Lista 5

Adrian Mucha 236526

6 stycznia 2019

1 Problem

2 Rozwiązanie

2.1 Wykorzystanie pamięci

Zważywszy na małą ilość elementów niezerowych w macierzy zdecydowano się użyć struktury danych `SparseMatrixCSC` z języka `Julia`. Przechowuje ona dane w porządku kolumnowym, a więc iterując w tej strukturze najpierw po kolumnach zmniejszamy czas dostępu do wybranego elementu.

2.2 Eliminacja Gaussa

Algorytm eliminacji Gaussa posłuży nam do rozwiązywania układów równań oraz wyznaczania rozkładu LU. Metoda dzieli się na dwa zasadnicze etapy. Pierwszym z nich jest *Redukcja do macierzy górnej*. Polega na zerowaniu wszystkich elementów pod elementami głównymi występującymi na diagonali macierzy. W celu wyzerowania elementu a_{ik} znajdującego się pod a_{kk} obliczymy *mnożnik* $z_{ik} = a_{ik}/a_{kk}$ i odejmiemy wiersz k-ty pomnożony przez mnożnik od wiersza i-tego. Przypadek, w którym $a_k k$ jest zerem algorytm zawodzi i wymaga użycia permutacji wierszy. Następnie wykonujemy etap drugi, który nazywamy *Podstawieniem wstecz*. Polega on na konsekwentnym wykorzystywaniu wierszy znajdujących się poniżej do obliczenia danej wartości wektora spełniającego równanie. Ta metoda działa ze względu na wcześniej przygotowaną macierz trójkątną górną. Rozpoczynamy od podstawienia wiersza ostatniego do przedostatniego itd. Wartość i-tą wektora x obliczamy za pomocą wzoru

$$x_i = \frac{b_i - \sum_{j=i+1}^{n} a_{ij}}{a_{ii}}$$

2.2.1 Rozwiązywanie układów równań

Standardowa eliminacja Gaussa cechuje się złożonością $O(n^3)$ a podstawianie wstecz $O(n^2)$. Dzięki specyficznej budowie macierzy zmniejszymy tę złożoność do $O(n)$ przy pewnych założeniach. W sposobie rozwiązywania równań uwzględniona została specjalna trójdiagonalna blokowa postać macierzy. Należy zauważyć, że dzięki specyficznej postaci macierzy znacząco zmniejsza się ilość zerowanych wartości pod diagonalą. Wynika to z budowy podmacierzy A_i oraz B_i. Możemy wyprowadzić wzór na ostatni niezerowy element w kolumnie

$$lastRow(col) = min(n, l + l \cdot \lfloor \frac{col}{l} \rfloor)$$

Ponadto odejmując od siebie wiersze nie musimy modyfikować wszystkich wartości w wierszu, a jedynie odejmiemy te niezerowe. Wyprowadzimy wzór na najdalszą niezerową kolumnę w danych wierszu.

$$lastColumn(row) = min(n, row + l)$$

Wynika to z faktu iż ostatnim niezerowym elementem w danym wierszu jest zawsze element na przekątnej podmacierzy C_i. Przyjmując l jako stałą otrzymujemy żądaną złożoność obliczeniową $O(n)$.

2.2.2 Wybór elementu głównego

Powyższe metody nie radzą sobie gdy na przekątnej pojawia się zero. W tym przypadku stosuje się tzw. wybór elementu głównego, czyli zamianę wiersza z elementem zerowym na przekątnej z wierszem, którego wartość co do modułu jest największa w danej kolumnie, w której znajdował się element zerowy. Taka operacja dla dużych macierzy jest kosztowna, więc w praktyce wiersze pozostają na miejscu, ale stosuje się wektory permutacji zawierające informacje o zamienionych wierszach. Taki wektor dla macierzy niespermutowanej jest zwykłym wektorem w którym $p_i = i$. Odpowiednio należy również zmodyfikować algorytm eliminacji Gaussa - a mianowicie do wierszy odwołujemy się poprzez wektor permutacji ($a_i \rightarrow a_{p_i}$), co jest niewielką zmianą. Innym istotnym problemem jest zmiana formuły na ostatnią kolumnę w której mogły zajść zmiany. Zmodyfikujemy w tym celu poprzedni wzór na *lastColumn*.

$$lastColumn(row) = min(n, 2l + l \cdot \lfloor \frac{row - 1}{l} \rfloor)$$

Wynika on z obserwacji, że najdalszym niezerowym elementem podczas eliminowania współczynników w pierwszych l kolumnach jest element w kolumnie $2l$. Podczas eliminowania kolejnych l kolumn, najdalszą jest $3l$-kolumna itd. Zmiana dotyczy również algorytmu podstawiania wstecz. Wybór elementu głównego wpływa nieznacząco na złożoność pod warunkiem, że l jest stałą. Dzięki czemu nadal utrzymujemy $O(n)$.

2.3 Rozkład LU

Rozkład **LU** jest przedstawieniem macierzy A za pomocą iloczynu macierzy trójkątnej dolnej L oraz trójkątnej górnej U. Przyjmuje się że macierz trójkątna dolna zawiera na przekątnej same wartości 1. Uzyskanie rozkładu LU jest możliwe dzięki metodzie eliminacji Gaussa, która tworzy macierz trójkątną górną - macierz U. Macierz L powstanie poprzez zapisanie mnożników z w miejsce eliminowanych elementów pod diagonalą (z_{ik} zostanie zapisany w miejscu eliminowanego elementu a_{ik}). Rozkład LU ma złożoność obliczeniową $O(n^3)$. Można go jednak wykorzystać wielokrotnie do obliczania układów równań w przypadku gdy macierz pozostaje ta sama - wtedy obliczany jest tylko raz. Wtedy rozwiązanie układów równań sprowadza się do obliczenia $Lz = b$ oraz $Ux = z$ co sprowadza się do obliczenia dwóch układów macierzy trójkątnych, tym samym mających złożoność obliczeniową $O(n^2)$.

W przypadku gdy do znalezienia rozkładu LU użyjemy metody eliminacji Gaussa z założeniami dla macierzy specyficznej podanej w zadaniu (l jest stałą), to jej koszt spada do równowartości metody eliminacji Gaussa - $O(n)$. Algorytm zostaje jedynie zmodyfikowany o zapis mnożników w miejsca zerowanych elementów pod przekątną macierzy.

Znalezienie rozwiązania układu równań dla macierzy LU sprowadza się do przeprowadzenia *Podstawienie w przód* (które jest odwrotnym do podstawiania wstecz, z drobną modyfikacją) dla rozwiązania $Lz = b$ aby potem użyć wspomnianego już wcześniej *Podstawienia wstecz* by znaleźć ostateczne rozwiązanie równania $Ux = z$. Rozwiązywanie układów równań przy założeniach macierzy danej w zadaniu (l jest stałe) sprowadza się również do $O(n)$.

3 Wyniki

3.1 Błąd względny

Otrzymano następujące błędy względne dla obliczonych wektorów.

Metoda	$n = 16$	$n = 10000$	$n = 50000$
Gauss	$3.6912828624410226e-15$	$7.997607911218936e-14$	$2.6390293711611795e-14$
Gauss z Wyborem	$5.10280049072227e-16$	$3.905334752145677e-16$	$4.2015187204731675e-16$
LU	$3.6912828624410226e-15$	$7.997607911218936e-14$	$2.6390293711611795e-14$
LU z Wyborem	$5.10280049072227e-16$	$3.905334752145677e-16$	$4.2015187204731675e-16$

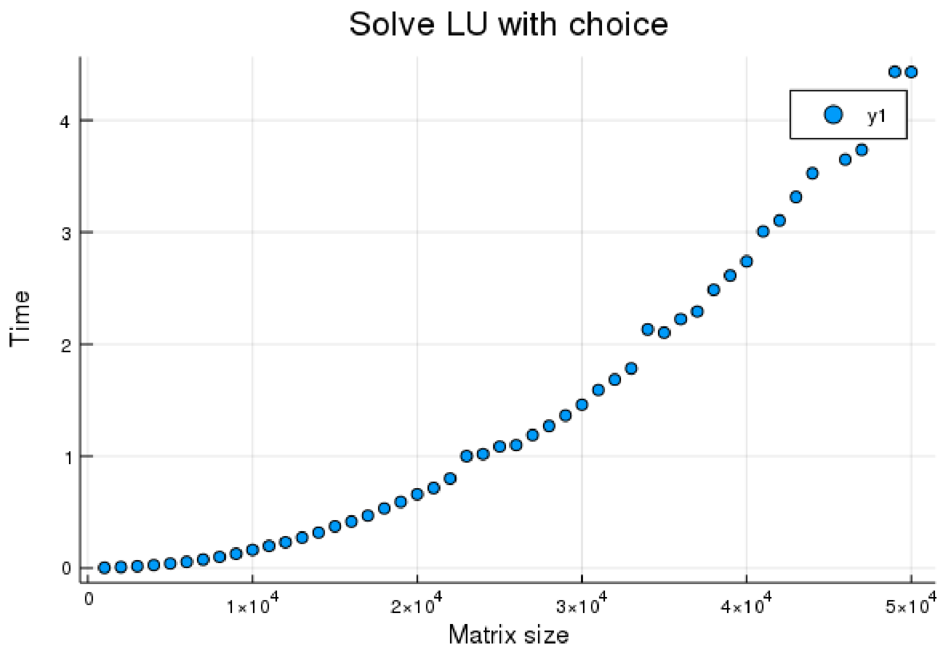
Tabela 1: Błędy względne obliczonych wektorów dla danej metody

3.1.1 Obserwacje

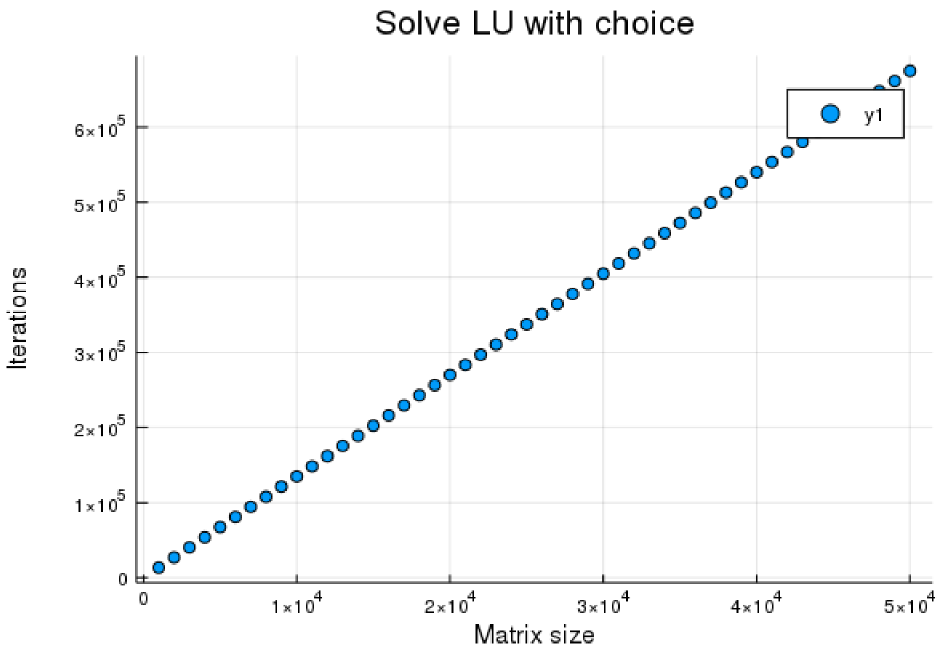
Błąd względny zdaje się być mniejszy w przypadkach gdy używamy wariantu z wyborem elementu głównego. Rozkład LU z wyborem radzi sobie znakomicie z każdym z trzech rozmiarów macierzy.

3.2 Złożoność czasowa oraz iteracyjna

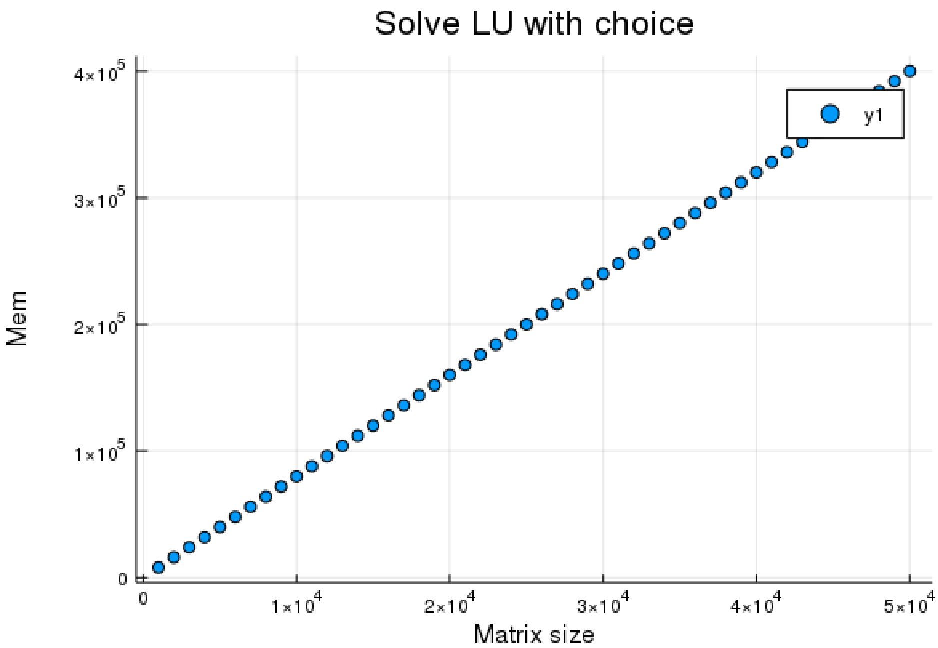
Otrzymanie wyniki złożoności czasowej odbiegają od liniowych (są bliskie kwadratowym) mimo że obliczona liczba iteracji jest liniowa. Jest to spowodowane założeniem, że *SparseMatrixCSC* posiada dostęp do elementów w czasie stałym - co niekoniecznie musi być prawdą.



Rysunek 1: Złożoność czasowa w sekundach dla rozwiązania układu równań za pomocą macierzy LU (generowanie LU wliczone w czas)



Rysunek 2: Ilośc iteracji dla rozwiązania układu równań za pomocą macierzy LU (generowanie LU wliczone)



Rysunek 3: Złożoność pamięciowa w bajtach dla rozwiązania układu równań za pomocą macierzy LU (generowanie LU wliczone)