Санкт-Петербургский Государственный Электротехнический Университет "ЛЭТИ"

кафедра физики

Задание №2 по дисциплине

"Физические основы информационных технологий"

Название: Численное решение уравнения Лапласа

| Фамилия И.О.: | Чубан Д.В. |
| --- | --- |
| 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

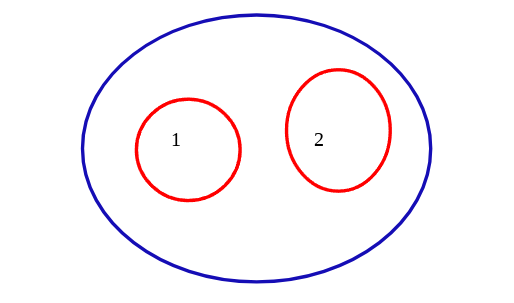


Рисунок 1. Пример электростатической системы

Исходные данные

| Вар. | Уравнение внешнего электрода | Уравнение электрода 1 | Уравнение электрода 2 | Потенциал искомой эквипотенциали, В | Потенциал на электроде 1, В | Потенциал на электроде 2, В |
| --- | --- | --- | --- | --- | --- | --- |
| 16 | x^2 + y^2 = 25 | 0.5\*Abs[-1.8 + x]^3 + 0.8\*Abs[-1.8 + y]^3 = 0.8 | 0.3\*Abs[1.8 + x]^2 + 0.8\*Abs[1.8 + y]^2 = 0.8 | -3 | -6 | 6 |

Выполнение работы

Из области, ограниченной внешним электродом вычтем области, ограниченные внутренними электродами.

В результате получим следующую область:

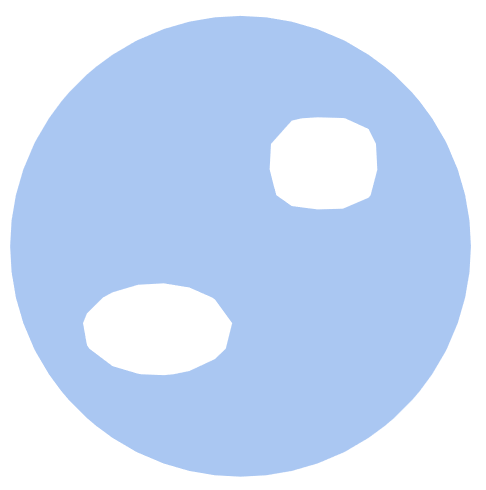


Рисунок 2. Область решения диф.уравнения

Зададим граничные условия Дирихле:

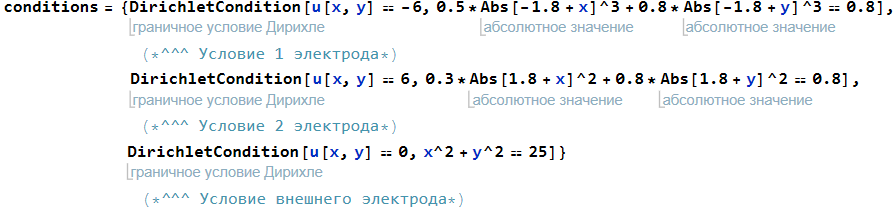


Рисунок 3. Условия Дирихле

Численно решим уравнение Лапласа с заданными условиями и построим график решения уравнения.

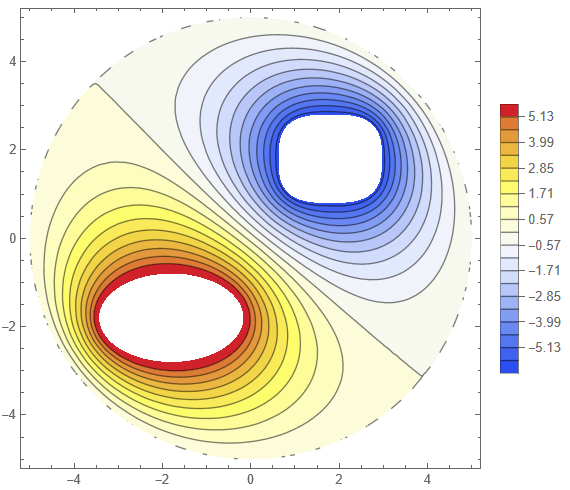


Рисунок 4. График решения уравнения Лапласа

Цветами обозначены потенциалы, выделенные линии – эквипотенциали.

Найдем эквипотенциаль с потенциалом -3В и построим ее на графике (выделена зеленым цветом):

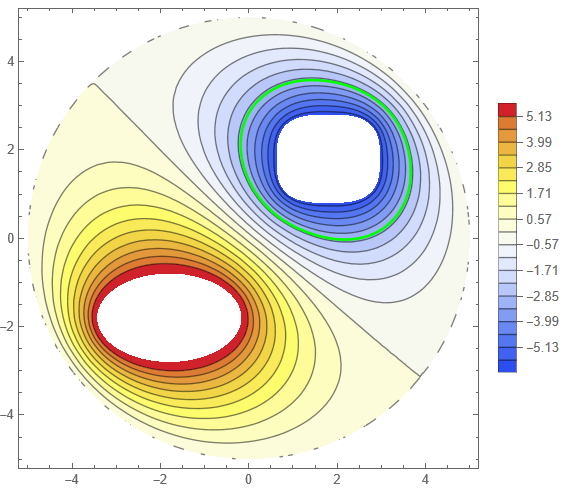


Рисунок 5. Эквипотенциаль с потенциалом -3В

Для того, чтобы приближенно подсчитать длину эквипотенциали, извлечем точки по которым построен график в массив и последовательно посчитаем евклидово расстояние между ними. Сумма данных расстояний – приближенное искомое значение.

Вывод

С помощью Wolfram Mathematica было решено уравнение Лапласа, построены эквипотенциали для электростатической системы из 3 электродов, найдена эквипотенциаль с заданным потенциалом и посчитана ее длина.

**Приложение A**

**Программа IDZ2.nb**

outerElectrode = x^2 + y^2 <= 25;

electrode1 = 0.5\*Abs[-1.8 + x]^3 + 0.8\*Abs[-1.8 + y]^3 <= 0.8;

electrode2 = 0.3\*Abs[1.8 + x]^2 + 0.8\*Abs[1.8 + y]^2 <= 0.8;

outerRegion = ImplicitRegion[outerElectrode, {x, y}];

innerRegion1 = ImplicitRegion[electrode1, {x, y}];

innerRegion2 = ImplicitRegion[electrode2, {x, y}];

fullRegion =

RegionDifference[RegionDifference[outerRegion, innerRegion1],

innerRegion2];

Region[fullRegion]

laplEquation = Laplacian[u[x, y], {x, y}] == 0;

conditions = {DirichletCondition[u[x, y] == -6,

0.5\*Abs[-1.8 + x]^3 + 0.8\*Abs[-1.8 + y]^3 == 0.8],

(\*^^^ Условие 1 электрода\*)

DirichletCondition[u[x, y] == 6,

0.3\*Abs[1.8 + x]^2 + 0.8\*Abs[1.8 + y]^2 == 0.8],

(\*^^^ Условие 2 электрода\*)

DirichletCondition[u[x, y] == 0, x^2 + y^2 == 25]}

(\*^^^ Условие внешнего электрода\*)

eqSol = NDSolve[{laplEquation, conditions},

u, {x, y} \[Element] fullRegion]

eqSolPlot =

ContourPlot[u[x, y] /. First[eqSol], {x, y} \[Element] fullRegion,

Contours -> 20, ColorFunction -> "TemperatureMap",

PlotLegends -> Automatic]

DensityPlot[u[x, y] /. First[eqSol], {x, y} \[Element] fullRegion,

ColorFunction -> "TemperatureMap", PlotLegends -> Automatic]

contourPlot =

ContourPlot[

Evaluate[u[x, y] /. eqSol] == -3, {x, y} \[Element] fullRegion,

PlotLegends -> Automatic, ContourStyle -> Green];

Show[eqSolPlot, contourPlot]

points = Cases[Normal@contourPlot, Line[pts\_] :> pts, Infinity];

pointPairs = Flatten[points, 1];

totalDistance = 0;

For[i = 1, i <= Length[pointPairs] - 1, i++,

totalDistance +=

EuclideanDistance[pointPairs[[i]], pointPairs[[i + 1]]]]

totalDistance