МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Липецкий государственный технический университет

Факультет автоматизации и информатики

Домашняя работа №2

по математическому программированию

Студент Станиславчук С. М.

Группа АС-21-1

Руководитель Качановский Ю. П.

Липецк 2023 г.

Содержание

1. Задание

Вариант: 5

$f(x) = x^4-14x^3+60x^2-70x$,

[4, 8]

метод половинного деления

2. Теория

Известен первоначальный интервал $[a_0, b_0]$ и необходимо найти минимум данной функции на заданном отрезке с некоторой точностью E

1. Положим $x_m = (a_0 + b_0) / 2$ и вычисляем значения функции $f(x_m)$
2. Вычисляем точки согласно следующим соотношениям:

   $x_1 = a_0 + ((b_0 - a_0) / 4)$ $x_2 = b_0 - ((b_0 - a_0) / 4)$

   Затем вычисляем значения функции: $f(x_1)$ и $f(x_2)$. Заметим, что точки x_1, x_m, x_2 делят интервал $[a_0, b_0]$ на 4 равные части.

3. Сравниваем $f(x_1)$ и $f(x_m)$. Если $f(x_1) < f(x_m)$ исключаем подынтервал $(x_m, b_0]$, положив $b_1 = x_m$. Средней точкой нового интервала поиска становится точка x_1. Следовательно, необходимо положить $x_m = x_1, a_1 = a_0$. Переходим к шагу 5. Если $f(x_1) >= f(x_m)$, переходим к шагу 4.
4. Сравниваем $f(x_2)$ и $f(x_m)$. Если $f(x_2) < f(x_m)$ исключаем подынтервал $[a_0, x_m)$, положив $a_1 = x_m$. Средней точкой нового интервала поиска становится точка x_2. => Необходимо положить $x_m = x_2$, $b_1 = b_0$. переходим к шагу
5. Если $f(x_2) >= f(x_m)$, то исключаем сразу 2 подынтервала $[a_0, x_1)$ и $(x_2, b_0]$, положив $a_1 = x_1$ и $b_1 = x_2$. x_m продолжает
6. оставаться средней точкой нового интервала. Переходим к шагу 5.

7. Проверяем критерий останова. Если длина оставшегося интервала неопределенности $[a_k, b_k]$ на текущей итерации k становится меньше заданной точности $|b_k - a_k| <= E$

Поиск заканчиваем. Минимум функции равен $f(x_m)$. В противном случае возвращаемся к шагу 2 и начинаем новую итерацию3. Решение

Для поиска точек минимума функции с использованием метода бисекции сначала определим нашу функцию f(x):

```
def f(x):
    return x**4 - 14*x**3 + 60*x**2 - 70*x
```

Теперь мы можем использовать метод половинного деления для нахождения точки минимума функции на заданном интервале :

```
def find_minimum_bisection_derivative(a, b, tol=1e-5):
    iteration = 0

    while (b - a) > tol:
        # Шаг 1: Вычисление средней точки xm и двух дополнительных точек x1 и x2
        xm = (a + b) / 2
        x1 = a + (b - a) / 4
        x2 = b - (b - a) / 4

        # Шаг 2: Вычисление значений функции в трех точках: x1, xm и x2
        f_a = f(x1)
        f_m = f(xm)
        f_b = f(x2)

        # Шаг 3: Сравнение значений функции и выбор перспективного подинтервала
        if f_a < f_m and f_a < f_b:
            b = xm
            xm = x1
        elif f_b < f_m:
            a = xm
            xm = x2
        else:
            a = x1
            b = x2

        iteration += 1
        print(f"Iteration {iteration}: x = {xm:.6f}, f(x) = {f(xm):.6f}")
```

```
    minimum_x = xm

    return minimum_x
```

Определим границы нашего интервала и вычислим на нем точку минимума (применим нашу функцию):

```
a = 4

b = 8


minimum = find_minimum_bisection_derivative(a, b)
```

Результат программы при толерантности = 1e^(-5) (повезло, конечно, с точкой, ведь минимум функции находится очень близко к центру интервала, полученной сразу с первой итерации) :

```
Iteration 1: x = 6.000000, f(x) = 12.000000
Iteration 2: x = 6.000000, f(x) = 12.000000
Iteration 3: x = 6.000000, f(x) = 12.000000
Iteration 4: x = 6.000000, f(x) = 12.000000
Iteration 5: x = 5.937500, f(x) = 11.966324
Iteration 6: x = 5.968750, f(x) = 11.960633
Iteration 7: x = 5.953125, f(x) = 11.957959
Iteration 8: x = 5.960938, f(x) = 11.957902
Iteration 9: x = 5.957031, f(x) = 11.957584
Iteration 10: x = 5.957031, f(x) = 11.957584
Iteration 11: x = 5.957031, f(x) = 11.957584
Iteration 12: x = 5.957031, f(x) = 11.957584
Iteration 13: x = 5.957275, f(x) = 11.957584
Iteration 14: x = 5.957153, f(x) = 11.957584
Iteration 15: x = 5.957214, f(x) = 11.957583
Iteration 16: x = 5.957184, f(x) = 11.957583
Iteration 17: x = 5.957199, f(x) = 11.957583
Iteration 18: x = 5.957191, f(x) = 11.957583
Iteration 19: x = 5.957195, f(x) = 11.957583
Точка минимума функции на отрезке [4.0, 8.0]: x = 5.957195
Минимальное значение функции: 11.957583
>
```

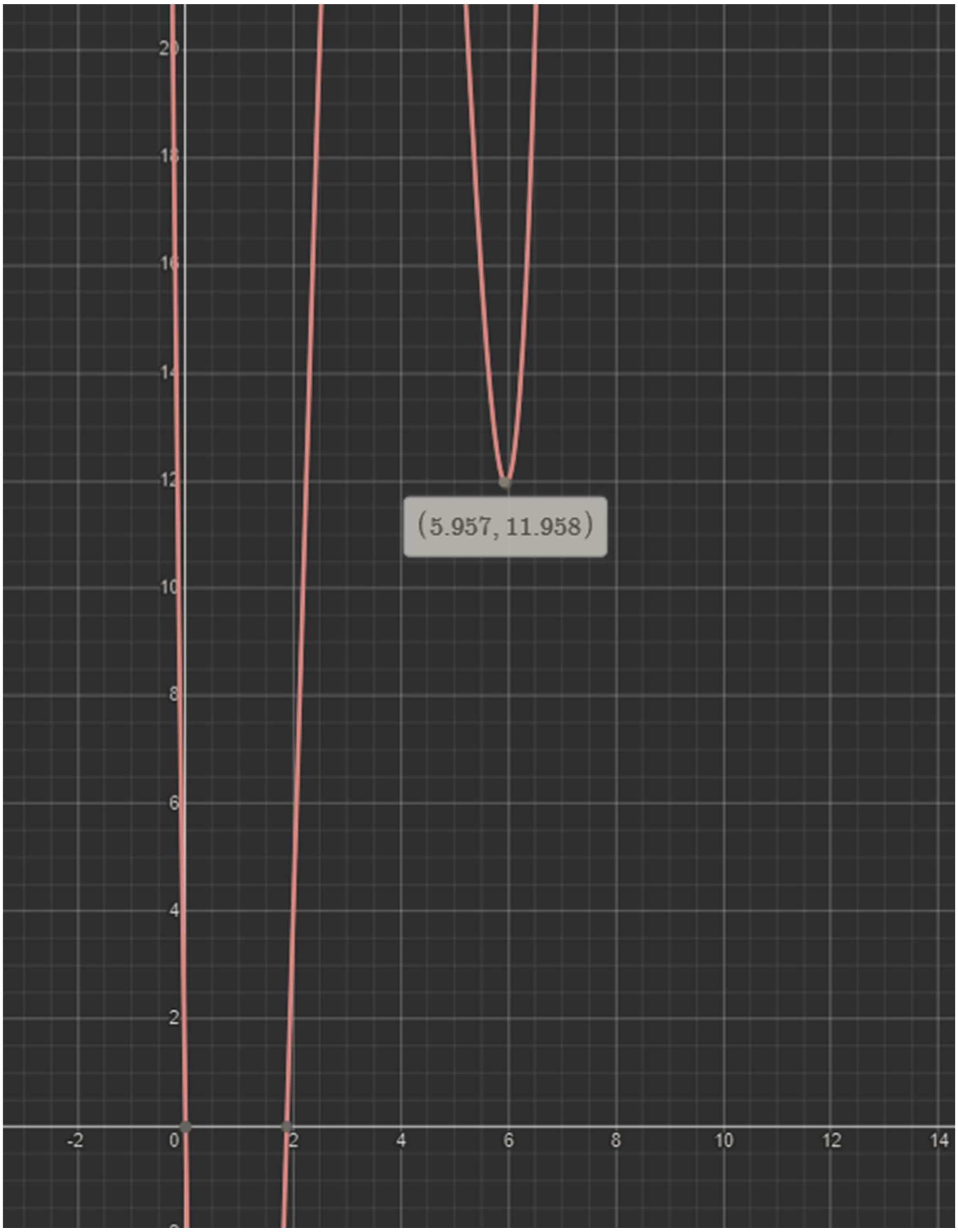

Рисунок 1. График функции y = x^{4}-14x^{3}+60x^{2}-70x

(сайт Desmos.com)

Как видно из графика, результаты программы сошлись с реальными. Действительно, округлив, получим значение точки минимума: 5.957 при значении функции 11.958. Это и будет наш локальный минимум на интервале [4:8].

4. Ответ: минимум исходной функции на интервале [4, 8] найден в точке x = 5.957191 методом бисекции.