

# Рабочий протокол и отчёт по моделированию №2

## «Частица в конденсаторе»

### Вариант №10

#### 1)Цель работы

Спроектировать цилиндрический конденсатор.

Найти минимальную разность потенциалов.

Построить графики зависимостей  $y(x)$ ,  $Vy(t)$ ,  $ay(t)$ ,  $y(t)$ .

Расчитать время полёта и конечную скорость электрона.

#### 2)Формулы

$$E = \frac{U_{\min}}{d} \quad F = \bar{e} \cdot E = \frac{\bar{e} \cdot U_{\min}}{d}$$

$$m \cdot a = F = \frac{\bar{e} \cdot U_{\min}}{d} \Rightarrow a = \frac{\bar{e} \cdot U_{\min}}{m \cdot d}$$

$$\frac{d}{2} = \frac{1}{2} \cdot a \cdot \tau^2 = \frac{1}{2} \cdot \frac{\bar{e} \cdot U_{\min}}{m \cdot d} \cdot \left(\frac{l}{v}\right)^2 \quad U_{\min} = \frac{m}{\bar{e}} \cdot \left(\frac{d}{l} \cdot v\right)^2$$

#### 2)Основа кода

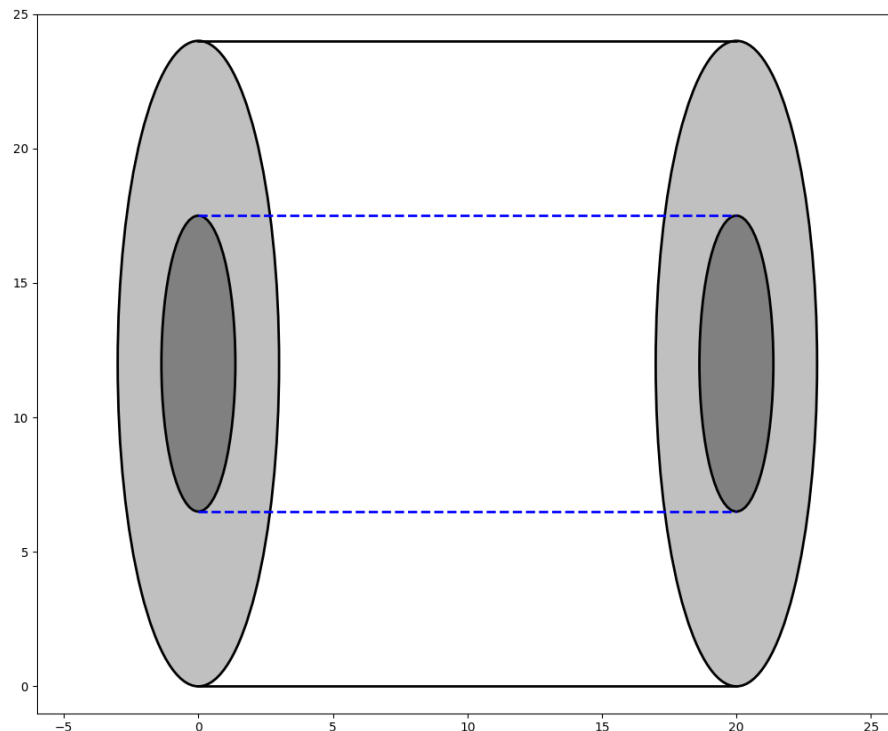
```
import numpy as np
# Внутренний радиус
r = 5.5
# Внешний радиус
R = 12
# Начальная скорость
v = 4 * (10 ** 6)
# Длина конденсатора
L = 20
# удельный заряд электрона
e_m0 = 1.6 / 9.1 * (10 ** 12)
E = (e_m0 * (v ** 2)) / ((L ** 2) * (R-r))
# X
X = np.linspace(0, 10, 100)
# Время
t = L / v
```

```

# Конечная скорость
v_res = v * np.sqrt(1 - ((R - r) ** 2) / (L ** 2))
# Зависимости, которые мы ищем
# y(x)
y_x = (R - r) / 2 - 0.5 * E * e_m0 * (X ** 2) / (v ** 2)
# vvy(t)
v_yt = -E * e_m0 * t
# y(t)
y_t = (R - r) / 2 - 0.5 * E * e_m0 * (t ** 2)
# avy(t)
a_yt = [E * e_m0 for _ in t]
# Минимальная разность потенциалов
U_min = ((R - r) ** 2) * 1 * (v ** 2) / ((L ** 2) * e_m0)

```

### 3) Схема установки



```

import matplotlib.pyplot as plt
import numpy as np
from matplotlib.patches import Ellipse
# Внутренний радиус
r = 5.5
# Внешний радиус
R = 12
# Начальная скорость
v = 4 * (10 ** 6)
# Длина конденсатора
L = 20

Graph = [
    Ellipse(xy=(0, R), width=R / 2, height= 2 * R, angle=0,facecolor='silver',
    edgecolor = 'black', linewidth = 2),
    Ellipse(xy=(0, R), width=r / 2, height=2 * r, angle=0,facecolor='gray',
    edgecolor = 'black', linewidth = 2),
    Ellipse(xy=(L, R), width=R / 2, height= 2 * R, angle=0,facecolor='silver',
    edgecolor = 'black', linewidth = 2),
    Ellipse(xy=(L, R), width=r / 2, height=2 * r, angle=0,facecolor='gray',
    edgecolor = 'black', linewidth = 2)
]

```

```

fig = plt.figure(0)
ax = fig.add_subplot(111, aspect='equal')

for e in Graph:
    ax.add_artist(e)
    e.set_clip_box(ax.bbox)
    e.set_alpha(1)
X = np.linspace(0, L, 100)
Y1 = [2 * R for x in X]
Y2 = [R + r for x in X]
Y3 = [0 for x in X]
Y4 = [R - r for x in X]
ax.plot(X, Y1, 'black', X, Y2, 'b--', X, Y3, 'black', X, Y4, 'b--',
        linewidth = 2)
ax.scatter(0, 7 * R / 4, c = 3)
ax.set_xlim(-R / 2, L + R / 2)
ax.set_ylim(-1, 2 * R + 1)
fig.set_figheight(15)
fig.set_figwidth(15)
plt.show()

```

#### 4)Расчёт результатов

Минимальная разность потенциалов: 9.611875 В

Время: 0.000005 с

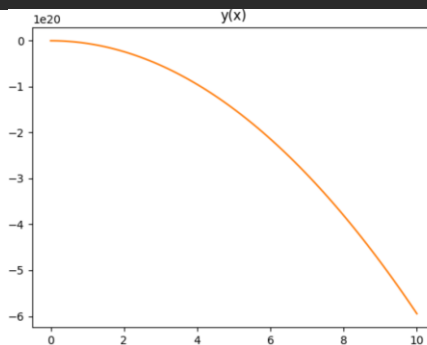
Конечная скорость: 3782856.0638755476 м/с

#### 5)Графики

```

y_x = (R - r) / 2 - 0.5 * E * e_m0 * (X ** 2) / (v ** 2)
plt.plot(X, y_x, 'tab:orange')
plt.title("y(x) ")
plt.show()

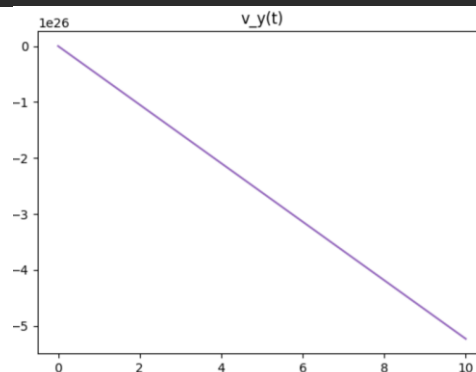
```



```

v_yt = -E * e_m0 * t
plt.plot(X, v_yt, 'tab:purple')
plt.title("v_y(t) ")
plt.show()

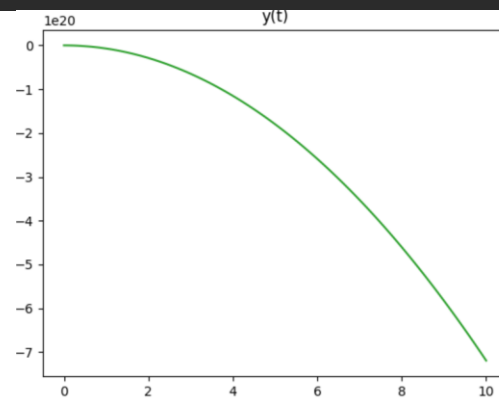
```



```

y_t = (R-r) / 2 - 0.5 * E * e_m0 * (t ** 2)
plt.plot(X, y_t, 'tab:green')
plt.title("y(t)")
plt.show()

```



```

a_yt = [E * e_m0 for _ in t]
plt.plot(X, a_yt, 'tab:blue')
plt.title("a_y(t)")
plt.show()

```

