Metody numeryczne - NUM3

Dawid Bytys gr. 3

8 listopada 2022

1 Wstęp

Rozwiązywanie układów równań liniowych dla macierzy kwadratowych jest obarczone złożonością obliczeniową rzędu $O(n^3)$. Biorąc pod uwagę specyficzne rodzaje macierzy, możemy znacznie zredukować liczbę niepotrzebnych operacji i uzyskać nawet złożoność rzędu $O(n)$. Przykładem takiej macierzy może być np. macierz wstęgowa *(ang. band matrix)*, która przyjmuje następującą postać:

$$\begin{bmatrix} a_1 & b_1 & 0 & \cdots & \cdots & 0 \\ c_1 & a_2 & b_2 & \ddots & \ddots & \vdots \\ 0 & c_2 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & b_{n-1} \\ 0 & \cdots & \cdots & 0 & c_{n-1} & a_n \end{bmatrix},$$

gdzie **n** to wymiar macierzy, a wszystkie liczby poza wstęgami to zera. Daje to bardzo dużo możliwości, jeżeli chodzi o samo przechowywanie macierzy w kodzie, ponieważ możemy wyeliminować wszystkie zera i trzymać w kontenerze same wstęgi. Jest to bardzo duża optymalizacja pamięci, ponieważ na przechowanie jednej liczby dla typu `float64` `(double)` potrzebujemy zarezerwować aż 8MB, co w skali całej macierzy daje ogromną ilość konsumowanych niepotrzebnie bajtów.

Podczas wykonywania faktoryzacji LU metodą Doolittle'a, możemy zredukować przedziały sumowania tak, aby nie mnożyć niepotrzebnie przez zero. Mamy zatem następujące wzory dla $i \in [0, n)$, ze zmodyfikowanymi przedziałami, gdzie lb to szerokość dolna, a ub to szerokość górna:

$$u_{ij} = a_{ij} - \sum_{k=q}^{j-1} l_{ik} u_{kj} \qquad j \in [i, n) \land q = \max\{0, i - lb\}$$

$$l_{ji} = \frac{1}{u_{ii}} (a_{ji} - \sum_{k=w}^{i-1} l_{jk} u_{ki}) \qquad j \in [i+1, n) \land w = \max\{0, j - ub\}.$$

Dodatkowo możemy ograniczyć przedziały sumowania dla **backward-substitution** i **forward-substitution**, żeby pominąć dodawanie i mnożenie przez zero.

Rozpatrzmy macierz A i znajdźmy $y = A^{-1}x$, dla $N = 100$, gdzie:

$$A = \begin{bmatrix} 1.2 & \frac{0.1}{1} & \frac{0.4}{1^2} & & & & \\ 0.2 & 1.2 & \frac{0.1}{2} & \frac{0.4}{2^2} & & & \\ & 0.2 & 1.2 & \frac{0.1}{3} & \frac{0.4}{3^2} & & \\ & & \ddots & \ddots & \ddots & \ddots & \\ & & & 0.2 & 1.2 & \frac{0.1}{N-2} & \frac{0.4}{(N-2)^2} \\ & & & & 0.2 & 1.2 & \frac{0.1}{N-1} \\ & & & & & 0.2 & 1.2 \end{bmatrix}, \quad x = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ \vdots \\ N \end{bmatrix}.$$

2 Wyniki

Wyniki, jaki mi wyszły zgadzają się z wynikami z bilbioteki **numpy**:

$$A^{-1}x = \begin{bmatrix} 0.03287133486041399 \\ 1.339622798096375 \\ 2.066480295894664 \\ 2.825543605175336 \\ 3.557571715528883 \\ \vdots \\ 71.53915685603329 \end{bmatrix}, \quad det(A) = 78240161.00959387.$$

3 Podsumowanie

Reasumując, chcąc obliczać skończone układy równań liniowych, zawsze sprawdzajmy z jakim typem macierzy mamy do czynienia i dobierajmy odpowiednie algorytmy. Pozwoli to na znaczne zwiększenie wydajności programu oraz mniejsze zużycie pamięci, jeżeli będziemy przechowywać macierz w sposób optymalny.