Scientific Computing I

Winter Semester 2013 / 2014
Prof. Dr. Beuchler
Bastian Bohn and Alexander Hullmann

Excercise sheet 2.

Closing date **29.10.2013**.

Theoretical exercise 1. (Brownian motion and diffusion [5 points])

We consider the grid $\{j\Delta x : j \in \mathbb{Z}\}$ on the real axis and the points in time $t = 0, \Delta t, 2\Delta t, \ldots$. A particle that is in $x = m\Delta x$ at $t = n\Delta t$ will move to $x = (m+1)\Delta x$ with probability $p = 1/2$ and to $x = (m-1)\Delta x$ with probability $p = 1/2$ in the next time step. At $t = 0$, a single particle is in $x = 0$.

a) Show that the probability of the particle to be in point $x = k\Delta x - (N-k)\Delta x$ at time $t = N\Delta t$ is equal to $\binom{N}{k} \cdot 2^{-N}$.

b) Let $P(y, s; J, t)$ be the conditional probability that the particle is in the interval J at time t given that it was in position y at time s. Compute the limit $N \to \infty$ with $\Delta x \to 0$ and subsequently $\Delta t \to 0$ in oder to show that

$$P(y, s; J, t) = \int_J p(y, s; x, t)dx \qquad (1)$$

with

$$p(y, s; x, t) := \frac{1}{\sqrt{2\pi(t-s)}} \exp\Big(-\frac{(x-y)^2}{2(t-s)}\Big) .$$

Hint: Use the limit theorem of Moivre-Laplace!

c) Now we consider a concentration function $\rho(x, t)$ on the real axis in x at time t. Show that we can deduce from (1) that

$$\rho(x, t) = \int_{-\infty}^{\infty} \int_0^t \rho(y, s)p(y, s; x, t)dsdy .$$

Derive the diffusion equation

$$\rho_t = \frac{1}{2}\rho_{xx} .$$

Theoretical exercise 2. (Lamé's equations [5 bonus points])

Lamé's equations

$$\rho\frac{\partial^2 \vec{u}}{\partial t^2} - \mu\Delta\vec{u} - (\lambda + \mu)\,\mathrm{grad}\,\mathrm{div}\,\vec{u} = \vec{f}(\vec{x}, t) \qquad (2)$$

are characterized by two parameters $\lambda, \mu > 0$. Furthermore, $\vec{f}$ is an external volume force and ρ is the density of Ω. In the stationary case, the above PDE simplifies to

$$-\mu\Delta\vec{u} - (\lambda + \mu)\,\mathrm{grad}\,\mathrm{div}\,\vec{u} = \vec{f}(\vec{x}, t) . \qquad (3)$$

a) Is the instationary case (2) elliptic, parabolic or hyperbolic? Why?

b) Is the stationary case (3) elliptic, parabolic or hyperbolic? Why?

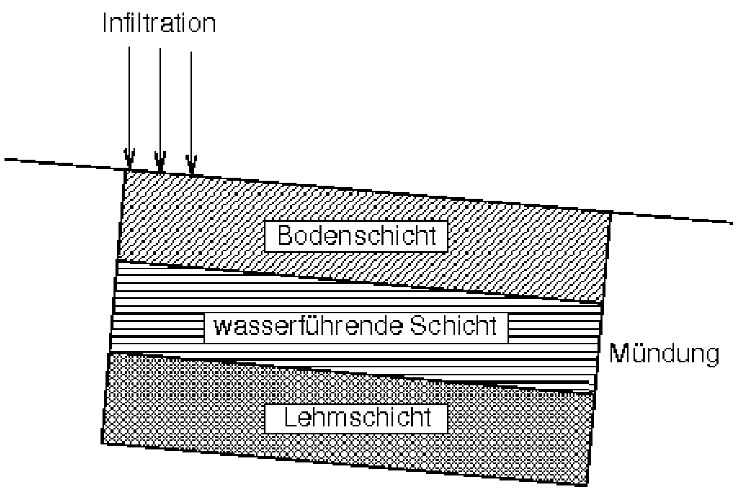

Figure 1: Ground-water infiltration

Theoretical exercise 3. (Modeling of the transport equation [5 points])

Through excessive fertilization phosphate can end up in the ground-water. In general, ground-water flows slowly and its speed (defined as volume that moves through a unit area in a unit time interval) is usually some meters per month. We now assume that the ground-water flow is one-dimensional and uniform. The phosphate will be passively transported without diffusion and not absorbed by the ground. The phosphate concentration $c(x,t)$ of the ground-water is measured in $[kg/m^3]$. Now, at $t > 0$ and $x = 0$, a certain amount Q $[kg/(m^2 \cdot s)]$ of phosphate per unit area and unit time infiltrates the ground-water, see Fig. 1.
Show that the conservation of phosphate results in the equation

$$c_t(x,t) + uc_x(x,t) = 0, \quad x > 0 \ ,$$

where u denotes the flow speed of the ground-water and the situation described above leads to

$$c(0,t) = Q/u, \quad c(x,0) = 0 \ .$$

Programming exercise 1. (Linear and quadratic Lagrange polynomials [10 points])

On last week's exercise sheet you calculated the linear and quadratic Lagrange polynomials ϕ_α and their derivatives. The evaluation and several integration routines for these will be implemented this week. When numbering the basis functions, please be consistent with the numbering of the theoretical exercise on sheet 1. As we begin our indices from 0, the indices from the theoretical exercise from the last sheet all have to be subtracted by 1 for the implementation.
The terms stiffness matrix, mass matrix and load vector appear which will be introduced and defined accurately in the lecture during the semester.

Tasks:

a) [3 points] Implement a class/struct `Basis`. It consists of the following information

- the basis type (i.e. linear Lagrangian basis or quadratic Lagrangian basis; you should use an `enum` or int for this)
- the number of basis functions on the reference triangle (3 for linear Lagrangian basis, 6 for quadratic Lagrangian basis)

Furthermore the following (member) functions have to be implemented:

- `int getNumBasisFunctions()` – returns the number of basis functions
- `int getBasisType()` – returns the basis type (e.g. 0 for linear, 1 for quadratic Lagrangian basis)
- `void setBasisType(int basisType)` – sets the basis type
- `double operator()(int i, double x1, double x2)` – evaluates the `i`-th basis function at the point $x = (\texttt{x1},\texttt{x2})$.
- `void deriv(int i, double x1, double x2, double* d1, double* d2)` – evaluates the derivative of the `i`-th basis function at the point $x = (\texttt{x1},\texttt{x2})$. The pointers `d1` and `d2` contain the first and second coordinates of the result.

b) [7 points] For the following methods you have to calculate several integrals over the reference triangle $\hat{T} = \{(x_1, x_2) \in \mathbb{R}^2 \mid x_1, x_2 \geq 0, x_1 + x_2 \leq 1\}$ or the interval $[0, 1]$ by hand and implement the results. Implement the following member functions:

- `double calcMassMatrixEntry(int i, int j, double factor)` – the method returns
$$\texttt{factor} \cdot \int_{\hat{T}} \phi_{\texttt{i}}(x)\phi_{\texttt{j}}(x)\mathrm{d}x$$
- `double calcLoadVectorEntry(int i, double factor)` – the method returns
$$\texttt{factor} \cdot \int_{\hat{T}} \phi_{\texttt{i}}(x)\mathrm{d}x$$
- `double calcStiffnessMatrixEntry(int i, int j, double** A, double d11, double d22, double factor)` – **This method has to be implemented only for the linear Lagrangian basis**. The matrix `A` passed as a parameter has the size 2×2. The method computes the integral
$$\texttt{factor} \cdot \int_{\hat{T}} (\texttt{A} \cdot \nabla\phi_{\texttt{i}}(x))^T \cdot \begin{pmatrix} \texttt{d11} & 0 \\ 0 & \texttt{d22} \end{pmatrix} \cdot \left(\texttt{A} \cdot \nabla\phi_{\texttt{j}}(x)\right) \mathrm{d}x$$
- `double calcBoundaryIntegral(int i, double factor)` – the method returns the value of the **one-dimensional** integral
$$\texttt{factor} \cdot \int_0^1 \bar{\phi}_{\texttt{i}}(x)\mathrm{d}x$$
where $\bar{\phi}_{\texttt{i}}$ is the `i`-th one-dimensional Lagrange polynomial on the interval $[0, 1]$, i.e.
  - $\bar{\phi}_0(x) := x$
  - $\bar{\phi}_1(x) := 1 - x$

  for the linear Lagrangian basis and
  - $\bar{\phi}_0(x) := (x - 1)(2x - 1)$
  - $\bar{\phi}_1(x) := x(2x - 1)$
  - $\bar{\phi}_2(x) := 4x(1 - x)$

  for the quadratic Lagrangian basis.

Test your implementation (`factor` is 1 for all examples):

- Write a main function which calculates the 3×3 mass matrix (the (i, j)-th entry is the result of the function call with i and j), the 3-dimensional load vector and the two boundary integrals for the linear Lagrangian basis. Calculate the 3×3 stiffness matrix for parameters $\texttt{A} = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$ and $\texttt{d11} = 2$, $\texttt{d22} = 1$.

- Now calculate the 6×6 mass matrix, the 6-dimensional load vector and the three boundary integrals for the quadratic Lagrangian basis.

Feel free to use the incomplete code from the website.