

Sprawozdanie z czwartej listy zadań na laboratorium

Obliczenia Naukowe

Paweł Rubin

Grudzień 2019

1 Interpolacja wielomianowa

Interpolacja wielomianowa, nazywana też interpolacją Lagrange’a, od nazwiska pioniera badań nad interpolacją Josepha Lagrange’a lub po prostu interpolacją – metoda numeryczna przybliżania funkcji tzw. wielomianem Lagrange’a stopnia n przyjmującym w $n+1$ punktach, zwanych węzłami interpolacji, wartości takie same jak przybliżana funkcja.

Interpolacja jest często stosowana w naukach doświadczalnych, gdzie dysponuje się zazwyczaj skończoną liczbą danych do określenia zależności między wielkościami.

Twierdzenie (Karl Weierstraß). *Dowolną funkcję $y = f(x)$ ciągłą na przedziale domkniętym, można dowolnie przybliżyć za pomocą wielomianu odpowiednio wysokiego stopnia.*

2 Obliczanie ilorazów różnicowych

W celu wyznaczenia wektora ilorazów różnicowych posłużymy się następującą zależnością rekurencyjną:

$$\begin{cases} f[x_i] = f(x_i) (0 \leq i \leq N) \\ f[x_k, x_{k+1}, \dots, x_{k+m}] = \frac{f[x_{k+1}, x_{k+2}, \dots, x_{k+m}] - f[x_k, x_{k+1}, \dots, x_{k+m-1}]}{x_{k+m} - x_k} \end{cases} \quad (1)$$

Poniżej znajduje się pseudo-kod

```
1: function ILORAZY-RÓŻNICOWE( $x, f$ )
2:    $len \leftarrow \text{LENGTH}(f)$ 
3:   for  $i \leftarrow 1$  to  $len$  do
4:      $f_x[i] \leftarrow f[i]$ 
5:   end for
6:   for  $i \leftarrow 2$  down to  $len$  do
7:      $f_x[j] \leftarrow \frac{f_x[j] - f_x[j-1]}{x[j] - x[j-1]}$ 
8:   end for
9:   return  $f_x$ 
10: end function
```

3 Obliczanie wartości wielomianu interpolacyjnego stopnia n w postaci Newtona

Skorzystamy z uogólnionego algorytmu Hornera. Wartości wielomianu w punkcie t obliczymy ze wzoru:

$$w(x) = (x - t) \cdot q(z) \cdot w(t) \quad (2)$$

W kolejnych iteracjach zwiększamy odpowiednią wartość wektora o poprzednią, pomnożoną przez różnicę wartości węzła i wielomianu oraz t :

$$n_t = f_x[i] + (t - w[i]) \cdot n_t \quad (3)$$

gdzie f_x to wektor ilorazów różnicowych.

```

1: function WAR-NEWTON( $x, f_x, t$ )
2:    $len \leftarrow \text{LENGTH}(x)$ 
3:    $n_t = f_x$ 
4:   for  $i \leftarrow len - 1$  down to 1 do
5:      $n_t \leftarrow f_x[i] + (t - x[i]) \cdot n_t$ 
6:   end for
7:   return  $n_t$ 
8: end function

```

4 Obliczanie współczynników wielomianu Newtona w postaci naturalnej

Wykorzystamy uogólnione wzory Hornera. W każdej iteracji liczymy kolejne wartości częściowe wielomianu - doprowadzając jednocześnie każdy składowy wielomian do postaci naturalnej,

Złożoność algorytmu to $O(n^2)$, ponieważ schemat Hornera to $O(n)$, a mnożenie wielomianów to również $O(n)$.

```

1: function NATURALNA( $x, f_x$ )
2:    $len \leftarrow \text{LENGTH}(x)$ 
3:    $a[len] \leftarrow f_x[len]$ 
4:   for  $i \leftarrow len - 1$  down to 1 do
5:      $a[i] = f_x[i] - a[i + 1] \cdot x[i]$ 
6:     for  $j \leftarrow i + 1$  to  $len - 1$  do
7:        $a[j] \leftarrow a[j] - a[j + 1] \cdot x[i]$ 
8:     end for
9:   end for
10:  return  $a$ 
11: end function

```

5 Wizualizacja wielomianu Newtona

Najpierw obliczamy odległość pomiędzy kolejnymi węzłami, a następnie wartości f w kolejnych węzłach.

Następnie obliczamy kolejne ilorazy różnicowe i liczymy wartości wielomianu w n równo oddalonych od siebie punktach.

Wykresy rysowane są za pomocą pakietu **PyPlot**.

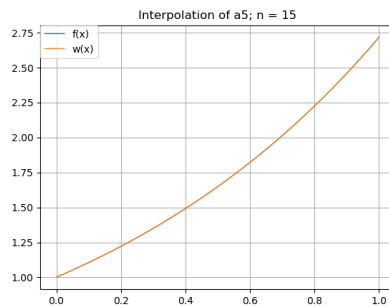
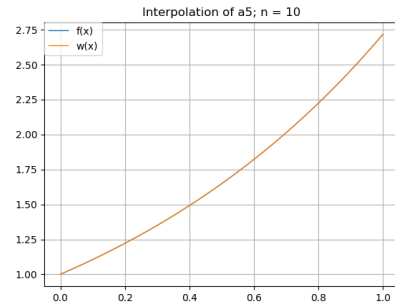
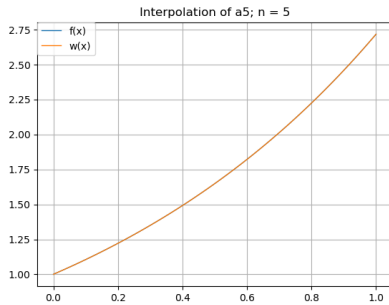
```

1: function RYSUJ-NNFX( $f, a, b, n$ )
2:    $nodes\_limit \leftarrow n + 1$ 
3:    $h \leftarrow \frac{b-a}{n}$ 
4:    $kh \leftarrow 0$ 
5:    $mult \leftarrow 20$ 
6:   for  $i \leftarrow 1$  to  $nodes\_limit$  do
7:      $x[i] \leftarrow a + kh$ 
8:      $y[i] \leftarrow f(x[i])$ 
9:      $kh \leftarrow kh + h$ 
10:  end for
11:   $kh \leftarrow 0$ 
12:   $f_x \leftarrow \text{ILORAZY-RÓŻNICOWE}(x, y)$ 
13:   $nodes\_limit \leftarrow nodes\_limit \cdot mult$ 
14:   $h \leftarrow \frac{b-a}{nodes\_limit-1}$ 
15:  for  $i \leftarrow 1$  to  $nodes\_limit$  do
16:     $args[i] \leftarrow a + kh$ 
17:     $w\_plot[i] \leftarrow \text{WAR-NEWTON}(x, f_x, args[i])$ 
18:     $f\_plot[i] \leftarrow f(args[i])$ 
19:     $kh \leftarrow kh + h$ 
20:  end for
21:   $\text{DRAW-PLOT}(q)$ 
22: end function

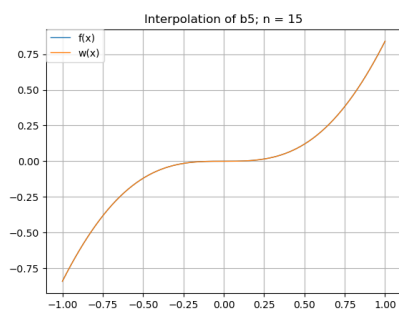
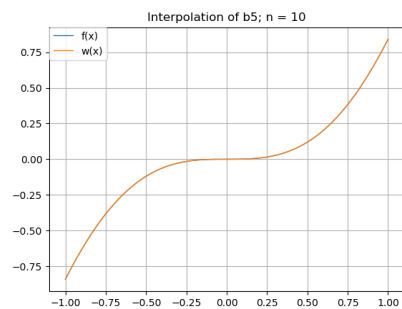
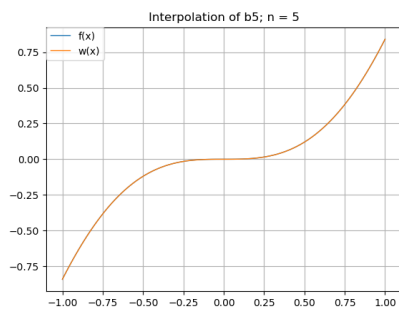
```

5.1 Na żywych przykładach

$f(x) = e^x$, przedział $[0, 1]$, $n = 5, 10, 15$

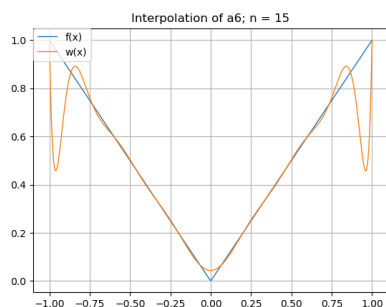
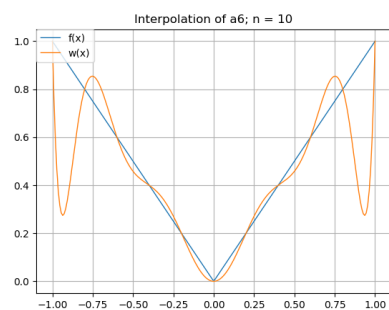
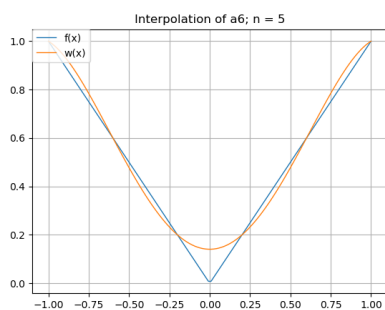


$f(x) = x^2 \sin(x)$, przedział $[-1, 1]$, $n = 5, 10, 15$

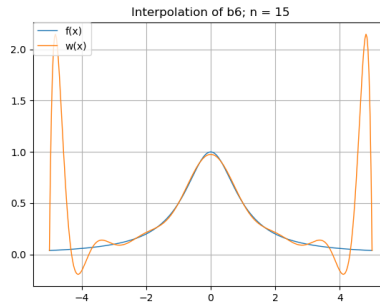
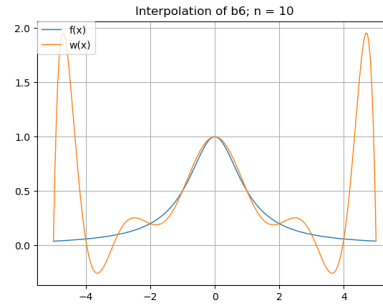
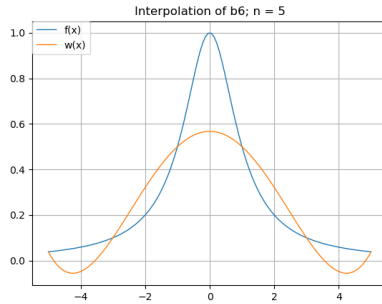


Wielomian interpolacyjny pokrywa się z funkcją dla powyższych przykładów - im większe n tym lepiej.

$f(x) = |x|$, przedział $[-1, 1]$, $n = 5, 10, 15$



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



Powyższe funkcje nie pokrywają się ze swoimi wielomianami interpolacyjnymi, ponieważ funkcja $f(x) = |x|$ nie jest różniczkowalna, natomiast w przypadku funkcji $f(x) = \frac{1}{1+x^2}$ obserwujemy zjawisko Runge'go - pogorszenie jakości interpolacji wielomianowej, mimo zwiększenia liczby jej węzłów. Początkowo ze wzrostem liczby węzłów n przybliżenie poprawia się, jednak po dalszym wzroście n , zaczyna się pogarszać, co jest szczególnie widoczne na końcach przedziałów.

Takie zachowanie się wielomianu interpolującego jest zjawiskiem typowym dla interpolacji za pomocą wielomianów wysokich stopni przy stałych odległościach węzłów. Występuje ono również, jeśli interpolowana funkcja jest nieciągła albo odbiega znacząco od funkcji gładkiej.

Ponieważ zgodnie z twierdzeniem Weierstrassa istnieje ciąg interpolujących wielomianów coraz wyższych stopni, które przybliżają jednostajnie funkcje ciągłą, można uważać to za paradoks, iż efekt Rungego ma dokładnie odwrotny wynik. Jest to spowodowane nałożeniem warunku na równoodległość węzłów.

Aby uniknąć tego efektu, stosuje się interpolację z węzłami coraz gęściej upakowanymi na krańcach przedziału interpolacji. Np. węzłami interpolacji n -punktowej wielomianowej powinny być miejsca zerowe wielomianu Czebyszewa n -tego stopnia.