Homework 7. Due April 2

1. **6 pts** Consider the following BVP in 1D:

$$-u_{xx} = f(x), \quad 0 < x < 1, \quad u(0) = u(1) = 1.$$

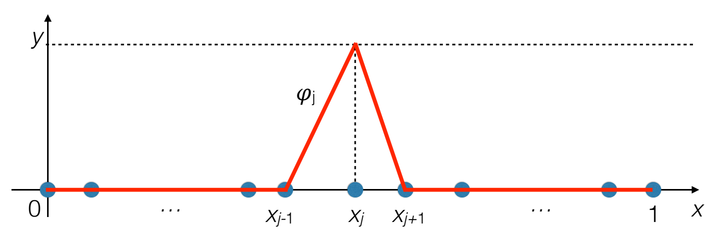
Work out all steps of the FEM on it.

(a) Let $w(x)$ be a twice continuously differentiable function on $(0,1)$ such that $w(0) = w(1) = 0$. Use integration by parts to reduce the BVP to an integral equation.

(b) Partition the interval $[0,1]$:

$$0 = x_0 < x_1 < \ldots < x_N < x_{N+1} = 1.$$

Define the basis functions $\phi_i(x)$, $1 \le i \le N$ as shown in the figure below ($\phi_i(x_i) = 0$, $\phi_i(x_j) = 0$, $j \ne i$, ϕ_i is piecewise linear).



$$\phi_i(x) = \begin{cases} 0, & x \le x_{i-1},\ x \ge x_{i+1}, \\ \frac{x - x_{i-1}}{x_i - x_{i-1}}, & x_{i-1} < x \le x_i, \\ \frac{x_{i+1} - x}{x_{i+1} - x_i}, & x_i < x < x_{i+1}. \end{cases}$$

Calculate the stiffness matrix and the load vector.

(c) In what case the FEM solution would coincide with the finite difference solution using the central difference scheme?

2. **(6 pts)**

(a) Let η_1, η_2, and η_3 be linear functions in a triangle T with vertices (x_1, y_1), (x_2, y_2), and (x_3, y_3) such that $\eta_j = 1$ at (x_j, y_j) and $\eta_j = 0$ and the other two vertices of T, $j = 1, 2, 3$. Prove that the matrix G introduced in Section 5 in [1] is equal to

$$G = \begin{bmatrix} \frac{\partial \eta_1}{\partial x} & \frac{\partial \eta_1}{\partial y} \\ \frac{\partial \eta_2}{\partial x} & \frac{\partial \eta_2}{\partial y} \\ \frac{\partial \eta_3}{\partial x} & \frac{\partial \eta_3}{\partial y} \end{bmatrix}. \tag{1}$$

(b) Let u be a finite element solution to some problem. This means that u is continuous, piecewise linear, and linear within each mesh triangle. Let T be a mesh triangle with vertices (x_1, y_1), (x_2, y_2), and (x_3, y_3), and $u(x_j, y_j) = u_j$, $j = 1, 2, 3$. Suppose that we need to compute the gradient of u. Find an exact expression for ∇u within the mesh triangle in terms of (x_j, y_j) and u_j, $j = 1, 2, 3$.

3. **(6 pts)** *You can use Python or Matlab. You you might find helpful the codes MyFEMcat.m or MyFEMcat.ipynb.*

Consider the problem of finding the density of the electric current in a thin square plate Ω made out of two metals with different conductivities $a = a_1$ and $a = a_2$ as shown in Fig. 1. The voltage $u = 0$ at the left side of the square, and $u = 1$ on the right side. Use the finite element method (FEM) to find the voltage and the density of the electric current inside the plate. Plot the voltage and the absolute value of the current density. Consider two cases: (a): $a_1 = 1.2$ and $a_2 = 1$; (b): $a_1 = 0.8$ and $a_2 = 1$. Comment on the distribution of the current in the plate. Link your codes to your pdf file.

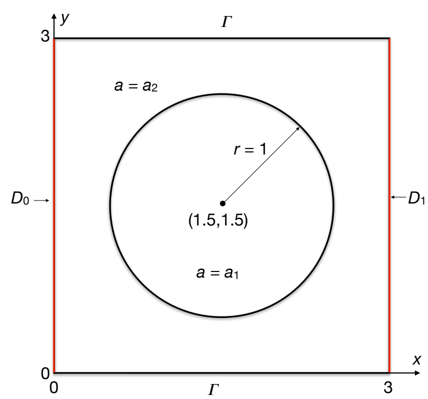


Figure 1: The conducting plate. An illustration to Problem 1.

The voltage is the solution to the following boundary value problem (BVP):

$$\begin{cases} -\nabla \cdot (a(x, y)\nabla u) = 0, & (x, y) \in \Omega := [0, 3]^2, \\ u = 0, & (x, y) \in D_0, \\ u = 1, & (x, y) \in D_1, \\ \frac{\partial u}{\partial n} = 0, & (x, y) \in \Gamma. \end{cases} \tag{2}$$

Note that the solution must satisfy the flux continuity condition: at the boundary separating the regions where $a(x, y) = a_1$ and $a(x, y) = a_2$, the voltage u and the current density $a(x, y)\nabla u$ must be continuous. This condition is automatically satisfied if you generate the mesh so that $a(x, y)$ is continuous in each mesh triangle and then solve the BVP using FEM. This can be easily achieved using e.g. `distmesh2d.m` requiring the points on the boundary separating these two regions to be fixed.

Once you have computed the voltage u, you can find the current density:

$$j = -a(x, y)\nabla u.$$

Use the formula for the gradient found in Problem 1(b). As a result, you have the gradient evaluated at the centers of the mesh triangles. Then compute the absolute value of the density of the current. In order to visualize it using `trisurf` you need to recompute it at the vertices of the mesh triangles. This can be done by averaging over all triangles adjacent to a given vertex:

```
abs_current_verts = zeros(Npts,1);
count_tri = zeros(Npts,1);
for j = 1:Ntri
    abs_current_verts(tri(j,:)) = abs_current_verts(tri(j,:)) ...
         + abs_current_centers(j);
    count_tri(tri(j,:)) = count_tri(tri(j,:)) + 1;
end
abs_current_verts = abs_current_verts./count_tri;
```

References

[1] Jochen Alberty, Carsten Carstensen and Stefan A. Funken, Remarks around 50 lines of Matlab: short finite element implementation