Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И

ОПТИКИ

Факультет систем управления и робототехники



**Отчет по проекту №1 «Моделирование силовых линий и сечений**

**эквипотенциальных поверхностей»**

**по дисциплине «Физика с элементами компьютерного моделирования»**

Выполнил: студент гр. **R32362**

Михин Н. С.

Преподаватель: Хвастунов Н. Н.,

Санкт-Петербург 2022

1. **Цели работы:**

Моделирование силовых линий и сечений эквипотенциальных поверхностей электрического поля для нескольких (четырех и более) точечных зарядов, одинаковых по модулю, находящихся друг от друга на одинаковых расстояниях.

1. **Теоретическая часть**

Силовые линии (линии напряженности) – это линии, касательные к которым в каждой точке совпадают с направлением вектора напряженности в этой точке.

В системе точечных зарядов справедлив принцип суперпозиции для вектора напряженности и для потенциала системы.

Напряженность электростатического поля системы равна векторной сумме напряженностей полей, которые создавал бы каждый из этих зарядов в отсутствии остальных

Потенциал электрического поля системы точечных зарядов равен алгебраической сумме потенциалов электрических полей, созданных каждым зарядом в отдельности.

Из определения силовой линии следует, что касательная к ней в любой точке составляет с осью абсцисс угол

Так как в нашем случае мы имеем дело с кривой на плоскости, то выражение примет следующий вид

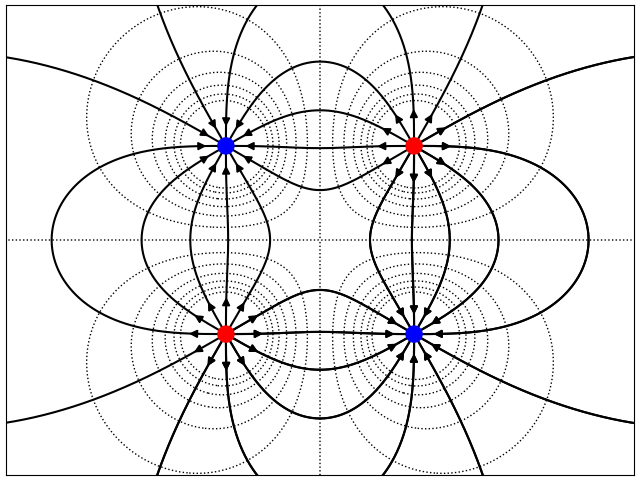
Получаем следующее дифференциальное уравнение:

Получается следующее аналитическое решение:

1. **Ход моделирования**

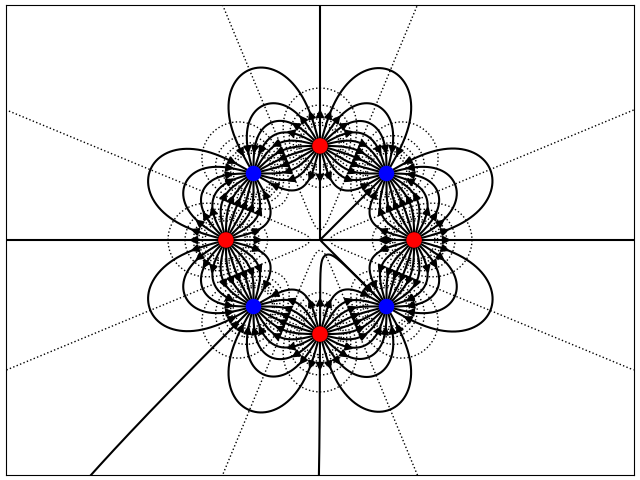
Моделирование проводилось с помощью средств языка программирования Python. Разобьём моделирование на несколько структур точечных зарядов.

1. **Квадруполь с одноименными зарядами по диагонали**

****

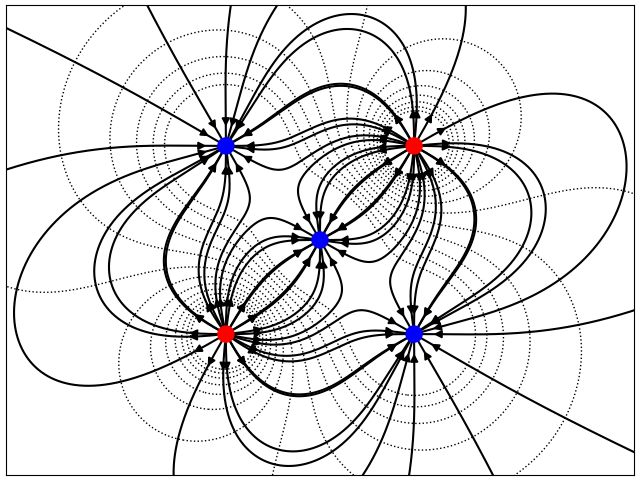
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | 1 | 2 | 3 | 4 |
| Заряд (Кл) | **-2** | **2** | **-2** | **2** |
| Координата | **(-2, 2)** | **(2, 2)** | **(2, -2)** | **(-2, -2)** |

1. **Окружность с чередующимися на ней точками зарядов**

****

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Заряд (Кл) | **-1** | **1** | **-1** | **1** | **-1** | **1** | **-1** | **1** |
| Координата  (x, y) | **(-2, 0)** | **(-1.414, 1.414)** | **(0, 2)** | **(1.414, 1.414)** | **(2, 0)** | **(1.414, -1.414)** | **(0, -2)** | **(-1.414, -1.414)** |

1. **Квадруполь с центральным отрицательным зарядом**

****

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | 1 | 2 | 3 | 4 | 5 |
| Заряд (Кл) | **-2** | **2** | **-2** | **2** | **-2** |
| Координата | **(-2, 2)** | **(2, 2)** | **(2, -2)** | **(-2, -2)** | **(0, 0)** |

1. **Квадруполь с центральным положительным зарядом**

**Изображение выглядит как аксессуар, векторная графика

Автоматически созданное описание**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | 1 | 2 | 3 | 4 | 5 |
| Заряд (Кл) | **-2** | **2** | **-2** | **2** | **-2** |
| Координата | **(-2, 2)** | **(2, 2)** | **(2, -2)** | **(-2, -2)** | **(0, 0)** |

1. **Исходный код программы**

**from** **matplotlib** **import** pyplot

**from** **numpy** **import** radians

**import** **electrostatics**

**from** **electrostatics** **import** PointCharge, ElectricField, Potential, GaussianCircle

**from** **electrostatics** **import** finalize\_plot

**def** **show\_electrostatic\_field**(type\_of\_struct, charge):

XMIN, XMAX = -**40**, **40**

YMIN, YMAX = -**30**, **30**

ZOOM = **5**

XOFFSET = **0**

electrostatics.init(XMIN, XMAX, YMIN, YMAX, ZOOM, XOFFSET)

**if** type\_of\_struct == **1**:

charges = [PointCharge(**2**, [**2**, **2**]),

PointCharge(**2**, [-**2**, -**2**]),

PointCharge(-**2**, [**2**, -**2**]),

PointCharge(-**2**, [-**2**, **2**])]

**elif** type\_of\_struct == **2**:

count\_of\_points = **8**

coordinates = [[-**2**, **0**], [-**1.414**, **1.414**], [**0**, **2**], [**1.414**, **1.414**], [**2**, **0**],

[**1.414**, -**1.414**], [**0**, -**2**], [-**1.414**, -**1.414**]]

charges = [PointCharge(charge \* (-**1**) \*\* i, coordinates[i]) **for** i **in** range(count\_of\_points)]

**elif** type\_of\_struct == **3**:

charges = [PointCharge(**2**, [**2**, **2**]),

PointCharge(**2**, [-**2**, -**2**]),

PointCharge(-**2**, [**2**, -**2**]),

PointCharge(-**2**, [-**2**, **2**]),

PointCharge(**2**, [**0**, **0**])]

**elif** type\_of\_struct == **4**:

charges = [PointCharge(**2**, [**2**, **2**]),

PointCharge(**2**, [-**2**, -**2**]),

PointCharge(-**2**, [**2**, -**2**]),

PointCharge(-**2**, [-**2**, **2**]),

PointCharge(-**2**, [**0**, **0**])]

**else**:

count\_of\_points = **0**

coordinates = []

charges = [PointCharge(charge \* (-**1**) \*\* i, coordinates[i]) **for** i **in** range(count\_of\_points)]

field = ElectricField(charges)

potential = Potential(charges)

gaussian\_list = [GaussianCircle(charges[i].x, **0.1**) **for** i **in** range(len(charges))]

gaussian\_list[**2**].a0 = radians(**90**)

gaussian\_list[**3**].a0 = radians(-**90**)

fieldlines = []

**for** g\_ **in** gaussian\_list[**0**:len(gaussian\_list)]:

**for** x **in** g\_.fluxpoints(field, **12**):

fieldlines.append(field.line(x))

**for** fieldline **in** fieldlines:

fieldline.plot()

**for** charge **in** charges:

charge.plot()

potential.plot()

finalize\_plot()

pyplot.show()

**def** **main**():

show\_electrostatic\_field(**2**, **2**)

**if** \_\_name\_\_ == '\_\_main\_\_':

main()