```
l=0;
n=5;                                                    %grid size
for a=1:1:n
    for b=1:1:n
            l=l+1;
            A(a,b)=rand;                                %potential seed function
            position(l,1)=a;                            %Storing the x position of the grid
            position(l,2)=b;                            %Storing the y position of the grid
    end
end

q=1;                                                    %Particle charge
m=1;                                                    %Particle mass
particlex=5.5 ;                                         %initial particle x-position
particley=2 ;                                           %initial particle y-position

for t=0:pi/4:2*pi
    l=0;                                                %Reset the array position to 0
   for j=1:1:n
       for k=1:1:n
         l=l+1;                                         %Move to next row in the array
         fprintf('%f',t);
         potentialt=(0.9)*A(j,k)+0.0001*(rand-0.5);     %Random walk based on potential at previous time
         A(j,k)=potentialt                              %potential at each grid point
         position(l,3)=potentialt                       %Stores potential in an array related to a grid point

       end
   end
end
```

```
for l=1:1:n
    A(l,n)=A(l,2)
    A(l,1)=A(l,n-1)
    A(1,l)=A(n-1,l)
    A(n,l)=A(2,l)
    A(n,n)=A(2,2)
    A(1,1)=A(n-1,n-1)
    A(n,1)=A(2,n-1)
    A(1,n)=A(n-1,2)
end
fprintf('%f',A(1,2))

        scatter3(position(:,1),position(:,2),position(:,3));
        sf=fit([position(:,1), position(:,2)], position(:,3), 'thinplateinterp');    %fits surface to potential
        plot(sf,[position(:,1), position(:,2)], position(:,3));

if (particlex<=n && particley<=n)                       %For particle within the grid
    [gx, gy] = differentiate(sf, [particlex,particley]) %Electric field calculated as derivative of potential at particle positi
     gx=-ex
     gy=-ey
     fx = q*ex                                          %Force at particle position in x direction
     fy = q*ey                                          %Force at particle position in y direction
     ax = fx/m                                          %Acceleration of particle in x direction
     ay = fy/m                                          %Acceleration of particle in y direction
     particlex = particlex+ax*(pi/4)^2                  %Particle position along the x axis
     particley = particley+ay*(pi/4)^2                  %Particle position along the y axis

elseif (particlex>n-1 && particley<=n-1)                %Particle moves out of the grid in the positive x direction
    partpot=interp2(A,particlex-(n-1),particley)

elseif (particlex<=n-1 && particley>n-1)                %Particle moves out of the grid in the positive y direction
    partpot=interp2(A,particlex,particley-(n-1))

else
```

```
l=0;
n=5;                                                    %grid size
for a=1:1:n
    for b=1:1:n
            l=l+1;
            A(a,b)=rand;                                %potential seed function
            position(l,1)=a;
            position(l,2)=b;
    end
end
l=0;
for a=2:1:n-1
    for b=2:1:n-1
            l=l+1;
            A(a,b)=rand;                                 %potential seed function
            grid(l,1)=a;
            grid(l,2)=b;
    end
q=1;
m=1;
particlex=2 ;                                           %initial particle x-position
particley=2 ;                                            %initial particle y-position
for t=0:pi/4:2*pi
    l=0;
   for j=1:1:n
       for k=1:1:n
         l=l+1;
         fprintf('%f',t);
         potentialt=(0.9)*A(j,k)+0.0001*(rand-0.5);
         A(j,k)=potentialt                              %potential at each grid point
         position(l,3)=potentialt

         if (2<=k<=n-1)
             B(j,k-1)=potentialt
             grid(l,3)=potentialt
         end

       end
   end
end
```

```
A =

    0.0179    0.0124    0.1072    0.0179    0.0124
    0.1520    0.2935    0.2879    0.1520    0.2935
    0.2735    0.2539    0.0663    0.2735    0.2539
    0.0179    0.0124    0.1072    0.0179    0.0124
    0.1520    0.2935    0.2879    0.1520    0.2935
```