

Проект № 1: Тело, брошенное под углом к горизонту

Факультет систем управления и робототехники

Выполнил: студент гр. R3137

Кожевникова Полина Вадимовна

Преподаватель: Смирнов Александр Витальевич

Санкт-Петербург, 2022

Выполнение задачи с пренебрежением сопротивления воздуха

Задача: Зенитное орудие сообщает снаряду скорость v_0 и α , рассчитать зону поражения снарядами

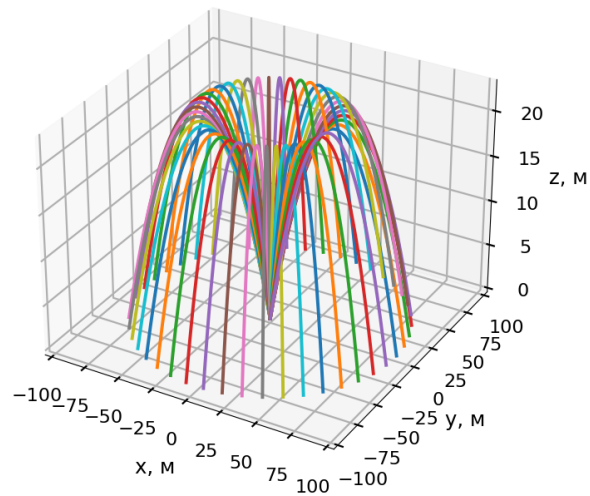
$$x(t) = V_{0x} * t, y(t) = V_{0y} * t - \frac{g * t^2}{2}$$

$$V_x = V_{0x} = V_0 * \cos(\alpha), V_y = V_{0y} - g * t$$

$$t = \frac{2 * V_0 * \sin(\alpha)}{g}, L = V_{0x} * t = \frac{V_0^2 * \sin(2 * \alpha)}{g}$$

$$H = \frac{V_0^2 * \sin(\alpha)}{2g}$$

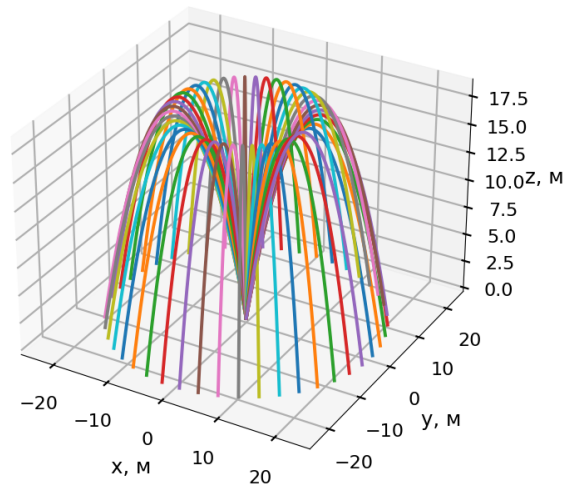
↓ **График № 1 Зона поражения при $v_0 = 30 \frac{м}{с}$ и $\alpha = 45^\circ$**



Дальность полета : $l = 91.654$ м

Высота полета : $h = 22.914$ м

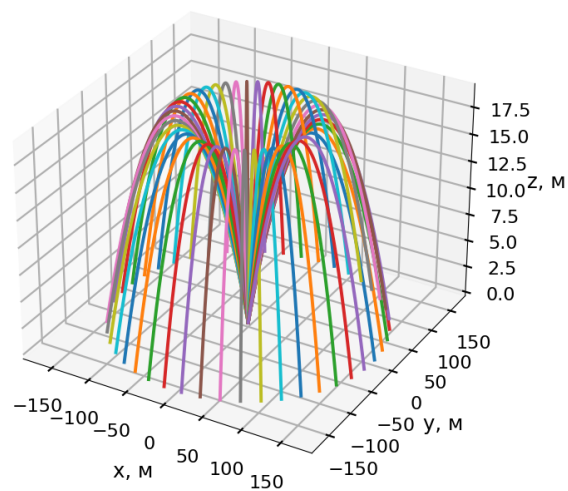
↓ **График № 2 Зона поражения при $v_0 = 20 \frac{\text{м}}{\text{с}}$ и $\alpha = 73^\circ$**



Дальность полета : $l = 22.779$ м

Высота полета : $h = 18.626$ м

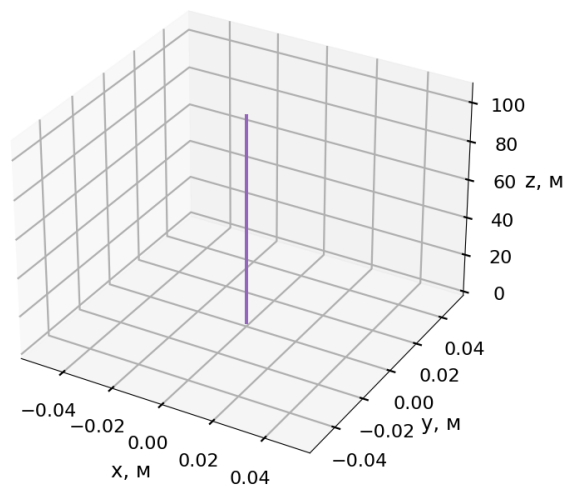
↓ **График № 3 Зона поражения при $v_0 = 46 \frac{\text{м}}{\text{с}}$ и $\alpha = 25^\circ$**



Дальность полета : $l = 165.075$ м

Высота полета : $h = 19.244$ м

↓ **График № 4 Зона поражения при $v_0 = 46 \frac{\text{м}}{\text{с}}$ и $\alpha = 90^\circ$**



Дальность полета : $l = 0$ м

Высота полета : $h = 107.745$ м

Код программы:

```

import matplotlib.pyplot as plt
import numpy as np
plt.style.use('seaborn-poster')

def plot_trace(v0, alpha, **kwargs):

    # acceleration of gravity
    g = 9.8195

    deg2rad=np.pi/180
    # time to max height
    tp = v0 * np.sin(alpha*deg2rad) / g

    # change size for figure
    fig = plt.figure(figsize=(8, 8))
    ax = plt.axes(projection='3d')
    ax.grid()

    # converting to time range
    t = np.linspace(0, 2 * tp, 1000)
    for thetta in range(1, 360, 8):
        # x axis
        x = v0 * np.around(np.cos(alpha*deg2rad), decimals=5) * t * np.cos(thetta*d
eg2rad)

        # y axis
        y = v0 * np.around(np.cos(alpha*deg2rad), decimals=5) * t * np.sin(thetta*d
eg2rad)

        # z axis
        z = v0 * np.around(np.sin(alpha*deg2rad), decimals=5) * t - g * (t ** 2) / 2
        ax.plot3D(x, y, z)

    # calculating length
    l =(v0**2*np.around(np.sin(2*alpha*deg2rad), decimals=5))/g
    #height
    h=v0**2*np.sin(alpha*deg2rad)**2/(2*g)

    # axis labels
    ax.set_xlabel('x, m', labelpad=20)
    ax.set_ylabel('y, m', labelpad=20)
    ax.set_zlabel('z, m', labelpad=20)

    # chart title
    print(l)
    print(h)
    print(tp)

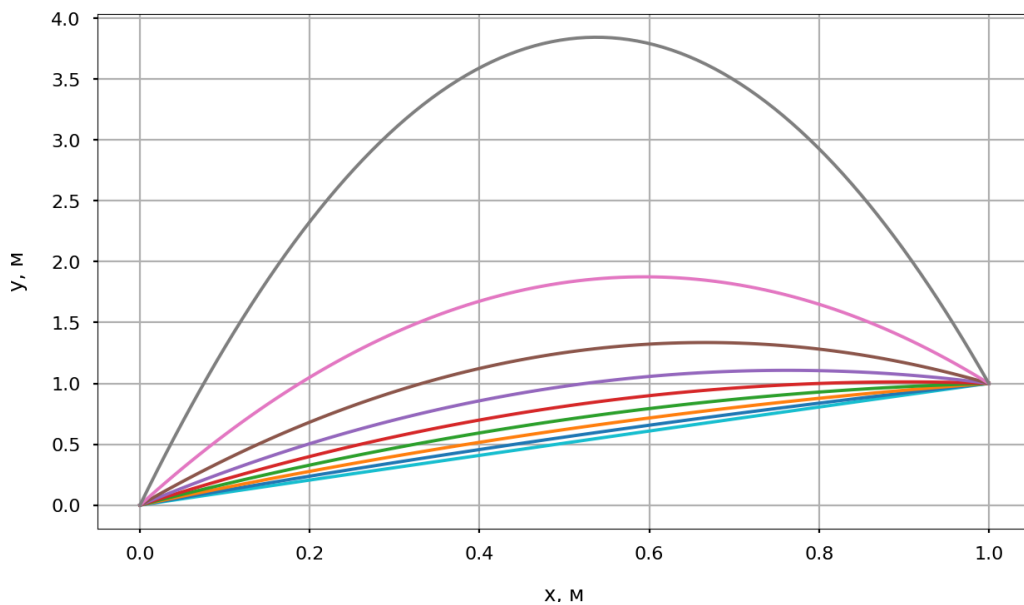
```

```
plt.grid(True)
plt.show()

plot_trace(46, 90, color='m')
```

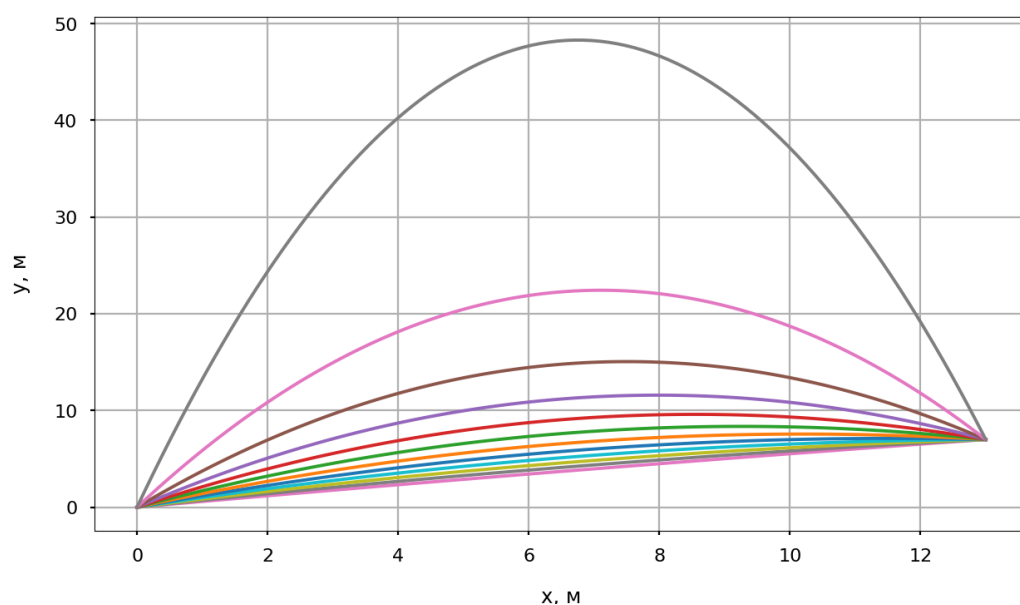
Задача: Даются координаты x, y цели, рассчитать v_0 и α для их достижения

↓ **График № 5 Координаты $x = 1$ м, $y = 1$ м**



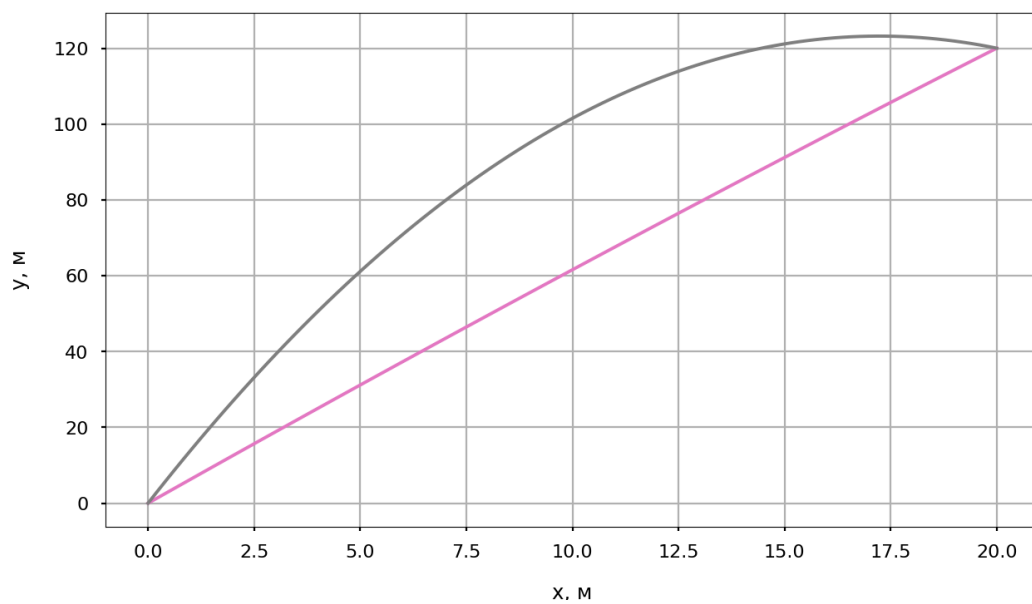
v_0 , м/с	α , °
16.922280699561554	46
7.264722100219481	51
5.704163198495082	56
5.097034091713708	61
4.880349893814645	66
4.932081573013854	71
5.278553578102439	76
6.144642596051387	81
8.7098058210987	86

↓ **График № 6 Координаты $x = 13$ м, $y = 7$ м**



$v_0, \text{ м/с}$	$\alpha, ^\circ$
37.31184074804157	31
22.770432016987588	36
18.404423204800334	41
16.312549818090627	46
15.212096597957565	51
14.703842649509909	56
14.648200245082142	61
15.031355345344892	66
15.954201916064715	71
17.722147861804906	76
21.251129936484034	81
30.872573990220978	86

↓ График № 7 Координаты $x = 13 \text{ м}$, $y = 120 \text{ м}$

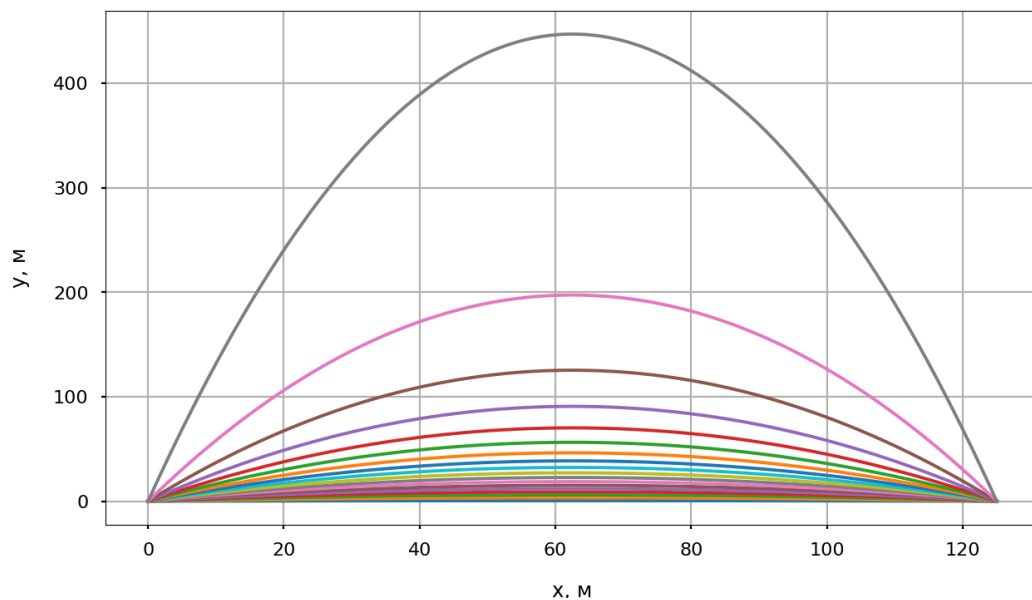


$v_0, \text{ м/с}$ $\alpha, ^\circ$

113.08873088736891 81

49.30644516621357 86

↓ **График № 8 Координаты $x = 125 \text{ м}$, $y = 0 \text{ м}$**



$v_0, \text{ м/с}$ $\alpha, ^\circ$

187.53835911675924 1

76.83520221732272 6

57.24162908528541 11
48.12767034588472 16
42.82962246126436 21
39.46695510630568 26
37.28483541056106 31
35.924979664064644 36
35.20653676582406 41
35.04548012982965 46
35.423991642806776 51
36.38452951757541 56
38.04428723838525 61
40.64086703733357 66
44.65074447310173 71
51.13226702678423 76
63.024369795598886 81
93.91220588855256 86

Код программы:

```
import matplotlib.pyplot as plt
import numpy as np
plt.style.use('seaborn-poster')

def plot_trace(xg, yg, **kwargs):
    # calculating length
    l = (xg ** 2 + yg ** 2) ** 0.5
    # acceleration of gravity
    g = 9.8195

    deg2rad=np.pi/180

    # change size for figure
    fig = plt.figure(figsize=(8, 8))
    ax=plt.axes()
    ax.grid()

    for theta in range(1,90,5):
        v0=((xg**2*g)/(xg*np.sin(2*theta*deg2rad)-2*yg*np.cos(theta*deg2rad)**
2))**0.5
        # time to max height
        tp = v0 * np.sin(theta*deg2rad) / g

        if yg==0:
            # full time
            t = np.linspace(0, 2 * tp, 100)
```



```

x = v0 * np.cos(theta*deg2rad) * t
y = v0 * np.sin(theta*deg2rad) * t - g * (t ** 2) / 2
else:
    # full time
    toime=xg/(v0*np.cos(theta*deg2rad))
    t = np.linspace(0,toime, 100)
    x = v0 * np.cos(theta*deg2rad) * t
    y = v0 * np.sin(theta*deg2rad) * t - g * (t ** 2) / 2

print(v0, ' ', theta)
plt.plot(x,y)

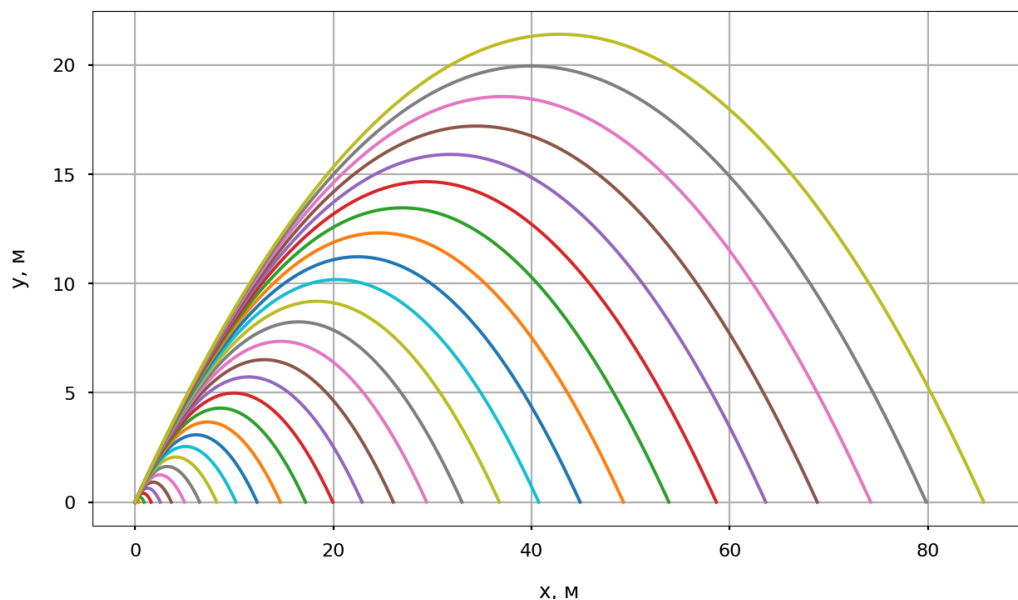
# axis labels
ax.set_xlabel('x, м', labelpad=20)
ax.set_ylabel('y, м', labelpad=20)

plt.show()

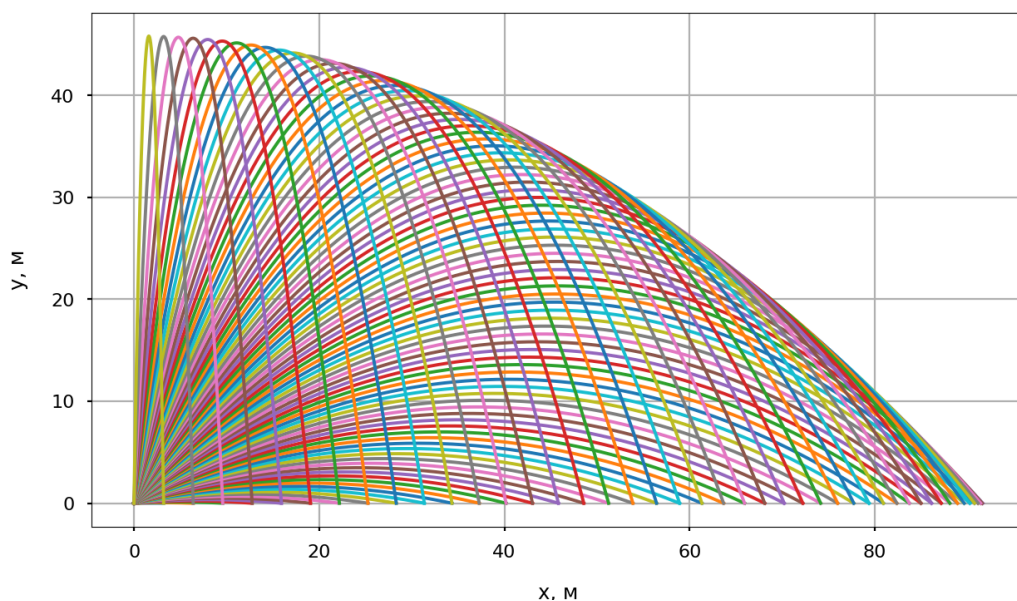
plot_trace(125, 0, color='m')

```

↓ **График № 9 $\alpha = 45^\circ$, v_0 изменяется в промежутке от 1 до 30 м/с**



↓ **График № 10 $v_0 = 30$ м/с, α изменяется в промежутке от 1 до 89°**



Выполнение задачи с сопротивлением воздуха

Задача: Зенитное орудие сообщает снаряду скорость v_0 и α , рассчитать зону поражения снарядами

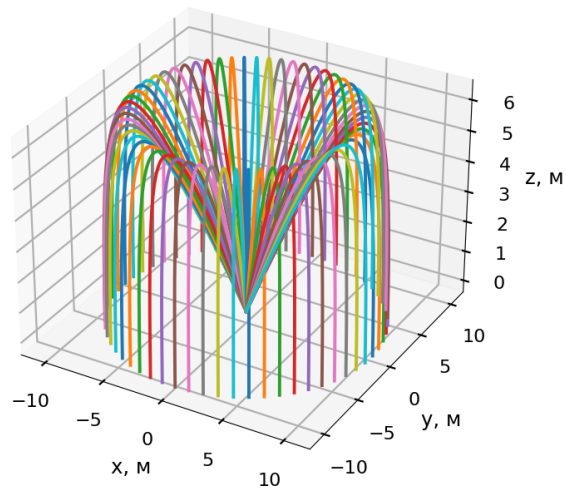
$$m * a_x = F_{air} = -k * v_x \Rightarrow a_x = -\frac{k * v_x}{m}$$

$$m * a_y = -m * g - F_{air} = -m * g - k * v_y \Rightarrow a_y = -g - \frac{k * v_y}{m}$$

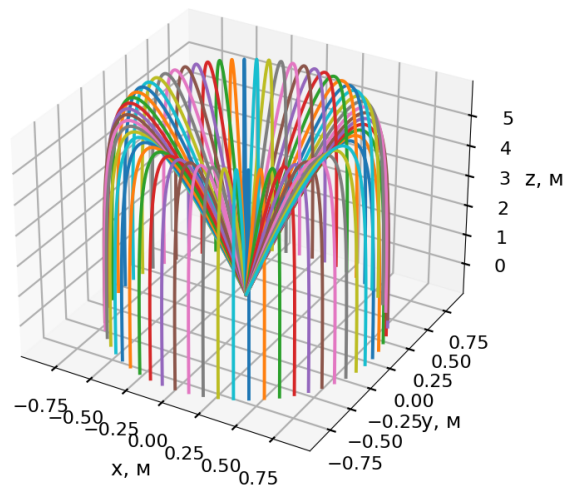
$$v_x = v_{0x} * e^{-\frac{k*t}{m}}, \quad x(t) = \frac{m}{k} * v_{0x} * (1 - e^{-\frac{k*t}{m}})$$

$$v_y = -\frac{g*m}{k} + (v_{0y} + \frac{g*m}{k}) * e^{-\frac{k*t}{m}}, \quad y(t) = \frac{m}{k} \left(-g * t + (v_{0y} + \frac{g*m}{k}) * (1 - e^{-\frac{k*t}{m}}) \right)$$

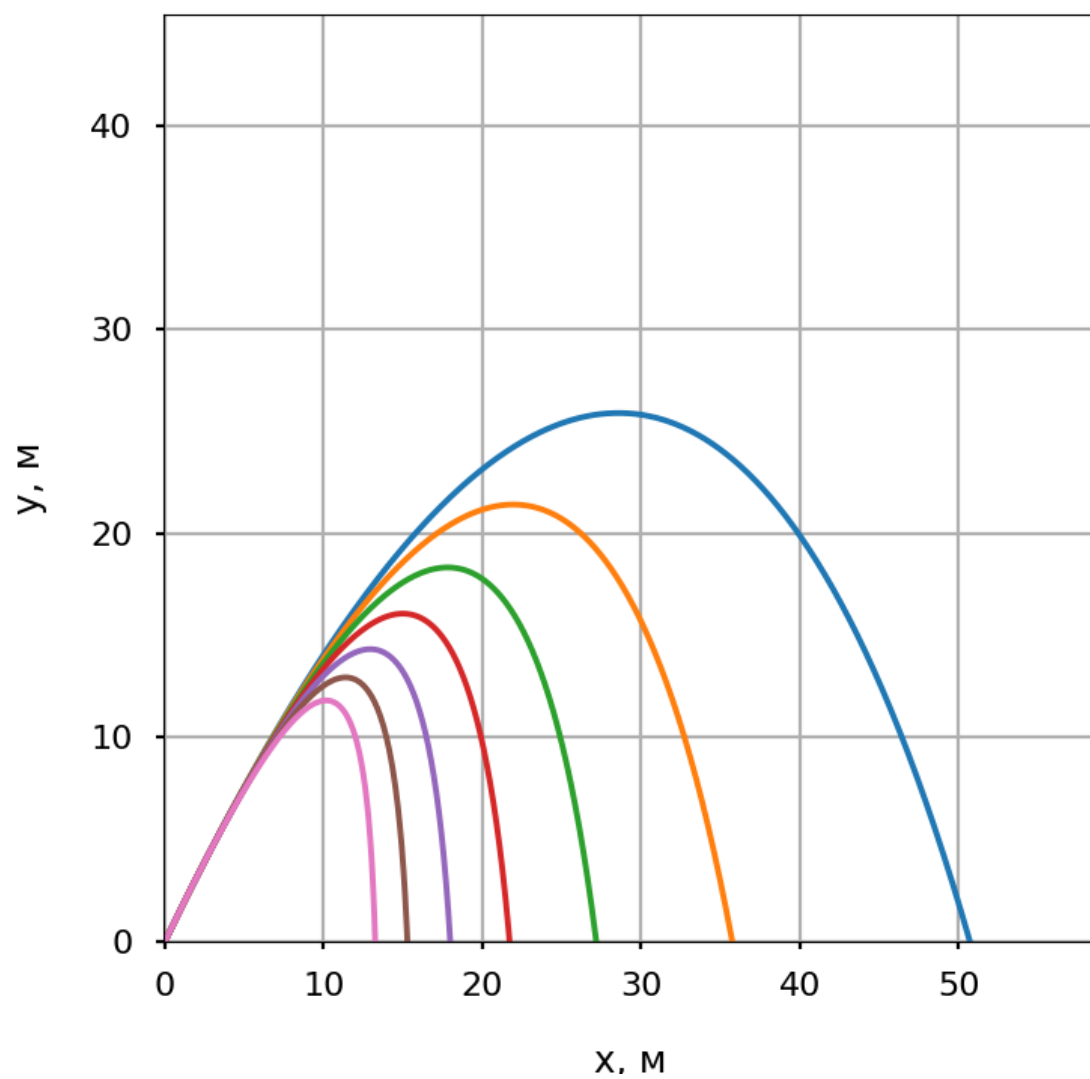
↓ **График № 11** $v_0 = 30 \text{ м/с}$, $\alpha = 45^\circ$, $k = 2 \text{ кг/с}$, $m = 1 \text{ кг}$



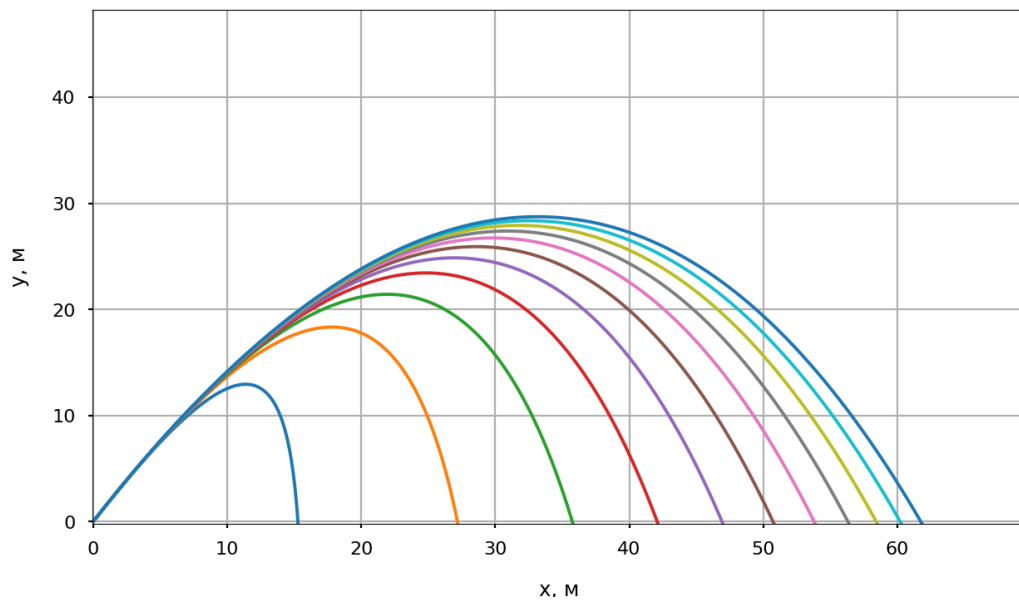
↓ *График № 12* $v_0 = 30 \text{ м/с}$, $\alpha = 45^\circ$, $k = 2 \text{ кг/с}$, $m = 1 \text{ кг}$



↓ *График № 13* $v_0 = 30 \text{ м/с}$, $\alpha = 45^\circ$, $m = 6 \text{ кг}$ *Влияние коэффициента k*



↓ **График № 14** $v_0 = 30 \text{ м/с}$, $\alpha = 45^\circ$, $k = 1 \text{ кг/с}$ Влияние массы снаряда m



Код программы:

```
import matplotlib.pyplot as plt
import numpy as np
plt.style.use('seaborn-poster')

def plot_trace(v0, alpha, k, m, **kwargs):

    # acceleration of gravity
    g = 9.8195

    alphax = alpha * np.pi / 180

    #time equation
    tp=v0 * np.sin(alphax) / g

    # degrees to radians
    deg2rad=np.pi/180

    if alpha == 90:
        cosa = 0
    else:
        cosa = np.cos(alphax)

    # change size for figure
    fig = plt.figure(figsize=(8, 8))
    ax = plt.axes(projection='3d')
    ax.grid()
```

```

# converting to time range
t = np.linspace(0,2.7, 1000)

for thetta in range(1,360,6):
    # x axis
    x = (-v0*cosa / k*m*np.exp(-k/m*t) + v0*cosa / k*m)* np.cos(thetta*deg
2rad)

    # y axis
    y = (-v0*cosa / k*m*np.exp(-k/m*t) + v0*cosa / k*m)*np.sin(thetta*deg2
rad)

    # z axis
    z = (v0*np.sin(alphax) + g * m / k) * m / k*(1- np.exp(-k*t/m)) - g*t*
m / k

    ax.plot3D(x, y, z)

# calculating length
l = v0 ** 2 * np.sin(2 * alphax) / g

# chart title
ax.set_xlabel('x, м', labelpad=20)
ax.set_ylabel('y, м', labelpad=20)
ax.set_zlabel('z, м', labelpad=20)
plt.show()

plot_trace(20, 85, 2,1,color='m')

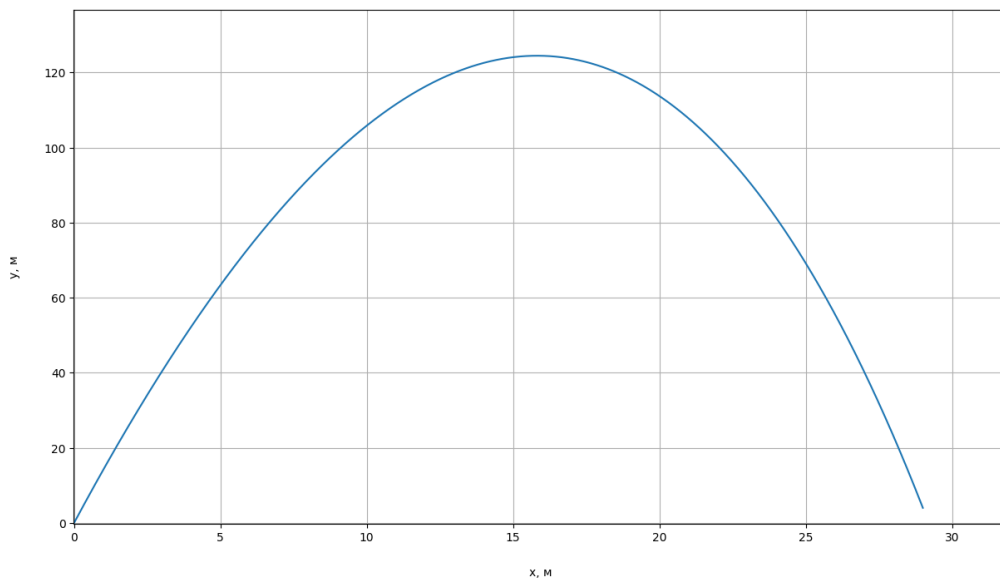
```

Задача: Даются координаты x, y цели, рассчитать v_0 и α для их достижения

$$\operatorname{tg}(\alpha) = \frac{y * k^2 + g * m^2 * (1 - e^{-\frac{k*t}{m}}) + g * t * m * k}{x * k^2} \Rightarrow \alpha = \arctg(\operatorname{tg}(\alpha))$$

$$v_0 = \frac{x * k}{m * (1 - e^{-\frac{k*t}{m}}) * \cos(\alpha)}$$

↓ **График № 14** $x = 29$ м, $y = 4$ м, $k = 0.1$ кг/с, $m = 2$ кг



$$v_0 = 53.807 \text{ м/с} \quad \alpha = 86^\circ$$

Код программы:

```
import matplotlib.pyplot as plt
import numpy as np

def plot_trace(xg, yg, k, m):

    # acceleration of gravity
    g = 9.8195

    # change size for figure
    fig = plt.figure(figsize=(8, 8))
    ax=plt.axes()
    ax.grid()

    tf=10
    ef=np.exp(-k*tf/m)
    tgalpha=(yg*k*k-g*m*m*(1-ef)+g*tf*k*m)/(xg*k*k)
    alpha=np.arctan(tgalpha)
    v0=(xg*k)/(m*(1-ef)*np.cos(alpha))

    t=np.linspace(0, tf, 1000)
    e=np.exp(-k*t/m)
    x=m/k*v0*np.cos(alpha)*(1-e)
    y=(m/k)*((v0*np.sin(alpha)+m*g/k)*(1-e)-t*g)
    print(v0)
    print(alpha * 180 / np.pi)

    plt.plot(x, y)
```

```
# axis labels
ax.set_xlabel('x, м', labelpad=20)
ax.set_ylabel('y, м', labelpad=20)
plt.show()
```

```
plot_trace(29, 4, 0.1, 2)
```

Метод Эйлера:

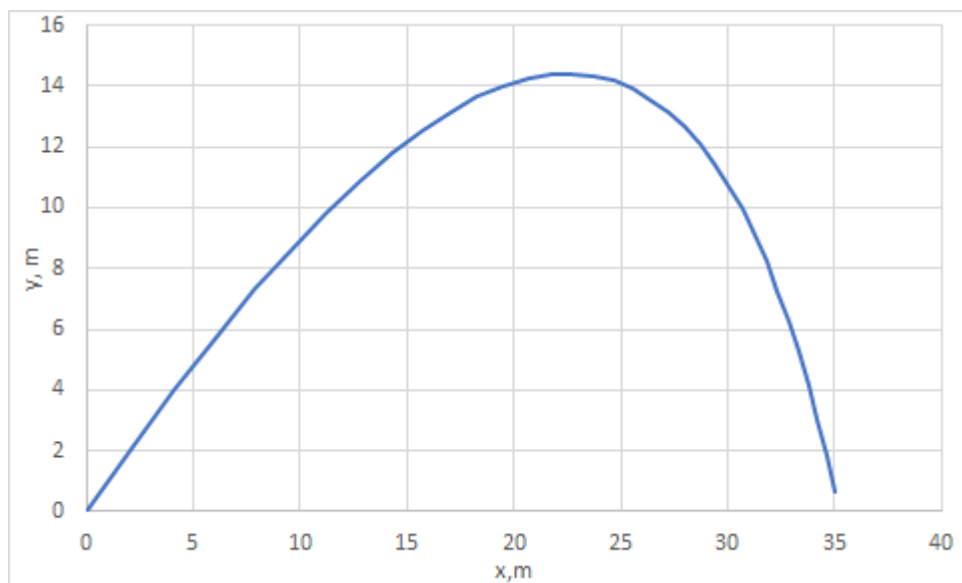
$$v_x(i+1) = v_x(i) + h * v'_x(i), \quad v'_x = a_x$$

$$x(i+1) = x(i) + h * x'(i), \quad x' = v_x$$

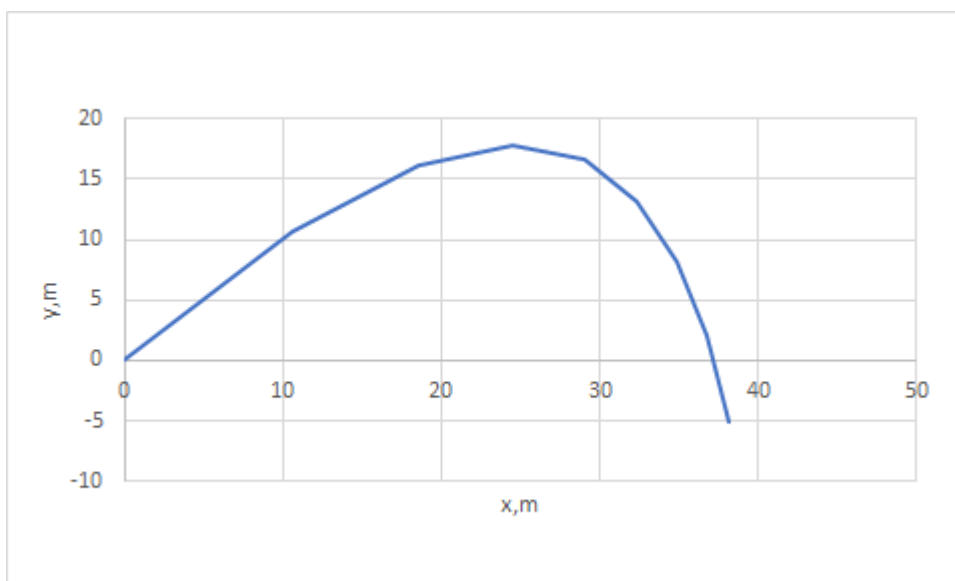
$$v_y(i+1) = v_y(i) + h * v'_y(i), \quad v'_y = a_y$$

$$y(i+1) = y(i) + h * y'(i), \quad y' = v_y$$

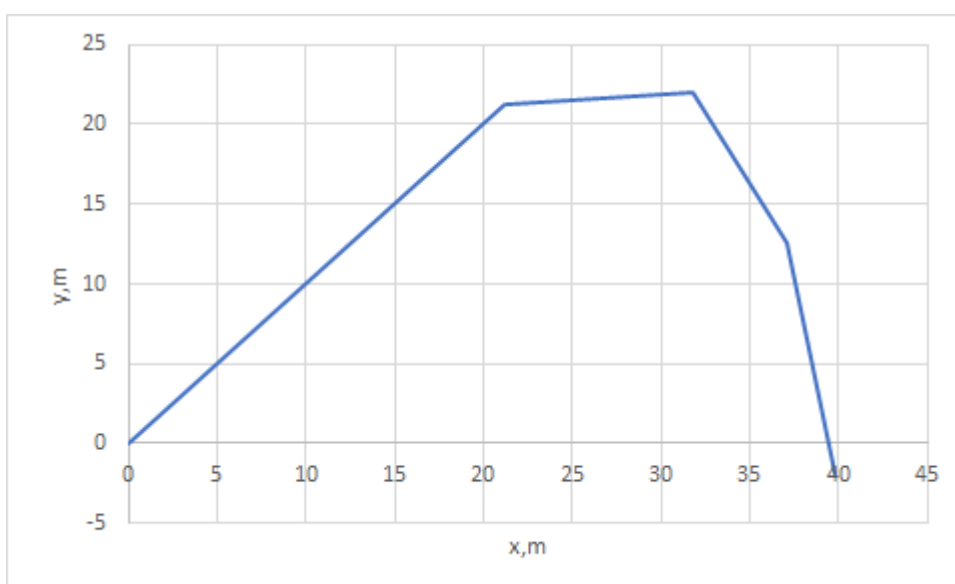
$$h = \Delta t$$



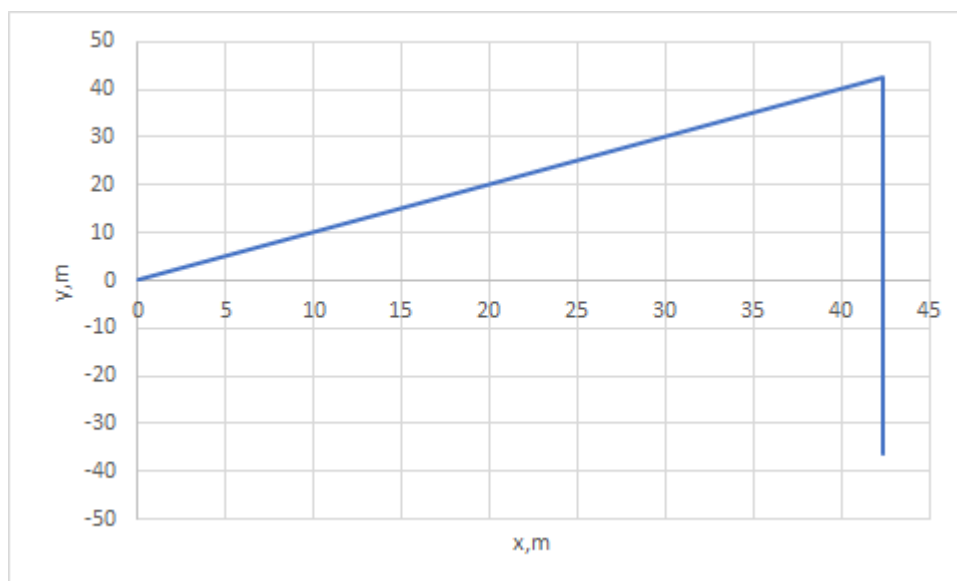
↑ **График № 15** $v_0 = 30 \text{ м/с}$, $\alpha = 45^\circ$, $k = 2 \text{ кг/с}$, $m = 1 \text{ кг}$, $h = 0.1$



↑ График № 16 $v_0 = 30 \text{ м/с}$, $\alpha = 45^\circ$, $k = 2 \text{ кг/с}$, $m = 1 \text{ кг}$, $h = 0.5$



↑ График № 17 $v_0 = 30 \text{ м/с}$, $\alpha = 45^\circ$, $k = 2 \text{ кг/с}$, $m = 1 \text{ кг}$, $h = 1$



↑ График № 18 $v_0 = 30 \text{ м/с}$, $\alpha = 45^\circ$, $k = 2 \text{ кг/с}$, $m = 1 \text{ кг}$, $h = 2$

Вывод:

Мы можем убедиться аналитическим путем, что дальность полета будет максимальна при угле $\alpha = 45^\circ$, а высота полет при угле $\alpha = 90^\circ$. Также если сравнивать графики с сопротивлением воздуха и без можно заметить, что график без сопротивления симметричен по прямой параллельной оси Y , в то время как график с сопротивлением такой симметричности не имеет и падает быстрее после достижения максимальной для заданных параметров высоты. Еще можно заметить тот факт, что при стремлении коэффициента k нулю дальность полета увеличивается.