

Projekt 1

Sprawozdanie

Daminika Dzeranhouskaya

335908

Temat: Wykorzystanie metody Newtona do znajdowania pierwiastków wielomianów Legendre’a w dziedzinie zespolonej.

Opis matematyczny metody

Metoda Newtona to jedna z najbardziej efektywnych metod numerycznych stosowanych do znajdowania pierwiastków równań nieliniowych. W przypadku wielomianów Legendre’a, iteracje metody Newtona dla wartości zespolonych są prowadzone zgodnie z następującym wzorem:

$$z_{n+1} = z_n - \frac{w_n(z)}{w'_n(z)}$$

gdzie:

- $w_n(z) = \sum_{k=0}^n a_k P_k(z)$ – wielomian Legendre’a
- $w'_n(z)$ – pochodna wielomianu $w_n(z)$
- z_n – punkt startowy iteracji

Proces iteracyjny jest kontynuowany do momentu spełnienia warunku zbieżności:

$$|z_{n+1} - z_n| < \varepsilon$$

Gdzie ε to zdefiniowana tolerancja błędu.

Wielomiany Legendre’a

Wielomiany Legendre’a są definiowane rekurencyjnie:

$$\begin{aligned} P_0(x) &= 1, & P_1(x) &= x, & (n+1)P_{n+1}(x) \\ & & & = (2n+1)xP_n(x) - nP_{n-1}(x), & n = 1, 2, \dots \end{aligned}$$

Opis programu

Funkcja legendre_polynomial

Działanie:

Funkcja `legendre_polynomial` oblicza wartości wielomianów Legendre'a $P_n(z)$ oraz ich pochodnej w zadanym punkcie z . Wykorzystuje rekurencyjną definicję wielomianów Legendre'a dla optymalizacji obliczeń.

Parametry wejściowe:

- **n** – stopień wielomianu Legendre'a,
- **x** – wartość (lub wartości), dla których obliczane są wielomiany i ich pochodne.

Parametry wyjściowe:

- **wn** – wartość wielomianu Legendre'a stopnia n w punkcie x ,
- **wn_prime** – wartość pochodnej wielomianu Legendre'a stopnia n w punkcie x .

Funkcja metoda_newtona**Działanie:**

Funkcja `metoda_newtona` iteracyjnie oblicza pierwiastki wielomianu Legendre'a $P_n(z)$ przy użyciu **metody Newtona** dla zadanych parametrów, takich jak punkt początkowy, tolerancja błędu i maksymalna liczba iteracji.

Parametry wejściowe:

- **n** – stopień wielomianu Legendre'a,
- **z0** – punkt początkowy iteracji (liczba zespolona),
- **tol** – tolerancja błędu, określająca kryterium stopu,
- **max_iter** – maksymalna liczba iteracji.

Parametry wyjściowe:

- **root** – obliczony pierwiastek wielomianu Legendre'a $P_n(z)$,
- **iterations** – liczba wykonanych iteracji,
- **trajectory** – wektor zawierający kolejne przybliżenia z w trakcie iteracji.

Funkcja newton_legendre_gui**Działanie:**

Funkcja `newton_legendre_gui` tworzy **graficzny interfejs użytkownika (GUI)**, który pozwala na wizualizację **trajektorii iteracji metody Newtona** dla wielomianów Legendre'a. Dodatkowo oblicza i porównuje wyniki metody Newtona oraz funkcji wbudowanej **`fsolve`**, przedstawiając kluczowe informacje numeryczne. Funkcja zabezpiecza się przed niepoprawnym formatem danych wejściowych i wyświetla komunikat o błędzie.

Uwaga: Funkcja `fsolve` wymaga zainstalowanego **Optimization Toolbox**.

Parametry wejściowe:

Funkcja nie posiada jawnych parametrów wejściowych. Wartości są wprowadzane przez użytkownika za pomocą GUI.

Wczytywane są następujące dane od użytkownika:

- **Stopień wielomianu nnn ,**
- **Punkt startowy z_0** (liczba zespolona),
- **Maksymalna liczba iteracji,**
- **Tolerancja błędu.**

Interfejs zawiera również przycisk **Start**, który uruchamia obliczenia.

Parametry wyjściowe:

- **Graficzny interfejs użytkownika (GUI)** z wizualizacją trajektorii iteracji,
- **Wyniki numeryczne** wyświetlone w etykietach GUI:
 - Pierwiastek obliczony metodą Newtona,
 - Pierwiastek obliczony funkcją `fsolve`,
 - Liczba iteracji potrzebnych do osiągnięcia zbieżności,
 - Błąd bezwzględny między wynikami Newtona i `fsolve`,
 - Błąd względny między wynikami Newtona i `fsolve`.

Funkcja `generate_test_cases`

Działanie:

Funkcja `generate_test_cases` służy do analizy i porównania wyników **metody Newtona** z wynikami uzyskanymi za pomocą wbudowanej funkcji MATLAB-a **`fsolve`** dla wielomianów Legendre'a o różnych stopniach i punktach startowych. Wyniki obliczeń są prezentowane w formie tabelarycznej, zawierającej błędy numeryczne, liczbę iteracji oraz przybliżone rozwiązania. Jeżeli metoda Newtona lub `fsolve` nie zbiegną, wynik jest zastępowany wartością **NaN**, aby zaznaczyć brak rozwiązania.

Parametry wejściowe:

Funkcja nie przyjmuje parametrów wejściowych – punkty testowe oraz inne ustawienia są zdefiniowane wewnątrz funkcji:

- **test_cases** – tablica zawierająca:
 - **Stopień wielomianu n ,**
 - **Punkt startowy z_0 ,**
 - **Tolerancja błędu ϵ ,**
 - **Maksymalna liczba iteracji max_iter .**

Parametry wyjściowe:

Funkcja wyświetla tabelę w konsoli MATLAB-a zawierającą:

- Stopień wielomianu n ,
- Punkt startowy z_0 ,
- Tolerancję błędu ϵ ,
- Maksymalną liczbę iteracji,
- Liczbę iteracji wykonanych przez metodę Newtona,
- Wynik metody Newtona (część rzeczywista i urojona),
- Wynik funkcji fsolve (część rzeczywista i urojona),
- Błąd bezwzględny,
- Błąd względny.

Pomocnicza funkcja `legendre_polynomial1`

Działanie:

Funkcja `legendre_polynomial1` oblicza wartość wielomianu Legendre’a w zadanym punkci. Jest **prostsza wersją** funkcji `legendre_polynomial`, która może być używana w sytuacjach, gdzie **pochodna nie jest potrzebna**

Parametry wejściowe:

- **n** – stopień wielomianu Legendre’a,
- **x** – wartość (lub wektor wartości), dla których obliczana jest wartość wielomianu $P_n(z)$

Parametry wyjściowe:

- **polynomial** – wartość wielomianu Legendre’a dla zadanego stopnia n i punktu x .

Wyniki

Przypadki testowe dla metody Newtona z porównaniem do funkcji fsolve:

n	z0	Tolerancja	MaxIter	Iteracje Newtona	Newton (Re + Im)	fsolve (Re + Im)	Błąd bezwzgl.	Błąd względny
2	100.0000 +	100.0000i 1.0e-06	50	12	0.5774 + -0.0000i	0.5774 + 0.0000i	4.26e-10	7.37e-10
3	-1.5000 +	2.0000i 1.0e-06	100	10	0.0000 + 0.0000i	-0.7746 + 0.0000i	7.75e-01	1.00e+00
5	0.5000 +	0.5000i 1.0e-06	100	6	0.5385 + 0.0000i	0.5385 + -0.0000i	1.47e-09	2.73e-09
7	0.1000 +	3.0000i 1.0e-08	100	15	0.0000 + 0.0000i	0.0000 + 0.0000i	9.27e-15	1.00e+00
8	-2.0000 +	0.5000i 1.0e-08	100	15	-0.9603 + 0.0000i	-0.9603 + -0.0000i	8.21e-17	8.54e-17
85	25.0000 +	25.0000i 1.0e-02	900	322	0.7720 + 0.3205i	25.0000 + 25.0000i	3.46e+01	9.78e-01
100	50.0000 +	50.0000i 1.0e-12	1000	497	0.6931 + -0.0000i	50.0000 + 50.0000i	7.02e+01	9.93e-01
100	0.0010 +	0.0000i 1.0e-12	1000	5	0.1092 + 0.0000i	0.1092 + 0.0000i	4.88e-15	4.47e-14

Do przeanalizowania starałam się dobrać dobre przykłady, które by pokazały od czego zależy szybkość zbieżności metody Newtona. Przedstawione przykłady pokazują, że kluczowy wpływ na liczbę iteracji ma nie stopień wielomianu, tylko odległość punktu od którego startujemy (z_0) do pierwiastka wielomianu. Szczególnie dobrze to widać w przykładzie gdzie wielomian ma 10 stopień. W pierwszej sytuacji z_0 jest bardzo daleko od pierwiastka i potrzebuje 497 iteracji żeby znaleźć pierwiastek. Ale mimo dużej liczbie iteracji błędy numeryczne są dość duże. W przeciwnej sytuacji, kiedy z_0 był blisko pierwiastku, były potrzebne tylko 5 iteracji, i dodatkowo błędy numeryczne są wystarczająco małe. Z kolei dla wielomianu drugiego stopnia też podałam z_0 dalekie od pierwiastka i w wyniku były potrzebne 12 iteracje. To się nie wydaje małym, wręcz naodwrot niemałą liczbą dla wielomianu 2 stopnia. Ale w porównaniu z wielomianem 10 stopnia błędy numeryczne są znacząco mniejsze, w sumie jak i liczba iteracji.

Także na błędy numeryczne znacząco wpływa, szczególnie dla wielomianów z dużym stopniem i odległością od pierwiastka, wybrana tolerancja i ograniczenia maksymalnej liczby iteracji. W moich przykładach większy wpływ okazała tolerancja.

Podsumowując można dojść do wniosku, że metoda Newtona jest efektywna dla niskich stopni wielomianów, dobrze dobranych punktów startowych oraz dobrze zdefiniowanych parametrów, takich jak tolerancja błędu i maksymalna liczba iteracji.

Wyniki przedstawione w GUI

Punkt zielonego koloru – to pierwiastek obliczony za pomocą wbudowanej funkcji fsolve.

Czerwonego koloru – obliczony zaimplementowaną przeze mnie metodą Newtona

Czasem czerwonego punktu nie widać, bo nakrywa się zielonym.

(Dodatkowo w niektórych przykładach chciałam pokazać jaką ciekawą trajektorię mogą mieć iteracje.)







