T U T O R I A L

"Numerical Methods for the Solution of Elliptic Partial Differential Equations"

to the lecture

"Numerics of Elliptic Problems"

Tutorial 12 Tuesday, 26 June 2018, Time: $10^{\underline{15}}$ – $11^{\underline{45}}$, Room: S2 346.

Programming (continued)

L_2-error and H^1-error

62 Write a function

```
double calcElErrorL2 (const Point2D& p0, const Point2D& p1,
                      const Point2D& p2, ScalarField exact,
                      double v0, double v1, double v2);
```

that approximates the element L^2-error $\|v - v_h\|_{L^2(\delta_r)}$, where `exact`$=v$ and $v_h(x_{\delta_r}(\xi)) = \sum_{\alpha\in A} v^{(r,\alpha)}\, p^{(\alpha)}(\xi)$ with `v0`$=v^{(r,1)}$ etc.

Hint: Use the quadrature rule from Exercise 34 to approximate

$$\|v - v_h\|^2_{L^2(\delta_r)} \;=\; \int_{\delta_r} |v(x) - v_h(x)|^2\, dx \;=\; \int_\Delta |v(x_{\delta_r}(\xi)) - v_h(x_{\delta_r}(\xi))|^2\, |\det J_{\delta_r}|\, d\xi$$

63 Write a function

```
double calcElErrorH1 (const Point2D& p0, const Point2D& p1,
                      const Point2D& p2,
                      ScalarField Dx1exact, ScalarField Dx2exact,
                      double v0, double v1, double v2);
```

that approximates the element H^1-error $|Dv - \nabla v_h|_{L^2(\delta_r)}$, where $Dv = \nabla v = (\frac{\partial v}{\partial x_1}, \frac{\partial v}{\partial x_2})^T$, with `Dx1exact`$=\frac{\partial v}{\partial x_1}$, `Dx2exact`$=\frac{\partial v}{\partial x_2}$ and $v_h(x_{\delta_r}(\xi)) = \sum_{\alpha\in A} v^{(r,\alpha)}\, p^{(\alpha)}(\xi)$ with `v0`$=v^{(r,1)}$ etc.

Hint: Use the quadrature rule from Exercise 34 to approximate

$$|v - v_h|^2_{H^1(\delta_r)} \;=\; \int_{\delta_r} |Dv(x) - \nabla_x v_h(x)|^2\, dx \;=\; \int_\Delta |Dv(x_{\delta_r}(\xi)) - J_r^{-T}\nabla_\xi v_h(x_{\delta_r}(\xi))|^2\, |\det J_{\delta_r}|\, d\xi$$

64 Write a function

```
double calcErrorL2 (const Mesh& mesh, ScalarField exact,
                    const Vector& solution);
```

that approximates the global L^2-error $\|v - v_h\|_{L^2(\Omega)}$, where `exact`$=v$ and `solution`$=v_h$.

Hint: use `calcElErrorL2` in a loop over all elements.

Show that $u(x_1, x_2) = \frac{1}{4}\cos(2\pi\, x_1)\cos(4\pi\, x_2)$ is the unique solution of (3.23) (see Tutorial 07, Exercise 39). Compute $\|u - u_h\|_{L^2(\Omega)}$ for each finite element solution u_h from Exercise 39 for the different meshes.

65 Write a function

```
double calcErrorH1 (const Mesh& mesh, ScalarField exact,
                    ScalarField Dx1exact, ScalarField Dx2exact,
                    const Vector& solution);
```

that approximates the global H^1-error $\|v - v_h\|_{H^1(\Omega)}$, where `exact`$=v$, `Dx1exact`$=\frac{\partial v}{\partial x_1}$, `Dx2exact`$=\frac{\partial v}{\partial x_2}$ and `solution`$=v_h$.

Hint: use `calcElErrorL2` and `calcElErrorH1` in a loop over all elements.

Compute $\|u - u_h\|_{H^1(\Omega)}$ for each finite element solution u_h from Exercise 39 for the different meshes.

The CHIP-Problem

Recall the CHIP-Problem from the lecture (T08a, T08b, T09)!

66 Prepare the initial mesh for the CHIP problem as proposed on T09 in your mesh-format, taking care of the appropriate boundary conditions.

Hint: If possible use symmetric reduction.

67 Modify your functions from 33, 35 and 36, such that you can assemble the stiffness matrix `K` according to the bilinear form

$$a(u,v) \;=\; \int_\Omega \lambda(x)\nabla u(x)\cdot\nabla v(x) + a(x)u(x)\,v(x)\,dx,$$

where $\lambda(x)$ and $a(x)$ are given coefficient functions.

68 Solve the finite element system corresponding to the CHIP problem on T08a with the parameter setting of T08b for the initial mesh of 66. Solve the same system for uniformly refined meshes with $h/h_0 = 2, 3, 8, 16$ and visualize the solution.

Hint: For incorporating the BC, use the following order: First natural BC, than essential BC.

A posteriori error estimates

69⋆ Implement the residual error estimator for the CHIP-problem as derived in Exercise 61.

70⋆ Compute the residual error for the CHIP-problem for uniformly refined meshes with $h/h_0 = 2, 3, 8, 16$ and visualize the error on each element!