Politechnika Wrocławska

Anna Modrzejewska

236642

Obliczenia naukowe

Lista nr 5

6 stycznia 2019

1 Opis problemu

Należy rozwiązać układ równań liniowych $Ax = b$ dla danej macierzy współczynników $A \in \mathbb{R}^{n \times n}$ i wektora prawych stron $b \in \mathbb{R}^n$, przy czym macierz A jest macierzą rzadką o następującej strukturze:

$$A = \begin{bmatrix} A_1 & C_1 & 0 & 0 & 0 & \dots & 0 \\ B_2 & A_2 & C_2 & 0 & 0 & \dots & 0 \\ 0 & B_3 & A_3 & C_3 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & B_{v-2} & A_{v-2} & C_{v-2} & 0 \\ 0 & \dots & 0 & 0 & B_{v-1} & A_{v-1} & C_{v-1} \\ 0 & \dots & 0 & 0 & 0 & B_v & A_v \end{bmatrix}$$

$v = \frac{n}{l}$, gdzie l jest wielkością wszystkich kwadratowych macierzy wewnętrznych ($A_k, B_k, C_k \in \mathbb{R}^{l \times l}$).

- $A_k (k = 1, \dots, v)$ jest macierzą gęstą,
- 0 jest macierzą zerową stopnia l,
- $B_k (k = 2, \dots, v)$ jest postaci:

$$B_k = \begin{bmatrix} 0 & \dots & 0 & b_1^k \\ 0 & \dots & 0 & b_2^k \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & b_l^k \end{bmatrix},$$

- $C_k (k = 1, \dots, v-1)$ jest macierzą diagonalną postaci:

$$C_k = \begin{bmatrix} c_1^k & 0 & 0 & \dots & 0 \\ 0 & c_2^k & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & c_{l-1}^k & 0 \\ 0 & \dots & 0 & 0 & c_l^k \end{bmatrix}$$

Przy założeniu, że l jest stałą, należy zaimplementować rozwiązywanie układu $Ax = b$ w czasie liniowym.

2 Metoda eliminacji Gaussa

Rozwiązywanie układu równań liniowych metodą eliminacji Gaussa opiera się na sprowadzeniu układu do macierzy schodkowej poprzez wyzerowanie elementów poniżej głównej przekątnej, a następnie zastosowanie podstawiania wstecz. Zostaną przedstawione dwa warianty eliminacji Gaussa - bez wyboru elementu głównego oraz z wyborem.

2.1 Bez wyboru elementu głównego

Pierwszym krokiem algorytmu jest przekształcenie macierzy do równoważnej macierzy trójkątnej górnej poprzez elementarne operacje, takie jak zamiana miejscami dwóch wierszy, dodawanie do siebie wierszy, mnożenie przez niezerową liczbę. Nie trzeba rozpatrywać całej macierzy poniżej diagonali, wystarczy jedynie elementy niezerowe. Należy zauważyć, że w każdym rzędzie macierzy A z wyjątkiem ostatniego, ostatnim niezerowym elementem jest ten należący do diagonali macierzy C. Jego indeks to $i+l$, gdzie i oznacza numer kolumny, a l to stała wielkość bloków. W ostatnim rzędzie ostatni element niezerowy należy do macierzy A_v, zatem jego indeks to n (wielkość macierzy A). Ostatecznie dla każdej i-tej kolumny wystarczy rozpatrywać $\min((i+l), n)$ elementów. Analogicznie dla rzędów, w pierwszych $l-1$ kolumach niezerowe elementy to jedynie te z bloku A_k, czyli l rzędów. Dla kolejnych l kolumn ostatnimi (rzędem) niezerowymi elementami jest ostatnia kolumna bloku B_k i $l-1$ kolumn bloku A_k itd. Zatem dla każdego i-tego rzędu indeks ostatniego niezerowego elementu to $\min((l + l \cdot \lfloor \frac{1+i}{l} \rfloor), n)$.

Po uzyskaniu macierzy schodkowej należy wyliczyć rozwiązanie od dołu ze wzoru:

$$x_n = \frac{b_n}{a_{nn}}, x_k = \frac{b_k - \sum_{j=k+1}^{n} a_{kj} x_j}{x_{kk}}.$$

Ta metoda nie jest niezawodna, tj. w przypadku, gdy któryś z elementów diagonalnych x_{kk} będzie mniejszy od epsilonu maszynowego, nastąpi dzielenie przez 0. Należy wtedy zastosować podejście przedstawione poniżej (punkt 2.2).

2.2 Z wyborem elementu głównego

W tym przypadku zerowanie macierzy poniżej diagonali przebiega tak samo. Jednak później, aby nie dopuścić do sytuacji, że któryś z elementów diagonalnych będzie za bardzo blisku zeru, należy w każdej kolumnie wyszukać maksymalny element i zamienić wierszami z tym leżącym na przekątnej. Dla każdego elementu diagonalnego należy szukać największego elementu w rzędzie o wyższym indeksie (np. dla elementu x_{kk} szukać wśród x_{mk}, gdzie $n > m > k$), aby macierz pozostała zerową trójkątną.

2.3 Złożoność obliczeniowa i pamięciowa

Jako że macierz A jest macierzą rzadką, do jej przechowywania została użyta struktura `SparseMatrixCSC` (*Compressed Sparse Column - Format Skompresowany Kolumnowy*).
Standardowy algorytm eliminacji Gaussa ma złożoność obliczeniową $O(n^3)$, jednak przy uwzględnianiu w implementacji przechodzenia po jedynie niezerowych elementów macierzy złożoność spada do $O(cl^2n)$ $(c - const)$, a więc dla stałej l jest liniowa.

3 Rozkład LU

Metoda LU polega na takim rozkładzie macierzy, że $A = LU$, gdzie L to dolna macierz trójkątna (ang. *lower*, U to górna macierz trójkątna (ang. *upper*). Dodatkowo na przekątna macierzy L wypełniona jest jedynkami. Macierze L i U otrzymuje się z metody eliminacji Gaussa - końcowa trójkątna macierz to macierz U, natomiast mnożniki użyte do wyzerowania macierzy A to macierz L. W skutek takiego rozkładu układ równań przyjmuje postać $L \cdot U \cdot x = b$, a rozwiązanie sprowadza się do dwóch równań:

$$L \cdot z = b$$
$$U \cdot x = z$$

$$\text{U} = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \dots & a_{1,n} \\ 0 & a_{2,2} & a_{2,3} & \dots & a_{2,n} \\ 0 & 0 & a_{3,3} & \dots & a_{2,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & a_{n,n} \end{bmatrix}$$

$$\text{L} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ z_{2,1} & 1 & \dots & 0 \\ z_{3,1} & z_{3,2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ z_{n-1,1} & z_{n-1,2} & \dots & 1 \end{bmatrix}$$

Żeby rozwiązać układ równań należy zastosować algorytm podstawiania wprzód i wstecz. Po uwzględnieniu w implementacji struktury macierzy złożoność obliczeniowa metody sięga $O(n)$.

4 Porównanie wyników zaimplementowanych funkcji

Dla przykładowych macierzy testowych A umieszczonych na stronie kursu obliczony został wektor prawych stron b. Dla poszczególnych metod zanotowano następujące błędy względne rozwiązań:

Metoda eliminacji Gaussa:

rozmiar macierzy	16	10 000	50 000
bez wyboru el. gł.	$3.7646425162 \cdot 10^{-15}$	$6.60110327624 \cdot 10^{-14}$	$2.98591584981 \cdot 10^{-14}$
z wyborem el. gł	$5.01907996276 \cdot 10^{-16}$	$3.83527021916 \cdot 10^{-16}$	$4.145686351718 \cdot 10^{-16}$

Rozkład LU:

rozmiar macierzy	16	10 000	50 000
bez wyboru el. gł.	$4.2141885581 \cdot 10^{-15}$	$7.33495787222 \cdot 10^{-14}$	$2.9447331392 \cdot 10^{-14}$
z wyborem el. gł.	$5.20000542492 \cdot 10^{-16}$	$3.777908043776 \cdot 10^{-16}$	$4.116796199387 \cdot 10^{-16}$

Po analizie powyższej tabeli można stwierdzić, że wszystkie metody zwróciły niewielkie błędy względne (rzędu 10^{-15}), zatem zostały zaimplementowane poprawnie. Ponadto w każdym przypadku błędy względne były mniejsze, gdy był wybierany element główny.