Министерство науки и высшего образования Российской Федерации
Калужский филиал
федерального государственного бюджетного
образовательного учреждения высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ***ИУК «Информатика и управление»***

КАФЕДРА ***ИУК4 «Программное обеспечение ЭВМ, информационные технологии»***

ЛАБОРАТОРНАЯ РАБОТА №1.1

«Минимизация функций»

ДИСЦИПЛИНА: «Моделирование»

Выполнил: студент гр. ИУК4-72Б ________________ (Карельский М.К.)
(Подпись)

Проверил: ________________ (Никитенко У.В.)
(Подпись)

Дата сдачи (защиты):

Результаты сдачи (защиты):
- Балльная оценка:
- Оценка:

Калуга, 2023

Цель: сформировать практические навыки анализа возможностей построения и выделения наиболее важных свойств объектов моделей для моделирования и использования специализированных программных пакетов и библиотек для решения задачи минимизации функции и визуализации результатов решения.

Задачи: найти минимум функции, указанной в варианте предложенным методом, сравнить результаты, выдвинуть и обосновать гипотезу целесообразности использования того или иного метода в зависимости от предложенной задачи и ее вариаций, точности результата, трудоемкости, сложности алгоритма, сложности обоснования применимости метода, вычислительной эффективности алгоритма. Визуализировать результаты.

Вариант 7

Задание 1.7.

Методом Ньютона найти минимум и максимум унимодальной на отрезке $[a; b]$ функции $f(x)$ с точностью $\varepsilon = 10^{-6}$. Предусмотреть подсчет числа итераций, потребовавшихся для достижения заданной точности.

$$f(x) = x^3 - e^x$$
$$a = 0$$
$$b = 3$$

Решение:

$$x_{i+1} = x_i - \frac{f'(x)}{f''(x)}$$

$$f'(x) = 3x^2 - e^x$$
$$f''(x) = 6x - e^x$$

В качестве начального значения возьмем:

$$x_0 = \frac{a+b}{2}$$

Результаты вычислений:

$$x = 0{,}910008$$
$$f(x) = -1{,}730752$$
$$n = 5$$

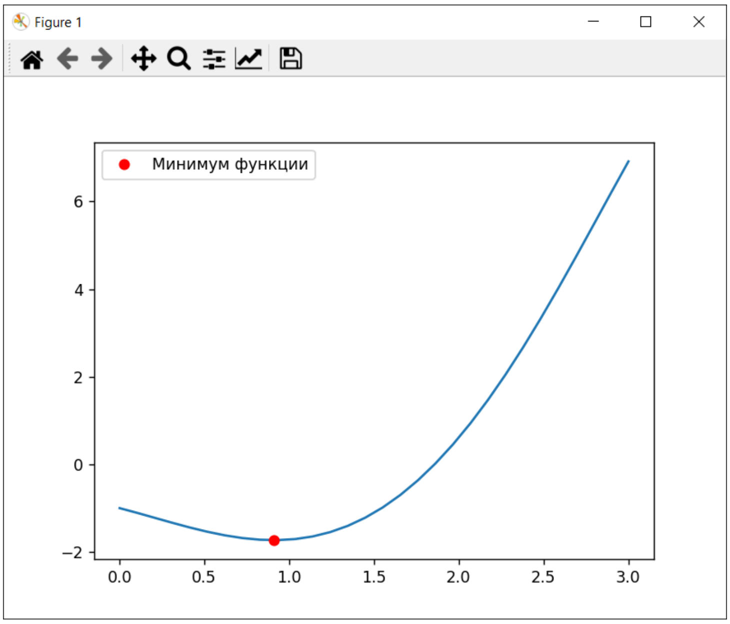

Рис. 1. Метод Ньютона

Задание 2.3.

Указанным в индивидуальном варианте методом найти минимумы и максимумы функции $f(x)$ на отрезке $[x_1, x_2]$ с точностью $\varepsilon = 10^{-6}$. Предусмотреть подсчет числа итераций, потребовавшихся для достижения заданной точности.

$$f(t) = \cos e^t$$
$$x_1 = 1$$
$$x_2 = 2$$

Метод – Фибоначчи

Решение:

В качестве константы различимости выберем:

$$\epsilon = 0.01$$

Установим, что:

$$k = 1$$
$$a_k = x_1$$
$$b_k = x_2$$
$$l = 2\varepsilon$$
$$F_n - \text{число Фибоначчи}$$

Тогда общее число вычислений n определим из:

$$\mathrm{F}_n > \frac{b-a}{l}$$

Вычислим:

$$\lambda_k = a_k + \frac{F_{n-2}}{F_n}(b_k - a_k)$$
$$\mu_k = a_k + \frac{F_{n-1}}{F_n}(b_k - a_k)$$

Шаг 1: если:

$$f(\lambda_k) > f(\mu_k),$$

то:

$$a_{k+1} = \lambda_k$$
$$b_{k+1} = b_k$$
$$\lambda_{k+1} = \mu_k$$
$$\mu_{k+1} = a_{k+1} + \frac{F_{n-k-1}}{F_{n-k}}(b_{k+1} - a_{k+1})$$

иначе:

$$a_{k+1} = a_k$$
$$b_{k+1} = \mu_k$$
$$\mu_{k+1} = \lambda_k$$
$$\lambda_{k+1} = a_{k+1} + \frac{F_{n-k-2}}{F_{n-k}}(b_{k+1} - a_{k+1})$$

Шаг 2: если:

$$k \neq n - 2,$$

то увеличить k на 1 и перейти к первому шагу.
Шаг 3:

$$\lambda_n = \lambda_{n-1}$$
$$\mu_n = \lambda_n + \epsilon$$

Если:

$$f(\lambda_n) = f(\mu_n),$$

то:

$$a_n = \lambda_n$$
$$b_n = b_{n-1}$$

иначе:

$$a_n = a_{n-1}$$
$$b_n = \mu_n$$

В результате имеем:

$$x = \frac{a_n + b_n}{2}$$

Результаты вычислений:

$$n = 28$$
$$x_{min} = 1{,}149728 \pm 10^{-6}$$
$$f(x_{min}) = -0.999876$$
$$x_{max} = 1{,}842877 \pm 10^{-6}$$
$$f(x_{max}) = -0.999504$$

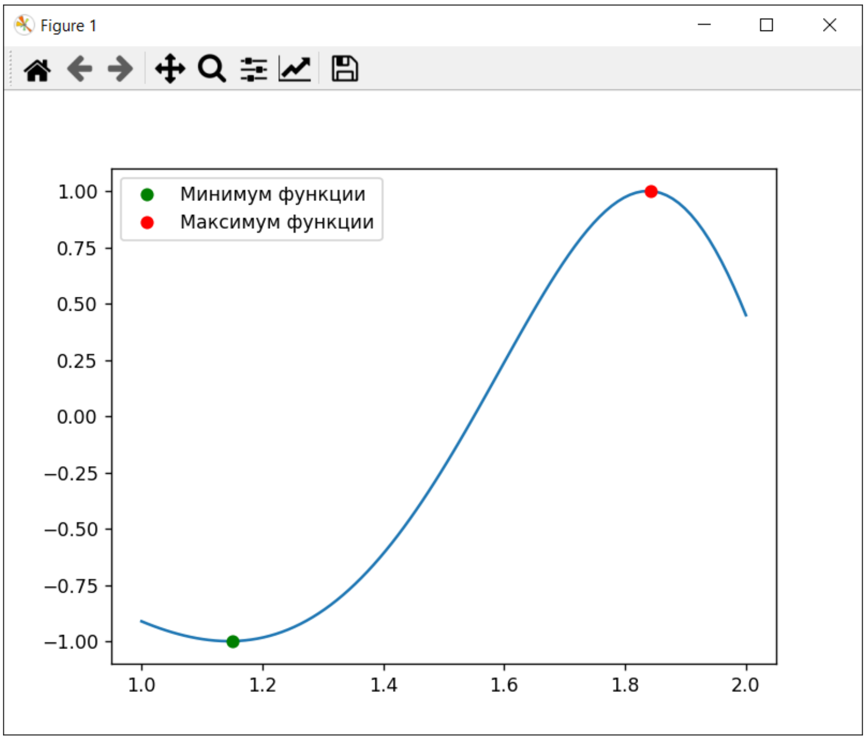

Рис. 2. Метод Фибоначчи

Задание 5.7.

Найти минимум функции 2-х переменных $f(x, y)$ с точностью $\varepsilon = 10^{-6}$ на прямоугольнике $[x_1, x_2] \times [y_1, y_2]$.

Порядок решения задачи:

1. Задать указанную в варианте функцию $f(x, y)$.
2. Построить графики функции и поверхностей уровня $f(x, y)$.
3. По графикам найти точки начального приближения к точкам экстремума.
4. Найти экстремумы функции с заданной точностью.

$$f(x,y) = 3x^2 + y^2 + \ln(x^2 + y^2 + 2x - 2y + 3)$$
$$x_1 = -2$$
$$x_2 = 2$$
$$y_1 = -2$$
$$y_2 = 2$$

Решение:

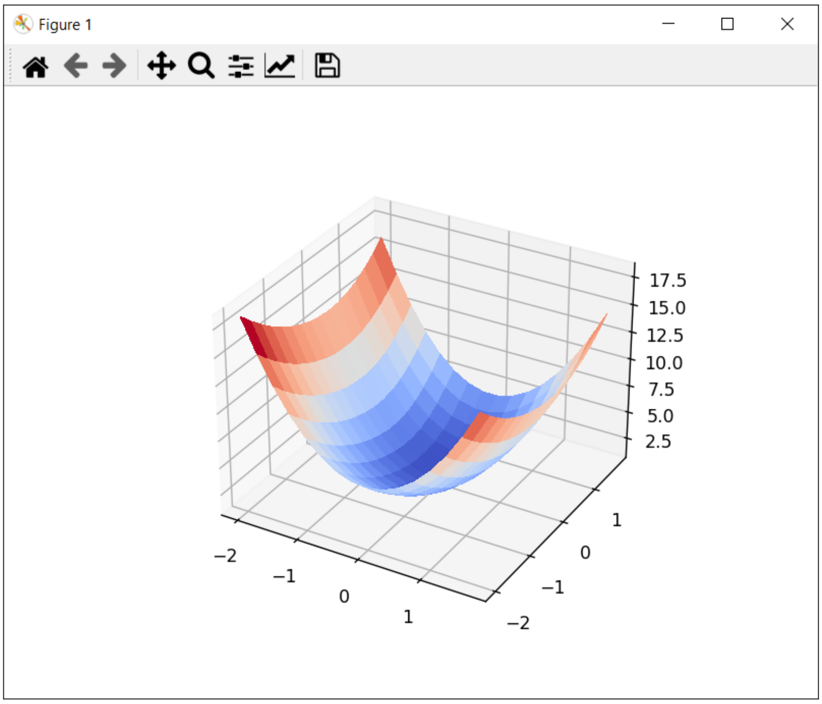

Рис. 3. График функции

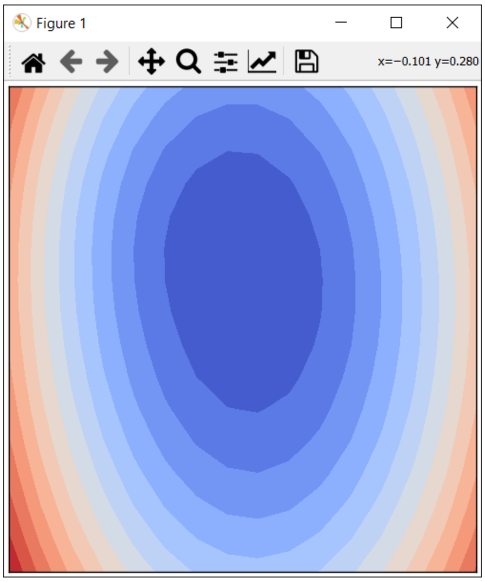

Рис. 4. График поверхностей уровня

Исходя из графиков, можно предположить, что в указанных границах функция имеет точку минимума M примерно в $(-0.1, 0.3)$.

Вычислим точное значение:

$$f'_x(x,y) = 6x + \frac{2x+2}{x^2+y^2+2x-2y+3}$$
$$f'_y(x,y) = 2y + \frac{2y-2}{x^2+y^2+2x-2y+3}$$

$$\begin{cases} 6x + \dfrac{2x+2}{x^2+y^2+2x-2y+3} = 0 \\ 2y + \dfrac{2y-2}{x^2+y^2+2x-2y+3} = 0 \end{cases}$$

```
x = -0.129812
y = 0.309169
```

Рис. 5. Решение системы

$$f''_{xx}(x,y) = 6 + \frac{2(x^2+y^2+2x-2y+3)-(2x+2)(2x+2)}{(x^2+y^2+2x-2y+3)^2}$$
$$f''_{xx}(x,y) = 6 + \frac{-2x^2+2y^2-4x-4y+2}{(x^2+y^2+2x-2y+3)^2}$$

$$f''_{xy}(x,y) = -\frac{(2x+2)(2y-2)}{(x^2+y^2+2x-2y+3)^2}$$

$$f''_{yy}(x,y) = 2 + \frac{2(x^2+y^2+2x-2y+3)-(2y-2)(2y-2)}{(x^2+y^2+2x-2y+3)^2}$$
$$f''_{yy}(x,y) = 2 + \frac{2x^2-2y^2+4x+4y+2}{(x^2+y^2+2x-2y+3)^2}$$

$$A = f''_{xx}(M)$$
$$B = f''_{xy}(M)$$
$$C = f''_{yys}(M)$$
$$D = AC - B^2$$

```
A = 6.288419
B = 0.481608
C = 2.512723
D = 15.569108
```

Рис. 6. Проверка достаточного условия экстремумы

$$D > 0 \to M - \text{экстремум}$$
$$A > 0 \to M - \text{точка минимума}$$

Задание 6.7.

Указанным в индивидуальном варианте методом найти минимум квадратичной функции $f(x,y) = a_{11}x^2 + 2a_{12}xy + a_{22}y^2 + 2a_{13}x + 2a_{23}y$ с точностью $\varepsilon = 10^{-6}$. Для решения задачи многомерной минимизации использовать метод Ньютона. Построить график функции f. Предусмотреть подсчет числа итераций, потребовавшихся для достижения заданной точности.

$$a_{11} = 1$$
$$2a_{12} = 0.5$$
$$a_{22} = 2.5$$
$$2a_{13} = -2$$
$$2a_{23} = -10.5$$

Метод – сопряженных направлений

Решение:

$$f(x, y) = x^2 + 0.5xy + 2.5y^2 - 2x - 10.5y$$

$$\frac{\partial f}{\partial x} = 2x + 0.5y - 2$$
$$\frac{\partial f}{\partial y} = 0.5x + 5y - 10.5$$

$$\frac{\partial^2 f}{\partial x^2} = 2$$
$$\frac{\partial^2 f}{\partial y^2} = 5$$

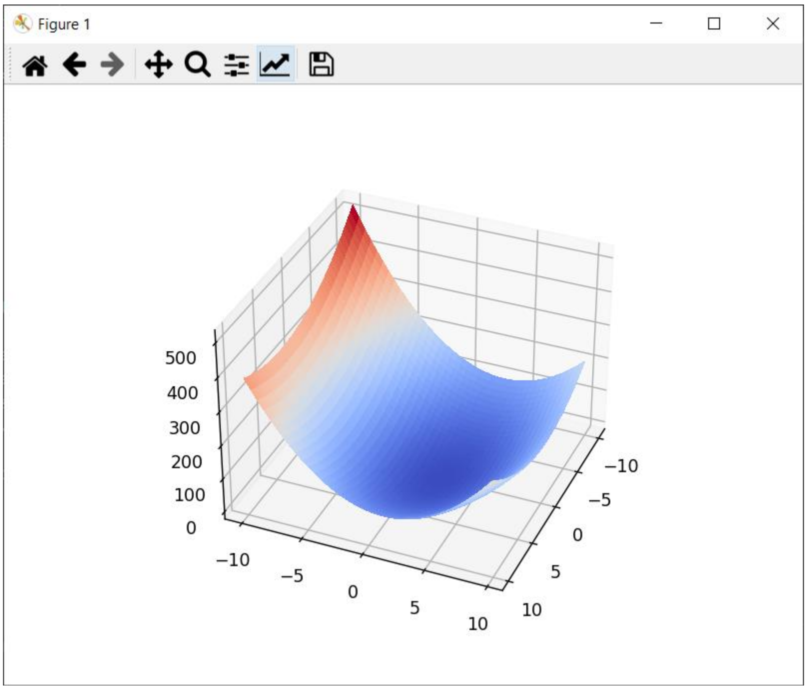

Рис. 6. График функции

Возьмем за исходную точку $A_0(0, 3)$.

$$p_x^0 = \left(\frac{\partial f}{\partial x}\right)_{A_0}$$
$$p_y^0 = \left(\frac{\partial f}{\partial y}\right)_{A_0}$$

$$\lambda = \frac{\left(\frac{\partial f}{\partial x} p_x^0\right)_{A_0} + \left(\frac{\partial f}{\partial y} p_y^0\right)_{A_0}}{\left((p_x^0)^2 + \left(p_y^0\right)^2\right)_{A_0} \begin{vmatrix} \frac{\partial^2 f}{\partial x^2} & 0 \\ 0 & \frac{\partial^2 f}{\partial y^2} \end{vmatrix}_{A_0}}$$

$$x_1 = x_0 - \lambda p_x^0$$
$$y_1 = y_0 - \lambda p_y^0$$

Следующие вычисления будут проводиться в цикле, пока $\sqrt{\left(p_x^k\right)^2 + \left(p_y^k\right)^2} > \varepsilon$, где k – текущая итерация

$$p_x^k = \left(\frac{\partial f}{\partial x}\right)_{A_k} + p_x^{k-1} \frac{\left(\frac{\partial f}{\partial x}\right)_{A_k}^2 + \left(\frac{\partial f}{\partial y}\right)_{A_k}^2}{\left(\frac{\partial f}{\partial x}\right)_{A_{k-1}}^2 + \left(\frac{\partial f}{\partial y}\right)_{A_{k-1}}^2}$$
$$p_y^k = \left(\frac{\partial f}{\partial y}\right)_{A_k} + p_y^{k-1} \frac{\left(\frac{\partial f}{\partial x}\right)_{A_k}^2 + \left(\frac{\partial f}{\partial y}\right)_{A_k}^2}{\left(\frac{\partial f}{\partial x}\right)_{A_{k-1}}^2 + \left(\frac{\partial f}{\partial y}\right)_{A_{k-1}}^2}$$

$$\lambda = \frac{\left(\frac{\partial f}{\partial x} p_x^k\right)_{A_k} + \left(\frac{\partial f}{\partial y} p_y^k\right)_{A_k}}{\left(\left(p_x^k\right)^2 + \left(p_y^k\right)^2\right)_{A_k} \begin{vmatrix} \frac{\partial^2 f}{\partial x^2} & 0 \\ 0 & \frac{\partial^2 f}{\partial y^2} \end{vmatrix}_{A_k}}$$

$$x_{k+1} = x_k - \lambda p_x^k$$
$$y_{k+1} = y_k - \lambda p_y^k$$

Результаты вычислений:

$$x = 0.487179$$
$$y = 2.051282$$

$$f(x,y) = -11.256410$$
$$n = 74$$

Решение методом Ньютона:

$$\begin{pmatrix} x \\ y \end{pmatrix}^{k+1} = \begin{pmatrix} x \\ y \end{pmatrix}^{k} - H^{-1}(f(x,y)^k)\nabla f(x,y)^k$$

$$H(f) = \begin{pmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial x^2} \end{pmatrix} = \begin{pmatrix} 2 & 0.5 \\ 0.5 & 5 \end{pmatrix}$$

$$|H| = 9.75$$
$$H^{-1} = \frac{1}{9.75}\begin{pmatrix} 5 & -0.5 \\ -0.5 & 2 \end{pmatrix}$$

$$\nabla \mathrm{f} = \begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{pmatrix} = \begin{pmatrix} 2x + 0.5y - 2 \\ 0.5x + 5y - 10.5 \end{pmatrix}$$

Результаты вычислений:

$$x = 0.487179$$
$$y = 2.051282$$
$$f(x,y) = -11.256410$$
$$n = 1$$

Вывод: в ходе выполнения лабораторной работы были получены практические навыки анализа возможностей построения и выделения наиболее важных свойств объектов моделей для моделирования и использования специализированных программных пакетов и библиотек для решения задачи минимизации функции и визуализации результатов решения.

ПРИЛОЖЕНИЯ

Листинг:

Task_1.py

```
import math, numpy
import matplotlib.pyplot as plt

f = lambda x: x**3 - math.exp(x)
a = 0
b = 3
eps = 10**-6

df = lambda x: 3*x**2 - math.exp(x)
d2f = lambda x: 6*x - math.exp(x)

x = (a + b) / 2

err = None
n = 0
while err == None or err > eps:
    if err == None:
        err = abs(df(x))
        continue
    err = abs(df(x))
    x = x - df(x)/d2f(x)
    n += 1

print(f"x = {x:.6f}; f(x) = {f(x):.6f}; n = {n}")

X = numpy.linspace(a, b, (b-a)*10)
Y = [f(x) for x in X]
plt.plot(X, Y)
plt.plot(x, f(x), 'ro', label='Минимум функции')
plt.legend()
plt.show()
```

Task_2.py

```
import math, numpy
import matplotlib.pyplot as plt

f = lambda x: math.cos(math.exp(x))
x_1 = 1
x_2 = 2
vareps = 10**-6

eps = 0.01
l = 2*vareps

def Fibonacci(f):
    a = x_1
```

```
    b = x_2
    F_x = (b - a)/l

    F = [1, 1]
    while F[-1] < F_x:
        F.append(F[-1] + F[-2])
    n = len(F) - 1

    lmb = a + F[-3] / F[-1] * (b - a)
    mu = a + F[-2] / F[-1] * (b - a)
    k = 0

    while k != n - 2:
        k += 1
        if f(lmb) > f(mu):
            a = lmb
            lmb = mu
            mu = a + F[-k-2] / F[-k-1] * (b - a)
        else:
            b = mu
            mu = lmb
            lmb = a + F[-k-3] / F[-k-1] * (b - a)

    mu = lmb + eps
    if f(lmb) == f(mu):
        a = lmb
    else:
        b = mu

    print(f"n = {n}")
    return (a + b) / 2

x_min = Fibonacci(f)
print(f"x = {x_min:.6f}; f(x) = {f(x_min):.6f}")
x_max = Fibonacci(lambda x: -f(x))
print(f"x = {x_max:.6f}; f(x) = {f(x_max):.6f}")

X = numpy.linspace(x_1, x_2, (x_2-x_1)*100)
Y = [f(x) for x in X]
plt.plot(X, Y)

plt.plot(x_min, f(x_min), 'go', label='Минимум функции')
plt.plot(x_max, f(x_max), 'ro', label='Максимум функции')
plt.legend()

plt.show()
```

Task_3.py

```
import matplotlib.pyplot as plt
from matplotlib import cm
import numpy as np
```

```python
from scipy import optimize

x_1 = -2
x_2 = 2
y_1 = -2
y_2 = 2
eps = 10**-6

fig, ax = plt.subplots(subplot_kw={"projection": "3d"})

X = np.arange(x_1, x_2, 0.25)
Y = np.arange(y_1, y_2, 0.25)
X, Y = np.meshgrid(X, Y)
Z = 3*X**2 + Y**2 + np.log(X**2 + Y**2 + 2*X - 2*Y + 3)

surf = ax.plot_surface(X, Y, Z, cmap=cm.coolwarm,
                       linewidth=0, antialiased=False)
plt.show()

plt.style.use('_mpl-gallery-nogrid')
levels = np.linspace(Z.min(), Z.max(), 15)
fig, ax = plt.subplots()
ax.contourf(X, Y, Z, levels=levels, cmap=cm.coolwarm, antialiased=False)
plt.show()

def equations(vars):
    x, y = vars
    eq1 = 6*x + (2*x + 2) / (x**2 + y**2 + 2*x - 2*y + 3)
    eq2 = 2*y + (2*y - 2) / (x**2 + y**2 + 2*x - 2*y + 3)
    return [eq1, eq2]

x, y = optimize.fsolve(equations, (-0.1, 0.3), xtol=eps)
print(f'x = {x:.6f}')
print(f'y = {y:.6f}')

A = 6 + (-2*x**2 + 2*y**2 - 4*x - 4*y + 2) / (x**2 + y**2 + 2*x - 2*y + 3)**2
B = -(2*x + 2) * (2*y - 2) / (x**2 + y**2 + 2*x - 2*y + 3)**2
C = 2 + (2*x**2 - 2*y**2 + 4*x + 4*y + 2) / (x**2 + y**2 + 2*x - 2*y + 3)**2

print(f'A = {A:.6f}')
print(f'B = {B:.6f}')
print(f'C = {C:.6f}')

D = A*C - B**2
print(f'D = {D:.6f}')
```

Task_4.py

```python
import numpy as np
from matplotlib import cm
import matplotlib.pyplot as plt
import math
```

```
eps = 10**-6

f = lambda x, y: x**2 + 0.5*x*y + 2.5*y**2 - 2*x - 10.5*y

df_x = lambda x, y: 2*x + 0.5*y - 2
df_y = lambda x, y: 0.5*x + 5*y - 10.5

d2f_x = 2
d2f_y = 5
det = d2f_x * d2f_y

x = [0]
y = [3]

p = [(df_x(x[0], y[0]), df_y(x[0], y[0]))]

lmd = (df_x(x[0], y[0]) * p[0][0] + df_y(x[0], y[0]) * p[0][1]) / ((p[0][0]**2 +
p[0][1]**2) * det)

x.append(x[0] - lmd * p[0][0])
y.append(y[0] - lmd * p[0][1])

n = 0
while math.sqrt(p[-1][0]**2 + p[-1][1]**2) > eps:
    frac = (df_x(x[-1], y[-1])**2 + df_y(x[-1], y[-1])**2) / (df_x(x[-2], y[-
2])**2 + df_y(x[-2], y[-2])**2)
    p_1 = df_x(x[-1], y[-1]) + p[-1][0] * frac
    p_2 = df_y(x[-1], y[-1]) + p[-1][1] * frac
    p.append((p_1, p_2))

    lmd = (df_x(x[-1], y[-1]) * p_1 + df_y(x[-1], y[-1]) * p_2) / ((p_1**2 +
p_2**2) * det)

    x.append(x[-1] - lmd * p_1)
    y.append(y[-1] - lmd * p_2)

    n += 1

print(f"x = {x[-1]:.6f}")
print(f"y = {y[-1]:.6f}")
print(f"f(x, y) = {f(x[-1], y[-1]):.6f}")
print(f"n = {n}")


x = [0]
y = [3]

det_inv = 1 / 9.75
H_inv = [[det_inv * 5, det_inv * -0.5],
         [det_inv * -0.5, det_inv * 2]]
```

```
n = 0
while math.sqrt(df_x(x[-1], y[-1])**2 + df_y(x[-1], y[-1])**2) > eps:
    x_k = x[-1] - (H_inv[0][0] * df_x(x[-1], y[-1]) + H_inv[0][1] * df_y(x[-1],
y[-1]))
    y_k = y[-1] - (H_inv[1][0] * df_x(x[-1], y[-1]) + H_inv[1][1] * df_y(x[-1],
y[-1]))

    x.append(x_k)
    y.append(y_k)

    n += 1

print(f"x = {x[-1]:.6f}")
print(f"y = {y[-1]:.6f}")
print(f"f(x, y) = {f(x[-1], y[-1]):.6f}")
print(f"n = {n}")

fig, ax = plt.subplots(subplot_kw={"projection": "3d"})

X = np.arange(-10, 10, 0.25)
Y = np.arange(-10, 10, 0.25)
X, Y = np.meshgrid(X, Y)
Z = X**2 + 0.5*X*Y + 2.5*Y**2 - 2*X - 10.5*Y

surf = ax.plot_surface(X, Y, Z, cmap=cm.coolwarm,
                       linewidth=0, antialiased=False)
plt.show()
```