Scientific Computing I

(Wissenschaftliches Rechnen I)

Winter term 2019/20

Priv.-Doz. Dr. Christian Rieger
Fabian Hoppe

UNIVERSITÄT BONN

5th exercise sheet

Submission on November 14, before the lecture

Exercise 1. (3 points)

On the unit interval $\Omega = [0,1]$ we consider the equation

$$-u'' = f \text{ on } \Omega, \qquad u(0) = u(1) = 0.$$

For finite element discretization, we divide $[0,1]$ into n subintervalls $[x_i, x_{i+1}]$ with $x_i := \frac{i}{n}$, $i = 0, \dots, n-1$. On this partition we consider the piecewise linear nodal basis $\{\psi_0, \dots, \psi_n\}$ uniquely defined by $\psi_k(x_i) = \delta_{ik}$ for $i, k = 0, \dots, n$.

Compute the entries of the corresponding stiffness matrix. What do you observe (compare to sheet 1, exercise 1)?

Exercise 2. (2 + 3 + 2 = 7 points)

This exercise is a preparation for the programming exercise below.

In finite element codes the assembly of stiffness and mass matrices is usually done by a loop over all elements. On each element a *local* stiffness and mass matrix is computed and its contribution is added to the respective global matrix. Before implementing piecewise linear finite elements on a 2D triangular mesh in `python` we therefore start with some preliminary calculations concerning the local stiffness and mass matrix:

Let $a_0, a_1, a_2 \in \mathbb{R}^2$ be the corners of an arbitrary non-degenerate triangle T and let $r_0 = (0,0)$, $r_1 = (0,1)$, $r_2 = (1,0)$ be the corners of the reference triangle T_{ref}.

a) Compute the affine map $F_T \colon T_{\mathrm{ref}} \longrightarrow T$ such that $F_T(r_i) = a_i$, $i = 0,1,2$, i.e. compute $B_T \in \mathbb{R}^{2\times 2}$, $c_T \in \mathbb{R}^2$, such that

$$B_T r_i + c_T = a_i, \qquad i = 0,1,2.$$

b) With

$$\varphi_0(x,y) = 1 - x - y, \qquad \varphi_1(x,y) = x, \qquad \varphi_2(x,y) = y$$

we denote the nodal basis on the reference element T_{ref} and with $\varphi_{i,T}$, $i = 0,1,2$, we refer to the corresponding nodal basis on T defined by $\varphi_{i,T} = \varphi_i \circ F_T^{-1}$. The *local stiffness matrix* $\mathbf{A}_T \in \mathbb{R}^{3\times 3}$ and the *local mass matrix* $\mathbf{M}_T \in \mathbb{R}^{3\times 3}$ are defined as follows:

$$\mathbf{A}_{T,ij} := \int_T \nabla\varphi_{i,T} \nabla\varphi_{j,T}, \qquad \mathbf{M}_{T,ij} := \int_T \varphi_{i,T}\varphi_{j,T}.$$

Show that the following formulas hold for their entries:

$$\mathbf{A}_{T,ij} = \frac{1}{2}|\det B| \int_{T_{\mathrm{ref}}} \nabla\varphi_i (B^T B)^{-1} \nabla\varphi_j,$$

$$\mathbf{M}_{T,ij} = |\det B| \begin{cases} \frac{1}{12}, & i = j \\ \frac{1}{24}, & i \neq j \end{cases}.$$

Conclude that the local stiffness matrix $\mathbf{A}_T$ can be computed as

$$\mathbf{A}_T = \frac{1}{2}\left|\det\begin{pmatrix} x_1 - x_0 & x_2 - x_0 \\ y_1 - y_0 & y_2 - y_0 \end{pmatrix}\right| \mathbf{G}\mathbf{G}^T, \qquad \text{with } \mathbf{G} := \begin{pmatrix} 1 & 1 & 1 \\ x_0 & x_1 & x_2 \\ y_0 & y_1 & y_2 \end{pmatrix}^{-1} \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix},$$

where $a_0 = (x_0, y_0)$, $a_1 = (x_1, y_1)$, $a_2 = (x_2, y_2)$ are the coordinates of the corners of T.

c) Derive similiar expressions for the local load vector $\mathbf{f}_T \in \mathbb{R}^3$ defined by

$$\mathbf{f}_{T,i} := \int_T \varphi_{i,T}, \qquad i = 0,1,2.$$

Programming exercise 1. (2 + 2 + 2 + 2 + 2 = 10 points)

In this exercise want to solve the model problem

$$\begin{aligned} -\Delta u &= f && \text{on } \Omega, \\ u &= 0 && \text{on } \Gamma_D \subset \partial\Omega. \end{aligned} \tag{1}$$

on a two dimensional domain $\Omega \subset \mathbb{R}^2$ with piecewise linear finite elements on a triangular mesh. The underlying mesh on Ω will be provided in the following way:

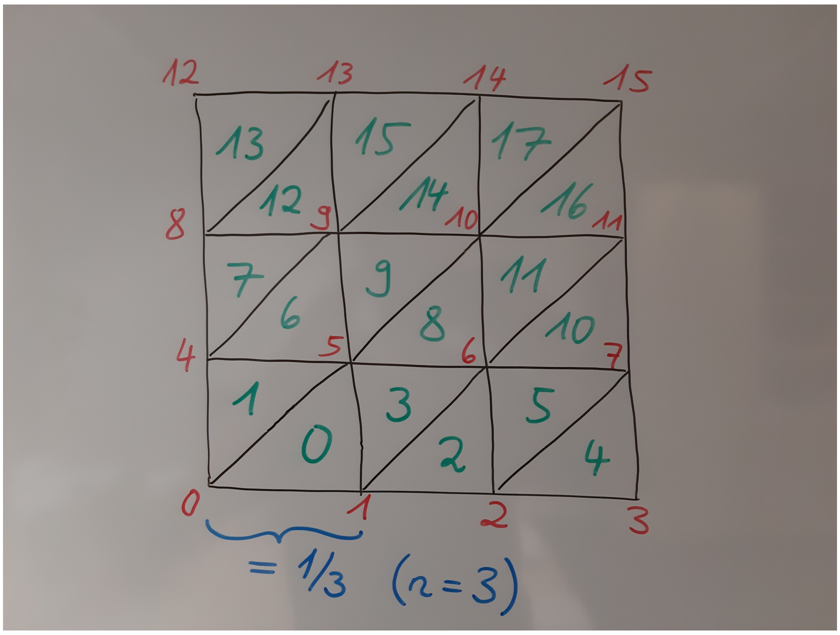
- an array `coordinates` containing x- and y-coordinates (2 columns) of the nodes of the triangulation. The nodes are numbered by the number of the row in which its coordinates are saved.
- an array `elements` containing for each triangle the numbers identifying its corner-nodes (3 columns). The triangles are numbered by the number of the row in which its corners are saved.
- an array `dirichletboundary` containing the numbers of the nodes belonging to the Dirichlet boundary Γ_D

With ψ_i we denote the nodal basis functions of the piecewise linear finite element space on the triangulation. The index i of the basis function is the number of the corresponding node. A finite element function v will be identified with a vector $\mathbf{v}$ via

$$v = \sum_i \mathbf{v}_i \psi_i, \qquad \text{i.e. } \mathbf{v}_i = v(x_i, y_i),$$

where (x_i, y_i) denote the coordinates of the i-th node of the triangulation.

a) Write a function that generates for given $n \in \mathbb{N}$ a criss-cross mesh (see the numbering of nodes (red) and triangles (green) below in the example for $n = 3$) with $(n + 1)^2$ nodes and $2n^2$ elements in the format described above.



The maximal diameter of the elements (triangles) in such a mesh is given by $h = \frac{\sqrt{2}}{n}$.

b) Implement routines `assemble_stiffness_local` and `assemble_mass_local` that assemble the local stiffness matrix $\mathbf{A}_T$ and the local mass matrix $\mathbf{M}_T$ for given triangle T with nodes a_0, a_1, a_2, cf. the exercise above.

Repeat the same procedure for the right hand side vector, i.e. write a routine `assemble_load_local` that assembles the local load vector with entries $\int_T f\varphi_{i,T}$. For the latter use the approximation

$$\int_T f\varphi_{i,T} \approx f(x_T)\int_T \varphi_{i,T},$$

where $x_T := \frac{1}{3}(a_0 + a_1 + a_2)$ is the barycenter of T.

c) Write routines `assemble_stiffness, assemble_mass, assemble_load` that compute the global stiffness and mass matrix and the global load vector, i.e.

$$\mathbf{A} = \left(\int_\Omega \nabla\psi_i\nabla\psi_j\,dx\right)_{i,j}, \qquad \mathbf{M} = \left(\int_\Omega \psi_i\psi_j\,dx\right)_{i,j}, \qquad \mathbf{f} = \left(\int_\Omega f\psi_i\right)_i$$

where the ψ_i denote the global (nodal) basis functions numbered by the number of the nodes. To do so loop over all elements T and utilize your routine for the local assembly. Use the sparse matrix format provided by scipy for $\mathbf{A}$ and $\mathbf{M}$.

d) Write a routine `solve` that computes the finite element approximation to (1), i.e. the vector $\mathbf{v}$ described above. To do so assign zeros to those entries of $\mathbf{v}$ corresponding to nodes on the Dirichlet boundary and solve the linear system that determines the values at the remaining free nodes, i.e. those nodes not on the Dirichlet boundary.

e) On $\Omega = [0,1]^2$ with $f(x,y) = 2\pi^2\sin(\pi x)\sin(\pi y)$ and $\Gamma_D = \partial\Omega$, the true solution of (1) is $u(x,y) = \sin(\pi x)\sin(\pi y)$. Plot your approximate solution and the mesh for mesh size $n = 25$ with help of the routines `triplot` and `tricontourf`. Compare it with the true solution.

For $n \in \{2,4,8,16,32,64\}$ compute the L^2-difference of the finite element approximation to the nodal interpolation of the true solution on the respective mesh. (That is the reason for implementing the mass matrix as well...) Plot this L^2-error with respect to the maximal element-diameter $h = \frac{\sqrt{2}}{n}$ of the respective mesh. Which order of convergence can you observe?

Please submit the programming exercise til November 14, before the lecture, directly to your tutor via Email.