M27_10_13_Interpolating_potential.... | M27_10_13_Interpolating_potential_... | M27_10_13_Interpolating_potential_... | M27_10_13_phantom_grid.m | M27_10_13_phantom_grid_sn

```
    l=0

for a=1:1:5

        for b=1:1:5
            l=l+1;
            A(a,b)=rand;                                %Potential seed function
            position(l,1)=a;
            position(l,2)=b;
        end
end

q=1                                                     %Particle charge
particlex=2                                             %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=1:1:5
        for k=1:1:5
          l=l+1;                                        %Move to next row in 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;                     %Store each potential in an array related to a grid point
        [ex, ey] = differentiate(sf, [j, k])            %Electric field calculated as derivative of potential
        fx(j,k) = q*ej                                  %Force in x direction
        fy(j,k) = q*ek                                  %Force in y direction
        end
    end

        scatter3(position(:,1),position(:,2),position(:,3))
        sf=fit([position(:,1), position(:,2)], position(:,3), 'biharmonicinterp')  %fits surface to potential
        plot(sf,[position(:,1), position(:,2)], position(:,3))
        figure
        partpot=interp2(A,particlex,particley)                                    %potential at particle position
        partforcej=interp2(fj,particlex,particley)                                %force in j direction at particle position
        partforcek=interp2(fk,particlex,particley)                                %force in k direction at particle position

end
```

M27_10_13_Interpolating_potential.... | M27_10_13_Interpolating_potential_... | M27_10_13_Interpolating_potential_... | M27_10_13_phantom_grid.m | M27_10_13_phantom_g

```
    l=0

for a=1:1:5

        for b=1:1:5
            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=2                                             %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=1:1:5
        for k=1:1:5
          l=l+1;                                        %move to next row in 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;                     %store each potential in an array related to a grid point

        end
    end
        scatter3(position(:,1),position(:,2),position(:,3))
        sf=fit([position(:,1), position(:,2)], position(:,3), 'biharmonicinterp')    %fits surface to potential
        plot(sf,[position(:,1), position(:,2)], position(:,3))
        figure
        [ex, ey] = differentiate(sf, [particlex,particley])                  %Electric field calculated as derivative of potential at particle position
        fx = q*ex                                       %Force at particle position in j direction
        fy = q*ey                                       %Force at particle position in k direction
        ax = fx/m                                       %acceleration in x direction
        ay = fy/m                                       %acceleration in y direction
        particlex = particlex+ax*(pi/4)^2               %particle x position at each time
        particley = particley+ay*(pi/4)^2               %particle y position at each time
end
```

M27_10_13_Interpolating_potential.... | M27_10_13_Interpolating_potential_... | M27_10_13_Interpolating_potential_... | M27_10_13_phantom_grid.m | M27_10_13_phantom_grid_smaller_...

```
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
dt=pi/4;                                    %Time step

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 array
         fprintf('%f',n);
         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;          %Store each potential in an array related to a grid point
       end
   end

        scatter3(position(:,1),position(:,2),position(:,3));
        sf=fit([position(:,1), position(:,2)], position(:,3), 'biharmonicinterp');    %fits surface to potential
        plot(sf,[position(:,1), position(:,2)], position(:,3));
       if (particlex<=n && particley<=n)                                    %For particle within the grid
        [ex, ey] = differentiate(sf, [particlex,particley])                 %Electric field calculated as derivative of potential at particle position
        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*dt^2                                       %Particle x position at each time
        particley = particley+ay*dt^2                                       %Particle y position at each time
       elseif (particlex>n-1 && particley<=n-1)                             %If particle moves out of the grid
        partpot=interp2(A,particlex-(n),particley)
       elseif (particlex<=n-1 && particley>n-1)
        partpot=interp2(A,particlex,particley-(n))
       else


       end

end

```