20 października 2019

Maciej Salwowski
298904
G3

Wizualizacja szybkości zbieżności metody Halleya

Projekt nr 1

1 Wstęp

W poniższym dokumencie zaprezentowany jest skrypt języka Matlab, za pośrednictwem którego można otrzymać wizualizację szybkości zbieżności metody rekurencyjnego wyszukiwania miejsc zerowych wielomianu, zwanej metodą Halleya, w ciele liczb zespolonych. Algorytm ten korzysta z pomocniczej metody Hornera oraz tworzy siatkę losowo generowanych punktów przestrzeni zespolonej.

2 Opis metody

Metoda Halleya polega na rekurencyjnym przybliżaniu pierwiastka wielomianu. Warunkiem tego algorytmu jest ciągłość funkcji $f(x)$ oraz jej pochodnych $f'(x)$ i $f''(x)$. Algorytm ten korzysta bezpośrednio z metody Newtona

$$x_{n+1} - x_n = -\frac{f(x_n)}{f'(x_n)},$$

po przekształceniu

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)},$$

oraz z rozwinięcia funkcji $f(x)$ w szereg Taylora dla $f(x) = 0$, przy zaokrągleniu do drugiego stopnia rozwinięcia, to znaczy

$$0 \approx f(x_n) + f'(x_n)(x_{n+1} - x_n) + \frac{1}{2}f''(x_n)(x_{n+1} - x_n)^2.$$

Zatem przyjmując x_n jako punkt startowy przybliżania, rekurencja zachodzi według wzoru

$$x_{n+1} = x_n - \frac{2 \cdot f(x_n) \cdot f'(x_n)}{2 \cdot (f'(x_n))^2 - f(x_n) \cdot f''(x_n)}.$$

3 Opis programu obliczeniowego

3.1 Funkcja horner

```
function [f,df,ddf] = horner(x,x0)
%funkcja obliczajaca za pomoca algorytmu Hornera wartosc drugiej pochodnej
%wielomianu w danym punkcie
%parametry wejsciowe:
%x - wspolczynniki wielomianu
%x0 - punkt, ktorego wartosc szukamy
%parametry wyjscia:
%ddf - wartosc drugiej pochodnej wielomianu o wspolczynnikach x w punkcie x0
%df - wartosc pierwszej pochodnej wielomianu o wspolczynnikach x w punkcie x0
%f - wartosc wielomianu o wspolczynnikach x w punkcie x0

m = length(x);

f = 0;
df = 0;
ddf = 0;

for i = 1:m-2
    f = f*x0 + x(i);
    df = df*x0 + f;
    ddf = ddf*x0 + df;
end

f = f*x0 + x(m-1);
df = df*x0 + f;
f = f*x0 + x(m);
ddf = 2*ddf;
```

W tym konkretnym przypadku w celu skorzystania z metody Halleya przy zadanym wielomianie oraz punkcie startowym, niezbędny był numeryczny sposób na obliczanie wartości zarówno wielomianu w danym punkcie $f(x_0)$, jak i jego pierwszej oraz drugiej pochodnej $f'(x_0)$ i $f''(x_0)$. Do wyżej wspomnianych obliczeń wykorzystany został algorytm Hornera, pozwalający, przy stosunkowo niewielkiej złożoności, uzyskać niezbędne wartości.

3.2 Funkcja halley

```
function l = halley(x,x0)
%funkcja obliczajaca za pomoca algorytmu Halleya najblizsze miejsce zerowe
%wielomianu przy zadanym punkcie
```

```
%parametry wejsciowe:
%x - wspolczynniki wielomianu
%x0 - zadany punkt
%parametr wyjscia:
%l - liczba iteracji potrzebna do uzyskania miejsca zerowego

x1 = x0 + 1;
l = 0;
e = 10^(-10);

while ((abs(x1 - x0)/abs(x0)) > e)
    if l ~= 0
        x0 = x1;
    end
    [f,df,ddf] = horner(x,x0);
    x1 = x0 - (2*f*df)/((2*(df)^2) - f*ddf);
    l = l + 1;
end
```

Rekurencja trwa dopóki różnica pomiędzy kolejnymi przybliżeniami, w stosunku do wartości poprzedniego przybliżenia, jest większa od ustalonego błędu e, zgodnie z warunkiem

$$\frac{|x_{n+1} - x_n|}{|x_n|} > e.$$

Błąd przyjęty do obliczeń w tym programie równy jest 10^{-10}.

3.3 Generowanie losowej siatki punktów oraz wizualizacja szybkości zbieżności

```
%skrypt rysujacy wizualizacje szybkosci zbieznosci metody Halleya w ciele
%liczb zespolonych
%przykladowe wielomiany:
%a) W(x) = 2x^3 - 3x^2 - 5x + 6
%b) W(x) = -5x^4 + 14x^3 - x^2 + 2x - 24
%c) W(x) = x^3 + 1
%d) W(x) = x^13 + 1

%x = [2,-3,-5,6]; %wariant a)
%x = [-5,14,-1,2,-24]; %wariant b)
%x = [1,0,0,-1]; %wariant c)
x = [1,0,0,0,0,0,0,0,0,0,0,0,0,1]; %wariant d)

a = -500; %dobranie wartosci startowej dla czesci rzeczywistej
c = -450; %dobranie wartosci startowej dla czesci zespolonej
N = 1000; %gorna granica czesci rzeczywistej
M = 900; %gorna granica czesci zespolonej
n = 250; %liczba punktow czesci rzeczywistej
m = 300; %liczba punktow czesci zespolonej
h1 = N/n; %skok czesci rzeczywistej
h2 = M/m; %skok czesci zespolonej

B = zeros(M,N);
```

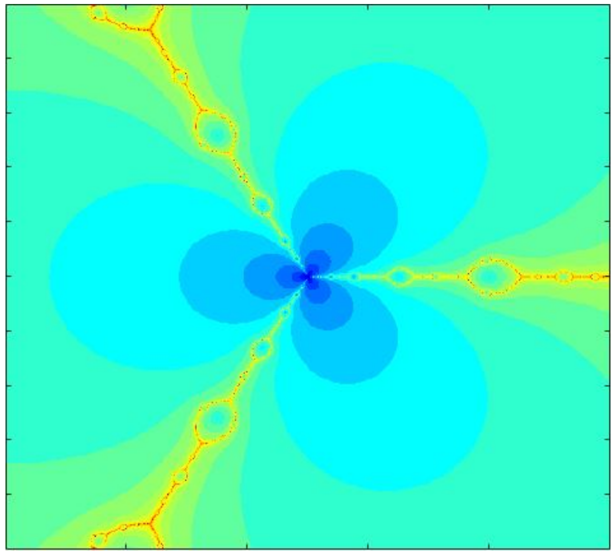
```
%utworzenie siatki punktow w prostokacie NxM

for k = 1:h2:M
    for l = 1:h1:N
        B(k,l) = a + l + 1i*(k + c);
    end
end

A = zeros(m,n);
C = nonzeros(B.');
r = 1;

%A) przyporzadkowanie liczby iteracji prowadzacej do zakonczenia procesu
%   metody Halleya dla kazdego punktu startowego
%B) przyporzadkowanie miejsca zerowego, do ktorego zbiezna jest metoda
%   Halleya przy danym punkcie startowym

for p = 1:m
    for q = 1:n
        A(p,q) = halley(x,C(r));
        r = r + 1;
    end
end

imagesc(A); %wariant A)
%imagesc(real(A)); %wariant B)
colorbar;
colormap jet;

%wykonal:
%   Maciej Salwowski
%   MiNI Politechnika Warszawska
```

W celu uzyskania dużej próby, pozwalającej na otrzymanie satysfakcjonujących rezultatów program korzysta z siatki punktów generowanej przez użytkownika poprzez wprowadzenie wartości początkowych, skoku oraz ilości punktów. Siatka zostaje utworzona na przestrzeni liczb zespolonych z osobnymi wartościami startu odpowiednio dla części rzeczywistej jak i urojonej oraz różnymi długościami przesunięcia (skoku).

Po dokonaniu obliczeń dla każdego kolejnego wygenerowanego punktu startowego, liczba iteracji potrzebna do znalezienia miejsca zerowego zapisywana jest w nowej macierzy, której liczba wierszy odpowiada liczbie różnych części rzeczywistych generowanych punktów, natomiast liczba kolumn dotyczy części urojonych. Ostatecznie macierz ta wyświetlana jest na ekranie, dzięki czemu użytkownik może zobaczyć rozkład szybkości zbieżności metody Halleya dla wcześniej wybranych punktów.

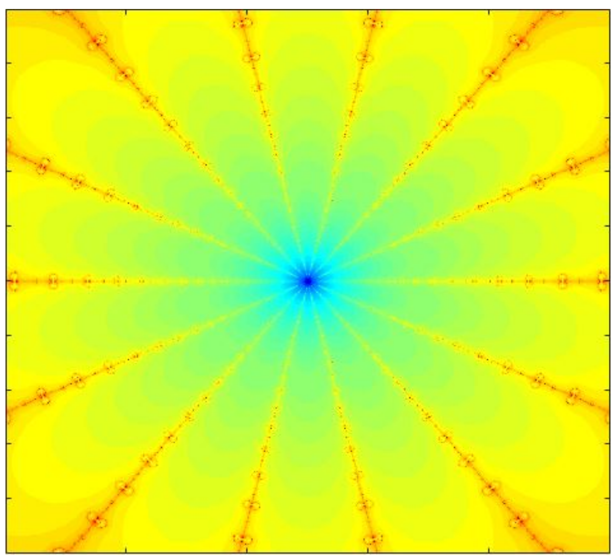
4 Przykłady obliczeniowe

Program przetestowany został na różnych wielomianach z doborem całkowicie różnych siatek punktów startowych. Rezultaty są szczególnie interesujące dla wielomianów postaci $W(x) = x^n + 1$, gdzie $n \in N$, przy których liczba charakterystycznych promieni odchodzących od środku układu jest równa n. Spójrzmy na dwa przykłady:



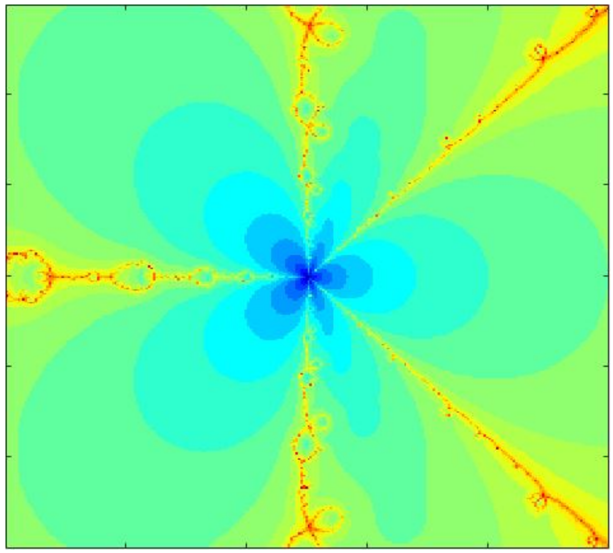
Rysunek 1: Wizualizacja dla wielomianu $W(x) = x^3 + 1$

oraz



Rysunek 2: Wizualizacja dla wielomianu $W(x) = x^{14} + 1$

Wizualizacja wygląda podobnie, z zachowaniem charakterystycznych promieni, dla dowolnego wielomianu. Różnią się one zazwyczaj natężeniem liczby iteracji oraz kształtem promieni. Dla porównania zwróćmy uwagę na wielomian $W(x) = -5x^4 + 14x^3 - x^2 + 2x - 24$, którego wizualizacja wygląda następująco:



Rysunek 3: Wizualizacja dla wielomianu $W(x) = -5x^4 + 14x^3 - x^2 + 2x - 24$

5 Podsumowanie

Zgodnie z oczekiwaniami, już na pierwszy rzut oka widać, że szybkość zbieżności dla każdego wielomianu nie jest przypadkowa. Kolejne punkty startowe tworzą figury według bliżej nieznanego klucza, zachowując swoją charakterystykę wraz z rosnącą wartością punktów zerowych.