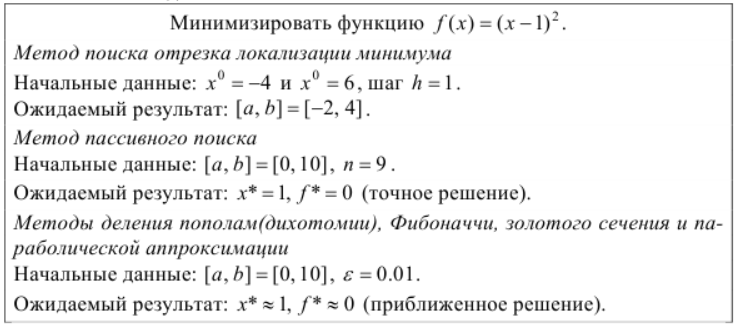
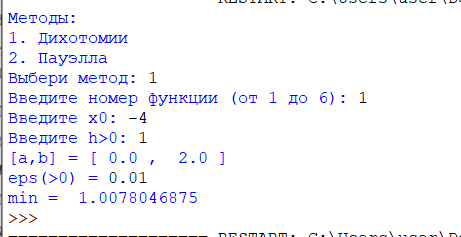
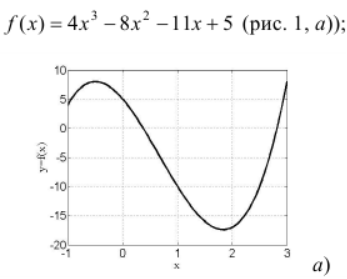
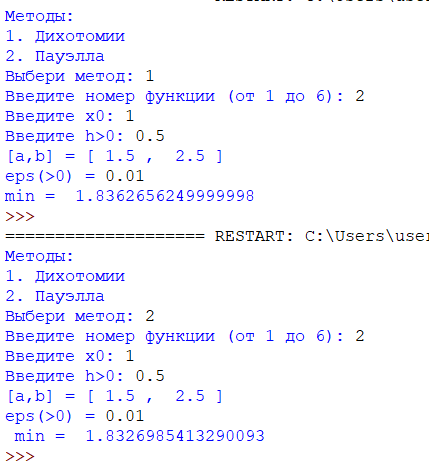
|  |  |  |
| --- | --- | --- |
| Лабораторная работа №1  Методы одномерной минимизации | Ф.И.О. | Бокова Ольга Дмитриевна |
| Группа | ИВТ-363 |
| Преподаватель | О.Н. Асанова |
| Дата сдачи | 30.09.2022 |

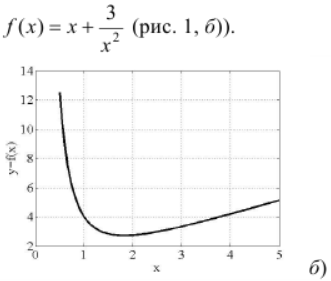
Методы: ДСК, дихитомии и Пауэлла

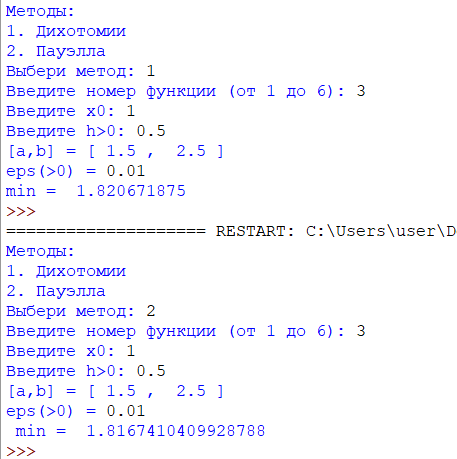


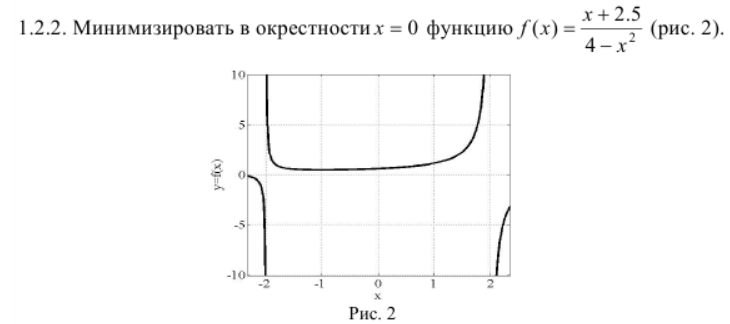


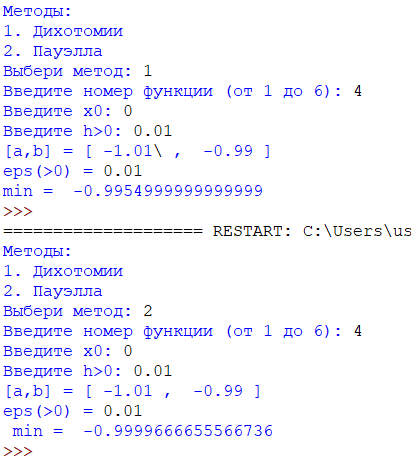


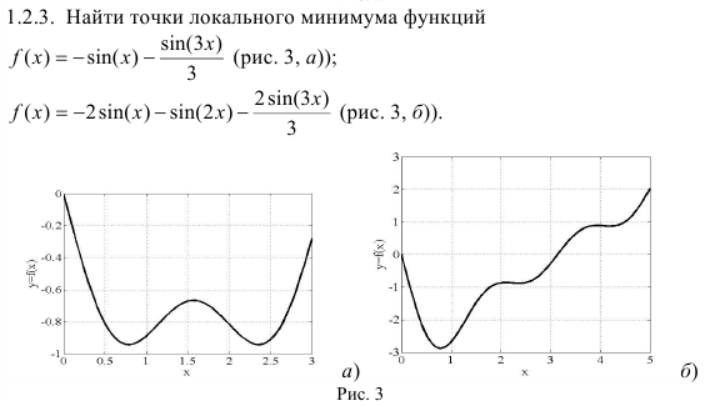


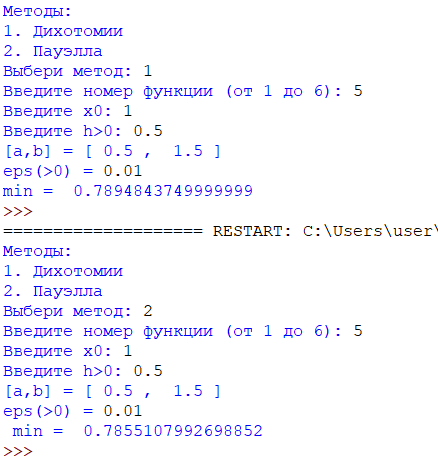


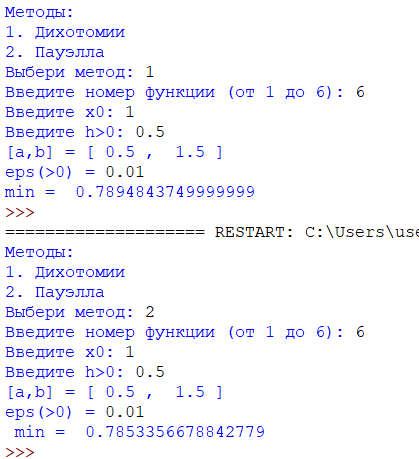












Код программы:

#import numpy as np

import math

# Анализируемые функции

def function(x,n):

if n == 1: return pow((x-1),2)

if n == 2: return 4\*pow(x,3) - 8\*pow(x,2) - 11\*x + 5

if n == 3:

if x == 0:

print('Программа вылетит, поменяйте x0')

quit()

else:

return x+(3/(x\*\*2))

if n == 4:

if x == 2.0 or x == -2.0:

print('Программа вылетит, поменяйте x0')

quit()

else:

return (x+2.5) / (4-pow(x,2))

if n == 5: return -math.sin(x)-(math.sin(3\*x)/3)

if n == 6: return -2\*math.sin(x) - math.sin(2\*x) - 2\*math.sin(3\*x)/3

def DSK(x0, h, n):

a = 0

b = 0

k = 0

incorrect\_flag = 0

if function(x0 + k\*h, n) >= function(x0 + (k+1)\*h, n):

a = x0

k += 1

while True:

if k>100:

incorrect\_flag = 1

break

if function(x0 + k\*h, n) < function(x0 + (k+1)\*h, n):

b = x0 + (k+1)\*h

break

else:

a = x0 + k\*h

k += 1

elif function(x0 - k\*h, n) >= function(x0 - (k+1)\*h, n):

b = x0

k += 1

while True:

if k > 100:

incorrect\_flag = 1

break

if function(x0 - k \* h, n) < function(x0 - (k+1)\*h, n):

a = x0 - (k + 1) \* h

break

else:

b = x0 - k \* h

k += 1

else:

a = x0 - h

b = x0 + h

if incorrect\_flag == 0:

print('[a,b] = [',a,', ',b,']')

return [a,b]

else: return 'Incorrect'

# Choose method

print('Методы:')

print('1. Дихотомии')

print('2. Пауэлла')

#print('3. Хука-Дживса')

method = int(input('Выбери метод: '))

if method > 2 or method < 1:

print('Такого метода нет')

method = int(input('Выбери метод: '))

# Input

f\_N = int(input('Введите номер функции (от 1 до 6): '))

while (f\_N > 6 or f\_N < 1):

print('Такой функции нет')

f\_N = int(input('Введите номер функции (от 1 до 6): '))

x0 = float(input('Введите x0: '))

h = float(input('Введите h>0: '))

if (h <= 0):

print('Шаг должен быть больше 0')

h = float(input('Введите h>0: '))

segment = DSK(x0, h, f\_N)

if segment == 'Incorrect':

print('Введите другие x0 и h')

method = -1

else:

a = segment[0]

b = segment[1]

# Деление пополам

if method == 1:

eps = float(input('eps(>0) = '))

if (eps <= 0):

print('Параметр точности поиска должен быть больше 0')

eps = float(input('eps(>0) = '))

delta = eps / 10

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

x1 = (a + b) / 2 - delta

x2 = (a + b) / 2 + delta

if function(x1, f\_N) <= function(x2, f\_N):

b = x2

else:

a = x1

rezult = (a + b) / 2

print('min = ', rezult)

# Метод Пауэлла

if method == 2:

eps = float(input('eps(>0) = '))

if (eps <= 0):

print('Параметр точности поиска должен быть больше 0')

eps = float(input('eps(>0) = '))

delta = eps / 10

# step 1

x1 = a

x2 = (a + b) / 2

x3 = b

X = [x1, x2, x3]

# step 2

while True:

min\_x = X[0]

for i in X:

if (function(i, f\_N) < function(min\_x, f\_N)):

min\_x = i

# step 3

num = (X[1] \*\* 2 - X[2] \*\* 2) \* function(X[0], f\_N) + (X[2] \*\* 2 - X[0] \*\* 2) \* function(X[1], f\_N) + (

X[0] \*\* 2 - X[1] \*\* 2) \* function(X[2], f\_N)

denum = (X[1] - X[2]) \* function(X[0], f\_N) + (X[2] - X[0]) \* function(X[1], f\_N) + (X[0] - X[1]) \* function(X[2], f\_N)

X.append(0.5 \* (num / denum))

# step 4

if (abs(X[3] - min\_x) <= eps):

break

# step 5

X.sort()

# step 6

max\_x = 0

for i in range(4):

if (function(X[i], f\_N) > function(X[max\_x], f\_N)):

max\_x = i

X.pop(max\_x)

# step 7

rezult2 = X[-1]

print(' min = ', rezult2)