diffusion-fundamentals

The Open-Access Journal for the Basic Principles of Diffusion Theory, Experiment and Application

Cube-based diffusion algorithm

Stephan Zschiegner,[1,2] Stefanie Russ,[3] Armin Bunde,[1] Jörg Kärger[2]

[1]Universität Giessen, Institut für Theoretische Physik III, Germany
[2]Universität Leipzig, Fakultät für Physik und Geowissenschaften, Germany
[3]Freie Universität Berlin, Institut für Theoretische Physik, Germany

Corresponding author:
Stephan Zschiegner,
Justus-Liebig-Universität Giessen
Institut für Theoretische Physik III
35392 Giessen, Germany
E-Mail: stephan.zschiegner@physik.uni-giessen.de

Abstract

We study molecular diffusion in (linear) nanopores with different types of roughness in the so-called Knudsen regime. Knudsen diffusion represents the limiting case of molecular diffusion in pores, where mutual encounters of the molecules within the free pore space may be neglected and the time of flight between subsequent collisions with the pore walls significantly exceeds the interaction time between the pore walls and the molecules. In this paper we present in full detail the algorithm, which we used in our previous studies, where we showed complete equivalence of self- (or tracer-) and transport diffusion.

Keywords

Knudsen, Lambert, diffusion, algorithm, correction method

1. Introduction

Diffusion phenomena of gases in disordered and porous media have been subject to intense research for several decades [1-6] with applications in heterogenous catalysis [7], adsorption [8] and separation [9]. Recent progress in synthesizing nanostructured porous materials [8,10] has provided essentially unlimited options for the generation of purpose-tailored pore architectures and there is an increasing demand for clarification of the main features of molecular transport in such systems [11,12].

In matter conversion and separation, as two prominent technical applications, bimodal porous materials have attained particular attention. These materials contain “transport pores” that ensure fast molecular exchange between the microporous regions, in which the actual conversion and separation takes place [13]. In these transport pores the so-called Knudsen diffusion dominates, where the interaction of the molecules with the pore walls play the crucial role and the intermolecular interaction can be neglected. In this case, the molecules perform a series of free flights and change direction statistically after each collision with the pore wall.

Two different kinds of problems can be considered in a typical diffusion experiment or simulation, respectively: The so-called transport diffusion, where the particles diffuse in a non-equilibrium situation from one side of the system to the opposite side (under the influence of a concentration gradient) and the self- (or tracer-) diffusion under equilibrium conditions. These processes are described by the transport diffusion coefficient D_t and the self- (or tracer) diffusion coefficient D_s, respectively. D_t is defined by Fick's 1st law as the proportionality constant between the current density $\vec{j}$ and the concentration gradient $\partial\vec{c}/\partial x$,

$$\vec{j} = -D_t \frac{\partial \vec{c}}{\partial x} \tag{1}$$

while D_s is defined by the mean square displacement $\langle x^2(t)\rangle$ of a random walker

$$\langle x^2(t)\rangle = 2dD_s t \tag{2}$$

after time t, where d is the dimension of the pore.

In our previous works [14-18] we showed equivalence of self- (or tracer-) diffusion and transport diffusion in the Knudsen regime. In this paper we present our diffusion algorithm in full detail, including structure generation, particle attributes, diffusion simulation and the correction method (Enhanced f_t method, EFM, see [18]) which we used to obtain complete equivalence between both diffusion coefficients.

2. System structure

Usually a direct approach is used to calculate collisions of particle trajectories with pore walls. Thereby every possible wall has to be checked for possible hits of the particle trajectory. For simple systems with easy geometries and only few different walls, especially in two dimensions, such a direct approach may be sufficient (indeed we used this approach in the 2D calculations in [14]). But for systems with many walls even an optimized wall sorting scheme that searches for possible collisions in order of ascending distance of the wall from the starting point of the particle (often used in 3D graphics) becomes too time-consuming.

In our algorithm, we use cube elements (*cubicles*) to generate the pore structure. Particle trajectories are calculated successively by short path elements through adjoining cubicles. Since only those cube elements are needed that are actually passed by the particle, the calculation time hereby scales strictly linear with the system size, see Fig. 1. Hence large systems can easily be implemented into our simulation. Additionally (not shown in the figure), for simple geometries, the calculation time can be optimized using larger cubicles.

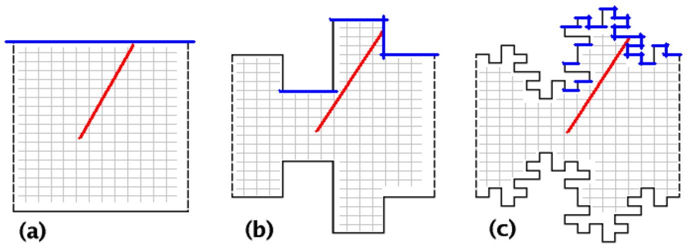

Fig. 1: Different approaches to determine the particle trajectory. The computation time for a cube based algorithm depends linearly on the system size, whereas a direct calculation of the intersection of a particle trajectory with the pore wall grows with the number of possible wall elements like $\mathcal{O}(\mathrm{nlogn})$. Of course, the size of a cubicle may be increased in (a) and (b).

2.1. The Framework

Our pore system is based on a periodic cubic lattice of size $c_{x,max} \times c_{y,max} \times c_{z,max}$, consisting of elementary *cubicles*. The cubicles assemble the porous structure of the whole system and can be either empty, i.e. part of the inner pore, or solid, i.e. part of the pore walls. Clearly, the particles only move inside the empty cubicles. If a particle hits a cubicle belonging to the wall, it is reflected back into the empty part of the inner pore, where it can move freely.

The location of any given particle is represented by three different sets of coordinates, i.e. by the 'house number' n that summarizes the cubicles inside a certain region (see Fig. 2), the cubicle vector $\vec{c} = (c_x, c_y, c_z)$ to the center of the given cubicle in the given 'house' and the incubicle coordinate $\vec{r} = (r_x, r_y, r_z)$ inside the cubicle, with $-1 \leq r_x, r_y, r_z \leq 1$ and the geometrical center at $r_x = r_y = r_z = 0$. Trajectories within individual cubicles are calculated using basic mathematical formulas and the trajectories through neighboring cubicles are combined to the flight path. The particle velocity is given by $\vec{v} = (v_x, v_y, v_z)$, where the absolute value v stays constant during our simulations, but this can be changed if necessary.

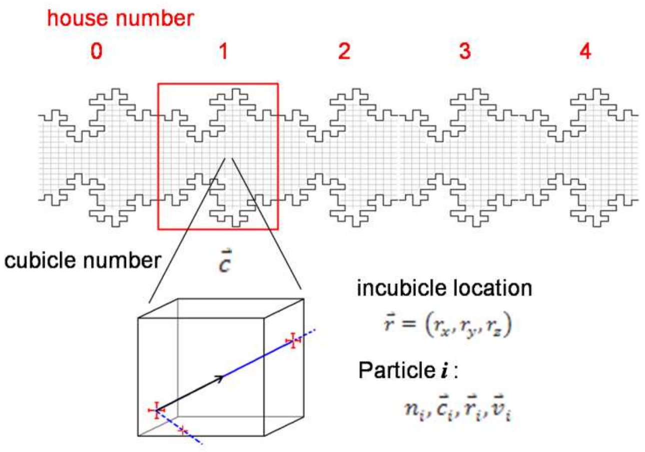

Fig. 2: The framework of the simulation. The pore system is based on a coordinate hierarchy: Empty or filled cubicles determine the local geometry of the pore. They are embedded into a superstructure of house numbers which repeat the local geometry throughout the pore length. The location of a particle i is described by its house number n_i, cubicle coordinate $\vec{c}_i$ and incubicle coordinates $\vec{r}_i$.

Summarizing the cubicles of a certain geometrical region under one 'house number' is very convenient to keep the program flexible for different tasks. When implementing identical periodic structures along the pore, the house number n simply enlarges the system of cubicles by a multiplicative factor n_{max} by repeating the basic structure. In this picture, we compare the cubicles to chambers in identical houses repeating along the pore. The three different coordinate frames of house number n, cubicle coordinate $\vec{c}$ and incubicle coordinate $\vec{r}$, are visualized in Fig. 2. In this case of exactly repeating houses, only the cubicle structure of one house has to be stored, making it possible to generate arbitrary large pores without needing lots of computer memory. If, on the other hand, the houses are not identical, we have to store the information of all houses. In this case the most convenient way in terms of computer memory is, to store the necessary information to re-generate every house “on the fly” at the moment when the particle enters, e.g. to store the seed of the random number generator for each house.

Networks of pores can be implemented by using a three component “house numbering” scheme like already used in the cubicle coordinates. The substructure then is repeated periodically in all three directions

2.2. Structure generation

Every cubicle of the pore system can be generated individually, giving rise to a whole variety of shapes, ranging from simple linear pores with a smooth surface to percolation clusters. In our studies [14-18] we used different iterations of a generalized Koch surface. For this task we applied a random iterator that is shown in Fig. 3(a). Every surface element of a given generation of the pore is divided into a grid of size 4×4. The four grid elements in the middle are alternating augmented or reduced, resulting in 2 cubicles more above und 2 cubicles less below the original surface. Thus the iterator is designed to be volume conserving while doubling the surface with every generation. Figure 3(b)-(d) show pores with different roughnesses that were generated using this iterator.

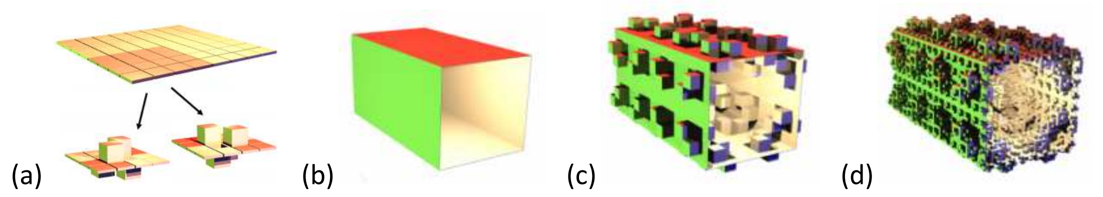

Fig. 3: Examples of pore geometries. (a) Volume conserving and surface doubling random iterator used to generate rough surfaces from (b) the basic smooth pore, (c) and (d) show the 1st and 2nd iteration, respectively.

3. Algorithm

Diffusion of particles through the above described pores is completely based on the underlying cubicle geometry. Only trajectories inside the cubicles are calculated. Whenever a particle leaves such a cubicle, it enters a neighboring one and the next part of the trajectory is determined.

3.1. Particles

The diffusion particles move within the framework of the above described hierarchy of *house numbers* and *cubicles*. For every particle, at least the following data have to be stored: House number n, cubicle coordinate $\vec{c}$, location $\vec{r}$ within the cubicle and velocity $\vec{v}$. The range of the coordinates is $0 \leq n < n_{max}$, $0 \leq c_{x,y,z} < c_{x,y,z,max}$, and $-1 \leq r_{x,y,z} \leq 1$, respectively. The latter coordinate does not start from zero, which optimizes the programming code as shown in the next chapter. It is convenient to keep track of additional parameters, e.g. the time t after the particle was inserted at the insertion location (e.g. $n_0, \vec{c}_0, \vec{r}_0$). Since this algorithm is lattice based instead of time based, it is mandatory to also keep track of the time. An example to extract discrete time steps is shown below.

3.2. From Cubicle to Cubicle

From the exact location and velocity of a particle after j steps ($n_j, \vec{c}_j, \vec{r}_j, \vec{v}_j$), given by the house number, cubicle coordinate and location within the cubicle, it is possible to determine the next location where the particle will hit the surface of this cubicle, using the *intercept theorem* from basic mathematics. On that basis, it is determined, if the particle is reflected or if it enters the adjacent cubicle or if it even enters the next house number.

Depending on entry point and flight direction three walls of the cubicle are possible candidates for the next collision. To determine the actual wall hit by the particle, one has to calculate the time to get from the current location $\vec{r} = (r_x, r_y, r_z)$ to all three (infinite) bounding planes, embedding the three possible wall elements. The minimal time t^* to reach one of those bounding planes is

$$t^* = \min\left(t_x = \frac{\frac{v_x}{|v_x|} - r_x}{v_x}, t_y = \frac{\frac{v_y}{|v_y|} - r_y}{v_y}, t_z = \frac{\frac{v_z}{|v_z|} - r_z}{v_z}\right). \tag{3}$$

The two remaining bounding planes would be crossed outside of the current cubicle, so the time to reach them can be discarded. The new incubicle coordinates $\vec{r}' = (r'_x, r'_y, r'_z)$ of the particle will then be

$$r'_{x,y,z} = r_{x,y,z} + v_{x,y,z} \cdot t^* \qquad \text{or} \qquad \vec{r}' = \vec{r} + \vec{v} \cdot t^*. \tag{4}$$

Accordingly, the simulation time has to be increased by t^*. That component of $\vec{r}'$ which is on the boundary of the present cubicle always has an integer value of ± 1. It can directly be used to determine the adjacent cubicle $\vec{c}'$ on whose boundary the particle is located now. We truncate all three vector components of $\vec{r}$ to integer values and add the corresponding component of $\vec{c}$, yielding the new cubicle coordinate $\vec{c}'$:

$$c'_{x,y,z} = c_{x,y,z} + \mathrm{int}(r'_{x,y,z}). \tag{5}$$

This simple formula includes every possibility of entry point and flight direction by exploiting the properties of the range of the incubicle coordinate $\vec{r}'$.
Including periodic boundary conditions in the calculation of $\vec{c}'$, this equation is changed to

$$c'_{x,y,z} = (c_{x,y,z} + \mathrm{int}(r'_{x,y,z}) + c_{x,y,z,max}) \mathrm{mod} c_{x,y,z,max}. \tag{6}$$

Hereby the house number has to be increased or decreased appropriately.

Now, two cases have to be taken into account. If $\vec{c}'$ is empty, the particle starts the next computation step in the new cubicle $c_{x,y,z} = c'_{x,y,z}$ with the new incubicle coordinates

$$r_{x,y,z} = r'_{x,y,z} - 2 \cdot \mathrm{int}(r'_{x,y,z}). \tag{7}$$

Simultaneously that coordinate that shares the boundary between old and new cubicle inverts its sign, while the others remain unchanged. If $\vec{c}'$ is filled and the particle may not diffuse inside, $\vec{c}'$ is discarded, the particle remains in cubicle $\vec{c}$, and the starting point of the next computation step is the end point of the previous one $r_{x,y,z} = r'_{x,y,z}$. In this case, a new velocity vector through the cubicle has to be chosen according to an appropriate reflection law.

3.3. Reflections

From many possible reflection laws, like specular or glossy reflection, we choose the diffuse reflection governed by *Lambert's cosine law*. A particle emitted thermally from a surface leaves with an angle $\vartheta \in [-\pi/2, \pi/2]$ to the normal component of the surface, where ϑ occurs with a probability $dP(\vartheta, \varphi) \sim \cos\vartheta d\Omega$. Hereby $d\Omega$ is the solid angle, which is equal to $d\vartheta$ in two dimensions and equal to $\sin\vartheta d\vartheta d\varphi$ in three dimensions. This establishes the same luminance in every direction. In some works, the solid angle $d\Omega$ is not clarified and we therefore believe that sometimes the same $dP(\vartheta, \varphi) \sim \cos\vartheta d\vartheta$ behavior is applied in 2D and 3D. In this case, anomalous diffusion effects may arise, that are beyond the scope of this article. These effects can be used to test the correct implementation of Lambert's law.

4. Simulations

Simulations of different properties usually imply different approaches. As examples we show, how transport diffusion and self-diffusion, both in the Knudsen regime, have to be implemented. Additionally we illustrate the correction method that was used in [18] and how to relax a many particle system into a stationary state.

4.1. Self- (or tracer-) diffusion

Calculating self- (or tracer-) diffusion implies a distinct time scale, since for every discrete time t_i the mean square displacement $\langle x^2(t_i) \rangle$ has to be calculated. This is accomplished by linear interpolation within the cubicles. Every calculation step starts at a certain time t at $r_{x,y,z}$ and ends at a time $t' = t + t^*$ at $r'_{x,y,z}$. If $(int)t < (int)t'$ the simulation has advanced more than one time step since the last measurement. In this case the exact location at the discrete times is determined by

$$\vec{r}(\mathrm{int}(t')) = \vec{r}(t) + \left(\vec{r}(t') - \vec{r}(t)\right) \cdot (\mathrm{int}(t') - t)/t^{*}. \tag{8}$$

Depending on the flight path, within the cubicle more than one discrete time step may have passed, which has to be taken into account accordingly. From these discrete measurements of $\vec{r}$, the mean square displacement $\langle x^2(t_i)\rangle$ can easily be averaged, leading to the self diffusion coefficient D_s

This method to extract discrete time steps also has to be used for other measurements, e.g. relaxation into stationarity or the later described correction method for transport diffusion.

4.2. Transport diffusion

In the simulations of the transport diffusion coefficient D_t, a concentration gradient $\partial\vec{c}(x)/\partial x$ along the pore length L is applied with the concentrations $c = c_0 = 1$ on the left hand side and $c = 0$ on the right hand side $(x \geq L)$. All particles start at $x = 0$, which means $n_0 = 0, c_{x,0} = 0, r_{x,0} = -1$, with a randomly distributed velocity $\vec{v}_0$. Then they perform a random walk in 2D or 3D, according to the chosen reflection law. They are absorbed when they hit either the left boundary ($x = 0$, where they entered) or the right boundary at $x = L$ $(n_i = n_{max} - 1, c_{x,i} = c_{x,max}, r_{x,i} = +1)$, which corresponds to Dirichlet boundary conditions. After some relaxation time, this leads to a constant current density $\vec{j}$, as described above.

Since relaxation of a particle flow into a stationary state is usually very time-consuming, as the particle flux has to be monitored throughout the system, we use a method proposed by Evans *et al.* [19]. The particle flux j_x in *x*-direction is derived from the (transmission) probability f_t that a particle starting at the left boundary will leave the pore through the right boundary,

$$j_x = c_0 f_t \langle v_x \rangle \tag{9}$$

where $\langle v_x\rangle$ is the mean velocity in *x*-direction.
Combining this equation with Fick's law yields

$$D_t = -c_0 f_t \langle v_x \rangle \left(\frac{dc}{dx}\right)^{-1}. \tag{10}$$

Usually the concentration gradient is assumed to be constant and equal to

$$\frac{dc}{dx} = -c_0/L. \tag{11}$$

We can now combine both equations to

$$D_t = f_t \langle v_x \rangle L. \tag{12}$$

Accordingly, for calculating f_t, *N* random trajectories are considered that start at $x = 0$ and end when either $x = 0$ or $x = L$ is reached.

4.3. Correction method

A problem arises if the concentration gradient $\partial\vec{c}/\partial x$ within the pore is not well approximated by a constant. This can be due to simple entrance effects, as shown in [18], where the *Enhanced f_t method* was introduced for correction. If deviations from a constant concentration gradient occur, this gradient can only be taken as a starting point.

To obtain the correct value of D_t, we need the concentration $c(x)$ within the pore and the associated concentration gradient $\partial c(x)/\partial x$. For this purpose we need to determine the exact location for each particle after every integer time step. The discrete times can be easily extracted through linear interpolation as shown above. Figure 4 shows the schematics of such a procedure 2D, where $c(x)$ is calculated from the trajectories used to calculate f_t. The particle positions at every time step are indicated by dots. The histogram over all particle positions of the whole simulation is shown at the bottom of the figure.

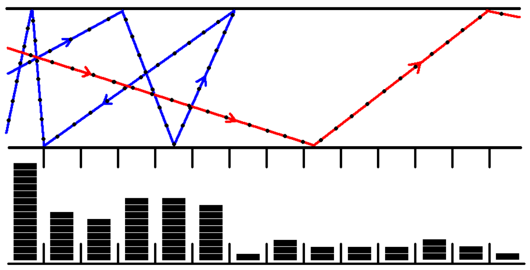

Fig. 4: Schematic for generating the particle histogram and hence c(x) for two particle trajectories in two dimensions. Every time step, the histogram (below) is incremented at the appropriate location.

Normalized this histogram describes the time-averaged concentration $\tilde{c}(x)$, which can be identified as equal to the ensemble averaged concentration $c(x)$ using the ergodic hypothesis. The corresponding concentration gradient then is applied to Fick's law.

4.4. Relaxation into stationarity

To compare the results from Evans' method or its enhanced version, it might be necessary to, indeed, relax the system into a stationary state, where the particle concentration does not depend on the time. This can be very time consuming, since the particle flux has to be monitored throughout the system and deviations have to be compared.

In our simulations we divide the pore length into equal sized bins (e.g. every house number corresponds to one bin) and insert a fixed number of particles every time step until the flux of entering and exiting particles is statistically equal. Again, the particle locations at discrete time steps are determined as shown above. We consider a state as stationary, if the particle concentration between two given discrete time steps at a fixed number of bins differ at maximum by a predetermined threshold value (see [18]). The discrete time steps are determined as described above.

5. Conclusions

We have presented the algorithm that we used in our studies of molecular diffusion in narrow pores in the Knudsen regime in full detail. We laid out the structure of the pore system, including an example to generate Koch shaped rough pores. This framework of the pore system is based on small cubicles wherein the successive steps of free flight or reflection are calculated. These elementary building blocks are embedded into a periodic structure to enlarge the system size. The diffusing particles are described by their velocity and three different coordinate sets, incubicle coordinate $\vec{r}$, cubicle vector $\vec{c}$ and house number n. The algorithm depends only on the system size and is completely independent of the number of walls, making it perfectly adequate for very rough pore surfaces. A discrete time scale can be extracted in run-time during the simulation; this is needed since many applications are based on discrete time intervals. We also showed examples of different tasks the algorithm can accomplish, e.g. how to determine self-diffusion or transport properties and relaxation into a stationary state. Based on these examples it should be possible to enhance the algorithm to serve many other purposes, e.g. the inclusion of particle interactions.

* * *

We gratefully acknowledge stimulating discussions with Prof. M.-O. Coppens and Dr. Ton Dammers, with the International Research Training Group "Diffusion in Porous Materials" and financial support by the "Deutsche Forschungsgemeinschaft" and by the "Fonds der Chemischen Industrie".

The authors are open for any further questions regarding the simulation algorithm.

References

[1] F.J. Keil, R. Krishna, and M.-O. Coppens, Rev. Chem. Eng. **16** (2000) 71.
[2] J. Kärger and D.M. Ruthven, Diffusion in Zeolites and Other Microporous Solids, (Wiley&Sons, New York) 1992.
[3] N.Y. Chen, T.F. Degnan, and C.M. Smith, Molecular Transport and Reaction in Zeolites, (VCH, New York) 1994.
[4] S.B. Santra and B. Sapoval, Phys. Rev. E **57** (1998) 6888.
[5] J.S. Andrade jr., H.F. da Silva, M. Baqui, and B. Sapoval, Phys. Rev. E **68** (2003) 041608.
[6] K. Malek and M.-O. Coppens, Phys. Rev. Lett. **87** (2001) 125505.
[7] G. Ertl, H. Knötzinger, and J. Weitkamp, Handbook of Heterogeneous Catalysis, (Wiley-VCH, Chichester) 1997.
[8] F. Schüth, K.S.W. Sing, and J. Weitkamp, Handbook of Porous Solids, (Wiley-VCH, Weinheim) 2002.
[9] R. Krishna, B. Smit, and S. Calero, Chem. Soc. Rev. **31** (2002) 185.
[10] J. Kärger, R. Valiullin, and S. Vasenkov, New J. Phys. **7** (2005) 1.
[11] S. Anandan and M. Okazaki, Microporous Mesoporous Mater. **87** (2005) 77.
[12] J. Caro, M. Noack, and P. Kölsch, Adsorption **11** (2005) 215.
[13] J.H. Sun, Z. Shan, Th. Maschmeyer, and M.-O. Coppens, Langmuir **19** (2003) 8395.
[14] S. Russ, S. Zschiegner, A. Bunde, and J. Kärger, Adv. In Solid State Physics **45** (2005) 59.
[15] A. Bunde, J. Kärger, S. Russ, and S. Zschiegner, Diffusion Fundamentals **2** (2005) 6.
[16] S. Zschiegner, S. Russ, A. Bunde, and J. Kärger, Diffusion Fundamentals **2** (2005) 42.
[17] S. Russ, S. Zschiegner, A. Bunde, and J. Kärger, Phys, Rev. E **72** (2005) 030101(R).
[18] S. Zschiegner, S. Russ, A. Bunde, and J. Kärger, Europhys. Lett. **78** (2007) 20001.
[19] J.W. Evans, M.H. Abbasi, and A. Sarin, J. Chem. Phys. **72** (1980) 2967.