

Krzysztof Czarnowus	zadanie NUM3	grupa 3
---------------------	--------------	---------

1. Wstęp

Rozwiązywanie układów równań macierzowych na maszynach cyfrowych jest problemem bardzo kosztownym obliczeniowo. Jednym z proponowanych rozwiązań poprawienia wydajności procesu jest wykonanie faktoryzacji wyjściowej macierzy i jej rozkładu na macierze L oraz U w odpowiedni sposób:

$$A y = x \quad (1)$$

$$L (U y) = x \quad (2)$$

gdzie macierz L jest macierzą trójkątną dolną zawierającą na diagonalu jedynek, a U macierzą trójkątną górną.

Najbardziej wyraźny spadek czasu wykonywania obliczeń w tej metodzie można zaobserwować dla szczególnego przypadku macierzy rzadkiej: macierzy wstęgowej, która posiada niezerowe wartości jedynie na diagonalu oraz tuż pod i nad nią. W zadaniu NUM3 dokonano faktoryzacji LU bez wyboru elementu podstawowego macierzy wstęgowej A:

$$A = \begin{pmatrix} 1.2 & \frac{0.1}{1} & \frac{0.15}{1^2} & & & & & & & \\ 0.2 & 1.2 & \frac{0.1}{2} & \frac{0.15}{2^2} & & & & & & \\ & 0.2 & 1.2 & \frac{0.1}{3} & \frac{0.15}{3^2} & & & & & \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ & & & & & 0.2 & 1.2 & \frac{0.1}{N-2} & \frac{0.15}{(N-2)^2} & \\ & & & & & & 0.2 & 1.2 & \frac{0.1}{N-1} & \\ & & & & & & & 0.2 & 1.2 & \end{pmatrix}$$

Dla takiego przypadku macierzy nie trzeba rezerwować w pamięci tablicy o rozmiarach $n \times n$, a jedynie $4 \times n$. Zadany wektor x we wzorze 1. ma postać:

$$x = [1, 2, \dots, n]^T$$

Na podstawie znajomości schematów obliczania elementów macierzy L oraz U w metodzie Doolittle'a oraz analizie tego, które elementy rozwinięcia będą się zerowały, wyprowadzono wzory dotyczące badanej macierzy:

$$u_{i,i} = a_{i,i} - l_{i,i-1} u_{i-1,i} \quad (3)$$

$$u_{i,i+1} = a_{i,i+1} - l_{i,i-1} u_{i-1,i+1} \quad (4)$$

$$u_{i,i+2} = a_{i,i+2} \quad (5)$$

$$l_{i+1,i} = a_{i+1,i} / u_{ii} \quad (6)$$

Obliczane za ich pomocą kolejne elementy nadpisywano w tablicy zawierającej początkowo macierz A, a następnie przekształcając odpowiednie wzory obliczono poszukiwany wektor rozwiązań y dla wymiaru macierzy równego 124.

Otrzymane wartości porównano z wynikiem uzyskanym za pomocą biblioteki NumPy, potwierdzając poprawność metody przez zgodność wartości. Zbadano również złożoność obliczeniową obu rozwiązań, wykonując dla napisanych programów powtarzalne obliczenia z wzrastającym wymiarem macierzy A.

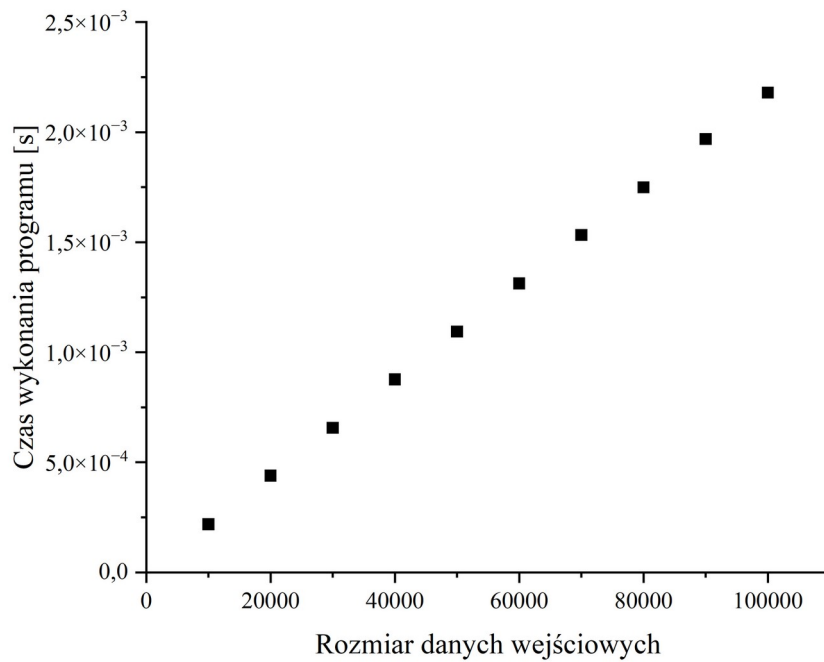
Dla programu stworzonego w języku C++ wykonującego faktoryzację LU zmierzono czas działania dla wymiarów macierzy z zakresu 10 000 – 100 000 z krokiem równym 10 000. Każde obliczenia powtórzono 5000 razy, aby otrzymać odpowiednio wysoką wartość, po czym podzielono go przez ilość powtórzeń, aby znać dokładny czas jednokrotnego przeprowadzenia obliczeń.

Program w języku Python wykorzystujący bibliotekę NumPy bada natomiast macierze o wymiarach z zakresu 1000 – 10 000 z krokiem 1000.

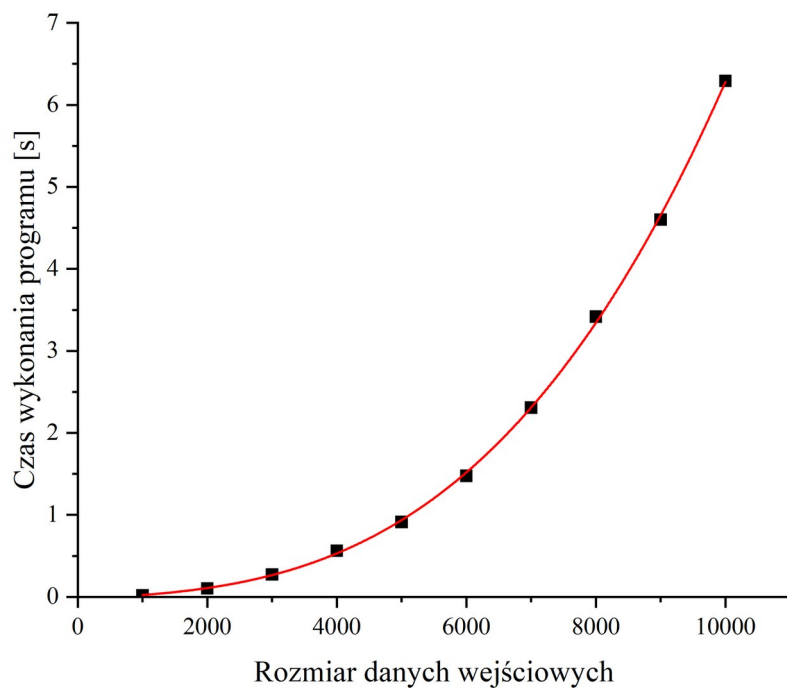
2. Otrzymane wyniki i dyskusja

Na rysunkach 1. oraz 2. przedstawiono zależność czasu obliczania wektora rozwiązań od wymiaru macierzy, a więc danych wejściowych. Niemożliwe było ich przedstawienie na jednym wykresie ze względu na zbyt dużą rozbieżność odnotowanych wartości – program napisany specjalnie pod badaną macierz okazał się być zbyt szybki w porównaniu do stosowania biblioteki algebry liniowej.

Można zaobserwować, że dla programu dedykowanego dla badanej macierzy i uwzględniającego jej strukturę zależność czasu wykonywanych obliczeń od rozmiaru danych wejściowych jest liniowa – zgodnie z oczekiwaniami złożoność obliczeniowa napisanego algorytmu jest $O(n)$. Do punktów otrzymanych w wyniku wykorzystania procedur bibliotecznych algebry liniowej zdefiniowanych w bibliotece NumPy za pomocą programu Origin dopasowano krzywą wielomianową trzeciego stopnia; złożoność obliczeniowa tego rozwiązania wynosi zatem $O(n^3)$.



Rysunek 1. Zależność czasu działania programu od rozmiaru danych wejściowych dla programu dedykowanego dla zadanej macierzy napisanego w języku C++.



Rysunek 2. Zależność czasu działania programu od rozmiaru danych wejściowych dla programu używającego biblioteki NumPy napisanego w języku Python wraz z dopasowaną krzywą wielomianową trzeciego stopnia.

W tabeli 1. przedstawiono otrzymane dla obu programów wartości przedstawione na rysunkach 1. oraz 2.

Tabela 1. Czas obliczania rozwiązania równania dla różnych wymiarów macierzy A.

program dedykowany		biblioteka NumPy	
wymiar macierzy	czas działania [ms]	wymiar macierzy	czas działania [s]
10 000	0.219	1000	0.020
20 000	0.439	2000	0.104
30 000	0.657	3000	0.273
40 000	0.876	4000	0.562
50 000	1.094	5000	0.914
60 000	1.312	6000	1.476
70 000	1.533	7000	2.309
80 000	1.749	8000	3.418
90 000	1.968	9000	4.601
100 000	2.180	10 000	6.292

Obie badane metody doprowadziły do otrzymania tego samego wektora rozwiązań dla wymiaru macierzy równego 124:

$$y = \begin{bmatrix} 0.4487 \\ 1.4133 \\ 2.1349 \\ \vdots \\ \vdots \\ \vdots \\ 87.9078 \\ 88.6820 \end{bmatrix}$$

Dzięki przeprowadzeniu faktoryzacji LU bardzo szybko staje się również obliczenie wyznacznika wyjściowej macierzy. Wiadomo, że wyznacznik iloczynu macierzy równy jest iloczynowi ich wyznaczników, zatem:

$$\det(A) = \det(LU) = \det(L) \times \det(U) \quad (7)$$

Ponadto wyznacznik macierzy trójkątnej równy jest iloczynowi elementów na jej diagonalu. Ponieważ diagonalą macierzy L składa się wyłącznie z jedynek, dla zadanej macierzy o wymiarze 124 wyznacznik wynosi:

$$\det(A) = \det(U) = \prod_{i=1}^{124} u_{ii} = 6,142 \times 10^9$$

3. Podsumowanie

Odpowiednie dobranie metody obliczeniowej specyficznej dla badanej macierzy rzadkiej przełożyło się na wyjątkowo wyraźnie poprawienie kosztów obliczeniowych liczenia rozwiązań równań macierzowych z jej udziałem. Problemem napisanego programu jest jednak brak uniwersalności – nadaje się on do wykonywania obliczeń jedynie dla macierzy wstęgowych o strukturze analogicznej do badanej macierzy; obecność innych niezerowych elementów prowadziłaby do konieczności przekształcenia wzorów 3.-6., co wiązałoby się z dodatkową pracą.