Изображение выглядит как Шрифт, логотип, Графика, белый

Автоматически созданное описание**Университет ИТМО**

**Физико-технический мегафакультет Физический факультет**

|  |  |
| --- | --- |
| Группа M3201 | К работе допущен |
| Студенты Ткачук С.A. и Чуб Д.О. | Работа выполнена |
| Преподаватель Шоев В.И. | Отчет принят |

****

**Отчет по моделированию № 2.3**

**Частица в конденсаторе**

1. **Теоретическая часть**

В данной работе исследуется движение электрона в конденсаторе под действием электрического поля, создаваемого разностью потенциалов между пластинами конденсатора. Используются уравнения движения в электрическом поле. Используется модель движения заряженной частицы в конденсаторе. Решение основано на численном методе, а именно, методе бинарного поиска, для определения минимальной разности потенциалов между пластинами конденсатора, при которой электрон не покидает конденсатор.

1. **Используемые формулы:**

(1) -ускорение по оси y, зависящее от заряда частицы, ее массы, расстоянии d частицы от оси симметрии цилиндров, радиусов цилиндров и разности потенциалов обкладок

Формула (1) получена из 3-х формул:

- формула ускорения электрона в электрическом поле

– формула напряженности электрического поля между обкладками цилиндрического конденсатора, на расстоянии d от оси цилиндров

– формула разности потенциалов между обкладками цилиндрического конденсатора

Выражая через получаем, что

(2) – изменение скорости по оси y

(3) – изменение координаты y

1. **Численный алгоритм для решения уравнений**

Используется бинарный поиск для поиска минимальной разности потенциалов между обкладками конденсатора

1. **Программный код на языке python:**

import matplotlib.pyplot  
import math  
import numpy  
  
r = 0.075  
R = 0.16  
v\_x = 2 \* (10 \*\* 6)  
L = 0.24  
q = -1.6021766208 \* (10 \*\* -19)  
m = 9.1093837015 \* (10 \*\* -31)  
  
  
def get\_next\_v\_y(v\_y\_pred, a\_y):  
 return v\_y\_pred + a\_y \* delta\_t  
  
  
def get\_next\_y(y\_pred, v\_y, a\_y):  
 return y\_pred + v\_y \* delta\_t + a\_y \* delta\_t \* delta\_t / 2  
  
  
def get\_a\_y(u, y):  
 return (u \* q) / (m \* (y + r) \* math.log(R / r))  
  
  
total\_flight\_time = L / v\_x  
  
delta\_t = 10 \*\* -9  
u\_left, u\_right = 0, 1000  
u\_mid = 0  
  
while u\_right - u\_left > (10 \*\* -15):  
  
 u\_mid = (u\_right + u\_left) / 2  
  
 y = (R - r) / 2  
 v\_y = 0  
 a\_y = get\_a\_y(u\_mid, y)  
  
 is\_out = False  
  
 for i in range(1, int(total\_flight\_time \* (10 \*\* 9))):  
 y = get\_next\_y(y, v\_y, a\_y)  
 v\_y = get\_next\_v\_y(v\_y, a\_y)  
 a\_y = get\_a\_y(u\_mid, y)  
 if y < 0 or y > R - r:  
 is\_out = True  
 break  
  
 if is\_out:  
 u\_right = u\_mid  
 else:  
 u\_left = u\_mid  
  
y = (R - r) / 2  
v\_y = 0  
  
y\_values = []  
v\_y\_values = []  
a\_values = []  
t\_values = []  
x\_values = numpy.linspace(0, L, int(total\_flight\_time \* (10 \*\* 9)))  
  
for i in range(0, int(total\_flight\_time \* (10 \*\* 9))):  
 a\_y = get\_a\_y(u\_left, y)  
 y = get\_next\_y(y, v\_y, a\_y)  
 v\_y = get\_next\_v\_y(v\_y, a\_y)  
  
 y\_values.append(y)  
 v\_y\_values.append(v\_y)  
 a\_values.append(a\_y)  
 t\_values.append(i \* (10 \*\* -9))  
  
figure = matplotlib.pyplot.figure(figsize=(10, 10))  
matplotlib.pyplot.subplots\_adjust(wspace=0.6, hspace=0.6)  
  
subplot1 = figure.add\_subplot(2, 2, 1)  
subplot1.set\_title("y(x)")  
subplot1.set\_xlabel("x, m", color='black')  
subplot1.set\_ylabel("y, m", color='black')  
subplot1.plot(x\_values, y\_values, color='green')  
  
subplot2 = figure.add\_subplot(2, 2, 2)  
subplot2.set\_title("v\_y(t)")  
subplot2.set\_xlabel("t, s", color='black')  
subplot2.set\_ylabel("v\_y, m/s", color='black')  
subplot2.plot(t\_values, v\_y\_values, color='green')  
  
subplot3 = figure.add\_subplot(2, 2, 3)  
subplot3.set\_title("a\_y(t)")  
subplot3.set\_xlabel("t, s", color='black')  
subplot3.set\_ylabel("a, m/s^2", color='black')  
subplot3.plot(t\_values, a\_values, color='green')  
  
subplot4 = figure.add\_subplot(2, 2, 4)  
subplot4.set\_title("y(t)")  
subplot4.set\_xlabel("t, s", color='black')  
subplot4.set\_ylabel("y, m", color='black')  
subplot4.plot(t\_values, y\_values, color='green')  
  
matplotlib.pyplot.show()  
  
print("Potential difference: {:.7f}".format(u\_right))  
v = math.sqrt(v\_x \*\* 2 + v\_y \*\* 2)  
print("Speed: {:.7f}".format(v))  
print("Time: {:.7f}".format(total\_flight\_time))

1. **Вывод**

В данной работе проведено численное моделирование движения электрона в цилиндрическом конденсаторе под воздействием электрического поля. Используется метод бинарного поиска для определения минимальной разности потенциалов между пластинами конденсатора, при которой электрон не покидает его пределов. Исследование позволяет оценить влияние электрического поля на движение электрона и определить условия его задержания в конденсаторе.