Scientific Computing I

Winter semester 2022/2023
Prof. Dr. Carsten Burstedde
Uta Seidler and Denis Düsseldorf

UNIVERSITÄT BONN

Exercise sheet 10.

Submission (theory) by **19.01.2023**.

Exercise 25. (Scaling)

(6 Points)

Let $K \subset \mathbb{R}^d$ be an open and bounded set and $u \in W^{k,p}(K)$ for some $1 \leq p < \infty$. Furthermore, let $0 < s \in \mathbb{R}$ and $L(x) = sx$ be a scaling operation on $\mathbb{R}^d$ and define $\hat{K} = L(K) = \{L(x) \mid x \in K\}$ as well as new coordinates $\hat{x} = L(x)$. For the function $\hat{u} \in W^{k,p}(\hat{K})$ defined via $\hat{u}(\hat{x}) = u(x)$, show that

$$|\hat{u}|_{W^{k,p}(\hat{K})} = s^{\frac{d}{p}-k} \, |u|_{W^{k,p}(K)} \, .$$

Exercise 26. (Interpolation error)

(6 Points)

For $(x, y) \in \mathbb{R}^2$ set $u(x, y) \coloneqq 1 - x^2$ and let $s > 0$.

a) Let T be the triangle with corners $(-1, 0), (1, 0)$ and $(0, s)$, and let $I_T u$ be the nodal interpolant of u in $P_1(T)$. Show that

$$\|\nabla(u - I_T u)\|_{L^2(T)} \geq \frac{1}{2s} \|D^2 u\|_{L^2(T)}.$$

b) Let T be triangle with corners $(0, 0), (1, 0)$ and $(0, s)$, and let $I_T u$ be the nodal interpolant of u in $P_1(T)$. Show that

$$\|\nabla(u - I_T u)\|_{L^2(T)} = \frac{1}{\sqrt{12}} \|D^2 u\|_{L^2(T)}.$$

Exercise 27. (Piecewise C^1-functions)

(0 Points)

Let $u_1\colon \left[0, \frac{1}{2}\right] \to \mathbb{R}$ and $u_2\colon \left[\frac{1}{2}, 1\right] \to \mathbb{R}$ be continuously differentiable functions. We define

$$u\colon [0, 1] \to \mathbb{R}, \qquad x \mapsto \begin{cases} u_1(x) & \text{if } x \leq 1/2, \\ u_2(x) & \text{if } x > 1/2. \end{cases}$$

Show that $u \in H^1([0, 1])$ if and only if $u_1\left(\frac{1}{2}\right) = u_2\left(\frac{1}{2}\right)$.

Programming exercise 3. (Piecewise linear finite elements)

(12 Points)

The goal of this programming exercise is to solve the Poisson problem in 2D with mixed boundary conditions,

$$\begin{aligned} -\Delta u &= f \quad \text{in } \Omega, \\ u &= g \quad \text{on } \partial\Omega_D, \\ \nabla u \cdot \nu &= 0 \quad \text{on } \partial\Omega_N, \end{aligned} \tag{1}$$

using a piecewise linear finite element method. The domain $\Omega \subset \mathbb{R}^2$ and our degrees of freedom shall be encoded by lookup tables, and we would like to write code only once that can be adapted to new choices for the Dirichlet and Neumann boundaries.

a) Consider a triangulation $\mathcal{T}$ of Ω and store the relevant information for this triangulation $\mathcal{T}$. By this we mean a division of Ω into triangles T_k, $k = 1, \ldots, K$ where each triangle is defined by $M = 3$ vertices $z_{i(k,r)}$, $r = 1, \ldots, M$.

   We encode the topology of the discretized domain by the map from local to global indices. For each element T_k, we can identify any local index $r \in \{1 \ldots, M\}$ with one matching global index $i \in \{1, \ldots, I\}$ where I is the total number of vertices in $\mathcal{T}$. To this end, define the following table in the program:

$$i = i_k(r) = i(k, r) \quad \text{for } k = 1, \ldots, K \text{ and } r = 1, \ldots, M. \tag{2}$$

   The global indices are used to compose the numerical solution using global basis functions,

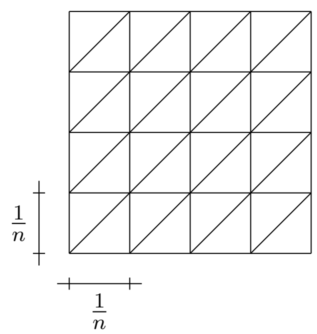
$$u_h(x) = \sum_{i=1}^{I} u_i \psi_i(x). \tag{3}$$

   We will also separate the global degrees of freedom into two nonempty sets. The first set of cardinality $L < I$ contains interior (volume) vertices and vertices on the Neumann boundary and the second set the complementary Dirichlet set. To this end, we need to configure two non-intersecting lists,

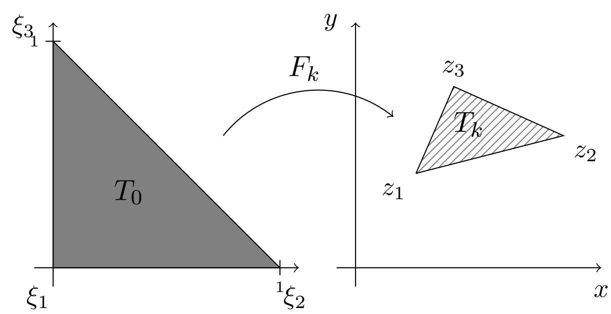
$$i_V = i_V(j) \quad \text{where } j = 1, \ldots, L,\ i_V \in 1, \ldots, I, \tag{4a}$$

$$i_D = i_D(j') \quad \text{where } j' = 1, \ldots, I - L,\ i_D \in 1, \ldots, I. \tag{4b}$$

   Create the lists described in (2) and (4). For example, in the case where $\Omega = (0, 1)^2$, you may consider the discretization where the square is divided into $2n^2$ similar triangles (see left figure).



(Uniform) Triangulation of the unit square $(0, 1)^2$.



Reference element T_0 and transformation F_k onto a triangle T_k.

b) We pull each of the T_k back to the reference element T_0 with vertices ξ_r, $r = 1, \dots, M$. We choose as reference triangle the convex hull of the points $\xi_1 = (0,0)$, $\xi_2 = (1,0)$ and $\xi_3 = (0,1)$.

On this reference triangle, we define nodal linear functions Ψ_r, that is, $\Psi_r(\xi_s) = \delta_{rs}$, $r, s \in \{1, \dots, M\}$. Moreover, we define for each element T_k an affine linear transformation

$$F_k : T_0 \to T_k \tag{5}$$

such that the basis functions ψ_i over this element satisfy

$$\Psi_r \circ F_k^{-1} = \psi_{i(k,r)}|_{T_k}. \tag{6}$$

For each element T_k compute the local mass and stiffness matrices $M_{T_k}, A_{T_k} \in \mathbb{R}^{M \times M}$ as

$$(M_{T_k})_{rs} = \int_{T_k} \psi_{i(k,r)} \psi_{i(k,s)} \, \mathrm{d}x, \tag{7a}$$

$$(A_{T_k})_{rs} = \int_{T_k} \nabla \psi_{i(k,r)} \cdot \nabla \psi_{i(k,s)} \, \mathrm{d}x \tag{7b}$$

for $r, s = 1, \dots, M$, using the transformation theorem to integrate over T_0. Write a routine that carries out the matrix-vector multiplication for the global stiffness matrix $\mathbf{A} \in \mathbb{R}^{I \times I}$ given by

$$\mathbf{A}_{i(k,r), i(k,s)} = \sum_{k=1}^{K} (A_{T_k})_{r,s}, \tag{8}$$

and analogously for the global mass matrix $\mathbf{M}$. Test your implementation by verifying that for the I-vector $\mathbf{e}$ of all ones

$$\mathbf{e}^T \mathbf{M} \mathbf{e} = |\Omega|, \qquad \mathbf{e}^T \mathbf{A} \mathbf{e} = 0. \tag{9}$$

c) Write a routine that returns the local load vector $f_{T_k} \in \mathbb{R}^M$ for each element T_k, $k = 1, \dots, K$ with

$$(f_{T_k})_r = \int_{T_k} f \psi_{i(k,r)} \, \mathrm{d}x \tag{10}$$

for $r = 1, \dots, M$. To this end, transform the integral to the reference triangle T_0 and approximate it with the quadrature

$$\int_{T_0} f(x,y) \, \mathrm{d}x \, \mathrm{d}y \approx \frac{1}{6} \left[f\left(\frac{1}{6}, \frac{1}{6}\right) + f\left(\frac{4}{6}, \frac{1}{6}\right) + f\left(\frac{1}{6}, \frac{4}{6}\right) \right] \tag{11}$$

Combine them into a global load vector $\mathbf{f} \in \mathbb{R}^I$.

d) Use the conjugate gradient method from the last programming exercise to solve the discrete problem. We utilize the projector $\mathbf{P}_V \in \mathbb{R}^{I \times I}$ that sets all Dirichlet entries of a vector to zero and its complement projector $\mathbf{P}_D = \mathrm{id} - \mathbf{P}_V$. With given Dirichlet values $(\mathbf{u}_D)_i = g(z_i)$, $i = i_D(j')$, and $(\mathbf{u}_D)_i = 0$, $i = i_V(j)$, solve the equation

$$(\mathbf{P}_V \mathbf{A} \mathbf{P}_V + \mathbf{P}_D) \mathbf{u} = \mathbf{P}_V (\mathbf{f} - \mathbf{A} \mathbf{u}_D) + \mathbf{u}_D. \tag{12}$$

Plot the solution that you obtained. Try your code for different problems. To this end, you can choose a true solution u and define the right hand side and boundary conditions such that (1) is satisfied. Then use your code to find an approximation u_h to u.

For example, solve the Laplace problem on $\Omega = (0,1)^2$ where the true solution is given by

i)
$$u(x,y) = (1-x)x(1-y)y$$
and zero-Dirichlet boundary conditions on the entire boundary, i.e. $\partial\Omega_D = \partial\Omega$

ii)
$$u(x,y) = 1 + 3x^2y - 2x^3y + 3x^2y^3 - 2y^3x^3$$
with $\partial\Omega_N = \{0\} \times (0,1) \cup \{1\} \times (0,1)$ and $\partial\Omega_D = (0,1) \times \{0\} \cup (0,1) \times \{1\}$

and the values on the boundary and right hand side are chosen accordingly.

e) Now we want to measure the error between the approximate solution u_h and the true solution u. For that, use the quadrature rule in part (c) to evaluate $\|\nabla(u-u_h)\|_{L^2(\Omega)}$. Alternatively, interpolate the true solution u to the finite element space, i.e.

$$I_h u(x) = \sum_{i=1}^{I} \tilde{u}_i \psi_i(x) \qquad \text{with} \qquad \tilde{u}_i = u(z_i)$$

and use the stiffness matrix to compute $\|\nabla(u-u_h)\|_{L^2(\Omega)}$. Compare the two errors.

Moreover, compute the $L^2(\Omega)$-error $\|u-u_h\|_{L^2(\Omega)}$ by either using the quadrature rule or interpolating the true solution to the finite element space and using the mass matrix.

Analyze the convergence of the finite element method by solving the PDE with the uniform triangulation for different n. Plot both errors versus n using a logarithmic scaling for both axis.

Examine the error for the following problems

(i)
$$\begin{aligned}
-\Delta u &= 20\pi^2 \sin(2\pi x)\cos(4\pi y) && \text{in } \Omega = (0,1)^2 \\
u &= \sin(2\pi x) && \text{on } (0,1) \times \{0\}, \\
u &= 0 && \text{on } \{0\} \times (0,1) \cup \{1\} \times (0,1), \\
\nabla u \cdot \nu &= 0 && \text{on } (0,1) \times \{1\},
\end{aligned}$$
such that the true solution is $u(x,y) = \sin(2\pi x)\cos(4\pi y)$.

(ii)
$$\begin{aligned}
-\Delta u &= f \quad \text{in } \Omega = (0,1)^2 \\
u &= g \quad \text{on } \partial\Omega
\end{aligned}$$
such that the true solution is $u(x,y) = \frac{\sqrt{x}}{1+y^2}$.

f) Run your code also for a non-rectangular domain of your choice.

You have until Monday, 23rd January 2023 to work on this programming exercise. The points will mainly be given for the parts (d)-(f) this time.