Sprawozdanie

OBLICZENIA NAUKOWE, LISTA NR 4

ŁUKASZ KLEKOWSKI 229738

Wrocław 10.12.2017

1. Zadania 1-4

1.1.Opis problemu

Polecenia polegały na napisaniu funkcji obliczających ilorazy różnicowe, wartość wielomianu interpolującego w danym punkcie, wyznaczających współczynniki naturalne takiego wielomianu oraz funkcji która narysuje wykres funkcji oraz jej interpolacji. Należało także napisać programy testujące.

1.2.Rozwiązanie zadań

Napisałem funkcje opisane powyżej za pomocą języka Julia. Zapakowałem wszystko w moduł ON_L4_Z1-4.jl. Następnie napisałem testy sprawdzające, które testują napisane metody przez podanie im przykładowego zestawu danych, a następnie porównanie ich z rzeczywistymi wynikami.

1.2.1. Funkcja $ilorazyRoznicowe()$:

I. Opis funkcji:

Funkcja służy do obliczania ilorazów różnicowych potrzebnych do wyznaczenia wielomianu interpolacyjnego Newtona. Iloraz różnicowy jest to iloraz przyrostu wartości funkcji przez przyrost argumentu funkcji (inaczej – wielkość opisująca przyrost funkcji w danym przedziale). Wielkość tą opisuje się wzorem: $\frac{f(x_1)-f(x_0)}{x_1-x_0}$. Interpolacja natomiast jest to metoda numeryczna, która polega na wyznaczaniu tzw. Funkcji interpolacyjnej, która często jest wykorzystywana do upraszczania skomplikowanych funkcji.

Postaci ilorazów rzędu zerowego i pierwszego są oczywiste:

$$f[x_0] = f(x_0),\ \ f[x_0, x_1] = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

Ilorazy wyższych rzędów są obliczane za pomocą tego rekurencyjnego wzoru:

$$f[x_0, x_1, \dots, x_k] = \frac{f[x_1, x_2, \dots, x_k] - f[x_0, x_1, \dots, x_{k-1}]}{x_k - x_0}$$

Pseudokod środka funkcji:

```
for i=0 to n do
      fx[i]=f[i]
end
for j=1 to n do
      for i=n to j step -1 do
            fx[i]=(fx[i]-fx[i-1])/(x[i]-x[i-j])
      end
end
return fx
```

II. Przyjmowane argumenty:

Nasza funkcja przyjmuje 2 parametry:

x – wektor długości $n + 1$ zawierający węzły $x_0, \ldots, x_n \; gdzie \; x[1] = x_0, \ldots, x[n+1] = x_n$

f – wektor długości $n + 1$ zawierający wartości interpolowanej funkcji w węzłach $f(x_0), \ldots, f(x_n)$

III. Zwracane wartości:

fx - wektor długości $n + 1$ zawierający obliczone ilorazy różnicowe $fx[1] = f[x_0]$, $fx[2] = f[x_0, x_1], \ldots,$ $fx[n] = f[x_0, \ldots, x_{n-1}]$, $fx[n+1] = f[x_0, \ldots, x_n]$

1.2.2. Funkcja $warNewton()$:

I. Opis metody:

Funkcja służy do obliczania wartości wielomianu interpolacyjnego Newtona w punkcie $x = t$. Mieliśmy w zadaniu wykorzystać uogólniony algorytm Hornera oraz złożoność obliczeniowa powinna wynosić $O(n)$.

Do napisania funkcji wykorzystałem algorytm który jest na 4 liście ćwiczeniowej w zadaniu 8:

$$w_n(x) := f[x_0, x_1, \ldots, x_n]$$
$$w_k(x) := f[x_0, x_1, \ldots, x_k] + (x - x_k)\, w_{k+1}(x) \quad (k = n-1, \ldots, 0)$$
$$N_n(x) := w_0(x)$$

Algorytm na samym początku przypisuje do w[n] wartość ilorazu różnicowego $f[x_0, x_1, \ldots, x_n]$. Następnie wchodzimy do pętli i korzystamy z drugiego wzoru, czyli do w[n-1] przypisujemy sumę danego ilorazu różnicowego i iloczynu różnicy zadanego punktu z danym węzłem z obliczoną poprzednią wartością czyli w[n].

Po wyjściu z pętli zwracana jest wartość wielomianu która znajduje się w w[1].

Pseudokod środka funkcji:

```
w[n] = fx[n]
for i=n-1 to 1 step -1 do
      w[i] = fx[i] + (t-x[i])*w[i+1]
end
return w[1]
```

II. Przyjmowane argumenty:

Nasz algorytm przyjmuje 3 parametry:

x – wektor długości $n + 1$ zawierający węzły $x_0, \ldots, x_n \; gdzie \; x[1] = x_0, \ldots, x[n+1] = x_n$

fx – wektor długości $n + 1$ zawierający ilorazy różnicowe $fx[1] = f[x_0],\ fx[2] = f[x_0, x_1]\ , \ldots,\ fx[n] = f[x_0, \ldots, x_{n-1}],\ fx[n+1] = f[x_0, \ldots, x_n]$

nt – punkt, w którym należy obliczyć wartość wielomianu

III. Zwracane wartości:

nt – wartość wielomianu w punkcie t.

1.2.3. Funkcja $naturalna()$:

I. Opis metody:

Funkcja służy do obliczania naturalnych współczynników wielomianu interpolacyjnego Newtona znając węzły oraz ilorazy różnicowe. Złożoność obliczeniowa powinna wynosić $O(n^2)$ Nasz wielomian interpolacyjny możemy napisać tak:

$$p(x) = fx_0 + (x - x_0)(fx_1 + (x - x_1)(fx_2 + (x - x_2)(fx_3 + (x - x_3)(\ldots (fx_{n-1} + (x - x_{n-1})fx_n)\ \ldots))))$$

Gdzie $fx_0, \ldots, fx_n$ to nasze ilorazy różnicowe ,a $x_0, \ldots, x_{n-1}$ to nasze węzły.

Teraz możemy obliczać rekursywnie nasze współczynniki zaczynamy od obliczenia współczynnika przy najwyższej potędze. Jest to oczywiście fx_n.

W każdym kolejnym kroku nasz wielomian może być zapisany w ten sposób (podobnie jak w funkcji wyżej):

$$p_k = fx_k + (x - x_k)p_{k+1} = fx_k + (x - x_k)(b_0 + b_1 x + b_2 x^2 + \ldots)$$

Teraz jesteśmy w stanie obliczać współczynniki dla wielomianu pomocniczego p_k znając współczynniki wielomianu p_{k+1}

Dla przykładu:

a to współczynniki wielomianu p_k a b to współczynniki wielomianu p_{k+1}

$a_0 = fx_k - x_k b_0$

$a_1 = b_0 - x_k b_1$

…

W ten sposób obliczane są współczynniki przy każdej potędze.

Pseudokod programu:

```
a[n] = fx[n]
for i=n-1 to 1 step -1 do
      a[i] = fx[i]
      for j=I to n-1 do
            a[j]=a[j] - (a[j+1]*x[i])
      end
end
return a
```

II. Przyjmowane argumenty:

Nasz algorytm przyjmuje 2 parametry:

x – wektor długości $n + 1$ zawierający węzły $x_0, \ldots, x_n \; gdzie \; x[1] = x_0, \ldots, x[n+1] = x_n$

fx – wektor długości $n + 1$ zawierający ilorazy różnicowe $fx[1] = f[x_0],\ fx[2] = f[x_0, x_1]\ , \ldots,\ fx[n] = f[x_0, \ldots, x_{n-1}],\ fx[n+1] = f[x_0, \ldots, x_n]$

III. Zwracane wartości:

a – wektor długości $n + 1$ zawierający obliczone współczynniki postaci naturalnej $a[1] = a_0,\ a[2] = a_1\ , \ldots,\ a[n] = a_{n-1}, a[n+1] = a_n$.

1.2.4. Funkcja $rysNnfx()$:

I. Opis metody:

Funkcja rysuje wykres funkcji oraz wykres wielomianu interpolującego tej funkcji. Do narysowania wykresu użyłem dwóch pierwszych funkcji dzięki którym poznałem ilorazy różnicowe, oraz wartości wielomianów w danych punktach dzięki którym mogłem narysować wykres.

Węzły równoodległe były obliczane ze wzoru:

$$x_k = a + kh, h = \frac{b-a}{n}, k = 0, 1, \ldots, n$$

gdzie $a \; i \; b$ to koniec i początek przedziału.

Obliczam wartości funkcji oraz wielomianu dla 100000 argumentów równo oddalonych od siebie w zadanym przedziale i z tak uzyskanymi danymi jestem w stanie wygenerować wykres. Robię to za pomocą pakietu PlotlyJS.

II. Przyjmowane argumenty:

Nasza funkcja przyjmuje 4 parametry:

f – anonimowa funkcja

a, b – początek i koniec przedziału

n – stopień wielomianu

III. Zwracane wartości:

Funkcja rysuje wykresy funkcji oraz wielomianu w zadanym przedziale.

2. Zadanie piąte

2.1.Opis zadania

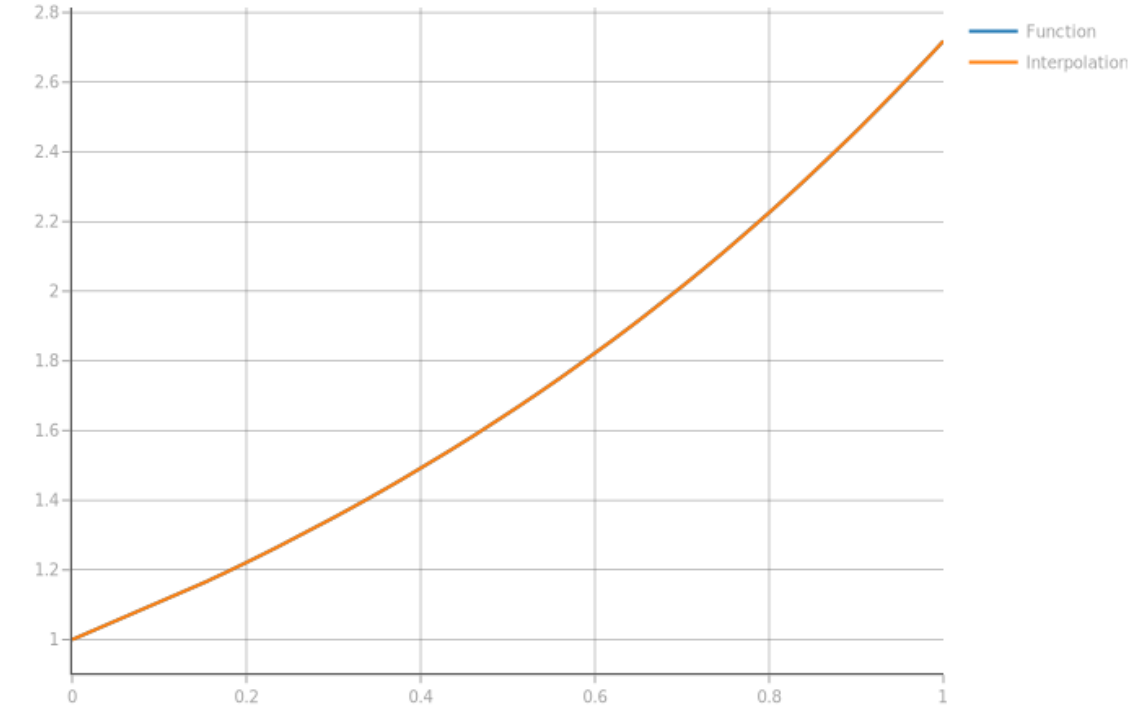
Polecenie polegało na wykorzystaniu funkcji $rysujNnfx()$ na dwóch zestawach danych:

a) $e^x, [0,1],\ n = 5, 10, 15$
b) $x^2 \sin(x), [-1, 1],\ n = 5, 10, 15$

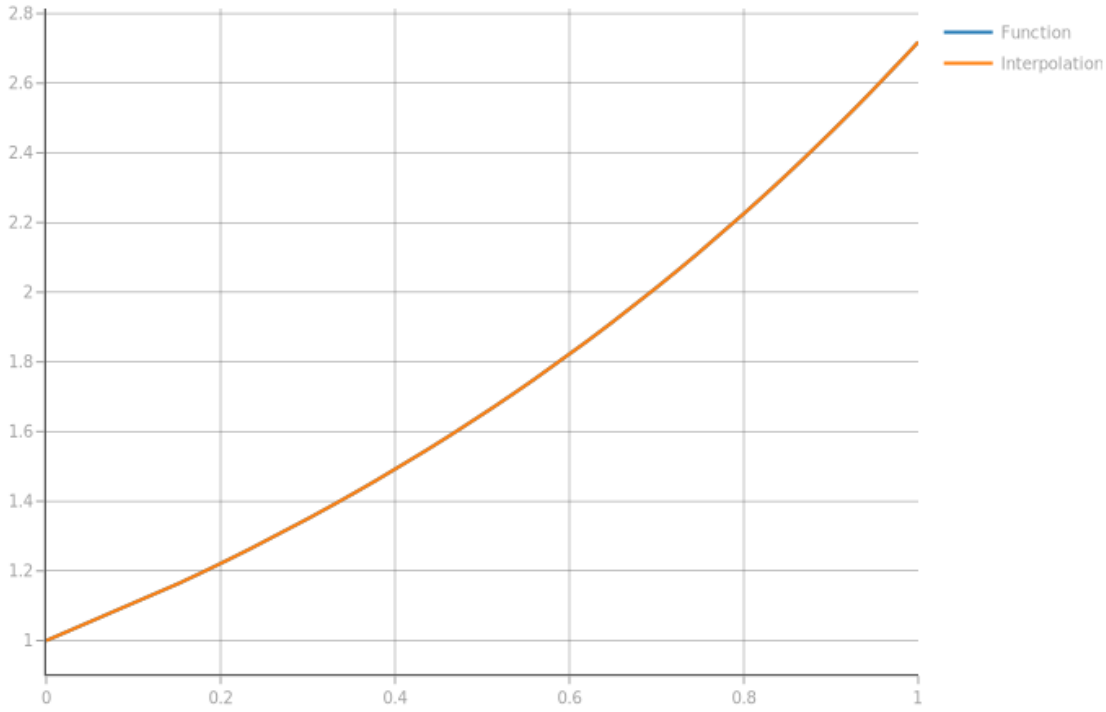
2.2.Wyniki

2.2.1. Funkcja e^x

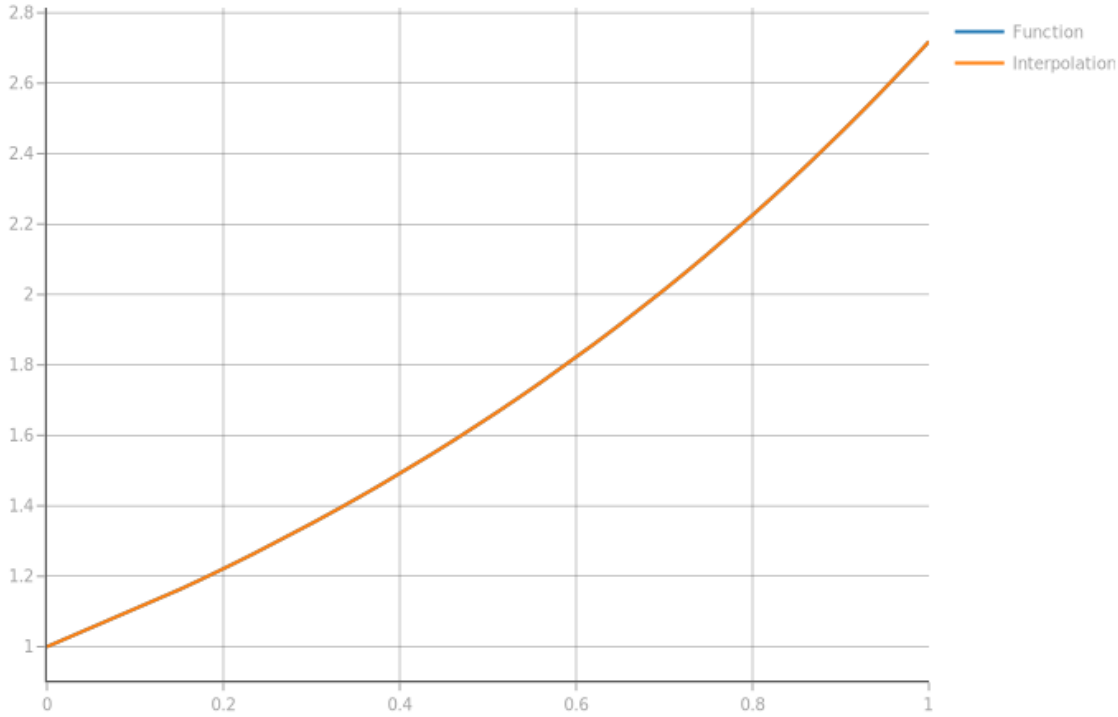
$n = 5$



$n = 10$

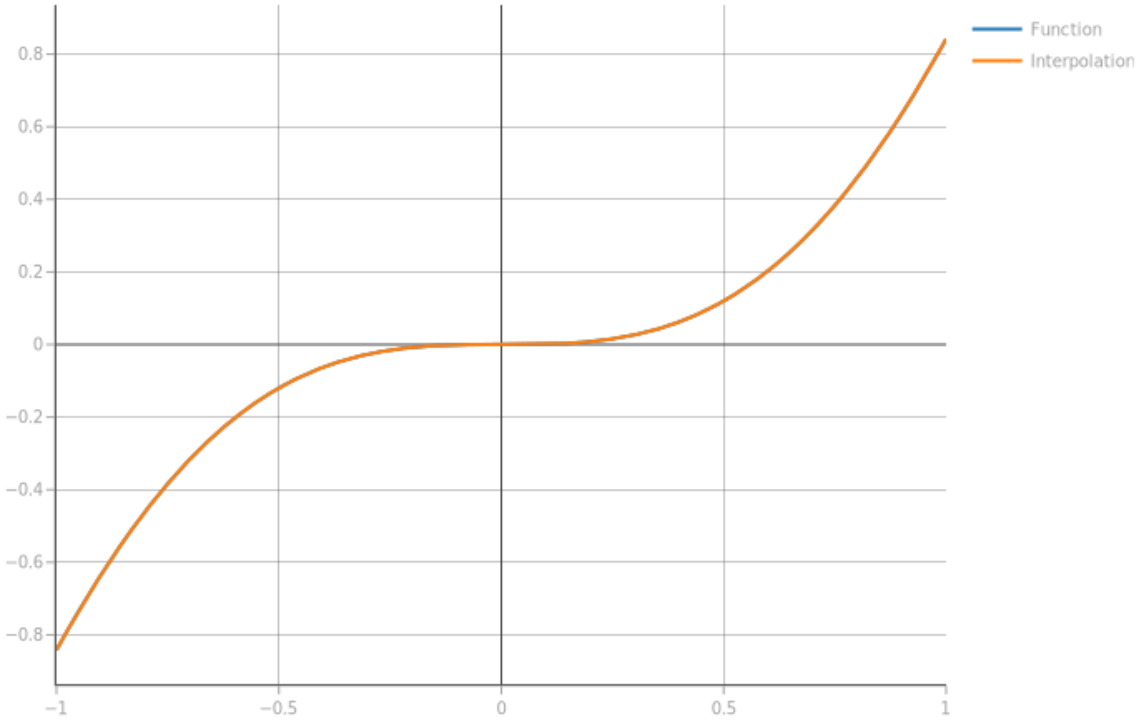

Function
Interpolation
2.8
2.6
2.4
2.2
2
1.8
1.6
1.4
1.2
1
0
0.2
0.4
0.6
0.8
1

$n = 15$

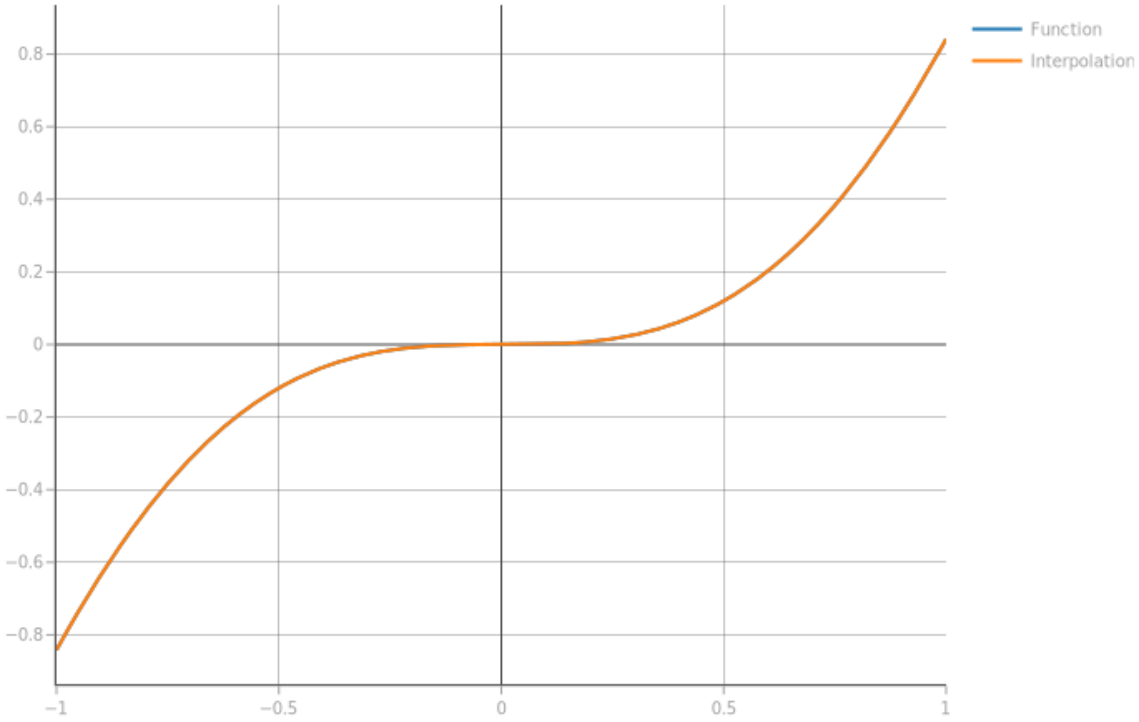

Function
Interpolation
2.8
2.6
2.4
2.2
2
1.8
1.6
1.4
1.2
1
0
0.2
0.4
0.6
0.8
1

2.2.2. Funkcja $x^2 \sin(x)$

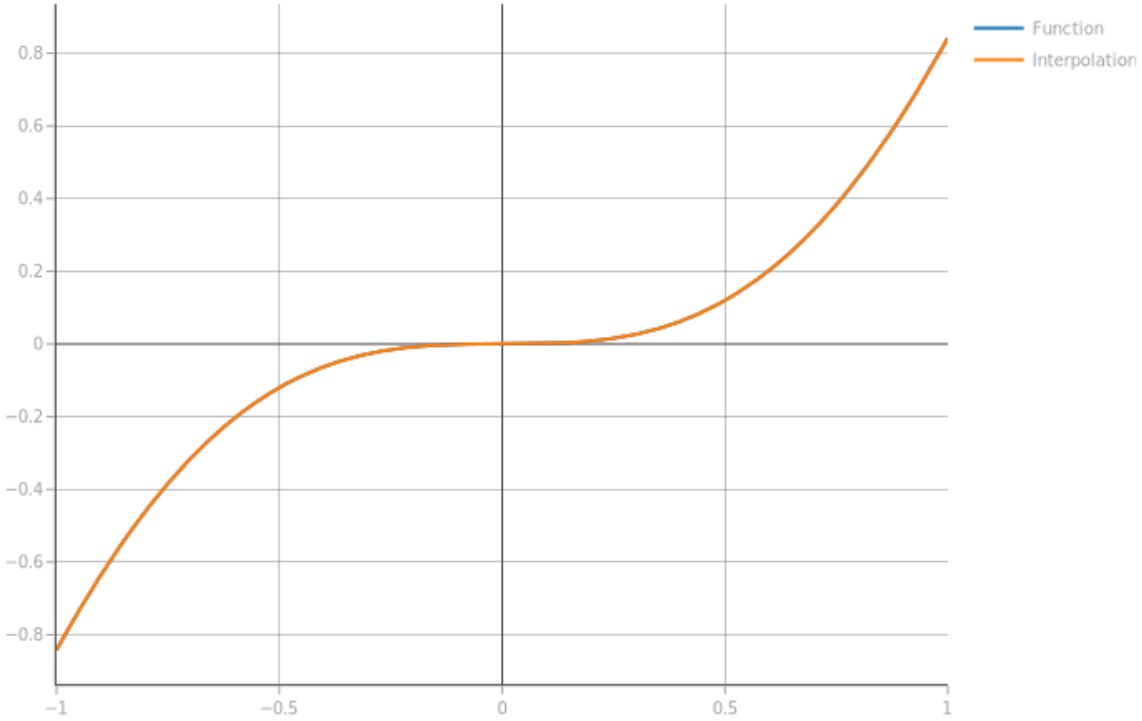
$n = 5$



$n = 10$



$n = 15$



2.3. Wnioski

Powyższe przykłady pokazują na czym polega zjawisko interpolacji. Jak widać narysowane wykresy pokrywają się. Znając jedynie niektóre punkty jesteśmy w stanie odtworzyć przebieg funkcji. Oczywiście im więcej węzłów tym dokładniej możemy zinterpolować daną funkcję.

3. Zadanie szóste

3.1.Opis zadania

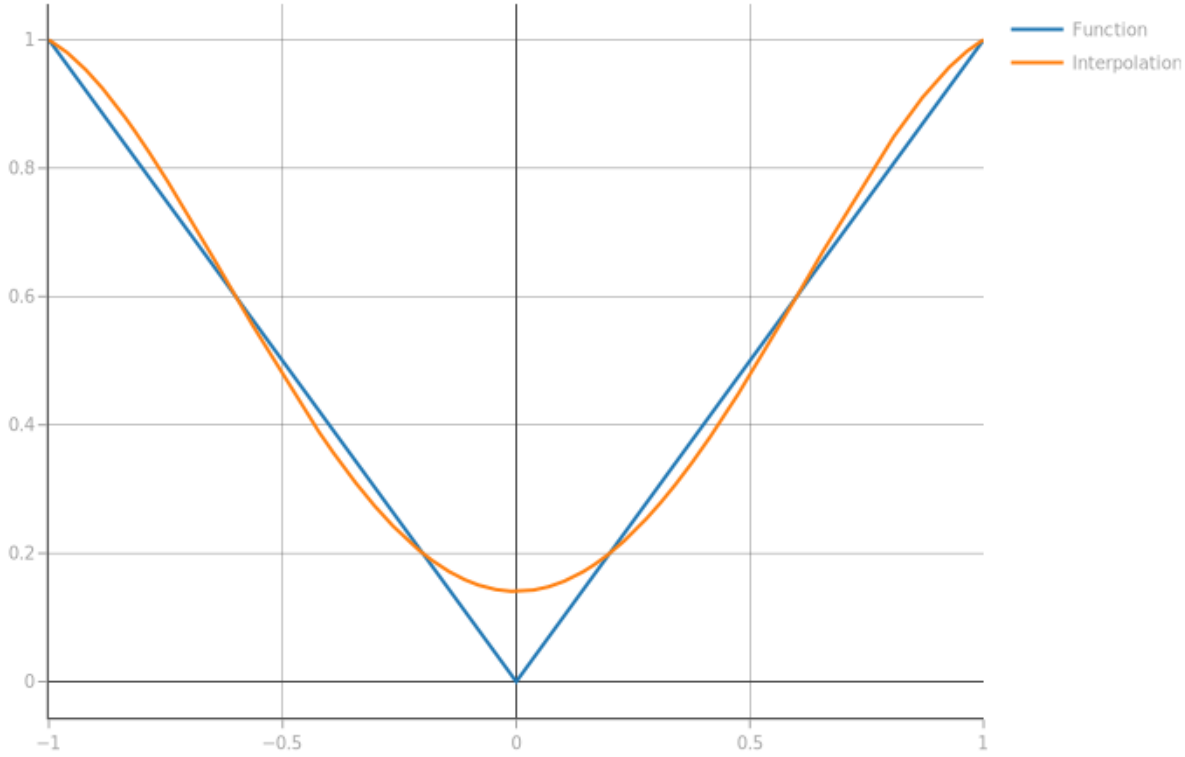
Polecenie polegało na wykorzystaniu funkcji $rysujNnfx()$ na dwóch zestawach danych:

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

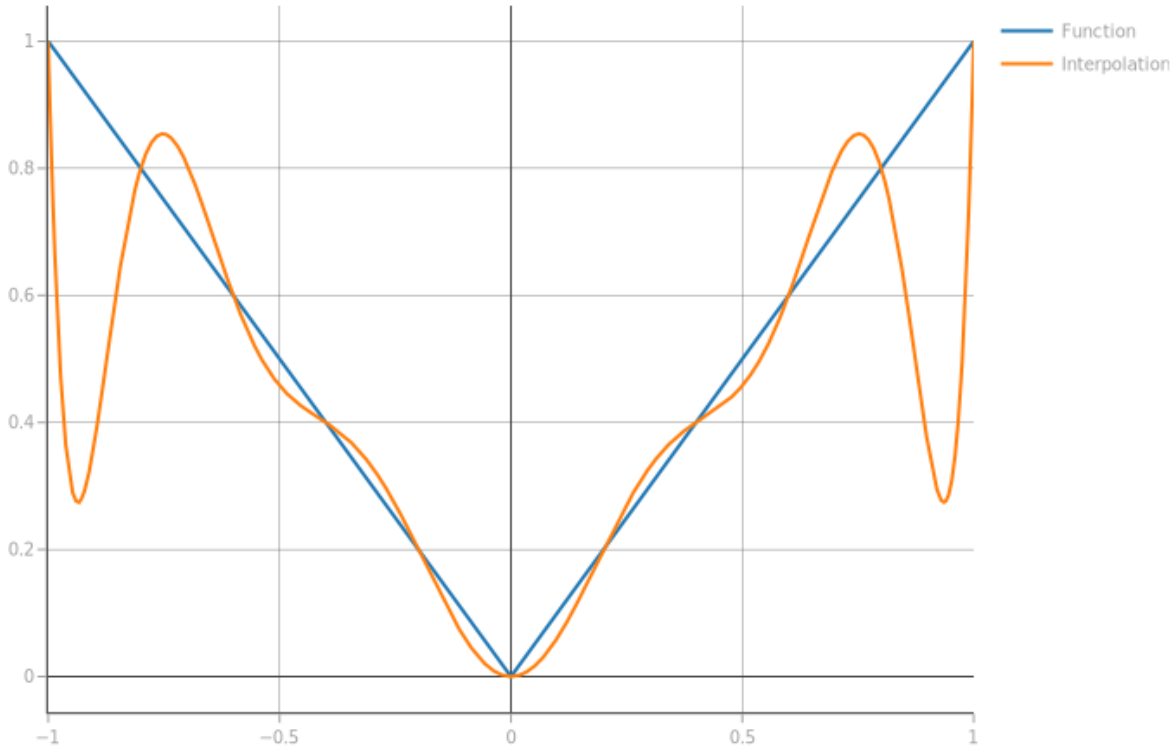
3.2.Wyniki

3.2.1. Funkcja $|x|$

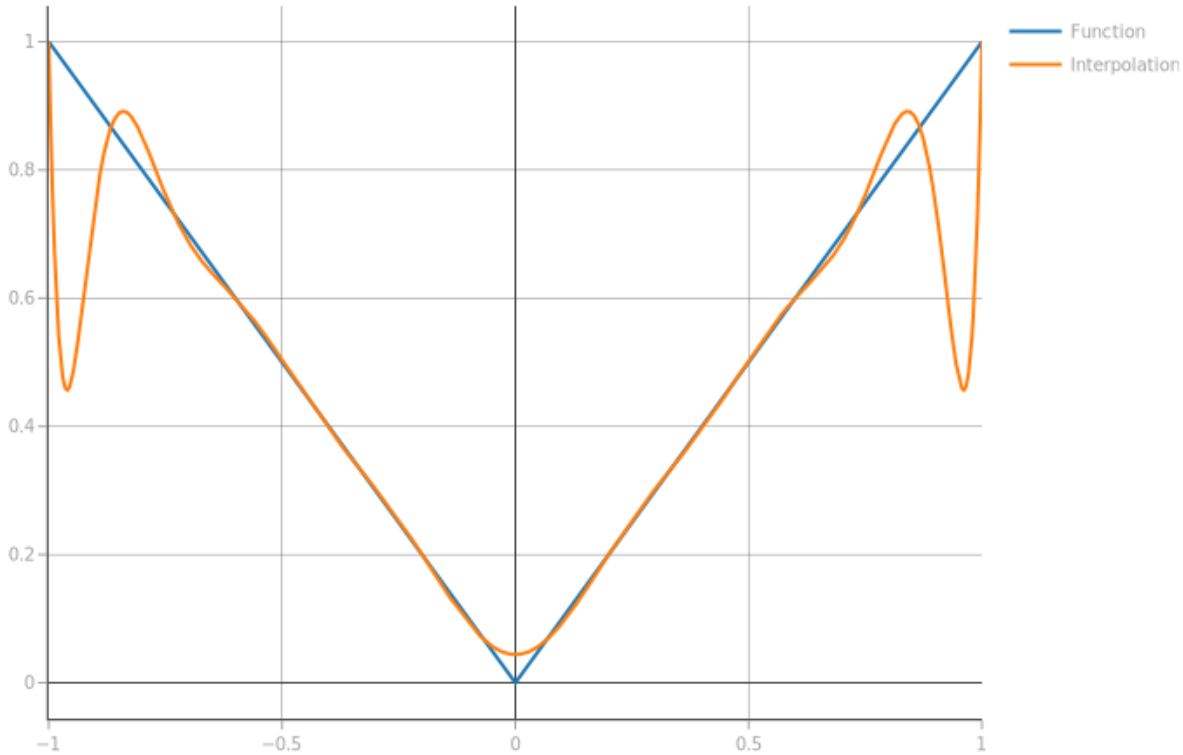
$n = 5$



$n = 10$

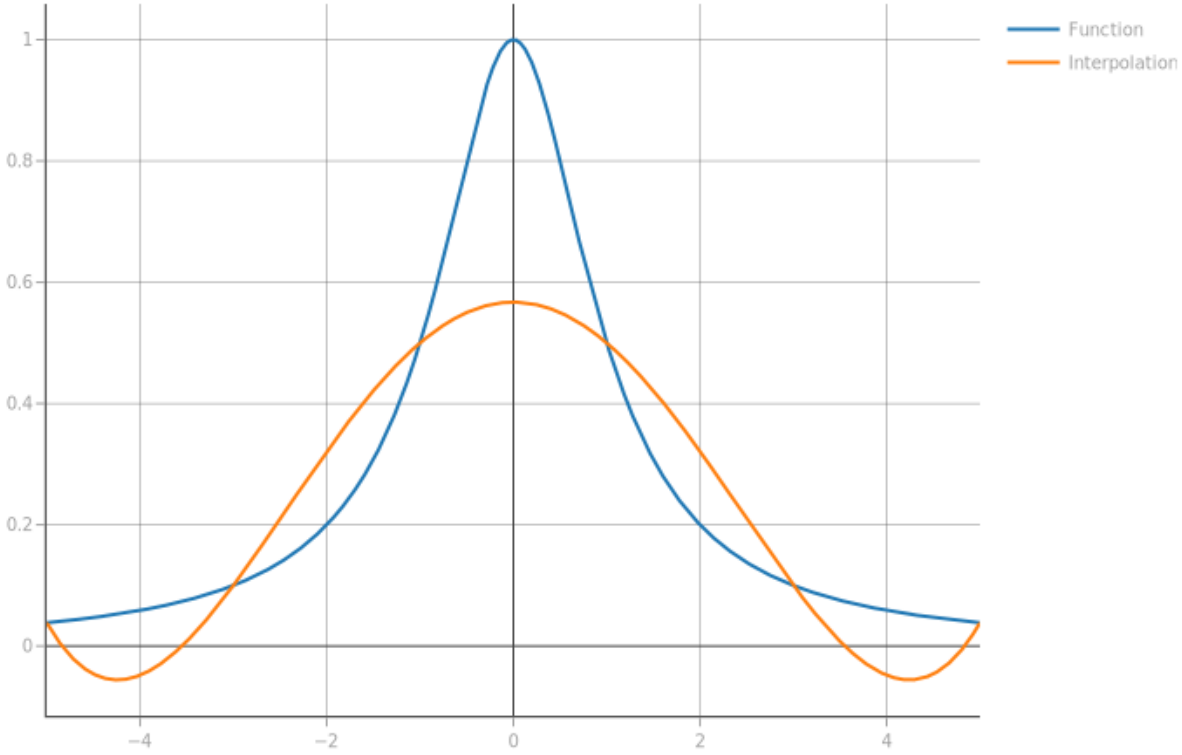

Function
Interpolation
1
0.8
0.6
0.4
0.2
0
−1
−0.5
0
0.5
1

$n = 15$

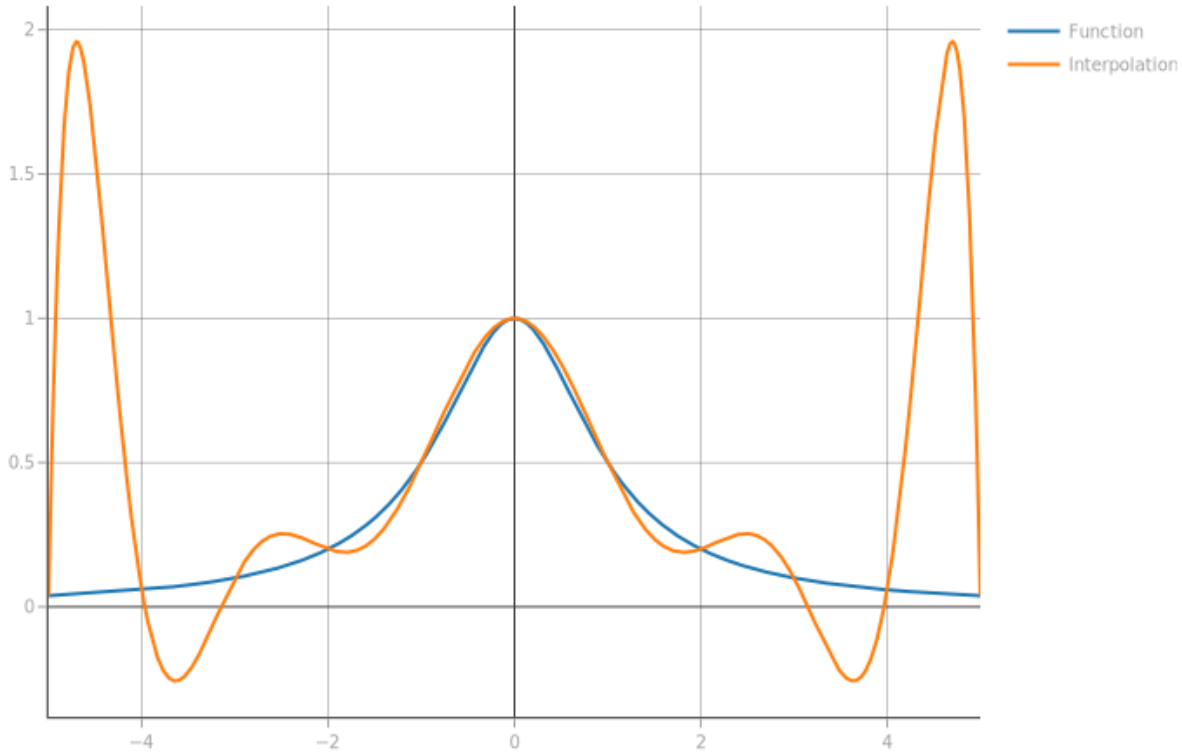

Function
Interpolation
1
0.8
0.6
0.4
0.2
0
−1
−0.5
0
0.5
1

3.2.2. Funkcja $\frac{1}{1+x^2}$

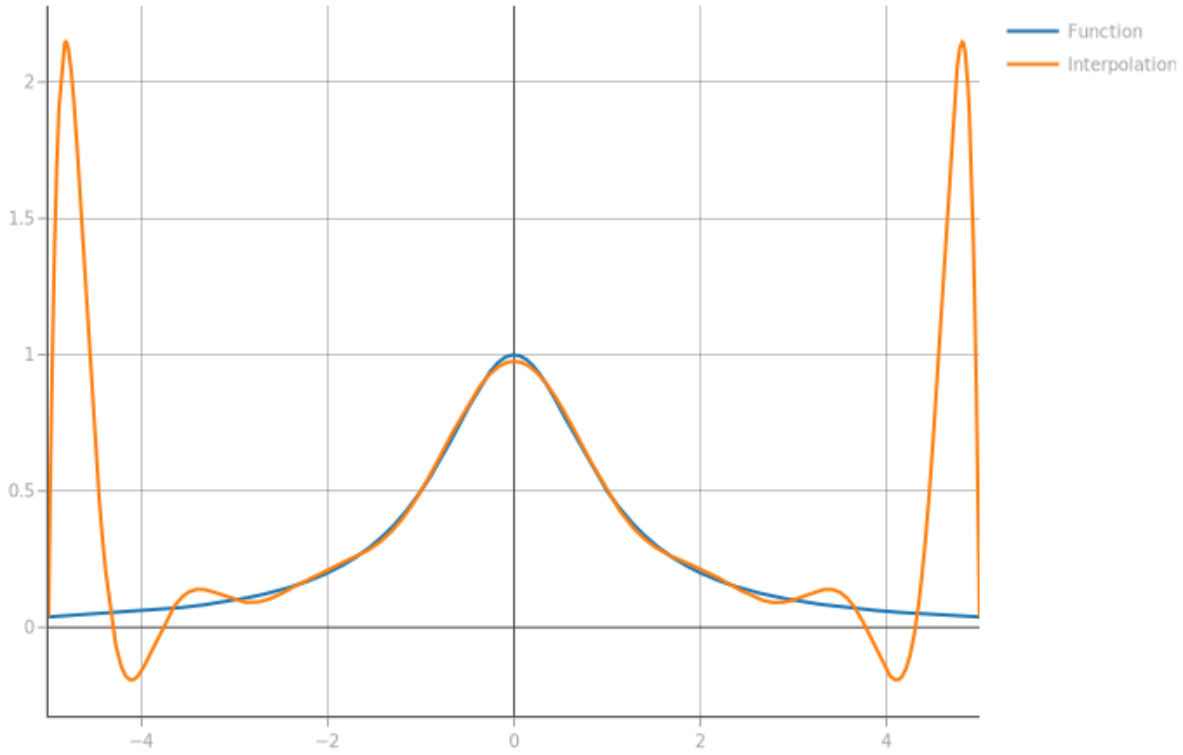
$n = 5$



$n = 10$



$n = 15$



3.3.Wnioski

Jak widać na wykresach załączonych powyżej tym razem nie udało nam się tak ładnie zinterpolować funkcji. Zwiększanie ilości węzłów dało nam lepsze przybliżenia na środku przedziału i jednocześnie coraz to gorsze na jego krańcach. Jest to tak zwane zjawisko Rungego. Jest ono typowe dla interpolacji gdzie zastosowano równoodległe węzły. Aby sobie z tym poradzić należy zwiększyć gęstość węzłów na końcach przedziału. Wtedy zamiast używać węzłów równoodległych używa się węzłów leżących w zerach wielomianów Czebyszewa. Ten problem zanika ponieważ miejsca zerowe wielomianów Czebyszewa zagęszczają się ku krańcom przedziału.