Politechnika Świętokrzyska

Laboratorium

Metody numeryczne

Ćwiczenie 2

Dokładne metody rozwiązywania układów równań liniowych

Cel ćwiczenia

Celem ćwiczenia jest zapoznanie studentów z dokładnymi metodami rozwiązywania układów równań liniowych. Implementacjąalgorytmów rozwiazywania tych równań w języku programowania Python. Wykorzystaniem bibliotek do obliczeń numerycznych dostępnych w języku Python.

dr inż Robert Kazała

Dokładne metody rozwiązywania układów równań liniowych

Dany jest zgodny i oznaczony układ równań liniowych:

$$\mathbf{Ax} = \mathbf{b} \tag{1}$$

gdzie, $A = [a_{i,j}]$, $i=1,...,m$, $j=1,...,n$
jest macierzą m × n znanych współczynników (m - liczba wierszy, n - liczba kolumn), $\mathbf{b} = [b_i]$ $i=1,...,m$ jest wektorem m znanych wyrazów wolnych a $\mathbf{x} = [x_i]$, $j=1,...,n$ jest wektorem n niewiadomych.

Metoda Cramera

Jeżeli macierz współczynników **A** jest macierzą nieosobliwą, to znaczy det **A** ≠ 0, to istnieje macierz odwrotna $\mathbf{A}^{-1}$.

W celu rozwiązania układu (1), obie strony równania należy pomnożyć przez $\mathbf{A}^{-1}$:

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b} \qquad (2)$$

gdzie: $\mathbf{A}^{-1} = \mathbf{A}^{D}/\det\mathbf{A}$, $\mathbf{A}^{D}$ jest macierzą dopełnień algebraicznych (macierzą dołączoną)

Z równania (2), wynikają wzory Cramera:

$$x_k = \mathbf{D}_k/\det\mathbf{A}$$

gdzie: $k = 1, 2,\ldots, n$.

Algorytm 1 Metoda Kramera

```
1: dA ← det A                                   ▷ wyznacznik macierzy A
2: if dA ≠ 0 then
3:     for j ← 1...n do
4:         D ← A
5:         [d_{i,j}]_{i=1,...,m} ← b
6:         x_i ← det D / dA
7:     end for
8: end if
```

Macierze trójkątne

Macierz trójkątna dolna (lewa)

$$\mathbf{L} = \begin{bmatrix} l_{11} & 0 & \cdots & 0 \\ l_{21} & l_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{k1} & l_{k2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{bmatrix}$$

Macierz trójkątna górna (prawa)

$$\mathbf{U} = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_{kn} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_{nn} \end{bmatrix}$$

Wyznaczniki macierzy **L** oraz **U** są iloczynami elementów leżących na głównej przekątnej.

Metoda eliminacji Gaussa pozwala sprowadzić układ do postaci trójkątnej.

Jeżeli macierz układu **Ax** = **b** jest macierzą trójkątną, to taki układ można łatwo rozwiązać.

Rozwiązanie układu równań liniowych opisanego macierzą trójkątną

Dla układu z macierzą trójkątną górną, rozwiązanie można opisać ogólnymi zależnościami:

$$x_n = \frac{b_n}{a_{nn}}$$

$$x_i = \frac{b_i - \sum_{k=i+1}^{n} a_{ik} x_k}{a_{ii}}$$

dla $i = n - 1, n - 2, \ldots 1$.

Dla układu z macierzą trójkątną dolną:

$$x_n = \frac{b_1}{a_{11}}$$

$$x_i = \frac{b_i - \sum_{k=1}^{n-1} a_{ik} x_k}{a_{ii}}$$

dla $i = 2, 3, \ldots n$.

Powyższa procedura nazywa się metodą wstecznego podstawiania.

Wiele metod numerycznego rozwiązywania układów równań liniowych polega na:
• sprowadzeniu układu do postaci trójkątnej
• a następnie wykorzystaniu metody wstecznego podstawiania.

Metoda eliminacji Gaussa

Stosując metodę eliminacji Gaussa można sprowadzić układ równań do postaci z macierzą trójkątną górną.

Uogólnione zależności umożliwiające wyznaczenie współczynników macierzy trójkątnej górnej i wyrazów wolnych układu równań liniowych dowolnego rzędu (n-tego) są następujące:

$$a_{ij}^{k} = a_{ij}^{k-1} - \frac{a_{ik}^{k-1}}{a_{kk}^{k-1}} a_{kj}^{k-1} \qquad b_{i}^{k} = b_{i}^{k-1} - \frac{a_{ik}^{k-1}}{a_{kk}^{k-1}} b_{k}^{k-1}$$

Dalej, układ równań można rozwiązać stosując metodę wstecznego podstawiania.

Algorytm 2 Metoda eliminacji Gaussa

1: **for** $k \leftarrow 1 \ldots n-1$ **do** ▷ **faza eliminacji**
2: **for** $i \leftarrow k+1 \ldots n$ **do**
3: $\lambda \leftarrow \frac{a_{i,k}}{a_{k,k}}$
4: **for** $j \leftarrow k+1 \ldots n$ **do**
5: $a_{i,j} \leftarrow a_{i,j} - \lambda a_{k,j}$
6: **end for**
7: $b_i \leftarrow b_i - \lambda b_k$
8: **end for**
9: **end for**
10: x ← b
11: **for** $i \leftarrow n \ldots 1$ **do** ▷ **faza wstecznego podstawiania**
12: $S \leftarrow 0$
13: **for** $j \leftarrow i+1 \ldots n$ **do**
14: $S \leftarrow S + a_{i,j} x_j$
15: **end for**
16: $x_i \leftarrow \frac{b_i - S}{a_{i,i}}$
17: **end for**

Algorytm 3 Metoda eliminacji Gaussa - zapis wektorowy

Na podstawie materiałów do wykładów z przedmiotu Technika obliczeniowa i symulacja, prof. dr hab. inż. M.Wciślik

1: $r \leftarrow \operatorname{rank} \mathbf{A}$ ▷ rząd macierzy $\mathbf{A}$

2: $\mathbf{Ab} \leftarrow [\mathbf{A}\ \mathbf{b}]$ ▷ konkatenacja macierzy $\mathbf{A}$ i wektora $\mathbf{b}$

3: $[ab_{i,j}]_{\substack{i=1 \\ j=1,\ldots,r+1}} \leftarrow [ab_{i,j}]_{\substack{i=1 \\ j=1,\ldots,r+1}} \Big/ [ab_{i,j}]_{\substack{i=1 \\ j=1}}$ ▷ faza eliminacji

4: **for** $k \leftarrow 2 \ldots r$ **do**

5: $\quad [ab_{i,j}]_{\substack{i=k,\ldots,r \\ j=1,\ldots,r+1}} \leftarrow [ab_{i,j}]_{\substack{i=k,\ldots,r \\ j=1,\ldots,r+1}} - [ab_{i,j}]_{\substack{i=k,\ldots,r \\ j=k-1}} [ab_{i,j}]_{\substack{i=k-1 \\ j=1,\ldots,r+1}}$

6: $\quad [ab_{i,j}]_{\substack{i=k \\ j=1,\ldots,r+1}} \leftarrow [ab_{i,j}]_{\substack{i=k \\ j=1,\ldots,r+1}} \Big/ [ab_{i,j}]_{\substack{i=k \\ j=k}}$

7: **end for**

8: **for** $k \leftarrow (r-1) \ldots 1$ **do** ▷ faza wstecznego podstawiania

9: $\quad [ab_{i,j}]_{\substack{i=1,\ldots,k \\ j=1,\ldots,r+1}} \leftarrow [ab_{i,j}]_{\substack{i=1,\ldots,k \\ j=1,\ldots,r+1}} - [ab_{i,j}]_{\substack{i=1+k \\ j=1,\ldots,r+1}} [ab_{i,j}]_{\substack{i=1,\ldots,k \\ j=k+1}}$

10: $\quad [ab_{i,j}]_{\substack{i=1,\ldots,k \\ j=k+1}} \leftarrow [0_{i,j}]_{\substack{i=1,\ldots,k \\ j=1}}$ ▷ wyrażenie opcjonalne

11: **end for**

Literatura

The Python Tutorial, https://docs.python.org/3/tutorial/index.html

Python Tutorial, https://www.w3schools.com/python/

Numpy tutorial, https://numpy.org/devdocs/user/quickstart.html

Scipy tutorial, https://docs.scipy.org/doc/scipy/reference/

Matplotlib, https://matplotlib.org/index.html

The Python Standard Library, https://docs.python.org/3/library/index.html

The Python Language Reference, https://docs.python.org/3/reference/index.html

Zadania

1. Stosując metodę Cramera rozwiązać podany układ równań liniowych:

   $10x_1 + 40x_2 + 70x_3 = 300$

   $20x_1 + 50x_2 + 80x_3 = 360$

   $30x_1 + 60x_2 + 80x_3 = 390$

2. Rozwiązać układ równań z zadania 1 stosując metodę eliminacji Gaussa.
3. Dany jest układ równań liniowych:

   $2x_2 + 2x_3 = 1$

   $3x_1 + 3x_2 = 3$

   $x_1 + x_3 = 2$

   Sprawdzić czy macierz A podanego układu jest osobliwa, jeżeli nie to rozwiązać układ metodą eliminacji Gaussa.
4. Stosując język Python, zaimplementować wektorową postać metody eliminacji Gaussa.