Obliczenia naukowe - lista druga

Bartosz Rajczyk

10 listopada 2019

Spis treści

1 Zadanie 1.

1.1 Opis problemu

Zadanie polegało na powtórzeniu eksperymentu mnożenia wektorów z poprzedniej listy z delikatnie zmienionymi danymi.

1.2 Rozwiązanie

Rozwiązanie zaprezentowane w pliku `1.jl` polega na dosłownej implementacji algorytmów ze zmienionymi danymi.

1.3 Wyniki i interpretacja

Tabele 1. oraz 2. przedstawiają wyniki uzyskane po zmodyfikowaniu wartości z poprzedniej listy wraz z różnicą względem oryginalnych wyników.

algorytm	**wynik**	**różnica**
1	-0.004296342739891585	0.00429634284
2	-0.004296342998713953	0.00429634284
3	-0.004296342842280865	-0.004296342842280865
4	-0.004296342842280865	-0.004296342842280865

Tabela 1: Wyniki dla Float64

algorytm	**wynik**	**różnica**
1	-0.3472038161889941	3.6379788e-12
2	-0.3472038162872195	0
3	-0.5	0
4	-0.5	0

Tabela 2: Wyniki dla Float32

Jak widać, niewielkie zmiany w danych (rząd wielkości 10^{-10}) wpływają na wynik końcowy w rzędzie nawet 10^{-3}.

1.4 Wnioski

Daje nam to wyobrażenie o złym poziomie uwarunkowania zadania, ponieważ dane wejściowe wpływają na uzyskiwane błędy w znacznym stopniu i już niewielka ich zmiana powoduje znaczne odchylenia wyników w przyjętej arytmetyce.

2 Zadanie 2.

2.1 Opis problemu

Problem polega na narysowaniu wykresu funkcji

$$f(x) = e^x ln(1 + e^{-x})$$

w różnych pakietach do wizualizacji i porównanie ich z oczekiwaną granicą funkcji.

2.2 Rozwiązanie

Rozwiązanie polegało na wprowadzeniu równania do pakietów WolframAlpha oraz Matlab.

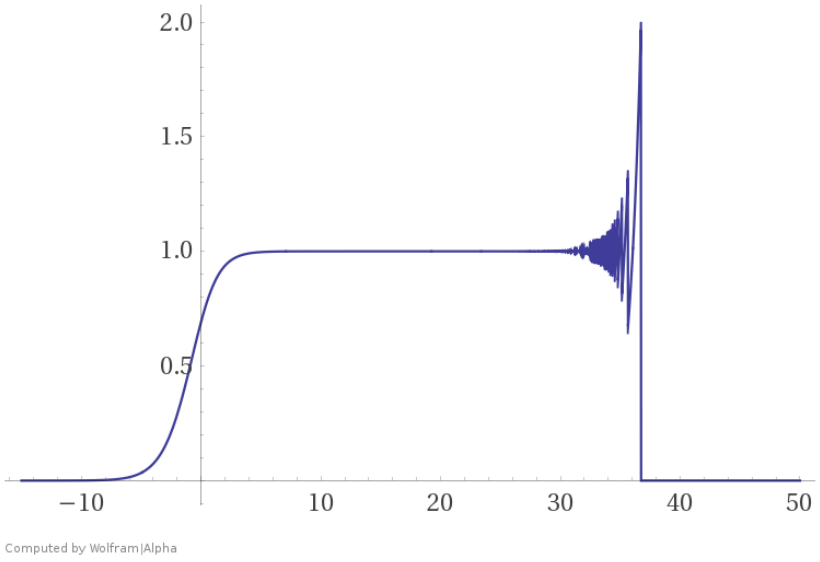
2.3 Wyniki i interpretacja

Obliczona granica to

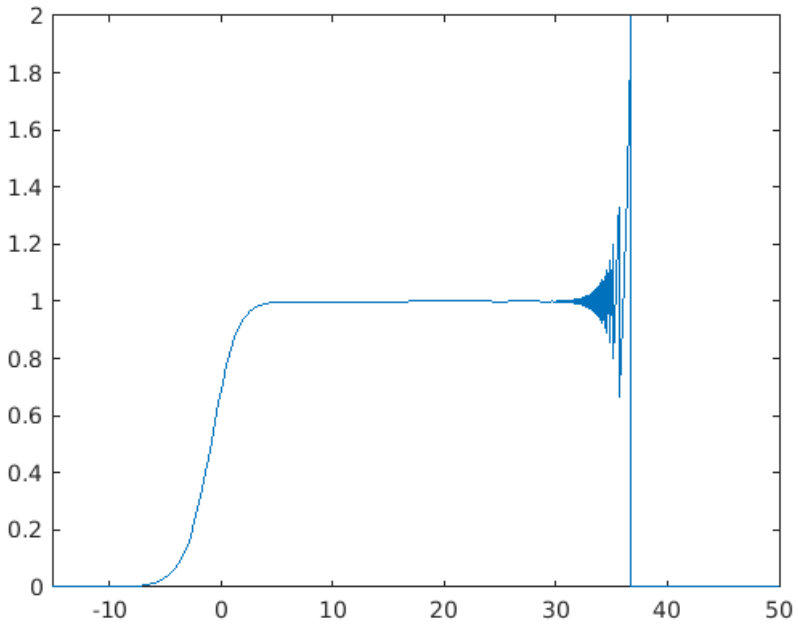
$$\lim_{x \to \infty} e^x ln(1 + e^{-x}) = 1$$

Wykresy uzyskane z pakietów do wizualizacji prezentują rysunki 1. oraz 2.

Rysunek 1: wykres z WolframAlpha



Rysunek 2: wykres z Matlab



Dosyć wyraźnie możemy spostrzec, że dla wartości powyżej 30 uzyskiwany wykres zaczyna odbiegać od oczekiwanej wartości, a następnie wartość funkcji na wizualizacji spada do zera, aby już tam pozostać, co nie zgadza się z wyliczoną granicą równą 1.

2.4 Wnioski

Obliczając wartość e^{-x} dla wartości w okolicach 30 - 40 możemy zauważyć, że przez ujemny wykładnik zbliża się ona do okolic epsilona maszynowego i ostatecznie spada do zera, co sprawia, że najpierw wykres zaczyna oscylować wokół jeden i coraz bardziej od niego odbiegać, a następnie spada do zera.

3 Zadanie 3.

3.1 Opis problemu

Problem polegał na porównaniu rozwiązań równania macierzowego metodą macierzy odwrotnej i eliminacji Gaussa w przypadku macierzy Hilberta oraz macierzy losowej o danym uwarunkowaniu.

3.2 Rozwiązanie

Rozwiązanie zaprezentowane w pliku `3.jl` korzysta z funkcji napisanych przez dr. Zielińskiego do generowania odpowiednich macierzy. Język Julia posiada wiele wbudowanych metod umożliwiających prostą pracę z macierzami, więc rozwiązanie jest dosłowną implenetacją odpowiednich algorytmów.

3.3 Wyniki i interpretacja

Tabele 3. oraz 4. prezentują otrzymane różnice względem poprawnego wyniku.

n	**rank**	**cond**	**odwrotna**	**gauss**
1	1	1.0	0.0	0.0
3	3	524.05	0.0	8.022593772267726e-15
5	5	4.76e5	3.3544360584359632e-12	1.6828426299227195e-12
7	7	4.75e8	4.713280397232037e-9	1.2606867224171548e-8
9	9	4.93e11	4.541268303176643e-6	3.8751634185032475e-6
11	10	5.22e14	0.007618304284315809	0.00015827808158590435
13	11	3.34e18	5.331275639426837	0.11039701117868264
15	12	3.67e17	7.344641453111494	4.696668350857427
17	12	1.26e18	10.516942378369349	13.707236683836307
19	13	6.47e18	12.233761393757726	9.720589712655698
21	13	3.29e18	43.4753048667801	56.40267595616145
23	13	6.31e17	13.803784630487236	12.483655076018373
25	13	1.37e18	16.93987792970947	10.15919484338797
27	14	4.42e18	28.105714901589447	30.802979621424512
29	14	8.06e18	267.9428700307017	17.744955787688642

Tabela 3: różnice względem poprawnego wyniku dla macierzy Hilberta

Jak widać, macierz Hilberta jest przykładem macierzy wyjątkowo źle uwarunkowanej, ponieważ wskaźnik uwarunkowania $cond(X)$ przekracza rząd 10^{11} dla macierzy 10 x 10, a następnie jedynie wzrasta wraz z jej rozmiarem.

n	rank	cond	odwrotna	gauss
5	5	1.0	2.482534153247273e-16	1.2161883888976234e-16
5	5	10.0	3.1401849173675503e-16	1.719950113979703e-16
5	5	1000.0	9.010196885116183e-15	2.577056993127896e-15
5	5	1.00e7	2.557993259869828e-10	2.3081002354773747e-10
5	5	9.99e11	3.901681985794751e-5	3.5064789180322344e-5
5	4	1.36e16	0.15313297173127022	0.14608058005390304
10	10	1.0	1.6837365821701475e-16	2.696722356863272e-16
10	10	10.0	4.800992561380036e-16	5.573275351449751e-16
10	10	1000.0	8.899530037265428e-15	1.1261624290316674e-14
10	10	1.005e7	3.407137587788345e-10	3.5771960585083103e-10
10	10	9.99e11	1.58033934680766e-5	1.5255865047639428e-5
10	9	8.25e15	0.22229274054723425	0.19669009252789274
20	20	1.0	4.1168171577819285e-16	4.644396126208125e-16
20	20	10.0	6.986470390014553e-16	6.600927136532929e-16
20	20	1000.0	2.7241172029510876e-14	2.4721702828668427e-14
20	20	9.99e6	3.464274750660281e-10	3.3267894585119536e-10
20	20	1.0e12	1.0559547832997443e-5	1.1012415441136603e-5
20	19	9.12e15	0.0736149369051341	0.08784784062469024

Tabela 4: różnice względem poprawnego wyniku dla macierzy losowej

3.4 Wnioski

Wyraźnie widać, że uwarunkowanie macierzy ma bezpośredni wpływ na otrzymywane błędy. Macierz Hilberta z dużym $cond(X)$ otrzymuje rosnące błędy przy dowolnej metodzie wraz ze wzrostem rozmiaru. Macierz losowa o rozmiarze 20 x 20, ale wskaźniku uwarunkowania 1.0 posiada błędy mniejsze niż macierz 5x5 o wskaźniku $1.36 * 10^{16}$.

4 Zadanie 4.

4.1 Opis problemu

Zadanie polega na znalezieniu zer wielomianu Wilkonsona używając pakietu Polynomials, sprawdzeniu wartości tego wielomianu dla odnalezionych zer i porównanie obliczonych zer z wartością tych z góry znanych, a następenie powtórzenie eksperymentu dla drugiego współczynnika zmienionego o 2^{-23}.

4.2 Rozwiązanie

Rozwiązanie zaprezentowane w pliku `4.jl` polega na użyciu konstruktorów `Poly` i `poly` do utworzenia obiektów wielomianu, a następnie wywołaniu funkcji `roots` w celu znalezieniu zer.

4.3 Wyniki i interpretacja

Tabele 5. i 6. prezentują kolejno wyniki dla wielomianu Wilkinsona i wielomianu Wilkinsona zmodyfikowanego o -2^{-23} w drugim współczynniku.

k	$\lvert P(z_k)\rvert$	$\lvert p(z_k)\rvert$	$\lvert z_k - k\rvert$
1	36352.0	38400.0	3.0109248427834245e-13
2	181760.0	198144.0	2.8318236644508943e-11
3	209408.0	301568.0	4.0790348876384996e-10
4	3.106816e6	2.844672e6	1.626246826091915e-8
5	2.4114688e7	2.3346688e7	6.657697912970661e-7
6	1.20152064e8	1.1882496e8	1.0754175226779239e-5
7	4.80398336e8	4.78290944e8	0.00010200279300764947
8	1.682691072e9	1.67849728e9	0.0006441703922384079
9	4.465326592e9	4.457859584e9	0.002915294362052734
10	1.2707126784e10	1.2696907264e10	0.009586957518274986
11	3.5759895552e10	3.5743469056e10	0.025022932909317674
12	7.216771584e10	7.2146650624e10	0.04671674615314281
13	2.15723629056e11	2.15696330752e11	0.07431403244734014
14	3.65383250944e11	3.653447936e11	0.08524440819787316
15	6.13987753472e11	6.13938415616e11	0.07549379969947623
16	1.555027751936e12	1.554961097216e12	0.05371328339202819
17	3.777623778304e12	3.777532946944e12	0.025427146237412046
18	7.199554861056e12	7.1994474752e12	0.009078647283519814
19	1.0278376162816e13	1.0278235656704e13	0.0019098182994383706
20	2.7462952745472e13	2.7462788907008e13	0.00019070876336257925

Tabela 5: wyniki dla wielomianu Wilkinsona

Już w tym momencie możemy zauważyć, że żadne z policzonych zer ewaluowane jako argument wielomianu nie dało poprawnego wyniku, czyli zera.

k	$\|P(z_k)\|$	$\|p(z_k)\|$	$\|z_k - k\|$
1	20992.0	22016.0	1.6431300764452317e-13
2	349184.0	365568.0	5.503730804434781e-11
3	2.221568e6	2.295296e6	3.3965799062229962e-9
4	1.046784e7	1.0729984e7	8.972436216225788e-8
5	3.9463936e7	4.3303936e7	1.4261120897529622e-6
6	1.29148416e8	2.06120448e8	2.0476673030955794e-5
7	3.88123136e8	1.757670912e9	0.00039792957757978087
8	1.072547328e9	1.8525486592e10	0.007772029099445632
9	3.065575424e9	1.37174317056e11	0.0841836320674414
10	7.143113638035824e9	1.4912633816754019e12	0.6519586830380406
11	7.143113638035824e9	1.4912633816754019e12	1.1109180272716561
12	3.357756113171857e10	3.2960214141301664e13	1.665281290598479
13	3.357756113171857e10	3.2960214141301664e13	2.045820276678428
14	1.0612064533081976e11	9.545941595183662e14	2.5188358711909045
15	1.0612064533081976e11	9.545941595183662e14	2.7128805312847097
16	3.315103475981763e11	2.7420894016764064e16	2.9060018735375106
17	3.315103475981763e11	2.7420894016764064e16	2.825483521349608
18	9.539424609817828e12	4.2525024879934694e17	2.454021446312976
19	9.539424609817828e12	4.2525024879934694e17	2.004329444309949
20	1.114453504512e13	1.3743733197249713e18	0.8469102151947894

Tabela 6: wyniki dla zmodyfikowanego wielomianu Wilkinsona

4.4 Wnioski

Zaburzenie wyników w nieznacznym stopniu (2^{-23}) znacząco wpłynęło na otrzymywane wyniki, stąd wiemy, że zadanie jest źle uwarunkowane. Same niedokładności w obliczanych zerach wielomianu wynikały z tego, że przy tak dużych współczynnikach brakowało cyfr znaczących do ich bezbłędnej reprezentacji w obranej arytmetyce.

5 Zadanie 5.

5.1 Opis problemu

Problem polega na przetestowaniu zachowania wzoru rekurencyjnego (modelu logistycznego) w trzech różnych przypadkach:

1. 40 iteracji w arytmetyce Float64
2. 4 x 10 iteracji w arytmetyce Float64 z ucinaniem mantysy do trzech cyfr znaczących po każdych dziesięciu
3. 40 iteracji w arytmetyce Float32

Wzór:

$$p_{n+1} = p_n + rp_n(1 - p_n)$$

dla zadanego r i p_0.

5.2 Rozwiązanie

Rozwiązanie zaprezentowane w pliku `5.jl` polega na dosłownej implementacji wzoru rekurencyjnego jako funkcji rekurencyjnej. Następnie uruchomione zostaje klasyczne 40 iteracji oraz cztery powtórzenia 10 iteracji z zapamiętywaniem ich wyniku i ponownym uruchomienim z nim jako p_0 kolejnego powtórzenia.

5.3 Wyniki i interpretacja

Uzyskane wyniki prezentują się następująco:

1. 0.011611238029748606
2. 0.71587336
3. 0.25860548

Jak widać, obcinanie cyfr znaczących jak i zmiana arytmetyki przy wielu iteracjach znacząco wpływa na wyniki końcowe.

5.4 Wnioski

Równania rekurencyjne polegające na wynikach z poprzednich wywołań są bardzo podatne na niewielkie zmiany dokładności oraz precyzję arytmetyki.

6 Zadanie 6.

6.1 Opis problemu

Problem polega na rozpatrzeniu zachowania równania rekurencyjnego postaci

$$x_n = x_{n-1}^2 + c$$

6.2 Rozwiązanie

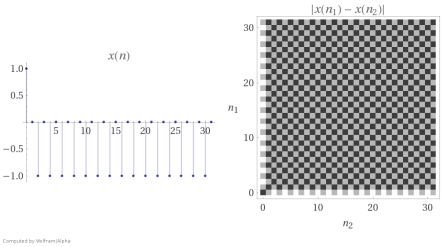
Rozwiązanie zaprezentowane w pliku `6.jl` polega na dosłownej implementacji wzoru rekurencyjnego jako funkcji rekurencyjnej.

6.3 Wyniki i interpretacja

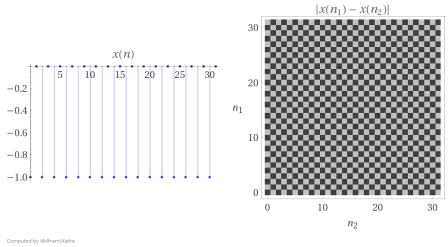
Wyniki dla różnych par c, x_0 po 40 iteracjach:

1. $c = -2, x_0 = 1$: -1
2. $c = -2, x_0 = 2$: 2
3. $c = -2, x_0 = 1.99999999999999$: -0.3289791230026702
4. $c = -1, x_0 = 1$: -1
5. $c = -1, x_0 = -1$: -1
6. $c = -1, x_0 = 0.75$: -1
7. $c = -1, x_0 = 0.25$: 0

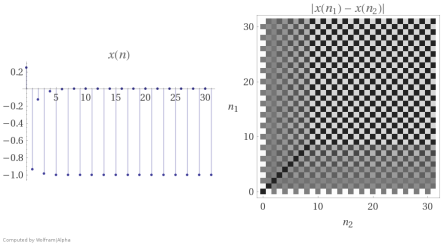
Dodatkowo utworzyłem wykresy iteracji tej funkcji korzystając z WolframAlpha:



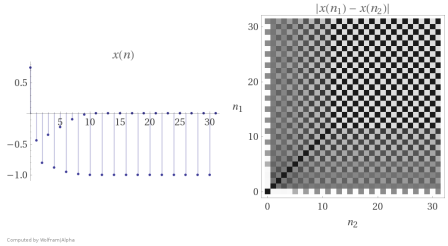
$c = -1, x_0 = 1$



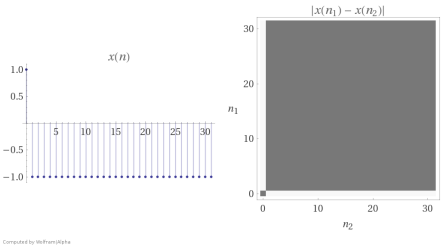
$c = -1, x_0 = -1$



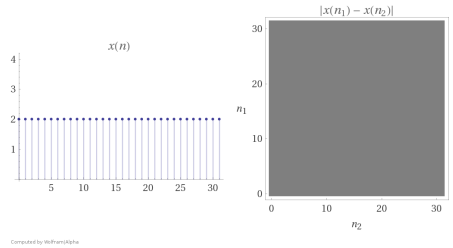
$c = -1, x_0 = 0.25$



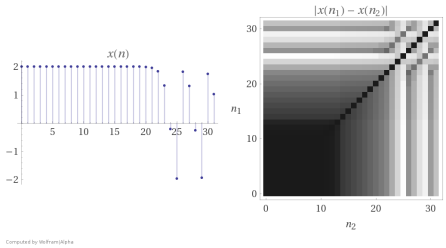
$c = -1, x_0 = 0.75$



$c = -2, x_0 = 1$



$c = -2, x_0 = 2$



$c = -2, x_0 = 1.99999999999999$

Otrzymane wyniki nie zaskakują w przypadku użycia za c oraz x_0 wartości całkowitych; możemy jednak dostrzec, że dla wartości $x_0 = 0.25$ i $x_0 = 0.75$ wartości powoli zbiegają do całkowitych, aby po 40 iteracjach zwrócić taką wartość. Dla odmiany $x_0 = 1.99999999999999$ na samym początku jest bliska liczbom całkowitym, a następnie wpada w mniej stabilne wartości.

6.4 Wnioski

Zadanie obrazuje dwie rzeczy:

1. skończona dokładność arytmetyki sprawia, że początkowe wartości 0.25 i 0.75 zbiegają ostatecznie do liczby całkowitej przez kumulację błędów
2. zmiana wartości z 2 na 1.99999999999999 znacząco wpływa na wyniki po 40 iteracjach

Jest to zdecydowany argument za tym, aby pamiętać o kumulacjach i narastaniu błędów przy powtarzanych operacjach.