Sprawozdanie nr 5

Obliczenia naukowe

Tomasz Krent nr indeksu 236595

Opis problemu: Rozwiązanie problemu, które sprowadza się do obliczenia układu równań liniowych $Ax = b$ dla danej macierzy $A \in \mathbb{R}^{\mathrm{n}\times\mathrm{n}}$ i wektora prawych stron $b \in \mathbb{R}^{\mathrm{n}}$. Macierz A jest macierzą rzadką tj. mającą dużą elementów zerowych, i blokową o następującej strukturze:

$$A = \begin{pmatrix} A_1 & C_1 & 0 & 0 & 0 & \dots & 0 \\ B_2 & A_2 & C_2 & 0 & 0 & \cdots & 0 \\ 0 & B_3 & A_3 & C_3 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & B_1 & A_{v-2} & C_{v-2} & 0 \\ 0 & \dots & 0 & 0 & B_{v-1} & A_{v-1} & C_{v-1} \\ 0 & \dots & 0 & 0 & 0 & B_v & A_v \end{pmatrix}$$

$v = n/l,$, zakładając, że n jest podzielne przez $l,$ gdzie l jest rozmiarem wszystkich kwadratowych macierzy wewnętrznych (bloków): A_k, B_k, C_k. Mianowicie $A_k \in \mathbb{R}^{\mathrm{l}\times\mathrm{l}}, k = 1, \dots, v$ jest macierzą gęstą 0 jest kwadratową macierzą zerową stopnia $l,$ macierz $B_k \in \mathbb{R}^{\mathrm{l}\times\mathrm{l}}, k = 2, \dots, v$ jest następującej postaci:

$$B_k = \begin{pmatrix} 0 & \dots & 0 & b_1^k \\ 0 & \cdots & 0 & b_2^k \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & b_l^k \end{pmatrix}$$

B_k ma tylko ostatnią kolumnę niezerową. Natomiast $C_k \in \mathbb{R}^{\mathrm{l}\times\mathrm{l}}, k = 1, \dots, v-1$ jest macierz diagonalną:

$$C_k = \begin{pmatrix} c_1^k & 0 & 0 & \dots & 0 \\ 0 & c_2^k & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & c_{l-1}^k & 0 \\ 0 & \dots & 0 & 0 & c_l^k \end{pmatrix}$$

Problemem efektywnego zadania problemu są wymagania czasowe i pamięciowe wynikające z dużego rozmiaru danej macierzy. Co wyklucza pamiętanie macierzy A jako tablicy $\mathrm{n} \times \mathrm{n}$. Także trzeba utworzyć specjalną strukturę pamiętającą tylko elementy niezerowe macierzy oraz adaptacja algorytmów, tak aby uwzględniały postać macierzy A czyli jej rzadkość i regularność występowania elementów niezerowych oraz zerowych.

Rozwiązanie:

(zadania zostały rozwiązane w języku Julia; wszystkie funkcje i zmienne opisane poniżej znajdują się w plikach dołączonych do sprawozdania)

1. Funkcja rozwiązująca układ $Ax = b$ metodą eliminacji Gaussa uwzględniająca specyficzną postać macierzy A składa się z dwóch części:
   - Pierwsza część składa się z funkcji $readA$ oraz $readB$ i $writeB$.
     - Funkcja $readA$ na wejściu przyjmuje nazwę pliku tekstowego z niezerowymi argumentami macierzy A. Na początku działania funkcja czyta wszystkie argumenty niezerowe i wpisuje je do tabeli m. Następnie na podstawie wartości przeczytanych umieszcza dane w specjalnie przygotowanej tablicy dwuwymiarowej tablicy Wx wielkości $\mathrm{n} \times (2 * \mathrm{l} + 2)$. Wx jest tablicą która posiada kolejne wartości z każdego rzędu macierzy A zaczynając od pierwszej niezerowej wartości w danym rzędzie. Wszystkie niezerowe argumenty macierzy A są zawarte w danej tablicy. Funkcja $readA$ zwraca w wyniku tablicę Wx, oraz liczby n oraz l, przeczytane z danego pliku tekstowego.

     - Funkcja $readB$ na wejściu przyjmuje nazwę pliku tekstowego z wszystkimi wartościami wektora prawych stron b i przypisuje wartości do specjalnej jednowymiarowej tablicy B o długości n. Funkcja zwraca w wyniku tablicę B.
     - Funkcja $writeB$ na wejściu przyjmuje specjalną tablicę Wx obliczoną przez funkcję $readA$, wartości n i l oraz nazwę nowego pliku tekstowego. Funkcja ta oblicza wektor prawych stron b na podstawie wektora x, tj $b = Ax$, gdzie $x = (1, \ldots, 1)^{\mathrm{T}}$ uwzględniając rzadką postać macierzy A. Następnie dany wektor zapisuje do pliku tekstowego o odpowiedniej nazwie.

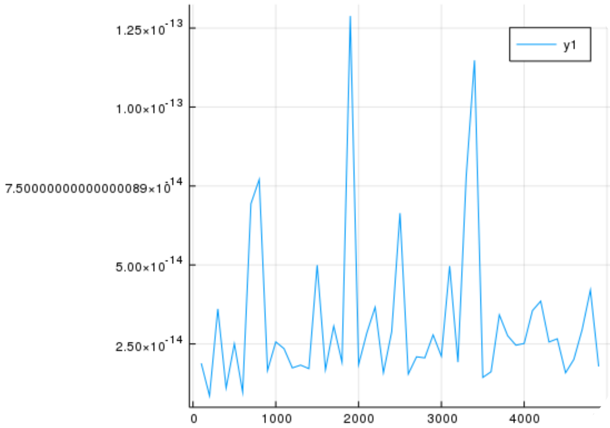
     W zależności od tego czy posiadamy plik z wektorem prawych stron b wykorzystujemy funkcję $writeB$.
   - Druga część składa się z funkcji $z1a$ oraz $z1b$:
     - Funkcja $z1a$ na wejściu przyjmuje specjalną tablicę Wx z niezerowymi argumentami macierzy A, tablicę B zawierającą wektor prawych stron b, wartości n i l oraz nazwę nowego pliku tekstowego. Następnie odpowiednio modyfikując algorytm Gaussa do rozwiązywania układu równań funkcja oblicza macierz trójkątną, która zostaje odpowiednio zapisana w specjalnej tablicy Wx. Modyfikacje tego algorytmu wykorzystują rzadkość i regularność macierzy A tak żeby ograniczyć dodatkowe obliczenia, omijając zerowe argumenty macierzy A. Wykorzystując tablicę Wx, funkcja w czasie $O(n * l^2)$ potrafi obliczyć macierz trójkątną i zapisać w tej samej tablicy Wx odpowiednie dane niezerowe. Wykorzystując tą tablicę funkcja w dalszym działaniu oblicza rozwiązanie czyli wektor x, w czasie $O(n * l)$, który zapisuje do pliku tekstowego i obliczany jest błąd względny względem wektora $(1, \ldots, 1)^{\mathrm{T}}$.

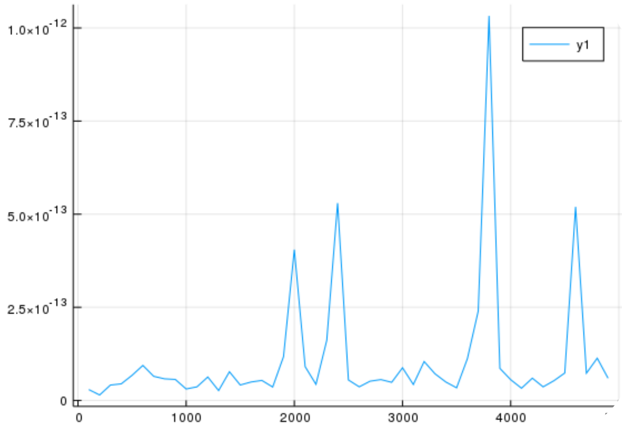
- Funkcja $z1b$ przyjmuje te same dane co funkcja $z1a$ i działa praktycznie tak samo jak ta funkcja tylko w funkcji $z1b$ wybierany jest częściowo element główny oraz odpowiednio zamieniane są rzędy tablicy Wx oraz B, w porównaniu z funkcją $z1a$, w której nie ma wyboru elementu głównego. Wybór elementu następuje w czasie $O(l)$.

W zależności od tego czy chcemy rozwiązać układ równań bez wyboru elementu głównego czy z jego częściowym wyborem używamy odpowiedniej funkcji.

- Do testowania rozwiązania układu równań zostały użyte funkcje $zadanie1a$ (funkcja bez wyboru elementu głównego) i $zadanie1b$ (funkcja z częściowym wyborem elementu głównego). Funkcje te wykorzystują funkcję $blockmat$ i $matcond$ napisane przez profesora Pawła Zielińskiego służące do generowania macierzy A. Następnie wykorzystując funkcję opisane powyżej generowane są pliki wynikowe z wektorem rozwiązań x oraz błędem względnym tych rozwiązań w porównaniu do wektora $(1, \ldots, 1)^{\mathrm{T}}$. Test zostały przeprowadzone dla $n = 100, 200, \ldots, 4900$ i $l = 5, 10, 20$.

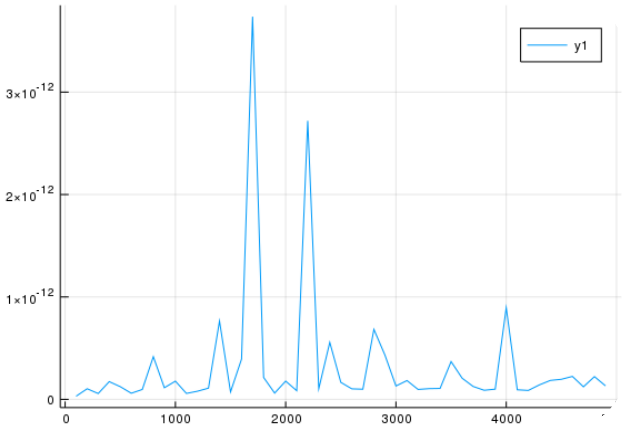
  Wyniki testów:
  //do generowania wykresów funkcji została użyta metoda plot z modułu plots w języku Julia

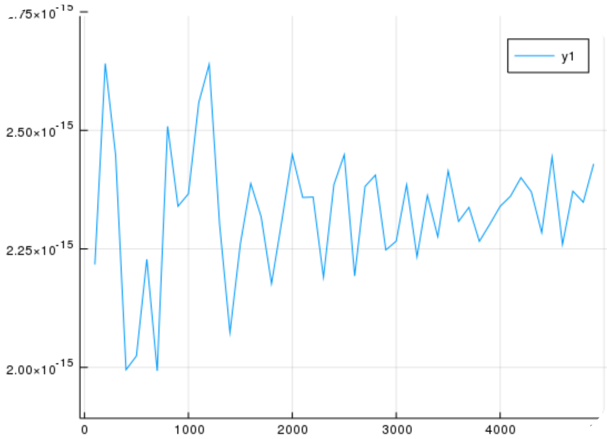




Wykres błędów względnych dla:
$n = 100,200, \ldots, 4900, l = 5, funkcja\ zadanie1a$

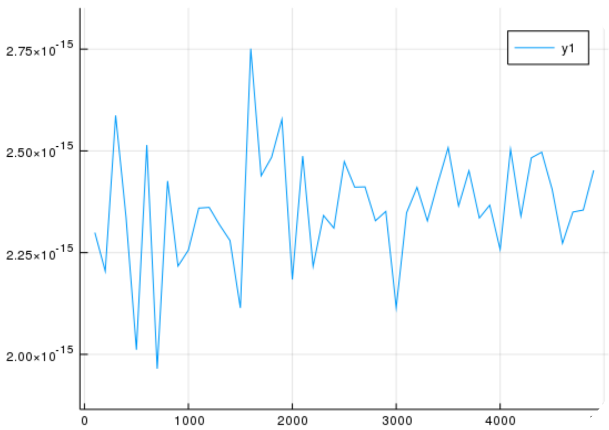
Wykres błędów względnych dla:
$n = 100,200, \ldots, 4900, l = 10, funkcja\ zadanie1a$

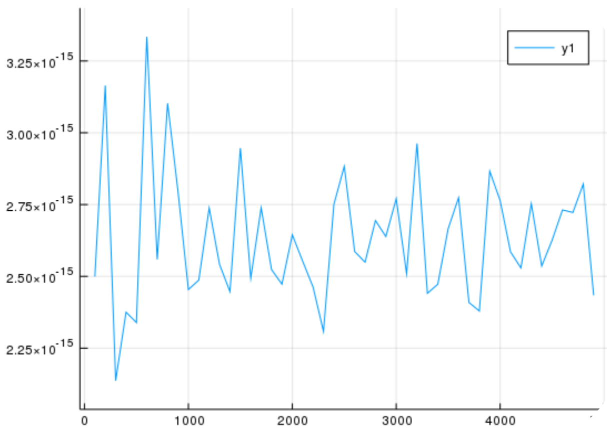




Wykres błędów względnych dla:
$n = 100,200, \ldots, 4900, l = 20, funkcja\ zadanie1a$

Wykres błędów względnych dla:
$n = 100,200, \ldots, 4900, l = 5, funkcja\ zadanie1b$





Wykres błędów względnych dla:
$n = 100,200, \ldots, 4900, l = 10, funkcja\ zadanie1b$

Wykres błędów względnych dla:
$n = 100,200, \ldots, 4900, l = 20, funkcja\ zadanie1b$

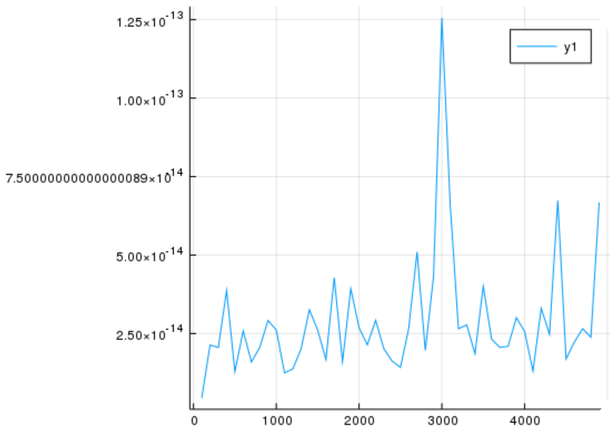
Interpretacja wyników testów: Wektory zostały obliczone w sposób prawidłowy. Błędy względne dla metody bez wyboru elementu głównego są rzędu od 10^{-14} do 10^{-12}. Błędy względne dla metody z częściowym wyborem elementu głównego są wynoszą około 10^{-15}. Wynika z tego że metody użyte do obliczenia układu równań zostały skonstruowane poprawnie i wyniki są bliskie wektorowi rzeczywistemu. Dodatkowo wiemy, że wyniki dla funkcji z częściowym elementem wyboru głównego są dokładniejsze niż te bez wyboru elementu głównego.

2. Funkcja wyznaczająca rozkład LU macierzy A metodą eliminacji Gaussa uwzględniająca jej specyficzną postać i na jej podstawie rozwiązująca układ równań $Ax = b$ składa się z trzech części:
    - Z części czytającej, takiej samej jak w przypadku rozwiązywania układu $Ax = b$ z metodą eliminacji Gaussa;
    - Z drugiej części składającej się z funkcji z2a oraz z2b:
        - Funkcja $z1a$ na wejściu przyjmuje specjalną tablicę Wx z niezerowymi argumentami macierzy A, tablicę B zawierającą wektor prawych stron b oraz wartości n i l. Następnie wykorzystując odpowiednio zmodyfikowany algorytm Gaussa do obliczenia rozkład LU w tablicy Wx otrzymujemy macierz trójkątną górną a w nowo utworzonej dwuwymiarowej tablicy $Wx2$ wielkości $\mathrm{n} \times (\mathrm{l} + 1)$ mamy zapisane niezerowe wartości macierzy trójkątnej dolnej. Modyfikacje tego algorytmu są podobne do modyfikacji wykorzystanych w zadaniu z rozwiązaniem układu równań, tylko dodatkowo zapisywane są odpowiednie wartości do nowo utworzonej tablicy $Wx2$. Funkcja w wyniku zwraca tablice Wx oraz Wy oraz tablicę B.
        - Funkcja $z2b$ przyjmuje te same dane co funkcja $z2a$ i działa praktycznie tak samo jak ta funkcja tylko w funkcji $z2b$ wybierany jest częściowo element główny oraz odpowiednio zamieniane są rzędy tablicy Wz, $Wz2$ oraz B, w porównaniu z funkcją $z1a$, w którym nie ma wyboru elementu głównego. Wybór elementu następuje w czasie $O(l)$. Funkcja ta w wyniku zwraca te sam dane co funkcja $z2a$.
    - Trzecia część składa się z jednej funkcji $z3$
        - Funkcja z3 przyjmuje na wejściu dane wynikowe z drugiej części czyli tablice Wx $Wx2$ i B, wartości n , l oraz nazwę nowego pliku tekstowego. Funkcja tworzy nowe tablice jednowymiarowe X i Y długości n. Następnie wylicza wektor y, z działania $L * y = b$, gdzie L jest macierzą dolną trójkątną, która jest uwzględniona w tablicy $Wx2$ i B jest wektorem prawych stron. Po obliczeniu tego wektora y obliczamy wektor x z działania $U * x = y$, gdzie U jest macierzą górną trójkątną , uwzględnioną w odpowiedni sposób w tablicy Wx. Działanie algorytmów jest podobne i polega na obliczeniu odpowiednich wartości wykorzystując specyficzność i regularność danych macierzy A i obliczonej na jej podstawie tablic Wx i $Wx2$. Wektor y obliczamy z

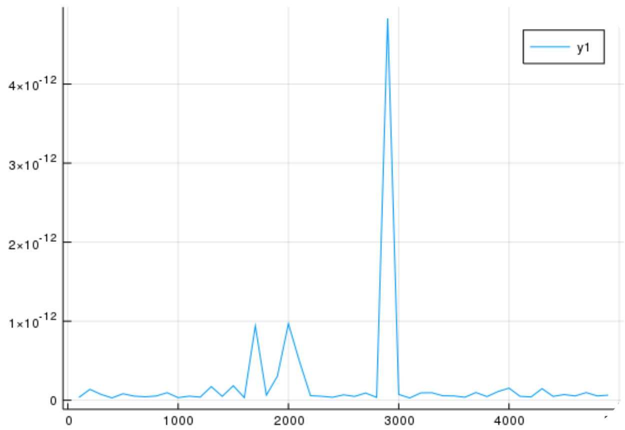
„góry na dół”, a później wektor x „z dołu do góry” znając odpowiednie zależności występujące w tablicach. Czas działania tych algorytmów to $O(n * l)$. Po zakończeniu działania programu wektor x zapisywany jest w pliku tekstowym i obliczany jest błąd względny względem wektora $(1, \dots, 1)^T$.

- Do testowania wyznaczenia rozkładu LU macierzy A i na podstawie tego rozkładu rozwiązania układu równań zostały użyte funkcje $zadanie2a$ (funkcja bez wyboru elementu głównego) i $zadanie2b$ (funkcja z częściowym wyborem elementu głównego). Funkcje te wykorzystują funkcję $blockmat$ i $matcond$ napisane przez profesora Pawła Zielińskiego służące do generowania macierzy A. Następnie wykorzystując funkcję opisane powyżej generowane są pliki wynikowe z wektorem rozwiązań x oraz błędem względnym tych rozwiązań w porównaniu do wektora $(1, \dots, 1)^T$. Test zostały przeprowadzone dla $n = 100, 200, \dots, 4900$ i $l = 5, 10, 20$.

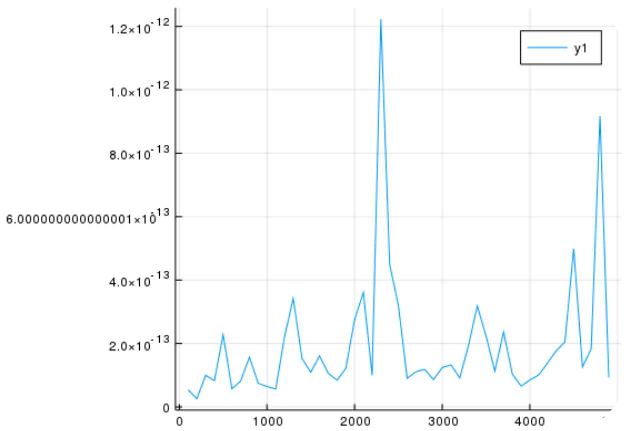
  Wyniki testów:



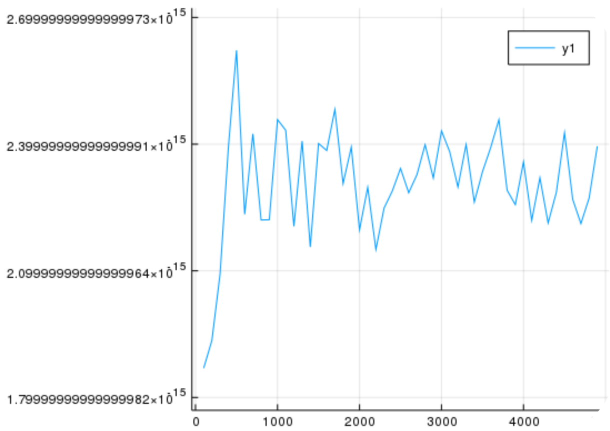
Wykres błęów względnych dla:
$n = 100{,}200, \ldots, 4900, l = 5, funkcja\ zadanie1a$



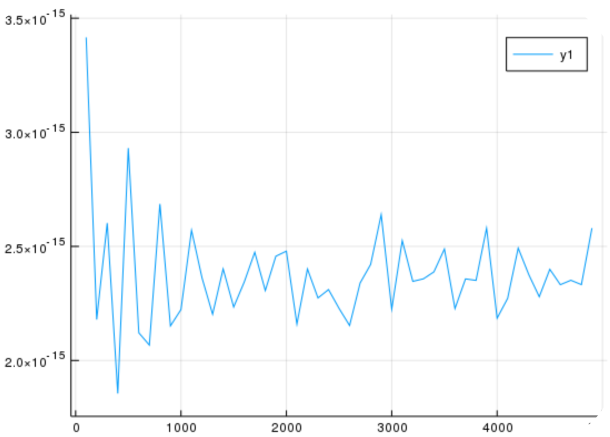
Wykres błęów względnych dla:
$n = 100{,}200, \ldots, 4900, l = 10, funkcja\ zadanie1a$



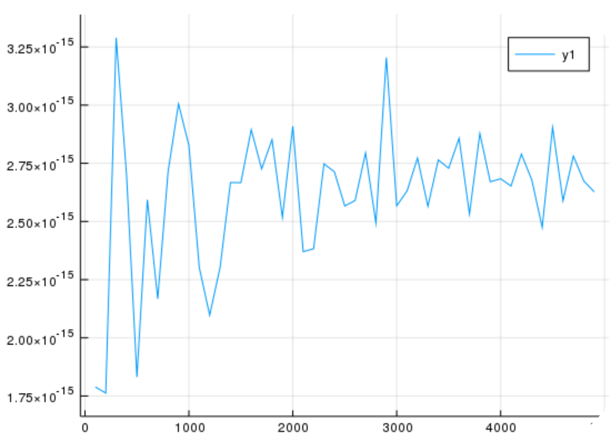
Wykres błęów względnych dla:
$n = 100{,}200, \ldots, 4900, l = 20, funkcja\ zadanie1aa$



Wykres błęów względnych dla:
$n = 100{,}200, \ldots, 4900, l = 5, funkcja\ zadanie1b$



Wykres błęów względnych dla:
$n = 100{,}200, \ldots, 4900, l = 10, funkcja\ zadanie1b$



Wykres błęów względnych dla:
$n = 100{,}200, \ldots, 4900, l = 20, funkcja\ zadanie1b$

Interpretacja wyników: Wektory również w tym przypadku zostały obliczone w sposób prawidłowy. Błędy względne dla metody bez wyboru elementu głównego są rzędu od 10^{-14} do 10^{-12}. Błędy względne dla metody z częściowym wyborem elementu głównego są wynoszą około 10^{-15}. Wynika z tego że metody użyte do obliczenia układu równań zostały skonstruowane poprawnie i wyniki są bliskie wektorowi rzeczywistemu, a z tego wynika, że prawidłowo został wyznaczony rozkład LU da podstawie którego zostały obliczone wyniki. Dodatkowo wiemy, że wyniki dla funkcji z częściowym elementem wyboru głównego są dokładniejsze niż te bez wyboru elementu głównego.

Wnioski:

Wykorzystując regularność i rzadkość macierzy A udało się skonstruować tablice wielkości $\mathrm{n} \times (2 * \mathrm{l} + 2)$, która jest znacznie mniejsza niż tablica $\mathrm{n} \times \mathrm{n}$, która miałaby przetrzymywać całą macierz A. Macierze trójkątne pochodzące z rozkładu LU również są przechowywane w taki sposób aby nie pamiętać całej macierzy. Również udało się skrócić czas działania programu w porównaniu do metody eliminacji Gaussa z $O(n^3)$ do $O(n * l * l)$ (jeśli l jest stałą to czas działania programu jest rzędu $O(n)$). Wyniki testów wykazały, że modyfikacje nie spowodowały znacznych zmian wynikach w porównaniu z tradycyjnymi algorytmami Gaussa. Z tego wynika, że udało się zmniejszyć wymagania czasowe i pamięciowe, które wynikały z bardzo dużego rozmiaru macierzy n.