ภาคผนวก

ในส่วนของภาคผนวกจะแสดงรายละเอียดของโปรแกรม MATLAB ที่ในการคำนวณเพื่อแก้ปัญหาศักย์ของสมการลาปลาซ ของตัวอย่างที่ 5.1-5.9 โดยแบ่งเนื้อหาออกเป็น 2 ส่วนคือ ส่วนโปรแกรมหลัก และส่วนของโปรแกรมย่อย ดังต่อไปนี้

ส่วนที่ 1 โปรแกรมหลัก (Main Program)

```
%*********************************************************************
%This is a program to solve the boundary problem
%By using Quadrtic element MQBEM51
%We use 10-point Gaussian quadrature method to compute
%integration values
%*********************************************************************
% Set ep(g) and w(g) values for 10-point Gaussian quadrature
%*********************************************************************
ep=[0.973906528517172;-0.973906528517172;0.865063366688985;-
   0.865063366688985;0.679409568299024;0.679409568299024;0.433395394
   129247;-0.433395394129247;0.148874338981631;-0.148874338981631];
w=[0.066671344308688;0.066671344308688;0.149451349150581;0.1494513491
   50581;0.219086362515982;0.219086362515982;0.269266719309996;0.2692
   66719309996;0.295524224714753;0.295524224714753];
%*********************************************************************
% Receive the input data and compute coordinate matrix
%*********************************************************************
disp(' Welcome to QABEM51 program')
n=input('Enter the number of boundary node: ');
X=coorx(n);
Y=coory(n);%coordinate of boundary nodes
%*********************************************************************
Hhat=matrixhq(n,X,Y,ep,w); % This is the matrix H-hat
G=matrixgq(n,X,Y,ep,w);    % This is the matrix G
%********************************************************************
%   input known values of U and Q on boundary of square
%********************************************************************
  N=ncode(n);
  t=ncodv(Y,n);
%********************************************************************
%       Changing an system of equation to be in form
%                       Ax=B
%       and solve to obtain the unknown values of U and Q
%********************************************************************
A=cheql(N,Hhat,G,t,n);
b=cheqr(N,Hhat,G,t,n);
B=b*t;
XX=(inv(A))*B;              % XX is x in Ax=B
%********************************************************************
%            Show  the matrix U and Q
%********************************************************************
U=matrixu(t,n,XX);
Q=matrixq(t,n,XX);
nod=1:1:n;
node=(nod');
disp('**********************************************')
disp('     node          X          Y        u        deriv.')
```

```
for i=1:n
      solu(i,1)=node(i,1);
      solu(i,2)=X(i,1);
      solu(i,3)=Y(i,1);
      solu(i,4)=U(i,1);
      solu(i,5)=Q(i,1);
end
solu
%**********************************************************************
% Next, we start to compute for internal nodes
%**********************************************************************
% ------- step 1 : input the data to compute ----------
disp('**************************************************')
l=input(' Enter the number of internal nodes: ');
n1=l;
disp('Next ,input the coordinate(X,Y) of each int node')
disp('**************************************************')
for i=1:l
fprintf('X-coordinate of node %.0f is',i);
Xin1(i)=input(': ');
fprintf('Y-coordinate of node %.0f is',i);
Yin1(i)=input(': ');
end
Xin=Xin1';
Yin=Yin1';
% ------- step 2 : Construct matrix H and G -----------
% both H2 and G2 are constructed in the same
% way to construct Hhat and G
H2=matrixhq2(n,l,X,Y,Xin,Yin,ep,w);
G2=matrixgq2(n,l,X,Y,Xin,Yin,ep,w);
% --- step 3 : compute the solutions at each int node --
Usol=0.5((G2)*(Q)-(H2)*(U));
MLBEM51=mlbem(n,l,n1,X,Y,Xin,Yin,ep,w);
for i=1:l
    Uvalex(i,1)=1+Xin(i)+2*Yin(i);
end
for i=1:l
    Error(i,1)=0;
    Errorq(i,1)=abs(Uvalex(i,1)-Usol(i,1));
    Errorl(i,1)=abs(Uvalex(i,1)-MLBEM51(i,1));
end
disp('---------- The solutions are shown as follows -------------')
nod=1:1:l;
node=(nod');
disp(' node    X     Y    U-MLBEM51   U-MQBEM51  U-Exact    EL   EQ')
for i=1:l
    solu1(i,1)=node(i,1);
    solu1(i,2)=Xin(i,1);
    solu1(i,3)=Yin(i,1);
    solu1(i,4)=MLBEM51(i,1);
    solu1(i,5)=Usol(i,1);
    solu1(i,6)=Uvalex(i,1);
    solu1(i,7)=Errorl(i,1);
    solu1(i,8)=Errorq(i,1);
end
solu1
disp('*****************************************************************')
%////////////////// Graph of Normal derivative ///////////////
```

```
D=1:1:1;
Subplot(2,1,1);
plot(D,Usol,'m-O',D,Uvalex,'b-s',D,MLBEM51,'g-*','LineWidth',2);
xlabel('Node');ylabel('Solution of u');
grid on
title('Solution of u from MQBEM51 and MLBEM51');
legend('MQBEM51','Exact','MLBEM51');
Subplot(2,1,2);
plot(D,Errorq,'m-O',D,Errorl,'g-*','LineWidth',2);
axis([0.9 4.1 -0.005 0.02])
xlabel('Node');ylabel('Error');
grid on
title('Error of u from MQBEM51 and MLBEM51');
legend('Error-Q','Error-L');


%*******************************************************************
%This is a program to solve the boundary problem  for Ex 5.2
%By using Quadrtic element MQBEM52
%We use 10-point Gaussian quadrature method to compute
%integration values
%*******************************************************************
% Set ep(g) and w(g) values for 10-point Gaussian quadrature
%*******************************************************************
ep=[0.973906528517172;-0.973906528517172;0.865063366688985;-
   0.865063366688985;0.679409568299024;0.679409568299024;0.433395394
   129247;-0.433395394129247;0.148874338981631;-0.148874338981631];
w=[0.066671344308688;0.066671344308688;0.149451349150581;0.1494513491
   50581;0.219086362515982;0.219086362515982;0.269266719309996;0.2692
   66719309996;0.295524224714753;0.295524224714753];
%*******************************************************************
% Receive the input data and compute coordinate matrix
%*******************************************************************
disp(' Welcome to QABEM52 program')
n=input('Enter the number of boundary node: ');
X=coorx(n);
Y=coory(n);%coordinate of boundary nodes
%*******************************************************************
Hhat=matrixhq(n,X,Y,ep,w); % This is the matrix H-hat
G=matrixgq(n,X,Y,ep,w);    % This is the matrix G
%******************************************************************
%   input known values of U and Q on boundary of square
%******************************************************************
  N=ncode(n);
  t=ncodv(Y,n);%will be used in computing B of Ax=B steps
%******************************************************************
%       Changing an system of equation to be in form
%                       Ax=B
%       and solve to obtain the unknown values of U and Q
%******************************************************************
A=cheql(N,Hhat,G,t,n);
b=cheqr(N,Hhat,G,t,n);
B=b*t;
XX=(inv(A))*B;              % XX is x in Ax=B
%******************************************************************
%            Show  the matrix U and Q
%******************************************************************
U=matrixu(t,n,XX);
```

```
Q=matrixq(t,n,XX);
nod=1:1:n;
node=(nod');
disp('************************************************')
disp('     node          X          Y        u       deriv.')
for i=1:n
      solu(i,1)=node(i,1);
      solu(i,2)=X(i,1);
      solu(i,3)=Y(i,1);
      solu(i,4)=U(i,1);
      solu(i,5)=Q(i,1);
end
solu
%*******************************************************************
% Next, we start to compute for internal nodes
%*******************************************************************
% ------- step 1 : input the data to compute ----------
disp('************************************************')
l=input(' Enter the number of internal nodes: ');
n1=l;
disp('Next ,input the coordinate(X,Y) of each int node')
disp('************************************************')
for i=1:l
fprintf('X-coordinate of node %.0f is',i);
Xin1(i)=input(': ');
fprintf('Y-coordinate of node %.0f is',i);
Yin1(i)=input(': ');
end
Xin=Xin1';
Yin=Yin1';
% ------- step 2 : Construct matrix H and G -----------
% both H2 and G2 are constructed in the same
% way to construct Hhat and G
H2=matrixhq2(n,l,X,Y,Xin,Yin,ep,w);
G2=matrixgq2(n,l,X,Y,Xin,Yin,ep,w);
% --- step 3 : compute the solutions at each int node --
Usol=0.5*((G2)*(Q)-(H2)*(U));
MLBEM52=mlbem(n,l,n1,X,Y,Xin,Yin,ep,w);
for i=1:l
    Uvalex(i,1)=300-50*Xin(i);
end
for i=1:l
    Error(i,1)=0;
    Errorq(i,1)=abs(Uvalex(i,1)-Usol(i,1));
    Errorl(i,1)=abs(Uvalex(i,1)-MLBEM52(i,1));
end
disp('--------------- The solutions are shown as follows ----------')
nod=1:1:l;
node=(nod');
disp('node     X      Y  U-MLBEM52   U-MQBEM52  U-Exact   EL      EQ')
for i=1:l
    solu1(i,1)=node(i,1);
    solu1(i,2)=Xin(i,1);
    solu1(i,3)=Yin(i,1);
    solu1(i,4)=MLBEM52(i,1);
    solu1(i,5)=Usol(i,1);
    solu1(i,6)=Uvalex(i,1);
    solu1(i,7)=Errorl(i,1);
```

```
    solu1(i,8)=Errorq(i,1);
end
solu1
disp('**********************************************************')
%////////////////// Graph of Normal derivative ///////////////
D=1:1:1;
Subplot(2,1,1);
plot(D,Usol,'m-O',D,Uvalex,'b-s',D,MLBEM52,'g-*','LineWidth',2);
xlabel('Node');ylabel('Solution of u');
grid on
title('Solution of u from MQBEM52 and MLBEM52 ');
legend('MQBEM52','Exact','MLBEM52');
Subplot(2,1,2);
plot(D,Errorq,'m-O',D,Errorl,'g-*','LineWidth',2);
axis([0.9 5.1 -0.2 4])
xlabel('Node');ylabel('Error');
grid on
title('Error of u from MQBEM52 and MLBEM52');
legend('Error-Q','Error-L');


%*******************************************************************
%This is a program to solve the boundary problem  for Ex 5.3
%By using Quadrtic element MQBEM53
%We use 10-point Gaussian quadrature method to compute
%integration values
%*******************************************************************
% Set ep(g) and w(g) values for 10-point Gaussian quadrature
%*******************************************************************
ep=[0.973906528517172;-0.973906528517172;0.865063366688985;-
   0.865063366688985;0.679409568299024;0.679409568299024;0.433395394
   129247;-0.433395394129247;0.148874338981631;-0.148874338981631];
w=[0.066671344308688;0.066671344308688;0.149451349150581;0.1494513491
   50581;0.219086362515982;0.219086362515982;0.269266719309996;0.2692
   66719309996;0.295524224714753;0.295524224714753];
%*******************************************************************
% Receive the input data and compute coordinate matrix
%*******************************************************************
disp(' Welcome to QABEM53 program')
n=input('Enter the number of boundary node: ');
X=coorx(n);
Y=coory(n);%coordinate of boundary nodes
%*******************************************************************
Hhat=matrixhq(n,X,Y,ep,w); % This is the matrix H-hat
G=matrixgq(n,X,Y,ep,w);    % This is the matrix G
%******************************************************************
%   input known values of U and Q on boundary of square
%******************************************************************
  N=ncode(n);
  t=ncodv(Y,n);%will be used in computing B of Ax=B steps
%******************************************************************
%       Changing an system of equation to be in form
%                       Ax=B
%       and solve to obtain the unknown values of U and Q
%******************************************************************
A=cheql(N,Hhat,G,t,n);
b=cheqr(N,Hhat,G,t,n);
B=b*t;
```

```
XX=(inv(A))*B;                  % XX is x in Ax=B
%*******************************************************************
%             Show  the matrix U and Q
%*******************************************************************
U=matrixu(t,n,XX);
Q=matrixq(t,n,XX);
nod=1:1:n;
node=(nod');
disp('************************************************')
disp('     node          X         Y       u       deriv.')
for i=1:n
      solu(i,1)=node(i,1);
      solu(i,2)=X(i,1);
      solu(i,3)=Y(i,1);
      solu(i,4)=U(i,1);
      solu(i,5)=Q(i,1);
end
solu
%*******************************************************************
% Next, we start to compute for internal nodes
%*******************************************************************
% ------- step 1 : input the data to compute ----------
disp('************************************************')
l=input(' Enter the number of internal nodes: ');
n1=l;
disp('Next ,input the coordinate(X,Y) of each int node')
disp('************************************************')
for i=1:l
fprintf('X-coordinate of node %.0f is',i);
Xin1(i)=input(': ');
fprintf('Y-coordinate of node %.0f is',i);
Yin1(i)=input(': ');
end
Xin=Xin1';
Yin=Yin1';
% ------- step 2 : Construct matrix H and G -----------
% both H2 and G2 are constructed in the same
% way to construct Hhat and G
H2=matrixhq2(n,l,X,Y,Xin,Yin,ep,w);
G2=matrixgq2(n,l,X,Y,Xin,Yin,ep,w);
% --- step 3 : compute the solutions at each int node --
Usol=0.5*((G2)*(Q)-(H2)*(U));
MLBEM53=mlbem(n,l,n1,X,Y,Xin,Yin,ep,w);
Uvalex=[0.432028;0.540529;0.432028;0.345349;0.250000;0.176531;0.09541
4];
for i=1:l
    Error(i,1)=0;
    Errorq(i,1)=abs(Uvalex(i,1)-Usol(i,1));
    Errorl(i,1)=abs(Uvalex(i,1)-MLBEM53(i,1));
end
disp('-------- The solutions are shown as follows -----------')
nod=1:1:l;
node=(nod');
disp('     node        X         Y      U-MLBEM53   U-MQBEM53  U-Exact
EL       EQ')
for i=1:l
    solu1(i,1)=node(i,1);
    solu1(i,2)=Xin(i,1);
```

```
    solul(i,3)=Yin(i,1);
    solul(i,4)=MLBEM53(i,1);
    solul(i,5)=Usol(i,1);
    solul(i,6)=Uvalex(i,1);
    solul(i,7)=Errorl(i,1);
    solul(i,8)=Errorq(i,1);
end
solul
disp('*****************************************************')
%///////////////// Graph of Normal derivative ///////////////
D=1:1:1;
Subplot(2,1,1);
plot(D,Usol,'m-O',D,Uvalex,'b-s',D,MLBEM53,'g-*','LineWidth',2);
xlabel('Node');ylabel('Solution of u');
grid on
title('Solution of u from MQBEM53 and MLBEM53 ');
legend('MQBEM53','Exact','MLBEM53');
Subplot(2,1,2);
plot(D,Errorq,'m-O',D,Errorl,'g-*','LineWidth',2);
axis([0.9 7.1 -0.001 0.005])
xlabel('Node');ylabel('Error');
grid on
title('Error of u from MQBEM53 and MLBEM53');
legend('Error-Q','Error-L');


%*******************************************************************
%This is a program to solve the boundary problem  for Ex 5.4
%By using Quadrtic element MQBEM54
%We use 10-point Gaussian quadrature method to compute
%integration values
%*******************************************************************
% Set ep(g) and w(g) values for 10-point Gaussian quadrature
%*******************************************************************
ep=[0.973906528517172;-0.973906528517172;0.865063366688985;-
   0.865063366688985;0.679409568299024;0.679409568299024;0.433395394
   129247;-0.433395394129247;0.148874338981631;-0.148874338981631];
w=[0.066671344308688;0.066671344308688;0.149451349150581;0.1494513491
   50581;0.219086362515982;0.219086362515982;0.269266719309996;0.2692
   66719309996;0.295524224714753;0.295524224714753];
%*******************************************************************
% Receive the input data and compute coordinate matrix
%*******************************************************************
disp(' Welcome to QABEM54 program')
n=12;
X=[0;1;2;2;2;2;2;1;0;0;0;0];
Y=[0;0;0;1;2;3;4;4;4;3;2;1];%coordinate of boundary nodes
%*******************************************************************
Hhat=matrixhq(n,X,Y,ep,w); % This is the matrix H-hat
G=matrixgq(n,X,Y,ep,w);    % This is the matrix G
%******************************************************************
%   input known values of U and Q on boundary of square
%******************************************************************
  N=ncode(n);
  t=ncodv(Y,n);%will be used in computing B of Ax=B steps
%******************************************************************
%       Changing an system of equation to be in form
%                     Ax=B
```

```
%       and solve to obtain the unknown values of U and Q
%*******************************************************************
A=cheql(N,Hhat,G,t,n);
b=cheqr(N,Hhat,G,t,n);
B=b*t;
XX=(inv(A))*B;              % XX is x in Ax=B
%*******************************************************************
%            Show  the matrix U and Q
%*******************************************************************
U=matrixu(t,n,XX);
Q=matrixq(t,n,XX);
nod=1:1:n;
node=(nod');
disp('************************************************')
disp('     node         X         Y       u       deriv.')
for i=1:n
      solu(i,1)=node(i,1);
      solu(i,2)=X(i,1);
      solu(i,3)=Y(i,1);
      solu(i,4)=U(i,1);
      solu(i,5)=Q(i,1);
end
solu
%*******************************************************************
% Next, we start to compute for internal nodes
%*******************************************************************
% ------- step 1 : input the data to compute ----------
disp('************************************************')
l=input(' Enter the number of internal nodes: ');
n1=l;
disp('Next ,input the coordinate(X,Y) of each int node')
disp('************************************************')
for i=1:l
fprintf('X-coordinate of node %.0f is',i);
Xin1(i)=input(': ');
fprintf('Y-coordinate of node %.0f is',i);
Yin1(i)=input(': ');
end
Xin=Xin1';
Yin=Yin1';
% ------- step 2 : Construct matrix H and G -----------
% both H2 and G2 are constructed in the same
% way to construct Hhat and G
H2=matrixhq2(n,l,X,Y,Xin,Yin,ep,w);
G2=matrixgq2(n,l,X,Y,Xin,Yin,ep,w);
% --- step 3 : compute the solutions at each int node --
Usol=0.5*((G2)*(Q)-(H2)*(U));
MLBEM54=mlbem(n,l,n1,X,Y,Xin,Yin,ep,w);
for i=1:l
    Uvalex(i,1)=Xin(i)^3-3*Xin(i)*Yin(i)^2;
end
for i=1:l
    Error(i,1)=0;
    Errorq(i,1)=abs(Uvalex(i,1)-Usol(i,1));
    Errorl(i,1)=abs(Uvalex(i,1)-MLBEM54(i,1));
end
disp('----------- The solutions are shown as follows ------------')
nod=1:1:l;
```

```
node=(nod');
disp('node     X      Y      U-MLBEM54    U-MQBEM54  U-Exact   EL     EQ')
for i=1:1
    solu1(i,1)=node(i,1);
    solu1(i,2)=Xin(i,1);
    solu1(i,3)=Yin(i,1);
    solu1(i,4)=MLBEM54(i,1);
    solu1(i,5)=Usol(i,1);
    solu1(i,6)=Uvalex(i,1);
    solu1(i,7)=Errorl(i,1);
    solu1(i,8)=Errorq(i,1);
end
solu1
disp('************************************************************')
%/////////////////  Graph of Normal derivative  ///////////////
D=1:1:1;
Subplot(2,1,1);
plot(D,Usol,'m-O',D,Uvalex,'b-s',D,MLBEM54,'g-*','LineWidth',2);
xlabel('Node');ylabel('Solution of u');
grid on
title('Solution of u from MQBEM54 and MLBEM54');
legend('MQBEM54','Exact','MLBEM54');
Subplot(2,1,2);
plot(D,Errorq,'m-O',D,Errorl,'g-*','LineWidth',2);
axis([0.9 7.1 -0.03 0.36])
xlabel('Node');ylabel('Error');
grid on
title('Error of u from MQBEM54 and MLBEM54');
legend('Error-Q','Error-L');



%*************************************************************
%This is a program to solve the boundary problem  for Ex 5.5
%By using Quadrtic element MQBEM55
%We use 10-point Gaussian quadrature method to compute
%integration values
%*************************************************************
% Set ep(g) and w(g) values for 10-point Gaussian quadrature
%*************************************************************
ep=[0.973906528517172;-0.973906528517172;0.865063366688985;-
   0.865063366688985;0.679409568299024;0.679409568299024;0.433395394
   129247;-0.433395394129247;0.148874338981631;-0.148874338981631];
w=[0.066671344308688;0.066671344308688;0.149451349150581;0.1494513491
   50581;0.219086362515982;0.219086362515982;0.269266719309996;0.2692
   66719309996;0.295524224714753;0.295524224714753];
%*************************************************************
% Receive the input data and compute coordinate matrix
%*************************************************************
disp(' Welcome to QABEM55 program')
n=input('Enter the number of boundary node: ');
X=coorx(n);
Y=coory(n);%coordinate of boundary nodes
%*************************************************************
Hhat=matrixhq(n,X,Y,ep,w); % This is the matrix H-hat
G=matrixgq(n,X,Y,ep,w);    % This is the matrix G
%************************************************************
%   input known values of U and Q on boundary of square
```

```
%*******************************************************************
  N=ncode(n);
  t=ncodv(Y,n);%will be used in computing B of Ax=B steps
%*******************************************************************
%       Changing an system of equation to be in form
%                       Ax=B
%       and solve to obtain the unknown values of U and Q
%*******************************************************************
A=cheql(N,Hhat,G,t,n);
b=cheqr(N,Hhat,G,t,n);
B=b*t;
XX=(inv(A))*B;              % XX is x in Ax=B
%*******************************************************************
%            Show  the matrix U and Q
%*******************************************************************
U=matrixu(t,n,XX);
Q=matrixq(t,n,XX);
nod=1:1:n;
node=(nod');
disp('***********************************************')
disp('    node         X         Y       u       deriv.')
for i=1:n
      solu(i,1)=node(i,1);
      solu(i,2)=X(i,1);
      solu(i,3)=Y(i,1);
      solu(i,4)=U(i,1);
      solu(i,5)=Q(i,1);
end
solu
%*******************************************************************
% Next, we start to compute for internal nodes
%*******************************************************************
% ------- step 1 : input the data to compute ----------
disp('***********************************************')
l=input(' Enter the number of internal nodes: ');
n1=l;
disp('Next ,input the coordinate(X,Y) of each int node')
disp('***********************************************')
for i=1:l
fprintf('X-coordinate of node %.0f is',i);
Xin1(i)=input(': ');
fprintf('Y-coordinate of node %.0f is',i);
Yin1(i)=input(': ');
end
Xin=Xin1';
Yin=Yin1';
% ------- step 2 : Construct matrix H and G -----------
% both H2 and G2 are constructed in the same
% way to construct Hhat and G
H2=matrixhq2(n,l,X,Y,Xin,Yin,ep,w);
G2=matrixgq2(n,l,X,Y,Xin,Yin,ep,w);
% --- step 3 : compute the solutions at each int node --
Usol=0.5*((G2)*(Q)-(H2)*(U));
MLBEM55=mlbem(n,l,n1,X,Y,Xin,Yin,ep,w);
for i=1:l
    Uvalex(i,1)=Xin(i)^2-Yin(i)^2;
end
for i=1:l
```

```
    Error(i,1)=0;
    Errorq(i,1)=abs(Uvalex(i,1)-Usol(i,1));
    Errorl(i,1)=abs(Uvalex(i,1)-MLBEM55(i,1));
end
disp('------------ The solutions are shown as follows -----------')
nod=1:1:l;
node=(nod');
disp(' node    X    Y    U-MLBEM55   U-MQBEM55  U-Exact   EL    EQ')
for i=1:l
    solu1(i,1)=node(i,1);
    solu1(i,2)=Xin(i,1);
    solu1(i,3)=Yin(i,1);
    solu1(i,4)=MLBEM55(i,1);
    solu1(i,5)=Usol(i,1);
    solu1(i,6)=Uvalex(i,1);
    solu1(i,7)=Errorl(i,1);
    solu1(i,8)=Errorq(i,1);
end
solu1
disp('**************************************************************')
%/////////////////  Graph of Normal derivative  ///////////////
D=1:1:l;
Subplot(2,1,1);
plot(D,Usol,'m-O',D,Uvalex,'b-s',D,MLBEM55,'g-*','LineWidth',2);
xlabel('Node');ylabel('Solution of u');
grid on
title('Solution of u from MQBEM55 and MLBEM55');
legend('MQBEM55','Exact','MLBEM55');
Subplot(2,1,2);
plot(D,Errorq,'m-O',D,Errorl,'g-*','LineWidth',2);
axis([0.9 5.1 -0.001 0.004])
xlabel('Node');ylabel('Error');
grid on
title('Error of u from MQBEM55 and MLBEM55');
legend('Error-Q','Error-L');


%*******************************************************************
%This is a program to solve the boundary problem  for Ex 5.6
%By using Quadrtic element MQBEM56
%We use 10-point Gaussian quadrature method to compute
%integration values
%*******************************************************************
% Set ep(g) and w(g) values for 10-point Gaussian quadrature
%*******************************************************************
ep=[0.973906528517172;-0.973906528517172;0.865063366688985;-
   0.865063366688985;0.679409568299024;0.679409568299024;0.433395394
   129247;-0.433395394129247;0.148874338981631;-0.148874338981631];
w=[0.066671344308688;0.066671344308688;0.149451349150581;0.1494513491
   50581;0.219086362515982;0.219086362515982;0.269266719309996;0.2692
   66719309996;0.295524224714753;0.295524224714753];
%*******************************************************************
% Receive the input data and compute coordinate matrix
%*******************************************************************
disp(' Welcome to QABEM56 program')
n=input('Enter the number of boundary node: ');
X=coorx(n);
Y=coory(n);%coordinate of boundary nodes
```

```
%*******************************************************************
Hhat=matrixhq(n,X,Y,ep,w); % This is the matrix H-hat
G=matrixgq(n,X,Y,ep,w);    % This is the matrix G
%******************************************************************
%   input known values of U and Q on boundary of square
%******************************************************************
  N=ncode(n);
  t=ncodv(Y,n);%will be used in computing B of Ax=B steps
%******************************************************************
%       Changing an system of equation to be in form
%                       Ax=B
%       and solve to obtain the unknown values of U and Q
%******************************************************************
A=cheql(N,Hhat,G,t,n);
b=cheqr(N,Hhat,G,t,n);
B=b*t;
XX=(inv(A))*B;              % XX is x in Ax=B
%******************************************************************
%            Show  the matrix U and Q
%******************************************************************
U=matrixu(t,n,XX);
Q=matrixq(t,n,XX);
nod=1:1:n;
node=(nod');
disp('***********************************************')
disp('     node          X          Y        u        deriv.')
for i=1:n
      solu(i,1)=node(i,1);
      solu(i,2)=X(i,1);
      solu(i,3)=Y(i,1);
      solu(i,4)=U(i,1);
      solu(i,5)=Q(i,1);
end
solu
%******************************************************************
% Next, we start to compute for internal nodes
%******************************************************************
% ------- step 1 : input the data to compute ----------
disp('***********************************************')
l=input(' Enter the number of internal nodes: ');
n1=l;
disp('Next ,input the coordinate(X,Y) of each int node')
disp('***********************************************')
for i=1:l
fprintf('X-coordinate of node %.0f is',i);
Xin1(i)=input(': ');
fprintf('Y-coordinate of node %.0f is',i);
Yin1(i)=input(': ');
end
Xin=Xin1';
Yin=Yin1';
% ------- step 2 : Construct matrix H and G -----------
% both H2 and G2 are constructed in the same
% way to construct Hhat and G
H2=matrixhq2(n,l,X,Y,Xin,Yin,ep,w);
G2=matrixgq2(n,l,X,Y,Xin,Yin,ep,w);
% --- step 3 : compute the solutions at each int node --
Usol=0.5*((G2)*(Q)-(H2)*(U));
```

```
MLBEM56=mlbem(n,l,n1,X,Y,Xin,Yin,ep,w);
for i=1:l
    Uvalex(i,1)=Xin(i)^2+Yin(i)*(1-Yin(i));
end
for i=1:l
    Error(i,1)=0;
    Errorq(i,1)=abs(Uvalex(i,1)-Usol(i,1));
    Errorl(i,1)=abs(Uvalex(i,1)-MLBEM56(i,1));
end
disp('----------- The solutions are shown as follows -------------')
nod=1:1:l;
node=(nod');
disp(' node     X      Y   U-MLBEM56   U-MQBEM56  U-Exact   EL   EQ')
for i=1:l
    solu1(i,1)=node(i,1);
    solu1(i,2)=Xin(i,1);
    solu1(i,3)=Yin(i,1);
    solu1(i,4)=MLBEM56(i,1);
    solu1(i,5)=Usol(i,1);
    solu1(i,6)=Uvalex(i,1);
    solu1(i,7)=Errorl(i,1);
end
solu1
disp('**********************************************************')
%/////////////////  Graph of Normal derivative  ///////////////
D=1:1:l;
Subplot(2,1,1);
plot(D,Usol,'m-O',D,Uvalex,'b-s',D,MLBEM56,'g-*','LineWidth',2);
xlabel('Node');ylabel('Solution of u');
grid on
title('Solution of u from MQBEM56 and MLBEM56');
legend('MQBEM56','Exact','MLBEM56');
Subplot(2,1,2);
plot(D,Errorq,'m-O',D,Errorl,'g-*','LineWidth',2);
axis([0.9 9.1 -0.00005 0.0009])
xlabel('Node');ylabel('Error');
grid on
title('Error of u from MQBEM56 and MLBEM56');
legend('Error-Q','Error-L');


%*******************************************************************
%This is a program to solve the boundary problem  for Ex 5.7
%By using Quadrtic element MQBEM57
%We use 10-point Gaussian quadrature method to compute
%integration values
%*******************************************************************
% Set ep(g) and w(g) values for 10-point Gaussian quadrature
%*******************************************************************
ep=[0.973906528517172;-0.973906528517172;0.865063366688985;-
   0.865063366688985;0.679409568299024;0.679409568299024;0.433395394
   129247;-0.433395394129247;0.148874338981631;-0.148874338981631];
w=[0.066671344308688;0.066671344308688;0.149451349150581;0.1494513491
   50581;0.219086362515982;0.219086362515982;0.269266719309996;0.2692
   66719309996;0.295524224714753;0.295524224714753];
%*******************************************************************
% Receive the input data and compute coordinate matrix
%*******************************************************************
```

```
disp(' Welcome to QABEM57 program')
n=input('Enter the number of boundary node: ');
for  i=1:n
    X(i,1)=2*cos((i-1)*2*pi/n);
    Y(i,1)=2*sin((i-1)*2*pi/n);
end
%*********************************************************************
Hhat=matrixhq(n,X,Y,ep,w); % This is the matrix H-hat
G=matrixgq(n,X,Y,ep,w);    % This is the matrix G
%********************************************************************
%   input known values of U and Q on boundary of square
%********************************************************************
  N=ncode(n);
  t=ncodv(Y,n);%will be used in computing B of Ax=B steps
%********************************************************************
%       Changing an system of equation to be in form
%                       Ax=B
%       and solve to obtain the unknown values of U and Q
%********************************************************************
A=cheql(N,Hhat,G,t,n);
b=cheqr(N,Hhat,G,t,n);
B=b*t;
XX=(inv(A))*B;              % XX is x in Ax=B
%********************************************************************
%            Show  the matrix U and Q
%********************************************************************
U=matrixu(t,n,XX);
Q=matrixq(t,n,XX);
nod=1:1:n;
node=(nod');
disp('**************************************************')
disp('     node        X         Y       u        deriv.')
for i=1:n
      solu(i,1)=node(i,1);
      solu(i,2)=X(i,1);
      solu(i,3)=Y(i,1);
      solu(i,4)=U(i,1);
      solu(i,5)=Q(i,1);
end
solu
%********************************************************************
% Next, we start to compute for internal nodes
%********************************************************************
% ------- step 1 : input the data to compute ----------
disp('**************************************************')
l=input(' Enter the number of internal nodes: ');
nl=l;
disp('Next ,input the coordinate(X,Y) of each int node')
disp('**************************************************')
for i=1:l
fprintf('X-coordinate of node %.0f is',i);
Xin1(i)=input(': ');
fprintf('Y-coordinate of node %.0f is',i);
Yin1(i)=input(': ');
end
Xin=Xin1';
Yin=Yin1';
% ------- step 2 : Construct matrix H and G -----------
```

```
% both H2 and G2 are constructed in the same
% way to construct Hhat and G
H2=matrixhq2(n,l,X,Y,Xin,Yin,ep,w);
G2=matrixgq2(n,l,X,Y,Xin,Yin,ep,w);
% --- step 3 : compute the solutions at each int node --
Usol=0.5*((G2)*(Q)-(H2)*(U));
MLBEM57=mlbem(n,l,n1,X,Y,Xin,Yin,ep,w);
for i=1:l
    Uvalex(i,1)=Xin(i)+Yin(i);
    Exact1(i,1)=0;
end
for i=1:l
    Error(i,1)=0;
    Errorq(i,1)=abs(Uvalex(i,1)-Usol(i,1));
    Errorl(i,1)=abs(Uvalex(i,1)-MLBEM57(i,1));
end
disp('---------- The solutions are shown as follows ---------------')
nod=1:1:l;
node=(nod');
disp('node     X      Y    U-MLBEM57   U-MQBEM57  U-Exact   EL    EQ')
for i=1:l
    solu1(i,1)=node(i,1);
    solu1(i,2)=Xin(i,1);
    solu1(i,3)=Yin(i,1);
    solu1(i,4)=MLBEM57(i,1);
    solu1(i,5)=Usol(i,1);
    solu1(i,6)=Uvalex(i,1);
    solu1(i,7)=Errorl(i,1);
    solu1(i,8)=Errorq(i,1);
end
solu1
disp('*************************************************************')
%/////////////////  Graph of Normal derivative  ///////////////
D=1:1:l;
Subplot(2,1,1);
plot(D,Usol,'m-O',D,Uvalex,'b-s',D,MLBEM57,'g-*','LineWidth',2);
xlabel('Node');ylabel('Solution of u');
grid on
title('Solution of u from MQBEM57 and MLBEM57');
legend('MQBEM57','Exact','MLBEM57');
Subplot(2,1,2);
plot(D,Errorq,'m-O',D,Errorl,'g-*','LineWidth',2);
axis([0.9 4.1 -0.002 0.0115])
xlabel('Node');ylabel('Error');
grid on
title('Error from MQBEM57 and MLBEM57');
legend('Error-Q','Error-L');
```

```
%*******************************************************************
%This is a program to solve the boundary problem  for Ex 5.8
%By using Quadrtic element MQBEM58
%We use 10-point Gaussian quadrature method to compute
%integration values
%*******************************************************************
% Set ep(g) and w(g) values for 10-point Gaussian quadrature
%*******************************************************************
ep=[0.973906528517172;-0.973906528517172;0.865063366688985;-
   0.865063366688985;0.679409568299024;0.679409568299024;0.433395394
   129247;-0.433395394129247;0.148874338981631;-0.148874338981631];
w=[0.066671344308688;0.066671344308688;0.149451349150581;0.1494513491
   50581;0.219086362515982;0.219086362515982;0.269266719309996;0.2692
   66719309996;0.295524224714753;0.295524224714753];
%*******************************************************************
% Receive the input data and compute coordinate matrix
%*******************************************************************
disp(' Welcome to QABEM58 program')
n=60;
X=[0;0.2;0.4;0.6;0.8;1;1.2;1.4;1.6;1.8;
   2;2.2;2.4;2.6;2.8;3;3.1;3.2;3.3;3.4;3.5;3.6;
   3.7;3.8;3.9;4;4;4;4;4;4;3.8;
   3.6;3.4;3.2;3.0;2.8;2.6;2.4;2.2;2;1.8;
   1.6;1.4;1.2;1;.8;.6;.4;.2;0;0;
   0;0;0;0;0;0;0;0];
Y=[0;0;0;0;0;0;0;0;0;0;
    0;0;0;0;0;0;.46;.6;.71;.8;
    .87;.92;.95;.98;.99;1;1.2;1.4;1.6;1.8;
    2;2;2;2;2;2;2;2;2;2;
    2;2;2;2;2;2;2;2;2;2;2;
    1.8;1.6;1.4;1.2;1;0.8;0.6;0.4;0.2];
%*******************************************************************
Hhat=matrixhq(n,X,Y,ep,w); % This is the matrix H-hat
G=matrixgq(n,X,Y,ep,w);    % This is the matrix G
%******************************************************************
%   input known values of U and Q on boundary of square
%******************************************************************
  N=ncode(n);
  t=ncodv(Y,n);%will be used in computing B of Ax=B steps
%******************************************************************
%       Changing an system of equation to be in form
%                       Ax=B
%       and solve to obtain the unknown values of U and Q
%******************************************************************
A=cheql(N,Hhat,G,t,n);
b=cheqr(N,Hhat,G,t,n);
B=b*t;
XX=(inv(A))*B;             % XX is x in Ax=B
%******************************************************************
%            Show  the matrix U and Q
%******************************************************************
U=matrixu(t,n,XX);
Q=matrixq(t,n,XX);
nod=1:1:n;
node=(nod');
disp('*********************************************')
disp('     node          X          Y        u       deriv.')
```

```
for i=1:n
      solu(i,1)=node(i,1);
      solu(i,2)=X(i,1);
      solu(i,3)=Y(i,1);
      solu(i,4)=U(i,1);
      solu(i,5)=Q(i,1);
end
solu
%*******************************************************************
% Next, we start to compute for internal nodes
%*******************************************************************
% ------- step 1 : input the data to compute ----------
disp('**************************************************')
l=input(' Enter the number of internal nodes: ');
n1=l;
disp('Next ,input the coordinate(X,Y) of each int node')
disp('**************************************************')
for i=1:l
fprintf('X-coordinate of node %.0f is',i);
Xin1(i)=input(': ');
fprintf('Y-coordinate of node %.0f is',i);
Yin1(i)=input(': ');
end
Xin=Xin1';
Yin=Yin1';
% ------- step 2 : Construct matrix H and G -----------
% both H2 and G2 are constructed in the same
% way to construct Hhat and G
H2=matrixhq2(n,l,X,Y,Xin,Yin,ep,w);
G2=matrixgq2(n,l,X,Y,Xin,Yin,ep,w);
% --- step 3 : compute the solutions at each int node --
Usol=0.5*((G2)*(Q)-(H2)*(U));
MLBEM58=mlbem(n,l,n1,X,Y,Xin,Yin,ep,w);
Uvalex=[0.245695;1.745666;0.196042;0.875023];
for i=1:l
    Error(i,1)=0;
    Errorq(i,1)=abs(Uvalex(i,1)-Usol(i,1));
    Errorl(i,1)=abs(Uvalex(i,1)-MLBEM58(i,1));
end
disp('------------- The solutions are shown as follows ------------')
nod=1:1:l;
node=(nod');
disp(' node   X      Y      U-MLBEM57    U-MQBEM57  U-Exact  EL   EQ')
for i=1:l
    solu1(i,1)=node(i,1);
    solu1(i,2)=Xin(i,1);
    solu1(i,3)=Yin(i,1);
    solu1(i,4)=MLBEM58(i,1);
    solu1(i,5)=Usol(i,1);
    solu1(i,6)=Uvalex(i,1);
    solu1(i,7)=Errorl(i,1);
    solu1(i,8)=Errorq(i,1);
end
solu1
disp('***********************************************************')
%/////////////////  Graph of Normal derivative  ///////////////
D=1:1:l;
Subplot(2,1,1);
```

```
plot(D,Usol,'m-O',D,Uvalex,'b-s',D,MLBEM58,'g-*','LineWidth',2);
xlabel('Node');ylabel('Solution of u');
grid on
title('Solution of u from MQBEM58 and MLBEM58');
legend('MQBEM58','LINBEM','MLBEM58');
Subplot(2,1,2);
plot(D,Errorq,'m-O',D,Errorl,'g-*','LineWidth',2);
axis([0.9 4.1 -0.002 0.006])
xlabel('Node');ylabel('Error');
grid on
title('Error of u from MQBEM58 and MLBEM58');
legend('Error-Q','Error-L');
```

```
%*****************************************************************
%This is a program to solve the boundary problem  for Ex 5.9
%By using Quadrtic element MQBEM59
%We use 10-point Gaussian quadrature method to compute
%integration values
%*****************************************************************
% Set ep(g) and w(g) values for 10-point Gaussian quadrature
%*****************************************************************
ep=[0.973906528517172;-0.973906528517172;0.865063366688985;-
    0.865063366688985;0.679409568299024;0.679409568299024;0.433395394
    129247;-0.433395394129247;0.148874338981631;-0.148874338981631];
w=[0.066671344308688;0.066671344308688;0.149451349150581;0.1494513491
    50581;0.219086362515982;0.219086362515982;0.269266719309996;0.2692
    66719309996;0.295524224714753;0.295524224714753];
%*****************************************************************
% Receive the input data and compute coordinate matrix
%*****************************************************************
disp(' Welcome to QABEM59 program')
n=20;
X=[10.0200;15;20;25;29.98;29.993;27.716;21.213;11.48;0.628;0;0;0;0;0;
0.2090;3.8260;7.0710;9.238;9.997];
Y=[0;0;0;0;0;0.628;11.48;21.213;27.716;29.993;29.98;25;20;15;10.02;9.
997;9.238;7.071;3.826;0.209];
%*****************************************************************
Hhat=matrixhq(n,X,Y,ep,w); % This is the matrix H-hat
G=matrixgq(n,X,Y,ep,w);    % This is the matrix G
%****************************************************************
%   input known values of U and Q on boundary of square
%****************************************************************
  N=ncode(n);
  t=ncodv(Y,n);%will be used in computing B of Ax=B steps
%****************************************************************
%       Changing an system of equation to be in form
%                       Ax=B
%       and solve to obtain the unknown values of U and Q
%****************************************************************
A=cheql(N,Hhat,G,t,n);
b=cheqr(N,Hhat,G,t,n);
B=b*t;
XX=(inv(A))*B;              % XX is x in Ax=B
%****************************************************************
%            Show  the matrix U and Q
```

```
%*********************************************************************
U=matrixu(t,n,XX);
Q=matrixq(t,n,XX);
nod=1:1:n;
node=(nod');
disp('*************************************************')
disp('     node          X          Y        u        deriv.')
for i=1:n
      solu(i,1)=node(i,1);
      solu(i,2)=X(i,1);
      solu(i,3)=Y(i,1);
      solu(i,4)=U(i,1);
      solu(i,5)=Q(i,1);
end
solu
%*********************************************************************
% Next, we start to compute for internal nodes
%*********************************************************************
% ------- step 1 : input the data to compute ----------
disp('*************************************************')
l=input(' Enter the number of internal nodes: ');
nl=l;
disp('Next ,input the coordinate(X,Y) of each int node')
disp('*************************************************')
for i=1:l
fprintf('X-coordinate of node %.0f is',i);
Xin1(i)=input(': ');
fprintf('Y-coordinate of node %.0f is',i);
Yin1(i)=input(': ');
end
Xin=Xin1';
Yin=Yin1';
% ------- step 2 : Construct matrix H and G -----------
% both H2 and G2 are constructed in the same
% way to construct Hhat and G
H2=matrixhq2(n,l,X,Y,Xin,Yin,ep,w);
G2=matrixgq2(n,l,X,Y,Xin,Yin,ep,w);
% --- step 3 : compute the solutions at each int node --
Usol=0.5*((G2)*(Q)-(H2)*(U));
MLBEM59=mlbem(n,l,nl,X,Y,Xin,Yin,ep,w);
Uvalex=[621.1442;621.1442;621.1442];
for i=1:l
    Error(i,1)=0;
    Errorq(i,1)=abs(Uvalex(i,1)-Usol(i,1));
    Errorl(i,1)=abs(Uvalex(i,1)-MLBEM59(i,1));
end
disp('---------------------- The solutions are shown as follows -----
----------------')
nod=1:1:l;
node=(nod');
disp('node    X      Y    U-MLBEM510    U-MQBEM510  U-Exact  EL    EQ')
for i=1:l
    solu1(i,1)=node(i,1);
    solu1(i,2)=Xin(i,1);
    solu1(i,3)=Yin(i,1);
    solu1(i,4)=MLBEM59(i,1);
    solu1(i,5)=Usol(i,1);
    solu1(i,6)=Uvalex(i,1);
```

```
    solu1(i,7)=Errorl(i,1);
    solu1(i,8)=Errorq(i,1);
end
solu1
disp('***********************************************************')
%//////////////////  Graph of Normal derivative  ///////////////
D=1:1:1;
Subplot(2,1,1);
plot(D,Usol,'m-O',D,Uvalex,'b-s',D,MLBEM59,'g-*','LineWidth',2);
xlabel('Node');ylabel('Solution of u');
grid on
title('Solution of u from MQBEM59 and MLBEM59');
legend('MQBEM59','Exact','MLBEM59');
Subplot(2,1,2);
plot(D,Errorq,'m-O',D,Errorl,'g-*','LineWidth',2);
axis([0.9 3.1 -2 20])
xlabel('Node');ylabel('Error');
grid on
title('Error of u from MQBEM59 and MLBEM59');
legend('Error-Q','Error-L');
```

ส่วนที่ 2 โปรแกรมย่อย (Subrotines)

```
%*******************************************************************
function [X]=coorx(n)
    k=n/4;
    X=zeros(n,1);
    for i=1:k
       X(i+1)=X(i+1)+(i/k)*1;
    end
    for i=k+2:2*k+1
        X(i)=1;
    end
    for i=2:k
        X(2*k+i)=X(k-(i-2));
    end
%*******************************************************************

%*******************************************************************
function [Y]=coory(n)
    k=n/4;
    Y=zeros(n,1);
    for i=1:k
        Y(k+(i+1))=Y(k+(i+1))+(i/k)*1;
    end
    for i=2*k+2:3*k+1
        Y(i)=1;
    end
    for i=2:k
        Y(3*k+(i))=Y(2*k-(i-2));
    end
%*******************************************************************

%*******************************************************************
function [N]=ncode(n)
%-----------------------------------------------
%  N(i)=0 means at node i the value of Q is known
%  N(i)=1 means at node i the value of U is known
%-----------------------------------------------
     N=zeros(n,1);
     k=n/4;
    for i=1:k
        N(i)=0;
    end
    for i=k+1:2*k
        N(i)=1;
    end
    for i=2*k+1:3*k
        N(i)=0;
    end
    for i=3*k+1:n
        N(i)=1;
    end
%*******************************************************************
```

```
%**********************************************************************
function [t]=ncodv(Y,n)
k=n/4;
t=zeros(n,1);
for i=1:k
    t(i)=-2;
end
for i=k+1:2*k
    t(i)=2+2*Y(i);
end
for i=2*k+1:3*k
    t(i)=2;
end
for i=3*k+1:n
    t(i)=1+2*Y(i);
end
%**********************************************************************

%**********************************************************************
function [U]=matrixu(t,n,XX)
U=zeros(n,1);
k=n/4;
    for i=1:k
        U(i)=XX(i);
    end
    for i=k+1:2*k
        U(i)=t(i);
    end
    for i=2*k+1:3*k
        U(i)=XX(i);
    end
    for i=3*k+1:n
        U(i)=t(i);
    end
%**********************************************************************

%**********************************************************************
function [Q]=matrixq(t,n,XX)
Q=zeros(n,1);
k=n/4;
    for i=1:k
        Q(i)=t(i);
    end
    for i=k+1:2*k
        Q(i)=XX(i);
    end
    for i=2*k+1:3*k
        Q(i)=t(i);
    end
    for i=3*k+1:n
        Q(i)=XX(i);
    end
%**********************************************************************
```

```
%*******************************************************************
function [x]=matrixx(X,n)
x=zeros(n,1);
% Compute x middpoint of each element (see p.39 Davies )
    for i=1:n-1
        x(i)=x(i)+(X(i)+X(i+1))/2;
    end
    x(n)=x(n)+(X(n)+X(1))/2;
%*******************************************************************

%*******************************************************************
function [y]=matrixy(Y,n)
y=zeros(n,1);
% Compute y middpoint of each element (see p.39 Davies )
    for i=1:n-1
        y(i)=y(i)+(Y(i)+Y(i+1))/2;
    end
    y(n)=y(n)+(Y(n)+Y(1))/2;
%*******************************************************************

%*******************************************************************
% Construct distance matrix d(i,j)(=distance from node i to an
element j)
function [d]=dist(n,X,Y)
d=zeros(n);
for i=1:n
    for j=1:n-1
        ax=X(j+1)-X(j);
        ay=Y(j+1)-Y(j);
        if abs(ax)>=0.0001
            ta=(ay/ax);
            d(i,j)=abs(ta*X(i)-Y(i)+Y(j)-ta*X(j))/(sqrt(ta^2+1));
        else
            d(i,j)=abs(X(i)-X(j));
        end
    end
end
for i=1:n
        ax=X(1)-X(n);
        ay=Y(1)-Y(n);
        if abs(ax)>=0.0001
            ta=(ay/ax);
            d(i,n)=abs(ta*X(i)-Y(i)+Y(n)-ta*X(n))/(sqrt(ta^2+1));
        else
            d(i,n)=abs(X(i)-X(n));
        end
    end

%*******************************************************************

%*******************************************************************
% Construct distance matrix d(i,j)(=distance from node i to an
element j)
function [din]=distin(n,n1,Xin,Yin,X,Y)
din=zeros(n1,n);
for i=1:n1
```

```
    for j=1:n-1
        ax=X(j+1)-X(j);
        ay=Y(j+1)-Y(j);
        if abs(ax)>=0.0001
            ta=(ay/ax);
            din(i,j)=abs(ta*Xin(i)-Yin(i)+Y(j)-
ta*X(j))/(sqrt(ta^2+1));
        else
            din(i,j)=abs(Xin(i)-X(j));
        end
    end
end
for i=1:n1
        ax=X(1)-X(n);
        ay=Y(1)-Y(n);
        if abs(ax)>=0.0001
            ta=(ay/ax);
            din(i,n)=abs(ta*Xin(i)-Yin(i)+Y(n)-
ta*X(n))/(sqrt(ta^2+1));
        else
            din(i,n)=abs(Xin(i)-X(n));
        end
    end
%*********************************************************************

%*********************************************************************
function L=lenet(n,X,Y)
L=zeros(n,1);
for j=2:n-1
    L(j)=L(j)+sqrt((X(j+1)-X(j))^2+(Y(j+1)-Y(j))^2);
end
L(n)=sqrt((X(1)-X(n))^2+(Y(1)-Y(n))^2);
L(1)=sqrt((X(2)-X(1))^2+(Y(2)-Y(1))^2);
%*********************************************************************

%*********************************************************************
function [Usol1]=mlbem(n,l,n1,X,Y,Xin,Yin,ep,w)
%*********************************************************************
% Receive the input data and compute coordinate matrix
%*********************************************************************
x=matrixx(X,n);
y=matrixy(Y,n);
%********************************************************************
% Construct distance d(i,j) from a node i to an element j
%   and l(j) is a lenght of element j
%********************************************************************
d=dist(n,X,Y);
L=lenet(n,X,Y);
%********************************************************************
%Construct the matrix H and G for solving equation system HU=GQ
%************************************************************
Hhat=matrixh(n,X,Y,x,y,d,ep,w,L); % This is the matrix H-hat
G=matrixg(n,X,Y,x,y,d,ep,w,L);    % This is the matrix G
N=ncode(n);
t=ncodv(Y,n);%will be used in computing B of Ax=B step
%*********************************************************************
%       Changing an system of equation to be in form
%                       Ax=B
```

```
%       and solve to obtain the unknown values of U and Q
%*****************************************************************
A=cheql(N,Hhat,G,t,n);
b=cheqr(N,Hhat,G,t,n);
B=b*t;
XX=(inv(A))*B;             % XX is x in Ax=B
%***********************************************************
%            Show  the matrix U and Q
%***********************************************************
U=matrixu(t,n,XX);
Q=matrixq(t,n,XX);
%************************************************************
%     Next, we start to compute for internal nodes
%***********************************************************
%  ------- step 1 : input the data to compute ----------
din=distin(n,n1,Xin,Yin,X,Y);
H2=matrixh2(n,n1,X,Y,Xin,Yin,x,y,din,ep,w,L);
G2=matrixg2(n,n1,X,Y,Xin,Yin,x,y,din,ep,w,L);
%  ------- step 3 : compute the solutions at each int node ------
Usol1=(1/(2*pi))*((G2)*(Q)-(H2)*(U));
%********************************************************************


%********************************************************************
function [Hhat]=matrixhq(n,X,Y,ep,w)
%*************************************************************
Hhat=zeros(n);
% Next we shall construct H-hat by condidering in 3 parts
% first for R(i,j+1), second for R(i,j) and three for R(i,j-1)
% We use 10-point Gaussian quadrature method to compute
    for  g=1:10
        F1(g)=0.5*((ep(g))^2-ep(g));
        F2(g)=1-(ep(g))^2;
        F3(g)=0.5*((ep(g))^2+ep(g));
    end
%*********************** j=1 ******************************
    for i=1:n
        for j=1
% **** first consider for R(i,j+1)
            for g=1:10
                A=X(j)-2*X(j+1)+X(j+2);
                B=0.5*(X(j+2)-X(j));
                C=Y(j)-2*Y(j+1)+Y(j+2);
                D=0.5*(Y(j+2)-Y(j));
                Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                x1(g)=X(j)*F1(g)+X(j+1)*F2(g)+X(j+2)*F3(g);
                y1(g)=Y(j)*F1(g)+Y(j+1)*F2(g)+Y(j+2)*F3(g);
                R2ij1(g)=(X(i)-x1(g))^2+(Y(i)-y1(g))^2;
                dxds(g)=A*ep(g)+B;
                dyds(g)=C*ep(g)+D;
                nx(g)=(dyds(g))/Jco(g);
                ny(g)=-(dxds(g))/Jco(g);
                dudx(g)=x1(g)-X(i);       dudy(g)=y1(g)-Y(i);
                func(g)=nx(g)*dudx(g)+ny(g)*dudy(g);
                S1(g)=(func(g)*Jco(g)*w(g)*F1(g))/R2ij1(g);
            end
            ss1=sum(S1);
% **** next consider for R(i,j)
```

```
            for g=1:10
                A=X(n)-2*X(j)+X(j+1);
                B=0.5*(X(j+1)-X(n));
                C=Y(n)-2*Y(j)+Y(j+1);
                D=0.5*(Y(j+1)-Y(n));
                Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                x1(g)=X(n)*F1(g)+X(j)*F2(g)+X(j+1)*F3(g);
                y1(g)=Y(n)*F1(g)+Y(j)*F2(g)+Y(j+1)*F3(g);
                R2ij1(g)=(X(i)-x1(g))^2+(Y(i)-y1(g))^2;
                dxds(g)=A*ep(g)+B;
                dyds(g)=C*ep(g)+D;
                nx(g)=(dyds(g))/Jco(g);
                ny(g)=-(dxds(g))/Jco(g);
                dudx(g)=x1(g)-X(i);        dudy(g)=y1(g)-Y(i);
                func(g)=nx(g)*dudx(g)+ny(g)*dudy(g);
                S2(g)=(func(g)*Jco(g)*w(g)*F2(g))/R2ij1(g);
             end
             ss2=sum(S2);
  %**** next consider for R(i,j-1)
            for g=1:10
                A=X(n-1)-2*X(n)+X(j);
                B=0.5*(X(j)-X(n-1));
                C=Y(n-1)-2*Y(n)+Y(j);
                D=0.5*(Y(j)-Y(n-1));
                Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                x1(g)=X(n-1)*F1(g)+X(n)*F2(g)+X(j)*F3(g);
                y1(g)=Y(n-1)*F1(g)+Y(n)*F2(g)+Y(j)*F3(g);
                R2ij1(g)=(X(i)-x1(g))^2+(Y(i)-y1(g))^2;
                dxds(g)=A*ep(g)+B;
                dyds(g)=C*ep(g)+D;
                nx(g)=(dyds(g))/Jco(g);
                ny(g)=-(dxds(g))/Jco(g);
                dudx(g)=x1(g)-X(i);        dudy(g)=y1(g)-Y(i);
                func(g)=nx(g)*dudx(g)+ny(g)*dudy(g);
                S3(g)=(func(g)*Jco(g)*w(g)*F3(g))/R2ij1(g);
            end
            ss3=sum(S3);
%*****************************************************************
            Hhat(i,j)=-0.5*(ss1+ss2+ss3)/pi;
        end
    end
%************************  j=2  *********************************
    for i=1:n
        for j=2
% **** first consider for R(i,j+1)
            for g=1:10
                A=X(j)-2*X(j+1)+X(j+2);
                B=0.5*(X(j+2)-X(j));
                C=Y(j)-2*Y(j+1)+Y(j+2);
                D=0.5*(Y(j+2)-Y(j));
                Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                x1(g)=X(j)*F1(g)+X(j+1)*F2(g)+X(j+2)*F3(g);
                y1(g)=Y(j)*F1(g)+Y(j+1)*F2(g)+Y(j+2)*F3(g);
                R2ij1(g)=(X(i)-x1(g))^2+(Y(i)-y1(g))^2;
                dxds(g)=A*ep(g)+B;
                dyds(g)=C*ep(g)+D;
                nx(g)=(dyds(g))/Jco(g);
                ny(g)=-(dxds(g))/Jco(g);
```

```
                    dudx(g)=x1(g)-X(i);      dudy(g)=y1(g)-Y(i);
                    func(g)=nx(g)*dudx(g)+ny(g)*dudy(g);
                    S1(g)=(func(g)*Jco(g)*w(g)*F1(g))/R2ij1(g);
                end
                ss1=sum(S1);
% **** next consider for R(i,j)
                for g=1:10
                    A=X(j-1)-2*X(j)+X(j+1);
                    B=0.5*(X(j+1)-X(j-1));
                    C=Y(j-1)-2*Y(j)+Y(j+1);
                    D=0.5*(Y(j+1)-Y(j-1));
                    Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                    x1(g)=X(j-1)*F1(g)+X(j)*F2(g)+X(j+1)*F3(g);
                    y1(g)=Y(j-1)*F1(g)+Y(j)*F2(g)+Y(j+1)*F3(g);
                    R2ij1(g)=(X(i)-x1(g))^2+(Y(i)-y1(g))^2;
                    dxds(g)=A*ep(g)+B;
                    dyds(g)=C*ep(g)+D;
                    nx(g)=(dyds(g))/Jco(g);
                    ny(g)=-(dxds(g))/Jco(g);
                    dudx(g)=x1(g)-X(i);      dudy(g)=y1(g)-Y(i);
                    func(g)=nx(g)*dudx(g)+ny(g)*dudy(g);
                    S2(g)=(func(g)*Jco(g)*w(g)*F2(g))/R2ij1(g);
                end
                ss2=sum(S2);
 %**** next consider for R(i,j-1)
                for g=1:10
                    A=X(n)-2*X(j-1)+X(j);
                    B=0.5*(X(j)-X(n));
                    C=Y(n)-2*Y(j-1)+Y(j);
                    D=0.5*(Y(j)-Y(n));
                    Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                    x1(g)=X(n)*F1(g)+X(j-1)*F2(g)+X(j)*F3(g);
                    y1(g)=Y(n)*F1(g)+Y(j-1)*F2(g)+Y(j)*F3(g);
                    R2ij1(g)=(X(i)-x1(g))^2+(Y(i)-y1(g))^2;
                    dxds(g)=A*ep(g)+B;
                    dyds(g)=C*ep(g)+D;
                    nx(g)=(dyds(g))/Jco(g);
                    ny(g)=-(dxds(g))/Jco(g);
                    dudx(g)=x1(g)-X(i);      dudy(g)=y1(g)-Y(i);
                    func(g)=nx(g)*dudx(g)+ny(g)*dudy(g);
                    S3(g)=(func(g)*Jco(g)*w(g)*F3(g))/R2ij1(g);
                end
                ss3=sum(S3);
%*****************************************************************
                Hhat(i,j)=-0.5*(ss1+ss2+ss3)/pi;
            end
        end
%***************** j=3-n-2 *************************************
        for i=1:n
            for j=3:n-2
% **** first consider for R(i,j+1)
                for g=1:10
                    A=X(j)-2*X(j+1)+X(j+2);
                    B=0.5*(X(j+2)-X(j));
                    C=Y(j)-2*Y(j+1)+Y(j+2);
                    D=0.5*(Y(j+2)-Y(j));
                    Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                    x1(g)=X(j)*F1(g)+X(j+1)*F2(g)+X(j+2)*F3(g);
```

```
                  y1(g)=Y(j)*F1(g)+Y(j+1)*F2(g)+Y(j+2)*F3(g);
                  R2ij1(g)=(X(i)-x1(g))^2+(Y(i)-y1(g))^2;
                  dxds(g)=A*ep(g)+B;
                  dyds(g)=C*ep(g)+D;
                  nx(g)=(dyds(g))/Jco(g);
                  ny(g)=-(dxds(g))/Jco(g);
                  dudx(g)=x1(g)-X(i);       dudy(g)=y1(g)-Y(i);
                  func(g)=nx(g)*dudx(g)+ny(g)*dudy(g);
                  S1(g)=(func(g)*Jco(g)*w(g)*F1(g))/R2ij1(g);
              end
              ss1=sum(S1);
% **** next consider for R(i,j)
              for g=1:10
                  A=X(j-1)-2*X(j)+X(j+1);
                  B=0.5*(X(j+1)-X(j-1));
                  C=Y(j-1)-2*Y(j)+Y(j+1);
                  D=0.5*(Y(j+1)-Y(j-1));
                  Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                  x1(g)=X(j-1)*F1(g)+X(j)*F2(g)+X(j+1)*F3(g);
                  y1(g)=Y(j-1)*F1(g)+Y(j)*F2(g)+Y(j+1)*F3(g);
                  R2ij1(g)=(X(i)-x1(g))^2+(Y(i)-y1(g))^2;
                  dxds(g)=A*ep(g)+B;
                  dyds(g)=C*ep(g)+D;
                  nx(g)=(dyds(g))/Jco(g);
                  ny(g)=-(dxds(g))/Jco(g);
                  dudx(g)=x1(g)-X(i);       dudy(g)=y1(g)-Y(i);
                  func(g)=nx(g)*dudx(g)+ny(g)*dudy(g);
                  S2(g)=(func(g)*Jco(g)*w(g)*F2(g))/R2ij1(g);
               end
               ss2=sum(S2);
  %**** next consider for R(i,j-1)
              for g=1:10
                  A=X(j-2)-2*X(j-1)+X(j);
                  B=0.5*(X(j)-X(j-2));
                  C=Y(j-2)-2*Y(j-1)+Y(j);
                  D=0.5*(Y(j)-Y(j-2));
                  Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                  x1(g)=X(j-2)*F1(g)+X(j-1)*F2(g)+X(j)*F3(g);
                  y1(g)=Y(j-2)*F1(g)+Y(j-1)*F2(g)+Y(j)*F3(g);
                  R2ij1(g)=(X(i)-x1(g))^2+(Y(i)-y1(g))^2;
                  dxds(g)=A*ep(g)+B;
                  dyds(g)=C*ep(g)+D;
                  nx(g)=(dyds(g))/Jco(g);
                  ny(g)=-(dxds(g))/Jco(g);
                  dudx(g)=x1(g)-X(i);       dudy(g)=y1(g)-Y(i);
                  func(g)=nx(g)*dudx(g)+ny(g)*dudy(g);
                  S3(g)=(func(g)*Jco(g)*w(g)*F3(g))/R2ij1(g);
              end
              ss3=sum(S3);
%************************************************************
              Hhat(i,j)=-0.5*(ss1+ss2+ss3)/pi;
          end
      end
%********************** j= n-1 *************************************
      for i=1:n
% when j=2...n-2
          for j=n-1
% **** first consider for R(i,j+1)
```

```
            for g=1:10
                A=X(j)-2*X(j+1)+X(1);
                B=0.5*(X(1)-X(j));
                C=Y(j)-2*Y(j+1)+Y(1);
                D=0.5*(Y(1)-Y(j));
                Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                x1(g)=X(j)*F1(g)+X(j+1)*F2(g)+X(1)*F3(g);
                y1(g)=Y(j)*F1(g)+Y(j+1)*F2(g)+Y(1)*F3(g);
                R2ij1(g)=(X(i)-x1(g))^2+(Y(i)-y1(g))^2;
                dxds(g)=A*ep(g)+B;
                dyds(g)=C*ep(g)+D;
                nx(g)=(dyds(g))/Jco(g);
                ny(g)=-(dxds(g))/Jco(g);
                dudx(g)=x1(g)-X(i);        dudy(g)=y1(g)-Y(i);
                func(g)=nx(g)*dudx(g)+ny(g)*dudy(g);
                S1(g)=(func(g)*Jco(g)*w(g)*F1(g))/R2ij1(g);
            end
            ss1=sum(S1);
% **** next consider for R(i,j)
            for g=1:10
                A=X(j-1)-2*X(j)+X(j+1);
                B=0.5*(X(j+1)-X(j-1));
                C=Y(j-1)-2*Y(j)+Y(j+1);
                D=0.5*(Y(j+1)-Y(j-1));
                Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                x1(g)=X(j-1)*F1(g)+X(j)*F2(g)+X(j+1)*F3(g);
                y1(g)=Y(j-1)*F1(g)+Y(j)*F2(g)+Y(j+1)*F3(g);
                R2ij1(g)=(X(i)-x1(g))^2+(Y(i)-y1(g))^2;
                dxds(g)=A*ep(g)+B;
                dyds(g)=C*ep(g)+D;
                nx(g)=(dyds(g))/Jco(g);
                ny(g)=-(dxds(g))/Jco(g);
                dudx(g)=x1(g)-X(i);        dudy(g)=y1(g)-Y(i);
                func(g)=nx(g)*dudx(g)+ny(g)*dudy(g);
                S2(g)=(func(g)*Jco(g)*w(g)*F2(g))/R2ij1(g);
             end
             ss2=sum(S2);
 %**** next consider for R(i,j-1)
            for g=1:10
                A=X(j-2)-2*X(j-1)+X(j);
                B=0.5*(X(j)-X(j-2));
                C=Y(j-2)-2*Y(j-1)+Y(j);
                D=0.5*(Y(j)-Y(j-2));
                Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                x1(g)=X(j-2)*F1(g)+X(j-1)*F2(g)+X(j)*F3(g);
                y1(g)=Y(j-2)*F1(g)+Y(j-1)*F2(g)+Y(j)*F3(g);
                R2ij1(g)=(X(i)-x1(g))^2+(Y(i)-y1(g))^2;
                dxds(g)=A*ep(g)+B;
                dyds(g)=C*ep(g)+D;
                nx(g)=(dyds(g))/Jco(g);
                ny(g)=-(dxds(g))/Jco(g);
                dudx(g)=x1(g)-X(i);        dudy(g)=y1(g)-Y(i);
                func(g)=nx(g)*dudx(g)+ny(g)*dudy(g);
                S3(g)=(func(g)*Jco(g)*w(g)*F3(g))/R2ij1(g);
            end
            ss3=sum(S3);
%**********************************************************************
            Hhat(i,j)=-0.5*(ss1+ss2+ss3)/pi;
```

```
        end
    end
%****************** j= n *******************************************
    for i=1:n
% when j=2...n-2
        for j=n
% **** first consider for R(i,j+1)
            for g=1:10
                A=X(j)-2*X(1)+X(2);
                B=0.5*(X(2)-X(j));
                C=Y(j)-2*Y(1)+Y(2);
                D=0.5*(Y(2)-Y(j));
                Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                x1(g)=X(j)*F1(g)+X(1)*F2(g)+X(2)*F3(g);
                y1(g)=Y(j)*F1(g)+Y(1)*F2(g)+Y(2)*F3(g);
                R2ij1(g)=(X(i)-x1(g))^2+(Y(i)-y1(g))^2;
                dxds(g)=A*ep(g)+B;
                dyds(g)=C*ep(g)+D;
                nx(g)=(dyds(g))/Jco(g);
                ny(g)=-(dxds(g))/Jco(g);
                dudx(g)=x1(g)-X(i);        dudy(g)=y1(g)-Y(i);
                func(g)=nx(g)*dudx(g)+ny(g)*dudy(g);
                S1(g)=(func(g)*Jco(g)*w(g)*F1(g))/R2ij1(g);
            end
            ss1=sum(S1);
% **** next consider for R(i,j)
            for g=1:10
                A=X(j-1)-2*X(j)+X(1);
                B=0.5*(X(1)-X(j-1));
                C=Y(j-1)-2*Y(j)+Y(1);
                D=0.5*(Y(1)-Y(j-1));
                Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                x1(g)=X(j-1)*F1(g)+X(j)*F2(g)+X(1)*F3(g);
                y1(g)=Y(j-1)*F1(g)+Y(j)*F2(g)+Y(1)*F3(g);
                R2ij1(g)=(X(i)-x1(g))^2+(Y(i)-y1(g))^2;
                dxds(g)=A*ep(g)+B;
                dyds(g)=C*ep(g)+D;
                nx(g)=(dyds(g))/Jco(g);
                ny(g)=-(dxds(g))/Jco(g);
                dudx(g)=x1(g)-X(i);        dudy(g)=y1(g)-Y(i);
                func(g)=nx(g)*dudx(g)+ny(g)*dudy(g);
                S2(g)=(func(g)*Jco(g)*w(g)*F2(g))/R2ij1(g);
            end
            ss2=sum(S2);
  %**** next consider for R(i,j-1)
            for g=1:10
                A=X(j-2)-2*X(j-1)+X(j);
                B=0.5*(X(j)-X(j-2));
                C=Y(j-2)-2*Y(j-1)+Y(j);
                D=0.5*(Y(j)-Y(j-2));
                Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                x1(g)=X(j-2)*F1(g)+X(j-1)*F2(g)+X(j)*F3(g);
                y1(g)=Y(j-2)*F1(g)+Y(j-1)*F2(g)+Y(j)*F3(g);
                R2ij1(g)=(X(i)-x1(g))^2+(Y(i)-y1(g))^2;
                dxds(g)=A*ep(g)+B;
                dyds(g)=C*ep(g)+D;
                nx(g)=(dyds(g))/Jco(g);
                ny(g)=-(dxds(g))/Jco(g);
```

```
                dudx(g)=x1(g)-X(i);       dudy(g)=y1(g)-Y(i);
                func(g)=nx(g)*dudx(g)+ny(g)*dudy(g);
                S3(g)=(func(g)*Jco(g)*w(g)*F3(g))/R2ij1(g);
            end
            ss3=sum(S3);
%*********************************************************************
            Hhat(i,j)=-0.5*(ss1+ss2+ss3)/pi;
        end
    end
    for i=1:n
        Hhat(i,i)=0;
    end
    % Next we consider Hhat's values when i=j
    Hhat1=sum(Hhat');
    for i=1:n
        Hhat(i,i)=-Hhat1(1,i);
    end
%*********************************************************************

%*********************************************************************
function [H2]=matrixhq2(n,l,X,Y,Xin,Yin,ep,w)
H2=zeros(l,n);
% Next we shall construct H2 by condidering in 3 parts
% first for R(i,j+1), second for R(i,j) and three for R(i,j-1)
% We use 10-point Gaussian quadrature method to compute
%*********************************************************************
    for  g=1:10
        F1(g)=0.5*((ep(g))^2-ep(g));
        F2(g)=1-(ep(g))^2;
        F3(g)=0.5*((ep(g))^2+ep(g));
    end
%************************ j=1 ***********************************
    for i=1:l
% when j=2...n-2
        for j=1
% **** first consider for R(i,j+1)
            for g=1:10
                A=X(j)-2*X(j+1)+X(j+2);
                B=0.5*(X(j+2)-X(j));
                C=Y(j)-2*Y(j+1)+Y(j+2);
                D=0.5*(Y(j+2)-Y(j));
                Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                x1(g)=X(j)*F1(g)+X(j+1)*F2(g)+X(j+2)*F3(g);
                y1(g)=Y(j)*F1(g)+Y(j+1)*F2(g)+Y(j+2)*F3(g);
                R2ij1(g)=(Xin(i)-x1(g))^2+(Yin(i)-y1(g))^2;
                dxds(g)=A*ep(g)+B;
                dyds(g)=C*ep(g)+D;
                nx(g)=(dyds(g))/Jco(g);
                ny(g)=-(dxds(g))/Jco(g);
                dudx(g)=x1(g)-Xin(i);       dudy(g)=y1(g)-Yin(i);
                func(g)=nx(g)*dudx(g)+ny(g)*dudy(g);
                S1(g)=(func(g)*Jco(g)*w(g)*F1(g))/R2ij1(g);
            end
            ss1=sum(S1);
% **** next consider for R(i,j)
            for g=1:10
                A=X(n)-2*X(j)+X(j+1);
                B=0.5*(X(j+1)-X(n));
```

```
                C=Y(n)-2*Y(j)+Y(j+1);
                D=0.5*(Y(j+1)-Y(n));
                Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                x1(g)=X(n)*F1(g)+X(j)*F2(g)+X(j+1)*F3(g);
                y1(g)=Y(n)*F1(g)+Y(j)*F2(g)+Y(j+1)*F3(g);
                R2ij1(g)=(Xin(i)-x1(g))^2+(Yin(i)-y1(g))^2;
                dxds(g)=A*ep(g)+B;
                dyds(g)=C*ep(g)+D;
                nx(g)=(dyds(g))/Jco(g);
                ny(g)=-(dxds(g))/Jco(g);
                dudx(g)=x1(g)-Xin(i);       dudy(g)=y1(g)-Yin(i);
                func(g)=nx(g)*dudx(g)+ny(g)*dudy(g);
                S2(g)=(func(g)*Jco(g)*w(g)*F2(g))/R2ij1(g);
             end
             ss2=sum(S2);
  %**** next consider for R(i,j-1)
            for g=1:10
                A=X(n-1)-2*X(n)+X(j);
                B=0.5*(X(j)-X(n-1));
                C=Y(n-1)-2*Y(n)+Y(j);
                D=0.5*(Y(j)-Y(n-1));
                Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                x1(g)=X(n-1)*F1(g)+X(n)*F2(g)+X(j)*F3(g);
                y1(g)=Y(n-1)*F1(g)+Y(n)*F2(g)+Y(j)*F3(g);
                R2ij1(g)=(Xin(i)-x1(g))^2+(Yin(i)-y1(g))^2;
                dxds(g)=A*ep(g)+B;
                dyds(g)=C*ep(g)+D;
                nx(g)=(dyds(g))/Jco(g);
                ny(g)=-(dxds(g))/Jco(g);
                dudx(g)=x1(g)-Xin(i);       dudy(g)=y1(g)-Yin(i);
                func(g)=nx(g)*dudx(g)+ny(g)*dudy(g);
                S3(g)=(func(g)*Jco(g)*w(g)*F3(g))/R2ij1(g);
            end
            ss3=sum(S3);
%*****************************************************************
            H2(i,j)=-0.5*(ss1+ss2+ss3)/pi;
        end
    end
%************************* j=2 ************************************
    for i=1:l
% when j=2...n-2
        for j=2
% **** first consider for R(i,j+1)
            for g=1:10
                A=X(j)-2*X(j+1)+X(j+2);
                B=0.5*(X(j+2)-X(j));
                C=Y(j)-2*Y(j+1)+Y(j+2);
                D=0.5*(Y(j+2)-Y(j));
                Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                x1(g)=X(j)*F1(g)+X(j+1)*F2(g)+X(j+2)*F3(g);
                y1(g)=Y(j)*F1(g)+Y(j+1)*F2(g)+Y(j+2)*F3(g);
                R2ij1(g)=(Xin(i)-x1(g))^2+(Yin(i)-y1(g))^2;
                dxds(g)=A*ep(g)+B;
                dyds(g)=C*ep(g)+D;
                nx(g)=(dyds(g))/Jco(g);
                ny(g)=-(dxds(g))/Jco(g);
                dudx(g)=x1(g)-Xin(i);       dudy(g)=y1(g)-Yin(i);
                func(g)=nx(g)*dudx(g)+ny(g)*dudy(g);
```

```
                S1(g)=(func(g)*Jco(g)*w(g)*F1(g))/R2ij1(g);
            end
            ss1=sum(S1);
% **** next consider for R(i,j)
            for g=1:10
                A=X(j-1)-2*X(j)+X(j+1);
                B=0.5*(X(j+1)-X(j-1));
                C=Y(j-1)-2*Y(j)+Y(j+1);
                D=0.5*(Y(j+1)-Y(j-1));
                Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                x1(g)=X(j-1)*F1(g)+X(j)*F2(g)+X(j+1)*F3(g);
                y1(g)=Y(j-1)*F1(g)+Y(j)*F2(g)+Y(j+1)*F3(g);
                R2ij1(g)=(Xin(i)-x1(g))^2+(Yin(i)-y1(g))^2;
                dxds(g)=A*ep(g)+B;
                dyds(g)=C*ep(g)+D;
                nx(g)=(dyds(g))/Jco(g);
                ny(g)=-(dxds(g))/Jco(g);
                dudx(g)=x1(g)-Xin(i);       dudy(g)=y1(g)-Yin(i);
                func(g)=nx(g)*dudx(g)+ny(g)*dudy(g);
                S2(g)=(func(g)*Jco(g)*w(g)*F2(g))/R2ij1(g);
            end
            ss2=sum(S2);
 %**** next consider for R(i,j-1)
            for g=1:10
                A=X(n)-2*X(j-1)+X(j);
                B=0.5*(X(j)-X(n));
                C=Y(n)-2*Y(j-1)+Y(j);
                D=0.5*(Y(j)-Y(n));
                Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                x1(g)=X(n)*F1(g)+X(j-1)*F2(g)+X(j)*F3(g);
                y1(g)=Y(n)*F1(g)+Y(j-1)*F2(g)+Y(j)*F3(g);
                R2ij1(g)=(Xin(i)-x1(g))^2+(Yin(i)-y1(g))^2;
                dxds(g)=A*ep(g)+B;
                dyds(g)=C*ep(g)+D;
                nx(g)=(dyds(g))/Jco(g);
                ny(g)=-(dxds(g))/Jco(g);
                dudx(g)=x1(g)-Xin(i);       dudy(g)=y1(g)-Yin(i);
                func(g)=nx(g)*dudx(g)+ny(g)*dudy(g);
                S3(g)=(func(g)*Jco(g)*w(g)*F3(g))/R2ij1(g);
            end
            ss3=sum(S3);
%*************************************************************************
            H2(i,j)=-0.5*(ss1+ss2+ss3)/pi;
        end
    end
%******************* j=3-n-2 *************************************
    for i=1:1
% when j=2...n-2
        for j=3:n-2
% **** first consider for R(i,j+1)
            for g=1:10
                A=X(j)-2*X(j+1)+X(j+2);
                B=0.5*(X(j+2)-X(j));
                C=Y(j)-2*Y(j+1)+Y(j+2);
                D=0.5*(Y(j+2)-Y(j));
                Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                x1(g)=X(j)*F1(g)+X(j+1)*F2(g)+X(j+2)*F3(g);
                y1(g)=Y(j)*F1(g)+Y(j+1)*F2(g)+Y(j+2)*F3(g);
```

```
                    R2ij1(g)=(Xin(i)-x1(g))^2+(Yin(i)-y1(g))^2;
                    dxds(g)=A*ep(g)+B;
                    dyds(g)=C*ep(g)+D;
                    nx(g)=(dyds(g))/Jco(g);
                    ny(g)=-(dxds(g))/Jco(g);
                    dudx(g)=x1(g)-Xin(i);        dudy(g)=y1(g)-Yin(i);
                    func(g)=nx(g)*dudx(g)+ny(g)*dudy(g);
                    S1(g)=(func(g)*Jco(g)*w(g)*F1(g))/R2ij1(g);
                end
                ss1=sum(S1);
% **** next consider for R(i,j)
                for g=1:10
                    A=X(j-1)-2*X(j)+X(j+1);
                    B=0.5*(X(j+1)-X(j-1));
                    C=Y(j-1)-2*Y(j)+Y(j+1);
                    D=0.5*(Y(j+1)-Y(j-1));
                    Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                    x1(g)=X(j-1)*F1(g)+X(j)*F2(g)+X(j+1)*F3(g);
                    y1(g)=Y(j-1)*F1(g)+Y(j)*F2(g)+Y(j+1)*F3(g);
                    R2ij1(g)=(Xin(i)-x1(g))^2+(Yin(i)-y1(g))^2;
                    dxds(g)=A*ep(g)+B;
                    dyds(g)=C*ep(g)+D;
                    nx(g)=(dyds(g))/Jco(g);
                    ny(g)=-(dxds(g))/Jco(g);
                    dudx(g)=x1(g)-Xin(i);        dudy(g)=y1(g)-Yin(i);
                    func(g)=nx(g)*dudx(g)+ny(g)*dudy(g);
                    S2(g)=(func(g)*Jco(g)*w(g)*F2(g))/R2ij1(g);
                 end
                 ss2=sum(S2);
 %**** next consider for R(i,j-1)
                for g=1:10
                    A=X(j-2)-2*X(j-1)+X(j);
                    B=0.5*(X(j)-X(j-2));
                    C=Y(j-2)-2*Y(j-1)+Y(j);
                    D=0.5*(Y(j)-Y(j-2));
                    Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                    x1(g)=X(j-2)*F1(g)+X(j-1)*F2(g)+X(j)*F3(g);
                    y1(g)=Y(j-2)*F1(g)+Y(j-1)*F2(g)+Y(j)*F3(g);
                    R2ij1(g)=(Xin(i)-x1(g))^2+(Yin(i)-y1(g))^2;
                    dxds(g)=A*ep(g)+B;
                    dyds(g)=C*ep(g)+D;
                    nx(g)=(dyds(g))/Jco(g);
                    ny(g)=-(dxds(g))/Jco(g);
                    dudx(g)=x1(g)-Xin(i);        dudy(g)=y1(g)-Yin(i);
                    func(g)=nx(g)*dudx(g)+ny(g)*dudy(g);
                    S3(g)=(func(g)*Jco(g)*w(g)*F3(g))/R2ij1(g);
                end
                ss3=sum(S3);
%******************************************************************
                H2(i,j)=-0.5*(ss1+ss2+ss3)/pi;
            end
        end
%****************** j= n-1 *************************************
        for i=1:1
% when j=2...n-2
            for j=n-1
% **** first consider for R(i,j+1)
                for g=1:10
```

```
                    A=X(j)-2*X(j+1)+X(1);
                    B=0.5*(X(1)-X(j));
                    C=Y(j)-2*Y(j+1)+Y(1);
                    D=0.5*(Y(1)-Y(j));
                    Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                    x1(g)=X(j)*F1(g)+X(j+1)*F2(g)+X(1)*F3(g);
                    y1(g)=Y(j)*F1(g)+Y(j+1)*F2(g)+Y(1)*F3(g);
                    R2ij1(g)=(Xin(i)-x1(g))^2+(Yin(i)-y1(g))^2;
                    dxds(g)=A*ep(g)+B;
                    dyds(g)=C*ep(g)+D;
                    nx(g)=(dyds(g))/Jco(g);
                    ny(g)=-(dxds(g))/Jco(g);
                    dudx(g)=x1(g)-Xin(i);       dudy(g)=y1(g)-Yin(i);
                    func(g)=nx(g)*dudx(g)+ny(g)*dudy(g);
                    S1(g)=(func(g)*Jco(g)*w(g)*F1(g))/R2ij1(g);
                end
                ss1=sum(S1);
% **** next consider for R(i,j)
                for g=1:10
                    A=X(j-1)-2*X(j)+X(j+1);
                    B=0.5*(X(j+1)-X(j-1));
                    C=Y(j-1)-2*Y(j)+Y(j+1);
                    D=0.5*(Y(j+1)-Y(j-1));
                    Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                    x1(g)=X(j-1)*F1(g)+X(j)*F2(g)+X(j+1)*F3(g);
                    y1(g)=Y(j-1)*F1(g)+Y(j)*F2(g)+Y(j+1)*F3(g);
                    R2ij1(g)=(Xin(i)-x1(g))^2+(Yin(i)-y1(g))^2;
                    dxds(g)=A*ep(g)+B;
                    dyds(g)=C*ep(g)+D;
                    nx(g)=(dyds(g))/Jco(g);
                    ny(g)=-(dxds(g))/Jco(g);
                    dudx(g)=x1(g)-Xin(i);       dudy(g)=y1(g)-Yin(i);
                    func(g)=nx(g)*dudx(g)+ny(g)*dudy(g);
                    S2(g)=(func(g)*Jco(g)*w(g)*F2(g))/R2ij1(g);
                 end
                 ss2=sum(S2);
 %**** next consider for R(i,j-1)
                for g=1:10
                    A=X(j-2)-2*X(j-1)+X(j);
                    B=0.5*(X(j)-X(j-2));
                    C=Y(j-2)-2*Y(j-1)+Y(j);
                    D=0.5*(Y(j)-Y(j-2));
                    Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                    x1(g)=X(j-2)*F1(g)+X(j-1)*F2(g)+X(j)*F3(g);
                    y1(g)=Y(j-2)*F1(g)+Y(j-1)*F2(g)+Y(j)*F3(g);
                    R2ij1(g)=(Xin(i)-x1(g))^2+(Yin(i)-y1(g))^2;
                    dxds(g)=A*ep(g)+B;
                    dyds(g)=C*ep(g)+D;
                    nx(g)=(dyds(g))/Jco(g);
                    ny(g)=-(dxds(g))/Jco(g);
                    dudx(g)=x1(g)-Xin(i);       dudy(g)=y1(g)-Yin(i);
                    func(g)=nx(g)*dudx(g)+ny(g)*dudy(g);
                    S3(g)=(func(g)*Jco(g)*w(g)*F3(g))/R2ij1(g);
                end
                ss3=sum(S3);
%******************************************************************
                H2(i,j)=-0.5*(ss1+ss2+ss3)/pi;
           end
```

```
    end
%************************* j= n ***********************************
    for i=1:1
% when j=2...n-2
        for j=n
% **** first consider for R(i,j+1)
            for g=1:10
                A=X(j)-2*X(1)+X(2);
                B=0.5*(X(2)-X(j));
                C=Y(j)-2*Y(1)+Y(2);
                D=0.5*(Y(2)-Y(j));
                Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                x1(g)=X(j)*F1(g)+X(1)*F2(g)+X(2)*F3(g);
                y1(g)=Y(j)*F1(g)+Y(1)*F2(g)+Y(2)*F3(g);
                R2ij1(g)=(Xin(i)-x1(g))^2+(Yin(i)-y1(g))^2;
                dxds(g)=A*ep(g)+B;
                dyds(g)=C*ep(g)+D;
                nx(g)=(dyds(g))/Jco(g);
                ny(g)=-(dxds(g))/Jco(g);
                dudx(g)=x1(g)-Xin(i);      dudy(g)=y1(g)-Yin(i);
                func(g)=nx(g)*dudx(g)+ny(g)*dudy(g);
                S1(g)=(func(g)*Jco(g)*w(g)*F1(g))/R2ij1(g);
            end
            ss1=sum(S1);
% **** next consider for R(i,j)
            for g=1:10
                A=X(j-1)-2*X(j)+X(1);
                B=0.5*(X(1)-X(j-1));
                C=Y(j-1)-2*Y(j)+Y(1);
                D=0.5*(Y(1)-Y(j-1));
                Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                x1(g)=X(j-1)*F1(g)+X(j)*F2(g)+X(1)*F3(g);
                y1(g)=Y(j-1)*F1(g)+Y(j)*F2(g)+Y(1)*F3(g);
                R2ij1(g)=(Xin(i)-x1(g))^2+(Yin(i)-y1(g))^2;
                dxds(g)=A*ep(g)+B;
                dyds(g)=C*ep(g)+D;
                nx(g)=(dyds(g))/Jco(g);
                ny(g)=-(dxds(g))/Jco(g);
                dudx(g)=x1(g)-Xin(i);      dudy(g)=y1(g)-Yin(i);
                func(g)=nx(g)*dudx(g)+ny(g)*dudy(g);
                S2(g)=(func(g)*Jco(g)*w(g)*F2(g))/R2ij1(g);
            end
            ss2=sum(S2);
 %**** next consider for R(i,j-1)
            for g=1:10
                A=X(j-2)-2*X(j-1)+X(j);
                B=0.5*(X(j)-X(j-2));
                C=Y(j-2)-2*Y(j-1)+Y(j);
                D=0.5*(Y(j)-Y(j-2));
                Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                x1(g)=X(j-2)*F1(g)+X(j-1)*F2(g)+X(j)*F3(g);
                y1(g)=Y(j-2)*F1(g)+Y(j-1)*F2(g)+Y(j)*F3(g);
                R2ij1(g)=(Xin(i)-x1(g))^2+(Yin(i)-y1(g))^2;
                dxds(g)=A*ep(g)+B;
                dyds(g)=C*ep(g)+D;
                nx(g)=(dyds(g))/Jco(g);
                ny(g)=-(dxds(g))/Jco(g);
                dudx(g)=x1(g)-Xin(i);      dudy(g)=y1(g)-Yin(i);
```

```
                func(g)=nx(g)*dudx(g)+ny(g)*dudy(g);
                S3(g)=(func(g)*Jco(g)*w(g)*F3(g))/R2ij1(g);
            end
            ss3=sum(S3);
%*****************************************************************
            H2(i,j)=-0.5*(ss1+ss2+ss3)/pi;
        end
    end
%*****************************************************************

%*****************************************************************
function [G]=matrixgq(n,X,Y,ep,w,L)
%*****************************************************************
G=zeros(n);
% Next we shall construct G by condidering in 3 parts
% first for R(i,j+1), second for R(i,j) and three for R(i,j-1)
% We use 10-point Gaussian quadrature method to compute
%*****************************************************************
    for  g=1:10
        F1(g)=0.5*((ep(g))^2-ep(g));
        F2(g)=1-(ep(g))^2;
        F3(g)=0.5*((ep(g))^2+ep(g));
    end
%*********************** j=1 ***********************************
    for i=1:n
        for j=1
% **** first consider for R(i,j+1)
            for g=1:10
                A=X(j)-2*X(j+1)+X(j+2);
                B=0.5*(X(j+2)-X(j));
                C=Y(j)-2*Y(j+1)+Y(j+2);
                D=0.5*(Y(j+2)-Y(j));
                Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                x1(g)=X(j)*F1(g)+X(j+1)*F2(g)+X(j+2)*F3(g);
                y1(g)=Y(j)*F1(g)+Y(j+1)*F2(g)+Y(j+2)*F3(g);
                R2ij1(g)=sqrt((X(i)-x1(g))^2+(Y(i)-y1(g))^2);
                if i~=j
                    S1(g)=(-0.5*Jco(g)*w(g)*F1(g)*log(R2ij1(g)))/pi;
                    SS1=0;
                else
                    dxds=-0.5*(3*X(j)-4*X(j+1)+X(j+2));
                    dyds=-0.5*(3*Y(j)-4*Y(j+1)+Y(j+2));
                    jacob1=sqrt((dxds)^2+(dyds)^2);
                    SS1=(jacob1)/pi;
                    red(g)=0.5*(1+ep(g));
                    SS2(g)=-0.5*w(g)*log(red(g))*(jacob1)/pi;
                    P1(g)=(-0.5*Jco(g)*w(g)*F1(g)*log(R2ij1(g)))/pi;
                    S1(g)=P1(g)-SS2(g) ;
                end
            end
            ss1=sum(S1)+SS1;
% **** next consider for R(i,j)
            for g=1:10
                    A=X(n)-2*X(j)+X(j+1);
                    B=0.5*(X(j+1)-X(n));
                    C=Y(n)-2*Y(j)+Y(j+1);
                    D=0.5*(Y(j+1)-Y(n));
```

```
                    Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                    x1(g)=X(n)*F1(g)+X(j)*F2(g)+X(j+1)*F3(g);
                    y1(g)=Y(n)*F1(g)+Y(j)*F2(g)+Y(j+1)*F3(g);
                    R2ij1(g)=sqrt((X(i)-x1(g))^2+(Y(i)-y1(g))^2);
                 if i~=j
                    S2(g)=(-0.5*Jco(g)*w(g)*F2(g)*log(R2ij1(g)))/pi;
                    SS1=0;
                 else
                    dxds=0.5*(X(j+1)-X(n));
                    dyds=0.5*(Y(j+1)-Y(n));
                    jacob1=sqrt((dxds)^2+(dyds)^2);
                    SS1=(jacob1)/pi;
                    red(g)=abs(ep(g));
                    SS2(g)=-0.5*w(g)*log(red(g))*(jacob1)/pi;
                    P2(g)=-0.5*Jco(g)*w(g)*F2(g)*log(R2ij1(g))/pi;
                    S2(g)=P2(g)- SS2(g);
                 end
            end
             ss2=sum(S2)+SS1;
%**** next consider for R(i,j-1)
            for g=1:10
                    A=X(n-1)-2*X(n)+X(j);
                    B=0.5*(X(j)-X(n-1));
                    C=Y(n-1)-2*Y(n)+Y(j);
                    D=0.5*(Y(j)-Y(n-1));
                    Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                    x1(g)=X(n-1)*F1(g)+X(n)*F2(g)+X(j)*F3(g);
                    y1(g)=Y(n-1)*F1(g)+Y(n)*F2(g)+Y(j)*F3(g);
                    R2ij1(g)=sqrt((X(i)-x1(g))^2+(Y(i)-y1(g))^2);
                if i~=j
                    S3(g)=-0.5*Jco(g)*w(g)*F3(g)*log(R2ij1(g))/pi;
                    SS1=0;
                else
                    dxds=0.5*(X(n-1)-4*X(n)+3*X(j));
                    dyds=0.5*(Y(n-1)-4*Y(n)+3*Y(j));
                    jacob1=sqrt((dxds)^2+(dyds)^2);
                    SS1=(jacob1)/pi;
                    red(g)=0.5*(1-ep(g));
                    SS3(g)=-0.5*w(g)*log(red(g))*(jacob1)/pi;
                    P3(g)=-0.5*Jco(g)*w(g)*F3(g)*log(R2ij1(g))/pi;
                    S3(g)=P3(g)- SS3(g);
                end
        end
            ss3=sum(S3)+SS1;
%*******************************************************************
         G(i,j)=(ss1+ss2+ss3);
        end
    end
%************************ j=2 ************************************
     for i=1:n
        for j=2
% **** first consider for R(i,j+1)
             for g=1:10
                    A=X(j)-2*X(j+1)+X(j+2);
                    B=0.5*(X(j+2)-X(j));
                    C=Y(j)-2*Y(j+1)+Y(j+2);
                    D=0.5*(Y(j+2)-Y(j));
                    Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
```

```
                x1(g)=X(j)*F1(g)+X(j+1)*F2(g)+X(j+2)*F3(g);
                y1(g)=Y(j)*F1(g)+Y(j+1)*F2(g)+Y(j+2)*F3(g);
                R2ij1(g)=sqrt((X(i)-x1(g))^2+(Y(i)-y1(g))^2);
            if i~=j
                S1(g)=(-0.5*Jco(g)*w(g)*F1(g)*log(R2ij1(g)))/pi;
                SS1=0;
            else
                dxds=-0.5*(3*X(j)-4*X(j+1)+X(j+2));
                dyds=-0.5*(3*Y(j)-4*Y(j+1)+Y(j+2));
                jacob1=sqrt((dxds)^2+(dyds)^2);
                SS1=(jacob1)/pi;
                red(g)=0.5*(1+ep(g));
                SS2(g)=-0.5*w(g)*log(red(g))*(jacob1)/pi;
                P1(g)=(-0.5*Jco(g)*w(g)*F1(g)*log(R2ij1(g)))/pi;
                S1(g)=P1(g)-SS2(g) ;
            end
        end
        ss1=sum(S1)+SS1;
% **** next consider for R(i,j)
        for g=1:10
                A=X(j-1)-2*X(j)+X(j+1);
                B=0.5*(X(j+1)-X(j-1));
                C=Y(j-1)-2*Y(j)+Y(j+1);
                D=0.5*(Y(j+1)-Y(j-1));
                Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                x1(g)=X(j-1)*F1(g)+X(j)*F2(g)+X(j+1)*F3(g);
                y1(g)=Y(j-1)*F1(g)+Y(j)*F2(g)+Y(j+1)*F3(g);
                R2ij1(g)=sqrt((X(i)-x1(g))^2+(Y(i)-y1(g))^2);
            if i~=j
                S2(g)=(-0.5*Jco(g)*w(g)*F2(g)*log(R2ij1(g)))/pi;
                SS1=0;
            else
                dxds=0.5*(X(j+1)-X(j-1));
                dyds=0.5*(Y(j+1)-Y(j-1));
                jacob1=sqrt((dxds)^2+(dyds)^2);
                SS1=(jacob1)/pi;
                red(g)=abs(ep(g));
                SS2(g)=-0.5*w(g)*log(red(g))*(jacob1)/pi;
                P2(g)=-0.5*Jco(g)*w(g)*F2(g)*log(R2ij1(g))/pi;
                S2(g)=P2(g)- SS2(g);
            end
        end
         ss2=sum(S2)+SS1;
 %**** next consider for R(i,j-1)
        for g=1:10
                A=X(n)-2*X(j-1)+X(j);
                B=0.5*(X(j)-X(n));
                C=Y(n)-2*Y(j-1)+Y(j);
                D=0.5*(Y(j)-Y(n));
                Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                x1(g)=X(n)*F1(g)+X(j-1)*F2(g)+X(j)*F3(g);
                y1(g)=Y(n)*F1(g)+Y(j-1)*F2(g)+Y(j)*F3(g);
                R2ij1(g)=sqrt((X(i)-x1(g))^2+(Y(i)-y1(g))^2);
            if i~=j
                S3(g)=-0.5*Jco(g)*w(g)*F3(g)*log(R2ij1(g))/pi;
                SS1=0;
            else
                dxds=0.5*(X(n)-4*X(j-1)+3*X(j));
```

```
                    dyds=0.5*(Y(n)-4*Y(j-1)+3*Y(j));
                    jacob1=sqrt((dxds)^2+(dyds)^2);
                    SS1=(jacob1)/pi;
                    red(g)=0.5*(1-ep(g));
                    SS3(g)=-0.5*w(g)*log(red(g))*(jacob1)/pi;
                    P3(g)=-0.5*Jco(g)*w(g)*F3(g)*log(R2ij1(g))/pi;
                    S3(g)=P3(g)- SS3(g);
                end
        end
             ss3=sum(S3)+SS1;
%**********************************************************************
         G(i,j)=(ss1+ss2+ss3);
        end
     end
%****************** j=2-(n-2) ***********************************
    for i=1:n
        for j=3:n-2
% **** first consider for R(i,j+1)
            for g=1:10
                A=X(j)-2*X(j+1)+X(j+2);
                B=0.5*(X(j+2)-X(j));
                C=Y(j)-2*Y(j+1)+Y(j+2);
                D=0.5*(Y(j+2)-Y(j));
                Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                x1(g)=X(j)*F1(g)+X(j+1)*F2(g)+X(j+2)*F3(g);
                y1(g)=Y(j)*F1(g)+Y(j+1)*F2(g)+Y(j+2)*F3(g);
                R2ij1(g)=sqrt((X(i)-x1(g))^2+(Y(i)-y1(g))^2);
                if i~=j
                    S1(g)=(-0.5*Jco(g)*w(g)*F1(g)*log(R2ij1(g)))/pi;
                    SS1=0;
                else
                    dxds=-0.5*(3*X(j)-4*X(j+1)+X(j+2));
                    dyds=-0.5*(3*Y(j)-4*Y(j+1)+Y(j+2));
                    jacob1=sqrt((dxds)^2+(dyds)^2);
                    SS1=(jacob1)/pi;
                    red(g)=0.5*(1+ep(g));
                    SS2(g)=-0.5*w(g)*log(red(g))*(jacob1)/pi;
                    P1(g)=(-0.5*Jco(g)*w(g)*F1(g)*log(R2ij1(g)))/pi;
                    S1(g)=P1(g)-SS2(g) ;
                end
            end
            ss1=sum(S1)+SS1;
% **** next consider for R(i,j)
            for g=1:10
                A=X(j-1)-2*X(j)+X(j+1);
                B=0.5*(X(j+1)-X(j-1));
                C=Y(j-1)-2*Y(j)+Y(j+1);
                D=0.5*(Y(j+1)-Y(j-1));
                Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                x1(g)=X(j-1)*F1(g)+X(j)*F2(g)+X(j+1)*F3(g);
                y1(g)=Y(j-1)*F1(g)+Y(j)*F2(g)+Y(j+1)*F3(g);
                R2ij1(g)=sqrt((X(i)-x1(g))^2+(Y(i)-y1(g))^2);
                if i~=j
                    S2(g)=(-0.5*Jco(g)*w(g)*F2(g)*log(R2ij1(g)))/pi;
                    SS1=0;
                 else
                    dxds=0.5*(X(j+1)-X(j-1));
                    dyds=0.5*(Y(j+1)-Y(j-1));
```

```
                    jacob1=sqrt((dxds)^2+(dyds)^2);
                    SS1=(jacob1)/pi;
                    red(g)=abs(ep(g));
                    SS2(g)=-0.5*w(g)*log(red(g))*(jacob1)/pi;
                    P2(g)=-0.5*Jco(g)*w(g)*F2(g)*log(R2ij1(g))/pi;
                    S2(g)=P2(g)- SS2(g);
                 end
              end
              ss2=sum(S2)+SS1;
%**** next consider for R(i,j-1)
              for g=1:10
                  A=X(j-2)-2*X(j-1)+X(j);
                  B=0.5*(X(j)-X(j-2));
                  C=Y(j-2)-2*Y(j-1)+Y(j);
                  D=0.5*(Y(j)-Y(j-2));
                  Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                  x1(g)=X(j-2)*F1(g)+X(j-1)*F2(g)+X(j)*F3(g);
                  y1(g)=Y(j-2)*F1(g)+Y(j-1)*F2(g)+Y(j)*F3(g);
                  R2ij1(g)=sqrt((X(i)-x1(g))^2+(Y(i)-y1(g))^2);
                  if i~=j
                      S3(g)=-0.5*Jco(g)*w(g)*F3(g)*log(R2ij1(g))/pi;
                      SS1=0;
                   else
                      dxds=0.5*(X(j-2)-4*X(j-1)+3*X(j));
                      dyds=0.5*(Y(j-2)-4*Y(j-1)+3*Y(j));
                      jacob1=sqrt((dxds)^2+(dyds)^2);
                      SS1=(jacob1)/pi;
                      red(g)=0.5*(1-ep(g));
                      SS3(g)=-0.5*w(g)*log(red(g))*(jacob1)/pi;
                      P3(g)=-0.5*Jco(g)*w(g)*F3(g)*log(R2ij1(g))/pi;
                      S3(g)=P3(g)- SS3(g);
                  end
              end
              ss3=sum(S3)+SS1;
%*****************************************************************
          G(i,j)=(ss1+ss2+ss3);
        end
    end
%************************ j=n-1 ******************************
    for i=1:n
        for j=n-1
% **** first consider for R(i,j+1)
              for g=1:10
                  A=X(j)-2*X(j+1)+X(1);
                  B=0.5*(X(1)-X(j));
                  C=Y(j)-2*Y(j+1)+Y(1);
                  D=0.5*(Y(1)-Y(j));
                  Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                  x1(g)=X(j)*F1(g)+X(j+1)*F2(g)+X(1)*F3(g);
                  y1(g)=Y(j)*F1(g)+Y(j+1)*F2(g)+Y(1)*F3(g);
                  R2ij1(g)=sqrt((X(i)-x1(g))^2+(Y(i)-y1(g))^2);
                  if i~=j
                      S1(g)=(-0.5*Jco(g)*w(g)*F1(g)*log(R2ij1(g)))/pi;
                      SS1=0;
                  else
                      dxds=-0.5*(3*X(j)-4*X(j+1)+X(1));
                      dyds=-0.5*(3*Y(j)-4*Y(j+1)+Y(1));
                      jacob1=sqrt((dxds)^2+(dyds)^2);
```

```
                    SS1=(jacob1)/pi;
                    red(g)=0.5*(1+ep(g));
                    SS2(g)=-0.5*w(g)*log(red(g))*(jacob1)/pi;
                    P1(g)=(-0.5*Jco(g)*w(g)*F1(g)*log(R2ij1(g)))/pi;
                    S1(g)=P1(g)-SS2(g) ;
                end
            end
            ss1=sum(S1)+SS1;
% **** next consider for R(i,j)
            for g=1:10
                    A=X(j-1)-2*X(j)+X(j+1);
                    B=0.5*(X(j+1)-X(j-1));
                    C=Y(j-1)-2*Y(j)+Y(j+1);
                    D=0.5*(Y(j+1)-Y(j-1));
                    Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                    x1(g)=X(j-1)*F1(g)+X(j)*F2(g)+X(j+1)*F3(g);
                    y1(g)=Y(j-1)*F1(g)+Y(j)*F2(g)+Y(j+1)*F3(g);
                    R2ij1(g)=sqrt((X(i)-x1(g))^2+(Y(i)-y1(g))^2);
                 if i~=j
                    S2(g)=(-0.5*Jco(g)*w(g)*F2(g)*log(R2ij1(g)))/pi;
                    SS1=0;
                 else
                    dxds=0.5*(X(j+1)-X(j-1));
                    dyds=0.5*(Y(j+1)-Y(j-1));
                    jacob1=sqrt((dxds)^2+(dyds)^2);
                    SS1=(jacob1)/pi;
                    red(g)=abs(ep(g));
                    SS2(g)=-0.5*w(g)*log(red(g))*(jacob1)/pi;
                    P2(g)=-0.5*Jco(g)*w(g)*F2(g)*log(R2ij1(g))/pi;
                    S2(g)=P2(g)- SS2(g);
                 end
            end
             ss2=sum(S2)+SS1;
 %**** next consider for R(i,j-1)
            for g=1:10
                    A=X(j-2)-2*X(j-1)+X(j);
                    B=0.5*(X(j)-X(j-2));
                    C=Y(j-2)-2*Y(j-1)+Y(j);
                    D=0.5*(Y(j)-Y(j-2));
                    Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                    x1(g)=X(j-2)*F1(g)+X(j-1)*F2(g)+X(j)*F3(g);
                    y1(g)=Y(j-2)*F1(g)+Y(j-1)*F2(g)+Y(j)*F3(g);
                    R2ij1(g)=sqrt((X(i)-x1(g))^2+(Y(i)-y1(g))^2);
                if i~=j
                    S3(g)=-0.5*Jco(g)*w(g)*F3(g)*log(R2ij1(g))/pi;
                    SS1=0;
                else
                    dxds=0.5*(X(j-2)-4*X(j-1)+3*X(j));
                    dyds=0.5*(Y(j-2)-4*Y(j-1)+3*Y(j));
                    jacob1=sqrt((dxds)^2+(dyds)^2);
                    SS1=(jacob1)/pi;
                    red(g)=0.5*(1-ep(g));
                    SS3(g)=-0.5*w(g)*log(red(g))*(jacob1)/pi;
                    P3(g)=-0.5*Jco(g)*w(g)*F3(g)*log(R2ij1(g))/pi;
                    S3(g)=P3(g)- SS3(g);
                end
        end
            ss3=sum(S3)+SS1;
```

```
%*****************************************************************
         G(i,j)=(ss1+ss2+ss3);
        end
    end
%********************** j=n ***************************************
    for i=1:n
        for j=n
% **** first consider for R(i,j+1)
            for g=1:10
                    A=X(j)-2*X(1)+X(2);
                    B=0.5*(X(2)-X(j));
                    C=Y(j)-2*Y(1)+Y(2);
                    D=0.5*(Y(2)-Y(j));
                    Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                    x1(g)=X(j)*F1(g)+X(1)*F2(g)+X(2)*F3(g);
                    y1(g)=Y(j)*F1(g)+Y(1)*F2(g)+Y(2)*F3(g);
                    R2ij1(g)=sqrt((X(i)-x1(g))^2+(Y(i)-y1(g))^2);
                if i~=j
                    S1(g)=(-0.5*Jco(g)*w(g)*F1(g)*log(R2ij1(g)))/pi;
                    SS1=0;
                else
                    dxds=-0.5*(3*X(j)-4*X(1)+X(2));
                    dyds=-0.5*(3*Y(j)-4*Y(1)+Y(2));
                    jacob1=sqrt((dxds)^2+(dyds)^2);
                    SS1=(jacob1)/pi;
                    red(g)=0.5*(1+ep(g));
                    SS2(g)=-0.5*w(g)*log(red(g))*(jacob1)/pi;
                    P1(g)=(-0.5*Jco(g)*w(g)*F1(g)*log(R2ij1(g)))/pi;
                    S1(g)=P1(g)-SS2(g) ;
                end
            end
            ss1=sum(S1)+SS1;
% **** next consider for R(i,j)
            for g=1:10
                    A=X(j-1)-2*X(j)+X(1);
                    B=0.5*(X(1)-X(j-1));
                    C=Y(j-1)-2*Y(j)+Y(1);
                    D=0.5*(Y(1)-Y(j-1));
                    Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                    x1(g)=X(j-1)*F1(g)+X(j)*F2(g)+X(1)*F3(g);
                    y1(g)=Y(j-1)*F1(g)+Y(j)*F2(g)+Y(1)*F3(g);
                    R2ij1(g)=sqrt((X(i)-x1(g))^2+(Y(i)-y1(g))^2);
                if i~=j
                    S2(g)=(-0.5*Jco(g)*w(g)*F2(g)*log(R2ij1(g)))/pi;
                    SS1=0;
                else
                    dxds=0.5*(X(1)-X(j-1));
                    dyds=0.5*(Y(1)-Y(j-1));
                    jacob1=sqrt((dxds)^2+(dyds)^2);
                    SS1=(jacob1)/pi;
                    red(g)=abs(ep(g));
                    SS2(g)=-0.5*w(g)*log(red(g))*(jacob1)/pi;
                    P2(g)=-0.5*Jco(g)*w(g)*F2(g)*log(R2ij1(g))/pi;
                    S2(g)=P2(g)- SS2(g);
                end
            end
             ss2=sum(S2)+SS1;
 %**** next consider for R(i,j-1)
```

```
          for g=1:10
               A=X(j-2)-2*X(j-1)+X(j);
               B=0.5*(X(j)-X(j-2));
               C=Y(j-2)-2*Y(j-1)+Y(j);
               D=0.5*(Y(j)-Y(j-2));
               Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
               x1(g)=X(j-2)*F1(g)+X(j-1)*F2(g)+X(j)*F3(g);
               y1(g)=Y(j-2)*F1(g)+Y(j-1)*F2(g)+Y(j)*F3(g);
               R2ij1(g)=sqrt((X(i)-x1(g))^2+(Y(i)-y1(g))^2);
            if i~=j
               S3(g)=-0.5*Jco(g)*w(g)*F3(g)*log(R2ij1(g))/pi;
               SS1=0;
            else
               dxds=0.5*(X(j-2)-4*X(j-1)+3*X(j));
               dyds=0.5*(Y(j-2)-4*Y(j-1)+3*Y(j));
               jacob1=sqrt((dxds)^2+(dyds)^2);
               SS1=(jacob1)/pi;
               red(g)=0.5*(1-ep(g));
               SS3(g)=-0.5*w(g)*log(red(g))*(jacob1)/pi;
               P3(g)=-0.5*Jco(g)*w(g)*F3(g)*log(R2ij1(g))/pi;
               S3(g)=P3(g)- SS3(g);
            end
        end
           ss3=sum(S3)+SS1;
%*********************************************************************
        G(i,j)=(ss1+ss2+ss3);
        end
    end


%*********************************************************************
function [G2]=matrixgq2(n,l,X,Y,Xin,Yin,ep,w)
%*********************************************************************
G2=zeros(l,n);
% Next we shall construct G2 by condidering in 3 parts
% first for R(i,j+1), second for R(i,j) and three for R(i,j-1)
% We use 10-point Gaussian quadrature method to compute
%*********************************************************************
    for  g=1:10
        F1(g)=0.5*((ep(g))^2-ep(g));
        F2(g)=1-(ep(g))^2;
        F3(g)=0.5*((ep(g))^2+ep(g));
    end
%*********************** j=1 *************************************
    for i=1:l
% when j=2...n-2
        for j=1
% **** first consider for R(i,j+1)
            for g=1:10
                A=X(j)-2*X(j+1)+X(j+2);
                B=0.5*(X(j+2)-X(j));
                C=Y(j)-2*Y(j+1)+Y(j+2);
                D=0.5*(Y(j+2)-Y(j));
                Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                x1(g)=X(j)*F1(g)+X(j+1)*F2(g)+X(j+2)*F3(g);
                y1(g)=Y(j)*F1(g)+Y(j+1)*F2(g)+Y(j+2)*F3(g);
                R2ij1(g)=sqrt((Xin(i)-x1(g))^2+(Yin(i)-y1(g))^2);
                S1(g)=Jco(g)*w(g)*F1(g)*log(R2ij1(g));
```

```
            end
            ss1=sum(S1);
% **** next consider for R(i,j)
            for g=1:10
                A=X(n)-2*X(j)+X(j+1);
                B=0.5*(X(j+1)-X(n));
                C=Y(n)-2*Y(j)+Y(j+1);
                D=0.5*(Y(j+1)-Y(n));
                Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                x1(g)=X(n)*F1(g)+X(j)*F2(g)+X(j+1)*F3(g);
                y1(g)=Y(n)*F1(g)+Y(j)*F2(g)+Y(j+1)*F3(g);
                R2ij1(g)=sqrt((Xin(i)-x1(g))^2+(Yin(i)-y1(g))^2);
                S2(g)=Jco(g)*w(g)*F2(g)*log(R2ij1(g));
             end
             ss2=sum(S2);
  %**** next consider for R(i,j-1)
            for g=1:10
                A=X(n-1)-2*X(n)+X(j);
                B=0.5*(X(j)-X(n-1));
                C=Y(n-1)-2*Y(n)+Y(j);
                D=0.5*(Y(j)-Y(n-1));
                Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                x1(g)=X(n-1)*F1(g)+X(n)*F2(g)+X(j)*F3(g);
                y1(g)=Y(n-1)*F1(g)+Y(n)*F2(g)+Y(j)*F3(g);
                R2ij1(g)=sqrt((Xin(i)-x1(g))^2+(Yin(i)-y1(g))^2);
                S3(g)=Jco(g)*w(g)*F3(g)*log(R2ij1(g));
            end
            ss3=sum(S3);
%*********************************************************************
            G2(i,j)=-0.5*(ss1+ss2+ss3)/pi;
        end
    end
%************************* j=2 ***********************************
    for i=1:1
% when j=2...n-2
        for j=2
% **** first consider for R(i,j+1)
            for g=1:10
                A=X(j)-2*X(j+1)+X(j+2);
                B=0.5*(X(j+2)-X(j));
                C=Y(j)-2*Y(j+1)+Y(j+2);
                D=0.5*(Y(j+2)-Y(j));
                Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                x1(g)=X(j)*F1(g)+X(j+1)*F2(g)+X(j+2)*F3(g);
                y1(g)=Y(j)*F1(g)+Y(j+1)*F2(g)+Y(j+2)*F3(g);
                R2ij1(g)=sqrt((Xin(i)-x1(g))^2+(Yin(i)-y1(g))^2);
                S1(g)=Jco(g)*w(g)*F1(g)*log(R2ij1(g));
            end
            ss1=sum(S1);
% **** next consider for R(i,j)
            for g=1:10
                A=X(j-1)-2*X(j)+X(j+1);
                B=0.5*(X(j+1)-X(j-1));
                C=Y(j-1)-2*Y(j)+Y(j+1);
                D=0.5*(Y(j+1)-Y(j-1));
                Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                x1(g)=X(j-1)*F1(g)+X(j)*F2(g)+X(j+1)*F3(g);
                y1(g)=Y(j-1)*F1(g)+Y(j)*F2(g)+Y(j+1)*F3(g);
```

```
                R2ij1(g)=sqrt((Xin(i)-x1(g))^2+(Yin(i)-y1(g))^2);
                S2(g)=Jco(g)*w(g)*F2(g)*log(R2ij1(g));
             end
             ss2=sum(S2);
  %**** next consider for R(i,j-1)
            for g=1:10
                A=X(n)-2*X(j-1)+X(j);
                B=0.5*(X(j)-X(n));
                C=Y(n)-2*Y(j-1)+Y(j);
                D=0.5*(Y(j)-Y(n));
                Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                x1(g)=X(n)*F1(g)+X(j-1)*F2(g)+X(j)*F3(g);
                y1(g)=Y(n)*F1(g)+Y(j-1)*F2(g)+Y(j)*F3(g);
                R2ij1(g)=sqrt((Xin(i)-x1(g))^2+(Yin(i)-y1(g))^2);
                S3(g)=Jco(g)*w(g)*F3(g)*log(R2ij1(g));
            end
            ss3=sum(S3);
%********************************************************************
            G2(i,j)=-0.5*(ss1+ss2+ss3)/pi;
        end
    end
%******************** j=3-n-2 **************************************
    for i=1:1
% when j=2...n-2
        for j=3:n-2
% **** first consider for R(i,j+1)
            for g=1:10
                A=X(j)-2*X(j+1)+X(j+2);
                B=0.5*(X(j+2)-X(j));
                C=Y(j)-2*Y(j+1)+Y(j+2);
                D=0.5*(Y(j+2)-Y(j));
                Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                x1(g)=X(j)*F1(g)+X(j+1)*F2(g)+X(j+2)*F3(g);
                y1(g)=Y(j)*F1(g)+Y(j+1)*F2(g)+Y(j+2)*F3(g);
                R2ij1(g)=sqrt((Xin(i)-x1(g))^2+(Yin(i)-y1(g))^2);
                S1(g)=Jco(g)*w(g)*F1(g)*log(R2ij1(g));
            end
            ss1=sum(S1);
% **** next consider for R(i,j)
            for g=1:10
                A=X(j-1)-2*X(j)+X(j+1);
                B=0.5*(X(j+1)-X(j-1));
                C=Y(j-1)-2*Y(j)+Y(j+1);
                D=0.5*(Y(j+1)-Y(j-1));
                Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                x1(g)=X(j-1)*F1(g)+X(j)*F2(g)+X(j+1)*F3(g);
                y1(g)=Y(j-1)*F1(g)+Y(j)*F2(g)+Y(j+1)*F3(g);
                R2ij1(g)=sqrt((Xin(i)-x1(g))^2+(Yin(i)-y1(g))^2);
                S2(g)=Jco(g)*w(g)*F2(g)*log(R2ij1(g));
             end
             ss2=sum(S2);
  %**** next consider for R(i,j-1)
            for g=1:10
                A=X(j-2)-2*X(j-1)+X(j);
                B=0.5*(X(j)-X(j-2));
                C=Y(j-2)-2*Y(j-1)+Y(j);
                D=0.5*(Y(j)-Y(j-2));
                Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
```

```
                x1(g)=X(j-2)*F1(g)+X(j-1)*F2(g)+X(j)*F3(g);
                y1(g)=Y(j-2)*F1(g)+Y(j-1)*F2(g)+Y(j)*F3(g);
                R2ij1(g)=sqrt((Xin(i)-x1(g))^2+(Yin(i)-y1(g))^2);
                S3(g)=Jco(g)*w(g)*F3(g)*log(R2ij1(g));
            end
            ss3=sum(S3);
%****************************************************************
            G2(i,j)=-0.5*(ss1+ss2+ss3)/pi;
        end
    end
%******************** j= n-1 ***********************************
    for i=1:1
% when j=2...n-2
        for j=n-1
% **** first consider for R(i,j+1)
            for g=1:10
                A=X(j)-2*X(j+1)+X(1);
                B=0.5*(X(1)-X(j));
                C=Y(j)-2*Y(j+1)+Y(1);
                D=0.5*(Y(1)-Y(j));
                Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                x1(g)=X(j)*F1(g)+X(j+1)*F2(g)+X(1)*F3(g);
                y1(g)=Y(j)*F1(g)+Y(j+1)*F2(g)+Y(1)*F3(g);
                R2ij1(g)=sqrt((Xin(i)-x1(g))^2+(Yin(i)-y1(g))^2);
                S1(g)=Jco(g)*w(g)*F1(g)*log(R2ij1(g));
            end
            ss1=sum(S1);
% **** next consider for R(i,j)
            for g=1:10
                A=X(j-1)-2*X(j)+X(j+1);
                B=0.5*(X(j+1)-X(j-1));
                C=Y(j-1)-2*Y(j)+Y(j+1);
                D=0.5*(Y(j+1)-Y(j-1));
                Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                x1(g)=X(j-1)*F1(g)+X(j)*F2(g)+X(j+1)*F3(g);
                y1(g)=Y(j-1)*F1(g)+Y(j)*F2(g)+Y(j+1)*F3(g);
                R2ij1(g)=sqrt((Xin(i)-x1(g))^2+(Yin(i)-y1(g))^2);
                S2(g)=Jco(g)*w(g)*F2(g)*log(R2ij1(g));
             end
             ss2=sum(S2);
  %**** next consider for R(i,j-1)
            for g=1:10
                A=X(j-2)-2*X(j-1)+X(j);
                B=0.5*(X(j)-X(j-2));
                C=Y(j-2)-2*Y(j-1)+Y(j);
                D=0.5*(Y(j)-Y(j-2));
                Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                x1(g)=X(j-2)*F1(g)+X(j-1)*F2(g)+X(j)*F3(g);
                y1(g)=Y(j-2)*F1(g)+Y(j-1)*F2(g)+Y(j)*F3(g);
                R2ij1(g)=sqrt((Xin(i)-x1(g))^2+(Yin(i)-y1(g))^2);
                S3(g)=Jco(g)*w(g)*F3(g)*log(R2ij1(g));
            end
            ss3=sum(S3);
%*******************************************************************
            G2(i,j)=-0.5*(ss1+ss2+ss3)/pi;
        end
    end
%*********************** j= n ***********************************
```

```
    for i=1:1
% when j=2...n-2
        for j=n
% **** first consider for R(i,j+1)
            for g=1:10
                A=X(j)-2*X(1)+X(2);
                B=0.5*(X(2)-X(j));
                C=Y(j)-2*Y(1)+Y(2);
                D=0.5*(Y(2)-Y(j));
                Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                x1(g)=X(j)*F1(g)+X(1)*F2(g)+X(2)*F3(g);
                y1(g)=Y(j)*F1(g)+Y(1)*F2(g)+Y(2)*F3(g);
                R2ij1(g)=sqrt((Xin(i)-x1(g))^2+(Yin(i)-y1(g))^2);
                S1(g)=Jco(g)*w(g)*F1(g)*log(R2ij1(g));
            end
            ss1=sum(S1);
% **** next consider for R(i,j)
            for g=1:10
                A=X(j-1)-2*X(j)+X(1);
                B=0.5*(X(1)-X(j-1));
                C=Y(j-1)-2*Y(j)+Y(1);
                D=0.5*(Y(1)-Y(j-1));
                Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                x1(g)=X(j-1)*F1(g)+X(j)*F2(g)+X(1)*F3(g);
                y1(g)=Y(j-1)*F1(g)+Y(j)*F2(g)+Y(1)*F3(g);
                R2ij1(g)=sqrt((Xin(i)-x1(g))^2+(Yin(i)-y1(g))^2);
                S2(g)=Jco(g)*w(g)*F2(g)*log(R2ij1(g));
             end
             ss2=sum(S2);
  %**** next consider for R(i,j-1)
            for g=1:10
                A=X(j-2)-2*X(j-1)+X(j);
                B=0.5*(X(j)-X(j-2));
                C=Y(j-2)-2*Y(j-1)+Y(j);
                D=0.5*(Y(j)-Y(j-2));
                Jco(g)=sqrt((A*ep(g)+B)^2+(C*ep(g)+D)^2);
                x1(g)=X(j-2)*F1(g)+X(j-1)*F2(g)+X(j)*F3(g);
                y1(g)=Y(j-2)*F1(g)+Y(j-1)*F2(g)+Y(j)*F3(g);
                R2ij1(g)=sqrt((Xin(i)-x1(g))^2+(Yin(i)-y1(g))^2);
                S3(g)=Jco(g)*w(g)*F3(g)*log(R2ij1(g));
            end
            ss3=sum(S3);
%*******************************************************************
            G2(i,j)=-0.5*(ss1+ss2+ss3)/pi;
        end
    end
%*********************************************************************
```

```
%*********************************************************************
function [Hhat]=matrixh(n,X,Y,x,y,d,ep,w,L)
Hhat=zeros(n);
% Next we shall construct H-hat by condidering in 2 parts
% first for R(i,j) and second, for R(i,j-1)
% ***** For the element 1 to n-1 we obtain *******
for i=1:n
    % when j=2...n-1
    for j=2:n-1
                % first consider for R(i,j)
                for g=1:10
                x1(g)=x(j)+(1/2)*((X(j+1)-X(j))*ep(g));
                y1(g)=y(j)+(1/2)*((Y(j+1)-Y(j))*ep(g));
                R2ij1(g)=(X(i)-x1(g))^2+(Y(i)-y1(g))^2;
                S1(g)=w(g)*((1-ep(g))/R2ij1(g));
                end
                s1=sum(S1);
                ss1=(-L(j)*d(i,j)/(4))*(s1);
                % **** next consider for R(i,j-1)
                for g=1:10
                x3(g)=x(j-1)+(1/2)*((X(j)-X(j-1))*ep(g));
                y3(g)=y(j-1)+(1/2)*((Y(j)-Y(j-1))*ep(g));
                R2ij3(g)=(X(i)-x3(g))^2+(Y(i)-y3(g))^2;
                S3(g)=w(g)*((1+ep(g))/R2ij3(g));
                 end
                s3=sum(S3);
                %-----
                ss2=(-L(j)*d(i,j-1)/(4))*(s3);
                %*********
                 Hhat(i,j)=ss1+ss2;
        end
    end
 % ********* For an element j=1 ***********
 % first consider for R(i,j)
       for i=1:n
                for g=1:10
                x1(g)=x(1)+(1/2)*((X(2)-X(1))*ep(g));
                y1(g)=y(1)+(1/2)*((Y(2)-Y(1))*ep(g));
                R2ij1(g)=(X(i)-x1(g))^2+(Y(i)-y1(g))^2;
                S1(g)=w(g)*((1-ep(g))/R2ij1(g));
                end
                s1=sum(S1);
                ss1=(-L(1)*d(i,1)/(4))*(s1);
                % **** next consider for R(i,j-1)
                for g=1:10
                x3(g)=x(n)+(1/2)*((X(1)-X(n))*ep(g));
                y3(g)=y(n)+(1/2)*((Y(1)-Y(n))*ep(g));
                R2ij3(g)=(X(i)-x3(g))^2+(Y(i)-y3(g))^2;
                S3(g)=w(g)*((1+ep(g))/R2ij3(g));
                 end
                s3=sum(S3);
                %----
                ss2=(-L(1)*d(i,n)/(4))*(s3);
                %*********
                Hhat(i,1)=ss1+ss2;
       end
%******* For an element j=n  ****************
   for i=1:n
```

```
 % first for R(i,j)
            for g=1:10
             x1(g)=x(n)+(1/2)*((X(1)-X(n))*ep(g));
             y1(g)=y(n)+(1/2)*((Y(1)-Y(n))*ep(g));
             R2ij1(g)=(X(i)-x1(g))^2+(Y(i)-y1(g))^2;
             S1(g)=w(g)*((1-ep(g))/R2ij1(g));
            end
           s1=sum(S1);
           %----
             ss1=(-L(n)*d(i,n)/(4))*(s1);
            % Next for R(i,j)
            for g=1:10
             x3(g)=x(n-1)+(1/2)*((X(n)-X(n-1))*ep(g));
             y3(g)=y(n-1)+(1/2)*((Y(n)-Y(n-1))*ep(g));
             R2ij3(g)=(X(i)-x3(g))^2+(Y(i)-y3(g))^2;
             S3(g)=w(g)*((1+ep(g))/R2ij3(g));
            end
           s3=sum(S3);
           %-----
             ss2=(-L(n)*d(i,n-1)/(4))*(s3);
             %-------
            Hhat(i,n)=ss1+ss2;
end
% Next we consider Hhat's values when i=j
Hhat1=sum(Hhat');
for i=1:n
   Hhat(i,i)=-Hhat1(1,i);
end
%**********************************************************************


%**********************************************************************
function [H2]=matrixh2(n,n1,X,Y,Xin,Yin,x,y,din,ep,w,L)
H2=zeros(n1,n);
% Next we shall construct H-hat by condidering in 2 parts
% first for R(i,j) and second, for R(i,j-1)
% ***** For the element 1 to n-1 we obtain *******
for i=1:n1
    % when j=2...n-1
    for j=2:n-1
            % first consider for R(i,j)
            for g=1:10
            x1(g)=x(j)+(1/2)*((X(j+1)-X(j))*ep(g));
            y1(g)=y(j)+(1/2)*((Y(j+1)-Y(j))*ep(g));
            R2ij1(g)=(Xin(i)-x1(g))^2+(Yin(i)-y1(g))^2;
            S1(g)=w(g)*((1-ep(g))/R2ij1(g));
            end
            s1=sum(S1);
            ss1=(-L(j)*din(i,j)/(4))*(s1);
            % **** next consider for R(i,j-1)
            for g=1:10
            x3(g)=x(j-1)+(1/2)*((X(j)-X(j-1))*ep(g));
            y3(g)=y(j-1)+(1/2)*((Y(j)-Y(j-1))*ep(g));
            R2ij3(g)=(Xin(i)-x3(g))^2+(Yin(i)-y3(g))^2;
            S3(g)=w(g)*((1+ep(g))/R2ij3(g));
             end
            s3=sum(S3);
            %-----
```

```
                ss2=(-L(j)*din(i,j-1)/(4))*(s3);
                %*********
                 H2(i,j)=(ss1+ss2);
        end
    end
 % ********* For an element j=1 ***********
 % first consider for R(i,j)
        for i=1:n1
                for g=1:10
                x1(g)=x(1)+(1/2)*((X(2)-X(1))*ep(g));
                y1(g)=y(1)+(1/2)*((Y(2)-Y(1))*ep(g));
                R2ij1(g)=(Xin(i)-x1(g))^2+(Yin(i)-y1(g))^2;
                S1(g)=w(g)*((1-ep(g))/R2ij1(g));
                end
                s1=sum(S1);
                ss1=(-L(1)*din(i,1)/(4))*(s1);
                % **** next consider for R(i,j-1)
                for g=1:10
                x3(g)=x(n)+(1/2)*((X(1)-X(n))*ep(g));
                y3(g)=y(n)+(1/2)*((Y(1)-Y(n))*ep(g));
                R2ij3(g)=(Xin(i)-x3(g))^2+(Yin(i)-y3(g))^2;
                S3(g)=w(g)*((1+ep(g))/R2ij3(g));
                 end
                s3=sum(S3);
                %----
                ss2=(-L(1)*din(i,n)/(4))*(s3);
                %*********
                H2(i,1)=(ss1+ss2);

        end
 %******* For an element j=n  ****************
    for i=1:n1
 % first for R(i,j)
              for g=1:10
               x1(g)=x(n)+(1/2)*((X(1)-X(n))*ep(g));
               y1(g)=y(n)+(1/2)*((Y(1)-Y(n))*ep(g));
               R2ij1(g)=(Xin(i)-x1(g))^2+(Yin(i)-y1(g))^2;
               S1(g)=w(g)*((1-ep(g))/R2ij1(g));
              end
             s1=sum(S1);
             %----
               ss1=(-L(n)*din(i,n)/(4))*(s1);
              % Next for R(i,j)
              for g=1:10
               x3(g)=x(n-1)+(1/2)*((X(n)-X(n-1))*ep(g));
               y3(g)=y(n-1)+(1/2)*((Y(n)-Y(n-1))*ep(g));
               R2ij3(g)=(Xin(i)-x3(g))^2+(Yin(i)-y3(g))^2;
               S3(g)=w(g)*((1+ep(g))/R2ij3(g));
              end
             s3=sum(S3);
             %-----
               ss2=(-L(n)*din(i,n-1)/(4))*(s3);
               %-------
              H2(i,n)=(ss1+ss2);
end
%*********************************************************************
```

```
%*********************************************************************
function [G]=matrixg(n,X,Y,x,y,d,ep,w,L)
G=zeros(n);
% Next we shall construct H-hat by condidering in 2 parts
% first for R(i,j) and second, for R(i,j-1)
% ***** For the element 1 to n-1 we obtain *******
for i=1:n
    % when j=2...n-1
    for j=2:n-1
                % first consider for R(i,j)
                for g=1:10
                x1(g)=x(j)+(1/2)*((X(j+1)-X(j))*ep(g));
                y1(g)=y(j)+(1/2)*((Y(j+1)-Y(j))*ep(g));
                Rij1(g)=sqrt((X(i)-x1(g))^2+(Y(i)-y1(g))^2);
                S1(g)=w(g)*(1-ep(g))*log(Rij1(g));
                end
                s1=sum(S1);
                ss1=(-L(j)/4)*(s1);
                % **** next consider for R(i,j-1)
                for g=1:10
                x3(g)=x(j-1)+(1/2)*((X(j)-X(j-1))*ep(g));
                y3(g)=y(j-1)+(1/2)*((Y(j)-Y(j-1))*ep(g));
                Rij3(g)=sqrt((X(i)-x3(g))^2+(Y(i)-y3(g))^2);
                S3(g)=w(g)*(1+ep(g))*log(Rij3(g));
                 end
                s3=sum(S3);
                %-----
                ss2=(-L(j)/4)*(s3);
                %*********
                 G(i,j)=ss1+ss2;
        end
    end
 % ********* For an element j=1 ***********
 % first consider for R(i,j)
        for i=1:n
                for g=1:10
                x1(g)=x(1)+(1/2)*((X(2)-X(1))*ep(g));
                y1(g)=y(1)+(1/2)*((Y(2)-Y(1))*ep(g));
                Rij1(g)=sqrt((X(i)-x1(g))^2+(Y(i)-y1(g))^2);
                S1(g)=w(g)*(1-ep(g))*log(Rij1(g));
                end
                s1=sum(S1);
                ss1=(-L(1)/4)*(s1);
                % **** next consider for R(i,j-1)
                for g=1:10
                x3(g)=x(n)+(1/2)*((X(1)-X(n))*ep(g));
                y3(g)=y(n)+(1/2)*((Y(1)-Y(n))*ep(g));
                Rij3(g)=sqrt((X(i)-x3(g))^2+(Y(i)-y3(g))^2);
                S3(g)=w(g)*(1+ep(g))*log(Rij3(g));
                 end
                s3=sum(S3);
                ss2=(-L(1)/4)*(s3);
                %*********
                G(i,1)=ss1+ss2;
        end
 %******* For an element j=n   ****************
    for i=1:n
 % first for R(i,j)
```

```
            for g=1:10
             x1(g)=x(n)+(1/2)*((X(1)-X(n))*ep(g));
             y1(g)=y(n)+(1/2)*((Y(1)-Y(n))*ep(g));
             Rij1(g)=sqrt((X(i)-x1(g))^2+(Y(i)-y1(g))^2);
             S1(g)=w(g)*(1-ep(g))*log(Rij1(g));
            end
           s1=sum(S1);
           %----
             ss1=(-L(n)/4)*(s1);
            % Next for R(i,j)
            for g=1:10
             x3(g)=x(n-1)+(1/2)*((X(n)-X(n-1))*ep(g));
             y3(g)=y(n-1)+(1/2)*((Y(n)-Y(n-1))*ep(g));
             Rij3(g)=sqrt((X(i)-x3(g))^2+(Y(i)-y3(g))^2);
             S3(g)=w(g)*(1+ep(g))*log(Rij3(g));
            end
           s3=sum(S3);
           %-----
             ss2=(-L(n)/4)*(s3);
             %-------
            G(i,n)=ss1+ss2;
end
% Next we consider Hhat's values when i=j
    for i=1:n
       G(i,i)=-(L(i)/2)*(log(L(i))-(3/2))-(L(i)/2)*(log(L(i))-(3/2));
    end

%**********************************************************************


%**********************************************************************
function [G2]=matrixg2(n,n1,X,Y,Xin,Yin,x,y,din,ep,w,L)
G2=zeros(n1,n);
% Next we shall construct H-hat by condidering in 2 parts
% first for R(i,j) and second, for R(i,j-1)
% ***** For the element 1 to n-1 we obtain *******
for i=1:n1
    % when j=2...n-1
    for j=2:n-1
            % first consider for R(i,j)
            for g=1:10
            x1(g)=x(j)+(1/2)*((X(j+1)-X(j))*ep(g));
            y1(g)=y(j)+(1/2)*((Y(j+1)-Y(j))*ep(g));
            Rij1(g)=sqrt((Xin(i)-x1(g))^2+(Yin(i)-y1(g))^2);
            S1(g)=w(g)*(1-ep(g))*log(Rij1(g));
            end
            s1=sum(S1);
            ss1=(-L(j)/4)*(s1);
            % **** next consider for R(i,j-1)
            for g=1:10
            x3(g)=x(j-1)+(1/2)*((X(j)-X(j-1))*ep(g));
            y3(g)=y(j-1)+(1/2)*((Y(j)-Y(j-1))*ep(g));
            Rij3(g)=sqrt((Xin(i)-x3(g))^2+(Yin(i)-y3(g))^2);
            S3(g)=w(g)*(1+ep(g))*log(Rij3(g));
             end
            s3=sum(S3);
            %-----
            ss2=(-L(j)/4)*(s3);
```

```
                %*********
                 G2(i,j)=(ss1+ss2);
        end
    end
% ********* For an element j=1 ***********
% first consider for R(i,j)
        for i=1:n1
                for g=1:10
                x1(g)=x(1)+(1/2)*((X(2)-X(1))*ep(g));
                y1(g)=y(1)+(1/2)*((Y(2)-Y(1))*ep(g));
                Rij1(g)=sqrt((Xin(i)-x1(g))^2+(Yin(i)-y1(g))^2);
                S1(g)=w(g)*(1-ep(g))*log(Rij1(g));
                end
                s1=sum(S1);
                ss1=(-L(1)/4)*(s1);
                % **** next consider for R(i,j-1)
                for g=1:10
                x3(g)=x(n)+(1/2)*((X(1)-X(n))*ep(g));
                y3(g)=y(n)+(1/2)*((Y(1)-Y(n))*ep(g));
                Rij3(g)=sqrt((Xin(i)-x3(g))^2+(Yin(i)-y3(g))^2);
                S3(g)=w(g)*(1+ep(g))*log(Rij3(g));
                 end
                s3=sum(S3);
                ss2=(-L(1)/4)*(s3);
                %*********
                G2(i,1)=(ss1+ss2);
        end
%******* For an element j=n  ****************
    for i=1:n1
% first for R(i,j)
            for g=1:10
             x1(g)=x(n)+(1/2)*((X(1)-X(n))*ep(g));
             y1(g)=y(n)+(1/2)*((Y(1)-Y(n))*ep(g));
             Rij1(g)=sqrt((Xin(i)-x1(g))^2+(Yin(i)-y1(g))^2);
             S1(g)=w(g)*(1-ep(g))*log(Rij1(g));
            end
           s1=sum(S1);
           %----
             ss1=(-L(n)/4)*(s1);
            % Next for R(i,j)
            for g=1:10
             x3(g)=x(n-1)+(1/2)*((X(n)-X(n-1))*ep(g));
             y3(g)=y(n-1)+(1/2)*((Y(n)-Y(n-1))*ep(g));
             Rij3(g)=sqrt((Xin(i)-x3(g))^2+(Yin(i)-y3(g))^2);
             S3(g)=w(g)*(1+ep(g))*log(Rij3(g));
            end
           s3=sum(S3);
           %-----
             ss2=(-L(n)/4)*(s3);
             %-------
            G2(i,n)=(ss1+ss2);
end
%**********************************************************************
```