5.016e-08	4.204e-08
5.016e-08	**3.912e-06**	1.926e-08
4.204e-08	1.926e-08	**3.707e-06**

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) using the user defined mean thermal velocity.



Calculation information

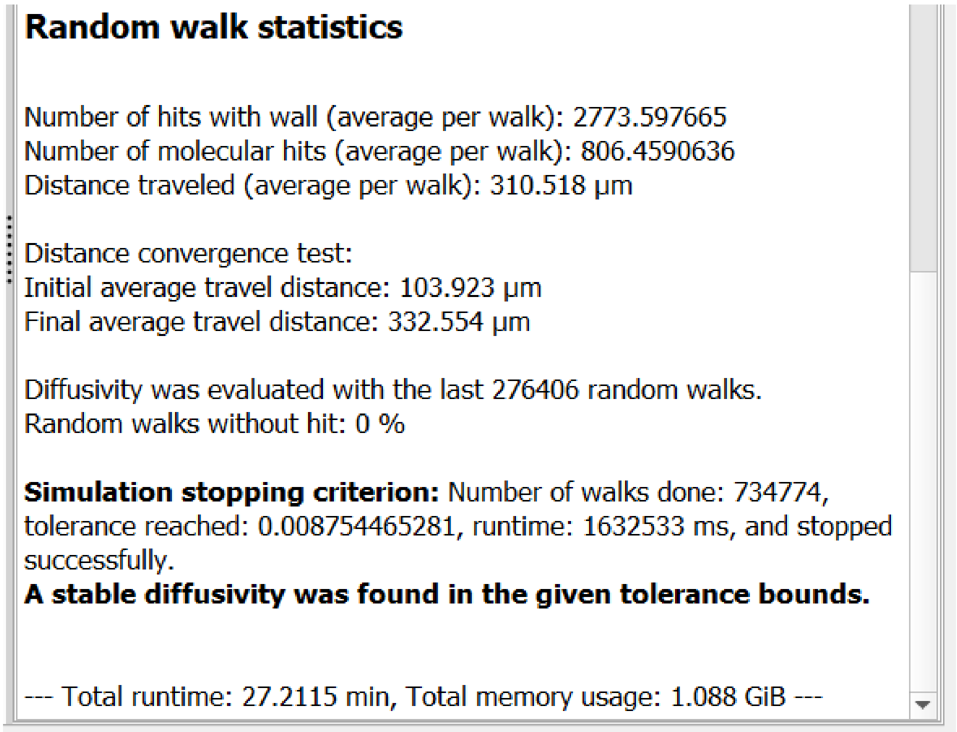
Particle diffusivity D_p is computed from the displacement of the simulated particles.
Effective diffusivity D_{eff} is the particle diffusivity D_p multiplied with the porosity.

Realized Knudsen number Kn: **0.60741**
Realized mean free path: **68.0025 nm**

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

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

- The **Realized Knudsen number** 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.



Random walk statistics

Number of hits with wall (average per walk): 2773.597665
Number of molecular hits (average per walk): 806.4590636
Distance traveled (average per walk): 310.518 µm

Distance convergence test:
Initial average travel distance: 103.923 µm
Final average travel distance: 332.554 µm

Diffusivity was evaluated with the last 276406 random walks.
Random walks without hit: 0 %

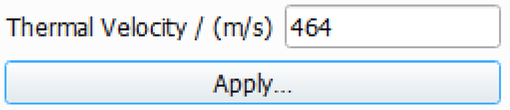
Simulation stopping criterion: Number of walks done: 734774, tolerance reached: 0.008754465281, runtime: 1632533 ms, and stopped successfully.
A stable diffusivity was found in the given tolerance bounds.

--- Total runtime: 27.2115 min, Total memory usage: 1.088 GiB ---

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.



Thermal Velocity / (m/s) 464

BOSANQUET APPROXIMATION

THEORETICAL BACKGROUND

Bosanquet's approximation [6] is an estimation of the effective diffusivity at intermediate Knudsen numbers:

$$D = \left(D_{Kn=0}^{-1} + D_{Kn=\infty}^{-1}\right)^{-1} \tag{21}$$

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** and the **Knudsen Diffusion** commands to compute the resulting diffusivity D. However, both commands have not computed $D_{Kn=0}^{-1}$ and $D_{Kn=\infty}^{-1}$, respectively, but the corresponding relative diffusivity.

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

$$D_{Kn=0} = \frac{1}{3}\lambda\,\bar{v}\,D_{Kn=0}^{*} \tag{22}$$

$$D_{Kn=\infty} = \frac{1}{3}L\bar{v}D_{Kn=\infty}^{*} \tag{23}$$

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

$$Kn = \frac{\lambda}{L} \tag{24}$$

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

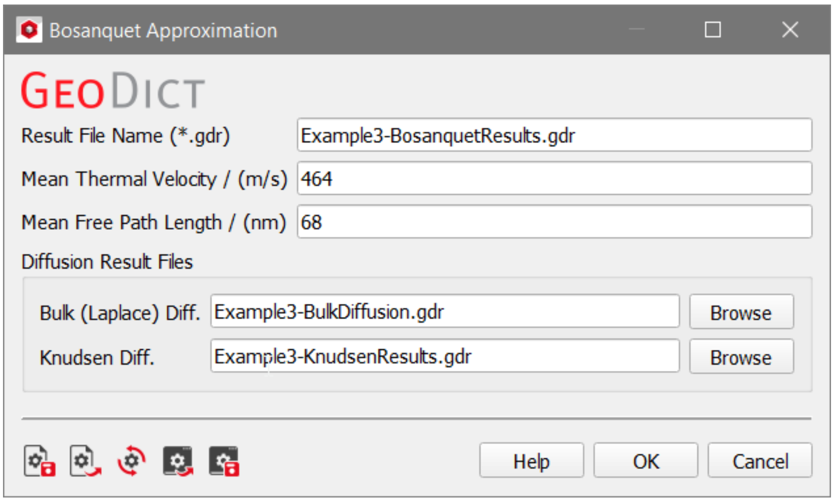
For binary gases, equation (17) no longer holds true, but is possible to use equation (7) where d_0 describes a measurable quantity:

$$D_{Kn=0} = d_0 D_{Kn=0}^{*} \tag{25}$$

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.

BOSANQUET APPROXIMATION OPTIONS

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



Result File Name (*.gdr) Example3-BosanquetResults.gdr
Mean Thermal Velocity / (m/s) 464
Mean Free Path Length / (nm) 68
Diffusion Result Files
Bulk (Laplace) Diff. Example3-BulkDiffusion.gdr Browse
Knudsen Diff. Example3-KnudsenResults.gdr Browse

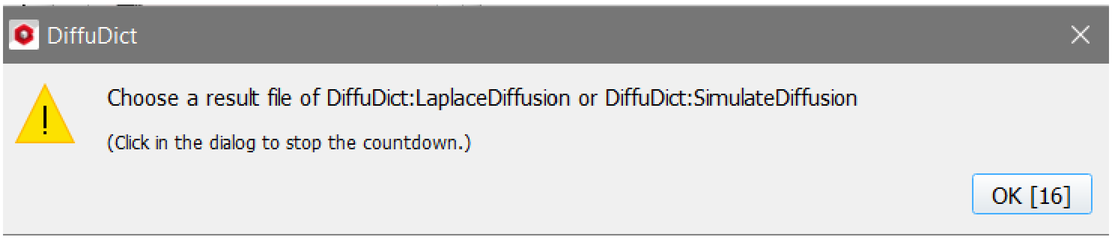
Mean Thermal Velocity is the average speed of the thermal motion of particles, which make up a gas, liquid, etc. It can be calculated by

$$v = \sqrt{\frac{8kT}{\pi m}} \tag{26}$$

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

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

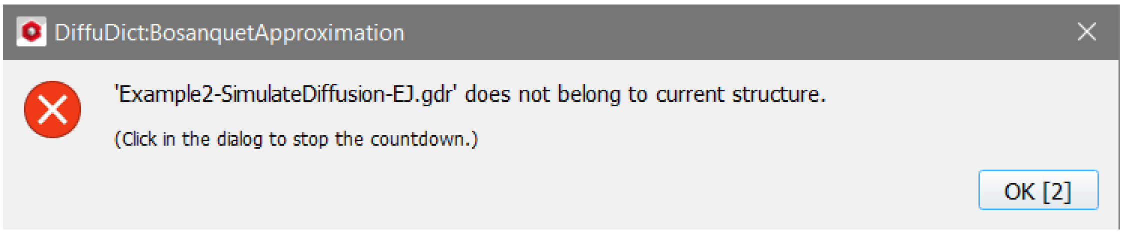
Bulk (Laplace) Diffusion and **Knudsen Diffusion** must be previously computed and the path to the corresponding result files must be loaded using the **Browse** buttons. The loaded result files must correspond to calculations of Bulk (Laplace) Diffusion and Knudsen Diffusion, respectively. Otherwise, a warning appears when clicking **OK** in the Bosanquet Approximation dialog box:



Choose a result file of DiffuDict:LaplaceDiffusion or DiffuDict:SimulateDiffusion

(Click in the dialog to stop the countdown.)

The loaded result files must also correspond to the current structure; otherwise, an error message is displayed after clicking Run.



'Example2-SimulateDiffusion-EJ.gdr' does not belong to current structure.

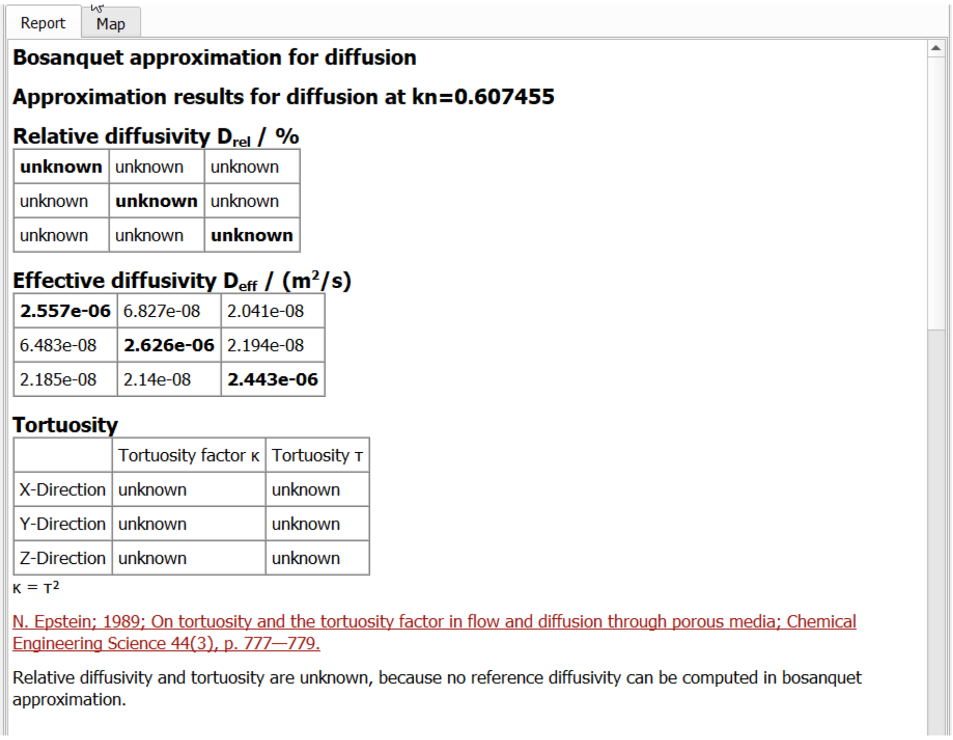
(Click in the dialog to stop the countdown.)

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):

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



Bosanquet approximation for diffusion

Approximation results for diffusion at kn=0.607455

Relative diffusivity D_{rel} / %

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

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

2.557e-06	6.827e-08	2.041e-08
6.483e-08	**2.626e-06**	2.194e-08
2.185e-08	2.14e-08	**2.443e-06**

Tortuosity

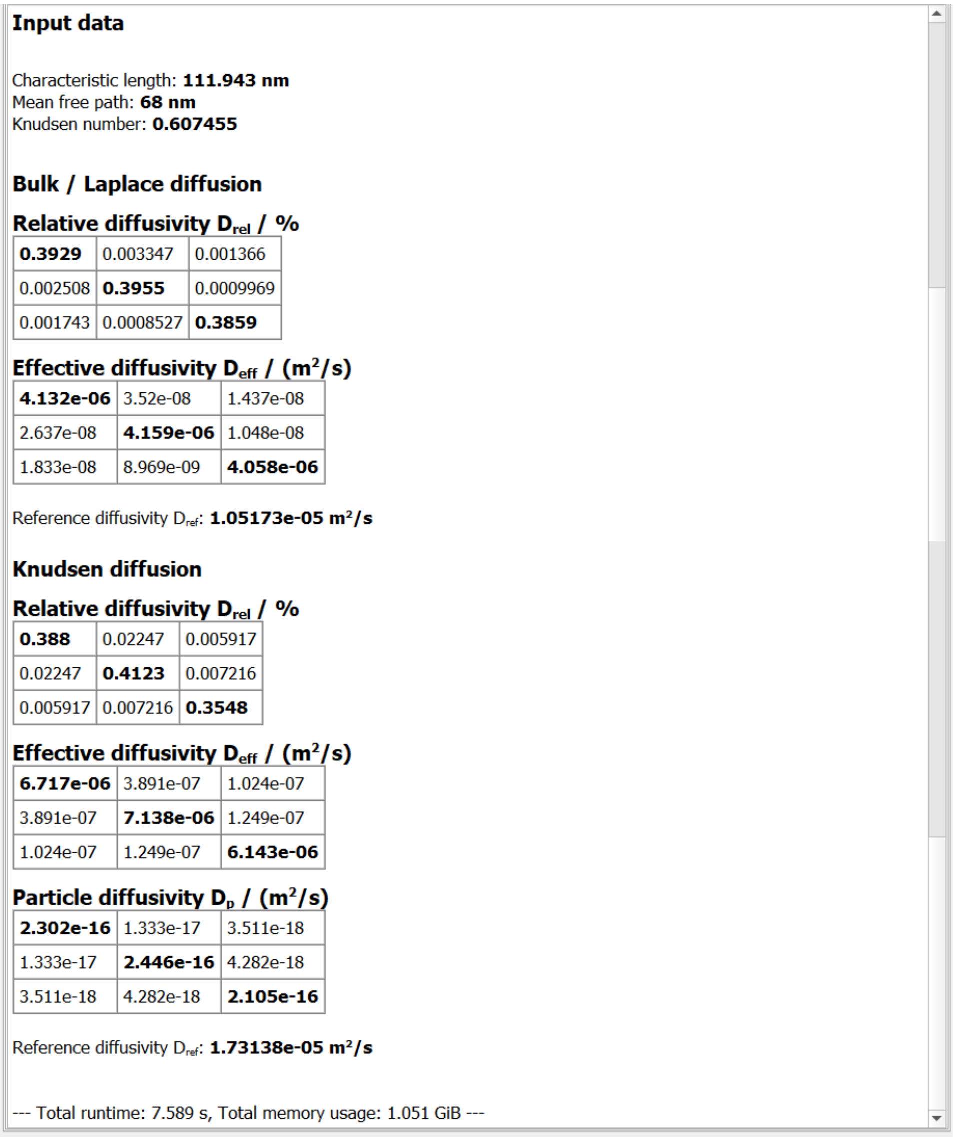
	Tortuosity factor κ	Tortuosity τ
X-Direction	unknown	unknown
Y-Direction	unknown	unknown
Z-Direction	unknown	unknown

$\kappa = \tau^2$

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

Relative diffusivity and tortuosity are unknown, because no reference diffusivity can be computed in bosanquet approximation.

- The overall diffusivity D, computed with Bosanquet's formula (21).
- Characteristic length L as used in equation (17) and computed with the Knudsen Diffusion command.
- Mean free path λ, as entered by the user.
- Knudsen number computed as $Kn = \frac{\lambda}{L}$.
- Relative bulk diffusivity $D^*_{Kn=0}$ as defined in equation (7) and computed with the Bulk (Laplace) Diffusion command.
- Effective bulk diffusivity $D_{Kn=0}$, determined by multiplication of the dimensionless bulk diffusivity with the reference diffusivity $\frac{1}{3}\lambda\bar{v}$ as in equation (22).
- Relative Knudsen diffusivity $D^*_{Kn=\infty}$ as defined in equation (18) and computed with the Knudsen Diffusion command.
- Effective Knudsen diffusivity $D_{Kn=\infty}$, determined by multiplication of the dimensionless Knudsen diffusivity with the reference diffusivity $\frac{1}{3}L\bar{v}$ as in equation (23).
- The Particles diffusivity for Knudsen diffusion computed by equation (16) using the user defined mean thermal velocity.
- Total runtime and total memory usage.



Input data

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

Bulk / Laplace diffusion

Relative diffusivity D_{rel} / %

0.3929	0.003347	0.001366
0.002508	**0.3955**	0.0009969
0.001743	0.0008527	**0.3859**

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} / %

0.388	0.02247	0.005917
0.02247	**0.4123**	0.007216
0.005917	0.007216	**0.3548**

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)

2.302e-16	1.333e-17	3.511e-18
1.333e-17	**2.446e-16**	4.282e-18
3.511e-18	4.282e-18	**2.105e-16**

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

--- Total runtime: 7.589 s, Total memory usage: 1.051 GiB ---

DIFFUSION GEOAPPS

All macros/scripts stored in the folder

`C:\Program Files\Math2Market GmbH\GeoDict 2024\GeoApps\Modules\DiffuDict`

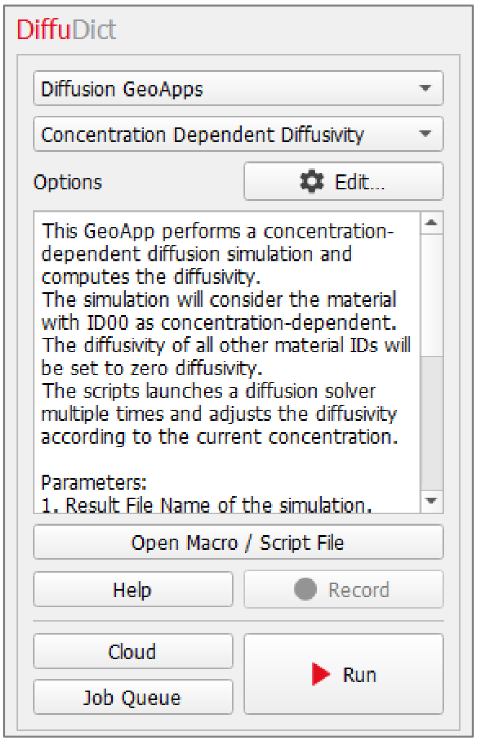
are shown under the Diffusion GeoApps drop-menu in the DiffuDict section.

Currently, this is the *Concentration Dependent Diffusivity.py* App, distributed with DiffuDict.

CONCENTRATION DEPENDENT DIFFUSIVITY

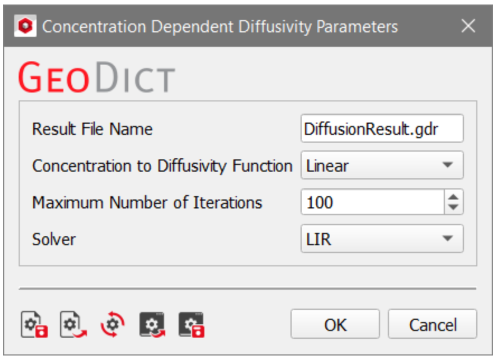
The window below the selection box shows the description of the script and its input parameters. The **Concentration Dependent Diffusivity** script approximates a diffusion process where the local diffusivity depends on the concentration of the diffusing species. This is achieved by using the existing constant diffusivity solver in an iterative process, where the local diffusivity in one iteration depends on the local concentration computed in the previous iteration.

Clicking **Edit...** opens the dialog to enter the available parameters.



This GeoApp performs a concentration-dependent diffusion simulation and computes the diffusivity.
The simulation will consider the material with ID00 as concentration-dependent.
The diffusivity of all other material IDs will be set to zero diffusivity.
The scripts launches a diffusion solver multiple times and adjusts the diffusivity according to the current concentration.

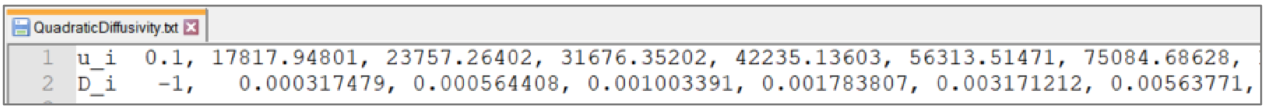
Parameters:
1. Result File Name of the simulation.



Result File Name: DiffusionResult.gdr
Concentration to Diffusivity Function: Linear
Maximum Number of Iterations: 100
Solver: LIR

Under **Concentration to Diffusivity Function**, if **Linear** instead of **Custom** is selected**,** a text file (*.txt) describing the relationship between diffusivity value (D_i) and concentration value (u_i) can be given. Examples for such files can be found in

`C:\Program Files\Math2Market GmbH\GeoDict 2024\GeoApps\Modules\DiffuDict\DiffusivityFuncs`



```
u_i   0.1, 17817.94801, 23757.26402, 31676.35202, 42235.13603, 56313.51471, 75084.68628,
D_i    -1,    0.000317479, 0.000564408, 0.001003391, 0.001783807, 0.003171212, 0.00563771,
```

For example, one of these can be customized, by copying them to a project specific file location and editing them 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.

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

$$div\big(D(u)grad(u)\big) = 0 \tag{27}$$

$$\frac{d}{dn}u = 0 \tag{28}$$

$$u(z_{min}) = u_{min};\ u(z_{max}) = u_{max} \tag{29}$$

$$\hat{u} = u \tag{30}$$

Here, **u** is the concentration and **D** the concentration-dependent diffusivity. For the surfaces the boundary condition (28) is applied. In Z-direction, constant Dirichlet-boundary conditions (29) 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 (30). The solution to the non-linear equation (27) 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 (27) 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.

Open Macro /Script File opens 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.

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

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

REFERENCES

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

[2] 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 (2011), 1360-1368.

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

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

[5] S. Linden, A. Wiegmann and H. Hagen, *The LIR space partitioning system applied to the Stokes equations*, Graphical Models 82 (2015), 58-66.

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

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

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

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

[10] 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).

[11] 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 (2004), 2947-2960.

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

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

Technical documentation:

Jürgen Becker
Andreas Wiegmann
Liping Cheng
Barbara Planas

Math2Market GmbH

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