Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Фізико-технічний інститут

Комп'ютерний практикум №2

Розв'язання систем лінійних алгебраїчних рівнянь прямими методами

предмет «Методи обчислень»

Роботу виконав:
Студент 3 курсу ФТІ, групи ФІ-91
Цибульник Антон Владиславович

Приймала:
Стьопочкіна Ірина Валеріївна

2022

Завдання

Розв'язати систему рівнянь з кількістю значущих цифр $m = 6$. Якщо матриця системи симетрична, то розв'язання проводити за методом квадратних коренів, якщо матриця системи несиметрична, то використати метод Гауса. Вивести всі проміжні результати (матриці A, що отримуються в ході прямого ходу методу Гауса, матрицю зворотного ходу методу Гауса, або матрицю T та вектор y для методу квадратних коренів), та розв'язок системи. Навести результат перевірки: вектор нев'язки $r = |b - Ax|$, де x – отриманий розв'язок.

Варіант завдання

Розглядається така система рівнянь:

$$\begin{cases} 5{,}5x_1 + 7x_2 + 6x_3 + 5{,}5x_4 = 23 \\ 7x_1 + 10{,}5x_2 + 8x_3 + 7x_4 = 32 \\ 6x_1 + 8x_2 + 10{,}5x_3 + 9x_4 = 33 \\ 5{,}5x_1 + 7x_2 + 9x_3 + 10{,}5x_4 = 31 \end{cases} \tag{1}$$

З цієї системи запишемо матрицю коефіцієнтів A та матрицю вільних членів B:

$$A = \begin{pmatrix} 5.5 & 7 & 6 & 5.5 \\ 7 & 10.5 & 8 & 7 \\ 6 & 8 & 10.5 & 9 \\ 5.5 & 7 & 9 & 10.5 \end{pmatrix}, B = \begin{pmatrix} 23 \\ 32 \\ 33 \\ 31 \end{pmatrix}.$$

Матриця A є симетричною, отже, завдання зводиться до знаходження розв'язку системи рівнянь методом квадратного кореня.

Теоретичні відомості

Метод квадратного кореня полягає у тому, щоб спершу представити початкову матрицю A у вигляді добутку двох взаємно транспонованих матриць $A = TT'$, а далі зворотнім ходом віднайти шуканий розв'язок x початкової системи рівнянь. Тож на першому кроці відбувається так звана факторизація, тобто прямий хід, а на другому кроці – зворотній хід, тобто фінальне знаходження коренів.

Крок 1. Прямий хід (факторизація):

$$\begin{pmatrix} t_{11} & 0 & 0 & 0 \\ t_{21} & t_{22} & 0 & 0 \\ t_{31} & t_{32} & t_{33} & 0 \\ t_{41} & t_{42} & t_{43} & t_{44} \end{pmatrix} \begin{pmatrix} t_{11} & t_{12} & t_{13} & t_{14} \\ 0 & t_{22} & t_{23} & t_{24} \\ 0 & 0 & t_{33} & t_{34} \\ 0 & 0 & 0 & t_{44} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix}, \tag{2}$$

де елементи матриці T шукатимемо за співвідношеннями, які можна отримати послідовним перемноженням елементів у рівнянні (2):

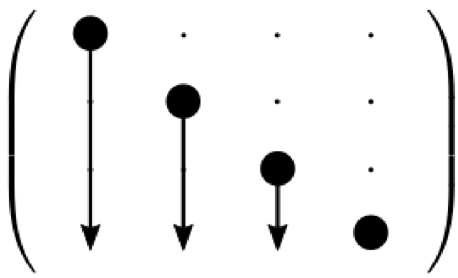
$$t_{11} = \sqrt{a_{11}}\ , t_{i1} = \frac{a_{i1}}{t_{11}} \qquad \text{для елементів першого стовпця,}$$

$$t_{ii} = \sqrt{a_{ii} - \sum_{k<i} t_{ik}^2} \qquad \text{для діагональних елементів,}$$

$$t_{ij} = \frac{a_{ij} - \sum\limits_{k<j} t_{ik} t_{jk}}{t_{jj}} \qquad \text{для елементів під діагоналлю,}$$

причому важливим є порядок застосування формул.

Правило обрахунку елементів матриці T можна описати так: спочатку з'ясовуємо перший діагональний елемент a_{11}, далі обраховуємо всі елементи під ним, тобто елементи відповідного стовпця. Наступним кроком рахуємо другий діагональний елемент, а тоді знову всі елементи відповідного стовпця і так далі. На рисунку праворуч зображена відповідна схема.



Крок 2. Зворотній хід:

$$\text{Із початкової системи рівнянь} \qquad Ax = B, \tag{3}$$

$$\text{тоді враховуючи, що } A = TT' \qquad TT'x = B, \tag{4}$$

$$\begin{array}{l}\text{розіб'ємо отримане рівняння на}\\ \text{такі два послідовних підкроки}\end{array} \qquad \begin{cases} T'x = y \\ Ty = B. \end{cases} \tag{5}$$

Оскільки кожна з матриць T та T' має діагональний вид, то обидва матричні рівняння системи (5) розв'язуються зворотнім ходом (як в методі Гауса). Наведемо формули обрахунку:

$$y_1 = \frac{b_1}{t_{11}},\ y_i = \frac{b_i - \sum\limits_{k=1}^{i-1} t_{ik} y_k}{t_{ii}}$$

$$x_n = \frac{y_n}{t'_{nn}},\ x_i = \frac{y_i - \sum\limits_{k=i+1}^{n} t'_{ik} x_k}{t'_{ii}}.$$

Програмний код

```
import numpy as np

a = np.array([[5.5, 7.0, 6.0, 5.5],
              [7.0, 10.5, 8.0, 7.0],
              [6.0, 8.0, 10.5, 9.0],
              [5.5, 7.0, 9.0, 10.5]])

length = a[0].size

b = np.array([23.0, 32.0, 33.0, 31.0])

y = np.array([0.0, 0.0, 0.0, 0.0])

x = np.array([0.0, 0.0, 0.0, 0.0])

t = np.array([[0.0, 0.0, 0.0, 0.0],
              [0.0, 0.0, 0.0, 0.0],
              [0.0, 0.0, 0.0, 0.0],
              [0.0, 0.0, 0.0, 0.0]])

t_rev = np.array([[0.0, 0.0, 0.0, 0.0],
                  [0.0, 0.0, 0.0, 0.0],
                  [0.0, 0.0, 0.0, 0.0],
                  [0.0, 0.0, 0.0, 0.0]])

def build_t(a,t):
    # first column is unuque
    d = 0 # index of a diagonal element

    # diagonal element
    t[d][d] = np.sqrt(a[d][d])

    # else elements in the first column
    for i in range(d+1, length):
        t[i][d] = a[i][d]/t[d][d]

    # general formula for the next diagonal element
    S_row = 0
    for k in range(0, d+1):
        S_row = S_row + t[d+1][k]*t[d+1][k]

    t[d+1][d+1] = np.sqrt(a[d+1][d+1] - S_row)

    for d in range(1, length-1):
        # elements in each column
        for i in range(d+1, length):
            S_cross = 0
            for k in range(0,d):
                S_cross = S_cross + t[i][k]*t[d][k]

            t[i][d] = (a[i][d] - S_cross)/t[d][d]
```

```python
        # the next diagonal element
        S_row = 0
        for k in range(0, d+1):
            S_row = S_row + t[d+1][k]*t[d+1][k]

        t[d+1][d+1] = np.sqrt(a[d+1][d+1] - S_row)

    return t

t = build_t(a, t)

def build_t_rev(t):
    for i in range(length):
        for j in range(length):
            t_rev[i][j] = t[j][i]
    return t_rev

t_rev = build_t_rev(t)

def find_y(t,b,y):
    y[0]=b[0]/t[0][0]

    for i in range(1, length):
        S_y=0
        for k in range(0, i):
            S_y = S_y + t[i][k]*y[k]
        y[i] = (b[i] - S_y)/t[i][i]

    return y

y = find_y(t,b,y)

def find_x(t_rev,y):
    x[length-1]=y[length-1]/t_rev[length-1][length-1]

    for i in range(length-2, -1, -1):
        S_x=0
        for k in range(length-1, i, -1):
            S_x = S_x + t_rev[i][k]*x[k]
        x[i] = (y[i] - S_x)/t_rev[i][i]

    return x

x = find_x(t_rev,y)

r = abs(b - np.dot(a,x))
```

Результати й проміжні кроки

Наведемо проміжні значення матриць T, T' методу квадратного кореня:

$$T = \begin{pmatrix} 2.34520788 & 0 & 0 & 0 \\ 2.98481003 & 1.26131245 & 0 & 0 \\ 2.5584086 & 0.28829998 & 1.96759461 & 0 \\ 2.34520788 & 0 & 1.52470431 & 1.63562733 \end{pmatrix} \tag{6}$$

$$T' = \begin{pmatrix} 2.34520788 & 2.98481003 & 2.5584086 & 2.34520788 \\ 0 & 1.26131245 & 0.28829998 & 0 \\ 0 & 0 & 1.96759461 & 1.52470431 \\ 0 & 0 & 0 & 1.63562733 \end{pmatrix} \tag{7}$$

Визначити правильність зазначених матриць можна, власне кажучи, зробивши таку перевірку: $A = TT'$. Також варто зазначити проміжний вектор y, отриманий після розв'язку зворотнім ходом рівняння $Ty = B$:

$$y = (9.80723295, 2.16224991, 3.70285334, 1.43935205).$$

І наостанок був обрахований вектор нев'язки $r = |B - Ax|$, який вказує похибку обчислень (коротше кажучи, це перевірка правильності розв'язку):

$$r = (0, 0, 0, 0).$$

Судячи з вектора нев'язки r, отриманий розв'язок $x = (0.16, 1.44, 1.2, 0.88)$ і є розв'язком початкової системи рівнянь $Ax = B$, зображеної у виразі (1). Отже, остаточний розв'язок:

$$x_1 = 0.16,\ x_2 = 1.44,\ x_3 = 1.2,\ x_4 = 0.88.$$

Контрольні запитання

1. *В чому полягає відмінність схеми Гауса із вибором головного елементу та схеми єдиного ділення?*

   Фактично, відмінність між цими двома методами полягає у виборі головного елементу. Наприклад, у схемі єдиного ділення головним елементом завжди обирають перший коефіцієнт, натомість у методі вибору елементу можемо назначити головний елемент власноруч (наприклад, обрати деякий максимальний). Це запобігає накопиченню похибок округлення. Крім того, це дає можливість уникати ситуацій, коли першим коефіцієнтом є нуль.

2. *В чому переваги схеми Гауса із вибором головного елементу порівняно із схемою єдиного ділення, схемою повного виключення?*

   Переваги схеми повного виключення порівняно зі схемою єдиного ділення аналогічні до попереднього питання.

3. *Коли факторизація за методом квадратного кореня неможлива для симетричної матриці?*

   Відповідь на питання можна визначити із співвідношень, наведених на початку першого кроку теоретичних відомостей. Власне, ситуація факторизації можлива для комплексних значень, проте неможлива у випадку утворення нульових величин на місці діагональних елементів матриць T та T'.