Metody Halleya i Homeiera

Jakub Mieczkowski

June 6, 2022

Spis treści

1 Wstep

W ramach zadania projektowego zbadałem metody iteracyjne Halleya i Homeiera, służace do wyznaczania miejsc zerowych wielomianów o współczynnikach naturalnych. Jak doskonale wiadomo ze szkoły średniej, dla wielomianów stopnia wyższego niż 3 znalezienie rozwiazań jest bardzo trudne, a czasem wiec niemożliwe. W raporcie pokaże, że metody iteracyjne Halleya oraz Homeiera doskonale sprawdzaja sie dla wielomianów: sa bardzo szybkie, prawie zawsze zbieżne oraz nie potrzebuja zbyt wielu iteracji do zakończenia działania.

2 Opis metod

2.1 Metoda Halleya

Metoda Halleya jako przybliżenie x_{k+1} przyjmuje miejsce zerowe hiperboli przybliżajace funkcje $f(x)$ w punkcie x_k. Jest ona określona nastepujacym wzorem:

$x_{k+1} = x_k - \frac{f(x_k))}{f'(x_k)+f"(x_k)\frac{f(x_k)}{2f'(x_k)}}$, gdzie $k = 0, 1, ...$.

Gdzie warunkiem stopu jest $|x_{k+1} - x_k| < d$. Za d przyjmuje bazowo 10^{-4} ale zmienie je podczas jednego testu.

2.2 Metoda Homeiera

Metode Homeiera implementujemy nastepujacym wzorem:

$y_k = x_k - \frac{f(x_k}{f'(x_k)}$,

$x_{k+1} = x_k - \frac{1}{2}f(x_k)(\frac{1}{f'(x_k)} + \frac{1}{f'(y_k)})$.

Warunek stopu przyjmujemy za każdym razem identyczny jak w metodzie Halleya, aby można było porównywać ze soba te metody.

3 Przykłady obliczeniowe

3.1 Przykład, w którym metoda jest rozbieżna

Jak wspomniałem wcześniej, obie metody sa zbieżne prawie zawsze i bardzo ciežko znaleźć przykład pokazujacy, że zbieżność nie zachodzi zawsze. Jednakże, jeżeli wielomian ma miejsca zerowe jedynie zespolone, a za punkt poczatkowy przyjmiemy liczbe z osi rzeczywistej, to nie otrzymamy poprawnego wyniku. Jeśli weźmiemy wielomian $x^4 + 16$, który nie ma rozwiazań rzeczywistych, a za przybliżenie poczatkowe weźmiemy liczbe 5, to nie otrzymamy rozwiazań. Obie funkcje zatrzymuja sie po 1000 iteracji, co jest maksymalna możliwa wartościa w moim programie i zwracaja różne liczby z osi rzeczywistej zupełnie nie powiazane z prawdziwymi rozwiazaniami (nie jest to cześć rzeczywista żadnego z rozwiazań. Natomiast, jeśli zmienie punkt poczatkowy na $5 + 0.01i$ to metoda zwraca poprawny wynik.

3.2 Różne wyniki

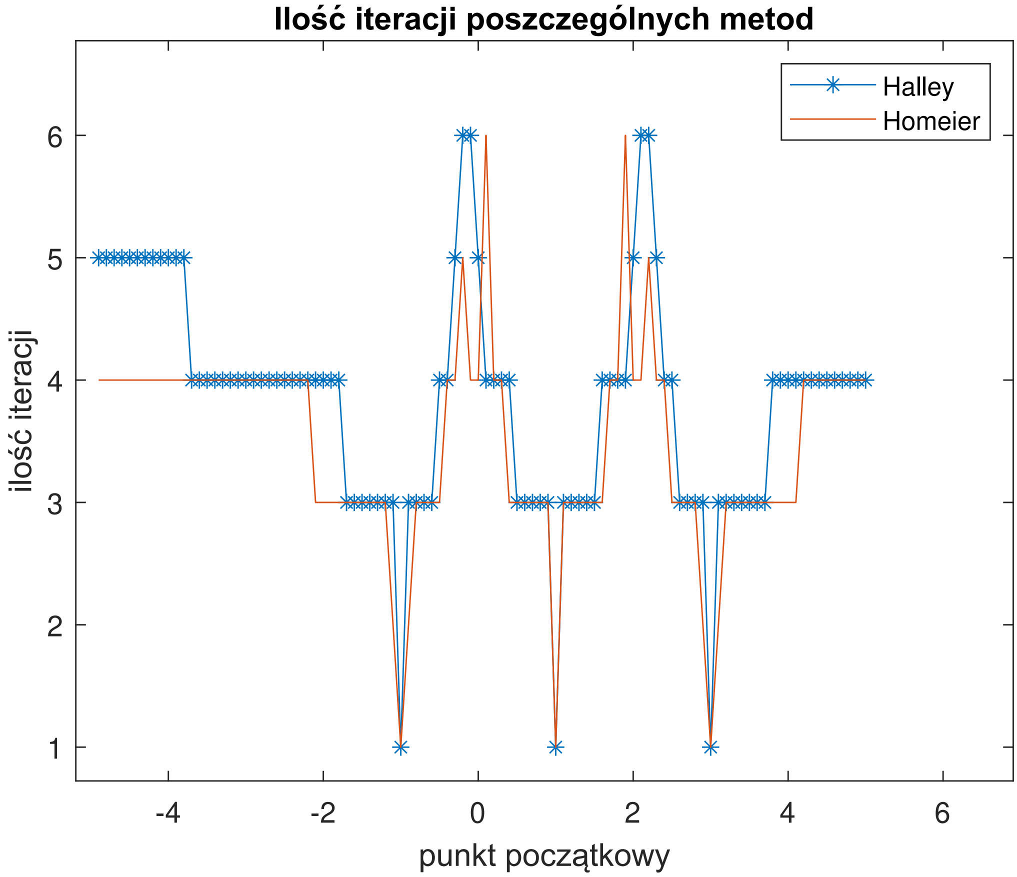
Czesto obydwie metody zwracaja te miejsca zerowe, jednakże dla niektórych punktów poczatkowych metody potrafia zwrócić różne miejsca zerowe. Przykładowo weźmy wielomian $x^3 - 3x^2 - x + 3$ oraz przybliżenie poczatkowe $x_0 = -0.1$. Wtedy metoda Halleya zwraca $x = 1$, zaś metoda Homeiera - $x = 3$. Jednakże, jeśli zmienimy wartość $x_0 = 2$ to obydwie metody zwracaja miejsce zerowe $x = 1$. A wiec metody nie wyznaczaja miejsca zerowego, które jest najbliższe co do modułu i czasem wynik jest inny.

3.3 Szybkość działania

Jak napisałem we wstepie obie metody działaja w matlabie bardzo szybko. Dzieje sie tak, ponieważ ich implementacja jest dość optymalna obliczeniowo oraz dlatego, że te metody sa bardzo szybko zbieżne (co pokaże potem). Badania prowadziłem za pomoca funkcji *tictok* w matlabie na różnych punktach poczatkowych oraz wielomianach. Czas wykonania oscylował miedzy 10^{-3} a 10^{-5} sekund. Ponadto, w niektórych przykładach metoda Halleya jest odrobine szybsza, zaś w niektórych metoda Homeiera troche szybciej zwraca wynik.

3.4 Ilość iteracji

W tej cześci pokaże na przykładzie jak zmienia sie ilość iteracji w obu metodach lekko manipulujac punktem poczatkowym. Zbadam to ponownie na przykładzie wielomianu $x^3 - 3x^2 - x + 3$ w przedziale $[-5, 5]$.



Pierwszy fakt, na który chciałbym zwrócić uwage to maksymalna ilość iteracji. W tych przykładach wynosi ona 6. Można zauważyć również, że obydwie metody zachowuja sie podobnie - przyjmuja jedna iteracje w miejscach zerowych, a najwiecej - pomiedzy miejscami zerowymi. Właśnie w okolicy zera i dwójki liczba iteracji sie zdecydowanie zmienia - może to być spowodowane tym, że w tej okolicy zmienia sie zwrace miejsce zerowe funkcji.

4 Podsumowanie

W powyższym raporcie pokazałem, że metody Halleya i Homeiera sa świetnymk metodami wyznaczania zer wielomianów. Bardzo szybko działaja, sa zbieżne prawie zawsze, a ich implementacja ma mała złożoność. Porównujac je wzgledem siebie, nie jestem w stanie ocenić czy któraś jest lepsza - w zależności od przykładu szybkość oraz ilość iteracji lekko sie różnia, ale sa do siebie bardzo zbliżone