Contents

- plot mesh
- FEM
- Boundary conditions - Dirichlet
- Sink
- Source
- Current
- LU
- Covariance


```
% HW 6
```

plot mesh

```
clear;close all

% File names
nodf='hw44.nod';
belf='hw44.bel';
dndf='hw44.dnd';

node=load(nodf);
node=node(:,2:3); % All nodes
bel=load(belf);
bel=bel(:,2:3); % Boundary element incidence list
bnod=unique(bel(:)); % Boundary nodes
dnd=load(dndf);
dnd=dnd(:,1); % Type 1 BC nodes (ground)

% Plot nodes
plot(node(:,1),node(:,2),'b.')
title(nodf)
ylim([-.2 1.5])
xlim([-1 1.5])


hold on
% Plot Boundary nodes
plot(node(bnod,1),node(bnod,2),'g.')
% Plot type 1 BC nodes
plot(node(dnd,1),node(dnd,2),'bo')
% Plot Current sink
plot(node(492:493,1),node(492:493,2),'ro')
% Plot Current source
plot(node(503,1),node(503,2),'gx','markersize',18,'linewidth',2)
legend('Nodes','Boundary Nodes','Dirichlet nodes','Boundary current sink','current source')
```

FEM

```
elem = load( 'hw44.ele' );
bcs = load( 'hw44.dnd' );
A_e = zeros( 3 );
A = zeros( size(node,1)-3 );
B = zeros( size(node,1)-3, 1 );

for i=1:size(elem,1)
    n1 = elem(i,2);
    n2 = elem(i,3);
    n3 = elem(i,4);

    x1 = node(n1,1);
    y1 = node(n1,2);
    x2 = node(n2,1);
    y2 = node(n2,2);
    x3 = node(n3,1);
    y3 = node(n3,2);

    area = (x1*(y2-y3) + x2*(y3-y1) + x3*(y1-y2)) / 2;

    t1 = 1 + x1/10 + y1/5;
    t2 = 1 + x2/10 + y2/5;
    t3 = 1 + x3/10 + y3/5;


    A_e(1,1) = -t1 * (((y2-y3)*(y2-y3))+((x2-x3)*(x2-x3))) / (4*area);
    A_e(1,2) = -t1 * (((y2-y3)*(y3-y1))+((x2-x3)*(x3-x1))) / (4*area);
    A_e(1,3) = -t1 * (((y2-y3)*(y1-y2))+((x2-x3)*(x1-x2))) / (4*area);
    A_e(2,1) = -t2 * (((y3-y1)*(y2-y3))+((x3-x1)*(x2-x3))) / (4*area);
    A_e(2,2) = -t2 * (((y3-y1)*(y3-y1))+((x3-x1)*(x3-x1))) / (4*area);
    A_e(2,3) = -t2 * (((y3-y1)*(y1-y2))+((x3-x1)*(x1-x2))) / (4*area);
    A_e(3,1) = -t3 * (((y1-y2)*(y2-y3))+((x1-x2)*(x2-x3))) / (4*area);
    A_e(3,2) = -t3 * (((y1-y2)*(y3-y1))+((x1-x2)*(x3-x1))) / (4*area);
    A_e(3,3) = -t3 * (((y1-y2)*(y1-y2))+((x1-x2)*(x1-x2))) / (4*area);

    A(n1,n1) = A(n1,n1) + A_e(1,1);
    A(n1,n2) = A(n1,n2) + A_e(1,2);
    A(n1,n3) = A(n1,n3) + A_e(1,3);
    A(n2,n1) = A(n2,n1) + A_e(2,1);
    A(n2,n2) = A(n2,n2) + A_e(2,2);
    A(n2,n3) = A(n2,n3) + A_e(2,3);
    A(n3,n1) = A(n3,n1) + A_e(3,1);
    A(n3,n2) = A(n3,n2) + A_e(3,2);
    A(n3,n3) = A(n3,n3) + A_e(3,3);
end
```

Boundary conditions - Dirichlet

```
for i = 1:size(bcs,1)
   B(i) = bcs(i,2);
   A(i,:) = 0;
   A(i,i) = 1;
end
```

Sink

```
B(492) = .25;
B(493) = .25;
```

Source

```
n1 = elem(288,2);
n2 = elem(288,3);
n3 = elem(288,4);

x1 = node(n1,1);
y1 = node(n1,2);
x2 = node(n2,1);
y2 = node(n2,2);
x3 = node(n3,1);
y3 = node(n3,2);

area = (x1*(y2-y3) + x2*(y3-y1) + x3*(y1-y2)) / 2;

% source at node 503, goes through 504,505?
B( elem(288,2) ) = -( (x2*y3-x3*y2) + node(503,1)*(y2-y3) - node(503,2)*(x2-x3) ) / (2*area);
B( elem(288,3) ) = -( (x3*y1-x1*y3) + node(503,1)*(y3-y1) - node(503,2)*(x3-x1) ) / (2*area);
B( elem(288,4) ) = -( (x1*y2-x2*y1) + node(503,1)*(y1-y2) - node(503,2)*(x1-x2) ) / (2*area);


Sol = linsolve(A,B);

% for i = 1:size(elem,1)
%     patch( node(elem(i,2:4),1), node( elem(i,2:4),2), Sol(elem(i,2:4)), 'FaceColor', 'interp
' );
%     hold on;
% end
% colorbar;
% title( 'Potential');
```

Current

```
for i = 1:size(elem,1)
    n1 = elem(i,2);
    n2 = elem(i,3);
    n3 = elem(i,4);

    x1 = node(n1,1);
    y1 = node(n1,2);
    x2 = node(n2,1);
    y2 = node(n2,2);
    x3 = node(n3,1);
    y3 = node(n3,2);

    area = (x1*(y2-y3) + x2*(y3-y1) + x3*(y1-y2)) / 2;
```

```
    curr_x(i) = -Sol(n1)*(y2-y3)/(2*area) - Sol(n2)*(y3-y1)/(2*area) - Sol(n3)*(y1-y2)/(2*area
);
    curr_y(i) = ( Sol(n1)*(x2-x3)/(2*area) + Sol(n2)*(x3-x1)/(2*area) + Sol(n3)*(x1-x2)/(2*are
a));

    x_centers(i) = (x1+x2+x3)/3;
    y_centers(i) = (y1+y2+y3)/3;
end
%
% figure(2);
% for i = 1:size(elem,1)
%     patch( node(elem(i,2:4),1), node( elem(i,2:4),2), 'w' );
%     hold on;
% end
% quiver( x_centers, y_centers, curr_x, curr_y, 'b', 'LineWidth', 1.5);
% title('Current');
```

LU

```
[L,U] = lu(A);

y = linsolve(L,B);
sol_lu = linsolve(U,y);

% figure(3);
% for i = 1:size(elem,1)
%     patch( node(elem(i,2:4),1), node( elem(i,2:4),2), sol_lu(elem(i,2:4)), 'FaceColor', 'int
erp' );
%     hold on;
% end
% colorbar;
% title('Potential with LU Decompositon');
```

Covariance

```
Ba = zeros(size(node,1)-3,1);
Ba( elem(288,2) ) = .5 * B( elem(288,2) );
Ba( elem(288,3) ) = .5 * B( elem(288,3) );
Ba( elem(288,4) ) = .5 * B( elem(288,4) );

Cov_Ua = inv(A)*Ba*transpose(Ba)*transpose(inv(A));
Sol_a = sqrt( diag(Cov_Ua) );
% figure(4);
% for i = 1:size(elem,1)
%     patch( node(elem(i,2:4),1), node( elem(i,2:4),2), Sol_a(elem(i,2:4)), 'FaceColor', 'inte
rp' );
%     hold on;
% end
% colorbar;
% title('Potential with 50% Input Variation');


Bb = zeros(size(node,1)-3,1);
```

```
Bb( 492 ) = .6 * B( 492 );
Bb( 493 ) = .6 * B( 493 );

Cov_Ub = inv(A)*Bb*transpose(Bb)*transpose(inv(A));
Sol_b = sqrt( diag(Cov_Ub) );
% figure(5);
% for i = 1:size(elem,1)
%     patch( node(elem(i,2:4),1), node( elem(i,2:4),2), Sol_b(elem(i,2:4)), 'FaceColor', 'inte
rp' );
%     hold on;
% end
% colorbar;
% title('Potential with 40% Output Variation');


Bc = zeros(size(node,1)-3,1);
Bc( dnd ) = .5^2;

Cov_Uc = inv(A)*Bc*transpose(Bc)*transpose(inv(A));
Sol_c = sqrt( diag(Cov_Uc) );
% figure(6);
% for i = 1:size(elem,1)
%     patch( node(elem(i,2:4),1), node( elem(i,2:4),2), Sol_c(elem(i,2:4)), 'FaceColor', 'inte
rp' );
%     hold on;
% end
% colorbar;
% title('Ground potential +/- .5 V');


Bd = zeros(size(node,1)-3,1);
Bd( elem(288,2) ) = .5 * B( elem(288,2) );
Bd( elem(288,3) ) = .5 * B( elem(288,3) );
Bd( elem(288,4) ) = .5 * B( elem(288,4) );
Bd( 492 ) = .6 * B( 492 );
Bd( 493 ) = .6 * B( 493 );
Bd( dnd ) = .5;

Cov_Ud = inv(A)*Bd*transpose(Bd)*transpose(inv(A));
Sol_d = sqrt( diag(Cov_Ud) );
% figure(7);
% for i = 1:size(elem,1)
%     patch( node(elem(i,2:4),1), node( elem(i,2:4),2), Sol_d(elem(i,2:4)), 'FaceColor', 'inte
rp' );
%     hold on;
% end
% colorbar;
% title('Combined Parameters');
```

Published with MATLAB® R2017b