Sprawozdanie NUM4

Szymon Tomaszewski

16 grudnia 2024

1. Wstęp

1.1. Treść zadania

Zadana jest macierz

$$\mathbf{A} = \begin{pmatrix} 5 & 3 & 1 & 1 & \dots & 1 & 1 & 1 & 1 \\ 1 & 5 & 3 & 1 & \dots & 1 & 1 & 1 & 1 \\ 1 & 1 & 5 & 3 & \dots & 1 & 1 & 1 & 1 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & 1 & 1 & 1 & \dots & 1 & 5 & 3 & 1 \\ 1 & 1 & 1 & 1 & \dots & 1 & 1 & 5 & 3 \\ 1 & 1 & 1 & 1 & \dots & 1 & 1 & 1 & 5 \end{pmatrix}$$

oraz wektor $b \equiv (2, \dots, 2)^T$ Macierz A ma liczby 5 na diagonali, 3 na pierwszej pozycji nad diagonalą, a pozostałe elementy są równe 1. Wymiary macierzy ustalamy na N=120

- Rozwiąż numerycznie równanie Ay=b, stosując odpowiednią metodę. Uwaga: algorytm należy zaimplementować samodzielnie.
- Sprawdź swój wynik 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?

1.2. Cel zadania

Celem zadania jest napisanie programu który efektywnie rozwiąże równanie Ay=b, gdzie A jest zadaną macierzą.

1.3. Wprowadzenie

Pojawienie się w wzorze “$A^{-1}b$” powinniśmy zawsze rozumieć jako zaproszenie do znalezienia wektora **z** takiego, że A**z**=b tym samym odradzane jest jawne skonstruowanie macierzy A^{-1}, ponieważ jest to obliczeniowo kosztowne.

W tym zadaniu musimy rozwiązać już niejako wyprowadzony owy wzór korzystając z struktury macierzy A. Ważne jest aby skorzystać z tej struktury, ponieważ pozwoli to na znaczną redukcję złożoności obliczeniowej i pamięciowej.

Jesteśmy w stanie zauważyć, że poza danymi na wstędze macierz A zawiera jedynie elementy o wartości 1, więc jeśli sprowadzimy macierz A do macierzy rzadkiej to obliczenia znacząco się skrócą. Jesteśmy to w stanie zrealizować, ponieważ zbudowana jest głównie z jedynek.

$$\begin{pmatrix} 5 & 3 & 1 & \cdots & 1 \\ 1 & 5 & 3 & \cdots & 1 \\ 1 & 1 & 5 & \cdots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \cdots & 5 \end{pmatrix} = \begin{pmatrix} 4 & 2 & \cdots & 0 \\ 0 & 4 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 4 \end{pmatrix} + \begin{pmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{pmatrix}$$

(Każda z tych macierzy to macierz N na N)

Teraz jesteśmy w stanie przedstawić tą macierz w postaci równania: $y = (B + uv^T)^{-1}b,\ gdzie\ A = B + uv^T$ *(1)*

Teraz skorzystamy z wzoru Shermana-Morrisona, gdzie wymogiem jest przedstawienie macierzy przy pomocy zaburzenia: :

$$(B + uv^T)^{-1} = B^{-1} - \frac{B^{-1}uv^TB^{-1}}{1+v^TB^{-1}u} \quad (2)$$

Podstawiając *(2)* do równania nr *(1)* otrzymamy:

$$y = (B^{-1} - \frac{B^{-1}uv^TB^{-1}}{1+v^TB^{-1}u})^{-1}b = B^{-1}b - \frac{B^{-1}uv^TB^{-1}}{1+v^TB^{-1}u}b$$

Możemy dostrzec, że macierz we wzorze pojawia się w formie B^{-1}, otrzymujemy wtedy:

$$y = p - \frac{qv^{-1}p}{1+v^Tq},\ gdzie\ p = B^{-1}b,\ q = B^{-1}u$$

Macierz B jest trójkątna górna, co eliminuje konieczność przeprowadzania dodatkowego rozkładu. Zamiast tego można skorzystać z metody forward substitution i backward substitution. W konsekwencji musimy znaleźć rozwiązania równań dla **p** i **q**.
Metoda ta charakteryzuje się złożonością obliczeniową O(n), więc będzie miała lepsze osiągi niż metoda z NumPy, będzie to widoczne zwłaszcza przy dużych macierzach.

2. Kod

Program został zaimplementowany w Pythonie 3, a do analizy wyników i czasu wykonania wykorzystano bibliotekę NumPy oraz Matplotlib. Aby zminimalizować wahania na wykresie, pomiary były powtarzane wielokrotnie dla zadanego rozmiaru macierzy.

3. Wyniki

NumPy:

```
[0.01557936 0.01557936 0.01557936 0.01557936 0.01557936 0.01557936
 0.01557936 0.01557936 0.01557936 0.01557936 0.01557936 0.01557936
 0.01557936 0.01557936 0.01557936 0.01557936 0.01557936 0.01557936
 0.01557936 0.01557936 0.01557936 0.01557936 0.01557936 0.01557936
 0.01557936 0.01557936 0.01557936 0.01557936 0.01557936 0.01557936
 0.01557936 0.01557936 0.01557936 0.01557936 0.01557936 0.01557936
 0.01557936 0.01557936 0.01557936 0.01557936 0.01557936 0.01557936
 0.01557936 0.01557936 0.01557936 0.01557936 0.01557936 0.01557936
 0.01557936 0.01557936 0.01557936 0.01557936 0.01557936 0.01557936
 0.01557936 0.01557936 0.01557936 0.01557936 0.01557936 0.01557936
 0.01557936 0.01557936 0.01557936 0.01557936 0.01557936 0.01557936
 0.01557936 0.01557936 0.01557936 0.01557936 0.01557936 0.01557936
 0.01557936 0.01557936 0.01557936 0.01557936 0.01557936 0.01557936
 0.01557936 0.01557936 0.01557936 0.01557936 0.01557936 0.01557936
 0.01557936 0.01557936 0.01557936 0.01557936 0.01557936 0.01557936
 0.01557935 0.01557936 0.01557935 0.01557937 0.01557933 0.0155794
 0.01557928 0.01557948 0.01557915 0.0155797  0.01557879 0.01558031
 0.01557778 0.01558199 0.01557496 0.01558668 0.01556715 0.01559971
 0.01554544 0.01563588 0.01548515 0.01573636 0.01531768 0.01601548
 0.01485249 0.01679081 0.01356027 0.0189445  0.00997079 0.02492697]
```

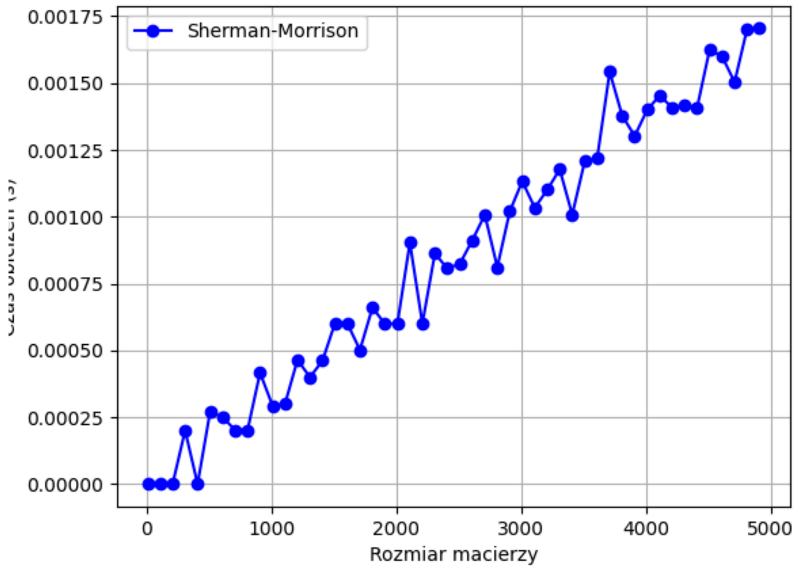
Sherman-Morrison:

```
[0.015579357351509254, 0.015579357351509254, 0.015579357351509254, 0.015579357351509254,
0.015579357351509254, 0.015579357351509254, 0.015579357351509254, 0.015579357351509254,
0.015579357351509254, 0.015579357351509254, 0.015579357351509254, 0.015579357351509254,
0.015579357351509254, 0.015579357351509254, 0.015579357351509254, 0.015579357351509254,
0.015579357351509254, 0.015579357351509254, 0.015579357351509254, 0.015579357351509254,
0.015579357351509254, 0.015579357351509254, 0.015579357351509254, 0.015579357351509254,
0.015579357351509254, 0.015579357351509254, 0.015579357351509254, 0.015579357351509254,
0.015579357351509254, 0.015579357351509254, 0.015579357351509254, 0.015579357351509254,
0.015579357351509254, 0.015579357351509254, 0.015579357351509254, 0.015579357351509254,
0.015579357351509254, 0.015579357351509254, 0.015579357351509254, 0.015579357351509254,
0.015579357351509254, 0.015579357351509254, 0.015579357351509254, 0.015579357351509254,
0.015579357351509254, 0.015579357351509254, 0.015579357351509254, 0.015579357351509254,
0.015579357351509254, 0.015579357351509254, 0.015579357351509254, 0.015579357351509254,
0.015579357351509254, 0.015579357351509282, 0.015579357351509227, 0.01557935735150931, 0.015579357351509171,
0.01557935735150942, 0.015579357351508977, 0.015579357351509698, 0.015579357351508505, 0.015579357351510531,
0.015579357351507145, 0.01557935735151278, 0.01557935735150337, 0.015579357351519052, 0.015579357351492934,
0.015579357351536455, 0.01557935735146393, 0.015579357351584805, 0.015579357351383327, 0.015579357351719142,
0.01557935735115945, 0.015579357352092232, 0.015579357350537615, 0.015579357353128653, 0.015579357348810247,
0.0155793573560076, 0.015579357344012001, 0.015579357364004676, 0.015579357330683552, 0.01557935738621874,
0.015579357293660084, 0.015579357447924519, 0.015579357190817156, 0.015579357619329437,
0.015579356905142283, 0.015579358095454215, 0.015579356111600995, 0.01557935941802302, 0.015579353907319654,
0.015579363091825255, 0.015579347784315939, 0.015579373296831456, 0.015579330775972233,
0.015579401644070928, 0.015579283530573113, 0.015579480386402805, 0.015579152293353327,
0.015579699115102486, 0.015578787745520545, 0.01558030669482377, 0.01557777511265171, 0.01558199441627181,
0.015574962243571633, 0.015586682531405271, 0.015567148718349216, 0.01559970507344266, 0.015545444481620263,
0.01563587880132425, 0.01548515493515093, 0.015736361378773128, 0.01531768397273614, 0.016015479649464454,
0.014852486854917246, 0.01679080817916262, 0.013560272638753645, 0.01894449853943525, 0.009970788704965927,
0.024926971762414818]
```

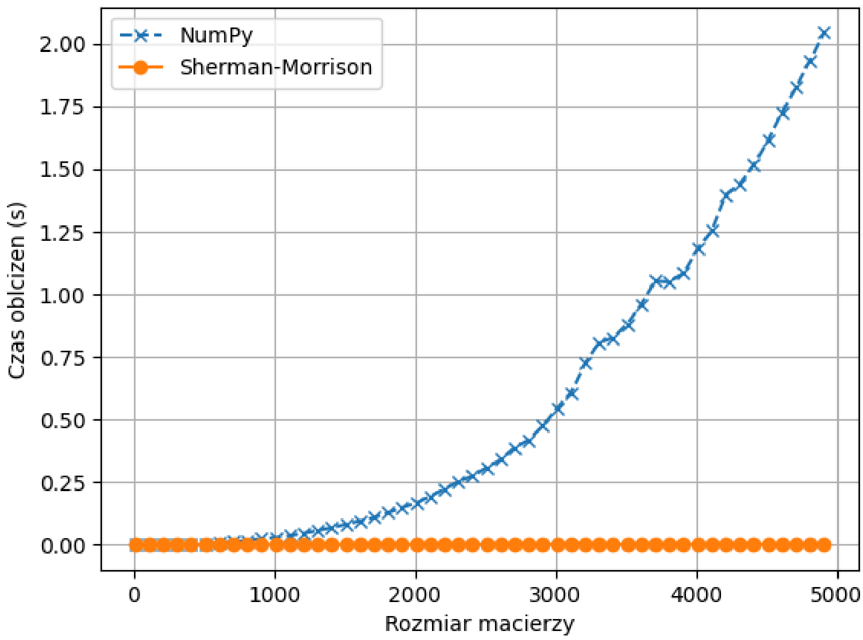
4. Wnioski

Zgodnie z naszymi oczekiwaniami, metoda oparta na Shermanie-Morrisonie okazuje się znacznie bardziej efektywniejsza, szczególnie w przypadku dużych macierzy, lecz technika ta jest skuteczna tylko w przypadku jeśli jesteśmy w stanie wyrazić tą macierz poprzez zaburzenie. Na wykresie można dopatrzyć się liniowej zależności O(n).

Różnice pomiędzy wynikami uzyskanymi między Sherman-Morrisonem a metodą z Numpy jest minimalna i wynosi średnio $7 * 10^{-18}$ a owy błąd wynika najprawdopodobniej z ograniczenia precyzji obliczeń.



Implementacja Sherman-Morrison ma złożoność liniową



Własna implementacja jest znacząco szybsza