

```
In [1]: import sympy  
from sympy import Matrix, Point, Line, Plane, Point3D
```

Занятие 6

Алгебра

<https://docs.sympy.org/latest/search.html?q=geometry>

Прямые и плоскости в пространстве. Модуль Geometry

Угол φ между векторами $a(x_1, y_1, z_1)$ и $b(x_2, y_2, z_2)$:

$$\cos \varphi = \frac{(a, b)}{|a| \cdot |b|},$$

где $(a, b) = x_1x_2 + y_1y_2 + z_1z_2$ — скалярное произведение, $|a| = \sqrt{x_1^2 + y_1^2 + z_1^2} = \sqrt{(a, a)}$.

Пример 1.

Найти угол между векторами $a(-3, 2, -6)$ и $b(7, -4, 4)$.

Зададим векторы в виде матриц-столбцов:

```
In [2]: a = Matrix([-3, 2, -6])  
b = Matrix([7, -4, 4])
```

Для вычисления скалярного произведения воспользуемся методом

.dot()

Скалярное произведение векторов X и Y:

X.dot(Y)

Вычислим скалярное произведение векторов a и b :

```
In [3]: a.dot(b)
```

```
Out[3]: -53
```

Найдем угол между векторами, для этого вначале вычислим косинус угла, а затем сам угол:

```
In [4]: cos_phi = a.dot(b)/sympy.sqrt((a.dot(a))*(b.dot(b)))  
phi = sympy.acos(cos_phi)  
cos_phi, phi, float(phi)
```

```
Out[4]: (-53/63, acos(-53/63), 2.5704241550242557)
```

Geometry

Модуль Geometry предназначен для работы с геометрическими объектами, такими как точки, прямые на плоскости и в пространстве, плоскости, плоские многоугольники и т.п.

Point

Класс точек в модуле Geometry называется Point, это родительский класс двумерных (Point2D) и трехмерных (Point3D) точек.

class sympy.geometry.point.Point

Описание класса и его методов и свойств здесь:

<https://docs.sympy.org/latest/modules/geometry/points.html#sympy.geometry.point.Point>

Point - точка в n -мерном Евклидовом пространстве.

Параметры:

coords: последовательность из n координат. При $n = 2$ или 3 используется соответственно Point2D или Point3D.

evaluate: если True (значение по умолчанию), все вещественные числа (float) приводятся к типу, соответствующему значению.

dim: число координат точки.

Пример 2

Найти угол между прямыми AB и CD , $A(-1, 2, -4)$, $B(2, -2, 1)$, $C(5, 3, -6)$, $D(0, 3, -6)$.

Вначале зададим точки:

```
In [5]: A = Point(-1, 2, -4)  
B = Point(2, -2, 1)  
C = Point(5, 3, -6)  
D = Point(0, 3, -6)  
display(A, B, C, D)
```

Point3D(-1, 2, -4)

Point3D(2, -2, 1)

Point3D(5, 3, -6)

Point3D(0, 3, -6)

Проверим, лежат ли точки в одной плоскости:

```
In [6]: Point3D.are_coplanar(A, B, C, D)
```

```
Out[6]: False
```

Вывод: не лежат, прямые скрещиваются

Зададим линии, проходящие через точки:

```
In [7]: AB = Line(A, B)  
CD = Line(C, D)  
display(AB, CD)  
Line3D(Point3D(-1, 2, -4), Point3D(2, -2, 1))  
Line3D(Point3D(5, 3, -6), Point3D(0, 3, -6))
```

Из геометрии мы знаем, что угол между прямыми - это минимальный из углов, образующийся при пересечении прямых. В Geometry реализованы два метода для угла между прямыми, angle_between, вычисляющий какой-нибудь из смежных углов, образующихся при пересечении прямых, и smallest_angle_between, который выдает именно тот угол, который в геометрии принято считать углом между прямыми.

Найдем какой-нибудь угол между прямыми и минимальный из углов, образующихся при пересечении прямой и пересекающей ее прямой, параллельной второй из скрещивающихся прямых:

```
In [8]: display(AB.angle_between(CD), AB.smallest_angle_between(CD))
```

$\cos\left(-\frac{3\sqrt{2}}{10}\right)$

$\cos\left(\frac{3\sqrt{2}}{10}\right)$

Пример 3

Найти угол между плоскостями ABC и ACD.

Вначале зададим плоскости тремя точками, не лежащими на одной прямой, затем вычислим угол (для плоскостей не реализован метод smallest_angle_between):

```
In [9]: ABC = Plane(A, B, C)  
ACD = Plane(A, C, D)  
ABC.angle_between(ACD)
```

$\cos\left(\frac{33\sqrt{1130}}{1130}\right)$

Методы и свойства точек

Point3D.are_collinear

лежат ли точки на одной прямой

Point3D.are_coplanar

лежат ли точки в одной плоскости

Методы и свойства прямой

equation()

уравнение прямой в формате tuple, состоящего из общих уравнений двух плоскостей, пересекающихся по этой прямой

is_parallel()

параллельны ли две прямые

is_perpendicular()

перпендикулярны ли две прямые

is_similar()

совпадают ли две прямые

parallel_line(P)

этот метод возвращает объект Line - прямую, проходящую через точку P параллельно заданной прямой

perpendicular_line(P)

этот метод возвращает объект Line - прямую, проходящую через точку P перпендикулярно заданной прямой

Методы и свойства плоскости

equation()

уравнение плоскости в формате выражения левой части общего уравнения плоскости

normal_vector

вектор нормали к плоскости (свойство, вызывается без ())

is_parallel()

параллельны ли две плоскости

is_perpendicular()

перпендикулярны ли две плоскости

parallel_plane(P)

этот метод возвращает объект Plane - плоскость, проходящую через точку P параллельно заданной плоскости

perpendicular_plane(A, B)

этот метод возвращает объект Plane - плоскость, проходящую через точки A и B перпендикулярно заданной плоскости

Пример 4

Уравнение плоскости:

```
In [10]: ABC.equation()
```

```
Out[10]: 3x + 36y + 27z + 39
```

Пример 5

Вектор нормали к плоскости:

```
In [11]: ABC.normal_vector
```

```
Out[11]: (3, 36, 27)
```

Пример 6

Уравнение плоскости, параллельной ABC, проходящей через D

```
In [12]: Plane6 = ABC.parallel_plane(D)  
Plane6.equation()
```

```
Out[12]: 3x + 36y + 27z + 54
```

Пример 7

Проверка параллельности и перпендикулярности плоскостей:

```
In [13]: print('Параллельны ли ABD и Plane6?')  
print('Перпендикулярны ли ABD и Plane6?')  
Plane6.is_parallel(ABC)
```

Параллельны ли ABD и Plane6?

True

Перпендикулярны ли ABD и Plane6?

False

Пример 8

Проверим, лежат ли точки в одной плоскости:

```
In [14]: Plane8AB = ABC.perpendicular_plane(A, B)
```

```
Out[14]: Plane(Point3D(-1, 2, -4), Point3D(2, -2, 1))
```

Проверим перпендикулярность плоскостей:

```
In [15]: ABC.is_perpendicular(Plane8AB)
```

```
Out[15]: True
```

Пример 9

Построение прямой, перпендикулярной плоскости ABC и проходящей через точку D:

```
In [16]: Line9 = ABC.perpendicular_line(D)  
Line9
```

```
Out[16]: Line3D(Point3D(0, 3, -6), Point3D(3, 39, 21))
```

Проверим, что Line9 перпендикулярна ABC

```
In [17]: ABC.is_perpendicular(Line9)
```

```
Out[17]: True
```

Пример 10

Угол между прямой и плоскостью:

```
In [18]: ACD.angle_between(Line9)
```

```
Out[18]: asin(33*sqrt(1130)/1130)
```

Проверим, что Line9 перпендикулярна ABC

```
In [19]: ABC.is_perpendicular(Line9)
```

```
Out[19]: True
```

Пример 11

Даны точки пространства $A(-4, -1, -5)$, $B(4, -6, 7)$, $C(12, -11, 20)$, $D(20, -16, 31)$, $K(28, -17, 41)$, $M(92, -49, 133)$. Составить уравнения всех (различных!) прямых, проходящих через все пары этих точек.

```
In [20]: A = Point(-4, -1, -5)  
B = Point(4, -6, 7)  
C = Point(12, -11, 20)  
D = Point(20, -16, 31)  
K = Point(28, -17, 41)  
M = Point(92, -49, 133)  
Points = (A, B, C, D, K, M)
```

num = len(Points)

LINES = [Line(A, B)]

for i in range(num):

for j in range(i + 1, num):

Line1 = Line(Points[i], Points[j])

res = True

for Line2 in LINES:

if Line1.is_similar(Line2):

res = False

break

if res:

Lines.append(Line1)

Lines_eq = [Line0.equation() for Line0 in Lines]

display(*Lines_eq)

(5x + 8y + 28, -3x + 2z - 2)