Institut für Numerische Simulation
Rheinische Friedrich-Wilhelms-Universität Bonn

universitätbonn

Exercises to Wissenschaftliches Rechnen I/Scientific Computing I (V3E1/F4E1)

Winter 2016/17

Prof. Dr. Martin Rumpf
Alexander Effland — Stefanie Heyden — Stefan Simon — Sascha Tölkes

Problem sheet 3

Please hand in the solutions of exercise 8 and 9 on Tuesday November 15!
Please prepare the solution of programming task 1 for Thursday November 24!

Exercise 8 **2+4 Points**

(i.) Show that $\Delta \log(|x|) = 0$ if $x \in \mathbb{R}^2 \setminus \{0\}$.

(ii.) Determine the weak solution of the following boundary value problem in $\mathbb{R}^2$:

$$\begin{cases} -\operatorname{div}(a\nabla u) = 0 & \text{in } B_2(0) \setminus B_{\frac{1}{2}}(0)\,, \\ u = 1 & \text{on } \partial B_{\frac{1}{2}}(0)\,, \\ u = 0 & \text{on } \partial B_2(0) \end{cases}$$

with

$$a(x) = \begin{cases} 1 & \text{if } |x| \leq 1\,, \\ 2 & \text{if } |x| > 1\,. \end{cases}$$

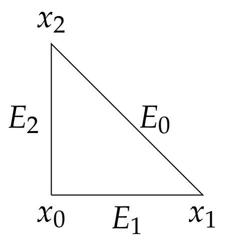
Exercise 9 **4+2+2 Points**

Consider the Argyris finite element $(T, \mathcal{P}_5, \Gamma)$ with the following 21 degrees of freedom

$$\Gamma(p) = (\Gamma_\alpha(P))_{\alpha=1,\dots,21} = \{p(x_i), \partial_k p(x_i), \partial_k \partial_l p(x_i), \partial_{n_i} p(m_i)\}_{i\in\{0,1,2\},k,l\in\{1,2\}}\,,$$

where n_i denotes the outer normal associated with the edge E_i at the edge midpoint m_i.

(i.) Show that any function in $\mathcal{P}_5(T)$ is uniquely determined by an Argyris finite element function on T.



Let $\mathcal{T}_h$ be any triangulation of a polygonal domain $\Omega \subset \mathbb{R}^2$, $\mathcal{V}_h$ be the space of Argyris finite elements on $\mathcal{T}_h$.
(ii.) Show that a function $v \in \mathcal{V}_h$ is continuous.
(iii.) Show that a function $v \in \mathcal{V}_h$ restricted to an edge has continuous normal and tangential derivatives and conclude that v is differentiable.

Programming task 1

Consider Poisson's problem

$$\begin{aligned} -\triangle u &= f \text{ on } \Omega\,, \\ u &= u^{\partial} \text{ on } \partial\Omega\,. \end{aligned} \tag{1}$$

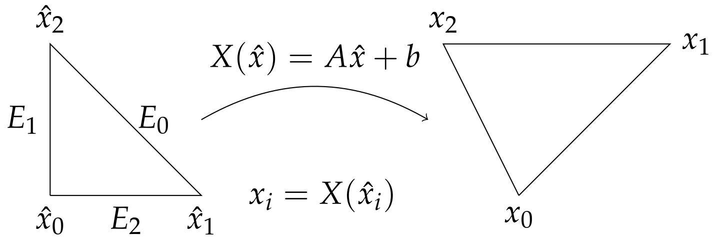
We assume $u^{\partial} \equiv 0$. Then the weak formulation of (1) is

$$\int_{\Omega} \nabla u \cdot \nabla \varphi \,\mathrm{d}y = \int_{\Omega} f \varphi \,\mathrm{d}y \tag{2}$$

for $f \in L^2(\Omega)$.
Consider two different functions f:

1. $f \equiv 1$ and
2. a function f that is obtained by mapping $[0,1]^2$ to $[-1,1]^2$, interpreting that domain as the complex plain and then taking the imaginary part of $z^{\frac{2}{3}}$. For details on the implementation and a simplified formula, we refer to (4) below.

In this programming task, problem (2) should be solved using linear ($\mathcal{P}^1$) finite elements. On the lecture web site, you can download a code framework written in C++ which implements a general two-dimensional triangular mesh for 2D problems and surfaces in 3D using $\mathcal{P}^1$ finite elements. All computations are performed on the unit triangle and then mapped to the respective element using a transformation X.



We denote the unit triangle by $\hat{T}$ and set $\hat{u} = u \circ X$ and $\hat{v} = v \circ X$ as the pullback of u and v. Using the transformation formula we derive

$$\int_T \nabla u \cdot \nabla v \,\mathrm{d}y = \int_{\hat{T}} \sqrt{\det DX^T DX}\, \nabla \hat{u}^T \left(DX^T DX\right)^{-1} \nabla \hat{v} \,\mathrm{d}\hat{y} \tag{3}$$

for a triangle $T \in \mathcal{T}_h$, where $\mathcal{T}_h$ is a triangulation of Ω. The transformation formula also has to be applied to the right hand side integral.
To assemble the system matrix, we use the common scheme of iterating over all elements, assembling local matrices there and then mapping local indices on the element to global node indices.

The following code fragments have to be filled in:

lib/triangleMesh/unityTriangleIntegratorShellFE.h

The local assembly of

$$\int_{\hat{T}} \sum_{k,l} \nabla(\phi_i)_k a_{kl} \nabla(\phi_j)_l \,\mathrm{d}\hat{y}$$

for basis functions ϕ_i and ϕ_j and a given matrix $A = (a_{kl})_{k,l}$.

lib/triangleMesh/unityTriangleIntegratorShellFE.h

The assembly of the global system matrix from local matrices.

course/stiffnessMatrixIntegrator.h

Implement the transformation of $\int_T \nabla u \cdot \nabla v \, \mathrm{d}y$ using formula (3).

course/rhs.h

The assembly of the right hand sides. Implement $f \equiv 1$ and

$$f(y) = \left((-1+2y_1)^2 + (-1+2y_2)^2\right)^{\frac{1}{3}} \sin\left(\tfrac{2}{3}\mathtt{atan2}(1+2y_2, -1+2y_1)\right) . \quad (4)$$

`atan2(`y_2`, `y_1`)` is a C++ function computing the angle in radians.

In the operator structure given, you only have to implement the evaluation of the numerical quadrature of f on a triangle T (given as element `el` in the code).

Further information: The code can be downloaded from the lecture web site, where you can also find input files containing discretized computational domains of varying grid size, additional information and a documentation of the classes needed to solve this task. A configuration file for `CMake`, which can be used to create Makefiles and project files for different IDEs, is provided. To compile, you need to download and install the `Eigen` library.
Results will be saved in the legacy VTK format which can be visualized by a number of tools, e. g. `Paraview`.

- http://numod.ins.uni-bonn.de/teaching/ws16/WissRechI/
- https://cmake.org/
- http://eigen.tuxfamily.org/
- http://www.paraview.org