Obliczenia Naukowe
Lista 5
Laboratoria

Piotr Szyma

02 stycznia 2017

1 Zadanie 1

1.1 Opis problemu

Celem tego zadania było zaimplementowanie metody rozwiązującej układ $Ax = b$ metodą eliminacji Gaussa uwzględniając specyficzną postać macierzy $A \in \mathbb{R}^{n\times n}$, tj. macierzy o charakterystycznej strukturze przedstawionej poniżej oraz wektora prawych stron $b \in \mathbb{R}^n$, $n > 4$.

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

gdzie $v = \frac{n}{l}$, przy załozeniu, że n jest podzielne przez l, a l jest rozmiarem wszystkich bloków wewnętrznych. Bloki $A_k \in \mathbb{R}^{l\times l}$ to macierze gęste, bloki 0 to macierze zerowe stopnia l bloki $B_k \in \mathbb{R}^{l\times l}$ następującej postaci:

$$\mathrm{B}_k = \begin{bmatrix} 0 & \dots & 0 & b^k_{1l-1} & b^k_{1l} \\ 0 & \dots & 0 & b^k_{2l-1} & b^k_{2l} \\ \vdots & & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & b^k_{ll-1} & b^k_{ll} \end{bmatrix}$$

Natomiast bloki $C_k \in \mathbb{R}^{l\times l}$ to macierze diagonalne:

$$\mathrm{C}_k = \begin{bmatrix} c^k_1 & 0 & 0 & \dots & 0 \\ 0 & c^k_2 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & c^k_{l-1} & 0 \\ 0 & \dots & 0 & 0 & c^k_l \end{bmatrix}$$

Należało stworzyć dwa warianty implementacji:

1. bez wyboru elementu głównego
2. z wyborem elementu głównego

Dodatkowym wymaganiem była złożoność algorytmu, która - po uwzględnieniu postaci macierzy - ma wynosić nie, tak jak w standardowym algortytmie eliminacji $O(n^3)$, a $O(n)$.

1.2 Analiza

Istotą metody eliminacji Gaussa są dwa etapy - pierwszy z nich to sprowadzenie macierzy do postaci schodkowej, tj. takiej, w której niezerowe komórki znajdują się jedynie powyżej komórek o indeksie wiersza i kolumny równym sobie, następny rozwiązanie odpowiadającego tej macierzy układu równań.

1.3 Rozwiązanie

1.3.1 Bez wyboru elementu głównego

Pierwsza część algorytmu to manipulacja macierzą - za pomocą operacji elementarych - w celu doprowadzenia macierzy do postaci TRÓJKĄTNEJ GÓRNEJ. Idea tej części opiera się na: wyborze wiersza głównego, następnie mnożeniu go przez odpowiedni czynnik i dodawaniu do każdych kolejnych wierszy tak, by wiersze te w danej kolumnie zostały wyzerowane.

$$W_n = W_n + \alpha W_1$$

$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \cdots & a_{1,n} \\ 0 & a_{2,2} & a_{2,3} & \cdots & a_{2,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & a_{n-1,2} & a_{2,3} & \cdots & a_{n-1,n} \\ 0 & a_{n,2} & a_{n,3} & \cdots & a_{n,n} \end{bmatrix}$$

W tym wypadku za wiersz główny został obrany wiersz W_1, a przy każdym innym wierszu współczynnik α_n wynosił $0 = a_{i,1} - \alpha\ a_{1,1}$, tj. $\alpha = \frac{a_{i,1}}{a_{1,1}}$ Po wykonaniu serii działań na kolejnych postaciach macierzy $A_1 \to A_2 \to \cdots \to A_n$ otrzymujemy macierz trójkątną.

$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \cdots & a_{1,n} \\ 0 & a_{2,2} & a_{2,3} & \cdots & a_{2,n} \\ 0 & 0 & a_{3,3} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & a_{n-1,n} \\ 0 & 0 & 0 & \cdots & a_{n,n} \end{bmatrix}$$

Z tej macierzy (zaczynając od ostatniego wiesza ostatnej kolumny) jesteśmy wstanie wyznaczyć szukany wektor X rozwiązań równania.

1.3.2 Wybór elementu głównego

W powyższym przykładzie musimy jednak poczynić pewne dodatkowe założenie - komórki a_{kk}, które używamy jako dzielniki w metodzie Gaussa nie mogą być równe zero. W przeciwnym wypadku dochodziłoby do dzielenia przez zero. Mało tego, ich wartości nie mogą być bliskie zero - dlatego doprowadzenie macierzy do postaci, w której komórki a_{kk} mają jak największą wartość, pozwala poprawić numeryczną dokładność. Szukam więc wiersza - spośród wierszy W_i t. że $i \in (k, \ldots, n)$. Wybór spośród wierszy $i < k$ zaburzyłby strukturę, do której dążymy - tj. zerową lewą dolną macierz trójkątną. Będąc w tej sytuacji:

$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \cdots & a_{1,n} \\ 0 & a_{2,2} & a_{2,3} & \cdots & a_{2,n} \\ 0 & a_{3,2} & a_{3,3} & \cdots & a_{3,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & a_{n-1,2} & a_{2,3} & \cdots & a_{n-1,n} \\ 0 & a_{n,2} & a_{n,3} & \cdots & a_{n,n} \end{bmatrix}$$

Wybieramy komórkę o największej wartości spośród $a_{i,2}$, $i \in (2, n)$, następnie zamieniamy miejscami wiersze W_2 i W_i.

1.4 Złożoność

W rozpatrywanej w zadaniu macierzy, w przypadku eliminacji Gaussa, nie osiągniemy oczekiwanej liniowej złożności przy klasycznym sposobie eliminacji. Aby osiągnąć odpowiednią złożoność, do eliminacji należy wziąć jedynie niezerowe komórki, które znajdują się w blokach A, B oraz C, a więc zawężając - w każdej iteracji - zakres wierszy do wyzerowania kolumny do l - taka jest maksymalna rozpiętość niezerowych wartości, a zakres kolumn w danym wierszu do $3l$. Tym sposobem otrzymujemy złożoność $O(n * l * 3l)$, a uwzględniając notację O duże oraz fakt, że l traktujemy jako stałą, dostajemy złożoność liniową. W celu osiągniecia odpowiedniej złożoności pamięciowej, do przechowywania macierzy zostały użyte macierze typu `SparseMatrixCSC`, zaimplementowane z myślą o macierzach rzadkich, pamiętające tylko elementy niezerowe.

2 Zadanie 2

2.1 Opis problemu

Celem tego zadania było, dla macierzy o strukturze takiej, jak w zadaniu pierwszym, stworzyć metodę wyznaczającą rozkład LU przy pomocy metody eliminacji Gaussa. W tym wypadku również należało stworzyć dwa warianty:

1. bez wyboru elementu głównego
2. z wyborem elementu głównego

2.2 Analiza

Rozkład LU to dwie macierze, L - będąca lewą dolną macierzą trójkątną, na indeksach a_{kk} mającą wartości 1, a U - prawą górną. Iloczyn tych macierzy winien dać wyjściową macierz, której rozkładu LU szukamy, tj.:

$$M = LU$$

2.3 Rozwiązanie

Do znalezienia układu LU zadanej macierzy można posłużyć się metodą eliminacji Gaussa podczas pierwszej pętli zerującej lewą dolną macierz trójkątną. Macierz po wyzerowaniu staje się szukaną prawą górną macierzą U, natomiast ze współczynników αij, które używaliśmy do wyzerowania zawartości komórek, tworzymy kolejną macierz - używając macierzy jednostkowej - następnie wypełniając ją odpowiednimi mnożnikami. Tak utworzona macierz jest szukaną macierzą L.

$$\text{U} = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \dots & a_{1,n} \\ 0 & a_{2,2} & a_{2,3} & \dots & a_{2,n} \\ 0 & 0 & a_{3,3} & \dots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & a_{n-1,n} \\ 0 & 0 & 0 & \dots & a_{n,n} \end{bmatrix}$$

$$\text{L} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ \alpha_{2,1} & 1 & 0 & \dots & 0 \\ \alpha_{3,1} & \alpha_{3,2} & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \alpha_{n-1,1} & \alpha_{n-1,2} & \alpha_{n-1,3} & \dots & 1 \end{bmatrix}$$

3 Zadanie 3

3.1 Opis problemu

W tym zadaniu należało stworzyć metodę, która, dla wyznaczonego podziału macierzy $A = LU$, znajdywała rozwiązanie układu równań $Ax = b$, t. $LUx = b$.

3.2 Analiza

W celu znalezienia rozwiązania równania $LUx = b$, należy rozbić obliczenia na dwa etapy, tj. wyliczanie:

1. $Ly = b$
2. $y = Ux$

3.3 Rozwiązanie

Pierwszy z etapów to znalezienie y, tj. rozwiązanie $Ly = b$. Macierz L jest w postaci trójkątnej, dlatego można w prosty sposób, wykonując na układzie *backward substitution*, znaleźć wektor y. Po odnalezieniu wektora y, należy rozwiązać $Ux = y$, tutaj wykonujemy $forward\ substitution$, które daje poszukiwany wektor x.

$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \dots & a_{1,n} \\ 0 & a_{2,2} & a_{2,3} & \dots & a_{2,n} \\ 0 & 0 & a_{3,3} & \dots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & a_{n-1,n} \\ 0 & 0 & 0 & \dots & a_{n,n} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{n-1} \\ y_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_{n-1} \\ b_n \end{bmatrix}$$

W powyższym układzie nieznane są jedynie współczynniki y_i wektora Y. *Backward substitution* zaczynamy od wyliczenia $y_n = \frac{b_n}{a_{nn}}$. Wyznaczanie kolejnych współczynników y_k wykonujemy podstawiając do wzoru $y_k = \frac{b_k - \sum_{j=k+1}^{n} a_{kj} x_j}{a_{kk}}$ - cofamy się do wcześniejszych wierszy i wyliczamy kolejne y_k korzystając z wcześniejszych wyników.

$$\begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ \alpha_{2,1} & 1 & 0 & \dots & 0 \\ \alpha_{3,1} & \alpha_{3,2} & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \alpha_{n-1,1} & \alpha_{n-1,2} & \alpha_{n-1,3} & \dots & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{n-1} \\ y_n \end{bmatrix}$$

W drugim etapie mamy analogiczną sytuację, choć tutaj występuje macierz trójkątna prawa górna. W tym przypadku użyjemy $forward\ substitution$. Zaczynamy od x_1, które jest po prostu równe y_1. Kolejne x_i wyliczamy za pomocą: $x_k = \frac{y_k - \sum_{j=1}^{k-1} \alpha_{kj} x_j}{\alpha_{kk}}$