Charge Distribution on a conducting plate

Bijin Joseph (182361). Hetanshu Bharadiya (182343)

26th February 2021

Abstract

In this project we are simulating a system consisting of a conducting plate with charge distributed on it and a charge at a distance d above the plate. We will be carrying out a set of experiments which include a neutral plate, a plate with a single charge distribution and a system with varying d. The simulation is carried out in python programming language and we will be making use of the Monte Carlo method.

Monte Carlo method, Charge Distribution, Python, Method of images

1 Introduction

The behaviour of charges in a conductor has always been an interesting field of study for physicist and the present theories took decades of analysis. However, this has led to a very solid theory of electrostatics and in this project we analyse a situation using this theory. We know that when we bring a positive or negative charge close to a grounded conductor, the conductor will no longer remain neutral. We also know that a conductor has equal number of positive charges and negative charges and that the charges outside will attract the opposite charge and repel the negative charge. Here in this project we analyse a similar and certain different situations by developing a python simulation and further we compare our results to experimentally proved results. Of course the result from this project is an approximation, as due the limited computing power we can only stick to a limited number of charges. However, even with the limitations, this project was successful in providing expected and certain interesting results.

2 Discussion

2.1 Theory

For this simulation we make use of Columbus law, which states that the force of on a charge will be

$$F = \frac{1}{4\pi\epsilon_o}\frac{q_1 q_2}{d^2} \tag{1}$$

where d is the distance between charges q_1 and q_2.

From this we get the energy to be

$$E = \frac{1}{4\pi\epsilon_o}\frac{q_1 q_2}{d} \tag{2}$$

Therefore if a conductor has n number of charges, then the total energy will be

$$\boxed{E = \Sigma_i^n \left(\Sigma_{j=i+1}^n \frac{1}{4\pi\epsilon_o}\frac{q_i q_j}{d_{i,j}}\right)} \tag{3}$$

We know the universal rule that the charges should be at the lowest energy and this is a fact that will be the key ingredient in our code.
Here we will be analysing several situations and all of these situations will have one thing in common and that is, there is a plate and a charge at a distance, d above the plate. The situations include,

- q_n positive charges on the plate and a positive charge at a distance d (very small) above the plate
- q_n negative charges on the plate and a positive charge at a distance d (very small) above the plate
- $q_n/2$ positive charges and $q_n/2$ negative charges on the plate (neutral) and a a positive charge at a distance d (very small) above the plate
- q_n positive charges on the plate and a positive charge at a different distance d above the plate

The reason why we take the first case with a very small distance d, is to make the plate infinite compared to d, so that we can compare the results with the theoretical results.
When including the charge at a distance d above the plate, the new energy term will be,

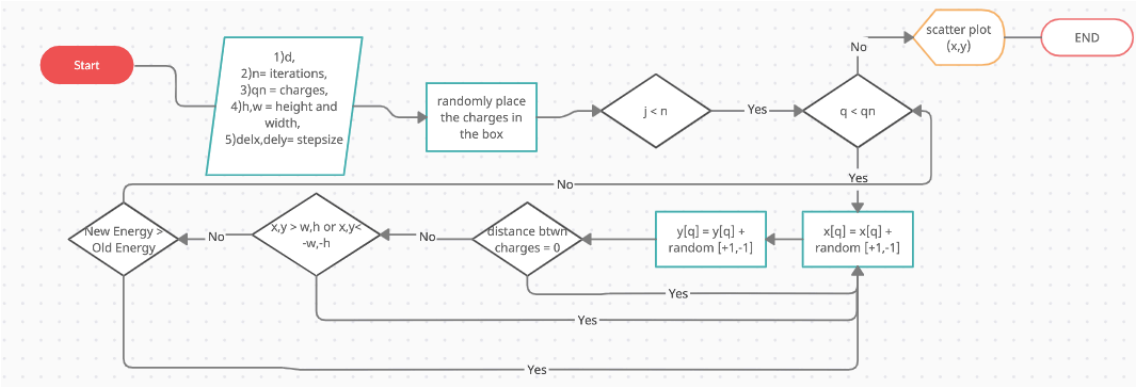
$$\boxed{E = \Sigma_i^n \left[\Sigma_{j=i+1}^n \left(\frac{1}{4\pi\epsilon_o}\frac{q_i q_j}{d_{i,j}}\right) + \frac{1}{4\pi\epsilon_o}\frac{Q q_i}{d_i}\right]} \tag{4}$$

Here Q is the charge at a distance d above the surface and q_n = Q/n, where n is the number of charges.
From the method of Images, it was found that the charge density on a conducting infinite plate, we a charge (consider positive) is held at a distance d above the plate is

$$\boxed{\sigma(x,y) = \frac{-qd}{2\pi\left(x^2+y^2+d^2\right)^{3/2}}} \tag{5}$$

We can use this equation to analyse our results.

2.2 Code



Flow chart of the code

```python
import numpy as np
import matplotlib.pyplot as plt
import random as rn
import time



def Emt(x,y, Em, Ed):
    Etotal = 0
    for i in range(len(x)):
        for j in range(i+1,len(x)):
            Em[i,j] = (1*Q[i]*Q[j]/(np.sqrt((x[i]-x[j])**2 + (y[i]-y[j])**2)))
            Etotal = Etotal+Em[i,j]
        Ed[i] = (1*Q[i]/np.sqrt(d**2 + x[i]**2 + y[i]**2))
        Etotal = Etotal+ Ed[i]
    return (Em, Ed, Etotal)


def Enew(Em,x,y, i, Q, d, Ed, Etotal):
    E = 0
    Eold = 0
    for q in range(0,i):
            Eold = Eold+ Em[q,i]
            Em[q,i] = 1*Q[q]*Q[i]/(np.sqrt((x[q]-x[i])**2 + (y[q]-y[i])**2))
            E += Em[q,i]
    for j in range(i+1, len(x)):
            Eold =Eold+ Em[i,j]
            Em[i,j] = 1*Q[i]*Q[j]/(np.sqrt((x[i]-x[j])**2 + (y[i]-y[j])**2))
            E = E+Em[i,j]
    Eold = Eold+ Ed[i]
    Ed[i] = (1*Q[i]/np.sqrt(d**2 + x[i]**2 + y[i]**2))
    E = E + Ed[i]
    Etotal = Etotal - Eold + E
    return(Etotal, Em, Ed)

def plot(x,y,c,j):
 plt.clf()
 plt.figure()
 for i in range(len(x)):
  plt.scatter(x[i],y[i], c = c[i])
 plt.plot(0,0, 'o', color = 'red')
 plt.title(j)
 plt.xlim(-w,w)
 plt.ylim(-h,h)
 plt.show()

def dis(x,y) :
   for i in range(len(x)):
        for j in range(i+1,len(x)):
            dist = np.sqrt((x[i]-x[j])**2 + (y[i]-y[j])**2)
            if dist <1:
                return (0 )
   return (1 )

d = 50
n=1000  #number of iteration
```

```
qn =  50   #number of charges
x = np.zeros(qn) #x position
y = np.zeros(qn) #y position
h = 100 #height
w = 100 #width
Q = np.zeros(qn)
c = []
Em = np.zeros([qn,qn])
Ed = np.zeros([qn])
delx =[-1,1] #step size
dely =[-1,1]
#initial position
for i in range(qn):
    x[i] = rn.randint(-h,h)
    y[i] = rn.randint(-w,w)
    if i < qn/2:
        Q[i] = -1/qn
        c.append('b')
    else:
        Q[i] = -1/qn
        c.append('b')
j = 0
plot(x,y,c,j)
Em, Ed, ET = Emt(x,y, Em, Ed)
print(ET)

for j in range(n):


  #plot(x,y,c,j)
  for qr in range(0,qn):

      #qr = rn.randint(0,qn-1) #choosing charges randomly
      xc = x[qr]
      yc = y[qr]
      Et = ET
      Emt = Em
      Edt = Ed
      dx = rn.randint(-1,1)   #changing the postion randomly
      dy = rn.randint(-1,1)
      x[qr] = x[qr]+dx
      y[qr] = y[qr]+dy

      #checking if the position is within the box
      if dis(x,y) == 0:
            x[qr] = xc
            y[qr] = yc
            j = j-1
            #print('x')
      elif (x[qr]>w or y[qr]>h) or (x[qr]<-w or y[qr]<-h):
            x[qr] = xc
            y[qr] = yc


      else:
            ET,Em, Ed = Enew(Em,x,y, qr, Q, d, Ed, ET)
           #checking if the energy condition is satisfied
            if ET > Et:
               x[qr] = xc
               y[qr] = yc
               ET = Et
               Em = Emt
               Ed = Edt


#plotting the result

plot(x,y,c,j)


def density(x,y,r1, r2):

 distance = np.sqrt(x**2 + y**2)
 keep =  (distance < r2) &  (distance  > r1)
 return (len(distance[keep]))
den = []

for r in np.arange(0,20):
    r1 = r*5
    r2 = (r1+5)
    den.append(density(x,y,r1, r2))

plt.plot(np.arange(0,10)*10, den)
print(den)
```

2.3 Results

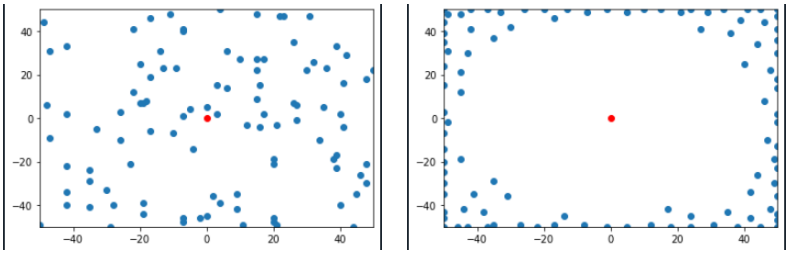


Figure 1: **100 positive charges on the plate and a positive charge at a distance d = 1 (very small) above the plate. The image on the left is for iteration 0 and image on the right for iteration 999**

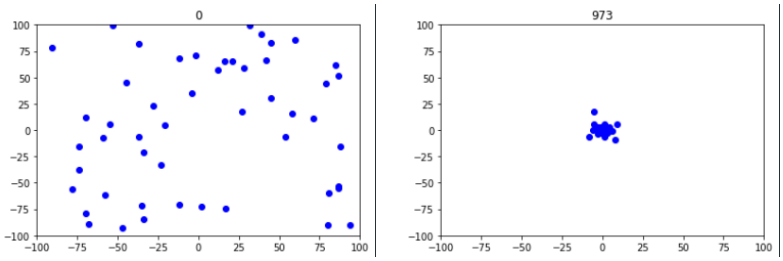


Figure 2: **100 negative charges on the plate and a positive charge at a distance d = 1 (very small) above the plate. The image on the left is for iteration 0 and image on the right for iteration 973**

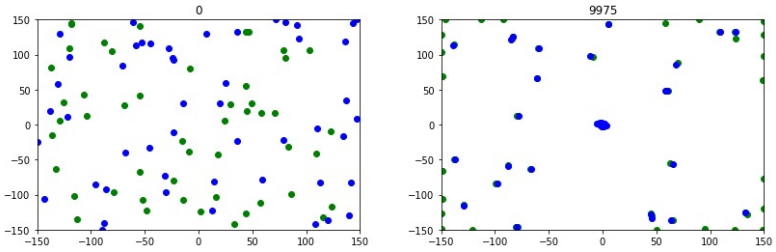


Figure 3: **500 positive charges and 500 negative charges on the plate (neutral) and a a positive charge at a distance d = 1 (very small) above the plate.The image on the left is for iteration 0 and image on the right for iteration 9975**

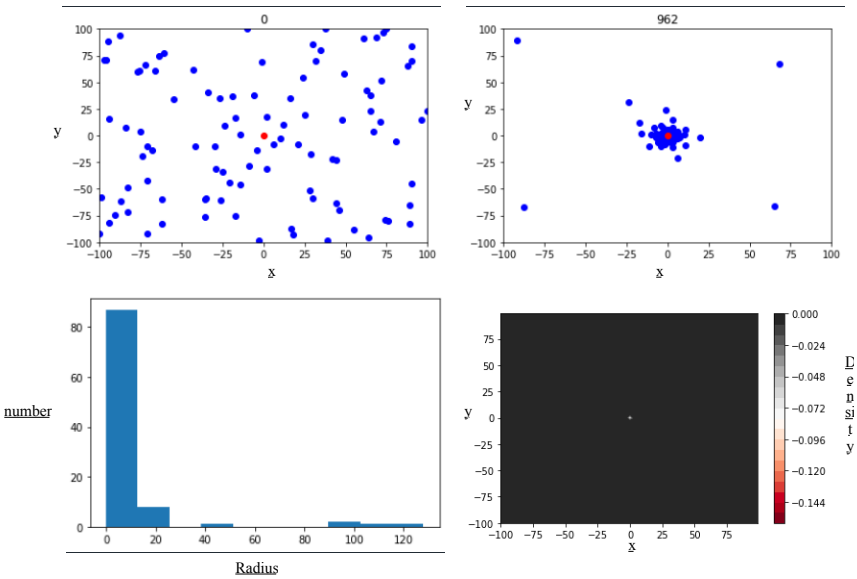


Figure 4: **100 negative charges at a distance of 1 unit above the plate.**

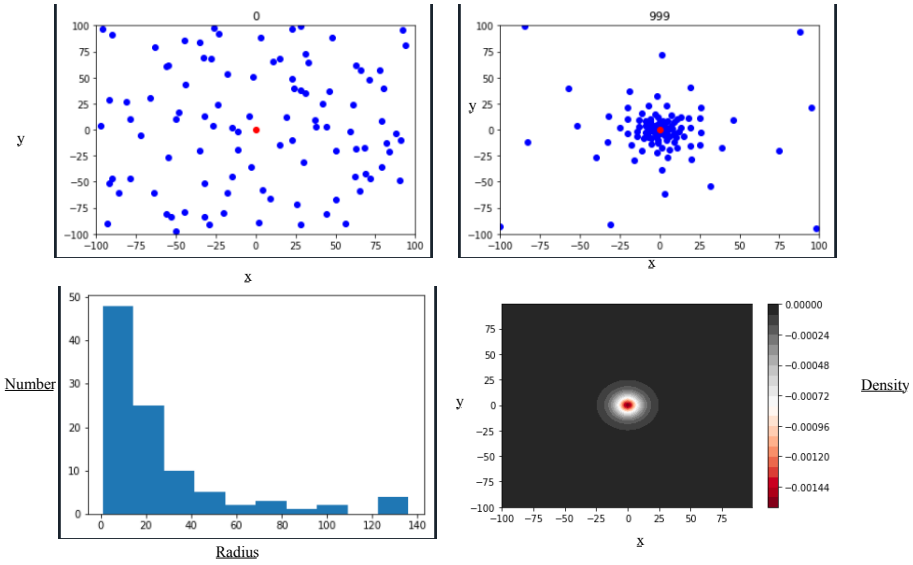


Figure 5: **100 negative charges at a distance of 10 unit above the plate.**

- **In Figure 1**, we have taken a conducting plate with 100 positive charges and at the centre about a distance 1 unit above it, we have placed a positive charge. From the result we can see that the positive charges spread out to the sides as expected. The charges have also spread away from each

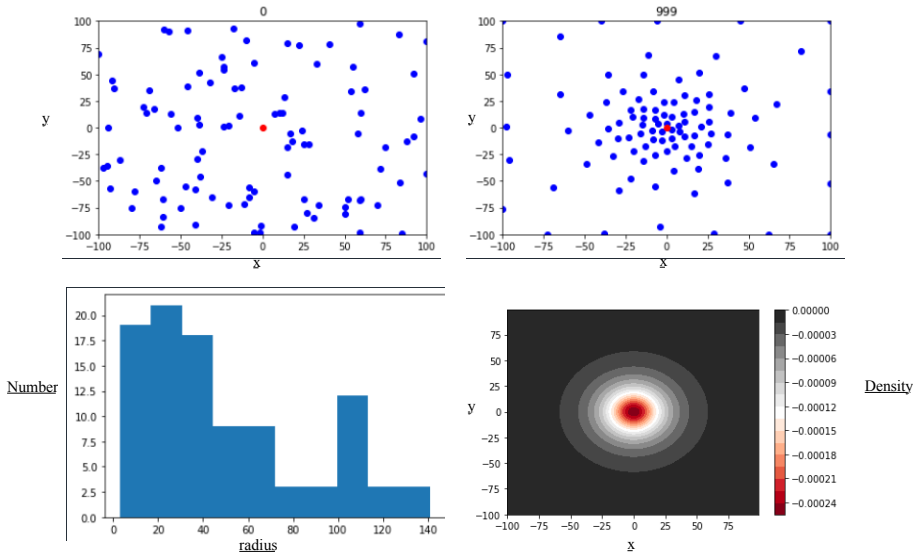


Figure 6: **100 negative charges at a distance of 25 unit above the plate.**

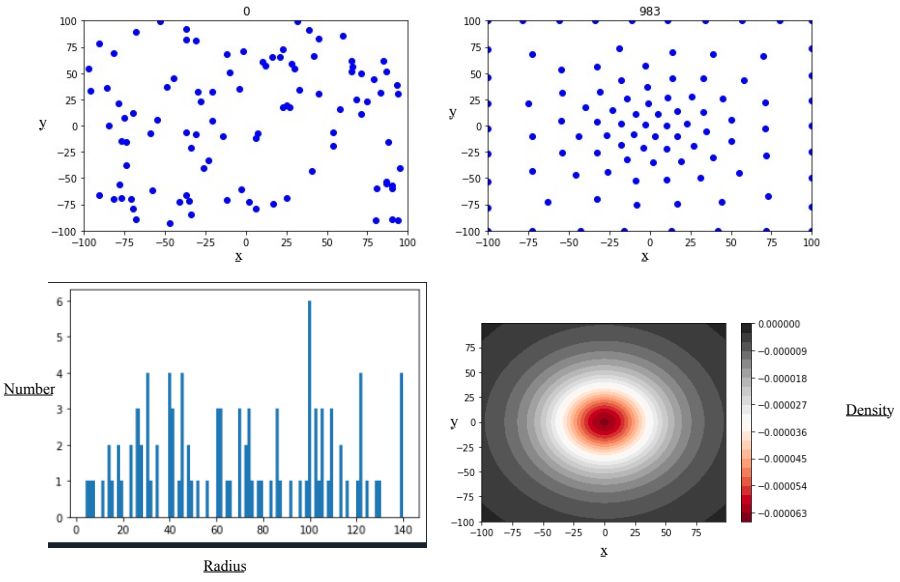


Figure 7: **100 negative charges at a distance 50 units above the plate.**

other due to repulsion.

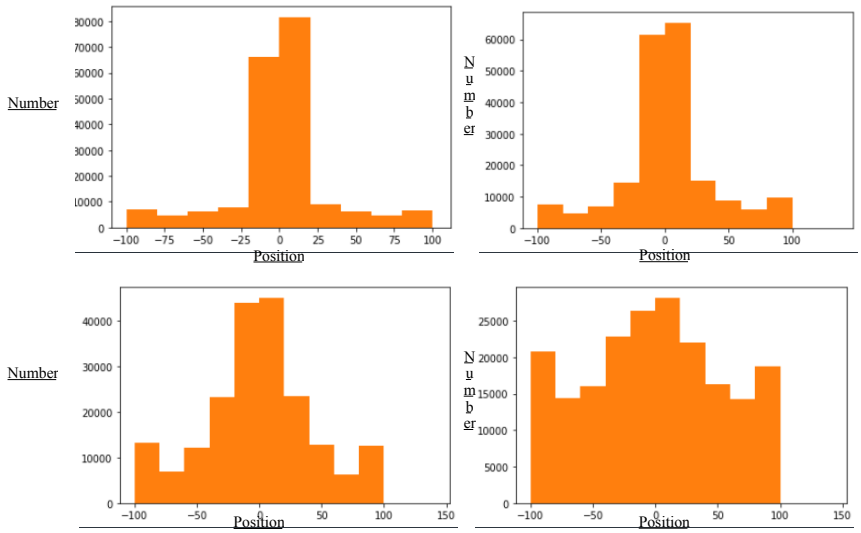


Figure 8: **for d= 1,10,25 and 50 respectively**

- **In Figure 2**, we have taken a conducting plate with 100 negative charges and at the centre about a distance 1 unit above it, we have placed a positive charge. As expected we see most of the charges get accumulated near the centre. However, since the code is in such a way that, no two charges can occupy the same position, all of the charges wont be found at the centre. Along with that the charges are repelling each other and therefore we see a spread in the distribution.

- **In Figure 3**, We have equal number of positive charges and negative charges (a neutral plate). Here, when we keep a positive charge at the centre, the negative charges(blue) are attracted towards the centre and the positive charges (green) are repelled towards the boundaries. However, we can also see pairs of positive charges and negative charges spread over the plate; this is because, the initial position of the charges are randomly assigned and sometimes positive charges and negative charges maybe close to each other. In such a situation they attract each other and come together to be neutral, and ones they are neutral they wont move and will remain stationary.

- **In Figure 4-7** we have negative charges distributed over the plate and a positive charge is placed at a distance d = 1,10,25 and 50, over the plate. The top two figure of each set, are the positions of the charges after the first and the last iteration. The third image in the set is a histogram which shows how many particles are there in the range r and $r + dr$, where r is the radius from the centre and the fourth image is a graph generated using equation 5 and by analysing the different graphs we can see that

the code is giving the correct result until when distance $d >= 25$. This is because of the limited number of charges we are taking and since we have assigned each charge a magnitude of q/n, there will be a significant repulsion between them and therefore the discrepancy.

- **In figure 8** We have plotted a histogram of the probabilities of the x and y positions that a particle can take to find out which position has maximum probability. Like expected, the centre, i.e the position of the charge above the plate, has the maximum probability. The boundaries also has a higher probability and that is because of the finite nature of the plate.

3 Conclusion

From this experiment we have successfully simulated how a charge affects a conducting plate under different conditions and we have compared it to theoretical results. From the comparison we can observe that the simulation results were correct but as the distance of the charge from the plate increased, the results was not as theoretically predicted. The difference is because of the finite number of charges and the limited computing power that was available.

4 Acknowledgment

First and foremost we would like to thank the physics department of St. Xaviers College, Mumbai, for giving us this opportunity to perform this project which has helped us to learn more and understand theory better. We specifically thank Professor Rohan Jadhav for approving this project and for being with us throughout this project to help us and guide us whenever it was required.

References

[1] David J Griffiths "Introduction to Electrodynamics"

[2] Werner Krauth "Statistical Mechanics: Algorithms and Computations"