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):

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

$$= \lim_{x\to\infty} \frac{\frac{1}{e^x+1}}{-e^{-x}} = \lim_{x\to\infty} \frac{-e^x(-1)}{e^x+1} = \lim_{x\to\infty} \frac{e^x}{e^x+1} = 1$$

Zatem:

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

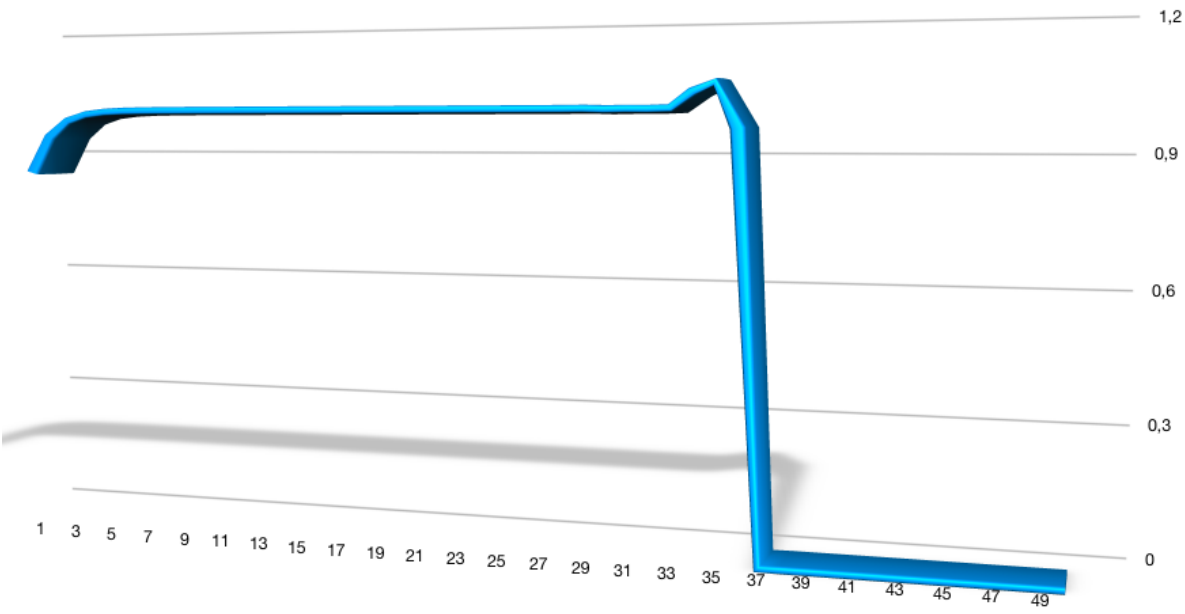
```
# Stworzenie logarytmu o podstawie e
function ln(x) return log(exp(1), x) end
# Stworzenie funkcji z zadania
function f(x) return (exp(1)^x) * ln(1 + exp(-x)) end
# Wyświetlenie wyników
function zad2() for i=0:50 println("$i, ", f(i)) end end
```

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	19.28147006790397	5.661048867003676e-16	1.4043333874306803e-15
3	3	524.0567775860644	8.022593772267726e-15	0.0
4	4	15513.73873892924	4.137409622430382e-14	0.0
5	5	476607.25024259434	1.6828426299227195e-12	3.3544360584359632e-12
6	6	1.4951058642254665e7	2.618913302311624e-10	2.0163759404347654e-10
7	7	4.75367356583129e8	1.2606867224171548e-8	4.713280397232037e-9
8	8	1.5257575538060041e10	6.124089555723088e-8	3.07748390309622e-7
9	9	4.931537564468762e11	3.8751634185032475e-6	4.541268303176643e-6
10	10	1.6024416992541715e13	8.67039023709691e-5	0.0002501493411824886
11	11	5.222677939280335e14	0.00015827808158590435	0.007618304284315809
12	11	1.7514731907091464e16	0.13396208372085344	0.258994120804705
13	11	3.344143497338461e18	0.11039701117868264	5.331275639426837
14	12	6.200786263161444e17	1.4554087127659643	8.71499275104814
15	12	3.674392953467974e17	4.696668350857427	7.344641453111494
16	12	7.865467778431645e17	54.15518954564602	29.84884207073541
17	12	1.263684342666052e18	13.707236683836307	10.516942378369349
18	12	2.2446309929189128e18	10.257619124632317	24.762070989128866
19	13	6.471953976541591e18	102.15983486270827	109.94550732878284
20	13	1.3553657908688225e18	108.31777346206205	114.34403152557572

Tabela1: Wyniki z Macierzy Hilberta, otrzymane z zad3.jl

Stopień	Rząd	Wskaźnik uwarunkowania	Błąd względny	
			Eliminacja Gaussa	Odwrotność macierzy A
5	5	1.0	1.719950113979703e-16	1.4895204919483638e-16
5	5	10.0	3.2934537262255424e-16	9.930136612989092e-17
5	5	1.0e3	1.5102624605787574e-14	1.6511521094205152e-14
5	5	1.0e7	1.9988766985489306e-10	1.5010355224632225e-10
5	5	1.0e12	2.221971369993138e-5	2.3109636277558834e-5
5	4	1.0e16	0.348665346160947	0.32264773747850767
10	10	1.0	3.3306690738754696e-16	2.742048596011341e-16
10	10	10.0	2.3551386880256624e-16	2.3551386880256624e-16
10	10	1.0e3	2.1485727154431577e-14	1.9002659797115724e-14
10	10	1.0e7	2.774674323484301e-10	2.1681865365464676e-10
10	10	1.0e12	1.9706842115917344e-5	1.9279942140900783e-5
10	9	1.0e16	0.09040595149920845	0.048412291827592706
20	20	1.0	3.040470972244059e-16	2.730787568572e-16
20	20	10.0	3.2934537262255424e-16	4.0792198665315547e-16
20	20	1.0e3	6.159887137044856e-15	1.7919426092099404e-14
20	20	1.0e7	4.245028176337727e-11	5.3549083199213835e-11
20	20	1.0e12	7.458600485844564e-6	4.40397258339646e-6
20	19	1.0e16	0.0079748940288458	0.02470529422006546

Tabela2: Wyniki z losowej macierzy, otrzymane z zad3.jl

3.4. Wnioski

Macierz Hilberta jest przykładem macierzy źle, a nawet bardzo źle uwarunkowanej. Wskaźnik jej uwarunkowania już dla niewielkich stopni jest wysoki, np. dla $n = 10, cond(n) \approx 1{,}6 \times 10^{13}$, a błąd względny jest już rzędu 10^{-5}. Jej elementy wylicza się ze wzoru $h_{i_j} = \frac{1}{i+j-1}$, co już mieliśmy okazję się przekonać w poprzedniej liście, nie jest w większości możliwa do dokładnego zapisu w reprezentacji binarnej, dlatego już dla niewielkich układów równań dokładny wynik nie jest możliwy do uzyskania.

Błąd macierzy losowej jest zależny od stopnia uwarunkowania, czym wskaźnik jest większy, tym błąd jest również większy. Zauważyłem również, że dla wskaźnika uwarunkowania wynoszącego 10^{16}, gdy stopień wynosi $n, rank(n) = n - 1$. W pozostałych przypadkach rząd jest równy stopniowi macierzy.