

### Лабораторная работа 3

#### Тело, брошенное под углом к горизонту

##### 1) Математическая модель

Артиллерийское орудие расположено на горе высотой  $h$ . Снаряд вылетает из ствола со скоростью  $v_0$ , направленной под углом  $\alpha$  к горизонту.

- I. Прямоугольную систему координат выбираем так, чтобы её начало совпало с точкой бросания, а оси координат направлены вдоль поверхности земли и по нормали к ней.
- II. Составляем уравнения скорости и перемещения для их проекций по каждому направлению.

а) В горизонтальном направлении снаряд летит равномерно. Его скорость и координаты в любой момент времени удовлетворяют уравнениям:

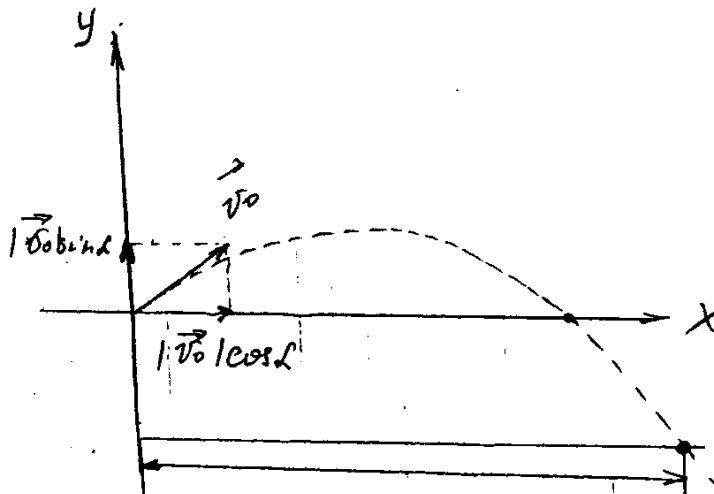
$$v_x = v_0 \cos \alpha$$

$$x = v_0 \cos \alpha * t$$

Для вертикального направления:

$$v_y = v_0 \sin \alpha - gt$$

$$y = v_0 \sin \alpha * t - gt^2$$



В момент времени  $t_1$ , когда снаряд упадёт на землю, его координаты равны:

$$x=S$$

$$y=-h$$

(Знак " – ", так как за время движения снаряд сместится относительно уровня отсчёта 0 высоты в сторону, противоположную направлению, принятому за положительное).

Результирующая скорость в момент падения:

$$v = \sqrt{v_x^2 + v_y^2}$$

В составленной системе уравнений нам нужно определить  $S$  и  $v$ .

Из уравнений найдём время полёта снаряда

$$t_1 = \frac{v_0 \sin \alpha + \sqrt{v_0^2 \sin^2 \alpha + 2gh}}{g}$$

$$v = \sqrt{v_0^2 + 2gh}$$

Чтобы найти уравнение траектории движения снаряда, нужно найти связь между её координатами  $x$  и  $y$  в произвольный момент времени  $t$ .

$$y = \tan \alpha * x - \frac{g}{2v_0^2 \cos^2 \alpha} * x^2$$

## 2) Код программы

```
1. from numpy import*
2. from matplotlib.pyplot import*
3.
4. V0 = 200
5. a_gr = 46
6. h = 70
7. g = 9.8
8. rad = 57.3
9. a = a_gr / rad
10. t_end = (V0 * sin(a) + sqrt(V0 * V0 * sin(a) * sin(a) + 2 * g * h)) / g
11. print(t_end)
12. S = V0 * cos(a) * t_end
13. print(S)
14.
15. y = []
16. x = []
17. j = int(t_end//0.0005)
18. for i in range(j +1):
```

```

19.     t = i / 2000
20.     curX = V0 * cos(a) * t
21.     curY = tan(a) * curX - g * curX * curX / (2 * V0 * V0 * cos(a) * cos(a))
22.     y.append(curY)
23.     x.append(curX)
24.
25.
26. plot(x, y)
27. plot([0, S], [-70, -70])
28. plot([0, S], [0, 0])
29. savefig('body to the horizon.jpg')
30. show()

```

### 3) Вывод программы:

