Федеральное государственное автономное

образовательное учреждение

Высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

|  |
| --- |
| Институт космических и информационных технологий |
| институт |
| Информатика |
| кафедра |

**ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ**

|  |
| --- |
| Метод Фибоначчи |
| тема |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Преподаватель | |  |  |  | В. В. Тынченко |
|  | |  | подпись, дата |  | инициалы, фамилия |
| Студент | КИ21-17/1Б, 032156940 |  |  |  | Н.А. Самарин |
|  | номер группы, зачётной книжки |  | подпись, дата |  | инициалы, фамилия |

Красноярск 2023

**СОДЕРЖАНИЕ**

1 Задание............................................................................................................... 3

2 Вариант задания................................................................................................ 3

3 Описание метода............................................................................................... 3

4 Реализация метода............................................................................................ 3

5 Анализ результатов.......................................................................................... 5

6 Вывод................................................................................................................. 6

**1 Задание**

Разработать программу, реализующую метод Фибоначчи. Найти  
безусловный экстремум функции, выбранной в соответствии с заданием, с  
использованием разработанной программы. Вариант задания необходимо  
выбрать по номеру в списке группы.

**2 Вариант задания**

24. f(x) = 2 \* e^5x − 6x + 2 \* x^3 → min. Интервал неопределённости  
[-6,6].

**3 Описание метода**

Метод Фибоначчи — это улучшение реализации поиска с помощью  
золотого сечения, служащего для нахождения минимума/максимума функции.  
Подобно методу золотого сечения, он требует двух вычислений функции на  
первой итерации, а на каждой последующей только по одному.

Метод основан на последовательности чисел Фибоначчи Fv, которая определяется следующим образом: Fv = Fv−1 + Fv−2, v = 1,2,3,…, F0 = F1 = 1

Предварительный этап. Выбрать допустимую конечную длину интервала  
неопределенности l > 0 и константу различимости ϵ. Пусть задан начальный  
интервал неопределенности (b1 − a1). Выбрать общее число вычислений  
функции n так, чтобы Fn > (b1 − a1) / l . Положить λ1 = a1 + (Fn − 2 / Fn) \* (b1 −  
a1), μ1 = a1 + (Fn−1 / Fn) \* (b1 − a1). Вычислить f(λ1) , f(μ1) , положить k = 1 и  
перейти к основному этапу.

Основной этап.

Первый шаг. Если f(λk) > f(μk) , то перейти ко второму шагу, в противном  
случае – к третьему шагу.

Второй шаг. Положить ak+1 = λk, bk+1 = bk. Затем положить λk+1 = μk,  
μk+1 = ak+1 + (Fn−k−1 / Fn−k)\* (bk+1 − ak+1). Если k = n−2, то перейти к  
пятому шагу, в противном случае вычислить f(μk+1) и перейти к четвертому  
шагу.

Третий шаг. Положить ak+1 = ak, bk+1 = μk, μk+1 = λk, λk+1 = ak+1 +  
(Fn−k−2 / Fn−k) \* (bk+1 − ak+1). Если k = n−2, то перейти к пятому шагу, в  
противном случае f(λk+1) и перейти к четвертому шагу.

Четвертый шаг. Заменить k на k + 1 и перейти к первому шагу.

Пятый шаг. Положить λn = λn−1, μn = λn + ϵ. Если f(λn) = f(μn), то  
положить an = λn, bn = bn−1. В противном случае (если f(λn) < f(μn)),  
положить an = an−1, bn = μn.

Конец: оптимальное решение содержится в интервале [an, bn]. Например,  
в середине интервала.

**4 Реализация метода**

Ниже в листинге 1 представлен текст python программы, реализующей

метод Фибоначчи для поиска минимума функции.

Листинг 1 – Текст python программы

import math  
def fibonacci(n):  
 a = 0  
 b = 1  
 for \_ in range(n):  
 a, b = b, a + b  
 return a  
def fibonacci\_method(func, a, b, l, eps=1e-9):  
 n = 1  
 while fibonacci(n) < abs(b - a) / l:  
 n += 1  
 fibonacci\_values = []  
 for i in range(1, n + 2):  
 fibonacci\_values.append(fibonacci(i))  
 z = a + (fibonacci\_values[n - 2] / fibonacci\_values[n]) \* (b - a)  
 z\_func = func(z)  
 y = a + (fibonacci\_values[n - 1] / fibonacci\_values[n]) \* (b - a)  
 y\_func = func(y)  
 calc\_num = 2  
 for i in range(1, n + 1):  
 if z\_func > y\_func:  
 a = z  
 z = y  
 z\_func = y\_func  
 y = a + (fibonacci\_values[n - i - 1] / fibonacci\_values[n - i]) \* (b  
 - a)  
 if i != n - 2:  
 y\_func = func(y)  
 calc\_num += 1  
 else:  
 b = y  
 y = z  
 y\_func = z\_func  
 z = a + (fibonacci\_values[n - i - 2] / fibonacci\_values[n - i]) \* (b  
 - a)  
 if i != n - 2:  
 z\_func = func(z)  
 calc\_num += 1  
 if i == n - 2:  
 break  
 y = z + eps  
 y\_func = func(y)  
 z\_func = func(z)  
 calc\_num = calc\_num + 2  
 if y\_func == z\_func:  
 a = z  
 else:  
 b = y  
 return (a + b) / 2, calc\_num  
def func(x):  
 return (2 \* math.exp(5 \* x)) - (6 \* x) + (2 \* math.pow(x, 3))  
a, b = -6, 6  
l = 0.0001  
eps = 1e-9  
minimum, calc\_num = fibonacci\_method(func, a, b, l, eps)  
print("Параметры метода: l -", l, ", epsilon -", eps)

Окончание листинга 1

print("Минимум функции:", minimum)  
print("Количество вычислений функции:", calc\_num)

**5 Анализ результатов**

Для начала найдём реальный минимум функции, для этого произведём  
следующие действия:

Находим первую производную функции: y′=6·x2 + 10·e5·x - 6 Вычисляем  
значения функции на концах интервала f(-0.104)=1.811 f(-6)=-396  
f(6)=21372949163445

Приравниваем ее к нулю: 6·x2 + 10·e5·x - 6=0

Получаем корни: x1=-0.104

Вычисляем значения функции:

f(-0.104) = 1.811

f(-6) = -396

f(6) = 21372949163445

В результате реальный минимум функции на интервале [-6; 6] находится  
в точке -6.

Теперь получим значения работы метода Фибоначчи при разных  
значениях параметров. Результаты представлены на рисунках ниже.

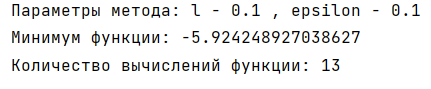


Рисунок 1 – Результат №1

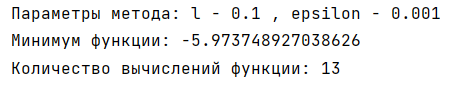


Рисунок 2 – Результат №2

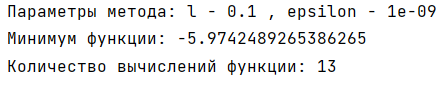


Рисунок 3 – Результат №3

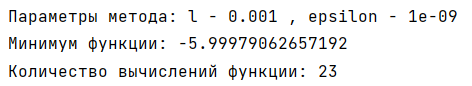


Рисунок 4 – Результат №4

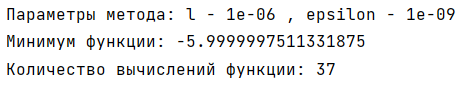


Рисунок 5 – Результат №5

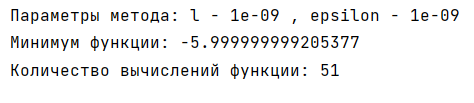


Рисунок 6 – Результат №6

Также построим графики зависимости количества вычислений целевой функции и отклонения от реального минимума для параметра l как наиболее влияющего на точность.

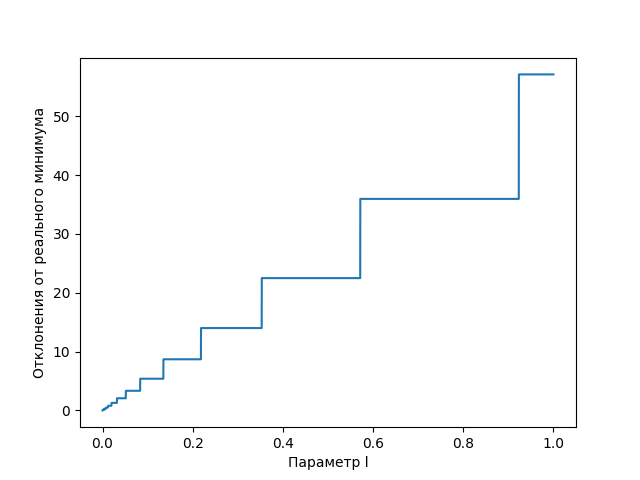


Рисунок 7 – График количества вычислений целевой функции

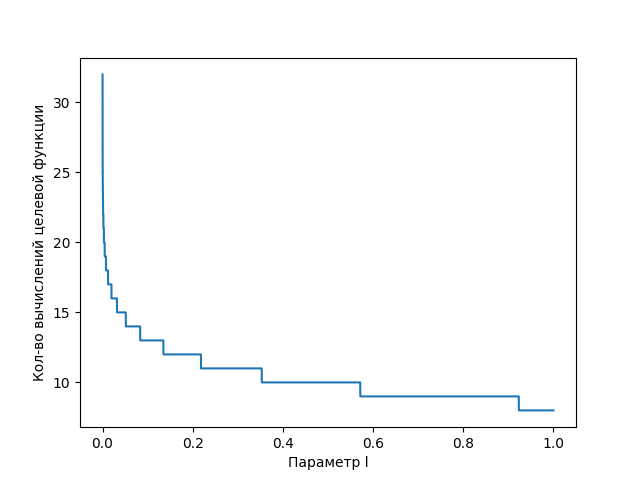


Рисунок 8 – График отклонения от реального минимума

В результате можно сказать, что найденные минимумы функции близки к  
реальному. При этом при приближении параметра ϵ к нулю  
увеличивается точность метода, при этом не увеличивая время нахождения  
решения (количество вычислений функции). При приближении параметра l к  
нулю увеличивается точность метода, но при этом увеличивается время  
нахождения решения.

**6 Вывод**

При выполнении задания был успешно реализован метод Фибоначчи,  
результаты работы метода сравнены с реальным и близки к нему, исследована  
зависимость работы метода от значений его параметров.