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

Кулешов А. С.

Вариант 16.

Проверил: доц. каф. «Информатика»

Мацкевич А. Г.

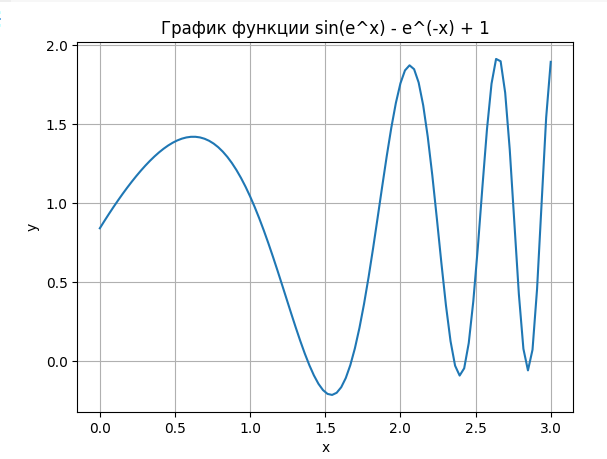
Москва, 2023 г.

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

* функция, для которой необходимо найти минимум –;

1. **Исследование задания:**

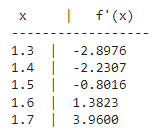
* график функции , построенный на достаточно большом отрезке ОДЗ функции:



* выберем по построенному графику функции начальный отрезок неопределенности (отрезок, содержащий точку минимума): отрезок [1.3;1.7];
* проверим выполнение аналитического условия унимодальности функции на выбранном отрезке

Решение достигается, при x=1.237, x=1.861 и x 1.861, как видно ни одно из решений не входит в отрезок [1.3;1.7].

* Значениясведем в следующую таблицу:

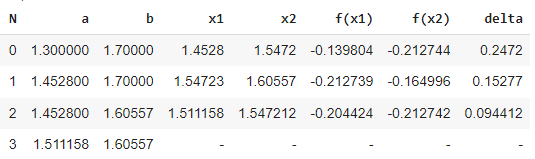


На отрезке [1.3;1.7**]** функция  монотонно возрастает, следовательно, функция f(x) - на выбранном отрезке унимодальная

### *Метод золотого сечения*

1. **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.



Как видно длина отрезка равна

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

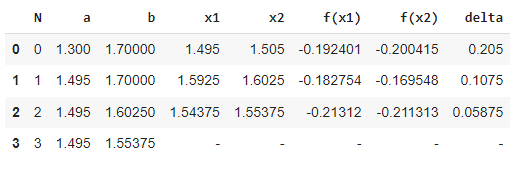


Длина отрезка равна **0.000069( )** при расчете, используя ЯП Python(N=18**)** . Точность достигнута при N=18. То есть, расчет совпадает с теоретической оценкой.

### *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 0.000069**.** Точность достигнута при N=13,т. е. расчет совпадает с теоретической оценкой

1. **Решение задачи оптимизации с использованием ЯП Python**

import numpy as np

import matplotlib.pyplot as plt

x = np.linspace(0, 3, 100)

y = np.sin(np.exp(x)) - np.exp(-x) + 1

plt.plot(x, y)

plt.xlabel('x')

plt.ylabel('y')

plt.title('График функции sin(e^x) - e^(-x) + 1')

plt.grid(True)

plt.show()

x = np.arange(1.3, 1.8, 0.1)

y = np.exp(-x) + np.exp(x) \* np.cos(np.exp(x))

# Построение таблицы значений для производной

print(" x     |   f'(x)   ")

print("------------------")

for i in range(len(x)):

    print(f"{x[i]:.1f}  |  {y[i]:.4f}")

# gold sech

def f(x):

  return np.sin(np.exp(x)) - np.exp(-x) + 1

data = {

    'N': [],

    'a': [],

    'b': [],

    'x1': [],

    'x2': [],

    'f(x1)': [],

    'f(x2)': [],

    'delta': []

}

k1 = 0.382

k2 = 0.618

a = 1.3

b = 1.7

for i in range(4):

  data['N'].append(i)

  data['a'].append(a)

  data['b'].append(b)

  x1 = a + k1 \* (b - a)

  data['x1'].append(x1)

  x2 = a + k2 \* (b - a)

  data['x2'].append(x2)

  fx1 = f(x1)

  data['f(x1)'].append(fx1)

  fx2 = f(x2)

  data['f(x2)'].append(fx2)

  if (fx1 > fx2):

    a = x1

  else:

    b = x2

  data['delta'].append(b - a)

data['delta'][-1] = "-"

data['f(x1)'][-1] = "-"

data['f(x2)'][-1] = "-"

data['x1'][-1] = "-"

data['x2'][-1] = "-"

import pandas as pd

# Создаем DataFrame из словаря

df = pd.DataFrame(data)

styled\_df = df.style.set\_properties(\*\*{'background-color': 'black', 'color': 'white'})

# Выводим таблицу значений

df

# dither

data = {

    'N': [],

    'a': [],

    'b': [],

    'x1': [],

    'x2': [],

    'f(x1)': [],

    'f(x2)': [],

    'delta': []

}

d = 0.00002

a = 1.3

b = 1.7

for i in range(20):

  data['N'].append(i)

  data['a'].append(a)

  data['b'].append(b)

  x1 = (a+b - d)/2

  data['x1'].append(x1)

  x2 = (a+b + d)/2 # Кажется в методичке очепятка...

  data['x2'].append(x2)

  fx1 = f(x1)

  data['f(x1)'].append(fx1)

  fx2 = f(x2)

  data['f(x2)'].append(fx2)

  if (fx1 > fx2):

    a = x1

  else:

    b = x2

  data['delta'].append(b - a)

data['delta'][-1] = "-"

data['f(x1)'][-1] = "-"

data['f(x2)'][-1] = "-"

data['x1'][-1] = "-"

data['x2'][-1] = "-"

import pandas as pd

# Создаем DataFrame из словаря

df = pd.DataFrame(data)

styled\_df = df.style.set\_properties(\*\*{'background-color': 'black', 'color': 'white'})

# Выводим таблицу значений

df