

# Obliczenia Naukowe Lista nr: 4

Bartosz Banasik

10 grudnia 2017

# 1 Zadanie 1

## 1.1 Opis problemu:

**Polecenie:** Napisać funkcję obliczającą ilorazy różnicowe.

**Dane:**  $x$  - wektor długości  $n + 1$  zawierający węzły  $x_0, \dots, x_n$

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

**Wyniki:**  $fx$  - wektor długości  $n + 1$  zawierający obliczone ilorazy różnicowe

## 1.2 Rozwiązanie:

Ilorazy różnicowe spełniają zależność:  $f[x_0, x_1, \dots, x_k] = \frac{f[x_1, x_2, \dots, x_k] - f[x_0, x_1, \dots, x_{k-1}]}{x_k - x_0}$

Ponadto wiemy, że  $f[x_0] = f(x_0)$ ,  $f[x_0, x_1] = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$ .

Stosując powyższe stwierdzenia jesteśmy w stanie zaproponować algorytm obliczający ilorazy różnicowe używając jedno wymiarowej tablicy. Kod źródłowy znajduje się w pliku *ilorazyRoznicowe.jl*.

---

**Algorithm 1** *ilorazyRoznicowe*( $x, f$ )

---

```
for  $i \leftarrow 0$  to  $n$  do
     $d_i \leftarrow f(x_i)$ 
end for
for  $j \leftarrow 1$  to  $n$  do
    for  $i \leftarrow n$  to  $j$  step  $-1$  do
         $d_i \leftarrow (d_i - d_{i-1}) / (x_i - x_{i-j})$ 
    end for
end for
```

---

## 1.3 Objaśnienia:

Dla poprawności obliczeń powinniśmy zastosować odpowiedni porządek węzłów. Tak aby algorytm był numerycznie poprawny, jednak biorąc pod uwagę wymagania kolejności sformułowane w poleceniu, porządkowanie węzłów zostało pominięte. Tablicę wynikową tworzymy na podstawie poprzednich wartości w niej zawartych. Dodatkowo mamy dwie zagnieżdżone pętle, a złożoność algorytmu wynosi  $O(n^2)$ .

## 2 Zadanie2

### 2.1 Opis Problemu:

Napisać funkcję obliczającą wartość wielomianu interpolacyjnego stopnia  $n$  w postaci Newtona  $N_n(x)$  w punkcie  $x = t$  za pomocą uogólnionego algorytmu Hornera, w czasie  $O(n)$ .

**Dane:**  $x$  - wektor długości  $n + 1$  zawierający węzły  $x_0, \dots, x_n$   
 $fx$  - wektor długości  $n + 1$  zawierający ilorazy różnicowe.  
 $t$  - punkt, w którym należy obliczyć wartość wielomianu

**Wyniki:**  $nt$  - wartość wielomianu w punkcie  $t$

### 2.2 Rozwiązanie:

Implementacja programu w języku *Julia* korzystając z uogólnionego algorytmu Hornera.

---

**Algorithm 2** *warNewton*( $x, fx, t$ )

---

```
 $n \leftarrow \text{length}(x)$   
 $q_n \leftarrow fx_n$   
for  $i \leftarrow n - 1$  to 1 step  $-1$  do  
     $q_i \leftarrow fx_i + (t - x_i) * q_{i+1}$   
end for  
return  $q_1$ 
```

---

### 2.3 Objaśnienia:

Algorytm oblicza kolejne wartości, bazując na poprzednio wyliczonych. Pętla wykonuje jeden przebieg, więc złożoność algorytmu jest liniowa.

### 3 Zadanie3

#### 3.1 Opis problemu

Znając współczynniki wielomianu interpolacyjnego w postaci Newtona oraz węzły, napisać funkcję obliczającą współczynniki w postaci naturalnej.

**Dane:**  $x$  - wektor długości  $n + 1$  zawierający węzły  
 $fx$  - wektor długości  $n + 1$  zawierający ilorazy różnicowe

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

#### 3.2 Rozwiązanie

Implementacja algorytmu w języku *Julia*. Każdy z wielomianów pomocniczych w metodzie Hornera możemy zapisać w postaci naturalnej:

$$w_i(x) = \sum_{k=1}^n a_k^{(i)} x^{k-i}, i = n, n-1, \dots, 0$$

Gdzie dostajemy układ równań rekurencyjnych na współczynniki  $a_k^{(i)}$

$$a_n^{(i)} = a_n^{(i+1)} = \dots = a_n^{(n)} = b_n$$

$$a_i^{(i)} = b_i - a_{i+1}^{(i+1)} x_i$$

$$a_k^{(i)} = a_k^{(i+1)} - a_{k+1}^{(i+1)} x_i, \text{ dla } k = i+1, i+2, \dots, n-1$$

Szukane współczynniki wielomianu postaci naturalnej to  $a_k^{(0)}$  Propozycja algorytmu:

---

**Algorithm 3** *naturalna*( $x, fx$ )

---

```
 $n \leftarrow \text{length}(x)$   
 $a_n \leftarrow fx_n$   
for  $i \leftarrow n-1$  to  $1$  do  
   $xi \leftarrow x_i$   
   $a_i \leftarrow fx_i$   
  for  $k \leftarrow i$  to  $n-1$  do  
     $a_k \leftarrow a_k - xi * a_{k+1}$   
  end for  
end for  
return  $a$ 
```

---

**Złożoność:** Złożoność tego algorytmu to  $n(n+1)/2$  mnożeń i odejmowań. Zatem spełnia on warunki zadania ( $O(n^2)$ ).

## 4 Zadanie 4

### 4.1 Opis problemu

Napisać funkcję która zinterpoluje zadaną funkcję  $f(x)$  w przedziale  $[a, b]$  za pomocą wielomianu interpolacyjnego stopnia  $n$  w postaci Newtona. Następnie narysuje wielomian interpolacyjny i interpolowaną funkcję.

**Dane:**  $f$  - funkcja  $f(x)$  zadana jako anonimowa funkcja ,  
 $a, b$  - przedział interpolacji  
 $n$  - stopień wielomianu interpolacyjnego

**Wyniki:** - funkcja rysuje wielomian interpolacyjny i interpolowaną funkcję na przedziale  $[a, b]$

### 4.2 Rozwiązanie:

Przedział  $[a, b]$  dzielimy na  $n$  równych części. Używamy węzłów równoodległych tj.  $x_k = a + kh$ ,  $h = (b - a)/n$ ,  $k = 0, 1, \dots, n$ , które posłużą nam do interpolacji wielomianu. Dla wybranych węzłów obliczamy wartość  $f(x)$ . Następnie przy użyciu funkcji *ilorazyRoznicowe* obliczamy ilorazy różnicowe. W ten sposób otrzymaliśmy wielomian  $N_n(x)$  interpolujący w postaci Newtona. Aby narysować wykres potrzebujemy wartości  $f(x)$  na przedziale  $[a, b]$  dla naszego wielomianu  $N_n(x)$ , więc dzielimy przedział  $[a, b]$  z krokiem co 0.0001 i dla każdego punktu obliczamy wartość przy użyciu funkcji *warNewton*. Teraz mamy już wszystkie dane potrzebne do narysowania wykresu, więc używając biblioteki *Plots* oraz silnika *plotly* rysujemy wykres. Wykres koloru czerwonego odpowiada funkcji interpolacyjnej, a koloru zielonego rzeczywistej funkcji  $f(x)$ .

## 5 Zadanie 5

### 5.1 Opis Problemu

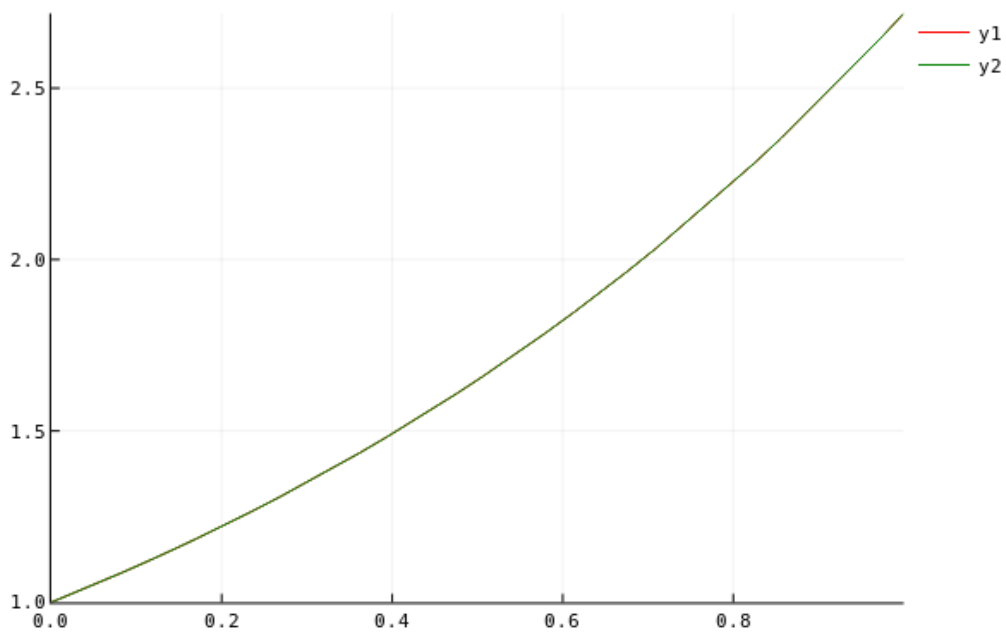
Przetestować funkcję  $rysujNnf(x, a, b, n)$  na następujących przykładach:

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

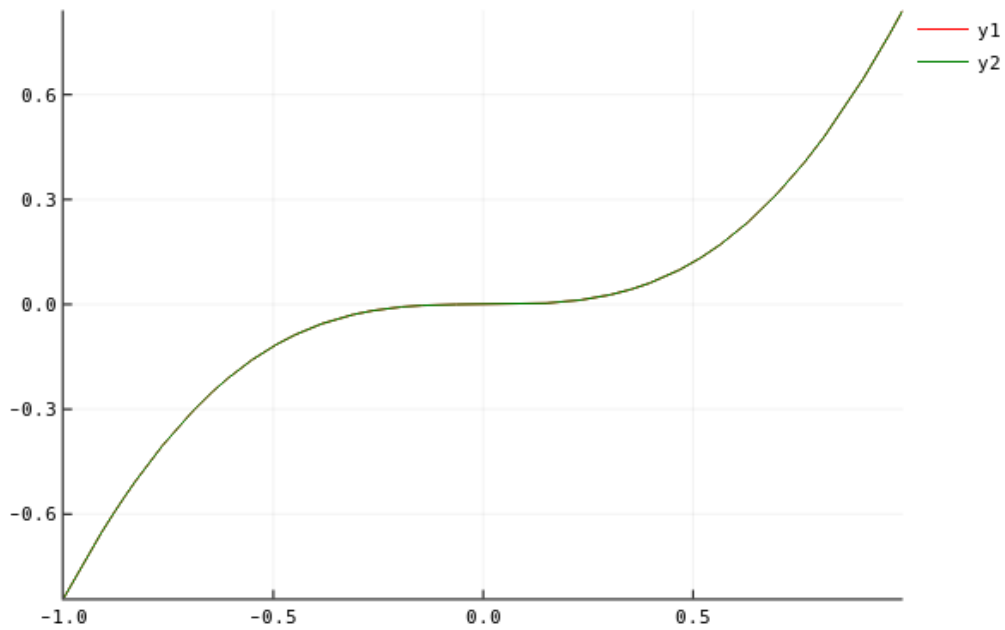
### 5.2 Rozwiązanie

Wywołanie funkcji  $rysujNnf(x)$  dla odpowiednich parametrów.

### 5.3 Wyniki:



Rysunek 1: Wykres funkcji  $e^x$  na przedziale  $[0, 1]$  dla  $n = 5, 10, 15$



Rysunek 2: Wykres funkcji  $x^2 \sin(x)$  na przedziale  $[-1, 1]$  dla  $n = 5, 10, 15$

## 5.4 Wnioski:

Dla zadanych funkcji oraz przedziałów, funkcja *rysujNnf(x)* interpoluje zadane funkcje z bardzo dużą dokładnością. Odchylenia na wykresie są niezauważalne. Ponadto wykres wygląda tak samo dla każdej ilości węzłów zadanej w poleceniu.

## 6 Zadanie 6

### 6.1 Opis problemu

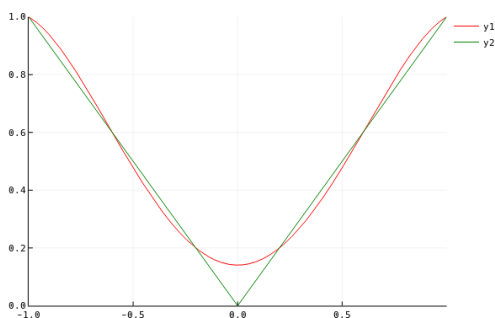
Przetestować funkcję *rysujNnf*( $f, a, b, n$ ) na następujących przykładach (zjawisko rozbieżności):

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

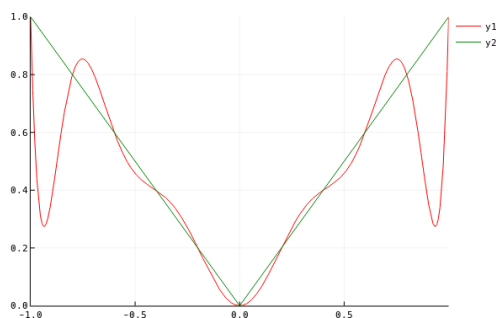
### 6.2 Rozwiązanie

Odpowiednie wywołanie metody *rysujNnf*( $\cdot$ ) z zadanymi parametrami.

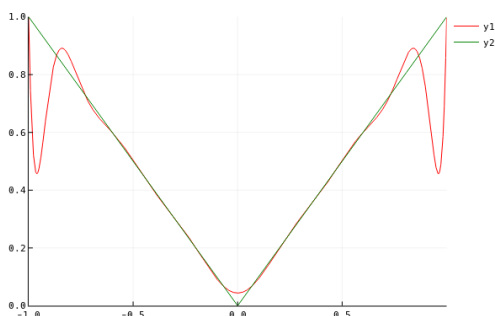
### 6.3 Wyniki



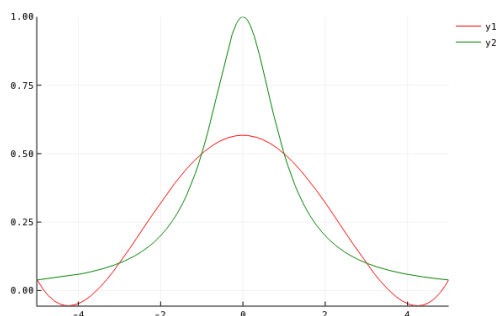
Rysunek 3: Wykres funkcji  $|x|$  na przedziale  $[-1, 1]$  dla  $n = 5$



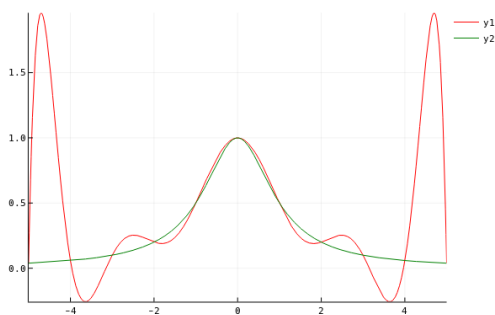
Rysunek 4: Wykres funkcji  $|x|$  na przedziale  $[-1, 1]$  dla  $n = 10$



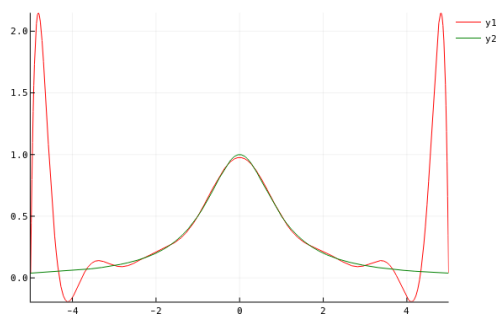
Rysunek 5: Wykres funkcji  $|x|$  na przedziale  $[-1, 1]$  dla  $n = 15$



Rysunek 6: Wykres funkcji  $\frac{1}{1+x^2}$  na przedziale  $[-5, 5]$  dla  $n = 5$



Rysunek 7: Wykres funkcji  $\frac{1}{1+x^2}$  na przedziale  $[-5, 5]$  dla  $n = 10$



Rysunek 8: Wykres funkcji  $\frac{1}{1+x^2}$  na przedziale  $[-5, 5]$  dla  $n = 15$



## 6.4 Wnioski

Możemy tutaj łatwo zauważyć, że początkowo wraz ze wzrostem liczby węzłów przybliżenie poprawia się w środkowej części przedziału, jednak tracąc swoją wartość na końcach. Jest to tak zwany efekt Rungego. Jest to typowe zachowanie dla interpolacji za pomocą wielomianów wysokich stopni, przy stałych odległościach węzłów. Dodatkowo funkcja  $|x|$  nie jest funkcją gładką co również ma wpływ na interpolację. Aby tego uniknąć można by zastosować węzły coraz gęściej upakowane na krańcach przedziału. Np. dla interpolacji  $n$ -punktowej węzłami powinny być miejsca zerowe wielomianu Czybyszewa  $n$ -tego stopnia.