Obliczenia Naukowe Lista 4

Bartłomiej Puchała

December 2023

1 Zadanie 1

1.1 Opis problemu

Problem polega na napisaniu funkcji obliczajacej ilorazy różnicowe. Ilorazy różnicowe sa wykorzystywane do budowy wielomianu interpolacyjnego. Funkcje należy zaprogramować bez użycia tablicy dwuwymiarowej majac podane jako parametry:

1. x - wektor długości $n+1$ zawierajacy wezły x_0,...x_n gdzie $x[1] = x_0$,...,$x[n+1] = x_n$
2. f - wektor długości $n + 1$ zawierajacy wartości interpolowanej funkcji w wezłach $f(x_0)$,...,$f(x_n)$

1.2 Rozwiazanie

Rozwiazanie polega na obliczeniu ilorazów różnicowych za pomoca podanej reukrencji:

$$f[x_0] = f(x_0)$$

$$f[x_i, x_j] = \frac{f[x_j] - f[x_i]}{x_j - x_i}$$

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

Aby nie wykorzystywać tablicy dwuwymiarowej należy wypełnić tablice na poczatku wartościami wezłów x_i funkcji f, a potem przy kolejnej iteracji rzedu aktualizować obliczone ilorazy przeliczajac je kolejny raz od dołu. W ten sposób zostana uwzglednione wcześniej wyliczone ilorazy różnicowe, od których sa zależne.

Ta funkcja znajduje sie w module Functions.jl

```
function ilorazyRoznicowe(x::Vector{Float64}, f::Vector{Float64})
    n = length(f)
    # fx[1] = f[1] # c0 = f(x0)
    # Algorytm Newtona wzoru interpolacyjnego

    res = [value for value in f]
    println(res)
    println("-----")
    for i in 1:n
        for j in n:-1:i+1
            res[j] = (res[j] - res[j-1]) / (x[j] - x[j - i])
        end
    end
    return res
end
```

2 Zadanie 2

2.1 Opis problemu

Problem polega na napisaniu funkcji obliczajacej wartość wielomianu interpolacyjnego stopnia n w postaci Newtona $N_n(x)$ w punkcie $x = t$ za pomoca uogólnionego algorytmu Hornera w czasie $O(n)$. Parametrami funkcji sa:

x - wektor długości $n + 1$ zawierajacy wezły x_0, ..., x_n, gdzie $x[1] = x_0$, ..., $x[n + 1] = x_n$

fx - wektor długości $n + 1$ zawierajacy ilorazy różnicowe, gdzie $fx[1] = f[x_0]$, $fx[2] = f[x_0, x_1]$, ..., $fx[n] = f[x_0, ..., x_{n-1}]$, $fx[n + 1] = f[x_0, ..., x_n]$

t - punkt, w którym należy obliczyć wartość wielomianu

2.2 Rozwiazanie

Algorytm korzysta z uogólnionego algorytmu Hornera, który przedstawia sie nastepujaco:

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

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

$$N_n(x) := w_0(x)$$

$f[x_0, x_1, ..., x_n]$ jest ilorazem różnicowym stopnia n dla funkcji $f(x)$ wzgledem wezłów $x_0, x_1, ..., x_n$. Oznacza on tempo zmiany funkcji w danym punkcie interpolacyjnym.

```
Ta funkcja znajduje sie w module Functions.jl
function warNewton(x::Vector{Float64}, fx::Vector{Float64}, t::Float64)
    n = length(fx)
    nt = fx[n] # wn(x) = f[x0, ..., xn] warunek poczatkowy
    for k in n-1:-1:1
        nt = fx[k] + (t - x[k]) * nt
    end
    return nt
end
```

3 Zadanie 3

3.1 Opis problemu

Problem polega na napisaniu funkcji obliczajacej w czasie $O(n^2)$ współczynniki postaci naturalnej wielomianu $a_0, ..., a_n$ tzn. $a_n x^n + a_{n-1}x^{n-1} + ... + a_1 x + a_0$ znajac współczynniki wielomianu interpolacyjnego w postaci Newtona $c_0 = f[x_0]$, $c_1 = f[x_0, x_1]$ oraz wezły $x_0, x_1, ..., x_n$.

3.2 Rozwiazanie

Wielomian w postaci interpolacyjnej przedstawia sie nastepujaco:

$$N_n(x) = f[x_0]+f[x_0,x_1](x-x_0)+f[x_0,x_1,x_2](x-x_0)(x-x_1)+...+f[x_0,x_1,...,x_n](x-x_0)(x-x_1)...(x-x_{n-1})$$

Współczynnikiem a_n stojacym przy najwyższej potedze x^n jest c_n. Startujac od najwyższych poteg można przy kolejnych przejściach aktualizować współczynniki przy odejmowanych aktualnie potegach w ten sposób, że sa one w postaci naturalnej. Przy iteracjach wykorzystane zostana obliczone wcześniej współczynniki cześciowe postaci naturalnej do zaktualizowania ich o nowe potegi.

```
Ta funkcja znajduje sie w module Functions.jl
function naturalna(x::Vector{Float64}, fx::Vector{Float64})
    n = length(fx)
    a = [val of val in fx]

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

    return a
end
```

4 Zadanie 4

4.1 Opis problemu

Zadanie polega na napisaniu funkcji, która zinterpoluje zadana funkcje $f(x)$ w przedziale $[a, b]$ za pomoca wielomianu interpolacyjnego stopnia n w postaci Newtona. Należy narysować wielomian interpolacyjny i interpolowana funkcje.

4.2 Rozwiazanie

Ta funkcja znajduje sie w module Functions.jl

```
function rysujNnfx(f, a::Float64, b::Float64, n::Int)
    if a > b
        a, b = b, a
    end

    x = Vector{Float64}(undef, n + 1)
    y = Vector{Float64}(undef, n + 1)
    delta = zero(Float64)

    for k in 1:n+1
        x[k] = a + delta
        y[k] = f(x[k])
        delta += (b - a) / n
    end


    ilorazy = ilorazyRoznicowe(x, y)
    interpolationX = Vector{Float64}(undef, n + 1)
    interpolationY = Vector{Float64}(undef, n + 1)
    realVals = Vector{Float64}(undef, n + 1)

    delta = zero(Float64)
    for i in 1:n+1
        interpolationX[i] = a + delta
        interpolationY[i] = warNewton(x, ilorazy, interpolationX[i])
        realVals[i] = f(interpolationX[i])
        delta += (b - a) / n
    end

    clf()
    plot(interpolationX, interpolationY, label = "interpolated", linewidth = 1.5)
    plot(interpolationX, realVals, label = "f(x)", linewidth = 1.5, alpha = 0.5)
    legend(title = "Interpolacja")
    savefig(string("wykresy/plot", f, "-", n, ".png"))
end
```

5 Zadanie 5

5.1 Opis problemu

Problem polega na przetestowaniu funkcji rysujNnfx(f, a, b, n) na przykładach:

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

5.2 Rozwiazanie

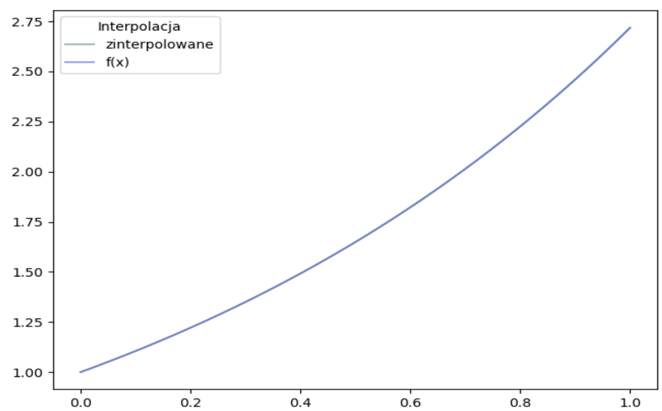

Figure 1: $\mathrm{e}^x, n = 5$

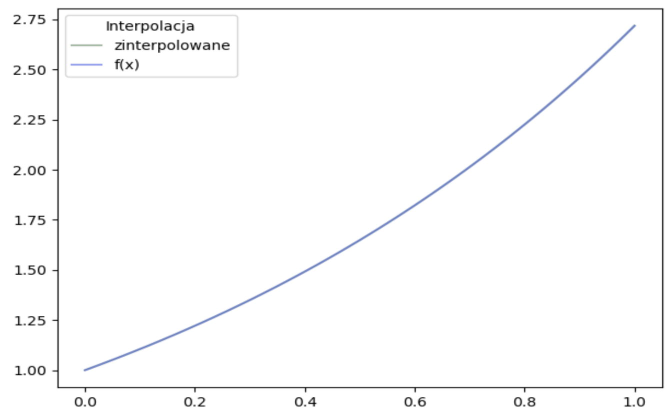

Figure 2: $\mathrm{e}^x, n = 10$

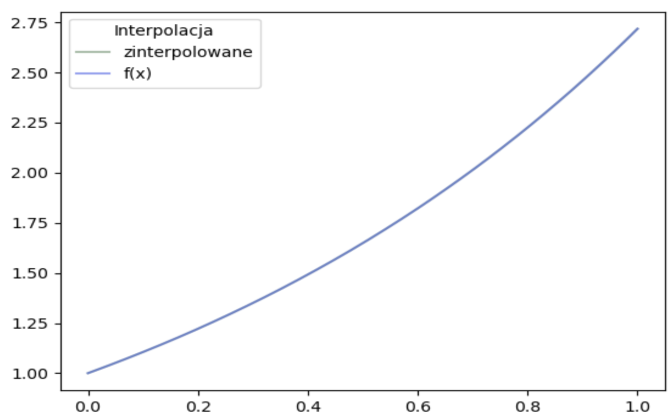

Figure 3: $\mathrm{e}^x, n = 15$

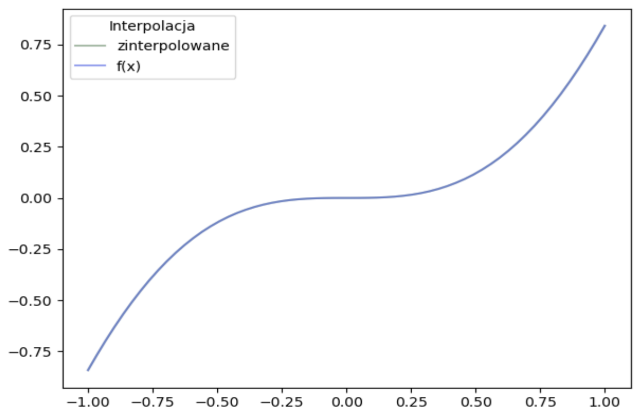

Figure 4: x$^2 sin(x), n = 5$

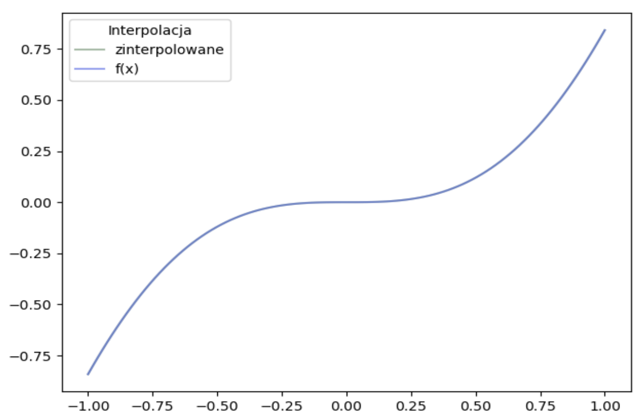

Figure 5: x$^2 sin(x), n = 10$

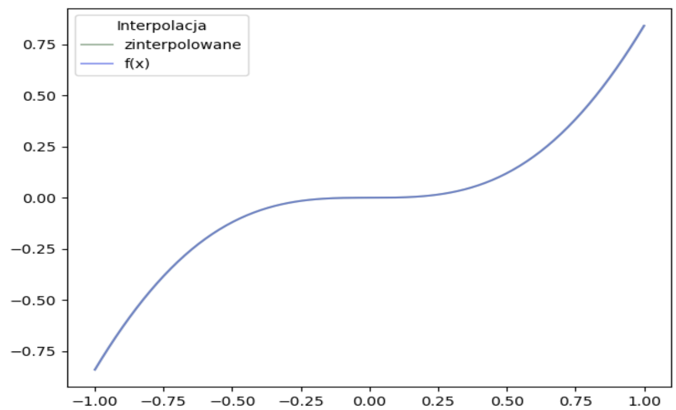

Figure 6: x$^2 sin(x), n = 15$

5.3 Wnioski

Dla wielomianu interpolacyjnego o małym stopniu wykresy pokrywaja sie dokładnie. Interpolacja funkcji których zmiana wartości jest niewielka, a pochodna nie zmienia znaku prowadzi do dokładnych wielomianów interpolacyjnych.

6 Zadanie 6

6.1 Opis problemu

Zadanie polega na pzetestowaniu funkcji z zadania 4 na kolejnych przykładach.

6.2 Rozwiazanie

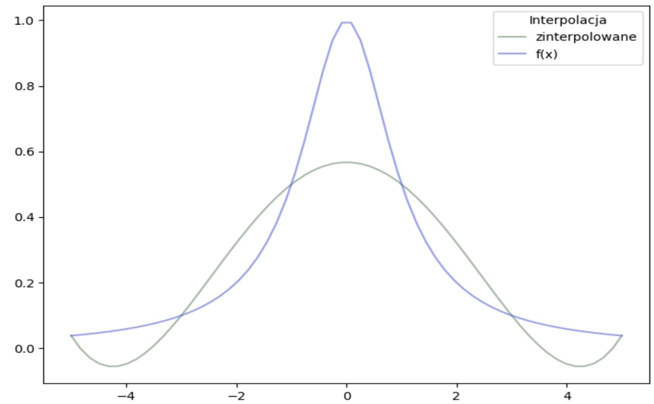

Figure 7: $1/(1+x^2)$, $n = 5$

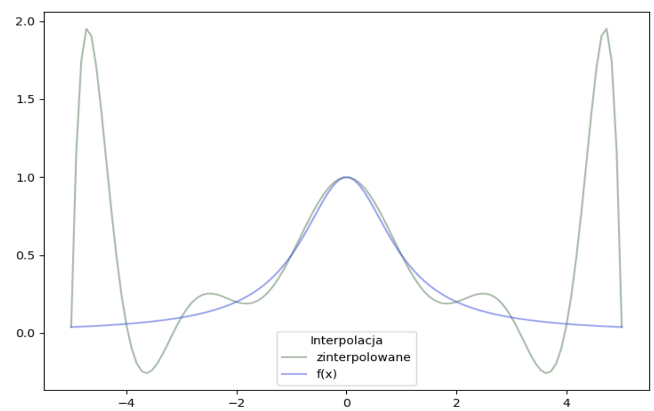

Figure 8: $1/(1+x^2)$, $n = 10$

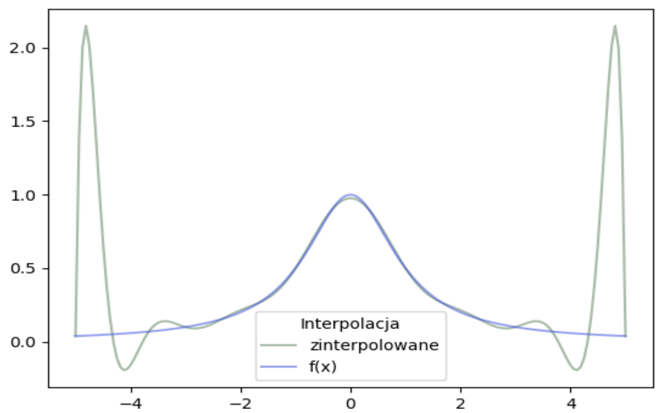

Figure 9: $1/(1+x^2)$, $n = 15$

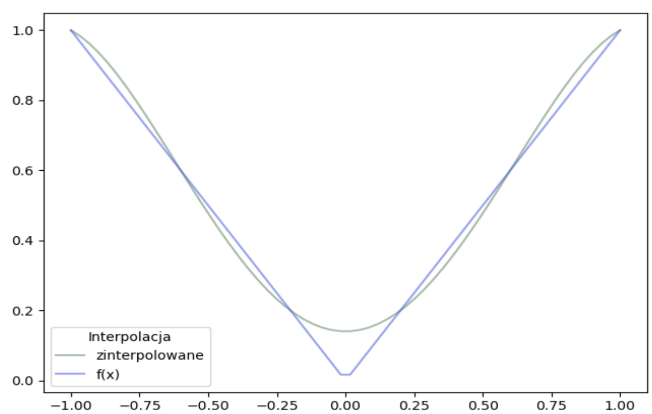

Figure 10: $abs(x)$, $n = 5$

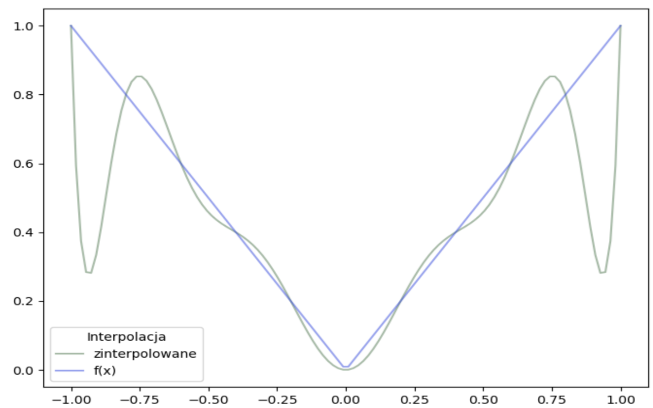

Figure 11: $abs(x)$, $n = 10$

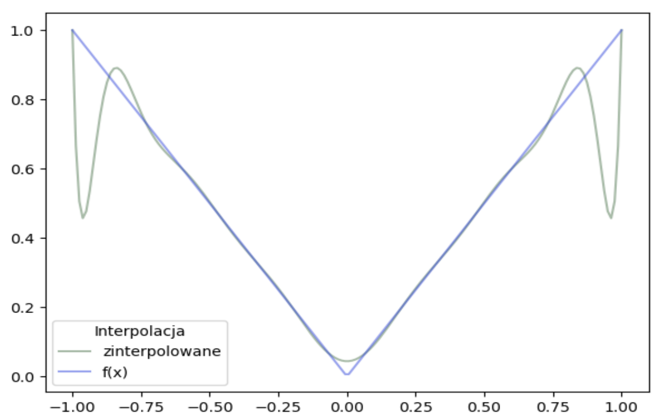

Figure 12: $abs(x)$, $n = 10$