Politechnika Wrocławska
Wydział Podstawowych Problemów Techniki

Obliczenia Naukowe
Sprawozdanie z Laboratorium

Lista 4

Autor:
Krzysztof Nowak
III Rok INF.
Nr Indeksu: 229807

10 grudnia 2017

1 Zadania 1-4

1.1 Opis problemu

Naszym zadaniem było zaimplementowanie 4 funkcji: funkcji do wyliczania ilorazów różnicowych, funkcji do wyznaczania wartości wielomianu interpolacyjnego w postaci Newtona w danym punkcie, funkcji wyznaczającej ze współczynników wielomianu interpolacyjnego w postaci Newtona jego współczynniki w postaci naturalnej, oraz funkcję rysującą, w przedziale $[a, b]$, wielomian interpolacyjny o podanym stopniu oraz interpolowaną funkcję.

1.2 Opis funkcji

1.2.1 ilorazyRoznicowe

Rozpatrzmy wielomian interpolacyjny pewnej funkcji f. Jak dobrze wiemy istnieje tylko jeden taki wielomian p dla danego stopnia interpolacji n taki, że $p(x_i) = f(x_i)$ dla $0 \leq i \leq n$. Intuicja podpowiada nam, że postać takiego wielomianu powinna wyglądać tak:
$p(x) = a_0 + a_1 x + a_2 x^n + \ldots + +a_n x^n$.
Niestety taka reprezentacja wielomianu prowadzi do układu równań z macierzą Vandermonde'a, która to jest źle uwarunkowana. Przedstawmy więc p w inny sposób. Niech:
$q_0(x) = 1$
$q_1(x) = (x - x_0)$
$q_2(x) = (x - x_0)(x - x_1)$
$\vdots$
$q_n(x) = (x - x_0)(x - x_1) \ldots (x - x_{n-1})$

Wtedy to $p(x) = \sum_{j=0}^{n} c_j q_j(x)$. Z faktu, że p spełnia warunki interpolacji, czyli $p(x_i) = \sum_{j=0}^{n} c_j q_j(x_i)$ otrzymujemy układ równań z którego możemy wyznaczyć $c_0, c_1, \ldots, c_n$.
Możemy zauważyć, że c_0 jest zależne od $f(x_0)$, c_1 od $f(x_0)$ i $f(x_1)$ itd. aż do c_n zależnego od f w punktach $x_0, x_1, \ldots, x_n$.
Gdy wprowadzimy notacje $c_n = f[x_0, x_1, \ldots, x_n]$, wtedy to $= f[x_0, x_1, \ldots, x_n]$ jest współczynnikiem przy x^n wielomianu stopnia co najwyżej n interpolującego f w węzłach $x_0, x_1, \ldots, x_n$. Wielkość tą nazywamy ilorazem różnicowym, inaczej nazywanym jako wielkość opisującą przyrost funkcji na danym przedziale.
Poprzez takie przedstawienie wielomianu p otrzymujemy jego postać Newtona. Ilorazy te spełniają następujące własności:
$f[x_0] = f(x_0)$
$f[x_0, x_1] = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$
$f[x_0, x_1, \ldots, x_n] = \frac{f[x_1, x_2, \ldots, x_n] - f[x_0, x_1, \ldots, x_{n-1}]}{x_n - x_0}$
Algorytm wyznaczania ilorazów wygląda następująco:
Najpierw została utworzona funkcja pomocnicza ilorazRoznicowy

Algorithm 1: ilorazRoznicowy

Data: x,f gdzie

x - wektor długości $n+1$ zawierający węzły $x_0, x_1, \ldots, x_n$,

f - wektor długości $n+1$ zawierający wartości interpolowanej funkcji w węzłach $f(x_0), f(x_1), \ldots, f(x_n)$

Result: fx gdzie

fx - iloraz różnicowy $f[x_0, x_1, \ldots, x_n]$

begin
  if $length(x) = 1$ **then**
    return $f[1]$
  else if *length(x) = 2* **then**
    return $\frac{f[2]-f[1]}{x[2]-x[1]}$
  else
    $a \longleftarrow ilorazRoznicowy(x[2, \ldots, length(x)], f[2, \ldots, length(f)])$
    $b \longleftarrow ilorazRoznicowy(x[1, \ldots, length(x)-1], f[1, \ldots, length(f)-1])$
    return $\frac{a-b}{x[length(x)]-x[1]}$
  end
end

Następnie główna funkcja algorytmu:

Algorithm 2: ilorazyRoznicowe

Data: x,f gdzie

x - wektor długości $n+1$ zawierający węzły $x_0, x_1, \ldots, x_n$,

f - wektor długości $n+1$ zawierający wartości interpolowanej funkcji w węzłach $f(x_0), f(x_1), \ldots, f(x_n)$

Result: fx gdzie

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

begin
  for $i = 1$ **to** $length(x)$ **do**
    $fx[i] \longleftarrow ilorazRoznicowy(x[1, \ldots, i], f[1, \ldots, i)]$
  end
  return fx
end

1.2.2 warNewton

Wielomian w postaci Newtona wygląda następująco (otrzymanie wielomianu zostało przedstawione przy omawianiu poprzedniej funkcji):
$p(x) = f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) + \ldots + f[x_0, x_1, \ldots, x_n](x - x_0)(x - x_1) \ldots (x - x_{n-1})$ Możemy wzór ten przekształcić do takiej postaci:
$p(x) = f[x_0] + (x - x_0)(f[x_0, x_1] + (x - x_1)(f[x_0, x_1, x_2] + (x - x_2)(\ldots + (x - x_{n-1})(f[x_0, x_1, \ldots, x_n])) \ldots))$
Zauważmy, że możemy zastosować tutaj algorytm Hornera z drobną modyfikacją taką, że:
$w_n(x) = f[x_0, x_1, \ldots, x_n]$
$w_i(x) = f[x_0, x_1, \ldots, x_i] + (x - x_i)w_{i+1}(x)$ dla $0 \leq i \leq n$
$p(x) = w_0(x)$
Jeżeli za x podstawimy jakąś wartość to możemy otrzymać wartość interpolacji funkcji w punkcie. Algorytm ten prezentuje się następująco:

Algorithm 3: warNewton

Data: x,fx,t gdzie

x - wektor długości $n+1$ zawierający węzły $x_0, x_1, \ldots, x_n$,

fx - wektor długości $n+1$ zawierający ilorazy różnicowe,

t - punkt, w którym należy obliczyć wartość wielomianu

Result: nt , gdzie

nt - wartość wielomianu w punkcie t,

begin

  $w[length(x)] \longleftarrow fx[length(fx)])$

  for $i = length(x) - 1$ **down to** 1 **do**

    $w[i] \longleftarrow fx[i] + (t - x[i])(w[i+1])$

  end

  return $w[1]$

end

1.2.3 naturalna

Każdy z wielomianów pomocniczych $w_i(x)$ z metody Hornera przedstawionej powyżej możemy zapisać w postaci naturalnej:
$w_i(x) = \sum_{j=i}^{n} a_k^{(i)} x^{(k-i)}$
Z tego zapisu dostajemy układ równań rekurencyjnych:
$a_n^{(i)} = a_n^{(i+1)} = \ldots = a_n^{(n)} = f[x_0, x_1, \ldots, x_n]$
$a_i^{(i)} = f[x_0, x_1, \ldots, x_i] - a_{i+1}^{(i+1)} x_i$
$a_k^{(i)} = a_k^{(i+1)} - a_{k+1}^{(i+1)} x_i$
Szukanym wspóczynnikiem postaci naturalnej wielomianu jest $a_k^{(0)}$
Algorytm ten wygląda następująco:

Algorithm 4: naturalna

Data: x,fx,t gdzie

x - wektor długości $n+1$ zawierający węzły $x_0, x_1, \ldots, x_n$,

fx - wektor długości $n+1$ zawierający ilorazy różnicowe,

Result: a , gdzie

a - wektor długości $n+1$ zawierający obliczone współczynniki postaci naturalnej

begin

  $a[length(x)] \longleftarrow fx[length(fx)])$

  for $i = length(x) - 1$ **down to** 1 **do**

    $a[i] \longleftarrow fx[i]$

    for $j = i$ **to** $length(x) - 1$ **do**

      $a[j] \longleftarrow a[j] - a[j+1]x[i]$

    end

  end

  return a

end

1.2.4 rysujNnfx

Mając dany przedział $[a, b]$, funkcję dla której tworzymy wielomian interpolacyjny, oraz podany stopień n wielomianu interpolacyjnego, tworzymy $n + 1$ węzłów równoodległych od siebie, następnie wyliczamy $f(x_i)$ dla każdego węzłu x_i. Mając zarówno węzły jak i wartości funkcji w tych węzłach wyznaczamy ilorazy różnicowe, a potem wartości wielomianu interpolacyjnego Newtona w tych węzłach. Po obliczeniu wszystkiego rysujemy wykres zawierający funkcję oraz jej interpolację wykorzystując funkcję plot z pakietu Plots dostępnego w języku programowania Julia.

Algorithm 5: naturalna

Data: f,a,b,n gdzie
  x - funkcja $f(x)$ zadana jako anonimowa funkcja ,
  a - początek przedziału,
  b - koniec przedziału,
  n - stopień wielomianu interpolacyjnego

Result: funkcja rysuje wielomian interpolacyjny i interpolowaną funkcję w przedziale $[a, b]$

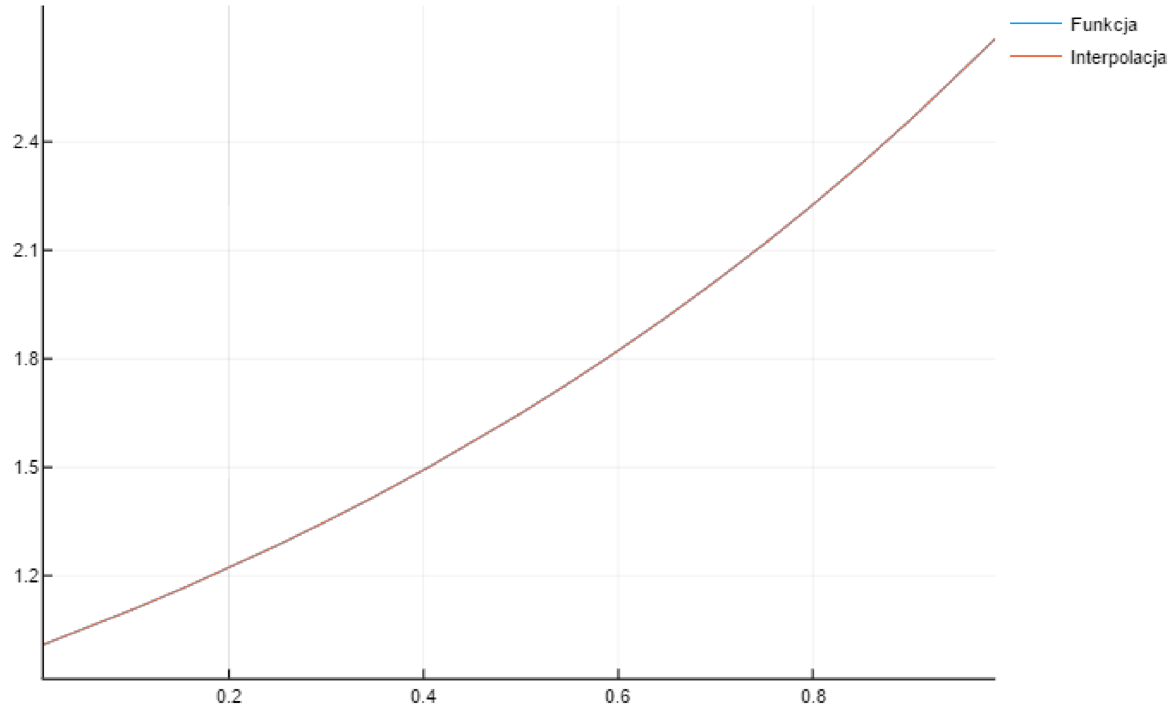
begin
  for $i = 0$ **to** n **do**
    $x[i + 1] \longleftarrow a + i * ((b - a)/n)$
    $fx[i + 1] \longleftarrow f(x[i + 1])$
  end
  $ilorazy \longleftarrow ilorazyRoznicowe(x, fx)$
  $fi \longleftarrow y \rightarrow warNewton(x, ilorazy, y)$ `/* Przypisanie warNewton jako funkcji anonimowej */`
  $plot(f, label = "Funkcja", a, b)$
  $plot!(fi, label = "Interpolacja", a, b)$
end

2 Zadanie 5

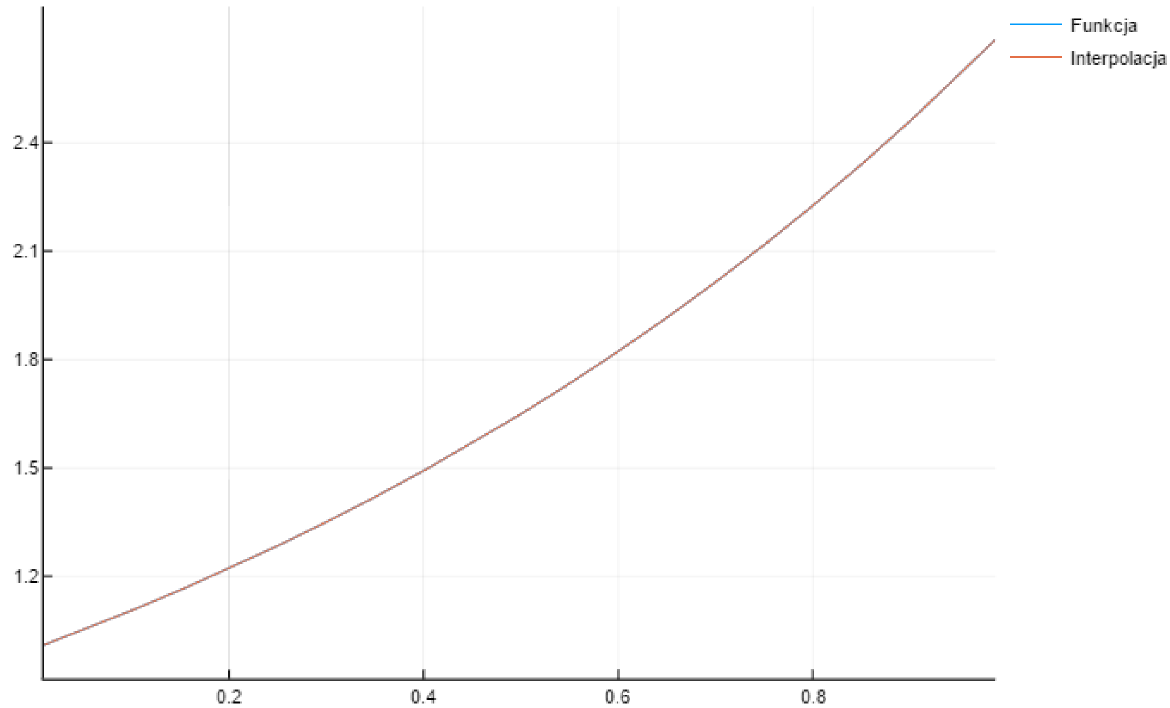
2.1 Opis problemu

Naszym zadaniem było wykorzystać napisane na poprzednich zadaniach metody do narysowania wykresów funkcji oraz ich interpolacji dla e^x na przedziale $[0, 1]$ oraz $x^2 \sin(x)$ na przedziale $[-1, 1]$. W obu przypadkach mieliśmy narysować wykresy dla zadanych stopni wielomianu $n = 5, 10, 15$.

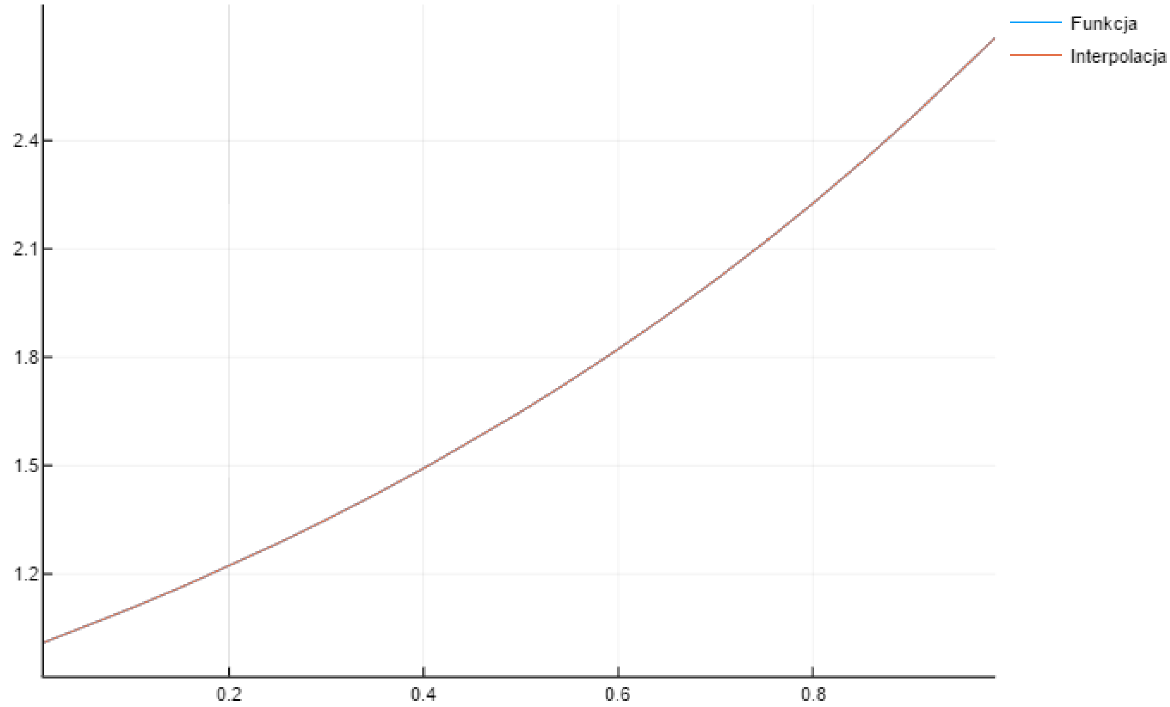
2.2 Rozwiązania i wyniki



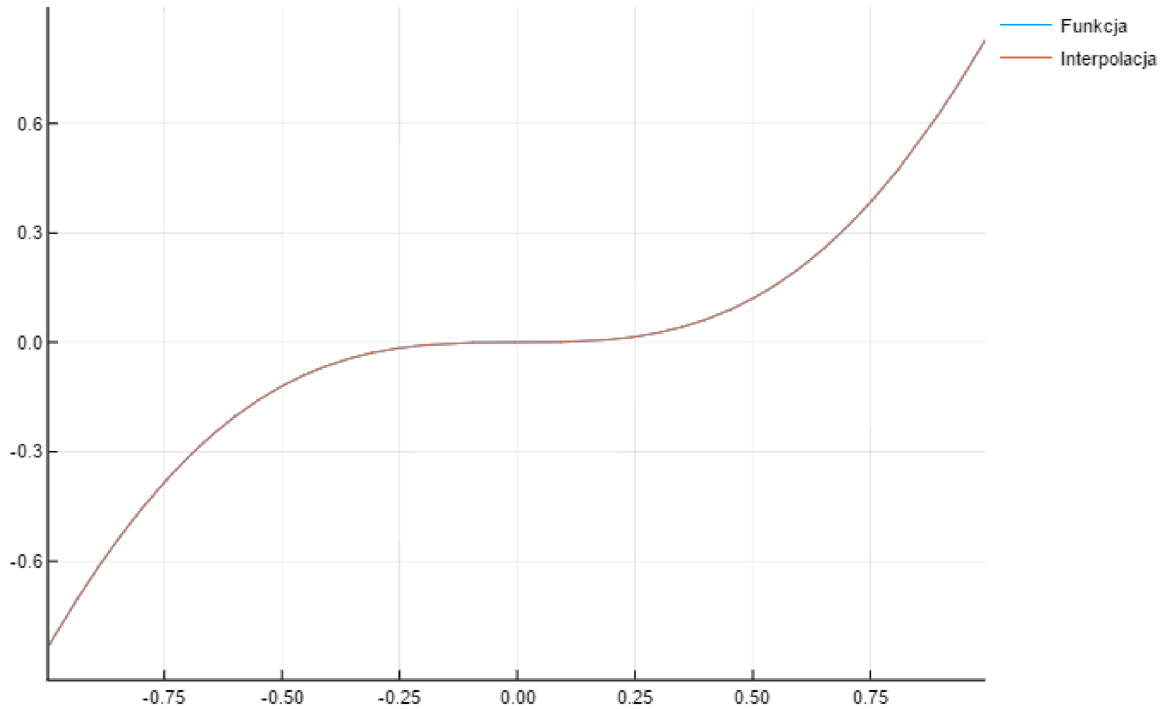
Rysunek 1: Wykres dla e^x i $n = 5$



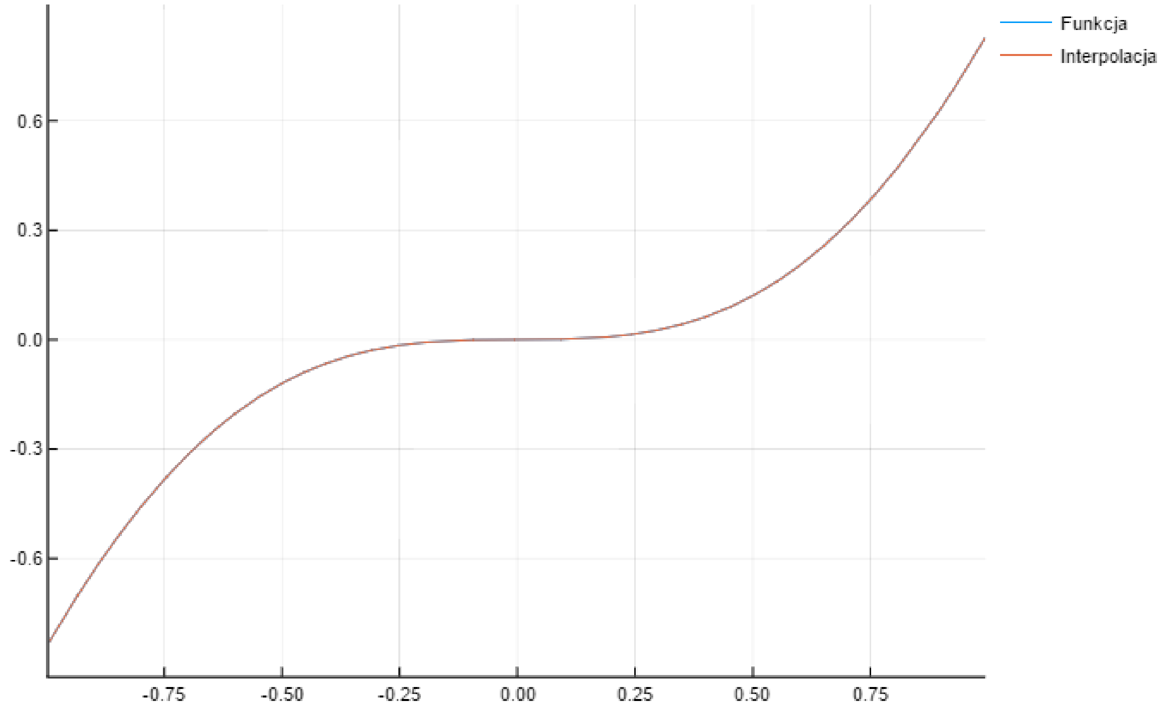
Rysunek 2: Wykres dla e^x i $n = 10$



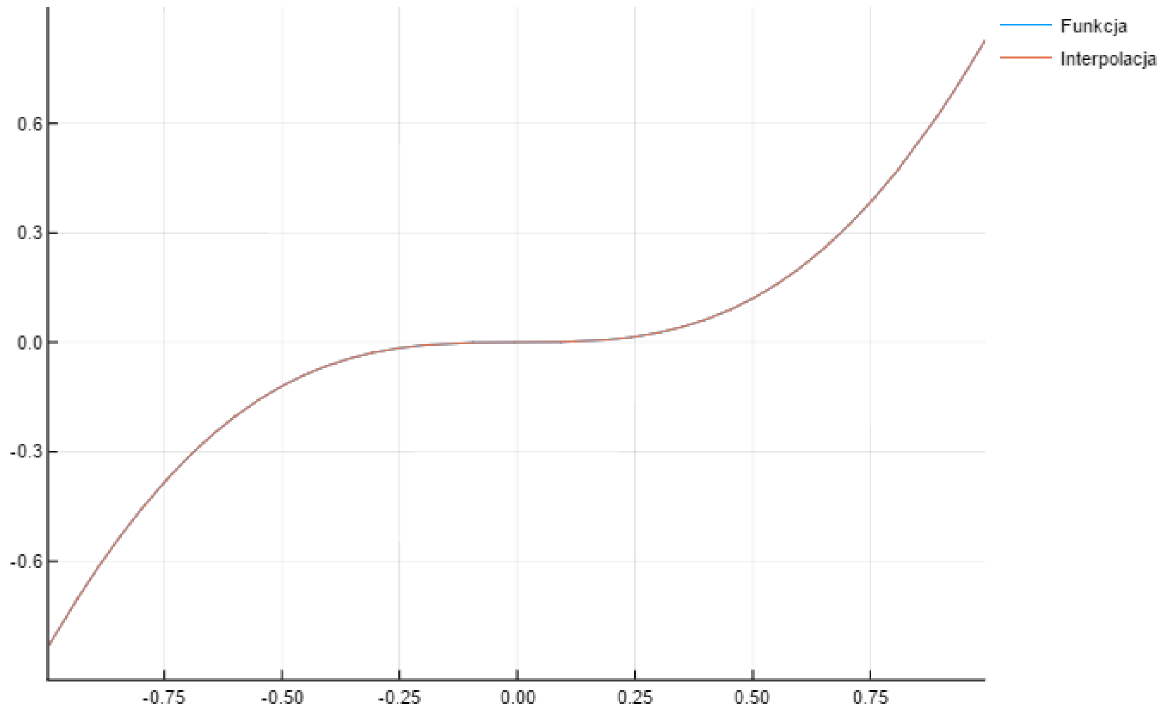
Rysunek 3: Wykres dla e^x i $n = 15$



Rysunek 4: Wykres dla $x^2 \sin(x)$ i $n = 5$



Rysunek 5: Wykres dla $x^2 \sin(x)$ i $n = 10$



Rysunek 6: Wykres dla $x^2 \sin(x)$ i $n = 15$

2.3 Wnioski

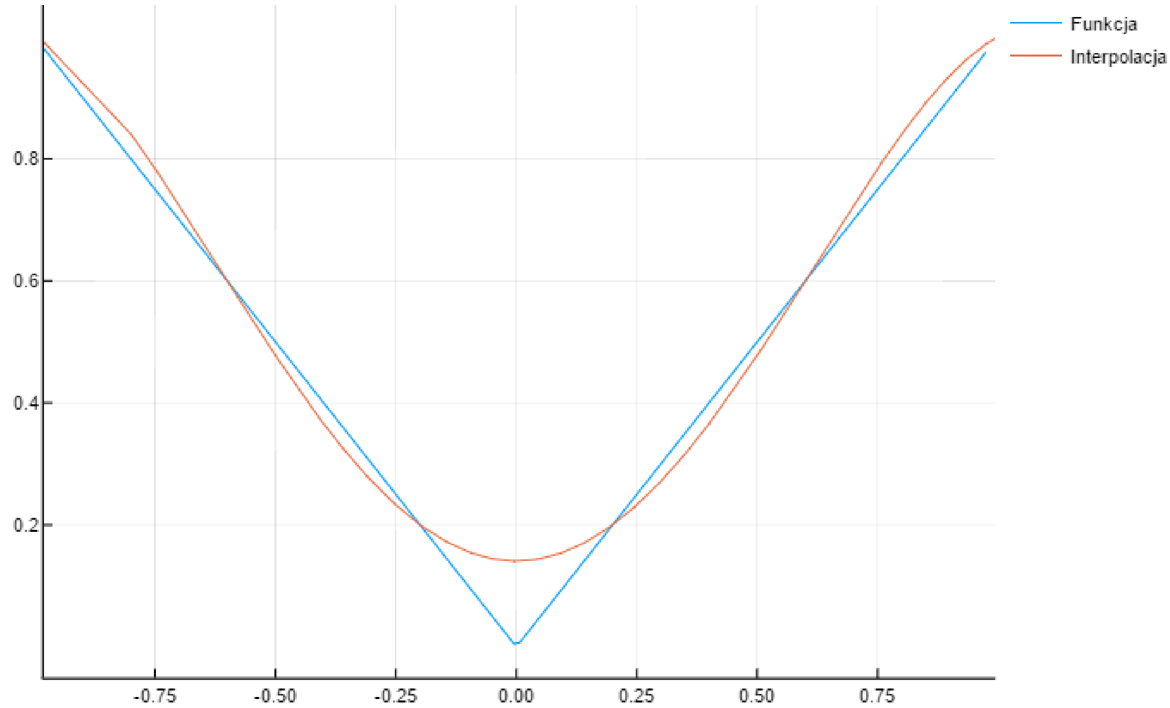
W obu przypadkach dla podanych stopni wielomianu oraz przedziałów, interpolacja prawie pokrywa się z funkcją. Różnice są na wykresach niezauważalne, więc widzimy na nich jedną pogrubioną linię. Jednak można zauważyć, że wraz ze wzrostem stopnia, wzrasta też dokładność interpolacji funkcji.

3 Zadanie 6

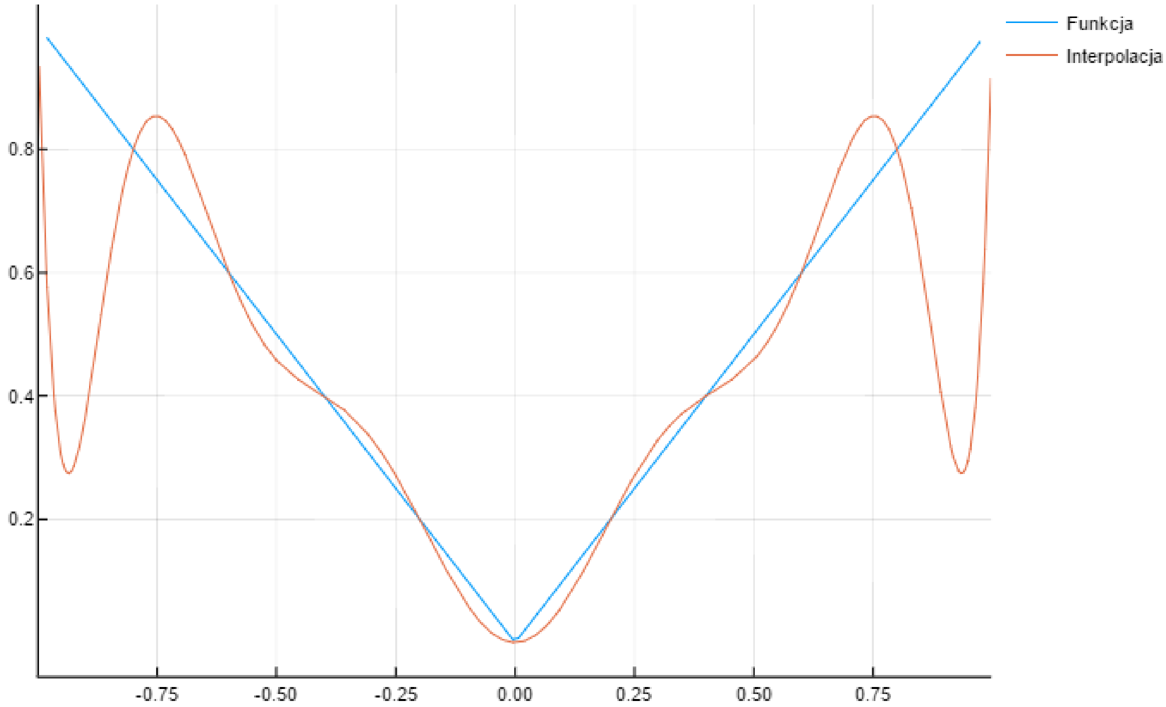
3.1 Opis problemu

Naszym zadaniem było wykorzystać napisane na poprzednich zadaniach metody do narysowania wykresów funkcji oraz ich interpolacji dla $|x|$ na przedziale $[-1, 1]$ oraz $\frac{1}{1+x^2}$ na przedziale $[-5, 5]$. W obu przypadkach mieliśmy narysować wykresy dla zadanych stopni wielomianu $n = 5, 10, 15$.

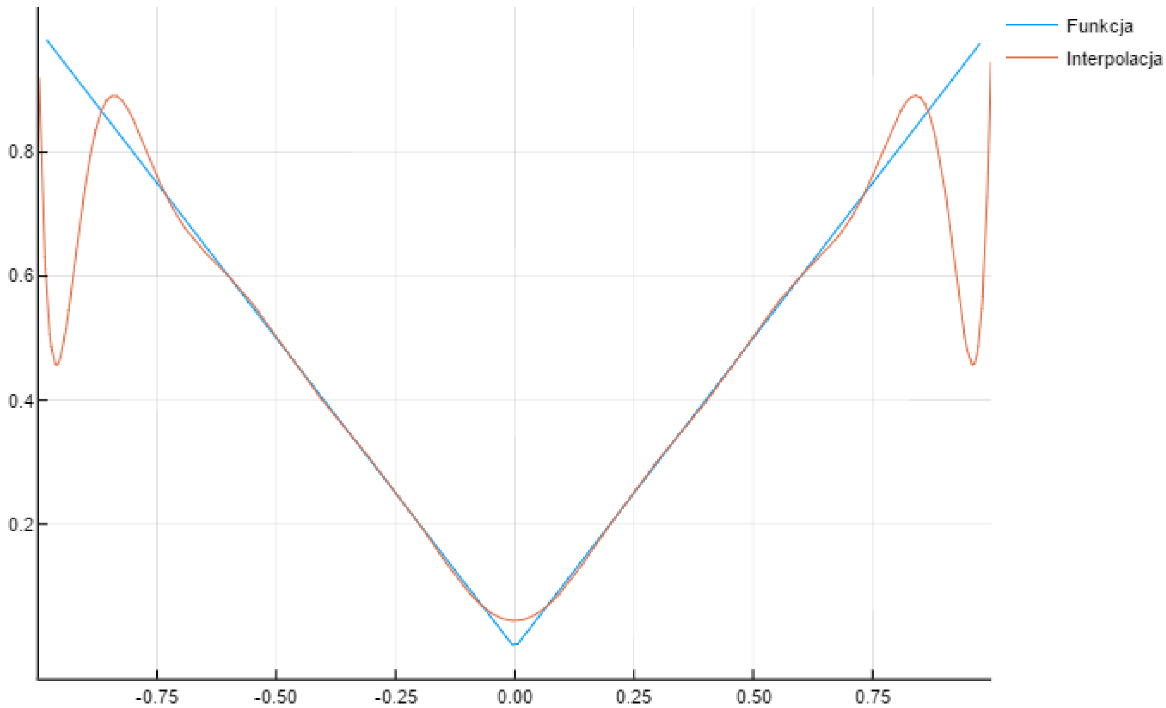
3.2 Rozwiązania i wyniki



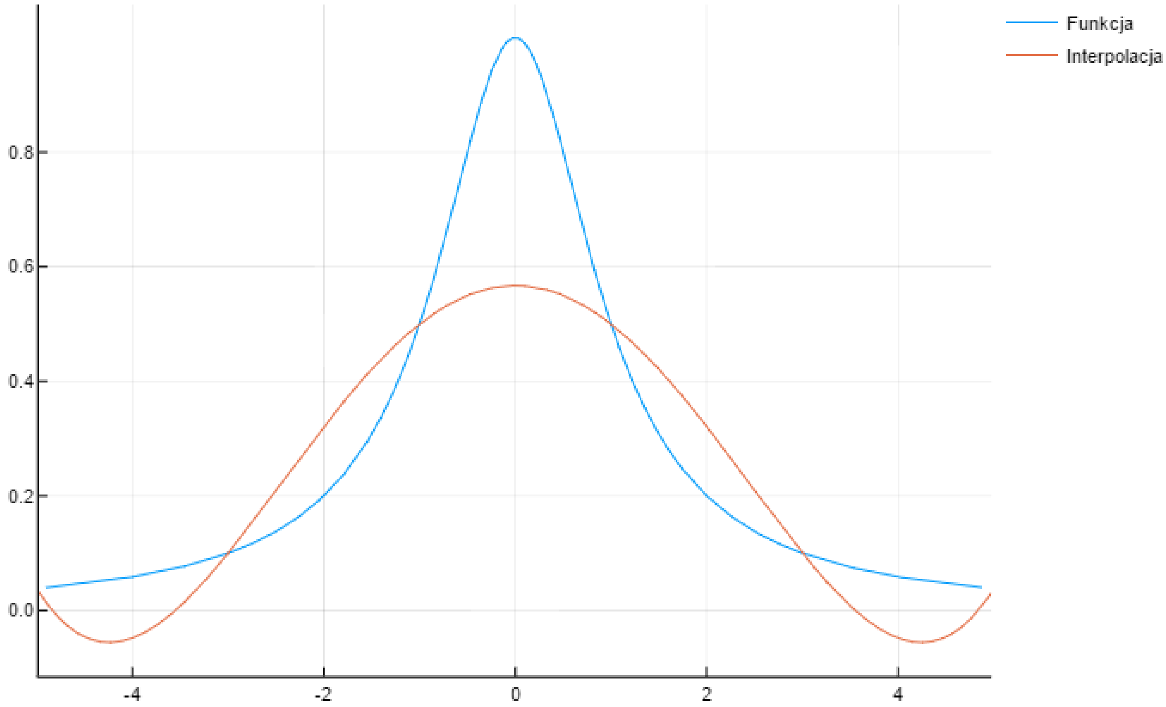
Rysunek 7: Wykres dla $|x|$ i $n = 5$



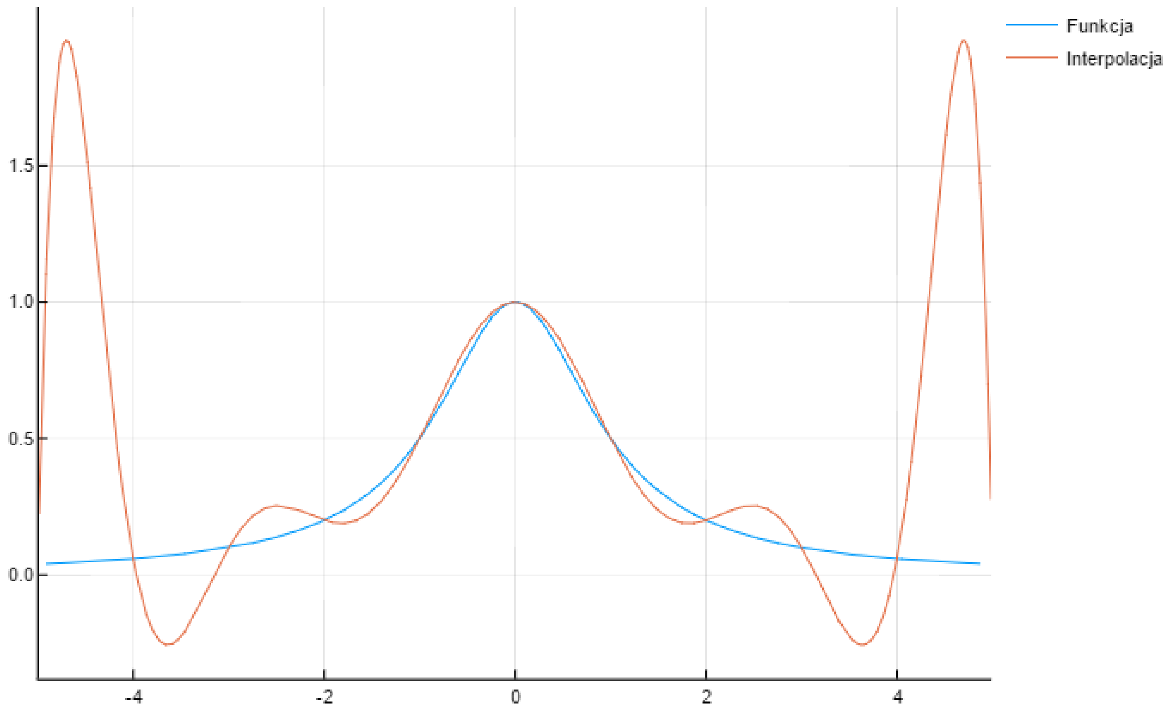
Rysunek 8: Wykres dla $|x|$ i $n = 10$



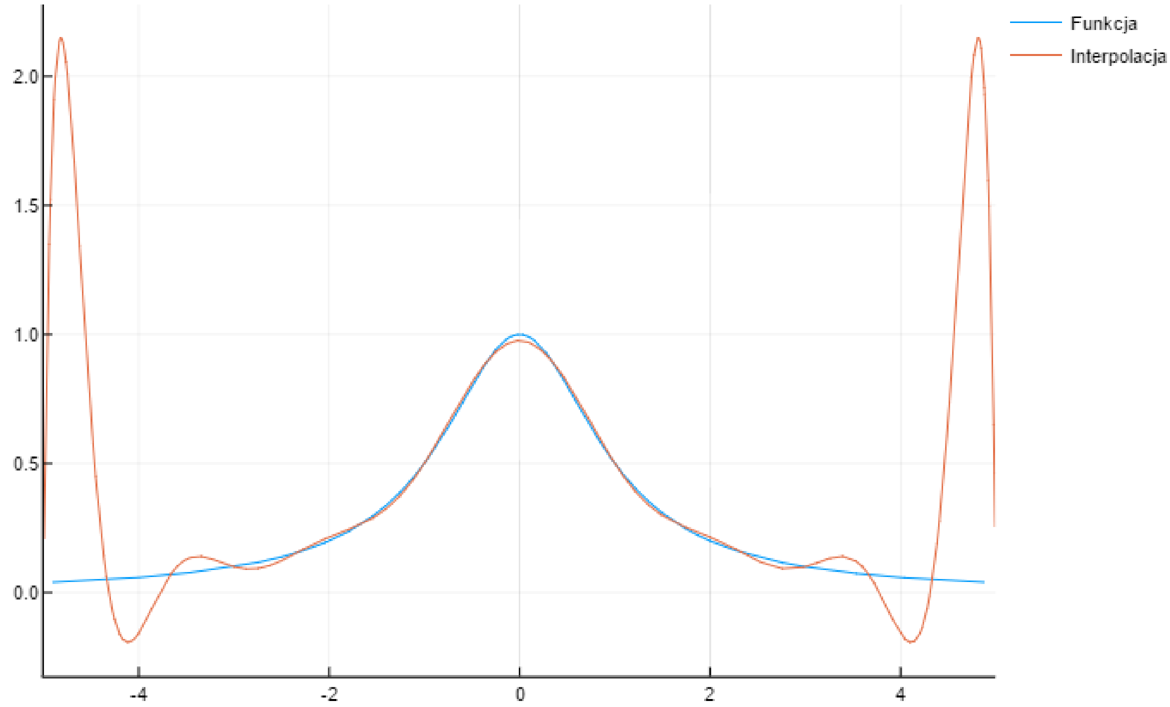
Rysunek 9: Wykres dla $|x|$ i $n = 15$



Rysunek 10: Wykres dla $\frac{1}{1+x^2}$ i $n = 5$



Rysunek 11: Wykres dla $\frac{1}{1+x^2}$ i $n = 10$



Rysunek 12: Wykres dla $\frac{1}{1+x^2}$ i $n = 15$

3.3 Wnioski

Tak jak w poprzednim zadaniu wzrost stopnia poprawia się dokładność interpolacji. Jednakże na końcach przedziału zaczyna dochodzić do dziwnych anomalii. Zjawisko to nazywa się zjawiskiem Rungego (znanym również jako Efekt Rungego). Dochodzi do niego dla wielomianów interpolujących z wysokim stopniem przy stałych odległościach węzłów. Aby zapobiec pojawieniu się tego efektu powinno się stosować interpolację z węzłami gęściej upakowanymi na krańcach przedziału interpolacji. Miejsca zerowe wielomianu Czebyszewa n-tego stopnia są dobrym doborem węzłów dla wielomianu interpolacyjnego n-tego stopnia.