Agnieszka Kurzajewska

Nr indeksu: 244994

Sprawozdanie

Lista 5

Sposób reprezentacji danych:

Z uwagi na problemy związane z przechowywaniem macierzy A jako tablicy dwuwymiarowej, została ona zaprezentowana za pomocą struktury *SparseMatrixCSC*. Jest to struktura zawarta w jednej z bibliotek języka Julia, która przechowuje dane macierzy w skompresowanym porządku kolumnowym.

Zadanie 1.

Metoda eliminacji Gaussa:

1. Redukcja macierzy górnej:
   - Sprowadzenie układu równań do układu równoważnego z macierzą trójkątną górną poprzez zerowanie pól macierzy pod polami z diagonali macierzy zaczynając od pierwszej kolumny i poniżej pierwszego wiersza
   - Zerowanie pola a_{k1} w pierwszej kolumnie i k-tym wierszu: dla każdego pola w wierszu k odejmujemy odpowiadające pole w wierszu pierwszym i mnożymy przez mnożnik a_{k1}/a_{11}
   - Powyższy punkt jest powtarzany kolejno dla 2, 3, 4, ... wiersza

2. Algorytm podstawiania wstecz:

Po kolei dla każdego wiersza od n do 1 wykonujemy:

$$x_k = \frac{b_k - \sum_{j=k+1}^{n} a_{kj}}{a_{kk}}$$

3. Modyfikacja algorytmu pod specyficzne dane wejściowe:

Macierz A ma budowę trójdiagonalną. Nie ma potrzeby zerowania wszystkich elementów w kolumnie. Dla $L-2$ pierwszych kolumn elementy nie będące zerem mogą znajdować się wyłącznie w L pierwszych wierszach (elementy bloku A_1). Dla pozostałych kolumn pola niezerowe mogą znajdywać się w pierwszych $2L$ rzędach (dwie ostatnie kolumny bloku B_2 oraz elementy bloku A_3) itd..

- Można wyznaczyć wzór na maksymalny indeks kolumny w danym rzędzie, w której znajduje się element niebędący zerem:

$$min\{n,\ k + l\}$$

- W trakcie odejmowania od siebie rzędów nie ma konieczności modyfikowania całego wiersza. Wystarczy odjąć wartości nie będące zerami, czyli te pola, których indeks kolumny jest mniejszy lub równy indeksowi obliczonemu we wzorze powyżej

`min{n, l+l*|(numer_komumny +1/l)|}`

4. Złożoność obliczeniowa liniowa:

- Ogólna: $O(n)$
- Zewnętrza pętla: $n-1$
- Środkowa pętla: $<2L$
- Wewnętrzna pętla $<L$
- Zewnętrzna pętla przebiegu wstecz: n
- Wewnętrzna pętla przebiegu wstecz: L

5. Rozwiązanie

- Dane wejściowe:
    - *A – podana w treści zadania macierz*
    - *b – wektor prawych stron*
    - *n – rozmiar macierzy A*
    - *L– rozmiar bloku macierzy A*
- Dane wyjściowe:
    - *x – wektor zawierający rozwiązanie równania Ax = b*

Zadanie 2.

Celem zadania było napisanie funkcji wyznaczającej rozkład LU macierzy A metodą eliminacji Gaussa uwzględniającą specyficzną postać macierzy dla dwóch wariantów:

a) bez wyboru elementu głównego
b) z częściowym wyborem elementu głównego

2. Rozkład LU:

Rozkład danej macierzy A na macierz dolnotrójkątną L oraz górnotrójkątną U, taki, że A=LU.

$$[\boldsymbol{A}] = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{12} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ l_{21} & 1 & 0 & 0 \\ l_{31} & l_{32} & 1 & 0 \\ l_{41} & l_{42} & l_{43} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{bmatrix}$$

Metoda Gaussa przekształca daną macierz A do macierzy trójkątnej górnej U. Macierz dolną L otrzymamy poprzez zapamiętanie mnożników, które zostały użyte do eliminacji kolejnych współczynników macierzy.

Rozkład **LU** dla niektórych przypadków umożliwia szybsze rozwiązywanie układów równań niż metoda eliminacji Gaussa. Etap, w którym wykonywana jest eliminacja Gaussa jest wykonywany tylko raz, a rozwiązanie układów równań zostaje sprowadzone do dwóch etapów:

$$\begin{cases} L * z = y \\ U * x = z \end{cases}$$

3. Złożoność:

Algorytm wyznaczania rozkładu **LU** ma złożoność podobną do algorytmu eliminacji Gaussa, bo w głównej mierze na nim opiera się stworzenie rozkładu i jest rzędu $O(n^3)$.

ponieważ macierze **L** i **U** są trójkątne to złożoność algorytmu jest rzędu $O(n^2)$.

4. Modyfikacja algorytmu pod specyficzne dane wejściowe:

Modyfikacja rozkładu LU przebiega prawie identycznie z modyfikacją obu warianów metody eliminacji Gaussa, poza wyjątkiem, że elementy a_{ij} nie są zerowane, ale zamieniane na mnożniki będące elementami macierzy L: a_{ij} / a_{jj}

Złożoność obliczeniowa tego algorytmu jest taka sama jak algorytmu eliminacji Gaussa: $O(n)$

5. Rozwiązanie

- Dane wejściowe:
  - *A – podana w treści zadania macierz*
  - *b – wektor prawych stron*
  - *n – rozmiar macierzy A*
  - *L– rozmiar bloku macierzy A*
- Dane wyjściowe:
  - *x – wektor zawierający rozwiązanie równania Ax = b*

Zadanie 3.

1. Opis

Program powinien rozwiązywać równanie *Ax=b* po wcześniejszym wyznaczeniu rozkładu LU.

Algorytm sprowadza się do użycia metody Gaussa (w miejsce zerowanych elementów zostają wstawione odpowiednie mnożniki).

2. Rozwiązanie

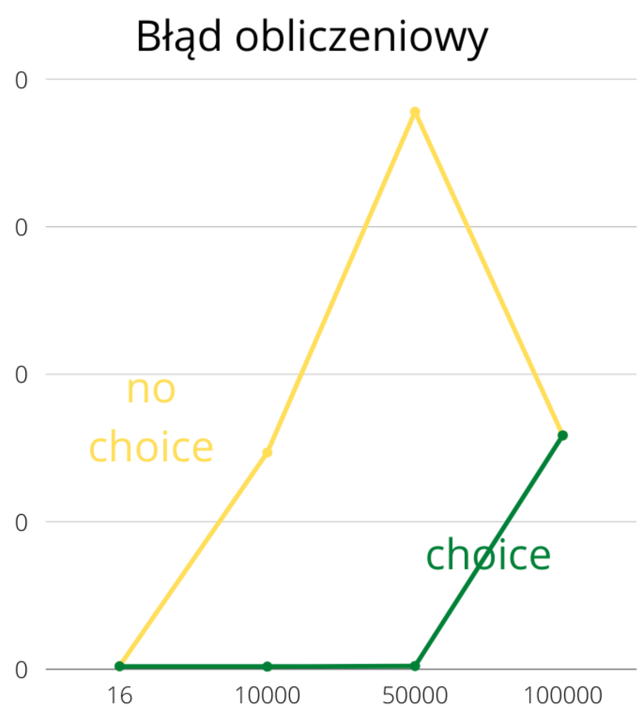
- Dane wejściowe:
  - *A – podana w treści zadania macierz*
  - *b – wektor prawych stron*
  - *n – rozmiar macierzy A*
  - *L– rozmiar bloku macierzy A*
  - *p - wektor permutacji*
- Dane wyjściowe:
  - *x – wektor zawierający rozwiązanie równania Ax = b*

Wyniki.

a) bez wyboru elementu głównego

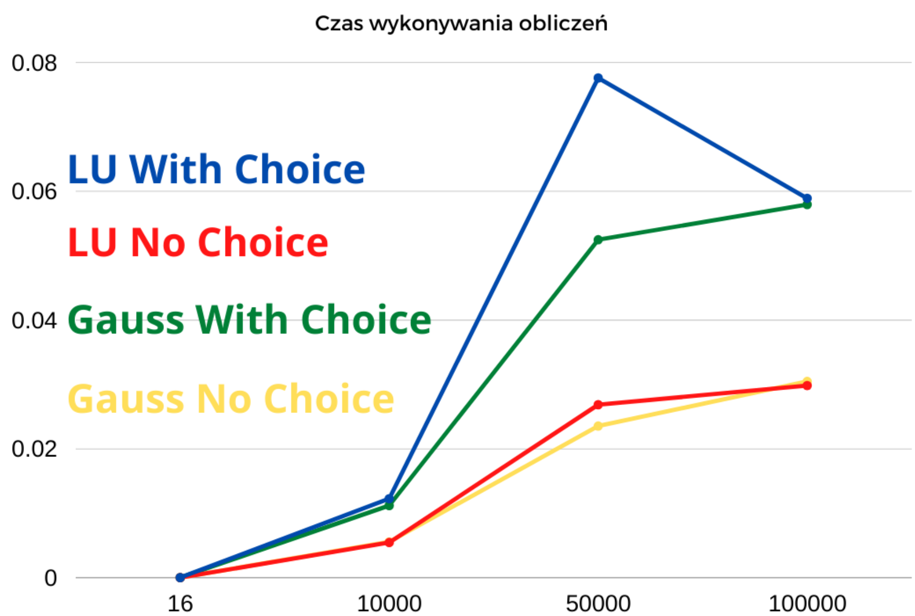
b) z wyborem elementu głównego

Rozmiar macierzy	Błąd względny obliczeń	
	a)	b)
16	5.45310538513047e-16	5.0037075531084e-16
10000	3.6738202190459565e-14	4.478865903518945e-16
50000	9.447293155699398e-14	5.427796231409846e-16
100000	3.963433526417144e-14	3.963433526417144e-14



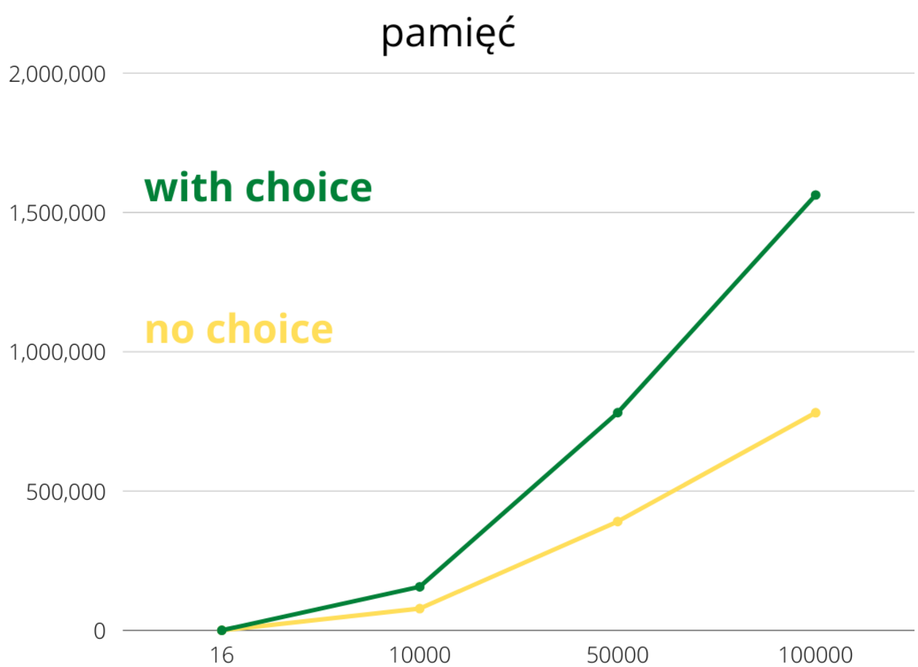
Wyniki dla metod eliminacji Gaussa i rozkładu LU są praktycznie takie same, dlatego w powyższej tabeli nie ma rozdziału na użytą metodę. ZAS

	Czas wykonywania obliczeń			
	Metoda eliminacji Gaussa		Metoda z rozkładem LU	
	a)	b)	a)	b)
16	0.000018s	0.000033s	0.000025s	0.000030s
10000	0.005574s	0.011216s	0.005482s	0.012282s
50000	0.023554s	0.052476s	0.026870s	0.077584s
100000	0.030475s	0.057945s	0.029841s	0.058908s



W powyższej tabeli zawierającej czas wykonywania obliczeń możemy zauważyć, że wraz ze wzrostem rozmiaru macierzy, wzrasta czas potrzebny na wykonanie wszystkich obliczeń. Rozmiar macierzy rośnie kolejno x625 i x5, a czas średnio x309 i x4. Czas wykonywania obliczen rośnie wolniej niż rozmiar macierzy.

Rozmiar macierzy	Zużycie pamięci dla metod obliczania równań liniowych	
	a)	b)
16	192 bytes	384 bytes
10000	78172 bytes	156344 bytes
50000	390672 bytes	781344 bytes
100000	781297 bytes	1562624 bytes



Tak jak w przypadku wyników błędów względnych, zużycie pamięci jest takie samo dla obu metod, różnią się natomiast warianty bez i z wyborem elementu głównego.

Widać znaczącą różnicę pomiędzy algorytmem, w którym nie dokonujemy częściowego wyboru elementu głównego, a algorytmem, w którym dokonujemy wyboru. Ilość pamięci potrzebnej do wykonania algorytmu z częściowym wyborem jest ponad dwukrotnie większa.

Wnioski.

Tabela zawierająca obliczone błędy względne obliczeń prezentuje, że obie zaprogramowane metody są prawidłowe, gdyż błędy są bardzo małej wielkości. Analizując dokładnie tabelę można zauważyć, że dla algorytmów z częściowym wyborem elementu głównego błąd względny jest mniejszy, czyli dokładność obliczeniowa jest większa. Wynika to z tego, że algorytm ten cechuje się lepszą poprawnością numeryczną.

Po przeanalizowaniu wyników dotyczących czasu wykonywania poszczególnych programów można zauważyć, że algorytmy, w których dokonujemy wyboru elementu głównego są znacznie wolniejsze od algorytmów bez wyboru. Niestety jak zostało wcześniej wspomniane w niektórych przypadkach niemożliwe jest korzystanie z algorytmu bez wyboru elementu głównego. Takim przypadkiem jest rozwiązywanie układu równań liniowych z macierzą diagonalną, której jakikolwiek element na diagonali jest zerowy.

Po dostosowaniu obu metod do specyficznej budowy macierzy udało się zredukować złożoność obliczeniową do liniowej. Szczególnie jest to widoczne na wykresie obrazującym ilość pamięci potrzebną do obliczenia poszczególnych algorytmów. W przypadku algorytmu, w którym nie dokonujemy wyboru elementu głównego udało się osiągnąć złożoność pamięciową rzędu $O(n)$. W przypadku, gdy stosowany jest algorytm z częściowym wyborem elementu głównego złożoność pamięciowa jest już nieco gorsza i wynosi $O(n^2)$. Wpływają na to dodatkowe permutacje rzędów przy wypełnianiu macierzy.