

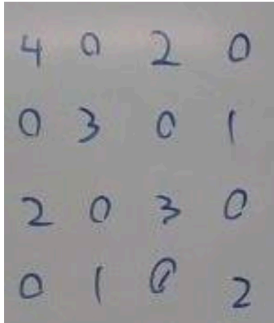
Київський національний університет імені Тараса Шевченка
Факультет комп'ютерних наук та кібернетики
Лабораторна робота №3
з дисципліни «Чисельні методи в інформатиці»
Тема: розв'язок систем нелінійних рівнянь та пошук власних значень.
Варіант 3

Виконала студентка третього курсу
Групи ІПС-32
Михайлова Софія

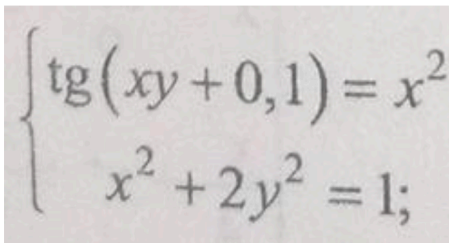
1) Постановка задачі

Варіант 3

Зайти найменше власне значення степеневим методом та наближення до всіх власних значень методом обертань Якобі (або виконати 3-4 ітерації):


$$A = \begin{pmatrix} 4 & 0 & 2 & 0 \\ 0 & 3 & 0 & 1 \\ 2 & 0 & 3 & 0 \\ 0 & 1 & 0 & 2 \end{pmatrix}$$

Розв'язати (або виконати 5 ітерацій) методом Ньютона:


$$\begin{cases} \operatorname{tg}(xy + 0,1) = x^2 \\ x^2 + 2y^2 = 1; \end{cases}$$

2.1) Теоретичні відомості та розрахунки

2.1.1) Степеневий метод для знаходження найменшого власного значення:

Степеневий метод — це ітераційний метод, який використовується для знаходження найбільшого за модулем власного значення матриці. Для симетричних матриць, зокрема для пошуку мінімального власного значення, цей метод є корисним і може бути реалізований як варіант для обчислення власних значень. Степеневий метод працює таким чином:

- Ітерація:** Початково задаємо вектор x_0 , і на кожній ітерації обчислюємо: $x^{k+1} = A \cdot x^k$, де x^{k+1} — це новий вектор, який отримується за допомогою матриці A і попереднього вектора x^k .
- Нормалізація:** Для забезпечення збіжності потрібно нормувати вектор: $x^{k+1} = A \cdot x^k / (\|A \cdot x^k\|)$
- Власне значення:** Як тільки вектор x^k сходиться, можна обчислити власне значення λ як: $\lambda = (x^{k+1} \cdot A \cdot x^k) / (x^k \cdot x^k)$. Це є наближенням до найбільшого за модулем власного значення.

2.1.2) Метод обертання Якобі:

Метод обертання Якобі використовується для знаходження всіх власних значень симетричних матриць. Він включає послідовність ортогональних перетворень, які призводять до діагоналізації матриці. Кроки методу:

1. **Нормування:** На кожній ітерації вибирається найбільший по модулю не діагональний елемент матриці A , після чого визначається кут обертання, за допомогою якого ці елементи можуть бути приведені до нуля.
2. **Обчислення кута:** Кут ϕ_k обчислюється з формули:

$$\tan(2\phi_k) = 2a_{ij} / (a_{ii} - a_{jj}),$$

де a_{ij} — це найбільший не діагональний елемент матриці.

3. **Матриця обертання:** Після визначення кута обертання обчислюється матриця U_k , яка застосовується до поточної матриці A_k для отримання нової матриці A_{k+1} :

$$A_{k+1} = U_k A_k U_k^T$$

4. **Ітерації:** Програма повторює кроки до досягнення бажаної точності або до певної кількості ітерацій.

Задача полягає у знаходженні найменшого власного значення за допомогою степеневого методу та всіх власних значень за допомогою методу обертання Якобі для матриці:

$$A = \begin{pmatrix} 4 & 0 & 2 & 0 \\ 0 & 3 & 0 & 1 \\ 2 & 0 & 3 & 0 \\ 0 & 1 & 0 & 2 \end{pmatrix}$$

2.1.3) Кроки розв'язку:

1. Метод степеневого розкладу:

Для початкової матриці A будемо використовувати степеневий метод для знаходження найбільшого власного значення. Як тільки власне значення стабілізується (різниця між ітераціями буде меншою за точність ϵ), воно буде взято за результат.

2. Метод обертання Якобі:

Метод Якобі буде використовуватися для знаходження всіх власних значень матриці. Вибиратимемо найбільший по модулю не діагональний елемент і виконуємо ортогональні обертання до тих пір, поки всі елементи поза діагоналлю не будуть близькими до нуля.

1. Початкова матриця A_0 :

$$A_0 = \begin{pmatrix} 4 & 0 & 2 & 0 \\ 0 & 3 & 0 & 1 \\ 2 & 0 & 3 & 0 \\ 0 & 1 & 0 & 2 \end{pmatrix}$$

2. Ітерації для методу обертання Якобі:

- Вибираємо найбільший елемент поза діагоналлю.
- Обчислюємо кут обертання ϕ_k .
- Оновлюємо матрицю A_k і матрицю U_k .
- Продовжуємо до досягнення бажаної точності.

2.1.4) Результати:

```
Initial Matrix A0:
[[4 0 2 0]
 [0 3 0 1]
 [2 0 3 0]
 [0 1 0 2]]
Eigenvalue (Power Iteration): 5.561552812808455
Eigenvector (Power Iteration): [7.88205438e-01 3.73804967e-07 6.15412209e-01 2.31024175e-07]
Iteration 1:
Matrix U (Rotation Matrix) at iteration 1:
[[ 0.78820544  0.          -0.61541221  0.          ]
 [ 0.          1.          0.          0.          ]
 [ 0.61541221  0.          0.78820544  0.          ]
 [ 0.          0.          0.          1.          ]]
Matrix A_1 (Updated Matrix) at iteration 1:
[[ 5.56155281e+00  0.00000000e+00 -2.70437804e-16  0.00000000e+00]
 [ 0.00000000e+00  3.00000000e+00  0.00000000e+00  1.00000000e+00]
 [-3.71898680e-16  0.00000000e+00  1.43844719e+00  0.00000000e+00]
 [ 0.00000000e+00  1.00000000e+00  0.00000000e+00  2.00000000e+00]]

Iteration 2:
Matrix U (Rotation Matrix) at iteration 2:
[[ 0.78820544  0.          -0.61541221  0.          ]
 [ 0.          0.85065081  0.          -0.52573111]
 [ 0.61541221  0.          0.78820544  0.          ]
 [ 0.          0.52573111  0.          0.85065081]]
Matrix A_2 (Updated Matrix) at iteration 2:
[[ 5.56155281e+00  0.00000000e+00 -2.70437804e-16  0.00000000e+00]
 [ 0.00000000e+00  3.61803399e+00  0.00000000e+00 -3.86220340e-17]
 [-3.71898680e-16  0.00000000e+00  1.43844719e+00  0.00000000e+00]
 [ 0.00000000e+00 -2.70369227e-16  0.00000000e+00  1.38196601e+00]]

Iteration 3:
Matrix U (Rotation Matrix) at iteration 3:
[[ 0.78820544  0.          -0.61541221  0.          ]
 [ 0.          0.85065081  0.          -0.52573111]
 [ 0.61541221  0.          0.78820544  0.          ]
 [ 0.          0.52573111  0.          0.85065081]]
Matrix A_3 (Updated Matrix) at iteration 3:
[[ 5.56155281e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00]
 [ 0.00000000e+00  3.61803399e+00  0.00000000e+00 -3.86220340e-17]
 [-1.01460876e-16  0.00000000e+00  1.43844719e+00  0.00000000e+00]
 [ 0.00000000e+00 -2.70369227e-16  0.00000000e+00  1.38196601e+00]]

Eigenvalues (Jacobi Method): [5.56155281 3.61803399 1.43844719 1.38196601]
```

2.2) Теоретичні відомості та розрахунки

Для системи нелінійних рівнянь методом Ньютона потрібно провести теоретичне обґрунтування збіжності цього методу.

$$\begin{cases} \tan(xy + 0.1) = x^2 \\ x^2 + 2y^2 = 1 \end{cases}$$

2.2.1) Умови збіжності

Метод Ньютона для системи нелінійних рівнянь збігається за умови виконання наступних критеріїв:

- Початкове наближення (x_0, y_0) повинно бути достатньо близьким до шуканого розв'язку.
- Функції $f_1(x, y)$ і $f_2(x, y)$ мають достатню гладкість (диференційовані функції) в околі розв'язку.
- Матрична Якобіанова матриця $J(x, y)$ повинна бути невинродженою в точці розв'язку, тобто її детермінант $\det(J(x, y)) \neq 0$.

Якобіан системи рівнянь:

Для системи:

$$\begin{cases} f_1(x, y) = \tan(xy + 0.1) - x^2 \\ f_2(x, y) = x^2 + 2y^2 - 1 \end{cases}$$

Якобіан $J(x, y)$ буде матрицею часткових похідних функцій $f_1(x, y)$ та $f_2(x, y)$:

$$J(x, y) = \begin{pmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} \end{pmatrix} = \begin{pmatrix} \frac{y}{\cos^2(xy+0.1)} - 2x & \frac{x}{\cos^2(xy+0.1)} \\ 2x & 4y \end{pmatrix}$$

Необхідність умов на Якобіан:

Для збіжності методу Ньютона необхідно, щоб матриця Якобіана була невинродженою в точці розв'язку, тобто:

$$\det(J(x, y)) = \left(\frac{y}{\cos^2(xy + 0.1)} - 2x \right) \cdot (4y) - \left(\frac{x}{\cos^2(xy + 0.1)} \right) \cdot (2x) \neq 0$$

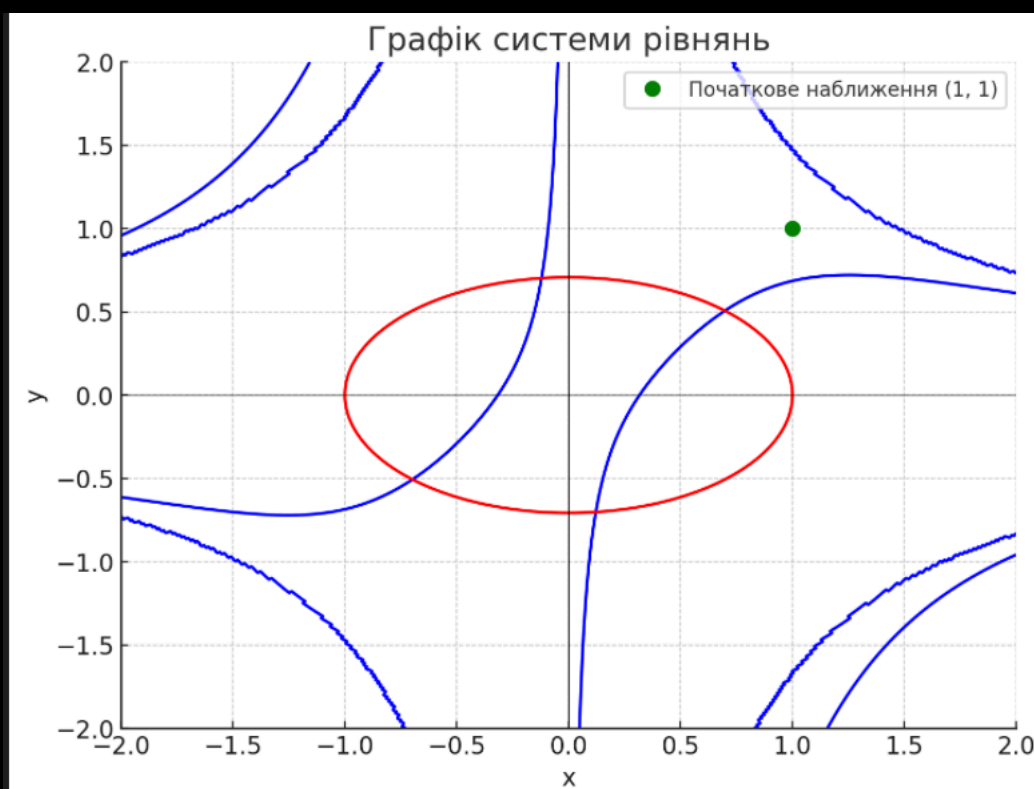
Якщо детермінант Якобіанова матриці в точці розв'язку не дорівнює нулю, метод Ньютона буде збігатися.

2.2.2) Графічна інтерпретація:

Для більш інтуїтивного розуміння, можемо побудувати графік функцій:

$$f_1(x, y) = \tan(xy + 0.1) - x^2$$

$$f_2(x, y) = x^2 + 2y^2 - 1$$



Сині криві відповідають рівнянню $f_1(x,y)=0$, а червоні — $f_2(x,y)=0$. Точки перетину цих кривих показують можливі розв'язки системи рівнянь. Це дає нам уявлення про те, де може знаходитися рішення для методу Ньютона.

З графіків видно, що точки перетину цих функцій можуть бути використані як початкові наближення для методу Ньютона, що забезпечить його збіжність.

2.2.3) Початкове наближення та його роль:

З графічного аналізу системи рівнянь видно, що точка (1,1) знаходиться досить близько до місця перетину кривих, що визначають рівняння системи. Тому, початкові значення $x_0=1$ і $y_0=1$ є хорошим наближенням, яке може призвести до швидкої збіжності методу Ньютона.

2.2.4) Висновок щодо збіжності:

Метод Ньютона буде збігатися до розв'язку, якщо:

1. Початкове наближення (x_0, y_0) знаходиться достатньо близько до шуканого розв'язку.
2. Матриця Якобіана не є виродженою в точці розв'язку, тобто $\det(J(x,y)) \neq 0$. Метод має квадратичну швидкість збіжності, що означає, що з кожною ітерацією точність наближення зростає в квадраті, за умови правильного початкового наближення.

2.2.5) Метод Ньютона для розв'язання системи нелінійних рівнянь

Метод Ньютона є потужним ітераційним методом для розв'язання систем нелінійних рівнянь. Ітераційна формула методу Ньютона для системи рівнянь виглядає так:

$$X_{k+1} = X_k - J(X_k)^{-1}F(X_k)$$

де:

- $X_k = \begin{pmatrix} x_k \\ y_k \end{pmatrix}$ — вектор змінних на k -ій ітерації,
- $F(X_k) = \begin{pmatrix} f_1(x_k, y_k) \\ f_2(x_k, y_k) \end{pmatrix}$ — вектор функцій, що визначає систему нелінійних рівнянь,
- $J(X_k)$ — матриця Якобі (матриця часткових похідних функцій відносно змінних x і y).

Якобіан:

Для системи рівнянь:

$$\begin{cases} f_1(x, y) = \tan(xy + 0.1) - x^2 \\ f_2(x, y) = x^2 + 2y^2 - 1 \end{cases}$$

Якобіан буде виглядати так:

$$J(x, y) = \begin{pmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} \end{pmatrix} = \begin{pmatrix} \frac{y}{\cos^2(xy+0.1)} - 2x & x \\ 2x & 4y \end{pmatrix}$$

Алгоритм:

1. Початкове наближення $X_0 = (x_0, y_0)$ повинно бути достатньо близьким до шуканого розв'язку.
2. Для кожної ітерації обчислюється Якобіан $J(X_k)$ та вектор функцій $F(X_k)$.
3. Визначається корекція $\Delta X_k = J(X_k)^{-1}F(X_k)$, і значення X_{k+1} оновлюється за допомогою формули:

$$X_{k+1} = X_k - \Delta X_k$$

4. Ітерації повторюються, поки різниця між значеннями на двох ітераціях не стане менша за задану точність ϵ .

2.2.6) Результати

Після виконання 5 ітерацій метод Ньютона дав наступні результати:

```
Ітерація 1:  
Якобіан J(x, y):  
[[2.86028051 4.86028051]  
 [2.         4.         ]]  
  
Xk:  
[1, 1]  
  
F(Xk):  
[0.96475966 2.         ]  
  
Ітерація 2:  
Якобіан J(x, y):  
[[-14.23470614 19.85240791]  
 [ 8.81350132 -4.81350132]]  
  
Xk:  
[4.406750658216354, -1.203375329108177]  
  
F(Xk):  
[-17.54728707 21.31567573]  
  
Ітерація 3:  
Якобіан J(x, y):  
[[ -5.05888212 1.2287469 ]  
 [ 2.44990941 -10.40368331]]  
  
Xk:  
[1.2249547046532525, -2.6009208274921702]  
  
F(Xk):  
[-1.44487427 14.03009233]  
  
Ітерація 4:  
Якобіан J(x, y):  
[[-142.58017871 143.14971261]  
 [ 2.53888109 -4.96750145]]  
  
Xk:  
[1.2694405464243308, -1.2418753633722424]  
  
F(Xk):  
[-12.18342256 3.69598814]  
  
Ітерація 5:  
Якобіан J(x, y):  
[[-4.96097821 3.9500294 ]  
 [ 5.25652369 0.78659759]]  
  
Xk:  
[2.628261844563882, 0.19664939731918984]  
  
F(Xk):  
[-6.19860196 5.98510229]
```

Результати ітерацій:

	x	y
0	4.406751	-1.203375
1	1.224955	-2.600921
2	1.269441	-1.241875
3	2.628262	0.196649
4	1.472117	0.313862

Отже, для даної системи рівнянь
розв'язком є
 $(x, y) = (1.472117; 0.313862)$