Изображение выглядит как металлоизделия, коллекция картинок

Автоматически созданное описание

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

**«Дальневосточный федеральный университет»**

**(ДВФУ)**

Институт математики и компьютерных технологий

Департамент математического и

компьютерного моделирования

ОТЧЕТ

по лабораторной работе №1

по дисциплине «Методы оптимизации»

Выполнил студент

гр. Б9119-02.03.01сцт

Петров С.Д.

«25» марта 2022 г.

Проверил

\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«25» марта 2022 г.

г. Владивосток

2022

**Постановка задачи**

Требуется найти минимум функции при условии, что

.

**Решение**

Данную задачу можно представить системой уравнений:

.

Для поиска оптимального решения воспользуемся теоремой

Куна – Таккера. Из данной теоремы следует, что , такой что:

1. , где ,
2. (1),
3. .

Из пункта 2 видно, что возможно два варианта. Если , то мы получаем задачу безусловного минимума, поэтому будем рассматривать только вариант .

Так как пространство является Гильбертовым, . Таким образом, можно представить как .

Итак, исходная задача преобразуется в задачу поиска безусловного минимума функции Лагранжа. Получим новое условие минимума:

Решая систему уравнений (1)-(2), получим решение .

– выбирается из условия, что , где – точка безусловного минимума.

**Код программы**

**import** **numpy** **as** **np**

**import** **matplotlib.pyplot** **as** **plt**

n = **6**

maxI = **10000000**

eps = **1e-6**

**def** **f**(x):

**return** (**0.5**\*np.reshape(x, (**1**, n))).dot(A).dot(np.reshape(x, (n, **1**))) + b.dot(x)

**def** **grad**(x):

**return** np.reshape(A.dot(x), (**1**, n)) + b

**def** **H**(x):

**return** A

**def** **Hinv**(x):

**return** np.linalg.inv(A)

A = np.random.randn(n, n)\***10**

b = np.random.rand(n)

x = np.zeros((n,**1**))

x0 = x

i = **0**

**while** np.linalg.norm(grad(x)) > eps:

x = x - Hinv(x).dot(np.reshape(grad(x), (n, **1**)))

i += **1**

**if** (i >= maxI):

**print**("Ошибка")

**break**

**print**("Безусловный минимум методом Ньютона")

**print**(x)

r = np.random.rand(**1**, **1**)

c = np.reshape((x - x0), (**1**, n)).dot(x - x0)

**while** c - r\*r <= **0**:

r = np.random.rand(**1**, **1**)

**print**("Число r")

**print**(r)

**print**("Проверим, что для безусловного минимума не выполнено условие (x - x0)\*(x - x0) -r\*r <= 0")

**print**("(x - x0)\*(x - x0) -r\*r = ", str(c - r\*r))

M = A

col = [**0.** **for** i **in** range(**0**,n)]

col[**0**] = float(c)

col = np.array(col, dtype=np.float64)

M = np.column\_stack((M, np.reshape(col, (n, **1**))))

c1 = np.append(x-x0, **0**)

M = np.vstack((M, c1))

B = np.append(-b, r\*r)

Y = np.linalg.inv(M.astype('float')).dot(B)

x = Y[**0**:n]

**print**("Условный минимум")

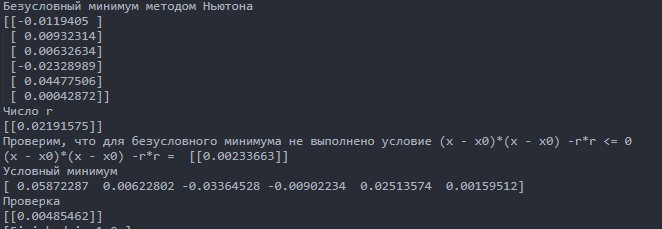
**print**(x)

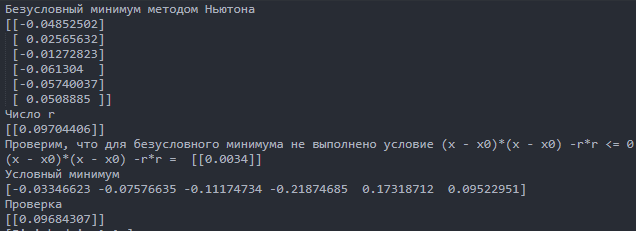
**print**("Проверка")

x = np.reshape(x, (n, **1**))

**print**(np.reshape((x - x0), (**1**, n)).dot(x - x0) - r\*r)

**Тесты**

****

****

**Заключение**

В ходе данной лабораторной работы, используя теорему Куна-Таккера, удалось найти минимум функции при условии, что .