# Obliczenia naukowe
# Lista 2

Joanna Szołomicka

2019-11-04

# 1 Zadanie 1

## 1.1 Opis problemu

Zadanie jest powtórzeniem zadania 5 z listy 1 z niewielką zmianą danych:
1) $x_5 =, 0.577215664$ i $x_7 = 0.301029995$ zamiast
2) $x_5 =, 0.5772156649$ i $x_7 = 0.3010299957$.

Polega na obliczeniu iloczynu skalarnego dwóch wektorów
$x' = [2.718281828, -3.141592654, 1.414213562, 0.577215664, 0.301029995]$
$y = [1486.2497, 878366.9879, -22.37492, 4773714.647, 0.000185049]$
na cztery sposoby dla typów Float32 i Float64.

## 1.2 Rozwiązanie

W programie użyto czterech algorytmów:

1. "w przód"
   $\sum_{i=1}^{n} x'_i y_i$
2. "w tył"
   $\sum_{i=n}^{1} x'_i y_i$
3. Dodanie liczb dodatnich od największej do najmniejszej, a liczb ujemnych od najmniejszej do największej.
4. Przeciwnie do algorytmu trzeciego.

## 1.3 Wyniki

| **algorytm** | **Float32** | **Float64** |
|---|---|---|
| Wyniki 1 | | |
| 1 | $-0.4999443$ | $-0.004296342739891585$ |
| 2 | $-0.4543457$ | $-0.004296342998713953$ |
| 3 | $-0.5$ | $-0.004296342842280865$ |
| 4 | $-0.5$ | $-0.004296342842280865$ |
| Wyniki 2 | | |
| 1 | $-0.4999443$ | $1.0251881368296672 * 10^{-10}$ |
| 2 | $-0.4543457$ | $-1.5643308870494366 * 10^{-10}$ |
| 3 | $-0.5$ | 0.0 |
| 4 | $-0.5$ | 0.0 |

Tabela 1: Wartości iloczynów skalarnych.

## 1.4 Wnioski

Wyniki dla Float32 są takie same dla oryginalnych danych i lekko zmienionych. Wprowadzony błąd jest mniejszy niż precyzja arytmetyki Float32. W arytmetyce Float64 niewielka zmiana w danych spowodowała dużą zmianę w wynikach. Z drugiej strony rozbieżność między wynikami poszczególnych sum się zmniejszyła. Jest to przykład źle uwarunkowanego zadania, co w tym przypadku jest spowodowane tym, że wektory są prawie prostopadłe.

# 2 Zadanie 2

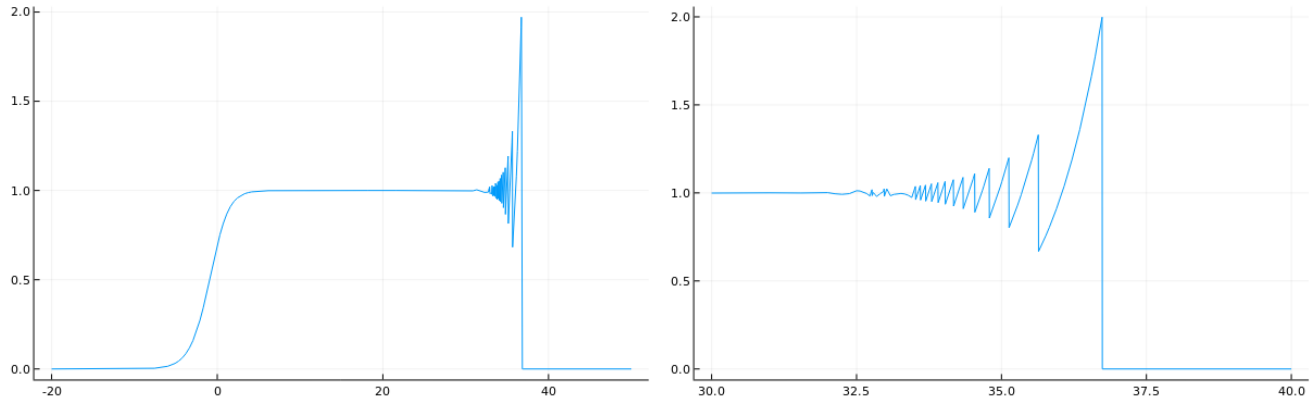
## 2.1 Opis problemu

Zadanie polega na porównaniu wykresów funkcji $f(x) = e^x \ln(1 + e^{-x})$ narysowanych w co najmniej dwóch programach graficznych z policzoną granicą $\lim_{x \to \infty} f(x)$.

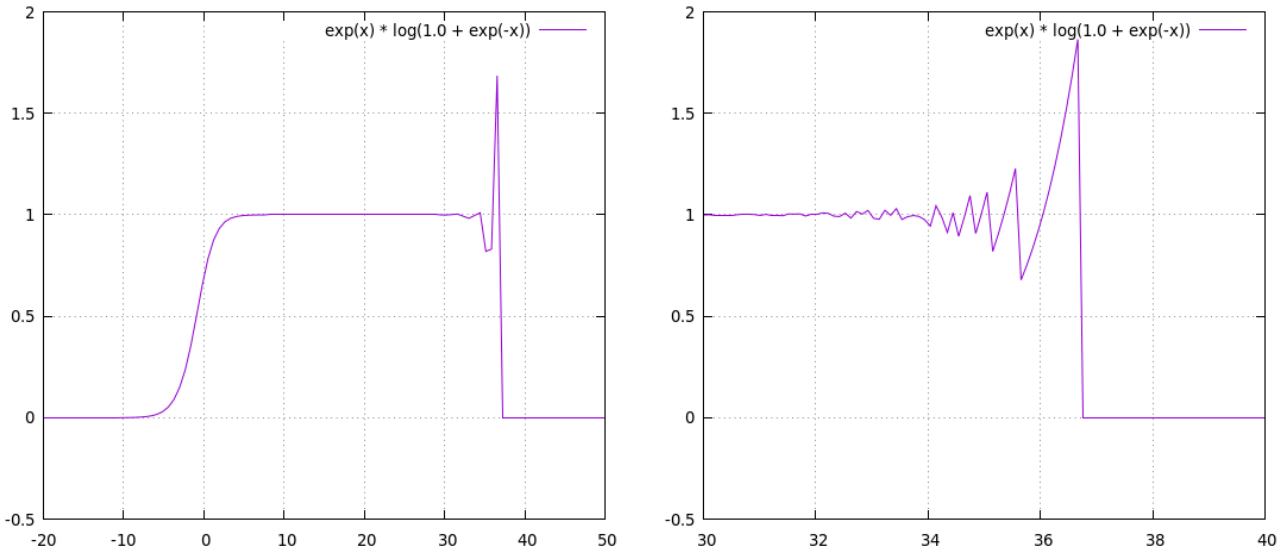
## 2.2 Rozwiązanie

Granica funkcji wynosi $\lim_{x \to \infty} (e^x \ln(1 + e^{-x})) = 1$. Wykresy zostały narysowane przy użyciu biblioteki SymPy w języku Julia oraz w programie Gnuplot. Rozpatruję wykresy funkcji w dwóch przedziałach $[-20, 50]$ oraz $[30, 40]$ dla argumentów o precyzji double.

## 2.3 Wyniki




Rysunek 1: Wykresy wykonane za pomocą biblioteki *SymPy*




Rysunek 2: Wykresy wykonane za pomocą programu *Gnuplot*

### 2.4 Wnioski

Można zauważyć, że w pewnym momencie wyniki odbiegają od prawidłowej wartości granicy, która wynosi 1. Zaburzenia są spowodowane dodaniem małej wartości $e^{-x}$ do dużej w porównaniu do niej 1. Następnie logarytm tej sumy, który dąży do 0 jest mnożony razy $e^x$ dążące do nieskończoności, mnożenie bardzo małej liczby razy bardzo dużą zwiększa błąd. Na wykresie widać, że błędny wynik jest generowany do momentu aż $\ln(1 + e^{-x})) = 0$, czyli gdy $e^{-x}$ zostanie "pochłonięte" przez 1. Granica jest wtedy równa $\infty * 0 = 0$.

## 3 Zadanie 3

### 3.1 Opis problemu

Zadanie polega na rozwiązaniu układu równań liniowych $\mathbf{Ax} = \mathbf{b}$, gdzie $\mathbf{A} \in \mathbb{R}^{n\times n}$ to macierz współczynników generowana jako:

1. macierz Hilberta $\mathbf{H}_n$ stopnia $n$,
2. macierz losowa $\mathbf{R}_n^c$ stopnia $n$ o zadanym wskaźniku uwarunkowania $c$.

oraz $\mathbf{b}$ to wektor prawych stron $\mathbf{b} \in \mathbb{R}^n$ zadany jako $\mathbf{b} = \mathbf{Ax}$, gdzie $\mathbf{A}$ jest wygenerowaną macierzą, a $\mathbf{x} = (1, ..., 1)^{\mathbb{T}}$, tak że znamy dokładne rozwiązanie dla $\mathbf{A}$ i $\mathbf{b}$.

Należy obliczyć $\tilde{\mathbf{x}}$ za pomocą dwóch algorytmów:

1. metody eliminacji Gaussa: $\tilde{\mathbf{x}} = \mathbf{A}\backslash\mathbf{b}$,
2. $\tilde{\mathbf{x}} = \mathbf{A}^{-1}\mathbf{b}$.

dla $\mathbf{H}_n$ z rosnącym stopniem $n > 1$ oraz $\mathbf{R}_n^c$ dla $n = 5, 10, 20$ i $c = 1, 10, 10^3, 10^7, 10^{12}, 10^{16}$. Rozwiązania $\tilde{\mathbf{x}}$ porównać z wartością dokładną $\mathbf{x}$.

### 3.2 Rozwiązanie

Zadane macierze zostały wygenerowane funkcjami $hilb(A)$ oraz $matcond(n, c)$. Następnie obliczony został $\tilde{x}$ metodą Gaussa oraz odwrotnej macierzy dla obu tych macierzy. Przy pomocy funkcji $cond(A)$ obliczyłam wskaźnik uwarunkowania wygenerowanych macierzy, dzięki funkcji $rank(A)$ sprawdziłam ich rząd. Błąd względny został obliczony ze wzoru $\frac{||x - \tilde{x}||}{||x||}$ .

### 3.3 Wyniki

| Macierz Hilberta | | | | |
|---|---|---|---|---|
| **Rozmiar** | **Rząd** | **Błąd względny metody Gaussa** | **Błąd względny metody inwersji** | **Wskaźnik uwarunkowania** |
| $1 \times 1$ | 1 | 0.0 | 0.0 | 1.0 |
| $2 \times 2$ | 2 | $5.661048867003676 * 10^{-16}$ | $1.4043333874306803 * 10^{-15}$ | 19.28147006790397 |
| $3 \times 3$ | 3 | $8.022593772267726 * 10^{-15}$ | 0.0 | 524.0567775860644 |
| $4 \times 4$ | 4 | $4.137409622430382 * 10^{-14}$ | 0.0 | 15513.73873892924 |
| $5 \times 5$ | 5 | $1.6828426299227195e * 10^{-12}$ | $3.3544360584359632 * 10^{-12}$ | 476607.25024259434 |
| $6 \times 6$ | 6 | $2.618913302311624 * 10^{-10}$ | $2.0163759404347654 * 10^{-10}$ | $1.4951058642254665 * 10^{7}$ |
| $7 \times 7$ | 7 | $1.2606867224171548 * 10^{-8}$ | $4.713280397232037 * 10^{-9}$ | $4.75367356583129 * 10^{8}$ |
| $8 \times 8$ | 8 | $6.124089555723088 * 10^{-8}$ | $3.07748390309622 * 10^{-7}$ | $1.5257575538060041 * 10^{10}$ |
| $9 \times 9$ | 9 | $3.8751634185032475 * 10^{-6}$ | $4.541268303176643 * 10^{-6}$ | $4.931537564468762 * 10^{11}$ |
| $10 \times 10$ | 10 | $8.67039023709691 * 10^{-5}$ | 0.0002501493411824886 | $1.6024416992541715 * 10^{13}$ |
| $11 \times 11$ | 10 | 0.00015827808158590435 | 0.007618304284315809 | $5.222677939280335 * 10^{14}$ |
| $12 \times 12$ | 11 | 0.13396208372085344 | 0.258994120804705 | $1.7514731907091464 * 10^{16}$ |
| $13 \times 13$ | 11 | 0.11039701117868264 | 5.331275639426837 | $3.344143497338461 * 10^{18}$ |
| $14 \times 14$ | 11 | 1.4554087127659643 | 8.71499275104814 | $6.200786263161444 * 10^{17}7$ |
| $15 \times 15$ | 12 | 4.696668350857427 | 7.344641453111494 | $3.674392953467974 * 10^{17}$ |
| $16 \times 16$ | 12 | 54.15518954564602 | 29.84884207073541 | $7.865467778431645 * 10^{17}$ |
| $17 \times 17$ | 12 | 13.707236683836307 | 10.516942378369349 | $1.263684342666052 * 10^{18}$ |
| $18 \times 18$ | 12 | 9.134134521198485 | 7.575475905055309 | $2.2446309929189128 * 10^{18}$ |
| $19 \times 19$ | 13 | 9.720589712655698 | 12.233761393757726 | $6.471953976541591 * 10^{18}$ |
| $20 \times 20$ | 13 | 7.549915039472976 | 22.062697257870493 | $1.3553657908688225 * 10^{18}$ |

Tabela 2: Wyniki dla macierzy Hilberta.

| Macierz losowa $\mathbf{R}_n^c$ | | | | |
|---|---|---|---|---|
| **Rozmiar** | **Rząd** | **Błąd względny metody Gaussa** | **Błąd względny metody inwersji** | **Wskaźnik uwarunkowania** |
| $5 \times 5$ | 5 | $2.482534153247273e-16$ | $1.9860273225978183e-16$ | 1.0000000000000013 |
| $5 \times 5$ | 5 | $2.6272671962866383e-16$ | $3.1401849173675503e-16$ | 10.0 |
| $5 \times 5$ | 5 | $1.369288066254624e-14$ | $1.18896540830928e-14$ | 1000.0000000000222 |
| $5 \times 5$ | 5 | $4.532689268483517e-10$ | $4.1852886817063086e-10$ | $1.0000000004302649e7$ |
| $5 \times 5$ | 5 | $2.2927486191515693e-5$ | $2.2519641331703858e-5$ | $9.99904436832464e11$ |
| $5 \times 5$ | 4 | 0.10260264000016615 | 0.08385254915624211 | $7.264870704482146e15$ |
| $10 \times 10$ | 10 | $2.248030287623391e-16$ | $2.9163173068810656e-16$ | 1.000000000000001 |
| $10 \times 10$ | 10 | $5.135906591666906e-16$ | $3.89370267511739e-16$ | 10.000000000000005 |
| $10 \times 10$ | 10 | $3.847899107996398e-14$ | $3.948150388114312e-14$ | 1000.0000000000583 |
| $10 \times 10$ | 10 | $2.7224743337994497e-10$ | $2.6120586498264907e-10$ | $9.999999992074e6$ |
| $10 \times 10$ | 10 | $8.466985253062836e-6$ | $1.041646379159023e-5$ | $1.0000301936811421e12$ |
| $10 \times 10$ | 9 | 1.240071747854101 | 1.381995698030569 | $1.0400037726311666e16$ |
| $20 \times 20$ | 20 | $4.591010489137866e-16$ | $3.439900227959406e-16$ | 1.0000000000000013 |
| $20 \times 20$ | 20 | $4.832978208675101e-16$ | $4.1836409184574146e-16$ | 10.000000000000007 |
| $20 \times 20$ | 20 | $1.3272065014729029e-14$ | $1.222203779958911e-14$ | 1000.0000000000152 |
| $20 \times 20$ | 20 | $1.1137910997375928e-10$ | $1.0381952732415501e-10$ | $1.0000000005650386e7$ |
| $20 \times 20$ | 20 | $1.5835965222350336e-5$ | $1.4151302577735613e-5$ | $9.999809635723966e11$ |
| $20 \times 20$ | 19 | 0.21821092110005602 | 0.24338163271008145 | $3.2581339672069984e16$ |

Tabela 3: Wyniki dla macierzy losowej.

### 3.4 Wnioski

Można zauważyć, że wielkość błędu jest zależna od wskaźnika uwarunkowania macierzy. Im większy wskaźnik uwarunkowania macierzy tym większy błąd względny. Macierz Hilberta jest bardzo źle uwarunkowaną macierzą, wraz ze zwiększaniem się rozmiaru wskaźnik gwałtownie rośnie, a przez to także błąd względny. Dla rozmiaru $20x20$ wskaźnik uwarunkowania wynosi aż $1.3553657908688225 * 10^{18}$. W przypadku macierzy Hilberta lepszą metodą jest metoda Gaussa, gdyż generuje mniejszy błąd względny. Analiza wyników macierzy losowej potwierdza poprawność tych zależności.

## 4 Zadanie 4

### 4.1 Opis problemu

Zadanie polega na obliczeniu 20 zer wielomianu Wilkinsona $P(x) = (x-20)(x-19)\dots(x-2)(x-1)$ w postaci kanoniczej. Należy sprawdzić otrzymane pierwiastki $z_k$ obliczając $|P(z_k)|$, $|p(z_k)|$ i $|z_k - k|$ dla $1 \leqslant x \leqslant 20$.Następnie powtórzyć eksperyment Wilkinsona zmieniając współczynnik $-210$ przy $x^{19}$ na $-210 - 2^{-23}$.

### 4.2 Rozwiązanie

W rozwiązaniu korzystałam z biblioteki *Polynomials*. Do uzyskania postaci kanonicznej wielomianu $P$ został użyty konstruktor *Poly*() z współczynnikami wielomianu jako argumentami. Miejsca zerowe wielomianu $P$ zostały obliczone za pomocą funkcji *roots*. Wielomian $p$ został stworzony przy użyciu metody *poly*. Wartości wielomianów w zadanych punktach zostały obliczone przy pomocy *polyval*.

### 4.3 Wyniki

| $k$ | $z_k$ | $\lvert P(z_k)\rvert$ | $\lvert p(z_k)\rvert$ | $\lvert z_k - k\rvert$ |
|---|---|---|---|---|
| 1 | 0.9999999999996989 | 36352.0 | 38400.0 | $3.0109248427834245e-13$ |
| 2 | 2.0000000000283182 | 181760.0 | 198144.0 | $2.8318236644508943e-11$ |
| 3 | 2.9999999995920965 | 209408.0 | 301568.0 | $4.0790348876384996e-10$ |
| 4 | 3.9999999837375317 | $3.106816e6$ | $2.844672e6$ | $1.626246826091915e-8$ |
| 5 | 5.000000665769791 | $2.4114688e7$ | $2.3346688e7$ | $6.657697912970661e-7$ |
| 6 | 5.999989245824773 | $1.20152064e8$ | $1.1882496e8$ | $1.0754175226779239e-5$ |
| 7 | 7.000102002793008 | $4.80398336e8$ | $4.78290944e8$ | 0.00010200279300764947 |
| 8 | 7.999355829607762 | $1.682691072e9$ | $1.67849728e9$ | 0.0006441703922384079 |
| 9 | 9.002915294362053 | $4.465326592e9$ | $4.457859584e9$ | 0.002915294362052734 |
| 10 | 9.990413042481725 | $1.2707126784e10$ | $1.2696907264e10$ | 0.009586957518274986 |
| 11 | 11.025022932909318 | $3.5759895552e10$ | $3.5743469056e10$ | 0.025022932909317674 |
| 12 | 11.953283253846857 | $7.216771584e10$ | $7.2146650624e10$ | 0.04671674615314281 |
| 13 | 13.07431403244734 | $2.15723629056e11$ | $2.15696330752e11$ | 0.07431403244734014 |
| 14 | 13.914755591802127 | $3.65383250944e11$ | $3.653447936e11$ | 0.08524440819787316 |
| 15 | 15.075493799699476 | $6.13987753472e11$ | $6.13938415616e11$ | 0.07549379969947623 |
| 16 | 15.946286716607972 | $1.555027751936e12$ | $1.554961097216e12$ | 0.05371328339202819 |
| 17 | 17.025427146237412 | $3.777623778304e12$ | $3.777532946944e12$ | 0.025427146237412046 |
| 18 | 17.99092135271648 | $7.199554861056e12$ | $7.1994474752e12$ | 0.009078647283519814 |
| 19 | 19.00190981829944 | $1.0278376162816e13$ | $1.0278235656704e13$ | 0.0019098182994383706 |
| 20 | 19.999809291236637 | $2.7462952745472e13$ | $2.7462788907008e13$ | 0.00019070876336257925 |

Tabela 4: Wyniki dla wielomianu $P$.

| $k$ | $z_k$ | $\lvert P(z_k)\rvert$ | | $\lvert p(z_k)\rvert$ | |
|---|---|---|---|---|---|
| 1 | 0.9999999999998357 + 0.0im | 2.0992 | $\times 10^4$ | 2.2016 | $\times 10^4$ |
| 2 | 2.0000000000550373 + 0.0im | 3.491 84 | $\times 10^5$ | 3.655 68 | $\times 10^5$ |
| 3 | 2.99999999660342 + 0.0im | 2.221 568 | $\times 10^6$ | 2.295 296 | $\times 10^6$ |
| 4 | 4.000000089724362 + 0.0im | 1.046 784 | $\times 10^7$ | 1.072 998 4 | $\times 10^7$ |
| 5 | 4.99999857388791 + 0.0im | 3.946 393 6 | $\times 10^7$ | 4.330 393 6 | $\times 10^7$ |
| 6 | 6.000020476673031 + 0.0im | 1.291 484 16 | $\times 10^8$ | 2.061 204 48 | $\times 10^8$ |
| 7 | 6.99960207042242 + 0.0im | 3.881 231 36 | $\times 10^8$ | 1.757 670 912 | $\times 10^9$ |
| 8 | 8.007772029099446 + 0.0im | 1.072 547 328 | $\times 10^9$ | 1.852 548 659 2 | $\times 10^{10}$ |
| 9 | 8.915816367932559 + 0.0im | 3.065 575 424 | $\times 10^9$ | 1.371 743 170 56 | $\times 10^{11}$ |
| 10 | 10.095455630535774 - 0.6449328236240688im | 7.143 113 638 035 824 | $\times 10^9$ | 1.491 263 381 675 401 9 | $\times 10^{12}$ |
| 11 | 10.095455630535774 + 0.6449328236240688im | 7.143 113 638 035 824 | $\times 10^9$ | 1.491 263 381 675 401 9 | $\times 10^{12}$ |
| 12 | 11.793890586174369 - 1.6524771364075785im | 3.357 756 113 171 857 | $\times 10^{10}$ | 3.296 021 414 130 166 4 | $\times 10^{13}$ |
| 13 | 11.793890586174369 + 1.6524771364075785im | 3.357 756 113 171 857 | $\times 10^{10}$ | 3.296 021 414 130 166 4 | $\times 10^{13}$ |
| 14 | 13.992406684487216 - 2.5188244257108443im | 1.061 206 453 308 197 6 | $\times 10^{11}$ | 9.545 941 595 183 662 | $\times 10^{14}$ |
| 15 | 13.992406684487216 + 2.5188244257108443im | 1.061 206 453 308 197 6 | $\times 10^{11}$ | 9.545 941 595 183 662 | $\times 10^{14}$ |
| 16 | 16.73074487979267 - 2.812624896721978im | 3.315 103 475 981 763 | $\times 10^{11}$ | 2.742 089 401 676 406 4 | $\times 10^{16}$ |
| 17 | 16.73074487979267 + 2.812624896721978im | 3.315 103 475 981 763 | $\times 10^{11}$ | 2.742 089 401 676 406 4 | $\times 10^{16}$ |
| 18 | 19.5024423688181 - 1.940331978642903im | 9.539 424 609 817 828 | $\times 10^{12}$ | 4.252 502 487 993 469 4 | $\times 10^{17}$ |
| 19 | 19.5024423688181 + 1.940331978642903im | 9.539 424 609 817 828 | $\times 10^{12}$ | 4.252 502 487 993 469 4 | $\times 10^{17}$ |
| 20 | 20.84691021519479 + 0.0im | 1.114 453 504 512 | $\times 10^{13}$ | 1.374 373 319 724 971 3 | $\times 10^{18}$ |

Tabela 5: Wyniki dla wielomianu $P$ ze zmienionym współczynnikiem przy $x^{19}$.

## 4.4 Wnioski

Porównując wyniki można zauważyć, że nawet niewielkie zakłócenie danych generuje duże zmiany w wynikach. Obliczenie pierwiastków Wilkinsona jest przykładem źle uwarunkowanego zadania. Ponadto widać, że błędy szybko się zwiększają, niewielkie błędy są mnożone przez duże czynniki. Wpływ ma także precyzja arytmetyki, która nie pozwala na dokładne zapisanie współczynników. Przy niewielkim zaburzeniu danych w wynikach pojawiły się pierwiastki zespolone mimo, że wielomian ma jedynie miejsca zerowe rzeczywiste.

| $k$ | $\lvert z_k - k\rvert$ | |
|---|---|---|
| 1 | 1.643 130 076 445 231 7 | $\times 10^{-13}$ |
| 2 | 5.503 730 804 434 781 | $\times 10^{-11}$ |
| 3 | 3.396 579 906 222 996 2 | $\times 10^{-9}$ |
| 4 | 8.972 436 216 225 788 | $\times 10^{-8}$ |
| 5 | 1.426 112 089 752 962 2 | $\times 10^{-6}$ |
| 6 | 2.047 667 303 095 579 4 | $\times 10^{-5}$ |
| 7 | 0.000 397 929 577 579 780 87 | |
| 8 | 0.007 772 029 099 445 632 | |
| 9 | 0.084 183 632 067 441 4 | |
| 10 | 0.651 958 683 038 040 6 | |
| 11 | 1.110 918 027 271 656 1 | |
| 12 | 1.665 281 290 598 479 | |
| 13 | 2.045 820 276 678 428 | |
| 14 | 2.518 835 871 190 904 5 | |
| 15 | 2.712 880 531 284 709 7 | |
| 16 | 2.906 001 873 537 510 6 | |
| 17 | 2.825 483 521 349 608 | |
| 18 | 2.454 021 446 312 976 | |
| 19 | 2.004 329 444 309 949 | |
| 20 | 0.846 910 215 194 789 4 | |

Tabela 6: Wyniki $|z_k - k|$ dla wielomianu $P$ ze zmienionym współczynnikiem przy $x^{19}$.

# 5 Zadanie 5

## 5.1 Opis problemu

Zadanie polega na zbadaniu modelu wzrostu populacji dla danego modelu logistycznego:
$p_{n+1} := p_n + rp_n(1 - p_n),$ dla $n = 0, 1, \ldots,$
gdzie $r$ jest pewną daną stałą, $r(1-p_n)$ jest czynnikiem wzrostu populacji, a $p_0$ jest wielkością populacji stanowiącą procent maksymalnej wielkości populacji dla danego stanu środowiska.

Należy wykonać eksperymenty:

1. W arytmetyce `Float32` dla $p_0 = 0.01$ i $r = 3$ wykonać 40 iteracji. Następnie ponownie wykonać 40 iteracji wyrażenia, ale po 10 iteracjach, zatrzymać, obciąć wynik odrzucając cyfry po trzecim miejscu po przecinku (daje to liczbę 0.722) i kontynuować dalej obliczenia (do 40-stej iteracji) tak, jak gdyby był to ostatni wynik na wyjściu. Należy porównać wyniki obu iteracji.

2. Wykonać 40 iteracji wyrażenia dla danych $p_0 = 0.01$ i $r = 3$ w arytmetyce `Float32` i `Float64`. Należy porównać wyniki iteracji dla obu arytmetyk.

## 5.2 Rozwiązanie

Zaimplementowanie podanej funkcji dla danych parametrów i wypisanie wyników poszczególnych iteracji.

## 5.3 Wyniki

| nr iteracji | bez modyfikacji | z modyfikacją |
|---|---|---|
| 1 | 0.0397 | 0.0397 |
| 2 | 0.15407173 | 0.15407173 |
| 3 | 0.5450726 | 0.5450726 |
| 4 | 1.2889781 | 1.2889781 |
| 5 | 0.1715188 | 0.1715188 |
| 6 | 0.5978191 | 0.5978191 |
| 7 | 1.3191134 | 1.3191134 |
| 8 | 0.056273222 | 0.056273222 |
| 9 | 0.21559286 | 0.21559286 |
| 10 | 0.7229306 | 0.722 |
| 15 | 1.2704837 | 1.2572169 |
| 16 | 0.2395482 | 0.28708452 |
| 17 | 0.7860428 | 0.9010855 |
| 18 | 1.2905813 | 1.1684768 |
| 19 | 0.16552472 | 0.577893 |
| 20 | 0.5799036 | 1.3096911 |
| 25 | 1.0070806 | 1.0929108 |
| 26 | 0.9856885 | 0.7882812 |
| 27 | 1.0280086 | 1.2889631 |
| 28 | 0.9416294 | 0.17157483 |
| 29 | 1.1065198 | 0.59798557 |
| 30 | 0.7529209 | 1.3191822 |
| 31 | 1.3110139 | 0.05600393 |
| 32 | 0.0877831 | 0.21460639 |
| 33 | 0.3280148 | 0.7202578 |
| 34 | 0.9892781 | 1.3247173 |
| 35 | 1.021099 | 0.034241438 |
| 36 | 0.95646656 | 0.13344833 |
| 37 | 1.0813814 | 0.48036796 |
| 38 | 0.81736827 | 1.2292118 |
| 39 | 1.2652004 | 0.3839622 |
| 40 | 0.25860548 | 1.093568 |

Tabela 7: Wyniki pierwszego eksperymentu dla `Float32`.

## 5.4 Wnioski

Model logistyczny przedstawiony w zadaniu pokazuje, że drobne błędy spowodowane np. niedokładnym przechowywaniem wyniku kumulują się. Wyniki różnią się w przypadku zaokrąglenia jednego wyniku w pierwszym eksperymencie, ale także w przypadku dwóch różnych precyzji arytmetyk. Jest to zjawisko chaosu w deterministycznych układach sprzężonych, które polega na niemożności przewidywania wyników.

| nr iteracji | Float32 | Float64 |
|---|---|---|
| 1 | 0.0397 | 0.0397 |
| 2 | 0.15407173 | 0.15407173000000002 |
| 3 | 0.5450726 | 0.5450726260444213 |
| 4 | 1.2889781 | 1.2889780011888006 |
| 5 | 0.1715188 | 0.17151914210917552 |
| 6 | 0.5978191 | 0.5978201201070994 |
| 7 | 1.3191134 | 1.3191137924137974 |
| 8 | 0.056273222 | 0.056271577646256565 |
| 9 | 0.21559286 | 0.21558683923263022 |
| 10 | 0.7229306 | 0.722914301179573 |
| 11 | 1.3238364 | 1.3238419441684408 |
| 12 | 0.037716985 | 0.03769529725473175 |
| 13 | 0.14660022 | 0.14651838271355924 |
| 20 | 0.5799036 | 0.5965293124946907 |
| 21 | 1.3107498 | 1.3185755879825978 |
| 22 | 0.088804245 | 0.058377608259430724 |
| 29 | 1.1065198 | 1.2321124623871897 |
| 30 | 0.7529209 | 0.37414648963928676 |
| 31 | 1.3110139 | 1.0766291714289444 |
| 32 | 0.0877831 | 0.8291255674004515 |
| 33 | 0.3280148 | 1.2541546500504441 |
| 36 | 0.95646656 | 1.1325322626697856 |
| 37 | 1.0813814 | 0.6822410727153098 |
| 38 | 0.81736827 | 1.3326056469620293 |
| 39 | 1.2652004 | 0.0029091569028512065 |
| 40 | 0.25860548 | 0.011611238029748606 |

Tabela 8: Wyniki drugiego eksperymentu dla `Float32` i `Float64`.

# 6 Zadanie 6

## 6.1 Opis problemu

Zadanie polegało na zbadaniu zachowania równania rekurencyjnego

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

gdzie $c$ jest pewną daną stałą, dla następujących danych:

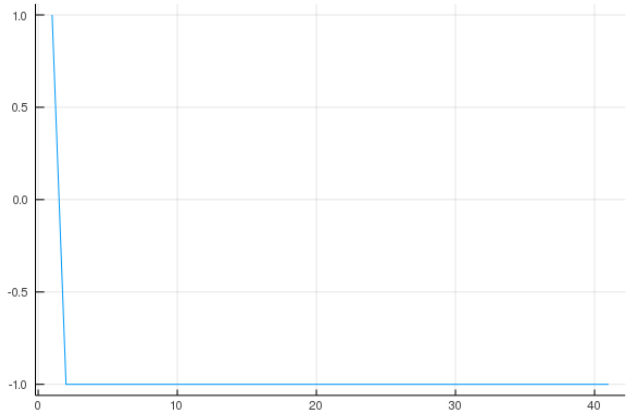
1. $c = -2$ i $x_0 = 1$
2. $c = -2$ i $x_0 = 2$
3. $c = -2$ i $x_0 = 1.99999999999999$
4. $c = -1$ i $x_0 = 1$
5. $c = -1$ i $x_0 = -1$
6. $c = -1$ i $x_0 = 0.75$
7. $c = -1$ i $x_0 = 0.25$

Należy wykonać 40 iteracji wyrażenia (1) w arytmetyce `Float64` i zaobserwować zachowanie generowanych ciągów. Następnie przeprowadzić iterację graficzną (1).

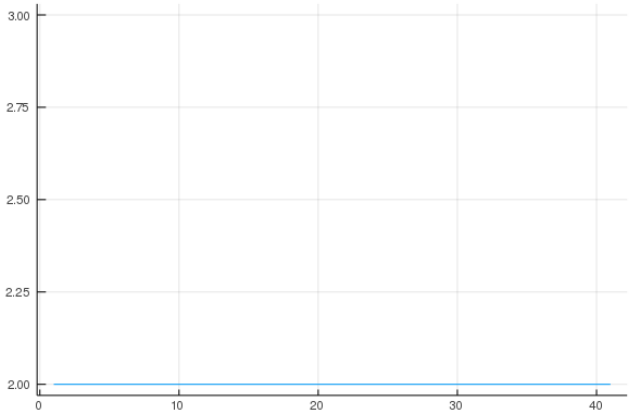
## 6.2 Rozwiązanie

Zaimplementowanie podanej funkcji dla danych parametrów i wypisanie wyników poszczególnych iteracji.

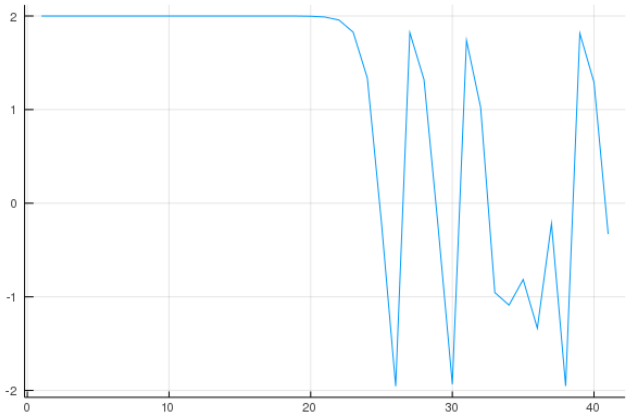
## 6.3 Wyniki



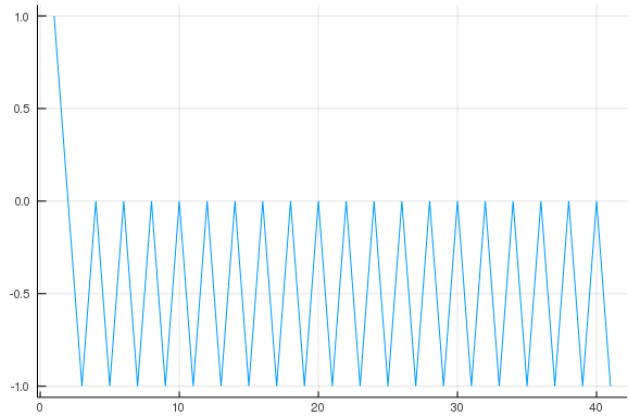

Rysunek 3: $c = -2$, $x_0 = 1$.




Rysunek 4: $c = -2$, $x_0 = 2$.



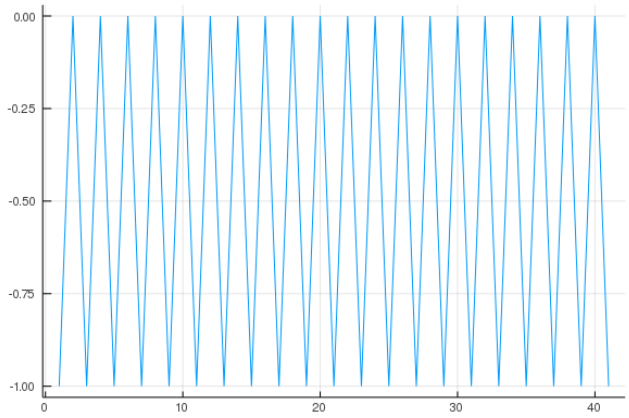

Rysunek 5: $c = -2$, $x_0 = 1.99999999999999$.



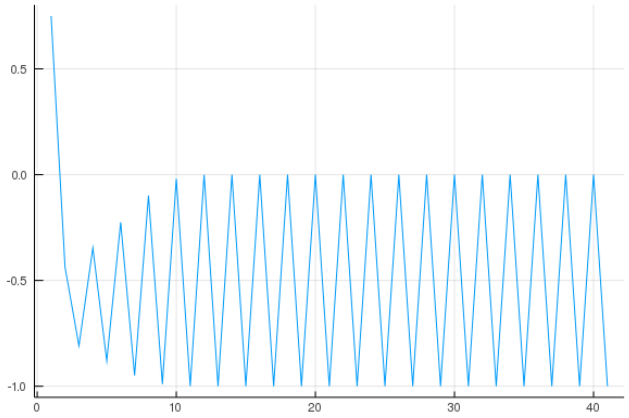

Rysunek 6: $c = -1$, $x_0 = 1$.

## 6.4 Wnioski

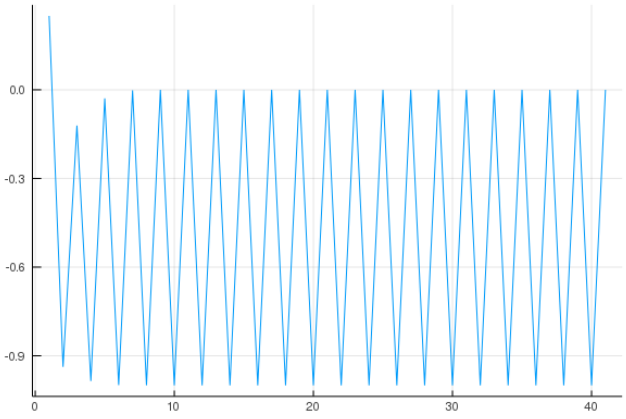
Eksperyment jest przykładem zjawiska chaosu. Można zauważyć, że dla $c = -2$ i $x_0 = 1$ oraz $x_0 = 2$ funkcja $x_{n+1} = x_n^2 + c$ zachowuje się stabilnie. Jednak już dla $c = -2$ i $x_0 = 1.9999999999999$ funkcja staje się nieuporządkowana. Jeżeli zmienimy $c = -1$ proces po pewnej liczbie iteracji ustala się i powtarzają się tylko wartości 0 i $-1$. Układ jest w stanie stabilnym.




Rysunek 7: $c = -1$, $x_0 = -1$.




Rysunek 8: $c = -1$, $x_0 = 0.75$.




Rysunek 9: $c = -1$, $x_0 = 0.25$.

| nr | $c=-2$ | | | $c=-1$ | | | |
|---|---|---|---|---|---|---|---|
| | $x_0=1$ | $x_0=2$ | $x_0=1.99999999999999$ | $x_0=1$ | $x_0=-1$ | $x_0=0.75$ | $x_0=0.25$ |
| 1 | -1.0 | 2.0 | 1.99999999999996 | 0.0 | 0.0 | -0.4375 | -0.9375 |
| 2 | -1.0 | 2.0 | 1.9999999999998401 | -1.0 | -1.0 | -0.80859375 | -0.12109375 |
| 3 | -1.0 | 2.0 | 1.9999999999993605 | 0.0 | 0.0 | -0.3461761474609375 | -0.9853363037109375 |
| 4 | -1.0 | 2.0 | 1.999999999997442 | -1.0 | -1.0 | -0.8801620749291033 | -0.029112368589267135 |
| 5 | -1.0 | 2.0 | 1.9999999999897682 | 0.0 | 0.0 | -0.2253147218564956 | -0.9991524699951226 |
| 6 | -1.0 | 2.0 | 1.9999999999590727 | -1.0 | -1.0 | -0.9492332761147301 | -0.0016943417026455965 |
| 7 | -1.0 | 2.0 | 1.999999999836291 | 0.0 | 0.0 | -0.0989561875164966 | -0.9999971292061947 |
| 8 | -1.0 | 2.0 | 1.9999999993451638 | -1.0 | -1.0 | -0.9902076729521999 | -5.741579369278327e-6 |
| 9 | -1.0 | 2.0 | 1.9999999973806553 | 0.0 | 0.0 | -0.01948876442658909 | -0.9999999999670343 |
| 10 | -1.0 | 2.0 | 1.999999989522621 | -1.0 | -1.0 | -0.999620188061125 | -6.593148249578462e-11 |
| 11 | -1.0 | 2.0 | 1.9999999580904841 | 0.0 | 0.0 | -0.0007594796206411569 | -1.0 |
| 12 | -1.0 | 2.0 | 1.9999998323619383 | -1.0 | -1.0 | -0.9999994231907058 | 0.0 |
| 13 | -1.0 | 2.0 | 1.9999993294477814 | 0.0 | 0.0 | -1.1536182557003727e-6 | -1.0 |
| 14 | -1.0 | 2.0 | 1.9999973177915749 | -1.0 | -1.0 | -0.9999999999986692 | 0.0 |
| 15 | -1.0 | 2.0 | 1.9999892711734937 | 0.0 | 0.0 | -2.6616486792363503e-12 | -1.0 |
| 16 | -1.0 | 2.0 | 1.9999570848090826 | -1.0 | -1.0 | -1.0 | 0.0 |
| 17 | -1.0 | 2.0 | 1.999828341078044 | 0.0 | 0.0 | 0.0 | -1.0 |
| 18 | -1.0 | 2.0 | 1.9993133937789613 | -1.0 | -1.0 | -1.0 | 0.0 |
| 19 | -1.0 | 2.0 | 1.9972540465439481 | 0.0 | 0.0 | 0.0 | -1.0 |
| 20 | -1.0 | 2.0 | 1.9890237264361752 | -1.0 | -1.0 | -1.0 | 0.0 |
| 21 | -1.0 | 2.0 | 1.9562153843260486 | 0.0 | 0.0 | 0.0 | -1.0 |
| 22 | -1.0 | 2.0 | 1.82677862987391 | -1.0 | -1.0 | -1.0 | 0.0 |
| 23 | -1.0 | 2.0 | 1.3371201625639997 | 0.0 | 0.0 | 0.0 | -1.0 |
| 24 | -1.0 | 2.0 | -0.21210967086482313 | -1.0 | -1.0 | -1.0 | 0.0 |
| 25 | -1.0 | 2.0 | -1.9550094875256163 | 0.0 | 0.0 | 0.0 | -1.0 |
| 26 | -1.0 | 2.0 | 1.822062096315173 | -1.0 | -1.0 | -1.0 | 0.0 |
| 27 | -1.0 | 2.0 | 1.319910282828443 | 0.0 | 0.0 | 0.0 | -1.0 |
| 28 | -1.0 | 2.0 | -0.2578368452837396 | -1.0 | -1.0 | -1.0 | 0.0 |
| 29 | -1.0 | 2.0 | -1.9335201612141288 | 0.0 | 0.0 | 0.0 | -1.0 |
| 30 | -1.0 | 2.0 | 1.7385002138215109 | -1.0 | -1.0 | -1.0 | 0.0 |
| 31 | -1.0 | 2.0 | 1.0223829934574389 | 0.0 | 0.0 | 0.0 | -1.0 |
| 32 | -1.0 | 2.0 | -0.9547330146890065 | -1.0 | -1.0 | -1.0 | 0.0 |
| 33 | -1.0 | 2.0 | -1.0884848706628412 | 0.0 | 0.0 | 0.0 | -1.0 |
| 34 | -1.0 | 2.0 | -0.8152006863380978 | -1.0 | -1.0 | -1.0 | 0.0 |
| 35 | -1.0 | 2.0 | -1.3354478409938944 | 0.0 | 0.0 | 0.0 | -1.0 |
| 36 | -1.0 | 2.0 | -0.21657906398474625 | -1.0 | -1.0 | -1.0 | 0.0 |
| 37 | -1.0 | 2.0 | -1.953093509043491 | 0.0 | 0.0 | 0.0 | -1.0 |
| 38 | -1.0 | 2.0 | 1.8145742550678174 | -1.0 | -1.0 | -1.0 | 0.0 |
| 39 | -1.0 | 2.0 | 1.2926797271549244 | 0.0 | 0.0 | 0.0 | -1.0 |
| 40 | -1.0 | 2.0 | -0.3289791230026702 | -1.0 | -1.0 | -1.0 | 0.0 |

Tabela 9: Iteracje funkcji $x_{n+1} := x_n^2 + c$.