

OBLICZENIA NAUKOWE
Lista nr 4 (laboratorium)

zad. 1 Napisać funkcję obliczającą ilorazy różnicowe.

```
function ilorazyRoznicowe (x::Vector{Float64}, f::Vector{Float64})
```

Dane:

- x – wektor długości $n + 1$ zawierający węzły x_0, \dots, x_n
 $x[1]=x_0, \dots, x[n+1]=x_n$
- 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
 $fx[1]=f[x_0],$
 $fx[2]=f[x_0, x_1], \dots, fx[n]=f[x_0, \dots, x_{n-1}], fx[n+1]=f[x_0, \dots, x_n]$.

Zaprogramować funkcję bez użycia tablicy dwuwymiarowej (macierzy).

zad. 2 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 $O(n)$, (implementacja algorytmu z zadania 8 lista nr 4 – ćwiczenia).

```
function warNewton (x::Vector{Float64}, fx::Vector{Float64}, t::Float64)
```

Dane:

- x – wektor długości $n + 1$ zawierający węzły x_0, \dots, x_n
 $x[1]=x_0, \dots, x[n+1]=x_n$
- fx – wektor długości $n + 1$ zawierający ilorazy różnicowe
 $fx[1]=f[x_0],$
 $fx[2]=f[x_0, x_1], \dots, fx[n]=f[x_0, \dots, x_{n-1}], fx[n+1]=f[x_0, \dots, x_n]$
- t – punkt, w którym należy obliczyć wartość wielomianu

Wyniki:

- nt – wartość wielomianu w punkcie t .

zad. 3 Znając współczynniki wielomianu interpolacyjnego w postaci Newtona $c_0 = f[x_0], c_1 = f[x_0, x_1], c_2 = f[x_0, x_1, x_2], \dots, c_n = f[x_0, \dots, x_n]$ (ilorazy różnicowe) oraz węzły x_0, x_1, \dots, x_n napisać funkcję obliczającą, w czasie $O(n^2)$, współczynniki jego postaci naturalnej a_0, \dots, a_n tzn. $a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$ (implementacja algorytmu z zadania 9 lista nr 4 – ćwiczenia).

```
function naturalna (x::Vector{Float64}, fx::Vector{Float64})
```

Dane:

- x – wektor długości $n + 1$ zawierający węzły x_0, \dots, x_n
 $x[1]=x_0, \dots, x[n+1]=x_n$
- fx – wektor długości $n + 1$ zawierający ilorazy różnicowe
 $fx[1]=f[x_0],$
 $fx[2]=f[x_0, x_1], \dots, fx[n]=f[x_0, \dots, x_{n-1}], fx[n+1]=f[x_0, \dots, x_n]$

Wyniki:

- a – wektor długości $n + 1$ zawierający obliczone współczynniki postaci naturalnej
 $a[1]=a_0,$
 $a[2]=a_1, \dots, a[n]=a_{n-1}, a[n+1]=a_n.$

zad. 4 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. Do rysowania zainstaluj np. pakiet Plots, PyPlot lub Gadfly.

W interpolacji użyć węzłów równoodległych, tj.

$x_k = a + kh$, $h = (b - a)/n$, $k = 0, 1, \dots, n$, albo węzłów będącymi zerami $n + 1$ wielomianu Czebyszewa T_{n+1} .

Nie wyznaczać wielomianu interpolacyjnego w jawnej postaci. Należy skorzystać z funkcji `ilorazyRoznicowe` i `warNewton`.

```
function rysujNnfx(f,a::Float64,b::Float64,n::Int; wezly::Symbol = :rownoodlegle)
```

Dane:

f – funkcja $f(x)$ zadana jako anonimowa funkcja,

a, b – przedział interpolacji

n – stopień wielomianu interpolacyjnego

`wezly` – jeśli `:rownoodlegle`, to węzły równoodległe, jeśli `:czebyszew`, to węzły $n + 1$ wielomianu Czebyszewa T_{n+1}

Wyniki:

- funkcja rysuje wielomian interpolacyjny i interpolowaną funkcję w przedziale $[a, b]$.

Uwagi: Powyższe funkcje powinny być zaprogramowane w języku Julia i umieszczone w module.

Napisać programy testujące!!!!!!

zad. 5 Przetestować funkcję `rysujNnfx(f,a,b,n; wezly=:rownoodlegle)` na następujących przykładach:

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

zad. 6 Przetestować funkcje `rysujNnfx(f,a,b,n; wezly=:rownoodlegle)` i `rysujNnfx(f,a,b,n; wezly=:czebyszew)` na następujących przykładach (zjawisko rozbieżności):

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

Uwagi: Do sprawozdania z eksperymentów opisanych w zad. 5 i 6 należy dołączyć wykresy rysowane przez funkcję `rysujNnfx`.

Rozwiązań zadań przedstawić w sprawozdaniu, plik pdf, które powinno zawierać:

1. krótki opis problemu,
2. rozwiązanie,
3. wyniki oraz ich interpretację,
4. wnioski.

Do sprawozdania należy dołączyć pliki z kodem (*.jl). Pliki powinny być skomentowane: imię i nazwisko autora (**anonimowe źródła nie będą sprawdzane**), opisane parametry formalne funkcji, komentarze zmiennych.

UWAGA: Ostateczną wersję programów proszę przetestować pod linuksem.