OBLICZENIA NAUKOWE

Lista 2

NIEBIESKI-KAPTUREK

INF-FIFA-PPT-TEAM®

LISTOPAD 2018

1. Zadanie 1 - "Iloczyn skalarny dwóch wektorów"

1.1. Opis problemu

Na poprzedniej liście mieliśmy napisać program w języku Julia realizujący następujący eksperyment obliczania iloczynu skalarnego dwóch wektorów:

x = [2.718281828, −3.141592654, 1.414213562, 0.5772156649, 0.3010299957]
y = [1486.2497, 878366.9879, −22.37492, 4773714.647, 0.000185049].

Zaimplementować mamy cztery algorytmy i policzyć sumę na cztery sposoby:

(a) "w przód" $\sum_{i=1}^{n} x_i\, y_i$, tj. algorytm

S := 0
for i := 1 to n **do**
$S := S + x_i * y_i$
end for

(b) "w tył" $\sum_{i=n}^{1} x_i\, y_i$, tj. algorytm

S := 0
for i := n **downto** *1* **do**
$S := S + x_i * y_i$
end for

(c) od największego do najmniejszego (dodać dodatnie liczby w porządku od największego do najmniejszego, dodaj ujemne liczby w porządku od najmniejszego do największego, a następnie daj do siebie obliczone sumy częściowe),

(d) od najmniejszego do największego (przeciwnie do metody (c)).

Teraz mamy powtórzyć to zadanie, przy założeniu, że usuwamy ostatnią 9 z x_4 i ostatnią 7 z x_5, oraz sprawdzić, jak zmieniły się wyniki.

1.2. Rozwiązanie

Cała implementacja rozwiązania pozostaje niezmieniona, w stosunku do zadania piątego z poprzedniej listy, oprócz fragmentu, gdzie tworzymy wektory. Tam usuwamy ostatnią cyfrę z x_4 i x_5, co wynika z treści polecenia zadania.

```
# Tworzenie wektorów
x = [2.718281828, -3.141592654, 1.414213562, 0.577215664, 0.301029995]
y = [1486.2497, 878366.9879, -22.37492, 4773714.647, 0.000185049]
```

Listing1: Jedyna zmiana w kodzie w stosunku do zad5.jl

1.3. Wyniki

```
julia> zad5()
Dokładny wynik: S = -1.00657107000000e-11
Algorytm A:
 pojedyncza precyzja: S = -0.4999443
 podwójna precyzja: S = 1.0251881368296672e-10
Algorytm B:
 pojedyncza precyzja: S = -0.4543457
 podwójna precyzja: S = -1.5643308870494366e-10
Algorytm C:
 pojedyncza precyzja: S = -0.5
 podwójna precyzja: S = 0.0
Algorytm D:
 pojedyncza precyzja: S = -0.5
 podwójna precyzja: S = 0.0
```

Listing2: Wyniki z zad5.jl (poprzednia lista)

```
julia> zad1()
Algorytm A:
 pojedyncza precyzja: S = -0.4999443
 podwójna precyzja: S = -0.004296342739891585
Algorytm B:
 pojedyncza precyzja: S = -0.4543457
 podwójna precyzja: S = -0.004296342998713953
Algorytm C:
 pojedyncza precyzja: S = -0.5
 podwójna precyzja: S = -0.004296342842280865
Algorytm D:
 pojedyncza precyzja: S = -0.5
 podwójna precyzja: S = -0.004296342842280865
```

Listing3: Wyniki z zad1.jl, pogrubione zostały wyniki różne od tych uzyskanych w listingu2

1.4. Wnioski

W przypadku arytmetyki Float32 usunięcie ostatnich cyfr w x_4 i x_5 nie miało wpływu na zmianę wyniku, w stosunku do rozwiązania z poprzedniej listy. Spowodowane jest to tym, że ta arytmetyka nie jest wystarczająco dokładna. Liczby pojedynczej precyzji *single*, pozwalają na zapis siedmiu cyfr znaczących w systemie dziesiętnym. Modyfikacja składowych wektora dotyczyła cyfr na dziesiątych pozycjach, więc automatycznie nic to nie zmieniło.

W przypadku arytmetyki Float64 ta kosmetyczna zmiana przyczyniła się do zmiany ostatecznego wyniku w każdym z czterech używanych algorytmów. Otrzymane dane są do siebie bardzo podobne i wynoszą: S ≈ -0.004296342..., a z ostatnich dwóch są dokładnie sobie równe. Wynika z tego, że dodanie cyfry na dziesiątej pozycji w dwóch składowych wektora x, powoduje mocne rozbieżności w wynikach zależnych od wybranego algorytmu.

2. Zadanie 2 - "Wykres funkcji logarytmicznej"

2.1. Opis problemu

W zadaniu drugim musimy narysować wykres funkcji $f(x) = e^x \ln(1 + e^{-x})$, a następnie policzyć granicę funkcji $\lim_{x\to\infty} f(x)$, porównać wykres funkcji z policzoną granicą oraz wyjaśnić zjawisko.

2.2. Rozwiązanie

Zacznijmy od policzenia granicy funkcji f(x):

$$\lim_{x\to\infty} e^x \ln(1 + e^{-x}) = \lim_{x\to\infty} \frac{\ln(1+e^{-x})}{e^{-x}} \overset{Reguła\ de\ l\prime Hospitala}{\Longleftrightarrow} \lim_{x\to\infty} \frac{(\ln(1+e^{-x}))\prime}{(e^{-x})\prime}$$
$$= \lim_{x\to\infty} \frac{-\frac{1}{e^x+1}}{-e^{-x}} = \lim_{x\to\infty} \frac{-e^x(-1)}{e^x+1} = \lim_{x\to\infty} \frac{e^x}{e^x+1} = 1$$

Zatem:

$$\lim_{x\to\infty} f(x) = 1$$

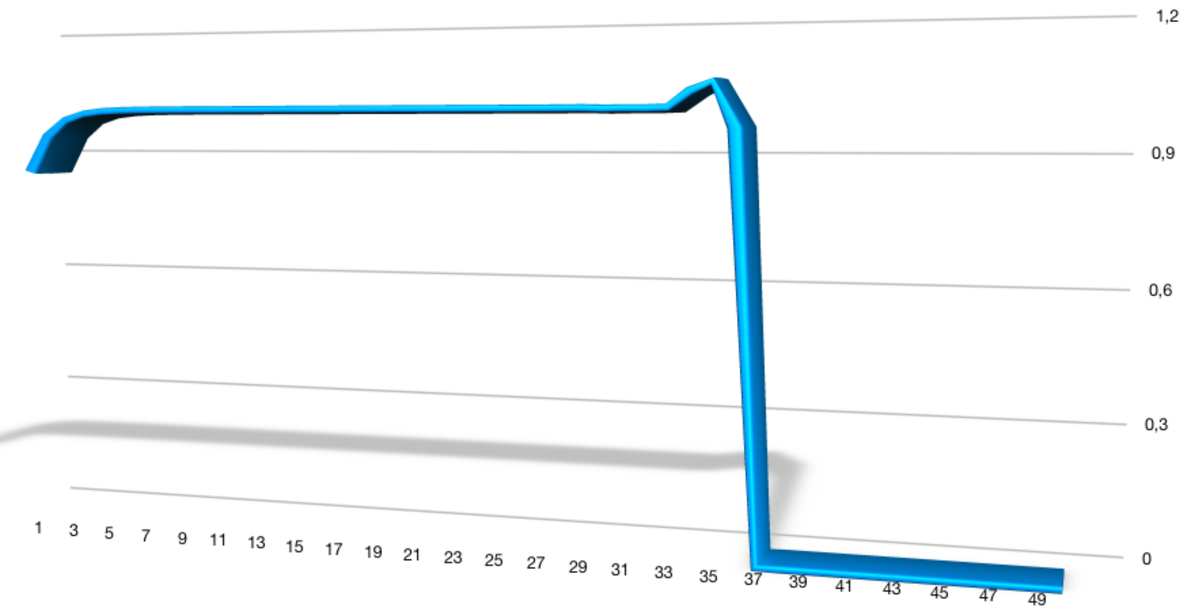
```
# Stworzenie logarytmu o podstawie e
function ln(x) return log(exp(1), x) end
# Stworzenie funkcji z zadania
function f(x) return (exp(1)^x) * ln(1 + exp(-x)) end
# Wyświetlenie wyników
function zad2() for i=0:50 println("$i, ", f(i)) end end
```

Listing4: Kod programu zad2.jl

2.3. Wyniki

Wyniki przedstawiam w postaci dwóch wykresów, wygenerowanych odpowiednio w programie *MS Excel* oraz *Numbers*.

Rysunek1: Wykres funkcji f(x) wykonany w Excelu.



Rysunek2: Wykres funkcji f(x) wykonany w Numbers.

2.4. Wnioski

Na podanych wykresach widać, że dla argumentu $x = 33$, wartość przekracza 1.0, co jest już odchyleniem. Następnie aż $f(35) \approx 1.0565$ funkcja rośnie, dla $f(36)$ przyjmuje ona wartość około 0.957, a każdy następny wynik wynosi 0.0. Spowodowane jest to prawdopodobnie podnoszeniem do coraz wyższych potęg liczby e.

3. Zadanie 3 - "Układ równań liniowych"

3.1. Opis problemu

W zadaniu trzecim mamy do rozwiązania układ równań liniowych $\boldsymbol{Ax} = \boldsymbol{b}$
za pomocą dwóch algorytmów: eliminacji Gaussa ($x = A/b$)
oraz inwersji ($x = A^{-1}b$ ($x = inv(A) * b$)).
Eksperymenty mamy przeprowadzić dla macierzy Hilberta H_n
z rosnącym stopniem $n > 1$ oraz dla macierzy losowej R_n, $n = 5, 10, 20$
z rosnącym wskaźnikiem uwarunkowania $c = 10^0, 10^1, 10^3, 10^7, 10^{12}, 10^{16}$.
Musimy policzyć błędy względne i porównać z dokładnym rozwiązaniem.

3.2. Rozwiązanie

Do wygenerowania macierzy Hilberta n-stopnia, użyto funkcji *hilb(n)*, co było podane w treści polecenia zadania, natomiast do wygenerowania losowej macierzy stopnia n i wskaźnika uwarunkowania użyto funkcji *matcond(n, c)*, co również było podane. Błędy względne zostały wyliczone przy pomocy normy wektora: $\partial = \frac{\|\tilde{x} - x\|}{\|x\|}$.

3.3. Wyniki

Stopień	Rząd	Wskaźnik uwarunkowania	Błąd względny	
			Eliminacja Gaussa	Odwrotność macierzy A
1	1	1.0	0.0	0.0
2	2			
3	3			
4	4			
5	5			
6	6			
7	7			
8	8			
9	9			
10	10			
11	11			
12	11			
13	11			
14	12			
15	12			
16	12			
17	12			
18	12			
19	13			
20	13			

Tabela1: Wyniki