

Agata Jasionowska 229726

Laboratorium – Lista 2

1. Zadanie 1

1.1. Opis problemu

Ponowne rozwiązanie zadania 5 z listy 1, jednak z wykorzystaniem nieznacznie zmienionych danych (usunięcie ostatnich cyfr w x_4 oraz x_5). Ta modyfikacja prezentuje się następująco:

$$x_4 = 0.5772156649 \quad x'_4 = 0.577215664$$

$$x_5 = 0.3010299957 \quad x'_5 = 0.301029995$$

W dalszej części sprawozdania wartości iloczynu skalarnego dla niezmienionych danych oznaczane będą jako I , zaś po modyfikacji jako I' .

1.2. Opis rozwiązania

W celu obliczenia iloczynów skalarnych użyto kodu zadania 5 listy 1 na zmodyfikowanych zgodnie z treścią zadania danych.

1.3. Wyniki

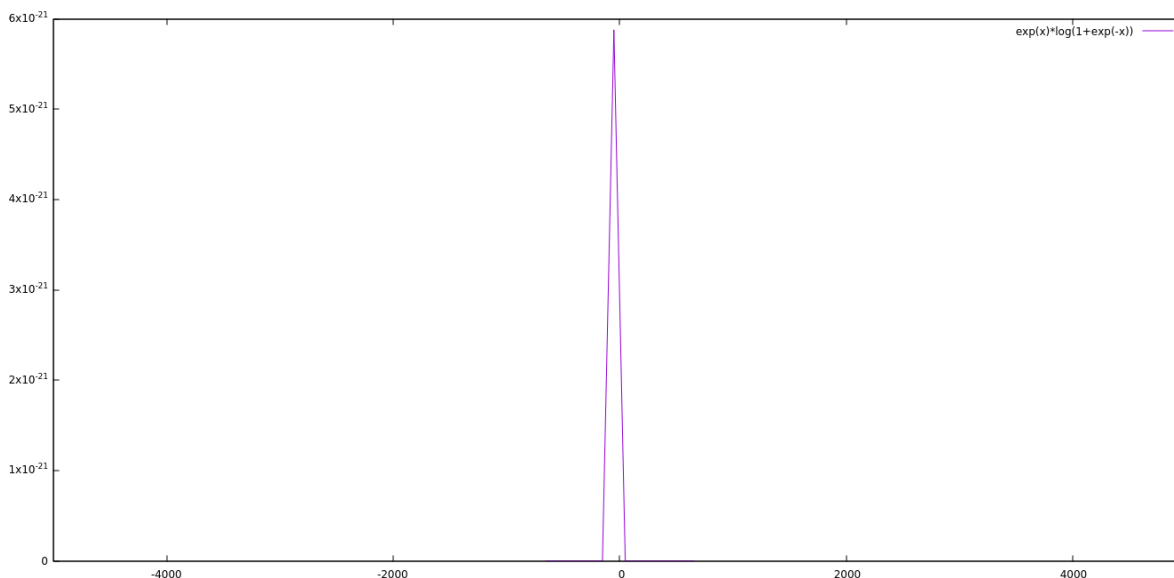
Poniższa tabela prezentuje uzyskane wyniki dla czterech algorytmów obliczających iloczyn skalarny, zestawiając rozwiązania dla I oraz I' :

1.4. Wnioski

Uzyskane wyniki pokazują, że usunięcie cyfr zgodnie z poleceniem nie wpłynęło na rezultaty dla obliczeń w arytmetyce `Float32`. Wynika to ze względu na stosunkowo niską precyzję obliczeń w przypadku tej arytmetyki. Potwierdza to analiza zapisu bitowego zmienionych wartości - reprezentacja x_4 oraz x'_4 wygląda tu identycznie, zaś dla x_5 różnica pojawia się dopiero na najmniej znaczącym bicie. Zupełnie inna sytuacja ma miejsce w przypadku arytmetyki `Float64` - rozbieżności są bardzo wyraźnie widoczne. Ostatecznie rezultaty są dalekie od oczekiwanych; spowodowane jest to faktem, iż dane wektory są prawie prostopadłe. W przypadku tego problemu nieznaczne zmiany wpływają na duże błędy końcowe, co potwierdza, że zadanie jest źle uwarunkowane.

podpunkt	I	I'
Float32		
1	-0.499 944 3	-0.499 944 3
2	-0.454 345 7	-0.454 345 7
3	-0.5	-0.5
4	-0.5	-0.5
Float64		
1	-0.499 944 3	-0.004 296 342 739 891 585
2	-0.454 345 7	-0.004 296 342 998 713 953
3	-0.5	-0.004 296 342 842 280 865
4	-0.5	-0.004 296 342 842 280 865

Tabela 1. iloczyn skalarny wektorów.



Rysunek 1. Wykres funkcji w GNUPlot

2. Zadanie 2

2.1. Opis problemu

W co najmniej dwóch wybranych programach do wizualizacji narysować wykres funkcji $f(x) = e^x \ln(1 + e^{-x})$ oraz policzyć granicę $\lim_{x \rightarrow \infty} f(x)$.

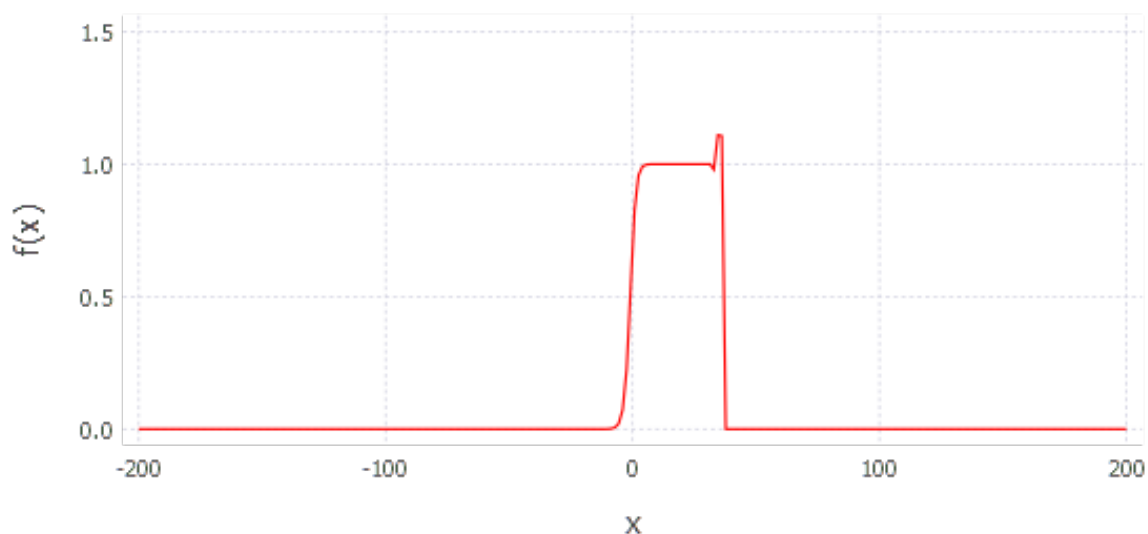
2.2. Opis rozwiązania

Wykresy zostały narysowane z użyciem programu GNUPlot oraz pakietu Gadfly, zaś granicę policzono przy pomocy funkcji `limit` języka Julia (SymPy).

2.3. Wyniki

Uzyskane wykresy przedstawione są na rysunkach poniżej na Rys. 1 oraz Rys. 2.

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



Rysunek 2. Wykres funkcji w Gadfly

2.4. Wnioski

WNIOSKI!!! Obecność fluktuacji: dla Float64 - $x \in (32, 37)$, Float32 - $x \in (12, 20)$

3. Zadanie 3

3.1. Opis problemu

Rozwiązanie układu równań liniowych $Ax = b$ dla danej macierzy współczynników $A \in \mathbb{R}^{n \times n}$ i wektora prawych stron $b \in \mathbb{R}^n$ za pomocą algorytmów: eliminacji Gaussa ($x = A/b$) oraz $x = A^{-1}b$ ($x = \text{inv}(A) * b$). Macierz A generowana jest w następujący sposób:

- (a) $A = H_n$, gdzie H_n jest macierzą Hilberta stopnia n ;
- (b) $A = R_n$, gdzie R_n jest losową macierzą stopnia n o zadanym wskaźniku uwarunkowania.

Przeprowadzenie eksperymentów dla macierzy Hilberta H_n z rosnącym stopniem $n > 1$ oraz dla macierzy losowej R_n , $n = 5, 10, 20$ z rosnącym wskaźnikiem uwarunkowania $c = 1, 10, 10^3, 10^7, 10^{12}, 10^{16}$, a także obliczenie błędów względnych.

3.2. Opis rozwiązania

3.3. Wyniki

3.4. Wnioski

WNIOSKI!!!

4. Zadanie 4

4.1. Opis problemu

Przeprowadzenie eksperymentu obliczania zer wielomianu Wilkinsona, zapisanego pod dwiema następującymi postaciami:

$$P(x) = x^{20} - 210x^{19} + 20615x^{18} \dots$$

$$p(x) = (x - 20)(x - 19)(x - 18) \dots (x - 2)(x - 1)$$

oraz określenie błędu bezwzględnego wartości tego wielomianu dla otrzymanych pierwiastków. Następnie powtórzenie testu dla współczynnika przy x^{19} zmienionego na $-210 - 2^{-23}$.

4.2. Opis rozwiązania

Rozwiązanie zadania przygotowano w języku Julia z użyciem trzech funkcji dostępnych w pakiecie `Polynomials`. Przebiega ono zgodnie z poniższymi krokami:

1. Utworzenie wielomianu na podstawie podanych współczynników w postaci kanonicznej (funkcja `Poly`), a także iloczynowej (funkcja `poly`);
2. Obliczenie pierwiastków wielomianu przy użyciu funkcji `roots`;
3. Prezentacja błędu bezwzględnego obu postaci wielomianu oraz miejsc zerowych.

4.3. Wyniki

k	$ P(z_k) $	$ p(z_k) $
1	$1.114\,453\,504\,512 \times 10^{13}$	$1.374\,373\,319\,724\,971\,3 \times 10^{18}$
2	$9.539\,424\,609\,817\,828 \times 10^{12}$	$4.252\,502\,487\,993\,469\,4 \times 10^{17}$
3	$9.539\,424\,609\,817\,828 \times 10^{12}$	$4.252\,502\,487\,993\,469\,4 \times 10^{17}$
4	$3.315\,103\,475\,981\,763 \times 10^{11}$	$2.742\,089\,401\,676\,406\,4 \times 10^{16}$
5	$3.315\,103\,475\,981\,763 \times 10^{11}$	$2.742\,089\,401\,676\,406\,4 \times 10^{16}$
6	$1.061\,206\,453\,308\,197\,6 \times 10^{11}$	$9.545\,941\,595\,183\,662 \times 10^{14}$
7	$1.061\,206\,453\,308\,197\,6 \times 10^{11}$	$9.545\,941\,595\,183\,662 \times 10^{14}$
8	$3.357\,756\,113\,171\,857 \times 10^{10}$	$3.296\,021\,414\,130\,166\,4 \times 10^{13}$
9	$3.357\,756\,113\,171\,857 \times 10^{10}$	$3.296\,021\,414\,130\,166\,4 \times 10^{13}$
10	$7.143\,113\,638\,035\,824 \times 10^9$	$1.491\,263\,381\,675\,401\,9 \times 10^{12}$
11	$7.143\,113\,638\,035\,824 \times 10^9$	$1.491\,263\,381\,675\,401\,9 \times 10^{12}$
12	$3.065\,575\,424 \times 10^9$	$1.371\,743\,170\,56 \times 10^{11}$
13	$1.072\,547\,328 \times 10^9$	$1.852\,548\,659\,2 \times 10^{10}$
14	$3.881\,231\,36 \times 10^8$	$1.757\,670\,912 \times 10^9$
15	$1.291\,484\,16 \times 10^8$	$2.061\,204\,48 \times 10^8$
16	$3.946\,393\,6 \times 10^7$	$4.330\,393\,6 \times 10^7$
17	$1.046\,784 \times 10^7$	$1.072\,998\,4 \times 10^7$
18	$2.221\,568 \times 10^6$	$2.295\,296 \times 10^6$
19	349 184.0	365 568.0
20	20 992.0	22 016.0

Tabela 2. Błędy bezwzględne uzyskanych pierwiastków wielomianu Wilkinsona.

4.4. Wnioski

WNIOSKI !!!

k	$ P(z_k) $
1	19.846 910 215 194 79
2	17.609 667 147 937 33
3	16.616 121 458 496 472
4	13.037 742 289 370 026
5	12.063 218 237 453 977
6	8.379 918 919 523 833
7	7.432 242 442 817 582
8	4.138 150 126 108 56
9	3.245 998 350 871 97
10	0.651 958 683 038 040 6
11	1.110 918 027 271 656 1
12	3.084 183 632 067 441 4
13	4.992 227 970 900 554
14	7.000 397 929 577 58
15	8.999 979 523 326 969
16	11.000 001 426 112 089
17	12.999 999 910 275 637
18	15.000 000 003 396 579
19	16.999 999 999 944 96
20	19.000 000 000 000 163

Tabela 3. Błędy bezwzględne uzyskanych pierwiastków wielomianu Wilkinsona.

5. Zadanie 5

5.1. Opis problemu

Przeprowadzenie dwóch eksperymentów przy wykorzystaniu następującego modelu wzrostu populacji:

$$p_{n+1} := p_n + rp_n(1 - p_n), \text{ dla } n = 1, 2, \dots,$$

gdzie r jest pewną stałą, $r(1 - p_n)$ jest czynnikiem wzrostu populacji, a p_0 jest wielkością populacji stanowiącą procent maksymalnej wielkości populacji dla danego stanu środowiska. Testy przeprowadzić dla danych wejściowych we wskazanych arytmetykach:

- (a) $p_0 = 0.01$, $r = 3$, $n = 40$ {Float32}
- (b) $p_0 = 0.01$, $r = 3$, $n = 40$ {Float64}
- (c) $p_0 = 0.01$, $r = 3$, $n = 40$, po 10 iteracjach zastosowanie obciążenia wyniku do trzeciego miejsca po przecinku {Float32}.

5.2. Opis rozwiązania

Rozwiązanie problemu zrealizowano w języku Julia.

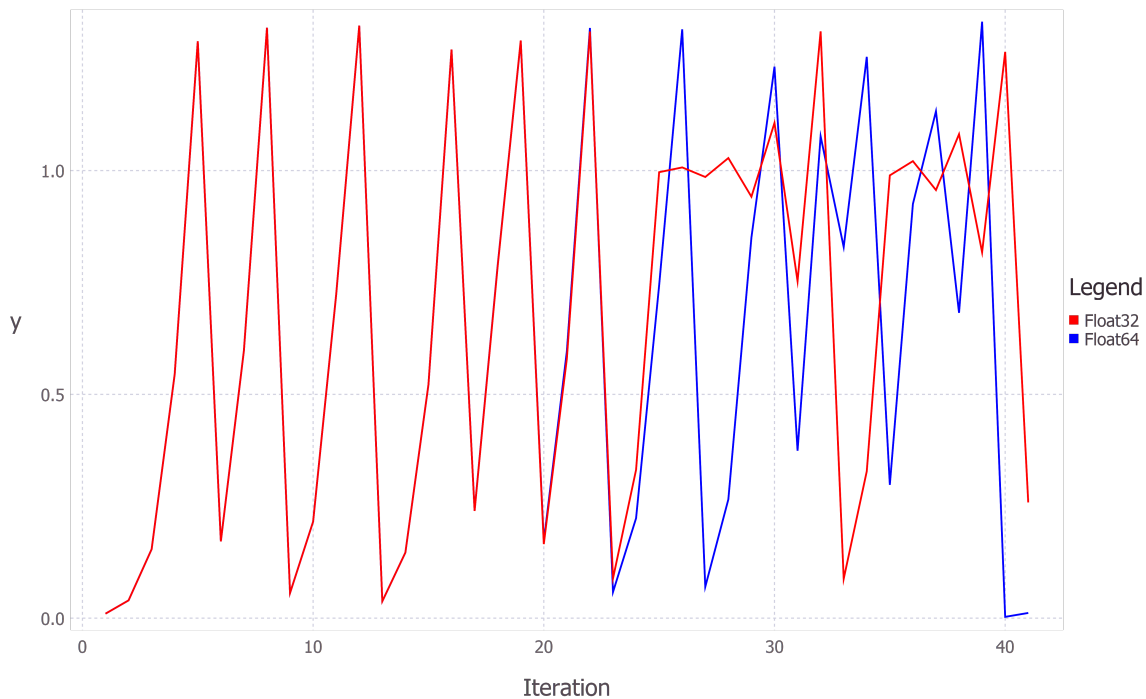
5.3. Wyniki

Uzyskane rezultaty dla przeprowadzonych eksperymentów:

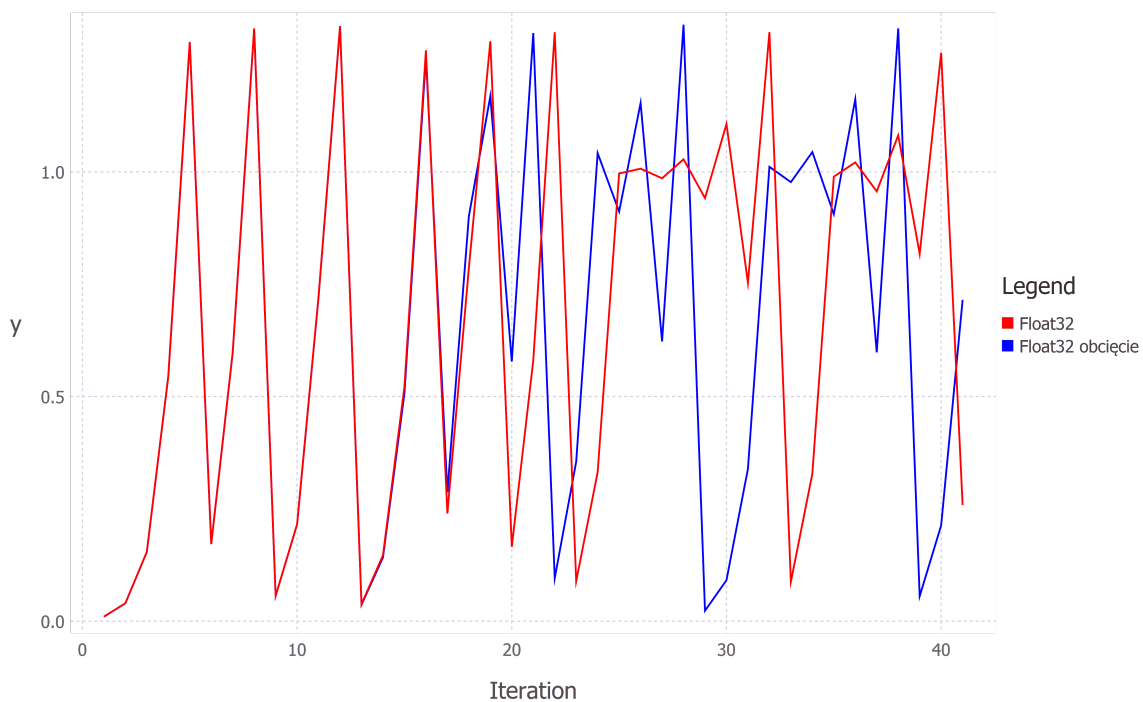
1. Zestawienie wyników dla danych (a.) oraz (c.)
2. Zestawienie wyników dla danych (a.) oraz (b.)

5.4. Wnioski

W przypadku pierwszego eksperymentu doskonale można zaobserwować, jak początkowo zdawałoby się znikomy błąd staje się wraz z kolejnymi iteracjami tak duży, iż stanowi oddzielną iterację. !!! Analiza drugiego wykresu pozwala dostrzec różnice w regularności przebiegu. Niższa



Rysunek 3. Zestawienie wyników dla danych (a.) oraz (c.).



Rysunek 4. Zestawienie wyników dla danych (a.) oraz (b.).

precyzja arytmetyki `Float32` wpłynęła na stale powiększające się różnice, by osiągnąć swoistą kulminacyjną wartość równą 1.0. Każda kolejna iteracja oscylowała w pobliżu właśnie tego wyniku. Ciekawym spostrzeżeniem jest fakt, iż dla arytmetyki `Float64` ma miejsce podobna

sytuacja, jednak ma ona miejsce w znacznie dalszej iteracji.

Powyższe eksperymenty pozwalają dostrzec, że błąd obliczeniowy wyniku, przeniesiony jako wejście kolejnej operacji, potęguje powstały błąd. Jest to typowy przykład sprzężenia zwrotnego - procesu, w którym dane wyjściowe jednego problemu są wejściem kolejnych obliczeń. W uniknięciu tego rodzaju sytuacji pomóc może zwiększenie precyzji obliczeń, jednak w daleko wybiegających symulacjach powstałe błędy mogą zniekształcić obliczenia w tak dużym stopniu, iż również te wyniki staną się dla badającego bezużyteczne.

6. Zadanie 6

6.1. Opis problemu

Przeprowadzenie eksperymentów w języku `Julia` w arytmetyce `Float64` dla następującego równania rekurencyjnego:

$$x_{n+1} := x_n^2 + c \text{ dla } n = 0, 1, \dots,$$

gdzie c jest pewną stałą, dla danych wejściowych:

1. $c = -2$ i $x_0 = 1$;
2. $c = -2$ i $x_0 = 2$;
3. $c = -2$ i $x_0 = 1.9999999999999999$;
4. $c = -1$ i $x_0 = 1$;
5. $c = -1$ i $x_0 = -1$;
6. $c = -1$ i $x_0 = 0.75$;
7. $c = -1$ i $x_0 = 0.25$;

oraz ilości iteracji $n = 40$.

6.2. Opis rozwiązania

6.3. Wyniki

6.4. Wnioski