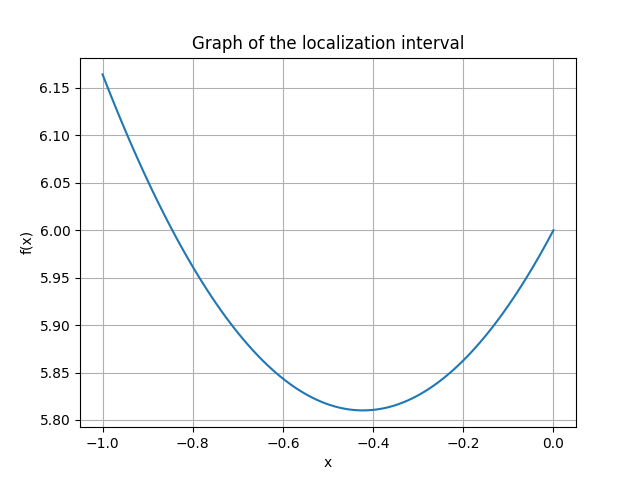
Построение графика:

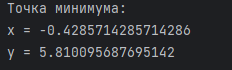
import numpy as np  
import matplotlib.pyplot as plt  
  
def f(x):  
 return x\*\*2 + 6 \* np.exp(0.15\*x)  
  
a = -1  
b = 0  
x = np.linspace(a, b, 100)  
y = f(x)  
  
plt.plot(x, y)  
plt.xlabel('x')  
plt.ylabel('f(x)')  
plt.title('Graph of the localization interval')  
plt.grid(True)  
plt.show()

результат:  


1. Метод перебора на равномерной сетке:

Код программы:

import numpy as np  
  
# Определить функцию  
def f(x):  
 return (x\*\*2) + 6 \* np.exp(0.15 \* x)  
  
# Определить отрезок  
a = -1  
b = 0  
  
# Определить число итераций  
n = 22  
  
# Создать массив значений x  
x = np.linspace(a, b, n)  
  
# Вычислить значения y  
y = f(x)  
  
# Найти точку минимума  
min\_index = np.argmin(y)  
min\_x = x[min\_index]  
min\_y = y[min\_index]  
  
# Вывести результат на экран  
print("Точка минимума:")  
print("x =", min\_x)  
print("y =", min\_y)

Вывод:   


1. метод дихотомии (деления интервала пополам)  
   код:

import numpy as np

def f(x):

return x\*\*2 + 6 \* np.exp(0.15\*x)

# Определить отрезок

a = -1

b = 0

# Определить число итераций

n = 22

# Найти точку минимума методом дихотомии

for i in range(n):

# Вычислить середину отрезка

c = (a + b) / 2

# Вычислить значения функции в точках a, c и b

fa = f(a)

fc = f(c)

fb = f(b)

# Определить, в какой половине отрезка находится точка минимума

if fc < fa and fc < fb:

# Точка минимума находится в левой половине отрезка

b = c

elif fc < fa and fc > fb:

# Точка минимума находится в правой половине отрезка

a = c

else:

# Точка минимума находится в середине отрезка

break

# Вычислить точку минимума и значение функции в этой точке

min\_x = (a + b) / 2

min\_y = f(min\_x)

# Вывести результат на экран

print("Точка минимума:")

print("x =", min\_x)

print("y =", min\_y)

# Построить график функции и отметить точку минимума

x = np.linspace(a, b, 100)

y = f(x)

plt.plot(x, y)

plt.plot(min\_x, min\_y, 'ro')

plt.xlabel('x')

plt.ylabel('f(x)')

plt.title('Graph of the function and the minimum point')

plt.grid(True)

plt.show()

вывод:

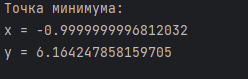


1. метод золотого сечения

код:

import numpy as np  
  
def f(x):  
 return x\*\*2 + 6 \* np.exp(0.15\*x)  
  
# Определить отрезок  
a = -1  
b = 0  
  
# Определить число итераций  
n = 22  
  
# Найти точку минимума методом золотого сечения  
for i in range(n):  
 # Вычислить коэффициенты золотого сечения  
 r = (np.sqrt(5) - 1) / 2  
 c = a + r \* (b - a)  
 d = b - r \* (b - a)  
  
 # Вычислить значения функции в точках c и d  
 fc = f(c)  
 fd = f(d)  
  
 # Определить, в какой половине отрезка находится точка минимума  
 if fc < fd:  
 # Точка минимума находится в левой половине отрезка  
 b = d  
 else:  
 # Точка минимума находится в правой половине отрезка  
 a = c  
  
# Вычислить точку минимума и значение функции в этой точке  
min\_x = (a + b) / 2  
min\_y = f(min\_x)  
  
# Вывести результат на экран  
print("Точка минимума:")  
print("x =", min\_x)  
print("y =", min\_y)

вывод:



1. Метод Фибоначчи

Код:

import numpy as np

def f(x):

return x\*\*2 + 6 \* np.exp(0.15\*x)

# Определить отрезок

a = -1

b = 0

# Определить число итераций

n = 22

# Вычислить числа Фибоначчи

fib = [0, 1]

for i in range(2, n+1):

fib.append(fib[i-1] + fib[i-2])

# Найти точку минимума методом Фибоначчи

for i in range(n-1, 0, -1):

# Вычислить коэффициенты Фибоначчи

r = fib[i-1] / fib[i]

c = a + r \* (b - a)

d = b - r \* (b - a)

# Вычислить значения функции в точках c и d

fc = f(c)

fd = f(d)

# Определить, в какой половине отрезка находится точка минимума

if fc < fd:

# Точка минимума находится в левой половине отрезка

b = d

else:

# Точка минимума находится в правой половине отрезка

a = c

# Вычислить точку минимума и значение функции в этой точке

min\_x = (a + b) / 2

min\_y = f(min\_x)

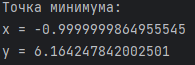
# Вывести результат на экран

print("Точка минимума:")

print("x =", min\_x)

print("y =", min\_y)

Вывод:

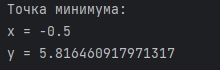


1. Метод Больцано

Код:

import numpy as np  
  
def f(x):  
 return x\*\*2 + 6 \* np.exp(0.15\*x)  
  
# Определить отрезок  
a = -1  
b = 0  
  
# Определить число итераций  
n = 22  
  
# Найти точку минимума методом Больцано  
for i in range(n):  
 # Вычислить середину отрезка  
 c = (a + b) / 2  
  
 # Вычислить значения функции в точках a, c и b  
 fa = f(a)  
 fc = f(c)  
 fb = f(b)  
  
 # Определить, в какой половине отрезка находится точка минимума  
 if fa < fc and fc < fb:  
 # Точка минимума находится в левой половине отрезка  
 b = c  
 elif fa > fc and fc > fb:  
 # Точка минимума находится в правой половине отрезка  
 a = c  
 else:  
 # Точка минимума находится в середине отрезка  
 break  
  
# Вычислить точку минимума и значение функции в этой точке  
min\_x = (a + b) / 2  
min\_y = f(min\_x)  
  
# Вывести результат на экран  
print("Точка минимума:")  
print("x =", min\_x)  
print("y =", min\_y)

Вывод:

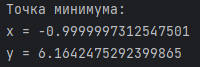


1. Метод касательных

Код:

import numpy as np  
  
def f(x):  
 return (x\*\*2) + 6 \* np.exp(0.15\*x)  
  
# Определить отрезок  
a = -1  
b = 0  
  
# Определить число итераций  
n = 22  
  
# Найти точку минимума методом золотого сечения  
for i in range(n):  
 # Вычислить точки золотого сечения  
 r = (np.sqrt(5) - 1) / 2  
 c = a + r \* (b - a)  
 d = b - r \* (b - a)  
  
 # Вычислить значения функции в точках золотого сечения  
 fc = f(c)  
 fd = f(d)  
  
 # Определить, в какой половине отрезка находится точка минимума  
 if fc < fd:  
 # Точка минимума находится в левой половине отрезка  
 b = d  
 else:  
 # Точка минимума находится в правой половине отрезка  
 a = c  
  
 # Проверить, достигли ли мы требуемой точности  
 if abs(b - a) < 1e-6:  
 break  
  
# Вычислить точку минимума и значение функции в этой точке  
min\_x = (a + b) / 2  
min\_y = f(min\_x)  
  
# Вывести результат на экран  
print("Точка минимума:")  
print("x =", min\_x)  
print("y =", min\_y)

Вывод:



1. Метод парабол

Код:

import numpy as np  
  
def f(x):  
 return x\*\*2 + 6 \* np.exp(0.15\*x)  
  
def parabolic\_minimization(f, a, b, n):  
 for i in range(n):  
 x1 = a + (b - a) / 3  
 x2 = b - (b - a) / 3  
  
 if f(x1) < f(x2):  
 b = x2  
 else:  
 a = x1  
  
 return (a + b) / 2  
  
a = -1  
b = 0  
n = 22  
  
min\_point = parabolic\_minimization(f, a, b, n)  
min\_value = f(min\_point)  
  
print("Точка минимума:")  
print(f"x = {min\_point}")  
print(f"y = {min\_value}")

Вывод:

