Theoretical Foundations and Simulation of Diffusion Processes by use of Finite Differences

1 Introduction

Transport phenomena in nature can be distinguished according to whether transport occurs through directed, collective movement of particles or whether it is governed by their random movement. The former phenomena are called convection or advection, while the latter are denoted as diffusion. If we take a look at the atomistic scale, then diffusion is the result of the non-directed motion of individual particles, driven by thermal fluctuations. If there exists a difference in concentration, statistically more particles move from higher to lower concentration than the other way. As a result, inhomogeneous materials (w.r.t. the atom types) can become homogeneous by diffusion. For diffusion to occur, the temperature should be high enough to overcome energy barriers to atomic motion.

In this practical we will take a different approach to diffusion, the phenomenological approach starting with Fick's laws of diffusion and their mathematical consequences: according to Fick's laws, the diffusion flux is proportional to the negative gradient of concentration. It goes from regions of higher concentration to regions of lower concentration.

This practical is divided into three main parts: the derivation and and numerical discretization of the governing partial differential equations, the numerical implementation for the special case of stationary diffusion and the interpretation/verification of the obtained results.

2 Governing Partial Differential Equations and Discretization

2.1 Constitutive equations

Subsequently, Fick's law of diffusion shall be used as a foundation for your numerical setup. As a model system we will assume a 2-dimensional domain on which a certain species is diffusing. This might for example be the diffusion of adsorbed atoms on a surface. Subsequently, you should use the following notations: $\rho(x, y)$ for the concentration of the diffusing species, $\boldsymbol{j}(x, y)$ for the diffusion flux and $\boldsymbol{D}(x, y)$ for the symmetrical tensor of proportionality constants (diffusion constants), which connects the flux to the concentration gradient.

Task 1.1: Derive and explain Fick's second law in 2-D under the assumption that the material has anisotropic diffusion properties. In addition neglect any source or sink terms.

Hint: The change of $\rho(x, y)$ in a control volume is under the given assumptions equal to the net flow $\boldsymbol{j}(x, y)$ across the volume's boundary. This can be expressed in a local form as the so called *continuity equation*

$$\frac{\partial \rho(x,y)}{\partial t} + \nabla \cdot \boldsymbol{j}(x,y) = 0. \tag{1}$$

Classify the resulting PDE (elliptic, parabolic or hyperbolic).

Task 1.2: Show that the general equation for *stationary diffusion* can be written as

Assumtion is anisotropic

$$\partial^2_{xx}\rho(x,y) + P\partial^2_{yy}\rho(x,y) = 0. \tag{2}$$

What is the meaning of x, y and P? How does the tensor of diffusion constants affect the parameter P?

Task 1.3: Show that by transformation of variables equation (2) can also be written as the so called *Laplace equation*

$$\partial^2_{\tilde{x}\tilde{x}}\tilde{\rho}(\tilde{x},\tilde{y}) + \partial^2_{\tilde{y}\tilde{y}}\tilde{\rho}(\tilde{x},\tilde{y}) = 0 \tag{3}$$

Note that stationary heat transfer has the same PDE structure and can be treated in full analogy. To obtain a numerical solution of equation (2) we have to numerically discretize it. We assume that the domain of interest is rectangular in the $x - y$ plane. To discretize the PDE by a spatial finite difference scheme we assume that the geometry is represented by a equidistant mesh of nodes (i, j) at which the function $\rho(x, y)$ will be evaluated. The node distances are h_x and h_y in the x and y directions respectively.

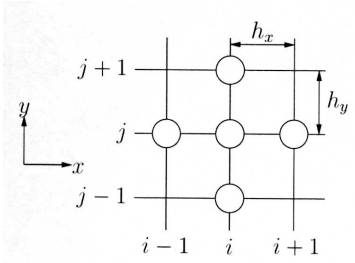

Figure 1: Small section of the discretize $x - y$-domain: circles denote the nodes, that are used for the finite difference scheme at node (i, j)

In the following, we will consider only the isotropic case, i.e. that $D^{xx} = D^{yy} =: D$.

Task 2.1: Discretize eq.(2) by a second order accurate central difference scheme. Use e.g. the notation $\rho_{i+1,j}$ if you refer to the right neighbour of node (i, j) (cf. Fig.1).

Task 2.2: Now we want to investigate a 2-dimensional rectangular region which is discretized by 5x5 nodes. As a simplification we assume that the boundaries of the rectangle are aligned with the principal axes of the diffusion tensor. You should use the equations from Task 2.1 and write the discretized scheme as a linear system of equations. This should be done in the following steps:
a) number your nodes row-wise from top left (nodes 1) to bottom right (node 25). Make a sketch of the mapping from the (i, j) notation to the consecutive node-numbering notation of a general system with NxM nodes. This should show how the 2D-indices can be mapped to a vector notation.
b) Sketch the *inner* part of a coefficient matrix, which should be used in the following form:

$$\begin{bmatrix} c_{1,1} & \cdots & \\ \vdots & \ddots & \vdots \\ & \cdots & c_{n,m} \end{bmatrix} \begin{bmatrix} \rho_1 \\ \vdots \\ \rho_m \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \tag{4}$$

What are the difficulties in the boundary regions?

Hint: The coefficients can be obtained as sum of the factors for the $\rho_{k,l}$ from the equation that you derived in Task 2.1.

2.2 Boundary conditions

So far the system does not consider boundary conditions. Now we assume that the top and lower boundary of the system have the temperature prescribed through Dirichlet boundary conditions ($\bar{\rho}_{\text{upper}}$ and $\bar{\rho}_{\text{lower}}$). Dirichlet boundary conditions are straightforward to apply: the nodes at which they are applied (capital letters) are no longer unknowns in the system, hence the new counting in Fig. 2.
For the left and right boundary we assume von Neumann boundary conditions where the diffusion flux $\bar{j}_{\text{left}}$ and $\bar{j}_{\text{right}}$ is prescribed, see Fig.2. We then can apply boundary conditions to the system by use of Fick's law together with a finite difference schemes, for expressing unknown nodes in terms of known nodal values in eq.(2). Fick's law (assuming isotropic diffusion, with diffusion constant D) takes the form

$$\frac{\partial \rho(x,y)}{\partial x} = -\frac{\bar{j}}{D} \tag{5}$$

with $\bar{j} = \bar{j}_{\text{left}}$ and $\bar{j} = \bar{j}_{\text{right}}$ for the flux values at the right and left boundary. As a convention we define $\vec{n}$ as the outward normal of the system (cf. Fig. 2), so that the fluxes at the boundaries are defined by

$$\vec{j} = j_{\text{left}} \vec{n}_{\text{left}}, \tag{6}$$

$$\vec{j} = j_{\text{right}} \vec{n}_{\text{right}}. \tag{7}$$

It follows, that $\bar{j}_{\text{left/right}}$ is negative, if the flux is directed into the domain.

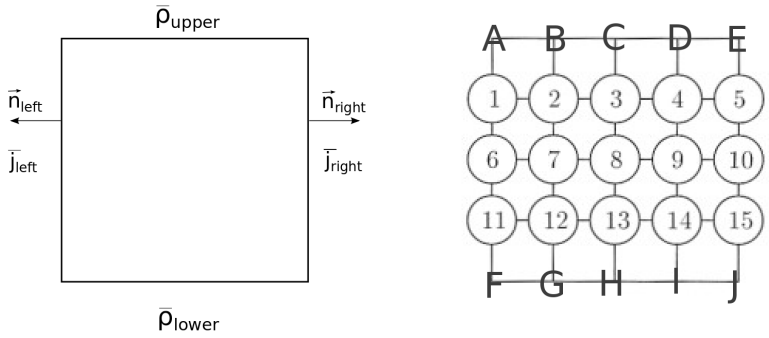

Figure 2: System and discretization.

Task 3.1: Dirichlet boundary conditions
Which nodes are required to evaluate the governing PDE at node 3? Derive the discrete (linear) equation for this node and use that ρ_C is known. Sort the variables such that all known variables are on the right hand of the equation.

Task 3.2: Neumann boundary conditions
Discretize Fick's equation by finite differences and use this linear equation to express unknown 'ghost nodes' at the boundary and consider the direction of flux. How does the resulting equation for node 10 look like? Again, sort all known values on the right hand side.

2.3 Assembling the system

The resulting discretized system with boundary conditions still has the form of eq. (4) derived in Task 2.2. It is a system of linear equations

$$\boldsymbol{A} \cdot x = b, \tag{8}$$

where x is the vector of unknowns, b is a vector with known values and $\boldsymbol{A}$ is a known coefficient matrix. To be solvable we should have obtained now as many equations as the number of unknowns is. The next steps are (i) assemble the coefficient matrix, (ii) assemble the vector on the r.h.s. and (iii) solve the linear system with a direct or iterative solver. Obviously, this is your next task:

Task 4.1: Assemble the the linear system of equations.
Use the template below to fill in all coefficients and values. The two vectors on the right hand side are for ρ and j. Which columns correspond to upper/lower/left/right boundaries, which correspond to internal nodes?

Task 4.2: What are numerical properties of the matrix (e.g. what is its structure)? Which parameters influence the shape of the coefficient matrix?

$$\begin{bmatrix} & & & & & & & & & & & & & & \\ & & & & & & & & & & & & & & \\ & & & & & & & & & & & & & & \\ & & & & & & & & & & & & & & \\ & & & & & & & & & & & & & & \\ & & & & & & & & & & & & & & \\ & & & & & & & & & & & & & & \\ & & & & & & & & & & & & & & \\ & & & & & & & & & & & & & & \\ & & & & & & & & & & & & & & \\ & & & & & & & & & & & & & & \\ & & & & & & & & & & & & & & \\ & & & & & & & & & & & & & & \\ & & & & & & & & & & & & & & \\ & & & & & & & & & & & & & & \end{bmatrix} \begin{bmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \\ \rho_4 \\ \rho_5 \\ \rho_6 \\ \rho_7 \\ \rho_8 \\ \rho_9 \\ \rho_{10} \\ \rho_{11} \\ \rho_{12} \\ \rho_{13} \\ \rho_{14} \\ \rho_{15} \end{bmatrix} = \begin{bmatrix} \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{bmatrix} - \begin{bmatrix} \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{bmatrix}$$

2.4 Numerical implementation

Task 5.1: Implement the solution of a system with given width l_x, height l_y, number of nodes N_x and N_y and arbitrary given values on the boundary for j and ρ, respectively. You can use either MatLab, Python or C/C++ for this. For assembling the matrix one way is to first assemble the matrix without considering boundaries and only in a second step to alter the respective rows/columns for considering boundary conditions. The reason for this becomes clear after you solved Task 4.1.

Task 5.2: Testproblems

1. Set $j_{left} = j_{right} =, \rho_{upper} = 100\,\mathrm{mol}\cdot\mathrm{m}^{-3}$, $\rho_{lower} = 100\,\mathrm{mol}\cdot\mathrm{m}^{-3}$. Now the concentration should be constant everywhere. Plot the result.
2. Set $\rho_{upper} = 100\,\mathrm{mol}\cdot\mathrm{m}^{-3}$, $\rho_{lower} = 50\,\mathrm{mol}\cdot\mathrm{m}^{-3}$. Now the concentration should be linear. Plot the result.
3. Change the Dirichlet BC to different pairs of concentration: $\{\rho_{upper}, \rho_{lower}\} = \{200, 100\}\,\mathrm{mol}\cdot\mathrm{m}^{-3}, \{50, 30\}\,\mathrm{mol}\cdot\mathrm{m}^{-3}, \{..., ...\}$. How can you set up a single simulation to completely describe the behaviour?
4. Compare the numerical solution of a coarse grid (e.g. 5x5) with the analytical solution. Where is the biggest discrepancy between the solutions and why?
5. What happens when you increase the number of nodes? Demonstrate the behaviour through a convergence (error) plot.
6. How can you speed up your simulation for the case that $j_{left} = j_{right} = 0$ and arbitrary ρ_{upper} and ρ_{lower} without affecting the result?
7. Plot your results with different boundary conditions and an appropriate number of nodes. Interpret and explain the results. In particular explain the behaviour towards the boundaries.