МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования

**«Московский технический университет связи и информатики»**

Кафедра «Информатика»

**Отчет по заданию №6**

**по дисциплине**

**«Численные методы»**

Выполнил: студент гр. БЭИ2202

Васильев В. А.

Вариант 8.

Проверил: доц. каф. «Информатика»

Мацкевич А. Г.

Москва, 2023 г.

1. **Задание для решения задачи одномерной оптимизации:**

Функция, для которой необходимо найти минимум –

;

1. **Исследование задания:**

* график функции , построенный на достаточно большом отрезке ОДЗ функции:

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

* выберем по построенному графику функции начальный отрезок неопределенности (отрезок, содержащий точку минимума): отрезок [1.5;2.3];
* проверим выполнение аналитического условия унимодальности функции на выбранном отрезке

Графики производных представлены ниже

Изображение выглядит как текст, линия, снимок экрана, График

Автоматически созданное описание

Изображение выглядит как текст, линия, График, снимок экрана

Автоматически созданное описание

Как видно первая производная имеет корень на отрезке и монотонна возрастает, а вторая не пересекает ось Х, следовательно условия унимодальности функции выполняются на заданном отрезке

### *Метод золотого сечения*

### **Результаты выполнения функции, реализующей метод золотого сечения, и длина отрезка, содержащего точку минимума после трех итераций.**

Для проведения расчетов по методу золотого сечения следует создать сценарий и выполнить расчеты 3-х итераций.

n = 15

a = 1.5

b = 2.3

phi = (1 + np.sqrt(5)) / 2

df\_gs = pd.DataFrame(columns=['a', 'b', 'x1', 'x2', 'y1', 'y2'])

delta = []

x1, x2 = b - (b - a) / phi, a + (b - a) / phi

y1, y2 = f(x1), f(x2)

for i in range(n):

  df\_gs.loc[len(df\_gs.index)] = [a, b, x1, x2, y1, y2]

  # print(i, a, b, x1, x2, y1, y2, abs(x1-x2), sep='\t')

  if y1 >= y2:

    a = x1

    x1 = x2

    x2 = a + (b - a) / phi

    y1 = y2

    y2 = f(x2)

  else:

    b = x2

    x2 = x1

    x1 = b - (b - a) / phi

    y2 = y1

    y1 = f(x1)

  delta.append(abs(a-b))

df\_gs['delta'] = delta

print((a+b)/2)

df\_gs

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Как видно длина отрезка равна

Теоретическая величина погрешности для метода золотого сечения определяется длиной конечного отрезка неопределенности после **N** итераций . Отсюда имеем ,



Длина отрезка равна **0.000086** при расчете, используя ЯП Python. То есть, расчет совпадает с теоретической оценкой.

### *Метод дихотомии*

**3. Результаты выполнения функции, реализующей метод золотого сечения и длина отрезка, содержащего точку минимума после трех итераций.** Значение параметра **d** метода дихотомии выберем равным **0.01.**

Для проведения расчетов по методу дихотомии следует создать сценарий и выполнить расчеты 3-х итераций.

Для метода дихотомии длина отрезка неопределенности после трех итераций равна

1. **Число итераций, необходимых для локализации точки минимума и Е=10-4**

Теоретическая величина погрешности для метода дихотомии определяется длиной конечного отрезка неопределенности после N итераций: . Отсюда, принимая во внимание, что , можно определить соответствующее число итераций: .

Если точность Е=0.0001, а параметр метода d==0.00002, то получим



В результате расчета на **ЯП Python** при N=13 длина отрезка равна 0.000089**.** Точность достигнута при N=13, т. е. расчет совпадает с теоретической оценкой.

d = 0.00002

n = 15

a = 1.5

b = 2.3

i = 0

df\_d = pd.DataFrame(columns=['a', 'b', 'x1', 'x2', 'y1', 'y2'])

delta = []

for \_ in range(n):

  x1, x2 = (a + b) / 2 - d, (a + b) / 2 + d

  y1, y2 = f(x1), f(x2)

  df\_d.loc[len(df\_d.index)] = [a, b, x1, x2, y1, y2]

  # print(i, a, b, x1, x2, y1, y2, ((b - a - 2\*d) / 2\*\*i) + 2\*d, sep='\t')

  if y1 > y2:

    a = x1

  else:

    b = x2

  delta.append(b - a)

print((x1 + x2) / 2)

df\_d['delta'] = delta

df\_d