

SPRAWOZDANIE - LABORATORIUM NR 3

Metoda największego spadku dla macierzy wstęgowej

Marek Kiełtyka

14 marca 2019

1 Wstęp

1.1 Metoda największego spadku

Jako jedna z iteracyjnych metod rozwiązywania układów równań liniowych postaci

$$[A][x] = [b] \quad (1)$$

(gdzie \mathbf{A} - macierz wejściowa, \mathbf{b} - wektor wyrazów wolnych), polega ona na estymowaniu coraz bardziej dokładnego wektora rozwiązań x na podstawie obliczeń w poprzednich iteracjach. Na potrzeby algorytmu wprowadza się wektor reszt r będący różnicą między obiema stronami równania (1).

1.2 Pseudokod

Poczynając od inicjalizacji b oraz x , poprzez kolejne przybliżenia wyniku dąży się do osiągnięcia zadanego progu zbieżności przez wektor reszt. Jeśli ten próg zostanie osiągnięty, problem można uznać za rozwiązany. Poniższe instrukcje definiują proces postępowania.

$$\begin{aligned} &do \{ r_k = b - Ax_k \\ &\quad \alpha_k = \frac{r_k^T \cdot r_k}{r_k^T \cdot Ar_k} \\ &\quad x_{k+1} = x_k + \alpha_k r_k \\ &\quad \} while(||r_k||_2 > 10^{-6}) \end{aligned}$$

Zmienna k oznacza tutaj numer iteracji, wartości r_k, α_k, x_k są pamiętane tylko podczas danego obiegu pętli. Dla pojedynczej precyzji próg zbieżności założono na poziomie 10^{-3} , dla podwójnej - jak w powyższym pseudokodzie. Transpozycje wektorów nie są konieczne w czasie wykonania programu z uwagi na jednoznaczność reprezentacji w pamięci komputera.

2 Zadanie do wykonania

2.1 Opis problemu

Celem laboratorium było rozwiązanie układu równań liniowych postaci (1) przy użyciu opisywanej metody i przedstawienie go na wykresie. Założono następujące warunki początkowe:

$$a_{i,j} = \frac{1}{1 + |i - j|}, \quad |i - j| \leq 5 \text{ dla } i, j = 1, \dots, n$$
$$a_{i,j} = 0, \quad |i - j| > 5 \text{ dla } i, j = 1, \dots, n$$

$$N = 1000 \text{ - rozmiar macierzy } A$$
$$b_i = i \quad \text{dla } i = 1, \dots, n$$
$$x_{i1} = 0 \quad \text{dla } i = 1, \dots, n$$
$$x_{i2} = 1 \quad \text{dla } i = 1, \dots, n$$

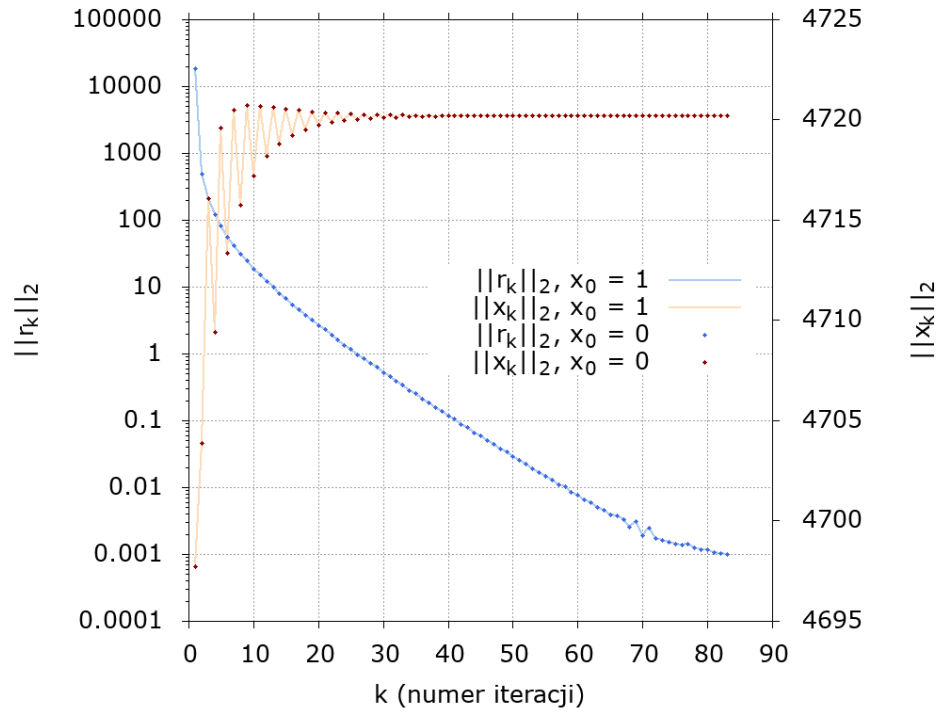
Należało zaimplementować algorytm i wykorzystać go osobno dla wektorów x_{i1} , x_{i2} . Ważne było wykonanie pomiaru czasu obliczeń i porównanie go z czasem odpowiadającym metodzie Gaussa-Jordana dla analogicznego problemu.

2.2 Wyniki

Do opracowania rozwiązania użyto bibliotek: *Numerical Recipes* i opracowanej samodzielnie, która bazuje na wspomnianej bibliotece *NR* i znacząco ułatwia zapis operacji algebraicznych. W każdej iteracji wyznaczano wartości norm euklidesowych wektorów reszt i rozwiązań.

$$\begin{cases} \|r_k\|_2 = \sqrt{r_k^T r_k} \\ \|x_k\|_2 = \sqrt{x_k^T x_k} \end{cases} \quad (2)$$

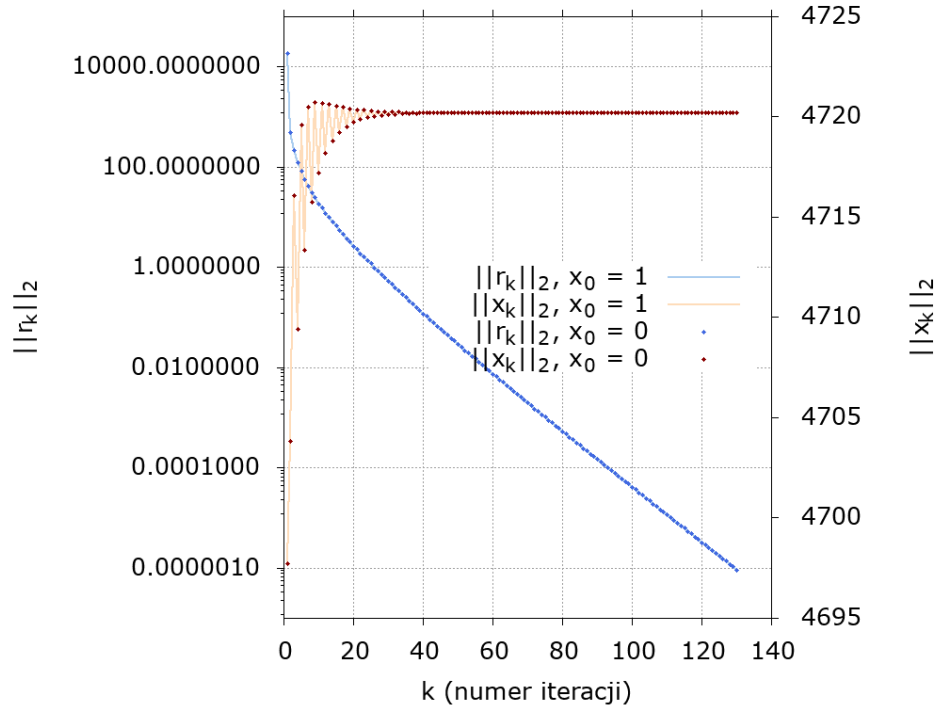
W sumie rozwiązano pięć układów równań - po dwa dla pojedynczej i podwójnej precyzji ($\vec{x}_1 = 0$ oraz $\vec{x}_2 = 1$), a także jeden metodą eliminacji zupełnej. Pomiary czasu wykonano tylko dla typu *float*.



Wykres 1: Norma wektora reszt $\|r_k\|_2$ oraz norma wektora rozwiązań $\|x_k\|_2$ w funkcji numeru iteracji dla pojedynczej precyzji i zbieżności 10^{-3} .

Lp.	Największego spadku ($N = 1000$)	Eliminacji zupełnej ($N = 400$)
1	719.163	758.858
2	710.449	757.143
3	703.752	754.007
4	708.115	763.167
5	703.155	757.843
6	701.794	752.966
7	702.043	754.356
8	703.772	754.839
9	699.344	753.149
10	704.403	751.317

Tablica 1: Porównanie czasów rozwiązania dla obu metod i różnych rozmiarów macierzy w milisekundach. Pojedyncza precyzja.



Wykres 2: Norma wektora reszt $\|r_k\|_2$ oraz norma wektora rozwiązań $\|x_k\|_2$ w funkcji numeru iteracji dla podwójnej precyzji i zbieżności 10^{-6} .

3 Wnioski

Patrząc na wykresy można stwierdzić, że początkowe wartości elementów wektora \vec{x} nie wpływają w żaden sposób na rozwiązanie - widać idealne pokrycie obu przypadków. Mimo założenia tak niskich progów zbieżności, norma wektora \vec{x} stabilizuje się w okolicach czterdziestej iteracji, co pokazuje szybkość metody największego spadku.

Kolejnym dowodem na efektywność jest zawartość tablicy (1). Macierz wejściowa może składać się nawet z miliona elementów, a mimo to uzyskujemy rozwiązanie szybciej niż choćby dla 160 000 elementów w przypadku klasycznej metody Gaussa-Jordana. Różnica bierze się również z tego, że do mnożenia macierzy przez wektor użyto ulepszanego algorytmu z uwagi na rzadkość macierzy wejściowej. Celowo nie umieściłem w tabeli wyniku metody eliminacji zupełnej dla $N = 1000$ przez rosnący wykładniczo czas oczekiwania na rezultaty.

Rozwiązując problem, możnaby również dokonać optymalizacji pamięciowej, gdyż

nonsensem jest pamiętanie licznych zerowych elementów tak dużej macierzy, które nie wnoszą do wyniku. Wystarczy przechowywać w pamięci tylko główną przekątną oraz znaczące przekątne znajdujące się nad oraz pod główną diagonalą, co znacząco oszczędziłoby niezbędne zasoby pamięci.