Homework assignment 2

Foundations of the Finite-Element Method

Note: The submission of homework assignments 1 to 4 is mandatory to pass the course. The assignments lead from the mathematical formulation of a model problem to the numerical solution of this problem. They build on each other. You must achieve at least 50 of the achievable points on each assignment.

For all tasks, include the solution steps and intermediate results. The final result alone is not sufficient! We recommend that you use Python for the solution of numerical tasks. Please create a PDF and attach the numerical codes as separate files. Please also include a PDF if you use Jupyter notebooks.

Note: Hints for the numerical tasks:

- Codes should always be as clear and easy to read as possible. Please note in particular:
  - Use names instead of symbols to identify your variables, e.g. `concentration` or `Konzentration` instead of `c`
  - Use suffixes to describe the dimensions of matrices and vectors. For example, `concentration_x` may indicate that the entries in the vector `concentration` depend on the spatial direction `x`.
  - Comment your code sufficiently!

- Use the `numpy` function `numpy.linalg.solve` to solve linear systems of equations. A custom algorithm for solving systems of equations is NOT part of this exercise. The following `numpy` functions may (but do not have to) be helpful : `full`, `linspace`, `zeros`, `eye`, `diag`, `maximum`.
- A full documentation of the `numpy` functions can be found at `https://numpy.org/doc/stable/`.
- To create plots, you can use, for example, `matplotlib.pyplot`. Documentation and examples for matplotlib can be found at `https://matplotlib.org/index.html`.
- Please follow the guidelines for creating figures!

Problem 1 Discretization with finite-elements

7 achievable points

The finite element method is a numerical method for solving partial differential equations (PDEs). In this assignment, we will consider the fundamentals of the finite-element method. As an example, we will again use the stationary diffusion equation in 1D (see homework 1):

$$D\frac{\partial^2 c}{\partial x^2} = f(x) \tag{2.1}$$

As in homework assignment 1, D is the diffusion constant, $c(x)$ is a material concentration and $f(x)$ is a source term. We solve the equation on a a finite domain $[0, L]$.

Task 1. Discretize the second derivative occurring on the left-hand side of Eq. (2.1) using finite elements by first deriving the weak formulation and then reducing the requirements on how often the basis function must be differentiable. Then use the Galerkin ansatz and finally insert the explicit expressions for the selected basis functions. You can initially neglect the grid points at the boundary of the domain. For discretization, use N uniformly distributed grid points. As basis functions, we choose the piecewise linear hat functions, as shown in Fig. 2.1.

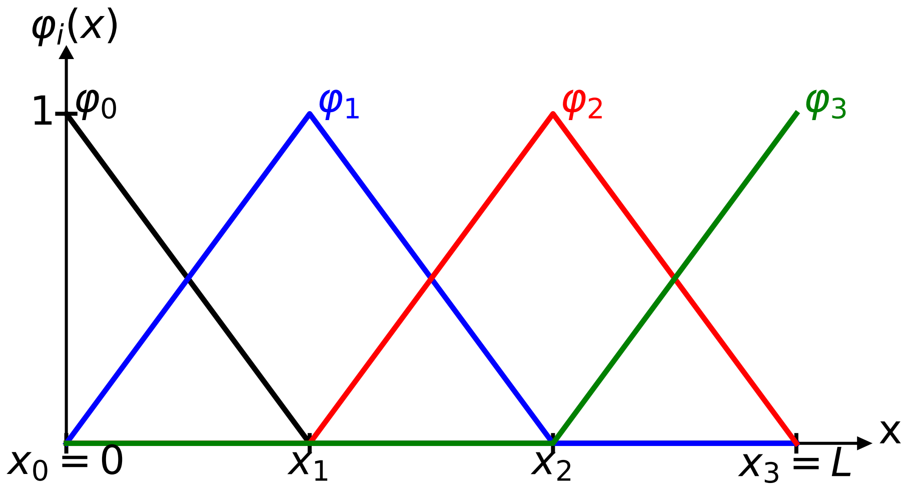

Figure 2.1: "Hat" functions for $N = 4$ equidistant grid points

Note: Remember that we approximate the solution by the series

$$c_N(x) = \sum_{i=1}^{N} a_i \varphi_i(x) \tag{2.2}$$

with the basis functions defined above. The discretized equations are then linear equations for the coefficients a_i.

Task 2. Now discretize the source term, i.e. the term on the right-hand side of Eq. (2.1). Derive the discretized form of the source terms

$$f(x) = -\delta\left(x - \frac{L}{3}\right)\mathrm{m}^{-2}\mathrm{s}^{-1} + \delta\left(x - \frac{2L}{3}\right)\mathrm{m}^{-2}\mathrm{s}^{-1} \tag{2.3}$$

$$f(x) = f_0 \quad \text{where } f_0 \text{ is a constant} \tag{2.4}$$

Note: Please note the following when calculating with δ-distributions as the source term:

- To make it easier to calculate the discretized terms, you can choose the number of grid points N so that the dirac source terms lie on a node, i.e. choose $N = 3N' + 1$ with $N' \in \mathbb{N}$.

- The delta distribution is defined by its filter property: $\int_0^L \mathrm{d}x\, \delta(x - a) f(x) = f(a)$ if $0 < a < L$.

Problem 2 Boundary conditions: Periodicity

10 achievable points

In this problem, you will use finite-elements to solve the diffusion equation with periodic boundary conditions on the domain $[0; L]$, which was solved analytically in homework 1. Use the same discretization and the same basis functions as in Problem 1. Just as in the analytical solution, the source term is $f(x) = -\delta\left(x - \frac{L}{3}\right) \mathrm{m}^{-2}\mathrm{s}^{-1} + \delta\left(x - \frac{2L}{3}\right) \mathrm{m}^{-2}\mathrm{s}^{-1}$.

As we saw in homework 1, the mean value of the solution cannot be determined without further information. In other words, if $c_1(x)$ is a periodic solution of the diffusion equation, then $c_2(x) = c_1(x) + C$ with arbitrary C is also a periodic solution of the diffusion equation. This means that the system of equations that we set up with finite elements cannot be solved unambiguously.

Task 1. To obtain a solvable system of equations, we need a problem with a unique solution. We must therefore supplement the requirement of 'periodic boundary conditions' with a further condition. We choose a given mean value of c_0, i.e.

$$\frac{1}{L} \int_0^L \mathrm{d}x\, c(x) = c_0. \tag{2.5}$$

Equation (2.5) results in an additional equation for the coefficients a_i. Determine this equation.

Task 2. The system of equations for the discrete coefficients a_i is usually written in matrix form:

$$\underline{K} \cdot \vec{a} = \vec{f} \tag{2.6}$$

$\underline{K}$ is then called the system matrix and $\vec{v}$ is called the load vector.

Write down the system matrix $\underline{K}$ for solving the 1D diffusion equation with periodic boundary conditions and a given mean value using linear finite elements for 4 lattice points.

Task 3. Write a (Python) function to solve the diffusion equation in periodic space with linear finite elements. This function should expect as arguments the number of grid points N, the distance between two grid points dx and a vector with the already discretized right-hand side of the problem. We recommend to use the following function signature:

```
def fem_laplace_linear_1d_periodic(nb_grid_pts, dx, rhs_x):
    """
    Function to solve the 1D Laplace equation with
    boundary conditions and an imposed average using a
    regular grid and linear finite elements.

    Arguments
    ---------
    nb_grid_pts: int
        Number of grid points
    dx: float
        Length between two adjacent grid points
    rhs_x: numpy.ndarray(nb_grid_pts) of floats
        Right-hand-side vector

    Returns
    -------
    func_x: numpy.ndarray(nb_grid_pts) of floats
        Solution of the discretized 1D Laplace equation
        at each grid point
    """
```

Create a plot that compares the FEM solution with the analytical solution from homework 1. The plot must show:

- Analytical solution from exercise sheet 1, task 4: $c(x) = -\frac{1}{D}\max(0\mathrm{m}, x - L/3)\mathrm{m}^{-2}\mathrm{s}^{-1} + \frac{1}{D}\max(0\mathrm{m}, x - 2L/3)\mathrm{m}^{-2}\mathrm{s}^{-1} + \frac{1}{3D}\mathrm{m}^{-2}\mathrm{s}^{-1}x + c_0$
- FEM solution for N=4
- FEM solution for N=7

Use the following values for the parameters in the plot:

- $D = 0.8\,\mathrm{m}^2\mathrm{s}^{-1}$
- $L = 1.5\,\mathrm{m}$
- $1/L\int_0^L \mathrm{d}x\, c(x) = 0.3\,\mathrm{m}^{-3}$

Task 4. Comment on what can be observed in the plot. What would you expect if the Dirac impulse did not lie directly on a lattice point? (A calculation is not necessary, a short comment is sufficient.)

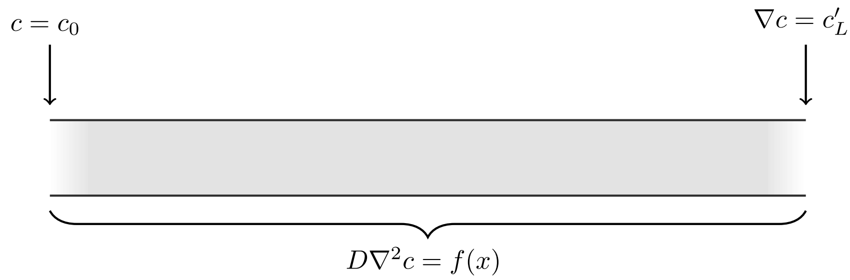

Figure 2.2: Sketch of the simulation domain. The left end is subject to a Dirichlet condition while the right end is subject to a Neumann condition.

Note: Please read the notes at the top of the exercise sheet!

Problem 3 Boundary conditions: Dirichlet and Neumann

12 achievable points

In this task, we consider the diffusion equation with a Dirichlet boundary condition at $x = 0\text{m}$ and a Neumann boundary condition at $x = L$:

$$c(x = 0) = c_0 \quad \text{(Dirichlet)} \tag{2.7}$$

$$\left.\frac{\partial c}{\partial x}\right|_{x=L} = c'_L \quad \text{(Neumann)} \tag{2.8}$$

This corresponds to a system that is connected on one side to an infinite reservoir in which the concentration always remains the same, while on the other side there is a constant particle flux in or out of the system (see Fig. 2.2). For the finite element solution, we choose the same discretization and the same basis functions as in Problem 1.

Task 1. Set up the equations for the coefficients a_i that describe the Dirichlet and Neumann boundary conditions. Set for the source term:
1. $f(x) = -\delta\left(x - \frac{L}{3}\right) \text{m}^{-2}\text{s}^{-1} + \delta\left(x - \frac{2L}{3}\right) \text{m}^{-2}\text{s}^{-1}$
2. $f(x) = f_0$

Task 2. Write a Python function to solve the diffusion equation with a Dirichlet and a Neumann boundary condition using linear finite elements. This function should expect the number of grid points N, the distance between two grid points dx and a vector with the already discretized right-hand side of the problem as arguments. An additional argument should indicate whether the system matrix is returned or not. So please use the interface:

```python
def FEM_Laplace_Linear_1D(nb_grid_pts, dx, rhs_x):
    """
    Function to solve the 1D Laplace equation with a
    Dirichlet boundary condition and a Neumann boundary
    condition using a regular grid and linear finite
    elements.

    Arguments
    ---------
    nb_grid_pts : int
        Number of grid points
    dx : float
        Length between two adjacent grid points
    rhs_x : numpy.ndarray(nb_grid_pts) of floats
        Right-hand-side vector
    return_system_matrix : bool, optional
        True if the system matrix should be returned.
        (Default: False)

    Returns
    -------
    func_x : numpy.ndarray(nb_grid_pts) of floats
        Solution of the discretized problem at each grid
        point
    system_matrix_xx : numpy.ndarray((nb_grid_pts, nb_grid_pts)) of floats
        System matrix of the discretized problem. Is only
        returned if the argument return_system_matrix is
        True.
    """
```

Task 3. Use your function to create two plots showing the following :

- Solution of the diffusion equation with two delta distributions as source term: FEM solution for $N = 4$, $N = 7$ and $N = 10$.
- Solution of the diffusion equation with constant source term: FEM solution for $N = 4$, $N = 7$ and $N = 10$ as well as the analytical solution

Comment briefly on what can be observed in the plots.

Use the following values for the parameters:

- $D = 0.8\,\frac{\mathrm{m}^2}{\mathrm{s}}$
- $L = 3\,\mathrm{m}$
- $c_0 = 2\,\mathrm{m}^{-3}$
- $c'_L = 0.7\,\mathrm{m}^{-4}$
- $f_0 = 0.5\,\mathrm{m}^{-3}\mathrm{s}^{-1}$