Sprawozdanie 2

Bohdan Tkachenko 256630

November 2022

Zadanie 1

Powtórzenie zadanie 5 z poprzedniej listy dla lekko zmienionych danych

Dane z poprzedniej listy : x = [2.718281828, -3.141592654, 1.414213562, 0.5772156649, 0.3010299957] y = [1486.2497, 878366.9879, -22.37492, 4773714.647, 0.000185049]

Zmienione dane: x = Float64[2.718281828, -3.141592654, 1.414213562, 0.577215664, 0.301029995] y = Float64[1486.2497, 878366.9879, -22.37492, 4773714.647, 0.000185049]

Wyniki z poprzedniej listy :

Typ	alg1	alg2	alg3	alg4
Float32	-0.4999443	-0.4543457	-0.5	-0.5
Float64	1.0251881368296672e-10	-1.5643308870494366e-10	0	0

Table 1: Do zadania 5 z listy 1

Wyniki dla zmienionych danych: Wniosek: Jak widać, niewielka zmiana

Typ	alg1	alg2	alg3	alg4
Float32	-0.4999443	-0.4543457	-0.5	-0.5
Float64	-0.004296342739891585	-0.004296342998713953	-0.004296342842280865	-0.0042963428422808

Table 2: Ze zmienionymi danymi

danych powoduje duże różnice w wyniku,co oznacza, ze zadanie jest źle uwarunkowane. Dla alg3 i alg4 nie dostaliśmy 0,tylko jakiś wynik,bliski do wyników alg1 i alg2

Zadanie 1

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

Wyliczona granica funkcji:

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

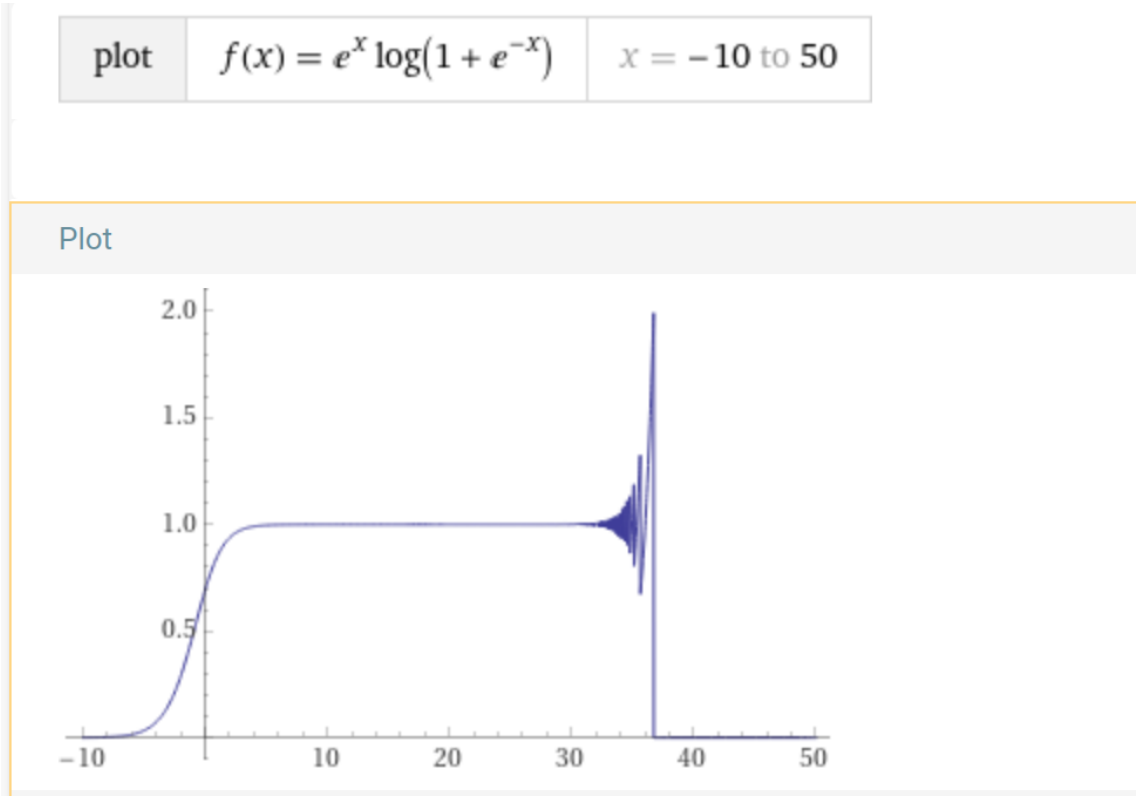

Figure 1: Wolfram alpha plot

Figure 2: Desmos plot

Na obu wykresach widać,że na poczatku wykres jest doby,ale od pewnego x zaczyna sie 'psuć',jest to spowodowane tym,że mnożymy duża liczbe e^x przez mała liczbe $ln(1+e^{-x})$ co daje zły wynik,co widać wy wykresie,jak to sie rozbiega od 1 . Później $ln(1+e^{-x})$ spada tak blisko do 0,że komputer zapisuje ta liczbe jako 0, odpowiednio w wyniku mnożenia też otrzymujemy 0.

Zadanie 1

Wyliczenie błedu wzglednego dla macierzy Hilberta oraz losowej macierzy stopnia n

Stosujemy 2 różne metody: 1)Eliminacja Gaussa 2)Metoda macierzy odwrotnej

Bład wzgledny liczymy za pomoca wzoru : $\frac{|x-x`|}{|x|}$

Otrzymane wyniki dla macierzy Hilberta:

N	rank	Eliminacja Gausa	Inverse	Wskaźnik uwarunkowania (Cond)
1	1	0	0	1
3	3	8.022593772267726e-15	0	524.0567775860644
5	5	1.6828426299227195e-12	3.3544360584359632e-12	476607.2502425855
7	7	1.2606867224171548e-8	4.713280397232037e-9	4.753673567446793e8
9	9	3.8751634185032475e-6	4.541268303176643e-6	4.9315375594102344e11
11	10	0.00015827808158590435	0.007618304284315809	5.222701316549833e14
13	11	0.11039701117868264	5.331275639426837	3.1883950689209334e18
15	12	4.696668350857427	7.344641453111494	3.67568286586649e17
17	12	13.707236683836307	10.516942378369349	1.249010044779401e18
19	13	102.15983486270827	109.94550732878284	6.472700911391398e18
...	...	...	...	...
39	15	118.20336501589891	263.5309838641091	9.058525451393528e18

Table 3: Do Hilberta

Jak widać, macierz Hilberta jest przykładem macierzy wyjatkowo źle uwarunkowanej, ponieważ wskaźnik uwarunkowania szybko rośnie,dla macierzy 9x9, juz to jest $10^{11}, itodalejrośnie.$

N	Wskaźnik uwarunkowania	rank	Eliminacja Gausa	Inverse
5	1	5 1.719950113979703e-16	2.432376777795247e-16	
5	10	5	5.15985034193911e-16	5.528866075183428e-
5	1000	5	2.7815906179775692e-14	2.37793081661856e-
5	1e7	5	2.6975780726065975e-10	2.375129349890999e-
5	1e12	5	5.8330001137372205e-5	6.080225377209794e
5	1e16	4	0.16060535607516435	0.134422684126647
10	1	10	2.4575834280036907e-16	2.895107444979072e-
...	...	...	...	...
10	1e16	9	0.05088574474439891	0.0346645299790993
20	1	20	4.723343357811471e-16	4.703730774727712e-
...	...	...	...	...
20	e16	19	0.26252113887598844	0.275599018005481

Table 4: Do losowej macierzy

Wyraźnie widać, że uwarunkowanie macierzy ma duży wpływ na otrzymywane błedy. Jednak dla tego samego Cond(M) rosnacy rozmiar M też obniża

dokładność otrzymanego wyniku.

Zadanie 4

Znalezienie zer wielomianu Wilkonsona

Zadanie polega na znalezieniu zer wielomianu Wilkonsona używajac pakietu Polynomials, sprawdzeniu wartości tego wielomianu dla odnalezionych zer i porównanie obliczonych zer

K	$\|P(z_k)\|$	$\|p(z_k)\|$	$\|z_k - k\|$
1	35696.50964788257	36720.50964788227	3.0109248427834245e-13
2	176252.60026668405	192636.60026691604	2.8318236644508943e-11
3	279157.6968824087	362101.69687113096	4.0790348876384996e-10
4	3.0271092988991085e6	2.7649652999648857e6	1.626246826091915e-8
5	2.2917473756567076e7	2.2277473671348542e7	6.657697912970661e-7
...	...	...	...
19	1.1435273749721195e13	1.14351402511197e13	0.0019098182994383706
20	2.7924106393680727e13	2.7923942556843e13	0.00019070876336257925

Table 5: Original Polynomial

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	20259.872313418207	36720.50964788227	1.6431300764452317e-13
2	346541.4137593836	192636.60026691604	5.503730804434781e-11
3	2.2580597001197007e6	362101.69687113096	3.3965799062229962e-9
4	1.0542631790395478e7	2.7649652999648857e6	8.972436216225788e-8
5	3.757830916585153e7	2.2277473671348542e7	1.4261120897529622e-6
...	...	...	...
19	4.557199223869993e12	1.14351402511197e13	2.0043294443099486
20	8.756386551865696e12	2.7923942556843e13	0.8469102151947894

Table 6: Polynomial with changed data

Wnioski: Zaburzenie wyników w nieznacznym stopniu (223) znaczaco wpłyneło na otrzymywane wyniki, stad wiemy, że zadanie jest źle uwarunkowane. Same niedokładności w obliczanych zerach wielomianu wynikały z tego, że przy tak dużych współczynnikach brakowało cyfr znaczacych do ich bezbłędnej reprezentacji w obranej arytmetyce.

Zadanie 5

Przetestowanie zachowania wzoru rekurencyjnego $p_{n+1} = p_n + rp_n(1 - p_n)$ (modelu logistycznego) w trzech różnych przypadkach

Uzyskane wyniki prezentuja sie nastepujaco:

1. 0.011611238029748606 dla 40 iteracji w arytmetyce Float64
2. 0.25860548 dla 40 iteracji w arytmetyce Float32
3. 0.71587336 z obcinaniem cyfr każde 10 iteracji

Jak widać, obcinanie cyfr znaczacych jak i zmiana arytmetyki przy wielu iteracjach znaczaco wpływa na wyniki końcowe.Równania rekurencyjne polegajace na wynikach z poprzednich wywołań sa bardzo podatne na niewielkie zmiany dokładności oraz precyzje arytmetyki,przy każdym wywołaniu pojawia sie co raz wiecej cyfr znaczacych,zapamietywanie których przestaje być możliwe po pewnej iteracji.

Zadanie 6

Rozpatrzenie zachowania równania rekurencyjnego postaci $x_n = x_{n-1}^2 + c$ dla pewnych danych

Dane:

1. c = 2 i x0 = 1 ; Wynik po 40 iteracjach : -1 (Stabilnie -1)
2. c = 2 i x0 = 2 Wynik po 40 iteracjach : 2 (Stabilnie 2)
3. c = 2 i x0 = 1.99999999999999; Wynik po 40 iteracjach : 2 -0.3289791230026702 (Chaos)
4. c = 1 i x0 = 1 ; Wynik po 40 iteracjach : -1 (Stabilnie na przemian 0,-1)
5. c = 1 i x0 = 1 ; Wynik po 40 iteracjach : -1 (Stabilnie na przemian 0,-1)
6. c = 1 i x0 = 0.75; Wynik po 40 iteracjach : -1 (Chaos do 17 iteracji,Stabilnie na przemian 0,-1)
7. c = 1 i x0 = 0.25 Wynik po 40 iteracjach : 0 (Chaos do 11 iteracji,Stabilnie na przemian 0,-1)

Wnioski: 1) Dla pewnych danych funkcja $x_n = x_{n-1}^2 + c$ zachowuje sie przewidywalnie i stabilnie, np dla (2,1),dla niektórych danych np licz niecałkowitych pojawia sie chaos niedeterministyczny

2)skończona dokładność arytmetyki sprawia, że poczatkowe wartości 0.25 i 0.75 zbiegaja ostatecznie do liczby całkowitej przez kumulacje błedów

3) drobna zmiana wartości np z 2 na 1.99999999999999 znaczaco wpływa na wyniki już po kilku iteracjach i funkcja zachowuje sie chaotycznie.

4) Z funkcjami rekurencyjnymi trzeba zwracać uwage na zgodność danych. Bład akumuluje sie bardzo szybko i nawet przy weilobitowej arytmetyce cyfry znaczace po pewnej liczbie iteracji zaczna być redukowane,co prowadzi do błednych wyników