**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МО ЭВМ**

**отчет**

**по лабораторной работе №4**

**по дисциплине «Физические основы информационных технологий»**

**Тема: Построение силовых линий электростатического поля**

**Вариант 19**

|  |  |  |
| --- | --- | --- |
| Студент гр. 6304 |  | Виноградов К.А. |
| Преподаватель |  | Альтмарк А.М. |

Санкт-Петербург

2018

**Цель работы**

Изучить моделирование и визуализацию электростатического поля.

**Задание**

Построить силовые линии для указанной на рисунке электростатической системы. Красный электрод является положительно заряженным с зарядом Q, черный электрод является отрицательно заряженным электродом с зарядом -Q. Необходимо построить 20 силовых линий, которые начинаются на положительных электродах, а также 10 эквипотенциальных линий (линий с одинаковым потенциалом). Вся длина положительно заряженного электрода должна быть равномерно разбита на 20 интервалов, границы которых будут являться стартовыми точками силовых линий. Результат работы представить в виде картинки в любом из нижеперечисленных форматах: bmp, tiff, emf или jpeg.

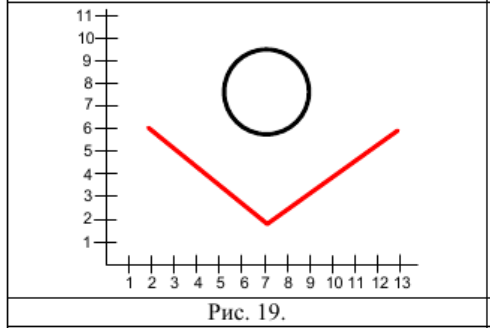


Рисунок 1 – Вариант задания

**Ход работы**

1. Зададим константы в программе, которые отвечают варианту.
2. Зададим точки отрицательного и положительного электрода.
3. Зададим функцию для получения следующей точки силовой линии от текущей, передаваемой в качестве параметра.
4. Построим картину силовых линий.

Исходный код программы приведён в приложении А. Построенная картина силовых линий в приложении Б.

**Выводы**

В ходе выполнения данной лабораторной работы были визуализированы силовые линии электростатического поля, которые наглядно отображают картину электростатического поля между электродами.

**ПРИЛОЖЕНИЕ А**

**Код программы**

# coding: utf-8  
  
import numpy as np  
from matplotlib import pyplot as plt  
  
# Кнстанты точек для линий  
  
num\_dots = 1000  
num\_dots\_half = int(num\_dots/2)  
num\_lines = 20  
lines\_half = int(num\_lines/2)  
  
  
# Константы варианта  
  
x\_left = 2  
y\_left = 6  
x\_mid = 7  
y\_mid = 1.5  
x\_mid\_left = 6.6  
y\_mid\_left = 1.8  
x\_mid\_right = 7.4  
y\_mid\_right = 1.8  
x\_right = 12  
y\_right = 6  
x\_plus = 5  
y\_plus\_min = 3.5  
y\_plus\_max = 8.5  
  
minus\_center = (7, 7.5)  
minus\_radius = 2  
  
# Массив точек (х, у) из которых будут строится силовые линии  
line\_dots\_left = np.array([np.linspace(x\_left, x\_mid\_left, lines\_half), np.linspace(y\_left, y\_mid\_left, lines\_half)])  
line\_dots\_right = np.array([np.linspace(x\_mid\_right, x\_right, lines\_half), np.linspace(y\_mid\_right, y\_right, lines\_half)])  
line\_dots = np.append(line\_dots\_left, line\_dots\_right, axis=1).T  
  
# Массив точек (х, у) отрицательного электрода  
minus = np.array([(minus\_center[0] + minus\_radius \* np.cos(i \* 360 / num\_dots \* np.pi / 180),  
 minus\_center[1] + minus\_radius \* np.sin(i \* 360 / num\_dots \* np.pi / 180))  
 for i in range(num\_dots)])  
  
# Массив точек (х, у) положительного электрода  
plus\_left = np.array([np.linspace(x\_left, x\_mid, num\_dots\_half), np.linspace(y\_left, y\_mid, num\_dots\_half)])  
plus\_right = np.array([np.linspace(x\_mid, x\_right, num\_dots\_half), np.linspace(y\_mid, y\_right, num\_dots\_half)])  
plus = np.append(plus\_left, plus\_right, axis=1).T  
  
# Функция получения следующей точки для силовой линии  
  
  
def get\_next\_point(x, y):  
  
 x1, y1, x2, y2 = 0, 0, 0, 0  
  
 for minus\_point, plus\_point in zip(minus, plus):  
  
 r1 = ((x - plus\_point[0]) \*\* 2 + (y - plus\_point[1]) \*\* 2) \*\* 0.5  
  
 if r1 != 0:  
 x1 += (x - plus\_point[0]) / r1 \*\* 2  
 y1 += (y - plus\_point[1]) / r1 \*\* 2  
  
 r2 = ((x - minus\_point[0]) \*\* 2 + (y - minus\_point[1]) \*\* 2) \*\* 0.5  
  
 if r2 != 0:  
 x2 += (minus\_point[0] - x) / r2 \*\* 2  
 y2 += (minus\_point[1] - y) / r2 \*\* 2  
  
 x1 = x + (x1 + x2) \* 1e-4  
 y1 = y + (y1 + y2) \* 1e-4  
  
 return x1, y1  
  
  
plt.figure(figsize=(9, 7))  
plt.axes().set\_aspect('equal')  
  
for idx, line\_dot in enumerate(line\_dots):  
 count = 500 # количество точек в силовой линии  
 delta = 4e-2 if idx < 10 else -4e-2 # сдвиг силовой линии от положительного электрода к отрицательному  
  
 x, y = line\_dot[0] + delta, line\_dot[1] + abs(delta)  
  
 res\_x = [x]  
 res\_y = [y]  
  
 while True:  
 count -= 1  
  
 x, y = get\_next\_point(x, y)  
 distance = (minus\_center[0] - x) \*\* 2 + (minus\_center[1] - y) \*\* 2  
 if distance <= minus\_radius \*\* 2 or count < 0:  
 break  
  
 res\_x.append(x)  
 res\_y.append(y)  
  
 plt.plot(res\_x, res\_y, lw=1, c='blue')  
  
plt.plot(minus[:, 0], minus[:, 1], c='black', lw=2)  
plt.plot(plus[:, 0], plus[:, 1], c='red', lw=2)  
plt.xticks(range(int(min(minus[:, 0])) - 1, int(max(minus[:, 0])) + 1))  
plt.grid()  
plt.title('Силовые линии', fontsize=20)  
plt.show()

**ПРИЛОЖЕНИЕ Б**

**Картина силовых линий**

