Министерство науки и высшего образования РФ ФГАОУ ВПО

Национальный исследовательский технологический университет «МИСИС»

Институт Информационных технологий и компьютерных наук (ИТКН)

Кафедра автоматизированных систем управления (АСУ)

Отчет по лабораторной работе №1

по дисциплине «Методы оптимизаций»

Вариант 5

Выполнил:

Студент: Безыкорнов Н.Б.

         Группа: БИВТ-20-4

Проверил:

Лычев А.В.

Москва, 2023

**Цель работы:**

Приобретение практических навыков для решения задач одномерной минимизации численными методами.

**Постановка задачи:**

Вариант 5

Требуется найти безусловный минимум функции одной переменной y = f(x) на отрезке [a, b], где функция является унимодальной. То есть найти такую точку x\* ∈ [a, b], что f(x\*) = min на x∈[a,b] f(x).

По условию использовались методы под номерами 1, 3 и 6. Т.е. метод равномерного поиска, метод деления отрезка пополам и метод Фибоначчи.

F(x) = (x+5)^4

a = -6

b = -2

Выполнение работы:

Изображение выглядит как диаграмма

Автоматически созданное описание

Рис. 1 – график функции

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название метода | Число итераций | Количество вычислений функций | Найденное решение | Значение функции |
| Метод равномерного поиска | 4000 | 4000 | -5.0 | 0 |
| Метод деления отрезка пополам | 12 | 48 | -4.9995117 |  |
| Метод Фибоначчи | 17 | 19 | -5.0002391 |  |

1. Метод равномерного поиска

def f(x):  
 global q  
 q += 1  
 return (x + 5) \*\* 4  
  
  
def ravnomerniySearch():  
 global a, b, e  
 N, k = int((b - a) / e), 0  
 fminlist = [f(a + (i \* (b - a)) / N) for i in range(N)]  
 minn = min(fminlist)  
 num = fminlist.index(minn)  
 answer = ((a + (num - 1) \* (b - a) / N) + (a + (num + 1) \* (b - a) / N)) / 2  
 return answer, k  
  
  
a, b, e, q = -6, -2, 0.001, 0  
  
answer, k = ravnomerniySearch()  
print("Минимум", answer)  
print("Итерации", k)  
print("Вычисления функции", q)  
print("Значение функции в минимуме", f(answer))

Листинг 1 – Метод равномерного поиска

Изображение выглядит как текст

Автоматически созданное описание

Рис. 2 – результат работы программы

1. Метод деления отрезка пополам

def f(x):  
 global q  
 q += 1  
 return (x + 5) \*\* 4  
  
  
def binarySearch(f, a, b, tol=0.001):  
 xmid = (a + b) / 2  
 L = b - a  
 iter\_count = 0  
 while L > tol:  
 x1, x2 = a + L / 4, b - L / 4  
 fx1, fmid, fx2 = f(x1), f(xmid), f(x2)  
 if fx1 < fmid:  
 b = xmid  
 xmid = x1  
 elif fx1 >= fmid:  
 if fx2 < fmid:  
 a = xmid  
 xmid = x2  
 else:  
 a = x1  
 b = x2  
 L = b - a  
 iter\_count += 1  
 print("Кол-во итераций ", iter\_count)  
 return xmid  
  
  
q = 0  
answer = binarySearch(f, -6, -2)  
print("Минимум", answer)  
print("Вычисления функции", q)  
print("Значение функции в минимуме", f(answer))

Листинг 2 – метод деления отрезка пополам

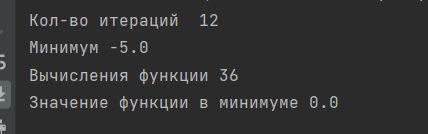


Рис. 3 – результат работы программы

1. метод Фибоначчи
2. def fibonacci\_search(f, a, b, eps):  
    def fibonacci(n):  
    if n < 1:  
    return 0  
    elif n == 1 or n == 2:  
    return 1  
    else:  
    return fibonacci(n - 1) + fibonacci(n - 2)  
     
    n = 1  
    while (b - a) / fibonacci(n + 2) >= eps:  
    n += 1  
     
    x1 = a + fibonacci(n) / fibonacci(n + 2) \* (b - a)  
    x2 = a + fibonacci(n + 1) / fibonacci(n + 2) \* (b - a)  
    f1 = f(x1)  
    f2 = f(x2)  
    print("кол-во итераций", n)  
    for i in range(1, n):  
    if f1 < f2:  
    b = x2  
    x2 = x1  
    f2 = f1  
    x1 = a + fibonacci(n - i) / fibonacci(n - i + 2) \* (b - a)  
    f1 = f(x1)  
    else:  
    a = x1  
    x1 = x2  
    f1 = f2  
    x2 = a + fibonacci(n - i + 1) / fibonacci(n - i + 2) \* (b - a)  
    f2 = f(x2)  
     
    return (a + b) / 2  
     
     
   def f(x):  
    global q  
    q += 1  
    return (x + 5) \*\* 4  
     
     
   a = -6  
   b = -2  
   eps = 0.001  
   q = 0  
     
   min\_x = fibonacci\_search(f, a, b, eps)  
   min\_f = f(min\_x)  
     
   print("Минимум", min\_f, "в точке", min\_x)  
   print("Вычисления функции", q)

листинг 3 - метод Фибоначчи

Изображение выглядит как текст

Автоматически созданное описание

Рис. 4 – результат работы программы

Сравнительная характеристика:

Методы бинарного поиска и Фибоначчи обеспечивают более быстрое нахождение минимумов и меньшее количество вычислений функции, в то время как метод равномерного поиска является менее эффективным, но может быть использован в случаях, когда нет возможности определить монотонность функции или при скорости изменения функции.

Вывод:

Я ознакомился с некоторыми методами нахождения минимумов функций.