УДК 537.8

Журавель А.И.
студент
3 курс, инженерно-физический факультет
Адыгейский государственный университет
Россия, г. Майкоп

КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ СИЛОВЫХ ЛИНИЙ ЭЛЕКТРИЧЕСКОГО ПОЛЯ СТАЦИОНАРНЫХ ПРОИЗВОЛЬНО РАСПОЛОЖЕННЫХ ЗАРЯДОВ НА ПЛОСКОСТИ

Аннотация: *В работе представлено исследование электрических полей, создаваемых системой стационарных зарядов, с использованием методов компьютерного моделирования. Основной целью является разработка программного кода на языке Python для визуализации силовых линий электрического поля и эквипотенциальных поверхностей системы двух точечных зарядов. Применены численные методы расчета напряженности поля и потенциала на основе принципа суперпозиции. Реализация включает построение графиков распределения напряженности* $E_z(z)$ *и потенциала* $\phi(z)$ *вдоль оси системы зарядов с использованием библиотек NumPy и Matplotlib. Результаты демонстрируют эффективность Python для решения задач электростатики и предоставляют наглядные инструменты для анализа конфигураций электрических полей.*

Ключевые слова: *электрическое поле, точечные заряды, электростатика, компьютерное моделирование, Python, NumPy, Matplotlib, визуализация данных, напряженность поля, эквипотенциальные линии.*

Abstract: *The study investigates electric fields generated by a system of stationary charges using computational modeling methods. The main objective is to develop Python code for visualizing electric field lines and equipotential surfaces of a two-point charge system. Numerical methods for calculating field strength and potential based on the superposition principle are applied. The implementation includes plotting Ez(z) and ϕ(z) distributions along the axis of the charge system using NumPy and Matplotlib libraries. The results demonstrate Python's efficiency for solving electrostatic problems and provide intuitive tools for analyzing electric field configurations.*

Keywords: *electric field, point charges, electrostatics, computer modeling, Python, NumPy, Matplotlib, data visualization, field strength, equipotential lines.*

Введение и постановка задачи

Изучение электрических полей, создаваемых системой стационарных зарядов, играет основную роль в электродинамике, поскольку позволяет понять принципы взаимодействия заряженных частиц и особенности распределения электромагнитной энергии в пространстве.

В данной работе главной целью является написание программного кода на языке Python, способного визуализировать силовые линии электрического поля стационарных произвольно расположенных зарядов на плоскости.

Построение эквипотенциальных линий электростатического поля двух зарядов

Задача. Два равных точечных заряда $q_1 = q_2 = q$ находятся в точках с координатами $(0,0,-a)$ и $(0,0,+a)$. Необходимо построить графики зависимостей $E_z(z)$ и $\varphi(z)$ для точек оси и $E_x(x), \varphi(x)$ для точек $x(z=0)$.

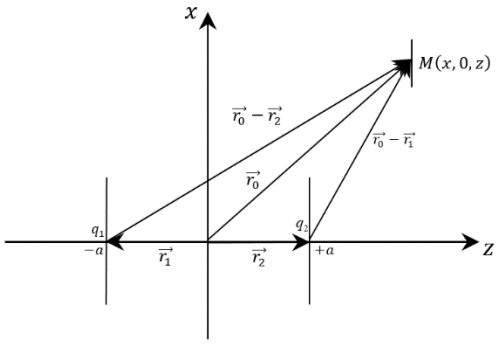

Рис.1. Два равных точечных заряда $q_1 = q_2 = q$ находящиеся в точках с координатами $(0,0,-a)$ и $(0,0,+a)$.

Решение. Расположим оси координат согласно рисунку 1. Выберем в плоскости xoz произвольную точку $M(x,0,z)$ и найдем в ней потенциал результирующего поля зарядов q_1 и q_2 по принципу суперпозиции, применяя соотношение (2):

$$\varphi = kq\left(\frac{1}{\left[(z+a)^2 + x^2\right]^{1/2}} + \frac{1}{\left[(z-a)^2 + x^2\right]^{1/2}}\right) \tag{5}$$

Проекции вектора напряженности на оси координат определим с помощью соотношения (1):

$$E_z = grad_z\varphi = -\frac{\partial\varphi}{\partial z};\ E_x = grad_x\varphi = -\frac{\partial\varphi}{\partial x},$$ то есть

$$E_z = kq\left(\frac{(z+a)}{\left[(z+a)^2 + x^2\right]^{3/2}} + \frac{(z-a)}{\left[(z-a)^2 + x^2\right]^{3/2}}\right),$$

$$E_x = kqx\left(\frac{1}{\left[(z+a)^2 + x^2\right]^{3/2}} + \frac{1}{\left[(z-a)^2 + x^2\right]^{3/2}}\right). \tag{6}$$

Выражения (5) и (6) задают значения потенциала φ и напряженности $\vec{E}$ в любой точке плоскости $x0z$.

В точках оси $z(x=0)$ выражения (5) и (6) имеют вид:

$$E_z(z) = kq\left(\frac{z+a}{|z+a|^3} + \frac{z-a}{|z+a|^3}\right), E_x = 0,$$
$$\varphi(z) = kq\left(\frac{1}{|z+a|} + \frac{1}{|z-a|}\right). \quad (7)$$

Попробуем показать графики $E_z(z)$ и $\varphi(z)$ с помощью Python. Для того, чтобы показать эти графики с помощью Python необходимо написать следующую программу:

```
import numpy as np
import matplotlib.pyplot as plt
# Параметры
k = 1   # Электростатическая постоянная
q = 1   # Величина заряда
a = 1   # Расстояние до зарядов
# Создаем массив значений z с повышенной плотностью точек возле зарядов
z_left = np.linspace(-100*a, -a*1.1001, 2000)
z_middle = np.linspace(-a*0.9999, a*0.9999, 2000)
z_right = np.linspace(a*1.1001, 100*a, 2000)
# Функция расчета Ez с обработкой особых точек
def calculate_ez(z, k, q, a):
 eps = 1e-10
term1 = (z + a + eps) / (np.abs(z + a + eps))**3
term2 = (z - a - eps) / (np.abs(z - a - eps))**3
return k * q * (term1 + term2)
Ez_left = calculate_ez(z_left, k, q, a)
Ez_middle = calculate_ez(z_middle, k, q, a)
Ez_right = calculate_ez(z_right, k, q, a)
# Создаем график с улучшенным качеством
plt.figure(figsize=(12, 6), dpi=100)
```

```
ax = plt.gca()
# Рисуем кривые с антиалиасингом (убраны несуществующие параметры)
plt.plot(z_left, Ez_left, 'b-', linewidth=2.5, antialiased=True)
plt.plot(z_middle, Ez_middle, 'b-', linewidth=2.5, antialiased=True)
plt.plot(z_right, Ez_right, 'b-', linewidth=2.5, antialiased=True)
# Настраиваем оси
ax.spines['left'].set_position('zero')
ax.spines['right'].set_color('none')
ax.spines['bottom'].set_position('zero')
ax.spines['top'].set_color('none')
# Убираем численные подписи
ax.set_xticks([])
ax.set_yticks([])
# Подписываем оси
ax.text(5*a*1.05, 0, 'z', fontsize=14, ha='center')
ax.text(0.51, 30.52, '$E_z$', transform=ax.get_yaxis_transform(),
        fontsize=14, ha='left', va='bottom', )
# Стрелочки
ax.plot(1, 0, ">k", transform=ax.get_yaxis_transform(), clip_on=False,
markersize=10)
ax.plot(0, 1, "^k", transform=ax.get_xaxis_transform(), clip_on=False,
markersize=10)
# Метки
label_offset = max(abs(Ez_left)) * 0.2
ax.annotate('$-a$', xy=(-a, 0), xytext=(-a, -label_offset),
            ha='center', fontsize=14, arrowprops=dict(arrowstyle='->', linewidth=1.5))
ax.annotate('$+a$', xy=(a, 0), xytext=(a, -label_offset),
            ha='center', fontsize=14, arrowprops=dict(arrowstyle='->', linewidth=1.5))
# Вертикальные линии
```

```
ax.axvline(x=-a, color='r', linestyle=':', linewidth=2.0, alpha=0.9)
ax.axvline(x=a, color='r', linestyle=':', linewidth=2.0, alpha=0.9)
# Настройки отображения
ax.set_xlim(-5*a, 5*a)
ax.set_ylim(-max(abs(Ez_left))*1.3, max(abs(Ez_left))*1.3)
plt.tight_layout()
plt.show()
```

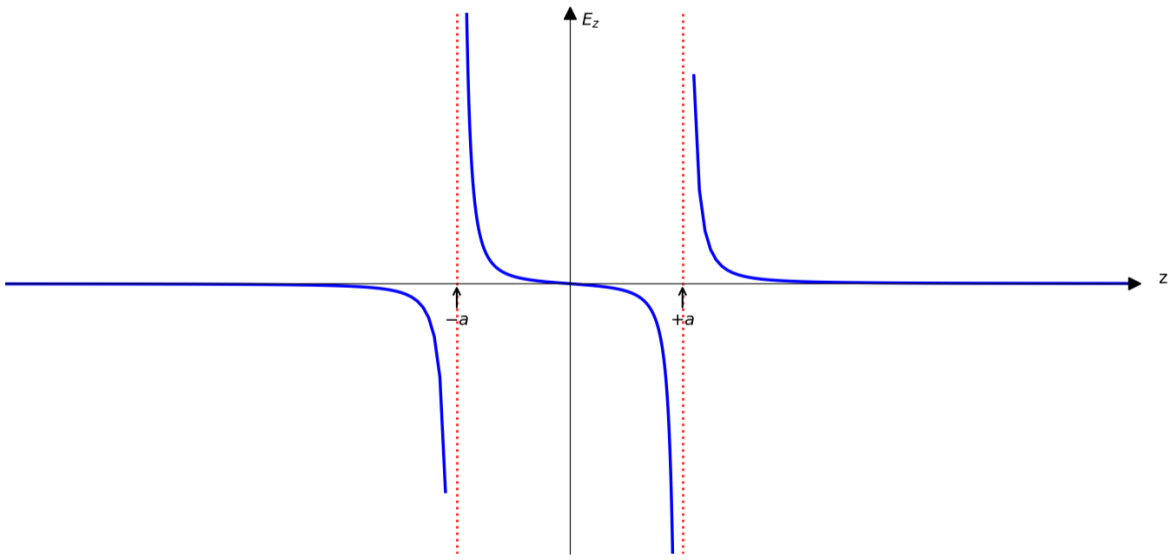

График 1 - $E_z(z)$

```
import numpy as np
import matplotlib.pyplot as plt
# Параметры системы
k = 8.99e9  # Электростатическая постоянная
q = 1e-9    # Величина заряда
a = 0.1     # Расстояние до зарядов
# Создаем массив координат z (исключаем точки на зарядах)
z = np.linspace(-5*a, 5*a, 1000)
z = z[(np.abs(z - a) > 0.01*a) & (np.abs(z + a) > 0.01*a)]
# Функция потенциала
def potential(z, k, q, a):
```

```
    return k * q * (1/np.abs(z + a) + 1/np.abs(z - a))
phi = potential(z, k, q, a)
# Создаем график
fig, ax = plt.subplots(figsize=(10, 6))
# Рисуем потенциал
ax.plot(z, phi, 'b-', linewidth=2)
# Настраиваем оси
ax.spines['left'].set_position('zero')
ax.spines['bottom'].set_position('zero')
ax.spines['right'].set_color('none')
ax.spines['top'].set_color('none')
# Убираем числовые метки
ax.set_xticks([])
ax.set_yticks([])
# Подписи осей (смещены к стрелкам)
ax.text(5*a*1.05, 0, 'z', fontsize=12, ha='center')
ax.text(0, max(phi)*1.05, 'φ(z)', fontsize=12, va='center')
# Стрелки на осях
ax.plot(1, 0, ">k", transform=ax.get_yaxis_transform(), markersize=10,
clip_on=False)
ax.plot(0, 1, "^k", transform=ax.get_xaxis_transform(), markersize=10,
clip_on=False)
# Метки зарядов (без стрелок)
label_offset = max(phi)*0.1
ax.text(-a, -label_offset, '$-a$', ha='center', fontsize=12)
ax.text(a, -label_offset, '$a$', ha='center', fontsize=12)
# Настройки отображения
ax.set_xlim(-5*a, 5*a)
ax.set_ylim(-max(phi)*0.1, max(phi)*1.1)  # Ось φ(z) из центра
```

```
plt.tight_layout()
plt.show()
```

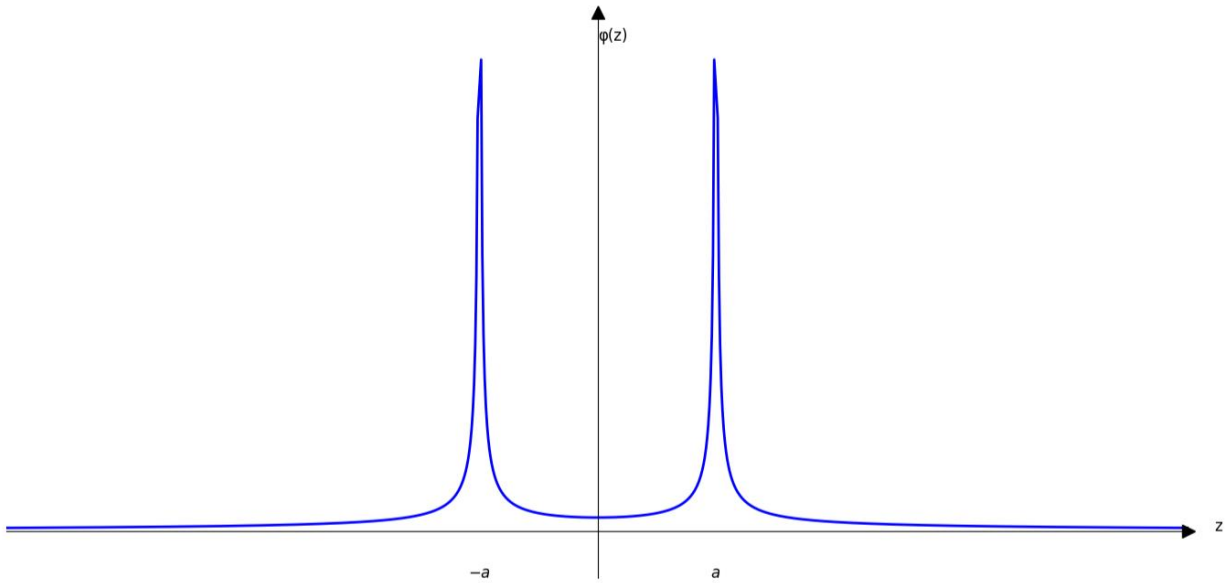

График 2 - $\varphi(z)$

Мы показали, что графики $E_z(z)$ и $\varphi(z)$ можно построить с помощью Python.

Заключение

В заключении работы можно подвести небольшие итоги: было выполнено компьютерное моделирование силовых линий электрического поля для системы стационарных точечных зарядов на плоскости с использованием языка программирования Python. Для расчёта напряжённости поля и построения силовых линий применялись численные методы, а также библиотеки NumPy и SciPy для эффективных математических вычислений. Визуализация результатов проводилась с помощью Matplotlib, что позволило получить наглядные графические представления конфигураций поля для различных распределений зарядов. Реализация на Python подтвердила гибкость и удобство этого языка для научных расчётов и моделирования физических процессов.

Список использованной литературы

1. Болсун, А.И. Краткий словарь физических терминов. – Минск: Вышэйшая школа, 1979. — С. 272.
2. Генденштейн, Л.Э., Дик, Ю.И. Физика. 10 класс. В 2 частях. Часть 1. Учебник (базовый уровень). — С. 277.
3. Мякишев, Г.Я., Буховцев, Б.Б., Сотский, Н.Н. Физика. 10 класс : учеб. для общеобразоват. организаций : базовый и углубл. уровни // 6-е изд., перераб. и доп. подред. Парфентьевой, Н. А.— М. : Просвещение, 2019. — С. 249.
4. Жукова, И.Н., Тлячев, В.Б. Об одной задаче электростатики // Труды ФОРА. – 1997. – №2. – С. 131-141.
5. Савельев, И.В. Основы теоретической физики в двух томах. — Механика, Электродинамика (издание второе). — Том 1. — С. 278.