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 №7**

**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 многомерной оптимизации:**

* функция –
* метод оптимизации для «ручного расчета» - значение параметра **p=3**;
* метод оптимизации для расчета на ПК – значение параметра t=1.

1. **Проверка существования минимума функции**

Известно, что всякий глобальный минимум выпуклой функции является одновременно и локальным.

Проверим, что функция является выпуклой на множестве R.

**Матрица Гессе** для функции :

а угловые миноры:

.

Таким образом, функция  - выпуклая на множестве R**.**

1. **Решение задачи многомерной оптимизации аналитическим методом**

Необходимые условия существования точки экстремума:



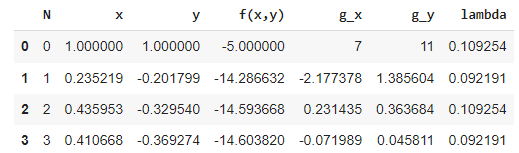
Откуда

1. **Начальная точка итерационного процесса численного решения задачи многомерной оптимизации**

Выберем начальную точку спуска - .

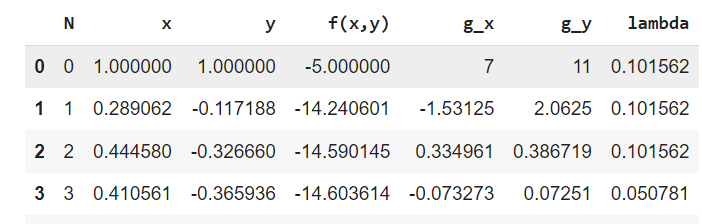
1. **Пример выполнения 1-й итерации методом наискорейшего спуска (НСА):**

**Пример выполнения 3-х итерации по методу НСА:**

**

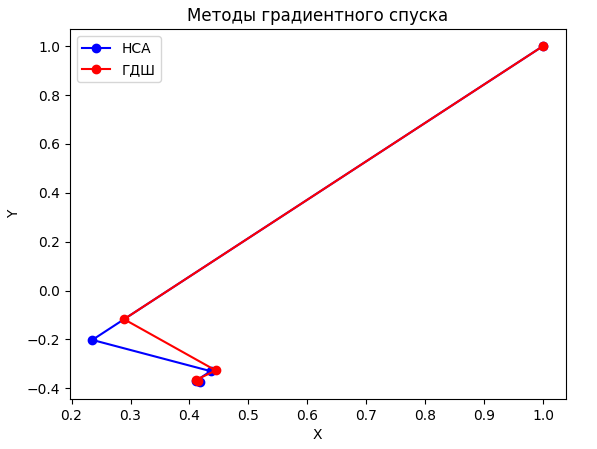
После 3-х итераций: Xmin=0.410668, ymin=-0.3692, f=-14.6038**.**

**Пример выполнение 3-х итерации по методу ГДШ:**

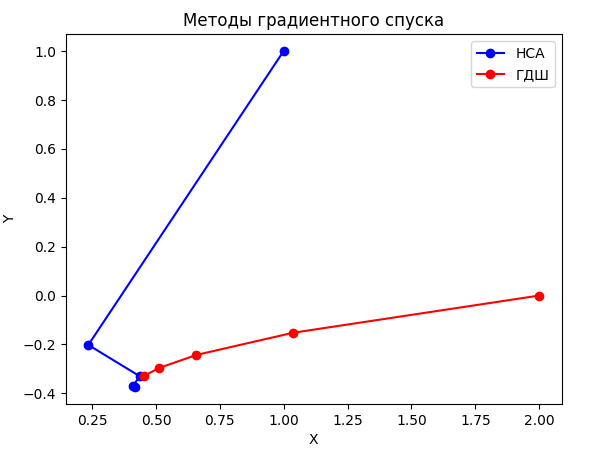
**

После 3-х итераций: Xmin=0.410561, ymin=-0.3659, f=-14.6036**.**

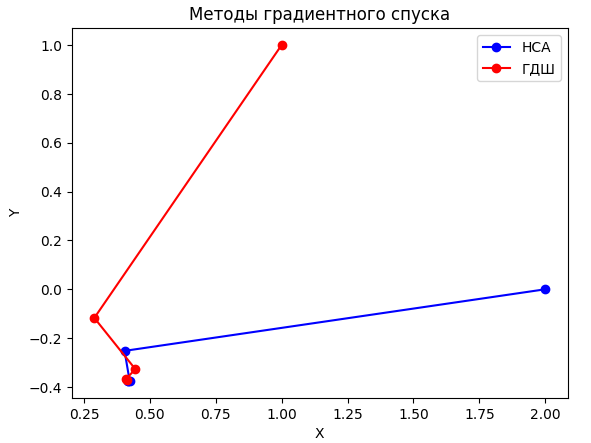
1. **Построение траектории поиска минимума методами НСА и ГДШ.**

**

Траектории методов НСА и ГДШ. Оба метода начинают в (1;1)



НСА - (1;1), ГДШ – (2,0)



НСА – (2;0), ГДШ – (1;1)

Был использован ЯП Python.

# Определение функции

def func(x, y):

    return 6 \* x\*\*2 + 4 \* y\*\*2 - 5 \* x + 3 \* y - 13

# Определение градиента функции

def grad(x, y):

    return (12 \* x - 5, 8 \* y + 3)

# Начальная точка

x0, y0 = 1, 1

# Количество итераций

num\_iterations = 5

import math

# Наискорейшего дробления шага

def lambda\_f\_NSA(x,y):

  return (17 - 60\*x + 72\*x\*\*2 + 24\*y + 32\*y\*\*2)/(186 - 720\*x + 864\*x\*\*2 + 192\*y + 256\*y\*\*2)

# Градиентного дробления шага

def lambda\_f\_GDSH(x,y, guess = 13):

  gradient = grad(x, y)

  x1 = x - guess \* gradient[0]

  y1 = y - guess \* gradient[1]

  if func(x,y) - func(x1,y1) >= (guess/2)\*(gradient[0]\*\*2 + gradient[1]\*\*2):

    return guess

  return lambda\_f\_GDSH(x,y, guess/2)

data = {

    'N': [],

    'x': [],

    'y': [],

    'f(x,y)': [],

    'g\_x': [],

    'g\_y': [],

    'lambda': []

}

functions = [lambda\_f\_NSA, lambda\_f\_GDSH]

for func\_ in functions:

  if func\_ == lambda\_f\_NSA:

    x0, y0 = 2, 0

  else:

    x0, y0 = 1, 1

  for i in range(num\_iterations):

      print(x0,y0)

      data['N'].append(i)

      data['x'].append(x0)

      data['y'].append(y0)

      data['f(x,y)'].append(func(x0,y0))

      gradient = grad(x0, y0)

      step\_size = func\_(x0,y0) # lambda\_f\_NSA(x0,y0) для метода НСА, для ГДШ 13

      data['g\_x'].append(gradient[0])

      data['g\_y'].append(gradient[1])

      data['lambda'].append(step\_size)

      x0 -= step\_size \* gradient[0]

      y0 -= step\_size \* gradient[1]

  data['g\_x'][-1] = "-"

  data['g\_y'][-1] = "-"

  data['lambda'][-1] = "-"

import pandas as pd

# Создаем DataFrame из словаря

df = pd.DataFrame(data)

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

# Вывод найденного минимума

minimum = func(x0, y0)

print(minimum)

df

import matplotlib.pyplot as plt

# Задайте значения x и y

x = [1, 2, 3, 4, 5]

y = [1, 4, 9, 16, 25]

plt.plot(data['x'][:num\_iterations], data['y'][:num\_iterations], marker='o', linestyle='-', color='blue', label='НСА')

plt.plot(data['x'][num\_iterations:], data['y'][num\_iterations:], marker='o', linestyle='-', color='red', label='ГДШ')

plt.xlabel('X')

plt.ylabel('Y')

plt.legend()

plt.title('Методы градиентного спуска')

df