Obliczenia naukowe
Sprawozdanie – Lista 4

Bartosz Rzepkowski

1. Zadanie 1 – Ilorazy różnicowe

1.1 Opis algorytmu

Funkcja ilorazyRoznicowe przyjmuje jako argumenty wektory x oraz f. Wektor x zawiera zawiera węzły x_0, …, x_n, natomiast wektor f zawiera wartości funkcji interpolowanej w węzłach $f(x_0)$, …, $f(x_n)$.

Ilorazy różnicowe wyliczane są z następującego wzoru:

$$f[x_0, x_1, \ldots, x_n] = \frac{f[x_1, x_2, \ldots, x_n] - f[x_0, f_1, \ldots, f_{n\text{-}1}]}{x_n - x_0}$$

Funkcja początkowo kopiuje wszystkie wartości f to tablicy fx. Kolejne ilorazy różnicowe wyliczane są za pomocą dwóch pętli, w następujący sposób:

1. I = 2 – jest to licznik pierwszej pętli,
2. J = length(x) – jest to indeks w drugiej pętli równy długości wektora x,
3. Dopóki I <= length(x):

3.1. Dopóki J >= I :

  3.1.1. Wartość ilorazu różnicowego w komórce fx[J] = fx[J] – fx[J-1] / x[J] – x[J - I + 1]

  3.1.2. Zmniejsz wartość J o 1,

  3.1.3. Przejdź do kroku 3.1,

3.2. Zwiększ I o 1.

3.3. Przejdź do kroku 3.

Powyższy algorytm ilustruje poniższy pseudokod:

```
for (I = 2, I <= length(x), I++)
   for (J = length(x), J >= I, J--)
     fx[J] = (fx[J] - fx[J-1])/(x[J] - x[J-I+1])
   end
 end
```

Pseudokod algorytmu obliczającego ilorazy różnicowe

Możemy zauważyć, że dzięki zwiększaniu I kolejne wartości $f[x_0]$, $f[x_0, x_1]$, …, $f[x_0, x_1, \ldots, x_n]$ zapamiętywane są w tablicy fx w następujący sposób:

$fx[1] = f[x_0]$,

$fx[2] = f[x_0, x_1]$,

…

$fx[n+1] = f[x_0, x_1, \ldots, x_n]$.

2. Zadanie 2 – Wartość wielomianu interpolacyjnego

2.1 Opis algorytmu

Funkcja warNewton przyjmuje na wejściu tablice x oraz fx (zawierające odpowiednio węzły $x_0, x_1, \ldots, x_n$ oraz wartości funkcji w tych węzłach) oraz stałą t, w której ma zostać obliczona wartość wielomianu. Algorytm oblicza wartość wielomianu interpolacyjnego stopnia n w postaci Newtona $N_n(x)$ w punkcie x = t. Korzysta on z uogólnionego algorytmu Hornera:

$w_n(x) := f[x_0, x_1, \ldots, x_n]$,

$w_k(x) := f[x_0, x_1, \ldots, x_k] + (x - x_k)w_{k+1}$ (k = n-1, …, 0)

$N_n(x) := w_0(x)$.

Algorytm działa w następujący sposób:
1. Pod zmienną nt podstaw wartość fx[length(fx)] (jest ona równa f[x_0, x_1, …, x_n]),
2. Podstaw pod zmienną K wartość length(fx)-1 (= n - 1),
3. Dopóki K >= 1:
    3.1 nt = fx[K] + (t – x[K])*nt
    3.2 Zmniejsz K o 1,
    3.3 Przejdź do kroku 3,
4. Zwróć wartość nt.

Po wykonaniu pętli n-1 razy funkcja zwraca wartość nt = N_n(x).

3. Zadanie 3 – Interpolacja funkcji f(x) w przedziale [a,b]

3.1. Opis algorytmu

Funkcja rysujNnfx interpoluje funkcję f(x) w przedziale [a,b], po czym rysuje wykres wielomianu interpolacyjnego i interpolowanej funkcji. Funkcja wykorzystuje wcześniej opisane funkcje ilorazyRoznicowe oraz warNewton.

Początkowo funkcja oblicza odstęp h między kolejnymi węzłami ze wzoru:

h = (b – a) / n.

Następnie tworzy tablicę x, która przechowuje węzły i kolejne komórki wypełnia wartościami x[k+1] = a + (k * h), gdzie k początkowo jest równe 0 i po każdym przypisaniu jest zwiększane o 1. Następnie algorytm tworzy tablicę y zawierającą wartość funkcji w danych węzłach (y[k] = f(x[k])).

Algorytm za pomocą funkcji ilorazyRoznicowe dla wektorów x oraz y oblicza ilorazy różnicowe i zapisuje je w tablicy fx. Następnie tworzy tablicę interpolation, której komórki wypełnia w następujacy sposób:

interpolation[i] = warNewton(x, fx, x[i]).

Na końcu funkcja, wykorzystując pakiet Winston, rysuje wykres funkcji interpolowanej oraz wielomian interpolacyjny w następujący sposób:

```
plot(x, interpolation, "r")
hold(true)
fplot(x → z(x), [ a ,b ], "g").
```

Wykres funkcji interpolowanej oznaczony jest na zielono, natomiast wielomianu interpolacyjnego na czerwono.

4. Zadanie 4

4.1 Opis problemu

Przetestować funkcję rysujNnfx na następujących przykładach:

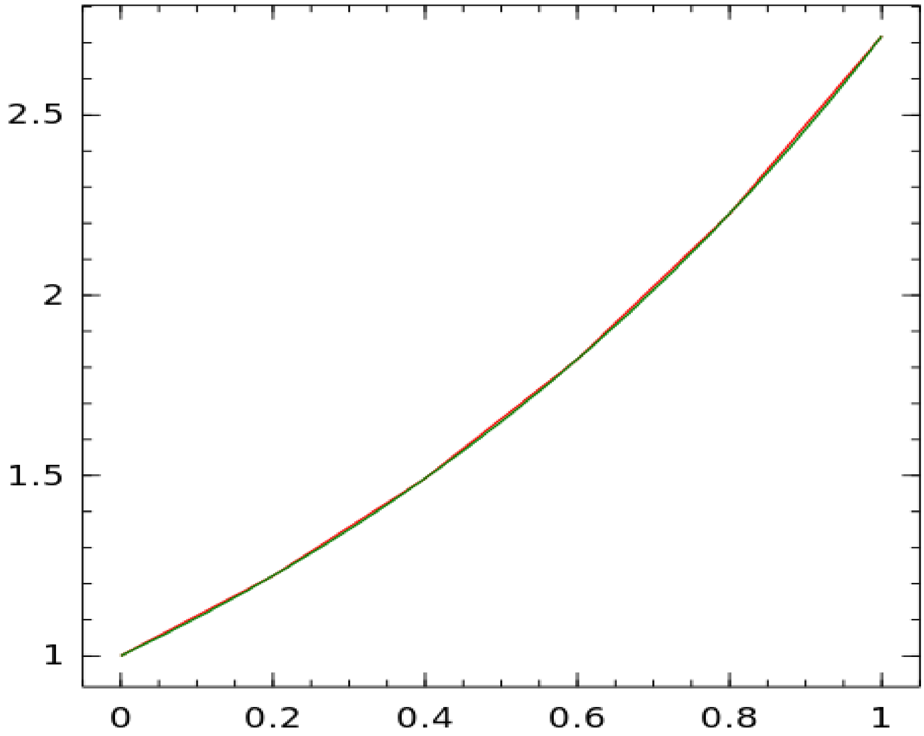
a) $e^x, [0,1], n=5,10,15,$

b) $x^2 \sin x, [-1,1], n=5,10,15.$

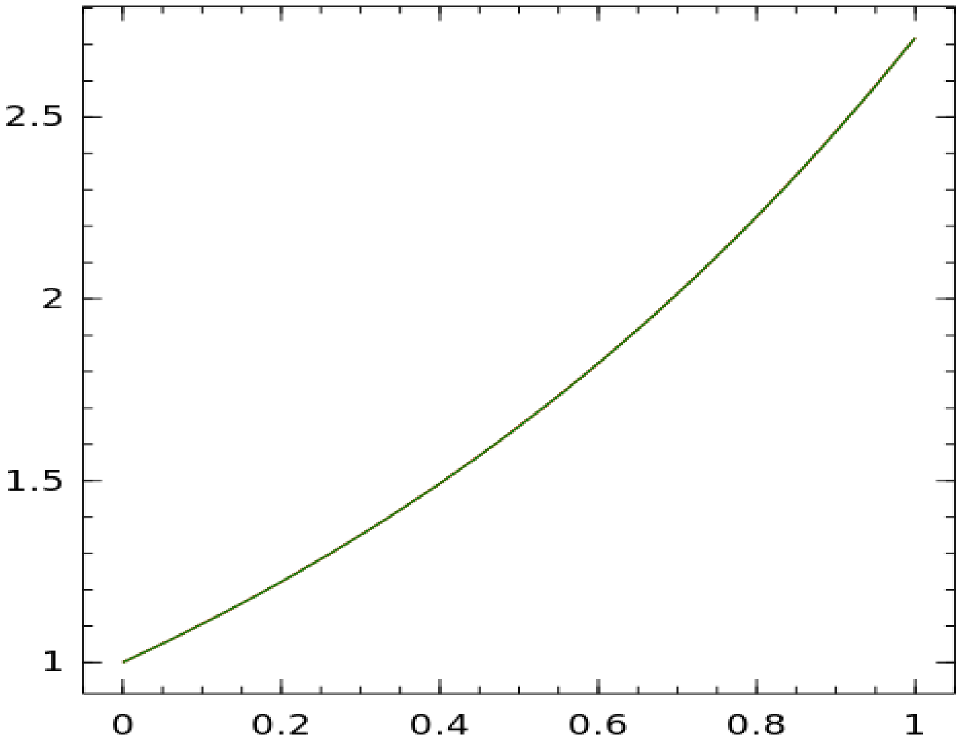
4.2 Wyniki

a)

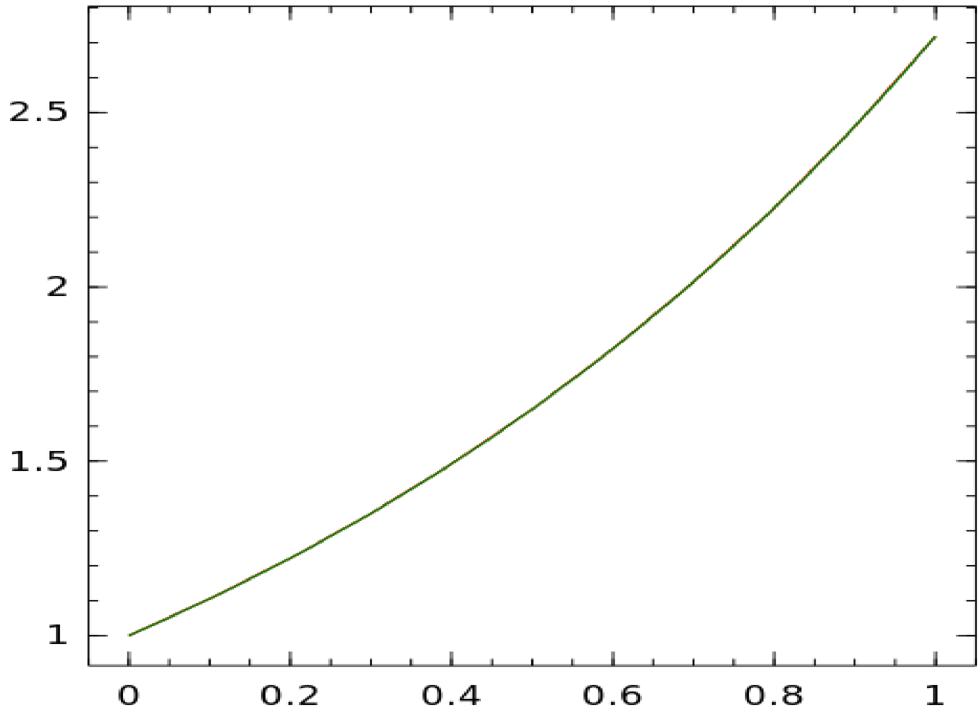
n = 5



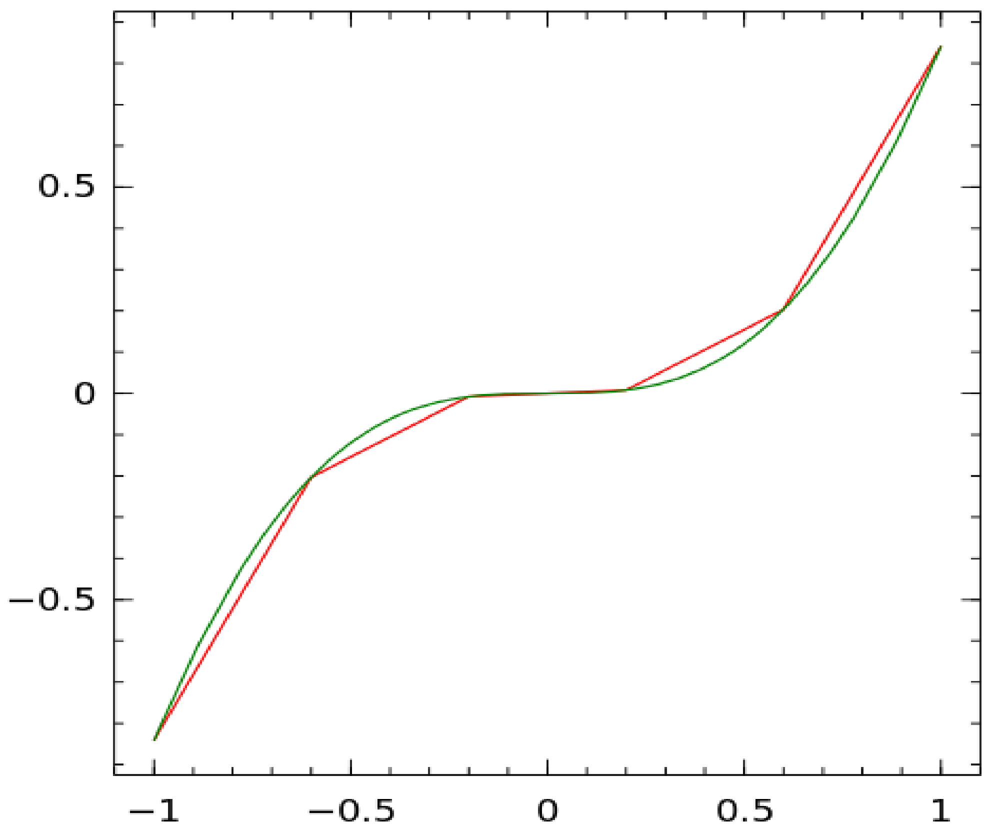
n = 10



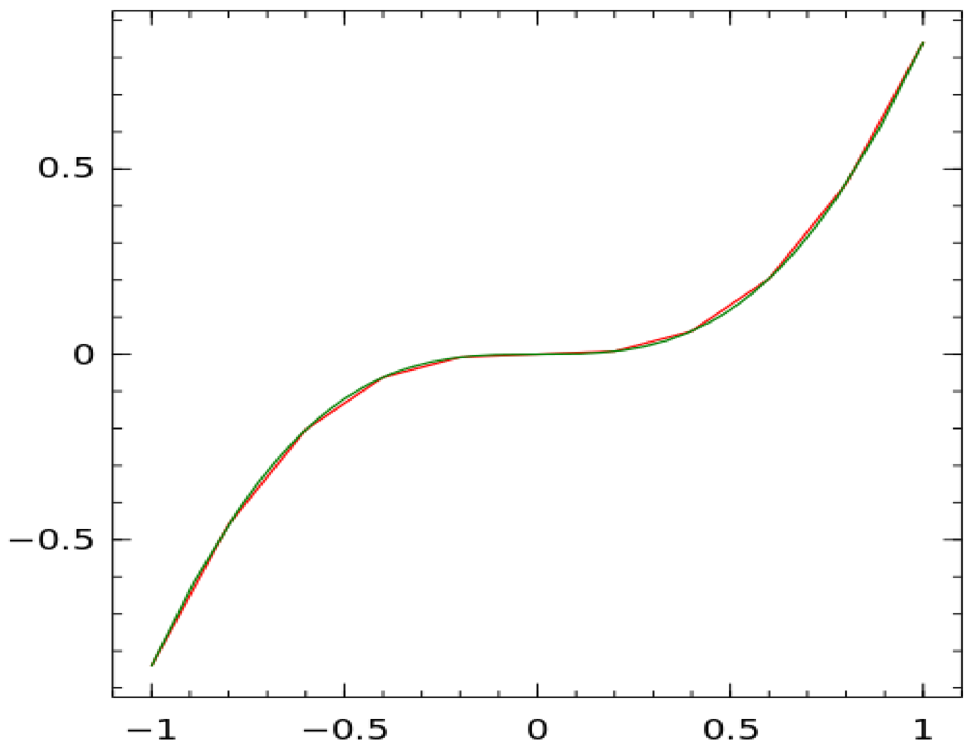
n = 15



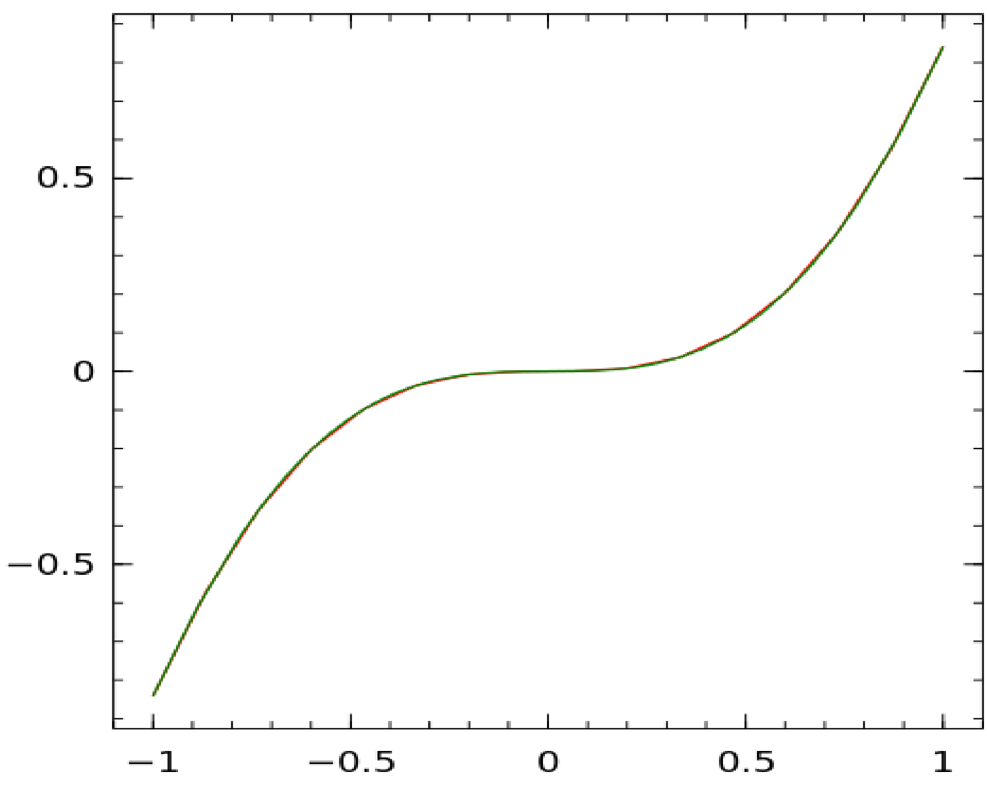
b)
n = 5



n = 10



n = 15



4.3 Wnioski

Możemy zauważyć, że dla funkcji danych w zadaniu zwiększenie stopnia wielomianu wpływa na poprawę interpolacji (linie czerwone odpowiadające wielomianowi interpolacyjnemu coraz bardziej pokrywają się z zielonymi, odpowiadającymi funkcjom interpolowanym).

5. Zadanie 5

5.1 Opis problemu

Przetestować funkcję rysujNnfx na następujących przykładach:

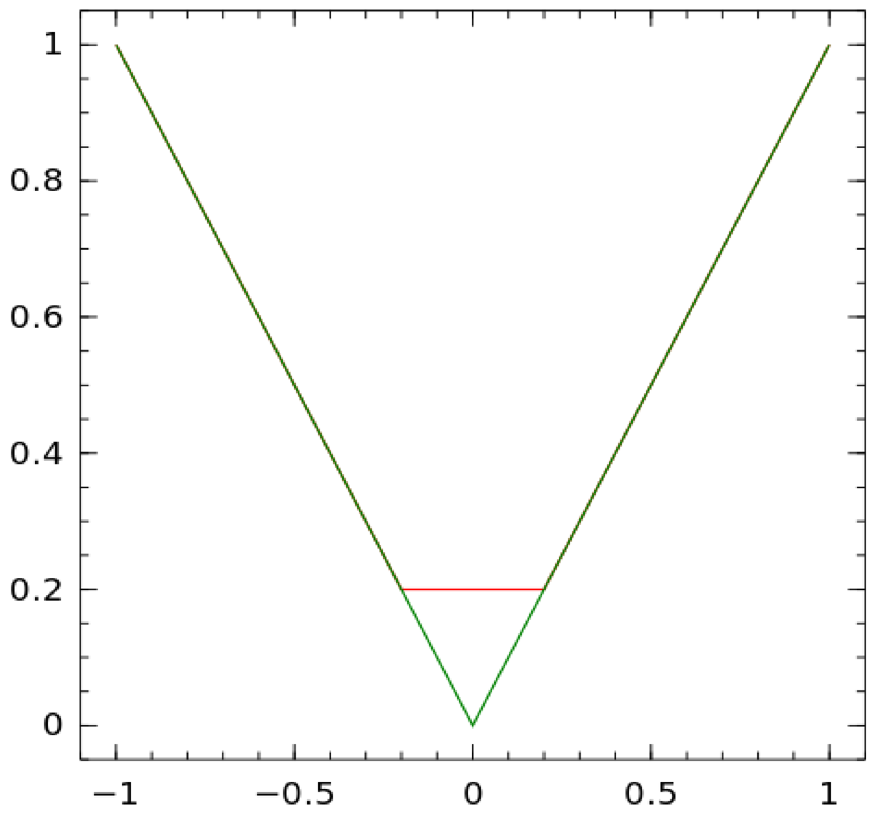
a) $|x|, [-1, 1], n=5, 10, 15,$

b) $\frac{1}{1+x^2}, [-5,5], n=5, 10, 15.$

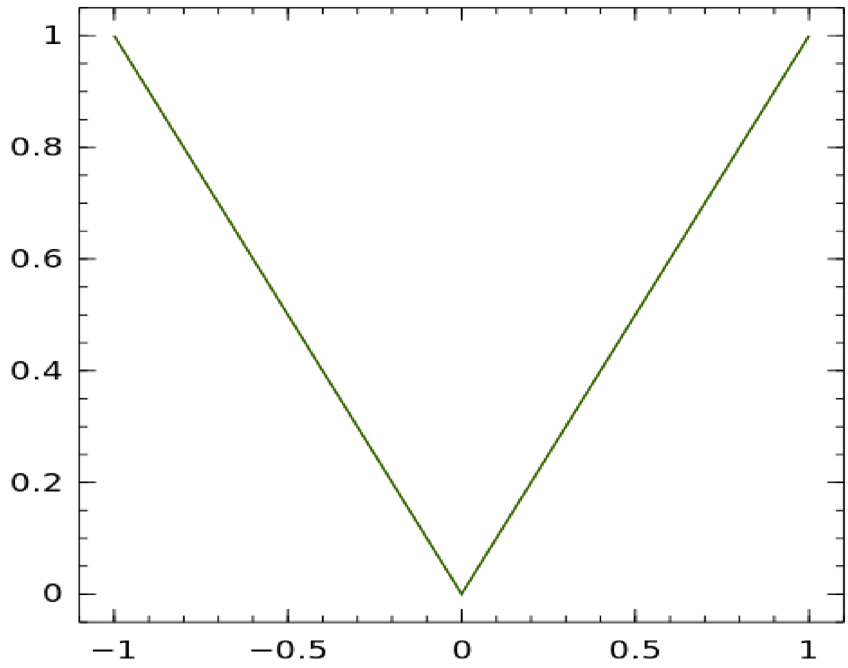
5.2 Wyniki

a)

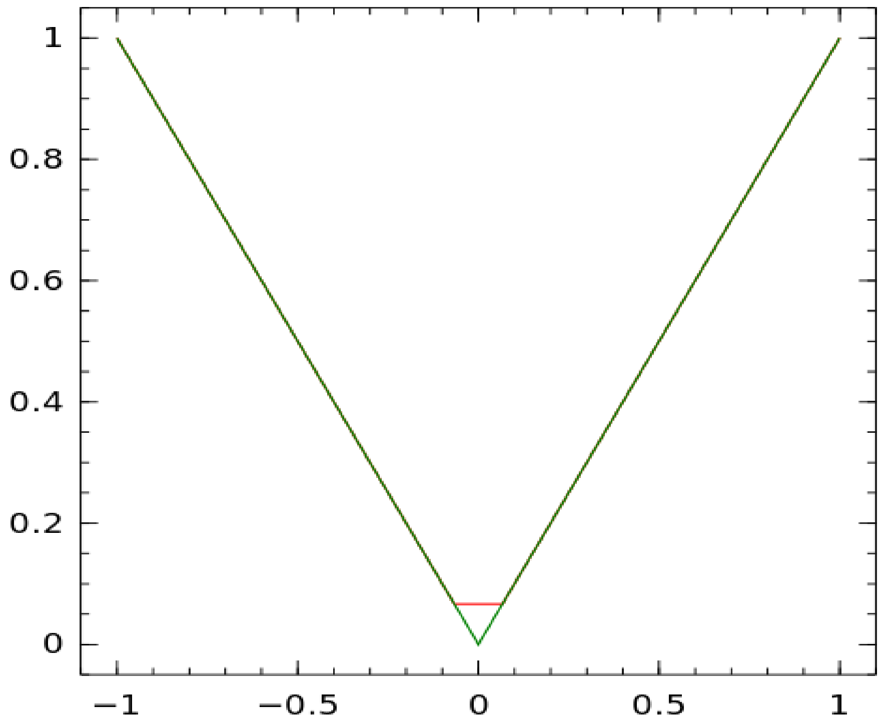
n = 5



n = 10

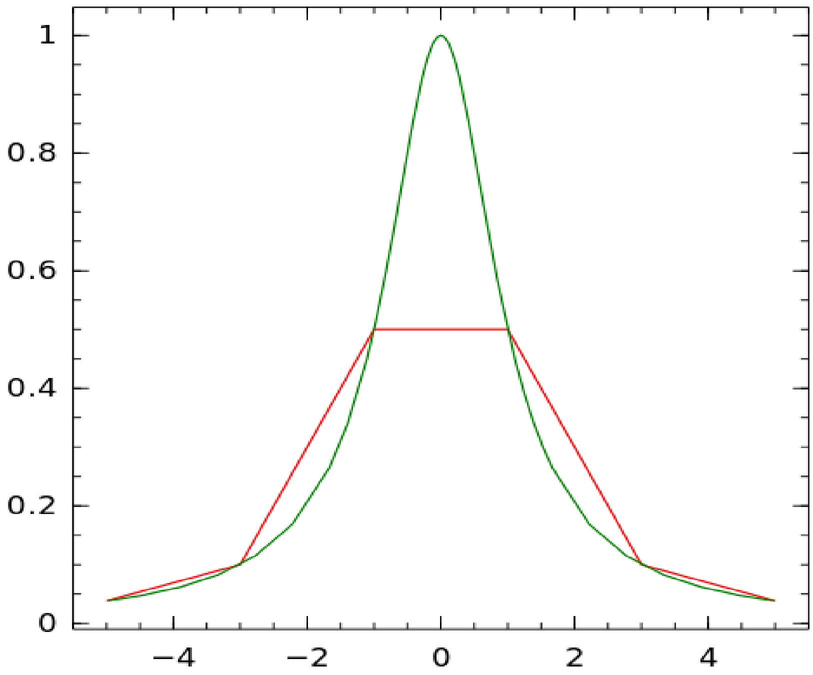


n = 15

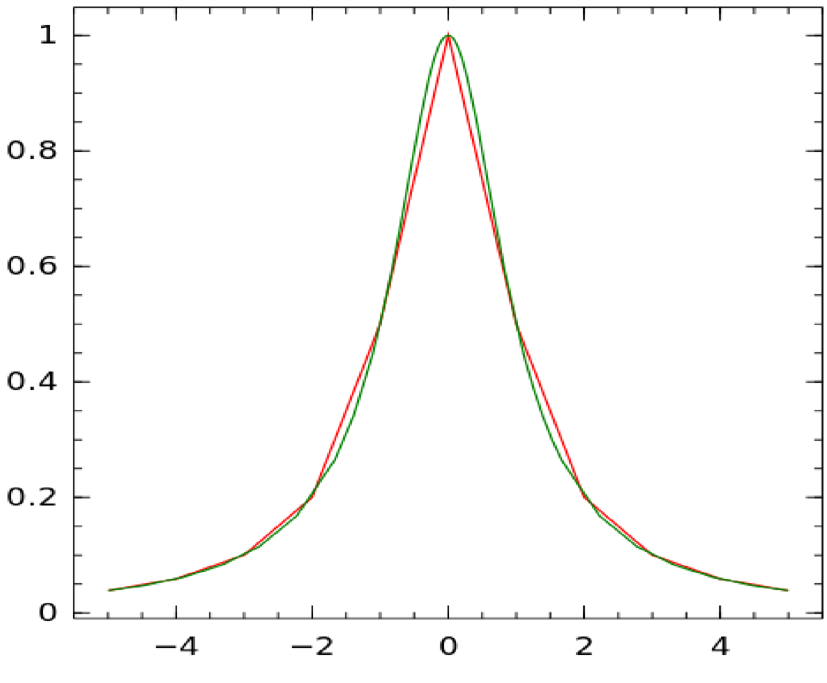


b)

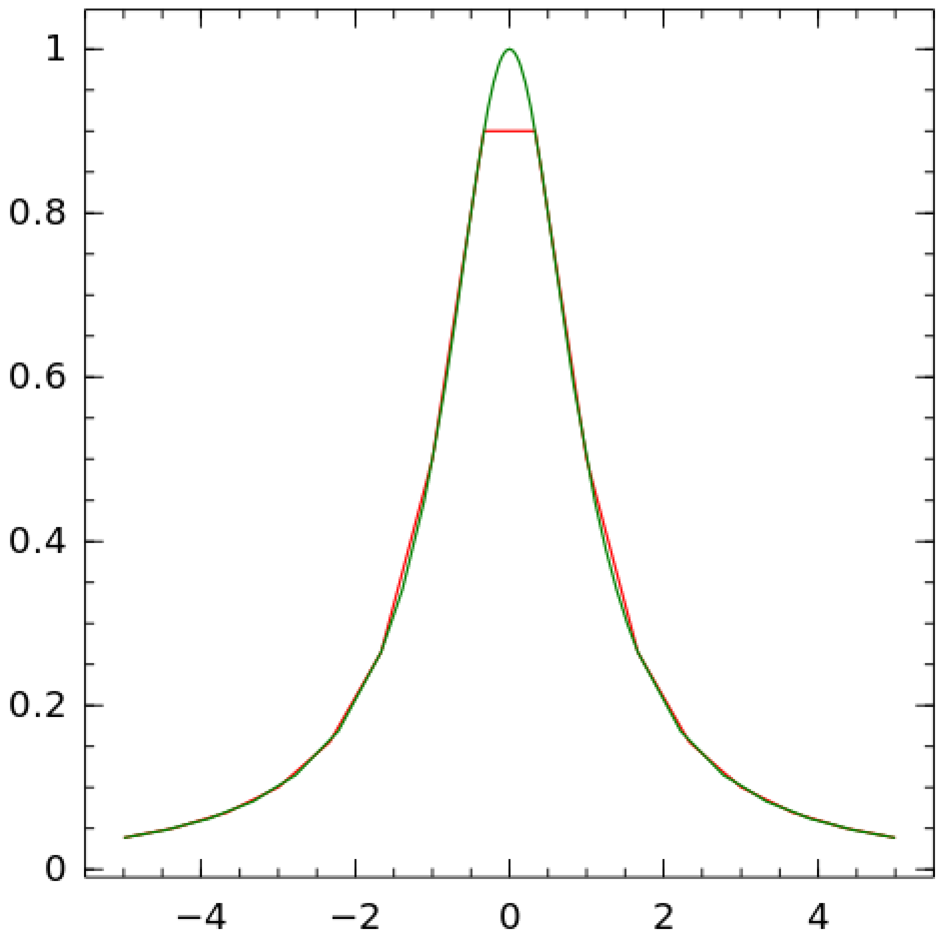
n = 5



n = 10



n = 15



5.3 Wyniki

Możemy zauważyć, że dla danych funkcji zwiększenie stopnia wielomianu poprawia interpolację funkcji tylko do pewnego momentu (w naszym przypadku do n = 10). Dalsze zwiększanie stopnia wielomianu wpływa na pogorszenie interpolacji funkcji. Jest to przykład zjawiska Runge'ego. Jest ono typowe dla wielomianów interpolacyjnych, w których odległość między kolejnymi węzłami jest stała. Aby uniknąć tego zjawiska stosuje się zmienną odległość między węzłami (odległość między kolejnymi węzłami na krańcach przedziału jest mniejsza, niż w środku przedziału).