БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет прикладной математики и информатики

Отчет

Методы численного анализа

Лабораторная работа 2

Выполнила

Юрковская Екатерина Артуровна

Студентка 2 курса 3 группы

Минск 2020

1) Постановка задачи.

Постановка задачи. Для заданной функции $f : [a, b] \to \mathbb{R}$ требуется:

- Произвести интерполяцию многочленом в указанной форме на отрезке $[-2, 2]$.
- Интерполирование следует проводить как по равноотстоящим узлам, так и по чебышевским.
- Для каждого типа узлов построить графики получившихся приближений для сеток с количеством узлов, равным $N_i = 10i, \quad i = 1, 2, \ldots, 10$. 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 $|f(x_i) - P(x_i)|, \quad i = 1, \ldots, 1000$. Результат представить в виде таблицы:

N	Норма (равноотстоящие узлы)	Норма (чебышовские узлы)
10		
20		
...	...	...
100		

Примечание: $f(x) = (\cos x)^2 - x$, ИМ в барицентрической форме

2) Теоретические сведения

2.1. Многочлен Лагранжа в барицентрической форме.

Запишем многочлен Лагранжа в виде:

$$P_n(x) = \sum_{i=0}^{n} y_i \frac{\omega_{n+1}(x)}{(x - x_i)\omega'_{n+1}(x_i)}, \qquad (5.37)$$

$$\omega_{n+1}(x) = (x - x_0)(x - x_1) \ldots (x - x_n).$$

Чтобы избежать громоздких умножений в каждом слагаемом формулы (5.37), достаточно просто вынести общий множитель и записать в следующем виде:

$$P_n(x) = \omega_{n+1}(x) \sum_{i=0}^{n} y_i \frac{v_i}{x - x_i}, \qquad (5.38)$$

где

$$v_i = \frac{1}{\omega'_{n+1}(x_i)} = \frac{1}{\prod_{j \neq i}(x_i - x_j)}, \quad i = \overline{0, n}. \qquad (5.39)$$

Формула (5.38) называется первой формой барицентрической интерполяционной формулы. Существует вторая, еще более эффективная форма записи барицентрической формулы. Строится она так: строим по формуле (5.38) интерполяционный многочлен для f(x) = 1, который, очевидно, тождественно равен 1. То есть,

$$\omega_{n+1}(x) \sum_{i=0}^{n} \frac{v_i}{x - x_i} = 1.$$

Тогда P(x) можно записать в виде

$$\boxed{P_n(x) = \frac{\sum_{i=0}^{n} y_i \frac{v_i}{x - x_i}}{\sum_{i=0}^{n} \frac{v_i}{x - x_i}}}$$

Эту формулу называют просто барицентрической интерполяционной формулой. [1]

2.2) Интерполирование по чебышевским узлам.

Для вычисления узлов интерполирования будем использовать формулу [2] :

$$\boxed{x_i = \frac{a + b}{2} + \frac{b - a}{2} \cos \frac{\pi(2i + 1)}{2n + 2}, \quad i = \overline{0, n}.}$$

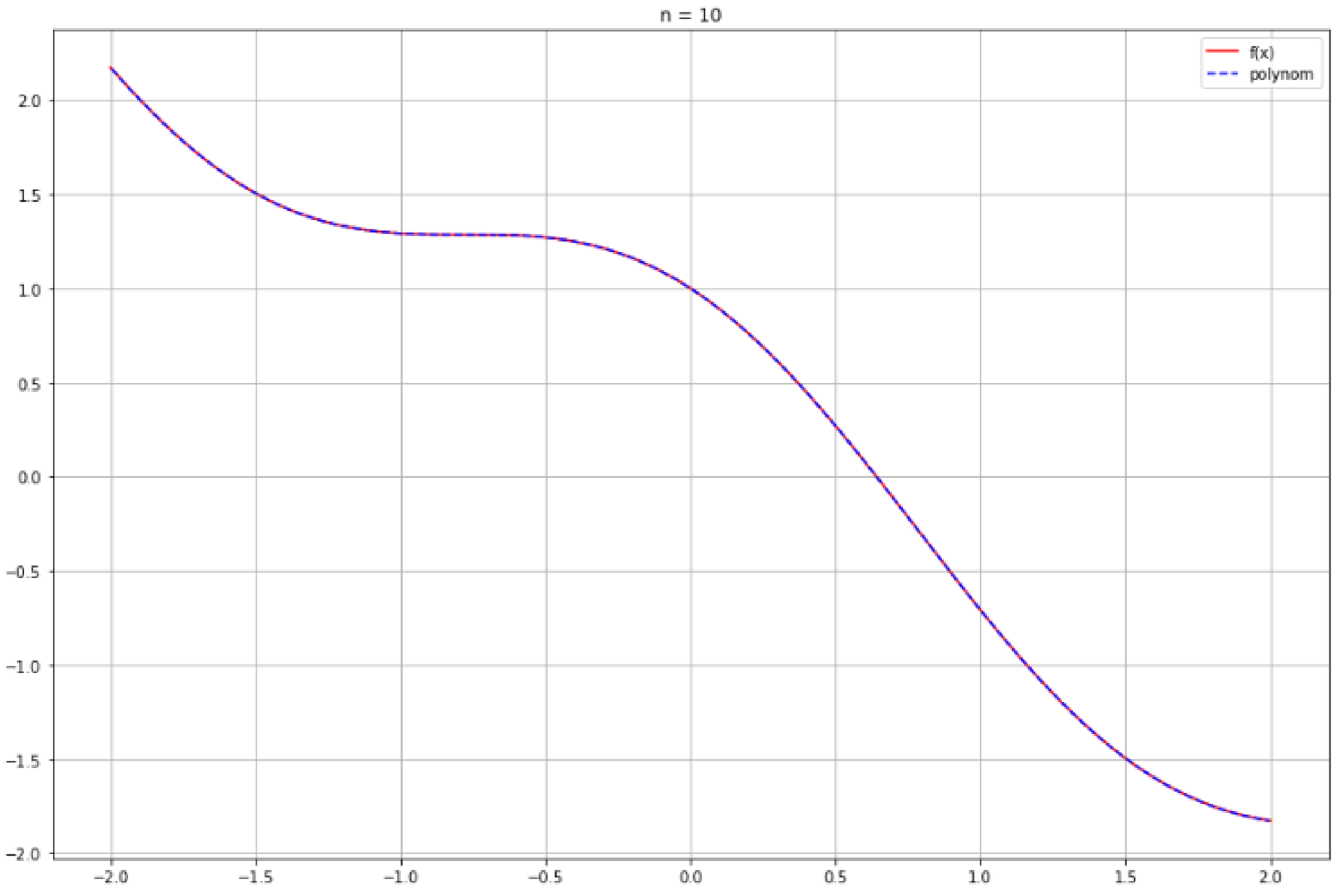
3) Исходный код программы.

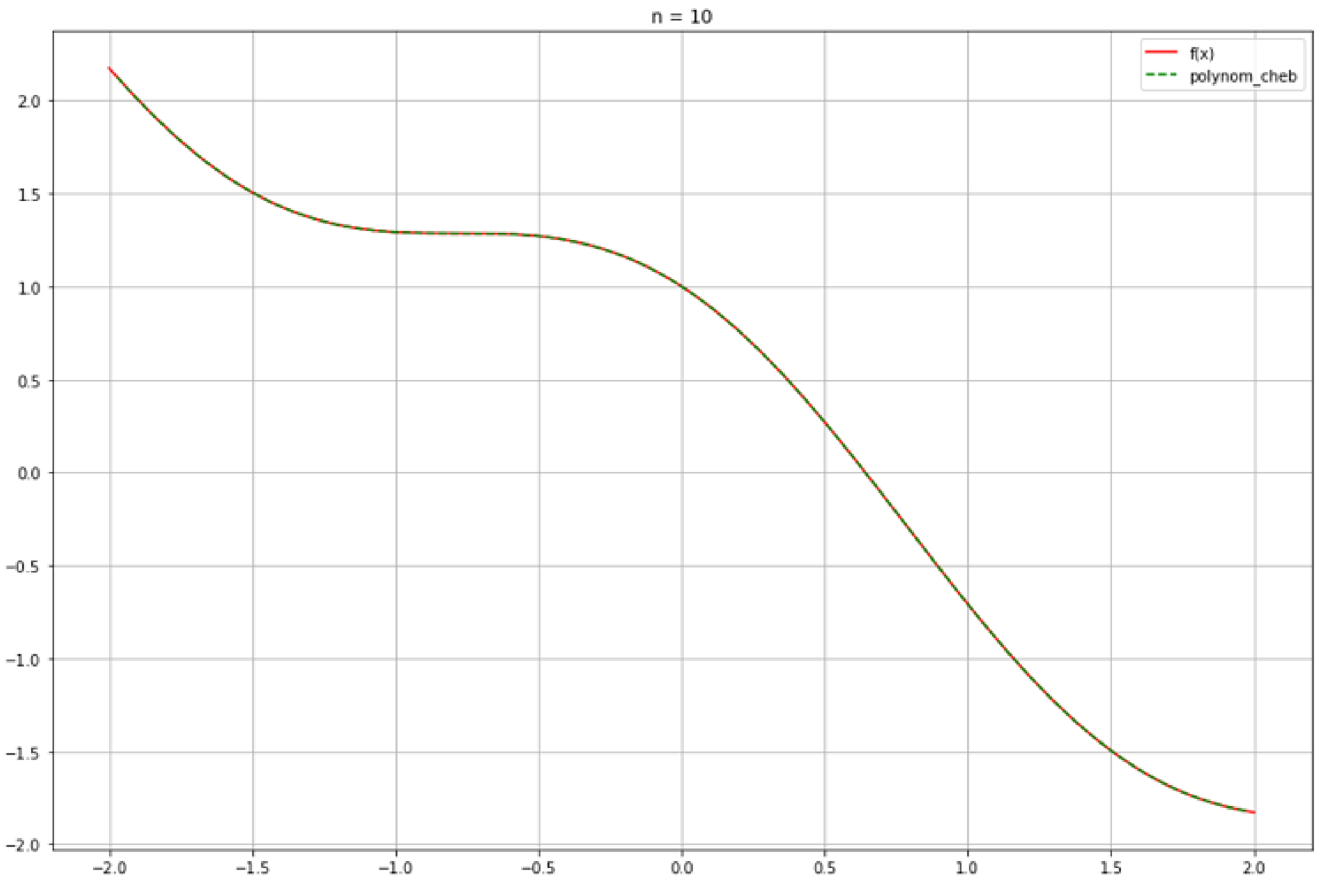
```
import numpy as np
import math
import matplotlib.pyplot as plt
from pylab import *

#исходная функция
func_x = np.linspace(-2.0, 2.0, 1000)
func_y = [math.cos(x)* math.cos(x) - x for x in func_x]
```

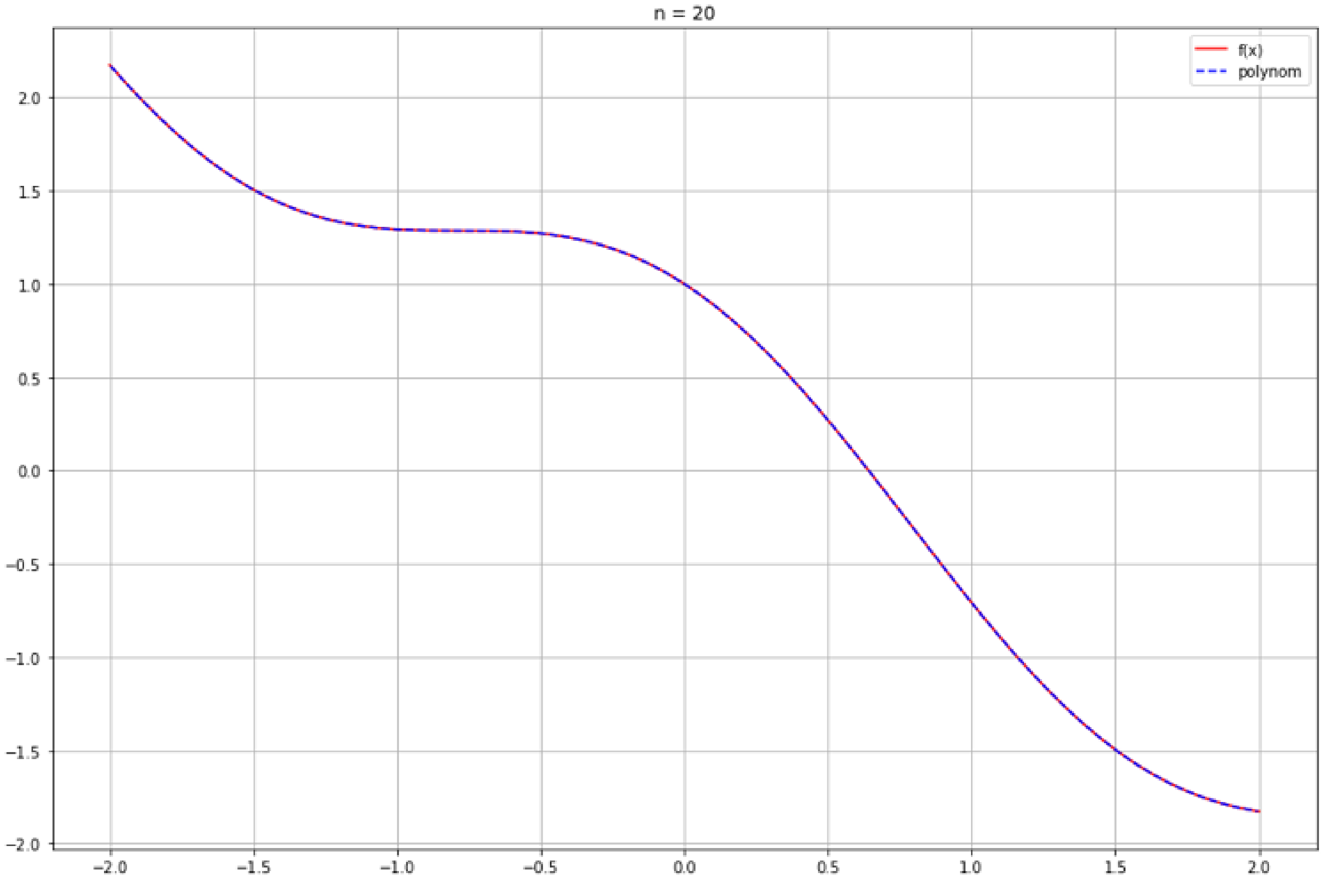
```
# вычисление интерполяционного полинома в барицентрической форме
def bar_form(x, y, val):
    v=[]
    for i in range(len(x)):
        x_=1.0
        for j in range(len(x)):
            if i == j:
                continue
            x_*= x[i]-x[j]
        v.append(1.0/x_)

    chisl = 0; znam = 0
    for i in range(len(y)):
            koef=v[i]/(val-x[i])
            znam+=koef
            chisl+=y[i]*koef
    return chisl/znam

# интерполяция функции по узлам Чебышева
def interpolate_cheb(a, b, nodes_cnt):
    nodes_x = np.asarray([0.5 * (a + b) + 0.5 * (b - a) * cos(pi * (2 * k - 1) / (2 *
nodes_cnt))
                       for k in range(1, nodes_cnt + 1)], dtype = np.float64)
    nodes_y = np.asarray([math.cos(x)* math.cos(x) - x for x in nodes_x], dtype = np.float64)

    x = np.linspace(a, b, 1000)
    y = [bar_form(nodes_x, nodes_y, x_val) for x_val in x]

    return x, y

# интерполяция функции по равноотстоящим узлам
def interpolate(a, b, nodes_cnt):
    nodes_x = np.linspace(a, b, nodes_cnt)
    nodes_y = [math.cos(x)* math.cos(x) - x for x in nodes_x]

    x = np.linspace(a,b, 1000)
    y = [bar_form(nodes_x, nodes_y, x_val) for x_val in x]

    return x, y


# построение графика для равноотстоящих узлов
def draw_plot(x, y, n):
    fig = plt.figure(figsize=(15,10))
    ax = plt.subplot(111)
    ax.plot(func_x, func_y, 'r-',  label = 'f(x)')
    ax.plot(x, y, 'b--', label = 'polynom')
    plt.grid(True)
    plt.title('n = {}'.format(n))
    ax.legend()
    show()

# построение графика для чебышевских узлов
def draw_plot_cheb(x, y, n):
    fig = plt.figure(figsize=(15,10))
    ax = plt.subplot(111)
    ax.plot(func_x, func_y, 'r-',  label = 'f(x)')
    ax.plot(x, y, 'g--', label = 'polynom_cheb')
    plt.grid(True)
    plt.title('n = {}'.format(n))
    ax.legend()
```

```
    show()


n=10
x1 = []; y1 = []
x1, y1 = interpolate (-2.0, 2.0, n)

x2 = []; y2 = []
x2, y2 = interpolate_cheb (-2.0, 2.0,n)

draw_plot(x1,y1,n)
draw_plot_cheb (x2,y2,n)


max_n1 = 0
max_n2 = 0

for i in range(1000):
    max_n1 = max(max_n1, abs(y1[i] - func_y[i]))
    max_n2 = max(max_n2, abs(y2[i] - func_y[i]))
print(max_n1)
print(max_n2)
```

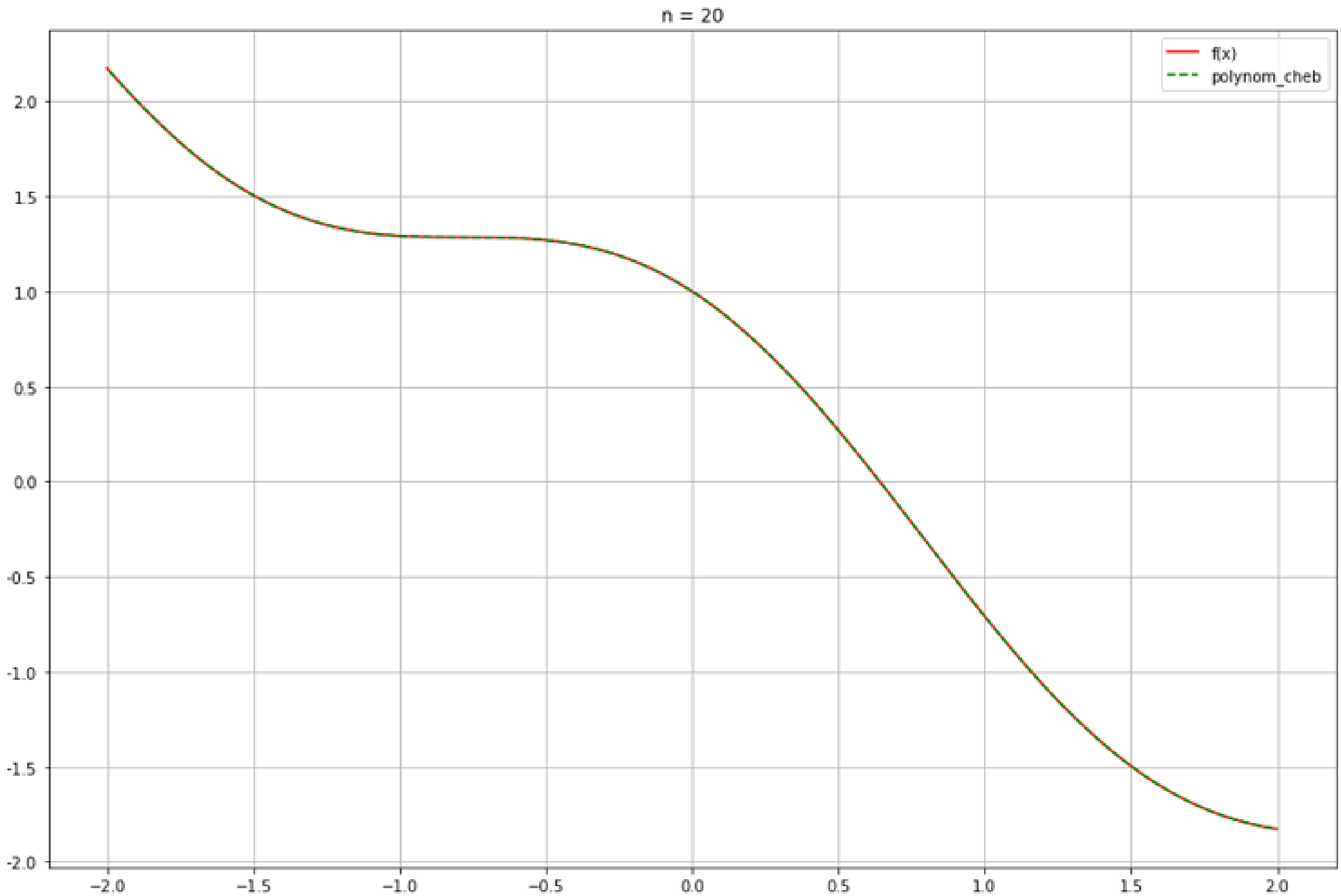
4) Графики.


n = 10
f(x)
polynom
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

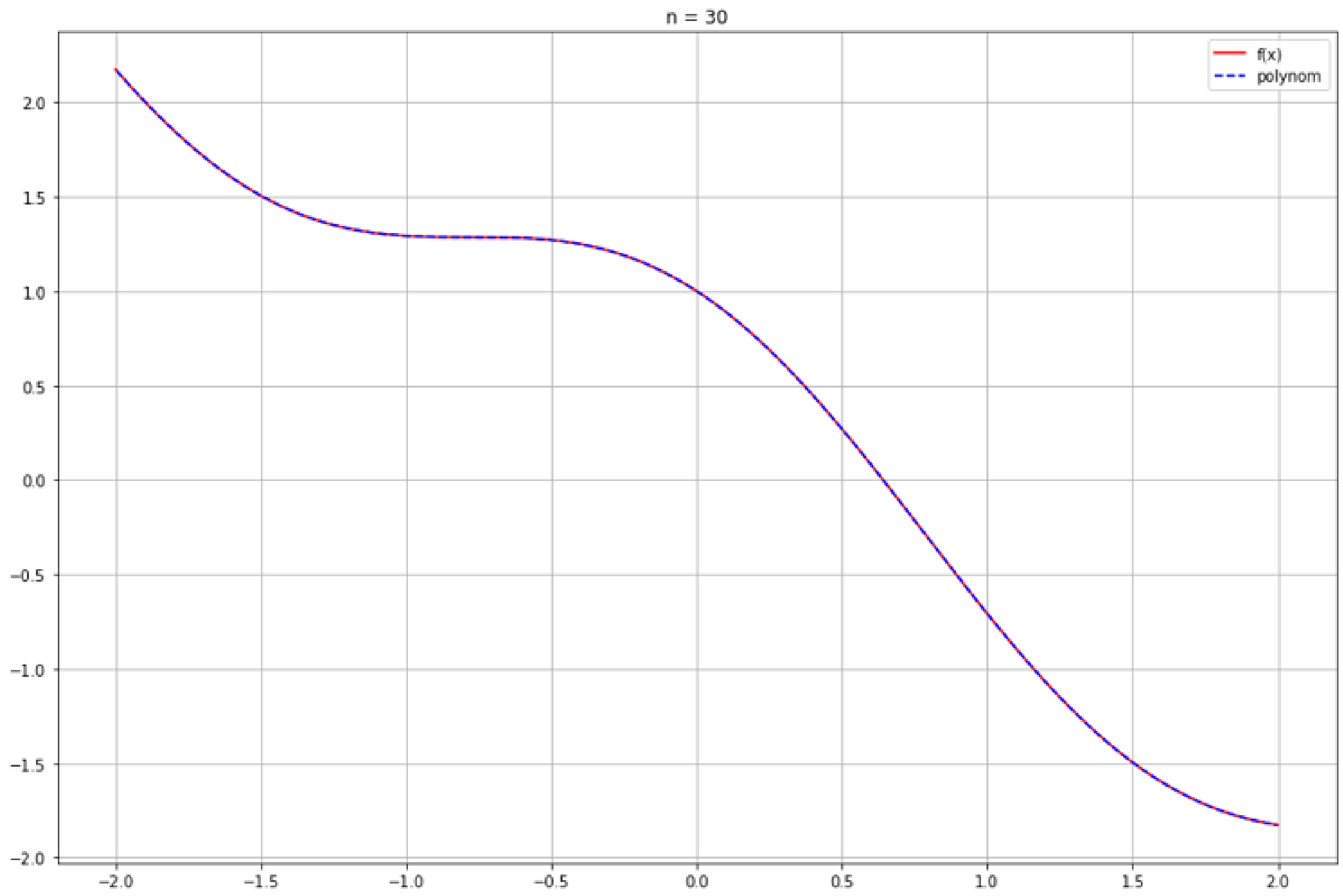

n = 10
f(x)
polynom_cheb
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

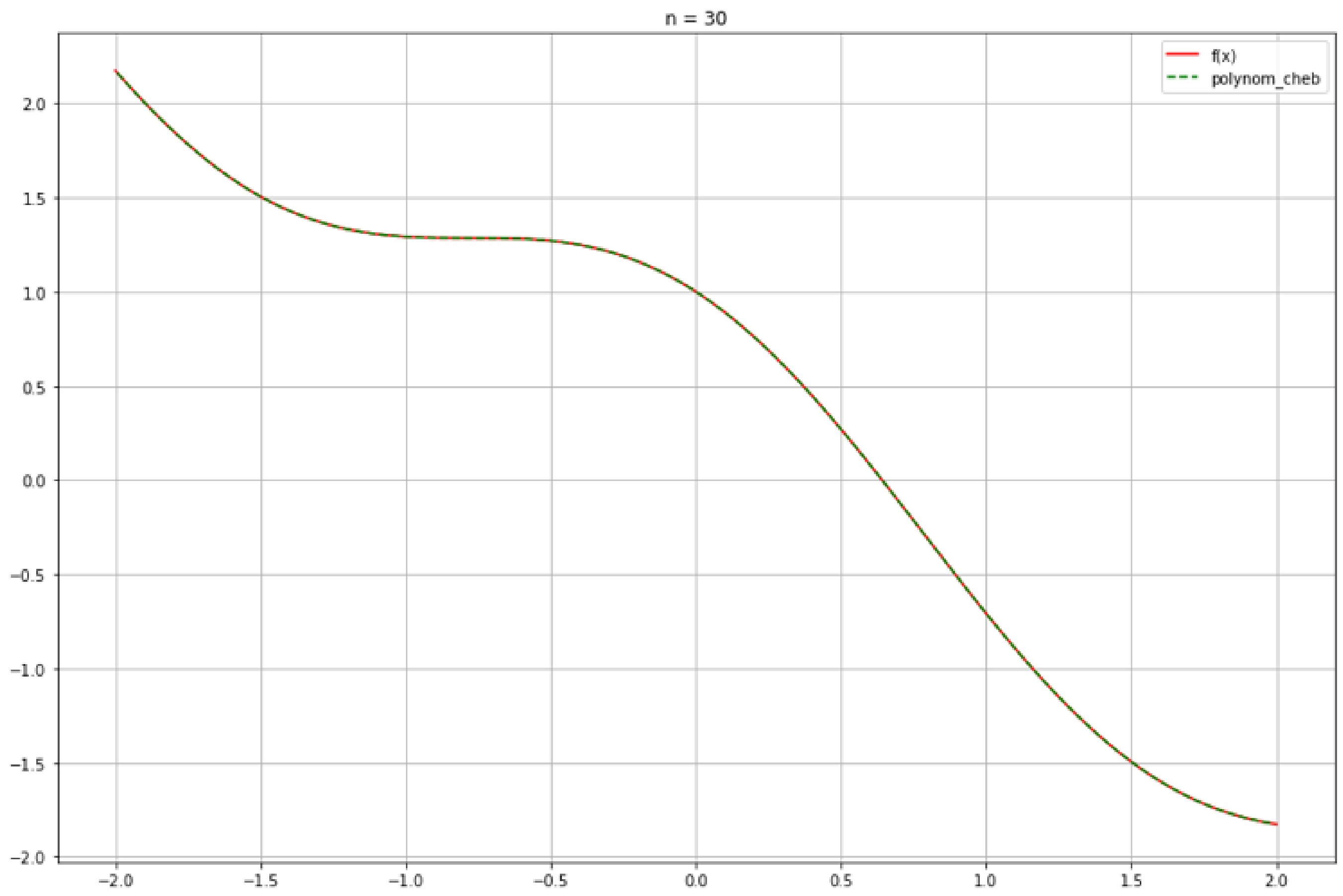


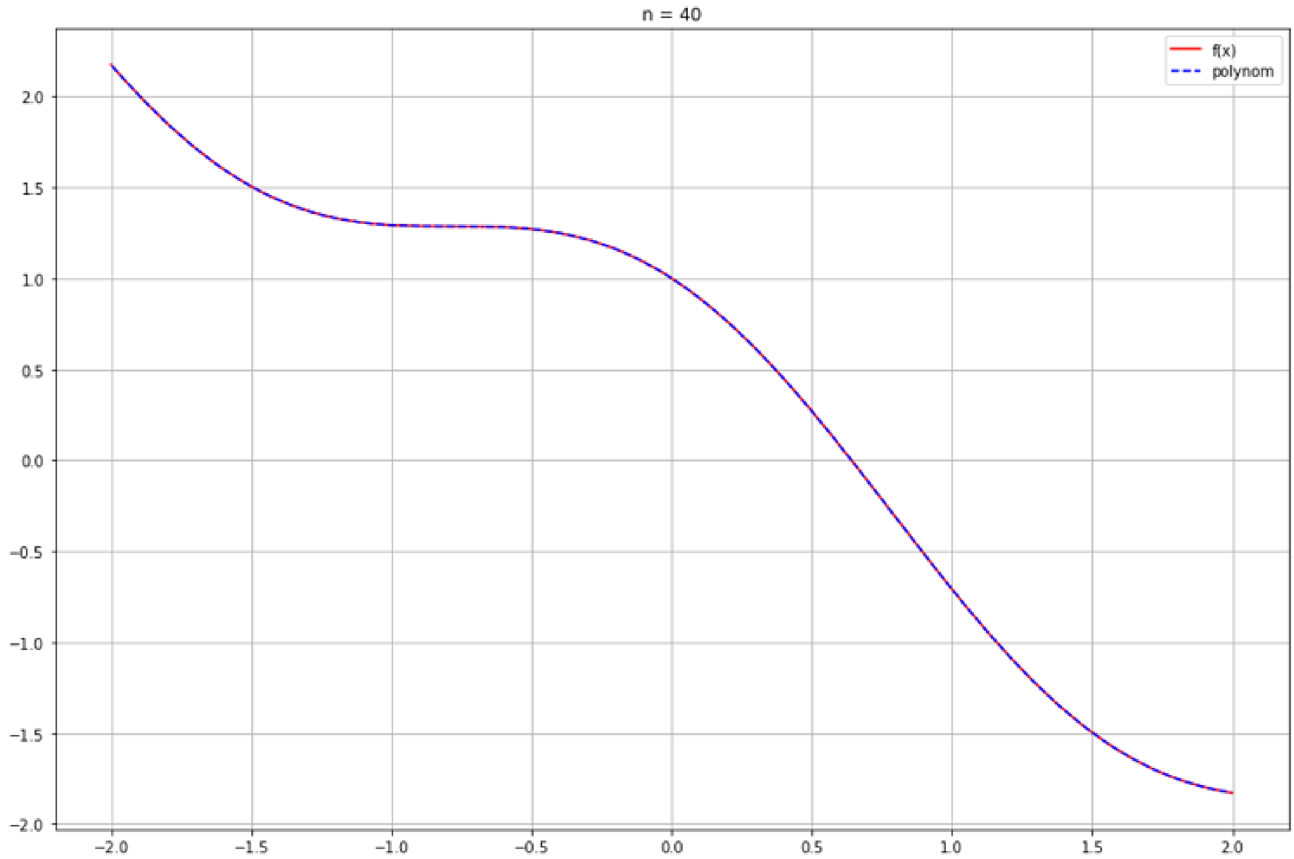
n = 20
f(x)
polynom
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

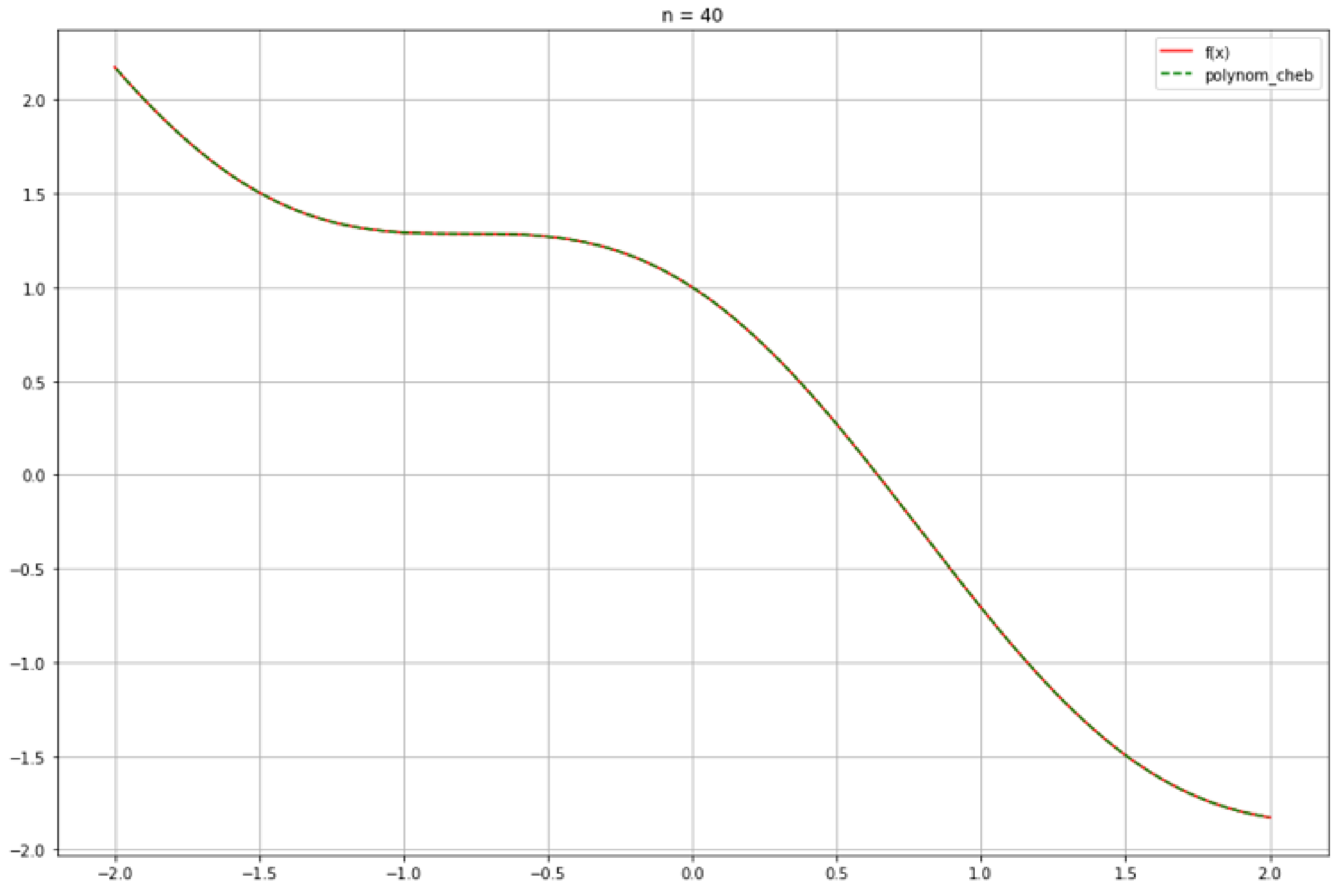


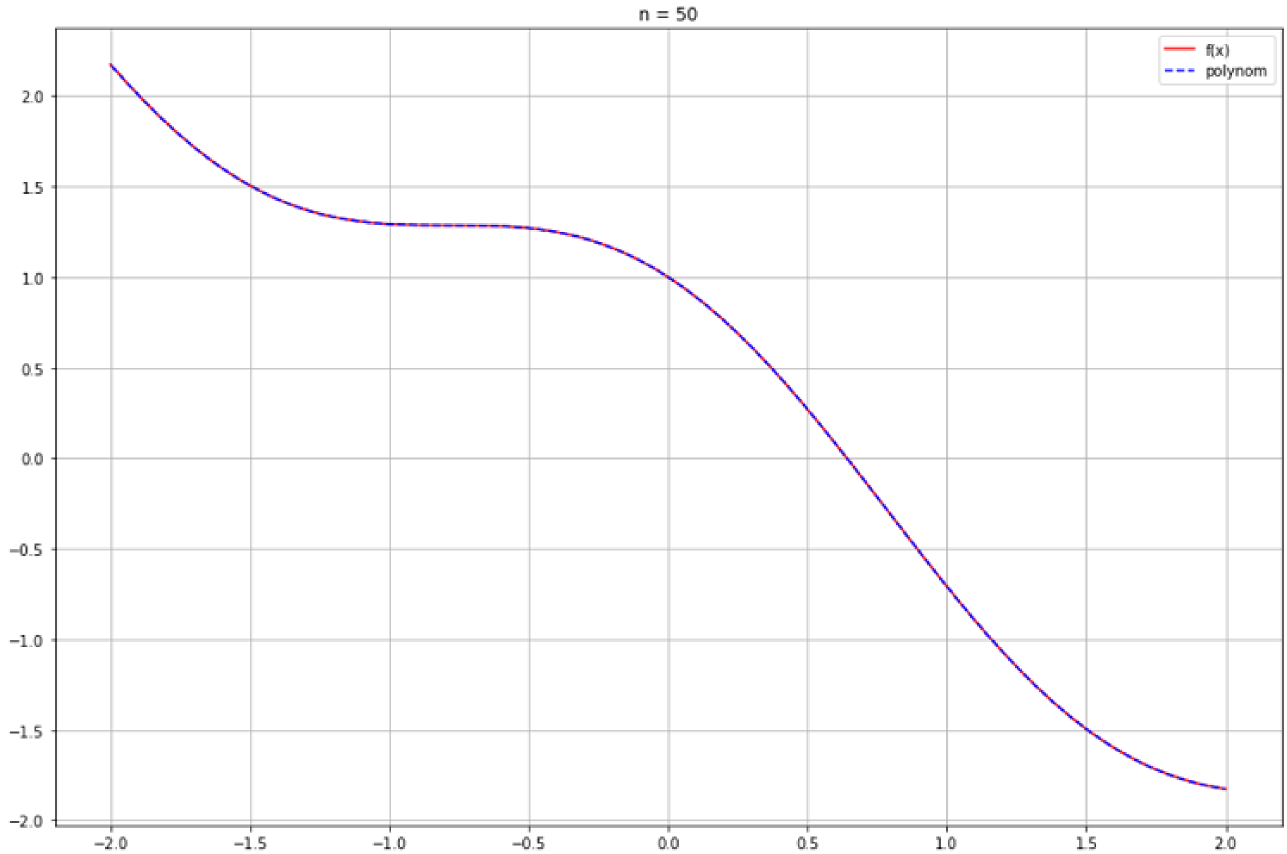
n = 20
f(x)
polynom_cheb
2.0
1.5
1.0
0.5
0.0
-0.5
-1.0
-1.5
-2.0
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

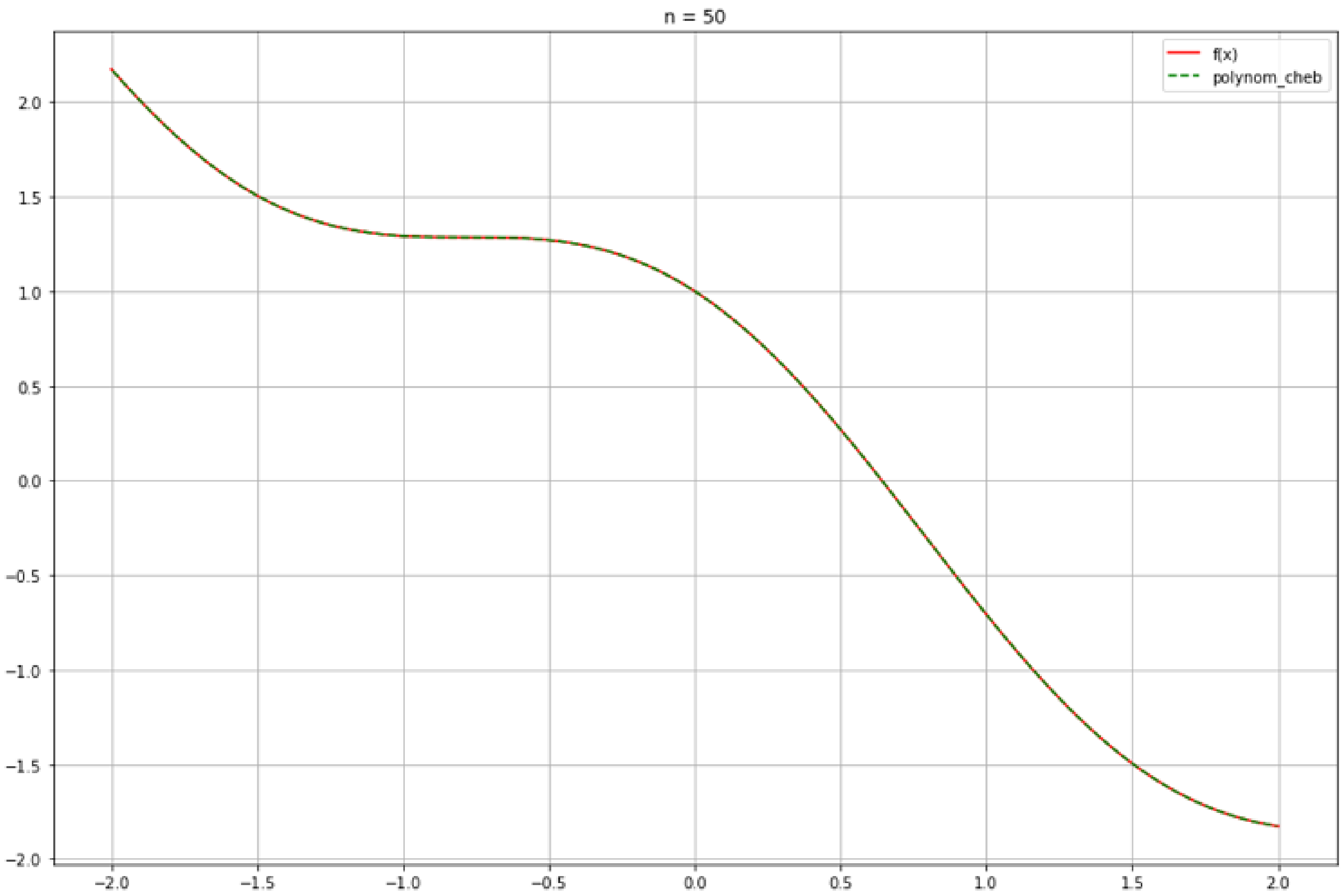

n = 30
f(x)
polynom
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

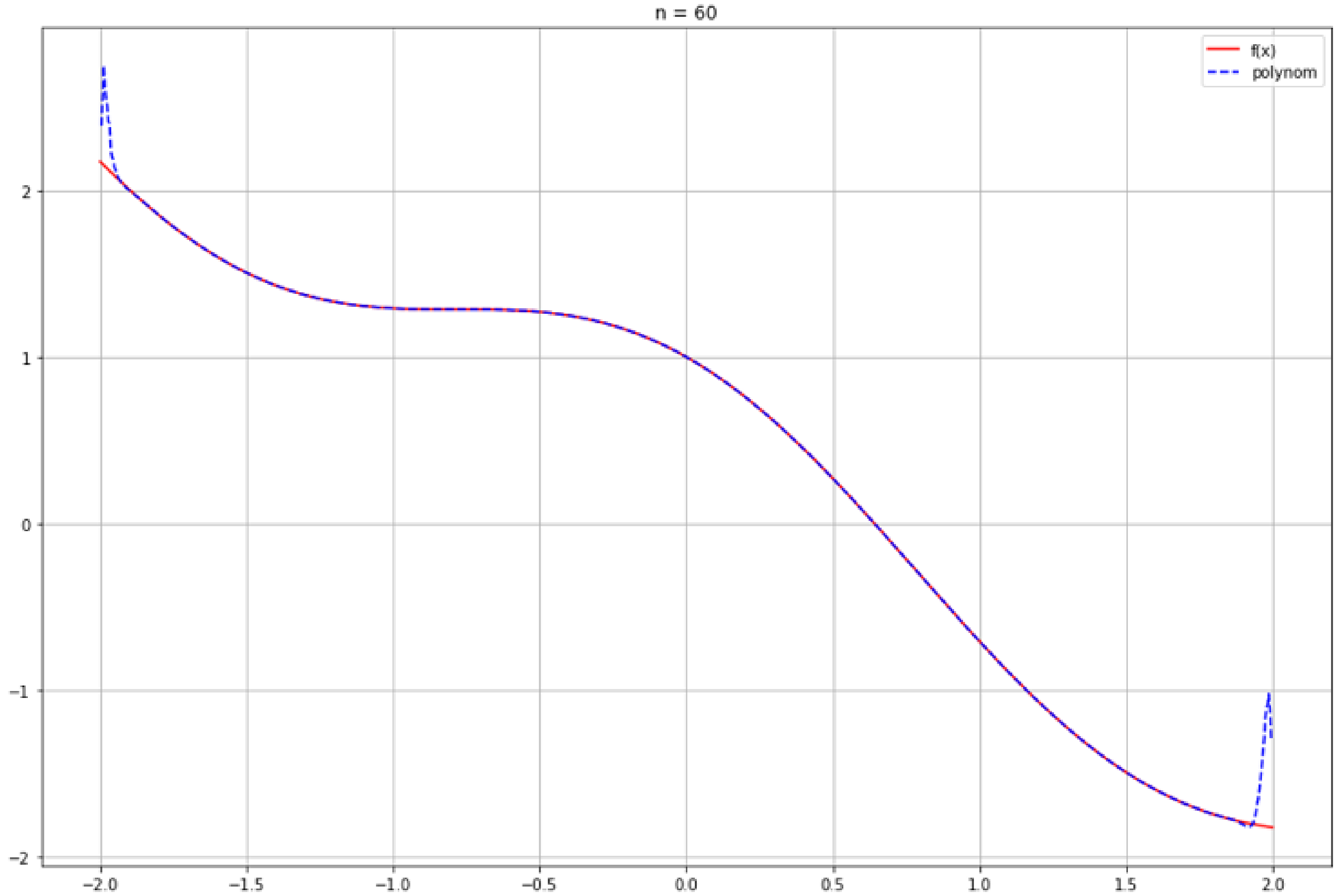

n = 30
f(x)
polynom_cheb
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

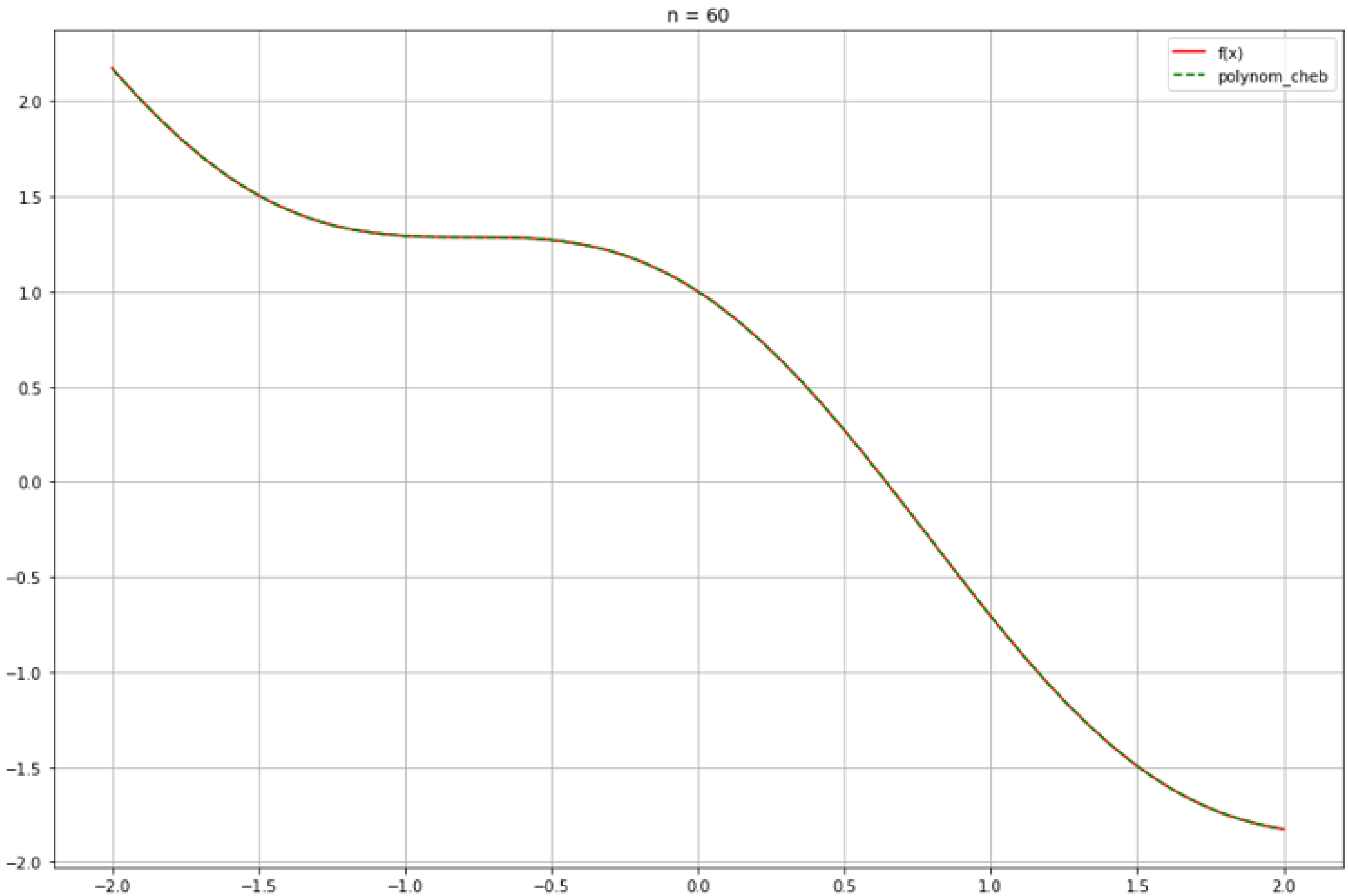

n = 40
f(x)
polynom
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

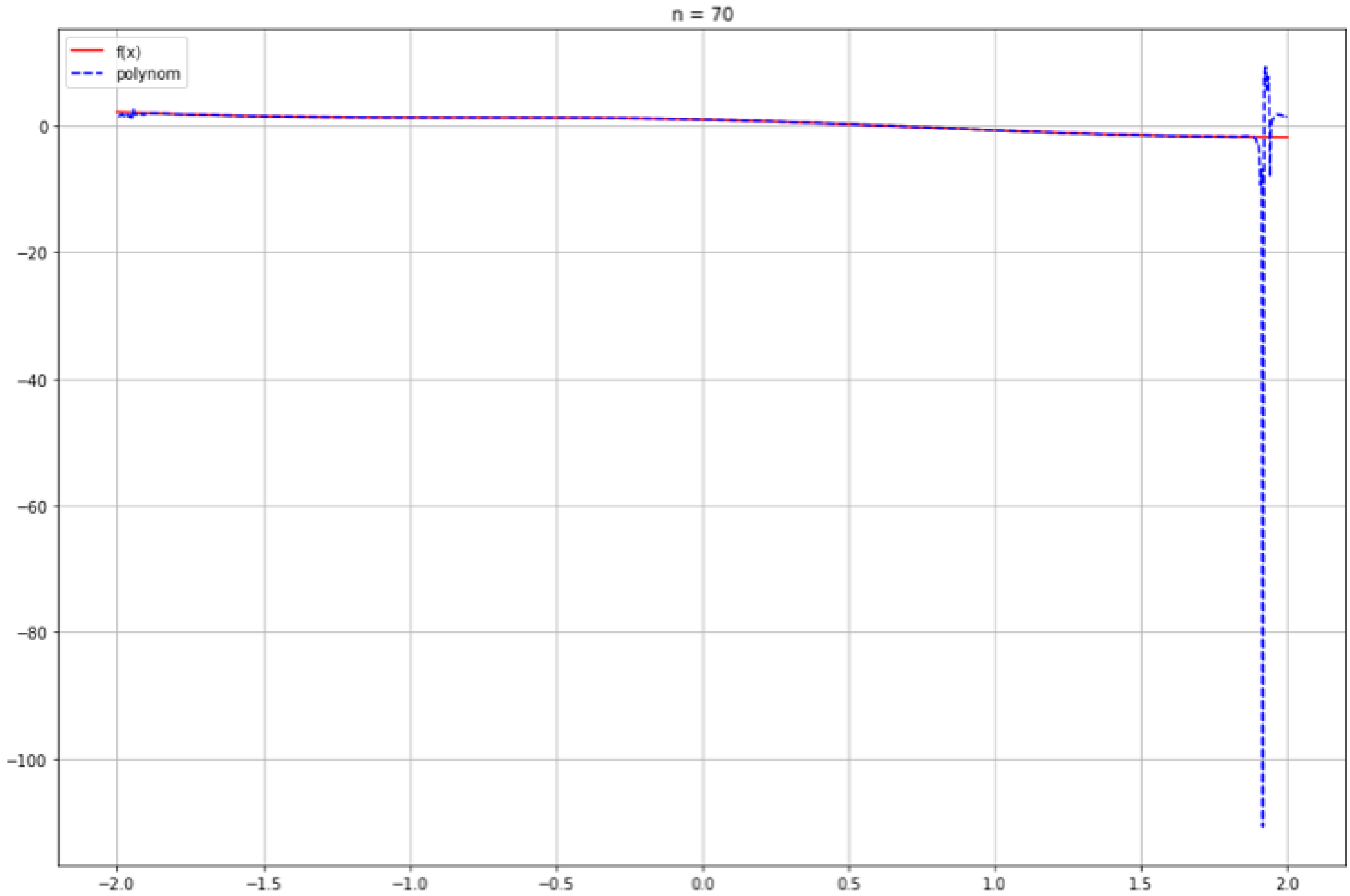

n = 40
f(x)
polynom_cheb
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0


n = 50
f(x)
polynom
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

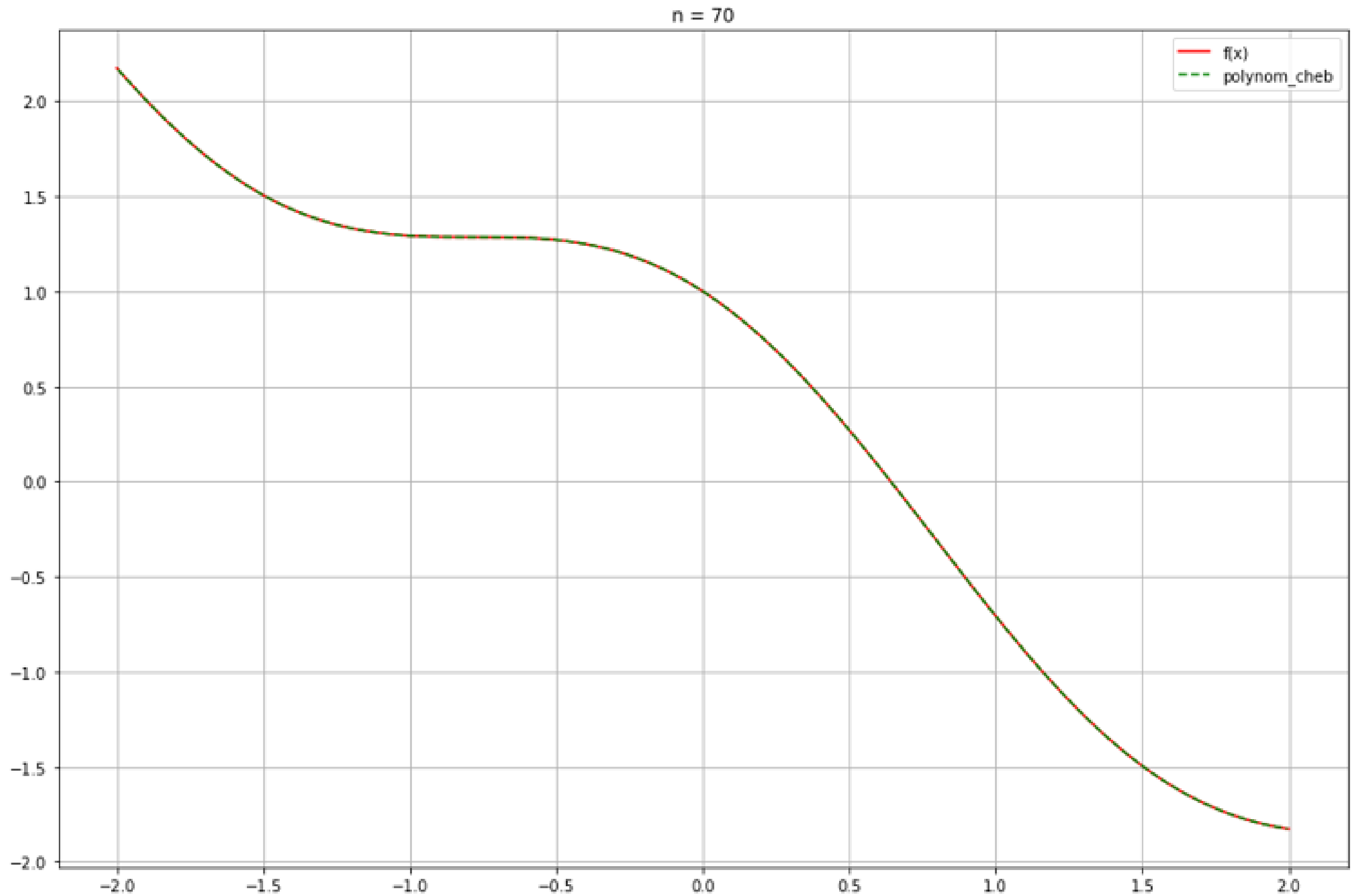

n = 50
f(x)
polynom_cheb
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0


n = 60
f(x)
polynom

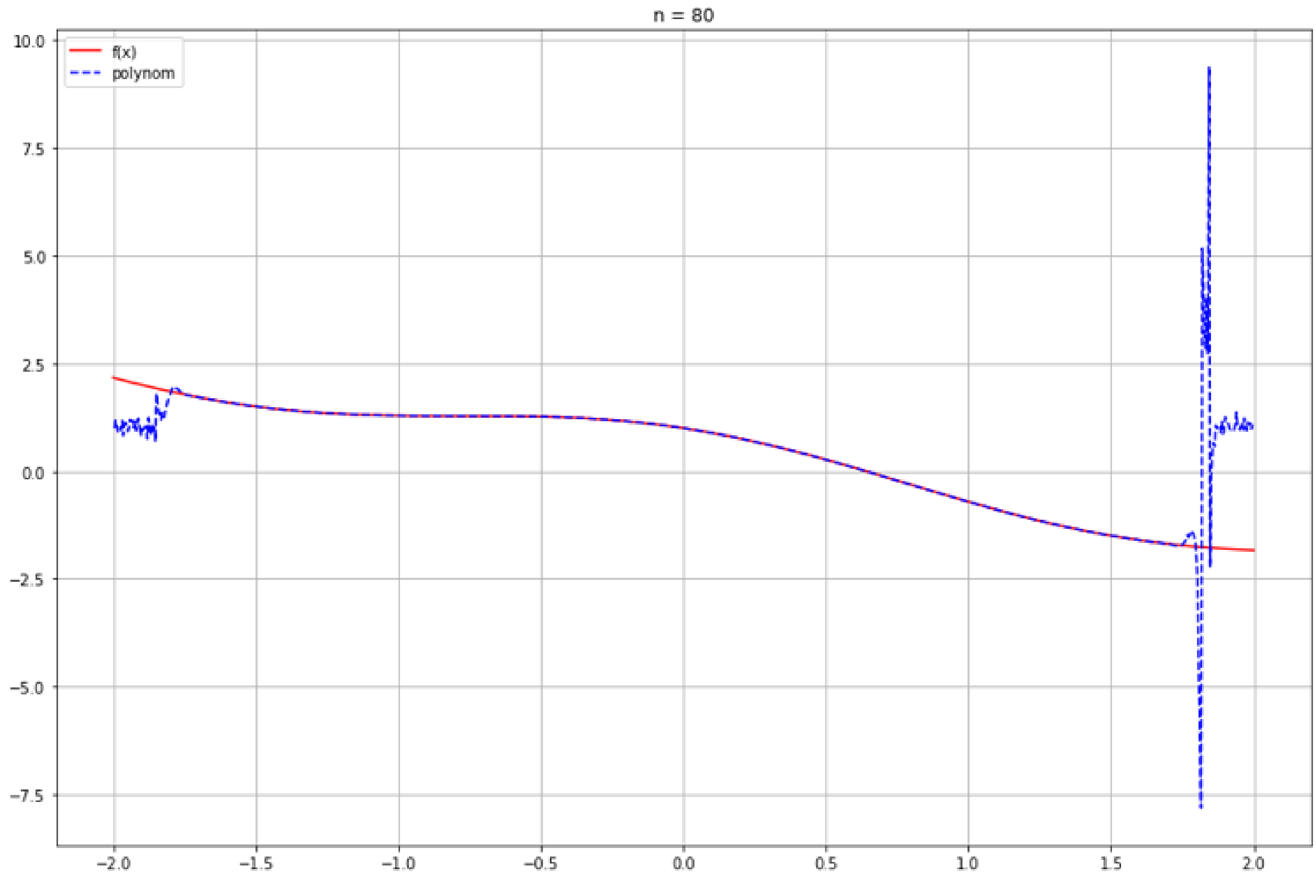

n = 60
f(x)
polynom_cheb

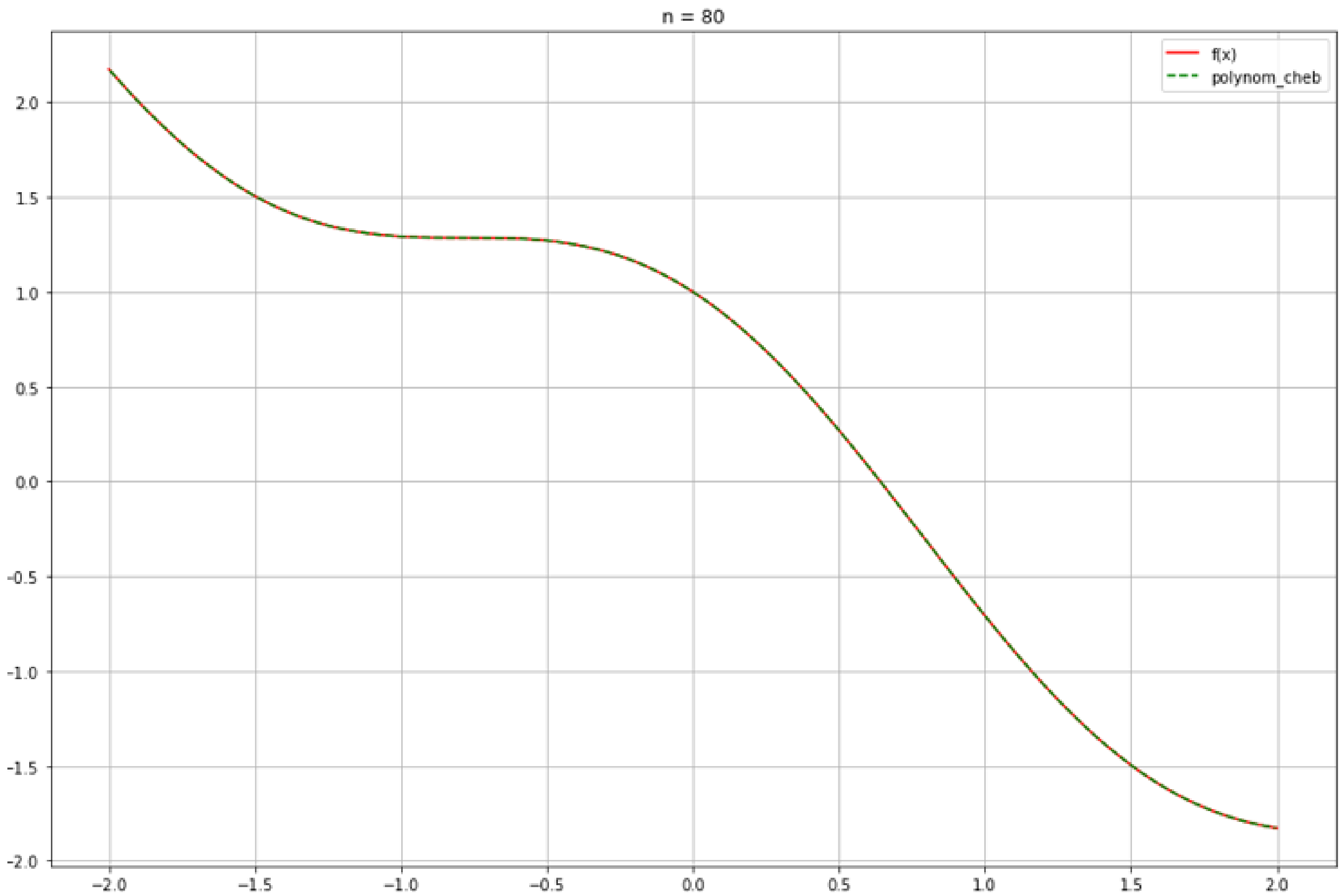


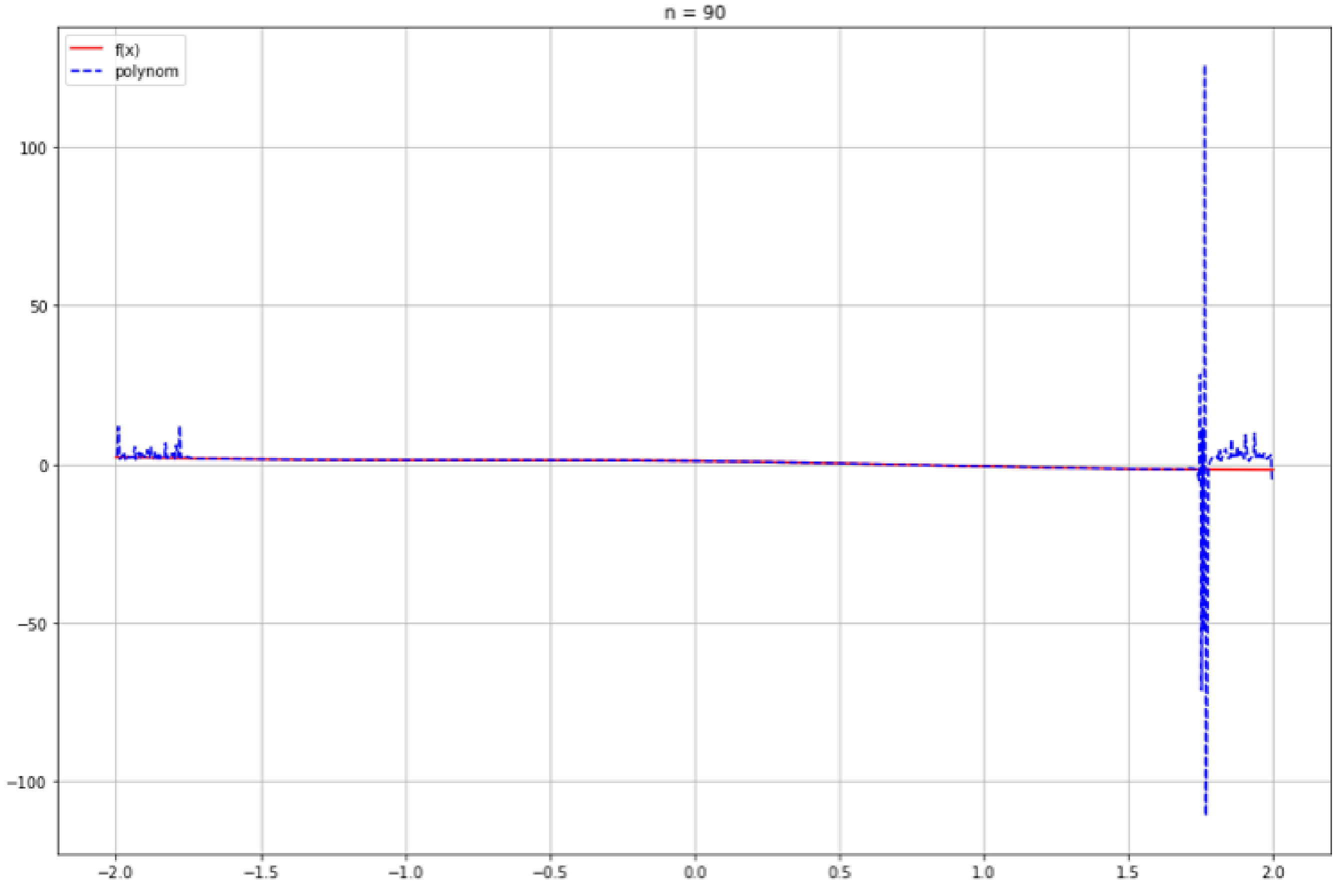
n = 70
f(x)
polynom
0
−20
−40
−60
−80
−100
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

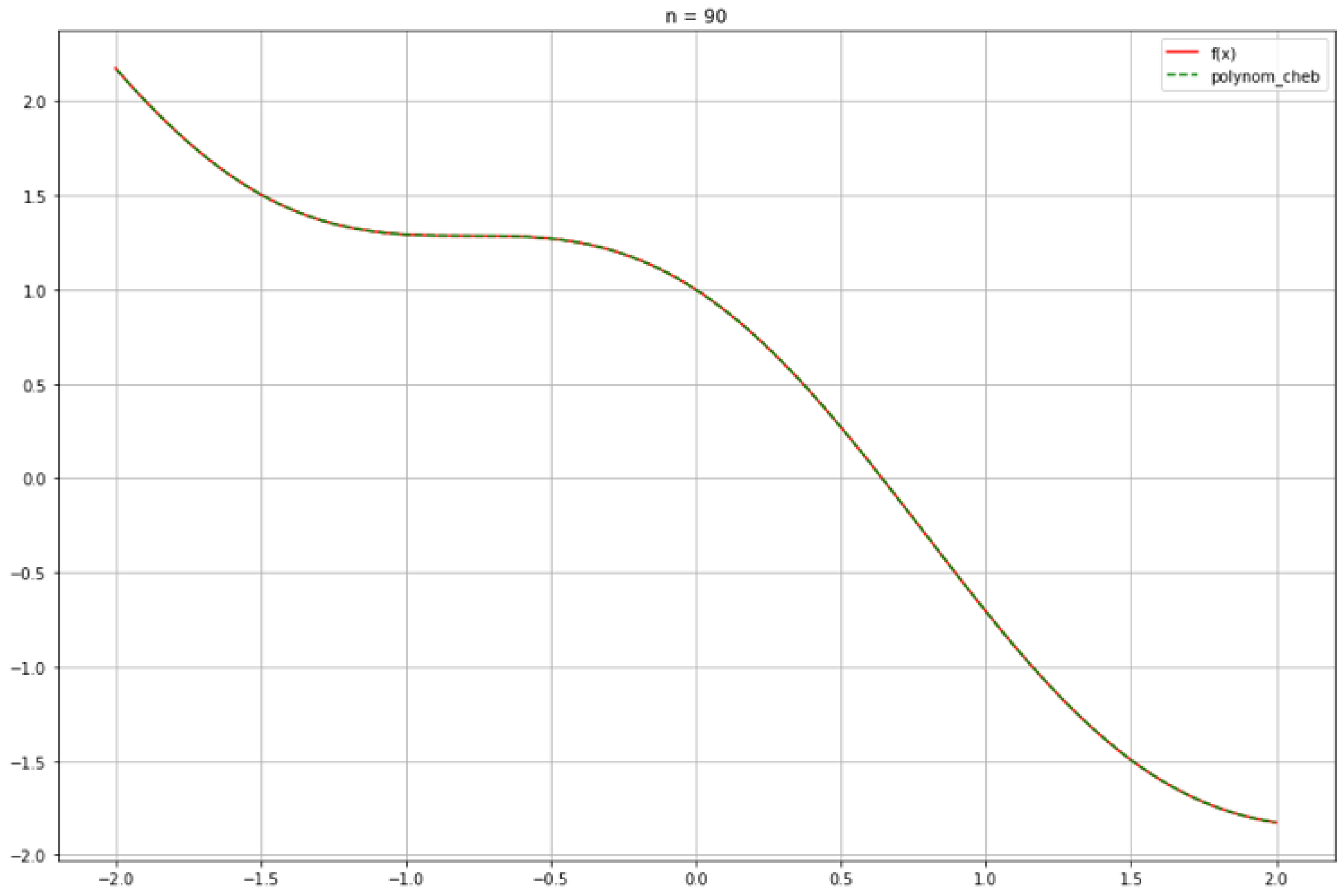


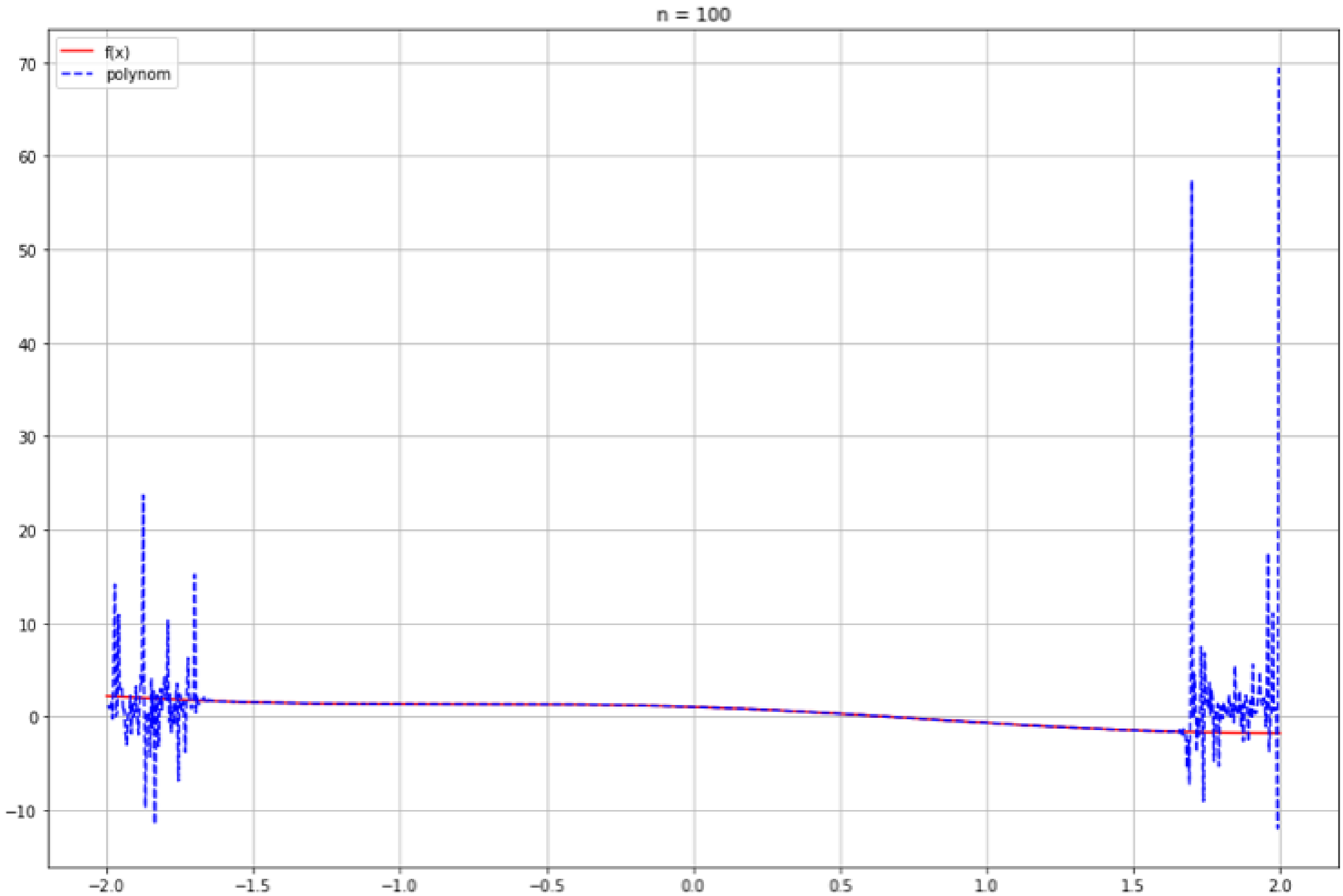
n = 70
f(x)
polynom_cheb
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

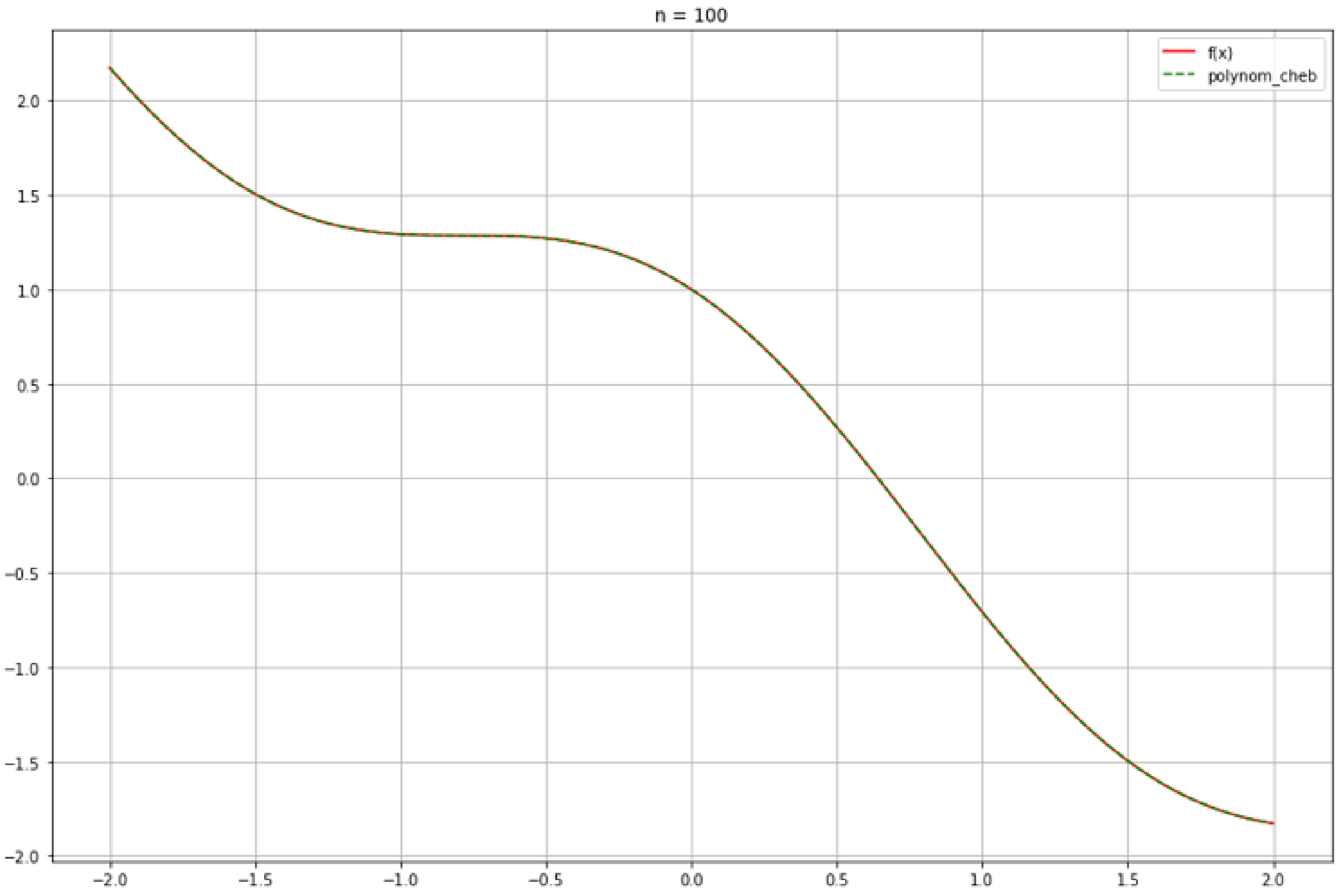

n = 80
f(x)
polynom
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0


n = 80
f(x)
polynom_cheb
2.0
1.5
1.0
0.5
0.0
-0.5
-1.0
-1.5
-2.0
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0


n = 90
f(x)
polynom
100
50
0
−50
−100
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0


n = 90
f(x)
polynom_cheb
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0


n = 100
f(x)
polynom
70
60
50
40
30
20
10
0
−10
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0


n = 100
f(x)
polynom_cheb
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

5) Таблица

N	Норма (равноотстоящие узлы)	Норма (чебышевские узлы)
10	0.0012723000737486778	0.00020785128659173147
20	6.515277206631254e-11	3.622657729351886e-13
30	5.714277939716794e-10	1.7763568394002505e-15
40	3.7820961362733385e-07	2.4424906541753444e-15
50	0.0004098737660453988	1.9984014443252818e-15
60	0.7988823946738641	1.7763568394002505e-15
70	109.00599986310604	2.220446049250313e-15
80	11.161996780941475	3.3306690738754696e-15
90	127.70328322270198	2.886579864025407e-15
100	71.3247088758354	2.886579864025407e-15

Список использованной литературы:

1. Б. В. Фалейчик «Методы вычислений» - с.126
2. Б. В. Фалейчик «Методы вычислений» - с.124