Eliminacja Gaussa

Filip Jędrzejewski

24 Grudnia 2023

1 Wstęp

1.1 Cel zadania

Celem zadania było zaprojektowanie i zaimplementowanie równoległego algorytmu eliminacji Gaussa.

1.2 Informacje o programie

Program został w całości napisany w języku `c++`. Dane wejściowe do programu są wczytywane z pliku `.txt`, natomiast wynik wypisywany jest w terminalu.

1.3 Zaimplementowane funkcjonalności

W programie zaimplementowano następujące funkcjonalności:

- Transponowanie danej macierzy
- Współbieżny algorytm eliminacji Gaussa doproawdzający daną macierz do postaci górnej trójkątnej
- Iteracyjny algorytm eliminacji Gaussa
- Obliczanie wyznacznika danej macierzy z definicji
- Obliczanie wyznacznika danej macierzy korzystając z równoległej eliminacji Gaussa
- Rozwiązywanie układu równań liniowych danego w postaci macierzowej

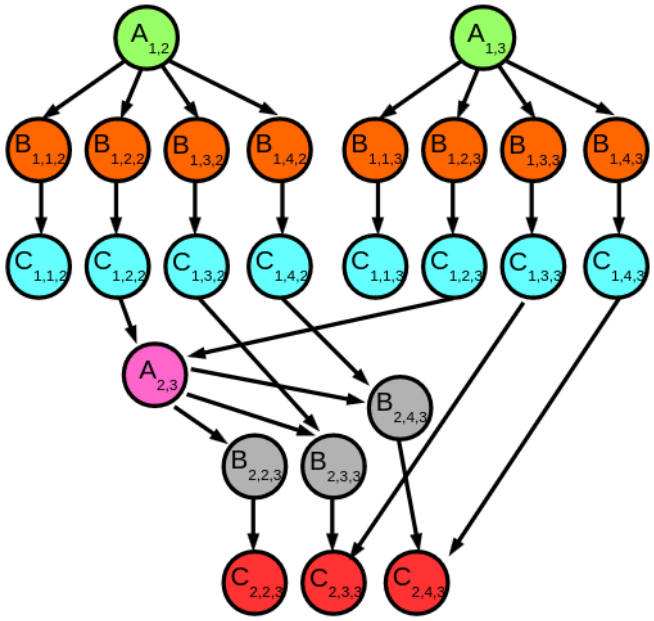
2 Opis teoretyczny

W celu rozwiązania zadania zdefiniowano następujące operacje:

- $A_{i,j}$ - wyznaczenie mnożnika dla wiersza i, by odjąć go od wiersza j.
- $B_{i,k,j}$ - przemnożenie elementu k z wiersza i, przez mnożnik dla wiersza j.

- $C_{i,k,j}$ - odjęcie elementu k z wiersza i od elementu k z wiersza j.

Na podstawie tej listy operacji stworzono grafy Diekerta dla współbieżnego algorytmu eliminacji Gaussa dla macierzy o rozmiarach: 2x2, 3x2, 3x3, 4x3 oraz 4x4, by znaleźć zależności, które by pozwoliły na uogólnienie zależności między poszczególnymi operacjami dla dowolnych rozmiarów macierzy.



Rysunek 1: Przykładowy graf Diekerta dla macierzy 4x3

3 Implementacja

3.1 Lista klas

W programie stworzono następujące klasy:

1. `InputParser` - klasa wykorzystywana do tworzenia obiektu macierzy (`Matrix`) z odczytanego pliku `.txt`.
2. `Matrix` - klasa reprezentująca macierz, przechowująca ją i udostępniająca metody do jej obsługi.
3. `Gauss` - klasa zawierająca metody do eliminacji Gaussa sposobem iteracyjnym lub współbieżnym oraz do rozwiązywania układów równań liniowych.

3.2 Opisy plików

3.2.1 ./src/main.cpp

Główny plik zawierający funkcję `int main()`m która otwiera plik z danymi wejściowymi, przekazuje je do obiektu klasy `InputParser`, uruchamia eliminację Gaussa, wyświetla otrzymaną macierz oraz rozwiązuje układ równań.

3.2.2 ./src/HeaderFiles/constants.h oraz interfaces.h

Pliki nagłówkowe zawierające stałe oraz definicje zaimplementowanych klas.

3.2.3 ./src/InputLogic/InputParser.cpp

Plik zawierający implementację klasy `InputParser` zdefiniowanej jako:

```
class InputParser{
    private:
        int nx;
        int ny;
        Matrix* matrix;
    public:
        InputParser();
        ~InputParser();
        void createNewMatrix(int x, int y);
        void setRow(int i, std::string line);
        Matrix* getMatrix();
};
```

3.2.4 ./src/Matrix/Matrix.cpp

Plik zawierający implementację klasy `Matrix` zdefiniowanej jako:

```
class Matrix{
    private:
        int nx;
        int ny;
        float** tab;
        Matrix* getMatrixWithout(int ii, int ij);
    public:
        Matrix();
        Matrix(int nx, int ny);
        ~Matrix();
        void setValue(int i, int j, float value);
        void subtractRowsWithMultiplicator(int i1, int i2, float value);
        void multiplicateRow(int i, float value);
        int getSizeX();
        int getSizeY();
```

```
        float getValue(int i, int j);
        Matrix* getTransposedMatrix();
        Matrix* addColumns(Matrix* v);
        Matrix* popLastColumn();
        void show(std::string description);
        float getDet(char method);  //g - Gauss OR d - definition
};
```

Metoda `Matrix* getMatrixWithout(int ii, int ij)` zwraca obiekt macierzy powstałej po usunięciu wiersza `ii` oraz kolumny `ij`.

Metoda `Matrix* popLastColumn()` zwraca macierz będącą ostatnią kolumną oraz usuwa ostatnią kolumnę z macierzy początkowej.

Metoda `float getDet(char method)` zwraca wartość wyznacznika macierzy. Argument `char method` służy do określenia, z której metody funkcja ma skorzystać: `'g'` - metoda wykorzystująca współbieżny algorytm eliminacji Gaussa, `'d'` - metoda obliczająca wyznacznik korzystając z definicji.

3.2.5 ./src/Matrix/Gauss.cpp

Plik zawierający implementację klasy `Gauss` zdefiniowanej jako:

```
class Gauss{
    private:
        bool merged = false;
        Matrix* m;
        Matrix* v;
    public:
        Gauss();
        Gauss(Matrix* matrix);
        Gauss(Matrix* matrix, Matrix* vector);
        ~Gauss();
        void classicElimination();
        void threadElimination();
        void toIdentityMatrix();
        Matrix* getMatrix();
        Matrix* getVector();
};
```

Konstruktor `Gauss(Matrix* matrix, Matrix* vector)` łączy daną macierz i wektor w jedną macierz (dodaje do macierzy kolumnę zawierającą wektor).

Metoda `void classicElimination()` wykonuje eliminację Gaussa metodą iteracyjną.

Metoda `void threadElimination()` wykonuje równoległą eliminację Gaussa.

Metoda `void toIdentityMatrix()` rozwiązuje układ równań liniowych dany w postaci macierzowej (do klasy należy przekazać macierz współczynników oraz wektor wyrazów wolnych).

Metody `Matrix* getMatrix()` oraz `Matrix* getVector()` zwracają macierz i wektor, a w przypadku, gdy zostały one połączone w konstruktorze, wtedy również je rozdziela.

4 Dane wejściowe

Dane wejściowe do programu powinny być podane w pliku `./data/inputData.txt` w postaci:

```
rozmiar macierzy
macierz
wektor w postaci transponowanej
```

Przykładowa zawartość pliku `./data/inputData.txt`:

```
3
2.0 1.0 3.0
4.0 3.0 8.0
6.0 5.0 16.0
6.0 15.0 27.0
```

4.1 Uwaga

Program nie uwzględnia ewentualnej konieczności zamiany miejscami wierszy.

5 Wyjście programu

Program wypisuje dane wyjściowe w terminalu. W obecnej konfiguracji pliku `./src/main.cpp` wyjście programu wygląda następująco:

```
macierz odczytana z pliku wejściowego
wektor odczytany z pliku wejsciowego
transponowany wektor
macierz po równoległej eliminacji Gaussa
wektor po równoległej eliminacji Gaussa
macierz jednostkowa (otrzymana podczas rozwiązywania układu równań)
wektor będący rozwiązaniem układu równań
```

Dane zwrócone przez program dla przykładowych danych podanych w poprzedniej sekcji:

```
Input matrix:
| 2.00  1.00    3.00    |
| 4.00  3.00    8.00    |
| 6.00  5.00    16.00   |
```

```
Input vector:
| 6.00  15.00   27.00   |

Transposed vector:
| 6.00  |
| 15.00 |
| 27.00 |

Matrix after elimination:
| 2.00  1.00    3.00    |
| 0.00  1.00    2.00    |
| 0.00  0.00    3.00    |

Vector after elimination:
| 6.00  |
| 3.00  |
| 3.00  |

Indentity matrix:
| 1.00  0.00    0.00    |
| 0.00  1.00    0.00    |
| 0.00  0.00    1.00    |

Vector:
| 1.00  |
| 1.00  |
| 1.00  |
```