

Obliczenia Naukowe

lista 1

Mateusz Tofil

6 listopada 2021

1 Zadanie 1

1.1 Opis zadania

Jest to powtórka zadania z listy poprzedniej - listy 1 zadania 5. Jedyna różnica jaka występuje, to usunięcie z x_4 ostatniej cyfry tj. 9. i z x_5 usunąć ostatnią 7.

1.2 Metoda rozwiązania

Metoda niczym się nie różni od wcześniejszego zdania z wcześniejszej liczby. Dane wejściowe są tylko inne - a dokładniej tylko x_4 i x_5 . Rozwiązanie znajduję się w pliku `zad1.jl` i jest dokładną duplikacją kodu z zadania 5 z poprzedniej listy ze zmienionymi danymi wejściowymi.

1.3 Otrzymane wyniki

Tabele poniżej przedstawiając sumy obliczone tak jak w zadaniu 5 z listy 1, natomiast lekko zmienionymi wartościami na wejściu. Wyniki przedstawione dla arytmetyki `Float64` i `Float32`.

podpunkt	sumy z listy 2	sumy z listy 1
<i>a</i>	-0.004296342739891585	1.0251881368296672e-10
<i>b</i>	-0.004296342998713953	-1.5643308870494366e-10
<i>c</i>	-0.004296342842280865	0.0
<i>d</i>	-0.004296342842280865	0.0

Tablica 1: Porównanie sum z poprzedniej listy i obecnej dla `Float64`

podpunkt	suma z listy 2	suma z listy 1
<i>a</i>	-0.3472038161889941	-0.3472038161853561
<i>b</i>	-0.3472038162872195	-0.3472038162872195
<i>c</i>	-0.5	-0.5
<i>d</i>	-0.5	-0.5

Tablica 2: Wyniki dla `Float32`

1.4 Wnioski

Zmiana wartości liczby, rzędu nawet 2^{-10} wpływa znacząco na wyniki ostateczne. Niewielkie zmiany spowodowały duże względne odkształcenia wyników, zatem możemy stwierdzić, że zadanie jest źle uwarunkowane.

2 Zadanie 2

2.1 Opis zadania

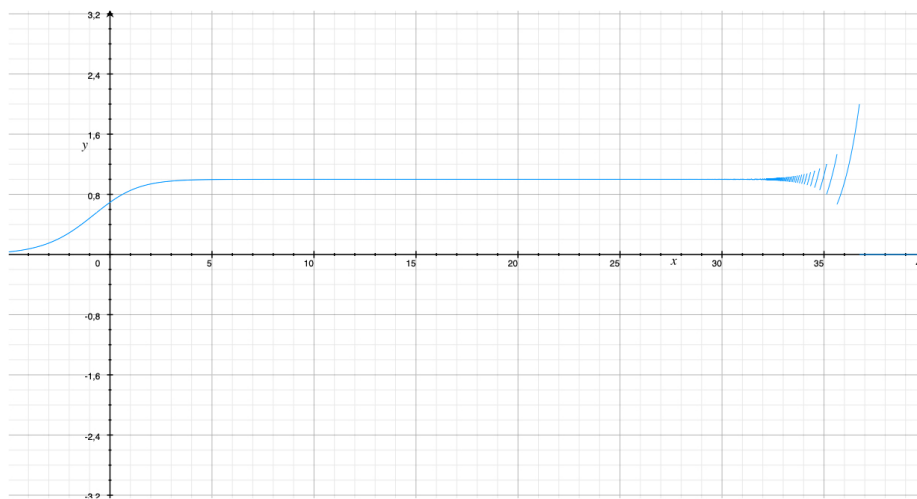
Należało narysować funkcję $f(x) = e^x \ln(1 + e^{-x})$ w conajmniej dwóch różnych program do rysowania wykresu, zbadać faktyczną granicę funkcji i porównać z otrzymanymi wykresami.

2.2 Metoda rozwiązania

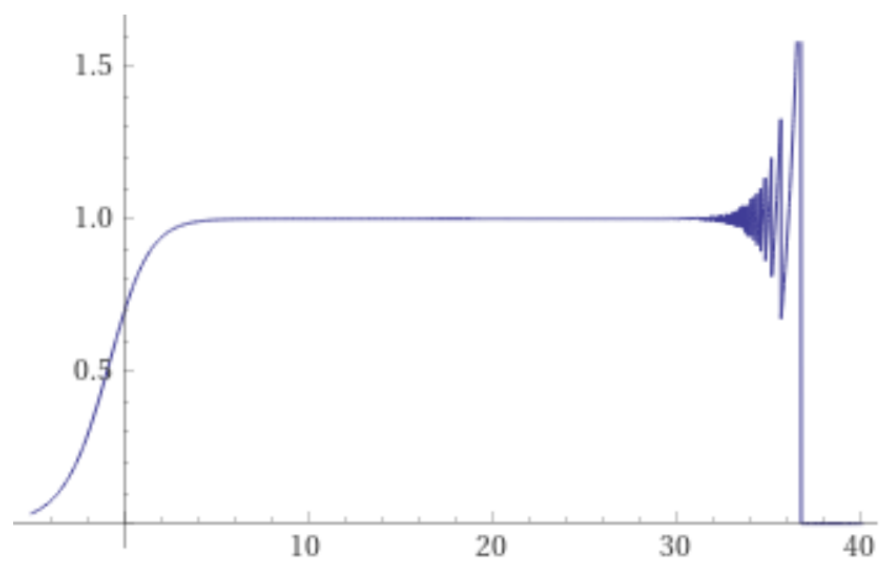
Funkcję $f(x)$ wpisać do programu umożliwiającego rysowanie wykresów. Programy, które wybrałem do narysowania funkcji $f(x)$ to: WolframAlpha, Grapher oraz postanowiłem napisać programy przedstawiający wykresy w języku programowania Python z wykorzystaniem biblioteki matplotlib in numpy

2.3 Otrzymane wyniki

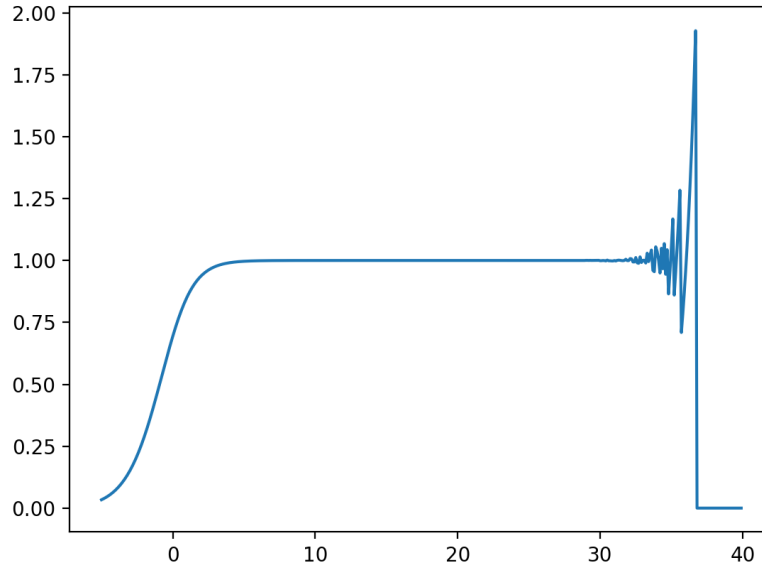
W programie Grapher należało wpisać funkcję w wyznaczone miejsce. Następnie przeskalować oś x, wykonując następujące czynności: View > Frame Limit ... i zmienić skalowanie. Na stronie WolframAlpha należało wpisać `plot <funkcja> from -5 to 40`. Program napisany w pythonie znajduje się w pliku o ścieżce `./zad2/plotpython.py`



Rysunek 1: Wykres funkcji $f(x)$ w programie Grapher



Rysunek 2: Wykres funkcji $f(x)$ w programie WolframAlpha



Rysunek 3: Wykres funkcji $f(x)$ w języku Python z wykorzystaniem biblioteki matplotlib

Obliczmy teraz granicę funkcji:

$$\begin{aligned} \lim_{x \rightarrow \infty} e^x \ln(e^{-x} + 1) &= \lim_{x \rightarrow \infty} \frac{e^x \ln(e^{-x} + 1) e^{-x}}{e^{-x}} = \lim_{x \rightarrow \infty} \frac{\ln(e^{-x} + 1)}{e^{-x}} \stackrel{H}{=} \\ &\stackrel{H}{=} \lim_{x \rightarrow \infty} \frac{\frac{d}{dx} \ln(e^{-x} + 1)}{\frac{d}{dx} e^{-x}} = \lim_{x \rightarrow \infty} \frac{-\frac{e^{-x}}{e^{-x} + 1}}{-e^{-x}} = \lim_{x \rightarrow \infty} \frac{1}{e^{-x} + 1} = 1 \quad (1) \end{aligned}$$

Jak łatwo zauważyć, granica funkcji, którą obliczyliśmy powyżej nie pokrywa się z granicą funkcji odczytując ją z wykresu. Na wykresie granica funkcji dąży do 0 natomiast z matematycznego punktu widzenia dąży do 1. Powyżej argumentów powyżej 30 obserwujemy, że wykresy zaczynają pokazywać nieprawidłowe wyniki.

2.4 Wnioski

Wartości e^{-x} dla każdego następnego argumentu zbliżając się do epsilon maszynowego, po przekroczeniu epsilon, funkcja spada do zera. Przed przekroczeniem epsilon maszynowego, funkcja dąży do jedynki, tak jak powinna.

wykładnik	$\exp(\text{wykładnik})$	$\exp(\text{wykładnik}) - \text{epsilon}$
30	9.357622968840175e-14	9.335418508347672e-14
31	3.442477108469977e-14	3.4202726479774736e-14
32	1.2664165549094176e-14	1.2442120944169144e-14
33	4.658886145103398e-15	4.4368415401783664e-15
34	1.713908431542013e-15	1.4918638266169817e-15
35	6.305116760146989e-16	4.084670710896676e-16
36	2.319522830243569e-16	9.907678099325606e-18
37	8.533047625744066e-17	-1.3671412866759066e-16
38	3.1391327920480296e-17	-1.90653277004551e-16
39	1.1548224173015786e-17	-2.104963807520155e-16
40	4.248354255291589e-18	-2.1779625066973972e-16

Tablica 3: Wartości $\exp()$ dla kolejnych argumentów porównanie z epsilonem maszynowym

3 Zadanie 3

3.1 Opis zadania

Porównanie rozwiązań równań macierzowych dwoma metodami: macierzy odwrotnej i eliminacja Gausa dla dwóch rodzajów macierzy: Macierzy Hilberta oraz macierzy losowej.

3.2 Metoda rozwiązywania

Rozwiązanie znajduje się w pliku `zad3.jl`, zawiera one funkcję napisane przez prowadzącego do generowania odpowiednich macierzy. Przez wbudowane funkcje w język Julia, bez trudności jest zaimplementowanie powyższego problemu.

3.3 Otrzymane wyniki

Rozwiązania przedstawie w dwóch tabelkach z podziałem na macierze, gdzie w pierwszej tabeli rozpatrujemy macierz Hilberta, natomiast w kolejnej macierz losową stopnia n ze wskaźnikiem uwarunkowania.

size	rank()	cond()	met. el. Gaussa	met. m. odwrotną
1	1	1.0	0.0	0.0
2	2	19.3	5.66e-16	1.4e-15
3	3	524.0	8.02e-15	0.0
4	4	15500.0	4.14e-14	0.0
5	5	477000.0	1.68e-12	3.35e-12
6	6	1.5e7	2.62e-10	2.02e-10
7	7	4.75e8	1.26e-8	4.71e-9
8	8	1.53e10	6.12e-8	3.08e-7
9	9	4.93e11	3.88e-6	4.54e-6
10	10	1.6e13	8.67e-5	0.00025
11	10	5.22e14	0.000158	0.00762
12	11	1.75e16	0.134	0.259
13	11	3.34e18	0.11	5.33
14	11	6.2e17	1.46	8.71
15	12	3.67e17	4.7	7.34
16	12	7.87e17	54.2	29.8
17	12	1.26e18	13.7	10.5
18	12	2.24e18	9.13	7.58
19	13	6.47e18	9.72	12.2
20	13	1.36e18	7.55	22.1

Tablica 4: Wyniki dla macierzy Hilberta używając metody macierzy odwrotnej i eliminacji Gaussa

Macierz Hilberta, jest bardzo dobrym przykładem macierzy źle uwarunkowanej, wraz ze wzrostem rozmiaru macierzy, wzrasta $\text{cond}(x)$ i wyniki są coraz bardziej rozstrzelane.

size	rank()	cond()	met. el. Gaussa	met. m. odwrotną
5	5	1.0	9.93e-17	9.93e-17
5	5	10.0	1.11e-16	2.81e-16
5	5	1000.0	2.31e-14	2.6e-14
5	5	1.0e7	1.18e-10	9.39e-11
5	5	1.0e12	5.46e-6	3.81e-6
5	4	1.46e16	0.33	0.161
10	10	1.0	2.72e-16	1.79e-16
10	10	10.0	2.19e-16	3.39e-16
10	10	1000.0	3.22e-14	2.69e-14
10	10	1.0e7	1.18e-10	1.38e-10
10	10	1.0e12	1.53e-5	1.26e-5
10	9	9.52e15	0.121	0.148
20	20	1.0	3.84e-16	2.66e-16
20	20	10.0	5.17e-16	7.69e-16
20	20	1000.0	9.71e-15	8.34e-15
20	20	1.0e7	2.04e-10	1.84e-10
20	20	1.0e12	3.54e-7	4.35e-6
20	19	1.2e16	0.156	0.125

Tablica 5: Wyniki dla macierzy generowanej losowo z używając metody macierzy odwrotnej i eliminacji Gaussa

Warto zauważyć, że dla rozmiaru macierzy 20x20 i wskaźniku uwarunkowania błędy są mniejsze niż dla macierzy rozmiaru 5x5 i wskaźniku uwarunkowania 1.46e16

3.4 Wnioski

Uwarunkowanie macierzy ma wpływ na otrzymane wyniki. Im większy $\text{cond}(x)$ otrzymujemy coraz większe błędy przy dowolnej metodzie. Dla macierzy Hilberta, wraz ze wzrostem rozmiaru macierzy, zwiększa się $\text{cond}(x)$ i z każdą iteracją otrzymujemy to coraz mniej dokładne wyniki. Dla macierzy losowej wyniki dla dużych rozmiarów ale z małym wskaźnikiem uwarunkowania dają lepsze wyniki niż dla małych macierzy z dużym wskaźnikiem.