Homework 4 Assignment - Spring 2009

1. Consider the following elliptic PDE

$$-\nabla \cdot (p(\vec{x}) \nabla u) = q(\vec{x}) \quad \text{for} \quad \vec{x} \in \Omega \subset \mathbb{R}^2,$$

$$u(\vec{x}) = 0 \quad \text{when} \quad \vec{x} \in \partial\Omega,$$

with $p(\vec{x}) \geq m > 0$.

(a) Write the weak formulation of this problem in the form,

Find $u \in V$, such that

$$a(u, v) = G(v), \text{ for all } v \in V.$$

What is $a(\cdot,\cdot)$? What is $G(\cdot)$? What is V? Why are u and v in the same space? Is $a(\cdot,\cdot)$ symmetric?

(b) Define the energy functional,

$$J(u) = \frac{1}{2} a(u, u) - G(u),$$

and show that the weak formulation above is equivalent to finding the "critical" or "stationary" point of J, by computing

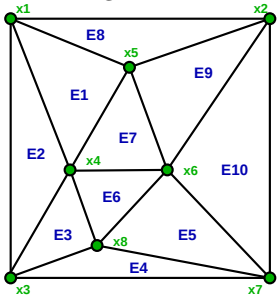
$$\frac{\mathrm{d}}{\mathrm{d}\tau} J(u + \tau v)\big|_{\tau=0} = 0.$$

What is the second directional-derivative of J?

$$\frac{\mathrm{d}^2}{\mathrm{d}\tau^2} J(u + \tau v)\big|_{\tau=0}$$

What can we conclude about the stationary point if $a(\cdot,\cdot)$ is symmetric and coercive (assuming continuity of $a(\cdot,\cdot)$ and $G(\cdot)$ has already been established)? You do not need to prove that $a(\cdot,\cdot)$ is coercive.

(c) Suppose the domain is a disjoint union of M triangles, $E_1, E_2, \cdots, E_M$, with shared vertices. Further suppose that there are a total of N vertices with $\vec{x} = (x, y)$ coordinates given by $\vec{x}_1, \vec{x}_2, \cdots, \vec{x}_N$. The figure below is an example of triangulated domain. Show that the bilinear form, $a(\cdot,\cdot)$, and linear functional, $G(v)$, derived in the previous part can be written as a sum of integrals over each triangle (element).



(d) Consider approximating the solution to the weak formulation by restricting the solution space to a subspace spanned by N basis functions, $\{\phi_1, \cdots, \phi_N\}$. Write the (approximate) solution as a linear combination of these basis functions,

$$u(\vec{x}) = \sum_{i=1}^{N} u_i \phi_i,$$

where $\phi_i \in V$ and $\text{span}(\phi_1, \cdots, \phi_N) \subset V$. Show that the minimizer of the energy, $J(u)$, over the subspace $\text{span}(\phi_1, \cdots, \phi_N)$, solves

$$\sum_{j=1}^{N} u_j a(\phi_j, \phi_i) = G(\phi_i), \quad \forall i = 1 \ldots N.$$

(e) Show that this approximation of u and v reduces the problem of solving the weak formulation to solving a linear system of equations of the form $A\vec{u} = b$ where $\vec{u}$ contains the unknown coefficients. What are the entries of A and b in terms of $a(\cdot,\cdot)$ and $G(\cdot)$? Show that each entry of A and b can be written as a sum of integrals over triangles, if the domain is triangulated as described before.

(f) Suppose we modified the PDE to include non-homogeneous Dirichlet boundary conditions,

$$-\nabla \cdot (p(\vec{x}) \nabla u) = q(\vec{x}) \quad \text{for} \quad \vec{x} \in \Omega \subset \mathbb{R}^2,$$

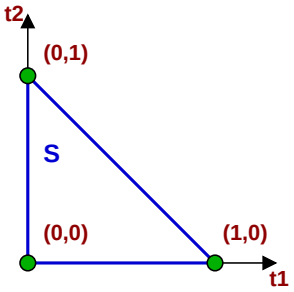
$$u(\vec{x}) = g(\vec{x}) \quad \text{when} \quad \vec{x} \in \partial\Omega,$$

with $p(\vec{x}) \geq m > 0$. Show this yields the problem,

$$a(\hat{u}, v) = \hat{G}(v) \quad \text{for all } v \in V,$$

with $\hat{u} \in V$ and $u := u^0 + \hat{u}$ and $\hat{G}(v) := G(v) - a(u^0, v)$. Here, u^0, is a known fixed function which satisfies the boundary conditions.

2. In this problem we will derive the analytical formulae necessary for implemenation of the finite element method using piecewise linear finite elements on a triangulated domain.



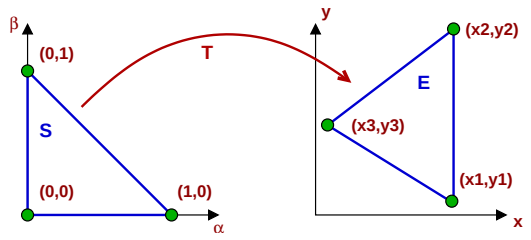
(a) Suppose $\lambda_1(\alpha, \beta)$ is the piecewise linear nodal basis function on the standard element (triangle), with $\lambda_1(0,0) = 1$ and $\lambda_1(1,0) = \lambda_1(0,1) = 0$. Show that when restricted to the standard triangle with vertices $(0,0), (1,0), (0,1) \in \mathbb{R}^2$, this basis function can be written as

$$\lambda_1(\alpha, \beta) = 1 - \alpha - \beta$$

and conclude that the gradient of λ_1 is

$$\nabla\lambda_1 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}.$$

Derive a similar result for λ_2 which satisfies $\lambda_2(1,0) = 1$ and $\lambda_2(0,0) = \lambda_2(0,1) = 0$. Derive a similar result for λ_3 which satisfies $\lambda_3(0,1) = 1$ and $\lambda_3(0,0) = \lambda_3(1,0) = 0$.



(b) Given three vertices of a general triangle, $\vec{x}_1, \vec{x}_2, \vec{x}_3 \in \mathbb{R}^2$, write down an affine transformation, T, that maps the vertices of the "standard triangle" $(0,0), (1,0), (0,1) \in \mathbb{R}^2$ to the "general triangle". Show that the Jacobian (derivative) of this transformation is

$$J_T = \begin{bmatrix} (\vec{x}_2 - \vec{x}_1) & (\vec{x}_3 - \vec{x}_1) \end{bmatrix} \in \mathbb{R}^{2\times 2},$$

where $\vec{x} = (x, y)$.

(c) Suppose ϕ_r is the piecewise linear nodal basis function for a general triangulation with $\phi_r(\vec{x}_r) = 1$ and $\phi_r(\vec{x}_s) = 0$ if $r \neq s$. Using the results of the prior two parts show that when restricted to the "general triangle" that ϕ_r can be written as a composition of λ_r and T^{-1},

$$\phi_r(\vec{x}) = \lambda_r\left(T^{-1}(\vec{x})\right)$$

and the gradient of ϕ_r is

$$\nabla\phi_r|_{\vec{x}} = J_T^{-T}\nabla\lambda_r$$

where $\nabla\lambda_r$ is constant.

(d) Show that

$$\iint_E p(\vec{x})\,\nabla\phi_r(\vec{x})^T\nabla\phi_s(\vec{x})\,d\vec{x} = \left(J_T^{-T}\nabla\lambda_r\right)^T\left(J_T^{-T}\nabla\lambda_s\right)|J_T|\iint_S p(T(\vec{\alpha}))\,d\vec{\alpha}$$

and

$$\iint_E q(\vec{x})\,\phi_r(\vec{x})\,d\vec{x} = |J_T|\iint_S q(T(\vec{\alpha}))\,\lambda_r(\vec{\alpha})\,d\vec{\alpha}$$

using the change of variables formula

$$\iint_E h(\vec{x})\,d\vec{x} = \iint_S h(T(\vec{\alpha}))\,|J_T|\,d\vec{\alpha}$$

Here $\vec{x} = (x, y) \in \mathbb{R}^2$, $\vec{\alpha} = (\alpha, \beta) \in \mathbb{R}^2$, E is the region defined by the general triangle, S is the region defined by the standard triangle, T is the affine transformation defined above, and $|J_T|$ is the determinant of the Jacobian of the transformation. You should use the results from the prior parts. Note that in the case that p and q are constants on E, these integrals can be computed exactly.

3. The problem will lead you through the steps of programing the Galerkin finite element method with piecewise linear elements on a triangulation of the unit square. We will consider the PDE

$$-\nabla\cdot(p(\vec{x})\nabla u) = q(\vec{x}) \quad \text{for} \quad \vec{x} \in \Omega \subset \mathbb{R}^2,$$
$$u(\vec{x}) = g(\vec{x}) \quad \text{when} \quad \vec{x} \in \partial\Omega,$$

for three test cases:

(i) $p = 0.1, q = 0$, on $\Omega = \Omega_1 \cup \Omega_2$
(ii) $p = 0.1, q = 0$, on Ω_1 and $p = 0.1, q = 40$, on Ω_2
(iii) $p = 0.1, q = 0$, on Ω_1 and $p = 40, q = 40$, on Ω_2,

with

$$g(x,y) = \begin{cases} 4x(1-x) & y = 0, \\ 0 & \text{otherwise}, \end{cases}$$

and unit square domain, $\Omega = (0,1)\times(0,1)$. The subdomains, which are defined in the supplied data files, are unions of triangular regions.

(a) Write a routine that reads the files "elems.dat" and "points.dat" into two arrays. The "points" array is of dimension $N \times 2$ where N is the number of vertices in the triangulation. The "elems" array is of dimension $M \times 4$ where M is the number of triangles. It contains the vertex numbers of each triangle in the first three columns and the fourth column indicates the subdomain number used above. In Matlab you can use the "load -ascii" command.

(b) Write a routine that takes as input three vertices of a general triangle and value of p and q on the triangle and returns a 3×3 matrix and a 3×1 vector. The matrix should have as entries the bilinear form restricted to the triangle for all combinations of the three basis functions supported on that triangle:

$$a_{r,s} = \iint_E p(\vec{x})\,\nabla\phi_r^T\nabla\phi_s d\vec{x} \quad r, s \in \{1,2,3\}$$

The vector should have as entries the linear functional restricted to the triangle for all three of the basis functions supported on that triangle:

$$b_r = \iint_E q(\vec{x})\,\phi_r d\vec{x} \quad r \in \{1,2,3\}$$

Use the precomputed analytical calculation of

$$\nabla\lambda_r \quad \text{and} \quad \iint_S \lambda_r\,d\vec{\alpha} \quad \text{and} \quad \iint_S 1\,d\vec{\alpha}.$$

together with the results from the prior exercise.

(c) Write a routine that loops over all triangles (elements), calls the routine defined above, and adds the contribution of each 3×3 matrix and 3×1 vector to a global stiffness matrix, A, and global right-hand side, b. Do not worry about the boundary conditions in this step, just assume all the vertices are interior points (or at least non-Dirichlet points) and we will enforce the boundary conditions in the next step.

(d) Write a routine that loops over all the vertices and checks to see if the vertex is on the boundary. If the vertex, $\vec{x}_i$, is on the boundary, define the ith entry of the "boundary" vector $\vec{u}^0$ to be equal to the boundary conditions, otherwise set it to zero. In other words, $\vec{u}_i^0 = g(\vec{x}_i)$ at Dirichlet boundary points, but zero otherwise.

(e) Modify the global right-hand side, b, to account for nonzero boundary function u^0. Recall that $\hat{G}(v) := G(v) - a(u^0, v)$, so let $b = b - A\vec{u}^0$.

(f) Modify the matrix A to account for the Dirichlet boundary conditions. Write a routine that loops over all the vertices and checks to see if the vertex is on the boundary. If the vertex, $\vec{x}_i$, is on the boundary replace the correponding row and column of the matrix A with the row and column of the identity matrix (i.e., first set to zeros then put one on the diagonal). Similarly, replace the entry in the right-hand side, $\hat{b}$, with zero.

(g) Solve the resulting sparse linear system of equations, $A\hat{u} = b$, using a linear algebra library or Matlab's backslash operator, $\hat{u} = A\backslash\hat{b}$. Consider the three cases listed above. Plot your results using Matlab's patch command, or another method for plotting a solution on an unstructured mesh.

For your reference, the following plotting command was used to produce these three solutions

```
for i=1:length(elems)
  x = points(elems(i,1:3),1);
  y = points(elems(i,1:3),2);
  z = uhat(elems(i,1:3),1) ...
      + uzero(elems(i,1:3),1);
  patch(x,y,z,sum(z)/3);
end
```

Plots may be initially flat, but if you turn on rotation you can get the three dimensional view.