Autor: Marcin Sitko

Sprawozdanie – NUM3

Macierz A

$$
\mathbf{A} = \begin{pmatrix}
1.2 & \frac{0.1}{1} & \frac{0.4}{1^2} & & & & & & \\
0.2 & 1.2 & \frac{0.1}{2} & \frac{0.4}{2^2} & & & & & \\
 & 0.2 & 1.2 & \frac{0.1}{3} & \frac{0.4}{3^2} & & & & \\
\dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\
 & & & & & 0.2 & 1.2 & \frac{0.1}{N-2} & \frac{0.4}{(N-2)^2} \\
 & & & & & & 0.2 & 1.2 & \frac{0.1}{N-1} \\
 & & & & & & & 0.2 & 1.2
\end{pmatrix}
$$

Wektor Wyrazów wolnych

$$x = (1, 2, \dots, N)^T$$

Treść zadania:

$$\textit{Wyznacz } y = A^{-1}x$$

Zadanie rozpocząłem od wybrania poprawnego algorytmu, dzięki któremu będę w stanie optymalnie policzyć wynik równania macierzowego, zadanego w treści zadania.

Wybrałem rozwiązanie które składa się z:

1) Faktoryzacji LU macierzy A
2) następnie wykonanie Forward Substitution dla macierzy L oraz wektora wyrazów wolnych
3) a końcowym etapie wykonanie Back Substitution dla macierzy U oraz otrzymanego wektora w podpunkcie 2.

Do Faktoryzacji LU macierzy A użyłem zmodyfikowanego przeze mnie algorytmu Doolittle'a, dzięki temu Faktoryzacji LU dla macierzy A można dokonać ze złożonością obliczeniową rzędu $O(n)$, jest to duży zysk , ponieważ zwykly algorytm Faktoryzacji LU z wykorzystaniem metody Doolitlte'a ma złożoność obliczenia rzędu $O(n^3)$.

Wzory równań potrzebnych do wyznaczania macierzy L oraz U za pomocą metody Doolittle'a:

$$u_{k_j} = a_{k_j} - \sum_{m-1}^{k-1} l_{k_m} u_{m_k}$$

$$l_{i_k} = \frac{\left(a_{i_k} - \sum_{m=1}^{k-1} l_{i_m} u_{m_k}\right)}{u_{k_k}}$$

Dla naszej macierzy, te równania będą się bardzo upraszczały, i dzięki temu nie będziemy iterować po elementach zerowych których jest znacznie więcej, ponieważ jest to macierz rzadka.

Dzięki temu osiągniemy znaczny wzrost wydajności naszego algorytmu rozkładu LU, tak że będziemy w stanie wykonać go w czasie liniowym, oto moje przekształcenia powyższych wzorów na rozkład LU macierzy A:

$$Moje\ Wyniki\ wyprowadzenia\ wzorów\ algorytmu\ Doolittle'a\ dla\ macierzy\ z\ zadania$$

$$u_{ii} = A_{ii} - L_{i(i-1)} U_{(i-1)i}$$

$$u_{i(i+1)} = A_{i(i+1)} - L_{i(i-1)} U_{(i-1)(i+1)}$$

$$L_{(i+1)i} = \frac{A_{(i+1)i}}{U_{ii}}$$

$$u_{i(i+2)} = A_{i(i+2)}$$

Wyprowadzenia powyższych wzorów załączam na końcu sprawozdania.

Po dokonaniu rozkładu LU, musimy także dokonać operacji Back Substitution oraz Forward Substitution. Obie operacje są rzędu $O(n^2)$, natomiast są one operacjami względnie tanimi, w porównaniu do operacji rozkładu LU, która jest rzędu $O(n^3)$. Jeżeli jesteśmy w stanie otrzymać rozkład LU w czasie liniowym to, możemy rozwiązać każde równanie z tym rozkładem oraz dowolnym wektorem wyrazów wolnych ze złożonością obliczeniową rzędu $O(n^2)$.

Mój algorytm powinien wykonać się znacznie szybciej w stosunku do algorytmy biblioteki numerycznej $Scipy$, ponieważ mój rozkład LU dokonuje się w czasie liniowym.

Po uruchomieniu mojego programu , oraz wybraniu opcji $numer$ 1,ukazuje się nam czas wykonania mojego algorytmu rozkładu LU oraz operacji Back Substitution, Forward Substitution. Widać także czas wykonania tej samej operacji rozkładu LU oraz Back Substitution ,Forward Substitution wykonany poprzez bibliotekę numeryczna $scipy.linagl.lu(macierz, parametr)$, gdzie parametr jest to zmienna typu $boolean$ która określa typ przeprowadzanej operacji LU czy chcemy uzyskać zwykły rozkład $LU\ parametr = False$, czy jednak chcemy uzyskać rozwiązanie z pivotingiem czyli $PLU\ parametr = True$

$Wynik\ operacji\ numer$ 1:

```
CZAS WYKONANIA ROWNANIA MACIERZOWEGO
Bilbioteka numeryczna:  0.00727110000000053
Moj Wyspecjalizowny algorytm:  0.0013382000000001781

1) Zadanie NUM3
2) Sprawdz Poprawnosc Olbiczen
3) Zadanie Dodatkowe
4) Wyznacznik macierzy A

5) Wyjscie
: |
```

Widać, że mój algorytm, wykonuje się szybciej nawet dla tak małego rozmiaru macierzy jak 100x100 co utwierdza w przekonaniu, iż mój algorytm ma znacznie mniejsza złożoność obliczeniową.

Dodatkowo w moim programie można także sprawdzić poprawność obliczeń, dokonywanych przez mój algorytm, klikając $opcje\ numer$ 2 z pierwszego menu, wyświetli się wtedy wynik który produkuje mój algorytm Doolittle'a, Back Substitution, Forward Substitution oraz wynik dokonany przez bibliotekę numeryczną. $Opcja\ numer$ 2 dokonuje także odejmowania mojego wyniku od wyniku uzyskanego przez bibliotekę numeryczna, Wynik powinien być bardzo bliski zeru (oczywiście mogą występować błędy numeryczne)

Widać ze pozyskana macierz jest praktycznie równa 0 z małymi wyjątkami spowodowanymi błędami numerycznymi, obliczenia jakie wykonałem to operacja odejmowania macierzy uzyskanej przez mój algorytm z macierzą uzyskana przez algorytm biblioteki numerycznej.

$Wynik\ Odejmowania\ mojego\ wyniku\ od\ wyniku\ bilbioteki\ numerycznej\ Opcja\ numer\ 2$:

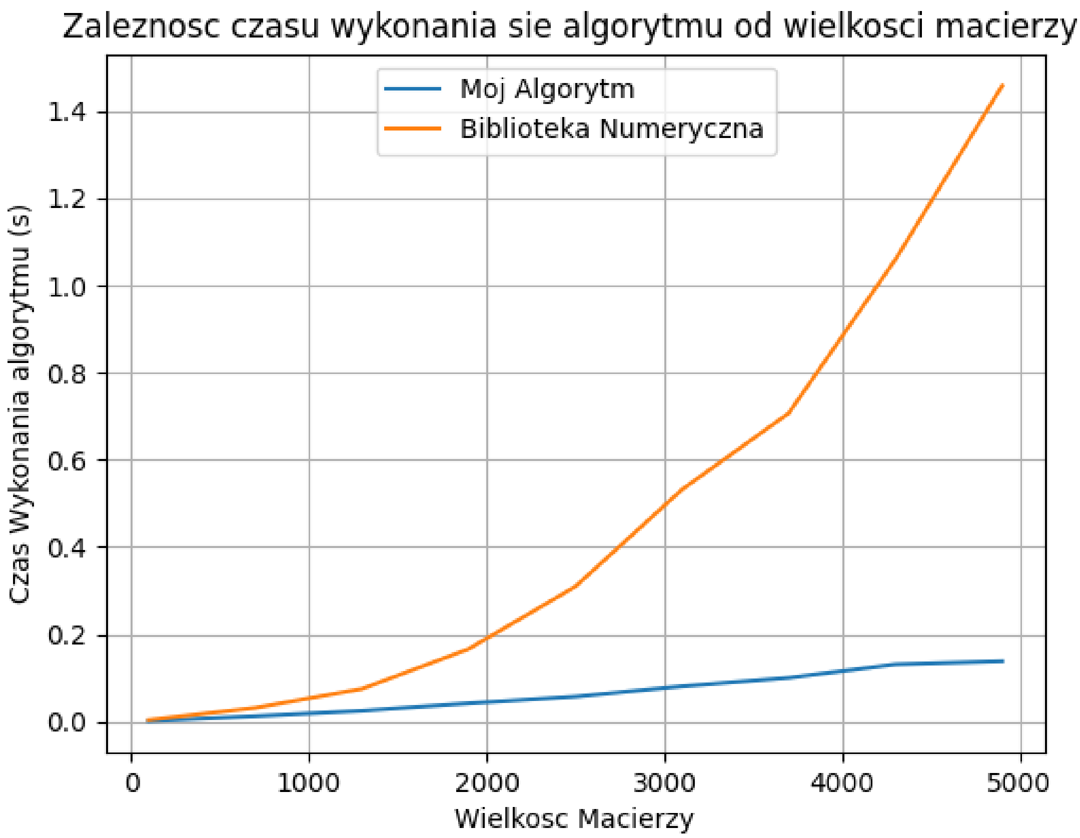
```
Sprawdzenie Poprawnosci Obliczen, vektor powinien byc bardzo bliski zeru

[0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.]
[0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.] [3.55271368e-15] [0.] [0.] [0.] [0.] [0.]
[0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.]
[0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.]
[0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.] [0.]
```

Widać, że moje wyniki pokrywają się z wynikami biblioteki numerycznej $Scipy$.

Dodatkowo wykonałem także „$zadanie\ dla\ ambitnych$”, $opcja\ numer\ 3$ z głównego menu Pojawia się dodatkowe menu które umożliwia nam wybranie konkretnych opcji.

```
1) Wykonaj Obliczenia Wystarczy Wykonac to 1 raz
2) Pokaz Wykres ilustrujacy zaleznosc pomiedzy wielkością n (macierzy) a czasem wykonania algorytmu
3) Pokaz czasy Wykonania algorytmow
4) Wyjscie
:
```

Obowiązkowo należy w pierwszej kolejności wybrać opcje $numer\ 1$, dzięki której, program dokona obliczeń potrzebnych do dalszych opcji, obliczenia wykonuje się kilka sekund ponieważ przyjąłem $n -\ z\ zakresu\ od\ 100 - 5.000\ z\ krokiem\ 600$, co oznacza że algorytmy (mój oraz biblioteka numeryczna) będą po kolei wykonywać Rozkładu LU, oraz Forward Substitution i Back Substitution z macierzami A oraz wektorami wyrazów wolnych o rozmiarach 100,600,1200,1800 … 5000. Następnie zliczone czasy wykonania tych algorytmow zostaną zebrane do tablicy i można je zobaczyć na wykresie wybierając $opcje\ numer\ 2$.



Widać ze mój algorytm znacznie lepiej poradził sobie z macierzami większych wielkości, jest dużo bardziej optymalny i działa w dużo krótszym czasie.

Dodatkowo można także zobaczyć wszystkie czasy z osobna wybierając *opcje numer* 3 z menu „zadania dla ambitnych”.

Przykladowy wynik opcji numer 3 *z menu dodatkowego zadania dla ambitnych*

```
MyTime:  0.0013947999999999183
MyTime:  0.012395500000000226
MyTime:  0.023569799999999752
MyTime:  0.03889549999999975
MyTime:  0.05524599999999946
MyTime:  0.07429449999999971
MyTime:  0.0946454999999986
MyTime:  0.11849319999999963
MyTime:  0.14854919999999971
Numercial module time:  0.009612999999999872
Numercial module time:  0.029428300000000185
Numercial module time:  0.07076949999999993
Numercial module time:  0.17888590000000004
Numercial module time:  0.48852790000000024
Numercial module time:  0.6062771000000007
Numercial module time:  0.759859500000001
Numercial module time:  1.0521794
Numercial module time:  1.3233633000000005
```

Widać ze mój algorytm jest zawsze szybszy w parowaniu do biblioteki numerycznej.

$$Wyprowadzania\ wzorów\ algorytmu\ Doolittle'a\ dla\ macierzy\ z\ zadania$$

$$\underline{U_{ii}} = a_{ii} - \sum_{k<i} l_{ik} U_{ki} = a_{ii} - l_{i,i-1}U_{i-1,i} + \underbrace{l_{i,i-2}U_{i-2,i}}_{0} + \dots = a_{ii} - l_{i,i-1}U_{i-1,i}$$

$$\underline{U_{i,i+1}} = a_{i,i+1} - \sum_{k<i} L_{ik}U_{k,i+1} = a_{i,i+1} - l_{i,i-1}U_{i-1,i+1} + \overbrace{l_{i,i-2}U_{i-2,i+1}}^{0} \dots = a_{i,i+1} - l_{i,i-1}U_{i-1,i+1}$$

$$\underline{L_{i+1,i}} = \frac{a_{i+1} - \sum_{k<i} l_{i+1,k}U_{k,i}}{U_{ii}} = \frac{a_{i+1} - \overbrace{l_{i+1,i-1}}^{0}U_{i-1,i}}{U_{ii}} = \underline{\frac{a_{i+1}}{U_{ii}}}$$

$$\underline{U_{i,i+2}} = a_{i,i+2} - \sum_{k<i} l_{ik}U_{k,i+2} = a_{i,i+2}\, l_{i,i-1}\underbrace{U_{i-1,i+2}}_{0} = \underline{a_{i,i+2}}$$