Obliczenia naukowe
Lista 2

Szymon Janiak

November 5, 2023

Zadanie 1

Opis problemu

Obliczenie iloczynu skalarnego róznymi funkcjami dwóch wektorów i porównanie wyników przy lekkiej zmianie danych wejściowych

Rozwiązanie

1. "w przód" t.j. $\sum_{i=1}^{n} x_i y_i$
2. "w tył" t.j. $\sum_{i=n}^{1} x_i y_i$
3. "liczby dodatnie od najwiekszego do najmniejszego a ujemne na odwrót"
4. "liczby ujemne od najwiekszego do najmniejszego a dodatnie na odwrót"

Wyniki

	Float64 stare dane	Float64 nowe dane	Prawidłowy wynik
"1"	1.0251881368296672e-10	-0.004296342739891585	-1.00657107000000e-11
"2"	-1.5643308870494366e-10	-0.004296342998713953	-1.00657107000000e-11
"3"	0.0	-0.004296342842280865	-0.004296342842280865
"4"	0.0	-0.004296342842280865	-0.004296342842280865

Wnioski

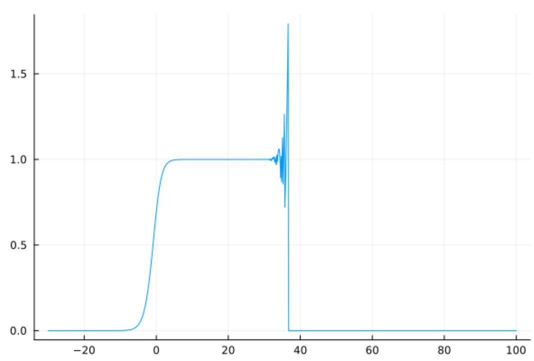
Przy usunięciu ostatniej 9 z x_4 oraz ostatniej 7 z x_5 dostajemy różne wyniki dla podwójnej precyzji. Po tak lekkiej zmianie danych możemy zauważyć, że wyniki dla wszystkich funkcji są znacznie bardziej przybliżone, prawie identyczne. Dla Float32 nie ma żadnej różnicy, gdyż jest to za mała precyzja. Według definicji jest to źle uwarunkowane zadanie.

Zadanie 2

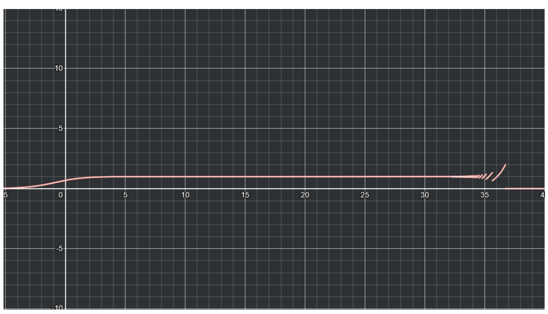
Opis problemu

Narysować wykres funkcji $f(x) = e^x ln(1 + e^{-x})$ w co najmniej dwóch dowolnych programach do wizualizacji.
Porównać wykres funkcji z policzoną granicą dla funkcji $\lim_{x\to\infty} f(x)$.

Wyniki



(a) Biblioteka Plots w języku Julia


(b) Desmos

Poniżej policzona granica:

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

Wnioski

Widzimy na wykresach, że w okolicach $x = 30$ pojawiają się problemy z obliczaniem wartości naszej funkcji.
Mimo, że funkcja dąży do 1 nasze programy mają wyraźny problem z obliczaniem wartości składających się na wynik naszej funkcji. Jest to prawdopodobnie spowodowane brakiem wystarczającej precyzji kiedy wartości e^{-x} robią się coraz mniejsze.

Zadanie 3

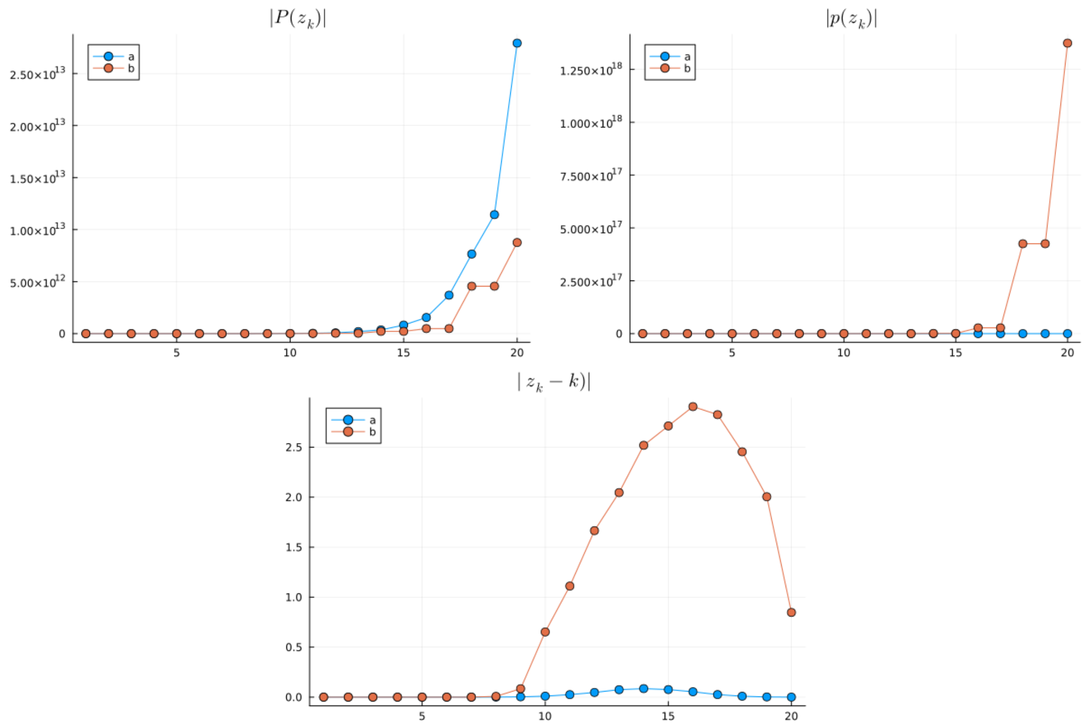
Opis problemu

Zadanie 4

Opis problemu

Użyć funkcji *roots* (z pakietu Polynomials) do obliczenia 20 zer wielominau P w postaci naturalnej.
P jest postacią naturalną wielomianu Wilkinsona p
$p(x) = (x-20)(x-19)(x-18)(x-17)(x-16)(x-15)$
$(x-14)(x-13)(x-12)(x-11)(x-10)(x-9)$
$(x-8)(x-7)(x-6)(x-5)(x-4)(x-3)(x-2)(x-1)$

Wyniki



Wnioski

Po wykresach widać, że obliczone przez nas miejsca zerowe nie pokrywają się z rzeczywistością.
Wynika to z ograniczeń arytmetyki w Float64, która ma od 15 do 17 cyfr znaczących w systemie dziesiętnym.
Po modyfikacji jednego współczynnika rozbieżności są jeszcze większe.

Zadanie 5

Opis problemu

Zadanie 6

Opis problemu

Dla równania rekurencyjnego

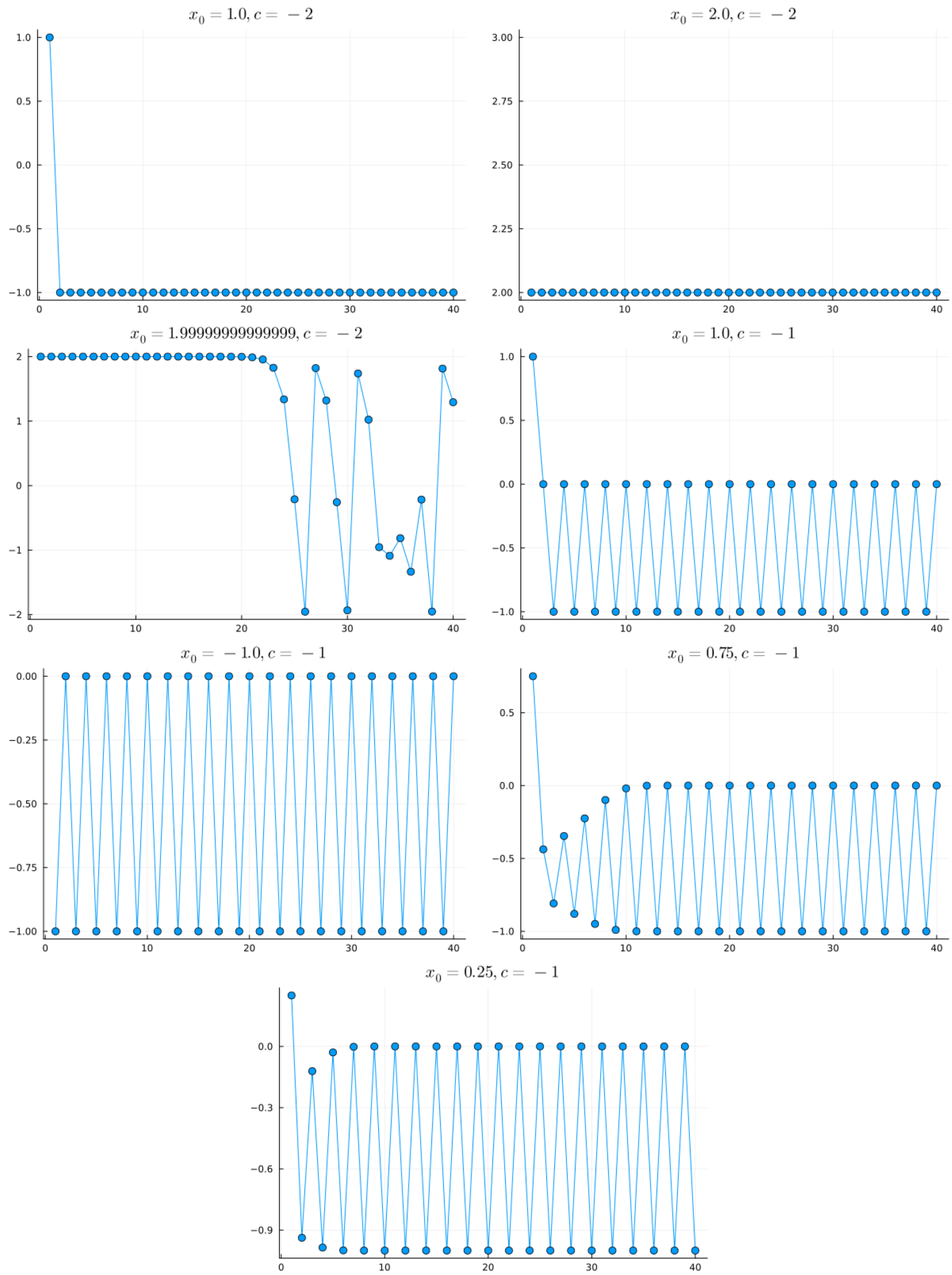
$$x_{n+1} := x_n^2 + c \text{ dla } n = 0, 1...,$$

Przeprowadzić następujące eksperymenty. Dla danych:

1. $c = -2$ i $x_0 = 1$
2. $c = -2$ i $x_0 = 2$
3. $c = -2$ i $x_0 = 1.99999999999999$
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 $x_0 = 0.25$

wykonać w arytmetyce Float64, 40 iteracji podanego wyrażenia i przeprowadzić iteracje graficzną.

Wyniki



Wnioski