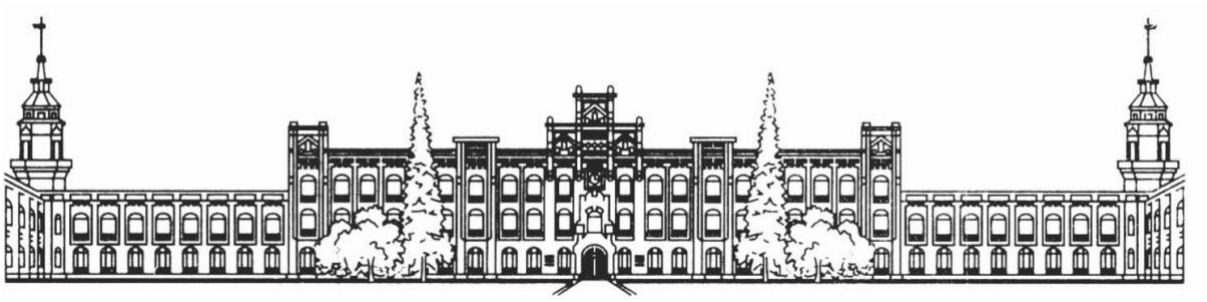


Міністерство освіти і науки України Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»
Інститут прикладного системного аналізу

Методи оптимізації 2
Лабораторна робота №2
«Числові методи безумовної оптимізації другого порядку.
Метод Ньютона та його варіації»
Бригада (варіант) №6

**Виконали:**
Студенти 3 курсу
Групи КА-03
Товкач Максим
Муравський Ігор
Страшук Віталій

**Перевірили:**
Яковлева Алла Петрівна
Спекторський Ігор Якович

Київ-2023

**Мета роботи:** реалізувати класичний та узагальнений метод Ньютона задля знаходження точки мінімуму заданої функції.

**Завдання лабораторної роботи**

1. У таблиці варіантів завдань знайти цільову функцію згідно з номером варіанта.
2. Скласти програму для мінімізації цільової функції одним із методів другого порядку (типу Ньютона). Конкретний тип градієнтного методу обрати самостійно, ураховуючи особливості функції (наприклад, ярність).

Під час складання програми треба:

- обчислити цільову функцію в окремій підпрограмі;
- частинні похідні цільової функції обчислити числово.

3. Знайти мінімум заданої цільової функції.

# Теоретичні відомості:

Метод Ньютона та його модифікації – найефективніший засіб числового розв'язання задач безумовної оптимізації. Надалі припускається, що функція $f$ строго опукла і двічі диференційовна на $\mathbb{R}^n$, причому матриця $f''(x)$ невироджена на $\mathbb{R}^n$. У методі Ньютона послідовність $x^1, x^2, \ldots, x^k, \ldots$ будують за правилом

$$x^{k+1} = x^k + h^k, \tag{2.1}$$
$$h^k = -[f''(x^k)]^{-1} \cdot f'(x^k), \quad k = 0, 1, \ldots$$

Отже, метод Ньютона – це метод мінімізації другого порядку. Як видно з (2.1), довжина кроку $\alpha_k = 1$; напрям, що визначається вектором $h^k$, є напрямом спадання функції $f$.

Квадратична апроксимація заданої функції в малому околі деякої точки значно точніша за лінійну апроксимацію. Тому в методі Ньютона природно сподіватися на більш точне наближення до розв'язку, ніж у градієнтному методі. Наведемо теорему про збіжність методу Ньютона.

**Теорема 2.1.** *Нехай функція $f$ двічі диференційовна, сильно опукла з константою $\Theta > 0$ на $\mathbb{R}^n$ та задовольняє умову*

$$\|f''(x) - f''(\widetilde{x})\| \le M\|x - \widetilde{x}\|,$$

*де $x, \widetilde{x} \in \mathbb{R}^n$, $M > 0$, а початкова точка $x^0$ така, що:*

$$\|f'(x^0)\| \le \frac{8\Theta^2}{M} \text{ або } \|f'(x^0)\| \le \frac{8\Theta^2}{M} q, \text{ де } q \in (0; 1).$$

*Тоді послідовність* $x^k$ $(k \geq 1)$, *що визначається формулами* (2.1), *збігається до точки мінімуму* $x^*$ *функції* $f$ *з квадратичною швидкістю:*

$$\left\|x^k - x^*\right\| < \frac{4\Theta^2}{M} q^{2^k}.$$

Збіжність методу Ньютона доведена лише для достатньо хорошого початкового наближення $x^0$. Недоліками методу є також складність пошуку потрібного початкового наближення та великий обсяг обчислень (на кожному кроці потрібно обчислювати й обертати матрицю других похідних цільової функції).

Модифікації методу Ньютона спрямовані на те, щоб, зберігши його основну перевагу – швидку збіжність, зменшити обсяг обчислень і послабити вимоги до вибору початкового наближення. В *узагальненому методі Ньютона* (так званий *метод Ньютона з регулюванням кроку*) послідовність $x^k$ $(k \geq 1)$ будується за правилом

$$x^{k+1} = x^k + \alpha_k h^k, \quad \alpha_k > 0, \; h^k = -[f''(x^k)]^{-1} f'(x^k) \tag{2.2}$$

(якщо $\alpha_k = 1$, метод збігається з класичним методом Ньютона).

Метод (2.2) можна також подати у вигляді

$$f''(x^k) \cdot h^k = -f'(x^k); \; x^{k+1} = x^k + \alpha_k h^k.$$

Отже, для визначення вектора $h^k$ можна розв'язувати систему лінійних рівнянь замість того, щоб обертати матрицю $f''(x^k)$.

Розглянемо два варіанти узагальненого методу Ньютона, які розрізняються способом вибору параметра $\alpha$.

**Перший спосіб**. 1. Вважаємо, що $\alpha = 1$.

2. За обраного $\alpha$ обчислюємо точку $x = x^k + \alpha h^k$ та значення функції $f(x) = f(x^k + \alpha h^k)$.

3. Перевіряємо нерівність:

$$f(x) - f(x^k) \leq \varepsilon\alpha \left\langle f'(x^k), h^k \right\rangle, \quad 0 < \varepsilon < \frac{1}{2}$$

($\varepsilon$ – довільна константа, однакова для всіх $k = 0, 1, 2, \ldots$).

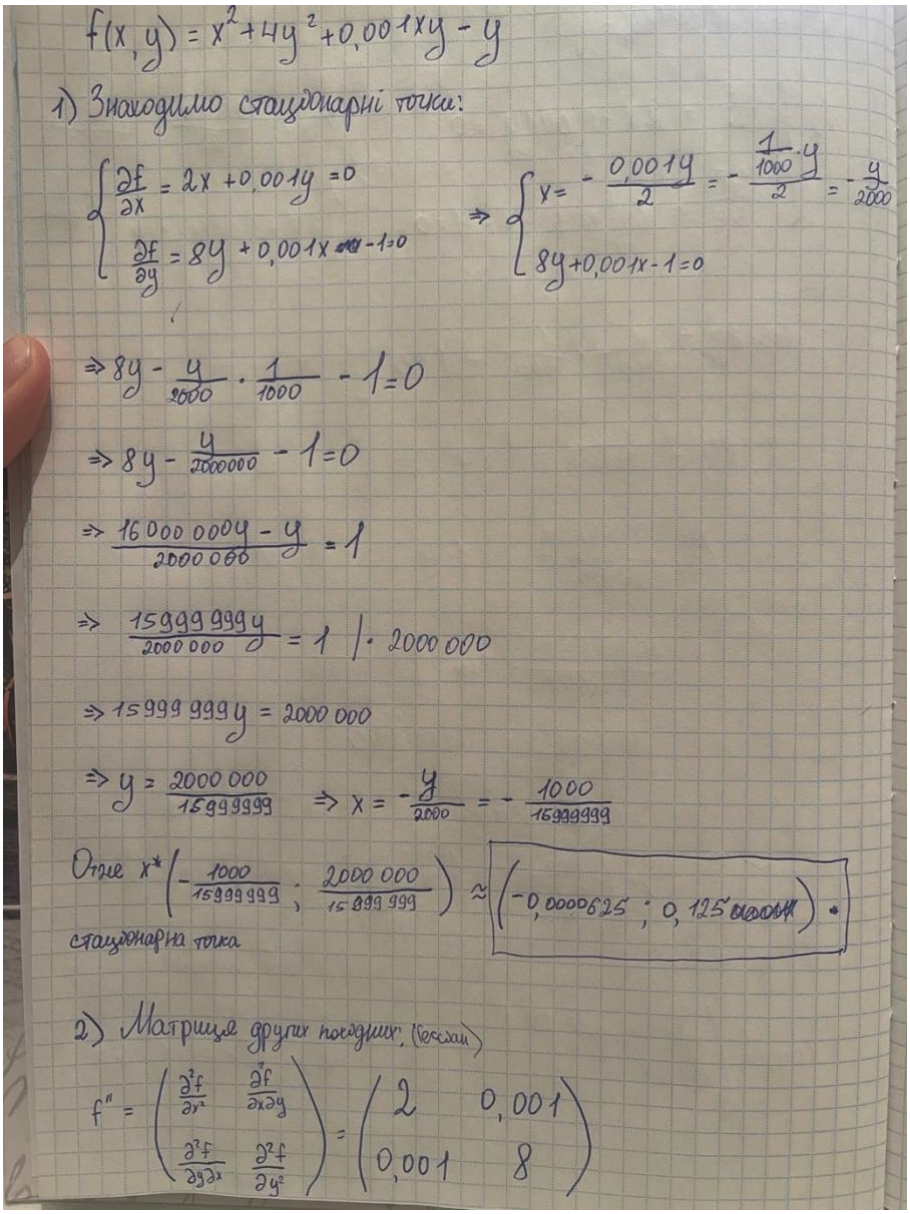
4. Якщо нерівність п. 3 виконується, беремо $\alpha_k = \alpha = 1$ і закінчуємо роботу алгоритму; інакше, виконуємо дроблення $\alpha$ і повертаємось до п. 2.

Для сильно опуклих двічі диференційовних функцій метод Ньютона збігається незалежно від вибору початкової точки $x^0$ із надлінійною або квадратичною швидкістю (за наявності додаткових умов на функцію $f$).

*Зауваження 2.1.* Для числового обчислення першої та другої похідної слід використовувати різницеві формули, що забезпечують порядок помилки не нижче за $h^2$, наприклад:

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} + O(h^2);$$
$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} + O(h^2).$$

## Аналітичний розв'язок задачі:



$$f''(x^*) = \begin{pmatrix} 2 & 0{,}001 \\ 0{,}001 & 8 \end{pmatrix} = A^*$$

3) Кутові мінори:

$M_1^1 = 2 > 0$

$M_{12}^{12} = 16 - (0{,}001)^2 > 0$

Отже, за критерієм Сильвестра, матриця $A^*$ є додатно визначеною

$\Rightarrow x^*$ – точка строгого мін.

4) Функція є опуклою (оскільки всі мінори $> 0$)

5) ~~$f(x^*) \approx -0{,}0625$~~

Запишемо функцію у вигляді $f(\bar{x}) = (A\bar{x}, \bar{x}) + (b, \bar{x})$ (квадратична форма)

$$f(x,y) = \left( \begin{pmatrix} 1 & 0{,}001 \\ 0{,}001 & 4 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}, \begin{pmatrix} x \\ y \end{pmatrix} \right) + \left( \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \begin{pmatrix} x \\ y \end{pmatrix} \right)$$

Отже $A = \begin{pmatrix} 1 & 0{,}001 \\ 0{,}001 & 4 \end{pmatrix}$

$M_1^1 = 1 > 0$

$M_{12}^{12} = 4 - (0{,}001)^2 > 0$

Всі кутові мінори матр. $A > 0$

$\Rightarrow$ За критерієм Сильвестра $A$ – додатно визнач.

$\Rightarrow (Ax, x) > 0$ (при $x \neq 0$)

$\Rightarrow$ функція має т. глоб. мін

6) Отже $x^*$ – точка строгого глоб. мін

7) $\boxed{f(x^*) \approx -0{,}0625}$

(-0.0000625)² + 4 × (0.1250001)² + 0.001 × (-0.0000625) × (0.1250001) - 0.1250001 =

-0.0625000039

| | | | | | | |
|---|---|---|---|---|---|---|
| Rad \| Deg | | x! | ( | ) | % | AC |
| Inv | sin | ln | 7 | 8 | 9 | ÷ |
| π | cos | log | 4 | 5 | 6 | × |
| e | tan | √ | 1 | 2 | 3 | − |
| Ans | EXP | $x^y$ | 0 | . | = | + |

Отзыв

## Вибір довжини кроку a:

1)Якщо розв’язувати задачу за допомогою класичного методу Ньютона, то a буде рівна 1.

2)Якщо розглядати узагальнений метод Ньютона (метод Ньютона з регулюванням кроку), то крок a ми знаходимо за допомогою методу дроблення кроку (find_a). Ми задаємо початкове a=1 та деяке eps з проміжку (0;0.5). В нашому випадку eps=0.1 . За обраного a обчислюємо $x^{k+1}$ та $y^{k+1}$ за формулою

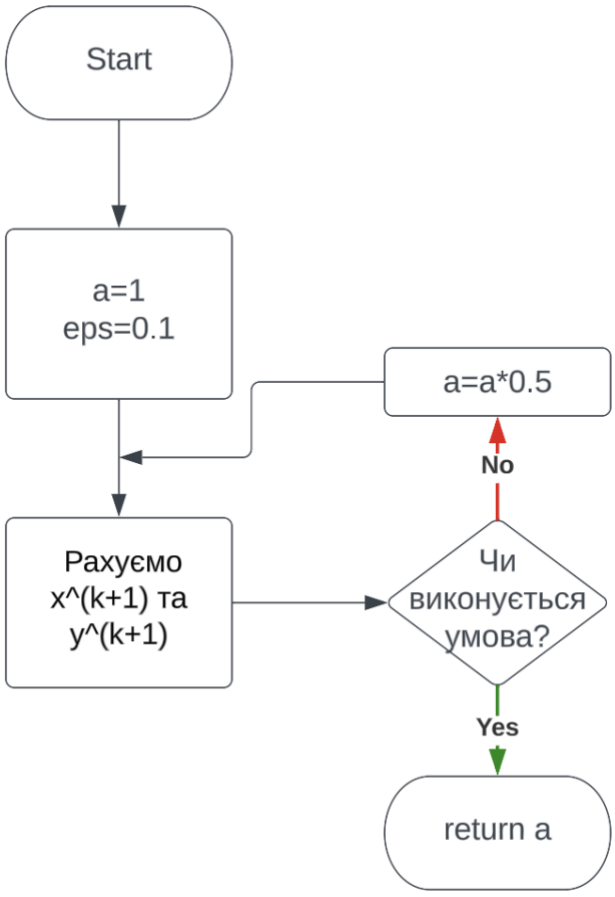
$$\bar{x}^{k+1} = \bar{x}^{k} + a * \bar{h}_k\ ,\ \text{де}\ \bar{h}_k = -\left(f''(\bar{x}^{k})\right)^{-1} * f'(\bar{x}^{k})$$

Потім перевіряємо чи справджується нерівність

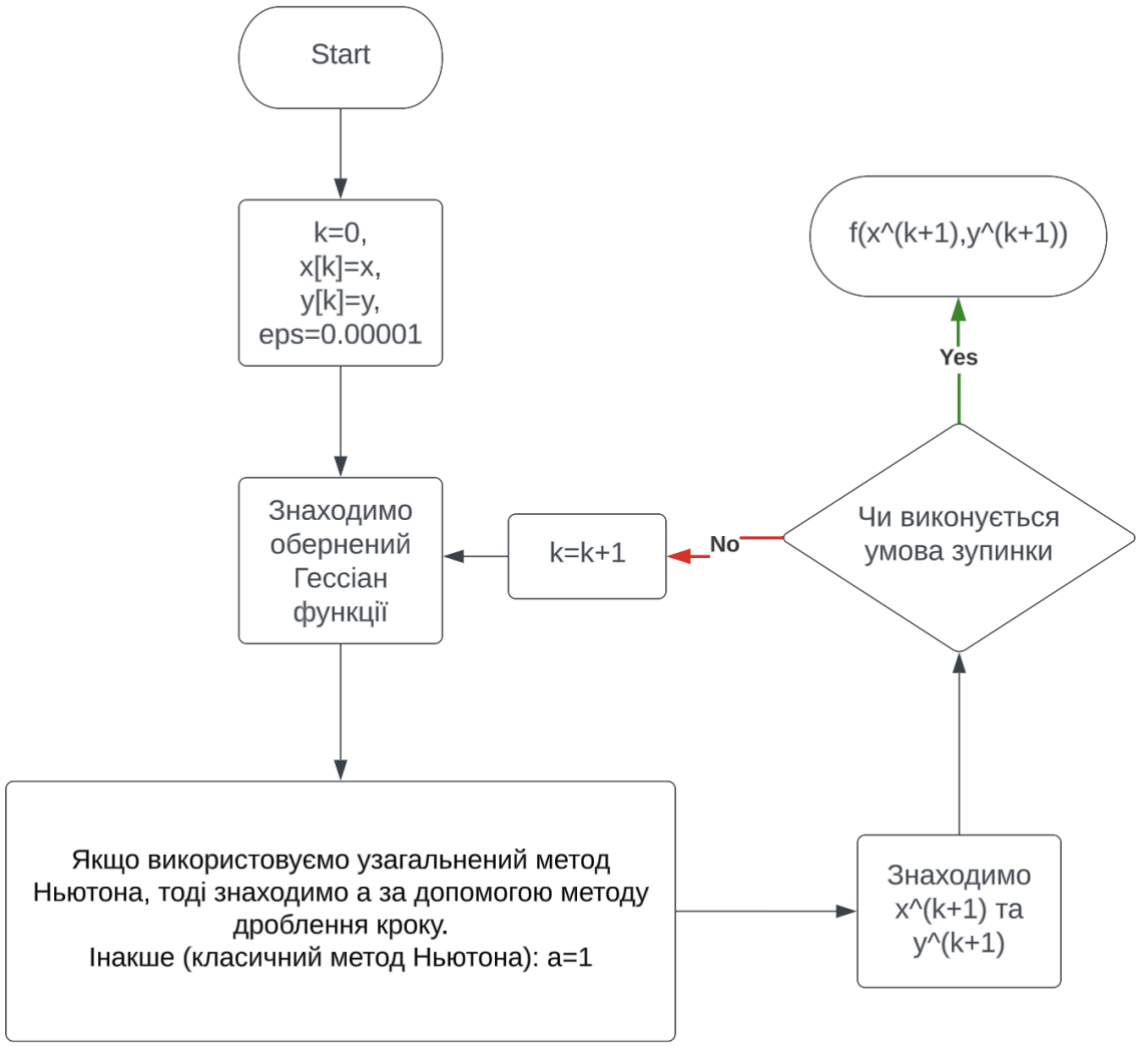
$$f(\bar{x}^{k+1}) - f(\bar{x}^{k}) \leq eps * a * (f'(\bar{x}^{k}), \bar{h}_k)$$

Якщо так, то забираємо a, якщо ні – домножуемо a на 0.5 та знову переходимо до пошуку $x^{k+1}$ та $y^{k+1}$.

Блок-схема алгоритму пошуку довжини кроку:



Блок-схема всього алгоритму:




В нашому випадку Гесіан функції не залежить від координат точки (є сталим). Тому його можно не рахувати на кожній ітерації.

Задля знаходження $\bar{x}^{k+1}$ використовується формула:

$\bar{x}^{k+1} = \bar{x}^k + a * \bar{h}_k$ ,

де $\bar{h}_k = -\left(f''(\bar{x}^k)\right)^{-1} * f'(\bar{x}^k)$

Умова зупинки: $\left|\bar{x}^{k+1} - \bar{x}^k\right| < eps$

**Код програми:**

```
#include <iostream>
#include <math.h>
using namespace std;

float T[1000][2];

float func(float x,float y){
    float f;
    f=pow(x,2)+4*pow(y,2)+0.001*x*y-y;//!
    return f;
}

float det_x(float x,float y){
    float f,h=0.01;
    f=(func(x+h,y)-func(x-h,y))/(2*h);
    return f;
}

float det_y(float x,float y){
    float f,h=0.01;
    f=(func(x,y+h)-func(x,y-h))/(2*h);
    return f;
}
```

```
float det_xx(float x, float y){
    float f,h=1;//0.1
    f=(func(x+h,y)-2*func(x,y)+func(x-h,y))/pow(h,2);
    return f;
}

float det_yy(float x, float y){
    float f,h=1;
    f=(func(x,y+h)-2*func(x,y)+func(x,y-h))/pow(h,2);
    return f;
}


float grad_x(float x, float y){
    float f;
    f = 2*x+0.001*y;
    return f;
}

float grad_y(float x, float y){
    float f;
    f = 8*y+0.001*x-1;
    return f;
}
```

```
float find_a(int k,float h_x, float h_y){//метод дроблення
    float a=1, eps=0.1;

    do{

        float x=T[k][0]+a*h_x;
        float y=T[k][1]+a*h_y;
        float f=func(x,y);


        if(f-func(T[k][0],T[k][1]) <=
eps*a*(det_x(T[k][0],T[k][1])*h_x+det_y(T[k][0],T[k][1])*h_y))break;
        a=a*0.5;

            }while(1);

    return a;
}




int main() {
```

```
float eps=0.00001;//0.0000001
float a=1,x,y;
int k=0;
float A[2][2];

cout<<"Введіть початкову координату x: ";
cin>>x;
cout<<"Введіть початкову координату y: ";
cin>>y;
cout<<endl<<endl;

T[0][0]=x;
T[0][1]=y;


while(1){

    A[1][0]=0.001;
    A[0][1]=0.001;
    A[0][0]=2;//!
    A[1][1]=8;//!

    float Atr[2][2];
    Atr[0][0]=A[1][1];
    Atr[1][0]=-A[1][0];
```

```
        Atr[0][1]=-A[0][1];
        Atr[1][1]=A[0][0];
        float det =A[0][0]*A[1][1]-A[1][0]*A[0][1];
        float Ar[2][2];
        Ar[0][0]=(1/det)*Atr[0][0];
        Ar[1][1]=(1/det)*Atr[1][1];
        Ar[1][0]=(1/det)*Atr[1][0];
        Ar[0][1]=(1/det)*Atr[0][1];



a=find_a(k,-
1*(Ar[0][0]*det_x(T[k][0],T[k][1])+Ar[0][1]*det_y(T[k][0],T[k][1])),-
1*(Ar[1][0]*det_x(T[k][0],T[k][1])+Ar[1][1]*det_y(T[k][0],T[k][1])));


        T[k+1][0]=T[k][0]-
a*(Ar[0][0]*det_x(T[k][0],T[k][1])+Ar[0][1]*det_y(T[k][0],T[k][1]));
        T[k+1][1]=T[k][1]-
a*(Ar[1][0]*det_x(T[k][0],T[k][1])+Ar[1][1]*det_y(T[k][0],T[k][1]));
        cout<<"k="<<k<<" a="<<a<<" x="<<T[k][0]<<" y="<<T[k][1]<<"
f="<<func(T[k][0],T[k][1])<<endl;
        if((abs(T[k+1][0]-T[k][0])<eps)&&(abs(T[k+1][1]-T[k][1])<eps))break;
        k++;
    }


    cout<<fixed;
```

```
    cout.precision(5);
    cout<<endl<<endl<<"Кількість ітерацій:
"<<k<<endl<<"F("<<T[k+1][0]<<"; "<<T[k+1][1]<<") = ";
    cout<<func(T[k+1][0],T[k+1][1])<<endl<<endl;


}
```

**Результати роботи:**

Початкова точка (1;1)
Використовуючи класичний метод Ньютона:

```
Введіть початкову координату x: 1
Введіть початкову координату y: 1


k=0 a=1 x=1 y=1 f=4.001
k=1 a=1 x=-6.22272e-05 y=0.125002 f=-0.0625


Кількість ітерацій: 1
F(-0.00006; 0.12500) = -0.06250
```

Використовуючи узагальнений метод Ньютона:

```
Введіть початкову координату x: 1
Введіть початкову координату y: 1


k=0 a=1 x=1 y=1 f=4.001
k=1 a=1.92593e-34 x=-6.22272e-05 y=0.125002 f=-0.0625


Кількість ітерацій: 1
F(-0.00006; 0.12500) = -0.06250
```

Початкова точка (100000;100000)

Використовуючи класичний метод Ньютона:

```
Введіть початкову координату x: 100000
Введіть початкову координату y: 100000


k=0 a=1 x=100000 y=100000 f=5.00099e+10
k=1 a=1 x=-2361.62 y=23212.8 f=2.16083e+09
k=2 a=1 x=-2350.42 y=812.787 f=8.16423e+06
k=3 a=1 x=-0.0107422 y=-0.0067749 f=0.00707397
k=4 a=1 x=-6.24983e-05 y=0.125 f=-0.0625


Кількість ітерацій: 4
F(-0.00006; 0.12500) = -0.06250
```

Використовуючи узагальнений метод Ньютона:

```
Введіть початкову координату x: 100000
Введіть початкову координату y: 100000


k=0 a=1 x=100000 y=100000 f=5.00099e+10
k=1 a=1 x=-2361.62 y=23212.8 f=2.16083e+09
k=2 a=1 x=-2350.42 y=812.787 f=8.16423e+06
k=3 a=1 x=-0.0107422 y=-0.0067749 f=0.00707397
k=4 a=3.9443e-31 x=-6.24983e-05 y=0.125 f=-0.0625


Кількість ітерацій: 4
F(-0.00006; 0.12500) = -0.06250
```

Початкова точка (0.126; 728.15)

Використовуючи класичний метод Ньютона:

```
Введіть початкову координату x: 0.26
Введіть початкову координату y: 728.15


k=0 a=1 x=0.26 y=728.15 f=2.12008e+06
k=1 a=1 x=0.624063 y=0.0249634 f=0.366999
k=2 a=1 x=-6.15716e-05 y=0.125 f=-0.0625


Кількість ітерацій: 2
F(-0.00006; 0.12500) = -0.06250
```

Використовуючи узагальнений метод Ньютона:

```
Введіть початкову координату x: 0.126
Введіть початкову координату y: 728.15


k=0 a=1 x=0.126 y=728.15 f=2.12008e+06
k=1 a=1 x=0.490063 y=0.0249634 f=0.217703
k=2 a=1.2326e-32 x=-6.2108e-05 y=0.125 f=-0.0625


Кількість ітерацій: 2
F(-0.00006; 0.12500) = -0.06250
```

Початкова точка (-732; 1830)

Використовуючи класичний метод Ньютона:

```
Введіть початкову координату x: -732
Введіть початкову координату y: 1830


k=0 a=1 x=-732 y=1830 f=1.39283e+07
k=1 a=1 x=-6.08429 y=-1.34082 f=45.5588
k=2 a=1 x=8.7738e-05 y=0.125003 f=-0.0625
k=3 a=1 x=-6.24826e-05 y=0.125 f=-0.0625


Кількість ітерацій: 3
F(-0.00006; 0.12500) = -0.06250
```

Використовуючи узагальнений метод Ньютона:

```
Введіть початкову координату x: -732
Введіть початкову координату y: 1830


k=0 a=1 x=-732 y=1830 f=1.39283e+07
k=1 a=1 x=-6.08429 y=-1.34082 f=45.5588
k=2 a=1 x=8.7738e-05 y=0.125003 f=-0.0625
k=3 a=3.9443e-31 x=-6.24826e-05 y=0.125 f=-0.0625


Кількість ітерацій: 3
F(-0.00006; 0.12500) = -0.06250
```

Тепер спробуємо нашу програму для «негарної» функції.
Візьмемо $f(x,y) = 100 * x^2 + y^2 + 0.001 * x * y - 1$
Гессіан цієї функції дорівнює $\begin{pmatrix} 200 & 0.001 \\ 0.001 & 2 \end{pmatrix}$
(відношення a22 до a11=1/100)
Отже матриця є погано обумовленою.

**Результати:**
Початкова точка (1; 1)

Використовуючи класичний метод Ньютона:

```
Введіть початкову координату x: 1
Введіть початкову координату y: 1


k=0 a=1 x=1 y=1 f=100.001
k=1 a=1 x=-1.3113e-06 y=0.500012 f=-0.25
k=2 a=1 x=-2.49961e-06 y=0.5 f=-0.25


Кількість ітерацій: 2
F(-0.00000; 0.50000) = -0.25000
```

Використовуючи узагальнений метод Ньютона:

```
Введіть початкову координату x: 1
Введіть початкову координату y: 1


k=0 a=1 x=1 y=1 f=100.001
k=1 a=1.20371e-35 x=-1.3113e-06 y=0.500012 f=-0.25


Кількість ітерацій: 1
F(-0.00000; 0.50001) = -0.25000
```

Початкова точка (10000; 10000)

Використовуючи класичний метод Ньютона:

```
Введіть початкову координату x: 10000
Введіть початкову координату y: 10000


k=0 a=1 x=10000 y=10000 f=1.01001e+10
k=1 a=1 x=272 y=10004.9 f=1.07488e+08
k=2 a=1 x=0.0490112 y=205 f=41820.2
k=3 a=1 x=0.000228517 y=0.605499 f=-0.238865
k=4 a=1 x=-2.50117e-06 y=0.5 f=-0.25


Кількість ітерацій: 4
F(-0.00000; 0.50000) = -0.25000
```

Використовуючи узагальнений метод Ньютона:

```
Введіть початкову координату x: 10000
Введіть початкову координату y: 10000


k=0 a=1 x=10000 y=10000 f=1.01001e+10
k=1 a=1 x=272 y=10004.9 f=1.07488e+08
k=2 a=1 x=0.0490112 y=205 f=41820.2
k=3 a=1 x=0.000228517 y=0.605499 f=-0.238865
k=4 a=2.46519e-32 x=-2.50117e-06 y=0.5 f=-0.25


Кількість ітерацій: 4
F(-0.00000; 0.50000) = -0.25000
```

Початкова точка (51322; 0.51)

Використовуючи класичний метод Ньютона:

```
Введіть початкову координату x: 51322
Введіть початкову координату y: 0.51


k=0 a=1 x=51322 y=0.51 f=2.63395e+11
k=1 a=1 x=-6022 y=29.182 f=3.62645e+09
k=2 a=1 x=-134 y=26.238 f=1.79626e+06
k=3 a=1 x=-0.0936127 y=-1.95395 f=6.64839
k=4 a=1 x=-2.49594e-06 y=0.499997 f=-0.25


Кількість ітерацій: 4
F(-0.00000; 0.50000) = -0.25000
```

Використовуючи узагальнений метод Ньютона:

```
Введіть початкову координату x: 51322
Введіть початкову координату y: 0.51


k=0 a=1 x=51322 y=0.51 f=2.63395e+11
k=1 a=1 x=-6022 y=29.182 f=3.62645e+09
k=2 a=1 x=-134 y=26.238 f=1.79626e+06
k=3 a=1 x=-0.0936127 y=-1.95395 f=6.64839
k=4 a=3.85186e-34 x=-2.49594e-06 y=0.499997 f=-0.25


Кількість ітерацій: 4
F(-0.00000; 0.50000) = -0.25000
```

**Висновок**

З результатів роботи програми можемо побачити, що розв’язки з усіма взятими початковими точками співпали між собою, а також з тим розв’язком, який ми знайшли аналітично (з похибкою 10^(-5)). Аналізуючи кількість кроків ми можемо зробити висновок, що метод Ньютона є дуже швидким (адже він має квадратичну швидкість збіжності) і є набагато швидшим за градієнтний метод (який має лінійну швидкість збіжності). Ось порівняння градієнтного методу (з використанням методу найшвидшого спуску) та методу класичного Ньютона з початковою точкою (1; 1):

```
Введіть початкову координату x: 1
Введіть початкову координату y: 1


k=0 a=0.130859 x=1 y=1 f=4.001
k=1 a=0.435547 x=0.73815 y=0.0838557 f=0.489199
k=2 a=0.126953 x=0.0951164 y=0.226897 f=-0.0118995
k=3 a=0.486328 x=0.0709369 y=0.123396 f=-0.0574489
k=4 a=0.123047 x=0.00187967 y=0.129603 f=-0.0624115
k=5 a=0.451172 x=0.00140116 y=0.125072 f=-0.0624978
k=6 a=0.123047 x=8.04309e-05 y=0.124812 f=-0.0624998
k=7 a=0.498047 x=4.52727e-05 y=0.124997 f=-0.0625
k=8 a=0.498047 x=-6.20604e-05 y=0.125009 f=-0.0625
k=9 a=0.123047 x=-6.25242e-05 y=0.124974 f=-0.0625
k=10 a=0.498047 x=-6.25242e-05 y=0.125 f=-0.0625


Кількість кроків: 10
F(-0.00006; 0.12500) = -0.06250
```

```
Введіть початкову координату x: 1
Введіть початкову координату y: 1


k=0 a=1 x=1 y=1 f=4.001
k=1 a=1 x=-6.22272e-05 y=0.125002 f=-0.0625


Кількість ітерацій: 1
F(-0.00006; 0.12500) = -0.06250
```

Бачимо, що метод Ньютона з точки (1; 1) збігається у 10 разів швидше ніж градієнтний меод з використанням методу найшвидшого спуску, та у 14 разів швидше за градієнтний метод з використанням дроблення кроку.

Якщо розглянути результати роботи з «негарною» функцією, то можемо побачити, що всі результати (з усіх взятих початкових точок) співпали. Цікавим є те, що при початковій точці (1; 1) узагальнений метод Ньютона був на 1 ітерацію швидшим за загальний метод Ньютона.
Хоч метод Ньютона і є найбільш ефективним для вирішення задач безумоної оптиматизації, він має і вади. Наприклад, в ньому, на відміну від градієнтного методу (першого порядку), треба на кожному кроці шукати матрицю других похідних (метод другого порядку).

| Початкова точка | Метод Ньютона | Кількість ітерацій | Координати точки мінінмума | Значення функції у цій точці |
|---|---|---|---|---|
| (1; 1) | Класичний | 1 | (-0.00006; 0.125) | -0.0625 |
| | Узагальнений | 1 | (-0.00006; 0.125) | -0.0625 |
| (100000; 100000) | Класичний | 4 | (-0.00006; 0.125) | -0.0625 |
| | Узагальнений | 4 | (-0.00006; 0.125) | -0.0625 |
| (0.26; 728.15) | Класичний | 2 | (-0.00006; 0.125) | -0.0625 |
| | Узагальнений | 2 | (-0.00006; 0.125) | -0.0625 |
| (-732; 1830) | Класичний | 3 | (-0.00006; 0.125) | -0.0625 |
| | Узагальнений | 3 | (-0.00006; 0.125) | -0.0625 |

Для «негарної функції»:

| Початкова точка | Метод вибору довжини кроку | Кількість ітерацій | Координати точки мінінмума | Значення функції у цій точці |
|---|---|---|---|---|
| (1; 1) | Класичний | 2 | (-0.000001; 0.50000) | -0.25 |
| | Узагальнений | 1 | (-0.000001; 0.50001) | -0.25 |
| (10000; 10000) | Класичний | 4 | (-0.000001; 0.5) | -0.25 |
| | Узагальнений | 4 | (-0.000001; 0.5) | -0.25 |
| (51322; 0.51) | Класичний | 4 | (-0.000001; 0.5) | -0.25 |
| | Узагальнений | 4 | (-0.000001; 0.5) | -0.25 |