Programming Exercise 2 (again based on the Alberty/Carstensen/Funken-paper). In exercise 1 we have generated a triangulation of our rectangular domain. Now, we want to define a hat basis for test and trial functions like in the lecture: For the node (also called vertex) p_j, which is not on a dirichlet boundary edge, let H_j be the function which is affine on each triangle of our triangulation and $H_j(p_k) = \delta_{j,k}$ for each node p_k of our triangulation.

Now consider a fixed triangle T in our triangulation. There are exactly 3 hat basis functions who have this triangle inside their support. Call them H_1, H_2 and H_3 for now. We call η_j the restrictions of H_j to T.

In the Uebung, we derived a neat representation of η_j by using the following facts:

The area of a triangle p_1, p_2, p_3 in two dimensions can be calculated as either $\frac{1}{2}\left|det\begin{pmatrix} p_2 - p_1 \\ p_2 - p_1 \end{pmatrix}\right|$ or $\frac{1}{2}\left|det\begin{pmatrix} 1 & p_1 \\ 1 & p_2 \\ 1 & p_3 \end{pmatrix}\right|$. The second representation arises from calculating the area of the tetrahedron with the corners $(1, p_1), (1, p_2), (1, p_3), (0, 0)$ by using the area of the paralleliped spanned by these four vectors.

Calling $A(a, b, c)$ the area of the triangle with the corners a, b and c, the next fact is, that $A_p := A(p, p_2, p_3)$ is linear in the point p. This can be seen by rotating the plane such that p_2 and p_3 lie on the x-axis.

The last fact is that $\frac{A_p}{A(p_1,p_2,p_3)}$ is also linear and 1 at p_1 and 0 at p_2 and p_3.

The representation of $\nabla\eta_1 = \frac{1}{2A}\begin{pmatrix} p_{2,y} - p_{3,y} \\ p_{3,x} - p_{2,x} \end{pmatrix}$ is much easier to find. The gradient must be perpendicular to $p_3 - p_2$ (90° positive rotation) and have the length $\frac{1}{h}$ where h is the height of the triangle when the edge p_2p_3 is considered as the base.

In the lecture we derived a formula for $B_T := (\int \nabla\eta_i \cdot \nabla\eta_j)_{i,j\in\{1,2,3\}}$.

Write a function in octave that calculates the matrix B_T for three given vertices. Then implement a finite element solver to solve

$$\triangle u = -1,\ u(\Gamma_D = \Omega) = 0.$$

on the rectangular grid.

It should load the triangulation data from the files generated in last weeks homework. Please use sparse matrices and visualize your solution.

Note that $sparse(T_K)$ from the lecture is exactly one line in the elements3.dat file.

Useful octave commands are:

```
load foo.dat;
foo(:,1)=[]; %deleting the first column
setdiff(1:n, dirichletBoundaryPoints);
A=sparse(sizex, sizey);
A([1,4,5],[1,4,5])=A(...,...)+localStiffnessMatrix B_T;
u([...])=A([...],[...])\b([...]); %use only subsets of rows and colums
trisurf(triangles,xCoordinates,yCoordinates,solution,'facecolor','interp');
view(...,...);
```