Metody Numeryczne - Sprawozdanie 3

Piotr Moszkowicz

21 marca 2019

Spis treści

1 Wstęp Teoretyczny

Na trzecich laboratoriach zajmowaliśmy się szybszym rozwiązaniem układów równań z wykorzystaniem metod iteracyjnych.

1.1 Wektor

Wektor to obiekt matematyczny opisywany za pomocą jego długości, zwrotu oraz kierunku, wykorzystywany głównie w fizyce oraz matematyce. Notacja, z którą wykorzystujemy w obliczeniach matematycznych do zapisu wektora to notacja macierzowa, zilustrowana poniżej:

$$v = \begin{bmatrix} x & y & z \end{bmatrix} \tag{1}$$

Jest to trójelementowy wektor wierszowy.

1.2 Macierz

Macierz to tablica prostokątna, która zawiera liczby. Notacja w jakiej zapisujemy macierze widoczna jest poniżej:

$$A = \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{bmatrix} \tag{2}$$

Na powyższym przykładzie widnieje macierz kwadratowa (ilość kolumn jest równa ilości wierszy) o wymiarze 2. Wyróżniamy kilka rodzajów macierzy, poniżej te najważniejsze, które są istotne dla przebiegu ćwiczenia.

Macierz diagonalna to taka, która posiada wartości różne od zera jedynie na przekątnej (tzw. diagonali).

Macierz wstęgowa to taka, której wszystkie elementy są zerowe poza diagonalą i w jej pobliżu. Mając daną macierz $n \times n$ jej elementy $a_{i,j}$ są niezerowe, gdy $i - k_1 \leqslant j \leqslant i + k_2$ gdzie $k_{1,2} \geqslant 0$ określają szerokość wstęgi.

Macierz trójdiagonalna to taka, która posiada wartości różne od zera jedynie na diagonali, oraz pierwszej naddiagonali i pierwszej poddiagonali.

2 Opis problemu

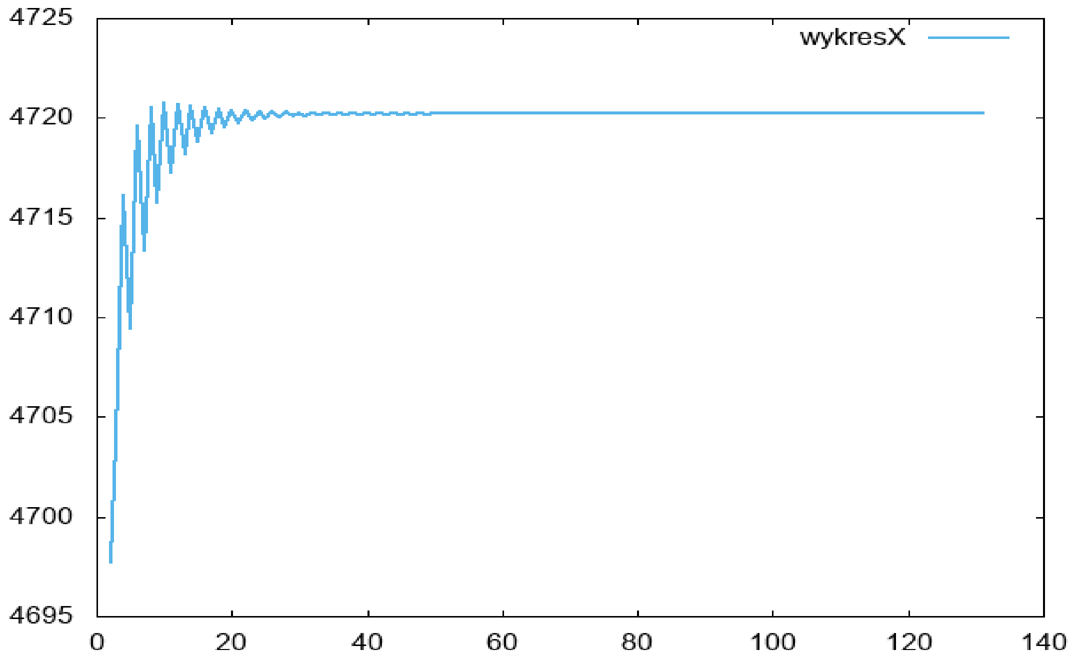
Naszym zadaniem jest rozwiązanie układu równań liniowych $Ax = b$ przy pomocy metody najmniejszego spadku. Aby to uczynić, przeniosłem schemat do kodu programu, który wygląda następująco:

```
do {
        std::array<double, 1000> mult{};
        std::array<double, 1000> mult2{};
        multiplyMatrixByVector(arr, x, mult);
        rk = minusVectors(b, mult);

        multiplyMatrixByVector(arr, rk, mult2);
        double ak = multiplyScalar(rk, rk) / multiplyScalar(rk, mult2);

        std::array<double, 1000> mult3 = multiplyByScalar(rk, ak);
        x = plusVectors(x, mult3);
```

```
        numOfIter++;
        saveNeededStuffToFile(fileName, numOfIter, ak, norm(rk), norm(x));
        saveNormStuffToFile(fileNameNormX, numOfIter, norm(x));
        saveNormStuffToFile(fileNameNormR, numOfIter, norm(rk));
    } while (norm(rk) > s);
```

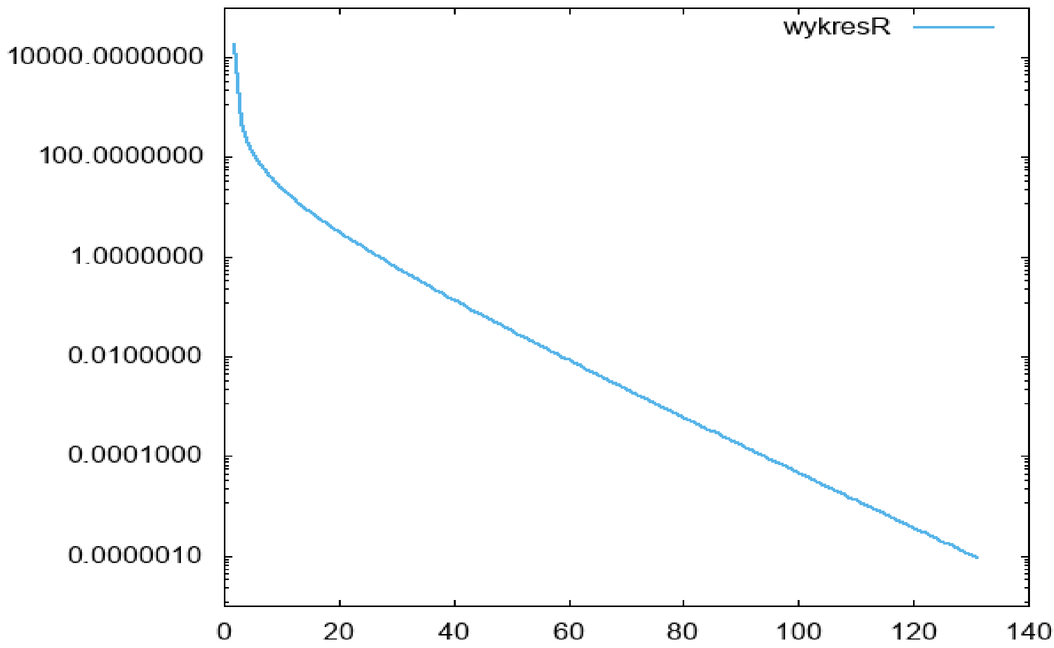
3 Wyniki

Nasz wynik możemy ujrzeć na poniższym wykresie:



Rysunek 1: Wykres wyników wektora x

Natomiast poniżej ilustruję wykres parametru r w skali logarytmicznej.



Rysunek 2: Wykres parametru r

4 Wnioski

Na powyższych wykresach widać, iż otrzymany wynik jest rozsądny oraz na drugim wykresie możemy zaobserwować spadek parametru r co świadczy o uzyskiwaniu coraz dokładniejszych wyników z każdą iteracją.

4.1 Porównanie czasowe z metodą eliminacji zupełnej

Metoda najmniejszego spadku jako metoda iteracyjna nie jest metodą naiwną. Wykorzystuje uproszczenie obliczeń, dzięki czemu możemy zaoszczędzić sporo czasu. Poniżej przedstawiam porównanie czasów wykonania programów:

- Metoda eliminacji zupełnej - 21 sekund
- Metoda najmniejszego spadku - 0.6 sekundy

Jak widać powyżej różnica jest bardzo duża co pokazuje sens stosowania innych metod w celu dalszej optymalizacji.