

Sprawozdanie

Obliczenia naukowe - Lista 2

Witold Karaś

1 zadanie 1

Zadanie polegało na niewielkiej zmianie wartości x_4 oraz x_5 .

1.1 Wnioski:

Zmiany wyników są widoczne przy obliczeniach w arytmetyce **Float64** (Double). Niewielka zmiana danych wejściowych znacząco wpłynęła na wynik działania algorytmów. W tym przypadku, mimo że ilorazy miały różny rząd wielkości, kolejność sumowania nie miała, w arytmetyce **Float64**, znaczenia dla wyników. Dla arytmetyki **Float32** zmiana nie miała wpływu na wyniki.

<i>typ zmiennej</i>	<i>output</i>
Float32	-0.4999443
	-0.4543457
	-0.5
	-0.5
Float64	-0.004296342739891585
	-0.004296342998713953
	-0.004296342842280865
	-0.004296342842280865

Tabela 1: wartości z wyjścia programu `zadanie1.jl`

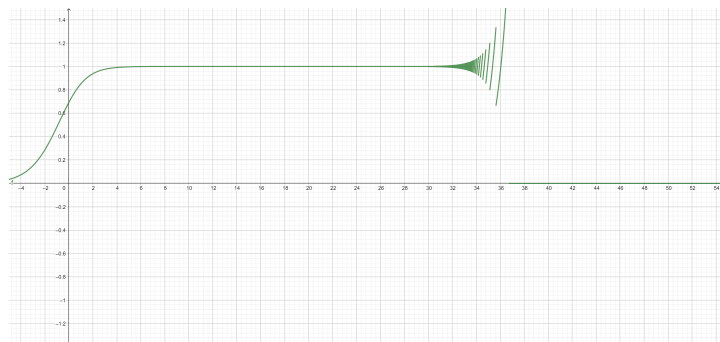
2 zadanie 2

Zadanie polegało na sprawdzeniu jak programy do wizualizacji funkcji radzą sobie z przedstawieniem funkcji $f(x) = e^x \ln(1+e^{-x})$ oraz policzenie granicy

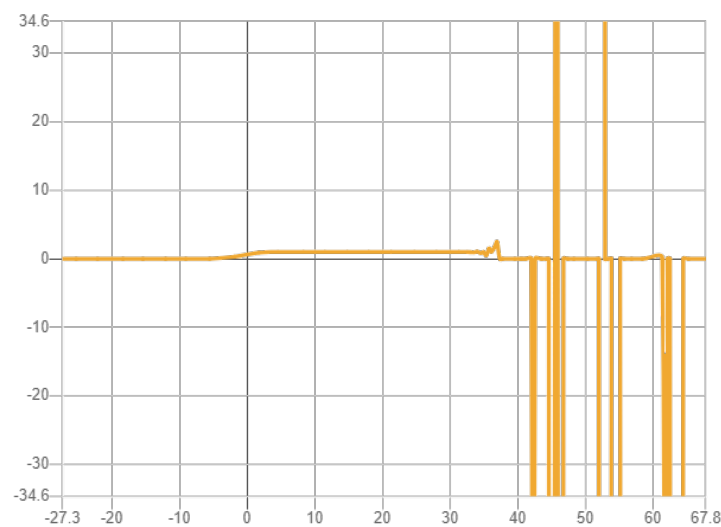
$$\lim_{x \rightarrow \infty} f(x) = 1$$

2.1 Wnioski:

Do narysowania wskazanej funkcji użyłem webowych aplikacji [GeoGebra](#) i [Microsoft Math Solver](#)



Rysunek 1: wykres z GeoGebra



Rysunek 2: wykres z Microsoft Math Solver

Oba programy rysują błędne wykresy. Wynika to z faktu, że e^x rośnie wykładniczo, a $\ln(1+e^{-x}) \rightarrow 0$, czyli mnożona jest bardzo duża i bardzo ma-

ła liczba. Dysponujemy arytmetyką o skończonej precyzji przez co mantysa traci na precyzji. Przy $x \approx 35$ wykresy stają się niewiarygodne.

3 zadanie 3

Zadanie polegało na rozwiązaniu układu równań linowych $\mathbf{Ax} = \mathbf{b}$ za pomocą dwóch algorytmów: eliminacji Gaussa $x = A \backslash b$ oraz $x = A^{-1}b$, gdzie $x = (1, \dots, 1)^T$, $A \in \mathbb{R}^{n \times n}$ jest losową macierzą lub macierzą Hilberta, $b \in \mathbb{R}^n$ jest wektorem prawych stron

Przedstawione wyniki *gauss err* i *inverted err* zostały policzone korzystając ze wzoru:

$$\frac{\text{norm}(\tilde{x} - x)}{\text{norm}(x)}$$

. Eksperyment pokazuje że dla macierzy Hilberta zadanie jest źle uwarunkowane przez błędy obliczeń, niezależnie od wybranej metody.

Dla macierzy Hilberta				
n	rank(A)	cond(A)	<i>gauss err</i>	<i>inverted err</i>
1	1	1.0	0.0	0.0
3	3	524.0567775860644	8.022593772267726e-15	0.0
5	5	476607.25024259434	1.6828426299227195e-12	3.3544360584359632e-12
7	7	4.75367356583129e8	1.2606867224171548e-8	4.713280397232037e-9
9	9	4.931537564468762e11	3.8751634185032475e-6	4.541268303176643e-6
11	10	5.222677939280335e14	0.00015827808158590435	0.007618304284315809
13	11	3.344143497338461e18	0.11039701117868264	5.331275639426837
15	12	3.674392953467974e17	4.696668350857427	7.344641453111494
17	12	1.263684342666052e18	13.707236683836307	10.516942378369349
19	13	6.471953976541591e18	9.720589712655698	12.233761393757726
21	13	3.290126328601399e18	56.40267595616145	43.4753048667801
23	13	6.313778670724671e17	12.483655076018373	13.803784630487236
25	13	1.3719347461445998e18	10.15919484338797	16.93987792970947
27	14	4.424587877361583e18	30.11850661319111	28.752075126924804
29	14	8.05926200352767e18	25.047149256115667	95.60461031775714

Tabela 2: wartości z wyjścia programu `zadanie3.jl`

Dla macierzy losowej					
n	c	$rank(A)$	$cond(A)$	$gauss\ err$	$inverted\ err$
5	1.0	5	1.0000000000000004	1.4895204919483638e-16	1.1102230246251565e-16
5	10.0	5	10.000000000000005	9.930136612989092e-17	1.9860273225978183e-16
5	1000.0	5	999.9999999999449	2.106710720622688e-14	2.1518860363986142e-14
5	1.0e7	5	9.999999986840717e6	3.3468138832372843e-10	3.2251456699322597e-10
5	1.0e12	5	1.0000120160444469e12	2.9337670118852183e-5	3.466970143890919e-5
5	1.0e16	4	9.058833363884072e15	0.2872201265139816	0.2867517436738616
10	1.0	10	1.0000000000000016	2.9996574304705467e-16	2.0770370905276122e-16
10	10.0	10	10.000000000000004	3.2177320244274193e-16	4.1093252186201184e-16
10	1000.0	10	1000.0000000000507	8.944843056870317e-15	8.39316858142599e-15
10	1.0e7	10	9.99999998732708e6	4.955625093760456e-11	5.707106429696928e-11
10	1.0e12	10	9.999449703833724e11	3.3926252911339658e-6	1.983321604069273e-6
10	1.0e16	9	9.231859885643694e15	0.012840193015659251	0.026087199776662497
20	1.0	20	1.0000000000000018	4.946414258176871e-16	4.3568297570458958e-16
20	10.0	20	9.999999999999993	4.1910000110727263e-16	5.1238927586247155e-16
20	1000.0	20	1000.000000000015	1.5451217583057598e-14	1.26308058855971e-14
20	1.0e7	20	9.999999995287322e6	1.2373756600196829e-11	3.903576264676482e-11
20	1.0e12	20	1.0000365381020696e12	1.7519210494067048e-5	1.5030584949375567e-5
20	1.0e16	19	1.07738779481172e16	0.034284522301952836	0.052557188402976115

Tabela 3: wartości z wyjścia programu `zadanie3.jl`

4 zadanie 4

Zadanie polegało na instalacji pakietu **Polynomials** oraz wykorzystaniu jego funkcji do znalezienia miejsc zerowych wielomianu Wilkinsona w dwóch postaciach - naturalnej oraz iloczynowej:

$$P(x) = x^{20} - 210x^{19} + 20615x^{18} - 1256850x^{17} + 53327946x^{16} - 1672280820x^{15} + 40171771630x^{14} - 756111184500x^{13} + 11310276995381x^{12} - 135585182899530x^{11} + 1307535010540395x^{10} - 10142299865511450x^9 + 63030812099294896x^8 - 311333643161390640x^7 + 1206647803780373360x^6 - 3599979517947607200x^5 + 8037811822645051776x^4 - 12870931245150988800x^3 + 13803759753640704000x^2 - 8752948036761600000x^1 + 2432902008176640000x^0$$

oraz

$$p(x) = \prod_{i=1}^{20} (x - i)$$

4.1 Wnioski:

Oczywiście $p(x) = P(x)$. Obliczone pierwiastki $P(x)$ różnią się od spodziewanych wyników. Niewielka zmiana współczynnika znacząco zmienia wyniki co implikuje, że zadanie jest źle uwarunkowane. Dokładność arytmetyki **Float64** ma 15-17 cyfr znaczących w systemie dziesiętnym. Wolny współczynnik wielomianu to $20!$ i ma 19 cyfr znaczących co wykracza poza precyzję jaką dysponujemy.

k	z_k	$ P(z_k) $	$ p(z_k) $	$ z_k - k $
1	0.9999999999996989	35696.50964788257	36720.50964788227	3.0109248427834245e-13
2	2.00000000000283182	176252.60026668405	192636.60026691604	2.8318236644508943e-11
3	2.9999999995920965	279157.6968824087	362101.69687113096	4.0790348876384996e-10
4	3.999999837375317	3.0271092988991085e6	2.7649652999648857e6	1.626246826091915e-8
5	5.000000665769791	2.2917473756567076e7	2.2277473671348542e7	6.657697912970661e-7
6	5.999989245824773	1.2902417284205095e8	1.2769707122070245e8	1.0754175226779239e-5
7	7.000102002793008	4.805112754602064e8	4.780526156335614e8	0.00010200279300764947
8	7.999355829607762	1.6379520218961136e9	1.6337585675856934e9	0.0006441703922384079
9	9.002915294362053	4.877071372550003e9	4.870348427548107e9	0.002915294362052734
10	9.990413042481725	1.3638638195458128e10	1.362843071072106e10	0.009586957518274986
11	11.025022932909318	3.585631295130865e10	3.584087897760478e10	0.025022932909317674
12	11.953283253846857	7.533332360358197e10	7.531256581876213e10	0.04671674615314281
13	13.07431403244734	1.9605988124330817e11	1.9602984002587503e11	0.07431403244734014
14	13.914755591802127	3.5751347823104315e11	3.574748406282602e11	0.08524440819787316
15	15.075493799699476	8.21627123645597e11	8.215740477766903e11	0.07549379969947623
16	15.946286716607972	1.5514978880494067e12	1.5514314565843672e12	0.05371328339202819
17	17.025427146237412	3.694735918486229e12	3.6946500070912217e12	0.025427146237412046
18	17.99092135271648	7.650109016515867e12	7.650001670877033e12	0.009078647283519814
19	19.00190981829944	1.1435273749721195e13	1.14351402511197e13	0.0019098182994383706
20	19.999809291236637	2.7924106393680727e13	2.7923942556843e13	0.00019070876336257925

Tabela 4: wartości z wyjścia programu **zadanie4.jl**, b)

k	z_k	$ P'(z_k) $	$ z_k - k $
1	0.999999999998357 + 0.0im	20259.872313418207	1.6431300764452317e-13
2	2.0000000000550373 + 0.0im	346541.4137593836	5.503730804434781e-11
3	2.99999999660342 + 0.0im	2.2580597001197007e6	3.3965799062229962e-9
4	4.000000089724362 + 0.0im	1.0542631790395478e7	8.972436216225788e-8
5	4.9999857388791 + 0.0im	3.757830916585153e7	1.4261120897529622e-6
6	6.000020476673031 + 0.0im	1.3140943325569446e8	2.0476673030955794e-5
7	6.99960207042242 + 0.0im	3.939355874647618e8	0.00039792957757978087
8	8.007772029099446 + 0.0im	1.184986961371896e9	0.007772029099445632
9	8.915816367932559 + 0.0im	2.2255221233077707e9	0.0841836320674414
10	10.095455630535774 - 0.6449328236240688im	1.0677921232930157e10	0.6519586830380407
11	10.095455630535774 + 0.6449328236240688im	1.0677921232930157e10	1.1109180272716561
12	11.793890586174369 - 1.6524771364075785im	3.1401962344429485e10	1.665281290598479
13	11.793890586174369 + 1.6524771364075785im	3.1401962344429485e10	2.0458202766784277
14	13.992406684487216 - 2.5188244257108443im	2.157665405951858e11	2.518835871190904
15	13.992406684487216 + 2.5188244257108443im	2.157665405951858e11	2.7128805312847097
16	16.73074487979267 - 2.812624896721978im	4.850110893921027e11	2.9060018735375106
17	16.73074487979267 + 2.812624896721978im	4.850110893921027e11	2.825483521349608
18	19.5024423688181 - 1.940331978642903im	4.557199223869993e12	2.4540214463129764
19	19.5024423688181 + 1.940331978642903im	4.557199223869993e12	2.0043294443099486
20	20.84691021519479 + 0.0im	8.756386551865696e12	0.8469102151947894

Tabela 5: wartości z wyjścia programu **zadanie4.jl**, b)

5 zadanie 5

Zadanie polegało na przeprowadzeniu eksperymentów dla funkcji rekurencyjnej:

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

dla danych początkowych $p_0 = 0.01, r = 3$ oraz 40 iteracji. W części **a)** rekurencja została podzielona na 4 interwały takie że po każdym stosowane było obcięcie do 3 miejsc po przecinku, obliczenia wykonywane były we Float32. W części **b)** program obliczał tą samą rekurencję w precyzji Float32 i Float64.

5.1 Wnioski:

Wyniki w tabeli wskazują jak ważna jest dokładność z jaką reprezentowane są liczby.

Float32, 4*10 iteracji z obcięciem 0.715	Float32 40 iteracji 0.25860548
Float32 40 iteracji 0.25860548	Float64 40 iteracji 0.011611238029748606

Tabela 6: wartości z wyjścia programu `zadanie5.jl`

6 zadanie 6

Zadanie polegało na przeprowadzeniu eksperymentów dla funkcji rekurencyjnej:

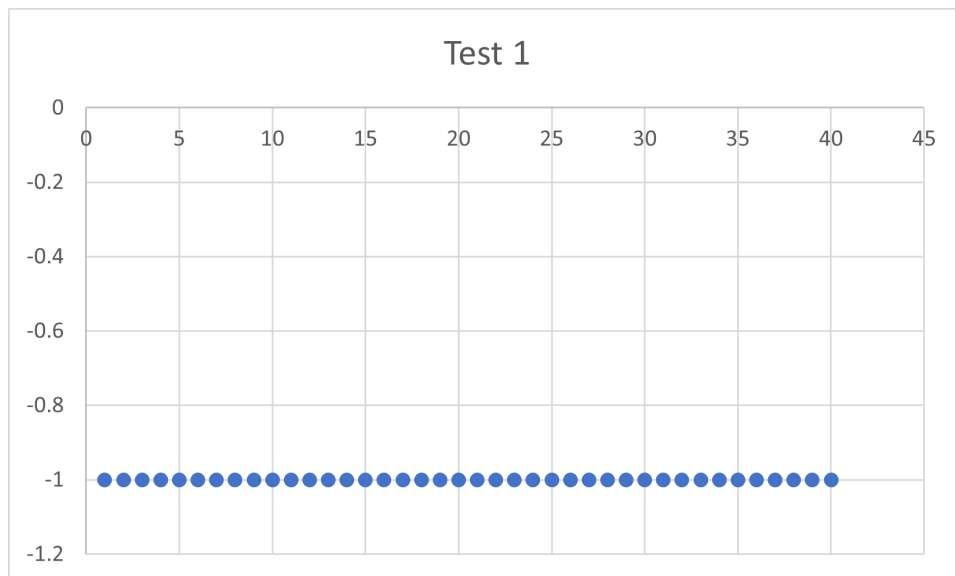
$$x_{n+1} := x_n^2 + c, \text{ dla } n = 0, 1, 2, \dots$$

dla danych początkowych:

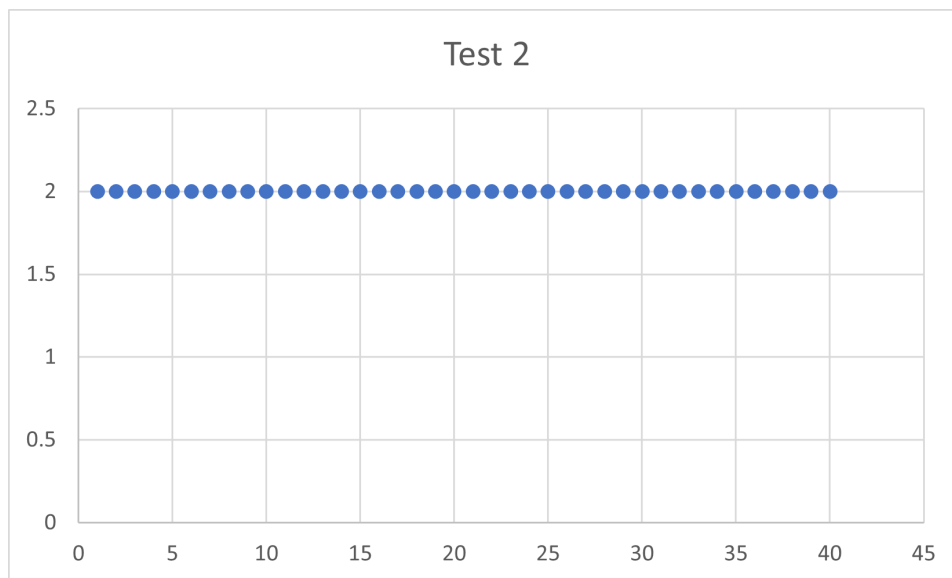
- **Test 1:** $c = -2, x_0 = 1$
- **Test 2:** $c = -2, x_0 = 2$
- **Test 3:** $c = -2, x_0 = 1.9999999999999999$
- **Test 4:** $c = -1, x_0 = 1$
- **Test 5:** $c = -1, x_0 = -1$
- **Test 6:** $c = -1, x_0 = 0.75$
- **Test 7:** $c = -1, x_0 = 0.25$

Dla danych początkowych program wykonywał 40 iteracji.

6.1 Wyniki:



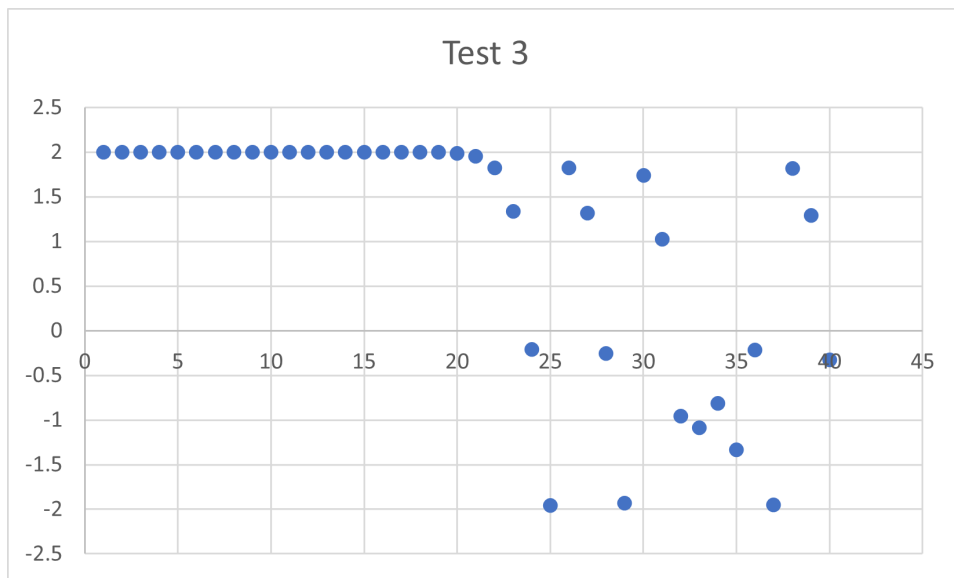
Rysunek 3: Test 1



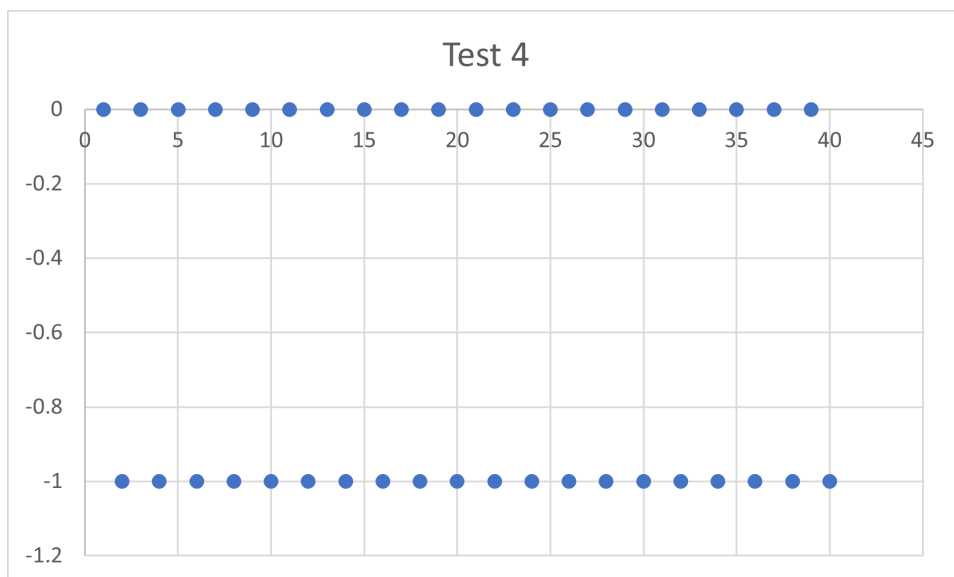
Rysunek 4: Test 2

6.2 Wnioski:

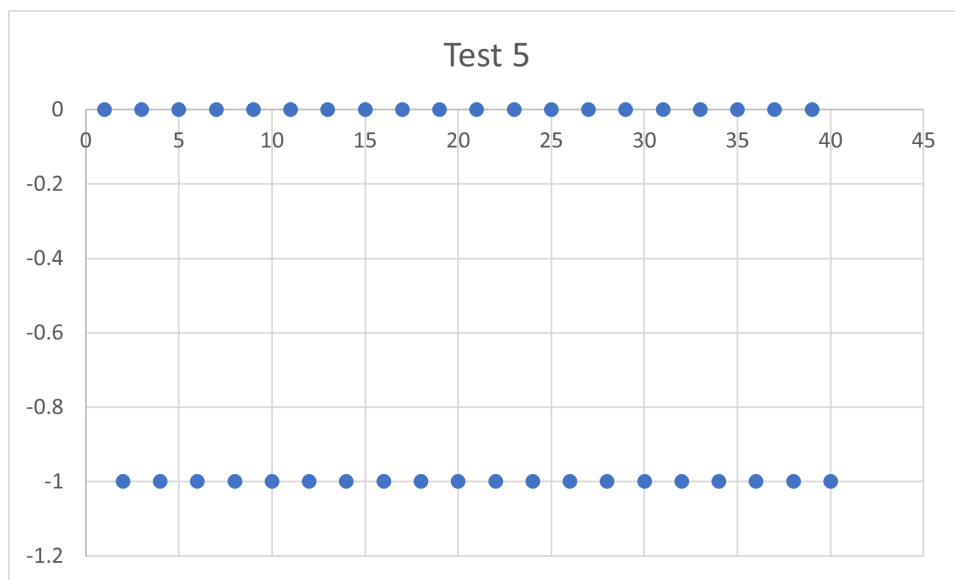
Dla testów 2 i 3 dla niewielkiej zmiany danych wejściowych dane wyjściowe mają duży rozrzut. Z kolei dla różnych danych wejściowych eksperymentów 4 i 5 dane wyjściowe zachowują się identycznie. Wykresy 6 i 7 również wykazują pewne podobieństwo mimo że różnica danych wejściowych jest znaczna.



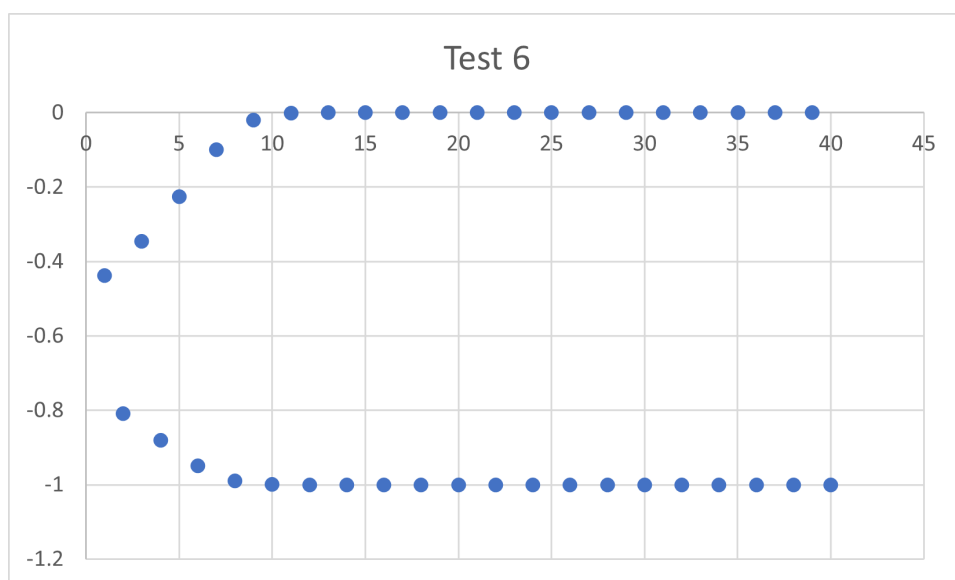
Rysunek 5: Test 3



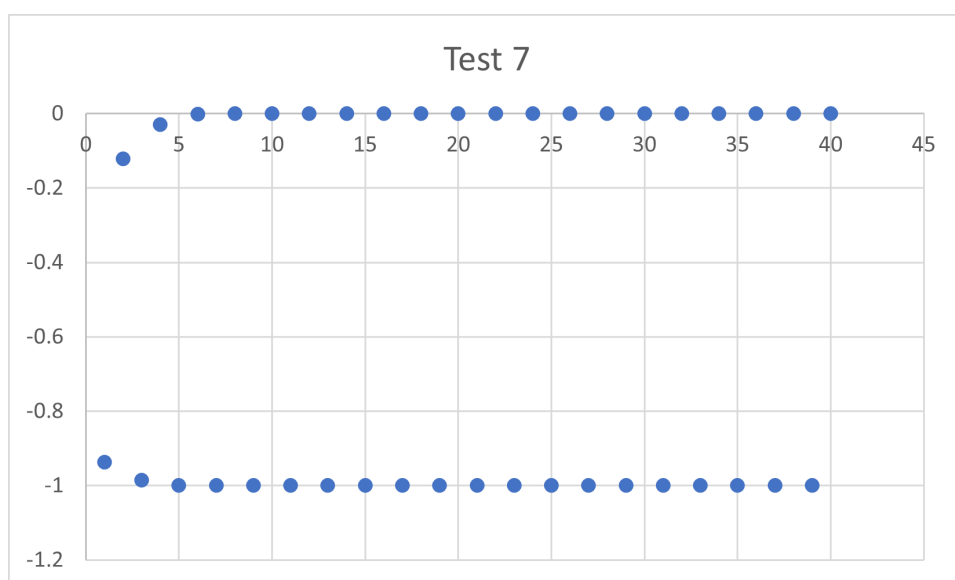
Rysunek 6: Test 4



Rysunek 7: Test 5



Rysunek 8: Test 6



Rysunek 9: Test 7