Київський національний університет імені Тараса Шевченка

ЗВІТ

до лабораторної роботи №2
з дисципліни “Чисельні методи”
на тему:
“Прямі і ітераційні методи розв’язання систем лінійних алгебраїчних рівнянь”

Виконав
Студент 3 курсу
Групи ТТП-31
Факультету комп’ютерних наук та кібернетики
Сіхневич Святослав

Київ, 2024

Зміст

Постановка задачі

Згенерувати матрицю 4x4 з цілими елементами за модулем менше 10 та вектор правої частини з урахуванням обмежень та достатніх умов збіжності, що накладаються методами у варіанті. Порахувати визначник матриці та обернену матрицю тими методами, що мають відповідне застосування. Для ітераційного методу бажану точність розв'язку СЛАР зробити параметром, який може вводити користувач.

Мій варіант - 1.

Необхідно використати метод Гаусса, метод прогонки, метод Якобі.

Для обчислень можна обрати наступну матрицю:

$$A = \begin{bmatrix} -9 & 2 & -6 & 1 \\ 4 & -9 & 1 & 0 \\ 1 & -1 & 6 & -1 \\ -5 & 2 & -1 & -8 \end{bmatrix}$$

Вектор правої частини:

$$b = \begin{bmatrix} 2 \\ -1 \\ 6 \\ 4 \end{bmatrix}$$

Теоретичні відомості та розрахунки

Теорія

В системі лінійних алгебраїчних рівнянь (СЛАР) Ax=b, де A - матриця розмірності n x n, det A ≠ 0, існує розв'язок і він єдиний.

Метод Гаусса

Для зменшення обчислювальної похибки в методі Гаусса використовують вибір головного елементу.

Метод Гауса

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = a_{1(n+1)} \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = a_{2(n+1)} \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n = a_{n(n+1)} \end{cases}$$

$$a_{i(n+1)} = b_i, \quad i = \overline{1,n}$$

$$a_{kj}^{(k)} = a_{kj}^{(k-1)} / a_{kk}^{(k-1)}, \; k = \overline{1,n}$$

$$a_{ij}^{(k)} = a_{ij}^{(k-1)} - a_{ik}^{(k-1)} a_{kj}^{(k)}, \; j = \overline{k+1, n+1}$$

$$a_{kk}^{(k-1)} \neq 0 \qquad i = \overline{k+1, n}$$

$$\begin{cases} x_1 + a_{12}^{(1)}x_2 + \ldots + a_{1n}^{(1)}x_n = a_{1(n+1)}^{(1)} \\ a_{22}^{(1)}x_2 + \ldots + a_{2n}^{(1)}x_n = a_{2(n+1)}^{(1)} \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{n2}^{(1)}x_2 + \ldots + a_{nn}^{(1)}x_n = a_{n(n+1)}^{(1)} \end{cases}$$

Прямий хід

$$a_{kj}^{(k)} = a_{kj}^{(k-1)} / a_{kk}^{(k-1)},\ k = \overline{1,n}$$

$$a_{ij}^{(k)} = a_{ij}^{(k-1)} - a_{ik}^{(k-1)} a_{kj}^{(k)},$$

$$j = \overline{k+1, n+1}, \quad i = \overline{k+1, n}$$

$$a_{kk}^{(k-1)} \neq 0$$

Зворотній хід

$$x_n = a_{n(n+1)}^{(n)}$$

$$x_i = a_{i(n+1)}^{(i)} - \sum_{j=i+1}^{n} a_{ij}^{(i)},\ i = \overline{n-1, 1}$$

$$x_n = a_{n(n+1)}^{n}$$

$$x_i = a_{i(n+1)}^{(i)} - \sum_{j=i+1}^{n} a_{ij}^{(i)},\ i = \overline{n-1, 1}$$

Метод прогонки

Є частковим випадком методу Гаусса.

Метод використовується лише для тридіагональних матриць.

Критерій перевірки $a_{ij} = 0$ для всіх $|i-j| > 1$

$$\begin{cases} -c_0 y_0 + b_0 y_1 = -f_0; \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_i y_{i-1} - c_i y_i + b_i y_{i+1} = -f_i, \quad i = \overline{1, n-1}; \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_n y_{n-1} - c_n y_n = -f_n; \end{cases}$$

Достатня умова стійкості. Нехай коефіцієнти $a_0, b_0 = 0$; $c_0, c_n \neq 0$; $a_i, b_i, c_i \neq 0$; $i = \overline{1, n-1}$. Якщо виконуються умови:

1) $|c_i| \geqslant |a_i| + |b_i|, \quad i = \overline{0, n}$;

2) $\exists\, i : |c_i| > |a_i| + |b_i|$,

то метод є стійким: $|\alpha_i| \leqslant 1$; $|z_i| > 1$, $i = \overline{1, n}$.

Прямий хід метода Гаусса в методі прогонки відповідає знаходженню прогонкових коефіцієнтів:

$$\alpha_1 = \frac{b_0}{c_0}; \qquad \beta_1 = \frac{f_0}{c_0}; \qquad \alpha_{i+1} = \frac{b_i}{z_i}; \qquad \beta_{i+1} = \frac{f_i + a_i\beta_i}{z_i};$$

$$z_i = c_i - \alpha_i a_i; \qquad i = \overline{1, n-1}.$$

Зворотній ход:

$$y_n = \frac{f_n + a_n\beta_n}{z_n}; \qquad y_i = \alpha_{i+1}y_{i+1} + \beta_{i+1}, \qquad i = \overline{n-1, 0}.$$

Складність методу прогонки: $Q(n) = 8n - 2$.

Зауваження. Методом прогонки можна знайти визначник:

$$DetA = -c_0 \cdot (-z_1) \cdot \ldots \cdot (-z_n).$$

Метод Якобі

$$x_i^{k+1} = -\sum_{j=1}^{i-1} \frac{a_{ij}}{a_{ii}} x_j^k - \sum_{j=i+1}^{n} \frac{a_{ij}}{a_{ii}} x_j^k + \frac{b_i}{a_{ii}}.$$

Достатня умова збіжності. Якщо $\forall i : i = \overline{1, n}$ виконується нерівність:

$$|a_{ii}| \geqslant \sum_{j=1, j\neq i}^{n} |a_{ij}|,$$

Умова припинення методу: $||x^n - x^{n-1}|| \leqslant \varepsilon$.

Зауваження. В якості норми зазвичай обирають неперервну (кубічну) норму вектору: $||x||_\infty = \max\limits_{1 \leqslant i \leqslant n} |x_i|$.

Необхідні і достатні умови збіжності. Для $\forall x^0$ ітераційний процес методу Якобі (21) збігається тоді і тільки тоді, коли $|\lambda| < 1$, де λ – це корені нелінійного рівняння:

$$\begin{vmatrix} \lambda a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & \lambda a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & \lambda a_{nn} \end{vmatrix} = 0.$$

Зауваження. При розв'язанні системи лінійних алгебраїчних рівнянь перевіряють достатні умови збіжності. Якщо в задачі необхідно щось довести, знайти область збіжності, то використовують необхідні і достатні умови збіжності.

Розрахунки

Визначник матриці (det A) = −3997. Не дорівнює нулю, отже є один розв'язок. Також матриця невироджена.

$\bar{A}_0 = \left[\begin{array}{cccc|c} -9 & 2 & -6 & 1 & 2 \\ 4 & -9 & 1 & 0 & -1 \\ 1 & -1 & 6 & -1 & 6 \\ -5 & 2 & -1 & -8 & 4 \end{array}\right]$ це розширена матриця.

Обернена матриця:

$$\mathbf{A}^{-1} = \begin{pmatrix} -\frac{431}{4189} & -\frac{169}{4189} & \frac{441}{4189} & -\frac{109}{4189} \\ -\frac{183}{4189} & -\frac{548}{4189} & \frac{265}{4189} & -\frac{56}{4189} \\ \frac{77}{4189} & -\frac{67}{4189} & \frac{621}{4189} & -\frac{68}{4189} \\ \frac{214}{4189} & -\frac{23}{4189} & -\frac{287}{4189} & -\frac{461}{4189} \end{pmatrix}$$

Розв’язання методом Гаусса:
Формуємо розширену матрицю. Прямий хід приводить матрицю до верхньотрикутникутного вигляду.
Зворотний хід поступово обчислює значення невідомих x.

Результат програмованих обчислень (файл gauss_method.py):

```
Початкова розширена матриця:

Крок Початок:

+---------+---------+---------+---------+---------+
|    x1   |    x2   |    x3   |    x4   |    b    |
+---------+---------+---------+---------+---------+
| -9.0000 | -2.0000 | -6.0000 |  1.0000 |  2.0000 |
|  4.0000 | -9.0000 |  1.0000 |  0.0000 | -1.0000 |
|  1.0000 | -1.0000 |  6.0000 | -1.0000 |  6.0000 |
| -5.0000 |  2.0000 | -1.0000 | -8.0000 |  4.0000 |
+---------+---------+---------+---------+---------+

Крок Прямий хід (крок 1):

+---------+---------+---------+---------+---------+
|    x1   |    x2   |    x3   |    x4   |    b    |
+---------+---------+---------+---------+---------+
| -9.0000 | -2.0000 | -6.0000 |  1.0000 |  2.0000 |
|  0.0000 | -9.8889 | -1.6667 |  0.4444 | -0.1111 |
|  0.0000 | -1.2222 |  5.3333 | -0.8889 |  6.2222 |
|  0.0000 |  3.1111 |  2.3333 | -8.5556 |  2.8889 |
+---------+---------+---------+---------+---------+

Крок Прямий хід (крок 2):

+---------+---------+---------+---------+---------+
|    x1   |    x2   |    x3   |    x4   |    b    |
+---------+---------+---------+---------+---------+
| -9.0000 | -2.0000 | -6.0000 |  1.0000 |  2.0000 |
|  0.0000 | -9.8889 | -1.6667 |  0.4444 | -0.1111 |
|  0.0000 |  0.0000 |  5.5393 | -0.9438 |  6.2360 |
|  0.0000 |  0.0000 |  1.8090 | -8.4157 |  2.8539 |
+---------+---------+---------+---------+---------+

Крок Прямий хід (крок 3):

+---------+---------+---------+---------+---------+
|    x1   |    x2   |    x3   |    x4   |    b    |
+---------+---------+---------+---------+---------+
| -9.0000 | -2.0000 | -6.0000 |  1.0000 |  2.0000 |
|  0.0000 | -9.8889 | -1.6667 |  0.4444 | -0.1111 |
|  0.0000 |  0.0000 |  5.5393 | -0.9438 |  6.2360 |
|  0.0000 |  0.0000 |  0.0000 | -8.1075 |  0.8174 |
+---------+---------+---------+---------+---------+
```

```
Крок Прямий хід (крок 4):

+---------+---------+---------+---------+---------+
|    x1   |    x2   |    x3   |    x4   |    b    |
+---------+---------+---------+---------+---------+
| -9.0000 | -2.0000 | -6.0000 |  1.0000 |  2.0000 |
|  0.0000 | -9.8889 | -1.6667 |  0.4444 | -0.1111 |
|  0.0000 |  0.0000 |  5.5393 | -0.9438 |  6.2360 |
|  0.0000 |  0.0000 |  0.0000 | -8.1075 |  0.8174 |
+---------+---------+---------+---------+---------+

Розрахунок x[4]: -0.1008

Розрахунок x[3]: 1.1086

Розрахунок x[2]: -0.1801

Розрахунок x[1]: -0.9324
```

Розв’язання методом прогонки:
Обчислення відбувалися з застосуванням програмованих засобів (файл thomas_algorithm.py). Також врахована перевірка чи є матриця тридіагональною.

```
Матриця не тридіагональна. Використовуємо метод Гаусса.
Розв'язок СЛАР: [-0.93244934 -0.1801351   1.10858144 -0.10082562]
```

Оскільки матриця не є тридіагональною, ми використовуємо метод Гаусса. Розв’язок збігається.

Розв’язання методом Якобі:
Початкове наближення вектора x обрано нульовим вектором.

Для кожного елемента ? обчислюємо нове значення за формулою:

$$x_i^{(k+1)} = \frac{b_i - \sum_{j=0, j\neq i}^{n-1} a_{ij} x_j^{(k)}}{a_{ii}}$$

де a_ii - діагональний елемент матриці A, а b_i - відповідний елемент вектора правої частини.

Ітерації тривають, доки норма різниці між новим і попереднім Наближеннями $||x^{(k+1)} - x^{(k)}||_\infty$ не стане меншою за задану точність або доки не буде досягнуто максимальну кількість ітерацій.

Точність буде дорівнювати 10^{-5}

Ітерація	x_1	x_2	x_3	x_4	Максимальна зміна
1	-0.222222	0.111111	1.000000	-0.500000	1.000000e+00
2	-0.969136	0.123457	0.972222	-0.458333	7.469136e-01
3	-0.948731	-0.211591	1.105710	0.015046	4.733796e-01
4	-0.910670	-0.187691	1.125364	-0.098155	1.132009e-01
5	-0.941662	-0.168591	1.104137	-0.118424	3.099214e-02
6	-0.934008	-0.184723	1.109108	-0.091626	2.679841e-02
7	-0.930758	-0.180769	1.109610	-0.101065	9.438648e-03
7	-0.930758	-0.180769	1.109610	-0.101065	9.438648e-03
8	-0.933020	-0.179269	1.108154	-0.102169	2.262027e-03
9	-0.932506	-0.180436	1.108597	-0.100199	1.970664e-03
10	-0.932323	-0.180159	1.108645	-0.100867	6.685804e-04
11	-0.932491	-0.180072	1.108550	-0.100918	1.681586e-04
12	-0.932452	-0.180157	1.108583	-0.100780	1.387375e-04
13	-0.932441	-0.180136	1.108586	-0.100830	4.985292e-05
14	-0.932453	-0.180131	1.108579	-0.100832	1.182926e-05
15	-0.932449	-0.180137	1.108582	-0.100822	9.609077e-06

```
Розв'язок методом Якобі:
Вектор розв'язку: [-0.93244948 -0.18013676  1.10858165 -0.10082227]
Кількість ітерацій: 15
```

За програмованими обрахунками (файл jacobi_method.py), за 15 ітерацій методом Якобі, ми отримали вектор розв’язку

$$\begin{bmatrix} -0,93244948 \\ -0,18013676 \\ 1,10858165 \\ -0,10082227 \end{bmatrix}$$

Зробимо перевірку, помноживши обрану матрицю A на отриманий вектор x:

$$\begin{pmatrix} -9 & 2 & -6 & 1 \\ 4 & -9 & 1 & 0 \\ 1 & -1 & 6 & -1 \\ -5 & 2 & -1 & -8 \end{pmatrix} \cdot \begin{pmatrix} -0.93244 \\ -0.18013 \\ 1.108581 \\ -0.10082 \end{pmatrix} = \begin{pmatrix} 1.279394 \\ -1.000009 \\ 5.999996 \\ 3.999919 \end{pmatrix}$$

Бачимо, що отриманий вектор співпадає з вектором b, що ми спершу обрали. Отже ми правильно розв’язали дану СЛАР.

Висновок:

Отже, ми запрограмували методи обчислень та отримали результат, що пройшов перевірку.

Методи Гаусса, прогонки (алгоритм Томпсона) та Якобі викладені в звіті. Репозиторій на відкритий код надано в Додатку звіту.

Додатки

Посилання на репозиторій з програмним кодом для обрахунку в даній лабораторній роботі: https://github.com/Svyatoslavk27/NM_2lab