Assignment 2

Finite Element Methods

R.G.A. Deckers

1.

1a) *Derivation*

Start with variational form of Poisson's Equation by Multiplying $f = -\Delta u$ by v and integrating over Ω:

$$-\int_\Omega -\Delta uv \,\mathrm{d}x = \int_\Omega fv \,\mathrm{d}x \tag{1.1}$$

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

$$\begin{aligned} -\int_\Omega -\Delta uv \,\mathrm{d}x &= \int_\Omega fv \,\mathrm{d}x \\ \int_\Omega \nabla u \cdot \nabla v \,\mathrm{d}x - \int_{\partial\Omega} n \cdot \nabla uv \,\mathrm{d}s &= \int_\Omega fv \,\mathrm{d}x \\ \int_\Omega \nabla u \cdot \nabla v \,\mathrm{d}x &= \int_\Omega fv \,\mathrm{d}x. \end{aligned} \tag{1.2}$$

Here v are members of the standard space V_0, $V = \left\{ v : \|v\|_{L^2(\Omega)} + \|\nabla v\|_{L^2(\Omega)} < \infty \right\}, V_0 = \{v \in V : v|_{\partial\Omega} = 0$. We now further restric us to the space V_h of continous piecewise linears on triangulation K (with $V_{h,0}$ defined similarly to V_0) then the FEM formulation is:

$$\int_\Omega \nabla u_h \cdot \nabla v \,\mathrm{d}x = \int_\Omega fv \,\mathrm{d}x, \; \forall v \in V_{h,0}. \tag{1.3}$$

now let the set of ϕ_i, $i = 1, \ldots, n$ (where n is the number of interior nodes) define a basis of $V_{h,0}$, then the above equation has to hold for all ϕ_i, that is

$$\int_\Omega \nabla u_h \cdot \nabla \phi_i \,\mathrm{d}x = \int_\Omega f\phi_i \,\mathrm{d}x, \; i = 1, \ldots, n. \tag{1.4}$$

Since $u_h \in V_{h,0}$ we can write

$$u_h = \sum_{j=1}^{n} \alpha_j \phi_j, \tag{1.5}$$

inserting this into the previous equation we find

$$\sum_{j=1}^{n} \alpha_j \int_\Omega \nabla \phi_j \cdot \nabla \phi_i \,\mathrm{d}x = \int_\Omega f\phi_i \,\mathrm{d}x, \; i = 1, \ldots, n. \tag{1.6}$$

Which can be written as $A\alpha = b$.

We have implemented this equation in Matlab (using the implementation details discussed in lab 2). We construct our A matrix and b vector as:

```
function [A,b] = assemble( p,e,t, f)

I = eye(length(p));
N = size(p,2);
A = sparse(N,N);
b = zeros(N,1);

```

```
for K = 1:size(t,2); % loop over the triangles
  nodes = t(1:3,K); % find triangle K's nodes

  x = p(1,nodes);
  y = p(2,nodes);

  [AK, bK] = create_AK_bK(x,y, f);

  % add AK(i,j), i,j=1,2,3, to A(nodes(i),nodes(j))
  A(nodes,nodes) = A(nodes,nodes)+AK;
  b(nodes) = b(nodes) + bK;
end
A(e(1,:),:) = I(e(1,:),:); % replace the rows corresponding
% to the boundary nodes by corresponding
% rows of I
b(e(1,:)) = 0; % put the boundary value into the RHS
end
```

with the helper function

```
function [ AK, BK, bK ] = create_AK_bK( x, y, f, epsilon, beta)
  %x, y are triplets of vertices, f is a function handle.
  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
  %compute f at the centroid
  BK = 1/12*area_K*[2 1 1; 1 2 1; 1 1 2];
  %AK = epsilon*(b.'*b+c.'*c)*area_K;
  bK = f(x_c, y_c)*area_K/3;
  AK =
end
```

1b) *Results*

We have computed the Galerkin results, the exact solution, and the error with $f(\vec{x}) = 8\pi^2 \sin(2\pi x_1)\sin(2\pi x_2)$ and $h = 1/8$ and $h = 1/16$ in figure 1, 2, and 3

1c) *convergence*

We have determined the convergence rate in the energy norm of the error $\|u - U\|_E^2$ for various values of $h_{\max}$ between $1/2$ and $1/32$, and the fitting a linear function to $\log \|u - U\|_E^2$ over $\log_2 h_{\max}$. The order of convergence is then given by the slope of the fit (as $\log x^a = a \log x$). We find an order of ≈ 0.84. The results are plotted in 4.

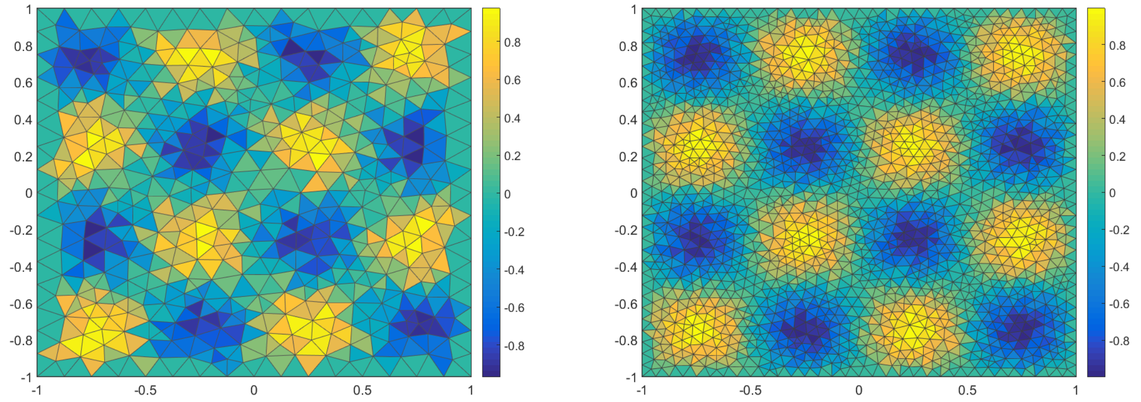

Figure 1: The Galerkin solution

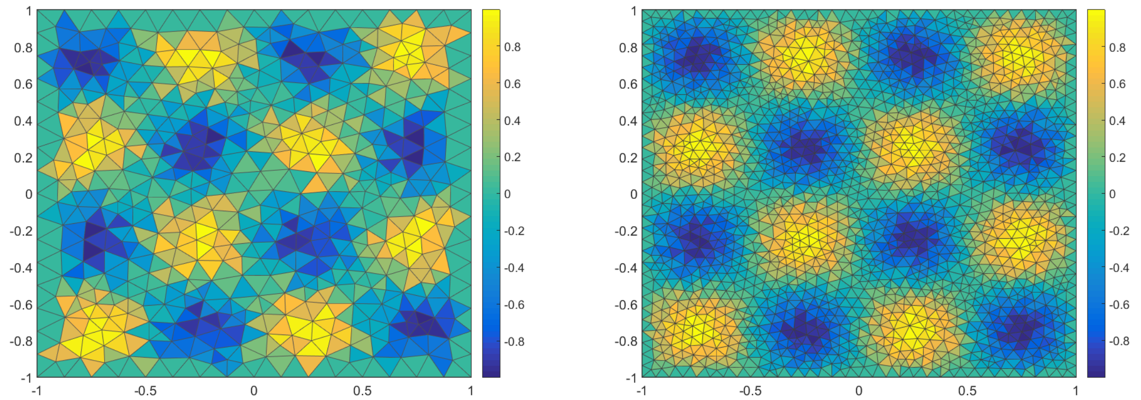

Figure 2: The exact solution

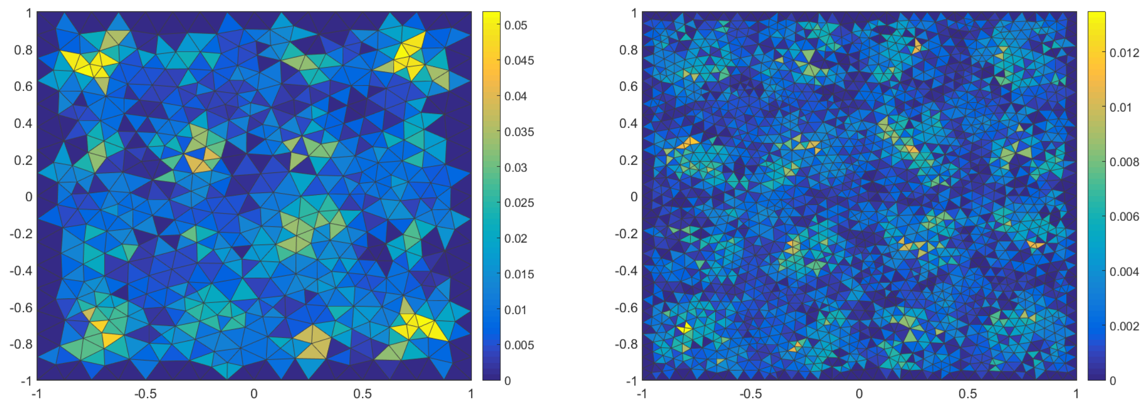

Figure 3: The absolute difference between the Galerkin and exact solution

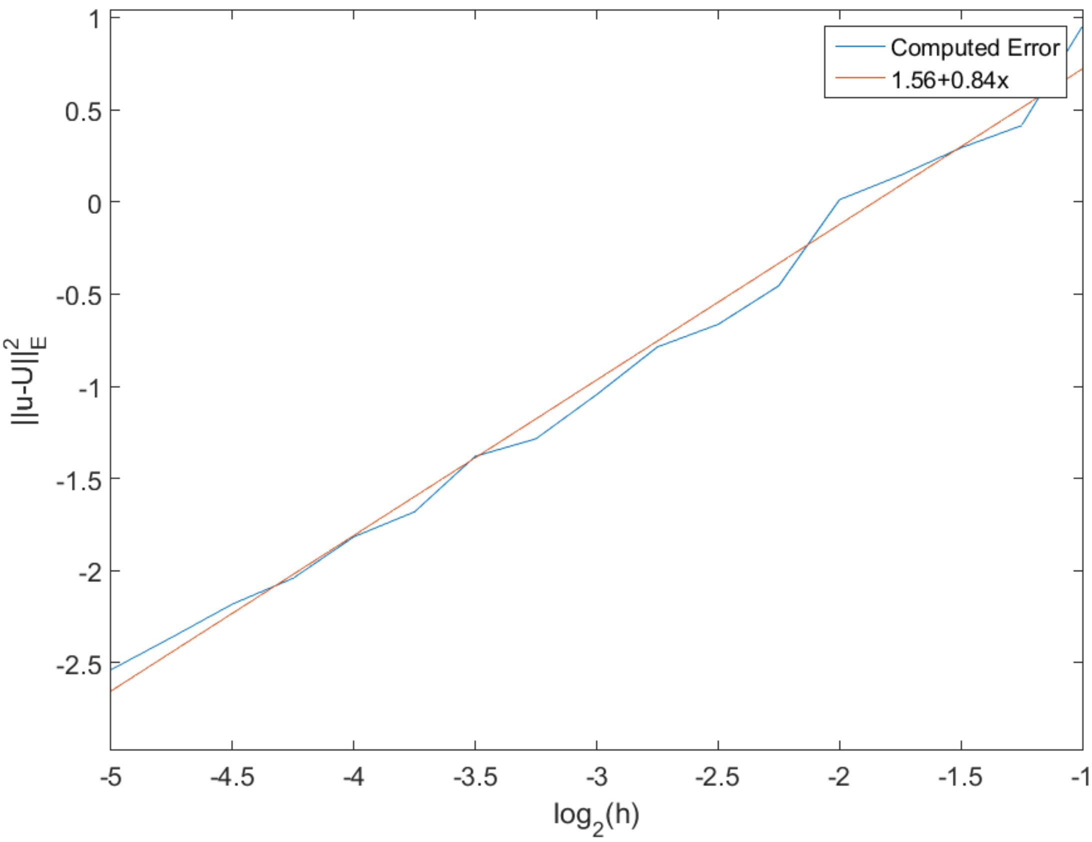

Figure 4: Convergence plot of our Galerkin plot.

2.

2a) *Galerkin Formulation*

The variational formulation for our problem is

$$
\begin{aligned}
\int_\Omega fv &= \int_\Omega \left(\partial_t u + \beta \cdot \nabla u - \nabla \cdot (\epsilon \nabla u)\right) v \, \mathrm{d}x \\
&= \int_\Omega \left(\partial_t u + \beta \cdot \nabla u + \epsilon \nabla u \cdot \nabla\right) v \, \mathrm{d}x - \int_{\partial\Omega} (\epsilon \nabla u \cdot \vec{n})\, v \, \mathrm{d}s \\
&= \int_\Omega \left(\partial_t u + \beta \cdot \nabla u + \epsilon \nabla u \cdot \nabla\right) v \, \mathrm{d}x
\end{aligned} \tag{2.1}
$$

here the last step follows from the fact the $v \in V_{h,0}$ and is therefore 0 along the boundary (which we have chosen as a boundary condition). Now we replace u with u_h and write both u_h and v in terms of the basis functions ϕ again, giving:

$$
\begin{aligned}
\int_\Omega f\phi_i \, \mathrm{d}x &= \alpha_j \int_\Omega \left(\epsilon \nabla \phi_j \cdot \nabla \phi_i + \beta \cdot \nabla \phi_j \phi_i\right) \, \mathrm{d}x + \dot{\alpha}_j \int_\Omega \phi_i \phi_j \, \mathrm{d}x \\
\vec{b} &= A\vec{\alpha} + B\dot{\vec{\alpha}}
\end{aligned} \tag{2.2}
$$

Now, noting that $\dot{\vec{\alpha}} = B^{-1}\left(\vec{b} - \epsilon A \vec{\alpha}\right)$, we apply time discretization using the Crank-Nicolson scheme:

$$\begin{aligned} \frac{\alpha^+ - \alpha}{\Delta t} &= \frac{1}{2}\left(\dot{\alpha}^+ + \dot{\alpha}\right) \\ \implies \left(B + \frac{\Delta t}{2} A\right)\alpha^+ &= \left(B - \frac{\Delta t}{2} A\right)\alpha + \frac{\Delta t}{2} b \end{aligned} \tag{2.3}$$