БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет прикладной математики и информатики

Отчёт по лабораторной работе №2 «Интерполяция алгебраическими многочленами»

Выполнил:
Гаргома А. О.

Преподаватель:
Горбачева Ю. Н.

Минск 2020

Содержание

1 Постановка задачи

На отрезке $[a, b]$ заданы функции $f_1(x)$ и $f_2(x)$. Построить многочлены степени $n = 3$, 5, 7, 10, 15, интерполирующие каждую из них по узлам.

1. равномерно расположены на указанном отрезке;
2. расположенным на указанном отрезке оптимальным (минимизирующим погрешность) образом.

2 Теория

Разделённые разности определяются следующим образом:

$$f(x_0,\ x_1,\ \ldots, x_{k+1} = \frac{f(x_1,\ x_2,\ \ldots,\ x_{k+1}) - f(x_0,\ x_1,\ \ldots,\ x_k)}{x_{k+1} - x_0}$$

Будем использовать интерполяционный многочлен в форме Ньютона.

$$P_n = f(x_0) + (x - x_0)f(x_0, x_1) + \cdots + (x - x_0)\ldots(x - x_{n-1})f(x_0, x_1, \ldots, x_k)$$

В качестве равномерно расположенных узлов, возьмём

$$x_m = a + \frac{b-a}{n-1}m, \quad m = 0,\ 1,\ \ldots,\ n-1$$

Для многочлена с минимальной погрешностью, в качестве узлов интерполирования возьмём корни многочлена Чебышёва после линейного преобразования $x = \frac{a+b}{2} - \frac{b-a}{2}t$

$$x_m = \frac{b+a}{2} + \frac{b-a}{2}\cos\left(\frac{\pi(2m+1)}{2n}\right), m = 0, 1, \ldots, n-1.$$

Для оценки точности, будем использовать модуль остатка интерполирования:

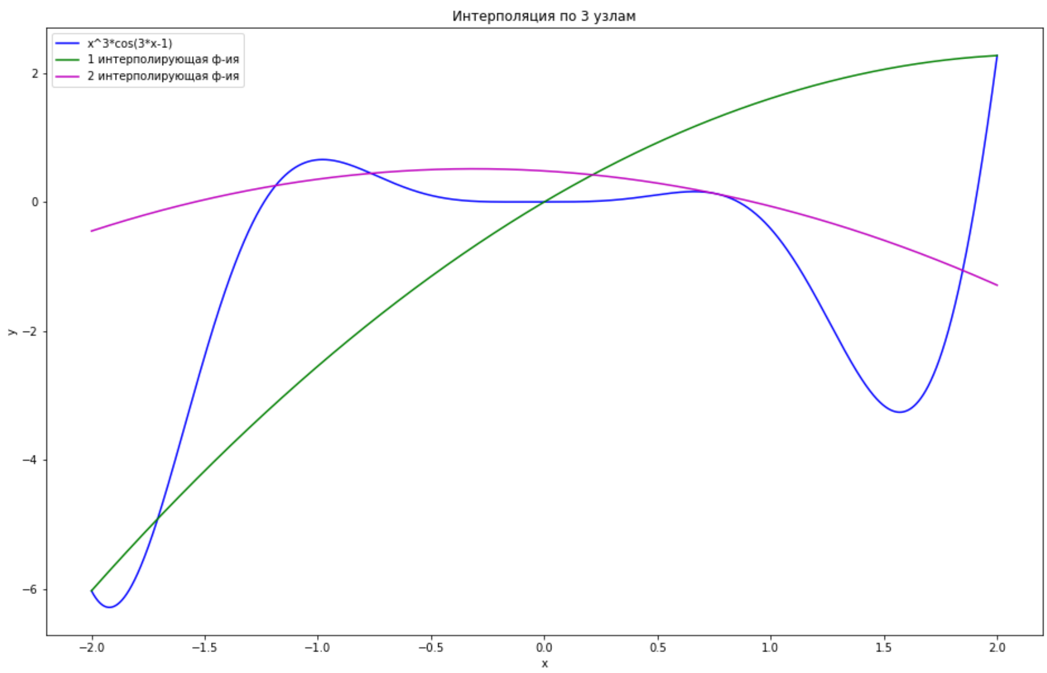
$$r_n = |f(x) - P_n(x)|$$

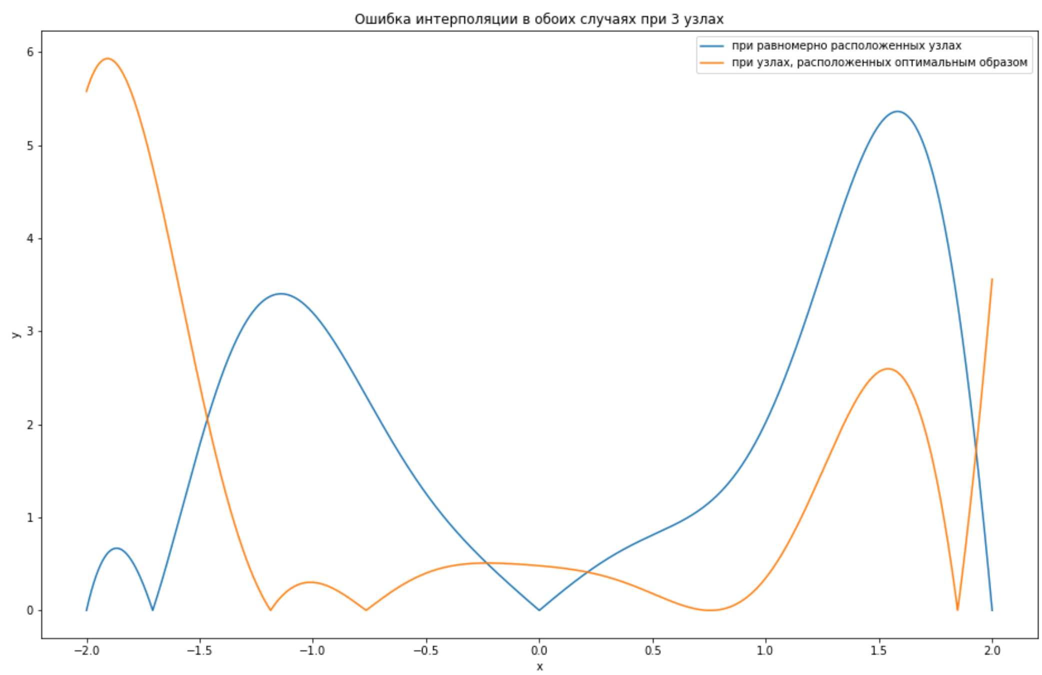
3 Программа

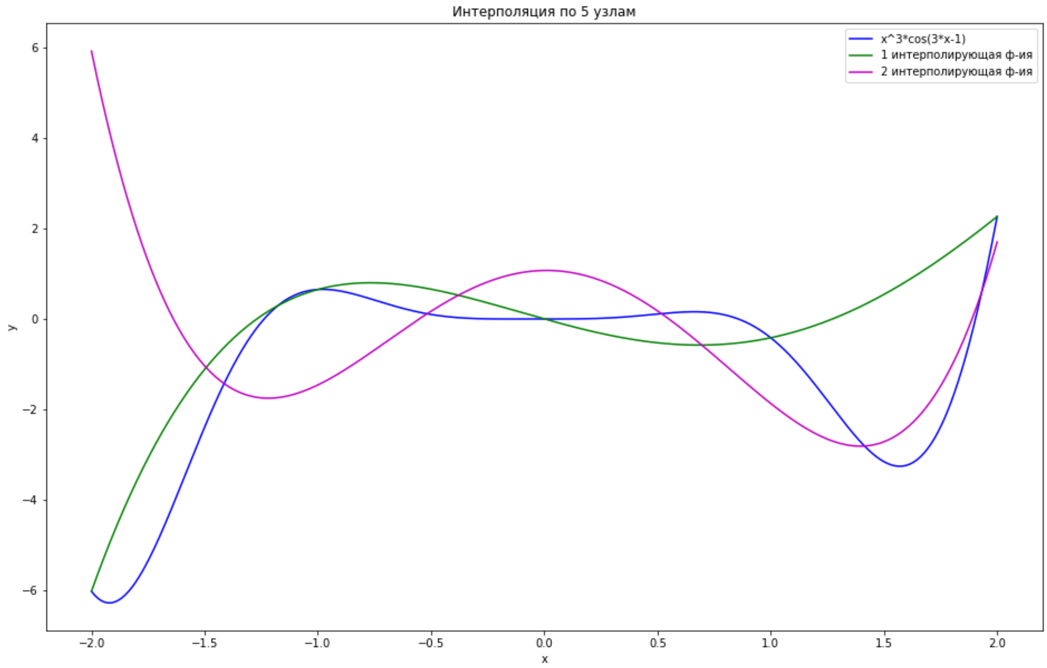
```
import numpy as np
import matplotlib.pyplot as plt
import math
a,b = -2,2
length = abs(b-a)
x = np.linspace(a,b,10000)
# функции
def f_1(x):
    return x**3*math.cos(3*x-1)
def f_2(x):
    return abs(5*math.cos(3*x)+3)
y0 = [np.array(list(map(f_1,x))),np.array(list(map(f_2,x)))]
def newton_pol(n,func,nodes):
    y = np.zeros((n,n+1))
    for j in range(n):
        y[j][0] = func(nodes[j])
    for h in range(n-2,-1,-1):
        k = n-h-1
        for i in range(0,h+1):
            y[i][k] = (y[i+1][k-1]-y[i][k-1])/(nodes[i+k]-nodes[i])
    def function(x):
        w = np.cumprod(x-nodes)
        result = y[0][0]+np.sum(np.dot(w,y[0,1:]))
        return result, func(x)-result
    return function
func_title = ['x^3*cos(3*x-1)','|5*cos(3*x)+3)|']
count = 1
for j,func in enumerate([f_1,f_2]):
    for n in [3,5,7,10,15]:
        # построение равномерно расположенных узлов
        h = length/(n-1)
        nodes = np.zeros(n)
        nodes[0] = a
        for i in range(1,n):
            nodes[i] = nodes[i-1]+h
        vnewtonpol1 = np.vectorize(newton_pol(n,func,nodes))
        y1,y2 = vnewtonpol1(x)
        # построение узлов, расположенных оптимальным образом
        nodes = np.zeros(n)
        for i in range(n):
            nodes[i] = (a+b)/2 + (b-a)/2*math.cos(math.pi*(2*i+1)/(2*n+2))
        vnewtonpol2 = np.vectorize(newton_pol(n,func,nodes))
        y3,y4 = vnewtonpol2(x)
        # построение графиков
        plt.figure(figsize=(10,6))
        plt.plot(x,y0[j],'b',label=func_title[j])
        plt.plot(x,y1,'g',label='1 интерполирующая ф-ия')
        plt.plot(x,y3,'m',label='2 интерполирующая ф-ия')
        plt.title(f'Интерполяция по {n} узлам')
        plt.xlabel('x')
        plt.ylabel('y')
        plt.legend()
        plt.savefig(f'images/{count}')
        count+=1
        plt.show()
        plt.figure(figsize=(10,6))
        plt.plot(x,y2,label='при равномерно расположенных узлах')
        plt.plot(x,y4,label='при узлах, расположенных оптимальным образом')
        plt.title(f'Ошибка интерполяции в обоих случаях при {n} узлах')
        plt.xlabel('x')
```

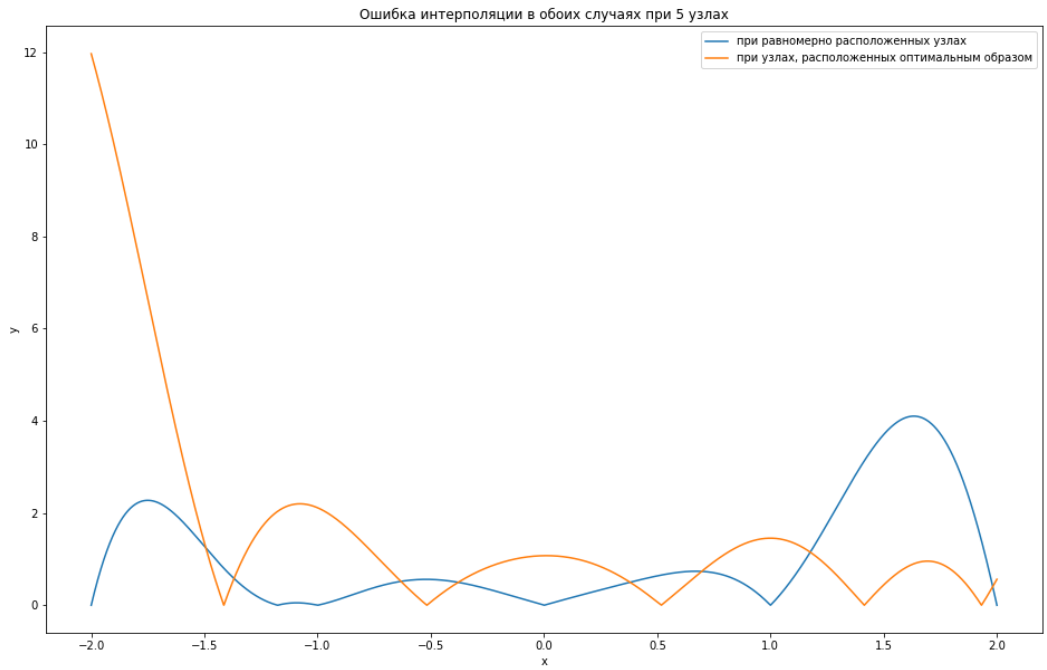
```
        plt.ylabel('y')
        plt.legend()
        plt.savefig(f'images/{count}')
        count+=1
        plt.show()
```

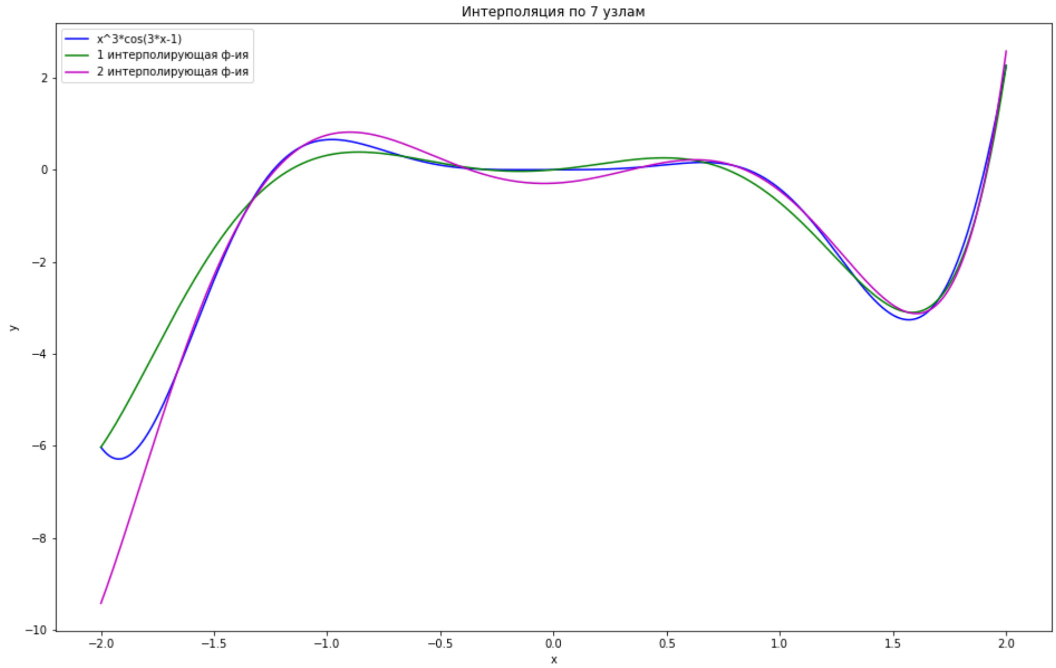
4 Результаты

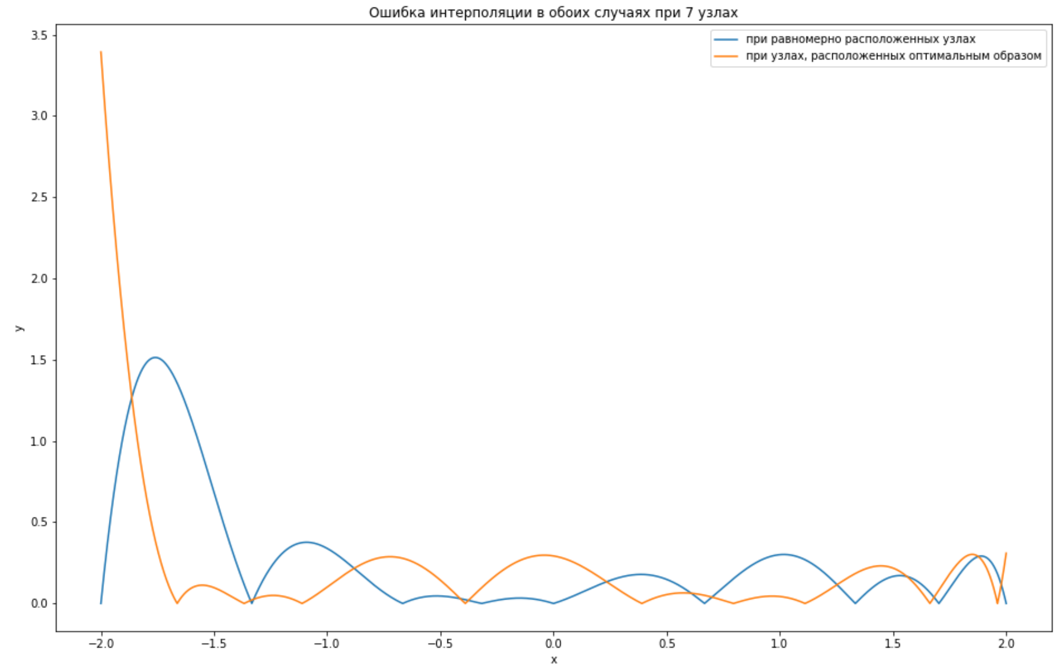


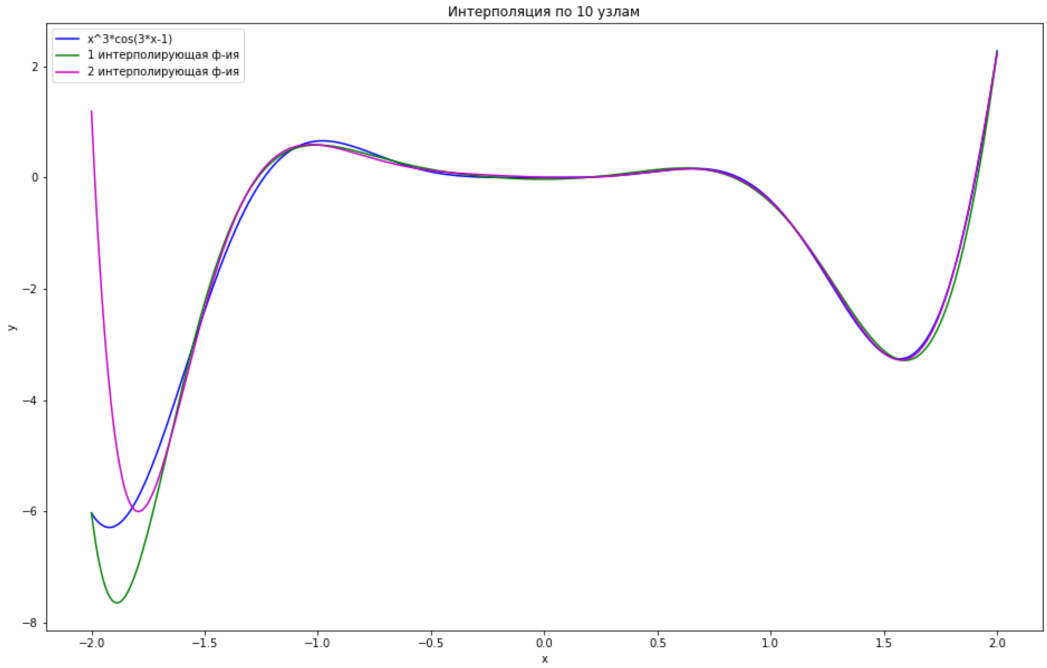


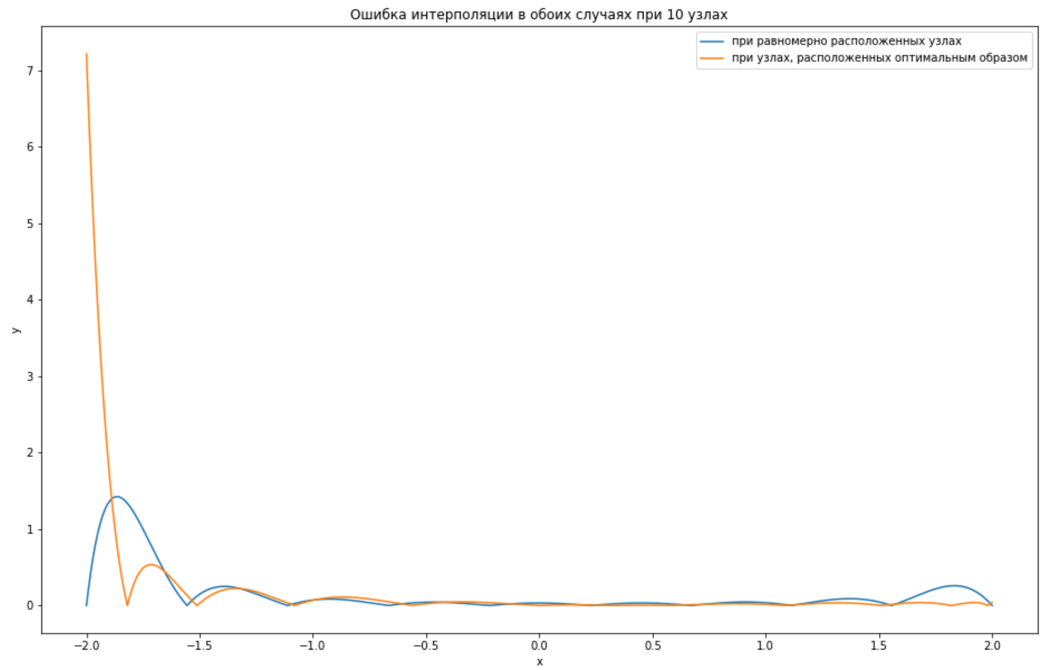

Интерполяция по 5 узлам
x^3*cos(3*x-1)
1 интерполирующая ф-ия
2 интерполирующая ф-ия
y
x
6
4
2
0
−2
−4
−6
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

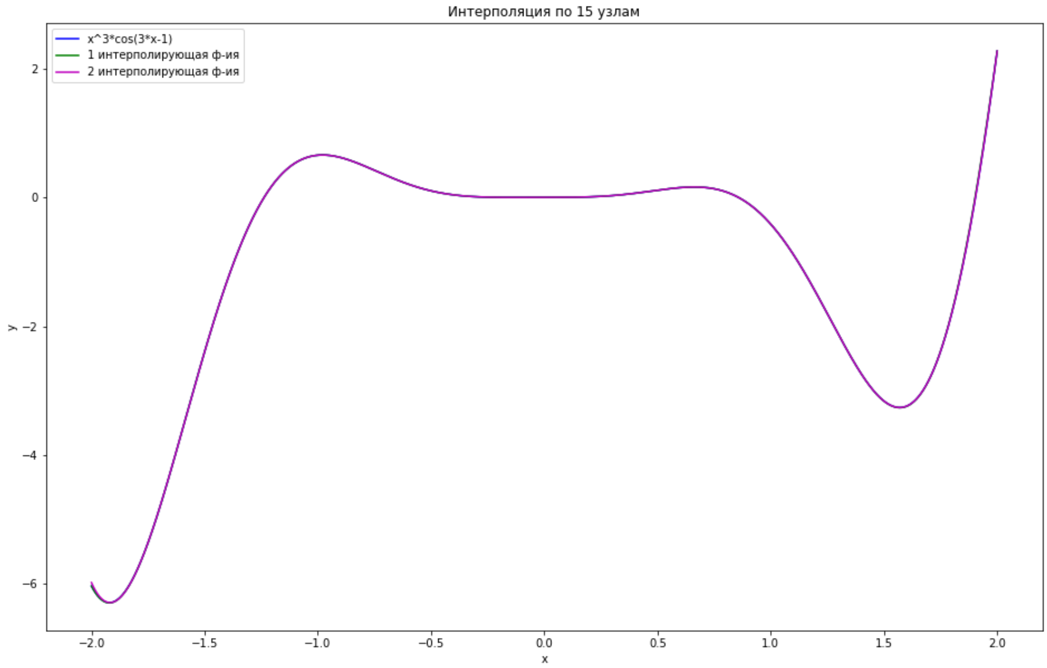

Ошибка интерполяции в обоих случаях при 5 узлах
при равномерно расположенных узлах
при узлах, расположенных оптимальным образом
y
x
12
10
8
6
4
2
0
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

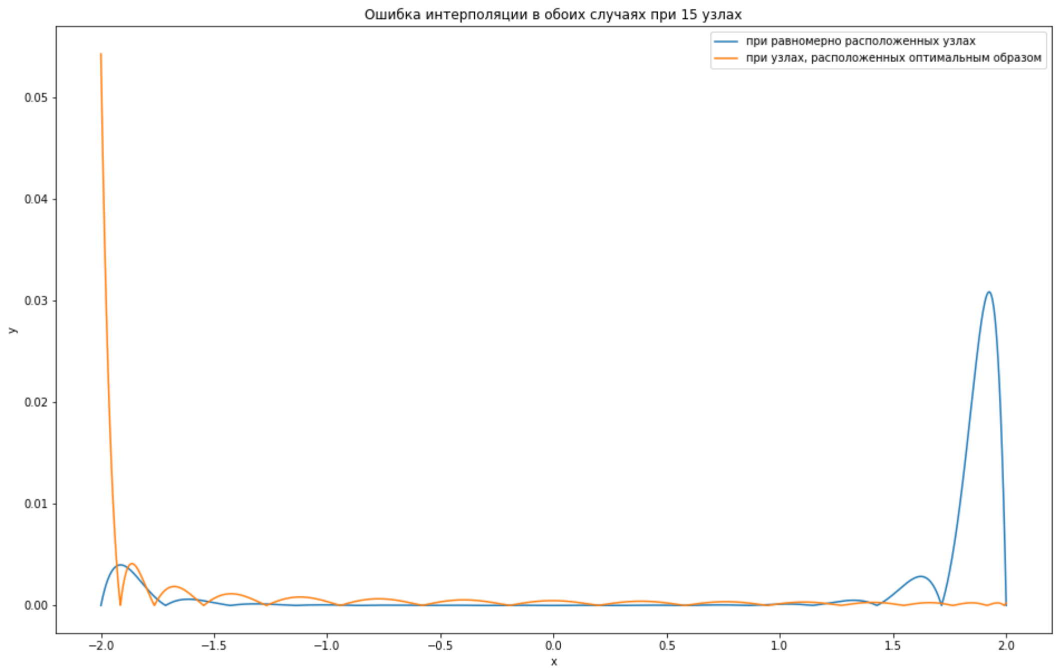

Интерполяция по 7 узлам
x^3*cos(3*x-1)
1 интерполирующая ф-ия
2 интерполирующая ф-ия
y
x

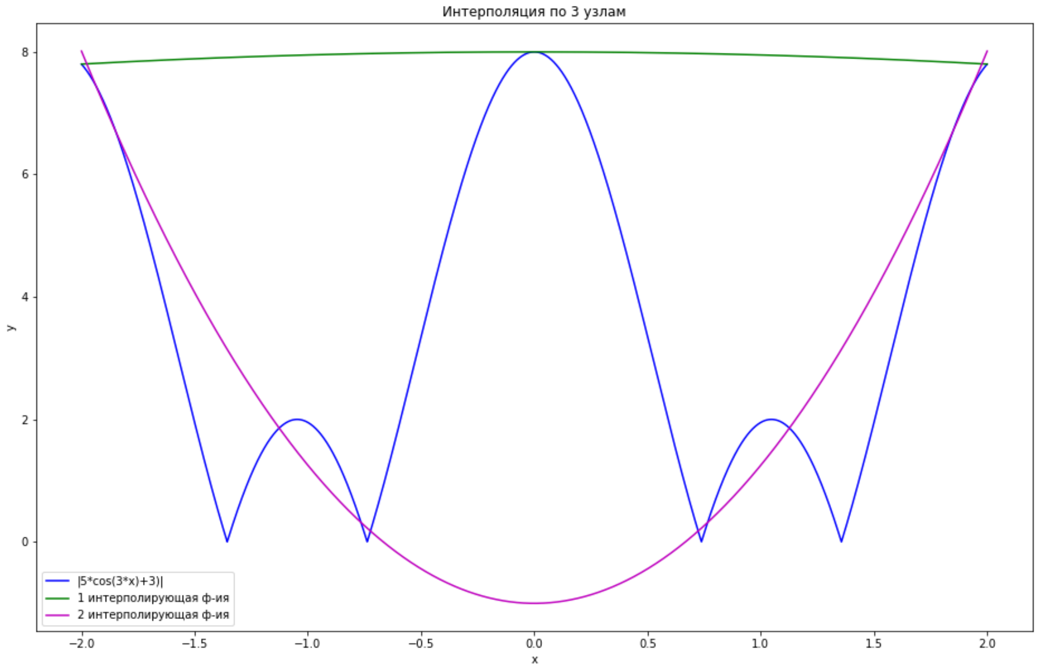

Ошибка интерполяции в обоих случаях при 7 узлах
при равномерно расположенных узлах
при узлах, расположенных оптимальным образом
y
x

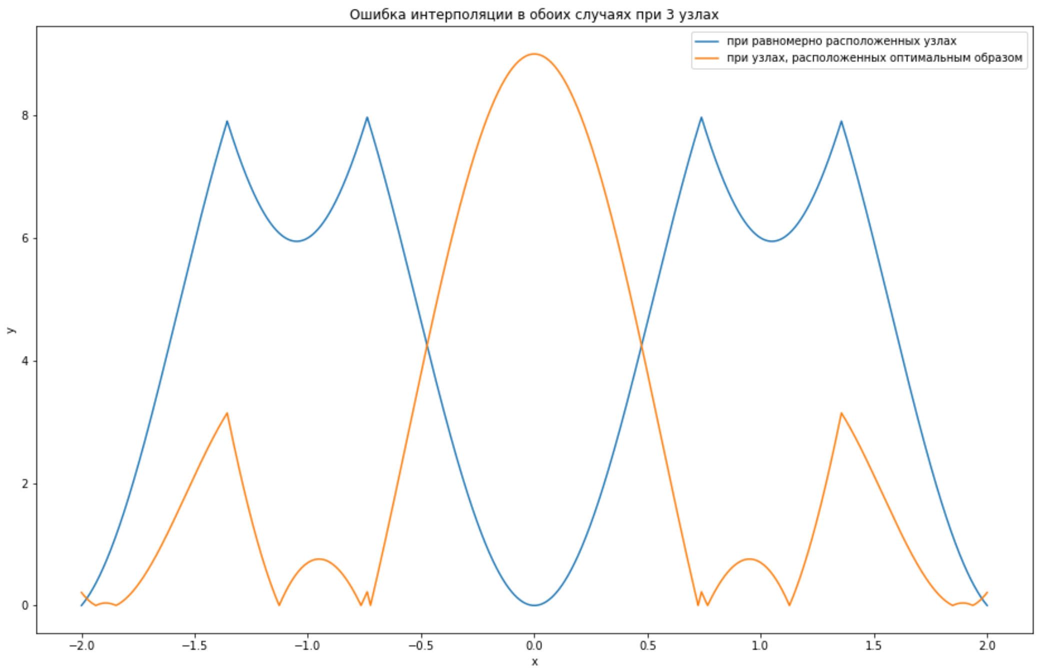

Интерполяция по 10 узлам
x^3*cos(3*x-1)
1 интерполирующая ф-ия
2 интерполирующая ф-ия
x
y

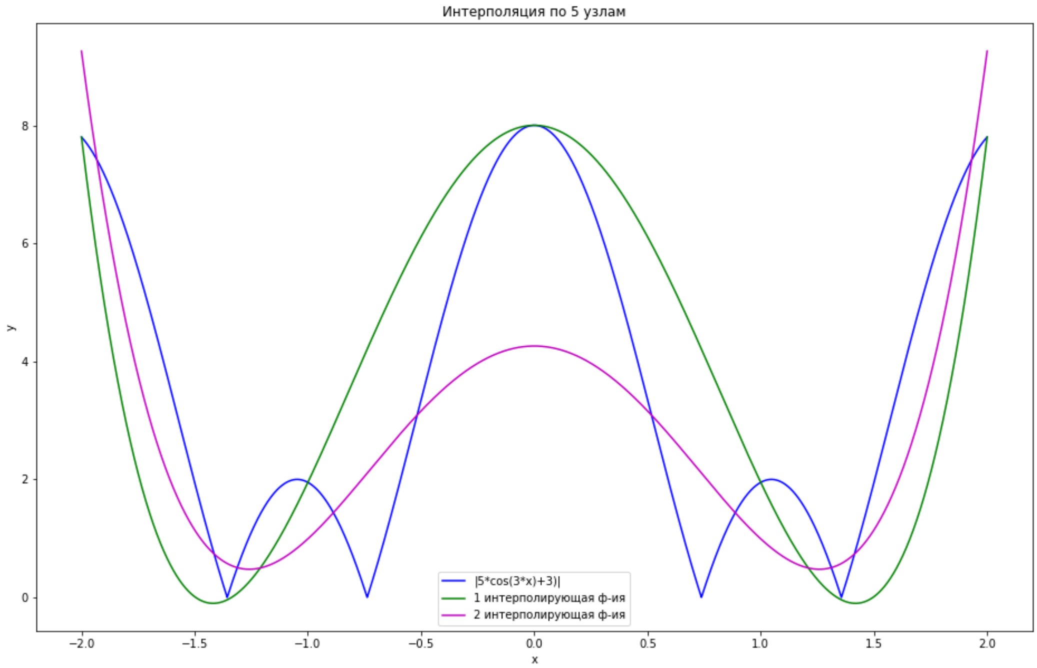

Ошибка интерполяции в обоих случаях при 10 узлах
при равномерно расположенных узлах
при узлах, расположенных оптимальным образом
x
y

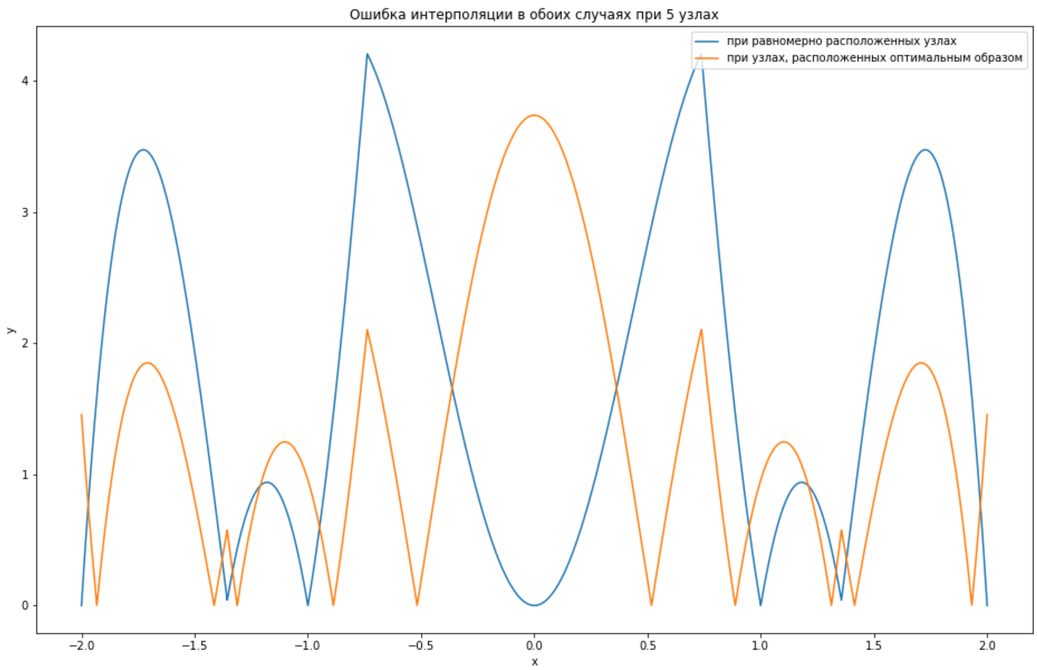

Интерполяция по 15 узлам
x^3*cos(3*x-1)
1 интерполирующая ф-ия
2 интерполирующая ф-ия
y
x

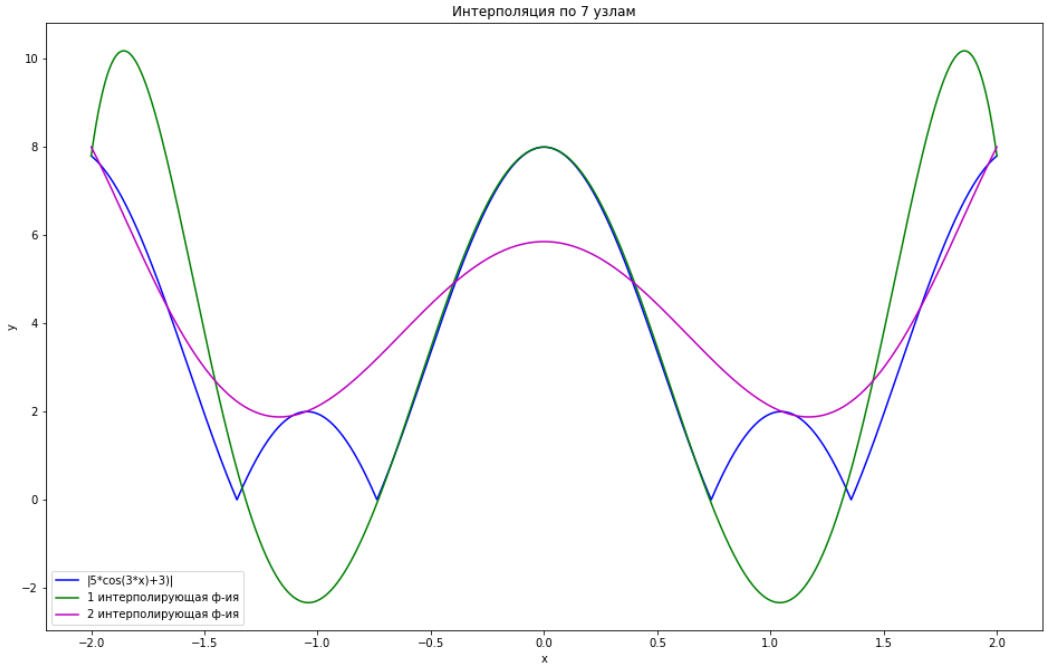

Ошибка интерполяции в обоих случаях при 15 узлах
при равномерно расположенных узлах
при узлах, расположенных оптимальным образом
y
x

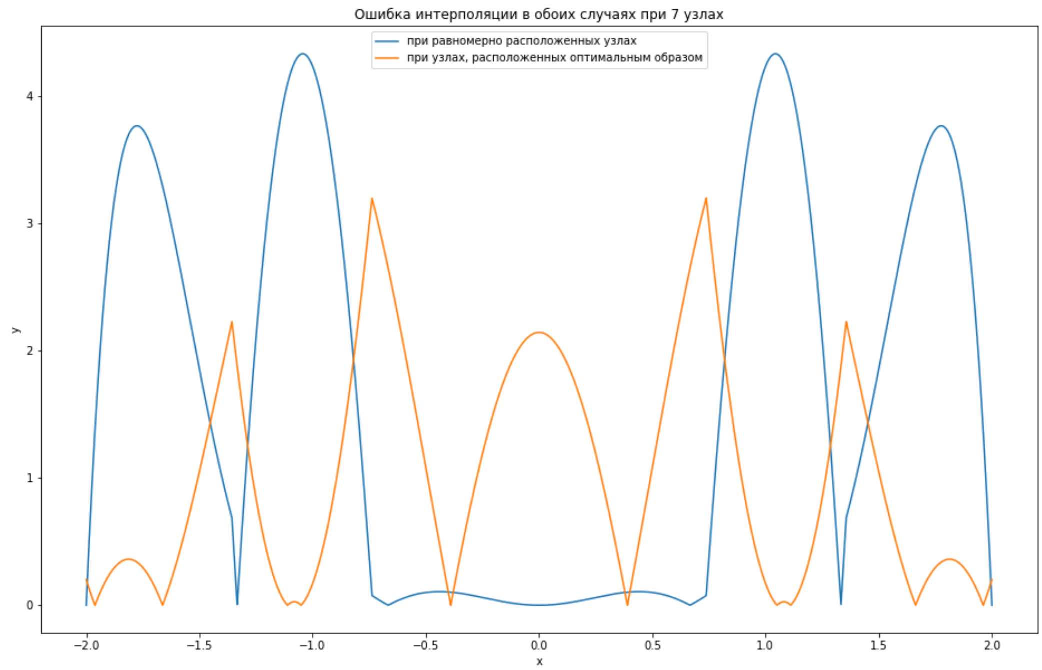

Интерполяция по 3 узлам
|5*cos(3*x)+3)|
1 интерполирующая ф-ия
2 интерполирующая ф-ия
x
y

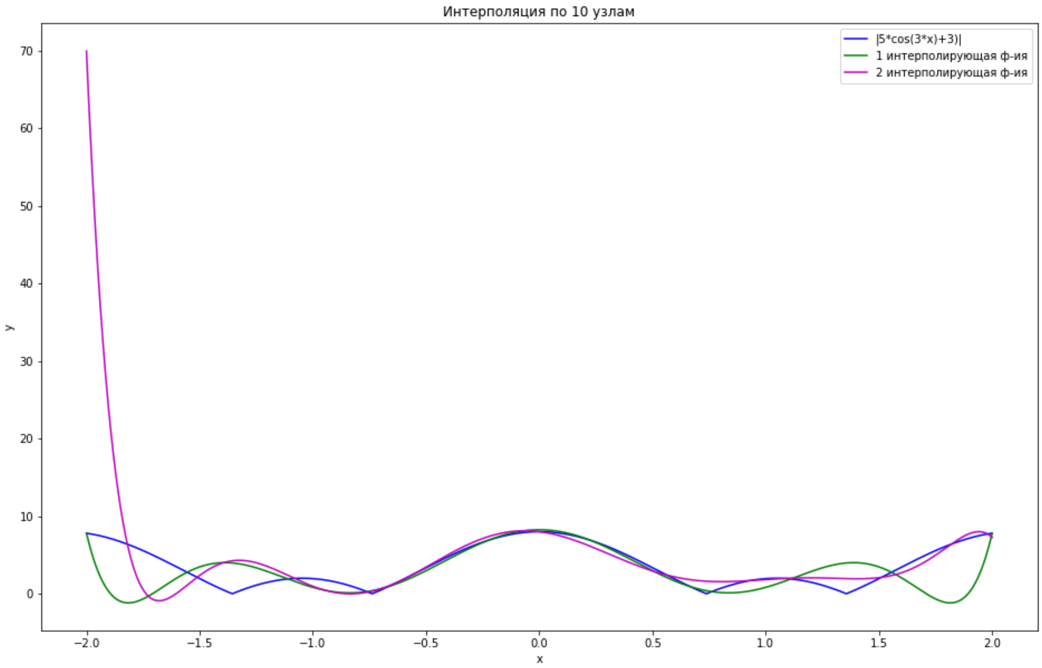

Ошибка интерполяции в обоих случаях при 3 узлах
при равномерно расположенных узлах
при узлах, расположенных оптимальным образом
x
y

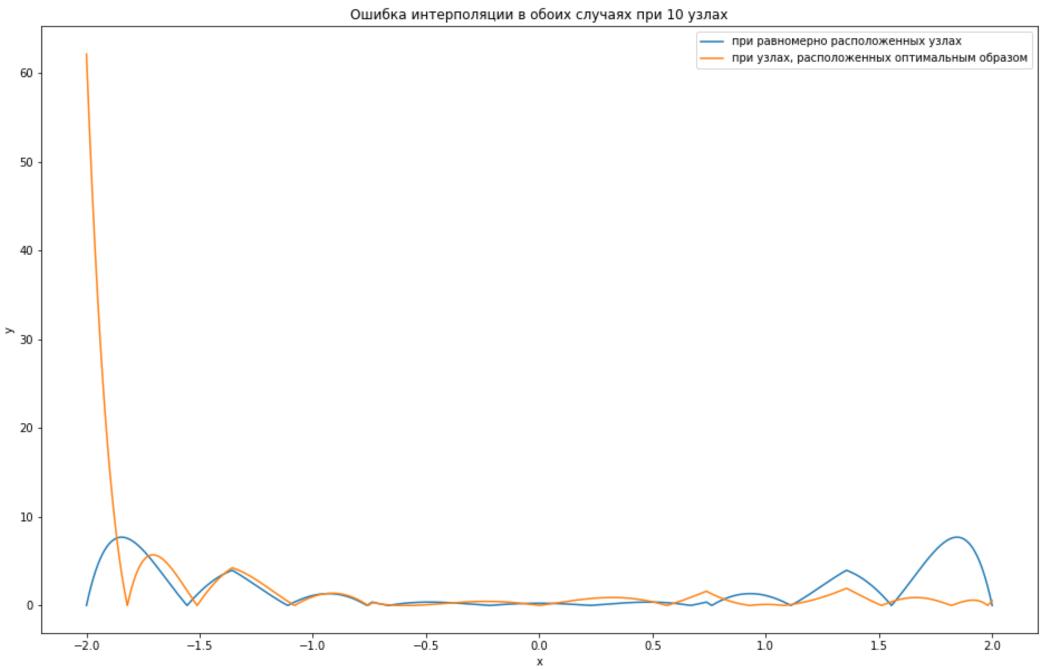

Интерполяция по 5 узлам
|5*cos(3*x)+3)|
1 интерполирующая ф-ия
2 интерполирующая ф-ия
x
y

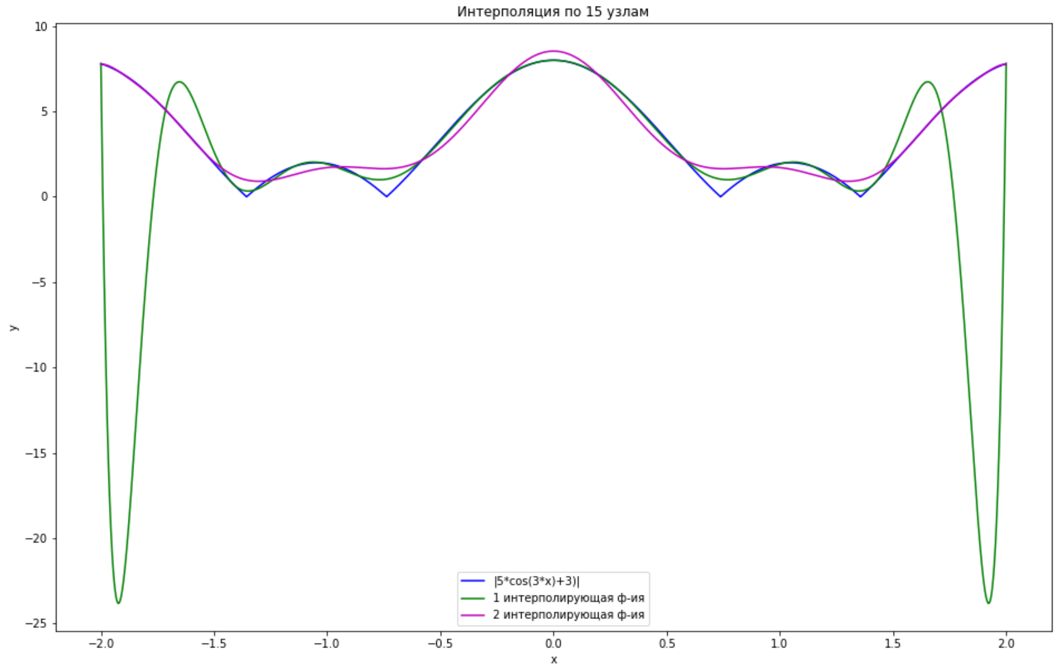

Ошибка интерполяции в обоих случаях при 5 узлах
при равномерно расположенных узлах
при узлах, расположенных оптимальным образом
x
y

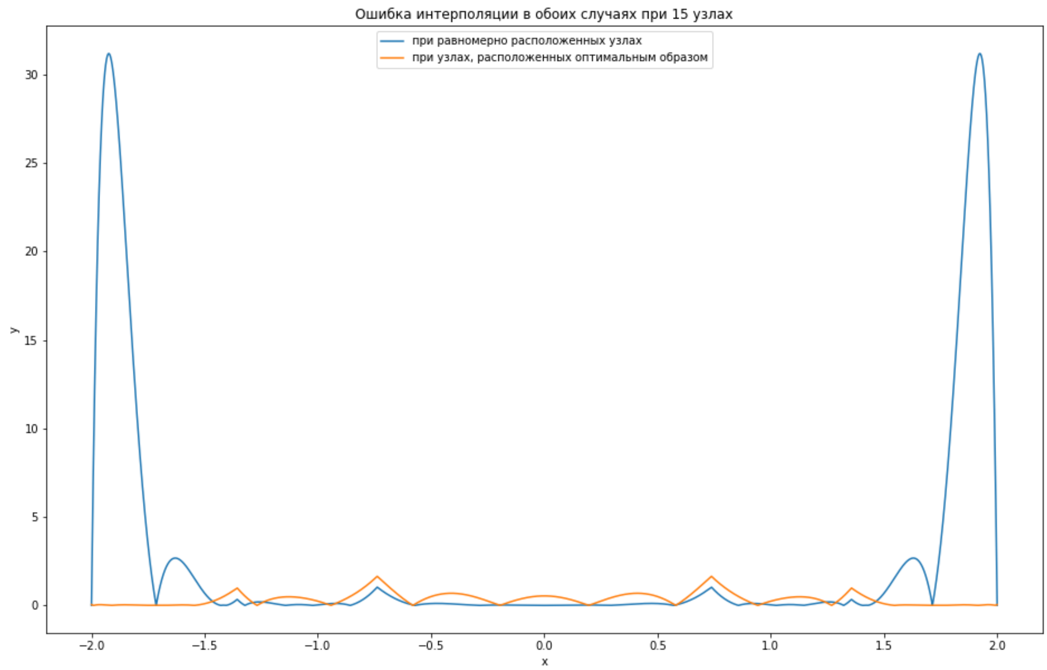

Интерполяция по 7 узлам
|5*cos(3*x)+3)|
1 интерполирующая ф-ия
2 интерполирующая ф-ия
x
y


Ошибка интерполяции в обоих случаях при 7 узлах
при равномерно расположенных узлах
при узлах, расположенных оптимальным образом
x
y


Интерполяция по 10 узлам
|5*cos(3*x)+3)|
1 интерполирующая ф-ия
2 интерполирующая ф-ия
y
x


Ошибка интерполяции в обоих случаях при 10 узлах
при равномерно расположенных узлах
при узлах, расположенных оптимальным образом
y
x





5 Выводы

Из результатов работы следует, что интерполяционный процесс сходится при увеличении количества узлов и если взять в качестве узлов корни многочлена наименее отклоняющегося от нуля, то существует алгебраический многочлен наилучшего приближения, который можно построить, например, методом Ньютона