

# MOwNiT - arytmetyka komputerowa

Paweł Podedworny

06.03.2024

## 1 Opis ćwiczenia

Dana jest zależność rekurencyjna  $3x_{k-1} - 10x_k + 3x_{k+1} = 0$ . Wartości początkowe  $x_0 = 1$ ,  $x_1 = \frac{1}{3}$ . Wyznaczyć wartości  $x_k$ ,  $x_{k+1}$  dla  $k = 45$ . Następnie korzystając z wyznaczonych wartości  $x_k$  i  $x_{k+1}$  obliczyć  $x_1$  i  $x_0$ , wykonując rekurencję w tył. Porównać wyznaczone wartości  $x_1$  i  $x_0$  z dokładnymi wartościami początkowymi 1 i  $\frac{1}{3}$ . Wykonać obliczenia dla różnej precyzji zmiennych (float, double, long double). Skomentować różnice. Co będzie, jeśli wszędzie liczbę 3 zastąpimy przez liczbę 2 lub 20, lub 30?

## 2 Dane techniczne

Komputer z systemem Windows 10 x64

Procesor: AMD Ryzen 5 3600 3.60GHz

Pamięć RAM: 16GB 3200MHz

Środowisko: CLion 2023.3.4 / PyCharm 2023.1

Język: C++17 / Python 3.10 z biblioteką mpmath

## 3 Przekształcenia

Z racji późniejszego zastąpienia liczby 3 innymi, podstawiam pod nią liczbę  $c$

$$c \cdot x_{k-1} - 10 \cdot x_k + c \cdot x_{k+1} = 0$$

Do obliczenia wartości  $x_k$ ,  $x_{k+1}$  dla  $k = 45$

$$x_k = \frac{10}{c} \cdot x_{k-1} - x_{k-2}$$

Analogicznie do ponownego obliczenia wartości  $x_0$ ,  $x_1$

$$x_k = \frac{10}{c} \cdot x_{k+1} - x_{k+2}$$

## 4 Porównanie typów

Float	Double	Long Double
4 bajty	8 bajtów	16 bajtów
Znak: 1 bit	Znak: 1 bit	Znak: 1 bit
Mantysa: 23 bity	Mantysa: 52 bity	Mantysa: 65 bity
Wykładnik: 8 bitów	Wykładnik: 11 bitów	Wykładnik: 62 bity
Dokładność: 6-7 cyfr znaczących	Dokładność: 15-16 cyfr znaczących	Dokładność 19-20 cyfr znaczących

Tabela 1: Porównanie specyfikacji typów float, double i long double w C++

## 5 Wyniki obliczeń

### 5.1 Liczba $c = 3$

	Obliczanie $x_{45}, x_{46}$	Ponowne obliczanie $x_0, x_1$
$x_0$	1.0000000000	-1258581798329908673481015296.000
$x_1$	0.3333333433	-419527253812140175353970688.000
$x_2$	0.1111111417	-139842414862922712833064960.000
$x_3$	0.0370371304	-46614138287640904277688320.000
$x_4$	0.0123459594	-15538045327265965021331456.000
$x_5$	0.0041160677	-5179348250268404239302656.000
$x_6$	0.0013742661	-1726449368717738721148928.000
$x_7$	0.0004648194	-575483110896313900728320.000
$x_8$	0.0001751318	-191827709636904136736768.000
$x_9$	0.0001189533	-63942571380167921369088.000
$x_{10}$	0.0002213791	-21314190460055973789696.000
$x_{11}$	0.0006189770	-7104730153351991263232.000
$x_{12}$	0.0018418776	-2368243478275655991296.000
$x_{13}$	0.0055206148	-789414516214800056320.000
$x_{14}$	0.0165601708	-263138166207538003968.000
$x_{15}$	0.0496799536	-87712722069179334656.000
$x_{16}$	0.1490396708	-29237573290052026368.000
$x_{17}$	0.4471189380	-9745857763350675456.000
$x_{18}$	1.3413567543	-3248619254450225152.000
$x_{19}$	4.0240702629	-1082873130629726208.000
$x_{20}$	12.0722103119	-360957710209908736.000
$x_{21}$	36.2166290283	-120319239599947776.000
$x_{22}$	108.6498870850	-40106413199982592.000
$x_{23}$	325.9496459961	-13368804042080256.000
$x_{24}$	977.8489379883	-4456268014026752.000
$x_{25}$	2933.5468750	-1485422626603008.000
$x_{26}$	8800.6406250	-495140875534336.000
$x_{27}$	26401.9218750	-165046952919040.000
$x_{28}$	79205.7656250	-55015649574912.000
$x_{29}$	237617.2968750	-18338549858304.000
$x_{30}$	712851.8750	-6112849428480.000
$x_{31}$	2138555.500	-2037614510080.000
$x_{32}$	6415666.500	-679199113216.000
$x_{33}$	19247000.000	-226382594048.000
$x_{34}$	57741000.000	-75409539072.000
$x_{35}$	173223008.000	-24982538240.000
$x_{36}$	519669024.000	-7865584640.000
$x_{37}$	1559007104.000	-1236077696.000
$x_{38}$	4677021184.000	3745325824.000
$x_{39}$	14031063040.000	13720497152.000
$x_{40}$	42093187072.000	41989664768.000
$x_{41}$	126279557120.000	126245052416.000
$x_{42}$	378838679552.000	378827177984.000
$x_{43}$	1136516005888.000	1136512204800.000
$x_{44}$	3409547886592.000	3409546838016.000
$x_{45}$	10228643921920.000	10228643921920.000
$x_{46}$	30685932814336.000	30685932814336.000

Tabela 2: Obliczone wyrazy funkcji rekurencyjnej dla typu float przy  $c = 3$

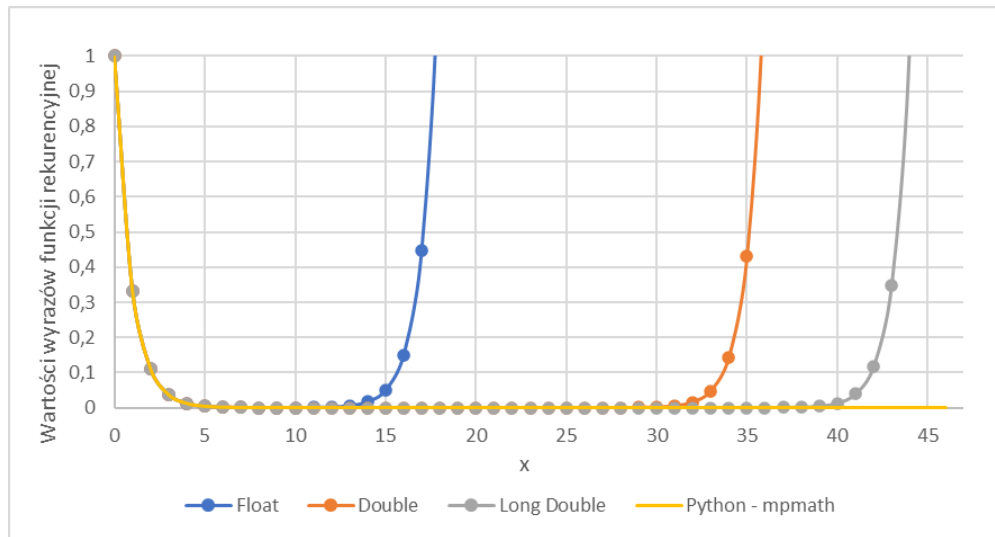
	Obliczanie $x_{45}, x_{46}$	Ponowne obliczanie $x_0, x_1$
$x_0$	1.0000000000	-8798768793.5659847260
$x_1$	0.3333333333	-2932922931.1886615753
$x_2$	0.1111111111	-977640977.0628871918
$x_3$	0.0370370370	-325880325.6876290441
$x_4$	0.0123456790	-108626775.2292096764
$x_5$	0.0041152263	-36208925.0764032230
$x_6$	0.0013717421	-12069641.6921344083
$x_7$	0.0004572474	-4023213.8973781359
$x_8$	0.0001524158	-1341071.2991260453
$x_9$	0.0000508053	-447023.7663753484
$x_{10}$	0.0000169351	-149007.9221251161
$x_{11}$	0.0000056450	-49669.3073750387
$x_{12}$	0.0000018817	-16556.4357916796
$x_{13}$	0.0000006272	-5518.8119305598
$x_{14}$	0.0000002091	-1839.6039768532
$x_{15}$	0.0000000698	-613.2013256176
$x_{16}$	0.0000000236	-204.4004418722
$x_{17}$	0.0000000089	-68.1334806231
$x_{18}$	0.0000000059	-22.7111602047
$x_{19}$	0.0000000109	-7.5703867260
$x_{20}$	0.0000000303	-2.5234622153
$x_{21}$	0.0000000902	-0.8411539917
$x_{22}$	0.0000002702	-0.2803844237
$x_{23}$	0.0000008106	-0.0934607540
$x_{24}$	0.0000024318	-0.0311514231
$x_{25}$	0.0000072954	-0.0103773229
$x_{26}$	0.0000218862	-0.0034396532
$x_{27}$	0.0000656586	-0.0010881879
$x_{28}$	0.0001969758	-0.0001876397
$x_{29}$	0.0005909273	0.0004627221
$x_{30}$	0.0017727819	0.0017300468
$x_{31}$	0.0053183457	0.0053041007
$x_{32}$	0.0159550371	0.0159502887
$x_{33}$	0.0478651112	0.0478635284
$x_{34}$	0.1435953335	0.1435948059
$x_{35}$	0.4307860006	0.4307858247
$x_{36}$	1.2923580017	1.2923579431
$x_{37}$	3.8770740051	3.8770739856
$x_{38}$	11.6312220154	11.6312220089
$x_{39}$	34.8936660463	34.8936660441
$x_{40}$	104.6809981389	104.6809981381
$x_{41}$	314.0429944166	314.0429944164
$x_{42}$	942.1289832498	942.1289832497
$x_{43}$	2826.3869497494	2826.3869497494
$x_{44}$	8479.1608492482	8479.1608492482
$x_{45}$	25437.4825477447	25437.4825477447
$x_{46}$	76312.4476432342	76312.4476432342

Tabela 3: Obliczone wyrazy funkcji rekurencyjnej dla typu double przy  $c = 3$

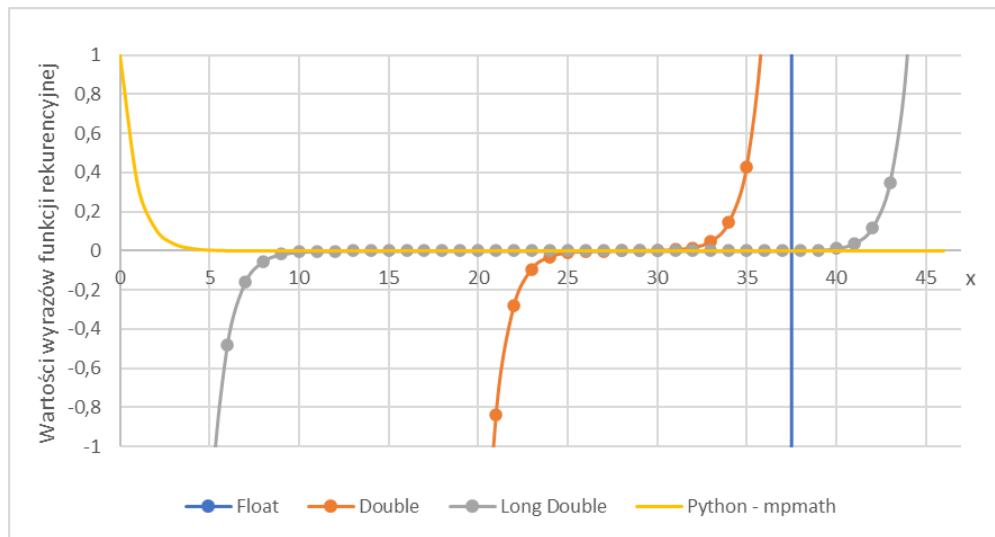
	Obliczanie $x_{45}, x_{46}$	Ponowne obliczanie $x_0, x_1$
$x_0$	1.0000000000	-351.5101364134
$x_1$	0.3333333333	-117.1700454711
$x_2$	0.1111111111	-39.0566818237
$x_3$	0.0370370370	-13.0188939412
$x_4$	0.0123456790	-4.3396313137
$x_5$	0.0041152263	-1.4465437712
$x_6$	0.0013717421	-0.4821812571
$x_7$	0.0004572474	-0.1607270857
$x_8$	0.0001524158	-0.0535756952
$x_9$	0.0000508053	-0.0178585651
$x_{10}$	0.0000169351	-0.0059528550
$x_{11}$	0.0000056450	-0.0019842850
$x_{12}$	0.0000018817	-0.0006614283
$x_{13}$	0.0000006272	-0.0002204761
$x_{14}$	0.0000002091	-0.0000734920
$x_{15}$	0.0000000697	-0.0000244973
$x_{16}$	0.0000000232	-0.0000081658
$x_{17}$	0.0000000077	-0.0000027219
$x_{18}$	0.0000000026	-0.0000009073
$x_{19}$	0.0000000009	-0.0000003024
$x_{20}$	0.0000000003	-0.0000001008
$x_{21}$	0.0000000001	-0.0000000336
$x_{22}$	0.0000000001	-0.0000000112
$x_{23}$	0.0000000001	-0.0000000036
$x_{24}$	0.0000000003	-0.0000000009
$x_{25}$	0.0000000009	0.0000000005
$x_{26}$	0.0000000027	0.0000000026
$x_{27}$	0.0000000081	0.0000000080
$x_{28}$	0.0000000243	0.0000000243
$x_{29}$	0.0000000728	0.0000000728
$x_{30}$	0.0000002185	0.0000002185
$x_{31}$	0.0000006555	0.0000006555
$x_{32}$	0.0000019666	0.0000019666
$x_{33}$	0.0000058999	0.0000058999
$x_{34}$	0.0000176998	0.0000176998
$x_{35}$	0.0000530995	0.0000530995
$x_{36}$	0.0001592984	0.0001592984
$x_{37}$	0.0004778951	0.0004778951
$x_{38}$	0.0014336854	0.0014336854
$x_{39}$	0.0043010561	0.0043010561
$x_{40}$	0.0129031684	0.0129031684
$x_{41}$	0.0387095052	0.0387095052
$x_{42}$	0.1161285157	0.1161285157
$x_{43}$	0.3483855472	0.3483855472
$x_{44}$	1.0451566417	1.0451566417
$x_{45}$	3.1354699250	3.1354699250
$x_{46}$	9.4064097751	9.4064097751

Tabela 4: Obliczone wyrazy funkcji rekurencyjnej dla typu long double przy  $c = 3$

## 5.2 Opracowanie wyników dla $c = 3$



Wykres 1: Porównanie obliczonych wyrazów funkcji rekurencyjnej dla  $k = 45$  z poprawnymi wartościami dla  $c = 3$



Wykres 2: Porównanie obliczonych wyrazów funkcji rekurencyjnej przy ponownym wyznaczaniu  $x_0$  i  $x_1$  z poprawnymi wartościami dla  $c = 3$

Jak można zobaczyć z tabel 2, 3 i 4 wyniki ponownego obliczenia  $x_0$  i  $x_1$  znacznie odbiegają od wartości 1 i  $\frac{1}{3}$ . Wynika to z faktu, iż dla  $c = 3$  kolejne wyrazy funkcji rekurencyjnej przyjmują postać  $\frac{1}{3^k}$  przez co szybko wartość ta zaczyna zbiegać do 0. Na wykresach 1 oraz 2 możemy zobaczyć porównanie z poprawnie obliczonymi wyrazami za pomocą biblioteki mpmath w języku python, która wspiera operacje na liczbach zmiennoprzecinkowych o dowolnej precyzji. Wywnioskować z niej możemy, że korzystając z podanych typów danych prędzej czy później zachodzi błąd zaokrąglania co prowadzi do utraty dokładności. Wykresy dodatkowo ukazują różnice w specyfikacji typów, ponieważ pierwszy dokładność traci float, następnie double a na końcu long double.

### 5.3 Liczba $c = 2$

	Obliczanie $x_{45}, x_{46}$	Ponowne obliczanie $x_0, x_1$
$x_0$	1.0	nan
$x_1$	0.5	nan
$x_2$	1.5	nan
$x_3$	7.0	nan
$x_4$	33.5	nan
$x_5$	160.5	nan
$x_6$	769.0	nan
$x_7$	3684.5	nan
$x_8$	17653.5	nan
$x_9$	84583.0	nan
$x_{10}$	405261.5	nan
$x_{11}$	1941724.5	nan
$x_{12}$	9303361.0	nan
$x_{13}$	44575080.0	nan
$x_{14}$	213572032.0	nan
$x_{15}$	1023285056.0	nan
$x_{16}$	4902853120.0	nan
$x_{17}$	23490979840.0	nan
$x_{18}$	112552042496.0	nan
$x_{19}$	539269234688.0	-inf
$x_{20}$	2583794024448.0	-inf
$x_{21}$	12379701116928.0	-170529551039559953916965045683508740096.0
$x_{22}$	59314710511616.0	-35591587584295153998695454517139341312.0
$x_{23}$	284193858781184.0	-7428397023120617902347438875813609472.0
$x_{24}$	1361654554034176.0	-1550396739026310370398363926489202688.0
$x_{25}$	6524078944944128.0	-323586632396852692512211959860428800.0
$x_{26}$	31258739768033280.0	-67536462572034349294864669584916480.0
$x_{27}$	149769615600254976.0	-14095680463319053962111388064153600.0
$x_{28}$	717589323200856064.0	-2941939744560920515692270735851520.0
$x_{29}$	3438176983224156160.0	-614018607656184665363167930482688.0
$x_{30}$	16473295524200448000.0	-128153148648904457368067951820800.0
$x_{31}$	78928301462411804672.0	-26747118663376146872363382734848.0
$x_{32}$	378168210688346947584.0	-5582448899216645644951073325056.0
$x_{33}$	1811912690406671777792.0	-1165124926012716641420102860800.0
$x_{34}$	8681395346898128207872.0	-243175730846937562149440978944.0
$x_{35}$	41595066014408806236160.0	-50753690443039306369940324352.0
$x_{36}$	199293939228745530343424.0	-10592721368258969700260642816.0
$x_{37}$	954874666158115864444928.0	-2209916398255542131362889728.0
$x_{38}$	4575079319504239753953280.0	-456860623018740956553805824.0
$x_{39}$	21920521931363082905321472.0	-74386716838162651406139392.0
$x_{40}$	105027533507845312441483264.0	84927038827927699523108864.0
$x_{41}$	503217143302020470088400896.0	499021910977801149021683712.0
$x_{42}$	2411058219895745185419624448.0	2410182516061078045585309696.0
$x_{43}$	11552073661028800277656895488.0	11551890669327589078904864768.0
$x_{44}$	55349308609508730306100723712.0	55349270830576867348939014144.0
$x_{45}$	265194463483556747665790205952.0	265194463483556747665790205952.0
$x_{46}$	1270623046587206870980012015616.0	1270623046587206870980012015616.0

Tabela 5: Obliczone wyrazy funkcji rekurencyjnej dla typu float przy  $c = 2$

	Obliczanie $x_{45}, x_{46}$	Ponowne obliczanie $x_0, x_1$
$x_0$	1.0	-77040105326218382941974731688109606193594368.0
$x_1$	0.5	-16079206213162800143285597701722345469640704.0
$x_2$	1.5	-3355925739595610965783040750910609209425920.0
$x_3$	7.0	-700422484815253292947061856777891315974144.0
$x_4$	33.5	-146186684480655421581016077642580189249536.0
$x_5$	160.5	-30510937588023737586766076098742449078272.0
$x_6$	769.0	-6368003459463273606369220538907104378880.0
$x_7$	3684.5	-1329079709292631805121573662250894360576.0
$x_8$	17653.5	-277395086999885532575443361218852552704.0
$x_9$	84583.0	-57895725706795763308313486450464129024.0
$x_{10}$	405261.5	-12083541534093265076658139554887237632.0
$x_{11}$	1941724.5	-2521981963670562074977211323972059136.0
$x_{12}$	9303361.0	-526368284259545150653964475296645120.0
$x_{13}$	44575080.5	-109859457627163752079587347349372928.0
$x_{14}$	213572041.5	-22929003876273586685542169313280000.0
$x_{15}$	1023285127.0	-4785561754204176736437480789639168.0
$x_{16}$	4902853593.5	-998804894747294834917413497077760.0
$x_{17}$	23490982840.5	-208462719532297474178383714713600.0
$x_{18}$	112552060609.0	-43508702914192454909711783821312.0
$x_{19}$	539269320204.5	-9080795038664802621975018078208.0
$x_{20}$	2583794540413.5	-1895272279131555385413539463168.0
$x_{21}$	12379703381863.0	-395566356992975290255097724928.0
$x_{22}$	59314722368901.5	-8255950583321347336925872128.0
$x_{23}$	284193908462644.5	-17231172173631446429531635712.0
$x_{24}$	1361654819944321.0	-3596355034835884810732306432.0
$x_{25}$	6524080191258960.0	-750603000547977624129896448.0
$x_{26}$	31258746136350480.0	-156659967904003309917175808.0
$x_{27}$	149769650490493440.0	-32696838972038925455982592.0
$x_{28}$	717589506316116736.0	-6824226956191317362737152.0
$x_{29}$	3438177881090090496.0	-1424295808917661357703168.0
$x_{30}$	16473299899134337024.0	-297252088396989425778688.0
$x_{31}$	78928321614581596160.0	-61964633067285771190272.0
$x_{32}$	378168308173773668352.0	-12571076939439430172672.0
$x_{33}$	1811913219254286811136.0	-890751629911379673088.0
$x_{34}$	8681397788097660321792.0	8117318789882531807232.0
$x_{35}$	41595075721234015584256.0	41477345579324038709248.0
$x_{36}$	199293980818072424939520.0	199269409106737661739008.0
$x_{37}$	954874828369128050393088.0	954869699954364269985792.0
$x_{38}$	4575080161027567827025920.0	4575079090665083688189952.0
$x_{39}$	21920525976768714305961984.0	21920525753371054170963968.0
$x_{40}$	105027549722815991354753024.0	105027549676190187166629888.0
$x_{41}$	503217222637311246762770432.0	503217222627579881662185472.0
$x_{42}$	2411058563463740019120799744.0	2411058563461709221144297472.0
$x_{43}$	11552075594681390635547623424.0	11552075594680966224059301888.0
$x_{44}$	55349319409943209860082434048.0	55349319409943121899152211968.0
$x_{45}$	265194521455034643271701757952.0	265194521455034643271701757952.0
$x_{46}$	1270623287865229953721868222464.0	1270623287865229953721868222464.0

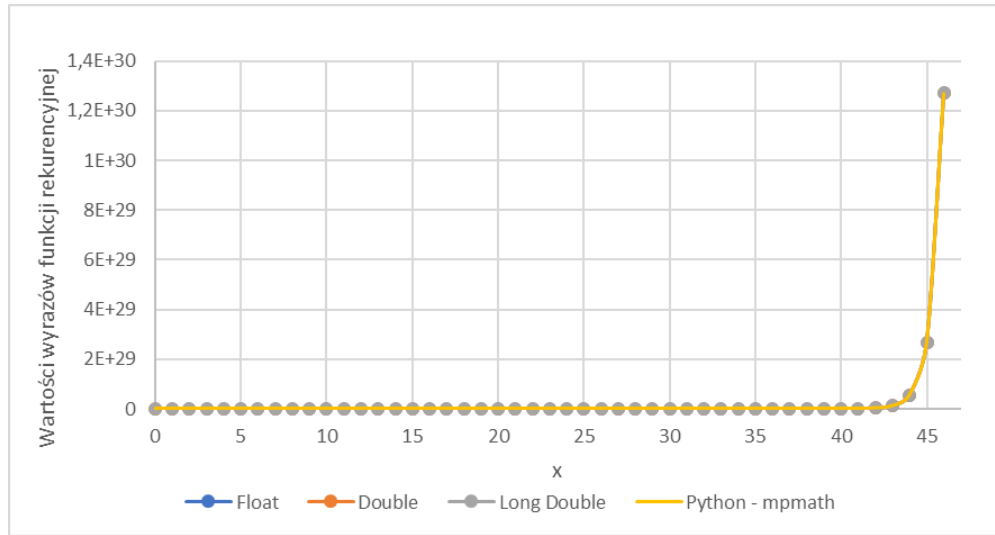
Tabela 6: Obliczone wyrazy funkcji rekurencyjnej dla typu double przy  $c = 2$

	Obliczanie $x_{45}, x_{46}$	Ponowne obliczanie $x_0, x_1$
$x_0$	1.0	-17351611458615731056908025604606504468480.0
$x_1$	0.5	-3621492177254477423108913495774291034112.0
$x_2$	1.5	-755849427656656058562754897970112495616.0
$x_3$	7.0	-157754961028802869806318086481673977856.0
$x_4$	33.5	-32925377487358290473447220456684781568.0
$x_5$	160.5	-6871926407988582569564927086301282304.0
$x_6$	769.0	-1434254552584622375962682043656044544.0
$x_7$	3684.5	-299346354934529310266497530488422400.0
$x_8$	17653.5	-62477222088024175450870402078736384.0
$x_9$	84583.0	-13039755505591566988980379812102144.0
$x_{10}$	405261.5	-2721555439933659490653797261246464.0
$x_{11}$	1941724.5	-568021694076730464605265842929664.0
$x_{12}$	9303361.0	-118553030449992832566045999890432.0
$x_{13}$	44575080.5	-24743458173233698216168063500288.0
$x_{14}$	213572041.5	-5164260416175658515344073424896.0
$x_{15}$	1023285127.0	-1077843907644594359727669903360.0
$x_{16}$	4902853593.5	-224959122047313283740952690688.0
$x_{17}$	23490982840.5	-46951702591972058942733811712.0
$x_{18}$	112552060609.0	-9799390912547010972716367872.0
$x_{19}$	539269320204.5	-2045251970762995920848027648.0
$x_{20}$	2583794540413.5	-426868941267968631523770368.0
$x_{21}$	12379703381863.0	-89092735576847236770824192.0
$x_{22}$	59314722368901.5	-18594736616267552330350592.0
$x_{23}$	284193908462644.5	-3880947504490524880928768.0
$x_{24}$	1361654819944321.0	-810000906185072074293248.0
$x_{25}$	6524080191258960.5	-169057026434835490537472.0
$x_{26}$	31258746136350481.5	-35284225989105378394112.0
$x_{27}$	149769650490493447.0	-7364103510691401433088.0
$x_{28}$	717589506316116753.5	-1536291564351628771328.0
$x_{29}$	3438177881090090320.5	-317354311066742423552.0
$x_{30}$	16473299899134334848.0	-50479990982083346432.0
$x_{31}$	78928321614581583920.0	64954356156325691392.0
$x_{32}$	378168308173773584768.0	375251771763711803392.0
$x_{33}$	1811913219254286339968.0	1811304502662233325568.0
$x_{34}$	8681397788097658115072.0	8681270741547454824448.0
$x_{35}$	41595075721234004234240.0	41595049205075040796672.0
$x_{36}$	199293980818072363057152.0	199293975283827749158912.0
$x_{37}$	954874828369127811055616.0	954874827214063704997888.0
$x_{38}$	4575080161027566692204544.0	4575080160786490775830528.0
$x_{39}$	21920525976768705648918528.0	21920525976718390174154752.0
$x_{40}$	105027549722815961550028800.0	105027549722805460094943232.0
$x_{41}$	503217222637311102109614080.0	503217222637308910300561408.0
$x_{42}$	2411058563463739549090316288.0	2411058563463739091407863808.0
$x_{43}$	11552075594681386643375521792.0	11552075594681386546738757632.0
$x_{44}$	55349319409943193663760760832.0	55349319409943193642285924352.0
$x_{45}$	265194521455034581664690864128.0	265194521455034581664690864128.0
$x_{46}$	1270623287865229714715528134656.0	1270623287865229714715528134656.0

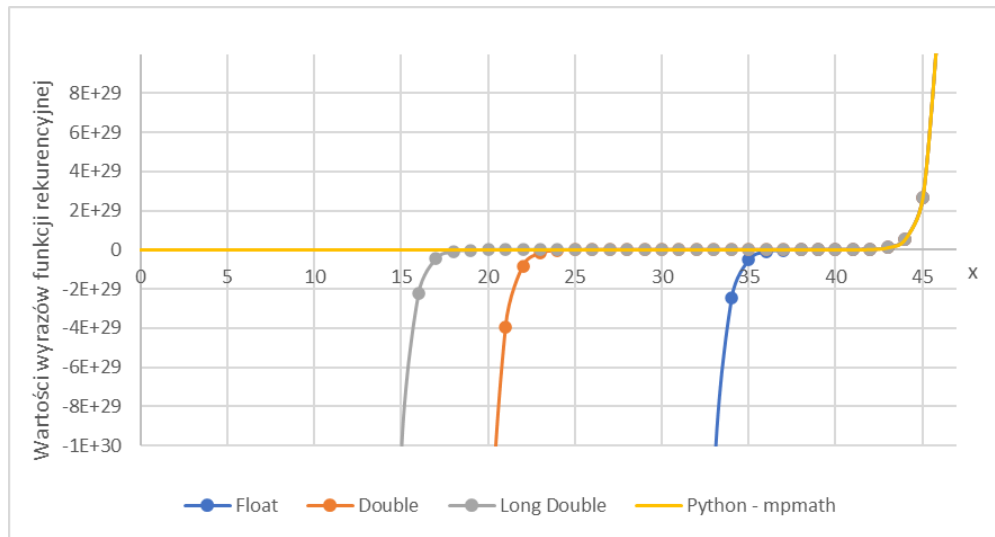
Tabela 7: Obliczone wyrazy funkcji rekurencyjnej dla typu long double przy  $c = 2$



## 5.4 Opracowanie wyników dla $c = 2$



Wykres 3: Porównanie obliczonych wyrazów funkcji rekurencyjnej dla  $k = 45$  z poprawnymi wartościami dla  $c = 2$



Wykres 4: Porównanie obliczonych wyrazów funkcji rekurencyjnej przy ponownym wyznaczaniu  $x_0$  i  $x_1$  z poprawnymi wartościami dla  $c = 2$

Jak można zauważyć w tabelach 4, 5 i 6 znowu nie udało się poprawnie obliczyć ponownie wartości  $x_0$  oraz  $x_1$ . Dla typu float doszło nawet to zwrócenia wartości  $-\text{inf}$  a następnie  $\text{nan}$  (not a number) czyli wartości numerycznej oznaczającej niezdefiniowaną lub niereprezentowalną wielkość. Jednak co warto podkreślić, początkowe cyfry wyrazów  $x_{45}$  i  $x_{46}$  są podobne oraz w zależności od dokładności danego typu pokrywają się do pewnego momentu z wartością rzeczywistą. Mimo wszystko kiedy chcemy ponownie wyznaczyć wartości początkowe, typy znowu tracą dokładność w analogiczny sposób jak dla  $c = 3$ .

## 5.5 Liczba $c = 20$

	Obliczanie $x_{45}, x_{46}$	Ponowne obliczanie $x_0, x_1$
$x_0$	1.0000000000	0.9999998808
$x_1$	0.0500000007	0.0499999523
$x_2$	-0.9750000238	-0.9749999046
$x_3$	-0.5375000238	-0.5374999046
$x_4$	0.7062500119	0.7062499523
$x_5$	0.8906250000	0.8906248808
$x_6$	-0.2609375119	-0.2609374821
$x_7$	-1.0210937262	-1.0210936069
$x_8$	-0.2496093512	-0.2496093214
$x_9$	0.8962890506	0.8962889314
$x_{10}$	0.6977539062	0.6977537870
$x_{11}$	-0.5474120975	-0.5474120378
$x_{12}$	-0.9714599848	-0.9714598060
$x_{13}$	0.0616821051	0.0616821051
$x_{14}$	1.0023009777	1.0023008585
$x_{15}$	0.4394683838	0.4394683242
$x_{16}$	-0.7825667858	-0.7825666666
$x_{17}$	-0.8307517767	-0.8307516575
$x_{18}$	0.3671908975	0.3671908081
$x_{19}$	1.0143471956	1.0143470764
$x_{20}$	0.1399827003	0.1399827302
$x_{21}$	-0.9443558455	-0.9443556666
$x_{22}$	-0.6121606231	-0.6121605635
$x_{23}$	0.6382755041	0.6382753849
$x_{24}$	0.9312983751	0.9312982559
$x_{25}$	-0.1726263165	-0.1726262271
$x_{26}$	-1.0176115036	-1.0176113844
$x_{27}$	-0.3361794353	-0.3361794651
$x_{28}$	0.8495217562	0.8495216966
$x_{29}$	0.7609403133	0.7609403133
$x_{30}$	-0.4690515995	-0.4690515399
$x_{31}$	-0.9954661131	-0.9954661131
$x_{32}$	-0.0286814570	-0.0286815166
$x_{33}$	0.9811253548	0.9811253548
$x_{34}$	0.5192441344	0.5192441940
$x_{35}$	-0.7215032578	-0.7215032578
$x_{36}$	-0.8799957633	-0.8799958229
$x_{37}$	0.2815053761	0.2815053463
$x_{38}$	1.0207484961	1.0207484961
$x_{39}$	0.2288688719	0.2288689017
$x_{40}$	-0.9063140750	-0.9063140154
$x_{41}$	-0.6820259094	-0.6820259094
$x_{42}$	0.5653011203	0.5653010607
$x_{43}$	0.9646764994	0.9646764398
$x_{44}$	-0.0829628706	-0.0829628706
$x_{45}$	-1.0061578751	-1.0061578751
$x_{46}$	-0.4201160669	-0.4201160669

Tabela 8: Obliczone wyrazy funkcji rekurencyjnej dla typu float przy  $c = 20$

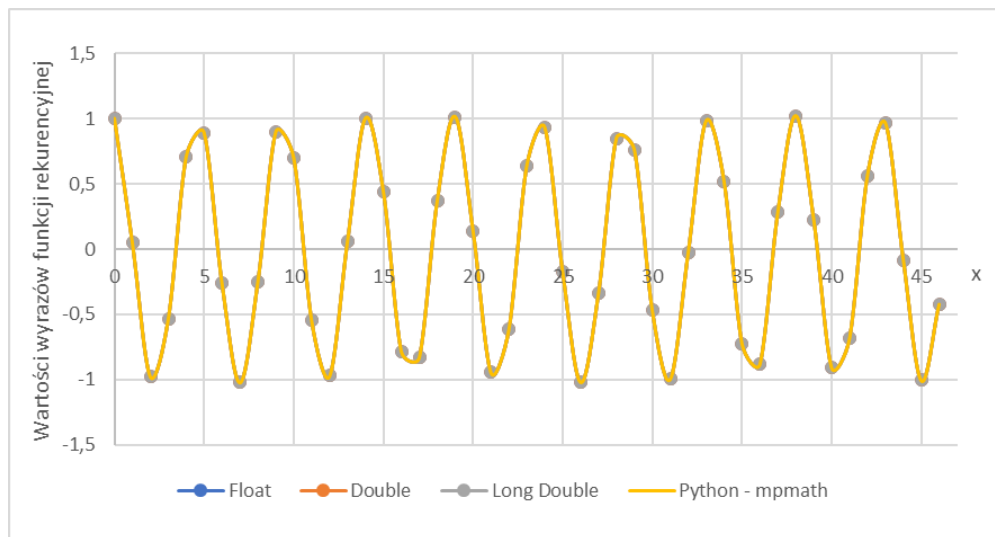
	Obliczanie $x_{45}, x_{46}$	Ponowne obliczanie $x_0, x_1$
$x_0$	1.0000000000	1.0000000000
$x_1$	0.0500000000	0.0500000000
$x_2$	-0.9750000000	-0.9750000000
$x_3$	-0.5375000000	-0.5375000000
$x_4$	0.7062500000	0.7062500000
$x_5$	0.8906250000	0.8906250000
$x_6$	-0.2609375000	-0.2609375000
$x_7$	-1.0210937500	-1.0210937500
$x_8$	-0.2496093750	-0.2496093750
$x_9$	0.8962890625	0.8962890625
$x_{10}$	0.6977539062	0.6977539063
$x_{11}$	-0.5474121094	-0.5474121094
$x_{12}$	-0.9714599609	-0.9714599609
$x_{13}$	0.0616821289	0.0616821289
$x_{14}$	1.0023010254	1.0023010254
$x_{15}$	0.4394683838	0.4394683838
$x_{16}$	-0.7825668335	-0.7825668335
$x_{17}$	-0.8307518005	-0.8307518005
$x_{18}$	0.3671909332	0.3671909332
$x_{19}$	1.0143472672	1.0143472672
$x_{20}$	0.1399827003	0.1399827003
$x_{21}$	-0.9443559170	-0.9443559170
$x_{22}$	-0.6121606588	-0.6121606588
$x_{23}$	0.6382755876	0.6382755876
$x_{24}$	0.9312984526	0.9312984526
$x_{25}$	-0.1726263613	-0.1726263613
$x_{26}$	-1.0176116332	-1.0176116332
$x_{27}$	-0.3361794554	-0.3361794554
$x_{28}$	0.8495219056	0.8495219056
$x_{29}$	0.7609404081	0.7609404081
$x_{30}$	-0.4690517015	-0.4690517015
$x_{31}$	-0.9954662589	-0.9954662589
$x_{32}$	-0.0286814280	-0.0286814280
$x_{33}$	0.9811255449	0.9811255449
$x_{34}$	0.5192442004	0.5192442004
$x_{35}$	-0.7215034447	-0.7215034447
$x_{36}$	-0.8799959228	-0.8799959228
$x_{37}$	0.2815054833	0.2815054833
$x_{38}$	1.0207486644	1.0207486644
$x_{39}$	0.2288688489	0.2288688489
$x_{40}$	-0.9063142400	-0.9063142400
$x_{41}$	-0.6820259689	-0.6820259689
$x_{42}$	0.5653012555	0.5653012555
$x_{43}$	0.9646765967	0.9646765967
$x_{44}$	-0.0829629572	-0.0829629572
$x_{45}$	-1.0061580753	-1.0061580753
$x_{46}$	-0.4201160804	-0.4201160804

Tabela 9: Obliczone wyrazy funkcji rekurencyjnej dla typu double przy  $c = 20$

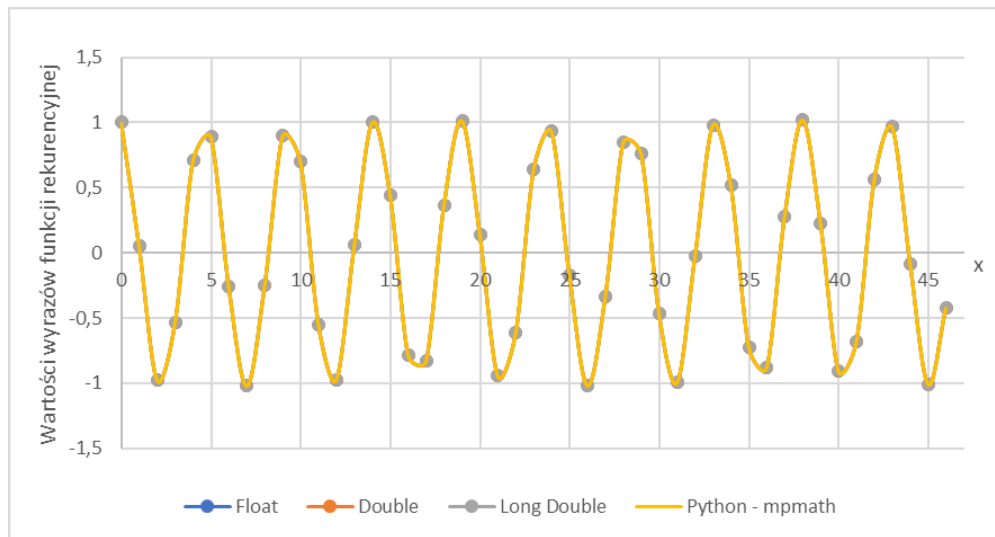
	Obliczanie $x_{45}, x_{46}$	Ponowne obliczanie $x_0, x_1$
$x_0$	1.0000000000	1.0000000000
$x_1$	0.0500000000	0.0500000000
$x_2$	-0.9750000000	-0.9750000000
$x_3$	-0.5375000000	-0.5375000000
$x_4$	0.7062500000	0.7062500000
$x_5$	0.8906250000	0.8906250000
$x_6$	-0.2609375000	-0.2609375000
$x_7$	-1.0210937500	-1.0210937500
$x_8$	-0.2496093750	-0.2496093750
$x_9$	0.8962890625	0.8962890625
$x_{10}$	0.6977539062	0.6977539062
$x_{11}$	-0.5474121094	-0.5474121094
$x_{12}$	-0.9714599609	-0.9714599609
$x_{13}$	0.0616821289	0.0616821289
$x_{14}$	1.0023010254	1.0023010254
$x_{15}$	0.4394683838	0.4394683838
$x_{16}$	-0.7825668335	-0.7825668335
$x_{17}$	-0.8307518005	-0.8307518005
$x_{18}$	0.3671909332	0.3671909332
$x_{19}$	1.0143472672	1.0143472672
$x_{20}$	0.1399827003	0.1399827003
$x_{21}$	-0.9443559170	-0.9443559170
$x_{22}$	-0.6121606588	-0.6121606588
$x_{23}$	0.6382755876	0.6382755876
$x_{24}$	0.9312984526	0.9312984526
$x_{25}$	-0.1726263613	-0.1726263613
$x_{26}$	-1.0176116332	-1.0176116332
$x_{27}$	-0.3361794554	-0.3361794554
$x_{28}$	0.8495219056	0.8495219056
$x_{29}$	0.7609404081	0.7609404081
$x_{30}$	-0.4690517015	-0.4690517015
$x_{31}$	-0.9954662589	-0.9954662589
$x_{32}$	-0.0286814280	-0.0286814280
$x_{33}$	0.9811255449	0.9811255449
$x_{34}$	0.5192442004	0.5192442004
$x_{35}$	-0.7215034447	-0.7215034447
$x_{36}$	-0.8799959228	-0.8799959228
$x_{37}$	0.2815054833	0.2815054833
$x_{38}$	1.0207486644	1.0207486644
$x_{39}$	0.2288688489	0.2288688489
$x_{40}$	-0.9063142400	-0.9063142400
$x_{41}$	-0.6820259689	-0.6820259689
$x_{42}$	0.5653012555	0.5653012555
$x_{43}$	0.9646765967	0.9646765967
$x_{44}$	-0.0829629572	-0.0829629572
$x_{45}$	-1.0061580753	-1.0061580753
$x_{46}$	-0.4201160804	-0.4201160804

Tabela 10: Obliczone wyrazy funkcji rekurencyjnej dla typu long double przy  $c = 20$

## 5.6 Opracowanie wyników dla $c = 20$



Wykres 5: Porównanie obliczonych wyrazów funkcji rekurencyjnej dla  $k = 45$  z poprawnymi wartościami dla  $c = 20$



Wykres 6: Porównanie obliczonych wyrazów funkcji rekurencyjnej przy ponownym wyznaczeniu  $x_0$  i  $x_1$  z poprawnymi wartościami dla  $c = 20$

Kiedy pod  $c$  podstawimy wartość 20 wyrazy podanej funkcji rekurencyjnej rozciągają się w granicach od -1 do 1. Z tego też powodu nawet jeżeli tracimy dokładność na  $x$  miejscu po przecinku nie powoduje to strasznej różnicy w wyniku końcowym. Jak możemy zauważyć na tabeli 8, przy użyciu typu float policzyliśmy  $x_0$  i  $x_1$  z błędem rzędu 7 cyfry po przecinku, a przy użyciu double i long double, bez żadnego błędu. Dodatkowo wykresy 5 i 6 pokazują, że obliczone wartości prawie idealnie pokrywają się z dokładnymi.

## 5.7 Liczba $c = 30$

	Obliczanie $x_{45}, x_{46}$	Ponowne obliczanie $x_0, x_1$
$x_0$	1.0000000000	0.9999998808
$x_1$	0.0333333351	0.0333333015
$x_2$	-0.9888888597	-0.9888888001
$x_3$	-0.3629629612	-0.3629629016
$x_4$	0.8679012060	0.8679011464
$x_5$	0.6522633433	0.6522632837
$x_6$	-0.6504800916	-0.6504800320
$x_7$	-0.8690900207	-0.8690899611
$x_8$	0.3607834280	0.3607833982
$x_9$	0.9893511534	0.9893510938
$x_{10}$	-0.0309997108	-0.0309996903
$x_{11}$	-0.9996843934	-0.9996843338
$x_{12}$	-0.3022284210	-0.3022284210
$x_{13}$	0.8989415765	0.8989415169
$x_{14}$	0.6018756032	0.6018756032
$x_{15}$	-0.6983163953	-0.6983163357
$x_{16}$	-0.8346477151	-0.8346477151
$x_{17}$	0.4201004803	0.4201004505
$x_{18}$	0.9746811986	0.9746811986
$x_{19}$	-0.0952067450	-0.0952067301
$x_{20}$	-1.0064167976	-1.0064167976
$x_{21}$	-0.2402655184	-0.2402655333
$x_{22}$	0.9263283014	0.9263283014
$x_{23}$	0.5490416288	0.5490416288
$x_{24}$	-0.7433144450	-0.7433144450
$x_{25}$	-0.7968131304	-0.7968131304
$x_{26}$	0.4777100682	0.4777100682
$x_{27}$	0.9560497999	0.9560497999
$x_{28}$	-0.1590268016	-0.1590268016
$x_{29}$	-1.0090587139	-1.0090587139
$x_{30}$	-0.1773260981	-0.1773260981
$x_{31}$	0.9499500394	0.9499499798
$x_{32}$	0.4939761162	0.4939760864
$x_{33}$	-0.7852913141	-0.7852913141
$x_{34}$	-0.7557398677	-0.7557398677
$x_{35}$	0.5333780050	0.5333780050
$x_{36}$	0.9335325360	0.9335325360
$x_{37}$	-0.2222004980	-0.2222005129
$x_{38}$	-1.0075993538	-1.0075993538
$x_{39}$	-0.1136659533	-0.1136659458
$x_{40}$	0.9697107077	0.9697107077
$x_{41}$	0.4369028509	0.4369028509
$x_{42}$	-0.8240764141	-0.8240764141
$x_{43}$	-0.7115949988	-0.7115949988
$x_{44}$	0.5868780613	0.5868780613
$x_{45}$	0.9072210193	0.9072210193
$x_{46}$	-0.2844710648	-0.2844710648

Tabela 11: Obliczone wyrazy funkcji rekurencyjnej dla typu float przy  $c = 30$

	Obliczanie $x_{45}, x_{46}$	Ponowne obliczanie $x_0, x_1$
$x_0$	1.0000000000	1.0000000000
$x_1$	0.0333333333	0.0333333333
$x_2$	-0.9888888889	-0.9888888889
$x_3$	-0.3629629630	-0.3629629630
$x_4$	0.8679012346	0.8679012346
$x_5$	0.6522633745	0.6522633745
$x_6$	-0.6504801097	-0.6504801097
$x_7$	-0.8690900777	-0.8690900777
$x_8$	0.3607834172	0.3607834172
$x_9$	0.9893512168	0.9893512168
$x_{10}$	-0.0309996782	-0.0309996782
$x_{11}$	-0.9996844429	-0.9996844429
$x_{12}$	-0.3022284694	-0.3022284694
$x_{13}$	0.8989416197	0.8989416197
$x_{14}$	0.6018756760	0.6018756760
$x_{15}$	-0.6983163944	-0.6983163944
$x_{16}$	-0.8346478074	-0.8346478074
$x_{17}$	0.4201004586	0.4201004586
$x_{18}$	0.9746812936	0.9746812936
$x_{19}$	-0.0952066941	-0.0952066941
$x_{20}$	-1.0064168583	-1.0064168583
$x_{21}$	-0.2402655921	-0.2402655921
$x_{22}$	0.9263283276	0.9263283276
$x_{23}$	0.5490417013	0.5490417013
$x_{24}$	-0.7433144272	-0.7433144272
$x_{25}$	-0.7968131770	-0.7968131770
$x_{26}$	0.4777100349	0.4777100349
$x_{27}$	0.9560498553	0.9560498553
$x_{28}$	-0.1590267498	-0.1590267498
$x_{29}$	-1.0090587719	-1.0090587719
$x_{30}$	-0.1773261742	-0.1773261742
$x_{31}$	0.9499500472	0.9499500472
$x_{32}$	0.4939761899	0.4939761899
$x_{33}$	-0.7852913172	-0.7852913172
$x_{34}$	-0.7557399623	-0.7557399623
$x_{35}$	0.5333779964	0.5333779964
$x_{36}$	0.9335326278	0.9335326278
$x_{37}$	-0.2222004538	-0.2222004538
$x_{38}$	-1.0075994457	-1.0075994457
$x_{39}$	-0.1136660281	-0.1136660281
$x_{40}$	0.9697107697	0.9697107697
$x_{41}$	0.4369029513	0.4369029513
$x_{42}$	-0.8240764526	-0.8240764526
$x_{43}$	-0.7115951022	-0.7115951022
$x_{44}$	0.5868780852	0.5868780852
$x_{45}$	0.9072211306	0.9072211306
$x_{46}$	-0.2844710417	-0.2844710417

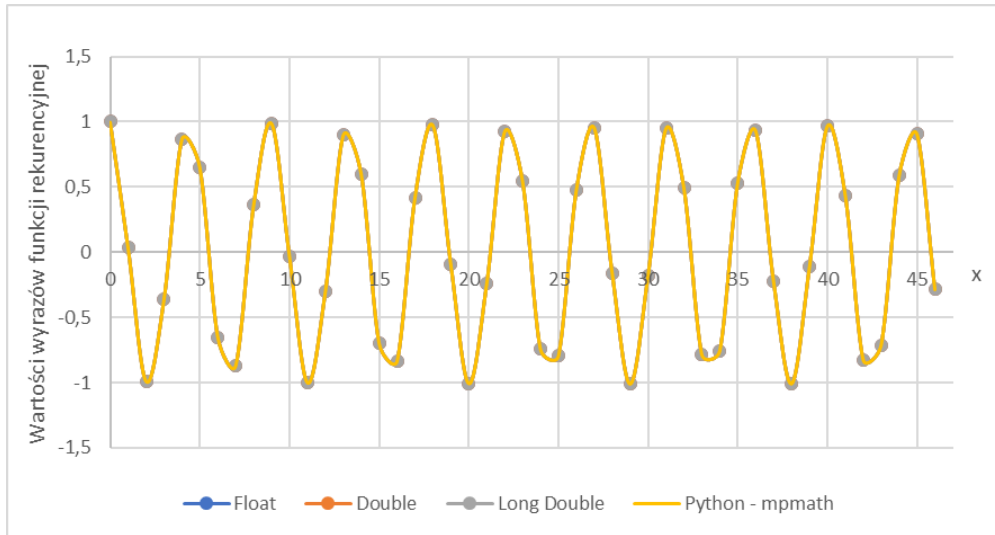
Tabela 12: Obliczone wyrazy funkcji rekurencyjnej dla typu double przy  $c = 30$

	Obliczanie $x_{45}, x_{46}$	Ponowne obliczanie $x_0, x_1$
$x_0$	1.0000000000	1.0000000000
$x_1$	0.0333333333	0.0333333333
$x_2$	-0.9888888889	-0.9888888889
$x_3$	-0.3629629630	-0.3629629630
$x_4$	0.8679012346	0.8679012346
$x_5$	0.6522633745	0.6522633745
$x_6$	-0.6504801097	-0.6504801097
$x_7$	-0.8690900777	-0.8690900777
$x_8$	0.3607834172	0.3607834172
$x_9$	0.9893512168	0.9893512168
$x_{10}$	-0.0309996782	-0.0309996782
$x_{11}$	-0.9996844429	-0.9996844429
$x_{12}$	-0.3022284694	-0.3022284694
$x_{13}$	0.8989416197	0.8989416197
$x_{14}$	0.6018756760	0.6018756760
$x_{15}$	-0.6983163944	-0.6983163944
$x_{16}$	-0.8346478074	-0.8346478074
$x_{17}$	0.4201004586	0.4201004586
$x_{18}$	0.9746812936	0.9746812936
$x_{19}$	-0.0952066941	-0.0952066941
$x_{20}$	-1.0064168583	-1.0064168583
$x_{21}$	-0.2402655921	-0.2402655921
$x_{22}$	0.9263283276	0.9263283276
$x_{23}$	0.5490417013	0.5490417013
$x_{24}$	-0.7433144272	-0.7433144272
$x_{25}$	-0.7968131770	-0.7968131770
$x_{26}$	0.4777100349	0.4777100349
$x_{27}$	0.9560498553	0.9560498553
$x_{28}$	-0.1590267498	-0.1590267498
$x_{29}$	-1.0090587719	-1.0090587719
$x_{30}$	-0.1773261742	-0.1773261742
$x_{31}$	0.9499500472	0.9499500472
$x_{32}$	0.4939761899	0.4939761899
$x_{33}$	-0.7852913172	-0.7852913172
$x_{34}$	-0.7557399623	-0.7557399623
$x_{35}$	0.5333779964	0.5333779964
$x_{36}$	0.9335326278	0.9335326278
$x_{37}$	-0.2222004538	-0.2222004538
$x_{38}$	-1.0075994457	-1.0075994457
$x_{39}$	-0.1136660281	-0.1136660281
$x_{40}$	0.9697107697	0.9697107697
$x_{41}$	0.4369029513	0.4369029513
$x_{42}$	-0.8240764526	-0.8240764526
$x_{43}$	-0.7115951022	-0.7115951022
$x_{44}$	0.5868780852	0.5868780852
$x_{45}$	0.9072211306	0.9072211306
$x_{46}$	-0.2844710417	-0.2844710417

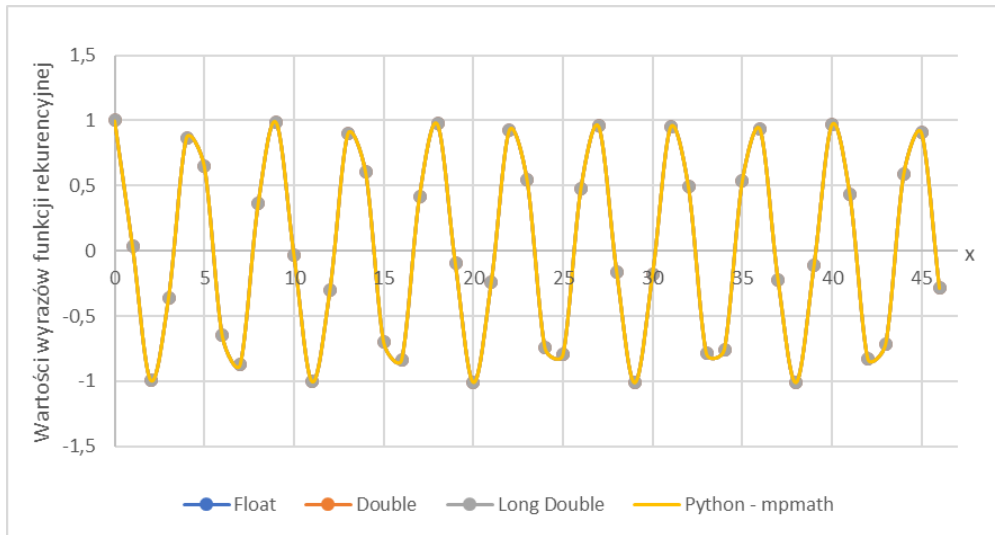
Tabela 13: Obliczone wyrazy funkcji rekurencyjnej dla typu long double przy  $c = 30$



## 5.8 Opracowanie wyników dla $c = 30$



Wykres 7: Porównanie obliczonych wyrazów funkcji rekurencyjnej dla  $k = 45$  z poprawnymi wartościami dla  $c = 30$



Wykres 8: Porównanie obliczonych wyrazów funkcji rekurencyjnej przy ponownym wyznaczeniu  $x_0$  i  $x_1$  z poprawnymi wartościami dla  $c = 30$

Podobnie jak w przypadku poprzednim obliczone wyrazy funkcji rekurencyjnej dla typów double i long double dokładnie pokrywają się z wartościami początkowymi, a dla typu float błąd jest rzędu 7 cyfry po przecinku.

## 6 Wnioski

Po wykonaniu wszystkich obliczeń dla 3 typów: float, double i long double pierwszym wnioskiem, który się od razu rzuca w oczy jest fakt, że dobór odpowiedniego narzędzia do obliczeń na liczbach zmiennoprzecinkowych ma znaczenie i może znacznie wpłynąć na wyniki naszych obliczeń.

Jak się okazało dla  $c$  równego 2 oraz 3 żaden z typów nie poradził sobie z poprawnym ponownym wyznaczeniem wartości  $x_0$  i  $x_1$ . Wszystkie typy prędzej czy później traciły dokładność w obliczeniach, przez co wyniki końcowe skrajnie różniły się od poprawnych.

Za to dla  $c$  równego 20 oraz 30 dostaliśmy wartości prawie dokładne, na co wpływ miał fakt, iż wyrazy funkcji przy takim parametrze oscylowały w granicach od -1 do 1 i nigdy nie uciekały do zera lub nieskończoności, jak w poprzednich przypadkach. Dzięki temu nawet jeżeli gubiliśmy dokładność na  $x$  miejscu po przecinku, to nie wpływała ona na tyle by zwrócić błędny wynik.

Ciekawym spostrzeżeniem jest fakt, że dla  $c = 3$  oraz  $c = 30$  drugi wyraz ciągu był liczbą o rozwinięciu nieskończonym, co powinno od samego początku utrudnić poprawne kolejne wyliczenia. Mimo, że dla pierwszej wartości wyniki okazały się nierealne, dla drugiej dostaliśmy poprawne. Wpływ jednak na to miał wzór funkcji rekurencyjnej jaki przyjmował dla danej stałej  $c$  oraz dokładność typów zmiennych.

Typ float, jak się można było spodziewać, najgorzej poradził sobie z przedstawionymi obliczeniami. Łatwo można było spostrzec, że obliczenia wykorzystujące ten typ danych najszybciej zaczynały tracić dokładność obliczeniową. Kiedy jednak dla  $c$  równego 20 lub 30 udawało się obliczyć ponownie wartości początkowe to różniły się one na 7 miejscu po przecinku.

Korzystając z typu double i long double uzyskaliśmy trochę lepsze wyniki. Specyfikacja tych typów nie pozwoliła na dokładne obliczenie wyrazów funkcji dla  $c$  równego 2 czy 3, za to dla 20 lub 30 wartości dla tych dwóch typów pokrywały się ze sobą (przynajmniej dla 10 cyfr znaczących) oraz udało się wyznaczyć dokładne poprawne wartości początkowe.