Grupa G5

Warszawa, 14. stycznia 2020r.

# Metoda bisekcji wyznaczania zer funkcji

Paweł Koźmiński

Projekt nr 2

## 1. Treść zadania

Metoda bisekcji wyznaczania zer funkcji
$f(x)=a_0+a_1|T_1(x)|+\ldots+a_n|T_n(x)|$,
gdzie $T_{0,}T_{1,}\ldots,T_n$ są wielomianami Czebyszewa I-go rodzaju.

**Uwaga.** Nie należy sprowadzać wielomianów Czebyszewa do postaci naturalnej!
Do obliczania wartości funkcji $f$ należy wykorzystać związek rekurencyjny spełniany przez wielomiany Czebyszewa.

## 2. Opis metody

Metoda bisekcji jest jedną z iteracyjnych metod znajdowania zera funkcji jednej zmiennej. Opiera się ona na twierdzeniu Darboux:

> Niech $f:[a,b]\to\mathbb{R}$ będzie funkcją ciągłą. Jeżeli $f(a)\cdot f(b)<0$,
> to istnieje taki punkt *c* w przedziale $(a,b)$, dla którego $f(c)=0$.

Zatem może być ona zastosowana na ciągłych funkcjach o wartościach na pewnym przedziale ze zbioru liczb rzeczywistych, o ile na krańcach owego przedziału przyjmuje przeciwne znaki.
W trakcie działania algorytmu metody wyszukiwane są argumenty $x_k$ - potencjalne miejsca zerowe funkcji, poprzez połowienie (wyznaczenie średniej arytmetycznej z argumentów krańcowych) przedziału. Tak długo, jak funkcja nie przyjmuje 0 w określonym miejscu, określany jest nowy przedział tak, aby warunek różnych znaków wartości na krańcach przedziału był zachowany. Krańcami nowego przedziału są: wyznaczone $x_k$ oraz jeden z krańców poprzedniego przedziału - dokładnie dla jednego z nich powyższy warunek jest spełniony, zatem nowy przedział wyznaczany jest jednoznacznie.

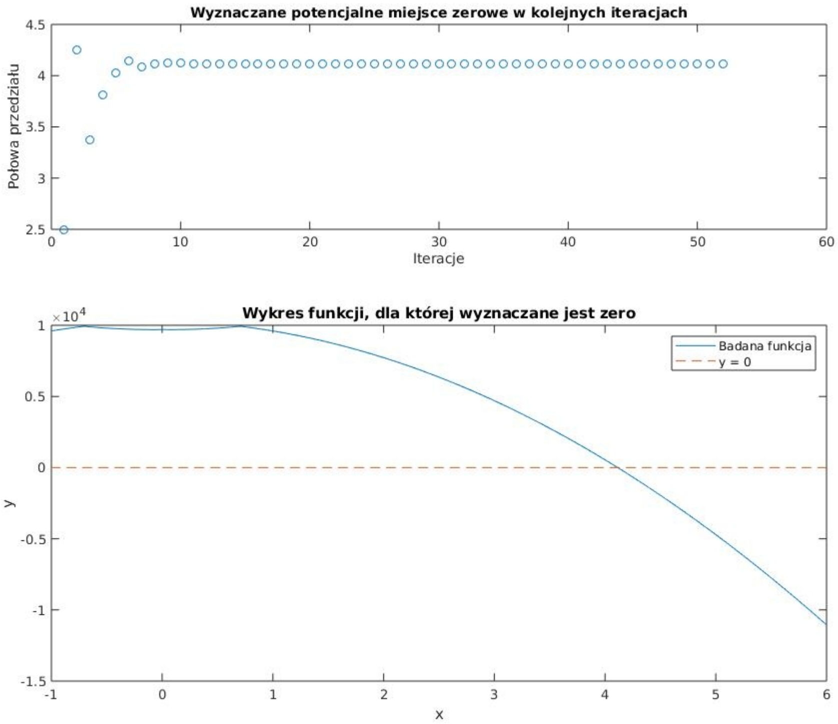
## 3. Opis działania programu obliczeniowego

W celu zaimplementowania powyższej metody w Matlabie stworzyłem dwie funkcje: *myBisection()* oraz pomocniczą *myCheb()*. *Funkcja myCheb()* przyjmuje na wejściu argument *x* oraz wektor współczynników $a_0, \ldots, a_n$. Zwraca wartość funkcji opisanej w treści zadania w punkcie *x*. Główna funkcja *myBisection()* zawiera w sobie serce zadania. Dla danego wektora współczynników oraz krańców przedziału wyznacza miejsce zerowe funkcji z wielomianami Czebyszewa. *MyBisection()* działa tak długo aż moduł wartości funkcji w wyznaczonym punkcie będzie mniejszy niż $10^{-12}$. Zwracane są dwie wartości: miejsce zerowe oraz liczba potrzebnych iteracji. Limitem liczby iteracji, po której działanie funkcji jest przerywane, jest 100 000. W trakcie dzialania, funkcja wyświetla także dwa estetyczne wykresy prezentujące wartości wyznaczonych potencjalnych miejsc zerowych w kolejnych iteracjach oraz wykres funkcji na poszukiwanym przedziale z zaznaczoną prostą $y=0$.

Dzięki implementacji ww. funkcji wyznaczenie zera jest dla użytkownika bardzo proste – jedyne co musi wykonać to wywołać funkcję *myBisection()* z wektorem współczynników oraz przedziałem jako argumentami. Na początku działania algorytmu, sprawdzane jest czy wartości na krańcach przedziału są różnych znaków, co jest kluczowym warunkiem powodzenia metody. W przypadku niepowodzenia, wyświetlany jest wykres funkcji. Powodzenie owego warunku, wraz z wiedzą o postaci funkcji, gwarantuje spełnienie wszystkich wymagań dotyczących zbieżności metody.

## 4. Przykłady obliczeniowe

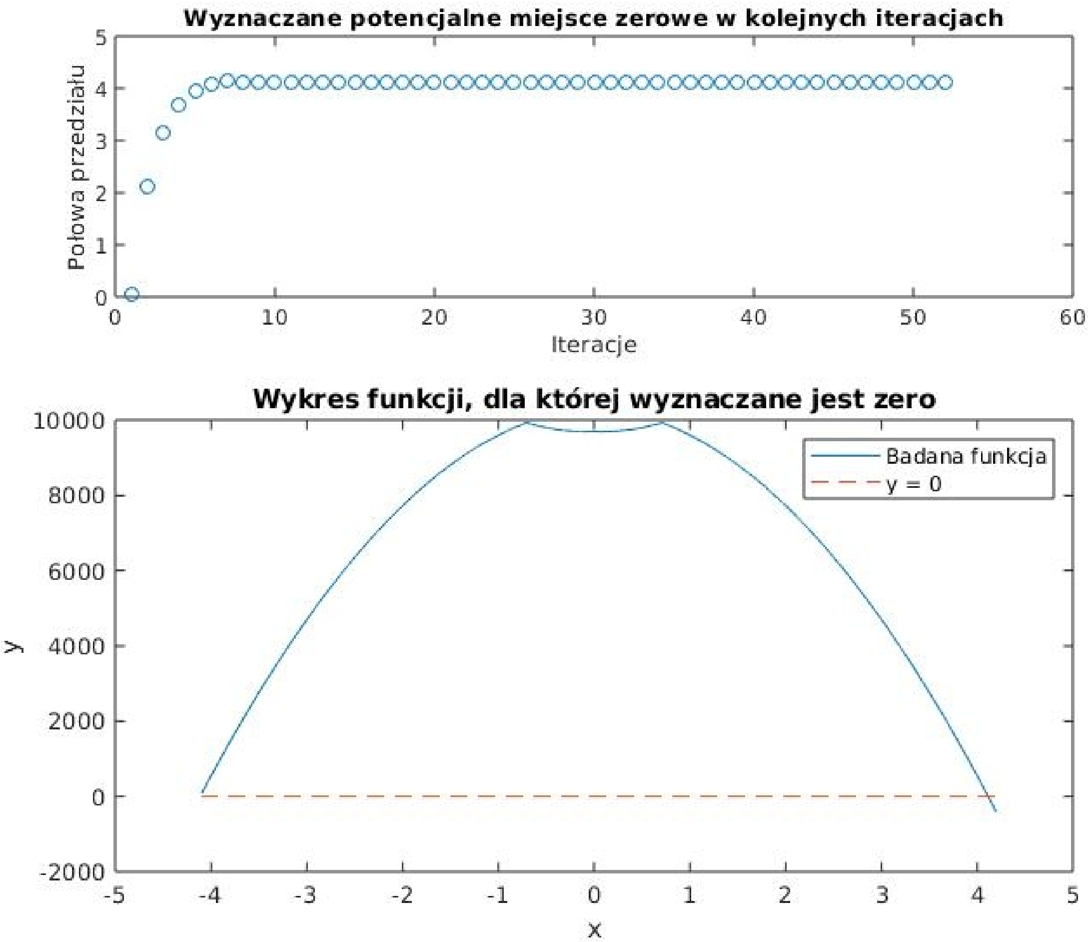
1. Wektor współczynników: *[10000, -100, -300, 1];*
   *Przedział: [-1, 6]*




*Wykres 1: Przykład 1*

Liczba potrzebnych iteracji: 52
Znalezione zero funkcji: 4.1140
Opis: Jako pierwszy przykład prezentuję funkcję z wyraźnie zaznaczonym jednym miejscem zerowym na zadanym przedziale. Do wyznaczenia zera program potrzebował wykonania 52 iteracji.

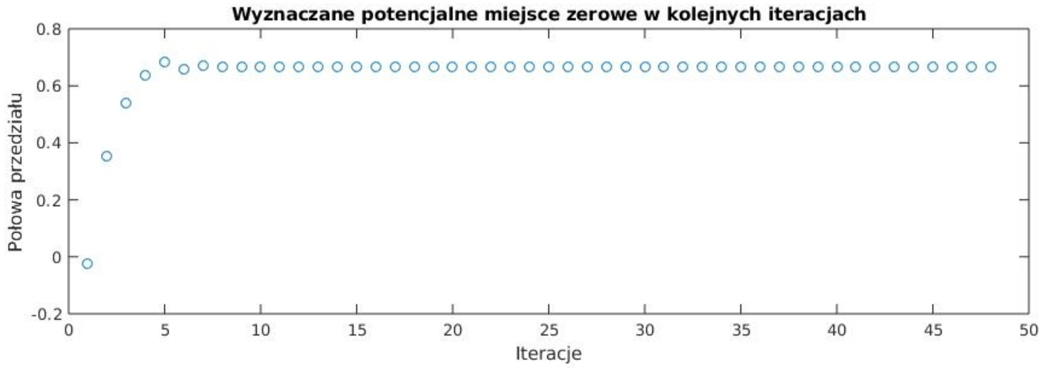
2. Wektor współczynników: *[10000, -100, -300, 1];*
   *Przedział: [-4.1, 4.2]*

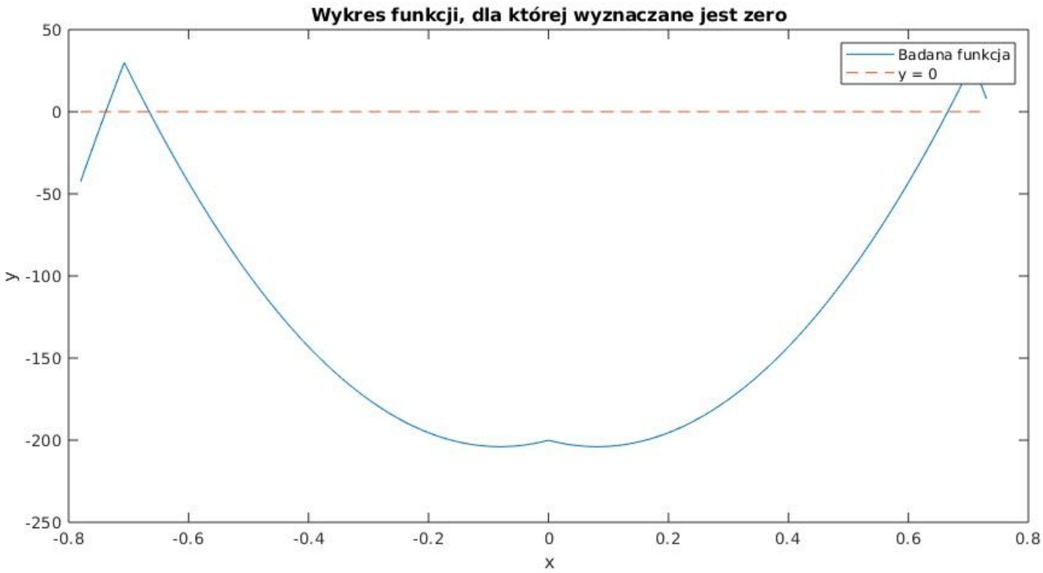



*Wykres 2: Przykład 2*

Liczba potrzebnych iteracji: 52
Znalezione zero funkcji: 4.1140
Opis: Metoda zastosowana na tej samej funkcji, jednak ze zmienionym przedziałem. Jak widać na wykresie, miejsce zerowe znajduje sie blisko prawego końca przedziału. Mimo to, metoda ponownie potrzebowała 52 iteracji. Z racji innego przedziału, inaczej wygląda pierwszy z wykresów: argumenty dążą do zera niemal ‘asymptotycznie’.

3. Wektor współczynników: *[100, -100, -300, 1];*
   *Przedział: [-0.78, 0.73]*

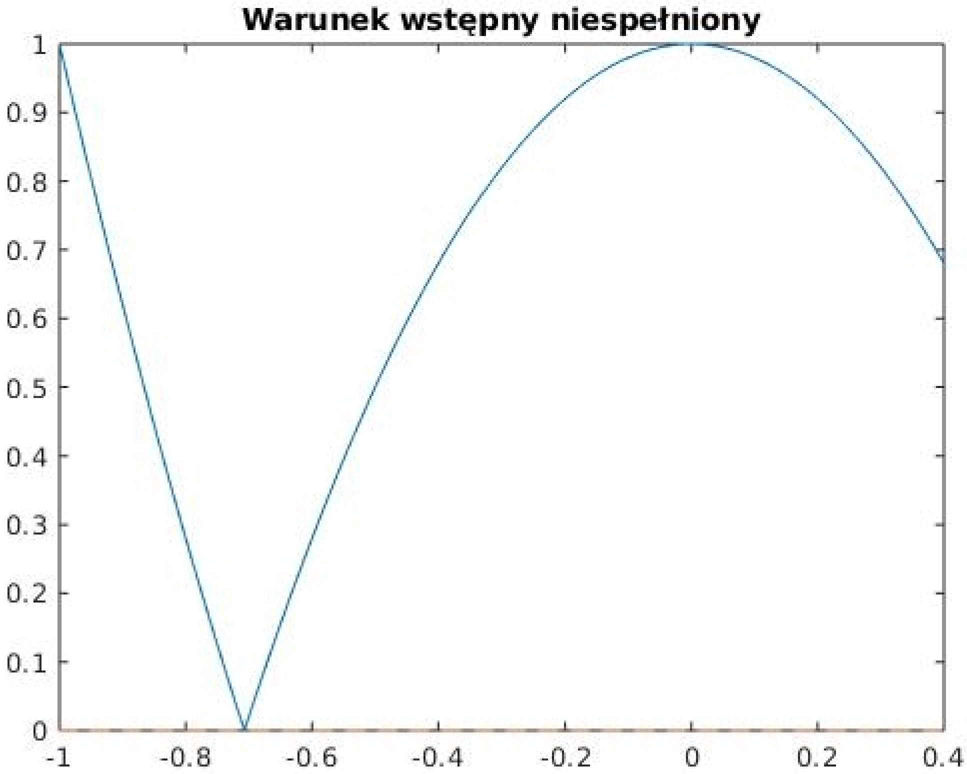



*Wykres 3: Przykład 3*

Liczba potrzebnych iteracji: 48
Znalezione zero funkcji: 0.6655
Opis: Zainspirowany przykładami nr 1 i 2, postanowiłem sprawdzić jak metoda zachowa się gdy na danym przedziale będzie się znajdować więcej niż jedno miejsce zerowe. Wyznaczone zero znajduje się z prawej strony wykresu z prostego powodu: lewa połówka została odrzucona już po pierwszej iteracji. Co ciekawe, tym razem potrzebne było mniej iteracji – 48.

4. Wektor współczynników: [0, 0, 1]
   Przedział: [-1, 0.4]



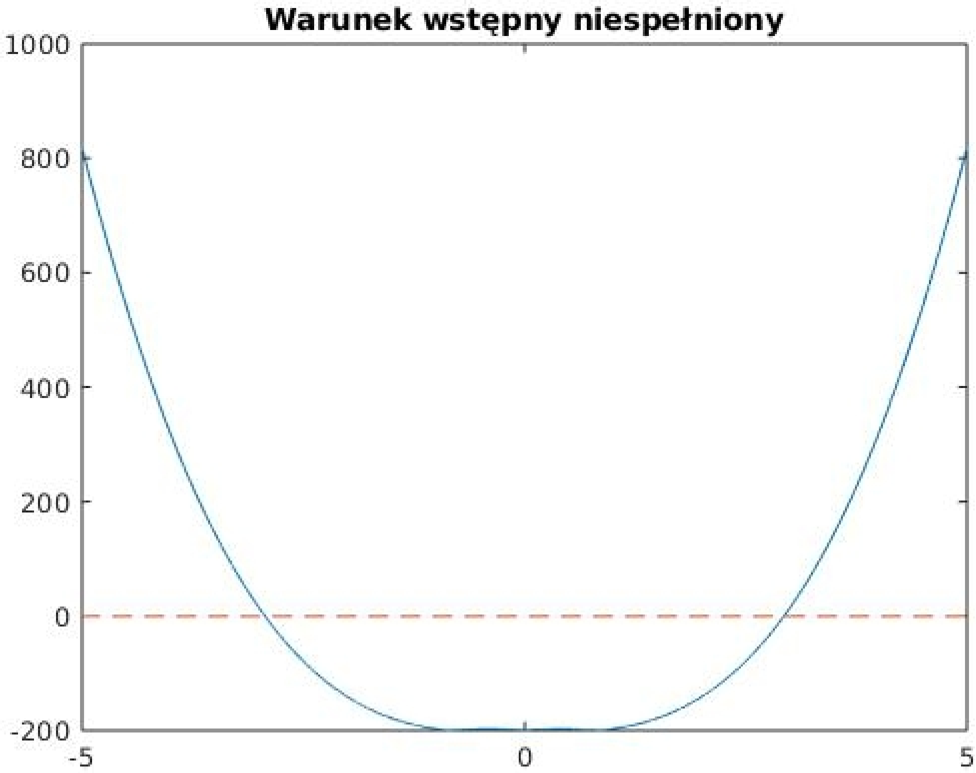

*Wykres 4: Przykład 4*

Liczba potrzebnych iteracji: nd.

Znalezione zero funkcji: brak

Opis: Niniejszy przykład obrazuje niewątpliwą wadę metody bisekcji. Badaną funkcją jest $y = |2x^2-1|$, która posiada dwa pierwiastki rzeczywiste. Niemniej nie przyjmuje ona wartości ujemnych, wobec czego zera nie mogą zostać wyznaczone przy pomocy twierdzenia Darboux.

5. Wektor współczynników: [-200, 0, 1, 2]
   Przedział: [-5, 5]




*Wykres 5: Przykład 5*

Liczba potrzebnych iteracji: nd.
Wyznaczone zero funkcji: brak
Opis: Fakt niepowodzenia metody bisekcji na danym przedziale, nie oznacza braku w nim miejsc zerowych. Powyższa ilustracja jest doskonałym kontrprzykładem.

# 5. Zalety i wady metody bisekcji

## 5.1 Zalety

- Metoda bisekcji, o ile spełnione są wstępne założenia, jest zawsze zbieżna (co istotne, wystarczy ciągłość jedynie owej funkcji)
- Każda iteracja dwukrotnie zmniejsza błąd obliczeń, wobec tego w łatwy sposób można określić błąd przybliżenia
- Prosta do zrozumienia idea

### 5.2 Wady

- Zbieżność metody jest stosunkowo wolna – liniowa
- Metoda nie ma zastosowania gdy wszystkie wartości są niedodatnie bądź nieujemne
- Źle zaimplementowana metoda mogłaby wyznaczyć zero funkcji $f(x)=\frac{1}{x}$ w punkcie $x$=0 (jednak funkcja ta nie jest w nim ciągła).

## 6. Dodatkowe uwagi

Jak oszacować liczbę iteracji potrzebną do uzyskania tolerancji błędu ε?

Po $k$ iteracjach rozmiar przedziału - jest to max. wartość błędu - maleje do $\varepsilon_0=\frac{b_0-a_0}{2^k}.(1)$

Po przekształceniu powyższej równości, otrzymujemy, że liczbę iteracji można oszacować przez:

$$k=\log_2(\frac{b_0-a_0}{\varepsilon_0})$$

Ponieważ ε można zapisać również jako $|x_k-a|$, gdzie α jest poszukiwanym zerem,
z $(1)$ otrzymujemy, że metoda bisekcji jest zbieżna liniowo, z ilorazem zbieżności $\frac{1}{2}$.

## Literatura

Źródła z których korzystałem:


- *materiały z wykładów p. Alicji Smoktunowicz*
- *pl.wikipedia.org*
- *Kaw, Autar; Kalu, Egwu (2008), Numerical Methods with Applications (1st ed.)*
- *wykład z Metod numerycznych dra hab. Piotra Fronczaka*
- *wazniak.mimuw.edu.pl*