Урок 5. Исследование параболы с помощью SymPy

Мы уже говорили в предыдущем уроке о том, что очень полезно знать свойства функции, с которой работаешь, потому что это позволяет упростить вычисления и проще найти решение. Но мы далеко не всегда работаем с функциями, которые нам хорошо известны. Если свойства функции неизвестны, их нужно выяснить. Для этого математики придумали ряд методов, с помощью которых можно исследовать функцию.

Мы не будем применять эти методы вручную, мы используем для этого удобный инструмент, с которым мы уже познакомились, — библиотеку SymPy. Она позволит нам избежать длинных выкладок в тетрадке и быстро получить необходимые ответы.

В данном уроке мы попробуем «поиграть с коэффициентами» квадратичной функции и проверить упомянутое нами свойство чётности для функций, вершины которых находятся на оси OY.

Итак, импортируем библиотеку и модуль для построения графиков, вызываем команду для красивой отрисовки формул. Затем создаём переменную и простейшую квадратичную функцию $f(x) = x^2$:

```
In [1]:
from sympy import *
from sympy.plotting import plot
```

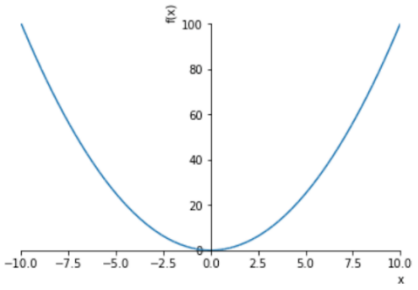
```
In [2]:
init_printing(use_unicode=False, wrap_line=False, no_global=True)
```

```
In [3]:
x = Symbol('x')
```

```
In [4]:
f = x ** 2
```

Давайте построим её график с помощью команды plot, это мы уже умеем:

```
In [7]:
plot(f)
```



Итак, мы построили параболу. Видим, что она получилась именно такой, как мы ожидали: симметричной, чётной и с ветвями, направленными вверх.

Давайте проверим чётность этой квадратичной функции, подставив в неё числа 100 и -100. По свойству чётности, значения функции должны получиться одинаковыми для противоположных чисел:

```
In [10]:
f.subs(x, 100)
```

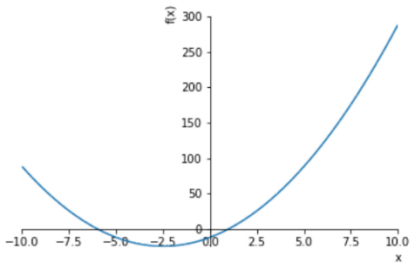
```
Out[10]: 10000
```

```
In [11]:
f.subs(x, -100)
```

```
Out[11]: 10000
```

Отлично, получились одинаковые значения. Похоже, что функция действительно чётная. Но не все квадратичные функции чётные. Симметричные все, а чётные — не все. Рассмотрим вот такую функцию: $f(x) = 2x^2 + 10x - 12$:

```
In [22]: 1 f2 = 2*x**2 + 10*x - 12
In [29]: 1 plot(f2)
```



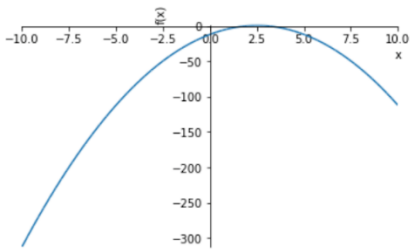
Сразу видно, что функция не симметрична относительно оси OY и не является чётной. Чётность ведь и выглядит как симметричность относительно вертикальной оси. Давайте для полной уверенности проверим чётность, подставив два противоположных числа в нашу функцию:

```
In [30]: 1 f2.subs(x, 200)
Out[30]: 81988
In [31]: 1 f2.subs(x, -200)
Out[31]: 77988
```

Видим, что результаты отличаются, значит, это точно не чётная функция. Обратите внимание, что тот факт, что функция не является чётной еще не значит, что она является нечётной! Функции бывают чётные, нечётные, и ни те, ни другие (в основном они такие и есть). Для нечётности требуется, чтобы для противоположных x значения функции были противоположны. Как мы видели, у нашей функции это не так, 20988 и 18988 не противоположны: для 20988 противоположным числом было бы -20988.

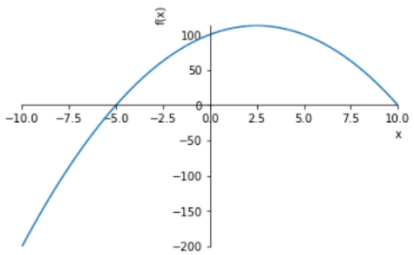
Теперь давайте попробуем менять коэффициенты квадратичной функции, чтобы посмотреть, как будет меняться график. Для начала поменяем коэффициент перед x^2. Получим функцию $f(x) = -2x^2 + 10x - 12$. Давайте построим её график:

```
In [11]: 1 f3 = -2*x**2 + 10*x - 12
         2 f3
Out[11]: −2x^2 + 10x − 12
In [12]: 1 plot(f3)
```



Действительно, свойство работает: смена знака перед x^2 повернула ветви графика вниз. Давайте проверим ещё одно свойство, заключающееся в том, что коэффициент *c* отвечает за высоту пересечения графика с осью OY. Сейчас наш график пересекает ось в точке с координатой y = -12. Давайте поднимем график вверх и сделаем так, чтобы он пересекал ось OY на высоте 100. Для этого нужно просто поменять *c* с -12 на 100:

```
In [13]: 1 f4 = -2*x**2 + 10*x + 100

In [14]: 1 plot(f4)
```



Отлично, у нас получилось! Мы теперь немного умеем исследовать функции с помощью SymPy.

Давайте расширим наш инструментарий. В предыдущих примерах при построении графиков мы не уточняли промежуток для x, SymPy сам его подбирал для нас. Но это не всегда удобно, а иногда и просто не подходит для задачи. Например, SymPy «ломается», если попросить его построить график функции, область определения которой — не вся числовая прямая, то есть не все значения x можно подставить в функцию. Например, корень из икс (√x). Мы помним, что в эту функцию нельзя подставлять отрицательные числа. Давайте попробуем немного похулиганить и попросим SymPy подставить в функцию корня -1.

Создадим функцию:

```
In [42]: 1 f = sqrt(x)
         2 f

Out[42]: √x
```

Может возникнуть закономерный вопрос, почему мы можем просто написать sqrt, почему python понял, чего мы от него хотим? Дело в том, что мы импортировали его, когда в начале написали:

```
In [ ]: 1 from sympy import *
```

Эта команда импортирует все базовые функции. Также она импортирует функцию натурального логарифма, то есть мы можем написать:

```
In [43]: 1 f = ln(x)
         2 f

Out[43]: log(x)
```

ln печатается как log, потому что это ещё одно его принятое обозначение, вы можете встретить и тот, и другой вариант.

Но мы отвлеклись от нашей хулиганской затеи подставить под корень отрицательное число. Вперёд:

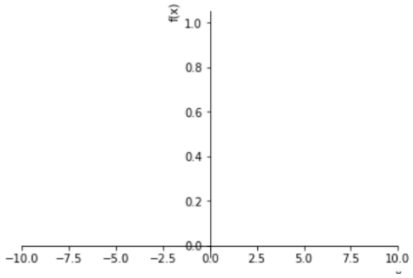
```
In [42]: 1 f = sqrt(x)
         2 f

Out[42]: √x

In [44]: 1 f.subs(x, -1)

Out[44]: iπ
```

Что же мы получили? Это так называемое комплексное число, оно получается, если подставить под корень отрицательное число. Его легко отличить по букве i, которая обозначает

мнимую единицу. В рамках нашего курса мы не будем рассматривать комплексные числа, поэтому для нас его появление эквивалентно знаку, что мы подставили в функцию что-то не то.

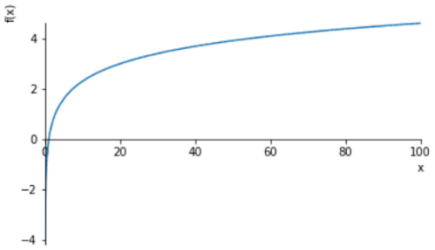
Что же будет с графиком? SymPy не будет его строить:

```
In [45]: 1 plot(f)
```



Он не знает, как изобразить корень отрицательного числа, поэтому так себя ведёт. Решение проблемы — указать корректный промежуток для переменной, входящий в область определения, например [0, 100]. Делается это достаточно просто: нужно передать в функцию plot ещё один аргумент типа кортеж следующего вида:

```
In [47]: 1 plot(f, (x, 0, 100))
```



В этом кортеже три значения: сама переменная, для которой мы задаём промежуток, начало промежутка и его конец.