# Лабораторный практикум 4. Прямая на плоскости. Виды уравнений

## Виды уравнений прямой на плоскости

Загрузите необходимые библиотеки:

>> import numpy as np

>> import matplotlib.pyplot as plt

>> from sympy import \*

%matplotlib widget

1. Уравнение прямой, проходящей через две точки.

Точки на плоскости задаются с помощью функции ***Line***:

Line((1, 4), (5, 3))

или

p1, p2 = Point(1, 4), Point(5, 3)

Для нахождения общего уравнения прямой, проходящей через две точки, используется метод **.*equation*()**.

**Пример 1.** Найти уравнение прямой, проходящей через точки (1,0) и (5,3).

p1, p2 = Point(1,0), Point(5,3)

L = Line(p1, p2)

L.equation()

2) Параметрические уравнения прямой – метод ***.arbitrary\_point*().**

**Упражнение 4.1.** Найти параметрические уравнения прямой из примера 1.

3) Уравнение прямой, проходящей через точку с заданным углом наклона к оси *Ох* (угол в виде углового коэффициента – параметр ***slope***).

**Пример 2.** Найти уравнение прямой, проходящей через точку (–2,3) с угловым коэффициентом, равным 2.

А = Point(-2,3)

к = 2

L = Line(A, slope = к)

print(L.equation())

Рассмотрим на примере один из методов построения графика прямой на плоскости.

**Пример 3.** Построить прямую, заданную общим уравнением на отрезке [–5,5]. В качестве заголовка задать общее уравнение данной прямой: (>>*plt.title('Уравнение прямой 5x-4y-8=0')*). Изобразить на графике:

а) вектор , берущий начало из точки (0, –2);

б) орт вектора , берущий начало из точки (0,0).

x = np.linspace(-5, 5, 100) #создаёт массив с нижней границей -5

#и верхней границей 5, в созданном массиве 100 элементов

n = [5, -4] # нормаль прямой

ortN = n/np.linalg.norm(n)#создаем орт - вектор единичной длины

plt.subplot()

plt.plot(x, (5\*x - 8)/4) #построение графика функции y=(5x-8)/4

#построение векторов

plt.arrow(0, -2, n[0], n[1], width=0.05, color='Red')

plt.arrow(0, 0, ortN[0], ortN[1], width=0.05, color='Green')

plt.title('Уравнение прямой 5x-4y-8=0') #название графика

plt.xlabel('x') #название горизонтальной оси

plt.ylabel('y') #название вертикальной оси

plt.grid () #накладываем сетку

plt.axis('equal')

plt.show()

## Методы, используемые при решении задач

1) Расстояние от точки до прямой – метод ***.distance*()**

**Упражнение 4.2.** Найти расстояние от точки (-1,1) до прямой проходящей через точки (1,4) и (5,3).

2) Угол между прямыми – метод ***.angle\_between*()** и наименьший по величине угол между прямыми – метод ***.smallest\_angle\_between*()**

**Пример 4.** Найти угол между прямыми, проходящими через точки (0,0), (1,0) и (1,1), (0,0). Найти наименьший по величине угол.

е = Line((0, 0), (1, 0))

w = Line((1, 1), (0, 0))

s1=w.angle\_between(e)

s2=w.smallest\_angle\_between(e)

[s1, s2]

3) Создание прямой, параллельной данной прямой и проходящей через заданную точку – метод ***.parallel\_line*()**

4) Создание прямой, перпендикулярной данной прямой и проходящей через заданную точку – метод ***.perpendicular\_line*()**

**Упражнение 4.3.** Прямая *L*1 проходит через точки (0,0) и (2,3). Найти уравнение прямой *L*2, параллельной прямой *L*1 и проходящей через точку (–2,2). Найти уравнение прямой *L*3, перпендикулярной прямой *L*1 и проходящей через точку (–2,2). Сделать рисунок.

5) Проекция точки на прямую – метод ***.projection*()*.***

**Упражнение 4.4.** Спроектировать точку (0.5, 0) на прямую .

6) Точка пересечения двух прямых – метод **.*intersection*().**

#### Упражнение 4.5. Найти точку пересечения прямых *L*1 и *L*3 из упражнения 4.3. Сделать рисунок.

**Пример 5.** Построить прямую, проходящую через точку *M*0(0.6,–0.4) перпендикулярно вектору . На график нанести точку M0 и нормальный вектор, вывести обозначение заданной точки (>> plt.text(0.6,-0.4,'$M\_0(x\_0,y\_0)$'). В качестве заголовка задать общее уравнение данной прямой.

Найти уравнение прямой, проходящей через точку *М*0 с нормальным вектором ***n*** можно с помощью метода *perpendicular\_line*, так как началом и концом нормального вектора всегда можно считать точки (0,0) и (–3,3). Следовательно, прямая, содержащая этот вектор, тоже проходит через эти же точки. Строим перпендикулярную ей прямую:

t1, t2, t3 = Point(0.6, -0.4), Point(0, 0), Point(-2, 2)

H1 = Line(t2, t3)

H2=H1.parallel\_line(t1)

H2=H2.equation()

print(H2)

Далее используем полученное уравнение прямой:

x = np.arange(-5, 5, 0.1)

n = [-3, 3]

M0 = [0.6, -0.4]

y = x-1

plt.plot(M0[0], M0[1],'or')

plt.arrow(M0[0], M0[1], n[0], n[1], width=0.08, color='Red')

plt.plot(x, y)

plt.text(M0[0] + 0.5, M0[1] - 0.5,'$M\_0(x\_0,y\_0)$')

plt.title('Уравнение прямой y=x-1')

plt.xlabel('x')

plt.ylabel('y')

plt.axis('equal')

plt.grid()

plt.show()

**Упражнение 4.6.** Построить прямую, проходящую через точку *M*(0.6,–0.4) параллельно вектору . Сделать рисунок (изобразите вектор ***q*** с началом в точке (0,–2). При решении использовать метод *parallel\_line*.

**Пример 6.** Построить прямую, заданную параметрическим уравнением Найти ее направляющий и нормальный векторы. Проверить их ортогональность. Найти расстояние от прямой до точки ***N***(–4,–1). Изобразить данные векторы исходящими из какой-нибудь точки, не лежащей на прямой.

Прямую, заданную в параметрическом виде удобно строить, придавая параметру *t* различные значения.

t = np.arange(-3, 3, 0.1)

M0 = [-1, 3]

N = [-4, -1]

q = [4, -3] # направляющий вектор из условия

n = [3, 4] # нормальный вектор подбираем

sp = np.dot(n, q) # проверка ортогональности

print('Скалярное произведение векторов (n,q) =', sp)

x = M0[0] + q[0] \* t # массив значений х

y = M0[1] + q[1] \* t # массив значений у

M1 = Point(3, 0) # возьмем еще одну точку на прямой (при t=1)

s = Line(M1, M0)

D=s.distance(N)

print('Расстояние от точки N до прямой =',D)

plt.plot(x, y)

plt.plot(M0[0], M0[1],'or',N[0], N[1],'oy',M1[0], M1[1],'ok')

plt.text(M0[0] + 0.8, M0[1] - 0.5,'$M\_0(-1,3)$')

plt.text(M1[0] + 0.8, M1[1] - 0.5,'$M\_1(3,0)$')

plt.text(N[0] + 0.8, N[1] - 0.5,'$N(-4,-1)$')

# из точки N строим векторы n и q

plt.arrow(0, 10, n[0], n[1], width=0.08,color='g')

plt.arrow(0, 10, q[0], q[1], width=0.08, color='g')

plt.title(f'Уравнение прямой x={M0[0]}+{q[0]}t, y={M0[1]}-{-q[1]}t')

plt.xlabel('x')

plt.ylabel('y')

plt.grid()

plt.axis('square')

plt.show()

**Дополнительное задание**

Даны вершины треугольника *А*(2,–2), *В*(3,5) и *С*(6,1). Найти

1. Уравнение прямой, на которой лежит высота, проведенная из точки *В*.
2. Длину этой высоты.
3. Уравнение прямой, на которой лежит медиана, проведенная из точки *А*.
4. Длину этой медианы.
5. Уравнение прямой, на которой лежит биссектриса угла *С*.

Сделать рисунок.