

Wrocław, 07.01.2022

Obliczenia naukowe

Lista nr 5

Małgorzata Orłowska

1. Opis problemu

Celem zadania było rozwiązanie układu równań liniowych $Ax = b$, dla danej macierzy współczynników $A \in \mathbb{R}^{n \times n}$ i wektora prawych stron $b \in \mathbb{R}^n$, $n \geq 4$. Macierz A jest tzw. macierzą rzadką, tj. mającą dużą elementów zerowych, i blokową o następującej strukturze:

$$\begin{array}{ccccccc} A_1 & C_1 & 0 & 0 & 0 & \cdots & 0 \\ B_2 & A_2 & C_2 & 0 & 0 & \cdots & 0 \\ 0 & B_3 & A_3 & C_3 & 0 & \cdots & 0 \\ \vdots & & & & & & \\ 0 & \cdots & 0 & B_{v-2} & A_{v-2} & C_{v-2} & 0 \\ 0 & \cdots & 0 & 0 & B_{v-1} & A_{v-1} & C_{v-1} \\ 0 & \cdots & 0 & 0 & 0 & B_v & A_v \end{array}$$

$v = n/\ell$, zakładając, że n jest podzielne przez ℓ , gdzie $\ell \geq 2$ jest rozmiarem wszystkich kwadratowych macierzy wewnętrznych (bloków): A_k, B_k i C_k . Mianowicie, $A_k \in \mathbb{R}^{\ell \times \ell}$, $k = 1, \dots, v$ jest macierzą gęstą, 0 jest kwadratową macierzą zerową stopnia ℓ , macierz $B_k \in \mathbb{R}^{\ell \times \ell}$, $k = 2, \dots, v$ jest macierzą posiadającą tylko dwie ostatnie kolumny niezerowe. Natomiast $C_k \in \mathbb{R}^{\ell \times \ell}$, $k = 1, \dots, v-1$ jest macierzą diagonalną.

W opracowaniu podano efektywny sposób rozwiązania powyższego problemu, uwzględniający specyficzną strukturę macierzy A , tj. jej rzadkość, regularność występowania elementów zerowych i niezerowych. Pozwala on na zredukowanie złożoności z $O(n^3)$ do $O(n)$ w stosunku do standardowych metod rozwiązywania takich układów równań.

2. Rozwiązanie – zaimplementowane algorytmy

Problem został rozwiązany dzięki zaadoptowaniu znanych metod – metody eliminacji Gaussa oraz sposobu wyznaczania rozkładu LU dla macierzy, do sytuacji w zadaniu. Poniżej znajduje się opis implementacji tych metod w dwóch wariantach, w wersji podstawowej oraz z częściowym wyborem elementu głównego, wraz z zastosowanymi optymalizacjami.

2.1 Metoda eliminacji Gaussa

Składa się z dwóch faz. W pierwszej przekształcamy macierz kwadratową w trójkątną, a następnie rozwiązujemy uzyskany, uproszczony układ równań.

2.1.1 Wersja podstawowa eliminacji Gaussa

Idea

W tej metodzie, aby uzyskać macierz trójkątną zerujemy elementy pod przekątną w kolejnych kolumnach, przy pomocy działań elementarnych na macierzach.

Dla $Ax = b$, $A \in \mathbb{R}^{n \times n}$, $x \in \mathbb{R}^n$, $b \in \mathbb{R}^n$, $\det(A) \neq 0$ musimy wykonać $n-1$ kroków. W k -tym kroku eliminujemy zmienną x_k (w k -tej kolumnie) z równań od $k+1$ do n . W tym celu mnożymy k -te równanie przez

$$l_{ik} = a_{ik}^{(k)} / a_{kk}^{(k)} \quad \text{dla } i = k + 1, \dots, n,$$

a następnie odejmujemy od pozostałych.

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} \quad \text{dla } i \leq k$$

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - (a_{ik}^{(k)} / a_{kk}^{(k)}) a_{kj}^{(k)} \quad \text{dla } i \geq k + 1, j \geq k + 1$$

$$a_{ij}^{(k+1)} = 0 \quad \text{dla } i \geq k + 1, j \leq k$$

Warto zauważyć, że ta wersja sprawdza się tylko dla macierzy nieosobliwych, których elementy na przekątnej $a_{kk} \neq 0$, $k = 1, \dots, n$. W przypadku a_{kk} , $k = 1, \dots, n$ bliskich zero jest problem z wyznaczaniem mnożników głównych macierzy ($l_{ik} = a_{ik}^{(k)} / a_{kk}^{(k)}$) mamy wówczas dzielenie przez zero.

W drugiej fazie algorytmu rozwiązujemy przekształcony układ równań. W tym celu wyznaczamy

$$x_n = b_n / a_{nn}$$

Dalej wyznaczamy x_k dla $k = n-1, \dots, 1$

$$x_k = (b_k - \sum_{j=k+1}^n a_{kj} x_j) / a_{kk}$$

Optymalizacje

Standardowa wersja eliminacji Gaussa ma złożoność $O(n^3)$, ponieważ dla k -tego wiersza (z $n-1$ wierszy), wykonujemy $n-k$ odejmowań wierszy, czyli uaktualniamy $n+1$ wartości w każdym wierszu.

Możemy zmniejszyć złożoność z $O(n^3)$ do $O(n)$ specyficznej macierzy danej w zadaniu, korzystając z regularności występowania elementów zerowych. Zastosowana optymalizacja polega na ograniczeniu ilości elementów, które należy uaktualnić w macierzy w trakcie dokonywania eliminacji. W tym celu dla każdej kolumny wystarczy wyznaczyć pierwszy niezerowy element pod przekątną (funkcja `findNoZeroRow()`), a dla każdego wiersza wyznaczyć ostatnią kolumnę, w której element nie jest zerem (funkcja `findNoZeroColumn()`). Dla wielkości bloku l , pod przekątną w danej kolumnie jest od 2 do $l - 1$ elementów niezerowych, a w trakcie odejmowań wystarczy uaktualnić l elementów (w najniższym położonym bloku A_v od $l - 1$ do 0 elementów). Zatem złożoność to $O((n - 1)(l - 1)l)$, czyli $O(n)$.

Łatwo zauważyć, że złożoność rozwiązywania macierzy trójkątnej to $O(n^2)$, ponieważ aby wyznaczyć wartość każdego z x (jest ich n), to dla x_k -tego obliczamy sumę $n - (k + 1)$ elementów.

Postępując w sposób opisany wyżej, w każdym wierszu możemy ograniczyć sumowanie do maksymalnie l elementów i w ten zmniejszyć złożoność do $O(n)$.

Zatem łączną złożoność dwóch faz algorytmu możemy ograniczyć do $O(n)$.

2.1.2 Eliminacja Gaussa z częściowym wyborem elementu głównego

Idea

Algorytm eliminacji Gaussa w wersji podstawowej przedstawiony powyżej jest często zawodny dla prostych układów równań. Dzieje się tak, gdy elementy przekątniowe macierzy a_{kk} (zwane elementami głównymi – używamy ich do wyzerowania znajdujących się pod nimi elementów w macierzy) są zbyt małe co do wartości bezwzględnej w porównaniu z innymi elementami wiersza, w którym się znajdują. Powstają wtedy błędy obliczeniowe związane z utratą miejsc po przecinku w trakcie odejmowania.

Aby zapewnić odpowiednią własność macierzy pozwalającą na wyeliminowanie tego błędu należy w k -tym kroku znaleźć taki element, że

$$|a_{pk}^{(k)}| = \max_{k \leq i \leq n} |a_{ik}^{(k)}|$$

i przestawieniu w macierzy $A^{(k)}$ wiersza p -tego z k -tym oraz elementu p -tego z k -tym w wektorze $b^{(k)}$.

Optymalizacje

Łatwo zauważyć, że złożoność poszukiwania elementu głównego, w każdym z $n - 1$ kroków eliminacji Gaussa jest $O(n)$, zatem złożoność eliminacji z częściowym wyborem pozostaje $O(n^3)$.

W tym przypadku stosujemy podobne sposoby optymalizacji jak wcześniej. Możemy ograniczyć liczbę porównań w trakcie szukania elementu głównego do $l - 1$ (tyle jest co najwyżej elementów niezerowych w danej kolumnie pod przekątną) i tym samym zmniejszyć złożoność tej procedury do liczby stałej. Należy zauważyć, że z powodu przestawienia wierszy zmienia się ograniczenie kolumn, od których odejmujemy w trakcie eliminacji. Dosłowne przestawienie wierszy byłoby zbyt kosztowne, dlatego kolejne permutacje macierzy zapisujemy w wektorze p . W k -tym kroku element główny może zostać wybrany maksymalnie z $k + l + 1$ wiersza, zatem dla k -tego wiersza ostatni niezerowy element może znajdować się w $k + 2l + 1$ kolumnie. Zostajemy jednak dalej przy stałej liczbie operacji

i złożoności tej fazy równej $O(n)$. To ograniczenie zwiększa się także w trakcie rozwiązywania macierzy trójkątnej i wyznaczania poszczególnych x -ów.

Zatem udało nam się zachować złożoność $O(n)$ rozwiązywania problemu danego w zadaniu.

2.2 Rozkład LU macierzy

Eliminacja Gaussa jest równoważna rozkładowi macierzy A na iloczyn macierzy trójkątnej dolnej L i trójkątnej górnej U .

$$A = LU$$

Znając ten rozkład, układ równań $Ax = b$ możemy sprowadzić do rozwiązania dwóch układów:

$$Ly = b \text{ względem } y$$

$$Ux = y \text{ względem } x$$

Wyznaczenie najpierw rozkładu LU , a następnie rozwiązywanie układu jest korzystne w przypadku, gdy wielokrotnie wyznaczamy x dla różnych wektorów prawych stron b .

2.2.1 Wersja podstawowa

Idea

Wynikiem eliminacji Gaussa jest macierz trójkątna górna U. Stosując niewielką modyfikację możemy również uzyskać macierz L. Oznaczmy przez $L^{(k)}$ macierz przekształcenia A w k-tym kroku. $L^{(k)}$ ma postać:

1

...

1

$-l_{k+1,k}$ 1

$-l_{k+2,k}$

...

$l_{n,k}$ 1, gdzie $l_{ik} = a_{ik}^{(k)} / a_{kk}^{(k)}$ dla $i = k + 1, \dots, n$

Macierz wynikową U uzyskujemy w n-1 krokach:

$$U = A^{(n)} = L^{(n-1)} \dots L^{(2)} L^{(1)} A$$

$$A = L^{(1)-1} L^{(2)-1} \dots L^{(n-1)-1} U$$

$$L = L^{(1)-1} L^{(2)-1} \dots L^{(n-1)-1}$$

Zatem macierz L jest postaci:

1

l_{21} 1

l_{31} l_{32} 1

...

l_{n1} l_{n2} ... $l_{n,n-1}$ 1

W algorytmie Gaussa należy zatem wprowadzić następujące zmiany:

- wartości pod przekątną nie zastępujemy zerami tylko mnożnikami l_{ik} , które służyły do ich wyzerowania, dzięki temu oszczędzimy pamięć, a efektem algorytmu będzie macierz LU, jedynek na przekątnej L nie musimy pamiętać

- nie przekształcamy wektora b, będzie zmieniany dopiero w trakcie wyznaczania rozwiązań układu

Aby rozwiązać układ posiadając wyznaczony wcześniej rozkład LU, w dwóch pętlach wyznaczamy kolejno y z $Ly = b$, a potem x z $Ux = y$. Każdy z układów rozwiązujemy analogicznie do opisanej wcześniej metody eliminacji Gaussa.

Optymalizacje

Z powodu niewielkich zmian wprowadzonych do eliminacji Gaussa w celu wyznaczenia LU, algorytm ma tę samą złożoność $O(n^3)$. Wykonując te same optymalizacje redukujemy ją do $O(n)$.

Dodatkowym zabiegiem oszczędzającym pamięć jest nadpisywanie wektora b rozwiązaniami pierwszego układu. Struktura LU jest taka sama jak A , więc rozwiązywanie obu układów równań upraszczamy w ten sam sposób jak algorytm eliminacji Gaussa, zachowując złożoność $O(n)$.

2.2.2 Rozkład LU z częściowym wyborem elementu głównego

Idea

Tak jak we wcześniej opisanej eliminacji Gaussa, częściowy wybór elementu głównego pozwala na zapewnienie lepszych własności numerycznych macierzy A i wpływa na zwiększenie dokładności obliczeń. Dodajemy niewielkie zmiany do algorytmu analogicznie jak w przypadku eliminacji Gaussa. Należy teraz dodatkowo zwracać wektor permutacji wierszy, gdyż będzie potrzebny w kolejnych obliczeniach.

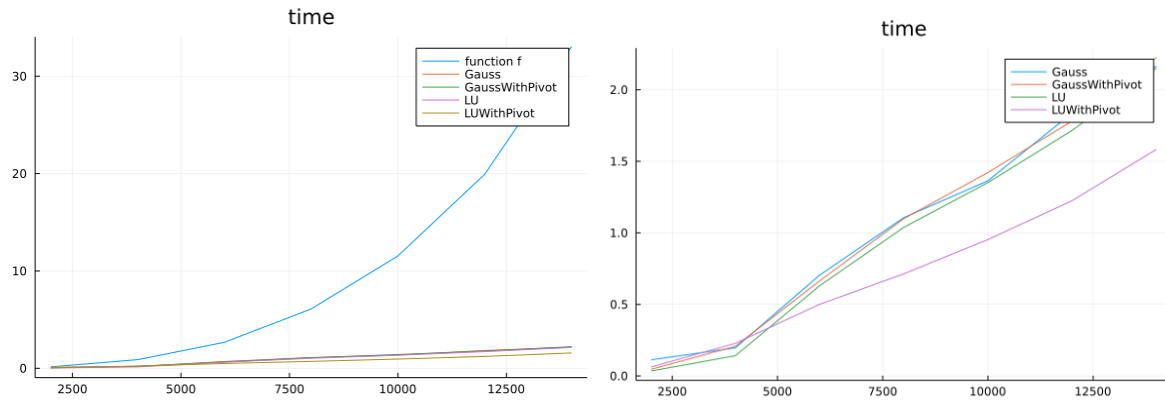
Optymalizacje

Stosujemy te same optymalizacje, co w przypadku eliminacji Gaussa z częściowym wyborem elementu głównego oraz rozkładu LU bez wyboru. Pamiętamy również o prawidłowym ograniczeniu wykonywania działań na elementach macierzy, tak jak w przypadku eliminacji Gaussa z częściowym wyborem elementu głównego. Udało nam się w tej części zredukować złożoność nieoptymalizowanego algorytmu wynoszącą $O(n^3)$ do $O(n)$.

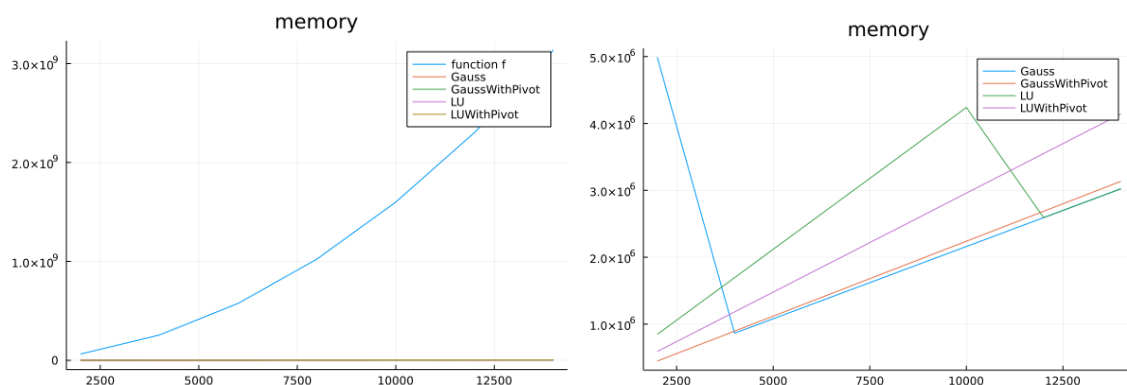
Wyznaczanie rozwiązań jest podobne do standardowego rozkładu LU, korzystamy wtedy dodatkowo z obliczonego wcześniej wektora permutacji p , tu również należy pamiętać o odpowiednim ograniczeniu iteracji. W tej fazie również redukujemy złożoność z $O(n^2)$ do $O(n)$.

3. Wyniki

Wykres 1 Czas wykonywania poszczególnych funkcji w zależności od liczby równań układu, $f=A/b$, eliminacji Gaussa, eliminacji Gaussa z częściowym wyborem elementu głównego, funkcji rozwiązującej układ z wyznaczonym wcześniej LU, funkcji rozwiązującej układ z wyznaczonym wcześniej LU z częściowym wyborem elementu głównego.



Wykres 2 Wykorzystana pamięć w stosunku do liczby równań układu, $f=A/b$, eliminacji Gaussa, eliminacji Gaussa z częściowym wyborem elementu głównego, funkcji rozwiązującej układ z wyznaczonym wcześniej LU, funkcji rozwiązującej układ z wyznaczonym wcześniej LU z częściowym wyborem elementu głównego.



4. Wnioski

Jeśli macierz jest „rzadka” z rozmieszczonymi regularnie zerami, to możemy wykorzystać jej specyficzną strukturę do zoptymalizowania algorytmów, w których z niej korzystamy. W sprawozdaniu pokazano, że złożoność wyznaczenia rozwiązania układu równań $Ax = b$ można zredukować z $O(n^3)$ do $O(n)$, zarówno w przypadku złożoności pamięciowej jak i czasowej, na co wskazują wykresy znajdujące się powyżej. Dodatkowo można zauważyć, że funkcja rozwiązująca układ z wyznaczonym wcześniej LU, z częściowym wyborem elementu głównego jest odrobinę szybsza od pozostałych.