Your Homework Assignment
Finite Elements and Elliptic PDEs
Notes on Solution
CMSC/AMSC 661 Spring 2005

The posted program `hmwk2.m` solves the 6 problems and produces the output table. The graph can be produced by `pdemesh(p,e,t,'zdata',abs(ucomp-utrue'))`, where `p, e, t,` and `ucomp` are returned by `getpetuc`.

The relevant convergence result is given in equation (5.53) on p. 66 of the text. (That is the only result in the book that concerns the sup-norm.) The hypotheses are:

- The triangulation is quasi-uniform. (Satisfied.)
- The boundary is smooth. (Satisfied only for Ω_1.)
- $\mathcal{A}$ coercive. (Satisfied.)
- The coefficients are smooth. (The coefficient a_2 has a discontinuity in its first derivative.)
- $u = 0$ on the boundary. (Violated.)

The conclusion is

$$\|u - u_h\|_{\mathcal{C}} \leq Ch^2 \log(1/h)\|u\|_{\mathcal{C}^2}.$$

Since u is smooth, the norm on the right is well behaved.

As usual, the book has not stated the result in full generality; more general boundary conditions are fine, and convex polygons should also be ok. The pacman domain works in our test problem because the true solution is so smooth. Since we compute the maximum only at the vertices of triangles rather than everywhere in the domain, the book's result should be an upper bound on our error.

I was surprised by the results, and a finite element theory expert that I consulted was also surprised by these two features:

- Intuition told us that the log term would not be seen, but that is wrong – until h is quite small, the log changes the convergence rate significantly.
- Sam Lamphier noticed that the error takes a big jump at vertices in which the number of intersecting triangles is not equal to 6. As far as we can tell, this effect is not discussed in the literature, and the cause is an interesting open question.

These features are best illustrated on a simpler problem, with zero boundary conditions, given in `pdedem`, derived from Matlab's `pdedemo1.m`. In this case, the error spikes dramatically at the center of the circle, where 8 triangles meet.

If we divide the computed error for mesh h by that for mesh $h/2$, we should expect to see

$$\frac{h^2 \log(1/h)}{(h^2 \log(1/(h/2))/4)} = \frac{4 \log h}{\log h - \log 2} \equiv \alpha.$$

Let $h = 10^{-p}$, and let's see how the log factor changes α from 4:

p	α
1	3.07
2	3.48
3	3.64
4	3.72
5	3.77
6	3.81
7	3.84
8	3.85
9	3.87
10	3.88

Some notes on your submissions:

- Yes, I really want to see the raw output from your Matlab programs. (See the solution to Homework 1.)
- Using `pdecirc` instead of `pdeellip` created a significantly different grid, one of Matlab's oddities.
- In your documentation, you should credit Mathworks (Matlab) for the code you borrowed.
- You can plot the mesh outside of the GUI using `pdemesh(p,e,t)`.
- To estimate h:
  - If we assume that the triangles have uniform area, then the longest side is proportional to `sqrt(area / nt)` where `nt =max(size(t))`. The proportionality constant is, for example, $\sqrt{2}$ if we assume equilateral right triangles.
  - (Courtesy of Patrick Rabenold) The actual maximum side of the ith triangle can be computed by

    ```
    x = p(1,t([1 2 3 1],i)); % x coordinates of vertices
    y = p(2,t([1 2 3 1],i)); % y coordinates of vertices
    len = sqrt((x(1:3)-x(2:4)).^2 + (y(1:3)-y(2:4)).^2);
    maxlen = max(len);
    ```
  - From the `x,y` data, it would also be possible to compute the diameter of the circumscribing circle.