Sprawozdanie 5

Aleksander Głowacki

19.01.2023

Spis Treści

1 Opis problemu

Rozwiązanie ukladu równań liniowych postaci:

$$Ax = b$$

dla zadanej macierzy A o charakterystycznej blokowej strukturze i zadanego wektora b.

2 Struktura Danych

Zadana macierz A po rozpisaniu na wiersze jest następującej postaci ($n = 12, l = 4$):

$$
\begin{pmatrix}
a & a & a & a & c & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
a & a & a & a & 0 & c & 0 & 0 & 0 & 0 & 0 & 0 \\
a & a & a & a & 0 & 0 & c & 0 & 0 & 0 & 0 & 0 \\
a & a & a & a & 0 & 0 & 0 & c & 0 & 0 & 0 & 0 \\
b & b & b & b & a & a & a & a & c & 0 & 0 & 0 \\
0 & 0 & 0 & b & a & a & a & a & 0 & c & 0 & 0 \\
0 & 0 & 0 & b & a & a & a & a & 0 & 0 & c & 0 \\
0 & 0 & 0 & b & a & a & a & a & 0 & 0 & 0 & c \\
0 & 0 & 0 & 0 & b & b & b & b & a & a & a & a \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & b & a & a & a & a \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & b & a & a & a & a \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & b & a & a & a & a
\end{pmatrix}
$$

Dane przechowuję w Sparse Arrayu. Ma on pewną wadę - problem z dostępem do niezainicjalizowanych pól. Gdy czytam plik wejściowy, w strukturę wpisuję tylko przeczytane dane. Algorytm eliminacji gaussa został zoptymalizowany pod strukturę macierzy, gdzie dane w rzędach i kolumnach są ograniczone do stałej długości $2 * l + 1$ w skrajnym przypadku. Jendak na tej długości zdarzają się puste miejsca, niezainicjowane. Dlatego muszę je dotatkowo wypełnić zerami. Na macierzy oznaczylem kolorem czerwonym jeden z przebiegów zewnętrznej pętli for. Algorytm jest opisany w dalszej sekcji.

3 Eliminacja Gaussa

Algorytm eliminacji Gaussa w podstawowej formie wygląda następujaco

for k ← 1 to n-1 **do**

  for i ← k+1 to n **do**

    $I_{ik}^{(k)} \leftarrow a_{ik}^{(k)} / a_{kk}^{(k)}$

    for j ← k+1 to n **do**

      $a_{ij}^{(k+1)} \leftarrow a_{ij}^{(k)} - I_{ik} a_{kj}^{(k)}$

    end for

    $b_i^{(k+1)} \leftarrow b_i^{(k)} - I_{ik} b_k^{(k)}$

  end for

end for

gdzie $a_{ik}^{(k)}$ jest elementem o indeksach i, k po $k{-}tym$ kroku, I_{ik} jest mnożnikiem rzędu i w $k - tym$ kroku.

Zauważmy, że w każdym kroku do wyzerowania mamy tylko elementy macierzy A_k których jest nie więcej niż $l - 1$ oraz elementy macierzy B_k których jest nie więcej niż 1. Wyjątkiem jest co l-ta kolumna, gdzie na macierz A_k przypada brak elementów do zerowania, zaś na macierz B_k - l elementów pod przekątną macierzy głównej.W związku z czym druga pętla wystarczy, że będzie się wykonywać l razy.

Podobnie, zauważmy że w każdym rzędzie, licząc od przekątnej, mamy conajwyżej $l + 1$ elementów: najwięcej mamy w pierwszym i ostatnim rzędzie $k - tych$ podmacierzy, w pierwszym mamy l w B_k, l w A_k oraz 1 w C_k a w ostatnim mamy 1 w B_k, l w A_k oraz l w C_k. Trzecia pętla może być wówczas wykonywana $l + 1$ razy.

Zmodyfukujmy algorytm następująco:

for k $\leftarrow$ 1 to n-1 **do**

  for i $\leftarrow$ k+1 to $min(k+l, n)$ **do**

    $I_{ik}^{(k)} \leftarrow a_{ik}^{(k)} / a_{kk}^{(k)}$

    for j $\leftarrow$ k+1 to $min(k+1+l$, n) **do**

      $a_{ij}^{(k+1)} \leftarrow a_{ij}^{(k)} - I_{ik} a_{kj}^{(k)}$

    end for

    $b_i^{(k+1)} \leftarrow b_i^{(k)} - I_{ik} b_k^{(k)}$

  end for

end for

do elementów a_{ik} dostajemy się w czasie stałym, przy użyciu funkcji pomocniczych, więc złożoność algorytmu jest $\Theta(n \cdot l^2)$. Spodziewamy się, że l jest małe i traktując to jako stałą otrzymujemy złożoność $\Theta(n)$

4 Eliminacja Gaussa z częsciowym wyborem

Przypomnę, że częściowy wybór w $k-tym$ kroku polega na znalezieniu elementu takiego, że $|a_{pk}^{(k)}| = \max_{k \le i \le n} |a_{ik}^{(k)}|$ a następnie przestawieniu wiersza p z $k-tym$ w macierzy i wektorze prawych stron.

Ze względu na postać naszej macierzy wystarczy sprawdzić l elementów, ponieważ reszta jest zerowa (o czym mówiłem wcześniej), tzn. znaleść element taki, że $|a_{pk}^{(k)}| = \max_{k \le i \le k+l} |a_{ik}^{(k)}|$ a następnie dokonać przestawień. Wybór jest $\Theta(l)$, a przestawienia są $\Theta(1)$ ponieważ dokonujemy przestawień w tablicy. Ponieważ l jest stała to wybór elementu głównego nie wpływa na złożoność algorytmu.

5 Rozwiązywanie układu po wykonaniu eliminacji Gaussa

Gdy mamy już macierz trójkątną przekształconą po eliminacji Gaussa, układ równań rozwiązujemy w następujący sposób dla dowolnej macierzy

$$x_n = \frac{b_n}{u_{nn}}$$

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

gdzie x_k jest $k-tym$ elementem wektora rozwiązań, b_k jest $k-tym$ elementem wektora prawych stron oraz u_{ik} jest elementem macierzy o indeksach i, k. Dla naszej specyficznej postaci macierzy wystarczy zsumować tyle elementów ile możemy mieć maksymalnie w rzędzie, czyli $2 \cdot l + 1$, tzn

$$x_k = \frac{b_k - \sum_{j=k+1}^{min(k+2+2l,n)} u_{kj} x_j}{u_{kk}}$$

Ponieważ obliczamy n składowych x, to złożoność jest $\Theta(n \cdot l) = \Theta(n)$ jeśli l jest stałą.

6 Eksperymenty

Algorytmy przetestowano przy użyciu macierzy testowych podanych na stronie. Każdą z nich pomnożono przez wektor jedynek i wynikowy wektor użyto jako wektor prawych stron. Rozwiazano równanie wszystkimi metodami. W tabeli przedstawiono moduły różnic między rozwiązaniem otrzymanym a rzeczywistym. Jak widać, algorytm z wwykorzystaniem pivota jest skuteczniejszy - prowadzi do mniejszych błędów.

	Gauss	GaussP
16	2.891003327845126e-15	4.666662269401757e-16
10000	2.959372701170333e-14	4.047989523512954e-16
50000	7.581797020478161e-14	4.1262156468807015e-16
100000	5.137266118617371e-13	4.950949311391626e-16
300000	8.999759241982931e-14	3.977397923826499e-16
500000	7.583582056241379e-14	3.9715730002224353e-16