MTH 655, Winter 2017, Assignment 4

In this HW you explore mesh building (Pbm 1), applications (Pbm 2-3), and theory of errors (Pbm 4). Please solve Pbms 1-4.

---

1. Shape of elements

The code demoELEMENT.m demonstrates the affine transformation $T : \hat{K} \to K$, with $T \in \mathbf{P}_1$ of the reference triangle $\hat{K}$ to some "true" element K, with vertices $(x1, y1), (x2, y2), (x3, y3)$.

Produce (plot) a "true" element $T(\hat{K})$ for the choices below. In particular, discuss the difference between (A) and (C). Discuss T^{-1}. (**Hint:** actually finding T^{-1} might be really messy. Describing what kind of function it is might be easier).

(A) Use some $T \in \mathbf{Q}_1 \setminus \mathbf{P}_1$ of your choice.
(B) Use some $T \in \mathbf{P}_2 \setminus \mathbf{P}_1$ of your choice.
(C) Now consider $\hat{K}$ to be the reference unit square, and use T as in (A).

---

2. Eigenvalues

Frequently, we need our FE code to calculate some quantity of interest other than just the solution. For example, we need boundary fluxes, or eigenvalues, or the average of the solution over some subdomain. These numerically obtained quantities have errors associated with them, and it is important to understand these errors.

Modify the code in `fem1d_2017.m` to obtain approximation to the eigenvalues of the problem

$$-u'' = \lambda u, \quad x \in (0, 1) \tag{1a}$$

$$u(0) = 0, \quad u(1) = 0 \tag{1b}$$

Here we use piecewise linears only.

The code similar to that below (added to `fem1d_2017.m`) can be used to find the eigenvalues $\lambda_{h,n}$ of the stiffness matrix K_h, i.e., solving

$$K_h V = \lambda_h V. \tag{2}$$

for the eigenvalues λ_h and eigenvectors V.

```
snod = length(freenodes);
matfull=zeros(snod,snod);
smat=mat(freenodes,freenodes);
matfull(:,:)=xnel*smat;
massmatfull(:,:)=xnel*smassmat;
[eigvec,eigvals]=eig(matfull);
```

```
deigvals=diag(eigvals);
deigvals(1),
```

Note the scaling in the code which is necessary to make this calculation in 1d produce the "proper" approximation to the "discrete Laplacian". Also note that I am suggesting using `eig` *rather than* `eigs` *for eigenvalue computations. (Check MATLAB documentation to understand the difference).*

It is known that this (continuous) problem has eigenvalues $\lambda_n = (n\pi)^2, n = 1, 2, \ldots$. Your code will produce approximations $\lambda_{h,n}$ to λ_n. **Determine and discuss** the (order) of accuracy of the first eigenvalue $n = 1$, and of eigenvalue $n = 21$ by using a well chosen set of the values of h.

---

3. Eigenvalues, more precisely

Write the variational and FE formulation of (1) to show that the discrete eigenvalue problem is really not (2) but rather the one involving the *generalized eigenvalue problem*

$$K_h V = \lambda_h M_h V. \tag{3}$$

where M_h is the mass matrix you used in HW 2 to solve Problem 4.

Extra: Redo Problem 2 and compare the eigenvalues you find from (3) to those in (2).

---

4. Superconvergence

In class we discussed the curious phenomenon that occurs when you calculate

$$\| u - u_h \|_{l_{\inf}} = \max_{j=1,\texttt{nnodes}} |u(x_j) - u_h(x_j)| \tag{4}$$

for the solution to $-u'' = f,\ x \in (0, 1), u(0) = u(1) = 0$.

In the code `fem1d_2017.m` this norm would be implemented as

```
norm(sol-exfun(xnod),inf))
```

As we said, you should **never** judge the accuracy of FE solution based on this discrete norm. The theoretical explanation for this phenomenon (see [CJ, Exercise]) involves the calculation

$$u(x_j) - u_h(x_j) = \delta_{x_j}(u - u_h) = \ldots, \tag{5}$$

and the Green's function G_{x_j} for the problem, defined by

$$a(G_{x_j}, v) = \delta_{x_j}(v), \quad \forall v \in V \tag{6}$$

A few further steps involve the FE solution to (6), and the use of G.O.

Use your work for HW 2 (with $\mathbf{P}_1$ elements) to calculate (4) for the case (a) $f(x) = 1$ and (b) $f(x) = \sin(\pi x)$. (Study the errors and the order of convergence). **Report and**

explain the difference between (a) and (b). **Suggest** how you would fix this so that there is no difference between these two cases. Is it worth it?

Extra: Is there any difference theoretically when you use piecewise quadratics? Confirm your answer using the code. What about when your problem is $-u'' + u = f,\ x \in (0,1), u(0) = u(1) = 0$?