Санкт-Петербургский Государственный Электротехнический Университет "ЛЭТИ"

кафедра физики

Задание №2 по дисциплине

"Физические основы информационных технологий"

Название: Численное решение уравнения Лапласа.

|  |  |
| --- | --- |
| Фамилия И.О.: | Самохин К.А. |
| группа: | 1303 |
| Преподаватель: | Альтмарк А.М. |
| Итоговый балл: |  |
| Крайний срок сдачи: | 5.11.23 |

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

2

1

Рисунок 1. Пример электростатической системы

Исходные данные

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Вар. | Уравнение внешнего электрода | Уравнения электрода 1 | Уравнения электрода 2 | Потенциал искомой эквипотенциали, В | Потенциал на электроде 1, В | Потенциал на электроде 2,В |
| 10 | x^2 + y^2 = 25 | Abs[-1.5 + x]^3.5 + 0.5 Abs[y]^3.5 = 0.6 | 0.8 Abs[1.5 + x]^4 + 0.5 Abs[y]^4 = 0.6 | 2 | 5 | -5 |

Выполнение работы

Прежде всего, введём уравнения кривых, задающих границы внешнего и внутренних электродов. Получим область, на которой будем решать дифференциальное уравнение. Для этого из области, ограниченной внешним электродом, вычтем области, ограниченные внутренними электродами.

Полученная область выглядит следующим образом:

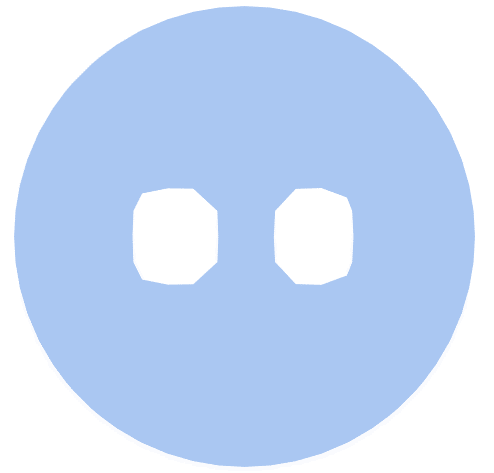


Рисунок 2. Область, на которой будет проводиться решение дифференциального уравнения.

Далее необходимо задать граничные условия Дирихле:

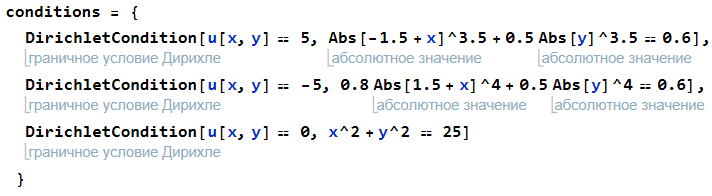


Рисунок 3. Условия Дирихле.

Используя функцию NDSolve и заданные условия, численно решим уравнение Лапласа и построим график решения уравнения в заданной ранее области.

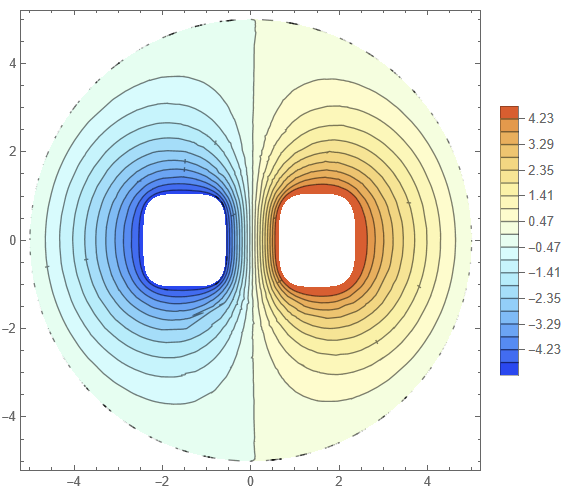


Рисунок 4. График решения уравнения Лапласа.

Найдём эквипотенциаль с потенциалом 2 В и построим её на графике:

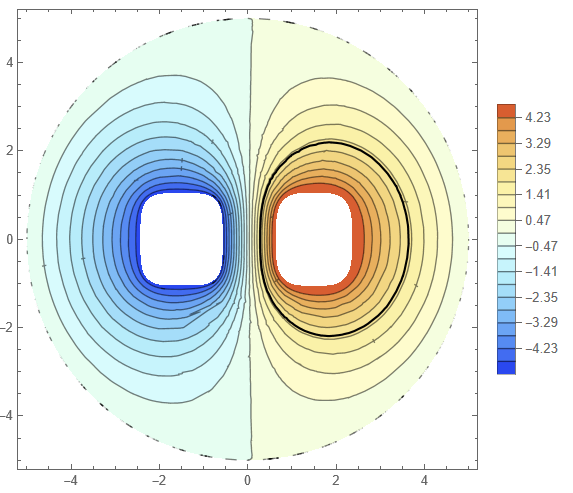


Рисунок 5. Искомая эквипотенциаль.

Для того, чтобы вычислить длину эквипотенциали, извлечём точки, по которым построен график и посчитаем эвклидово расстояние между соседними. Сумма всех этих расстояний и будет искомой длиной.

**Приложение А.**

Название файла: IDZ2.nb

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

innerEl1 = Abs[-1.5 + x]^3.5 + 0.5 Abs[y]^3.5 <= 0.6;

innerEl2 = 0.8 Abs[1.5 + x]^4 + 0.5 Abs[y]^4 <= 0.6;

outerArea = ImplicitRegion[outerEl, {x, y}];

innerArea1 = ImplicitRegion[innerEl1, {x, y}];

innerArea2 = ImplicitRegion[innerEl2, {x, y}];

fullArea = RegionDifference[RegionDifference[outerArea, innerArea1], innerArea2];

Region[fullArea]

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

conditions = {

DirichletCondition[u[x, y] == 5,

Abs[-1.5 + x]^3.5 + 0.5 Abs[y]^3.5 == 0.6],

DirichletCondition[u[x, y] == -5,

0.8 Abs[1.5 + x]^4 + 0.5 Abs[y]^4 == 0.6],

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

}

res = NDSolve[{laplaceEquation, conditions}, u, {x, y} \[Element] fullArea]

resPlt = ContourPlot[u[x, y] /. First[res], {x, y} \[Element] fullArea, Contours -> 20, ColorFunction -> "LightTemperatureMap", PlotLegends -> Automatic]

DensityPlot[u[x, y] /. First[res], {x, y} \[Element] fullArea, ColorFunction -> "LightTemperatureMap", PlotLegends -> Automatic]

contourPlt = ContourPlot[Evaluate[u[x, y] /. res] == 2, {x, y} \[Element] fullArea, Contours -> {1}, PlotLegends -> Automatic, ContourStyle -> Black];

Show[resPlt, contourPlt]

points = Cases[Normal@contourPlt, 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