Exercise 3

The goal of this exercise is to make you understand how the stiffness matrix and the right-hand side are assembled in the finite element method. We will focus on the Poisson equation on a square. You can use routines from the book by Larson and Bengzon and from MATLAB's PDE toolbox. In fact the exercise covers mostly Section 4.7 of the Larson–Bengzon book.

1. We consider the Poisson equation

$$\begin{aligned} -\Delta u(x,y) &= f(x,y) = 4, & (x,y) &\in \Omega = (-1,1)^2, \\ u(x,\pm 1) &= 1 - x^2, & -1 &< x < 1, \\ u(\pm 1, y) &= 1 - y^2, & -1 &< y < 1. \end{aligned} \tag{1}$$

   The analytic solution is $u(x,y) = 2 - x^2 - y^2$.

   Triangulate the square $\Omega = (-1,1)^2$. To this end you can generate a geometry array. To make things easier, you can use the function `initmesh` of the PDE toolbox:

   ```
   [p,e,t]=initmesh('square1');
   ```

   You find the file `square1.m` on the web site.

2. Use the functions `MassAssembler2D` and `StiffnessAssembler2D` by Larson and Bengzon to generate mass matrix M and stiffness matrix A. We will need the mass matrix later to deal with the right hand side $f(x,y)$, see eq. (4).

   The matrices generated in this way also contain the degrees of freedom associated with the nodes on the boundary. Of course we know the values of $u(x,y)$ at those points. They are given by the boundary conditions.

3. Now, we need to get rid of the boundary points. In particular we have to implement the equation

$$\sum_{j=1}^{n} \xi_j \int_\Omega \nabla\varphi_j \cdot \nabla\varphi_i \, d\mathbf{x} = \int_\Omega \varphi_i f \, d\mathbf{x} - \sum_{j=n+1}^{n+n_\partial} \int_\Omega g_D(\mathbf{x}_j) \nabla\varphi_j \cdot \nabla\varphi_i \, d\mathbf{x}, \quad 1 \le i \le n, \tag{2}$$

   on slide III-12. The desired solution will be

$$u_h = \sum_{j=1}^{n} \xi_j \varphi_j(x,y) + \sum_{j=n+1}^{n+n_\partial} g_D(x_j, y_j) \varphi_j(x,y). \tag{3}$$

   The edge matrix `e` generated by `initmesh` contains the necessary information. Rows 1 and 2 contain the indices of all boundary nodes (and no others). Extract those indices

Please submit your solution via e-mail to Peter Arbenz (`arbenz@inf.ethz.ch`) by Thursday October 12, 2017. (12:00). Please specify the tag **FEM17** in the subject field.

and employ MATLAB's command `unique`. Let's call this index vector `dirichlet`. This vector contains the indices that appear in the sum on the right of (2) and (3). Note that the indices in `dirichlet` are not contiguous numbers as we assumed in (2)–(3). You obtain the indices of the interior points using MATLAB's command `setdiff`,

```
interior = setdiff([1:np],dirichlet);
```

where `np` is the total number of points as generated by the mesher. (Use the size of the matrix `p` to get at `np`. `p` is generated by `initmesh`.)

4. We compute the domain integral on the right of (2) by interpolating f:

$$\int_\Omega \varphi_i f \, d\mathbf{x} = \sum_{j=1}^{n+n_\partial} f(\mathbf{x}_j) \int_\Omega \varphi_i \, \varphi_j \, d\mathbf{x}, \quad 1 \le i \le n, \tag{4}$$

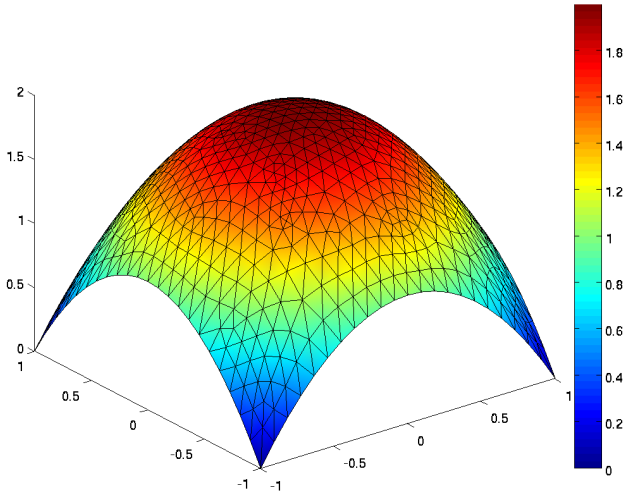
   where we can use the mass matrix M.

5. To solve for the ξ's in (2) we just have to extract the right pieces from the stiffness matrix A.

6. The solution should look like this. Use the command

   ```
   pdeplot(p,e,t,'xydata',u,'zdata',u,'mesh','on','colormap','jet');
   ```

   for this, where vector `u` stores the solution (including boundary points). Of course it looks nicer if you refine the mesh using MATLAB's `refinemesh`.



Please submit your solution via e-mail to Peter Arbenz (`arbenz@inf.ethz.ch`) by Thursday October 12, 2017. (12:00). Please specify the tag **FEM17** in the subject field.