

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение
высшего образования «Крымский федеральный университет им. В.И. Вернадского»

ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ

«Компьютерное построение линий поля»

обязательное задание по дисциплине

«Основы компьютерного моделирования»

студента 2 курса группы ПИ-б-о-201(1)

Шенгелай Всеволода Михайловича

направления подготовки 09.03.04 «Программная инженерия»

Кандидат технических наук,
доцент, заведующий кафедрой
компьютерной инженерии и
моделирования

90

(оценка)

Милюков В.В.



(подпись, дата)

Симферополь, 2022

Цели работы

- Ознакомиться методами моделирования электростатических полей, построить силовые линии поля
- Изучить способы верификации компьютерных моделей на основании физических законов и аналитических моделей;
- Повысить навыки создания графических приложений на Python, изучить интеграцию графиков библиотеки Matplotlib в программу с графическим интерфейсом Tkinter.

Перед выполнением лабораторной работы:

1. Были изучены теоретические сведения в методических указаниях к выполнению данной лабораторной работы; подробно рассмотрены приведенные практические примеры.
2. Прочитан соответствующий материал в электронном конспекте лектора по данному курсу.

Представлена 1 программа в электронной форме, продемонстрирована её работоспособность.

Программа, в которой производились все расчеты, была написана на языке Python 3 в интерактивной среде разработки IDLE (Python 3.6 32-bit). Технические характеристики рабочего компьютера следующие: процессор Intel(R) Core(TM) i3-10110U CPU @ 2.10GHz, 8 ГБ установленной оперативной памяти. Операционная система - Windows 10.

Ход работы

1. Физическая постановка задачи

1.1. Напряжённость электрического поля

Вектор напряженности \vec{E} – это силовая характеристика электрического поля. В некоторой точке поля, напряженность равна силе, с которой поле действует на единичный положительный заряд, размещенный в указанной точке, при этом направление силы и напряженности совпадают. Определение напряженности записывается так:

$$\vec{E} = \frac{\vec{F}}{q} \quad (1)$$

где \vec{F} – сила, с которой электрическое поле действует на неподвижный, «пробный», точечный заряд q , который размещают в рассматриваемой точке поля. При этом считают, что «пробный» заряд мал на столько, что не искажает исследуемого поля.

Если поле является электростатическим, то его напряженность от времени не зависит.

Если электрическое поле является однородным, то его напряженность во всех точках поля одинакова.

Графически электрические поля можно изображать при помощи силовых линий. Силовыми линиями (линиями напряженности) называют линии, касательные к которым в каждой точке совпадают с направлением вектора напряженности в этой точке поля.

1.2 Напряженность поля точечного заряда

Напряженность поля точечного заряда q равна:

$$\vec{E} = \frac{1}{4\pi\epsilon\epsilon_0} \frac{q}{r^3} \vec{r} \quad (2)$$

где $\epsilon_0 = 8,85 \cdot 10^{-12} \text{Ф/м}$ (система СИ) - электрическая постоянная.

1.3 Связь напряженности и потенциала

В общем случае напряженность электрического поля связана с потенциалом как:

$$\vec{E} = -grad\varphi - \frac{\delta \vec{A}}{\delta t} \quad (3)$$

где φ – скалярный потенциал, \vec{A} – векторный потенциал.

Для стационарных полей выражение (3) трансформируется в формулу:

$$\vec{E} = -grad\varphi \quad (4)$$

1.4 Принцип суперпозиции напряженностей электрических полей

Предположим, что мы хотим найти электрическое поле $E(r)$ в точке r , обусловленное точечными зарядами q_1, q_2, \dots, q_N . $E(r)$ удовлетворяет принципу суперпозиции и имеет вид:

$$E(r) = k \sum_i^N \frac{q_i}{|r - r_i|^3} (r - r_i) \quad (5),$$

где r_i – координата неподвижного i -го заряда и k – постоянная, зависящая от выбора системы единиц. В системе СИ постоянная k равна

$$k = \frac{1}{4\pi\epsilon_0} \approx 9,0 * 10^9 \text{Н} * \frac{\text{м}^2}{\text{Кл}^2},$$

где кулон (Кл) есть единица заряда в системе СИ, а ϵ_0 – диэлектрическая постоянная вакуума. С микроскопической точки зрения кулон – очень большая единица заряда. Например, величина заряда электрона равняется $e = 1,6 * 10^{-19}$ Кл.

2. Построение линий поля, математический алгоритм

Поле E является векторным. В каждой точке пространства такое поле характеризуется величиной и направлением. Как наглядно изобразить такое векторное поле? Одни из возможных способов – разбить пространство дискретной сеткой, найти E в каждой точке и начертить из этих точек стрелки

в направлении E . Однако этот способ не дает никакой информации о величине электрического поля. Более хорошим способом наглядного представления векторного поля является изображение *силовых линий электрического поля*. Эти линии обладают следующими свойствами:

1. Каждая силовая линия электрического поля представляет собой направленную линию, касательная к которой в каждой точке параллельна электрическому полю в этой точке.
2. Эти линии гладкие и непрерывные за исключением особых точек, таких как точечные заряды. (Бессмысленно говорить об электрическом поле в *точке* точечного заряда.)
3. Полное число электрических силовых линий, исходящих из точечного заряда, пропорционально величине этого заряда. Коэффициент пропорциональности выбирается из соображений наибольшей ясности изображения поля.

Старт линий можно осуществлять с окружностей, центры которых совпадают с зарядами (очевидно нельзя стартовать с самих зарядов). Величина шага задается заранее и в видимой области не изменяется, при выходе линии в дальнюю зону величина шага может быть увеличена, например, пропорционально расстоянию до центра системы, направление шага всегда совпадает с направлением поля. Линию обрываем в следующих случаях:

- а) точка приблизилась на расстояние ΔL к отрицательному заряду
- б) точка "убежала" на бесконечность
- в) поле E "очень мало"
- г) произошло "зацикливание" – перескакивание с линии на линию около точки $E=0$

Алгоритм построения силовых линий электрического поля в двумерном случае:

1. Выбираем точку (x, y) и вычисляем компоненты E_x и E_y вектора электрического поля E по формуле (5).

2. Проводим в этой точке небольшой прямолинейный отрезок заданной длины Δs в направлении E , Компоненты этого отрезка равны:

$$\Delta x = \Delta s \frac{E_x}{|E|} \quad \text{и} \quad \Delta y = \Delta s \frac{E_y}{|E|}$$

3. Повторяем данную процедуру с новой точки $(x + \Delta x, y + \Delta y)$.
Продолжаем до тех пор, пока силовая линия не уйдет в бесконечность или не подойдет к какому-нибудь отрицательному заряду.

На рис.1 представлено компьютерное изображение поля двух разноименных зарядов, построенное намеренно грубо, шаг $\Delta L = 0.1$, (расстояние между зарядами выбрано за единицу), число линий, стартующих с положительного заряда, выбрано равным 20. Мы можем наблюдать следующий дефект: “выгрызен” сектор слева от отрицательного заряда за счет того, что граница дальней зоны взята слишком близко.

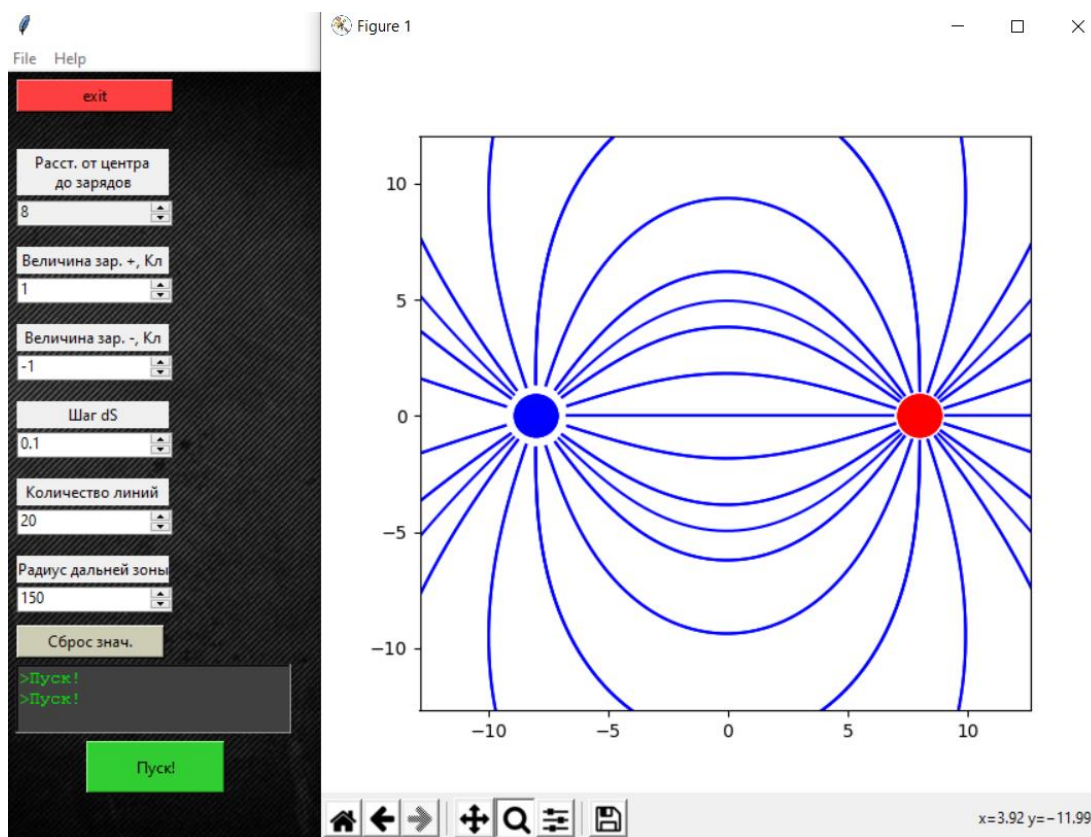


Рисунок 1. Точки старта и обрыва линий поля

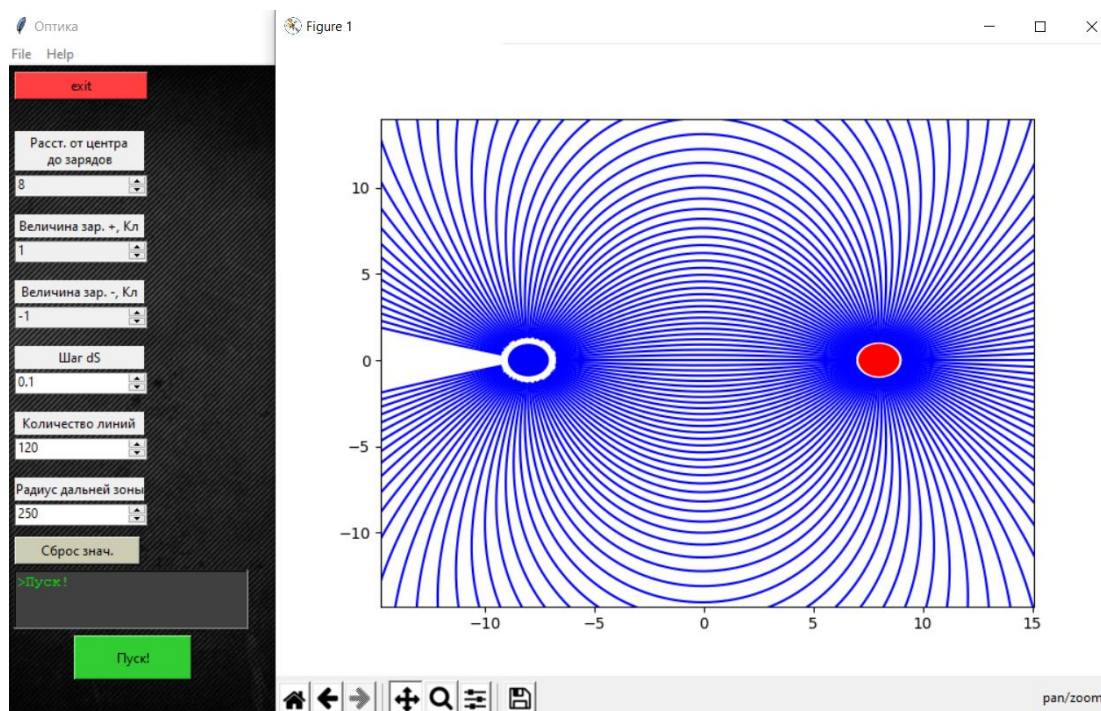


Рисунок 2. Отсутствие сектора с линиями слева от отрицательного заряда (радиус дальней зоны – 250м)

Этот недостаток можно исправить, если уменьшить величину шага и увеличить дальнюю зону, на границе с которой линии обрываются. На следующем рис.3 представлена более качественная картина поля для двух разноимённых зарядов. Здесь шаг выбран равным $\Delta L = 0.001$. Уплотнение линий на горизонтальной оси связано с проблемой плоского изображения трехмерного поля.

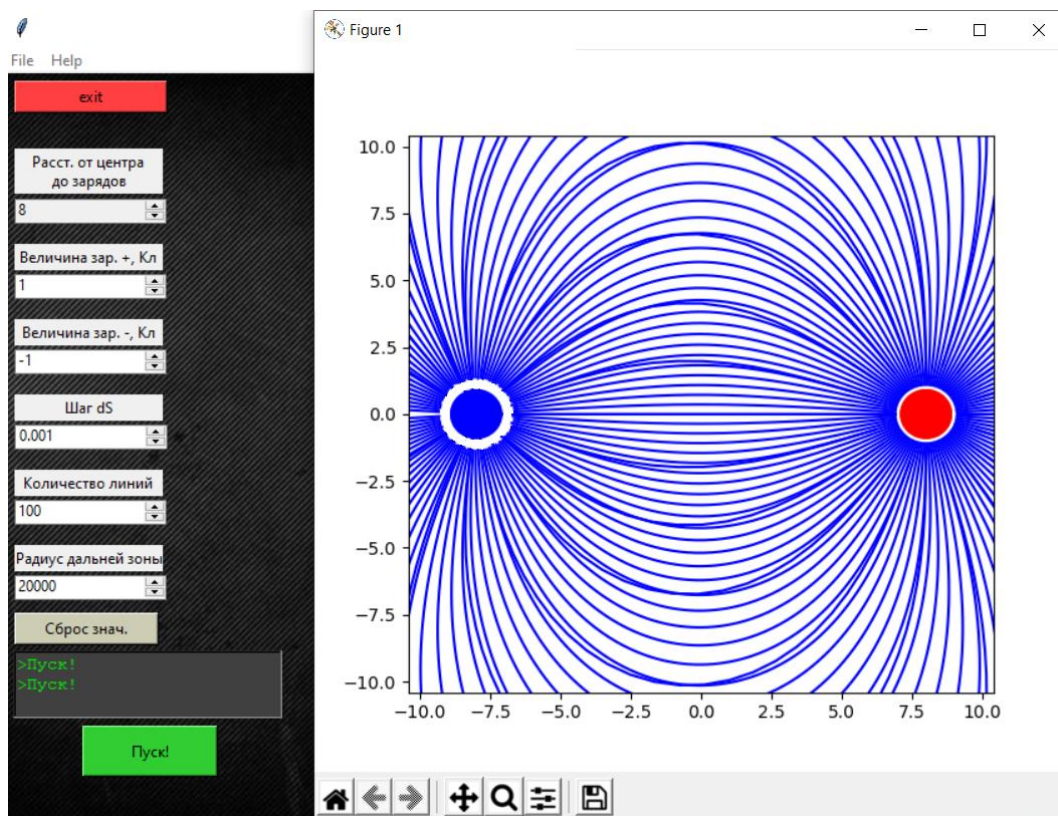


Рисунок 3. Поле двух разноимённо заряженных точечных зарядов

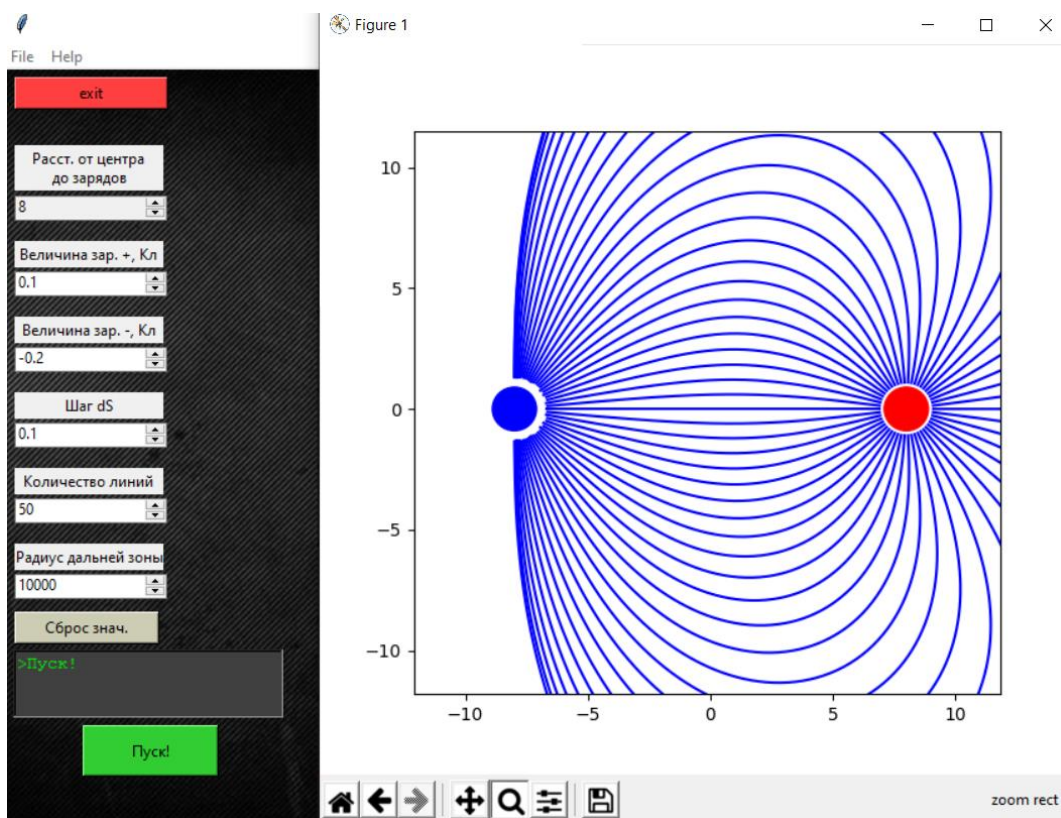


Рисунок 4. Поле двух разноимённо заряженных точечных зарядов, $|Q_{\text{neg}}| = 2 |Q_{\text{pos}}|$

3. Описание программы и ход работы

Программа строит силовые линии электрического поля для системы из двух разноимённо заряженных точечных зарядов.

В интерфейсе программы мы можем задать:

- Положение зарядов (чтобы получать картину линий поля, выравненную по горизонтальной линии, мы реализовали эту возможность изменением расстояния между зарядами);
- Величину положительного и отрицательного зарядов Q_{pos} , Q_{neg} ;
- Шаг приращения dS ;
- Количество линий N ;
- Радиус дальней зоны R_d .

С помощью панели инструментов инструментов для полотна библиотеки `matplotlib`, с помощью которой мы строим картину поля, можно изменить масштаб изображения, сохранить его в файл и перемещаться по полотну.

```
## Получаем точки окружности и корректируем их
def get_points(radius, number_of_points, correct):
    radians_between_each_point = 2*np.pi/number_of_points
    list_of_points = [[],[]]
    for p in range(0, number_of_points):
        list_of_points[0].append(radius*np.cos(p*radians_between_each_point))
        list_of_points[1].append(radius*np.sin(p*radians_between_each_point))
    for i in range(len(list_of_points[0])):
        list_of_points[0][i] += correct[0]
    for i in range(len(list_of_points[1])):
        list_of_points[1][i] += correct[1]

    return list_of_points
```

Рисунок 4. Функция радиального выбора точек старта у
положительного заряда

```

## Алгоритм расчёта вектора напряжённости в заданной точке
def calc_vect_E(charges, np_r, np_vector_array):
    #print(np_r)
    Sum = 0
    for i in range(0, len(charges)):
        newVect = np_r - np_vector_array[i]
        #print("newVect: ", newVect)
        vectModule = np.linalg.norm(newVect)**3
        #print("vectModule: ", vectModule)
        Ei = charges[i]*newVect/vectModule
        #print("Ei: ", Ei)
        Sum+= Ei
        #print(Sum)

    Sum *= k
    return Sum

```

Рисунок 5. Алгоритм расчёта вектора напряжённости в заданной точке

```

#функция отрисовки одной линии
def draw_a_line():

    control_mas = []

    x_glob = len(global_array)-2
    y_glob = len(global_array)-1

    global_array.append([])
    global_array.append([])

    iter_ = 0
    while True:
        iter_ = iter_+1
        x1 = np_vector_array[1][0]
        y1 = np_vector_array[1][1]

        if ((np_r[0]-x1)**2 + (np_r[1]-y1)**2)**0.5 <=1.4:
            break

        E = calc_vect_E(charges, np_r, np_vector_array)
        #print('E0= ',E[0], 'E1= ',E[1])
        Ex = E[0]
        Ey = E[1]
        dx = dS*Ex/np.linalg.norm(E)
        dy = dS*Ey/np.linalg.norm(E)
        #print('dx = ',dx,'|','dy = ',dy,'|','norm_E= ',np.linalg.norm(E))

        np_r[0]+=dx
        np_r[1]+=dy
        #print('np_r[0] = ',np_r[0], '|','np_r[1] = ',np_r[1], '|','norm_E= ',np.linalg.norm(E))

        global_array[x_glob].append(np_r[0])
        global_array[y_glob].append(np_r[1])

        control_mas.append(np_r[1])

```

Рисунок 6. Функция построения линии напряжённости.

4. Верификация алгоритма и полученных данных

Полученные нами изображения линий поля для разноимённо заряженных зарядов соответствуют изображениям, приведённым в методическом пособии [1].

Таблица 1. Расстояния между соседними точками при разном шаге приращения координат.

Вычисление №1, Шаг 0,9		
	Координата x	Координата y
x ₁	9,990009999950177	0,000019932254210221543
x ₂	10,980009999899416	0,000029957499967712618
x ₂ -x ₁ , y ₂ -y ₁	0,99	0,0000100252457574911
d, м	0,989999999999991	
Вычисление №2, Шаг 0,0001		
	Координата x	Координата y
x ₁	9,000109999999994	0,000010001003258001033
x ₂	9,000209999999988	0,000010002006516927978
x ₂ -x ₁ , y ₂ -y ₁	0,0000999999999891	0,0000000010032589269
d, м	0,0000999999999941415	

Вывод:

В ходе лабораторной работе мы ознакомились методами моделирования электростатических полей, построили силовые линии поля, для построения которых следует осуществить последовательные достаточно малые шаги вдоль направления напряженности поля; повысили навыки создания графических приложений на Python, с использованием библиотеки визуализации данных Matplotlib.

ЛИТЕРАТУРА

1. Милюков В.В. Учебно-методическое пособие по курсу «компьютерное моделирование в физике»
2. Гулд Х., Тобочник Я. Компьютерное моделирование в физике. Часть 1
3. Официальный сайт с документацией Python [Электронный ресурс] <https://www.python.org/doc/>.
4. Лучано Рамальо Python. К вершинам мастерства. – М.: ДМК Пресс, 2016. – 768 с.