

# **ЛАБОРАТОРНАЯ РАБОТА №3**

**«ИНТЕРПОЛИРОВАНИЕ:**

**КУБИЧЕСКИЕ СПЛАЙНЫ»**

***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 кубическими сплайнами на отрезке  $[-2, 2]$  по равноотстоящим узлам с естественными граничными условиями.
- 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  $|f(x_i) - S(x_i)|, i = 1, \dots, 1000$ . При каждом вычислении нормы замерять время с точностью до миллисекунды.

## Теоретические сведения

$$s_i(x) = \alpha_i + \beta_i(x - x_i) + \frac{\gamma_i}{2}(x - x_i)^2 + \frac{\delta_i}{2}(x - x_i)^3$$

$$\alpha_i = y_i, i = \overline{1, n}$$

$$\beta_i = \frac{y_i - y_{i-1}}{h_i} + \frac{2\gamma_i - \gamma_{i-1}}{6}h_i$$

$$\delta_i = \frac{\gamma_i - \gamma_{i-1}}{h_i}, i = \overline{2, n}$$

Если  $s''(a) = s''(b) = 0$ , то

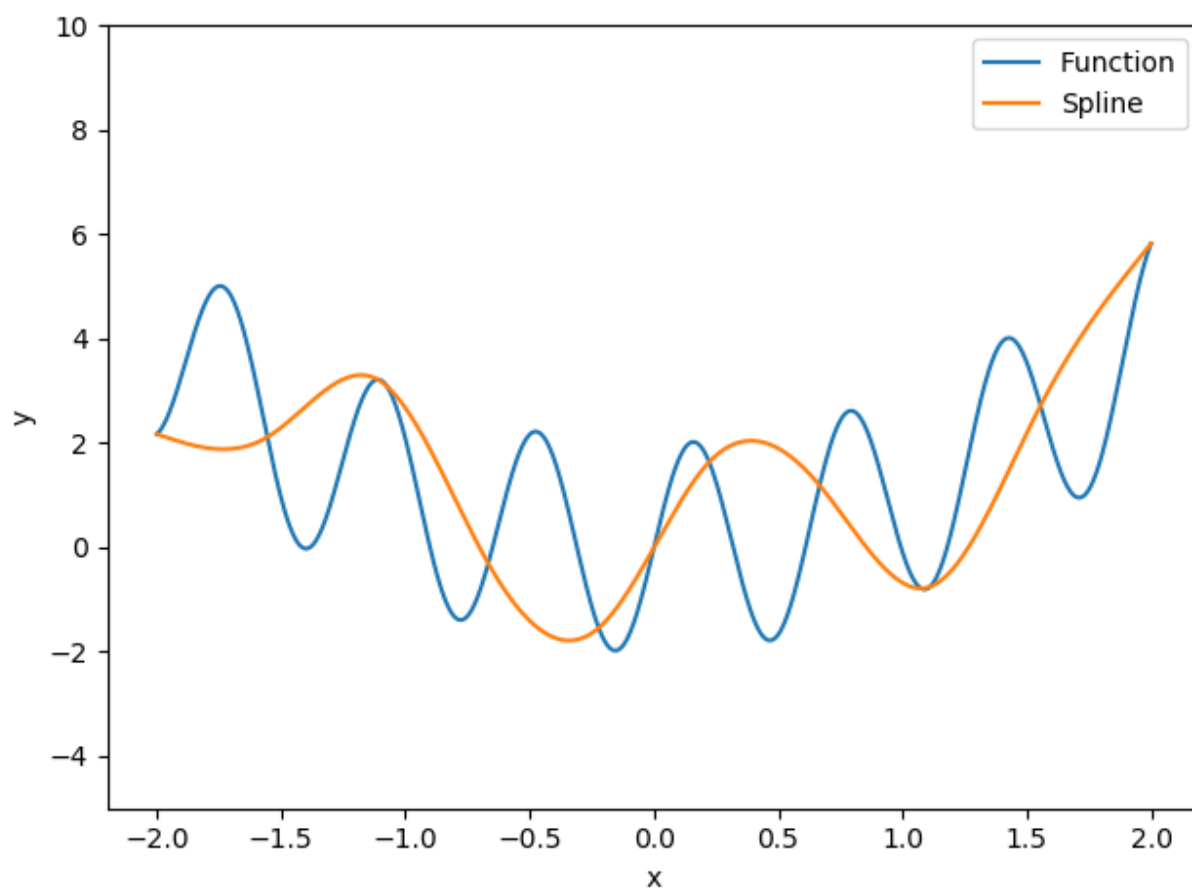
$$\gamma_0 = \gamma_n = 0$$

$$\begin{bmatrix} 2 & e_1 & & & \\ c_2 & 2 & e_2 & & \\ & c_3 & 2 & e_3 & \\ & & \ddots & \ddots & \ddots \\ & & & c_{n-2} & 2 & e_{n-2} \\ & & & & c_{n-1} & 2 \end{bmatrix} \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \\ \vdots \\ \gamma_{n-2} \\ \gamma_{n-1} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_{n-2} \\ b_{n-1} \end{bmatrix},$$

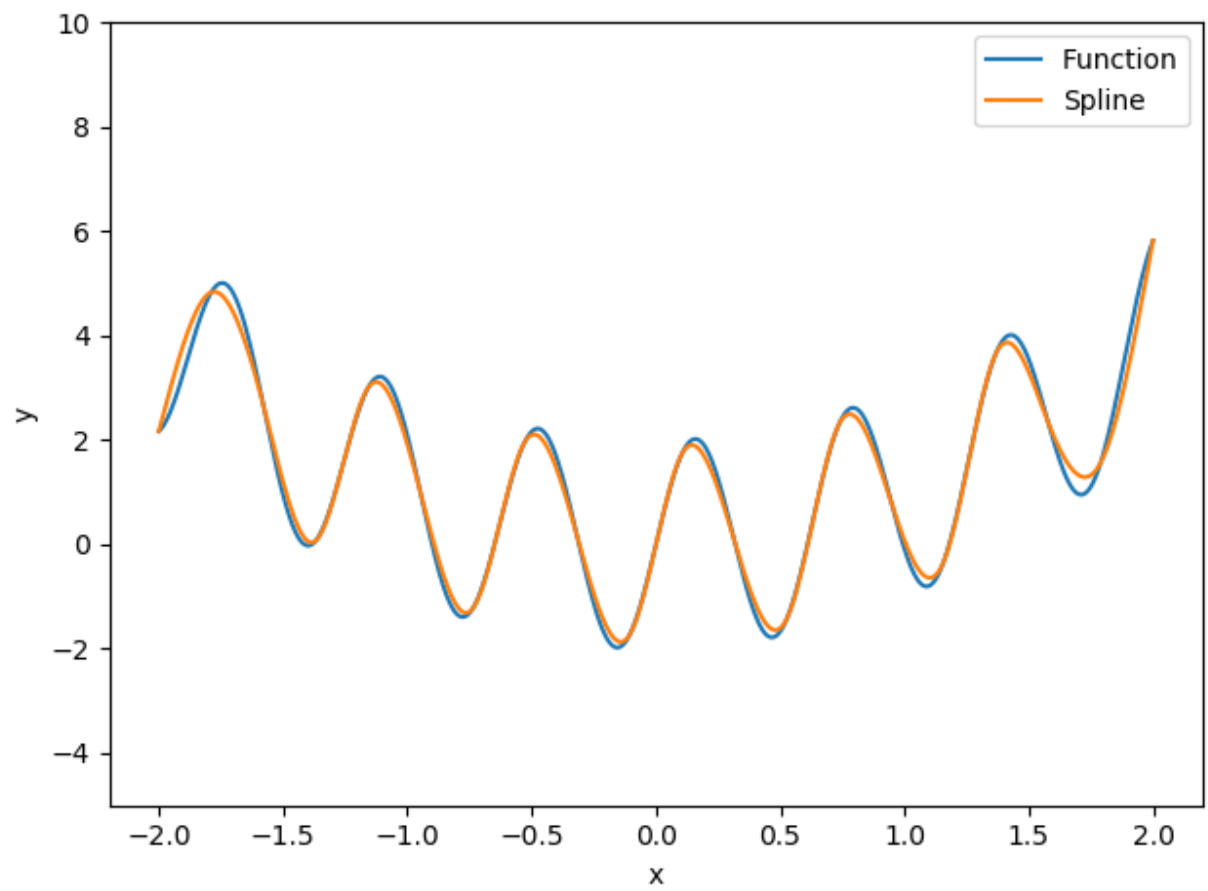
$$c_i = \frac{h_i}{h_i + h_{i+1}}, e_i = \frac{h_{i+1}}{h_i + h_{i+1}}, b_i = 6f[x_{i-1}, x_i, x_{i+1}]$$

# Эксперименты

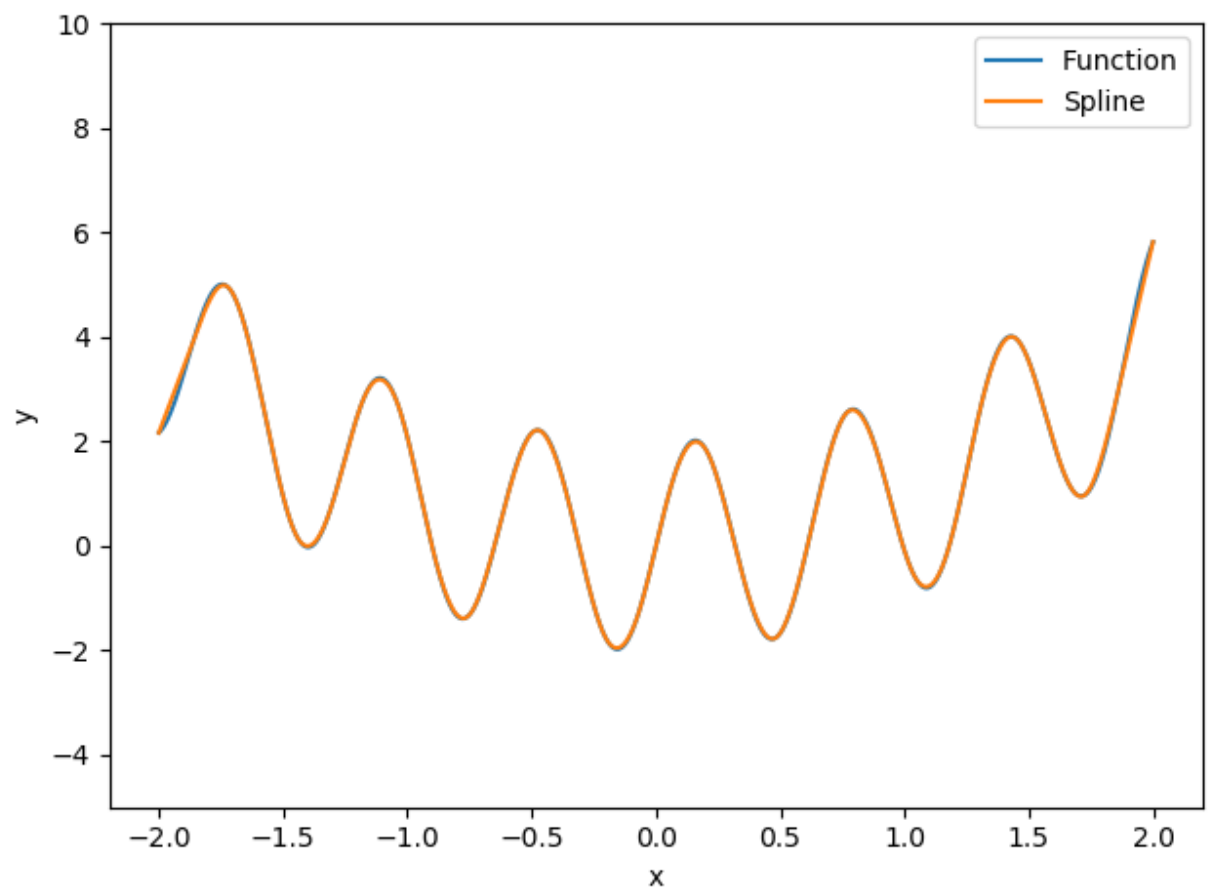
## Эксперимент 1: $N=10$



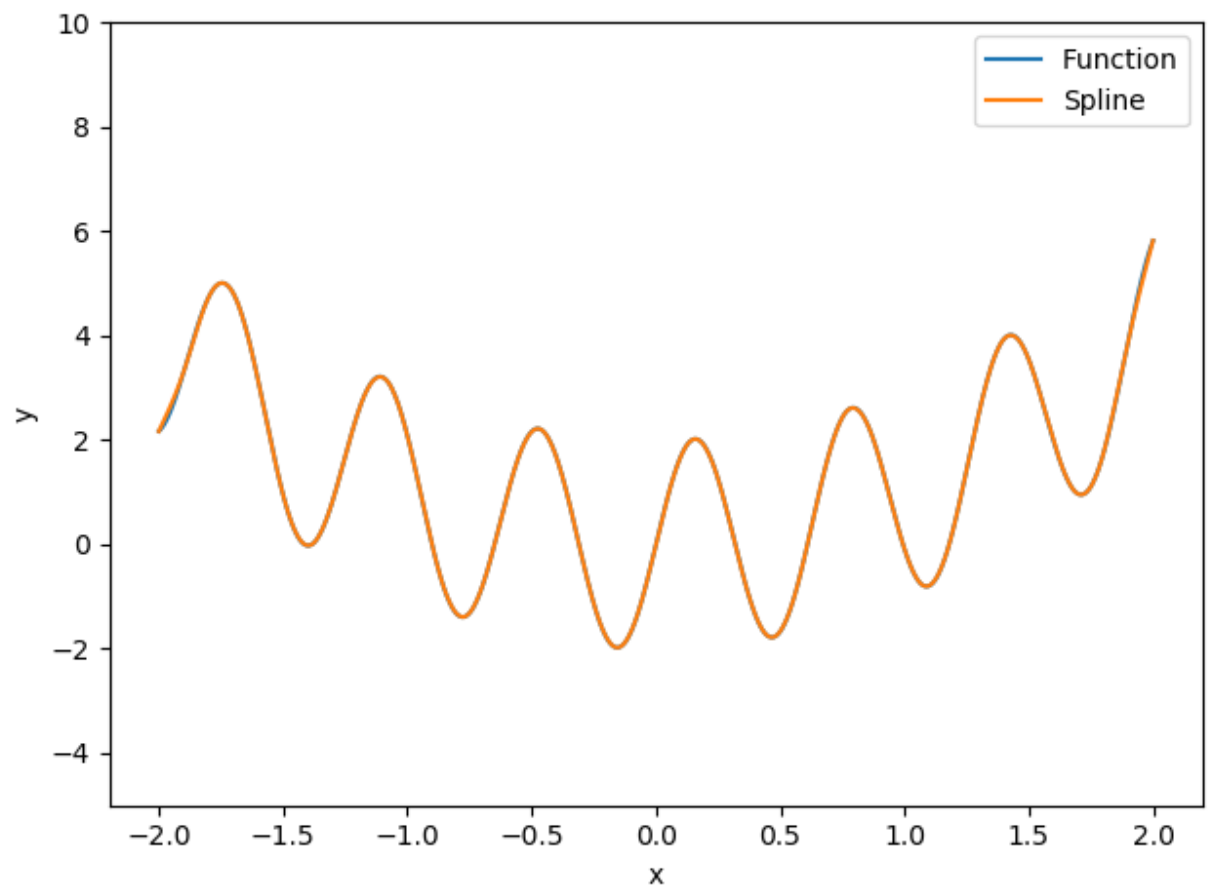
## Эксперимент 2: $N=20$



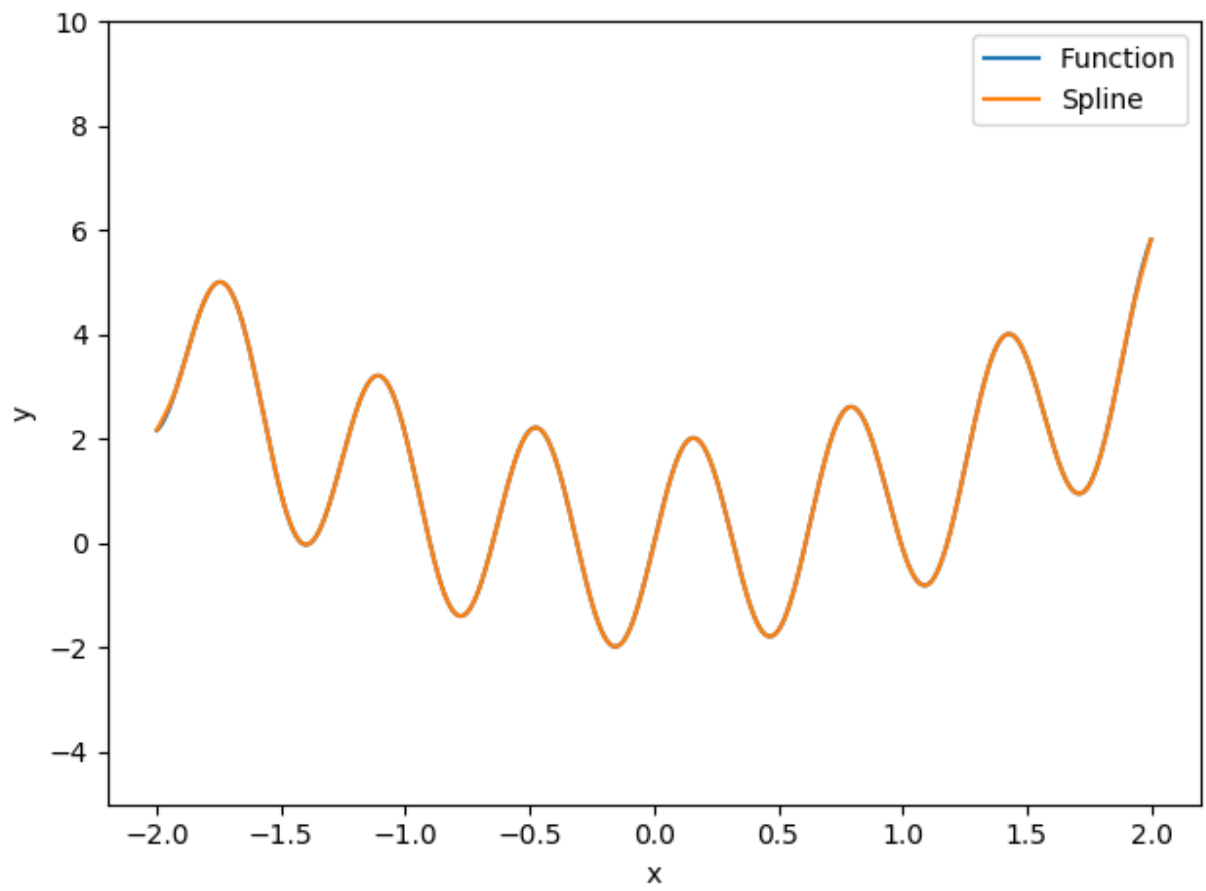
### Эксперимент 3: $N=30$



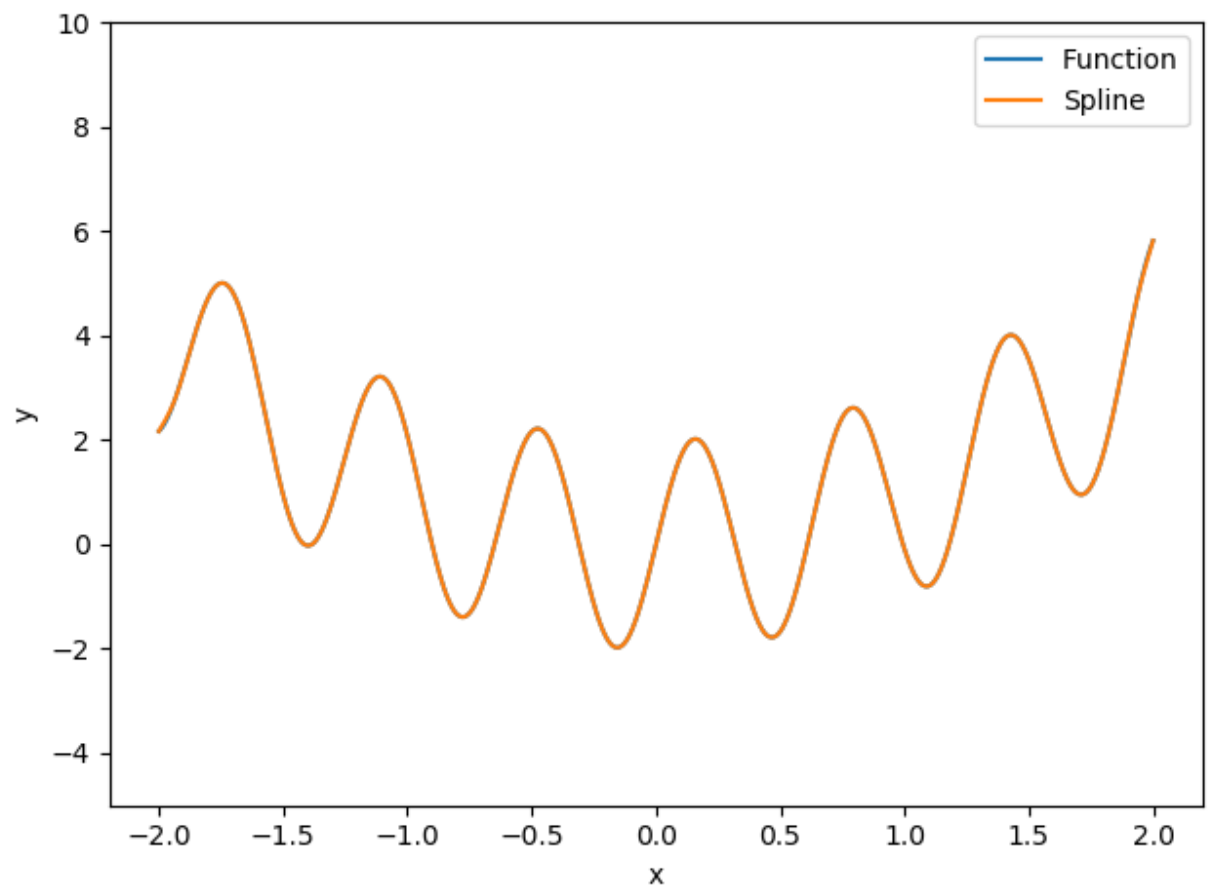
#### Эксперимент 4: $N=40$



## Эксперимент 5: $N=50$

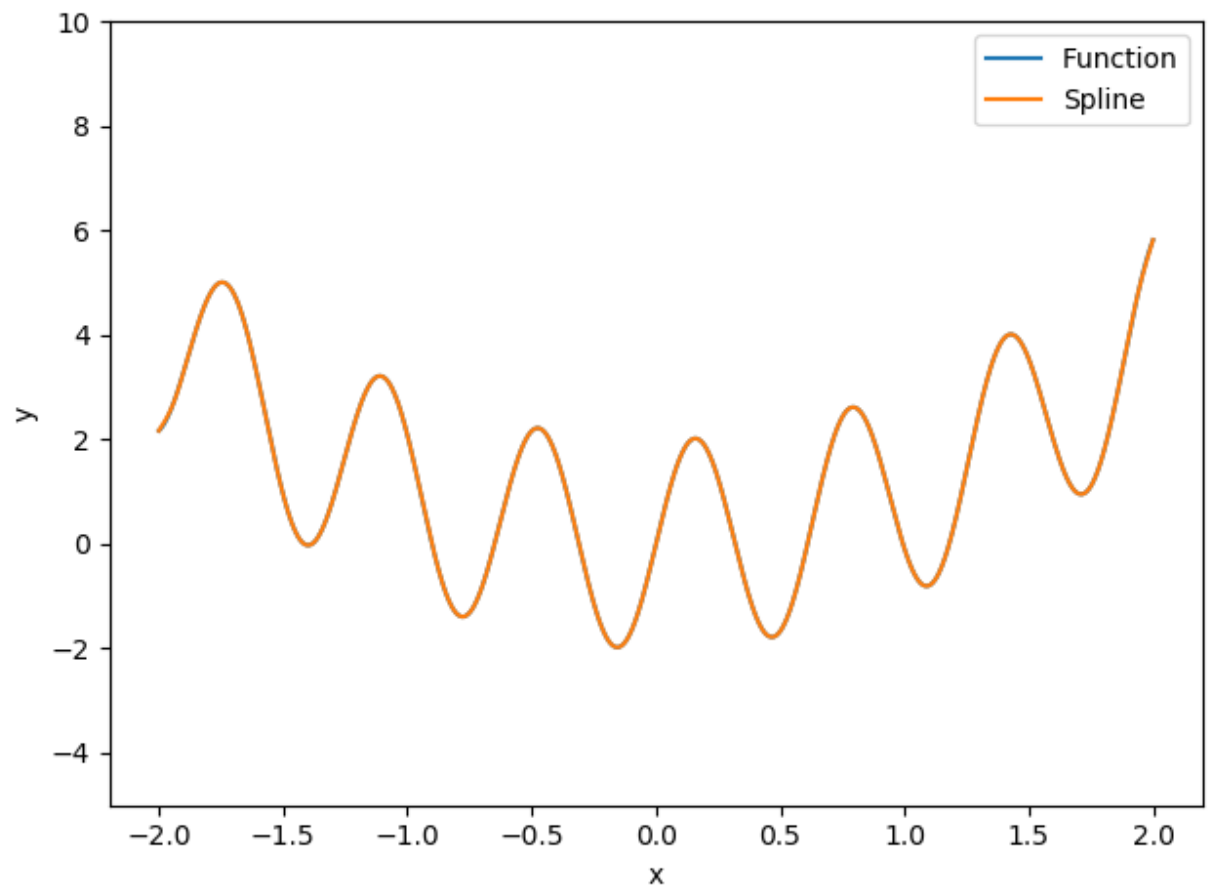


## Эксперимент 6: $N=60$

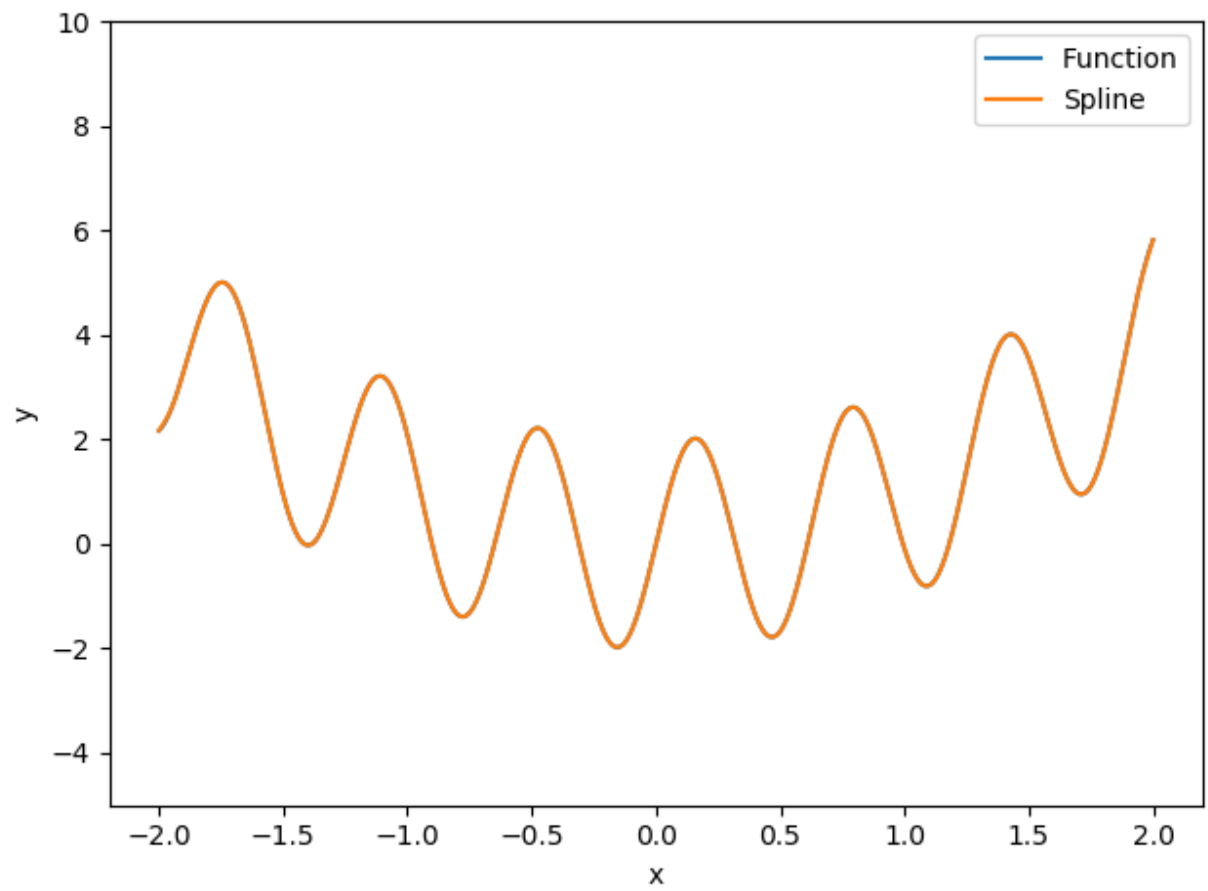




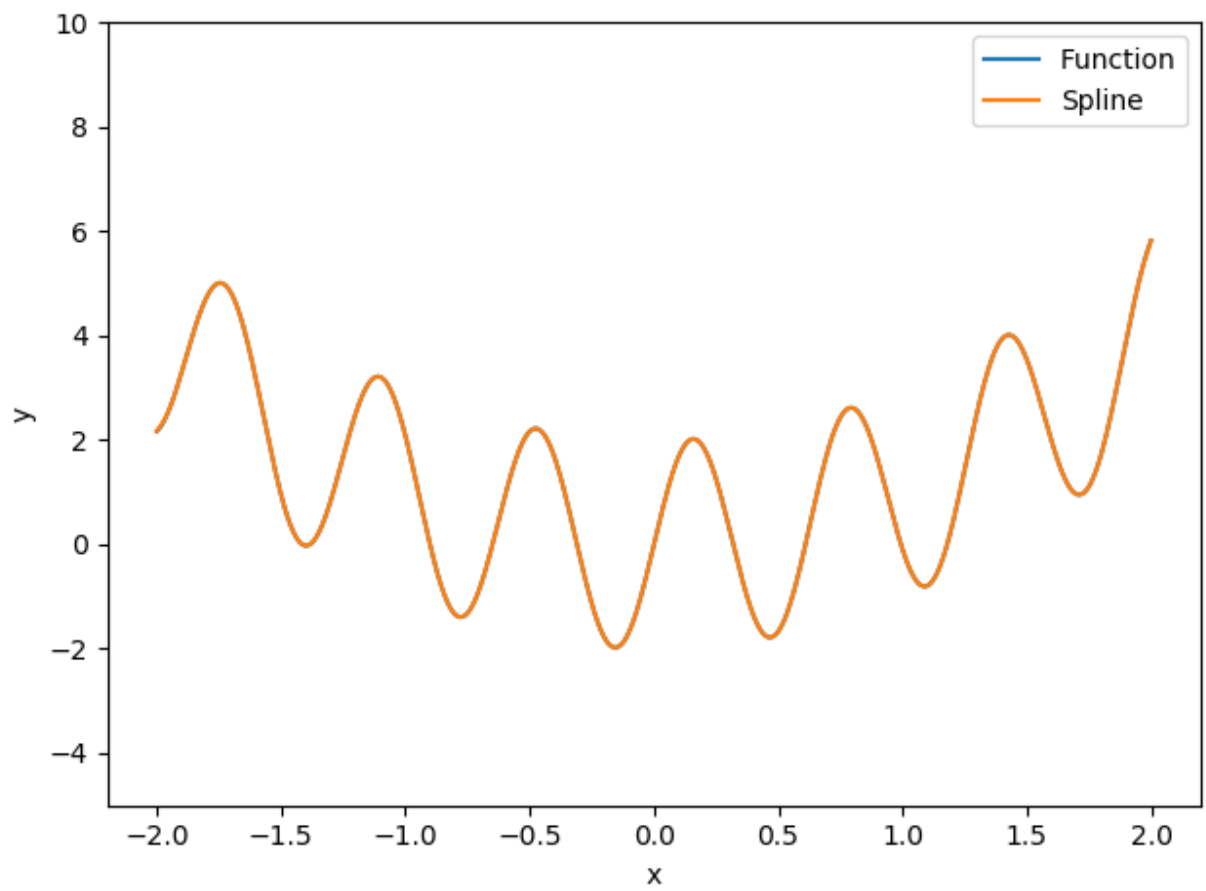
## Эксперимент 7: $N=70$



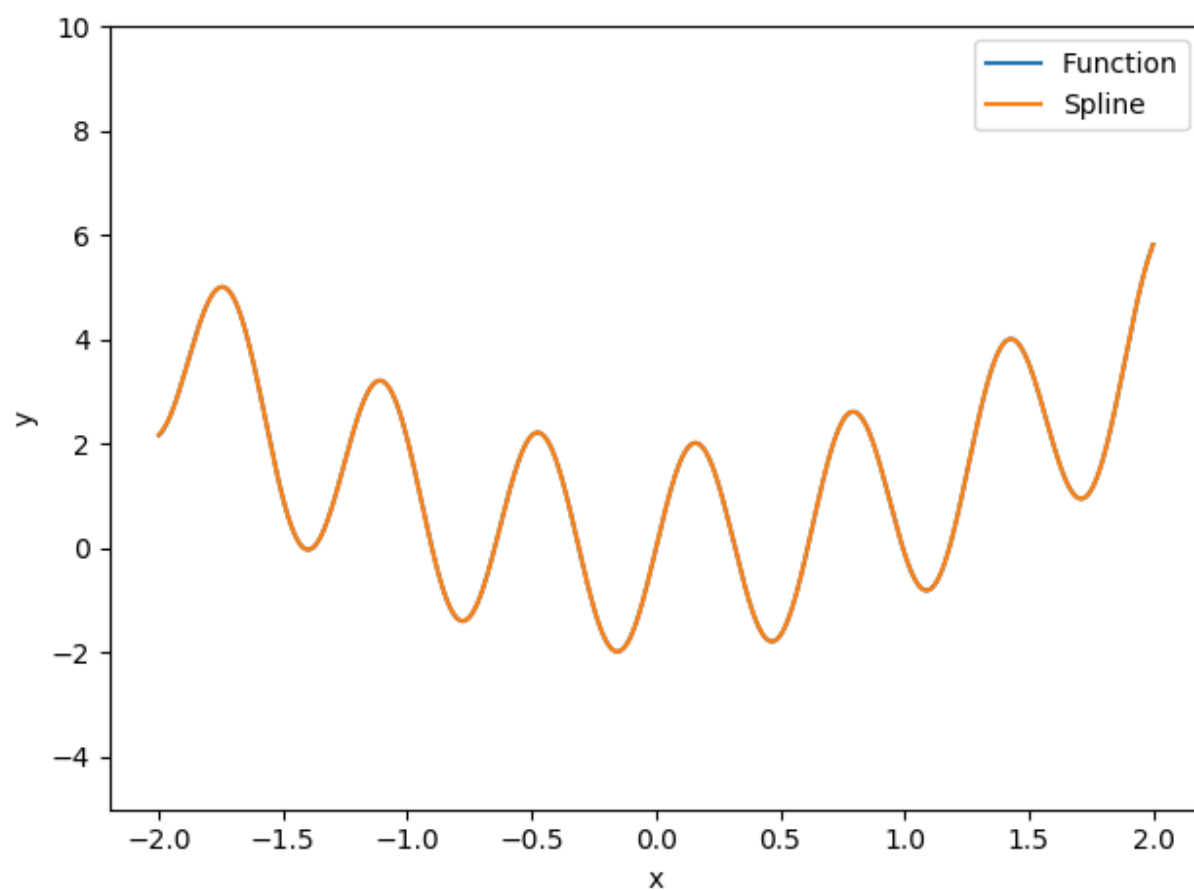
## Эксперимент 8: $N=80$



## Эксперимент 9: $N=90$



## Эксперимент 100: $N=100$



**Таблица**

N	Норма(сплайн)	Норма(Лагранж)	Время- Слайн с.	Время- Лагранж с.
10	3.76552664778278	3.552805780443978746393810162	0.008	0.617
20	0.557640360856308	0.79334932276108695781039164	0.009	2.309
30	0.210506416758476	0.000322282908693728490318008	0.010	4.952
40	0.107407969377276	0.000000001937781432476186115	0.010	7.990
50	0.06528563962244949	0.0000000000000007782160499475	0.009	12.462
60	0.04381757548167986	0.0000000000000007657918695710	0.010	17.850
70	0.03156282452157111	0.0000000000000007202743809959	0.010	24.186
80	0.023951305994033945	0.00000000000000010256494419157	0.009	32.746
90	0.01869770696849704	0.0000000000000007369520648702	0.009	42.142
100	0.015078642548047139	0.0000000000000009903770413542	0.009	50.419

## Код решения

```
import math
import numpy as np
from matplotlib import pyplot as plt
import time

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

def divided_difference(x1, x2, x3):
    f1 = (function(x2) - function(x1)) / (x2 - x1)
    f2 = (function(x3) - function(x2)) / (x3 - x2)
    return (f2 - f1) / (x3 - x1)

def spline(alpha, beta, gamma, delta, node, x):
    return alpha + beta * (x - node) + gamma * ((x - node) ** 2) / 2 + delta * ((x - node) ** 3) / 6

a = -2.0
b = 2.0
N = 100
h = (b - a) / (N - 1)
alpha_list = [0.0]
beta_list = [0.0]
gamma_list = np.array([0.0])
delta_list = [0.0]
nodes = np.linspace(a, b, N)
A = np.array([np.array([0.0 for i in range(N - 2)]) for j in range(N - 2)])
for i in range(N - 2):
    A[i][i] = 2
    if i > 0:
        A[i][i - 1] = 0.5
    if i + 1 < N - 2:
        A[i][i + 1] = 0.5
B = np.array([])
for i in range(N - 2):
    B = np.append(B, 6 * divided_difference(nodes[i], nodes[i + 1], nodes[i + 2]))
t = time.time()
gamma_list = np.append(gamma_list, np.linalg.solve(A, B))
gamma_list = np.append(gamma_list, 0.0)
for i in range(1, N):
    beta_list.append(((function(nodes[i]) - function(nodes[i - 1])) / h))
    beta_list[i] += (2 * gamma_list[i] + gamma_list[i - 1]) * h / 6
    delta_list.append((gamma_list[i] - gamma_list[i - 1]) / h)
    alpha_list.append(function(nodes[i]))
plot_x = np.linspace(a, b, 1000)
function_y = []
spline_y = []
pos = 1
norm = 0.0
for i in plot_x:
    if i > nodes[pos]:
        pos += 1
    f = function(i)
    s = spline(alpha_list[pos], beta_list[pos], gamma_list[pos], delta_list[pos], nodes[pos], i)
```

```
        function_y.append(f)
        spline_y.append(s)
        norm = max(norm, abs(f - s))
    print("Time is -", "%.3f" % (time.time() - t), ' ')
    plt.plot(plot_x, function_y, label="Function")
    plt.plot(plot_x, spline_y, label="Spline")
    plt.legend()
    plt.xlabel("x")
    plt.ylabel("y")
    plt.ylim(-5, 10)
    plt.show()
    print("Norm is -", norm, ' ')
```