Autor: Marcin Sitko

SPRAWOZDANIE NUM4

TRESC ZADANIA:

7. **(zadanie numeryczne NUM4)** Zadana jest macierz

$$
\mathbf{A} = \begin{pmatrix}
10 & 8 & 1 & 1 & \dots & 1 & 1 & 1 & 1 \\
1 & 10 & 8 & 1 & \dots & 1 & 1 & 1 & 1 \\
1 & 1 & 10 & 8 & \dots & 1 & 1 & 1 & 1 \\
\dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\
1 & 1 & 1 & 1 & \dots & 1 & 10 & 8 & 1 \\
1 & 1 & 1 & 1 & \dots & 1 & 1 & 10 & 8 \\
1 & 1 & 1 & 1 & \dots & 1 & 1 & 1 & 10
\end{pmatrix}
$$

oraz wektor $\mathbf{b} \equiv (5, \dots, 5)^T$. Macierz $\mathbf{A}$ ma liczby 10 na diagonali, 8 na pierwszej pozycji nad diagonalą, a pozostałe elementy są równe 1. Wymiar macierzy ustalamy na $N = 50$.

- Rozwiąż numerycznie równanie $\mathbf{Ay} = \mathbf{b}$, dobierając odpowiedni algorytm. **Uwaga:** algorytm należy go zaimplementować samodzielnie (mile widziane jest jednak sprawdzenie wyniku przy użyciu procedur bibliotecznych lub pakietów algebry komputerowej).
- Potraktuj N jako zmienną i zmierz czas potrzebny do uzyskania rozwiązania w funkcji N. Wynik przedstaw na wykresie. Czy wykres jest zgodny z oczekiwaniami?

Do rozwiązania tego zadania, a przede wszystkim rozwiązania równania macierzowego $Ay = b$ w tym zadaniu użyłem algorytmu Shermana-morissona którego wzór wyraża się w następujący sposób:

$$(A + uv^T)^{-1} = A^{-1} - \frac{A^{-1}uv^TA^{-1}}{1 + v^TA^{-1}u}.$$

Macierz podaną w zadaniu można w prosty sposób przekształcić do postaci macierzy rzadkiej dzięki poprzez odjecie od macierzy A macierzy takiej samej wielkości wypełnionej samymi jedynkami którą możemy uzyskać w prosty sposób mianowicie wykonując następujące mnożenie:

$$u = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}, \; u^T = \begin{bmatrix} 1 & 1 \cdots & 1 \end{bmatrix}, \; uu^T = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix}$$

$$
A - vv^T = \begin{bmatrix}
9 & 7 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\
0 & 9 & 7 & 0 & \cdots & 0 & 0 & 0 & 0 \\
0 & 0 & 9 & 7 & \cdots & 0 & 0 & 0 & 0 \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
0 & 0 & 0 & 0 & \cdots & 9 & 7 & 0 & 0 \\
0 & 0 & 0 & 0 & \cdots & 0 & 0 & 9 & 7 \\
0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 9
\end{bmatrix}
$$

Dzięki temu dla tak skonstruowanej macierzy można użyć w bardzo efektywny sposób algorytmu Shermana-Morissona, który w tym przypadku dla podanej macierzy pokazanej powyżej będzie rozwiązywał równanie macierzowe $Ay = b$ ze złożonością obliczeniową $O(n)$.

We wzorze Shermana-Morssina można zauważyć że wyrażenie $A^{-1}u$ można zapisać jako $Az = b$ oraz dodatkowo zauważamy ze $A^{-1}b$ można zapisać jako $Ax = u$, Dzięki takiemu zapsiowi otrzymuje dwa kolejne równania macierzowe

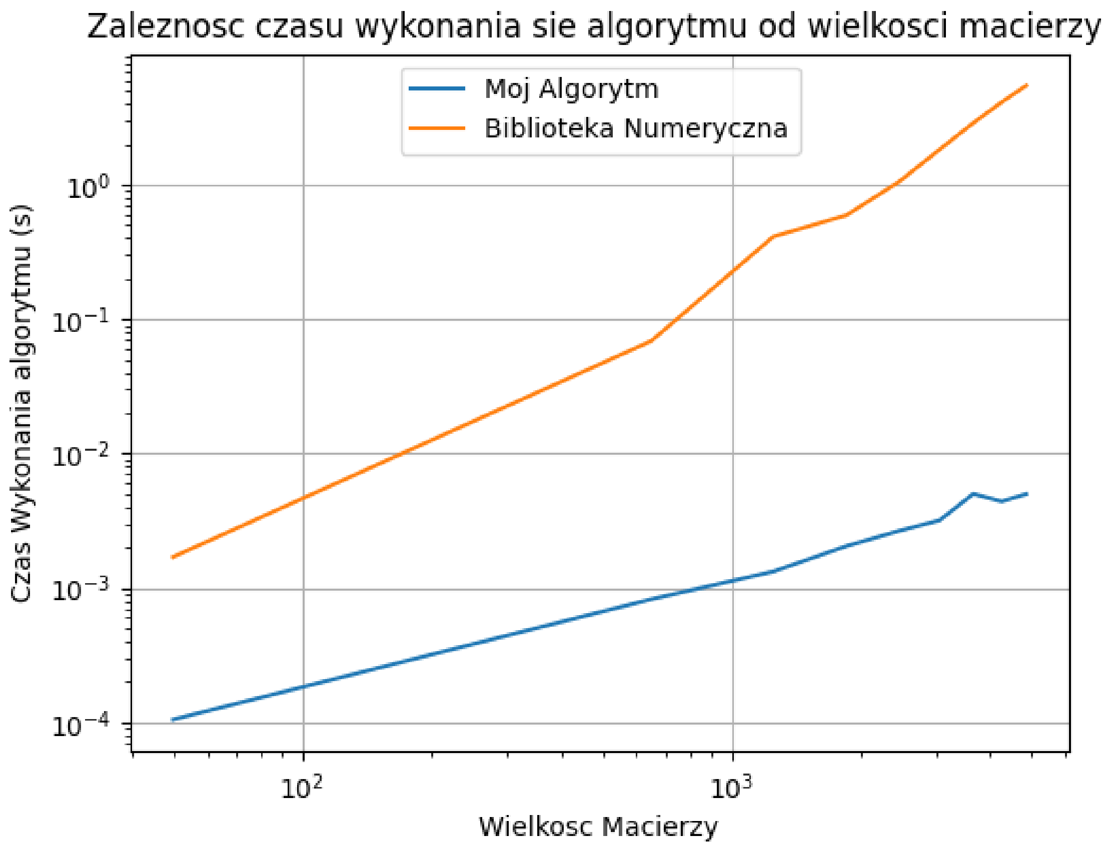
$$A\vec{z} = \vec{b}$$
$$A\vec{x} = \vec{u}$$

Dzięki tym równaniom nie musimy liczyć kosztownej odwrotności macierzy A, lecz rozwiązać te równanie korzystając z konstrukcji macierzy A

Podane dwa równania rozwiązujemy ze złożonością O(n) czyli bardzo szybko w porównaniu do pierwotnej wersji algorytmu Shermana-morissona którego złożoność w ogólności wynosi $O(n^3)$

Wyniki które uzyskuje moja implementacja algorytmu Shermana-Morssiona dla tej macierzy pokrywają się z wynikami które uzyskuje biblioteka numeryczna Aby dokonać tego sprawdzenia policzyłem normę z różnicy : wektora uzyskanego przez moja implementacje wraz oraz wektora uzyskanego przez bibliotekę numeryczna norma ta wynosi 0. Wiec wyniki są identyczne, dodatkowo po odjęciu tych wektorów otrzymuje wektor zerowy co utwierdza mnie w przekonaniu że moje rozwiązanie oraz wyniki biblioteki numerycznej są zgodne.

Wykonałem także wykres który pokazuje ile czas potrzebuje mój algorytm do dokonania wyliczeń, dodatkowo zestawiłem ten wykres z wykresem czasu dla tej samej czynności wykonanej przez bibliotekę numeryczną wyniki są bardzo zadowalające

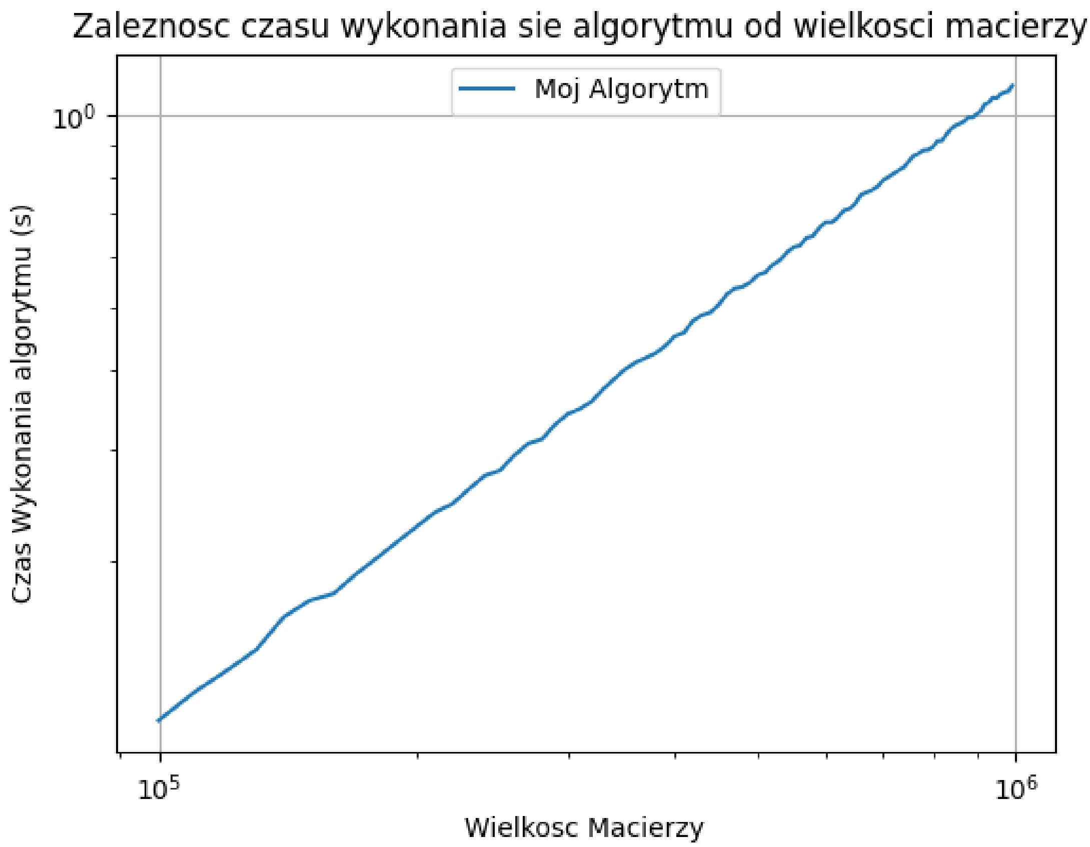


Zastosowałem skalę logarytmiczną tak aby było można zobaczyć wyraźniej różnice pomiędzy poszczególnymi czasami wykonania algorytmu a wielkością macierzy, użyłem tutaj niewielkiego rozmiaru macierzy jest to w maksymalnym momencie zaledwie macierz 5000x5000. Natomiast funkcje obrazująca wykres wykreśliłem na podstawie analizy czasu wykonania algorytmów dla macierzy za zakresu 100 – 5000 z krokiem 600 to znaczy ze algorytm wykonuje się dla macierzy o rozmiarach : 100,700,1200,1800 … itd.

Widać ze różnica pomiędzy czasem wykonania „mojego" algorytmu a czasem wykonania obliczeń za pomocą biblioteki numerycznej dla macierzy rozmiaru maksymalnie 5000x5000 wynosi około 3 rzędy wielkości mniej jest bardzo duża różnica na tak małych rozmiarów macierzy , ponieważ można zauważyć ze dla coraz to większych wielkości macierzy A ten wykres będzie coraz bardziej i znaczniej się rozbiegał.

Widać ze niebieski wykres w całym przebiegu na wykresie zmienił swój rząd wielkości zaledwie o 1 natomiast dla pomarańczowego wykresu zmiana rzędu wielkości wynosi około 3.5 ta rozbieżność będzie się powiększać jeżeli zaczniemy dobierać znacznie większe macierze co oznacza, że jeżeli wielkość macierzy A będzie rosła to „mój algorytm" będzie coraz wyraźniej zaznaczał swoją szybkość względem algorytmu biblioteki numerycznej.

Dodatkowo postanowiłem przetestować szybkość algorytmu i zastosowałem „mój algorytm" Shermana Morisona dla macierzy A rozmiarów z zakresu 100.000 - 1.000.000 z krokiem 10.000 uzyskany rezultat jest bardzo ciekawy ponieważ maksymalny czas jaki potrzebował algorytm na policzenie maksymalnej macierzy 1.000.000 x 1.000.000 wynosi zaledwie 1.2s natomiast większość poprzednich równań macierzowych udało się rozwiać poniżej 1 sekundy



Widać ze czas potrzebny na wykonanie algorytmu rośnie liniowo to zgadza się z moimi wcześniejszymi założeniami , iż algorytm powinien wykonywać się ze złożonością liniowa O(n).