Assignment 2

Finite Element Methods

R.G.A. Deckers

1.

Start with

$$-\int_\Omega \left[\nabla\cdot(\kappa\nabla u)\right] v \;\mathrm{d}\vec{x} = \int_\Omega f v \;\mathrm{d}\vec{x} \tag{1.1}$$

And then use Green's Theorem (specifically, equation 4.3 from the book) and simplify to find:

$$\begin{aligned}\int_\Omega \kappa\nabla u\cdot\nabla v \;\mathrm{d}\vec{x} - \int_{\partial\Omega} \vec{n}\cdot(\kappa\nabla u)\, v \;\mathrm{d}s &= \int_\Omega f v \;\mathrm{d}\vec{x}\\ \int_\Omega \kappa\nabla u\cdot\nabla v \;\mathrm{d}\vec{x} - \int_{\partial\Omega} \gamma\,(g-u)\, v \;\mathrm{d}s &= \int_\Omega f v \;\mathrm{d}\vec{x}\\ \int_\Omega \kappa\nabla u\cdot\nabla v \;\mathrm{d}\vec{x} + \int_{\partial\Omega} \gamma u v \;\mathrm{d}s &= \int_\Omega f v \;\mathrm{d}\vec{x} + \int_{\partial\Omega} \gamma g v \;\mathrm{d}s\end{aligned} \tag{1.2}$$

2.

We compute the AK (and bK) elements using the following code

```
function [ AK, bK ] = create_AK_bK( x, y, f, kappa)
  %x, y are triplets of vertices, f and kappa are funciton handles.
  area_K = polyarea(x,y);
  %abc matrix
  Z = [ones(1,3); x; y].';
  %solve for the three abc vectors
  abc = [Z\[1;0;0] Z\[0;1;0] Z\[0;0;1]];
  b = abc(2,:);
  c = abc(3,:);
  %take the centroid coordinates
  x_c = mean(x);
  y_c = mean(y);
  %evaluate the given expression for AK and bK
  %and compute kappa and f at the centroid
  AK = (b.'*b+c.'*c)*kappa(x_c, y_c)*area_K;
  bK = f(x_c, y_c)*area_K/3;
end
```

We then compute the eigenvalues of the reference triangle by explicily passing its vertices, using the following function

```
function [ eigen ] = problem_02( )
  AK = create_AK_bK([0 0 1], [0 1 0], @f_const, @kappa); %create AK over the reference triangle
  eigen = eig(AK); %and return its eigenvalues
end
```

We find that one of the eigenvalues is indeed 0.

3.

We have solved problem 3 using the following code, it automatically generates all the necessary plots (formatting was done afterwards).

```
function [order] = problem_03()
  X =[];
  Norm = [];
  %loop over different grid sizes
  %in logspace
  for j = 0.5:0.1:6.5
    dx = 2^-j;
    [p,e,t] = create_mesh(dx);
    [A,R, b, r] = assemble(p, e, t, 1e6, @f_sin, @g_const, @kappa);
    z = solve(A, R, b, r);
    for i = 1:size(p,2)
      x = p(1,i);
      y = p(2,i);
      z(i) = abs(z(i)-  exact(x,y));
    end
    norm = log(z.'*A*z);
    Norm = [Norm norm];
    X = [X log(dx)];
  end
  figure(1)
  plot(X, Norm)
  order = polyfit(X, Norm,1);
  hold on;
  fh = @(x) order(1)*x+order(2);
  ezplot(fh, [-7, 0]);

  dx = 0.05;
  [p,e,t] = create_mesh(dx);
  [A,R, b, r] = assemble(p, e, t, 1e6, @f_sin, @g_const, @kappa);
  z = solve(A, R, b, r);
  figure(2)
  make_plot(p,e,t, z);

end


function z = exact(x, y)
  z = sin(pi*x)*sin(pi*y);
end
```

3a)

We have solved the system with the known solution, the result is presented in figure 1.

3b)

In figure 2 we have plotted the convergence of the energy-norm in log-log space. The best line fit has a slope of 2.3 showing the expected 2nd order convergence of our method.

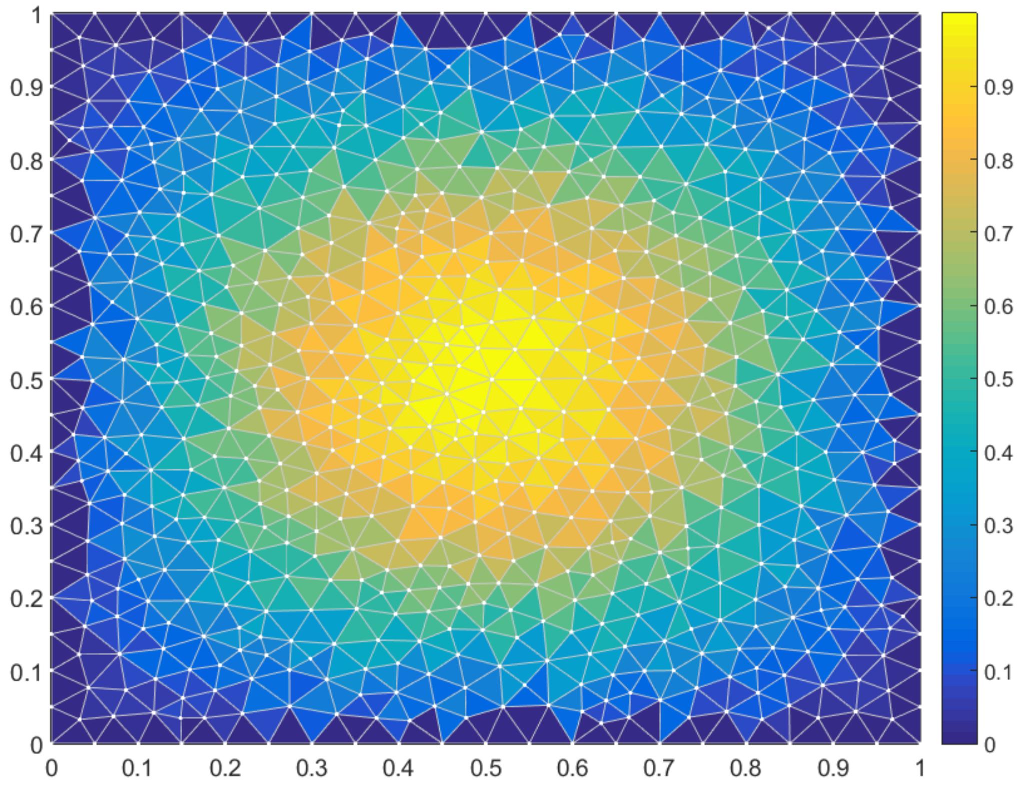

Figure 1: The solution to $-\Delta u = 2\pi^2 \sin(\pi x)\sin(\pi y)$ with a zero-boundary.

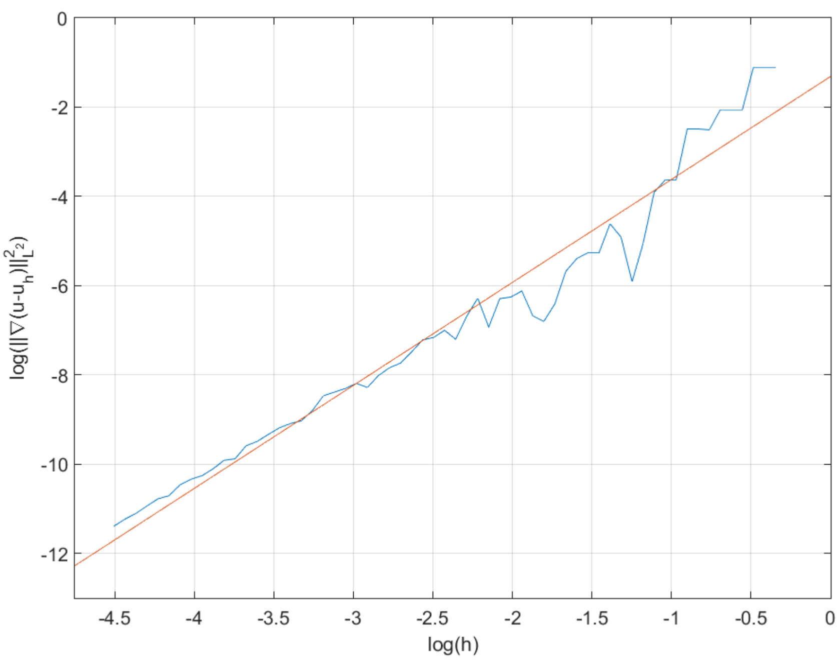

Figure 2: Convergence of the energy norm in log-log space. The slope of the best fit is ≈ 2 showing 2nd-order convergence.

4.

```
function [  ] = problem_04(  )
  dx = 0.05;
  [p,e,t] = create_mesh(dx);
  [A,R, b, r] = assemble(p, e, t, 1e6, @f_const, @g_cos, @kappa);
  z = solve(A, R, b, r);
  figure(3)
  make_plot(p,e,t, z);
end
```

We have solved the system with $-\Delta u = 1$ and boundary condition $u = cos(2\pi y)$ along the y-axis and 0 elsewhere, the results are shown in figure 3.

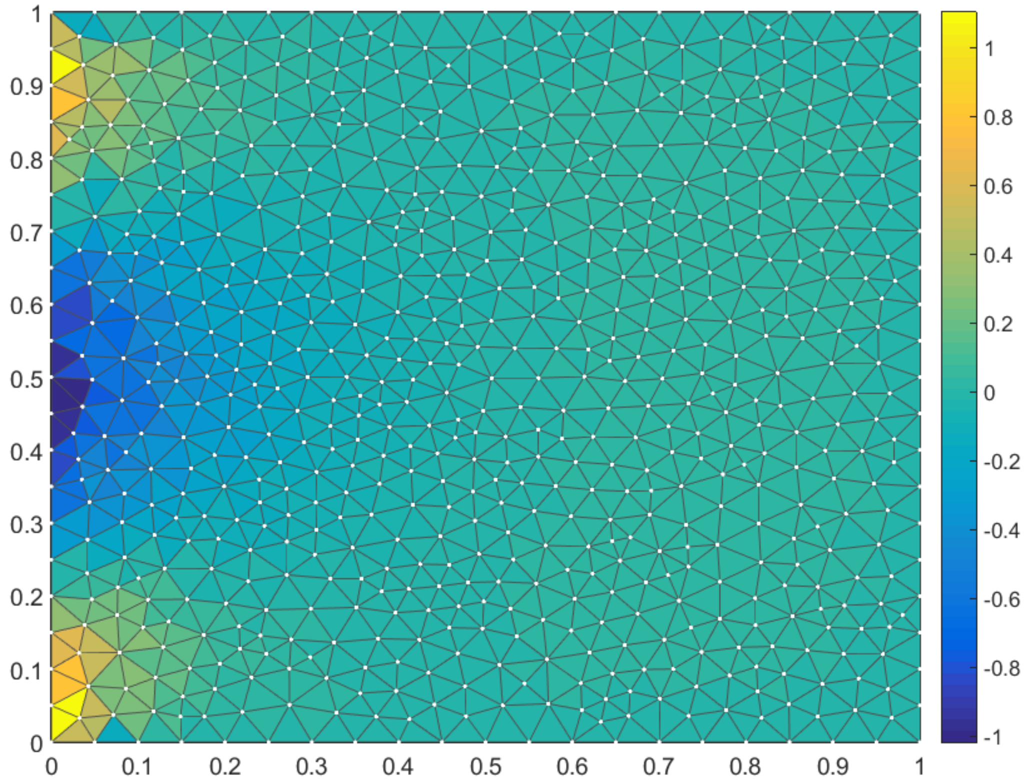

Figure 3: The solution to $-\Delta u = 1$. with boundary condition $u = cos(2\pi y)$ along the y-axis and 0 elsewhere.