# Metody numeryczne: projekt nr 2

20 stycznia 2020

## Spis treści



## 1 Autorzy


Jan Borowski
Piotr Fic
Grupa laboratoryjna: wtorki, godzina 10:15

# 2 Temat projektu

## 2.1 Treść zadania

**Wizualizacja szybkości zbieżności dla metody Newtona (w dziedzinie zespolonej) zastosowanej do znalezienia zera wielomianu:**

$$w_n(x) = \sum_{k=0}^{n} a_k H_k(x)$$

Nie należy sprowadzać wielomianu $w_n$ do postaci naturalnej! Do obliczania wartości wielomianu $w_n$ oraz jego pochodnej należy wykorzystać związek rekurencyjny spełniany przez wielomiany Hermite'a.

## 2.2 Związek rekurencyjny wielomianów Hermite'a

$H_0 = 1$
$H_1 = 2x$
$H_{n+1}(x) = 2xH_n(x) - 2nH_{n-1}(x)$

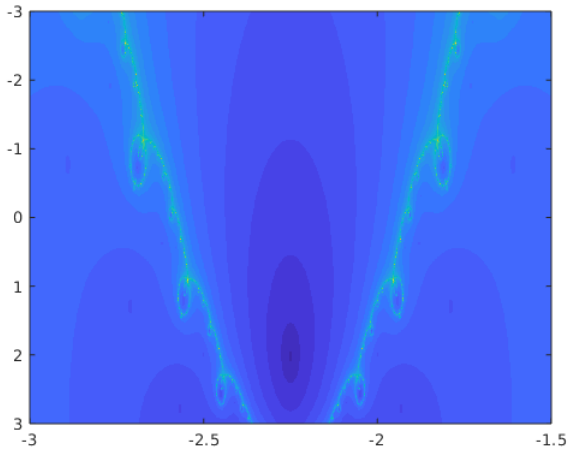
## 2.3 Metoda Newtona

Załóżmy, że $x_0 \in \mathbb{C}$ Kolejne przybliżenia są dane rekurencyjnym wzorem:

# 3 Przykłady

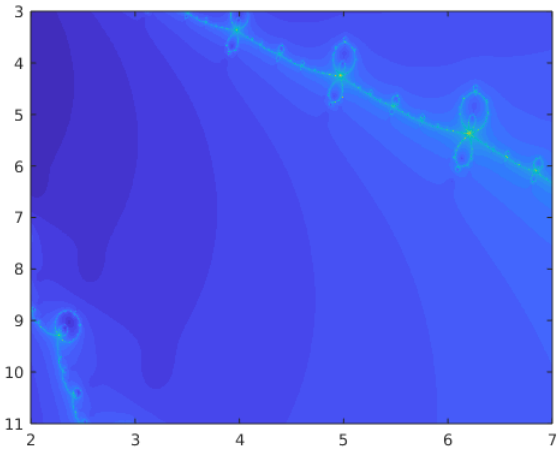
## 3.1 Przykład 1

Obszar działania metody: $[-3, -1.5] \times [-3, 3]$
Wektor $a_k = [1, 1, 1, 1, 1]$
Wizualizacja poszukiwania zer wielomianu:



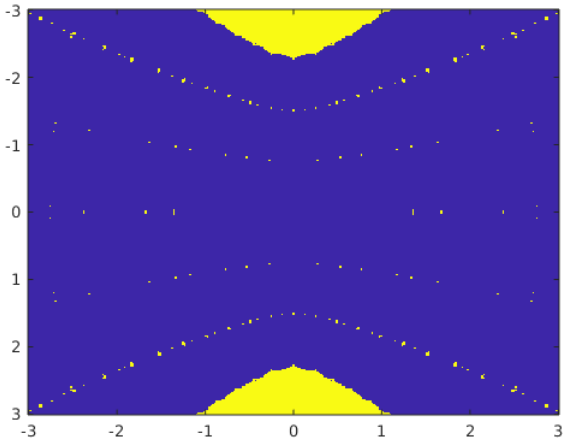
## 3.2 Przykład 2

Obszar działania metody: $[2, 7] \times [3, 11]$
Wektor $a_k = [1, 2, 3, 4, 5, 6]$
Wizualizacja poszukiwania zer wielomianu:



### 3.3 Przykład 3

Obszar działania metody: $[-3, 3] \times [3, 3]$
Wektor $a_k = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]$
Wizualizacja poszukiwania zer wielomianu:



# 4 Menu


MENU
Wizualizacji zbieżności
Podaj przedział [a,b]
Podaj przedział [c,d]
Podaj liczby punktów [n,m]
Podaj wektor współczynników ak
Podaj krotność n wielomianu Wn(x)
Wizualizuj zbieżność
Przykład Koala [-3,-1.5]x[-3,3]; ak=ones(5); n=4; nm=10x10
Liczby pierwsze
FINISH

# 5 Kod rozwiązania

Funkcja obliczająca wartość i pochodne wielomianu rekurencyjnie:

```
function [out,outd] = hermit(x,a,n)
%Function calculates Wn(x) and Wn'(x)
%Wn(x) = sum (0 to n) of { a[k] * H[k](x) }

H0 = 1;
H0d = 0;

H1 = 2*x;
H1d = 2;

out = a(2)*2*x+a(1)*1;
outd = a(2)*2;

for i=2:1:n
    disp(i)
    if rem(i,2) == 0
        H0d = (2*H1+2*x*H1d-2*(i-1)*H0d);
        H0 = (2*x*H1-2*(i-1)*H0);
        outd = outd + a(i+1)*H0d;
        out = out + a(i+1)*H0;
    end
    if rem(i,2) == 1
        H1d = (2*H0+2*x*H0d-2*(i-1)*H1d);
        H1 = (2*x*H0-2*(i-1)*H1);
        out = out + a(i+1)*H1;
        outd = outd +a(i+1)*H1d;
    end
end

end
```

Funkcja implementująca metodę Newtona:

```
function [k,x] = newton(a,b,x0,ak,n)
%NEWTON Newton method for polynomials from hermit
    function
%   Input:
%       [a, b] - interval
%       x0 - start point x0 in [a,b]
%       ak - vector of polynomial coefficients
%       n - size of Wn(x) polynomial
%   Output:
%       k - number of iterations made before stop
%       x - reached root approximation
%Test: [k,x] = newton(-5, -1.4, -5, ones(5), 4)

x = x0;
k = 1;

[w, wd] = hermit(x, ak, n);
d = 0 ;
while (abs(w) > 1e-10 && d < 10000)
    x = x - (w/wd);
    [w, wd] = hermit(x, ak, n);
    k = k+1;
    d = d+1;

end
end
```

Menu do obsługi powyższych funkcji:

```
clear
clc

finish=9;
kontrol=1;

while kontrol~=finish

    kontrol=menu('Wizualizacji zbie no ci', 'Podaj
       przedzia  [a,b]', 'Podaj przedzia  [c,d]', '
       Podaj liczby punkt w [n,m]', 'Podaj wektor
       wsp  czynnik w ak', 'Podaj krotno   n
       wielomianu Wn(x)', 'Wizualizuj zbie no  ','
       Przyk ad Koala [-3,-1.5]x[-3,3]; ak=ones(5); n
       =4; nm=10x10','Liczby pierwsze','FINISH');

    switch kontrol
        case 1
            ab = input('Podaj przedzia  jako wektor [
               a,b]');

        case 2
            cd = input('Podaj przedzia  jako wektor [
               c,d]');

        case 3
            nm = input('Podaj liczby punkt w jako
               wektor [n,m]');
        case 4
            ak = input('Podaj wektor wsp  czynnik w
                ak');

        case 5
            n = input('Podaj krotno   wielomianu n:'
               );

        case 6
            %Check if polynomial is correct
            if length(ak)-n ~= 1
                disp("B   d w d ugo ci wektora ak i
                    rozmiarze wielomianu n")
                break
            end
```

```
            a = ab(1);
            b = ab(2);
            c = cd(1);
            d = cd(2);

            [fa, fad] = hermit(a, ak, n);
            [fb, fbd] = hermit(b, ak, n);

            x = linspace(a, b, nm(1));
            y = linspace(c, d, nm(2));

            A = zeros(nm(1), nm(2));

            for i = 1:length(x)
                for j = 1:length(y)
                    x0 = x(i) + 1i*y(j);
                    [k, r] = newton(a, b, x0, ak, n);
                    A(i,j) = k;

                end
            end

            imagesc(A)
        case 7
            ab = [ -3,-1.5];
            cd = [-3,3];
            nm = [10,10];
            ak = ones(5);
            n = 4;
        case 8

            ab = [2;7];
            cd = [3;11];
            nm = [1000;1000];
            ak = 1:6;
            n = 5;
        case 9
            break
    end
end
```