Санкт-Петербургский Государственный Электротехнический Университет "ЛЭТИ"

кафедра физики

Задание №2 по дисциплине

"Физические основы информационных технологий"

Название: Численное решение уравнения Лапласа

|  |  |
| --- | --- |
| Фамилия И.О.: | Кузнецов Н.А. |
| группа: | 1303 |
| Преподаватель: | Альтмарк А.М. |
| Итоговый балл: |  |
|  |  |

Крайний срок сдачи: 05.11.23

Санкт-Петербург 2023

Условие задания

Дана электростатическая система, состоящая из трех электродов. Внешний электрод (на рисунке 1 отмечен синим цветом) обладает потенциалом 0 В. Внутренние электроды (на рисунке отмечены красным цветом и пронумерованы как 1 и 2) обладают потенциалами, отличными от 0. Исходные данные нужно взять в файле FOIT\_IDZ2.xlsx. Для одной из указанных в таблице эквипотенциальных линий необходимо найти длину и записать её в файл IDZ2.txt. Контуры электродов можно построить по формулам, указанным в таблице и сравнить с соответствующим изображением в jpeg – файле. Координаты в данном задании можно считать безразмерными.

Помимо текстового файла IDZ2.txt в папке IDZ2 должен находиться Word-файл с отчетом, а также файл с кодом (Python, Mathcad, Mathematica). Для лучшего понимания отчетности смотрите папку “Пример организации яндекс-папки студентов”.

Пример содержания файла IDZ2.txt:

4.53258

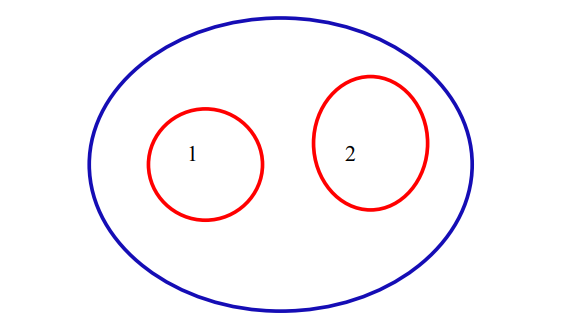


Рисунок 1. Пример электростатической системы

Исходные данные





Основные теоретические положения

Уравне́ние Пуассо́на — [эллиптическое](https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%BB%D0%B8%D0%BF%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B5_%D1%83%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D1%8F" \o "Эллиптические уравнения) [дифференциальное уравнение в частных производных](https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D1%84%D1%84%D0%B5%D1%80%D0%B5%D0%BD%D1%86%D0%B8%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D0%B5_%D1%83%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5_%D0%B2_%D1%87%D0%B0%D1%81%D1%82%D0%BD%D1%8B%D1%85_%D0%BF%D1%80%D0%BE%D0%B8%D0%B7%D0%B2%D0%BE%D0%B4%D0%BD%D1%8B%D1%85" \o "Дифференциальное уравнение в частных производных), которое описывает

* [электростатическое поле](https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D1%81%D1%82%D0%B0%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%BE%D0%B5_%D0%BF%D0%BE%D0%BB%D0%B5" \o "Электростатическое поле),
* [гравитационное поле](https://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D0%B2%D0%B8%D1%82%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D0%BE%D0%BB%D0%B5" \o "Гравитационное поле),
* стационарное поле температуры,
* поле давления,
* поле потенциала скорости в гидродинамике.

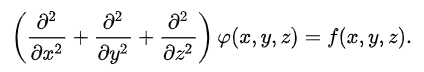
Оно названо в честь [французского](https://ru.wikipedia.org/wiki/%D0%A4%D1%80%D0%B0%D0%BD%D1%86%D0%B8%D1%8F" \o "Франция) [физика](https://ru.wikipedia.org/wiki/%D0%A4%D0%B8%D0%B7%D0%B8%D0%BA" \o "Физик) и [математика](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA" \o "Математик) [Симеона Дени Пуассона](https://ru.wikipedia.org/wiki/%D0%9F%D1%83%D0%B0%D1%81%D1%81%D0%BE%D0%BD,_%D0%A1%D0%B8%D0%BC%D0%B5%D0%BE%D0%BD_%D0%94%D0%B5%D0%BD%D0%B8" \o "Пуассон, Симеон Дени).

Это уравнение имеет вид:

{\displaystyle \Delta \varphi =f,} 

где {\displaystyle \varphi } — искомая функция, {\displaystyle \Delta } — [оператор Лапласа](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%82%D0%BE%D1%80_%D0%9B%D0%B0%D0%BF%D0%BB%D0%B0%D1%81%D0%B0" \o "Оператор Лапласа), или [лапласиан](https://ru.wikipedia.org/wiki/%D0%9B%D0%B0%D0%BF%D0%BB%D0%B0%D1%81%D0%B8%D0%B0%D0%BD" \o "Лапласиан), а {\displaystyle f} —заданная [вещественная](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D1%89%D0%B5%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%BD%D0%BE%D0%B5_%D1%87%D0%B8%D1%81%D0%BB%D0%BE" \o "Вещественное число) или [комплексная](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D0%BB%D0%B5%D0%BA%D1%81%D0%BD%D0%BE%D0%B5_%D1%87%D0%B8%D1%81%D0%BB%D0%BE" \o "Комплексное число) [функция](https://ru.wikipedia.org/wiki/%D0%A4%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D1%8F_(%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)" \o "Функция (математика)) на некотором [многообразии](https://ru.wikipedia.org/wiki/%D0%9C%D0%BD%D0%BE%D0%B3%D0%BE%D0%BE%D0%B1%D1%80%D0%B0%D0%B7%D0%B8%D0%B5" \o "Многообразие).

В трёхмерной [декартовой системе координат](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D1%8F%D0%BC%D0%BE%D1%83%D0%B3%D0%BE%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D0%BA%D0%BE%D0%BE%D1%80%D0%B4%D0%B8%D0%BD%D0%B0%D1%82" \o "Прямоугольная система координат) уравнение принимает форму:

{\displaystyle \left({\frac {\partial ^{2}}{\partial x^{2}}}+{\frac {\partial ^{2}}{\partial y^{2}}}+{\frac {\partial ^{2}}{\partial z^{2}}}\right)\varphi (x,y,z)=f(x,y,z).} 

В [декартовой системе координат](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D1%8F%D0%BC%D0%BE%D1%83%D0%B3%D0%BE%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D0%BA%D0%BE%D0%BE%D1%80%D0%B4%D0%B8%D0%BD%D0%B0%D1%82" \o "Прямоугольная система координат) оператор Лапласа записывается в форме {\displaystyle \Delta } {\displaystyle \nabla ^{2}} ({\displaystyle \nabla } — [оператор набла](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%82%D0%BE%D1%80_%D0%BD%D0%B0%D0%B1%D0%BB%D0%B0" \o "Оператор набла)) и уравнение Пуассона принимает вид:

{\displaystyle {\nabla }^{2}\varphi =f.}

Уравнение Пуассона с  {\displaystyle f\equiv 0} называется [уравнением Лапласа](https://ru.wikipedia.org/wiki/%D0%A3%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5_%D0%9B%D0%B0%D0%BF%D0%BB%D0%B0%D1%81%D0%B0" \o "Уравнение Лапласа):

{\displaystyle \Delta \varphi =0.}

Уравнение Пуассона может быть решено с использованием [функции Грина](https://ru.wikipedia.org/wiki/%D0%A4%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D1%8F_%D0%93%D1%80%D0%B8%D0%BD%D0%B0" \o "Функция Грина); см., например, статью [экранированное уравнение Пуассона](https://ru.wikipedia.org/wiki/%D0%AD%D0%BA%D1%80%D0%B0%D0%BD%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D1%83%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5_%D0%9F%D1%83%D0%B0%D1%81%D1%81%D0%BE%D0%BD%D0%B0" \o "Экранированное уравнение Пуассона). Есть различные методы для получения численных решений.

Значимость уравнения Пуассона для проблем электростатики заключается в том, что с его помощью решение может быть найдено практически всегда, а с помощью теоремы Гаусса только в исключительных случаях.

**ПРИЛОЖЕНИЕ А**

**ФАЙЛ IDZ2.PY**

import matplotlib.pyplot as plt

import numpy as np

from scipy.spatial import Delaunay

from matplotlib.tri import Triangulation

from skimage.measure import find\_contours

N = 400

R = 5

h = 2\*R/N

mesh = []

for i in range(N):

mesh.append([None]\*N)

x\_arr = []

y\_arr = []

a1 = (0.7/0.8)\*\*(1/2)

b1 = 0.7\*\*(1/2)

a2 = (0.5/0.3)\*\*(1/2.5)

b2 = 0.5\*\*(1/2.5)

v0 = 1

v1 = 5

v2 = -5

def f(x, y):

x = x\*h - 5

y = y\*h - 5

if 0.8\*abs(1.5 + x)\*\*2 + abs(-1.5 + y)\*\*2 > 0.7 and 0.3\*abs(-1.5 + x)\*\*2.5 + abs(1.5 + y)\*\*2.5 > 0.5 and x\*\*2 + y\*\*2 < 25:

return True

return False

def cathode(i, j):

if ((i-N/2)\*h)\*\*2 + ((j-N/2)\*h)\*\*2 > 24.7:

return False

if mesh[i+1][j] is None or mesh[i][j+1] is None or mesh[i-1][j] is None or mesh[i][j-1] is None:

return True

return False

def rho(x0, y0, x1, y1):

return ((x1-x0)\*\*2 + (y1-y0)\*\*2)\*\*(1/2)

def main():

data = [[], [], []]

for i in range(N):

for j in range(N):

if f(i, j):

mesh[i][j] = 0

for i in range(N):

for j in range(N):

if mesh[i][j] is not None and cathode(i, j):

if i > j:

mesh[i][j] = -5

else:

mesh[i][j] = 5

for k in range(400):

for i in range(N-1):

for j in range(N-1):

if mesh[i][j] is not None and mesh[i][j+1] is not None and not cathode(i, j+1):

mesh[i][j+1] = (mesh[i][j+1] + mesh[i][j])/2

if mesh[i][j] is not None and mesh[i+1][j] is not None and not cathode(i+1, j):

mesh[i+1][j] = (mesh[i+1][j] + mesh[i][j])/2

for i in range(N-1, 0, -1):

for j in range(N-1, 0, -1):

if mesh[i][j] is not None and mesh[i][j-1] is not None and not cathode(i, j-1):

mesh[i][j-1] = (mesh[i][j-1] + mesh[i][j])/2

if mesh[i][j] is not None and mesh[i-1][j] is not None and not cathode(i-1, j):

mesh[i-1][j] = (mesh[i-1][j] + mesh[i][j])/2

for i in range(N):

for j in range(N):

if mesh[i][j] is not None:

data[0].append(i\*h)

data[1].append(j\*h)

data[2].append(mesh[i][j])

#fig = plt.figure()

#ax = fig.add\_subplot(111, projection="3d")

#ax.plot\_trisurf(data[0], data[1], data[2], cmap="viridis", linewidth=0, alpha=1)

intersection\_points = []

for i in range(len(data[0]) - 1):

x1, y1, z1 = data[0][i], data[1][i], data[2][i]

x2, y2, z2 = data[0][i + 1], data[1][i + 1], data[2][i + 1]

if (z1 < 1 and z2 > 1) or (z1 > 1 and z2 < 1):

t = abs(z1 - 1) / (abs(z1 - 1) + abs(z2 - 1))

intersection\_x = x1 + t \* (x2 - x1)

intersection\_y = y1 + t \* (y2 - y1)

intersection\_points.append((intersection\_x, intersection\_y, 1))

intersection\_points = np.array(intersection\_points).T

#ax.plot(intersection\_points[0], intersection\_points[1], intersection\_points[2])

build\_points = [[], [], []]

for i in range(len(intersection\_points[0])):

build\_points[0].append(intersection\_points[0][i])

build\_points[1].append(intersection\_points[1][i])

build\_points[2].append(intersection\_points[2][i])

first = []

last = []

for i in range(len(intersection\_points[0])//2):

first.append(build\_points[0][2\*i+1])

last.insert(0, build\_points[0][2\*i])

build\_points[0] = []

build\_points[0].extend(first)

build\_points[0].extend(last)

first = []

last = []

for i in range(len(intersection\_points[1]) // 2):

first.append(build\_points[1][2 \* i + 1])

last.insert(0, build\_points[1][2 \* i])

build\_points[1] = []

build\_points[1].extend(first)

build\_points[1].extend(last)

length = 0

for i in range(len(build\_points[0])-1):

length += rho(build\_points[0][i], build\_points[1][i], build\_points[0][i+1], build\_points[1][i+1])

length += rho(build\_points[0][0], build\_points[1][0], build\_points[0][-1], build\_points[1][-1])

print(length)

#plt.show()

if \_\_name\_\_ == "\_\_main\_\_":

main()