

Parallel Gaussian Elimination

Szymon Maj & Jarosław Szczęśniak

Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie

January 22, 2012

Jedna z najszybszych metod, doprowadzająca do dokładnego rozwiązania przez wykonanie skończonej ilości działań.

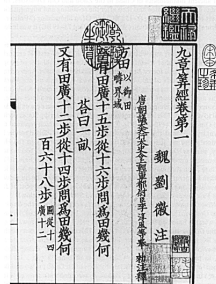
Pozwala na:

- rozwiązywanie układów równań liniowych,
- obliczanie rzędu macierzy,
- obliczanie macierzy odwrotnej,
- obliczanie wyznacznika macierzy:
 - Dodawanie wierszy nie zmienia wartości wyznacznika.
 - Pomnożenie wiersza powoduje pomnożenie wyznacznika.
 - Zamiana dwóch wierszy (lub kolumn) zmienia znak wyznacznika.

Nazwa metody pochodzi od nazwiska niemieckiego matematyka Carla Friedricha Gaussa, żyjącego na przełomie XVIII i XIX wieku.



Doszukano się podobieństw w „The Nine Chapters on the Mathematical Art”, chińskich zapiskach matematycznych, które powstawały między X wiekiem p.n.e. a I wiekiem n.e.



Metoda składa się z dwóch części:

- ❶ Eliminacja w przód
 - wynikiem jest macierz trójkątna
 - usuwanie niewiadomych z kolejnych wierszy macierzy
 - operacje elementarne - dodawanie, mnożenie, zamianę wierszy
- ❷ Podstawianie wstecz
 - warunek - macierz trójkątna
 - obliczenie wartości zmiennych od dołu macierzy
 - podstawianie wartości obliczonych zmiennych w poprzednich równaniach, dochodząc do początku

Złożoność obliczeniowa metody wynosi $O(n^3)$.

- Dekompozycja na poziomie wierszy
- W każdym kolejnym kroku jeden z wierszy (pivot), odejmujemy z odpowiednim współczynnikiem od każdego następnego wiersza
- Operacja odejmowania w tym przypadku jest niezależna, więc można ją zrównoleglić
- Odejmovany wiersz (pivot) zmieniany jest po zakończeniu pracy wszystkich procesorów
- Z każdym kolejnym krokiem ilość operacji zmniejsza się
- Problemem jest narzut komunikacji

```
1 | for pivot_row in (0..n)
2 |   parallel for row in (pivot_row+1..n)
3 |     factor = A[pivot_row][0]/A[row][0]
4 |     for col in (0..n)
5 |       A[row][col] = A[row][col]*factor - A[pivot_row][col]
6 |     end
7 |   end
8 | end
```

n	Communication Time	Workload Time	Total Time
400	2.15	0.72	2.88
800	18.28	5.76	24.03
1200	70.98	19.68	90.66

Figure: Wyniki równoległego algorytmu wykonanego przez 2 procesory [1]

n	Communication Time	Workload Time	Total Time
400	3.46	0.37	3.83
800	23.43	2.90	26.32
1200	82.73	9.85	92.58

Figure: Wyniki równoległego algorytmu wykonanego przez 4 procesory [1]

Bibliografia



S.F.McGinn and R.E.Shaw - „Parallel Gaussian Elimination Using OpenMP and MPI” - University of New Brunswick