Programming

Continue the program from Tutorial 9. As a sightly different example consider the boundary value problem

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

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

$$\int_\Omega \nabla u(x) \cdot \nabla v(x) + u(x)\, v(x)\, dx \;=\; \int_\Omega f(x)\, v(x)\, dx + \int_{\Gamma_N} g(x)\, v(x)\, ds \qquad \forall v \in V_0\,. \tag{12.1}$$

62 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

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

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

63 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`.

64 Solve the finite element system corresponding to (12.1) 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 **47**) and visualize the solution.