Лабораторная работа №2
Интерполирование

Выполнил студент 2 курса 3 группы ФПМИ
Сараев Владислав Максимович

Минск, 2020

Теоретические сведения

Дана функция $f(x) = (\sin(4x) - x)^3$. Необходимо построить интерполяционный многочлен данной функции в барицентрической форме на отрезке $[-2; 2]$. Интерполирование необходимо провести по N_i узлам ($N_i = 10i,\ i = 1, 2, \dots, 10$) как по равноотстоящим узлам, так и по чебышевским. Для каждого построение необходимо построить графики получившихся приближений и экспериментально определить максимум-норму погрешности (максимум величины $|f(x_i) - P(x_i)|, i = 1, \dots, 1000$).

Чебышевские узлы вычисляются по формуле:

$$x_k = \frac{1}{2}(a + b) + \frac{1}{2}(b - a)\cos\left(\frac{2k - 1}{2n}\pi\right), k = 1, \dots, n \qquad (1)$$

где n – число точек, а a и b – концы отрезка, а котором проводится интерполяция.

Интерполяционный многочлен в барицентрической форме имеет вид:

$$L_n(x) = \frac{\sum_{i=0}^{n} \frac{c_i\, f_i}{x - x_i}}{\sum_{i=0}^{n} \frac{c_i}{x - x_i}} \qquad (2)$$

Соответственно, для его нахождения необходимо предварительно вычислить только нормирующие множители c_i, которые вычисляются по формуле:

$$c_i = \frac{1}{\prod_{j=0, j\neq i}^{n}(x_i - x_j)} \qquad (3)$$

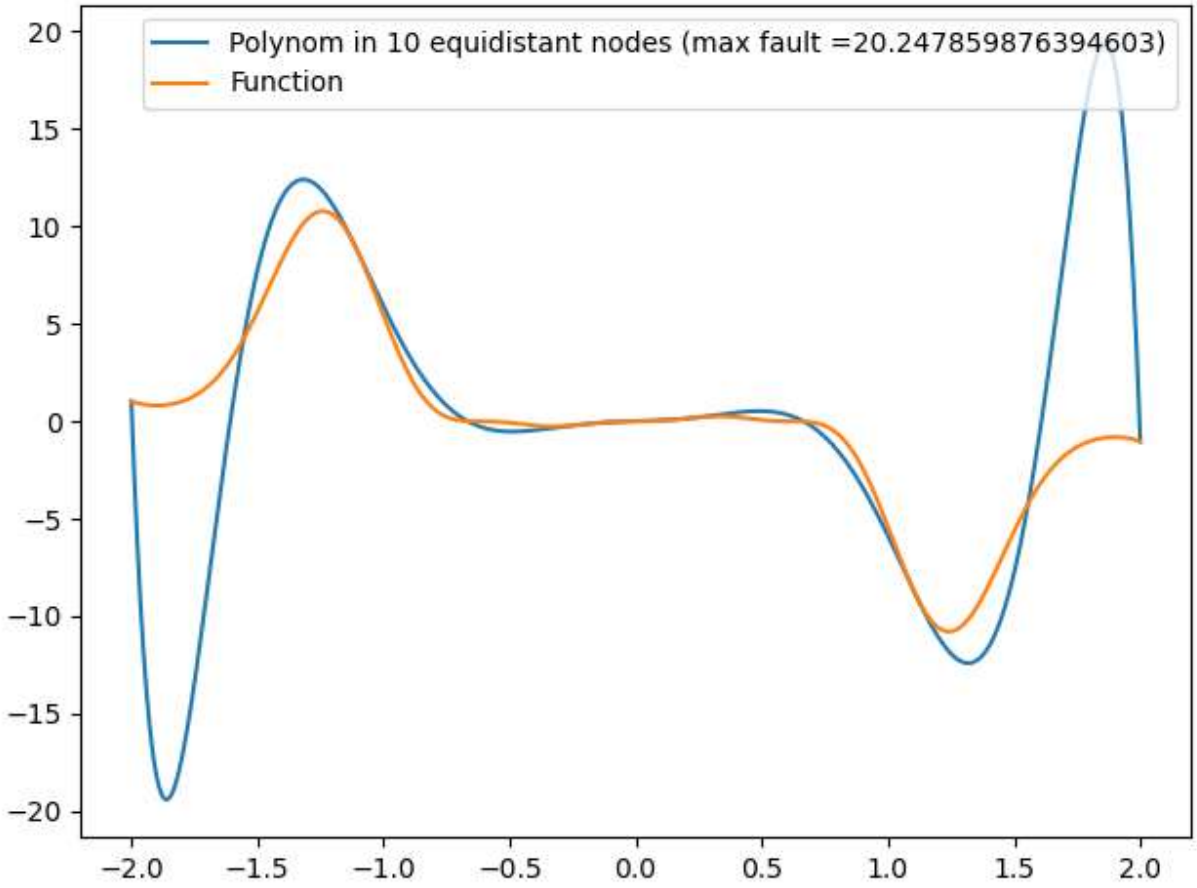
Вычислив один раз нормирующие множители c_i, можно найти значение интерполяционного многочлена $L_n(x)$ во всех интересующих точках.

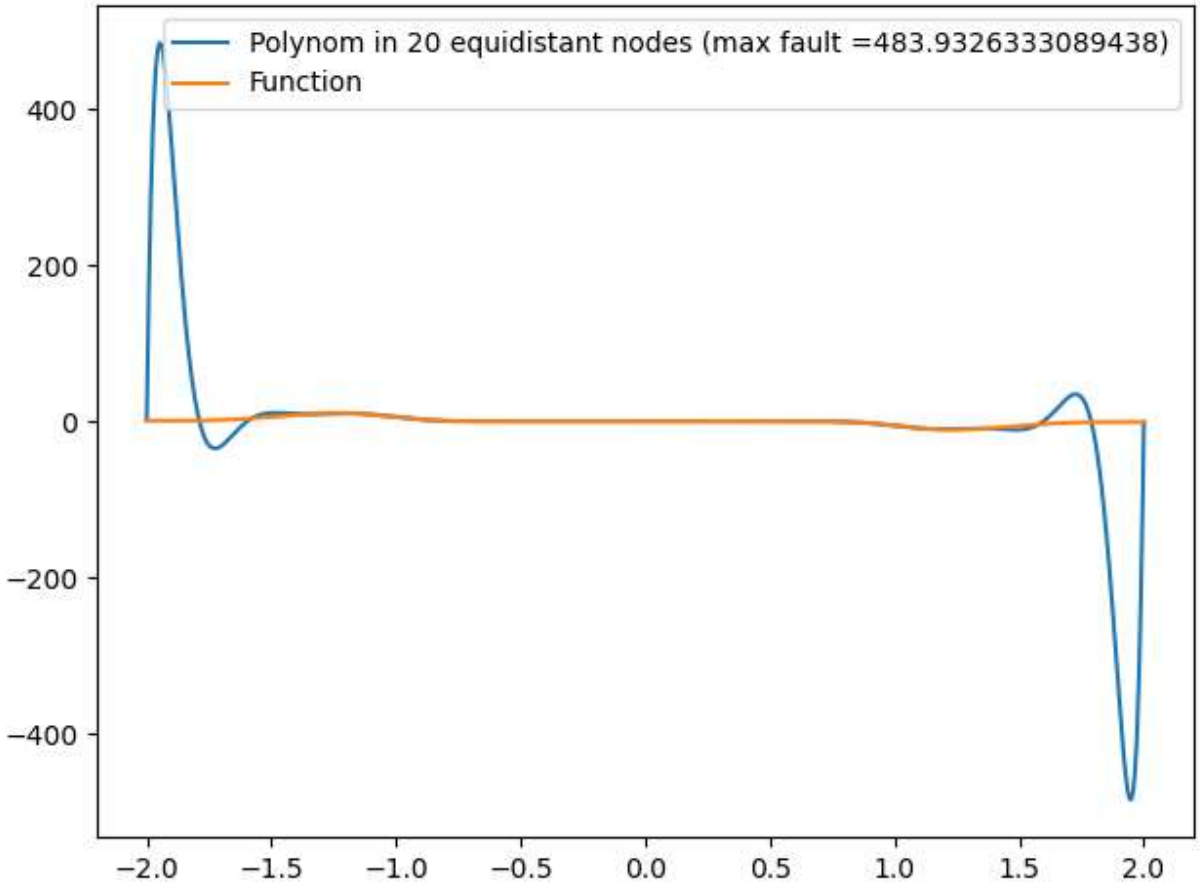
Соответственно, план выполнения задания следующий:

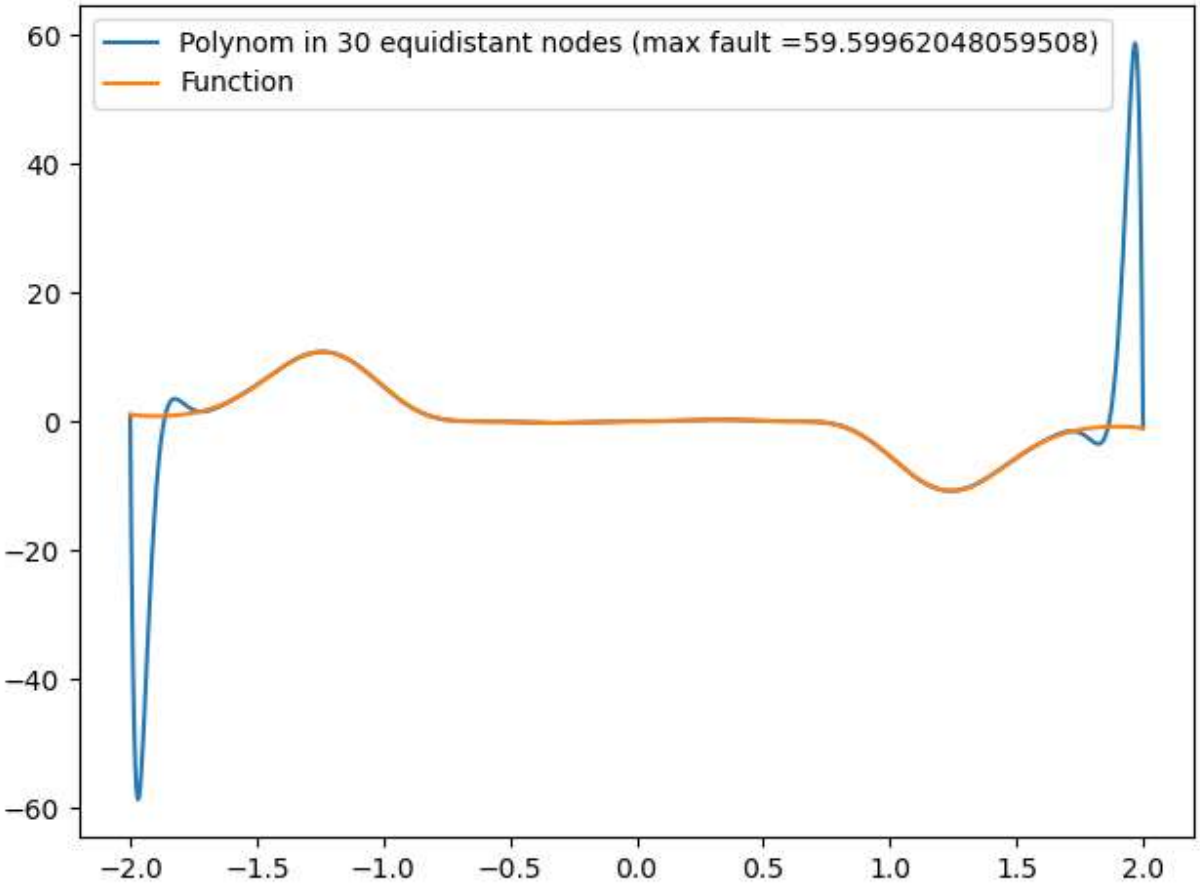
1) Разделить отрезок $[-2, 2]$ на нужное количество равноотстоящих узлов или вычислить необходимое количество чебышевских узлов по формуле (1).
2) Вычислить коэффициенты c_i по формуле (3) и, соответственно построить многочлен (2).
3) По формуле (2) найти значения в искомых точках
4) Построить график и вычислить максимум-норму погрешности

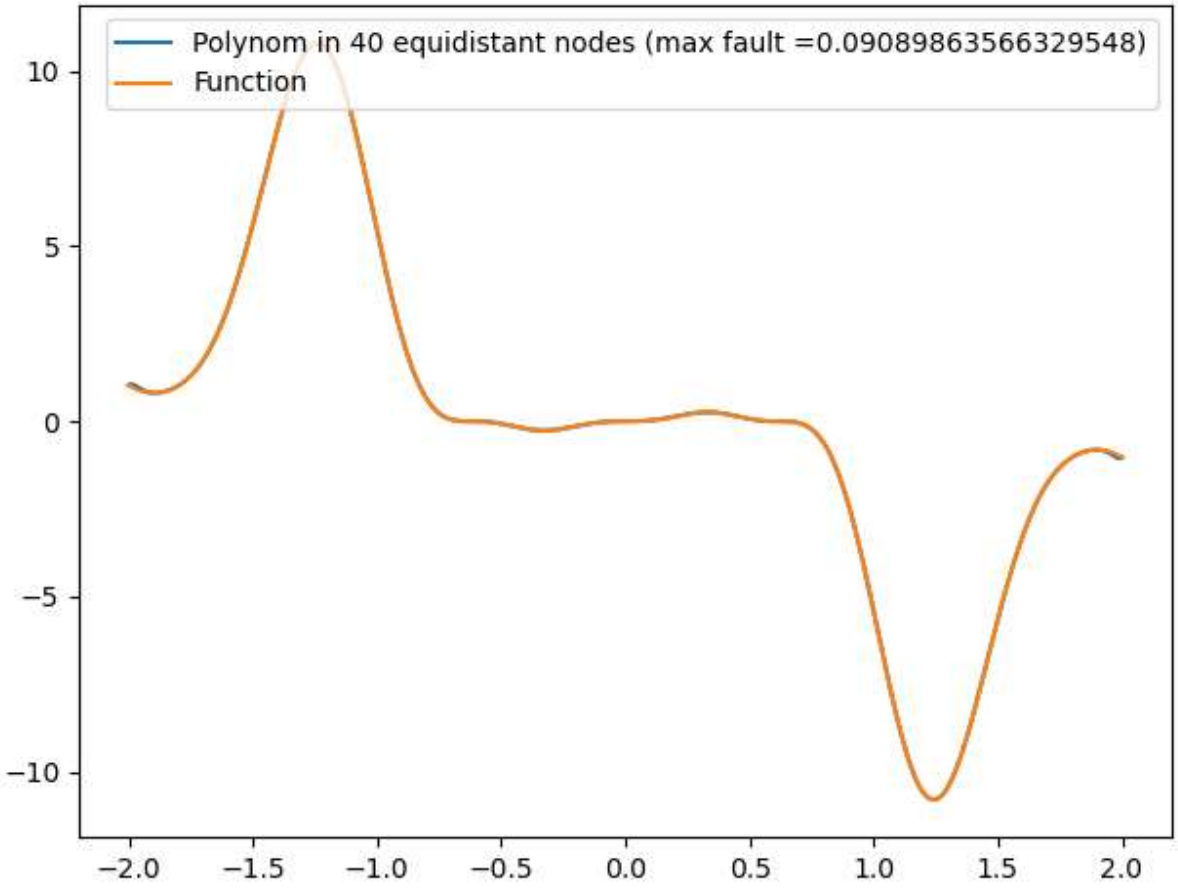
Полученное решение

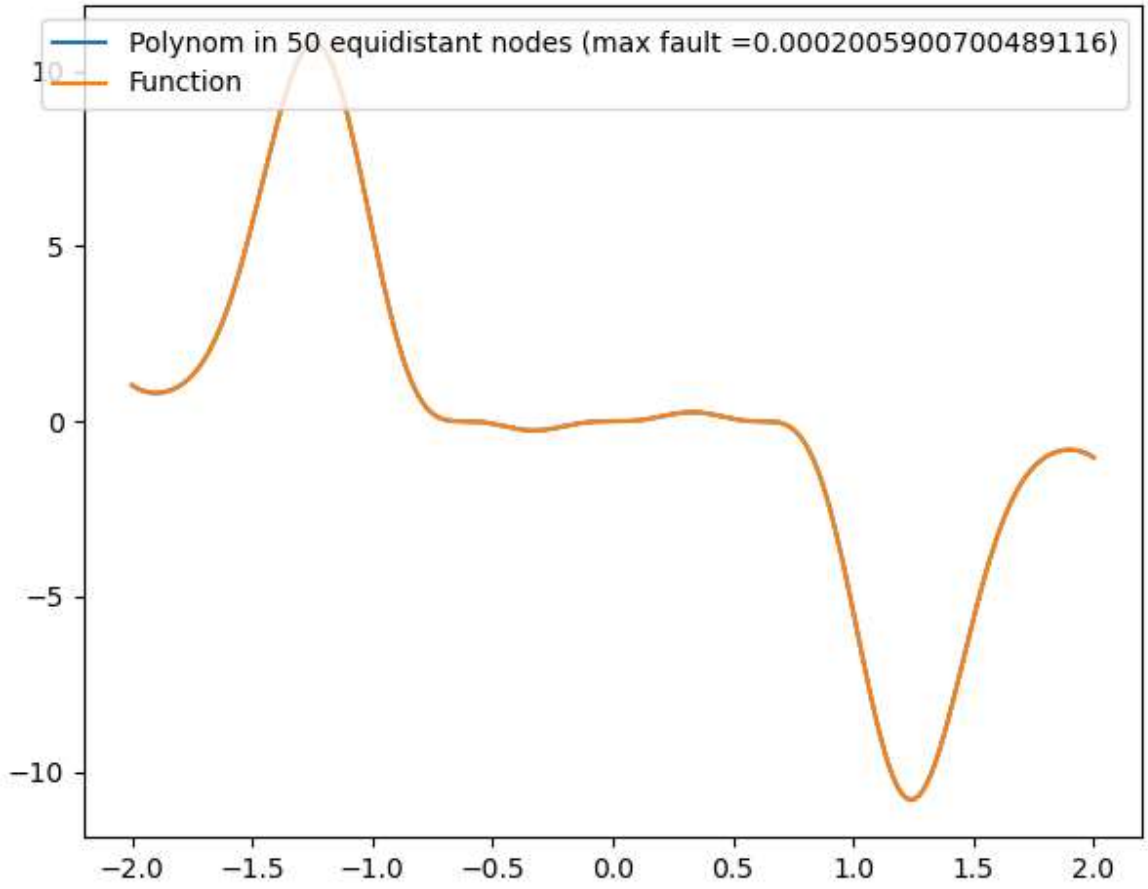
Графики, полученные по равноотстоящим узлам

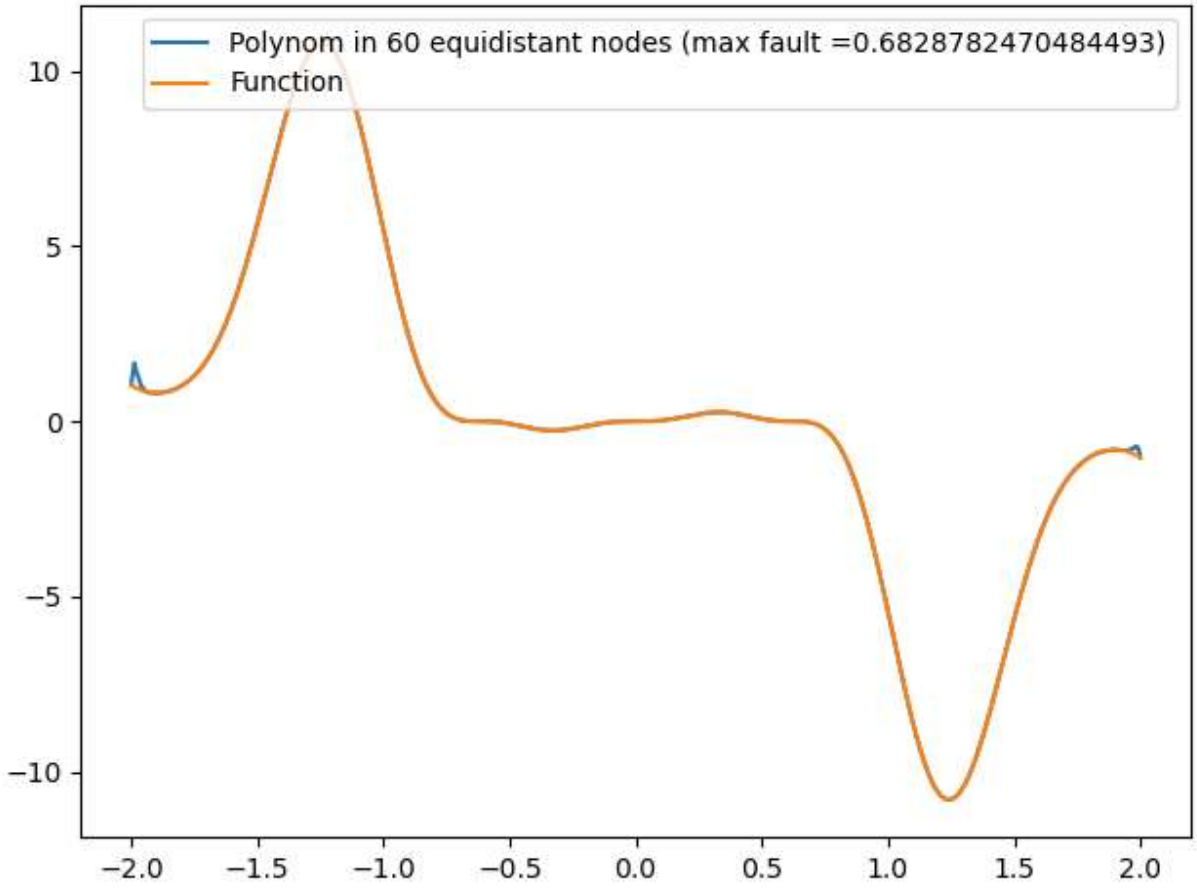


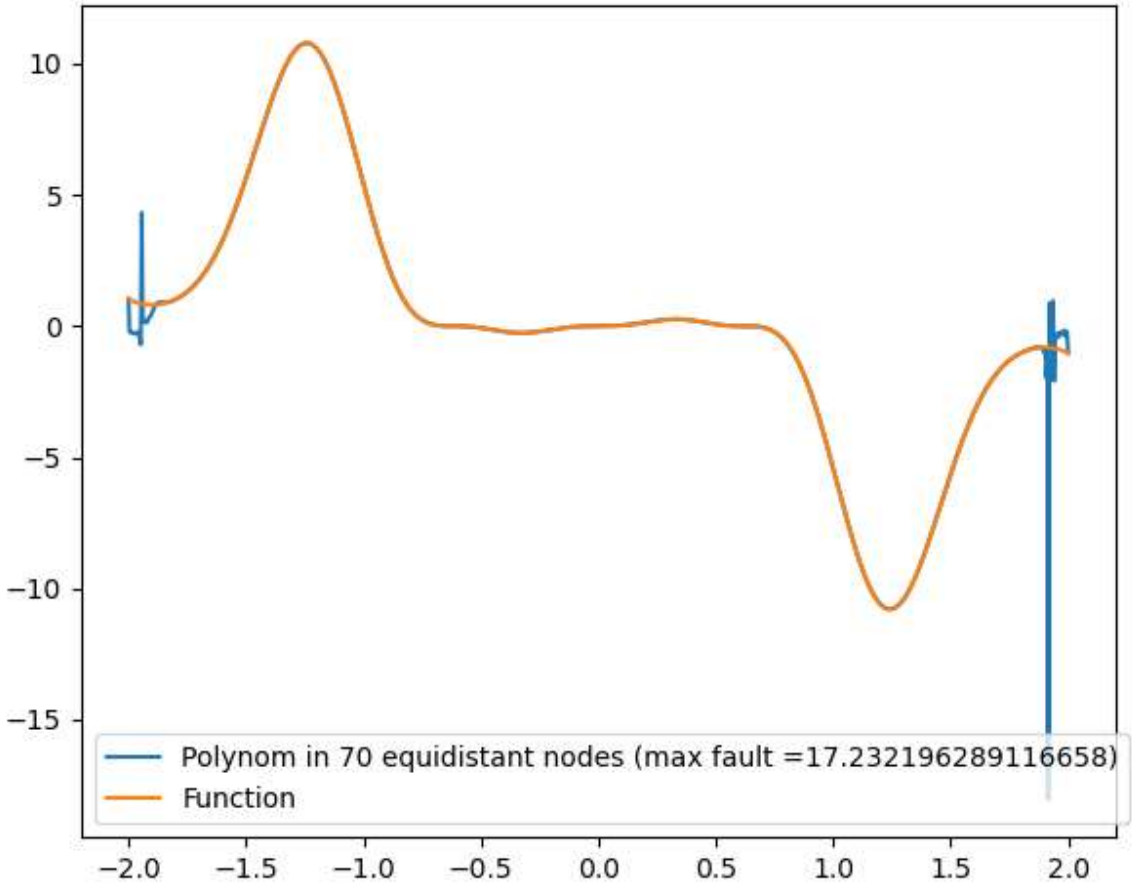

Polynom in 20 equidistant nodes (max fault =483.9326333089438)
Function
400
200
0
−200
−400
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

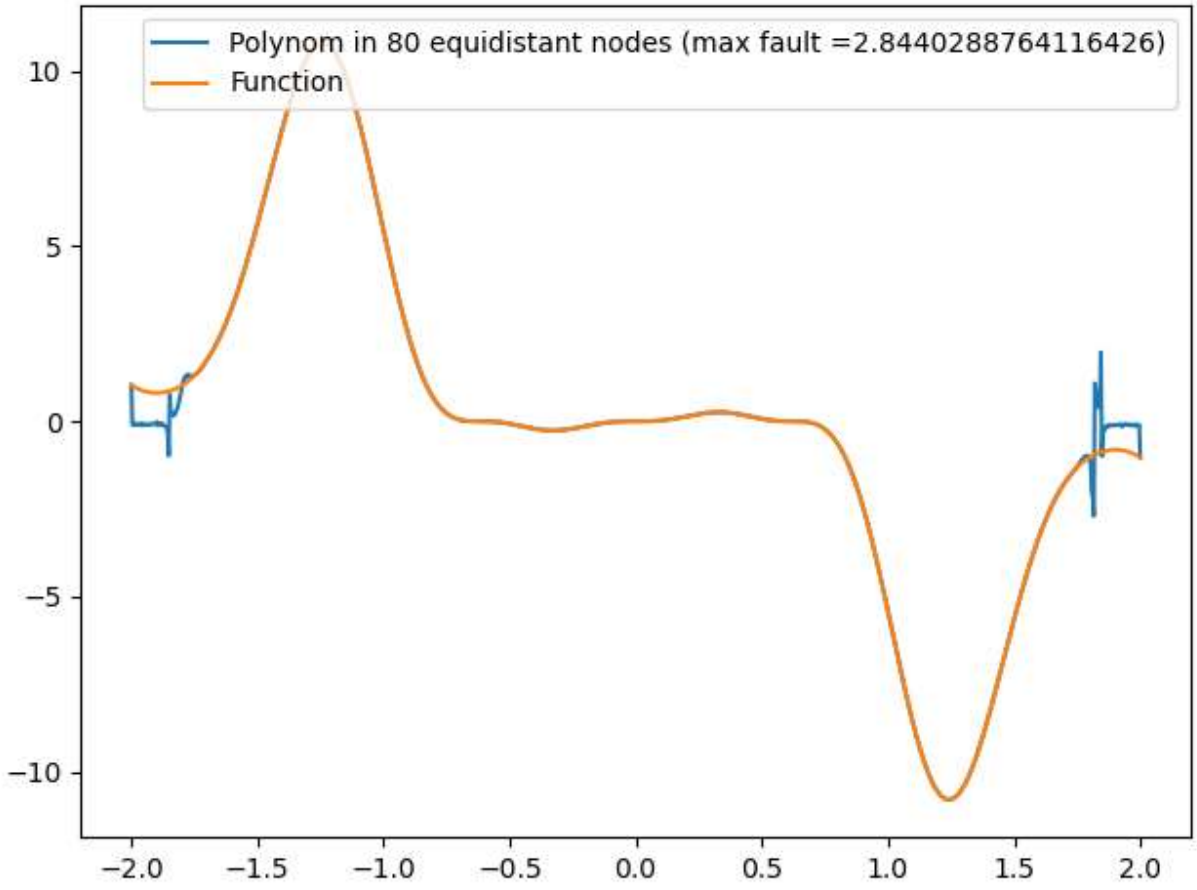

Polynom in 30 equidistant nodes (max fault =59.59962048059508)
Function
60
40
20
0
−20
−40
−60
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

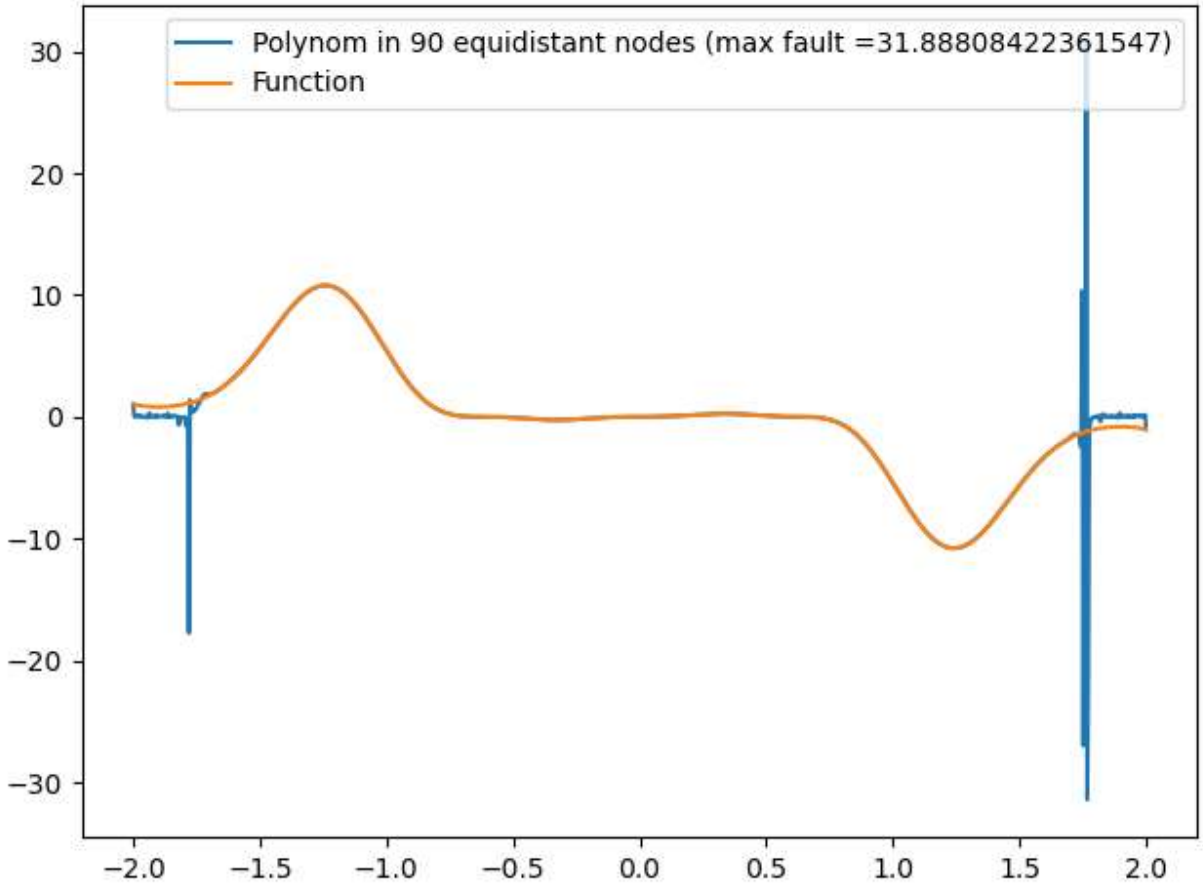

Polynom in 40 equidistant nodes (max fault =0.09089863566329548)
Function
10
5
0
−5
−10
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

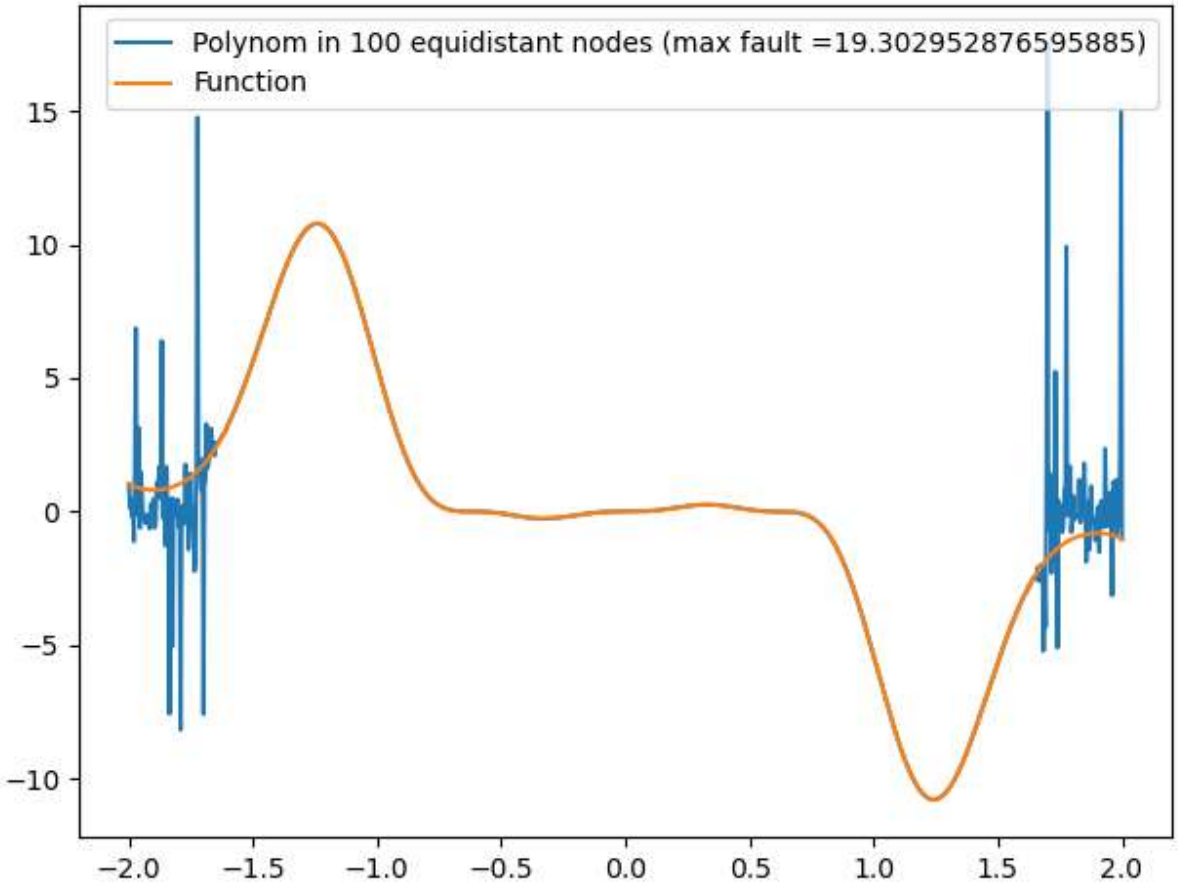

Polynom in 50 equidistant nodes (max fault =0.0002005900700489116)
Function
10
5
0
−5
−10
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0


Polynom in 60 equidistant nodes (max fault =0.6828782470484493)
Function
10
5
0
−5
−10
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

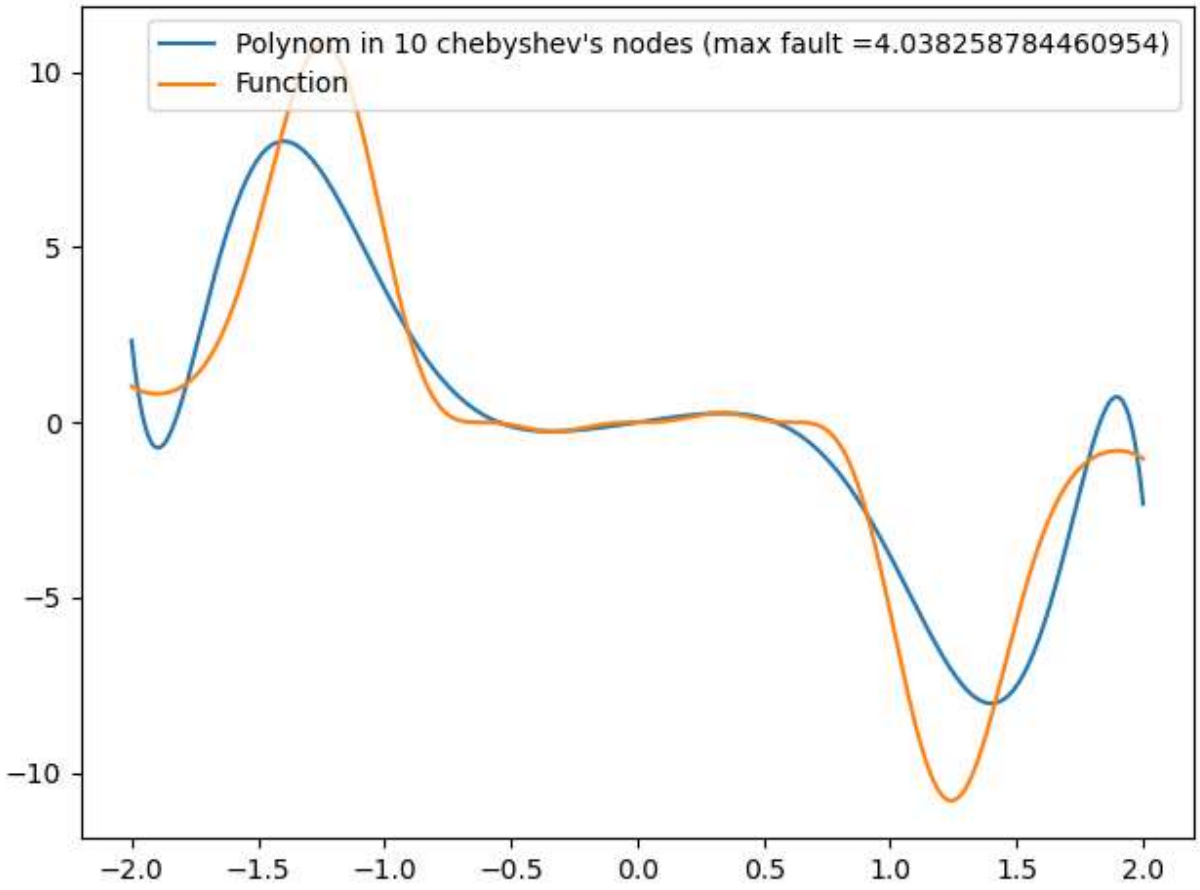

Polynom in 70 equidistant nodes (max fault =17.232196289116658)
Function
10
5
0
−5
−10
−15
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

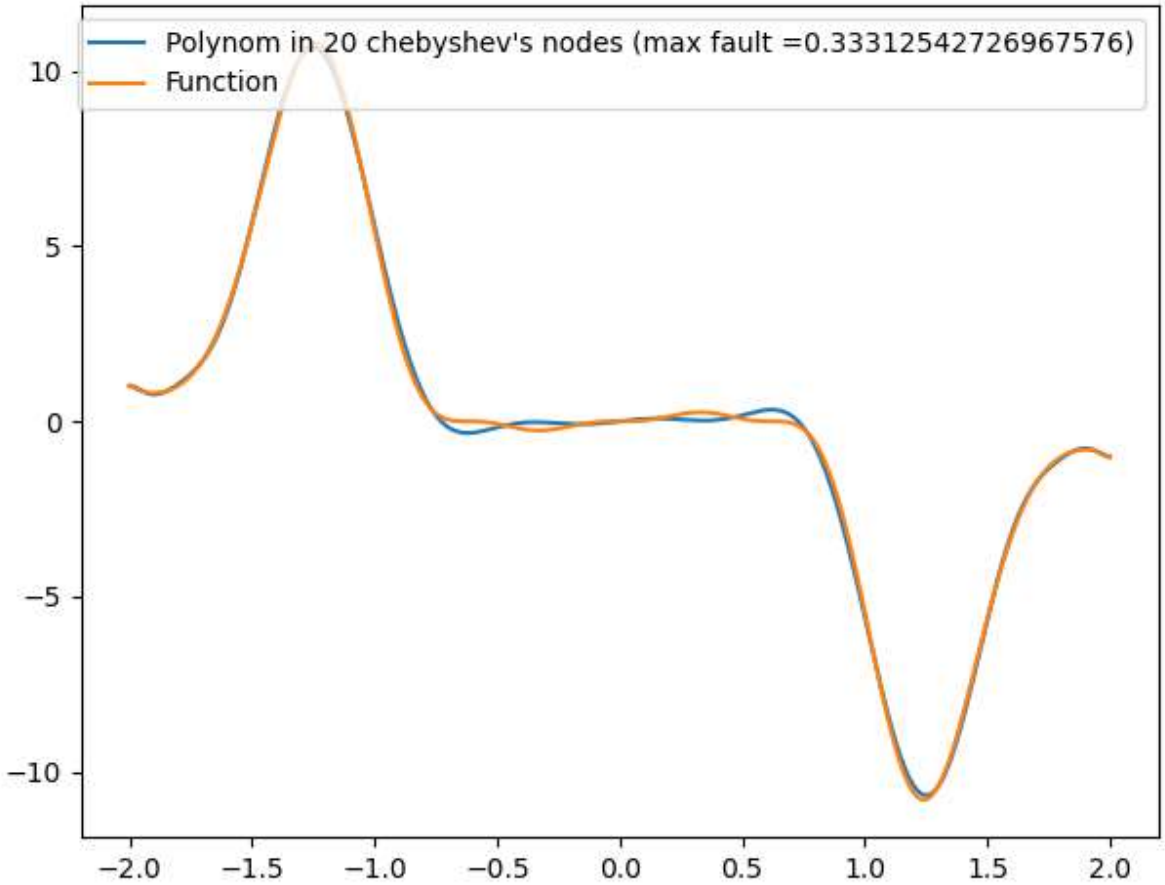

Polynom in 80 equidistant nodes (max fault =2.8440288764116426)
Function
10
5
0
−5
−10
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

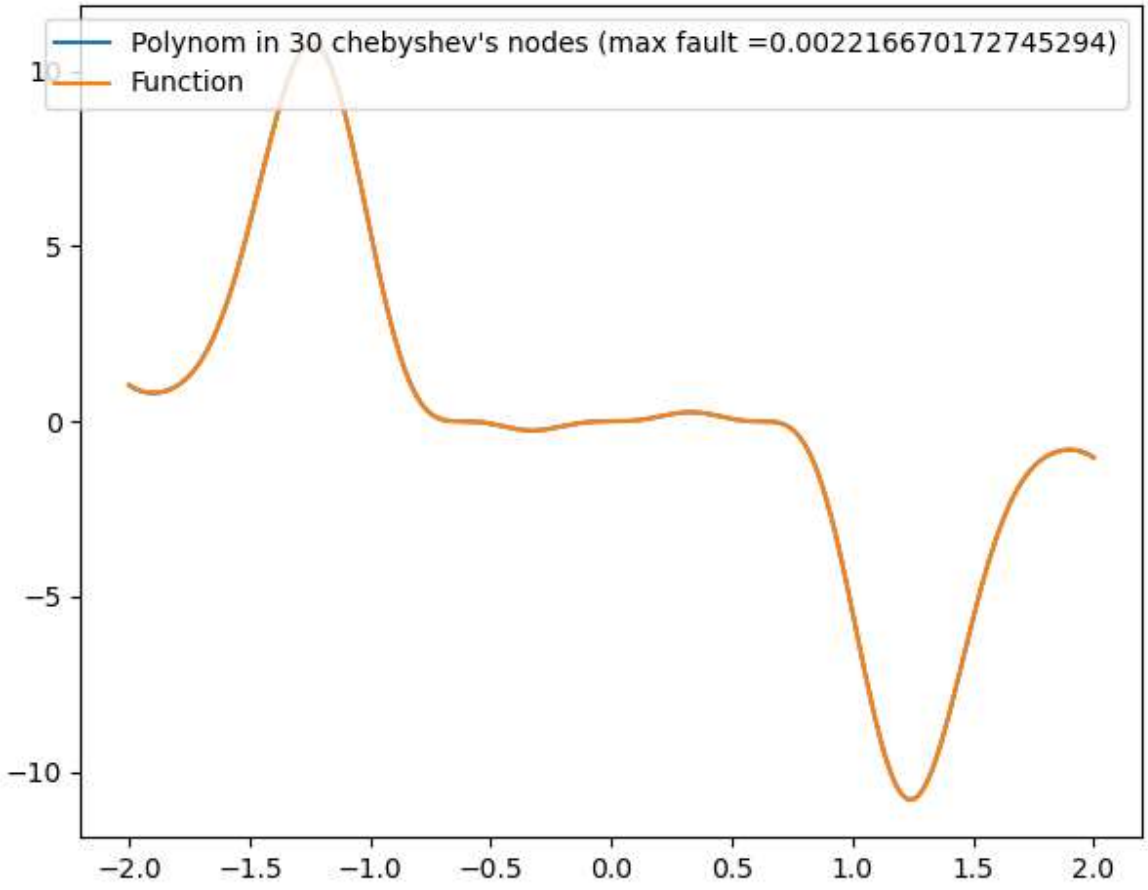

Polynom in 90 equidistant nodes (max fault =31.88808422361547)
Function
30
20
10
0
−10
−20
−30
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

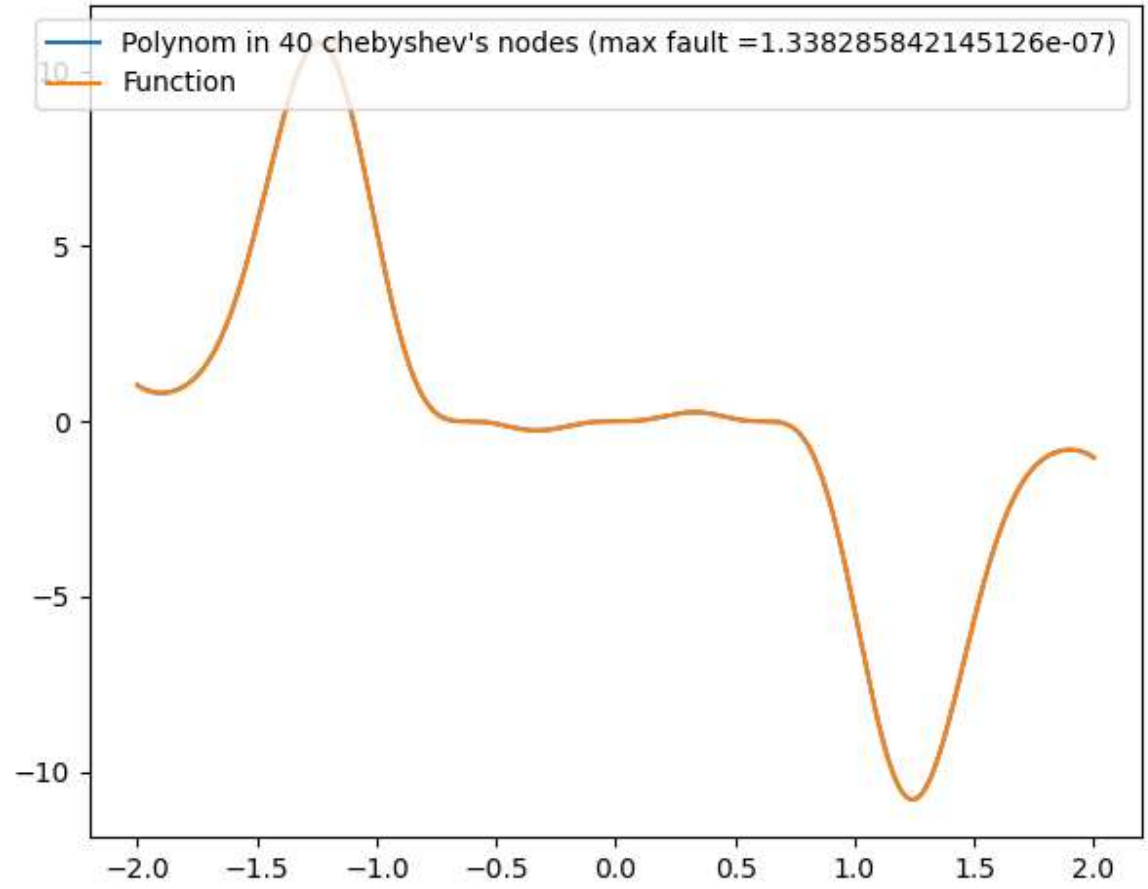

Polynom in 100 equidistant nodes (max fault =19.302952876595885)
Function
15
10
5
0
−5
−10
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

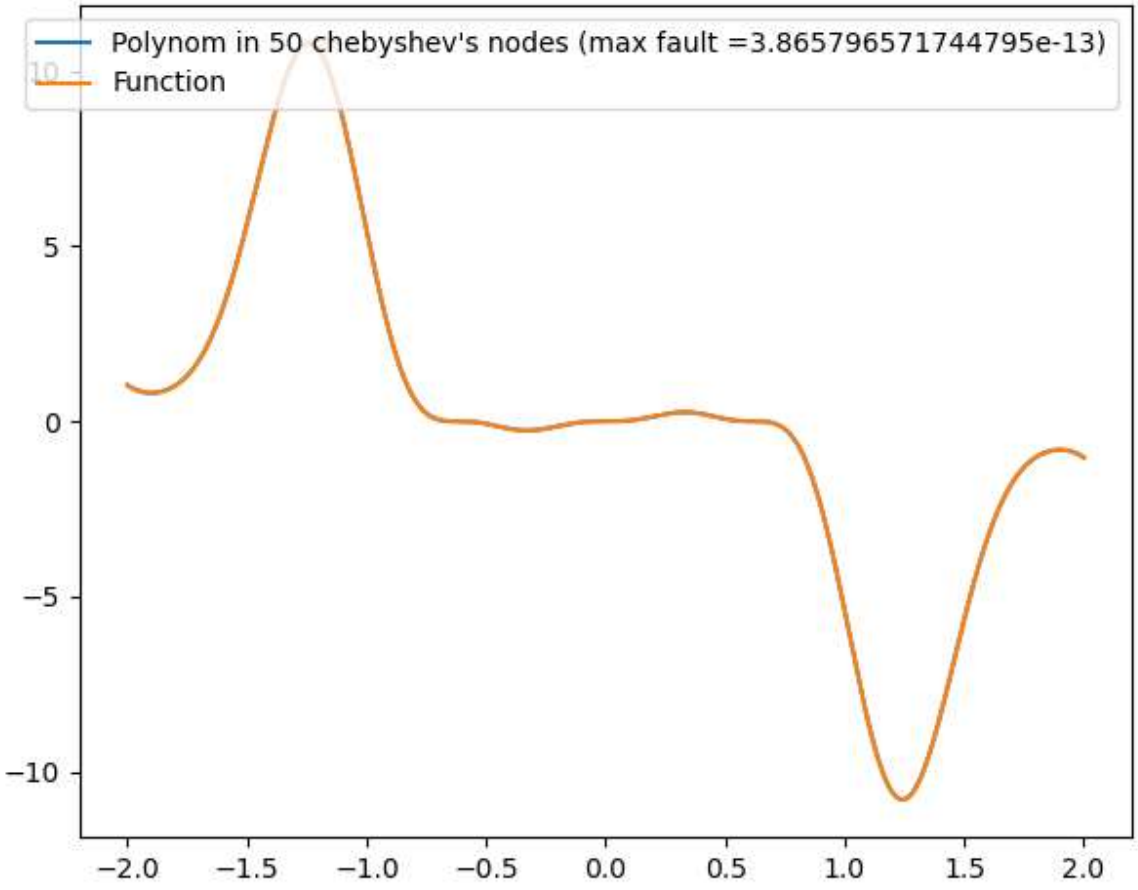
Графики, полученные по чебышевским узлам

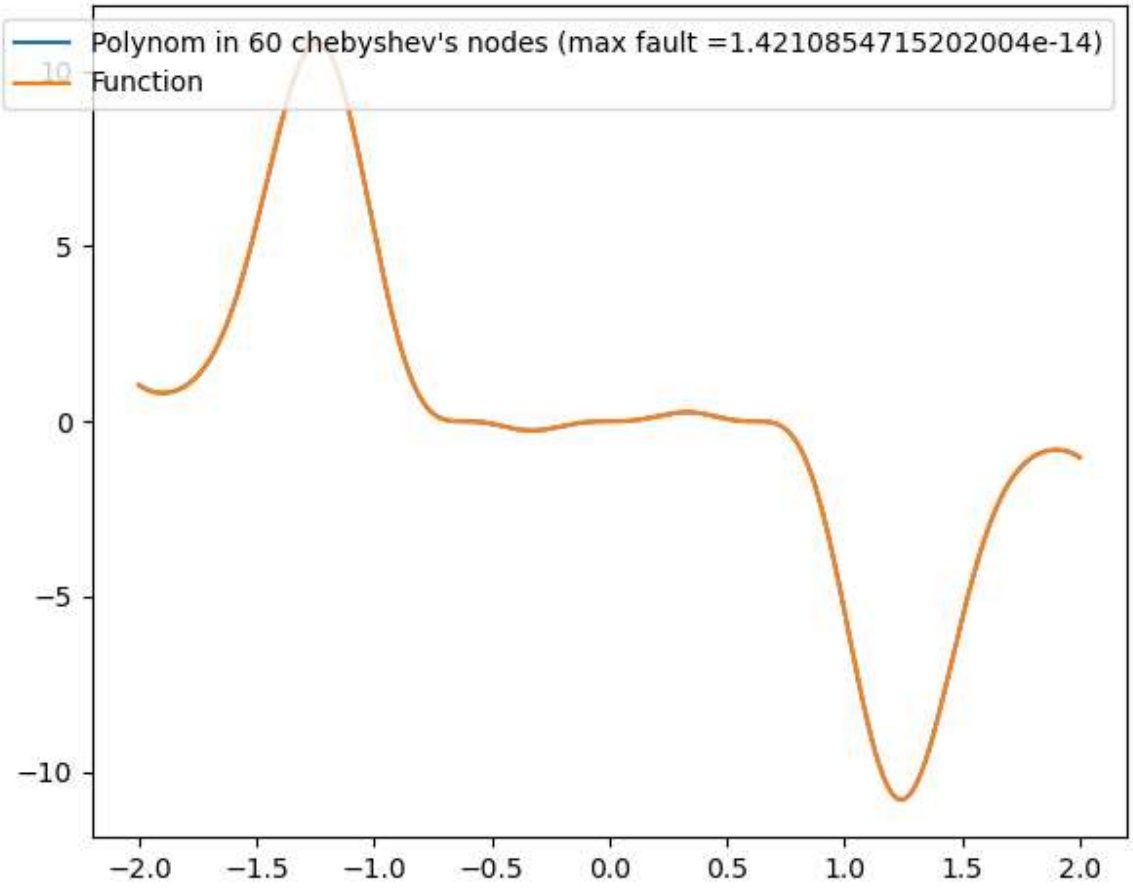


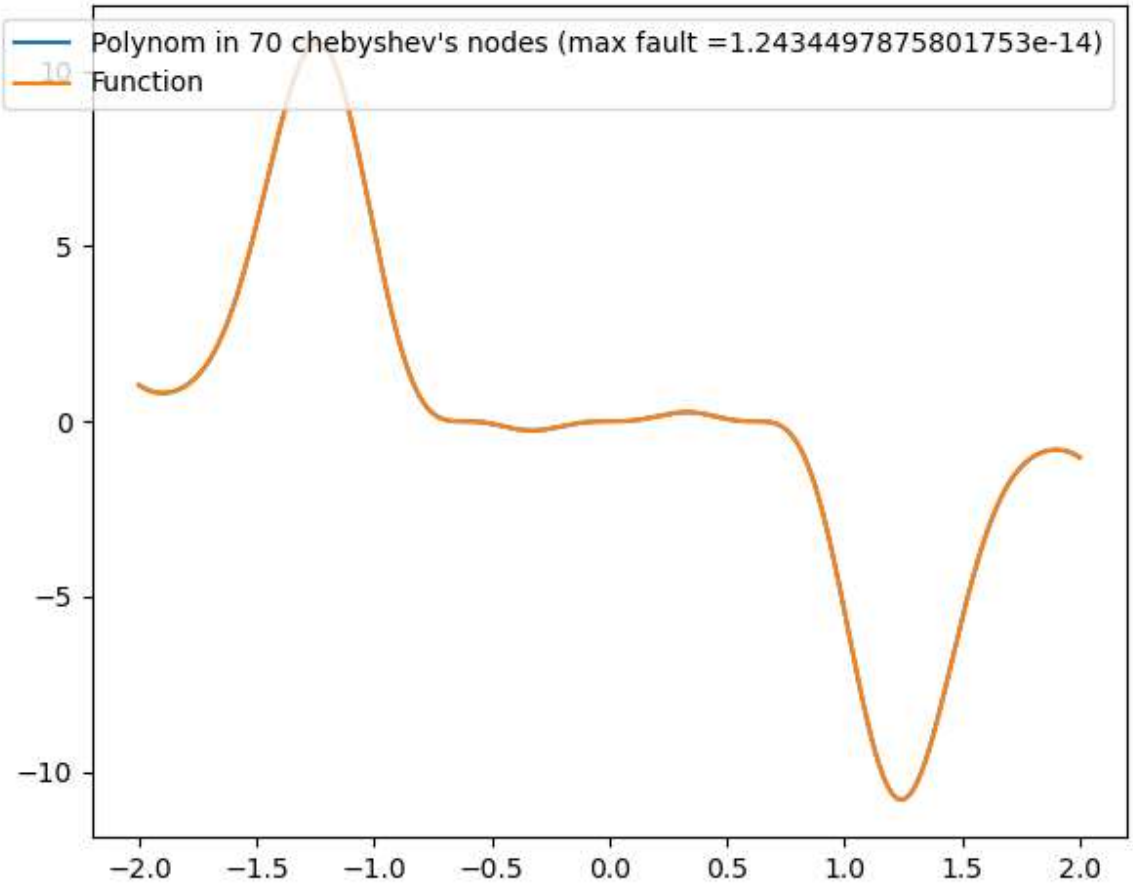


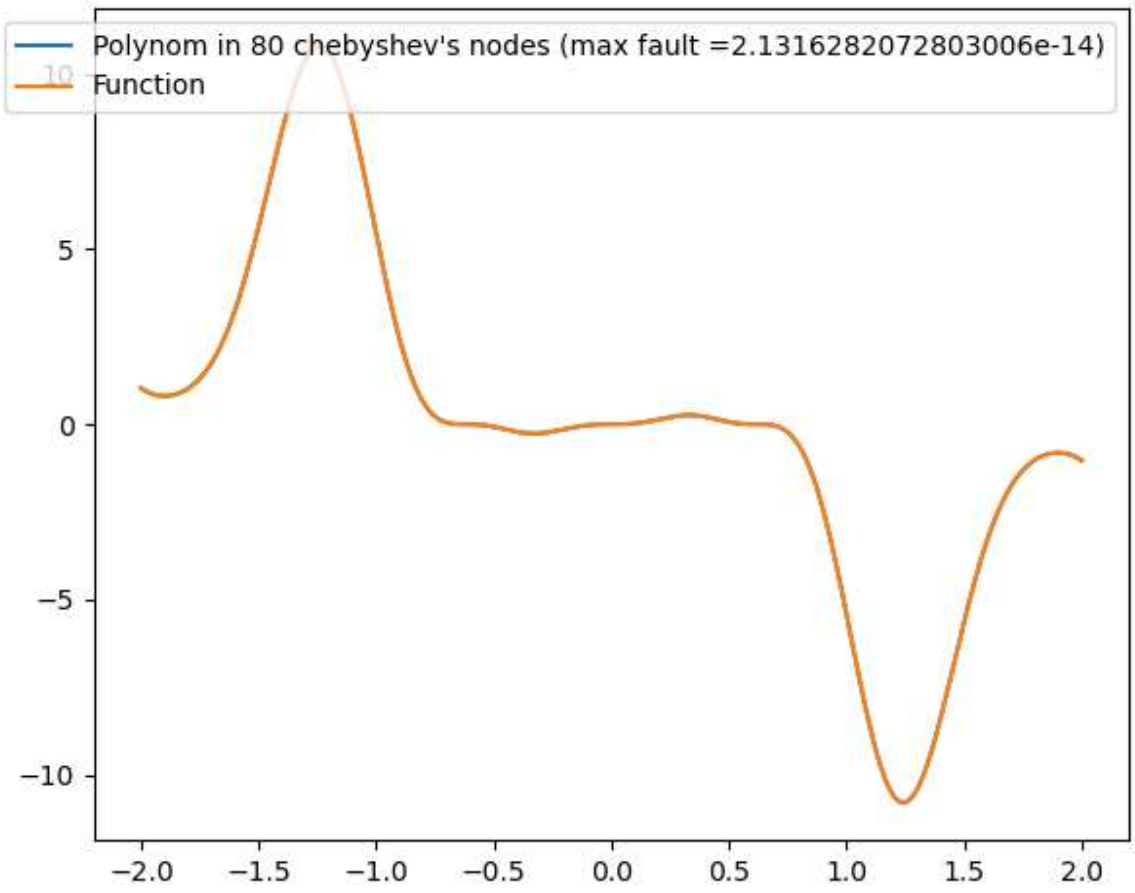

Polynom in 30 chebyshev's nodes (max fault =0.0022166701727452 94)
Function
10
5
0
−5
−10
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

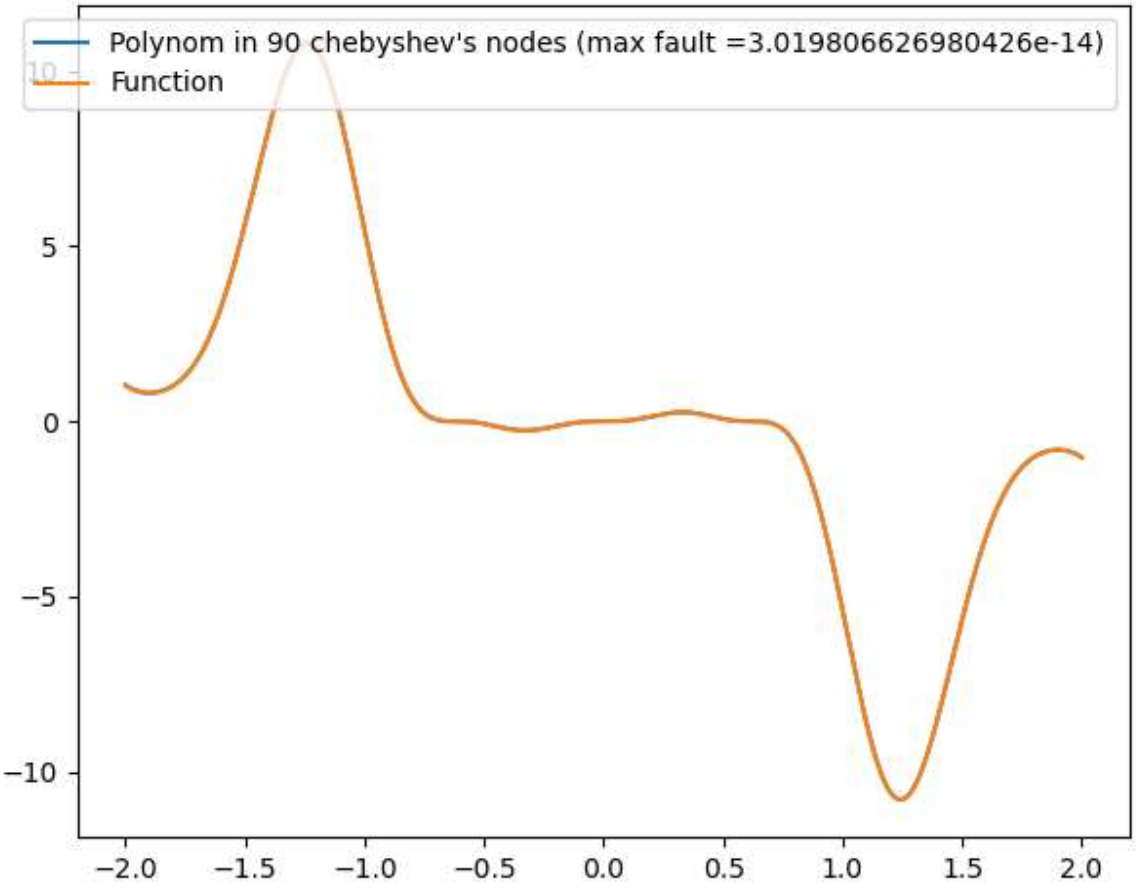

Polynom in 40 chebyshev's nodes (max fault =1.338285842145126e-07)
Function
10
5
0
−5
−10
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

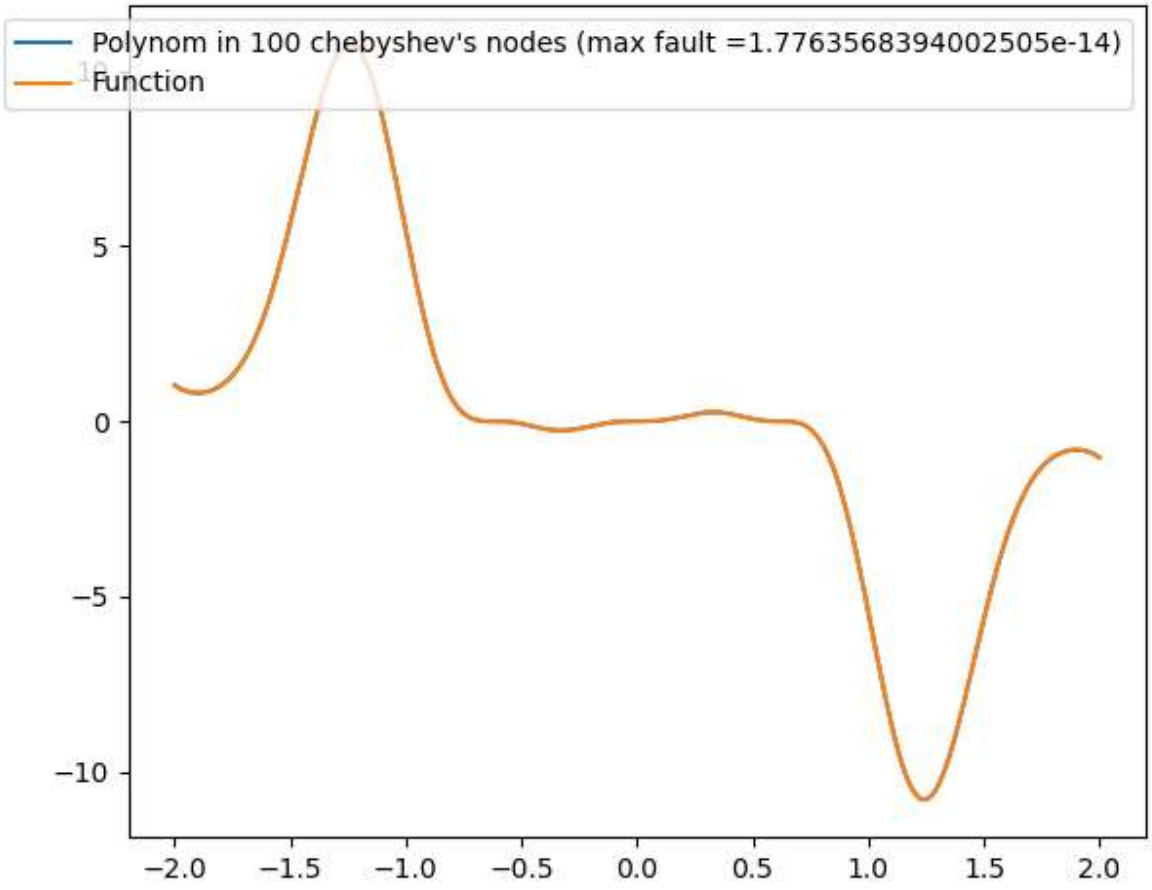

Polynom in 50 chebyshev's nodes (max fault =3.865796571744795e-13)
Function
10
5
0
−5
−10
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0


Polynom in 60 chebyshev's nodes (max fault =1.4210854715202004e-14)
Function
10
5
0
−5
−10
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0


Polynom in 70 chebyshev's nodes (max fault =1.2434497875801753e-14)
Function
10
5
0
−5
−10
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0


Polynom in 80 chebyshev's nodes (max fault =2.1316282072803006e-14)
Function
10
5
0
−5
−10
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0


Polynom in 90 chebyshev's nodes (max fault =3.019806626980426e-14)
Function
10
5
0
−5
−10
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0



Нормы

N	Норма (равноотстоящие узлы)	Норма (чебышевские узлы)
10	20.247859876394603	4.038258784460954
20	483.9326333089438	0.33312542726967576
30	59.59962048059508	0.002216670172745294
40	0.09089863566329548	1.338285842145126e-07
50	0.0002005900700489116	3.865796571744795e-13
60	0.6828782470484493	1.4210854715202004e-14
70	17.232196289116658	1.2434497875801753e-14
80	2.8440288764116426	2.1316282072803006e-14

90	31.88808422361547	3.019806626980426e-14
100	19.302952876595885	1.7763568394002505e-14

Исходный код

```
import numpy as np
from math import sin, cos, pi
import matplotlib.pyplot as plot


def function(t):
    return (sin(4 * t) - t) ** 3


def polynom(t):
    global x, y, c
    for i in range(len(x)):
        if t == x[i]:
            return y[i]

    den = [c[i] / (t - x[i]) for i in range(len(c))]
    return sum(den[i] * y[i] for i in range(len(c))) / sum(den)


L = -2
R = 2

dots = np.linspace(L, R, 1000)
count_of_dots = [10 * i for i in range(1, 11)]
type_of_nodes = "equidistant"
for l in range(2):
    for i in range(len(count_of_dots)):
        N = count_of_dots[i]
        if type_of_nodes == "equidistant":
            x = np.linspace(L, R, N)
        else:
            x = np.empty(N)
            for j in range(1, N + 1):
                x[j - 1] = (L+R)/2 + (R-L)/2*cos((2*j-1)*pi/(2*N))
        y = np.empty(N)

        for j in range(0, N):
            y[j] = function(x[j])

        c = np.ones(N)
        for j in range(len(x)):
            for k in range(len(x)):
                if j != k:
                    c[j] *= x[j] - x[k]
            c[j] = 1 / c[j]
        polynom_dots = [polynom(t) for t in dots]
        function_dots = [function(t) for t in dots]
        max_fault = max(abs(polynom_dots[i] - function_dots[i]) for i in
range(len(polynom_dots)))
        if type_of_nodes == "equidistant":
            label_info = "Polynom in " + str(N) + " equidistant nodes (max
fault =" + str(max_fault) + ")"
        else:
            label_info = "Polynom in " + str(N) + " chebyshev's nodes (max
fault =" + str(max_fault) + ")"

        plot.plot(dots, polynom_dots, label=label_info)
        plot.plot(dots, function_dots, label="Function")
        plot.legend()
        plot.show()
    type_of_nodes = "chebyshev's"
```