Homework 4

Table of Contents

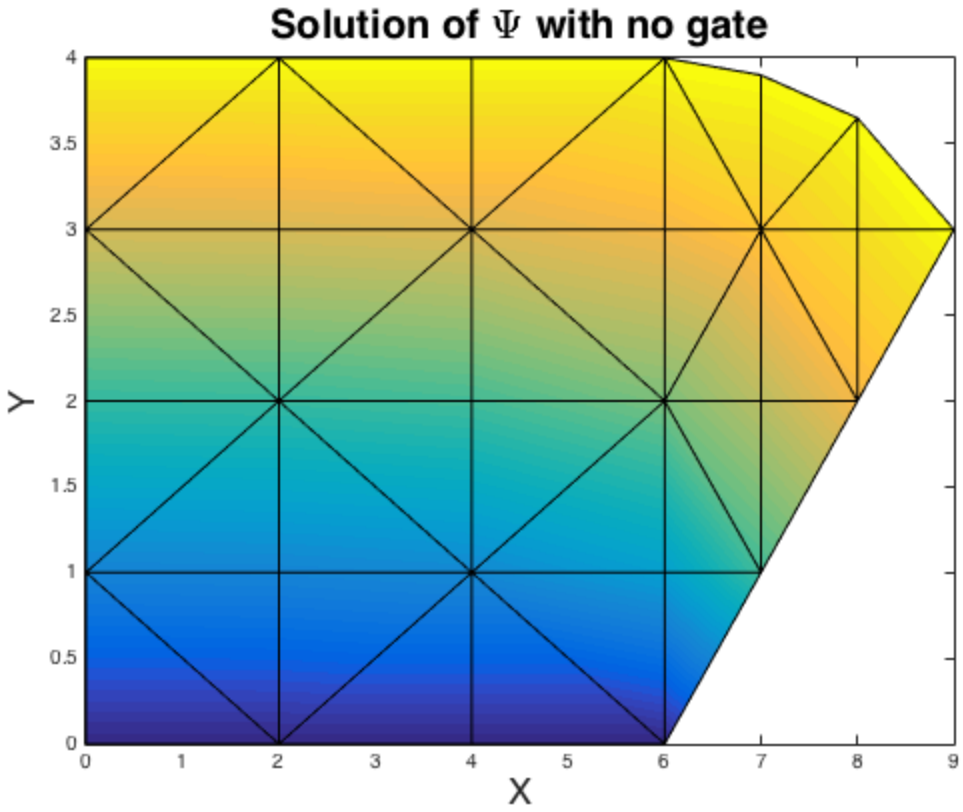
Boyu Meng 02/11/2018

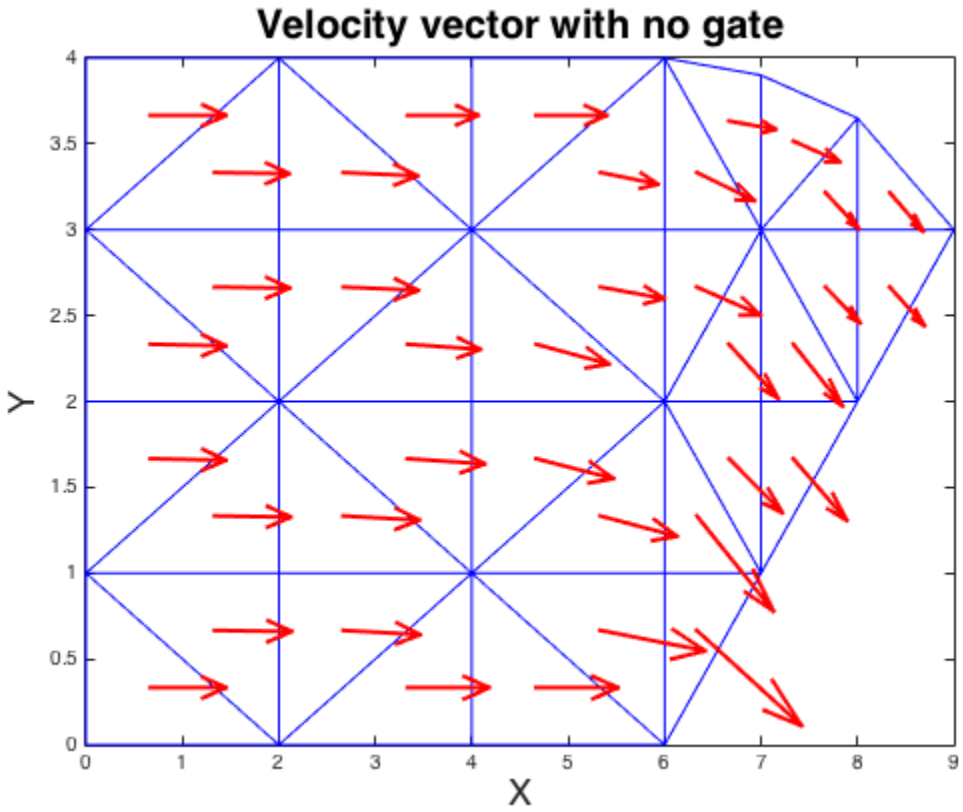
with no gate

make a matrix

```
clearvars
ele=dlmread('hw5.ele1.txt');
nod=dlmread('hw5.nod.txt');
a_mat=zeros(length(nod));
c_mat=a_mat;
mark=ones(3);
for k=1:length(ele)
    e_nod=ele(k,2:4);
    x_e=nod(e_nod,2);
    y_e=nod(e_nod,3);
    delt_x(1)=x_e(2)-x_e(3);
    delt_x(2)=x_e(3)-x_e(1);
    delt_x(3)=x_e(1)-x_e(2);
    delt_y(1)=y_e(2)-y_e(3);
    delt_y(2)=y_e(3)-y_e(1);
    delt_y(3)=y_e(1)-y_e(2);
    a_area=0.5*(x_e(1)*delt_y(1)+x_e(2)*delt_y(2)+x_e(3)*delt_y(3));
    for i=1:3
        for j=1:3
            amat_e(i,j)=-(delt_x(i)*delt_x(j)+delt_y(i)*delt_y(j))/
(4*a_area);

 a_mat(e_nod(i),e_nod(j))=a_mat(e_nod(i),e_nod(j))+amat_e(i,j);

 c_mat(e_nod(i),e_nod(j))=c_mat(e_nod(i),e_nod(j))+mark(i,j);
        end
    end
end
% modify A matrix base on the boundary condition
index_1=nod(nod(:,3)==0,1);
index_2=nod(nod(:,2)==0,1);
index_3=nod(nod(:,3)==4,1);
index_3=[index_3;24;27;28];
index_master=[index_1;index_2;index_3];
a_mat(index_master,:)=0;
%a_mat(:,index_master)=0;
for i=1:length(index_master)
```

```
    a_mat(index_master(i),index_master(i))=1;
end
% build B array
b_ary=zeros(length(nod),1);
b_ary(index_1)=0;
b_ary(index_2)=nod(nod(:,2)==0,3)/4;
b_ary(index_3)=1;
% solve for solution
u=a_mat\b_ary;
% plot
figure()
triplot([ele(:,2),ele(:,3),ele(:,4)],nod(:,2),nod(:,3));
hold on
trisurf([ele(:,2),ele(:,3),ele(:,4)],nod(:,2),nod(:,3),u,'FaceColor','interp');
title('Solution of \Psi with no gate','fontsize',20)
xlabel('X','fontsize',18)
ylabel('Y','fontsize',18)
% velocity
for k=1:length(ele)
    e_nod=ele(k,2:4);
    x_e=nod(e_nod,2);
    y_e=nod(e_nod,3);
    x_lab(k)=mean(x_e);
    y_lab(k)=mean(y_e);
    delt_x(1)=x_e(2)-x_e(3);
    delt_x(2)=x_e(3)-x_e(1);
    delt_x(3)=x_e(1)-x_e(2);
    delt_y(1)=y_e(2)-y_e(3);
    delt_y(2)=y_e(3)-y_e(1);
    delt_y(3)=y_e(1)-y_e(2);
    a_area=0.5*(x_e(1)*delt_y(1)+x_e(2)*delt_y(2)+x_e(3)*delt_y(3));
    fl=u(e_nod);
    v_x(k)=fl(1)*delt_y(1)/(2*a_area)+fl(2)*delt_y(2)/
(2*a_area)+fl(3)*delt_y(3)/(2*a_area);
    v_y(k)=-(fl(1)*delt_x(1)/(2*a_area)+fl(2)*delt_x(2)/
(2*a_area)+fl(3)*delt_x(3)/(2*a_area));
end
figure()
triplot([ele(:,2),ele(:,3),ele(:,4)],nod(:,2),nod(:,3));
hold on
%trisurf([ele(:,2),ele(:,3),ele(:,4)],nod(:,2),nod(:,3),u,'FaceColor','interp');
quiver(x_lab,y_lab,v_y,-v_x,'linewidth',2,'color','red')
title('Velocity vector with no gate','fontsize',20)
xlabel('X','fontsize',18)
ylabel('Y','fontsize',18)
```


Solution of Ψ with no gate
Y
X
0
0.5
1
1.5
2
2.5
3
3.5
4
0
1
2
3
4
5
6
7
8
9


Velocity vector with no gate
Y
X
0
0.5
1
1.5
2
2.5
3
3.5
4
0
1
2
3
4
5
6
7
8
9

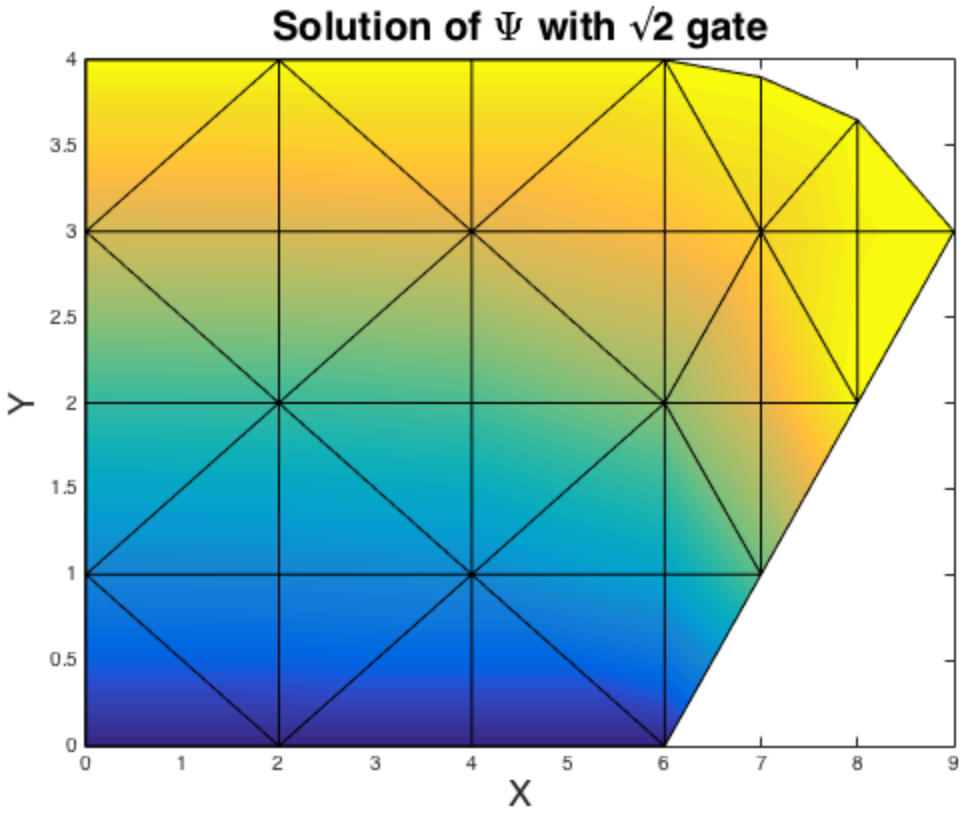
with short gate

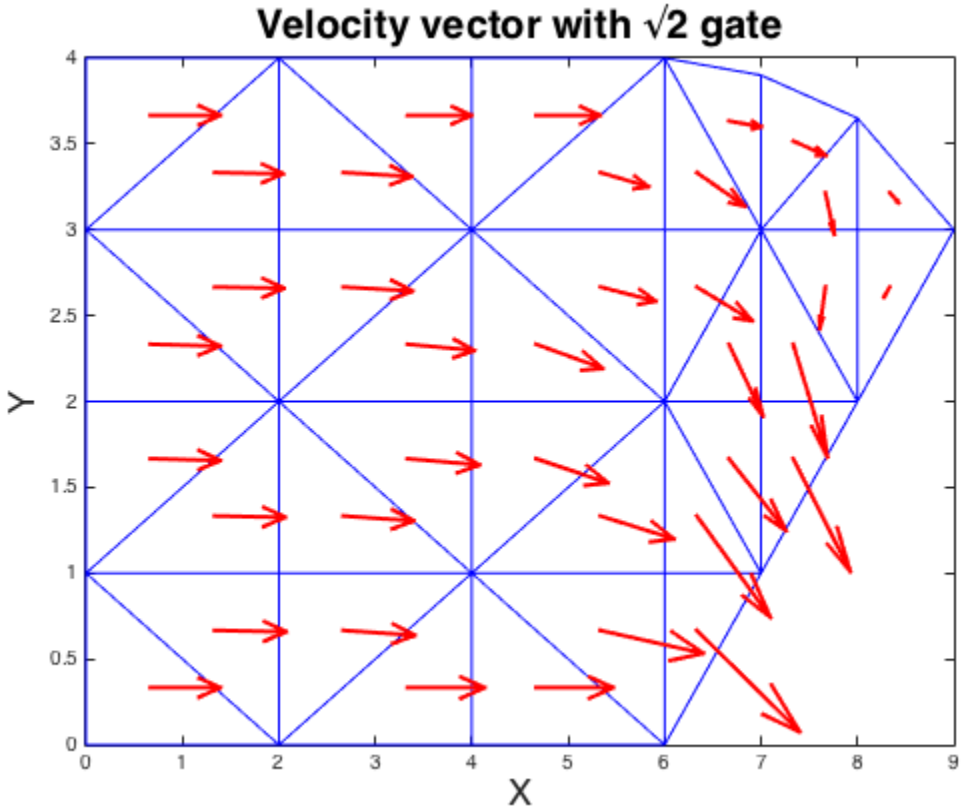
make a matrix

```
clearvars
ele=dlmread('hw5.ele1.txt');
nod=dlmread('hw5.nod.txt');
a_mat=zeros(length(nod));
c_mat=a_mat;
mark=ones(3);
for k=1:length(ele)
    e_nod=ele(k,2:4);
    x_e=nod(e_nod,2);
    y_e=nod(e_nod,3);
    delt_x(1)=x_e(2)-x_e(3);
    delt_x(2)=x_e(3)-x_e(1);
    delt_x(3)=x_e(1)-x_e(2);
    delt_y(1)=y_e(2)-y_e(3);
    delt_y(2)=y_e(3)-y_e(1);
    delt_y(3)=y_e(1)-y_e(2);
    a_area=0.5*(x_e(1)*delt_y(1)+x_e(2)*delt_y(2)+x_e(3)*delt_y(3));
    for i=1:3
        for j=1:3
            amat_e(i,j)=-(delt_x(i)*delt_x(j)+delt_y(i)*delt_y(j))/
(4*a_area);

 a_mat(e_nod(i),e_nod(j))=a_mat(e_nod(i),e_nod(j))+amat_e(i,j);

 c_mat(e_nod(i),e_nod(j))=c_mat(e_nod(i),e_nod(j))+mark(i,j);
        end
    end
end
% modify A matrix base on the boundary condition
index_1=nod(nod(:,3)==0,1);
index_2=nod(nod(:,2)==0,1);
index_3=nod(nod(:,3)==4,1);
index_3=[index_3;24;27;28];
index_master=[index_1;index_2;index_3];
a_mat(index_master,:)=0;
%a_mat(:,index_master)=0;
for i=1:length(index_master)
    a_mat(index_master(i),index_master(i))=1;
end
% add gate to a_mat
gate_l=sqrt(2);
dis_25=sqrt((nod(28,2)-nod(25,2))^2+(nod(28,3)-nod(25,3))^2);
dis_21=sqrt((nod(28,2)-nod(21,2))^2+(nod(28,3)-nod(21,3))^2);
dis_16=sqrt((nod(28,2)-nod(16,2))^2+(nod(28,3)-nod(16,3))^2);
if gate_l>=dis_21
    disp('Gate include node 21 25 28')
    a_mat(21,:)=0;
    a_mat(25,:)=0;
    a_mat(21,21)=1;
```

```
    a_mat(25,25)=1;
elseif gate_l>=dis_25
    disp('Gate include node 25 28')
    a_mat(25,:)=0;
    a_mat(25,25)=1;
end
% build B array
b_ary=zeros(length(nod),1);
b_ary(index_1)=0;
b_ary(index_2)=nod(nod(:,2)==0,3)/4;
b_ary(index_3)=1;
% add gate to B array
b_ary(25)=1;
% solve for solution
u=a_mat\b_ary;
% plot
figure()
triplot([ele(:,2),ele(:,3),ele(:,4)],nod(:,2),nod(:,3));
hold on
trisurf([ele(:,2),ele(:,3),ele(:,4)],nod(:,2),nod(:,3),u,'FaceColor','interp');
title('Solution of \Psi with \surd2 gate','fontsize',20)
xlabel('X','fontsize',18)
ylabel('Y','fontsize',18)
% velocity
for k=1:length(ele)
    e_nod=ele(k,2:4);
    x_e=nod(e_nod,2);
    y_e=nod(e_nod,3);
    x_lab(k)=mean(x_e);
    y_lab(k)=mean(y_e);
    delt_x(1)=x_e(2)-x_e(3);
    delt_x(2)=x_e(3)-x_e(1);
    delt_x(3)=x_e(1)-x_e(2);
    delt_y(1)=y_e(2)-y_e(3);
    delt_y(2)=y_e(3)-y_e(1);
    delt_y(3)=y_e(1)-y_e(2);
    a_area=0.5*(x_e(1)*delt_y(1)+x_e(2)*delt_y(2)+x_e(3)*delt_y(3));
    fl=u(e_nod);
    v_x(k)=fl(1)*delt_y(1)/(2*a_area)+fl(2)*delt_y(2)/
(2*a_area)+fl(3)*delt_y(3)/(2*a_area);
    v_y(k)=-(fl(1)*delt_x(1)/(2*a_area)+fl(2)*delt_x(2)/
(2*a_area)+fl(3)*delt_x(3)/(2*a_area));
end
figure()
triplot([ele(:,2),ele(:,3),ele(:,4)],nod(:,2),nod(:,3));
hold on
%trisurf([ele(:,2),ele(:,3),ele(:,4)],nod(:,2),nod(:,3),u,'FaceColor','interp');
quiver(x_lab,y_lab,v_y,-v_x,'linewidth',2,'color','red')
title('Velocity vector with \surd2 gate','fontsize',20)
xlabel('X','fontsize',18)
ylabel('Y','fontsize',18)
```

```
Gate include node 25 28
```


Solution of Ψ with √2 gate
Y
X
0
0.5
1
1.5
2
2.5
3
3.5
4
0
1
2
3
4
5
6
7
8
9


Velocity vector with √2 gate
Y
X
0
0.5
1
1.5
2
2.5
3
3.5
4
0
1
2
3
4
5
6
7
8
9

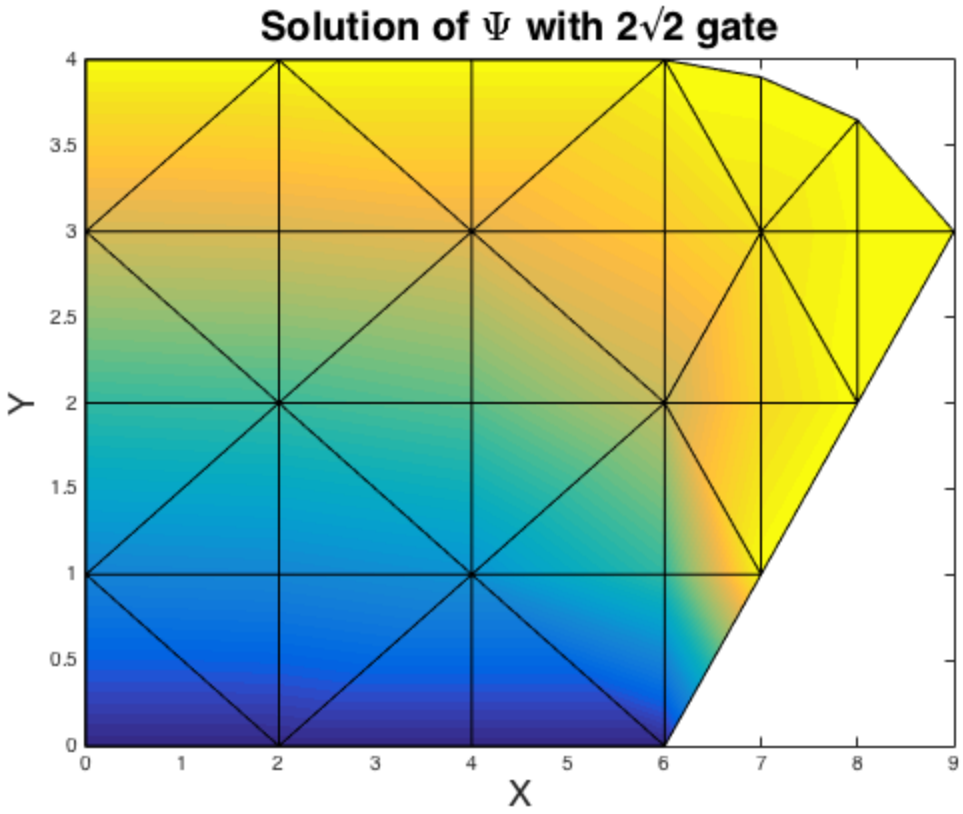
with long gate

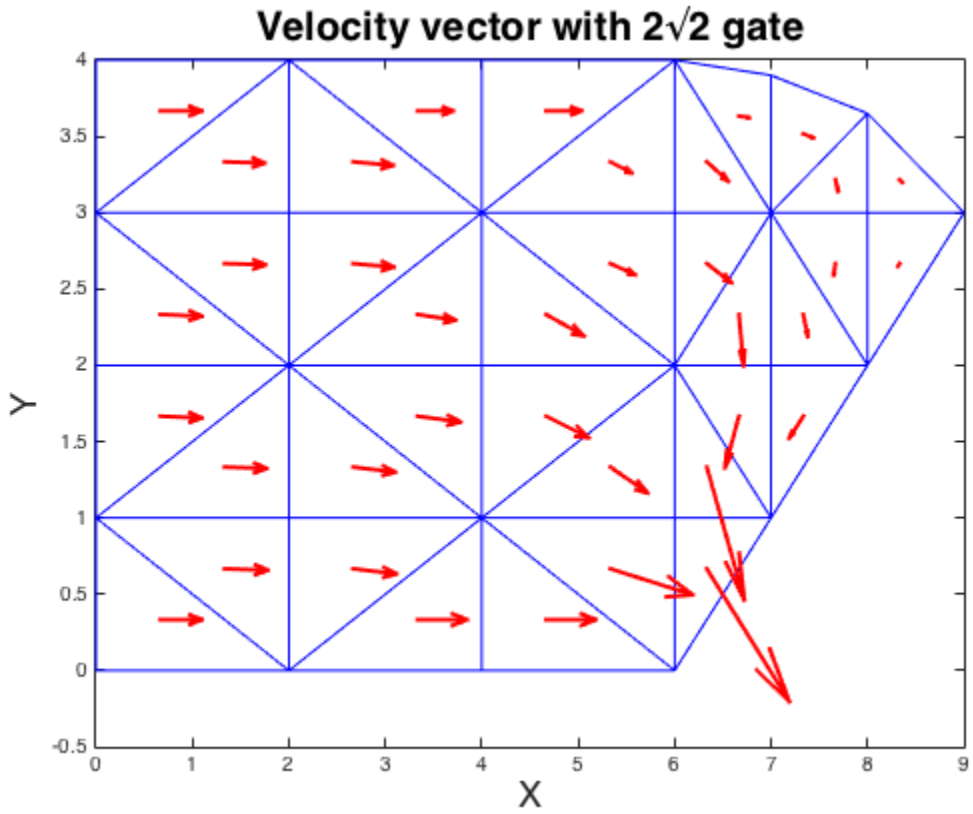
make a matrix

```
clearvars
ele=dlmread('hw5.ele1.txt');
nod=dlmread('hw5.nod.txt');
a_mat=zeros(length(nod));
c_mat=a_mat;
mark=ones(3);
for k=1:length(ele)
    e_nod=ele(k,2:4);
    x_e=nod(e_nod,2);
    y_e=nod(e_nod,3);
    delt_x(1)=x_e(2)-x_e(3);
    delt_x(2)=x_e(3)-x_e(1);
    delt_x(3)=x_e(1)-x_e(2);
    delt_y(1)=y_e(2)-y_e(3);
    delt_y(2)=y_e(3)-y_e(1);
    delt_y(3)=y_e(1)-y_e(2);
    a_area=0.5*(x_e(1)*delt_y(1)+x_e(2)*delt_y(2)+x_e(3)*delt_y(3));
    for i=1:3
        for j=1:3
            amat_e(i,j)=-(delt_x(i)*delt_x(j)+delt_y(i)*delt_y(j))/
(4*a_area);

 a_mat(e_nod(i),e_nod(j))=a_mat(e_nod(i),e_nod(j))+amat_e(i,j);

 c_mat(e_nod(i),e_nod(j))=c_mat(e_nod(i),e_nod(j))+mark(i,j);
        end
    end
end
% modify A matrix base on the boundary condition
index_1=nod(nod(:,3)==0,1);
index_2=nod(nod(:,2)==0,1);
index_3=nod(nod(:,3)==4,1);
index_3=[index_3;24;27;28];
index_master=[index_1;index_2;index_3];
a_mat(index_master,:)=0;
%a_mat(:,index_master)=0;
for i=1:length(index_master)
    a_mat(index_master(i),index_master(i))=1;
end
% add gate to a_mat
gate_l=2*sqrt(2);
dis_25=sqrt((nod(28,2)-nod(25,2))^2+(nod(28,3)-nod(25,3))^2);
dis_21=sqrt((nod(28,2)-nod(21,2))^2+(nod(28,3)-nod(21,3))^2);
dis_16=sqrt((nod(28,2)-nod(16,2))^2+(nod(28,3)-nod(16,3))^2);
if gate_l>=dis_21
    disp('Gate include node 21 25 28')
    a_mat(21,:)=0;
    a_mat(25,:)=0;
    a_mat(21,21)=1;
```

```
    a_mat(25,25)=1;
elseif gate_l>=dis_25
    disp('Gate include node 25 28')
    a_mat(25,:)=0;
    a_mat(25,25)=1;
end
% build B array
b_ary=zeros(length(nod),1);
b_ary(index_1)=0;
b_ary(index_2)=nod(nod(:,2)==0,3)/4;
b_ary(index_3)=1;
% add gate to B array
b_ary(25)=1;
b_ary(21)=1;
% solve for solution
u=a_mat\b_ary;
% plot
figure()
triplot([ele(:,2),ele(:,3),ele(:,4)],nod(:,2),nod(:,3));
hold on
trisurf([ele(:,2),ele(:,3),ele(:,4)],nod(:,2),nod(:,3),u,'FaceColor','interp');
title('Solution of \Psi with 2\surd2 gate','fontsize',20)
xlabel('X','fontsize',18)
ylabel('Y','fontsize',18)
% velocity
for k=1:length(ele)
    e_nod=ele(k,2:4);
    x_e=nod(e_nod,2);
    y_e=nod(e_nod,3);
    x_lab(k)=mean(x_e);
    y_lab(k)=mean(y_e);
    delt_x(1)=x_e(2)-x_e(3);
    delt_x(2)=x_e(3)-x_e(1);
    delt_x(3)=x_e(1)-x_e(2);
    delt_y(1)=y_e(2)-y_e(3);
    delt_y(2)=y_e(3)-y_e(1);
    delt_y(3)=y_e(1)-y_e(2);
    a_area=0.5*(x_e(1)*delt_y(1)+x_e(2)*delt_y(2)+x_e(3)*delt_y(3));
    fl=u(e_nod);
    v_x(k)=fl(1)*delt_y(1)/(2*a_area)+fl(2)*delt_y(2)/
(2*a_area)+fl(3)*delt_y(3)/(2*a_area);
    v_y(k)=-(fl(1)*delt_x(1)/(2*a_area)+fl(2)*delt_x(2)/
(2*a_area)+fl(3)*delt_x(3)/(2*a_area));
end
figure()
triplot([ele(:,2),ele(:,3),ele(:,4)],nod(:,2),nod(:,3));
hold on
%trisurf([ele(:,2),ele(:,3),ele(:,4)],nod(:,2),nod(:,3),u,'FaceColor','interp');
quiver(x_lab,y_lab,v_y,-v_x,'linewidth',2,'color','red')
title('Velocity vector with 2\surd2 gate','fontsize',20)
xlabel('X','fontsize',18)
ylabel('Y','fontsize',18)
```

Gate include node 21 25 28


Solution of Ψ with 2√2 gate
Y
X
0
0.5
1
1.5
2
2.5
3
3.5
4
0
1
2
3
4
5
6
7
8
9


Velocity vector with 2√2 gate
Y
X
-0.5
0
0.5
1
1.5
2
2.5
3
3.5
4
0
1
2
3
4
5
6
7
8
9

Published with MATLAB® R2015a