Miriam Jańczak
numer albumu: 229761

12 listopada 2017
prowadzący: dr hab. Paweł Zieliński

Obliczenia naukowe

Lista 2

1 Iloczyn skalarny raz jeszcze

1.1 Opis problemu

Ponowne obliczenie iloczynu skalarnego z zadania 5 z listy 1 dla typów `Float32` i `Float64` po wprowadzeniu niewielkich zmian danych w wektorze x, tj. obcięciu ostatniej cyfry znaczącej współrzędnych x_4 i x_5 (9 dla x_4, 7 dla x_5). W tym celu wykorzystano te same cztery algorytmy co w zadaniu z listy 1. Poniżej przedstawiono zmienione dane wejściowe:

$$\tilde{x} = [2.718281828, -3.141592654, 1.414213562, 0.577215664, 0.301029995]$$
$$y = [1486.2497, 878366.9879, -22.37492, 4773714.647, 0.000185049]$$

1.2 Rozwiązanie

Iloczyn skalarny dla typów `Float32` i `Float64` obliczono za pomocą czterech algorytmów z programu z listy 1:

(a) *w przód*,
(b) *w tył*,
(c) *od największego do najmniejszego*,
(d) *od najmniejszego do największego*.

1.3 Wyniki

Zestawienie wyników iloczynów skalarnych $x \cdot y$ (zadanie 5 z listy 1) oraz $\tilde{x} \cdot y$ przedstawia Tabela 1.

	`Float32`		`Float64`	
Algorytm	$x \cdot y$	$\tilde{x} \cdot y$	$x \cdot y$	$\tilde{x} \cdot y$
(a)	$-0.499\,944\,3$	$-0.499\,944\,3$	$1.025\,188\,136\,829\,667\,2 \cdot 10^{-10}$	$-0.004\,296\,342\,739\,891\,585$
(b)	$-0.454\,345\,7$	$-0.454\,345\,7$	$-1.564\,330\,887\,049\,436\,6 \cdot 10^{-10}$	$-0.004\,296\,342\,998\,713\,953$
(c)	-0.5	-0.5	0.0	$-0.004\,296\,342\,842\,280\,865$
(d)	-0.5	-0.5	0.0	$-0.004\,296\,342\,842\,280\,865$

Tabela 1: Iloczyny skalarne $x \cdot y$ oraz $\tilde{x} \cdot y$ obliczone w arytmetykach `Float32` i `Float64` przy użyciu danych algorytmów.

1.4 Wnioski

Przeprowadzając analizę danych z tabeli (1) można łatwo zauważyć, że w arytmetyce `Float32` zmiany dokonane w wektorze x nie miały wpływu na wyniki (pozostały one takie same). Jest to spowodowane stosunkowo niewielką precyzją tej arytmetyki, a co za tym idzie pewną niedokładnością w przechowywaniu poszczególnych składowych wektora. Ta niedokładność okazała się tutaj na tyle duża, że usunięcie 9 z x_4 nie zmieniło w żaden sposób zapisu wartości w arytmetyce `Float32`, a zmiana w x_5 zaważyła jedynie na najmniej znaczącym bicie zapisu liczby. Nie należy się zatem dziwić że wyniki nie uległy zmianie, jednak to powinno raczej martwić niż cieszyć, gdyż są one bardzo dalekie od poprawnych. Zgoła odmienna sytuacja miała miejsce w arytmetyce `Float64`. Tutaj wprowadzenie niewielkich zmian rzutowało bardzo mocno na wyniki obliczeń iloczynu skalarnego. Pierwszą rzeczą jaką można zauważyć jest to, że dane przy operacji obliczania iloczynu skalarnego są bardzo wrażliwe na zmiany. Mogłoby się wydawać że dokonanie modyfikacji rzędu 10^{-9} nie wpłynie znacząco na otrzymane wyniki, te jednak bardzo się zmieniły. Z drugiej strony, można zauważyć, że rozbieżności pomiędzy iloczynami skalarnymi uzyskanymi z użyciem poszczególnych algorytmów znacznie się zmniejszyły, a nawet powiedzieć że w granicach pewnego dopuszczalnego błędu są takie same, co pozwala z kolei wnioskować, że wynik iloczynu skalarnego w arytmetyce `Float64` przy zmienionych danych jest w gruncie rzeczy wynikiem poprawnym. Usunięcie ostatnich cyfr po przecinku z x_4 i x_5 sprawiło że wektory zostały dokładniej zapisane (zmniejszyła się także ich ortogonalność) i arytmetyka okazała się wystarczająca do obliczenia iloczynu skalarnego. Widać zatem wyraźnie, że precyzja arytmetyki odgrywa kluczową rolę wtedy, kiedy mamy do czynienia ze źle uwarunkowanym zadaniem, tj. wtedy kiedy niewielkie względne zmiany danych powodują duże względne odkształcenia wyników. Do rozwiązywania takich zadań należy zatem używać maksymalnej precyzji.

2 Nietypowa granica

2.1 Opis problemu

Porównanie wykresów funkcji

$$f(x) = e^x \ln(1 + e^{-x}) \tag{1}$$

(narysowanych w co najmniej dwóch programach do wizualizacji) z jej granicą $\lim\limits_{x\to\infty}$, a następnie wyjaśnienie zaistniałego zjawiska.

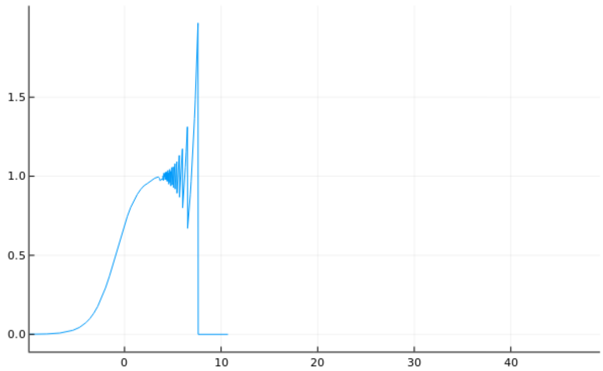
2.2 Rozwiązanie

Wykresy funkcji (1) wykonano za pomocą biblioteki *Plotly* (używając różnych typów zmiennopozycyjnych) w języku *Julia*, a także za pomocą pakietu matematycznego *Wolfram Alpha*. Granicę $\lim\limits_{x\to\infty}$ funkcji (1) wyliczono za pomocą biblioteki *SymPy* w języku *Julia* oraz w sposób analityczny.

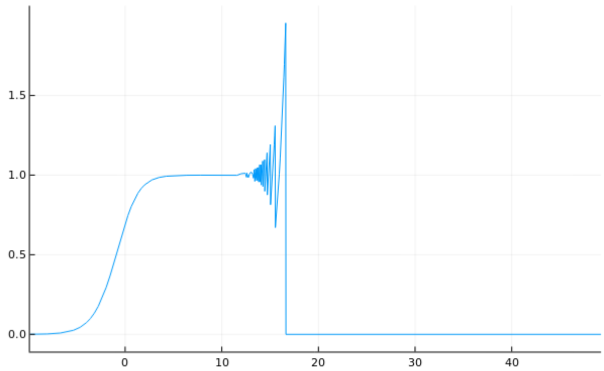
2.3 Wyniki

Poniżej przedstawiono (Rysunek 1, Rysunek 2) otrzymane wykresy funkcji (1).
Obliczona przez program granica $\lim\limits_{x\to\infty}$ funkcji (1) wynosi 1. Taki sam wynik został uzyskany przez rozwiązanie analityczne:

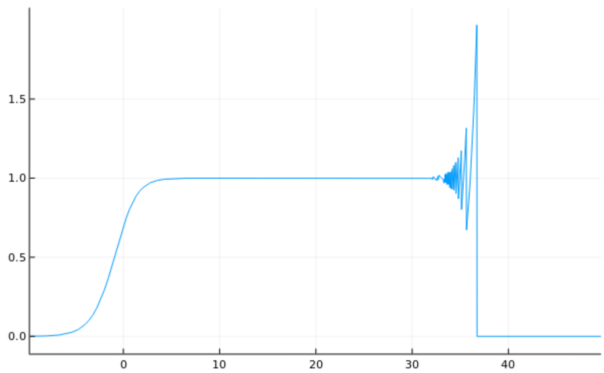
$$\lim_{x\to\infty} f(x) = \lim_{x\to\infty} e^x \ln(1 + e^{-x}) = \lim_{x\to\infty} \frac{\ln(1+e^{-x})}{e^{-x}} = \left[\frac{0}{0}\right] \overset{H}{=} \lim_{x\to\infty} \frac{(\ln(1+e^{-x}))'}{(e^{-x})'} =$$
$$= \lim_{x\to\infty} \frac{\frac{1}{1+e^{-x}} \cdot -e^{-x}}{-e^{-x}} = \lim_{x\to\infty} \frac{1}{1+e^{-x}} = 1.$$



(a) `Float16`



(b) `Float32`

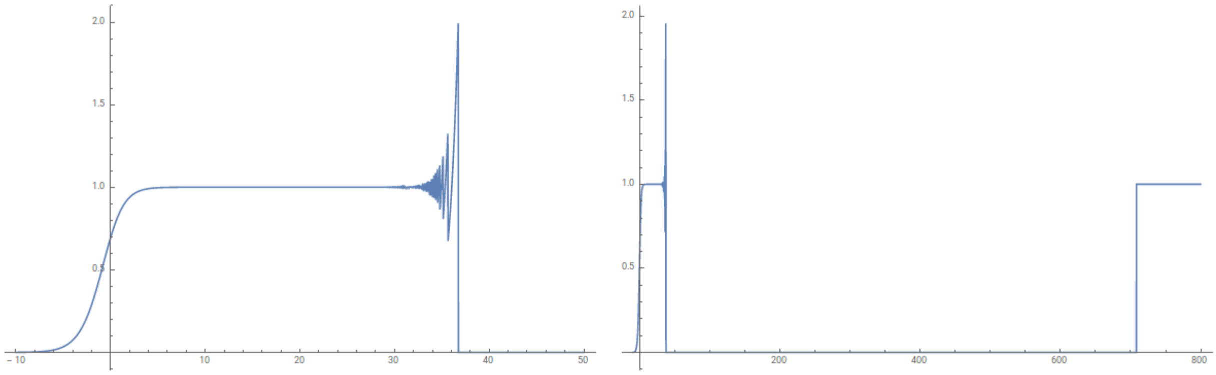


(c) `Float64`

Rysunek 1: Wykresy wykonane za pomocą biblioteki *Plotly*

2.4 Wnioski

Z analitycznego punktu widzenia oczywistym jest, że dla x dążącego do nieskończoności granica funkcji (1) jest równa jeden. Dużo trudniej natomiast podobnego wyniku dopatrzyć się na wykresach. Na każdym z nich pojawia się bowiem nietypowe zaburzenie, oscylacja niewłaściwa dla funkcji, po której wartości funkcji są równe zero. Na wykresach które przedstawia Rysunek 1, można zaobserwować, że moment pojawienia się owej oscylacji zależny jest od precyzji arytmetyki. Obserwowane zaburzenia wywołane są poprzez dodanie bardzo małej wartości e^{-x} do, w stosunku do niej dużej, jedynki przez co następuje utrata cyfr znaczących. Kluczowe jest jednak pomnożenie logarytmu tak przybliżonej wartości przez bardzo duże e^{x}, co potęguje względnie niewielki początkowy błąd i przez co można zaobserwować niepokojące zmiany na wykresie. Łatwo można wnioskować dalej że w momencie kiedy $1 + e^{-x} = 1$ (e^{-x} zostaje całkowicie pochłonięte przez 1), to $\ln(1 + e^{-x}) = 0$, co tłumaczy pojawienie się wartości 0 na wykresie. W wielu programach wykres funkcji w pewnym momencie się kończy. Wynika to z przepełnienia (*overflow*) dla e^{x}, wtedy pojawia się mnożenie $\infty \cdot 0$ co przyjmuje wartość **NaN**. Na wykresie 2b w momencie wystąpienia *overflow* zaobserwować można „oszustwo” jakiego dokonuje program *Wolfram Alpha* podstawiając po prostu wyliczoną analitycznie granicę równą jeden. W tym zadaniu przedstawiona jest sytuacja gdzie małe zmiany danych spowodowały duże odchylenia co pozwala stwierdzić że jest ono źle uwarunkowane.



(a) mniejszy zakres **(b)** większy zakres

Rysunek 2: Wykresy wykonane za pomocą programu *Wolfram Alpha*

3 Układ równań i wskaźnik uwarunkowania

3.1 Opis problemu

Rozwiązanie układu równań liniowych $\mathbf{Ax} = \mathbf{b}$ dla danej macierzy współczynników $\mathbf{A} \in \mathbb{R}^{n \cdot n}$ i wektora prawych stron $\mathbf{b} \in \mathbb{R}^n$.
Macierz $\mathbf{A}$ zadana była w następujący sposób:

(a) *macierz Hilberta* $\mathbf{H}_n$ stopnia n,

(b) *macierz losowa* $\mathbf{R}_n^c$ stopnia n o danym wskaźniku uwarunkowania c.

Wektor $\mathbf{b}$ natomiast jako $\mathbf{b} = \mathbf{Ax}$, gdzie $\mathbf{A}$ jest wygenerowaną macierzą, a $\mathbf{x} = (1, \ldots, 1)^T$, tak aby było znane dokładne rozwiązanie dla $\mathbf{A}$ i $\mathbf{b}$.
Układ równań $\mathbf{Ax} = \mathbf{b}$ należało rozwiązać za pomocą dwóch algorytmów:

(i) *metodą eliminacji Gaussa*: $\mathbf{x} = \mathbf{A} \backslash \mathbf{b}$,

(ii) *metodą macierzy odwrotnej*: $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$.

Obliczone rozwiązania $\tilde{\mathbf{x}}$ dla różnych macierzy wejściowych należało porównać z rozwiązaniem dokładnym $\mathbf{x} = (1, \ldots, 1)^T$ oraz obliczyć błędy względne.

3.2 Rozwiązanie

Macierz Hilberta $\mathbf{H}_n$ z rosnącym stopniem $n > 1$ wygenerowano za pomocą funkcji *hilb*(n), natomiast losową macierz $\mathbf{R}_n^c$, $n = 5, 10, 20$, z rosnącym wskaźnikiem uwarunkowania $c = 1, 10, 10^3, 10^7, 10^{12}, 10^{16}$ stworzono przy użyciu funkcji *matcond*(n, c). Dla każdej wygenerowanej macierzy rozwiązano układ równań metodą eliminacji Gaussa i macierzy odwrotnej, a także policzono błędy względne $\dfrac{||x - \tilde{x}||}{||x||}$ obu tych metod, wskaźniki uwarunkowania i rzędy macierzy.

3.3 Wyniki

Otrzymane wyniki dla macierzy Hilberta $\mathbf{H}_n$ prezentuje Tabela 2, natomiast dla macierzy losowej $\mathbf{R}_n^c$ Tabela 3, metoda eliminacji Gaussa jest oznaczona przez `GE`, a metoda macierzy odwrotnej przez `INV`.

Macierz Hilberta $\mathbf{H}_n$				
Rozmiar	Rząd	`COND`	Błędy względne	
			`GE`	`INV`
1x1	1	1.0	0.0	0.0
2x2	2	$1.928\,147\,006\,790\,397 \cdot 10^{1}$	$5.661\,048\,867\,003\,676 \cdot 10^{-16}$	$1.124\,015\,143\,811\,696 \cdot 10^{-15}$
3x3	3	$5.240\,567\,775\,860\,644 \cdot 10^{2}$	$8.022\,593\,772\,267\,726 \cdot 10^{-15}$	$9.825\,526\,038\,180\,824 \cdot 10^{-15}$
4x4	4	$1.551\,373\,873\,892\,924 \cdot 10^{4}$	$4.451\,545\,960\,181\,209 \cdot 10^{-13}$	$2.950\,477\,637\,286\,781 \cdot 10^{-13}$
5x5	5	$4.766\,072\,502\,425\,943 \cdot 10^{5}$	$1.682\,842\,629\,922\,719 \cdot 10^{-12}$	$8.500\,055\,777\,753\,297 \cdot 10^{-12}$
6x6	6	$1.495\,105\,864\,225\,467 \cdot 10^{7}$	$2.618\,913\,302\,311\,624 \cdot 10^{-10}$	$3.347\,413\,507\,036\,174 \cdot 10^{-10}$
7x7	7	$4.753\,673\,565\,831\,290 \cdot 10^{8}$	$1.260\,686\,722\,417\,155 \cdot 10^{-8}$	$5.163\,959\,183\,577\,243 \cdot 10^{-9}$
8x8	8	$1.525\,757\,553\,806\,004 \cdot 10^{10}$	$1.026\,543\,065\,687\,064 \cdot 10^{-7}$	$2.698\,715\,074\,276\,819 \cdot 10^{-7}$
9x9	9	$4.931\,537\,564\,468\,762 \cdot 10^{11}$	$4.832\,357\,120\,502\,150 \cdot 10^{-6}$	$9.175\,846\,868\,614\,517 \cdot 10^{-6}$
10x10	10	$1.602\,441\,699\,254\,171 \cdot 10^{13}$	$6.329\,153\,722\,983\,848 \cdot 10^{-4}$	$4.552\,142\,251\,740\,885 \cdot 10^{-4}$
11x11	11	$5.222\,677\,939\,280\,335 \cdot 10^{14}$	$1.154\,395\,859\,612\,211 \cdot 10^{-2}$	$8.044\,466\,773\,431\,160 \cdot 10^{-3}$
12x12	11	$1.751\,473\,190\,709\,146 \cdot 10^{16}$	$2.975\,640\,310\,734\,787 \cdot 10^{-1}$	$3.439\,293\,709\,120\,522 \cdot 10^{-1}$
13x13	11	$3.344\,143\,497\,338\,461 \cdot 10^{18}$	$2.375\,017\,867\,706\,776$	$5.585\,796\,893\,150\,773$
14x14	12	$6.200\,786\,263\,161\,444 \cdot 10^{17}$	$5.281\,004\,646\,755\,168$	$4.800\,641\,929\,017\,436$
15x15	12	$3.674\,392\,953\,467\,974 \cdot 10^{17}$	$1.177\,294\,734\,836\,712$	$4.827\,357\,721\,257\,648$
16x16	12	$7.865\,467\,778\,431\,645 \cdot 10^{17}$	$2.056\,465\,582\,380\,410 \cdot 10^{1}$	$3.173\,646\,749\,626\,613 \cdot 10^{1}$
17x17	12	$1.263\,684\,342\,666\,052 \cdot 10^{18}$	$1.774\,221\,463\,517\,907 \cdot 10^{1}$	$1.591\,033\,596\,260\,414 \cdot 10^{1}$
18x18	12	$2.244\,630\,992\,918\,913 \cdot 10^{18}$	$4.276\,456\,441\,115\,942$	$6.281\,223\,433\,472\,033$
19x19	13	$6.471\,953\,976\,541\,591 \cdot 10^{18}$	$2.211\,993\,729\,264\,891 \cdot 10^{1}$	$2.292\,561\,401\,563\,632 \cdot 10^{1}$
20x20	13	$1.355\,365\,790\,868\,823 \cdot 10^{18}$	$1.493\,006\,966\,929\,400 \cdot 10^{1}$	$2.153\,949\,860\,251\,383 \cdot 10^{1}$

Tabela 2: Wyniki obliczeń dla macierzy Hilberta $\mathbf{H}_n$

Macierz losowa $\mathbf{R}_n^c$				
Rozmiar	Rząd	`COND`	Błędy względne	
			`GE`	`INV`
5x5	5	1	$1.404\,333\,387\,430\,680 \cdot 10^{-16}$	$1.790\,180\,836\,524\,724 \cdot 10^{-16}$
5x5	5	10	0.0	$9.930\,136\,612\,989\,092 \cdot 10^{-17}$
5x5	5	1000	$6.467\,561\,325\,518\,618 \cdot 10^{-15}$	$6.138\,840\,652\,485\,208 \cdot 10^{-15}$
5x5	5	10^{7}	$2.932\,858\,554\,206\,356 \cdot 10^{-10}$	$2.541\,421\,917\,682\,778 \cdot 10^{-10}$
5x5	5	10^{12}	$2.431\,174\,605\,159\,248 \cdot 10^{-5}$	$2.445\,937\,707\,560\,239 \cdot 10^{-5}$
5x5	4	10^{16}	$9.228\,482\,506\,511\,224 \cdot 10^{-2}$	$1.358\,024\,596\,793\,109 \cdot 10^{-1}$
10x10	10	1	$2.328\,823\,463\,338\,184 \cdot 10^{-16}$	$2.302\,207\,463\,925\,367 \cdot 10^{-16}$
10x10	10	10	$5.324\,442\,579\,404\,919 \cdot 10^{-16}$	$5.916\,561\,726\,981\,507 \cdot 10^{-16}$
10x10	10	1000	$7.659\,734\,318\,226\,236 \cdot 10^{-16}$	$1.167\,815\,308\,046\,354 \cdot 10^{-14}$
10x10	10	10^{7}	$2.568\,414\,379\,855\,613 \cdot 10^{-10}$	$2.258\,463\,845\,088\,549 \cdot 10^{-10}$
10x10	10	10^{12}	$1.951\,994\,704\,671\,510 \cdot 10^{-5}$	$2.174\,813\,032\,517\,800 \cdot 10^{-5}$
10x10	9	10^{16}	$3.178\,399\,241\,163\,815 \cdot 10^{-1}$	$3.516\,778\,693\,816\,187 \cdot 10^{-1}$
20x20	20	1	$5.495\,323\,605\,393\,213 \cdot 10^{-16}$	$4.557\,326\,905\,135\,503 \cdot 10^{-16}$
20x20	20	10	$5.087\,681\,048\,627\,601 \cdot 10^{-16}$	$4.071\,658\,748\,137\,585 \cdot 10^{-16}$
20x20	20	1000	$5.808\,917\,732\,317\,164 \cdot 10^{-15}$	$3.747\,228\,857\,827\,342 \cdot 10^{-15}$
20x20	20	10^{7}	$1.511\,216\,720\,479\,130 \cdot 10^{-10}$	$1.241\,078\,754\,561\,024 \cdot 10^{-10}$
20x20	20	10^{12}	$4.740\,084\,259\,557\,948 \cdot 10^{-5}$	$4.819\,264\,617\,327\,459 \cdot 10^{-5}$
20x20	19	10^{16}	$8.613\,420\,159\,130\,484 \cdot 10^{-1}$	$8.466\,277\,602\,660\,599 \cdot 10^{-1}$

Tabela 3: Wyniki obliczeń dla macierzy losowej $\mathbf{R}_n^c$

3.4 Wnioski

W problemie rozwiązywania układu równań liniowych decydujący wpływ na wielkość błędu względnego ma wskaźnik uwarunkowania macierzy. Obserwując wielkości błędów dla macierzy losowej $\mathbf{R}_n^c$ (Tabela 3) zauważyć można że zgodnie z powyższym stwierdzeniem, że wielkości błędów zależą głównie od wskaźnika uwarunkowania (**cond**) i im większy wskaźnik uwarunkowania macierzy tym większy błąd względny. Problemy, w których pojawiają się macierze o dużym wskaźniku uwarunkowania, są źle uwarunkowane. Wyjątkowo kłopotliwe są zadania które sprowadzają się do obliczeń z macierzą Hilberta, ponieważ macierz ta jest bardzo źle uwarunkowana, co można zauważyć analizując otrzymane wartości „**cond**", które przedstawia Tabela 2. Wraz ze wzrostem rozmiaru macierzy rośnie wskaźnik uwarunkowania macierzy Hilberta. Można więc sobie wyobrazić, że zadanie w którym pojawia się taka macierz o dużym rozmiarze może być bardzo ciężko poprawnie rozwiązać. Z danych w tabeli można również wnioskować, że lepszym algorytmem w przypadku macierzy Hilberta jest eliminacja Gaussa (metoda macierzy odwrotnej nie jest zalecana z numerycznego punktu widzenia).

4 „Złośliwy wielomian" Wilkinsona

4.1 Opis problemu

Obliczenie dwudziestu zer wielomianu Wilkinsona p, tj. $p(x) = (x-20)(x-19)\dots(x-2)(x-1)$ w postaci naturalnej P i sprawdzenie otrzymanych pierwiastków z_k poprzez obliczenie $|P(z_k)|$, $|p(z_k)|$ i $|z_k-k|$ dla $1 \le x \le 20$. Powtórzenie eksperymentu Wilkinsona, tj. zmiana współczynnika -210 przy x^{19} na $-210 - 2^{-23}$ i wyjaśnienie zaistniałego zjawiska.

4.2 Rozwiązanie

Do rozwiązania zadania użyto pakietu `Polynomials`. Miejsca zerowe wielomianu P utworzonego z danych współczynników za pomocą funkcji `Poly` obliczono przy użyciu funkcji `roots`. Za pomocą funkcji `poly` stworzono natomiast wielomian p. Funkcja `polyval` posłużyła do obliczenia wartości wielomianów P i p w zadanych punktach. Obliczony został również błąd bezwzględny obliczonych pierwiastków wielomianu P. Podobne operacje zostały wykonane dla wielomianu P z zaburzonym współczynnikiem przy x^{19}.

4.3 Wyniki

Tabela 4 przedstawia obliczone pierwiastki wielomianu P oraz $|P(z_k)|$, $|p(z_k)|$ i $|z_k-k|$, natomiast Tabela 5 prezentuje te wartości dla wielomianu P z zaburzonym współczynnikiem.

4.4 Wnioski

W uproszczeniu można powiedzieć że zadanie polegało na policzeniu dwudziestu pierwiastków a następnie wykonaniu klasycznego „sprawdzenia" obliczając wartości funkcji w otrzymanych zerach. Tabela 4 pokazuje jednak że ta kontrola wyników się nie powiodła. Można by powiedzieć że pierwiastki wcale nie są źle obliczone, z otrzymanych danych wynika przecież, że odchylenia były niewielkie. Można by powiedzieć że odchylenie rzędu 10^{-13} jest bardzo niewielkie i zignorować taki błąd. Jednak przy zagadnieniu wielomianu Wilkinsona takie drobne odchylenie pierwiastka powoduje że zamiast oczekiwanego zera pojawia się około 20000, dla nieco większych odchyleń błędy rosną lawinowo i sięgają nawet bilionów. Dzieje się tak ponieważ w tym konkretnym wielomianie drobny błąd w wartości pierwiastka mnożony jest przez czynnik rzędu 19!. Wpływ na błędy przy obliczeniach miejsc zerowych mają współczynniki wielomianu P, które nie są dokładnie reprezentowane w arytmetyce `Float64`. Widać to wyraźniej w próbie powtórzenia eksperymentu Wilkinsona, gdzie celowo został zaburzony jeden ze współczynników. Tabela 5

k	$\lvert P(z_k)\rvert$	$\lvert p(z_k)\rvert$	$\lvert z_k - k\rvert$
1	$3.635\,200\,000\,000 \cdot 10^{4}$	$3.840\,000\,000\,000 \cdot 10^{4}$	$3.010\,924\,842\,783 \cdot 10^{-13}$
2	$1.817\,600\,000\,000 \cdot 10^{5}$	$1.981\,440\,000\,000 \cdot 10^{5}$	$2.831\,823\,664\,451 \cdot 10^{-11}$
3	$2.094\,080\,000\,000 \cdot 10^{5}$	$3.015\,680\,000\,000 \cdot 10^{5}$	$4.079\,034\,887\,638 \cdot 10^{-10}$
4	$3.106\,816\,000\,000 \cdot 10^{6}$	$2.844\,672\,000\,000 \cdot 10^{6}$	$1.626\,246\,826\,092 \cdot 10^{-8}$
5	$2.411\,468\,800\,000 \cdot 10^{7}$	$2.334\,668\,800\,000 \cdot 10^{7}$	$6.657\,697\,912\,971 \cdot 10^{-7}$
6	$1.201\,520\,640\,000 \cdot 10^{8}$	$1.188\,249\,600\,000 \cdot 10^{8}$	$1.075\,417\,522\,678 \cdot 10^{-5}$
7	$4.803\,983\,360\,000 \cdot 10^{8}$	$4.782\,909\,440\,000 \cdot 10^{8}$	$1.020\,027\,930\,076 \cdot 10^{-4}$
8	$1.682\,691\,072\,000 \cdot 10^{9}$	$1.678\,497\,280\,000 \cdot 10^{9}$	$6.441\,703\,922\,384 \cdot 10^{-4}$
9	$4.465\,326\,592\,000 \cdot 10^{9}$	$4.457\,859\,584\,000 \cdot 10^{9}$	$2.915\,294\,362\,053 \cdot 10^{-3}$
10	$1.270\,712\,678\,400 \cdot 10^{10}$	$1.269\,690\,726\,400 \cdot 10^{10}$	$9.586\,957\,518\,275 \cdot 10^{-3}$
11	$3.575\,989\,555\,200 \cdot 10^{10}$	$3.574\,346\,905\,600 \cdot 10^{10}$	$2.502\,293\,290\,932 \cdot 10^{-2}$
12	$7.216\,771\,584\,000 \cdot 10^{10}$	$7.214\,665\,062\,400 \cdot 10^{10}$	$4.671\,674\,615\,314 \cdot 10^{-2}$
13	$2.157\,236\,290\,560 \cdot 10^{11}$	$2.156\,963\,307\,520 \cdot 10^{11}$	$7.431\,403\,244\,734 \cdot 10^{-2}$
14	$3.653\,832\,509\,440 \cdot 10^{11}$	$3.653\,447\,936\,000 \cdot 10^{11}$	$8.524\,440\,819\,787 \cdot 10^{-2}$
15	$6.139\,877\,534\,720 \cdot 10^{11}$	$6.139\,384\,156\,160 \cdot 10^{11}$	$7.549\,379\,969\,948 \cdot 10^{-2}$
16	$1.555\,027\,751\,936 \cdot 10^{12}$	$1.554\,961\,097\,216 \cdot 10^{12}$	$5.371\,328\,339\,203 \cdot 10^{-2}$
17	$3.777\,623\,778\,304 \cdot 10^{12}$	$3.777\,532\,946\,944 \cdot 10^{12}$	$2.542\,714\,623\,741 \cdot 10^{-2}$
18	$7.199\,554\,861\,056 \cdot 10^{12}$	$7.199\,447\,475\,200 \cdot 10^{12}$	$9.078\,647\,283\,520 \cdot 10^{-3}$
19	$1.027\,837\,616\,282 \cdot 10^{13}$	$1.027\,823\,565\,670 \cdot 10^{13}$	$1.909\,818\,299\,438 \cdot 10^{-3}$
20	$2.746\,295\,274\,547 \cdot 10^{13}$	$2.746\,278\,890\,701 \cdot 10^{13}$	$1.907\,087\,633\,626 \cdot 10^{-4}$

Tabela 4: Obliczone wartości dla wielomianu P

k	z_k	$\lvert P(z_k)\rvert$	$\lvert p(z_k)\rvert$	$\lvert z_k - k\rvert$
1	0.999999999999836	$2.099\,200\,000\,000 \cdot 10^{4}$	$2.201\,600\,000\,000 \cdot 10^{4}$	$1.643\,130\,076\,445 \cdot 10^{-13}$
2	2.000000000055037	$3.491\,840\,000\,000 \cdot 10^{5}$	$3.655\,680\,000\,000 \cdot 10^{5}$	$5.503\,730\,804\,435 \cdot 10^{-11}$
3	2.999999996603420	$2.221\,568\,000\,000 \cdot 10^{6}$	$2.295\,296\,000\,000 \cdot 10^{6}$	$3.396\,579\,906\,223 \cdot 10^{-9}$
4	4.000000089724362	$1.046\,784\,000\,000 \cdot 10^{7}$	$1.072\,998\,400\,000 \cdot 10^{7}$	$8.972\,436\,216\,226 \cdot 10^{-8}$
5	4.999998573887910	$3.946\,393\,600\,000 \cdot 10^{7}$	$4.330\,393\,600\,000 \cdot 10^{7}$	$1.426\,112\,089\,753 \cdot 10^{-6}$
6	6.000020476673031	$1.291\,484\,160\,000 \cdot 10^{8}$	$2.061\,204\,480\,000 \cdot 10^{8}$	$2.047\,667\,303\,096 \cdot 10^{-5}$
7	6.999602070422420	$3.881\,231\,360\,000 \cdot 10^{8}$	$1.757\,670\,912\,000 \cdot 10^{9}$	$3.979\,295\,775\,798 \cdot 10^{-4}$
8	8.007772029099446	$1.072\,547\,328\,000 \cdot 10^{9}$	$1.852\,548\,659\,200 \cdot 10^{10}$	$7.772\,029\,099\,446 \cdot 10^{-3}$
9	8.915816367932559	$3.065\,575\,424\,000 \cdot 10^{9}$	$1.371\,743\,170\,560 \cdot 10^{11}$	$8.418\,363\,206\,744 \cdot 10^{-2}$
10	10.095455630535774 - 0.644932823624069i	$7.143\,113\,638\,036 \cdot 10^{9}$	$1.491\,263\,381\,675 \cdot 10^{12}$	$6.519\,586\,830\,380 \cdot 10^{-1}$
11	10.095455630535774 + 0.644932823624069i	$7.143\,113\,638\,036 \cdot 10^{9}$	$1.491\,263\,381\,675 \cdot 10^{12}$	$1.110\,918\,027\,272$
12	11.793890586174369 - 1.652477136407579i	$3.357\,756\,113\,172 \cdot 10^{10}$	$3.296\,021\,414\,130 \cdot 10^{13}$	$1.665\,281\,290\,598$
13	11.793890586174369 + 1.652477136407579i	$3.357\,756\,113\,172 \cdot 10^{10}$	$3.296\,021\,414\,130 \cdot 10^{13}$	$2.045\,820\,276\,678$
14	13.992406684487216 - 2.518824425710844i	$1.061\,206\,453\,308 \cdot 10^{11}$	$9.545\,941\,595\,184 \cdot 10^{14}$	$2.518\,835\,871\,191$
15	13.992406684487216 + 2.518824425710844i	$1.061\,206\,453\,308 \cdot 10^{11}$	$9.545\,941\,595\,184 \cdot 10^{14}$	$2.712\,880\,531\,285$
16	16.730744879792670 - 2.812624896721978i	$3.315\,103\,475\,982 \cdot 10^{11}$	$2.742\,089\,401\,676 \cdot 10^{16}$	$2.906\,001\,873\,538$
17	16.730744879792670 + 2.812624896721978i	$3.315\,103\,475\,982 \cdot 10^{11}$	$2.742\,089\,401\,676 \cdot 10^{16}$	$2.825\,483\,521\,350$
18	19.502442368818102 - 1.940331978642903i	$9.539\,424\,609\,818 \cdot 10^{12}$	$4.252\,502\,487\,993 \cdot 10^{17}$	$2.454\,021\,446\,313$
19	19.502442368818102 + 1.940331978642903i	$9.539\,424\,609\,818 \cdot 10^{12}$	$4.252\,502\,487\,993 \cdot 10^{17}$	$2.004\,329\,444\,310$
20	20.846910215194789	$1.114\,453\,504\,512 \cdot 10^{13}$	$1.374\,373\,319\,725 \cdot 10^{18}$	$8.469\,102\,151\,948 \cdot 10^{-1}$

Tabela 5: Obliczone wartości dla wielomianu P z zaburzonym współczynnikiem przy x^{19}

prezentuje dane dla tego eksperymentu. Można zauważyć, że w wielomianie, który ma miejsca zerowe rzeczywiste pojawiają się pierwiastki zespolone. Ze względu na tak duże odkształcenia wyników wynikłe z zaburzenia współczynników przy niewielkich zmianach danych można mówić o tym, że zadanie to jest źle uwarunkowane.

5 Model wzrostu populacji

5.1 Opis problemu

Zbadanie modelu wzrostu populacji (model logistyczny)

$$p_{n+1} := p_n + rp_n(1 - p_n), \text{ dla } n = 0, 1, \ldots, \tag{2}$$

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.

W tym celu należało przeprowadzić następujące eksperymenty.

(i) Wykonanie 40 iteracji wyrażenia (2) w arytmetyce `Float32` dla danych $p_0 = 0.01$ i $r = 3$. Ponowne wykonanie 40 iteracji wyrażenia (2) z niewielką modyfikacją tj. wykonanie 10 iteracji, zatrzymanie, zastosowanie obcięcia wyniku odrzucając cyfry po trzecim miejscu po przecinku (daje to liczbę 0.722) i kontynuowanie dalej obliczenia (do 40-stej iteracji) tak, jak gdyby był to ostatni wynik na wyjściu. Porównanie wyników obu iteracji.

(ii) Wykonanie 40 iteracji wyrażenia (2) dla danych $p_0 = 0.01$ i $r = 3$ w arytmetyce `Float32` i `Float64`. Porównanie wyników iteracji dla obu arytmetyk.

5.2 Rozwiązanie

Zaimplementowano podany model wzrostu populacji (2) i za pomocą stworzonej funkcji obliczono wyniki dla odpowiedniej liczby iteracji.

5.3 Wyniki

Zestawienie wyników dla obu eksperymentów prezentują odpowiednio Tabela 6 oraz Tabela 7.

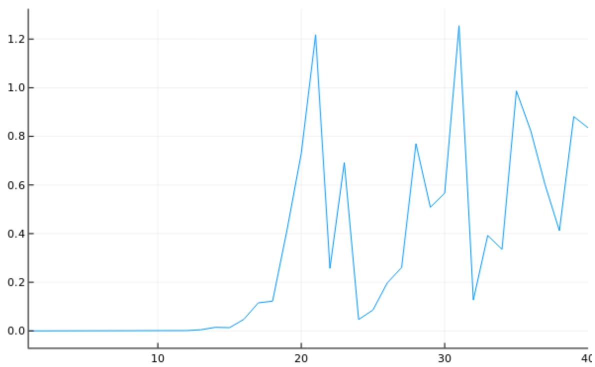
Iteracja	Bez modyfikacji	Z modyfikacją
1	0.0397	0.0397
2	0.154 071 73	0.154 071 73
3	0.545 072 6	0.545 072 6
4	1.288 978 1	1.288 978 1
5	0.171 518 8	0.171 518 8
10	0.722 930 6	0.722
11	1.323 836 4	1.324 147 9
12	0.037 716 985	0.036 488 414
15	1.270 483 7	1.257 216 9
17	0.786 042 8	0.901 085 5
19	0.165 524 72	0.577 893
20	0.579 903 6	1.309 691 1
25	1.007 080 6	1.092 910 8
30	0.752 920 9	1.319 182 2
35	1.021 099	0.034 241 438
40	0.258 605 48	1.093 568

Tabela 6: Wybrane wyniki kolejnych iteracji modelu logistycznego w arytmetyce `Float32` bez modyfikacji i z obcięciem wyniku 10 iteracji od 3 miejsca po przecinku

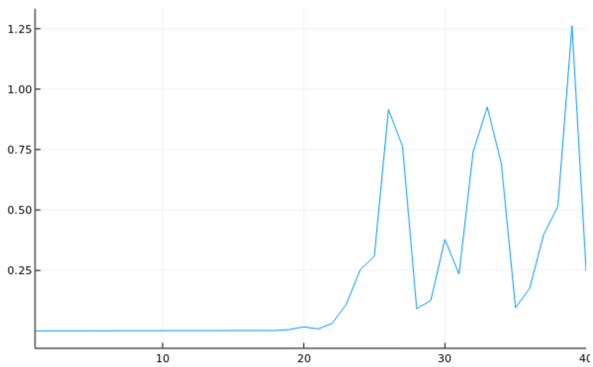
Dla lepszego ukazania rozbieżności pomiędzy kolejnymi iteracjami w eksperymentach zostały narysowane wykresy (Rysunek 3) przedstawiające wartość bezwzględną różnic otrzymanych wyników.

Iteracja	Float32	Float64
1	0.0397	0.0397
2	0.154 071 73	0.154 071 730 000 000 02
3	0.545 072 6	0.545 072 626 044 421 3
4	1.288 978 1	1.288 978 001 188 800 6
5	0.171 518 8	0.171 519 142 109 175 52
10	0.722 930 6	0.722 914 301 179 573
15	1.270 483 7	1.270 261 773 935 076 8
20	0.579 903 6	0.596 529 312 494 690 7
25	1.007 080 6	1.315 588 346 001 072
26	0.985 688 5	0.070 035 295 602 778 99
27	1.028 008 6	0.265 426 354 520 610 03
30	0.752 920 9	0.374 146 489 639 286 76
35	1.021 099	0.925 382 128 557 104 6
39	1.265 200 4	0.002 909 156 902 851 206 5
40	0.258 605 48	0.011 611 238 029 748 606

Tabela 7: Wybrane wyniki kolejnych iteracji modelu logistycznego w arytmetyce `Float32` i `Float64`



(a) `Float32` bez modyfikacji i z obcięciem



(b) `Float32` i `Float64`

Rysunek 3: Wykresy przedstawiają różnicę pomiędzy kolejnymi wynikami iteracji

5.4 Wnioski

Przeprowadzone w zadaniu eksperymenty są przykładami sprzężenia zwrotnego, czyli procesu, w którym dane wyjściowe jednego problemu są wejściem kolejnych obliczeń. Można zatem podejrzewać, że jeżeli w pierwszym eksperymencie wynik w dziesiątej iteracji jest zgodny tylko do trzeciego miejsca po przecinku, to w dalszych iteracjach też będzie istniała różnica między wynikami. Dziwić może jednak, że wyniki wyższych iteracji wydają się zupełnie nieskorelowane. Wskazuje to na istnienie pewnego chaosu w systemie, czy też, mówiąc inaczej niemożności przewidywania (sformułowanie Lorenza). Można jednak powiedzieć, że w pierwszym eksperymencie wprowadzony błąd był zbyt duży i postawić tezę, że jeżeli błąd ten zostanie zmniejszony, to dziwne zachowanie iteracji zniknie. W tym celu przeprowadzono drugi eksperyment. Dane nie zostały w żaden sposób zaburzone, jednak do wykonywania obliczeń zostały zastosowane różne precyzje. Również tutaj widać jednak, że małe odchylenie pojawiające się w pewnym momencie jest w dalszym etapie potęgowane i znów powoduje to nieskorelowanie wyników dla późniejszych iteracji. Mogłoby się wydawać że bardziej wiarygodne są obliczenia wykonane w arytmetyce `Float64`, jednak nie jest to do końca prawdziwy wniosek, gdyż jej precyzja jest ograniczona. Można zauważyć, że w każdej kolejnej iteracji rośnie liczba cyfr znaczących pozwalających na dokładne zapisanie wyniku. W pewnym momencie zatem precyzja arytmetyki `Float64` staje się niewystarczająca. Widać zatem, że niezależnie od tego, jak małe odchylenie od wartości początkowych zostanie wybrane, na skutek przeniesienia błędu jako wejście do kolejnej iteracji, a potem następnej, itd., będzie on gwałtownie rósł, tak że po stosunkowo niewielu iteracjach przewidywanie za pomocą komputera stanie się bezwartościowe. Zaobserwowana tutaj numeryczna niestabilność jest ciężka do uniknięcia, ponieważ nie istnieje arytmetyka o nieskończonej precyzji, która byłaby w stanie wykonać wszystkie obliczenia poprawnie, można jedynie przez wybór odpowiednio dużej precyzji opóźnić zjawisko niemożności przewidywania.

6 Iterowanie funkcji kwadratowej

6.1 Opis problemu

Zbadanie zachowania równania rekurencyjnego

$$x_{n+1} := x_n^2 + c, \text{ dla } n = 0, 1, \ldots, \tag{3}$$

gdzie c jest pewną daną stałą, dla następujących danych:

(i) $c = -2$ i $x_0 = 1$

(ii) $c = -2$ i $x_0 = 2$

(iii) $c = -2$ i $x_0 = 1.99999999999999$

(iv) $c = -1$ i $x_0 = 1$

(v) $c = -1$ i $x_0 = -1$

(vi) $c = -1$ i $x_0 = 0.75$

(vii) $c = -1$ i $x_0 = 0.25$

W tym celu należało wykonać 40 iteracji wyrażenia (3) i zaobserwować zachowanie generowanych ciągów, a także przeprowadzić iterację graficzną (3).

6.2 Rozwiązanie

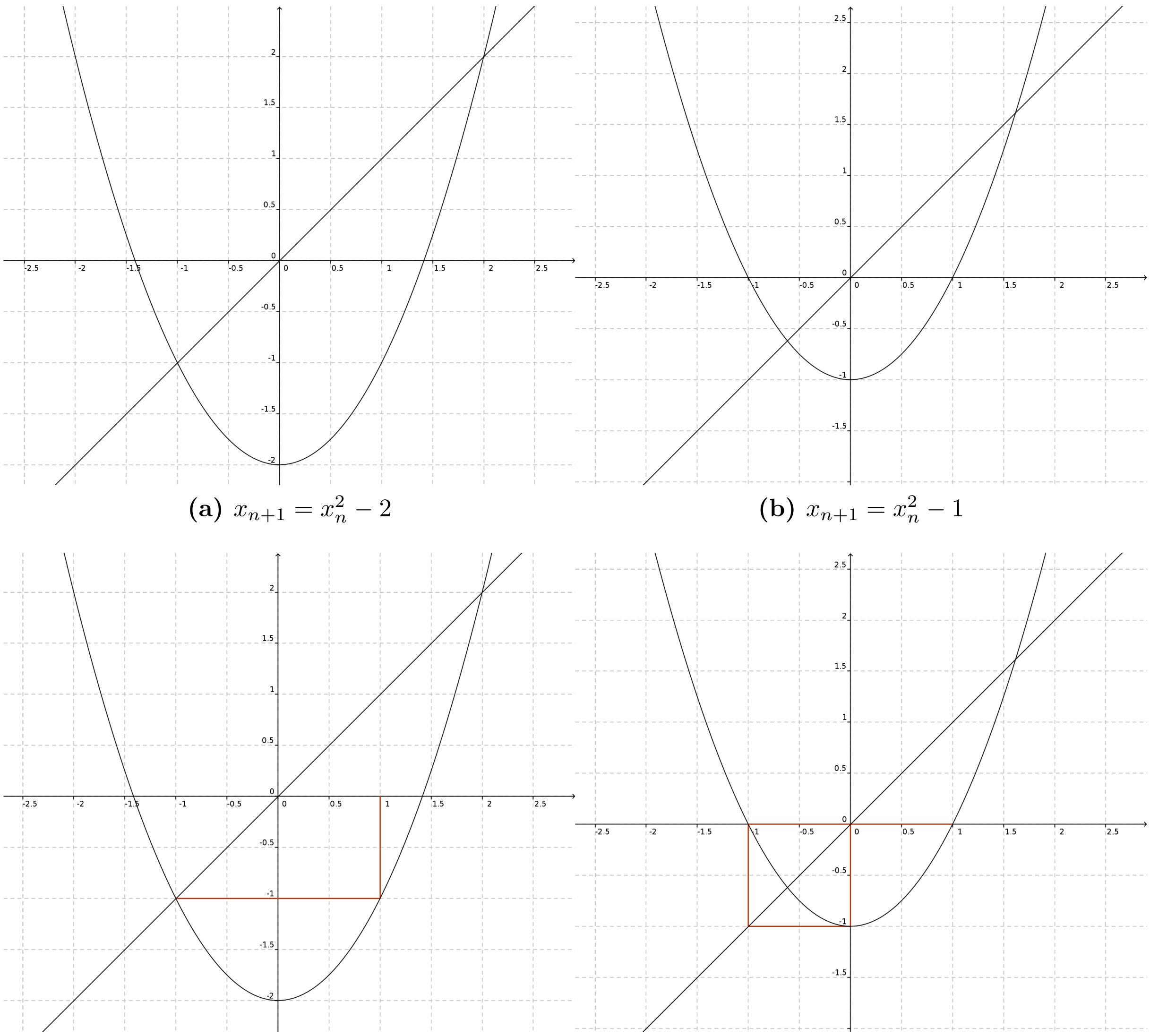
Zaimplementowano wyrażenie (3) i za pomocą stworzonej w programie funkcji dla danych parametrów wejściowych x_0 i c wykonano zadaną liczbę iteracji.

6.3 Wyniki

Wyniki kolejnych iteracji wyrażenia (3) dla różnych danych wejściowych przedstawia Tabela 8. W celu lepszego przedstawienia wyników metodą iteracji graficznej zostały stworzone wykresy, które przedstawia Rysunek 4.

6.4 Wnioski

Analizując zadanie poprzednie (model wzrostu populacji) można wyciągnąć wnioski, że za niestabilność numeryczną odpowiedzialna jest tylko i wyłącznie operacja podnoszenia do kwadratu. Iteracje wykonane w tym zadaniu obalają jednak ten wniosek. W tabeli 8 widać, że dla $x_0 = 1$ lub $x_0 = 2$ przy $c = -2$ iteracja zachowuje się stabilnie. W przypadku, gdy $x_0 = 1.99999999999999$ (jest to niewielkie odchylenie od 2) widać natomiast odmienne zachowanie, w którym występuje niestabilność. Doskonale obrazuje to metoda iteracji graficznej pokazana na wykresach 4g, 4h, 4i. Można zatem wyciągnąć wnioski, że niektóre z danych początkowych prowadzą do stabilnego zachowania, inne generują niestabilność wyników. W rzeczywistości w przedziale $[-2, 2]$ nie znajduje się zbyt wiele wartości prowadzących do rozwiązań stabilnych. Funkcja $\phi(x) = x^2 - 2$



(a) $x_{n+1} = x_n^2 - 2$

(b) $x_{n+1} = x_n^2 - 1$

(c) $x_0 = 1, c = -2$

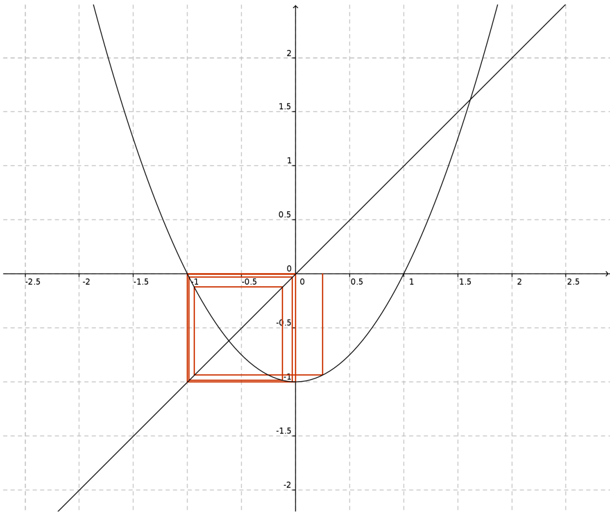
(d) $x_0 = 1, c = -1$

Rysunek 4: Iteracje graficzne wyrażenia $x_{n+1} = x_n^2 + c$ dla wybranych x_0 i c

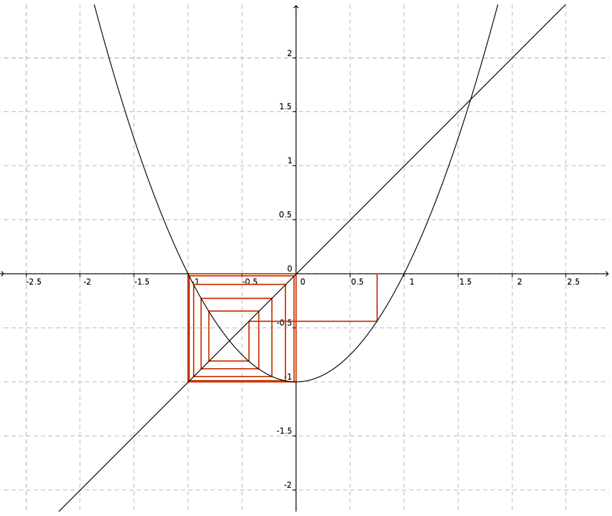
posiada dwa punkty stałe (tzn. takie x dla których $\phi(x) = x$) − -1, 2, co pokazuje wykres 4a. Wartości początkowe dla których $\phi(x)$ jest zbieżna do tych punktów prowadzą do stabilnych rozwiązań. Eksperymentalne sprawdzenie za pomocą iteracji graficznej pokazało jednak że w dużej mierze $\phi(x)$ jest rozbieżna. Zbieżność została zaobserwowana dla pojedynczych wartości x_0 takich jak: $-2, -1, 0, 1, 2$. Obserwacje te pokazują, że analiza błędu nie jest łatwa do przeprowadzenia, co staje się jeszcze bardziej widoczne kiedy wartość $c = -2$ została zamieniona na $c = -1$. Dla tak zdefiniowanej funkcji $\phi(x)$ otrzymano punkty stałe $\frac{1-\sqrt{5}}{2}$ i $\frac{1+\sqrt{5}}{2}$. Zaobserwować można jednak inną ciekawą rzecz. Startując od x_0 równego $1, -1, 0.75$ czy 0.25 po pewnej liczbie iteracji proces ustala się i powtarzają się tylko dwie wartości: 0 i -1. Do takiego efektu doprowadza również wybranie wielu innych wartości początkowych. Dla takich wartości x_0 układ sprzężenia zwrotnego jest w stanie idealnie stabilnym, co obrazują wykresy 4d, 4f i 4e. Dla tak przewidywalnych procesów niewielkie błędy podczas przebiegu zanikają lub ulegają redukcji, mogą więc zostać pominięte. Można posłużyć się zatem arytmetyką o skończonej precyzji, która staje się narzędziem jak najbardziej zdatnym do analizy i nie może zawieść.

It.	$c=-2$			$c=-1$			
	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	1.999 999 999 999 96	0.0	0.0	-0.4375	-0.9375
2	-1.0	2.0	1.999 999 999 999 840 1	-1.0	-1.0	$-0.808\,593\,75$	$-0.121\,093\,75$
3	-1.0	2.0	1.999 999 999 999 360 5	0.0	0.0	$-0.346\,176\,147\,460\,937\,5$	$-0.985\,336\,303\,710\,937\,5$
4	-1.0	2.0	1.999 999 999 997 442	-1.0	-1.0	$-0.880\,162\,074\,929\,103\,3$	$-0.029\,112\,368\,589\,267\,135$
5	-1.0	2.0	1.999 999 999 989 768 2	0.0	0.0	$-0.225\,314\,721\,856\,495\,6$	$-0.999\,152\,469\,995\,122\,6$
6	-1.0	2.0	1.999 999 999 959 072 7	-1.0	-1.0	$-0.949\,233\,276\,114\,730\,1$	$-0.001\,694\,341\,702\,645\,596\,5$
7	-1.0	2.0	1.999 999 999 836 291	0.0	0.0	$-0.098\,956\,187\,516\,496\,6$	$-0.999\,997\,129\,206\,194\,7$
8	-1.0	2.0	1.999 999 999 345 163 8	-1.0	-1.0	$-0.990\,207\,672\,952\,199\,9$	$-5.741\,579\,369\,278\,327 \cdot 10^{-6}$
9	-1.0	2.0	1.999 999 997 380 655 3	0.0	0.0	$-0.019\,488\,764\,426\,589\,09$	$-0.999\,999\,999\,967\,034\,3$
10	-1.0	2.0	1.999 999 989 522 621	-1.0	-1.0	$-0.999\,620\,188\,061\,125$	$-6.593\,148\,249\,578\,462 \cdot 10^{-11}$
11	-1.0	2.0	1.999 999 958 090 484 1	0.0	0.0	$-0.000\,759\,479\,620\,641\,156\,9$	-1.0
12	-1.0	2.0	1.999 999 832 361 938 3	-1.0	-1.0	$-0.999\,999\,423\,190\,705\,8$	0.0
13	-1.0	2.0	1.999 999 329 447 781 4	0.0	0.0	$-1.153\,618\,255\,700\,372\,7 \cdot 10^{-6}$	-1.0
14	-1.0	2.0	1.999 997 317 791 574 9	-1.0	-1.0	$-0.999\,999\,999\,998\,669\,2$	0.0
15	-1.0	2.0	1.999 989 271 173 493 7	0.0	0.0	$-2.661\,648\,679\,236\,350\,3 \cdot 10^{-12}$	-1.0
16	-1.0	2.0	1.999 957 084 809 082 6	-1.0	-1.0	-1.0	0.0
17	-1.0	2.0	1.999 828 341 078 044	0.0	0.0	0.0	-1.0
18	-1.0	2.0	1.999 313 393 778 961 3	-1.0	-1.0	-1.0	0.0
19	-1.0	2.0	1.997 254 046 543 948 1	0.0	0.0	0.0	-1.0
20	-1.0	2.0	1.989 023 726 436 175 2	-1.0	-1.0	-1.0	0.0
21	-1.0	2.0	1.956 215 384 326 048 6	0.0	0.0	0.0	-1.0
22	-1.0	2.0	1.826 778 629 873 91	-1.0	-1.0	-1.0	0.0
23	-1.0	2.0	1.337 120 162 563 999 7	0.0	0.0	0.0	-1.0
24	-1.0	2.0	$-0.212\,109\,670\,864\,823\,13$	-1.0	-1.0	-1.0	0.0
25	-1.0	2.0	$-1.955\,009\,487\,525\,616\,3$	0.0	0.0	0.0	-1.0
26	-1.0	2.0	1.822 062 096 315 173	-1.0	-1.0	-1.0	0.0
27	-1.0	2.0	1.319 910 282 828 443	0.0	0.0	0.0	-1.0
28	-1.0	2.0	$-0.257\,836\,845\,283\,739\,6$	-1.0	-1.0	-1.0	0.0
29	-1.0	2.0	$-1.933\,520\,161\,214\,128\,8$	0.0	0.0	0.0	-1.0
30	-1.0	2.0	1.738 500 213 821 510 9	-1.0	-1.0	-1.0	0.0
31	-1.0	2.0	1.022 382 993 457 438 9	0.0	0.0	0.0	-1.0
32	-1.0	2.0	$-0.954\,733\,014\,689\,006\,5$	-1.0	-1.0	-1.0	0.0
33	-1.0	2.0	$-1.088\,484\,870\,662\,841\,2$	0.0	0.0	0.0	-1.0
34	-1.0	2.0	$-0.815\,200\,686\,338\,097\,8$	-1.0	-1.0	-1.0	0.0
35	-1.0	2.0	$-1.335\,447\,840\,993\,894\,4$	0.0	0.0	0.0	-1.0
36	-1.0	2.0	$-0.216\,579\,063\,984\,746\,25$	-1.0	-1.0	-1.0	0.0
37	-1.0	2.0	$-1.953\,093\,509\,043\,491$	0.0	0.0	0.0	-1.0
38	-1.0	2.0	1.814 574 255 067 817 4	-1.0	-1.0	-1.0	0.0
39	-1.0	2.0	1.292 679 727 154 924 4	0.0	0.0	0.0	-1.0
40	-1.0	2.0	$-0.328\,979\,123\,002\,670\,2$	-1.0	-1.0	-1.0	0.0

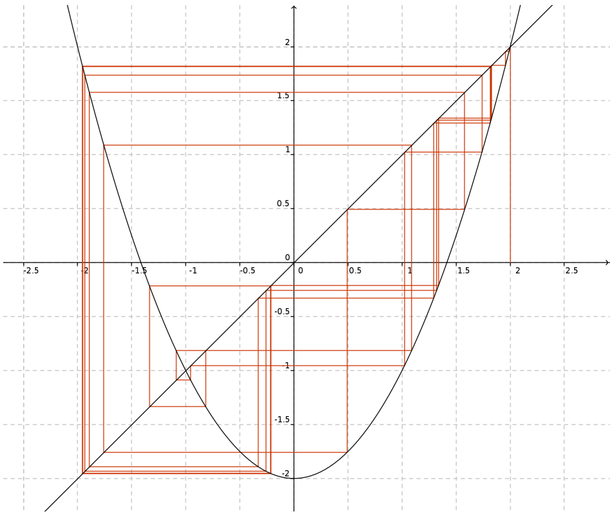
Tabela 8: Kolejne iteracje funkcji $x_{n+1} = x_n^2 + c$ dla danych x_0 i c



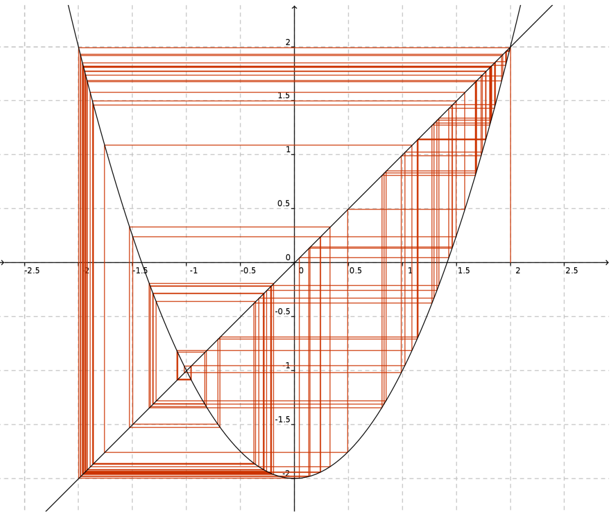
(e) $x_0 = 0.25, c = -1$



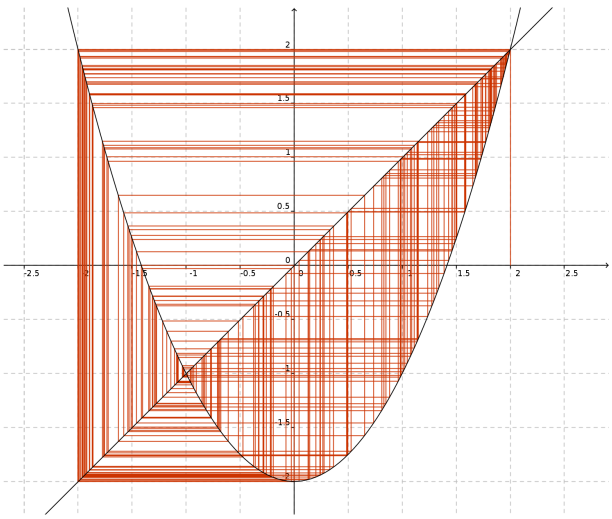
(f) $x_0 = 0.75, c = -1$



(g) $x_0 = 1.99999999999999, c = -2$ – 40 iteracji



(h) $x_0 = 1.99999999999999, c = -2$ – 100 iteracji



(i) $x_0 = 1.99999999999999, c = -2$ – 200 iteracji

Rysunek 4: Iteracje graficzne wyrażenia $x_{n+1} = x_n^2 + c$ dla wybranych x_0 i c