

```
In [1]: import matplotlib.pyplot as plt
from sympy import Ellipse, Point, S, plot_implicit, Eq
```

Занятие 14

Алгебра

Кривые второго порядка на плоскости: эллипс

<https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%BB%D0%B8%D0%BF%D1%81> В Geometry есть класс эллипсов Ellipse, экземпляры класса задаются точкой - центром эллипса и горизонтальной и вертикальной полуосями

Пример 1

Построим эллипс Ellipse1: центр в начале координат, полуоси 13 и 5

```
In [2]: Ellipse1 = Ellipse(Point(0, 0),  hradius=13,  vradius=5)
Ellipse1
```

Out[2]:



Эллипс можно задать центром, горизонтальной полуосью и эксцентриситетом

```
In [3]: Ellipse2 = Ellipse(Point(3, 1),  hradius=13,  eccentricity=S(12)/13)
Ellipse2
```

Out[3]:



Атрибуты класса Ellipse

center

центр эллипса, точка пересечения его осей

hradius

горизонтальная полуось

vradius

вертикальная полуось

area

площадь

circumference

длина (дуги) эллипса (аналог длины окружности)

eccentricity

эксцентриситет отношение фокусного расстояния к горизонтальной оси (той, на которой расположены фокусы)

periapsis

перифокусное расстояние (минимальное расстояние от фокуса до точки на эллипсе)

apoapsis

апофокусное расстояние (максимальное расстояние от фокуса до точки на эллипсе)

focus_distance

фокусное расстояние - половина расстояния между фокусами

foci

фокусы

Пример 2.

Выведем на экран атрибуты эллипса Ellipse2

```
In [4]: props = {'center': Ellipse2.center, 'hradius': Ellipse2.hradius, 'vradius': Ellipse2.vradius, 'area': Ellipse2.area,
               'circumference': Ellipse2.circumference, 'eccentricity': Ellipse2.eccentricity, 'periapsis': Ellipse2.periapsis,
               'apoapsis': Ellipse2.apoapsis, 'focus_distance': Ellipse2.focus_distance, 'foci': Ellipse2.foci}
for key in props.keys():
    display(key, props[key])
```

'center'

Point2D(3,1)

'hradius'

13

'vradius'

5

'area'

65π

'circumference'

$52E\left(\frac{144}{169}\right)$

'eccentricity'

$\frac{12}{13}$

'periapsis'

1

'apoapsis'

25

'focus_distance'

12

'foci'

(Point2D(-9, 1), Point2D(15, 1))

Методы класса Ellipse

encloses_point(p) - является ли внутренней точкой эллипса точка p?

equation(x='x', y='y', _slope=None)- уравнение эллипса

необязательный аргумент _slope - наклон главной оси

Пример 3.

Выведем на экран, является ли внутренней точкой эллипса точка (0, 0), (3, 6) и (20, 20).

```
In [5]: display(Ellipse2.encloses_point(Point(0, 0)),
               Ellipse2.encloses_point(Point(3, 6)),
               Ellipse2.encloses_point(Point(20, 20)))
```

True

False

False

Заметим, что граничные точки, т.е. точки на самом эллипсе, не являются внутренними!

Пример 4.

Выведем на экран уравнение эллипса Ellipse2, а также уравнение этого эллипса, повернутого на угол с тангенсом 2.

```
In [6]: from sympy.abc import x, y
Ellipse2.equation(x='x', y='y')
```

Out[6]: $\left(\frac{x}{13}-\frac{3}{13}\right)^2+\left(\frac{y}{5}-\frac{1}{5}\right)^2-1$

Посмотрим, как изменится уравнение эллипса при повороте, для этого изменим наклон главной оси с 0 на 2:

```
In [7]: EQ = Ellipse2.equation(_slope=2)
EQ
```

Out[7]: $\frac{(-2x+y+5)^2}{125}+\frac{(x+2y-5)^2}{845}-1$

Пример 5.

Изобразим на графике эллипс, повернутый на угол с тангенсом 2.

Для того, чтобы изобразить эллипс на графике, нужны его переменные, извлечем их с помощью .free_symbols

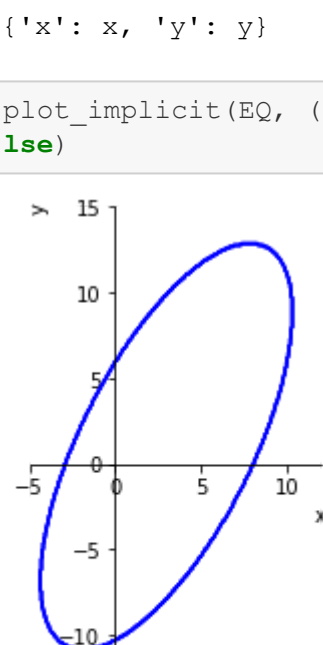
```
In [8]: symb_free = EQ.free_symbols
```

Обеспечим доступ к символам по имени, составив словарь dict_free по множеству set_free переменных уравнения эллипса:

```
In [9]: dict_free = {str(a): a for a in symb_free}
dict_free
```

Out[9]: {'x': x, 'y': y}

```
In [10]: plot_implicit(EQ, (dict_free['x'], -5, 12), (dict_free['y'], -15, 15), aspect_ratio=(1, 1), adaptive=False,
                      lse)
```



Out[10]: <sympy.plotting.plot.Plot at 0x2179012ee10>

Немного о plot_implicit

plot_implicit позволяет изобразить график одной кривой, заданной неявно с помощью уравнения.

Если нужно построить на одной плоскости два и более графика неявно заданных функций, можно поступить так:

вначале создаем все графики, указываем необязательный параметр show=False и запоминаем созданные с помощью plot_implicit объекты в переменные

затем объединяем объекты в один с помощью метода extend

Пример 6.

На графике изобразить эллипс Ellipse6 (с центром в (0, 0), осями 5 и 3) и его копии с наклоном 1, 2, -2, -1

```
In [11]: Ellipse6 = Ellipse(Point(0, 0), 5, 3)
eq6 = Ellipse6.equation(x='x', y='y')
dict_free = {str(a): a for a in eq6.free_symbols}
p = plot_implicit(Ellipse6.equation(x='x', y='y'), (dict_free['x'], -7, 6), (dict_free['y'], -6, 6),
                  aspect_ratio=(1, 1), show=False, adaptive=False)
for slope in (-2, -1, 1, 2):
    p.extend(plot_implicit(Ellipse6.equation(x='x', y='y', _slope=slope), aspect_ratio=(1, 1), show=False,
                           adaptive=False))
p.show()
```

