MATH 5620 NUMERICAL ANALYSIS II
OPTIONAL HOMEWORK 6, DUE WED APRIL 21 2010

The goal of this assignment is to use the finite element method with triangular piecewise linear (P1) finite elements to solve the 2D BVP

$$\begin{cases} -\Delta u = f(x,y) & \text{in } \Omega \\ u = 0 & \text{on } \partial\Omega \end{cases} \tag{1}$$

Here the domain is the unit square $\Omega = [0,1]^2$.

1. Solve the problem with $f(x,y) = 2(x(1-x) + y(1-y))$ with true solution being $u_{true}(x,y) = x(1-x)y(1-y)$, using $P1$ triangular finite elements. Include plots with $N \equiv N_x = N_y = 10$ and $N = 30$ (see below for construction of the grid)
2. Do a convergence study by plotting the error in $L^2(\Omega)$ and $H^1(\Omega)$ norms versus h in a log-log scale (you can take for example $N \in \{10, 20, 30, 40, 50\}$ and $h = 1/N$). Estimate the order of convergence. You have to convince me that your numerics agree with the theory i.e. that $\|u - u_h\|_{H^1(\Omega)} = \mathcal{O}(h)$ (first order accurate) and $\|u - u_h\|_{L^2(\Omega)} = \mathcal{O}(h^2)$ (second order accurate). See below for how to estimate these norms. For comparison when $N = 10$ I get: $\|u - u_h\|_{L^2(\Omega)} = 5.294373 \cdot 10^{-4}$ and $\|u - u_h\|_{H^1(\Omega)} = 0.005981$.

1. Implementation details

- The easiest way to obtain the grid and triangulation of Ω is as follows:

```
x=linspace(0,1,Nx+1);   % creates discretization in x direction
y=linspace(0,1,Ny+1);   % creates discretization in y direction
[X,Y]=meshgrid(x,y);    % creates (Nx+1)*(Ny+1) equally spaced points
px=X(:); py=Y(:);       % convert X and Y into vectors
tri = delaunay(px,py); % do actual triangulation
Nnodes = length(px);    % number of nodes (degrees of freedom)
Nelt = size(tri,1);     % number of elements (triangles)
```

- To assemble the stiffness matrix A and the right hand side F, see p106-108 in
  `http://www.math.utah.edu/~fguevara/math5620_s10/na015.pdf`.
  Do this without worrying about the boundary conditions (for now). **Use sparse matrices** otherwise solving the system will take a long time. You only need to initialize `A=sparse(Nnodes,Nnodes);`
- In order to deal with the boundary conditions we first need to find all nodes that are on the boundary. The array `ib` below contains the indices of such nodes (of course this could be done directly, but this method would work even if the grid were unstructured)

```
% find the nodes that are on the boundary
ib=find( abs(px-0)<1e-10 | abs(px-1)<1e-10 ...
       | abs(py-0)<1e-10 | abs(py-1)<1e-10 );
```

- Now to enforce that U is zero at the boundary nodes we can hardwire the Dirichlet boundary conditions in the linear system by modifying the matrix and right hand side as follows:

```
for i=1:length(ib), A(ib(i),:) = 0; A(ib(i),ib(i))=1; end;
F(ib) = 0;
```

- Solve the system $AU = F$ using Matlab's backslash.
- In order to plot your solution vector `U` you can use the following command: `trimesh(tri,px,py,U)`.

- A good approximation of $\|u - u_h\|_{L^2(\Omega)}$ is to look at the L^2 error between the interpolant of u_{true} on the grid and the computed u_h. Thus we need a way of computing the L^2 norm of some function $v_h \in V_h$ (piecewise $P1$) and then we will let $v_h = u - u_h$. If the elements cover Ω we have:

$$\|v_h\|^2_{L^2(\Omega)} = \int_\Omega (v_h(x))^2 dx = \sum_e \int_{K_e} (v_h(x))^2 dx.$$

  Then on each element we know that $v_h \in P1$ therefore $v_h(x) = \sum_{i=1}^3 v_h(x_{tri(e,i)})\phi_i^e(x)$, where the ϕ_i^e are the local basis functions. Thus the contribution of element e to the $L^2(\Omega)$ norm is:

$$\int_{K_e} v_h^2(x) dx = \int_{K_e} \left(\sum_{i=1}^3 v_h(x_{tri(e,i)})\phi_i^e(x) \right)^2 dx = \mathbf{v}_{loc}^T \mathbf{M}_{loc} \mathbf{v}_{loc},$$

  where $\mathbf{M}_{loc}$ is the same matrix given in p107 of the notes and $\mathbf{v}_{loc} = [v_h(x_{tri(e,1)}), v_h(x_{tri(e,2)}), v_h(x_{tri(e,3)})]^T$. **Do not forget to take square root** after having summed the contributions of all elements.
- Use a similar approximation for $\|v_h\|_{H^1(\Omega)}$:

$$\|v_h\|^2_{H^1(\Omega)} = \int_\Omega (v_h(x))^2 + \|\nabla v_h(x)\|^2 dx = \sum_e \int_{K_e} (v_h(x))^2 + \|\nabla v_h(x)\|^2 dx.$$

  It is clear that the first term in the integrand has been computed already by the L^2 norm procedure above. To compute the derivative term we do a similar procedure:

$$|v_h|^2_{H^1(K_e)} = \int_{K_e} \|\nabla v_h(x)\|^2 dx = \mathbf{v}_{loc}^T \mathbf{A}_{loc} \mathbf{v}_{loc},$$

  where $\mathbf{A}_{loc}$ is the same local stiffness matrix that is used in p106 of the notes to construct the stiffness matrix. Again: **do not forget to take square root** after having summed the contributions of all elements. This shows that FEM gives solutions that are meaningful even after differentiation (of course this depends on the polynomial space that is used)