Assume the notations of Tutorial 7 and let $\{p^{(\alpha)} : \alpha \in A\}$ denote the nodal basis on the reference element Δ.

37 Assume that the family $(\mathcal{T}_h)_{h\in\Theta}$ is shape-regular and denote by V_h the corresponding finite element space of the Courant element.

Show that there exist positive constants $\underline{\boldsymbol{c}}_0$ and $\overline{\boldsymbol{c}}_0$ such that

$$\underline{\boldsymbol{c}}_0 \left(\min_{r\in\mathbb{R}_h} h^{(r)} \right)^d (\underline{v}_h, \underline{v}_h)_{\ell^2} \;\leq\; (v_h, v_h)_{L^2(\Omega)} \;\leq\; \overline{\boldsymbol{c}}_0 \left(\max_{r\in\mathbb{R}_h} h^{(r)} \right)^d (\underline{v}_h, \underline{v}_h)_{\ell^2} \qquad \forall v_h \in V_h\,,$$

with $d = 2$.

38 Let the 3×3 matrices G_0 and G_1 be given (according to the lecture) by

$$G_0 \;=\; \left((p^{(\alpha)}, p^{(\beta)})_{L^2(\Delta)} \right)_{\alpha,\beta\in A}, \qquad G_1 \;=\; \left((p^{(\alpha)}, p^{(\beta)})_{H^1(\Delta)} \right)_{\alpha,\beta\in A},$$

Show that there exist positive constants $\underline{c}_G$ and $\overline{c}_G$ such that

$$\underline{c}_G\, (G_0\, \underline{v}, \underline{v}) \;\leq\; (G_1\, \underline{v}, \underline{v}) \;\leq\; \overline{c}_G\, (G_0\, \underline{v}, \underline{v}) \qquad \forall \underline{v} \in \mathbb{R}^3\,.$$

39 Let the family $(\mathcal{T}_{h\in\Theta})$ of subdivisions be shape-regular. Show that there exist positive constants $\underline{\boldsymbol{c}}$ and $\overline{\boldsymbol{c}}$ such that

$$\underline{\boldsymbol{c}}\, (v_h, v_h)_{L^2(\Omega)} \;\leq\; (v_h, v_h)_{H^1(\Omega)} \;\leq\; \overline{\boldsymbol{c}} \left(\min_{r\in\mathbb{R}_h} h^{(r)} \right)^{-2} (v_h, v_h)_{L^2(\Omega)} \qquad \forall v_h \in V_h\,.$$

Hint: Use the Exercise **38** to get the upper bound.

Programming

Download the updated version of `vec.hh` and the new files

- `vector.hh` – a vector class (for vectors of dynamic length)
- `sparsematrix.hh`, `sparsematrix.cc` – a sparse matrix class
- `mesh.hh`, and `mesh.cc` – a 2D triangular mesh

from the tutorial website.
There are also two demos:

- `smdemo.cc` – showing how to work with the sparse matrix and
- `mesh.cc` – showing how to work with the mesh.

Go through these demos and understand what is happening there.

40 Complete the implementation of

```
void Mesh :: getMatrixShape (SparseShape& ss) const;
```

in `mesh.cc`. The routine should give back the matrix pattern of the stiffness matrix corresponding to the mesh.

Hint: An index pair (i, j) is in the matrix pattern if and only if there is an element δ_r containing both vertex i and vertex j. Thus, loop over all elements and for each element, add 9 positions to `ss`.

41 Write a function

```
void assembleStiffnessMatrix (const Mesh& mesh, SparseMatrix& K);
```

that assembles the stiffness matrix `K` according to the bilinear form

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

for `mesh` being the triangulation of Ω.
Hint: Reuse the functions from Tutorial 6, in particular exercises 28 and 30.

42 Write a function

```
void assembleLoadVector (const Mesh& mesh, ScalarField f, Vector& b);
```

that assembles the load vector `b` according to the functional

$$\langle F,\, v\rangle \;=\; \int_\Omega f(x)\, v(x)\, dx$$

for `mesh` being the triangulation of Ω.
Hint: Reuse the function from exercise 29.

All routines should be tested for the two meshes created in `meshdemo.cc`