Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
институт

Кафедра «Информатика»
кафедра

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ № 1

Метод Фибоначчи
Тема

Преподаватель			В. В. Тынченко
		Подпись, дата	Инициалы, Фамилия
Студент	КИ19-17/1Б, №031939174		А. К. Никитин
	Номер группы, зачетной книжки	Подпись, дата	Инициалы, Фамилия

Красноярск 2021

1 Постановка задачи

Разработать программу, реализующую метод Фибоначчи.

Найти безусловный экстремум функции, выбранной в соответствии с заданием, с использованием разработанной программы.

Функция: $f(x) = 7x^2 - 2x - 2 \to min.$ Интервал неопределённости [-6,6].

2 Описание метода

Метод, использующий числа Фибоначчи, позволяет наиболее эффективно достичь заданной точности в поиске экстремума функции $Q\,(u)$. Числа Фибоначчи определяются соотношением:

$F0 = F1 = 1$; $Fk = Fk\text{-}1 + Fk\text{-}2$; $k = 2, 3, \ldots$

При большом "k" отношение соседних чисел Фибоначчи близко к отношению "золотого сечения".

Этот метод делит интервал неопределенности не в постоянном соотношении, а в переменном и предполагает некоторое, вполне определенное, зависящее от , число вычислений значений функции $Q\,(u)$.

По заданному определяется количество вычислений n и соответствующее ему число Фибоначчи Fn, исходя из соотношения.

3 Исходные тексты программ

На листинге 1 и 2 представлен код программы, реализующих задание

Листинг 1 – Функции вычисления чисел Фибоначчи

```
def fibonacci_generator():
    a, b = 1, 1
    while True:
        yield a
        a, b = b, a + b



def fibonacci(n: int):
    if n < 1:
        raise Exception('Incorrect n!')
```

Продолжение листинга 1

```
    generator = fibonacci_generator()
    for _ in range(n):
        number = next(generator)
    return number



def fibonacci_list(n: int):
    a, b = 1, 1
    series = []
    for _ in range(n):
        series.append(a)
        a, b = b, a + b
    return series
```

Листинг 2 – Метод Фибоначчи

```
from sympy import Symbol

x = Symbol('x')

def find_iteration_number(l0: float, l: float):
    fib_gen = fibonacci_generator()

    n = 1
    while next(fib_gen) < (l0 / l):
        n += 1
    return n



def fibonacci_method(func, interval: tuple, l=0.1, epsilon=0.1, float_n=3):
    if len(interval) != 2 or interval[0] > interval[1]:
        raise Exception('Wrong interval!')

    a0 = interval[0]
    b0 = interval[1]
    N = find_iteration_number(b0 - a0, l)
    F = fibonacci_list(N)

    ak = a0
    bk = b0
    yk = ak + F[N - 2 - 1] / F[N - 1] * (bk - ak)
```

```
    zk = ak + F[N - 1 - 1] / F[N - 1] * (bk - ak)

    for k in range(N - 3):
        fyk = func.subs(x, yk)
        fzk = func.subs(x, zk)

        if fyk <= fzk:
            bk = zk
            zk = yk
            yk = ak + F[N - k - 3 - 1] / F[N - k - 1 - 1] * (bk - ak)
        else:
            ak = yk
            yk = zk
            zk = ak + F[N - k - 2 - 1] / F[N - k - 1 - 1] * (bk - ak)

    yn = zk
    zn = zk + epsilon

    if func.subs(x, yn) <= func.subs(x, zn):
        return round(ak, float_n), round(zn, float_n)
    else:
        return round(yn, float_n), round(bk, float_n)


def main():
    f = 7 * (x ** 2) - 2 * x - 2
    interval = fibonacci_method(f, (-6, 6))
    print('Интервал неопределенности:', interval)
    print('Точка экстремума:', round((interval[0] + interval[1]) / 2, 3))
    print('Экстремум:', round(f.subs(x, (interval[0] + interval[1]) / 2), 3))


if __name__ == '__main__':
    main()
```

4 Исследование влияния параметров метода на точность и скорость нахождения решения

На рисунках 1 и 2 представлен график изменения точности (delta) и скорости нахождения решения (iterations_number) от параметра l и ε соответственно.

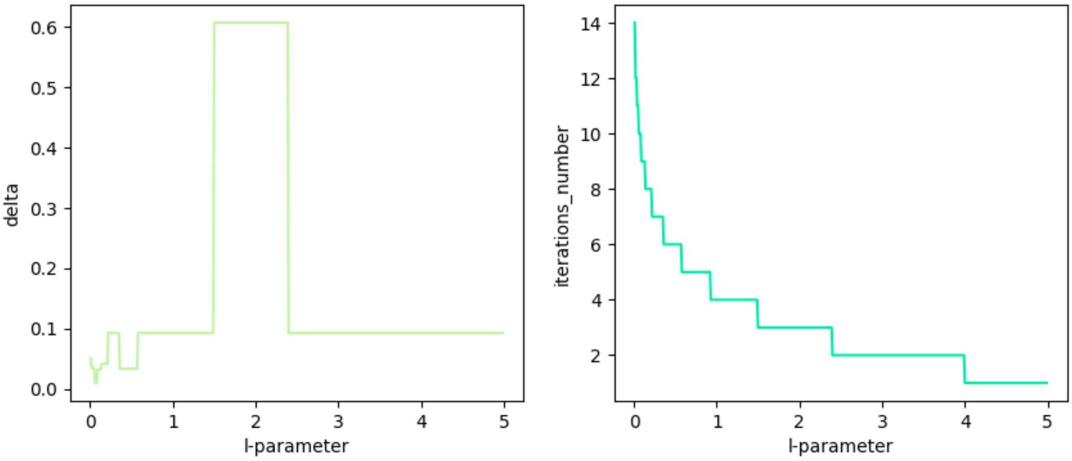

Рисунок 1 – Зависимость точности и скорости от параметра l

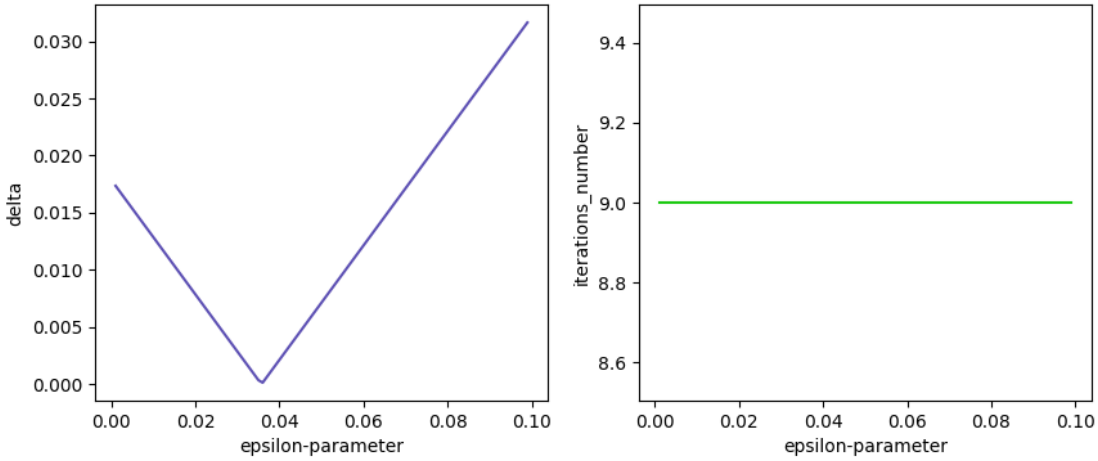

Рисунок 2 – Зависимость точности и скорости от параметра ε

Из графиков можно заключить, что параметр l показывает свою наибольшую результативность примерно в области до 1, и что чем меньше параметр l, тем больше времени необходимо алгоритму для работы. Параметр ε же никак не влияет на количество итераций алгоритма и имеет линейную зависимость с разницей между предполагаемым и реальным значением.

5 Вывод

В результате данной работы был реализован и проанализирован метод Фибоначчи для поиска локального минимум функции в заданном интервале.

Также были проанализированы гиперпараметры метода и их влияние на работу функции.