Lista 4

Adrian Mucha 236526

9 grudnia 2018

1 Zadanie 1

1.1 Problem

Napisać funkcję obliczającą ilorazy różnicowe bez użycia tablicy dwuwymiarowej (macierzy). Na wejściu otrzymujemy
`x` - wektor długości $n+1$ zawierający węzły $x_0, ..., x_n$, gdzie $x[1] = x_0, ..., x[n+1] = x_n$
`f` - wektor długości $n+1$ zawierający wartości interpolowanej funkcji w węzłach $f(x_0), ..., f(x_n)$

1.2 Rozwiązanie

Aby obliczyć ilorazy różnicowe, posłużymy się następującym twierdzeniem.

Theorem 1 *Ilorazy różnicowe spełniają zależność*

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

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

Corollary 1 *Powyższy wzór uogólnia się na*

$$f[x_i, x_{i+1}, ..., x_{i+j}] = \frac{f[x_{i+1}, x_{i+2}, ..., x_{i+j}] - f[x_i, x_{i+1}, ..., x_{i+j-1}]}{x_{i+j} - x_i}$$

Znając węzły x_i i wartości funkcji $f(x_i)$, czyli ilorazy $f[x_i]$ zerowego rzędu, można za pomocą powyższych wzorów tworzyć tablicę (kwadratową, trójkątną) ilorazów różnicowych wyższych rzędów. Jest to jednak nieefektywny sposób, w tym sensie, że wystarczy użyć tablicy d zmiennych, która reprezentuje poszczególne kolumny we wspomnianej wyżej macierzy. Początkową wartością zmiennej d_i są odpowiednie $f[x_i]$ pociąga za sobą wzór:

$$d_i = \frac{d_i - d_{i-1}}{x_i - x_{i-j}}$$

Gdzie j oznacza numer kolumny, a i jest numerem wiersza. Można zauważyć pewne podobieństwa z Wniosku (Corollary) 1. Kolejne ilorazy budowane są na podstawie poprzednich (poprzednich kolumn), od dołu do góry. W ten sposób unikamy konieczności alokowania zbędnej pamięci i wszystkie obliczenia wykonywane są w miejscu. Ideę powyższego rozumowania przedstawia Algorytm 1.

Algorithm 1 Algorytm wyznaczania ilorazów różnicowych

```
function ilorazyRoznicowe(x, f)
    n = length(x)
    d = []
    for i ← 1 to n do
        d_i = f_i
    end for
    for j = 1 to n do
        for i ← n downto j + 1 do
            d_i = (d_i − d_{i−1}) / (x_i − x_{i−j})
        end for
    end for
    return d
end function
```

2 Zadanie 2

2.1 Problem

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 $O(n)$. Postać wielomianu interpolacyjnego Newtona:

$$N_n(x) = \sum_{i=0}^{k} c_i \prod_{j=0}^{i-1} (x - x_j)$$

gdzie, $c_i = f[x_0, x_1, ..., x_i]$ (c_i jest *ilorazem różnicowym* rzędu i).

2.2 Rozwiązanie

Do rozwiązania zadania użyto *uogólnionego algorytmu Hornera*, wykorzystującego następujące zależności rekurencyjne:

$w_n(x) = f[x_0, x_1, ..., x_n]$

$w_k(x) = f[x_0, x_1, ..., x_k] + (x - x_k)w_{k+1}(x), \qquad (k = n - 1, ..., 0)$

$N_n(x) = w_0(x)$

Algorithm 2 Uogólniony algorytm Hornera do obliczania wartości wielomianu interpolacyjnego Newtona

```
function warNewton(x, f_x, t)
    n = length(x)
    w_k = f_{x_n}
    for k = n − 1 downto 1 do
        w_k = f_{x_k} + (t − x_k) w_k
    end for
    return w_k
end function
```

Dzięki *Algorytmowi Hornera*, wykonujemy jedynie n mnożeń i tyle samo dodawań, otrzymując zadaną złożoność obliczeniową $O(n)$.

3 Zadanie 3

3.1 Problem

Znając współczynniki wielomianu interpolacyjnego w *postaci Newtona* $c_0 = f[x_0]$, $c_1 = f[x_0, x_1]$, $c_2 = f[x_0, x_1, x_2]$,..., $c_n = f[x_0, ..., x_n]$ (ilorazy różnicowe) oraz węzły

$x_0, x_2, ..., x_n$ napisać funkcję obliczającą, w czasie $O(n^2)$, współczynniki jego postaci naturalnej $a_0, ..., a_n$ tzn. $a_n x^n + a_{n-1} x^{n-1} + ... + a_1 x + a_0$

3.2 Rozwiązanie

Współczynnik a_n przy najwyższej potędze jest równy c_n. Wracając do *algorytmu Hornera*, stąd wynika że $w_n = a_n$. Idąc dalej, następne iteracje algorytmu będą polegały na tworzeniu wartości a_k wykorzystując do tego współczynniki już obliczone (te, które stoją przy wyższych potęgach, z góry na dół). Zależność pomiędzy następującymi po sobie a_k polega na tym, żeby zmodyfikować je, odejmując od aktualnej wartości współczynnika a_k wartość w_{k+1}. Należy pamiętać o nadaniu początkowej wartości c_k przy pierwszym napotkaniu wpsółczynnika a_k. W każdej kolejnej iteracji, modyfikujemy o jeden współczynnik więcej. Ponownie można myśleć o iteracjach tego algorytmu, jak o przechodzeniu po tabeli trójkątnej, w której kolumny są kolejnymi iteracjami (aktualnym stopniem wielomianu) zawierającymi $n - k$ wierszy (współczynników) a wierszami są wartości współczynników a_i w k-tej iteracji. W ostatniej (k-tej) iteracji wszystkie współczynniki a_i zawierają końcowe wartości. Tę zawiłą ideę, przedstawia stosunkowo prosty Algorytm 3.

Algorithm 3 Algorytm wyznaczający współczynniki a_n postaci normalnej z wielomianu interpolacyjnego Newtona

```
function NATURALNA(x, c)
    n = length(x)
    a = zeros(n)
    w_k = f_{x_n}
    for k = n − 1 downto 1 do
        a_k = a_k + c_k
        for i = k to n − 1 do
            a_i = a_i − a_{i+1} · x_k
        end for
    end for
    return a
end function
```

4 Zadanie 4

4.1 Problem

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. Wykorzystać funkcje `ilorazyRoznicowe` oraz `warNewton`.

4.2 Rozwiązanie

Zaimplementowano funkcję `rysujNnfx`, która wyznacza węzły interpolacji z zadaną gęstością (ilość węzłów) oraz im odpowiadające wartości funkcji w węźle. Dzięki tym danym zostają obliczone ilorazy różnicowe przy użyciu funkcji `ilorazyRoznicowe`. Utworzona również zostaje funkcja obliczająca wartość w punkcie wielomianu interpolacyjnego odwzorowującego funkcję oryginalną. Dla uzyskania odpowiedniej gładkości drukowanego wykresu wykorzystujemy funkcję `linspace(a,b,200)`, która tworzy zakres na którym obliczane będą wartości funkcji interpolowanej jak i oryginalnej. Rysowanie zostało wykonane przy użyciu pakietu `Plots`, które jest następnie zapisywane w pliku wynikowym z rozszerzeniem `.png`. Metodę przedstawia Algorytm 4.

Algorithm 4 Funkcja rysująca wykresy funkcji oryginalnej oraz jej wersji zinterpolowanej

function RYSUJNNFX(f, a, b, n)
  $h = \frac{|b-a|}{n}$
  for $k = 0$ to n **do**
    $x_i = a + kh$
    $y_i = f(x_i)$
  end for
  $fx = ilorazyRoznicowe(x, y)$
  $fy(t) = warNewton(x, fx, t)$
  $range = linspace(a, b, 200)$
  $plot(range, f, "oryginalna")$
  $plot(range, fy, "interpolowana")$
end function

5 Zadanie 5

5.1 Problem

Przetestować funkcję $rysujNnfx(f, a, b, n)$ na następujących przykładach.

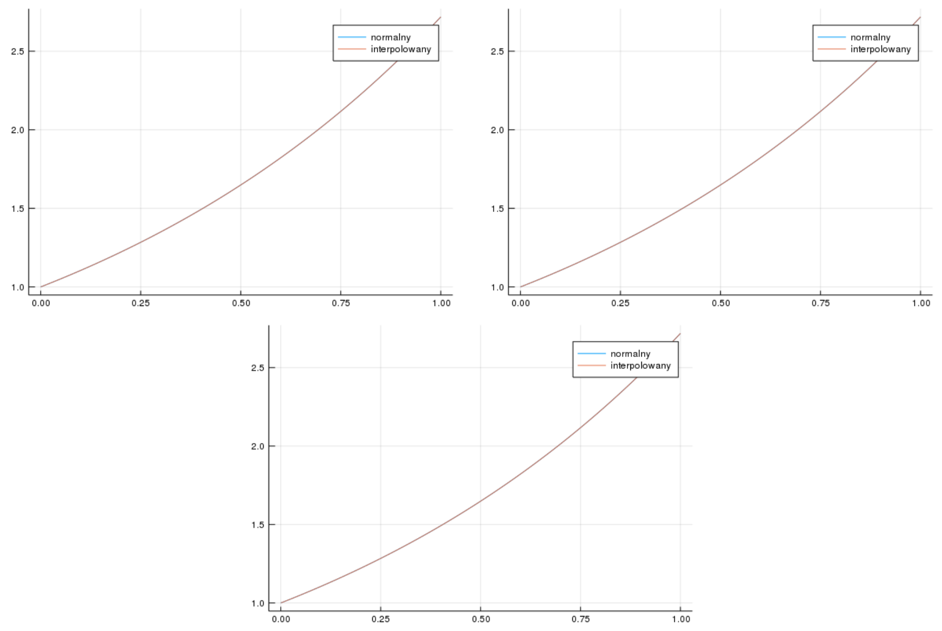
1. e^x, $[0, 1]$, $n = 5, 10, 15$
2. $x^2 sinx$, $[-1, 1]$, $n = 5, 10, 15$

5.2 Rozwiązanie

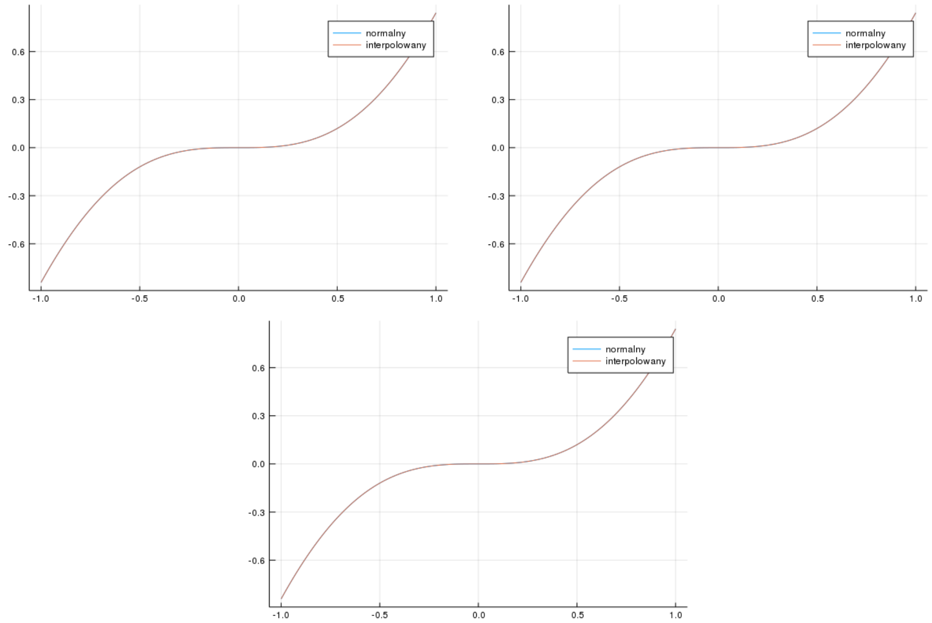
Wywołano funkcję $rysunNnfx$ na powyższych danych.

5.3 Wyniki

Uzyskano następujące wykresy widoczne na Rysunkach 1 oraz 2.



Rysunek 1: Kolejno wykresy funkcji e^x dla: $n = 5$, $n = 10$, $n = 15$



Rysunek 2: Kolejno wykresy funkcji $x^2 sinx$ dla: $n = 5$, $n = 10$, $n = 15$

5.4 Wnioski

Wykresy jednoznacznie pokazują brak rozbieżności w narysowanych funkcjach. Funkcja interpolowana świetnie pokrywa wartości funkcji oryginalnej.

6 Zadanie 6

6.1 Problem

Przetestować funkcję rysujNnfx(f,a,b,n) na następujących przykładach (zjawisko rozbieżności).

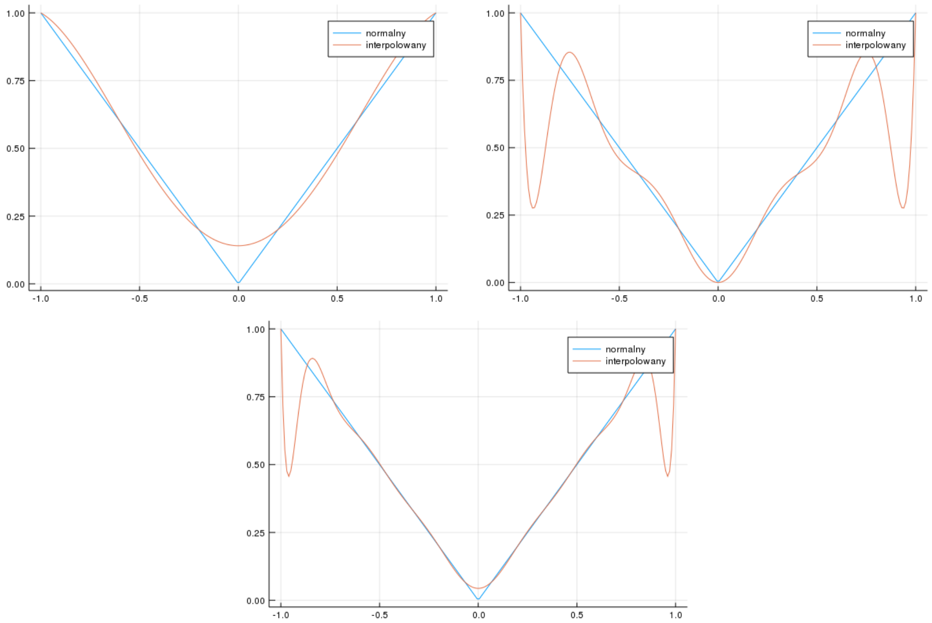
1. $|x|$, $[-1, 1]$, $n = 5, 10, 15$
2. $\frac{1}{1+x^2}$, $[-5, 5]$, $n = 5, 10, 15 (zjawiskoRunge'go)$

6.2 Rozwiązanie

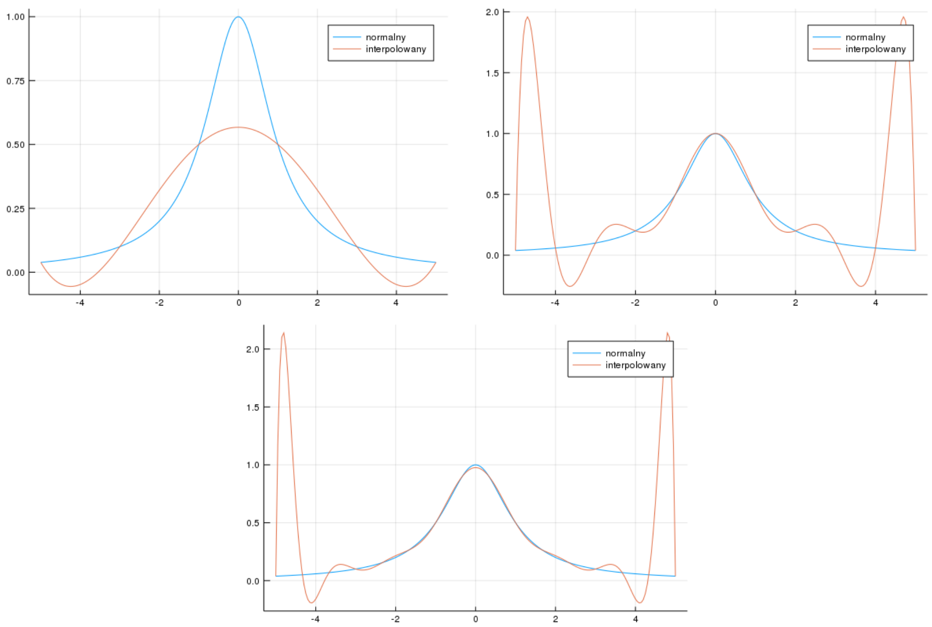
Wywołano funkcję $rysunNnfx$ na powyższych danych.

6.3 Wyniki

Uzyskano następujące wykresy widoczne na Rysunkach 3 oraz 4.



Rysunek 3: Kolejno wykresy funkcji $|x|$ dla: $n = 5$, $n = 10$, $n = 15$



Rysunek 4: Kolejno wykresy funkcji $\frac{1}{1+x^2}$ dla: $n = 5$, $n = 10$, $n = 15$

6.4 Wnioski

Wykresy funkcji interpolowanej znacznie rozbiegają w zależności od dobranego parametru n. Dla pierwszej funkcji $|x|$ jest to spowodowane brakiem różniczkowalności.

Dla drugiej funkcji, możemy zaobserwować, że w miarę jak zwiększamy ilość węzłów, jej przybliżenie początkowo poprawia się jednak po dalszym wzroście n, zaczyna się pogarszać, co jest najbardziej odczuwalne na końcach przedziałów. Takie zjawisko nazywane *efektem Rungego* jest typowe dla wielomianu interpolującego wysokich stopni przy stałych odległościach węzłów. Aby uniknąć tego fektu 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.