Politechnika Wrocławska
Wydział Podstawowych Problemów Techniki

Obliczenia naukowe

Sprawozdanie z zajęć laboratoryjnych

Lista 5

Autor:
Jakub Pezda
221426

1.1 Opis problemu

Ćwiczenie polega na napisaniu funkcji rozwiązującej układ `Ax = b` metodą eliminacji Gaussa uwzględniającą specyficzną postać macierzy `A` dla dwóch wariantów:

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

1.2 Rozwiązanie

Do rozwiązania równania użyto zmodyfikowanej wersji eliminacji Gaussa. W związku ze specyficzną budową macierzy `A` nie musimy zerować wszystkich elementów dla danej kolumny. Po drugie przy zerowaniu danego elementu nie trzeba zmieniać wszystkich wartości w tym rzędzie, wystarczy zmienić tylko te, nad którymi może być liczba różna od zera.
Można zauważyć, że ostatni indeks elementu o wartości różnej od zera w danej kolumnie zawsze jest równy wielokrotności liczby l. Dla kolumn [1, l) jest to l, dla kolumn [l, 2l) jest to 2l, itd. Aby obliczyć, do którego wiersza zerować elementy skorzystano ze wzoru $l + l * \left\lfloor \frac{k}{l} \right\rfloor$.
Ostatni indeks elementu w rzędzie k, który może mieć wartość różną od zera wynosi min(k + l, n). Z tego wynika, że zerując element w danym rzędzie nie musimy zmieniać wartości wszystkich pozostałych elementów tylko do elementu k + l.

Po wykonaniu eliminacji Gaussa skorzystano ze zmodyfikowanego algorytmu podstawiania wstecz. Z łatwością można zauważyć, że ostatni indeks elementu, który należy dodać do sumy dla danego rzędu `k` wynosi `k + l`.
Poniżej przedstawiono pseudokod całego algorytmu.

```
function solveLinearSystem(A, n, l, b)
   for k in 1:n - 1
      rowsToReset = l + l * floor(k / l)
      columns = min(k + l, n)
      for i in k + 1:rowsToReset
         z = A[i, k] / A[k, k]
         A[i, k] = 0
         for j in k + 1:columns
            A[i, j] = A[i, j] - z * A[k, j]
         b[i] = b[i] - z * b[k]

   for i in n:-1:1
      suma = 0.0
      limit = min(n, i + l)
      for j in i + 1:limit
         suma += A[i, j] * x[j]
      x[i] = (b[i] - suma) / A[i, i]
   return x
```

Listing 1: Funkcja rozwiązująca układ metodą eliminacji Gaussa

Funkcja jako argumenty przyjmuje macierz A, wielkość macierzy n, wielkość macierzy wewnętrznych l oraz wektor b.

Ilość elementów w kolumnie do wyzerowania wynosi maksymalnie l. Ilość elementów w rzędzie do zmiany może wynieść maksymalnie l. Ponieważ l jest stałą można ją pominąć więc złożoność czasowa zmodyfikowanej metody eliminacji Gaussa wynosi O(n). Podobnie można zauważyć, że pętla wewnętrzna algorytmu podstawiania wstecz jest zależna od stałej więc tu również złożoność czasowa wyniesie O(n). Dokładne porównanie czasów działania rozwiązywania układu Ax = b pokazano w kolejnym podpunkcie.

Do napisania funkcji rozwiązującej układ Ax = b metodą eliminacji Gaussa z częściowym wyborem elementu głównego zmodyfikowano funkcję przedstawioną na listingu 1.

```
function solveLinearSystemWithRootElement(A, n, l, b)
   p – tablica liczba od 1 do n
   for k in 1:n - 1
      rowsToReset = l + l * floor(k / l)
      for i in k + 1:rowsToReset
         #Jeżeli A[p[k], k] = 0
            #znajdź element o największej wartości co do modułu w tej kolumnie
            #zamień kolumny w tablicy p
         z = A[p[i], k] / A[p[k], k]
         A[p[i], k] = 0

         columns = min(p[i] + l, n)
         for j in k + 1:columns
            A[p[i], j] = A[p[i], j] - z * A[p[k], j]
         b[p[i]] = b[p[i]] - z * b[p[k]]

   for i in n:-1:1
      suma = 0.0
      limit = min(n, p[i] + l)
      for j in i + 1:n
         suma += A[p[i], j] * x[j]
      x[i] = (b[p[i]] - suma) / A[p[i], i]
   return x
```

Listing 2: Funkcja rozwiązująca układ metodą eliminacji Gaussa z częściowym wyborem elementu głównego

Przed obliczeniem ilorazu dwóch elementów należy sprawdzić czy element na przekątnej jest różny od zera. Jeżeli nie jest, to trzeba znaleźć element w tej samej kolumnie o największej wartości co do modułu i zamienić miejscami dwa rzędy. Rozwiązanie układu równań nie zależy od kolejności w jakiej równania są ustawione, więc zmiana rzędów macierzy nie wpłynie na końcowy wynik. W pozostałej części algorytm nie zmienia działania.

1.3 Wyniki

Poniżej przedstawiono czasy działania algorytmów dla danych testowych.

n – wielkość macierzy A
l – wielkość macierzy wewnętrznej
t – czas działania (w sekundach)
m – zużycie pamięci (w MB)

		X = A \ b		Bez wyboru elementu głównego		Z częściowym wyborem elementu głównego	
n	**l**	**t**	**m**	**t**	**m**	**t**	**m**
9	3	0.614677	22.480	0.060709	1.528	0.066813	1.132
1000	4	0.621196	23.881	0.063472	1.717	0.076107	1.329

Jak wynika z tabeli czas działania zaimplementowanego algorytmu dla danych testowych jest około 10 razy szybszy (w wersji bez wyboru elementu głównego). Pamięciowo algorytm jest również dużo lepszym rozwiązaniem.

1.4 Wnioski

Algorytm eliminacji Gaussa bez wyboru elementu głównego nie zadziała, gdy któryś z elementów głównych będzie wynosić 0. Algorytm wykonuje dużą liczbę działań arytmetycznych co może prowadzić do błędów związanych z reprezentacją liczb. Algorytm z częściowym wyborem elementu głównego wymaga dodatkowych przejrzeń w kolumnie lecz jest odporny na dzielenie przez zero.