

# 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 11**    Tuesday, 7 June 2011, Time: 10<sup>15</sup> – 11<sup>45</sup>, Room: SR / T 642.

### Programming (continued)

#### $L_2$ -error

**50** 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  $\mathbf{exact}=v$  and  $v_h(x_{\delta_r}(\xi)) = \sum_{\alpha \in A} v^{(r,\alpha)} p^{(\alpha)}(\xi)$  with  $\mathbf{v0}=v^{(r,1)}$  etc.

*Hint:* Use the quadrature rule from Exercise **36** to approximate

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

**51** 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  $\mathbf{exact}=v$  and  $\mathbf{solution}=v_h$ .

*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 (3.21) (see Tutorial 08, Exercise **41**). Compute  $\|u - u_h\|_{L^2(\Omega)}$  for each the finite element solution  $u_h$  from Exercise **41** for the different meshes.

## Incorporating boundary conditions

Consider the Neumann boundary value problem

$$\begin{aligned} -\Delta u(x) + u &= f(x) & \text{for } x \in \Omega := (0, 1)^2, \\ \frac{\partial u}{\partial n}(x) &= g(x) & \text{for } x \in \Gamma_N := \partial\Omega. \end{aligned}$$

The associated variational formulation is to find  $u \in V_0 := H^1(\Omega)$  such that

$$\int_{\Omega} \nabla u(x) \cdot \nabla v(x) + u(x) v(x) dx = \int_{\Omega} f(x) v(x) dx + \int_{\Gamma_N} g(x) v(x) ds \quad \forall v \in V_0. \quad (3.33)$$

**52** Let  $e \subset \Gamma_N$  be an element edge on the Neumann boundary with the two endpoints  $x^{(e,1)}$  and  $x^{(e,2)}$  and set  $h_e := |x^{(e,2)} - x^{(e,1)}|$ . Let us denote the two functions on the reference edge by  $p^{(1)}(\xi) = 1 - \xi$  and  $p^{(2)}(\xi) = \xi$ .

Write a function

```
void calcNeumannElVec (const Point2D& p0, const Point2D& p1,
                      ScalarField g, Vec<2>& elVec);
```

to approximate

$$g_e^{(\alpha)} := \int_e g(x) p^{(e,\alpha)}(x) ds \approx \frac{h_e}{2} \left( g(x^{(e,1)}) p^{(\alpha)}(0) + g(x^{(e,2)}) p^{(\alpha)}(1) \right)$$

as above by the trapezoidal rule;  $\text{elVec} \approx (g_e^{(1)}, g_e^{(2)})$ ,  $\text{p0} = x^{(e,1)}$ ,  $\text{p1} = x^{(e,2)}$ , and  $\text{g} = g$ .

**53** Write a function

```
void addNeumannLoadVector (const Mesh& mesh, ScalarField g, Vector& b);
```

which *adds* the contribution corresponding to  $\int_{\Gamma_N} g(x) v(x) ds$  to an (already existing) load vector  $\mathbf{b}$ .

*Hint:* Loop over all segments of the mesh and for those marked as Neumann (use `bcSegments[i] == BC_NEUMANN`) call `calcNeumannElVec`.

**54** Solve the finite element system corresponding to (3.33) with  $f(x_1, x_2) = -2.5 + x_1$  and  $g(x_1, x_2) = 0.5$  for a suitably refined mesh (see exercise [41](#)) and visualize the solution.

Consider the Dirichlet boundary value problem

$$\begin{aligned} -\Delta u(x) &= f(x) & \text{for } x \in \Omega := (0, 1)^2, \\ u(x) &= g & \text{for } x \in \Gamma_D := \partial\Omega. \end{aligned}$$

The associated variational formulation is to find  $u \in V_g := \{u \in H^1(\Omega) : u|_{\Gamma} = g\}$  such that

$$\int_{\Omega} \nabla u(x) \cdot \nabla v(x) dx = \int_{\Omega} f(x) v(x) dx \quad \forall v \in V_g. \quad (3.34)$$

55 Write a function

```
void incorporateHomogeneousDirichletBC (const Mesh& mesh,  
                                         SparseMatrix& K, Vector& b);
```

that incorporates the homogeneous Dirichlet boundary conditions ( $g = 0$ ) into the system matrix  $K$  and the load vector  $\mathbf{b}$ .

*Hint:* Loop over all segments of the mesh and search for those marked as Dirichlet (use `bcSegments[i] == BC_DIRICHLET`). For each such vertex with index  $i$  it sets all entries in row  $i$  and column  $i$  of  $K$  to zero and  $K_{i,i} = 1$ ,  $b_i = 0$ .

56 Solve the finite element system corresponding to (3.34) with  $f(x_1, x_2) = 20\pi^2 \sin(2\pi x_1) \sin(4\pi x_2)$  for a suitably refined mesh (see exercise 41) and visualize the solution.

57 Write a function

```
void incorporateInhomogeneousDirichletBC (const Mesh& mesh,  
                                           const Vector& ug, SparseMatrix& K, Vector& b);
```

that incorporates the inhomogeneous Dirichlet boundary conditions  $\mathbf{ug}$  into the system matrix  $K$  and the load vector  $\mathbf{b}$ . Here  $\mathbf{ug}$  is a vector of the same size as  $\mathbf{b}$  carrying the prescribed Dirichlet values (other values are ignored).

*Hint:* Ensure that the entries in  $\mathbf{ug}$ , that do not correspond to Dirichlet values are set to zero. The modification of the load vector  $\mathbf{b}$  can be done by

$$\mathbf{b}[i] = \begin{cases} \mathbf{ug}[i], & i \text{ corresponds to Dirichlet node} \\ \mathbf{b}[i] - (\mathbf{K} * \mathbf{ug})[i], & \text{else} \end{cases}$$

After that, in order to modify  $K$ , proceed as in Exercise 55.

58 Solve the finite element system corresponding to (3.34) with  $f(x_1, x_2) = 20\pi^2 \sin(2\pi x_1) \sin(4\pi x_2)$  and  $g(x_1, x_2)$  given by

$$g(x_1, x_2) = \begin{cases} 0, & x_2 = 1 \vee x_1 = 1 \\ (1 - x_1), & x_2 = 0 \\ (1 - x_2), & x_1 = 0 \end{cases}$$

for a suitably refined mesh (see exercise 41) and visualize the solution.