

ЛАБОРАТОРНАЯ РАБОТА №2

«ИНТЕРПОЛИРОВАНИЕ»

Выполнил

Святослав Артюшкевич, 3 группа, 2 курс

Условие задачи

Для заданной функции $f: [a, b] \rightarrow R, f(x) = x^2 + 2\sin(10x)$ требуется

- Произвести интерполяцию многочленом Лагранжа на отрезке $[-2, 2]$.
- Интерполирование стоит проводить как по равноотстоящим узлам, так и по чебышевским.
- Для каждого типа узлов построить графики получившихся приближений для сеток с количеством узлов, равным $N_i = 10i, i = 1, 2, \dots, 10$. На графике должны быть изображены построенное приближение и исходная функция.
- Для каждого построения экспериментально определить максимум-норму погрешности: взять сетку из 1000 равноотстоящих узлов и определить максимум величины $|f(x_i) - P(x_i)|, i = 1, \dots, 1000$

Теоретические сведения

Узлы Чебышева на произвольном отрезке $[a, b]$ получаются по формуле:

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

Интерполяционный многочлен Лагранжа это многочлен вида:

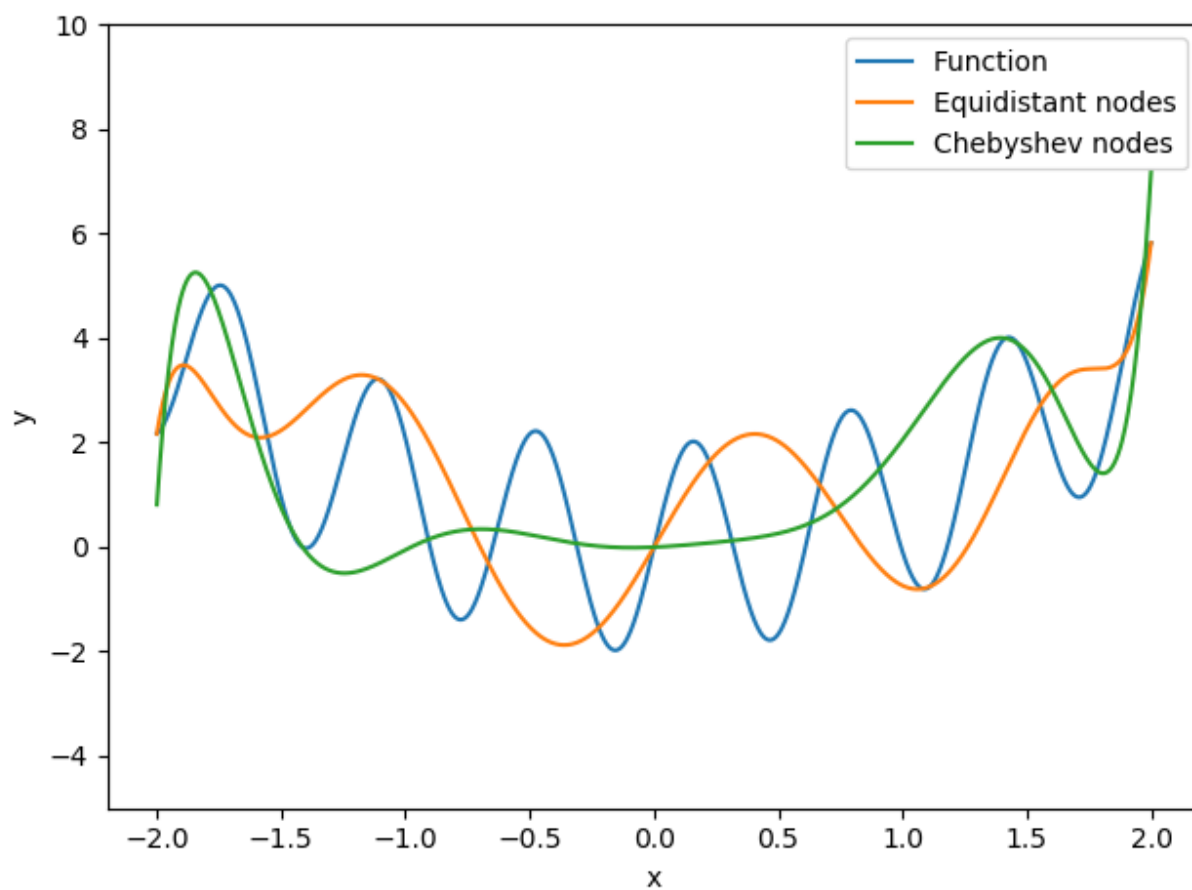
$$L(x) = \sum_{i=0}^n y_i l_i(x)$$

Где базисные полиномы определяются по формуле:

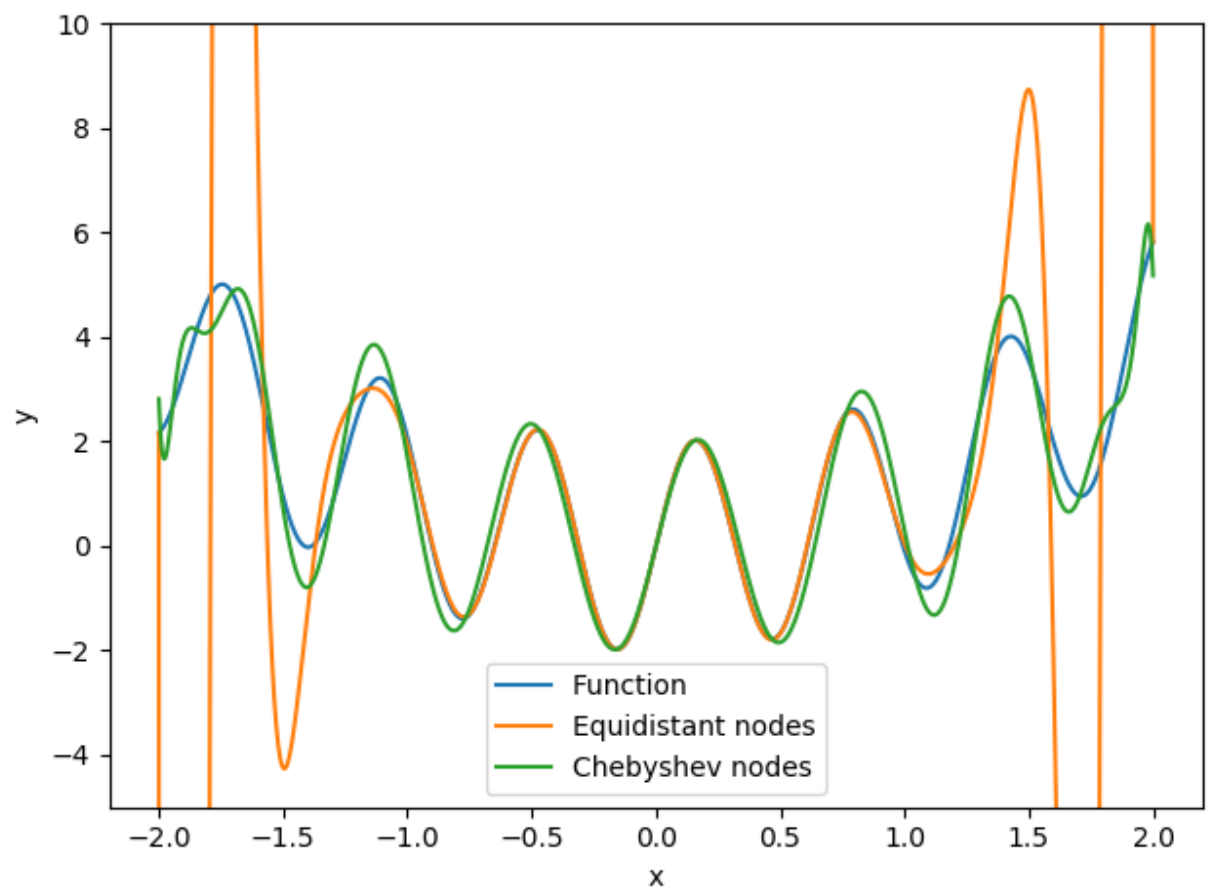
$$l_i(x) = \prod_{j=0, j \neq i}^n \frac{x - x_j}{x_i - x_j} = \frac{x - x_0}{x_i - x_0} \cdot \dots \cdot \frac{x - x_{i-1}}{x_i - x_{i-1}} \cdot \frac{x - x_{i+1}}{x_i - x_{i+1}} \cdot \dots \cdot \frac{x - x_n}{x_i - x_n}$$

Эксперименты

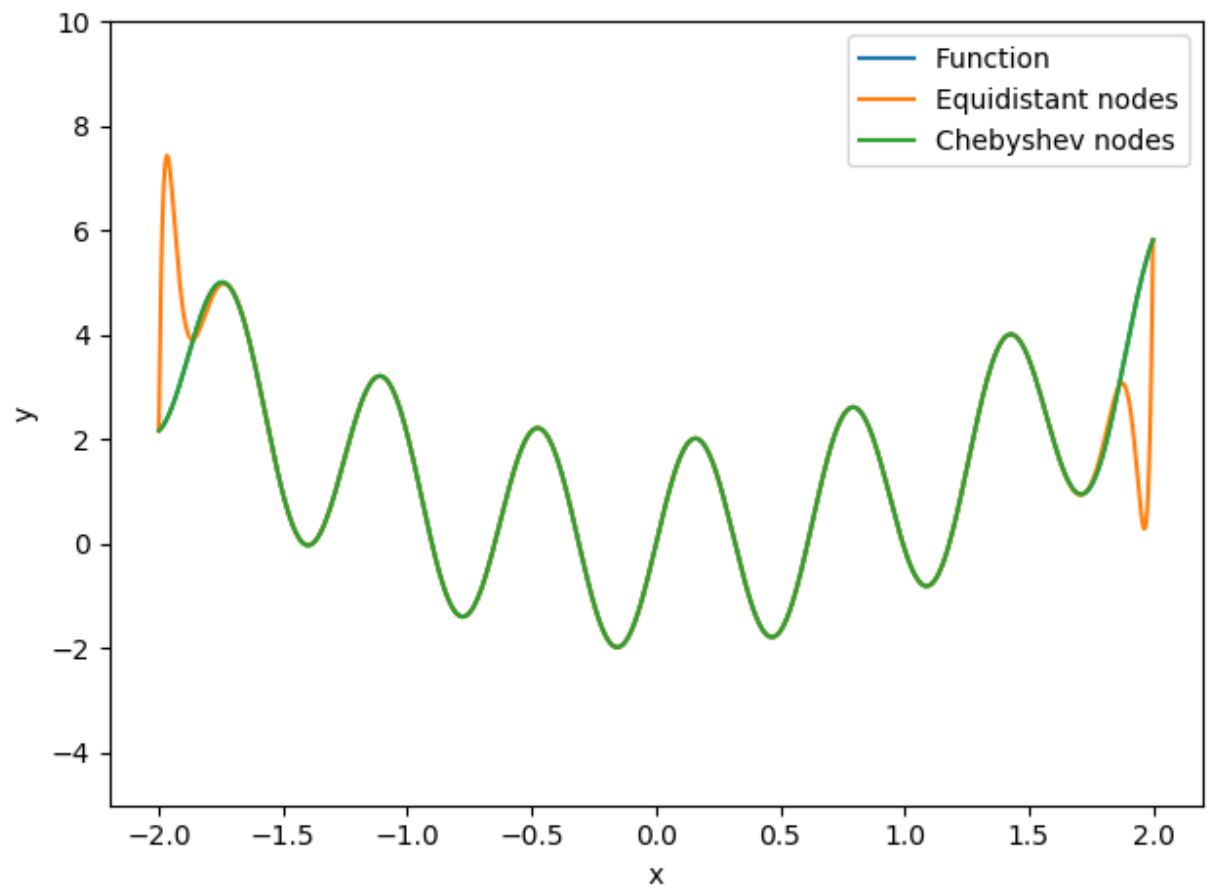
Эксперимент 1: $N=10$



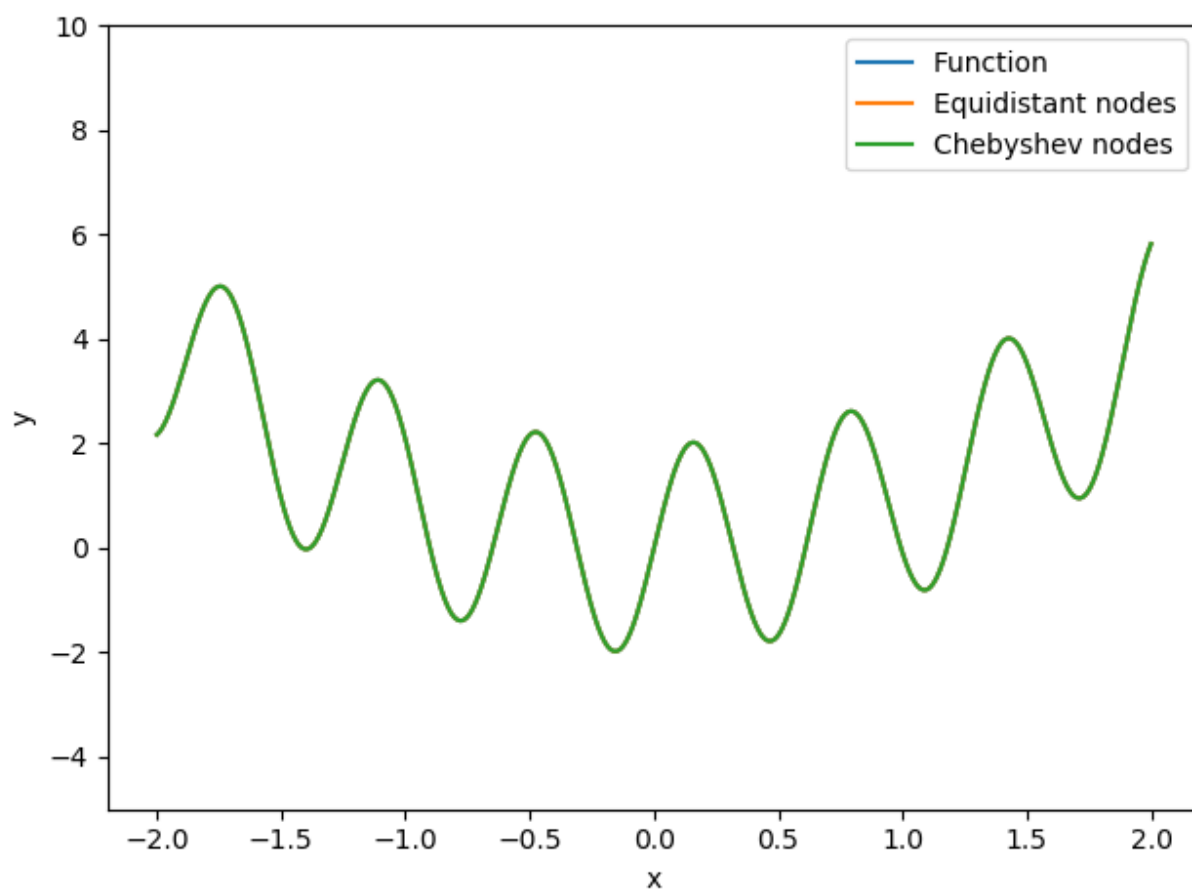
Эксперимент 2: $N=20$



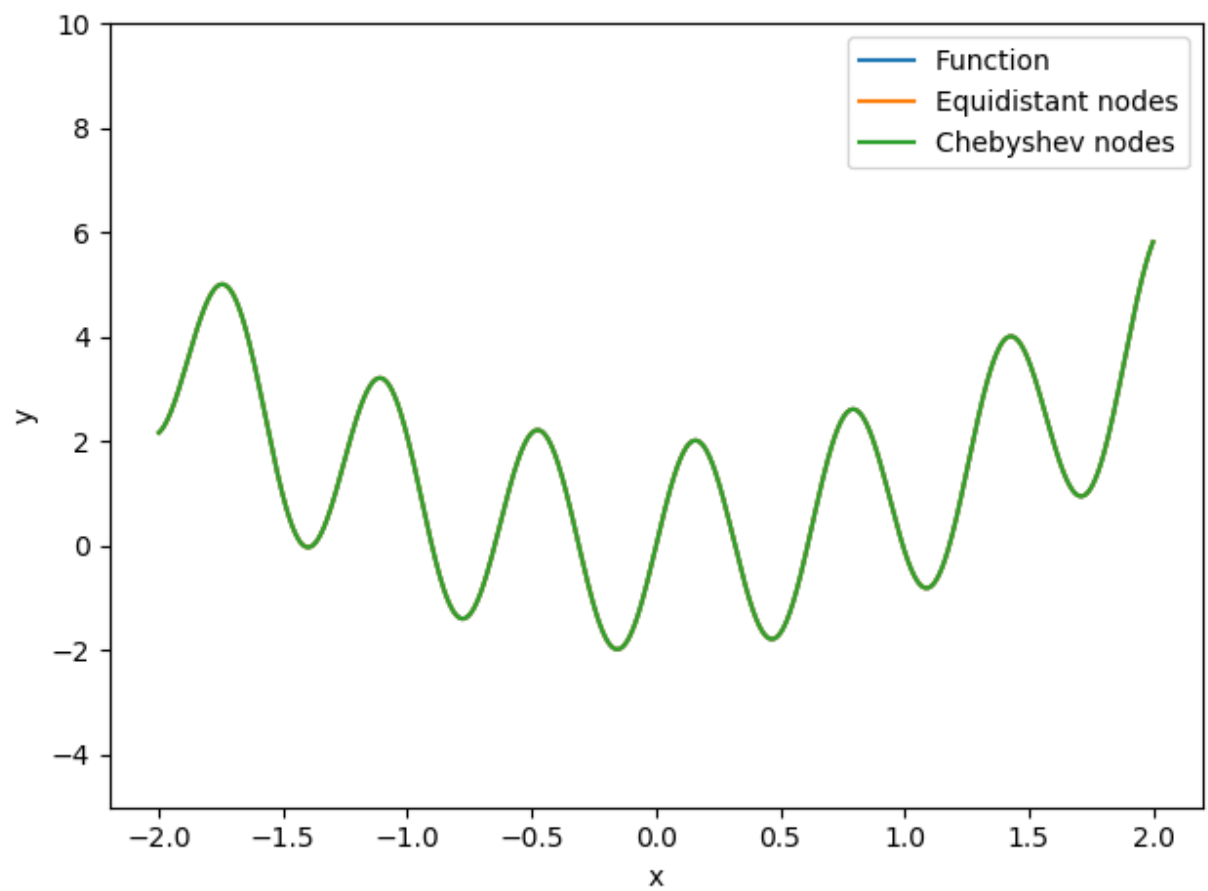
Эксперимент 3: $N=30$



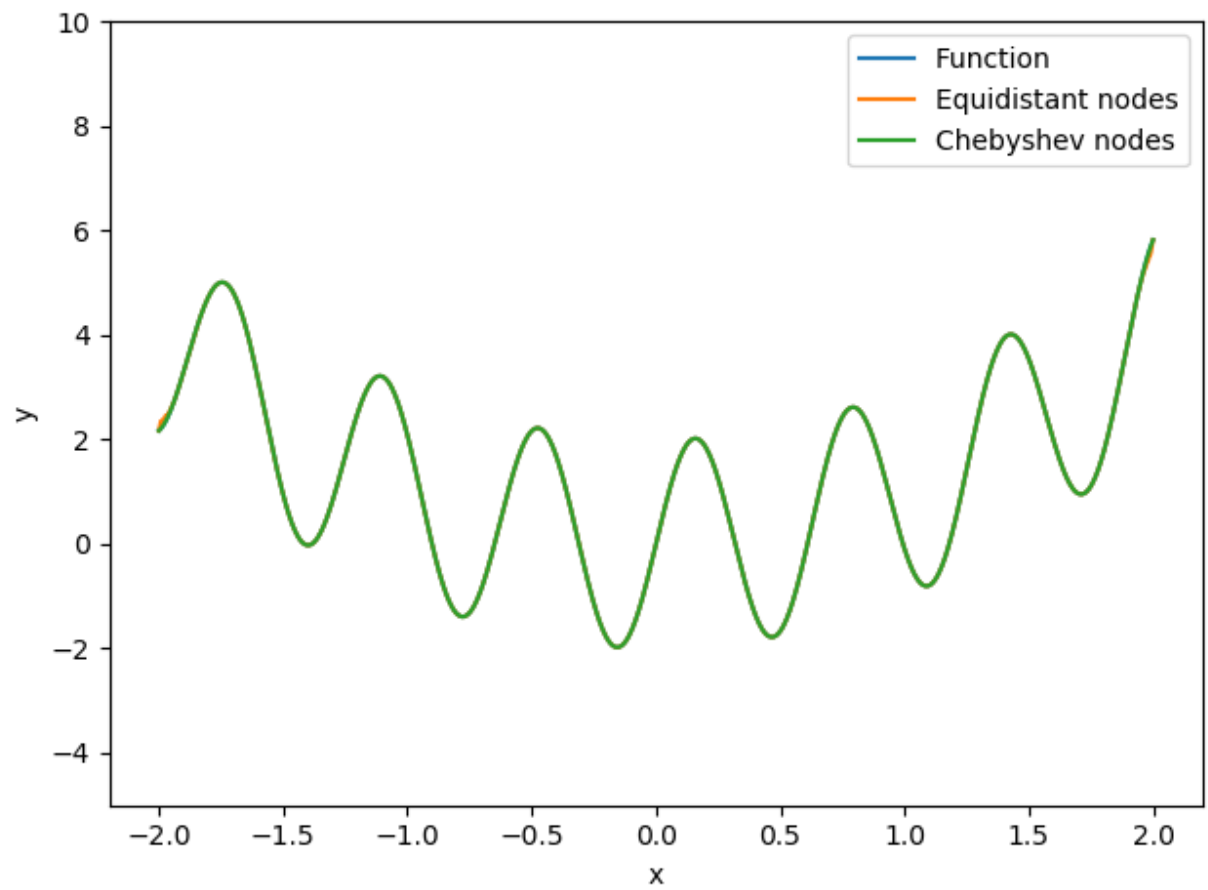
Эксперимент 4: $N=40$



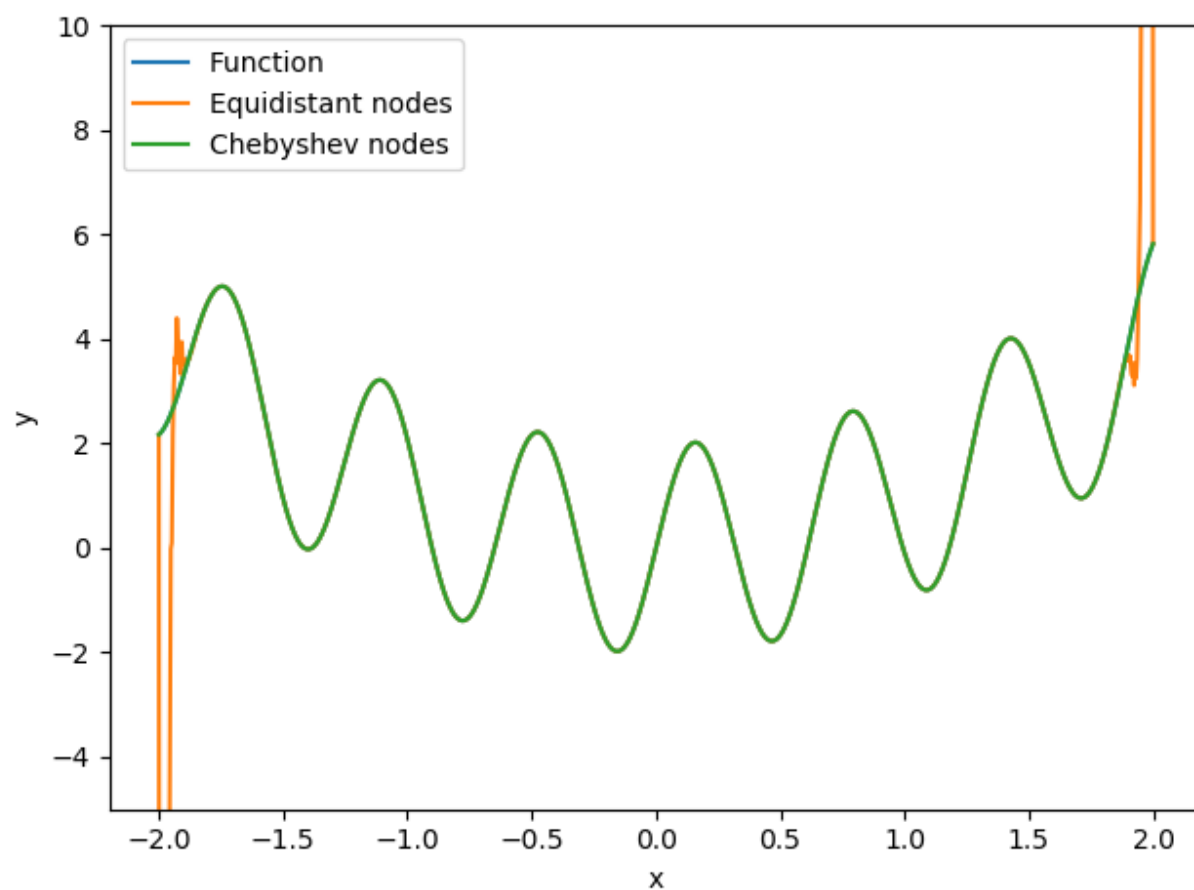
Эксперимент 5: $N=50$



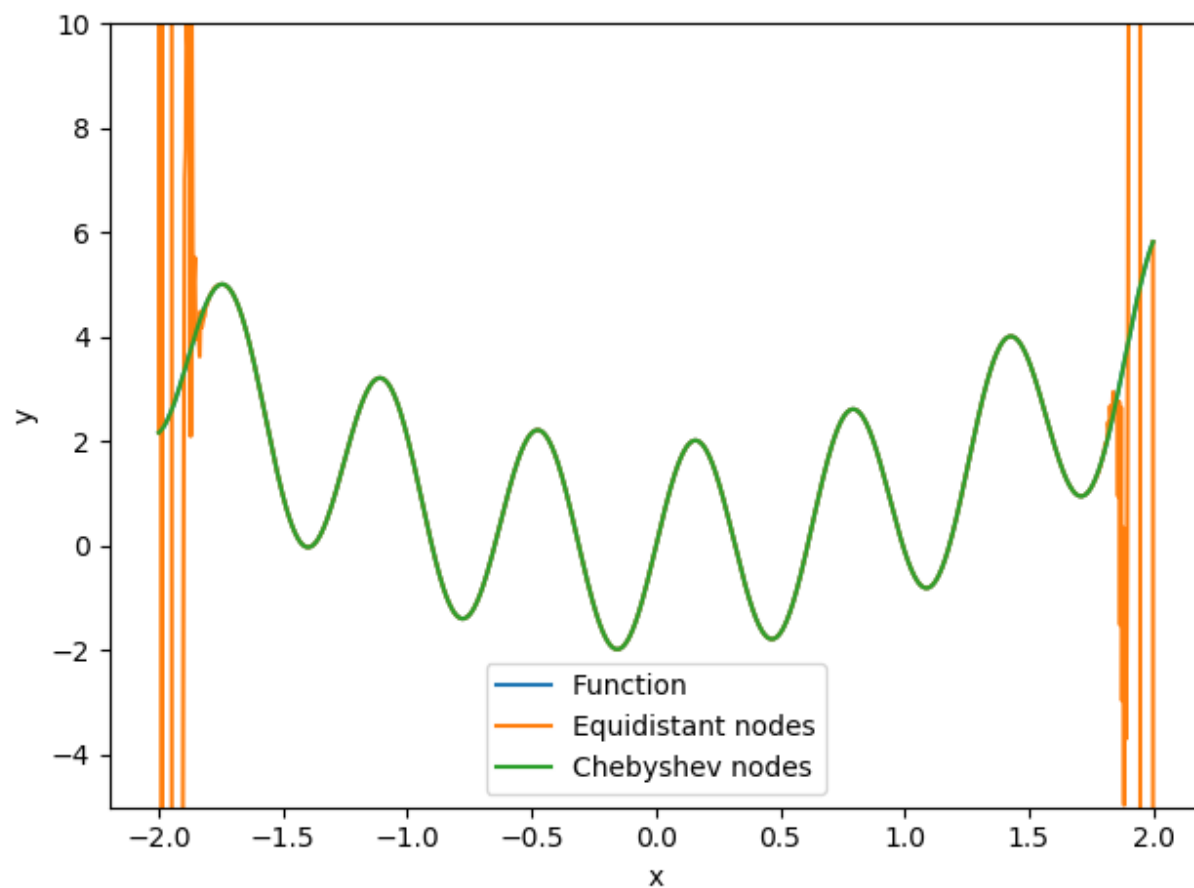
Эксперимент 6: $N=60$



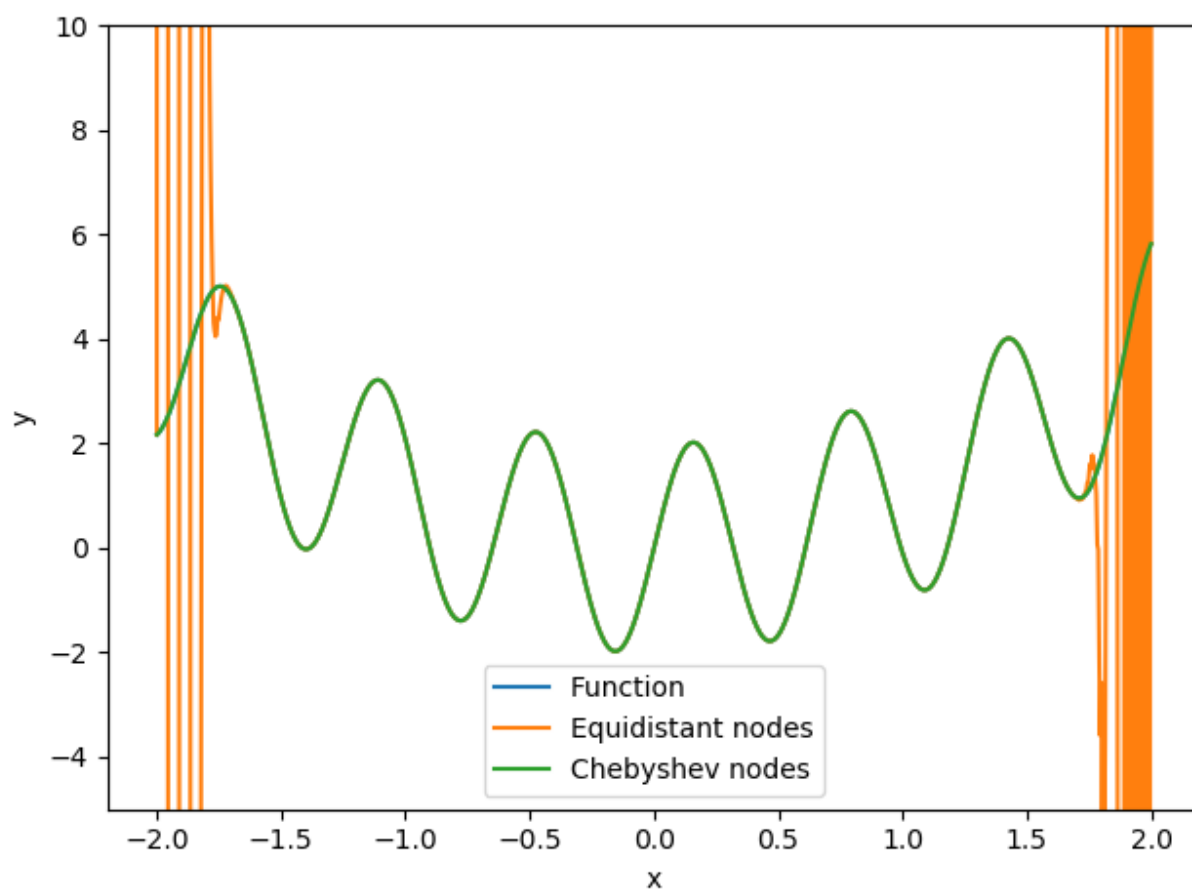
Эксперимент 7: $N=70$



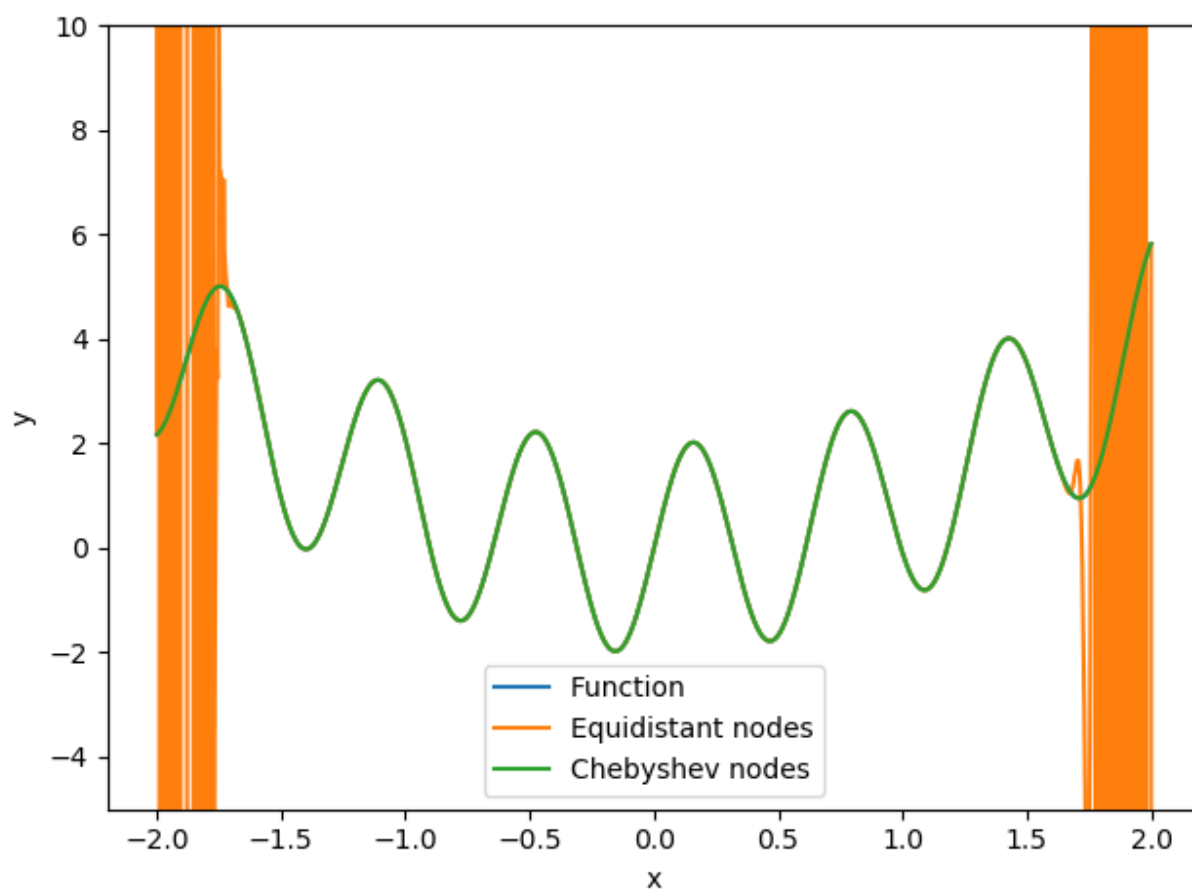
Эксперимент 8: $N=80$



Эксперимент 9: $N=90$



Эксперимент 10: $N=100$



Таблица

N	Норма (равноотстоящие узлы)	Норма (чебышевские узлы)
10	3.896402069083486320900196521	3.552805780443978746393810162
20	560.7618824201211474522041731	0.7933493227610869578103916405
30	5.045247810217645627955742629	0.0003222829086937284903180075772
40	0.0009155210912078368557074196955	1.937781432476186075422770500E-9
50	0.0001744064689795788817644306204	7.782160499475177812099456787E-15
60	0.1442166777573406210196898440	7.657918695709897374689579010E-15
70	76.15325304861984963531764922	7.202743809958643777728080750E-15
80	28864.79240602850214569597977	1.025649441915740193128585815E-14
90	29694666.89763907857011078567	7.369520648701719400405883789E-15
100	17572046324.40500604039642158	9.903770413542446947097778320E-15

Код решения

```
import numpy as np
import matplotlib.pyplot as plt
import math
from decimal import *

def function(x):
    return Decimal(x ** 2 + 2 * math.sin(10 * x))

def lagrange_polynomial(x, nodes):
    ans = Decimal(0)
    for lagrange_i in range(len(nodes)):
        basis = Decimal(1)
        for lagrange_j in range(len(nodes)):
            if lagrange_i != lagrange_j:
                basis *= Decimal((x - nodes[lagrange_j])) /
Decimal((nodes[lagrange_i] - nodes[lagrange_j]))
        ans += function(nodes[lagrange_i]) * basis
    return ans

a = -2
b = 2
N = 100
equidistant_nodes = np.linspace(a, b, N)
chebyshev_nodes = []
for i in range(1, N + 1):
    chebyshev_nodes.append((a + b) / 2 + 0.5 * (b - a) * math.cos((2 * i - 1) / (2 *
N) * math.pi))
chebyshev_nodes.reverse()
plot_x = np.linspace(a, b, 1000)
function_y = []
equidistant_y = []
chebyshev_y = []
equidistant_norm = 0.0
chebyshev_norm = 0.0
for i in plot_x:
    function_y.append(function(i))
    equidistant_P = lagrange_polynomial(i, equidistant_nodes)
    chebyshev_P = lagrange_polynomial(i, chebyshev_nodes)
    equidistant_y.append(equidistant_P)
    chebyshev_y.append(chebyshev_P)
    equidistant_norm = max(abs(function(i) - equidistant_P), equidistant_norm)
    chebyshev_norm = max(abs(function(i) - chebyshev_P), chebyshev_norm)
plt.plot(plot_x, function_y, label="Function")
plt.plot(plot_x, equidistant_y, label="Equidistant nodes")
plt.plot(plot_x, chebyshev_y, label="Chebyshev nodes")
plt.legend()
plt.xlabel("x")
plt.ylabel("y")
plt.ylim(-5, 10)
plt.show()
print("Norm for equidistant nodes is " + str(equidistant_norm))
print("Norm for Chebyshev nodes is " + str(chebyshev_norm))
```