# Obliczenia naukowe lista nr 4

Sebastian Woźniak 268491

November 2023

Na liście mamy do czynienia z problemem interpolacji. Interpolacja to aproksymacja wartości funkcji (funkcja interpolowana) w jakimś zakresie zmiennych na podstawie części wartości z tego zakresu. Wartości te nazywamy węzłami. Powstała aproksymacja nazywa się funkcją interpolacyjną lub wielomianem interpolacyjnym.

## 1 Ilorazy różnicowe

Ilorazem różnicowym nazywamy $f[x_0, ..., x_n] := \Sigma_{i=0}^{n} \frac{f(x_i)}{\Pi_{j=0}^{n}(x_i - x_j)}$.
Algorytm na wejściu przyjmuje x ( wektor długości $n+1$ zawierający węzły $x_1, x_2, ..., x_{n+1}$ ) oraz f ( wektor o długości $n+1$ zawierający wartości interpolowanej funkcji):

```
fx = copy(f)
for i in 2:n
    for j in n:-1:i
        fx[j] = (fx[j]-fx[j-1])/(x[j]-x[j-i+1])
     end
end
return fx
```

Tym sposobem otrzymuemy wektor $fx$ długości $n+1$ zawierający obliczone ilorazy różnicowe.$f[x_i, ..., x_{i+j}] = \frac{f[x_{i+1}, ..., x_{i+j}] - f[x_i, ..., x_{i+j-1}]}{x_{i+j} - x_i}$. Przyjmując oznaczenie $f[x_i, ..., x_{i+j}] = f_{i,i+j}$ z każdym obiegiem wewnętrznej pętli wyznaczamy $f_{j-i+2,j}$, z którego pomocą w następnej iteracji pętli zewnętrznej możemy wyznaczyc $f_{j-i+1,j}$. Finalnie otrzymujemy wektor o długości $n+1$ wyglądający następująco - $[f_{1,1}, ..., f_{1,n+1}]$, czyli szukanymi ilorazami różnicowymi.

$$\begin{array}{c} f[x_0]\ f[x_1]\ f[x_2] \\ \searrow\downarrow\ \searrow\downarrow \\ f[x_0, x_1]\ f[x_1, x_2] \\ \searrow\downarrow \\ f[x_0, x_1, x_2] \end{array}$$

## 2 Wartość wielomianu interpolacyjnego

Z pomocą uogólnionego schmatu Hornera posiadając węzły $x_0, ..., x_n$ oraz ilorazy różnicowe $f[x_0], ..., f[x_0, ..., x_n]$ jestśmy w stanie obliczyć wartość wielomianu interpolacyjnego w punkcie $t$ w czasie liniowym.
Przyjmujemy $n(t)_n = f[x_0, ..., x_n]$ oraz, że $n(t)_k = f[x_0, ..., x_k] + (t - x_k) \cdot n(t)_{k+1}$ wtedy $f(t) = n(t)_0$. Obliczenia wykonujemy w jednej pętli:

```
nt = fx[n] # f[x_0,...,x_n]
for i in n-1:-1:1
    nt = fx[i] + (t-x[i]) * nt
 end
```

## 3 Współczynniki postaci naturalnej

Znając węzły $x_0, ..., x_n$ funkcji interpolowanej oraz ilorazy różnicowe $f[x_0], ..., f[x_0, ..., x_n]$ jesteśmy w stanie obliczyć współczynniki funkcji w postaci naturalnej. Postać naturalna funkcji $f$ to:

$$f(x) = \Sigma_{i=0}^{n} a_i \cdot x^i$$

Używając uogólnionego schematu Hornera w dwóch pętlach o conajwyżej $n$ iteracjach w czasie $O(n^2)$ otrzymujemy współczynniki:

```
a[n] = fx[n]
for i in (n-1):-1:1
    a[i] = fx[i] - x[i] * a[i+1]
    for j in (i+1):(n-1)
        a[j] += -x[i] * a[j+1]
    end
end
```

Między 1, a 3 wierszem widzimy znajomy z poprzedniego algorytmu schemat Hornera. Następnie aktualizujemy obliczone współczynniki o przeciwność iloczynu $i$-tego węzła z $a_{j+1}$ współczynnikiem dla $a_j$. Te obliczenia odpowiadają iteracyjnemu kumulowaniu kolejnych potęg $x$ do odpowiednich współczynników.
$f(x) = (((...(a_n x + a_{n-1})x + a_{n-2})x + ...)x + a_1)x + a_0$

## 4 Wyznaczanie wielomianu interpolacyjnego

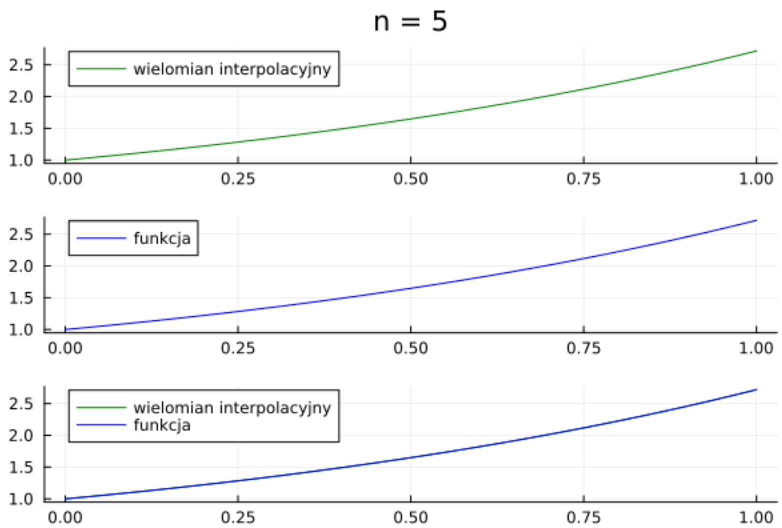
Używając poprzednio napisanych funkcji ilorazyRoznicowe() oraz warNewton() możemy wyznaczyć wielomian interpolacyjny. Funkcja rysujNnfx tworzy także wykres porównujący funkcje interpolowaną do wielomiany interpolacyjnego.

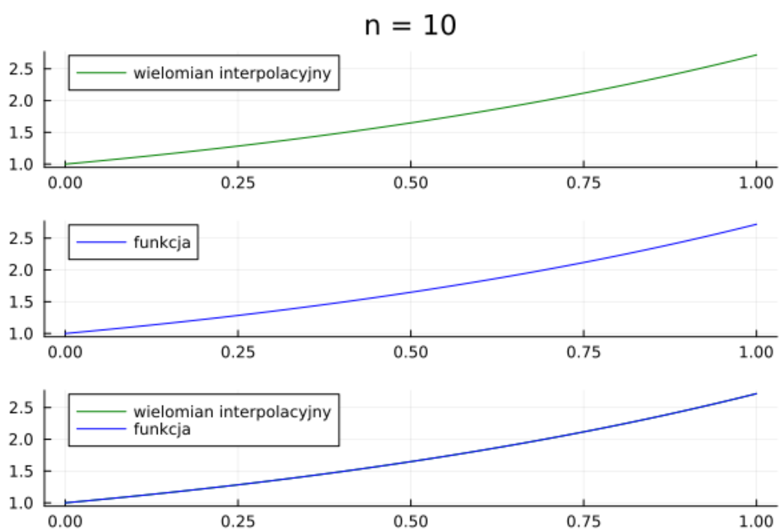
```
x = zeros(n+1)
y = zeros(n+1)
h = (b-a)/n
for k in 0:n
    x[k+1] = a + k*h
    y[k+1] = f(x[k+1])
end

c = ilorazyRoznicowe(x, y)
no_points = 30 *n
h = (b-a)/(no_points)

plot_x = zeros(no_points+1)
intp = zeros(no_points+1)
func = zeros(no_points+1)

plot_x[1] = a
intp[1] = func[1] = y[1]
for i in 2:no_points+1
    plot_x[i] = plot_x[i-1] + h
    intp[i] = warNewton(x, c, plot_x[i])
    func[i] = f(plot_x[i])
end
```

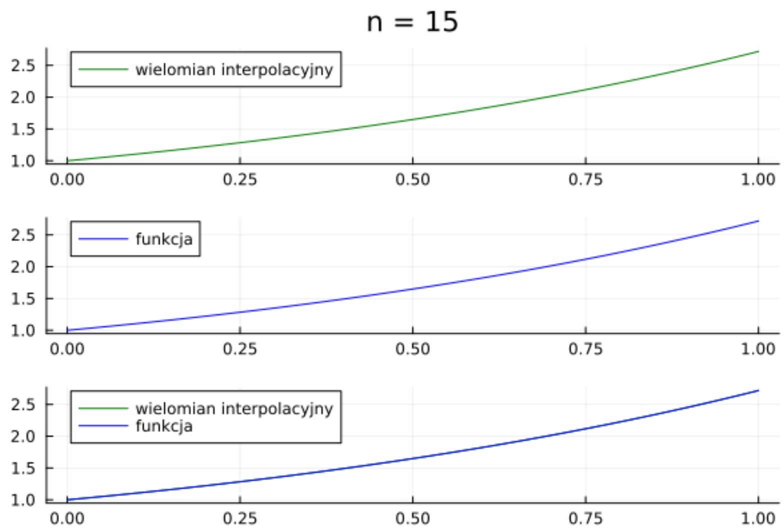
Zaczynamy od wyznacznia $n+1$ węzłów między którymi jest równa odległość $\frac{(b-a)}{n}$. Wyznaczamy także wektor $y$ przechowujący wartości funkcji $f$ w węzłach $x_0, ..., x_n$. Funkcja ilorazyRoznicowe z tymi danymi zwraca nam ilorazy różnicowe funkcji $f$, które pozwalają funkcji warNewton wyznaczyć wartości funkcji. Liczbą punktów na wykresie dla obu funkcji będzie $30 \cdot n$ a odległość między nimi $\frac{(b-a)}{30 \cdot n}$. Tablica plot_x przechowuje te punktu do prezentacji osi X na wykresach. W tablicy func znajdują się rzeczywiste wartości funkcji $f(x)$, a w tablicy intp wartości wielomianu interpolacyjnego.

# 5 Testy 1

## 5.1 $e^x$

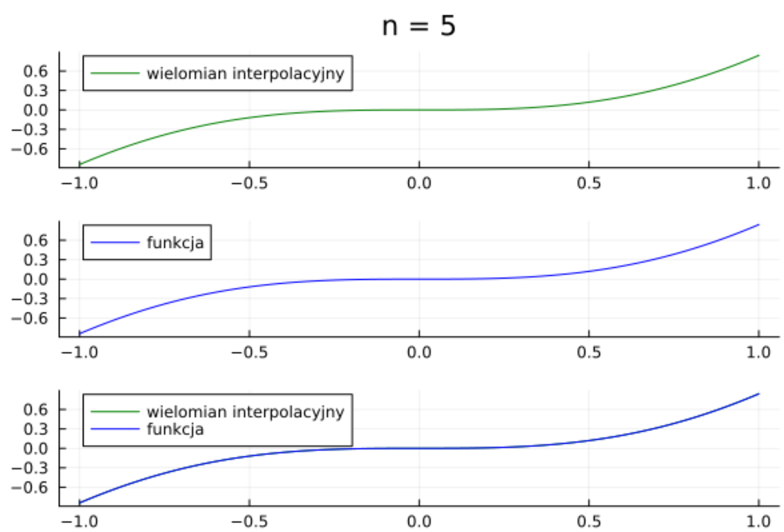


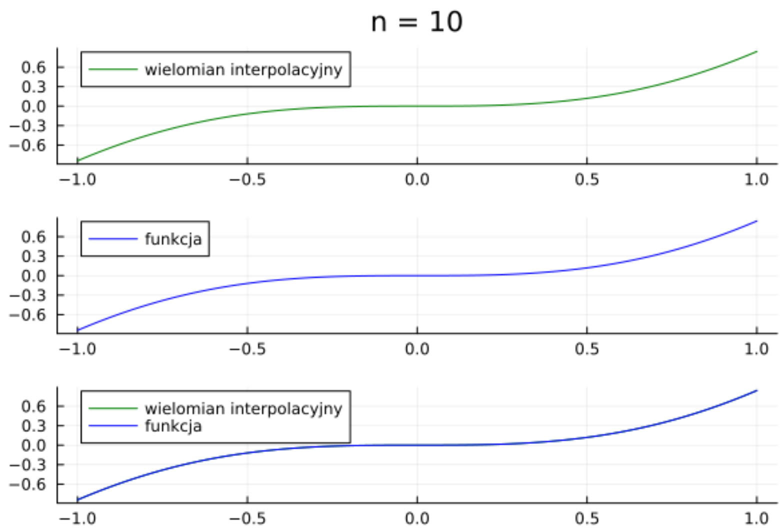



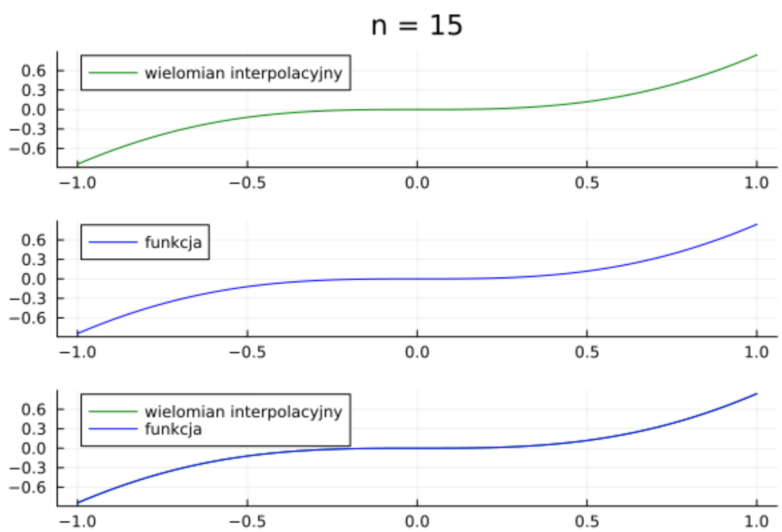



Funkcja bardzo dobrze daje się interpolować i wykresy funkcji oraz wielomianu interpolacyjnego pokrywają się.

### 5.2 $x^2 \cdot \sin(x)$



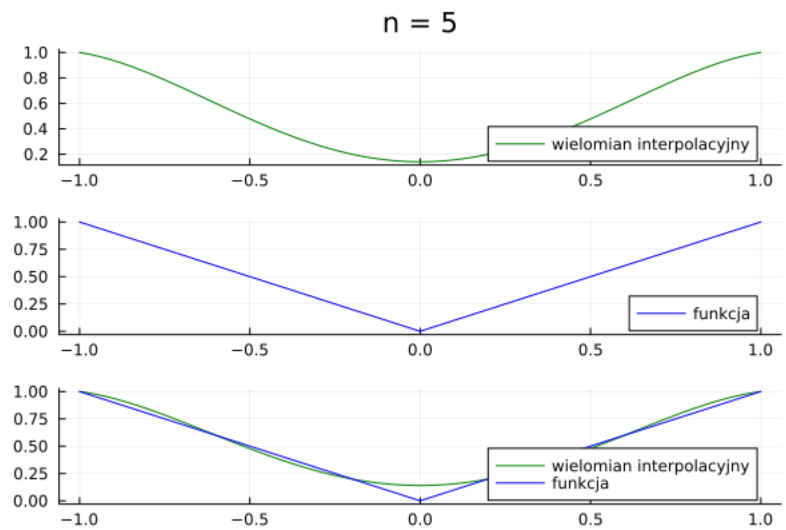


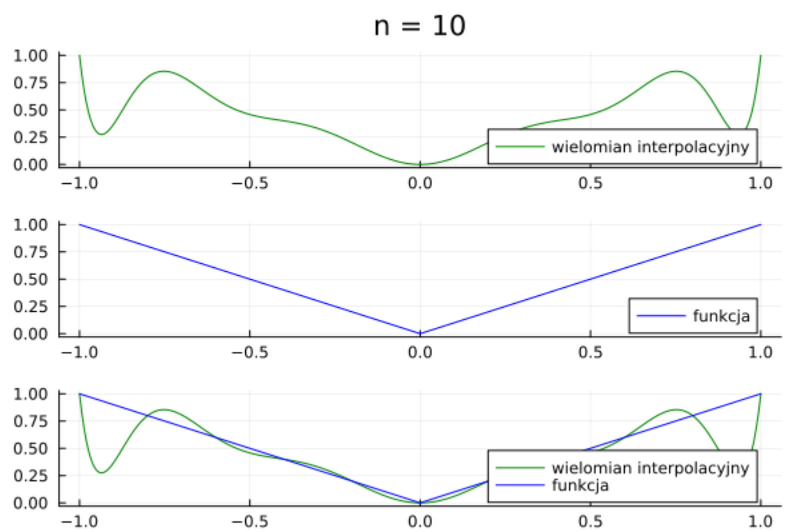



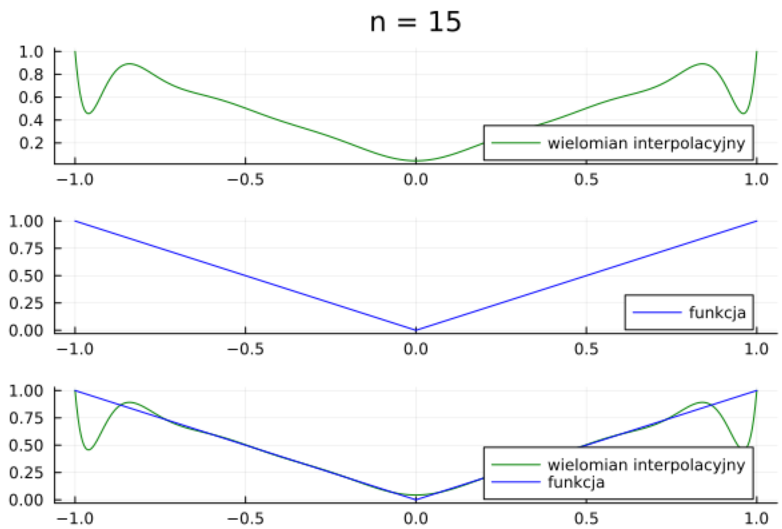

Dla tej funkcji otrzymaliśmy podobny efekt. Interpolowana funkcja niemalże pokrywa się z wielomianem interpolującym dla wszytkich wartości $n$.

# 6 Testy 2 - Zjawisko rozbieżności

## 6.1 $|x|$

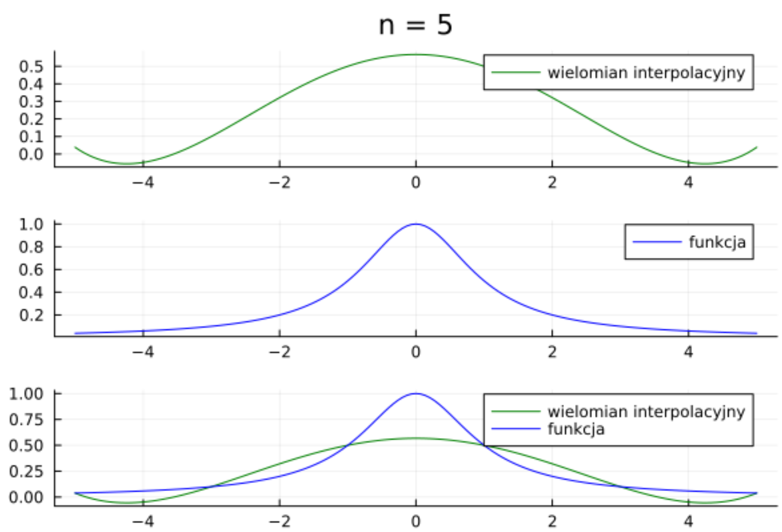


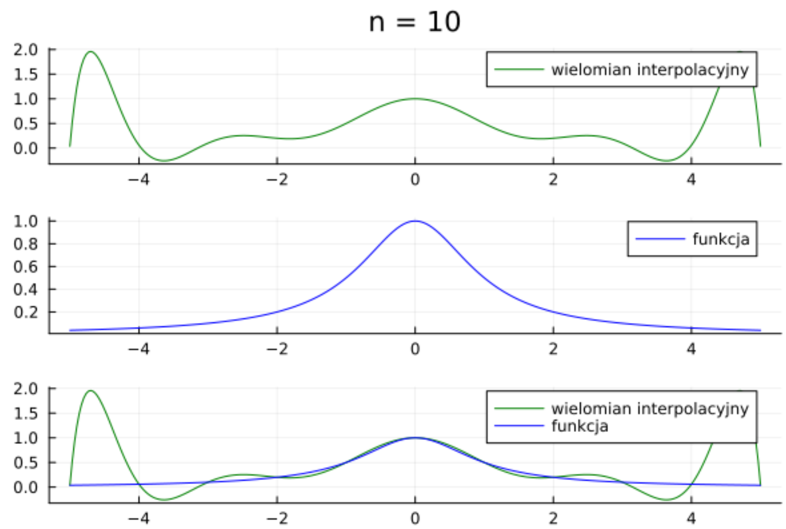



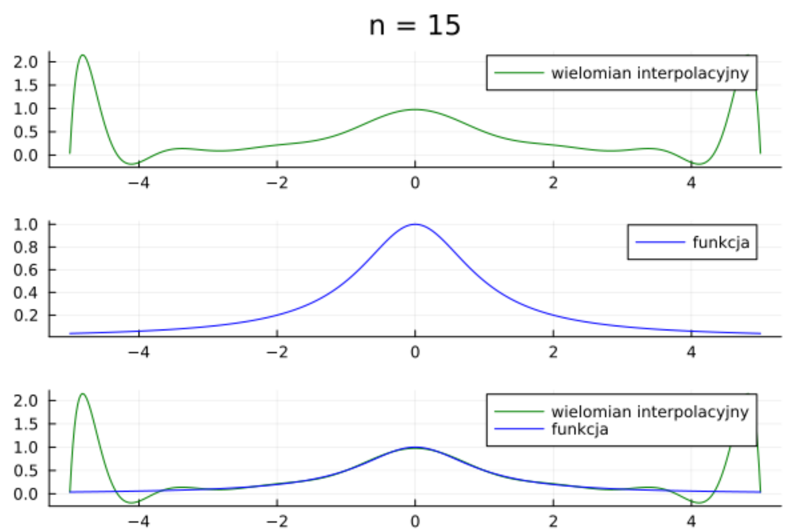



W tym przypadku funkcja nie interpoluje się tak dobrze. Problemem jest nieróżniczkowalnośc funkcji. Ostre krawędzie wykresu funkcji są nieosiągalne dla wielomianu interpolacyjnego.

### 6.2 $\frac{1}{1+x^2}$






Dla tej funkcji obserwujemy zjawisko, w którym wraz ze zwiększaniem liczby węzłów otrzymujemy coraz mniej dokładne interpolacje krańców przedziału. Aby temu zapobiec należałoby gęściej rozłożyć węzły w tych miejscach.

# 7 Wniosek

Interpolacja jest skuteczną metodą aproksymacji funkcji, gdy mamy do czynienia z funkcją, która nie posiada ostrych kątów. Należy także pamiętać, że zwiększenie ilości węzłów nie zawsze pomoże nam uzyskać dokładniejsze wyniki. Czasami większe znaczenie ma ich poprawne rozstawienie.