

## Команда:

Дмитрий Воынов

Глеб Колонин

Никита Пестрецов

# Основные методы

## Метод половинного деления

```
def half_divide(self, f, start, end):
    try:
        while self.__len(abs(end - start)) < self.accuracy:
            middle = (start + end) / 2
            y = f(start) * f(middle)

            if y > 0:
                start = middle
            elif y < 0:
                end = middle
            else:
                return middle
        else:
            return middle
    except:
        return None
```

## Метод секущих

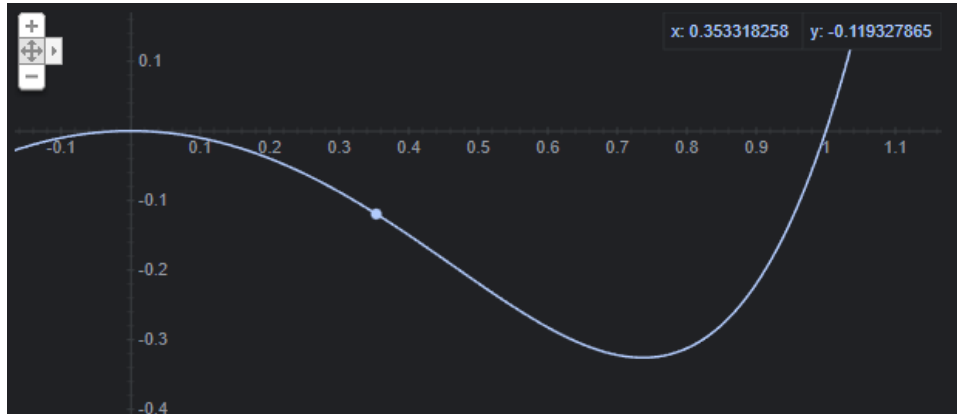
```
def secant(self, f, a, b):
    try:
        while abs(a - b) > self.eps:
            a = b - (b - a) * f(b) / (f(b) - f(a))
            b = a - (a - b) * f(a) / (f(a) - f(b))
        return b
    except:
        return None
```

В численном анализе метод секущих представляет собой алгоритм поиска корня который использует последовательность корней из секущих линий для лучшего приближения корня функции  $f$ . Метод секущей можно рассматривать как конечно-разностную аппроксимацию метода Ньютона. Однако этот метод был разработан независимо от метода Ньютона и предшествует ему более чем на 3000 лет.

## Тест 1

### Функция

$$f(x) = x^5 - x^2, x \in [0, 1]$$



### Время

Half divide 00.000170

Secant 00.000114

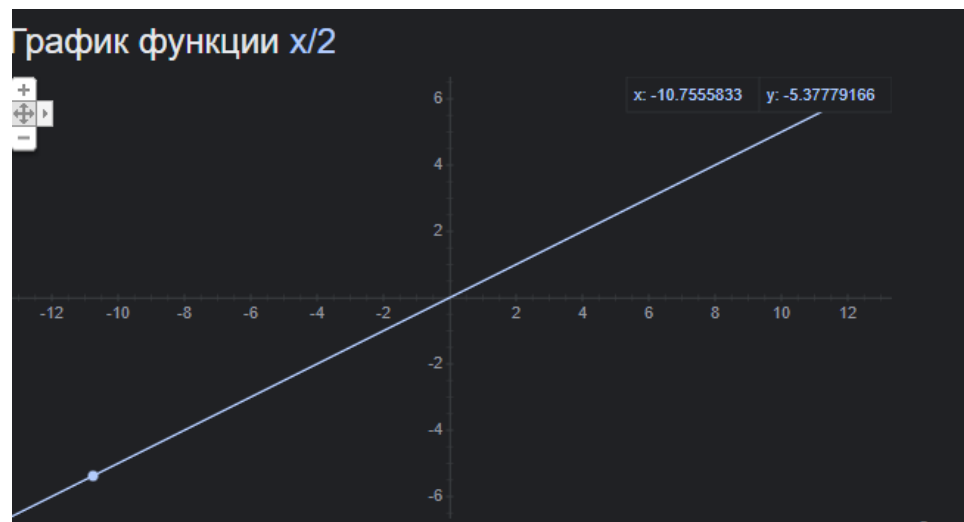
### Результат

Half divide: 0.5  
Secant: None

## Тест 2

### Функция

```
f = lambda x: x / 2  
a, b = -2, 2
```



### Время

Half divide	00.003362
Secant	00.001536

### Результат

```
Half divide: 0.0  
Secant: 0.0
```

## **Вывод**

Выбор метода нахождения корня функции зависит от требуемого временного ограничения. Метод половинного деления универсален и подходит для решения функций любого вида, но работает дольше.

[https://colab.research.google.com/drive/1y\\_81k7D6vHfIdE-Tui2\\_ubGNnnWgGQhs?usp=sharing](https://colab.research.google.com/drive/1y_81k7D6vHfIdE-Tui2_ubGNnnWgGQhs?usp=sharing)