Hleb Shypula
286720
20 marca 2020

Sprawozdanie 3

# Rozwiązywanie układu równań liniowych metodą największego spadku (SD)

## 1. Wstęp teoretyczny

**Metoda największego spadku –** gradientowy algorytm, w którym wielkość kroku $\alpha_k$ jest dobierana tak, aby otrzymać njwiększą wartość spadku wartości funkcji w każdym kolejnym punkcie.
Wartość $\alpha_k$ zazwyczaj wylicza się następująco:

$$\alpha_k = \arg\min_{a>0} f(x^{(k)} - \alpha \bigtriangledown f(x^{(k)}))$$

W naszym przypadku algorytm miał być zaimplementowany w taki sposób:

inicjalizacja: k=0, $\boldsymbol{x}$, $\boldsymbol{b}$, A
do{
k++
$\boldsymbol{r} = \boldsymbol{b} - A\boldsymbol{x}$
$\alpha = \frac{\boldsymbol{r}^T\boldsymbol{r}}{\boldsymbol{r}^T A\boldsymbol{r}}$
$\boldsymbol{x} = \boldsymbol{x} + \alpha\boldsymbol{r}$ //aktualizacja rozwiązania
} while ( $\|\boldsymbol{r}\|_2 > 10^{-6}$ && $k < 500$ )

gdzie: pogrubiona czcionka oznacza wektor, k – numer iteracji, ***x*** to aktualne przybliżenie wektora rozwiązań, ***b*** – wektor wyrazów wolnych, A – macierz nxn, a ***r*** jest wektorem reszt.
Jako warunek zakończenia iteracji przyjąłem $\|\boldsymbol{r}\|_2 > 10^{-6}$ i $k < 500$.

## 2. Problem

Na początku tworzenia programu zdefiniowałem **macierz A** o rozmiarze nxn i wypełniłem zgodnie z następującym wzorem:

$$A_{i,j} = \begin{cases} \frac{1}{1.0+|i-j|}, & |i-j| \leq m \\ 0, & |i-j| > m \end{cases}$$

gdzie: *i, j = 0, ... , n-1*.

Dalej utworzyłem wektor wyrazów wolnych $\boldsymbol{b}$ i wypełniłem następująco:

$$b_i = i, \quad i = 0, \ldots, n-1$$

W kolejnym kroku przeszedłem do głównego problemu laboratorium.Czyli do **rozwiązania układu równań** $A\boldsymbol{x} = \boldsymbol{b}$ metodą największego spadku dla dwóch wektorów startowych $\boldsymbol{x}$ tj. dla: a) $\boldsymbol{x} = 0$ , b) $\boldsymbol{x} = 1$.

W każdej iteracji zapisywaliśmy do pliku: aktualny numer iteracji (k), wartość normy euklidesowej wektora reszt ($\|\boldsymbol{r}\|_2 = (\boldsymbol{r}^T\boldsymbol{r})^{1/2}$), wartość $\alpha_k$, wartość normy euklidesowej wektora rozwiązań ($\|\boldsymbol{x}\|_2 = (\boldsymbol{x}^T\boldsymbol{x})^{1/2}$).

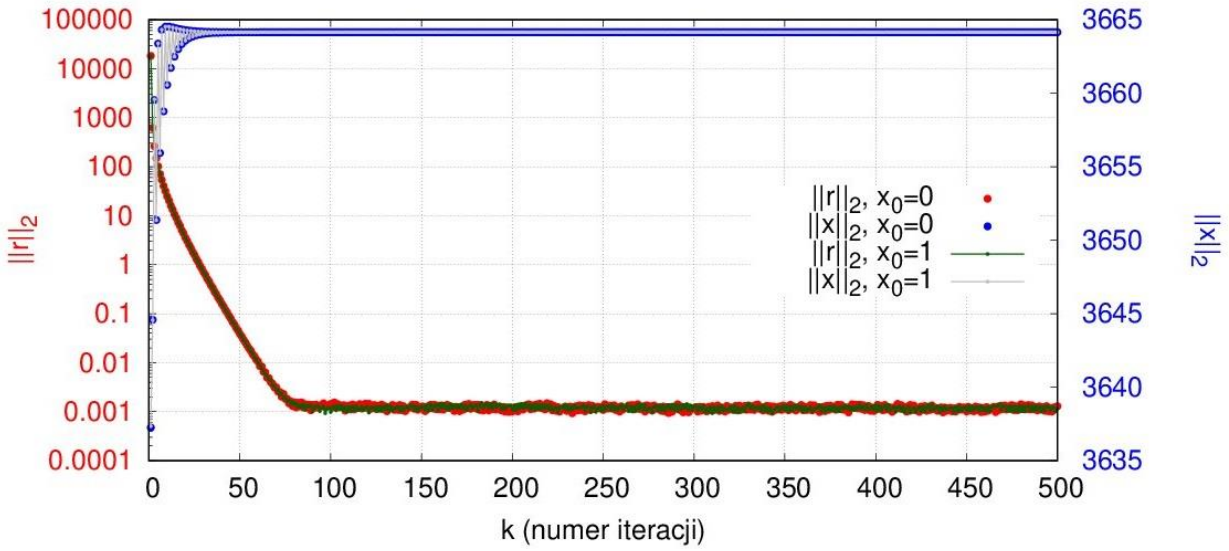
Obliczenia miałem wykonać w pojedynczej precyzji, czyli definiując odpoweidnie zmienne typu **float**, a następnie w podwójnej precyzji korzystając z typu **double**.

Na końcu wygenerowałem odpowiednie **wykresy** przedstawiające nasze wyniki za pomocą Gnuplot. Wykresy są przedstawione w kolejnej części sprawozdania.

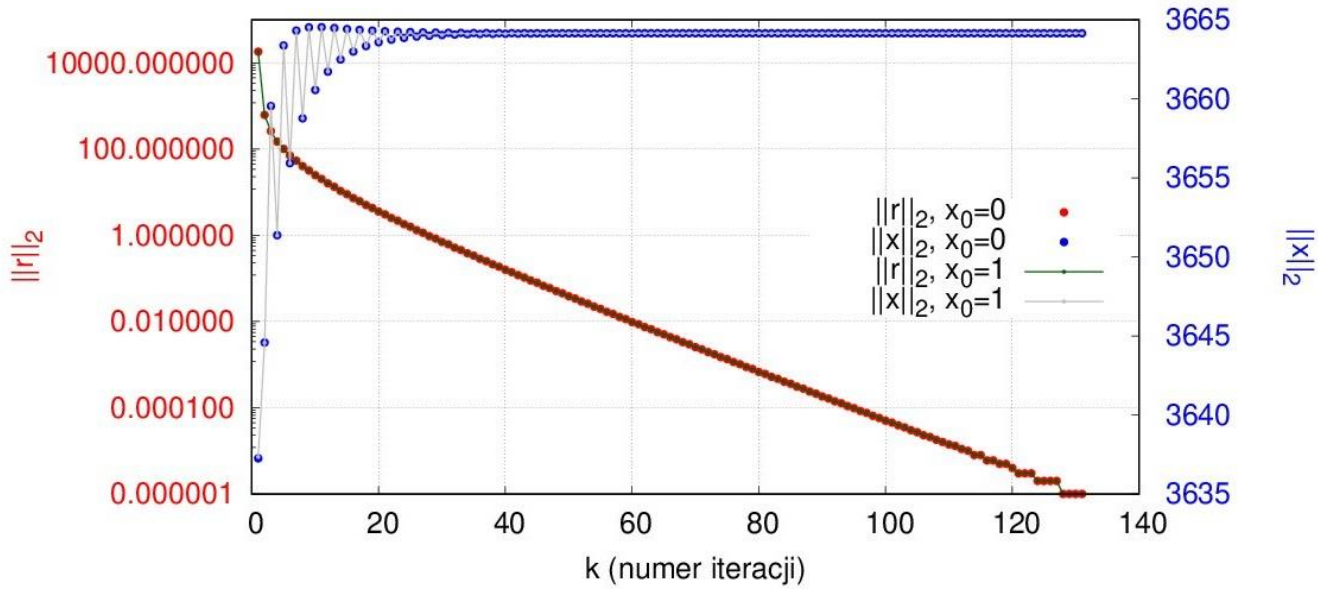
# 3. Wyniki

Ze względu na dużą ilość iteracji dołaczam tylko pierwsze i ostatnie 10 kroków dla wektora rozwiązań $\boldsymbol{x} = 1$.

| k | $\|\boldsymbol{r}\|_2$ | $\alpha_k$ | $\|\boldsymbol{x}\|_2$ |
|---|---|---|---|
| 1 | 18106.355127 | 0.199377 | 3637.337075 |
| 2 | 620.019682 | 0.371940 | 3644.555117 |
| 3 | 259.517897 | 0.351280 | 3659.584813 |
| 4 | 148.908031 | 0.365563 | 3651.351531 |
| 5 | 101.329839 | 0.351313 | 3663.397876 |
| 6 | 70.861661 | 0.366592 | 3655.918352 |
| 7 | 53.806976 | 0.352367 | 3664.330089 |
| 8 | 40.296372 | 0.367776 | 3658.754843 |
| 9 | 32.059010 | 0.353509 | 3664.534895 |
| 10 | 24.834639 | 0.369053 | 3660.554466 |
| ⋮ | ⋮ | ⋮ | ⋮ |
| 491 | 0.001001 | 0.494106 | 3664.147513 |
| 492 | 0.001221 | 0.354537 | 3664.147377 |
| 493 | 0.001058 | 0.483335 | 3664.147923 |
| 494 | 0.001287 | 0.382559 | 3664.147650 |
| 495 | 0.001253 | 0.370069 | 3664.147650 |
| 496 | 0.001061 | 0.430401 | 3664.147786 |
| 497 | 0.001199 | 0.375189 | 3664.147650 |
| 498 | 0.001219 | 0.362771 | 3664.147650 |
| 499 | 0.001005 | 0.434462 | 3664.147786 |
| 500 | 0.001140 | 0.414011 | 3664.147786 |




*Wykres 1:* Norma wektora reszt i rozwiązania dla pojedynczej precyzji.




*Wykres 2:* Norma wektora reszt i rozwiązania dla pojedynczej precyzji.

## 4. Wnioski

Korzystając z metody największego spadku rozwiązałem układ liniowy $A\boldsymbol{x} = \boldsymbol{b}$ dla dwóch różnych wektorów rozwiązań $\boldsymbol{x} = 0$ oraz $\boldsymbol{x} = 1$. W obu przypadkach liczba kroków była taka sama i wynosiła 131 dla podwójnej precyzji i 500 dla pojedynczej (z powodu ograniczenia iteracji). Z tego mogę wywnioskować, że postać wektora startowego nie wpływa na liczbę iteracji.

Zaobserwowałem, że metoda największego spadku działa wystarczająco szybko dla macierzy rzadkich o dużych wyniarach. Spowodowane jest to pominięciem operacji na elementach zerowych w tym algorytmie.