Scientific Computing I

Winter semester 2015/2016
Prof. Dr. Marc Alexander Schweitzer
Sa Wu

universitätbonn

Exercise sheet 11.

Submission due **Tue, 2016-02-02, before lecture**.

Exercise 40. (P^1 assembly via reference element)

In the lecture we already saw how the stiffness matrix for the Lagrange P^1 triangle is computed for a uniform triangulation of the square, i.e $F_T = F_{(1)}$ equivalent for all triangles, in particular $|\det J_F|$ being the same for each triangle. Now we consider the slightly more general case of triangles T which are not all just translations and rotations of each other.

Let a_0, a_1, a_2 be the corners of an arbitrary non-degenerate (a_i linearly independent) T and $r_0 = (0,0), r_1 = (0,1), r_2 = (1,0)$ the corners of the reference triangle T_{ref}.

a) Explicitly state the affine map $F_T : T_{ref} \to T$ such that $F_T(r_i) = a_i$, i.e. explicitly find $B_T \in \mathbb{R}^{2\times 2}, x_T \in \mathbb{R}^2$ such that

$$B_T r_i + x_T = a_i$$

and give F_T^{-1}.

b) The shape functions on T_{ref} are given by

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

and hence the shape functions on T are given by $L_{i,T} = \varphi_i \circ F_T^{-1}$. Compute $L_{i,T}$ and $\nabla L_{i,T}$ in global coordinates, i.e. expand $L_{i,T}, \nabla L_{i,T}$.

c) Show that

$$\int_T \nabla L_{i,T} \cdot \nabla L_{j,T} = \frac{1}{2} |\det B| \nabla \varphi_i \cdot (B^T B)^{-1} \nabla \varphi_j \ ,$$

$$\int_T L_{i,T} L_{j,T} = |\det B| \begin{cases} \frac{1}{12} \ , & i = j \ , \\ \frac{1}{24} \ , & i \neq j \ . \end{cases}$$

(4 Points)

Exercise 41. (Quadrilateral elements)

Let $Q_{ref} = [0,1]^2$ be the reference quadrilateral with corners

$$r_0 = (0,0) \qquad r_1 = (1,0) \qquad r_2 = (1,1)\,, \qquad r_3 = (0,1)\,.$$

and let $a_0, \dots, a_3$ be the corners of some quadrilateral Q in the the same counter-clockwise orientation. Recall that a map $F : Q_{ref} \to \mathbb{R}^2$ is bilinear, if F admits a representation

$$F(x,y) = a + bx + cy + dxy$$

with constants $a, b, c, d \in \mathbb{R}$. Show that

a) There exists a uniquely determined bilinear map F such that $F(r_i) = a_i$. Explicitly state this map F.

b) F is bijective if and only if Q is convex.

c) F is linear if and only if Q is a parallelogram.

(4 Points)

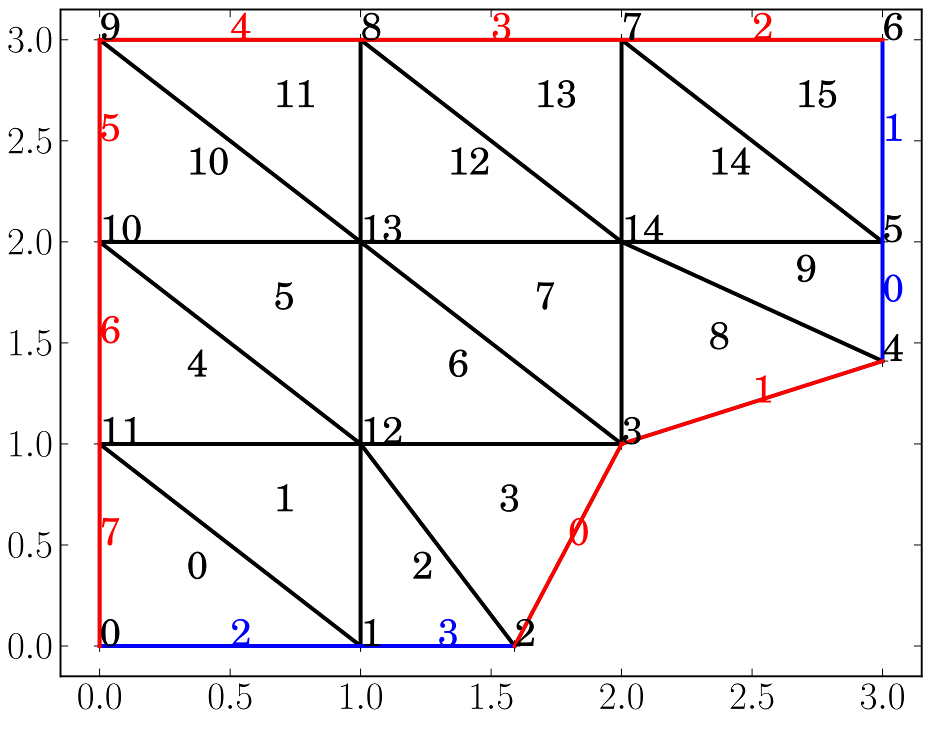

Figure 1: Mesh from example provided in `Python` template with numbering of nodes, elements and edges. Edges part of the Dirichlet boundary Γ_D are colored red. Edges part of the Neumann boundary Γ_N are colored blue.

Programming exercise 8. (A simple 2D FEM code)

Since the basic procedure of implementing a simple FEM code for P^1 Lagrange triangle elements has been already been discussed in the lecture, we attempt a simple implementation for the Poisson problem

$$\begin{aligned} -\Delta u &= f & x &\in \Omega \\ u &= g & x &\in \Gamma_D \subset \partial\Omega \\ \frac{\partial}{\partial \nu} u &= h & x &\in \Gamma_N = \partial\Omega \setminus \Gamma_D \end{aligned}$$

on a bounded Lipschitz domain Ω with a polygonal boundary $\partial\Omega$, Dirichlet boundary conditions on Γ_D and (natural) Neumann boundary conditions on Γ_N. This boundary value problem admits the weak formulation to find $v \in H_D^1(\Omega)$ such that

$$\int_\Omega \nabla v \cdot \nabla w \mathrm{d}x = \int_\Omega f w \mathrm{d}x + \int_{\Gamma_N} h w \mathrm{d}s - \int_\Omega \nabla u_D \cdot \nabla w \mathrm{d}x \,, \qquad w \in H_D^1(\Omega)$$

with the known inhomogeneous decomposition $v = u - u_D$ such that $u_D|_{\Gamma_D} = g$ and $v|_{\Gamma_D} = 0$ and the space

$$H_D^1(\Omega) := \{ w \in H^1(\Omega) : w = 0 \text{ on } \Gamma_D \} \,.$$

A template with function signatures and some plotting commands is provided on the website of the lecture.

The most important aspect in the beginning is the choice of data structures for the mesh/triangulation. In this reduced example we just store everything in lists and arrays in a straight forward fashion, i.e. for a simple example that is provided in the template and depicted in Figure 1, we have

- The vertices are just stored as arrays of the coordinates.

```
# x-coordinate, y-coordinate
coordinates = array([[0,0],[0,1],[1.59,0],[2,1],[3,1.41],[3,2],[3,3], \
                     [2,3],[1,3],[0,3],[0,2],[0,1],[1,1],[1,2],[2,2]])
```

- Triangles as just stored as lists of the indices of the vertices in counterclockwise ordering.

```
# index of node 0, node 1, node 2
triangles = array([[0,1,11],[1,12,11],[1,2,12],[2,3,12],\
                   [11,12,10],[12,13,10],[12,3,13],[3,14,13],\
                   [3,4,14],[4,5,14],[10,13,9],[13,8,9],\
                   [13,14,8],[14,7,8],[14,5,7],[5,6,7]])
```

- Dirichlet and Neumann boundary conditions are given as list of edges to which they are applied to.

```
# edge node left, edge node right
neumann = array([[4,5],[5,6],[0,1],[1,2]])
dirichlet = array([[2,3],[3,4],[6,7],[7,8],[8,9],[9,10],[10,11],[11,0]])
```

Implement a simple FEM code for triangulations given in this format by following the following steps.

a) Write a function `grid_mesh(nn,mm)` that creates the `coordinates`, `triangle`, `neumann` and `dirichlet` arrays for the domain $\Omega = [0,1]^2, \Omega_D = \partial\Omega$ with regular Courant triangles aligned in a grid with nodes (x_i, y_i), $x_i = i\frac{1}{\texttt{nn+1}}, y_j = j\frac{1}{\texttt{mm+1}}$, $i = 0, \ldots, \texttt{nn}+1$, $j = 0, \ldots, \texttt{mm}+1$. Plot the resulting triangulation using the methods provided in the template.

b) Show that the local stiffness matrix M for a single triangle with vertices

$$a_0 = (x_0, y_0)\,, \qquad a_1 = (x_1, y_1)\,, \qquad a_2 = (x_2, y_2)$$

can be computed via

$$M = \frac{1}{2}\det\begin{pmatrix} x_1 - x_0 & x_2 - x_0 \\ y_1 - y_0 & y_2 - y_0 \end{pmatrix} GG^T \qquad G := \begin{pmatrix} 1 & 1 & 1 \\ x_0 & x_1 & x_2 \\ y_0 & y_1 & y_2 \end{pmatrix}^{-1} \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Complete the function `assemble_element(vertices)` that computes this local stiffness matrix using the vertices of one such triangle.

Hint: `vstack` can be used to add lines to a matrix.

c) Assemble the global stiffness matrix `AA` via a loop over all triangles and adding all contributions.

d) The right hand side integrals are approximated using 1 point quadrature rules in the center of gravity (x_S, y_S) for triangles and midpoint (x_E, y_E) of edges, i.e. using a midpoint rule. That is the volume forces on a triangle T yield contributions

$$\int_T f\eta_j \mathrm{d}x \approx \frac{1}{6}\det\begin{pmatrix} x_1 - x_0 & x_2 - x_0 \\ y_1 - y_0 & y_2 - y_0 \end{pmatrix} f(x_S, y_S)$$

and the Neumann edges E with length $|E|$

$$\int_E h\eta_j \mathrm{d}s \approx \frac{|E|}{2} h(x_M, y_M)$$

for test functions η_j.

e) Assemble the right hand side `bb` via looping over all triangles for the volume forces and then over the edges of the Neumann boundary for the Neumann boundary conditions

f) Write a function `free_nodes(coordinates,dirichlet)` that determines the indices of free nodes, i.e. $i : x_i \notin \Gamma_D$. Apply the Dirichlet boundary conditions to the right hand side `bb` via

```
uu[unique(dirichlet)] = gg(coordinates(dirichlet))
bb -= AA.dot(uu)
```

and solving the resulting linear system only for the the reduced system of free nodes.

Hint: array slices.

g) Compute the solution via the reduced system of free nodes for the given triangulation and $f \equiv 1, g \equiv 0, h \equiv 0$.

h) Compute a solution on $\Omega = [0,1]^2$, $\Gamma_D = \partial\Omega$ for $f \equiv -6$, $g(x,y) = 1 + x^2 + 2y^2$ for sufficiently large `mm` and `nn` using the triangulation from a).

(8 Points)