

Lab 6c, Programowanie matematyczne

Piotr Onyszczyk, gr. C

9 I 2022

1 Treść zadania

Celem zadania jest znalezienie punktu najbliższego od początku układu współrzędnych spełniającego ograniczenia równościowe.

$$\Omega = \{x \in R^n : Ax = b, A \in R^{m \times n}, m < n, r(A) = m, b \in R^m\} \quad (1)$$

W tym celu wykorzystana zostanie funkcja MATLABa *quadprog* oraz metoda zewnętrznej kwadratowej funkcji kary.

1.1 Postać zadania

Funkcja minimalizowana:

$$\min(x_1^2 + x_2^2 + \dots + x_n^2) \quad (2)$$

Ograniczenia równościowe:

$$A * x - b = 0 \quad (3)$$

Dodatkową własnością zadania jest to, że mamy spełniony warunek Karlina.

1.2 Dane testowe

Do testów należy użyte zostały losowe macierze i wektory z wartościami z przedziału $[-5; 5]$ oraz różnymi rozmiarami.

2 Algorytmy

2.1 Algorytm ZFK

Algorytm ZFK składa się z następujących kroków:

1. Obliczenie kroku początkowego, x_0 -dowolny

2. Definicja funkcji kary i jej pochodnej
3. Definicja funkcji pomocniczej i jej pochodnej
4. Warunki stopu: wartość funkcji kary, liczba iteracji
5. Obliczenie kolejnego kroku z wykorzystaniem *fminunc* lub **FR**
6. Powrót do 4.

2.2 Algorytm FR

Poprawiona i uproszczona wersja algorytmu z poprzedniego zadania. Wykorzystuje minimalizację analityczną.

3 Wyniki i wnioski

3.1 WKT

Nasze zadanie nie posiada nieliniowych ograniczeń równościowych ani żadnych nierównościowych, więc jest wypukłe. Z racji tego że zadanie jest wypukłe, WKT są wystarczające do bycia RO. We wszystkich testach WKT były spełnione, więc rozwiązanie zawsze jest RO.

3.2 Testy

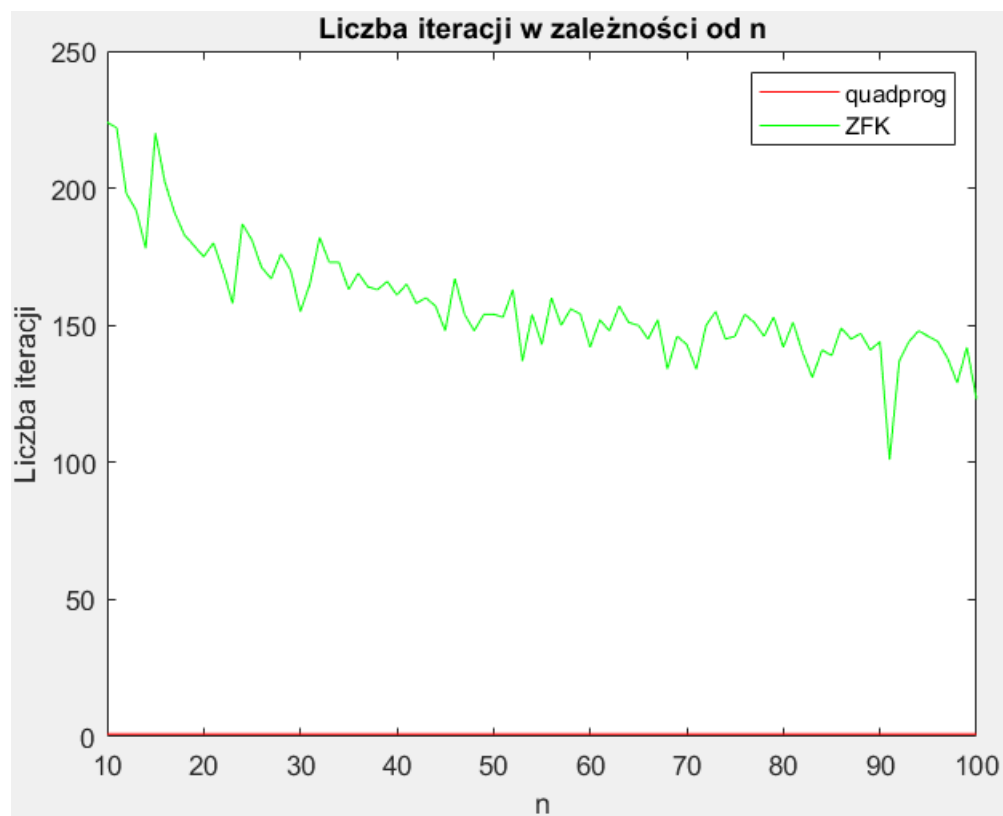
3.2.1 Porównanie *quadprog* i ZFK

W tym porównaniu przyjęta została wersja algorytmu **ZFK** z wykorzystaniem *fminunc*. Porównanie obu wersji znajduje się w dalszej części.

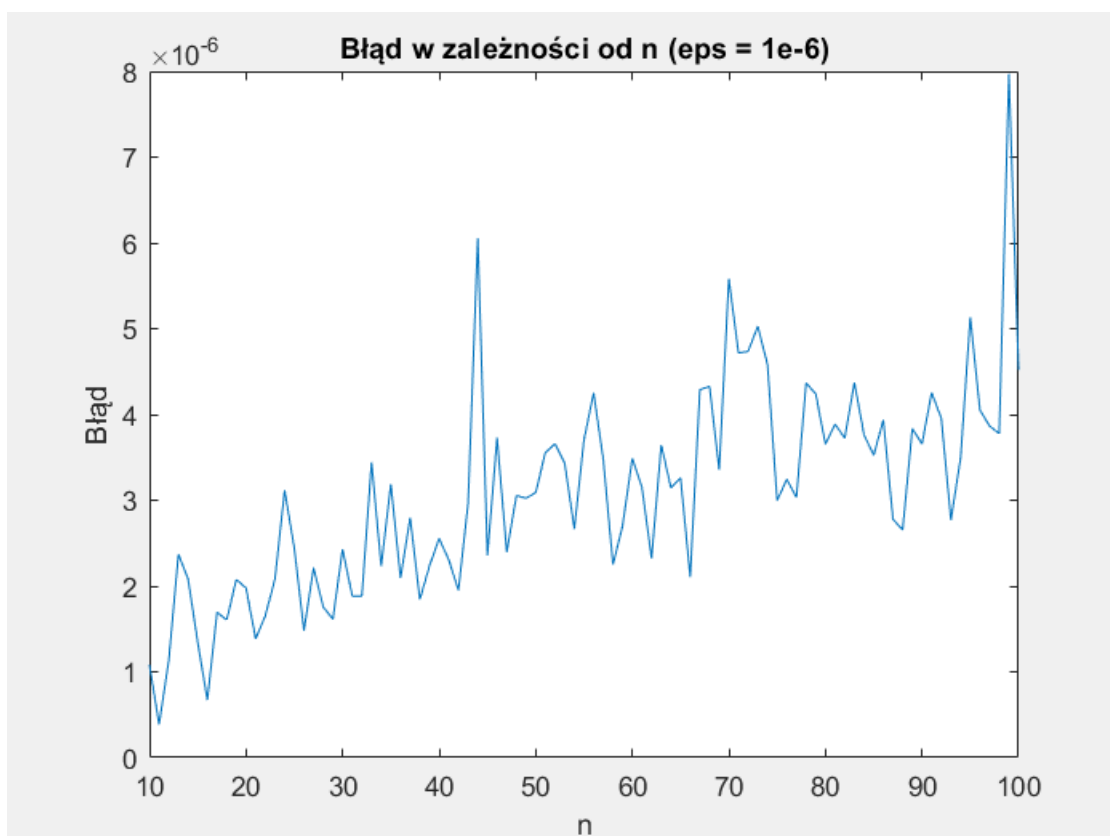
Zmienne 'n' Przy stałym epsilon (10^{-6}) oraz m (5) porównana została liczba iteracji obu algorytmów oraz błąd, czyli norma z różnicy rozwiązań (Wycinki 1 i 2). N zmieniało się od 10 do 100,

Można zauważyć kilka rzeczy:

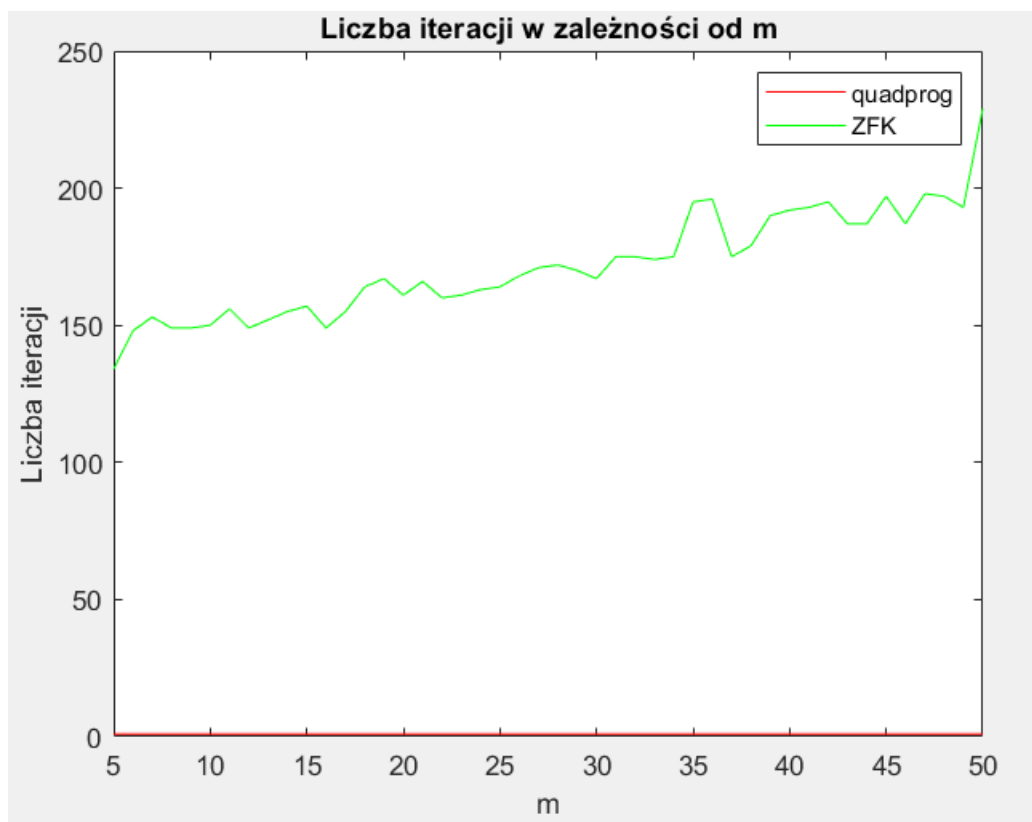
- Algorytm *quadprog* zawsze wykonuje jedną iterację
- Liczba iteracji ZFK utrzymuje się na podobnym poziomie. Ma nawet powolną tendencję malejącą, ale od pewnego momentu prawie nie maleje.
- Błąd ma tendencję rosnącą, ale cały czas utrzymuje się na poziomie epsilon



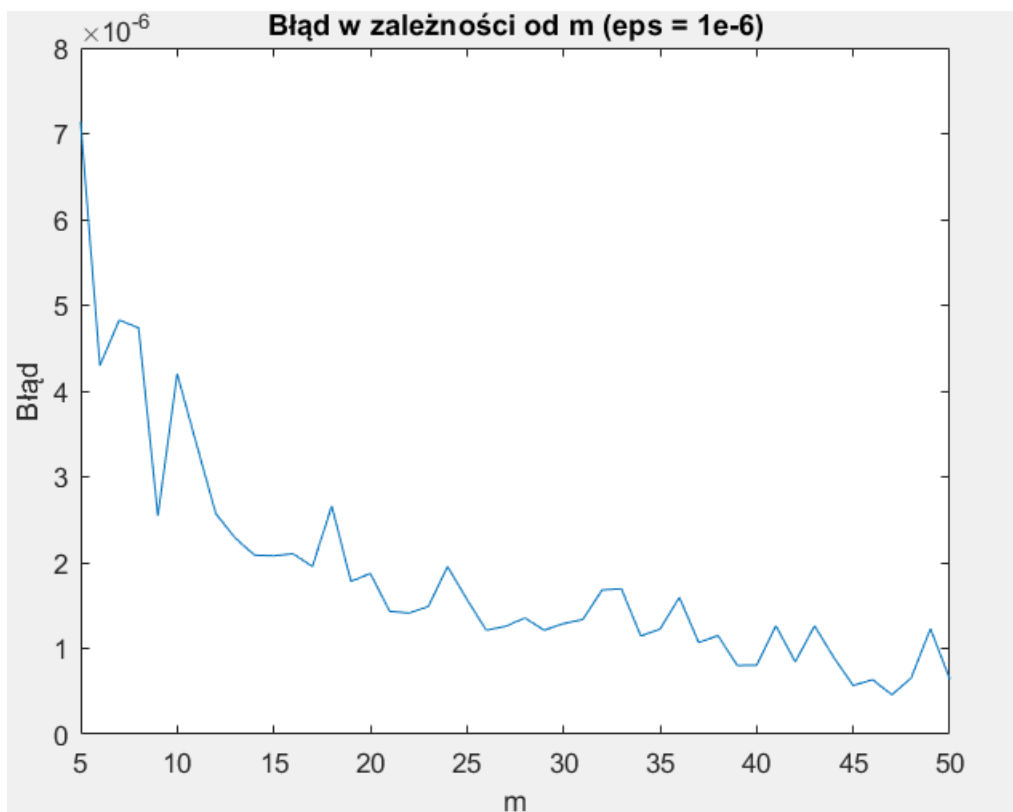
Wycinek 1: Liczba iteracji - zmienne n



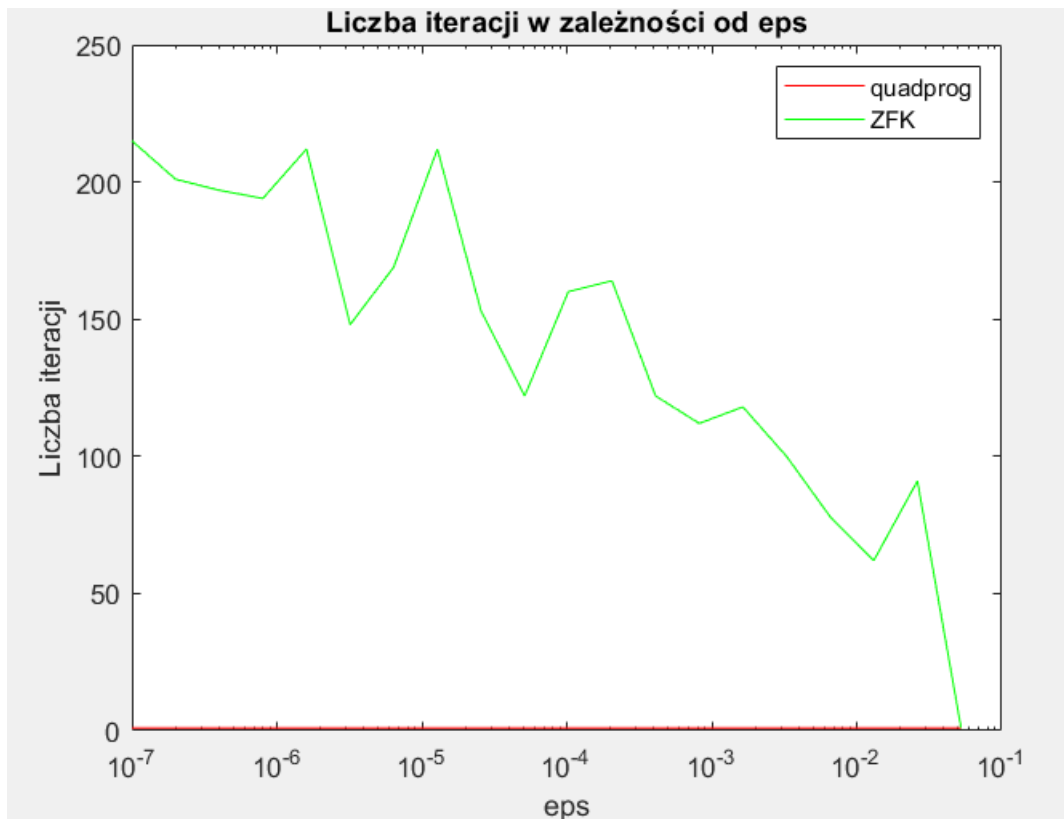
Wycinek 2: Błąd - zmienne n



Wycinek 3: Liczba iteracji - zmienne m



Wycinek 4: Błąd - zmienne m



Wycinek 5: Liczba iteracji - zmienne eps

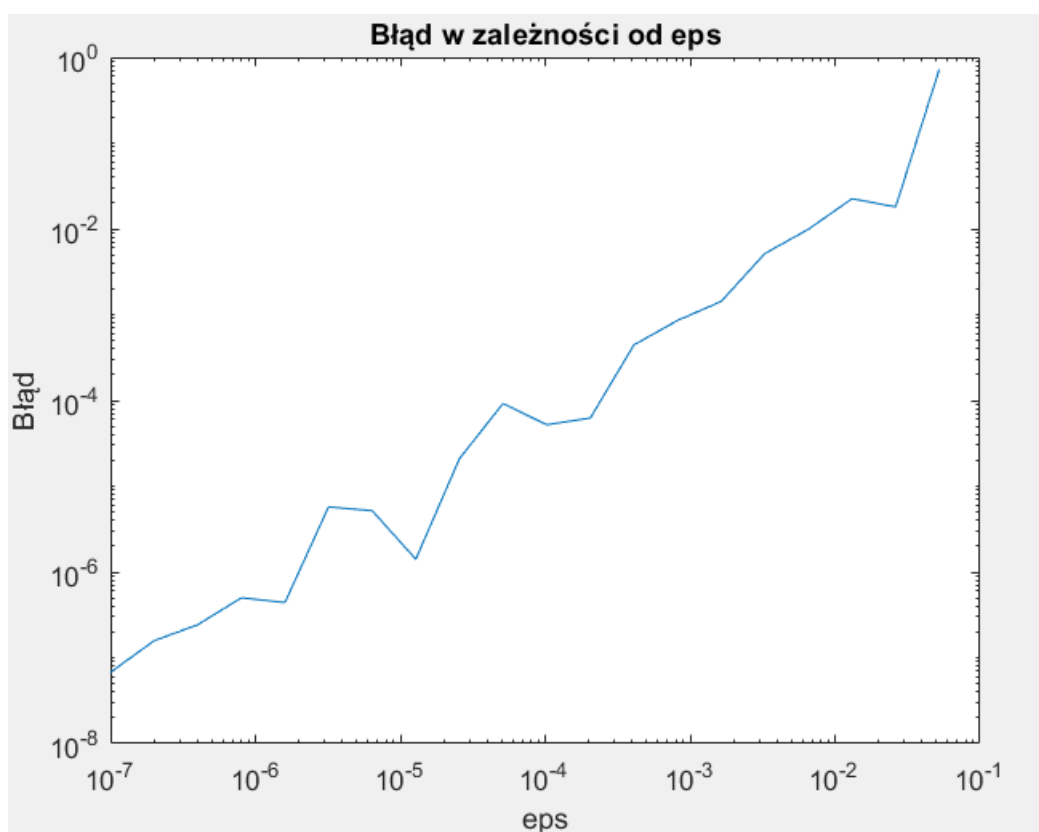
Zmienne 'm' Przy stałym epsilon (10^{-6}) oraz n (100) porównana została liczba iteracji obu algorytmów oraz błąd, czyli norma z różnicy rozwiązań (Wycinki 3 i 4). m zmieniało się od 5 do 50,

Można zauważyć kilka rzeczy:

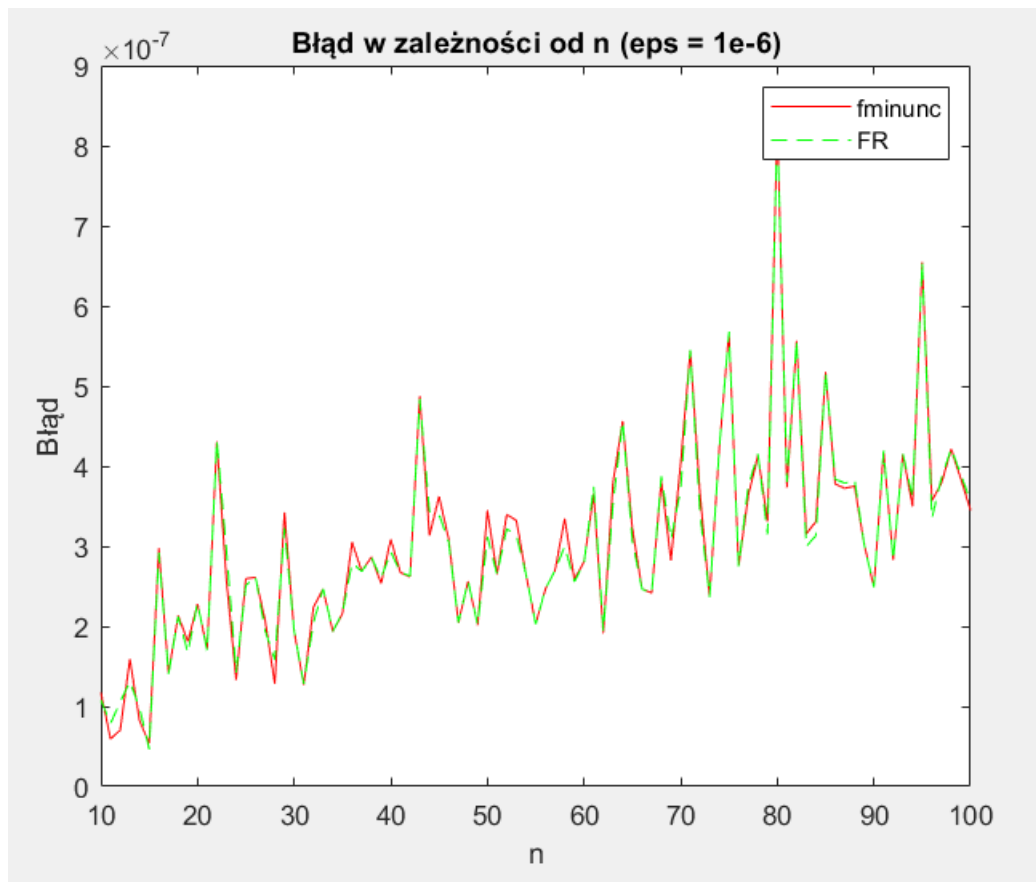
- Algorytm quadprog zawsze wykonuje jedną iterację
- Liczba iteracji ZFK utrzymuje się na podobnym poziomie. Ma powolną tendencję rosnącą.
- Błąd ma tendencję malejącą, ale cały czas utrzymuje się na poziomie epsilon

Zmienny 'epsilon' Przy stałym n (10) oraz m (5) porównana została liczba iteracji obu algorytmów oraz błąd, czyli norma z różnicy rozwiązań (Wycinki 5 i 6). Epsilon zmieniał się od 10^{-7} do 10^{-1} ,

Można zauważyć kilka rzeczy:



Wycinek 6: Błąd - zmienne eps



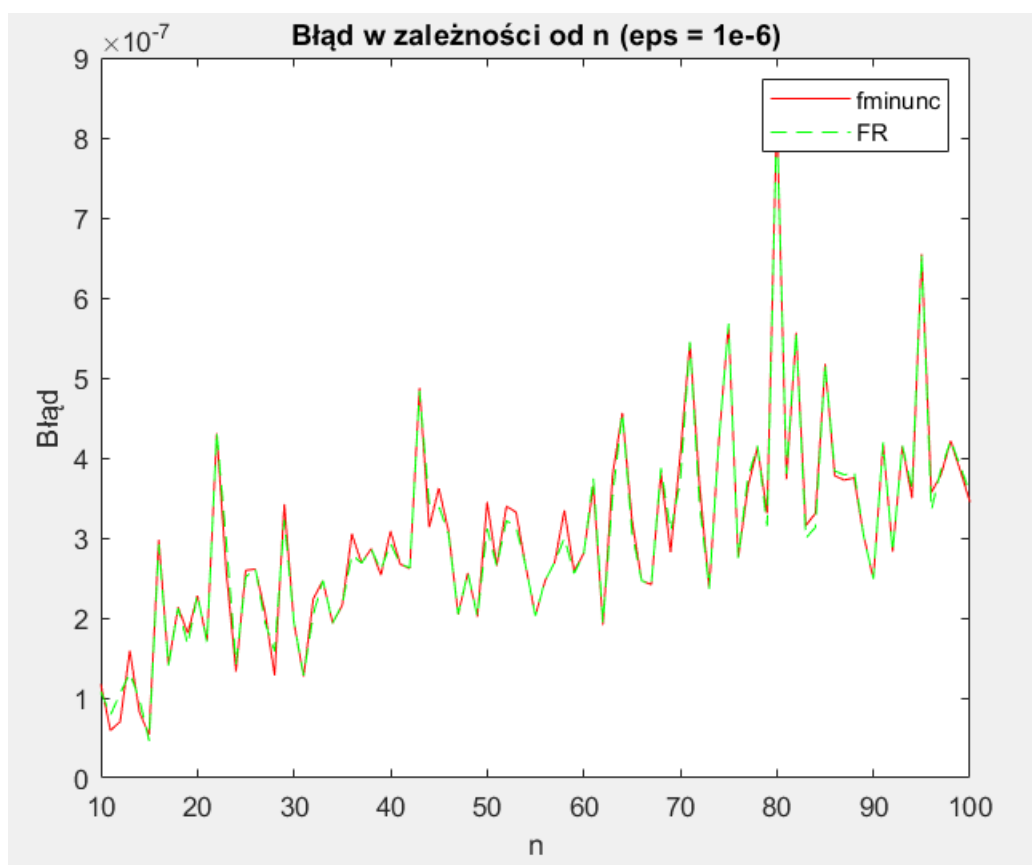
Wycinek 7: Liczba iteracji - zmienne n

- Algorytm quadprog zawsze wykonuje jedną iterację
- Liczba iteracji ZFK maleje szybko z rosnącym epsilon, co jest oczywistym zjawiskiem, bo zwiększamy dopuszczalny błąd
- Błąd utrzymuje się na poziomie epsilon

3.2.2 Porównanie $fminunc$ i FR

Porównanie to zostało zrobione tylko w zależności od ' n '. Pozostałe własności można zauważyć na podstawie poprzedniej sekcji. N zmieniało się od 10 do 100, $m = 5$, $eps = 10^{-6}$. Porównany jest algorytm **ZFK** z wykorzystaniem $fminunc$ oraz z wykorzystaniem **FR** (Wycinki 7 i 8).

W tym porównaniu prawie nie ma czego analizować. Oba wywołania zachowują się niemalże identycznie. Różnice są minimalne. Zarówno jeśli chodzi o liczbę iteracji, jak i o błąd.



Wycinek 8: Błąd - zmienne

3.3 Wnioski

Podsumowanie najważniejszych obserwacji:

- Algorytm *quadprog* zawsze wykonuje jedną iterację
- Zgodnie ze spostrzeżeniem wcześniej, jest to RO
- Różnica wyniku **ZFK** i *quadproga* utrzymuje się zawsze na poziomie epsilon
- Liczba iteracji nie zmienia się drastycznie w zależności od zadanych 'n' i 'm'
- **ZFK** z wykorzystaniem *fminunc* i **FR** zachowuje się prawie identycznie

4 Pliki

- Losowanie.m - skrypt losujący macierz A oraz wektor b zadanymi własnościami
- skrypt.m - skrypt wywołujący rozwiązanie funkcją *quadprog* oraz **ZFK**. Porównuje oba rozwiązania.
- ZFK.m - skrypt realizujący funkcję **ZFK**, wykorzystuje *fminunc* lub **FR**
- FR.m - implementacja algorytmu FR z krokiem analitycznym