Sprawozdanie z listy 2

Jarosław Socha
268463

2 listopada 2023

1 Iloczyn skalarny

Zadanie, tak jak na poprzedniej liście, polega na obliczeniu iloczynu skalarnego różnymi metodami. Metody to:

1. w kolejności rosnącej (po indeksach)
2. w kolejności malejącej (po indeksach)
3. od najmniejszego do największego osobno dla ujemnych i dodatnich
4. odwrotnie do 3.

Zadaniem jest porównać wyniki zwracane przy drobnych zmianach (tutaj rzędu 10^{-9}). Oto wyniki:

Metoda	Oryginał	Po zmianach	Błąd względny
1	-0.12593332	-0.13022967	0.034116052
2	-0.23105995	-0.23105995	0
3	$5.510957 \cdot 10^6$	$5.510957 \cdot 10^6$	0
4	4008.25	4008.25	0

Tabela 1: Wyniki dla Float32

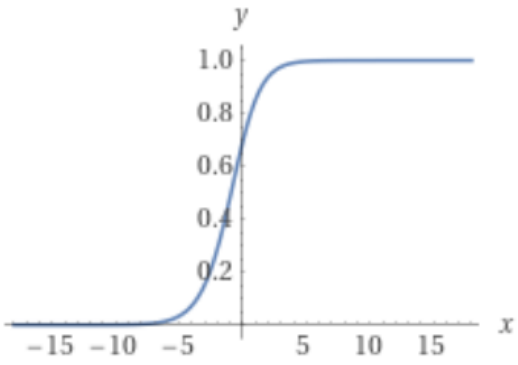
Metoda	Oryginał	Po zmianach	Błąd względny
1	$1.0251881368296672 \cdot 10^{-10}$	-0.004296342739891585	$4.19078478190021 \cdot 10^7$
2	$-1.5643308870494366 \cdot 10^{-10}$	-0.004296342998713953	$2.7464412279069766 \cdot 10^7$
3	$5.510957390881553 \cdot 10^6$	$5.510957373696182 \cdot 10^6$	$3.118400334134034 \cdot 10^{-9}$
4	4008.4028031835333	4008.398506840691	1.0718341078068914^{-6}

Tabela 2: Wyniki dla Float64

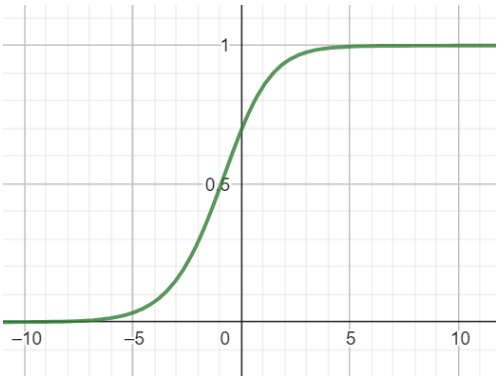
Dla Float32 zmiana ledwo wpływa na wyniki, ponieważ jest na granicy precyzji, a wyniki i tak nie są bliskie prawdziwym. Za to dla Float64 dla metod bliskich prawdzie, 1) i 2), różnice są ogromne, bo rzędów 10^7. Wielokrotne mnożenie i dodawanie błędu potęguje błąd i wynik znacząco odbiega od prawdziwej wartości $1.00657107000000 \cdot 10^{11}$.

2 Granica

Celem zadania jest sprawdzić poprawność obliczania granicy funkcji $e^x ln(1+e^{-x})$. Na początku sprawdzamy tę granicę za pomocą wykresów:



(a) Wykres narysowny przy pomocy WolframAlpha



(b) Wykres narysowny przy pomocy Geogebra

Widzimy, że $\lim_{x\to\infty} e^x ln(1+e^{-x}) = 1$. Teraz przeprowadzimy eksperyment z rosnącym x, dopóki nie osiągniemy NaN. Wyniki:

x	$e^x ln(1+e^{-x})$
1	0.8515335527332009
2	0.9378781941222128
4	0.9909524875473292
8	0.9998323061883384
16	0.9999999437324169
32	0.9999999999999936
64	1.0
128	0.9999999999999999
256	1.0
512	0.9999999999999999
640	1.0000000000000002
704	1.0
708	1.0
736	Inf
752	NaN

Tabela 3: Wynik działania w zależności od x

Widać, że choć z początku dążymy do 1 to po $x = 64$ skaczemy, osiągamy liczbę większą niż 1, a w pewnym momencie nawet nieskończoność, zanim jeszcze dotrzemy do NaN. Im wyższy jest x tym więcej dokładności tracimy, a w pewnym momencie x jest zbyt duży i przez ogromny czynnik e^x dostajemy nieskończoność.

3 Macierze

Zadanie polega na przetestowaniu dwóch algorytmów macierzowych rozwiązywania równań liniowych: Eliminacji Gaussa oraz odwrotności macierzy na złośliwych macierzach: Hilberta i o wysokim wskaźniku uwarunkowania. Należało policzyć błędy względne dla danej macierzy współczynników A oraz macierzy prawych stron b wygenerowanej przy założeniu, że wynikiem ma być macierz jedynek. Wyniki:

Rozmiar macierzy	eliminacja Gaussa	macierz odwrotna
2	$5.661048867003676 \cdot 10^{-16}$	$1.4043333874306803 \cdot 10^{-15}$
3	$8.022593772267726 \cdot 10^{-15}$	0.0
4	$4.4515459601812086 \cdot 10^{-13}$	$4.0996530221808757 \cdot 10^{-13}$
5	$1.6828426299227195 \cdot 10^{-12}$	$3.3544360584359632 \cdot 10^{-12}$
6	$3.641121827475282 \cdot 10^{-10}$	$3.4600984543575493 \cdot 10^{-10}$
7	$1.3205581245524007 \cdot 10^{-8}$	$1.1085268796510326 \cdot 10^{-8}$
8	$1.909875665491795 \cdot 10^{-7}$	$3.5917687006200656 \cdot 10^{-7}$
9	$1.300447094948228 \cdot 10^{-5}$	$1.5445339918749034 \cdot 10^{-5}$
10	0.00039932176841325925	0.0002775185876305333
11	0.00986617903362236	0.009474178855972404
12	0.7862760010041361	0.733781799038953
13	22.552104738091252	19.791274940267996
14	1.4379928825110613	8.692291052603988
15	2.4179669886885735	4.143383483677197
16	6.5528715985452655	4.756703115890609
17	6.805405581761646	8.201688865390627
18	31.423100504100923	34.447564628499634
19	1955.2812166935869	2426.5694199324626
20	13.326057064315599	17.790904042771082

Tabela 4: Błąd względny dla macierzy Hilberta o danym rozmiarze

Dla macierzy Hilberta błędy obydwu metod rosną szybko, ale w podobnym tempie, jednak metoda eliminacji Gaussa sprawdza się troszkę lepiej od metody macierzy odwrotnej.

wskaźnik uwarunkowania	$n = 5$	$n = 10$	$n = 20$
1	$4.965068306494546 \cdot 10^{-17}$	$2.742048596011341 \cdot 10^{-16}$	$4.902612130890297 \cdot 10^{-16}$
10	$1.4043333874306804 \cdot 10^{-16}$	$3.6821932062951477 \cdot 10^{-16}$	$5.715222817774256 \cdot 10^{-16}$
10^3	$7.0594821118363496 \cdot 10^{-15}$	$6.827375142205893 \cdot 10^{-15}$	$5.921767677594486 \cdot 10^{-15}$
10^7	$2.924754432861495 \cdot 10^{-10}$	$2.584225151520791 \cdot 10^{-10}$	$3.1333263766525503 \cdot 10^{-10}$
10^{12}	$2.7348349692560573 \cdot 10^{-5}$	$7.515953522018664 \cdot 10^{-6}$	$5.825775445322729 \cdot 10^{-6}$
10^{16}	0.05511634683180709	0.03616253324390385	0.16717769991984371

Tabela 5: Błąd względny macierzy o danym wskaźniku uwarunkowania metodą Gaussa

wskaźnik uwarunkowania	$n = 5$	$n = 10$	$n = 20$
1	$9.930136612989092 \cdot 10^{-17}$	$2.2752801345137457 \cdot 10^{-16}$	$5.606351459942525 \cdot 10^{-16}$
10	$2.0471501066083611 \cdot 10^{-16}$	$2.579925170969555 \cdot 10^{-16}$	$3.773125249565729 \cdot 10^{-16}$
10^3	$2.3381202007556434 \cdot 10^{-14}$	$3.191144845584111 \cdot 10^{-14}$	$1.7843325400473992 \cdot 10^{-14}$
10^7	$8.923065450948134 \cdot 10^{-11}$	$1.2105234399836872 \cdot 10^{-10}$	$5.231610852132886 \cdot 10^{-11}$
10^{12}	$3.90517696901542 \cdot 10^{-5}$	$6.3618011780387565 \cdot 10^{-6}$	$1.9370977687870217 \cdot 10^{-5}$
10^{16}	0.35185751510732294	0.1022214202112258	0.06722269245470144

Tabela 6: Błąd względny macierzy o danym wskaźniku uwarunkowania metodą odwrotnej macierzy

Błędy obydwu metod mają podobne rzędy wielkości i wzrastają wraz z większym wskaźnikiem uwarunkowania.

4 Wilkinson

Celem jest sprawdzenie poprawności obliczonych przez bibliotekę *Plynomials* w języku *Julia* pierwiastków wielomianu Wilkinsona $p(x) = (x-20)(x-19)...(x-2)(x-1)$. Niech $P(x)$ to wielomian w postaci wymnożonej, a z_k to k-ty pierwiastek wielomianu. Wyniki:

$$\sum_{k=1}^{20} |P(z_k)| = 5.159604088832 \cdot 10^{13}$$

$$\sum_{k=1}^{20} |p(z_k)| = 4.8209955749238805 \cdot 10^{13}$$

$$\sum_{k=1}^{20} |z_k - k| = 0.41037138509249393$$

Dodatkowo im większe było k tym większa wartość sumowanych czynników. Dla obu metod liczenia mamy podobny błąd (ten sam rząd wielkości).

Teraz zgodnie z poleceniem od czynnika przy x^{19} odejmiemy 2^{-23}:

$$\sum_{k=1}^{20} |P(z_k)| = 4.854192682582134 \cdot 10^{14}$$

$$\sum_{k=1}^{20} |p(z_k)| = 1.571499533141212 \cdot 10^{20}$$

$$\sum_{k=1}^{20} |z_k - k| = 34.83810615796398$$

Widzimy, że tu różnica jest jeszcze większa, ponieważ nie dość, że kumulowane są błędy, to jeszcze przez zmianę współczynnika kolejne liczby całkowite nie są już zerami wielomianu p

5 Populacja

Rozważamy tutaj model wzrostu populacji $p_{n+1} = p_n + rp_n(1-p_n)$ dla $p_0 = 0.001$ i $r = 3$ i patrzymy jaki błąd generuje niska precyzja i obcięcie do trzech miejsc po przecinku w 10 iteracji, przy 40 iteracjach. Dla Float32 otrzymujemy $p_{40} = 0.25860548$. Gdy obetniemy p_{10} do 3 miejsc po przecinku dostajemy $p_{40} = 1.093568$, co oznacza, że taka drobna zmiana silnie wpływa na wynik. Za to dla Float64 otrzymujemy $p_{40} = 0.011611238029748606$. Ten wynik także odbiega od tego dla Float32, a jako że zwiększyliśmy precyzję dwukrotnie, wydaje się to być prawdziwym wynikiem. Błąd jednak propagowany jest na tyle szybko, że potrzebowalibyśmy zdecydowanie większej precyzji (rzędu 2^{40} dla 40 iteracji), a zatem nawet przy precyzji Float64 robimy to co przy obcięciu do trzech cyfr dla Float32, tylko trochę mniej drastycznie, więc nie powinniśmy ufać temu wynikowi.

6 Zbieżność iteracji

Rozważamy tutaj równanie $x_{n+1} = x_n^2 + c$ dla różnych c i x_0. Wyniki:

c	x_0	x_{20}	x_{39}	x_{40}
-2	1.0	-1.0	-1.0	-1.0
-2	2.0	2.0	2.0	2.0
-2	1.99999999999999	1.9890237264361752	1.2926797271549244	-0.3289791230026702
-1	1.0	-1.0	0.0	-1.0
-1	-1.0	-1.0	0.0	-1.0
-1	0.75	-1.0	0.0	-1.0
-1	0.25	0.0	-1.0	0.0

Tabela 7: 20, 39 i 40 - iteracja dla danych x_0 i c

Za pomocą metody graficznej jesteśmy w stanie wywnioskować, że:

- dla $c = -1$
  - dla $x_0 = 1$ lub $x_0 = -1$ wartości oscylują między -1 i 0
  - dla wartości x_0 pomiędzy 0 a 1 otrzymujemy wartości coraz bliżej -1 i coraz bliżej 0 na przemian, aż zbiegniemy do powyższej oscylacji
- dla $c = -2$
  - dla $x_0 = 1$ mamy stale -1, a dla $x_0 = 2$ otrzymujemy stale 2
  - dla wartości bliskich 2 mamy rozbieżność, gdyż najpierw schodzimy coraz niżej, po czym lądujemy znowu na liczbie bliskiej 2 i powtarzamy proces

Co jest zgodne z wynikami eksperymentu.