

# 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 9**

Tuesday, 5 June 2012, Time: 10<sup>15</sup> – 11<sup>45</sup>, Room: S2 / 219.

## Programming (continued)

### 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

$$\int_{\Omega} \nabla u(x) \cdot \nabla v(x) + u(x) v(x) dx = \int_{\Omega} f(x) v(x) dx + \int_{\Gamma_N} g(x) v(x) ds \quad \forall v \in V_0. \quad (1.10)$$

- [42] 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  $\text{g} = g$ .

- [43] 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 `b`.

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

- [44] Solve the finite element system corresponding to (1.10) 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

$$\int_{\Omega} \nabla u(x) \cdot \nabla v(x) dx = \int_{\Omega} f(x) v(x) dx \quad \forall v \in V_g. \quad (1.11)$$

- [45] 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 `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  $K_{i,i} = 1$ ,  $b_i = 0$ .

- [46] Solve the finite element system corresponding to (1.11) 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.

- [47] Write a function

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

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

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

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

After that, in order to modify `K`, proceed as in Exercise [55].

- [48] Solve the finite element system corresponding to (1.11) 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.

Let's consider Robin boundary conditions of the type

$$\frac{\partial u}{\partial N} := \lambda \frac{\partial u}{\partial n} = \kappa(u_0 - u) = g_3 - \kappa u.$$

for given  $\lambda$ ,  $\kappa$  and  $u_0$  and the normal derivative  $n$ .

- [49] Let  $e \subset \Gamma_R$  be element edges on the Robin boundary with the two endpoints  $x^{(e,1)}$  and  $x^{(e,2)}$ . Let the reference edge be  $\Delta = (0, 1)$  with the corresponding nodal basis functions  $p^{(0)}(\xi) = 1 - \xi$  and  $p^{(1)}(\xi) = \xi$ . Write a function

```
void calcRobinElMat (const Vec<2>& x0, const Vec<2>& x1,
                    ScalarField kappa, Mat<2, 2>& elMat);
```

that computes the element Robin matrix  $K$

$$K_{\alpha\beta}^e = \int_e \kappa(x) p^{(e,\alpha)}(x) p^{(e,\beta)}(x) dx = \int_{\Delta} \kappa(x_e(\xi)) p^{(\alpha)}(\xi) p^{(\beta)}(\xi) \det(J_e) d\xi$$

using the quadrature rule on  $\Delta = (0, 1)$  given by

$$\int_{\Delta} g(\xi) d\xi \approx \frac{1}{6} [g(0) + 4g(0.5) + g(1)].$$

Show that this quadrature rule is exact for  $g \in P_3$ .

*Hint:* In order to get  $x_e(\xi)$ , implement a class modelling the affine linear transformation for edges, i.e. in 1D (compare [33], [34] and NumPDE-Tutorial).

**50** Write a function

```
void incorporateRobinBC (const Mesh& mesh, ScalarField kappa,  
                        ScalarField u0, SparseMatrix& K, Vector& b);
```

that incorporates the Robin boundary conditions into the system matrix **K** and the load vector **b**.

*Hint:* Loop over all segments of the mesh and search for those marked as Robin (use `bcSegments[i] == BC.ROBIN`) and reuse the function from the previous Exercise **49** to add the local contributions to the stiffness matrix.

*Hint:* For the contribution corresponding to  $\int_{\Gamma_R} g_3(x) v(x) ds$ , proceed as for the Neumann Boundary (see **42**).