

# Interpolacja funkcji wielomianami w postaci Lagrange'a i Newtona.

Wiktor Gut, 411 761

## Środowisko

Program wykonany w systemie Windows x64, na procesorze AMD Ryzen 5 3500U with Radeon Vega Mobile Gfx, za pomocą programu Pycharm w języku Python.

## Treść zadania:

Dla funkcji

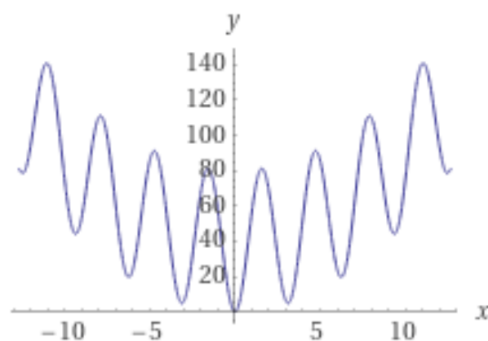
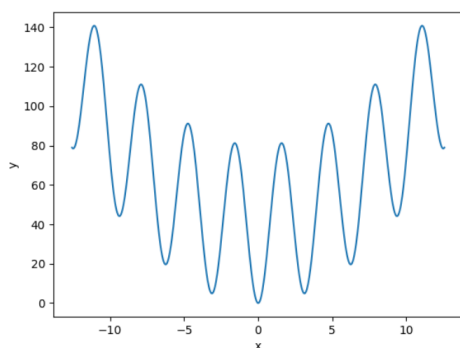
$$f(x) = 40 + x^2/2 - 40\cos(2x)$$

na przedziale  $[-4\pi, 4\pi]$

wyznacz dla zagadnienia Lagrange'a wielomian interpolujący w postaci Lagrange'a i Newtona. Interpolację przeprowadź dla różnej liczby węzłów (np.  $n = 3, 4, 5, 7, 10, 15, 20$ ). Dla każdego przypadku interpolacji porównaj wyniki otrzymane dla różnego rozmieszczenia węzłów: równoodległe oraz Czebyszewa\*. Oceń dokładność, z jaką wielomian przybliża zadaną funkcję. Poszukaj wielomianu, który najlepiej przybliża zadaną funkcję. Wyszukaj stopień wielomianu, dla którego można zauważyć efekt Rungego (dla równomiernego rozmieszczenia węzłów). Porównaj z wyznaczonym wielomianem dla węzłów Czebyszewa.

## Funkcja rysująca wykres a jego prawidłowy wygląd

Na potrzeby zadania funkcja została przepisana na język python a następnie narysowana przez zaimplementowaną funkcję rysującą wykres według zadanych  $n$  punktów, tutaj dla  $n=1000$ . Obok został przedstawiony obraz wyglądu funkcji według programu WolframAlpha.



Widać więc, że funkcja została zapisana dobrze, a program rysuje wykres w dokładny sposób.

## Implementacja funkcji liczących wielomiany dla postaci Lagrange'a i Newtona.

Zaimplementowane funkcje zostały zaczerpnięte z wykładu i przełożone na język python bez zewnętrznej pomocy, korzystając m.in. ze wzorów:

$$L_k(x) = \frac{d}{m} = \prod_{i=0, i \neq k}^n \frac{x - x_i}{(x_k - x_i)},$$

$$P_n(x) = \sum_{k=0}^n \underbrace{f(x_k)}_{\text{współczynniki}} \underbrace{L_k(x)}_{\text{baza Lagrange'a}}$$

Wzór 1.

$$P_n(x) = f[x_0] + (x - x_0)f[x_0, x_1] + (x - x_0)(x - x_1)f[x_0, x_1, x_2] + \dots + (x - x_0)(x - x_1) \dots (x - x_{n-1})f[x_0, x_1, \dots, x_n]$$

Wzór 2.

$x_0$	$f(x_0)$				
$x_1$	$f(x_1)$	$f[x_0, x_1]$			
$x_2$	$f(x_2)$	$f[x_1, x_2]$	$f[x_0, x_1, x_2]$		
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
$x_n$	$f(x_n)$	$f[x_{n-1}, x_n]$	$\dots$	$\dots$	$f[x_0, \dots, x_n]$

Wzór 3.

Gdzie Wzór 1 został użyty przy obliczaniu postaci Lagrange'a, a Wzory 2 i 3 przy postaci Newtona.

*Spostrzeżenie 1:* Jeżeli posługujemy się pojedynczą tablicą (najoszczędniejsze działanie), to obliczanie ilorazów różnicowych korzystając ze Wzoru 3. trzeba przeprowadzać od ostatniego wyrazu, ponieważ obliczanie ilorazu zależy od wyrazu na tej samej pozycji i o 1 niższej z poprzedniej serii liczenia. Jeśli zaczęlibyśmy od najniższych wyrazów, zmienilibyśmy wyniki,

które wciąż byłyby potrzebne w swojej poprzedniej wersji do obliczenia wartości z dalszych indeksów tablicy.

## Węzły w liczeniu funkcji

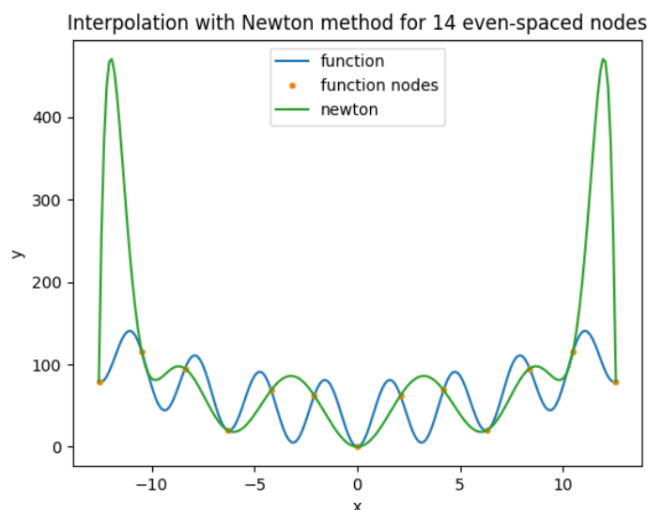
Wyżej wymienione funkcje liczące wielomiany interpolujące zostały uruchomione dla dwóch zbiorów węzłów: rozmieszczonych równomiernie na zadanym przedziale oraz wyznaczonych zerami wielomianu Czebyszewa. Drugi z nich posiada znacznie gęściej rozsięte punkty na końcach przedziału, co teoretycznie ma wpływać na zniwelowanie efektu Rungego, czego działanie zweryfikujemy.

Zanim jednak efekt Rungego, przyjrzymy się dokładności funkcji interpolowanych dla liczby węzłów z przedziału  $[5, 50]$ , liczonej zgodnie z błędem maksymalnym spośród 200 równomiernie rozłożonych punktów w przedziale.

## Dokładność interpolacji oraz najdokładniejszy wielomian

Jako pierwsze pomiary zostały wykonane dla punktów rozłożonych równomiernie:

nodecount	Lag	New
5	79.995	79.995
6	72.8721	72.8721
7	100.202	100.202
8	79.6615	79.6615
9	79.995	79.995
10	79.6762	79.6762
11	79.8877	79.8877
12	79.1969	79.1969
13	376.075	376.075
14	1028.2	1028.2
15	7245.26	7245.26
16	2477.3	2477.3
17	37173.1	37173.1
18	7118.65	7118.65
19	91853	91853
20	35424.2	35424.2



Na podstawie tabeli jesteśmy w stanie stwierdzić, że efekt Rungego zaczął występować około momentu wyznaczania funkcji dla 12 węzłów: błąd zamiast być mniejszy przez większą ilość węzłów (większą ilość danych) zaczyna gwałtownie rosnąć. Faktycznie, rysując wykres dla 14 węzłów widać efekt Rungego. Pozostałe obliczone błędy dla funkcji wyznaczonych na podstawie ilości węzłów  $[21, 50]$  zostały obcięte, ponieważ tylko rosną. Bazując na samym wyznaczaniu z równomiernie rozłożonymi węzłami, najdokładniejszy wielomian otrzymalibyśmy niezależnie od metody dla danych z 12 punktów.

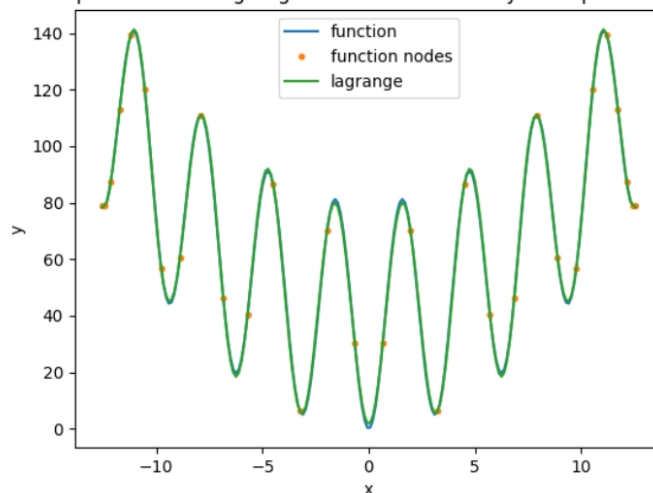
Następnie zmierzylem błędy maksymalne z wykorzystaniem zer wielomianu Czebyszewa:

<u>nodecount</u>	Lag	New
5	75.6399	75.6399
6	83.2037	83.2037
7	82.3374	82.3374
8	55.8558	55.8558
9	83.5142	83.5142
10	72.807	72.807
11	77.9555	77.9555
12	89.4368	89.4368
13	78.9565	78.9565
14	79.4446	79.4446
15	82.7452	82.7452
16	91.9617	91.9617
17	72.9265	72.9265
18	86.2533	86.2533
19	80.0718	80.0718
20	97.771	97.771
21	61.8925	61.8925
22	77.3814	77.3814
23	36.4904	36.4904
24	43.5823	43.5823
25	16.634	16.634

26	18.4172	18.4172
27	5.95771	5.9577
28	6.09923	6.09923
29	1.73022	1.73022
30	1.63559	1.63559
31	0.617173	0.418163
32	0.629792	0.364093
33	2.18695	0.0860674
34	15.0533	0.0686387
35	42.407	0.0185779
36	10.9485	0.0112074
37	95.0602	0.0153689
38	1179.85	0.0227156
39	1095.74	0.0412067
40	20967.6	0.0975513
41	10024.5	0.0497263
42	10963	0.121919
43	191970	0.0866837
44	1.23413e+06	0.344479
45	637517	0.330679
46	1.1173e+06	0.519449
47	1.39518e+07	0.742134
48	9.47449e+07	0.570064
49	2.91055e+08	1.35102

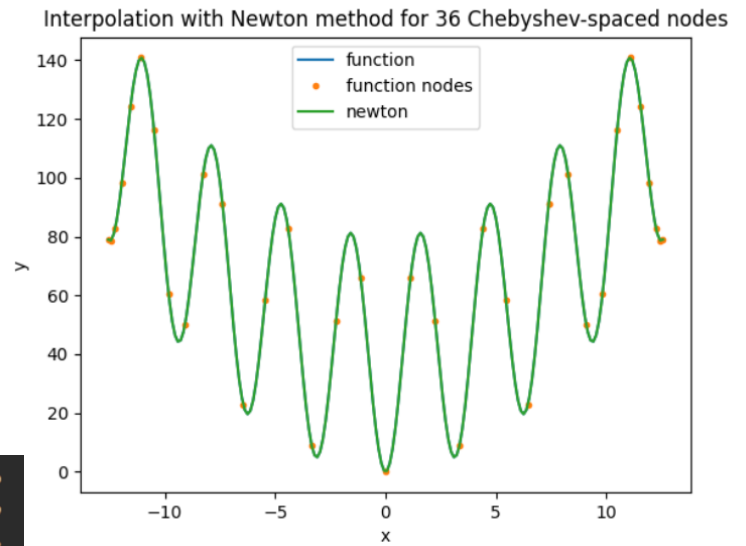
W porównaniu do wyników otrzymanych przy obliczeniach z równomiernie rozłożonymi węzłami, tutaj nie widać efektu Rungego. W przypadku wielomianu w postaci Lagrange'a po 31 węzle błąd zaczyna z powrotem rosnąć, lecz przy wielomianach takiego stopnia prawdopodobnie jest to kwestia błędów arytmetyki (wyznaczany wielomian jest zapisywany w postaci tablicy przy pomocy struktury "polynomial". Wartość dla wielomianu w postaci Newtona została policzona za każdym razem od razu z podstawioną wartością "x", dlatego takie błędy nie występują na naszej skali.

Interpolation with Lagrange method for 31 Chebyshev-spaced nodes



Ze względu na najmniejszy wyliczony błąd maksymalny najdokładniejszym wielomianem jest ten w postaci Newtona dla węzłów rozłożonych zgodnie z zerami wielomianu Czebyszewa dla 36 węzłów. Sposób implementacji funkcji uniemożliwia nam wyznaczenie samego wielomianu, ale funkcja Lagrange'a posiadająca niewiele większy błąd jest w stanie go wyznaczyć, dla wymienionych wcześniej 31 węzłów będzie to:

x's power	coefficient		
-----	-----		
0	1.65441	14	4.19549e-06
1	-3.13499e-14	15	2.30555e-19
2	75.7716	16	-5.46659e-08
3	-6.143e-14	17	3.32558e-21
4	-24.3977	18	5.09658e-10
5	1.61815e-14	19	1.65985e-23
6	3.12059	20	-3.38464e-12
7	-2.29677e-15	21	1.00438e-26
8	-0.20971	22	1.56544e-14
9	-5.82488e-16	23	-1.33367e-29
10	0.00853255	24	-4.79624e-17
11	1.28373e-16	25	3.9963e-32
12	-0.000227938	26	8.7544e-20
13	9.76862e-18	27	-9.21588e-35
		28	-7.20849e-23
		29	5.12442e-38



## Wniosek:

Z porównania wyników liczonych dla węzłów równomiernie rozłożonych oraz tych wyznaczonych zerami wielomianu Czebyszewa zgodnie z przewidywaniami wynika, że zastosowanie tego drugiego rozkładu zapobiega wystąpieniu efektu Rungego i znacząco dokładniej przybliża zadaną funkcję.