Домашнее задание №2 по дисциплине «Методы оптимизации»

Выполнил: Анисимов Максим Дмитриевич P3233

Вариант №2

Исходная функция:

График функции:



Код, выполненный на языке программирования Python:

1. Метод половинного деления:

import numpy as np

def f(x):

return np.log(1 + x\*\*2) - np.sin(x)

def half\_division(a, b, epsilon):

counter = 1

while(np.abs(b-a) >= 2\*epsilon):

if(counter > 25):

print("Достигнута 25-ая итерация")

break

x1 = (a+b - epsilon) / 2

x2 = (a+b + epsilon) / 2

print(f"n: {round(counter, 3)} | a= {round(a, 3)} | b= {round(b,3)} \n")

counter = counter + 1

if(np.abs(b-a)/2 < epsilon):

break

if(f(x1) <= f(x2)):

b=x2

else:

a=x1

print(f"Ответ: экстремум функции равен {round(f((b+a)/2), 10)} при {round((b+a)/2, 10)}")

if \_\_name\_\_ == '\_\_main\_\_':

half\_division(0, np.pi/4, 10\*\*(-10))

Результат работы программы:

n: 1 | a= 0 | b= 0.785

n: 2 | a= 0.393 | b= 0.785

n: 3 | a= 0.393 | b= 0.589

n: 4 | a= 0.491 | b= 0.589

n: 5 | a= 0.54 | b= 0.589

n: 6 | a= 0.54 | b= 0.565

n: 7 | a= 0.552 | b= 0.565

n: 8 | a= 0.552 | b= 0.558

n: 9 | a= 0.555 | b= 0.558

n: 10 | a= 0.555 | b= 0.557

n: 11 | a= 0.555 | b= 0.556

n: 12 | a= 0.556 | b= 0.556

n: 13 | a= 0.556 | b= 0.556

n: 14 | a= 0.556 | b= 0.556

n: 15 | a= 0.556 | b= 0.556

n: 16 | a= 0.556 | b= 0.556

n: 17 | a= 0.556 | b= 0.556

n: 18 | a= 0.556 | b= 0.556

n: 19 | a= 0.556 | b= 0.556

n: 20 | a= 0.556 | b= 0.556

n: 21 | a= 0.556 | b= 0.556

n: 22 | a= 0.556 | b= 0.556

n: 23 | a= 0.556 | b= 0.556

n: 24 | a= 0.556 | b= 0.556

n: 25 | a= 0.556 | b= 0.556

Достигнута 25-ая итерация

Ответ: экстремум функции равен -0.2584255601 при 0.555967

Метод золотого сечения:

import math

import numpy as np

def f(x):

return np.log(1 + x\*\*2) - np.sin(x)

def golden\_ratio(a, b, eps):

tau = (math.sqrt(5)-1)/2

x1 = a + ((3 - math.sqrt(5))/2)\*(b-a)

x2 = a + ((math.sqrt(5) - 1)/2)\*(b-a)

f\_x1 = f(x1)

f\_x2 = f(x2)

eps\_n = ((b-a)/2)

iterator = 1

while(((b-a)/2)>eps):

if(iterator > 25):

print("25-ая итерация достигнута")

break

if (f\_x1<f\_x2):

b = x2

x2 = x1

f\_x2 = f\_x1

x1 = b - tau\*(b-a)

f\_x1 = f(x1)

else:

a = x1

x1 = x2

f\_x1 = f\_x2

x2 = a + tau\*(b-a)

f\_x2 = f(x2)

eps\_n \*= tau

print(f"n: {iterator} | a = {a//0.001/1000} | b = {b//0.001/1000}")

iterator+=1

print(f"Ответ найден | Экстремум функции равен {f((a+b)/2)} при x= {(a+b)/2}")

if \_\_name\_\_ == "\_\_main\_\_":

golden\_ratio(0, np.pi/4, 10\*\*(-10))

Результат работы программы:

n: 1 | a = 0.299 | b = 0.785

n: 2 | a = 0.485 | b = 0.785

n: 3 | a = 0.485 | b = 0.67

n: 4 | a = 0.485 | b = 0.599

n: 5 | a = 0.529 | b = 0.599

n: 6 | a = 0.529 | b = 0.572

n: 7 | a = 0.545 | b = 0.572

n: 8 | a = 0.545 | b = 0.562

n: 9 | a = 0.552 | b = 0.562

n: 10 | a = 0.552 | b = 0.558

n: 11 | a = 0.554 | b = 0.558

n: 12 | a = 0.554 | b = 0.557

n: 13 | a = 0.555 | b = 0.557

n: 14 | a = 0.555 | b = 0.556

n: 15 | a = 0.555 | b = 0.556

n: 16 | a = 0.555 | b = 0.556

n: 17 | a = 0.555 | b = 0.556

n: 18 | a = 0.555 | b = 0.556

n: 19 | a = 0.555 | b = 0.556

n: 20 | a = 0.555 | b = 0.556

n: 21 | a = 0.555 | b = 0.555

n: 22 | a = 0.555 | b = 0.555

n: 23 | a = 0.555 | b = 0.555

n: 24 | a = 0.555 | b = 0.555

n: 25 | a = 0.555 | b = 0.555

25-ая итерация достигнута

Ответ найден | Экстремум функции равен -0.2584255600 при x= 0.5559677007

Метод Ньютона:

import math

import numpy as np

def f(x):

return np.log(1 + x\*\*2) - np.sin(x)

def derivative\_f(x):

return (2\*x)/(x\*\*2 + 1) - np.cos(x)

def second\_derivative\_f(x):

return np.sin(x) - (2\*(x\*\*2) - 2) / (x\*\*4 + 2\*(x\*\*2) + 1)

def newton\_method(x, eps):

counter = 1

while True:

if counter > 25:

print("Достигнута 25-ая итерация \n")

break

if abs(derivative\_f(x)) < eps:

print(f"Ответ: Экстремум функции {round(f(x),10)} при x = {round(x, 10)} \n")

break

x\_next = x - derivative\_f(x) / second\_derivative\_f(x)

print(f"n: {counter} | x = {round(x, 3)} | f(x) = {round(f(x), 3)} | f'(x) = {round(derivative\_f(x), 3)} | f''(x) = {round(second\_derivative\_f(x), 3)} | x\_next = {round(x\_next, 3)} \n")

x = x\_next

counter = counter + 1

if \_\_name\_\_ == "\_\_main\_\_":

newton\_method(0, 10\*\*-10)

Результат работы программы:

n: 1 | x = 0 | f(x) = 0.0 | f'(x) = -1.0 | f''(x) = 2.0 | x\_next = 0.5

n: 2 | x = 0.5 | f(x) = -0.256 | f'(x) = -0.078 | f''(x) = 1.439 | x\_next = 0.554

n: 3 | x = 0.554 | f(x) = -0.258 | f'(x) = -0.003 | f''(x) = 1.338 | x\_next = 0.556

n: 4 | x = 0.556 | f(x) = -0.258 | f'(x) = -0.0 | f''(x) = 1.334 | x\_next = 0.556

Ответ: Экстремум функции -0.2584255601 при x = 0.5559684307