

lab2

April 24, 2019

1 2

:

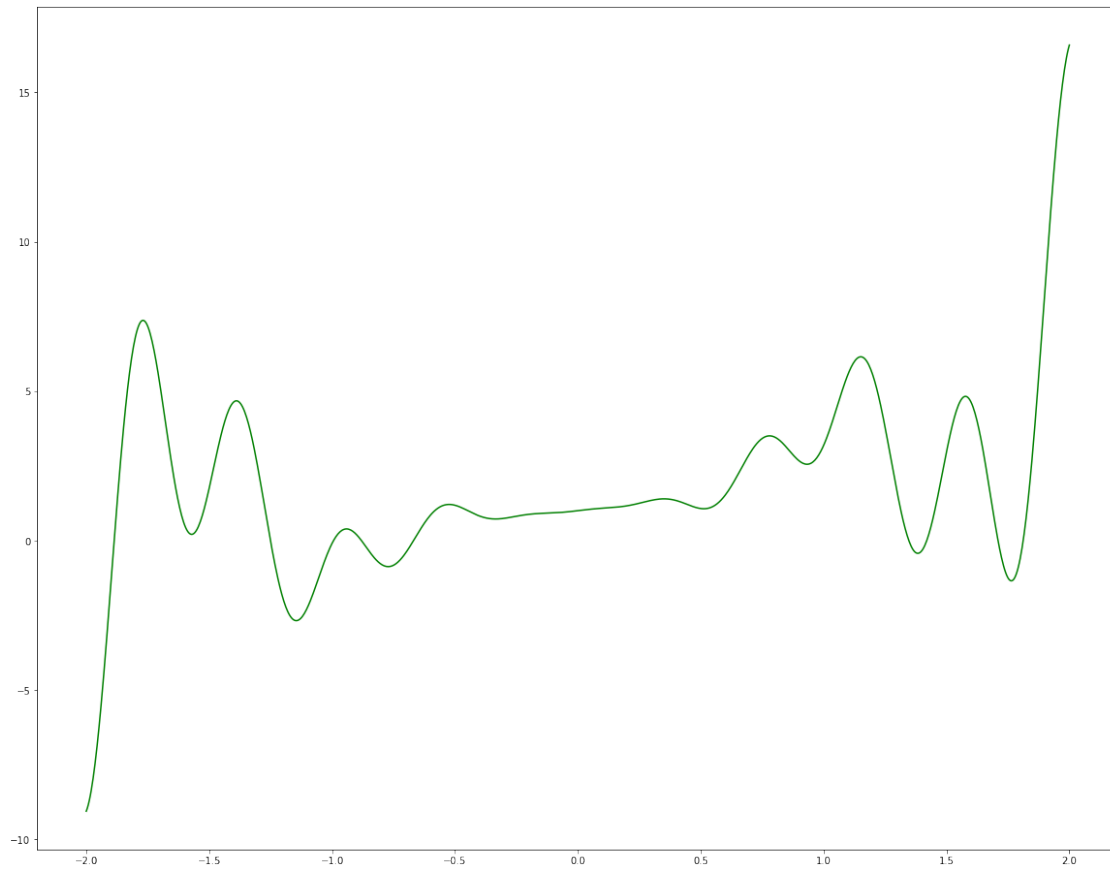
```
In [4]: #
        from math import *
        def f(x):
            return e**x - 3*x**2*sin(10*x)*cos(5*x)

In [5]: #
        def f_np(x):
            return e**x - 3*x**2*np.sin(10*x)*np.cos(5*x)

In [6]: # ( )
        import numpy as np
        import matplotlib.pyplot as plt

        f(x):

In [7]: # - , -
        rr = np.arange(-2, 2+1e-3, 4*1e-3)
        table = [[0 for x in range(7)] for y in range(10)]
        for i in range(0, 10, 1):
            table[i][0] = 10*(i+1)
        plt.figure(figsize=(20, 16))
        plt.plot(rr, f_np(rr), color='green')
        plt.show()
```



1.0.1 1:

```
In [8]: #
def coef(x, y):
    n = len(x)
    a = []
    for i in range(n):
        a.append(y[i])
    for j in range(1, n):
        for i in range(n-1, j-1, -1):
            a[i] = float(a[i]-a[i-1])/float(x[i]-x[i-j])
    return a

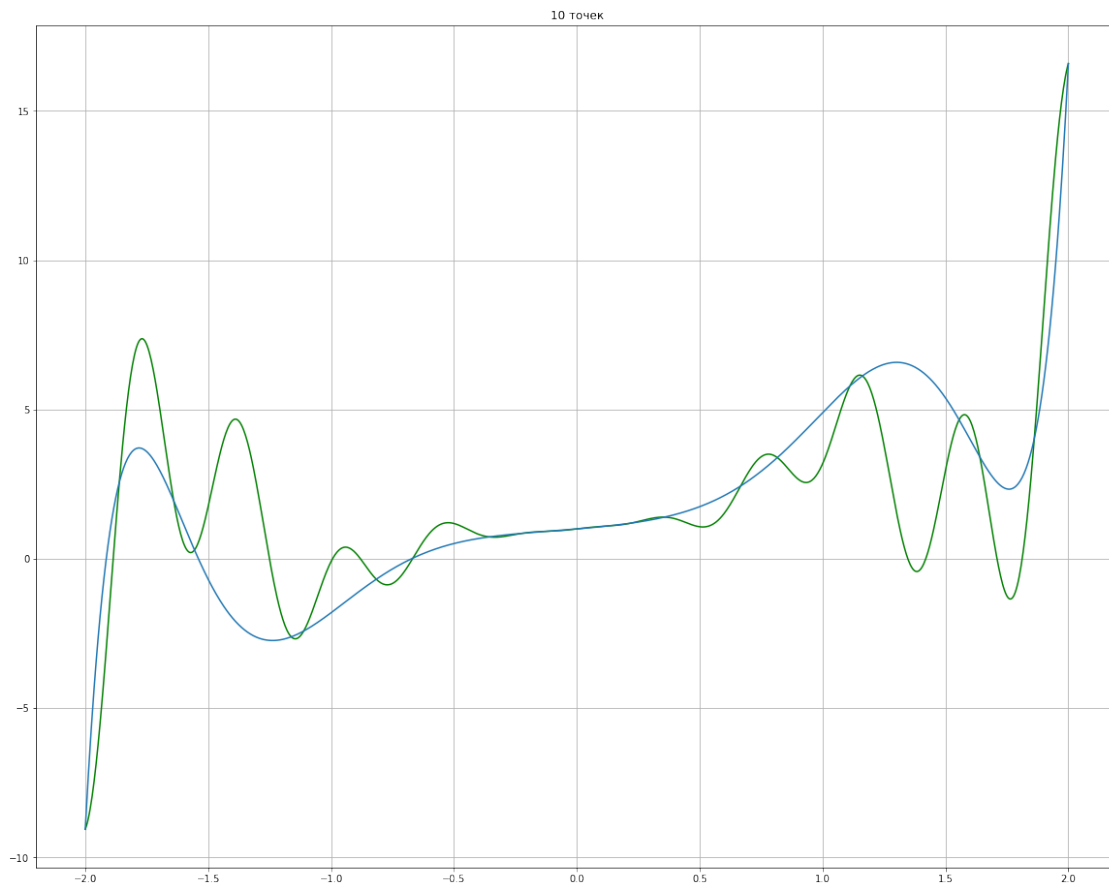
#
def eval(a, x, r):
    n = len(a) - 1
    temp = a[n]
    for i in range(n - 1, -1, -1):
        temp = temp * (r - x[i]) + a[i]
    return temp
```

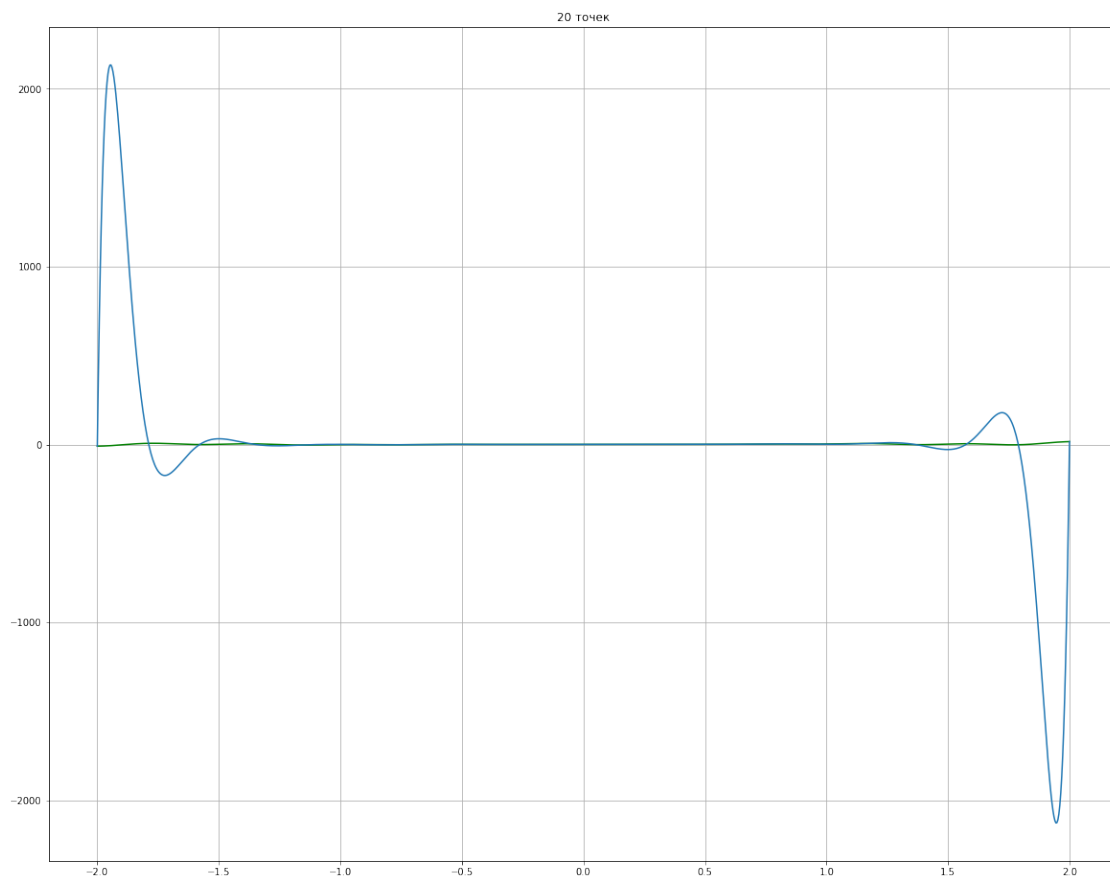
```
#10i
```

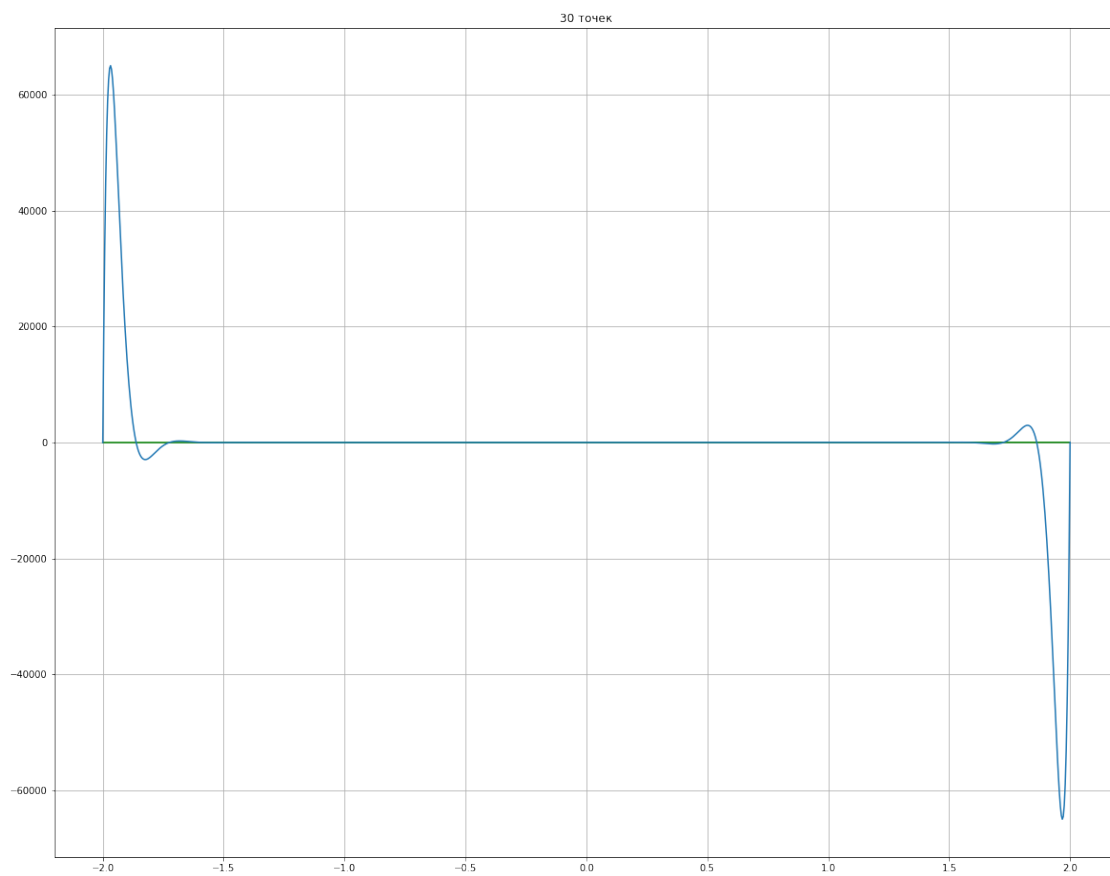
```
import time
for i in range(0, 10, 1):
    start_time=time.time()
    xs_i_nwt=[]
    ys_i_nwt=[]
    for j in np.arange(-2, 2+2/(10*i+9), 4/(10*i+9)):
        xs_i_nwt.append(j)
        ys_i_nwt.append(f(j))
    arr_i_nwt=coef(xs_i_nwt, ys_i_nwt)
    table[i][2]=(time.time() - start_time)*10e3

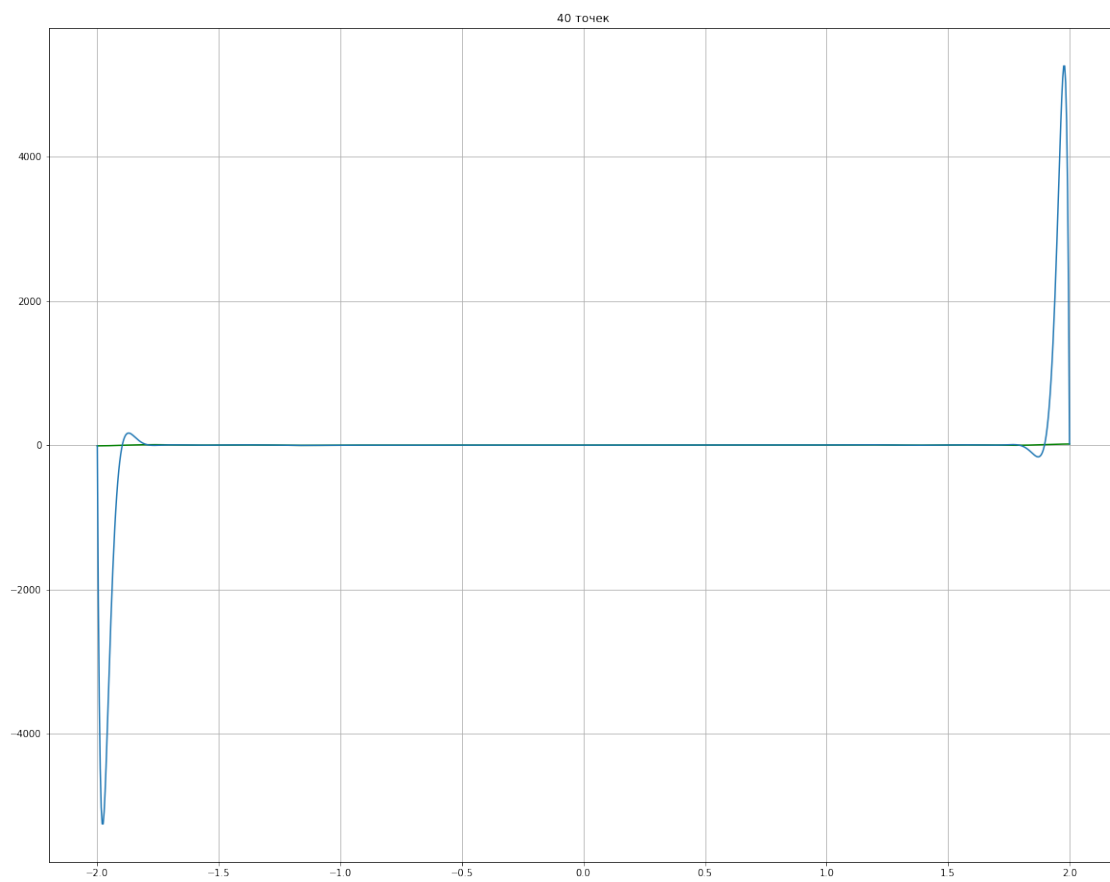
plt.figure(figsize=(20, 16))
plt.title(str(10 *(i+1)) + ' ')
plt.plot(rr, f_np(rr), color='green')
plt.plot(rr, list(map(lambda i : eval(arr_i_nwt, xs_i_nwt, i), rr)))
plt.grid()
plt.show()

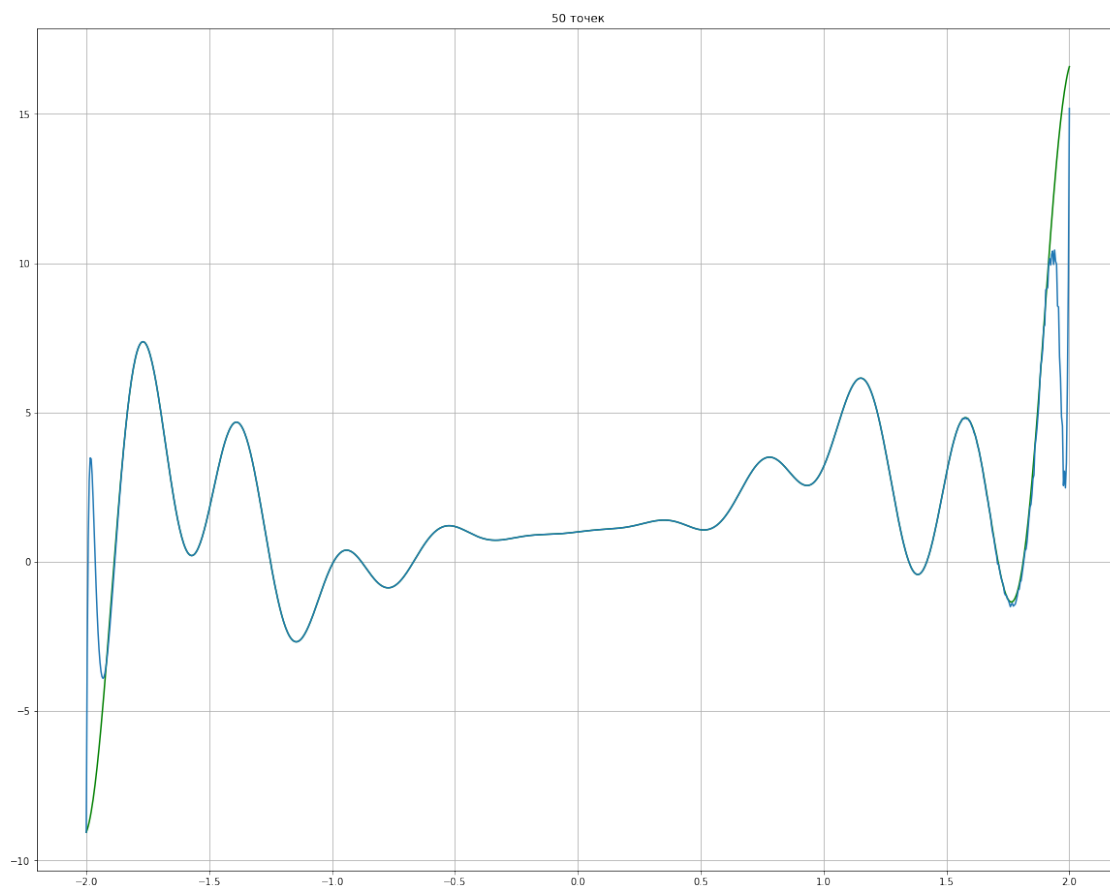
table[i][1]=max(list(map(lambda i : abs(f(i)-eval(arr_i_nwt, xs_i_nwt, i)), rr)))
```

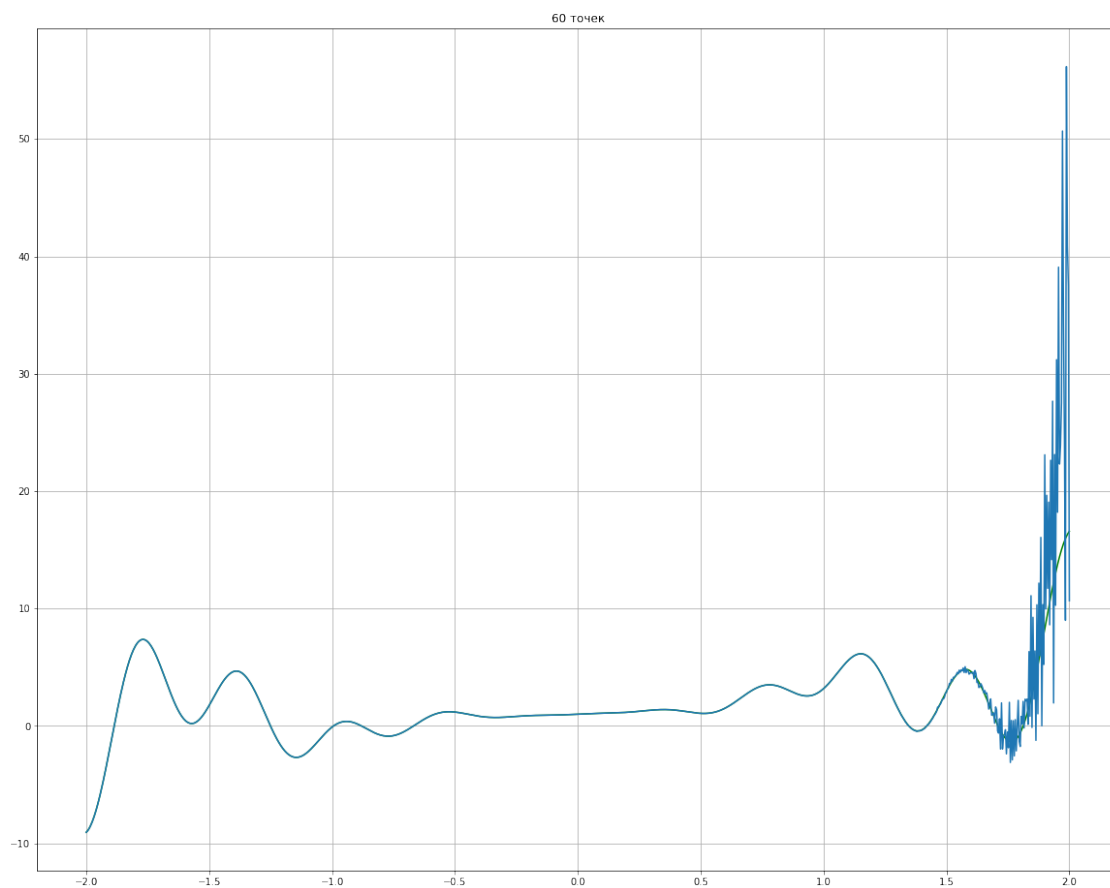


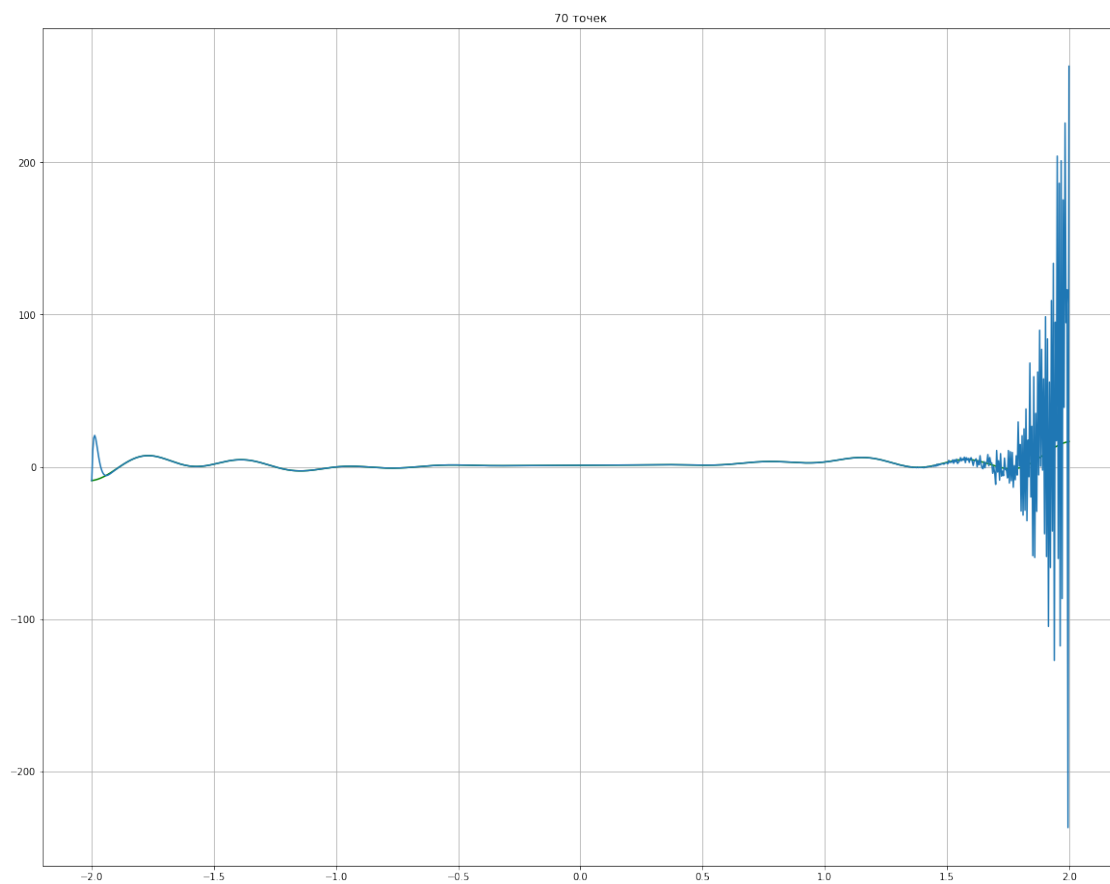


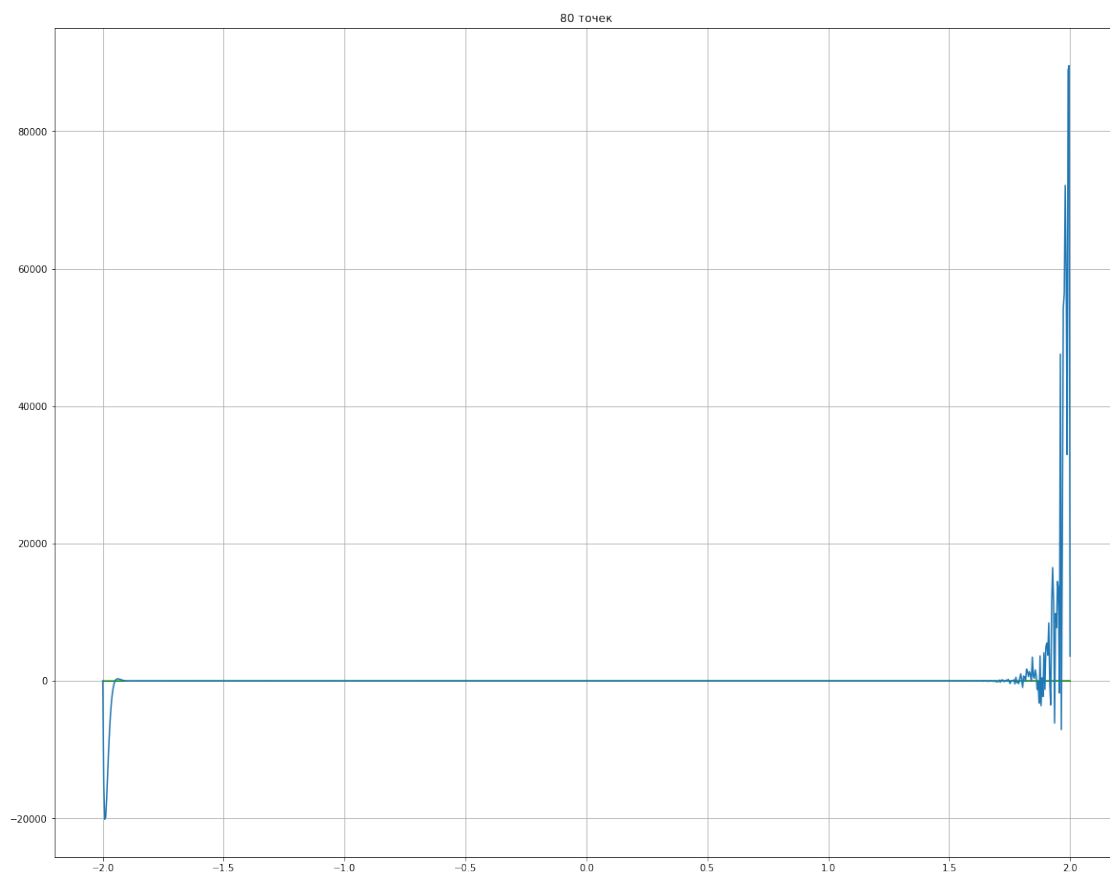


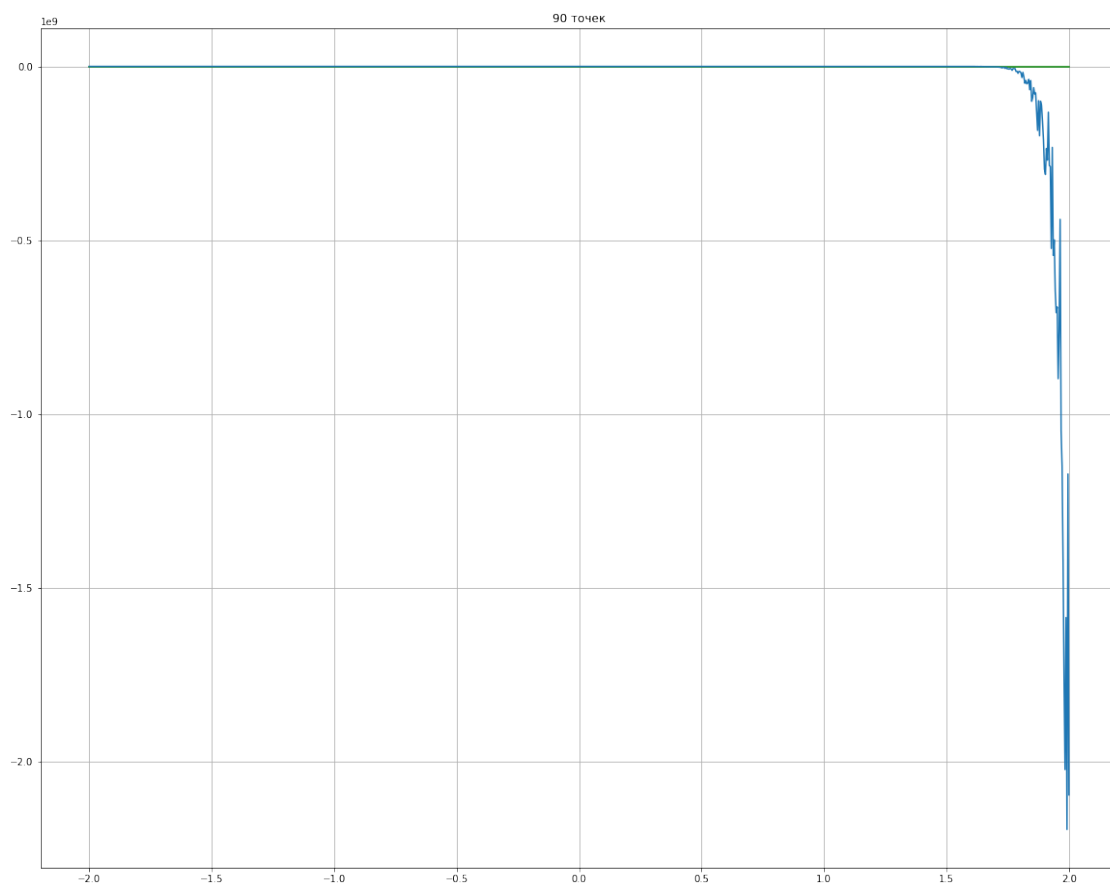


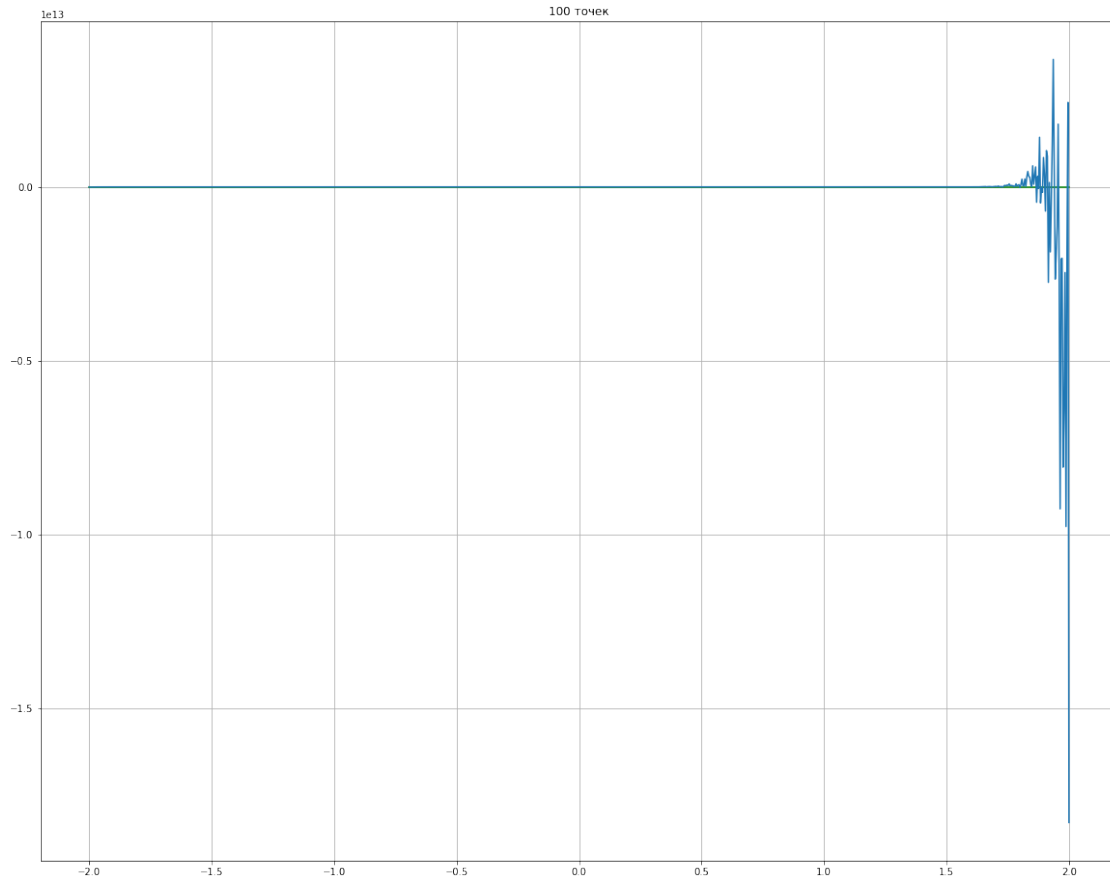












:

1.0.2 2:

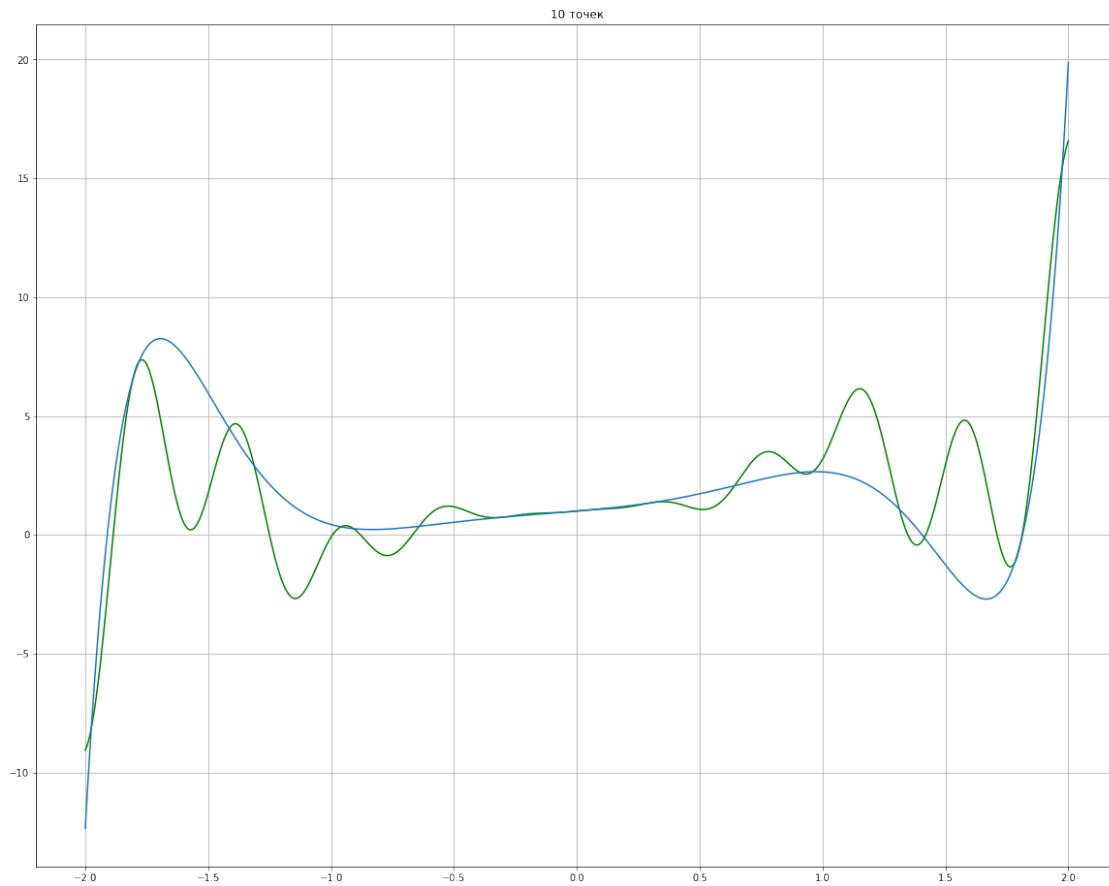
```
In [9]: # n      [a, b]
def chebyshev_newton(a, b, n):
    xs=[]
    for k in range(1, n+1):
        xs.append(0.5*(a+b)+0.5*(b-a)*cos((2*k-1)*pi/(2*n)))
    return sorted(xs)

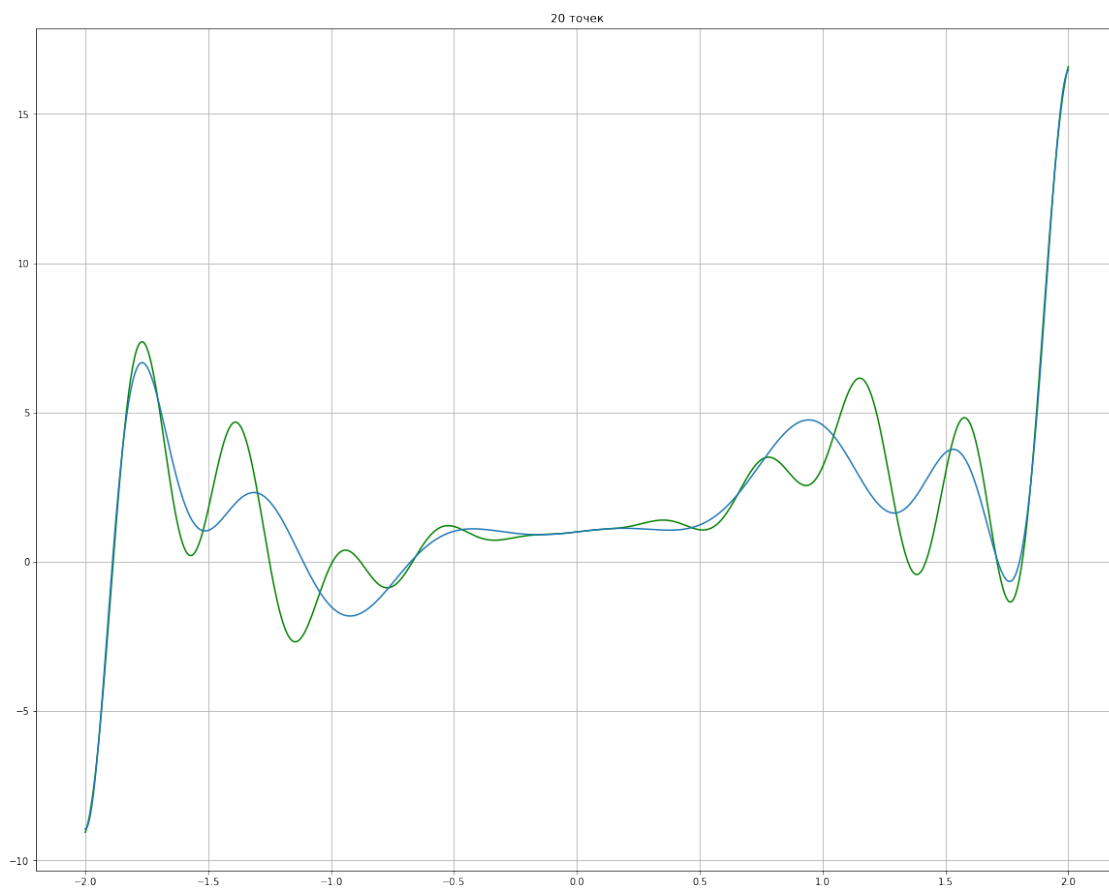
#10i
for i in range (0, 10, 1):
    start_time=time.time()
    xs_i_nwt_cheb=chebyshev_newton(-2, 2, 10*(i+1))
    ys_i_nwt_cheb=list(map(lambda i: f(i), xs_i_nwt_cheb))
    arr_i_nwt_cheb=coef(xs_i_nwt_cheb, ys_i_nwt_cheb)
    table[i][4]=(time.time() - start_time)*10e3

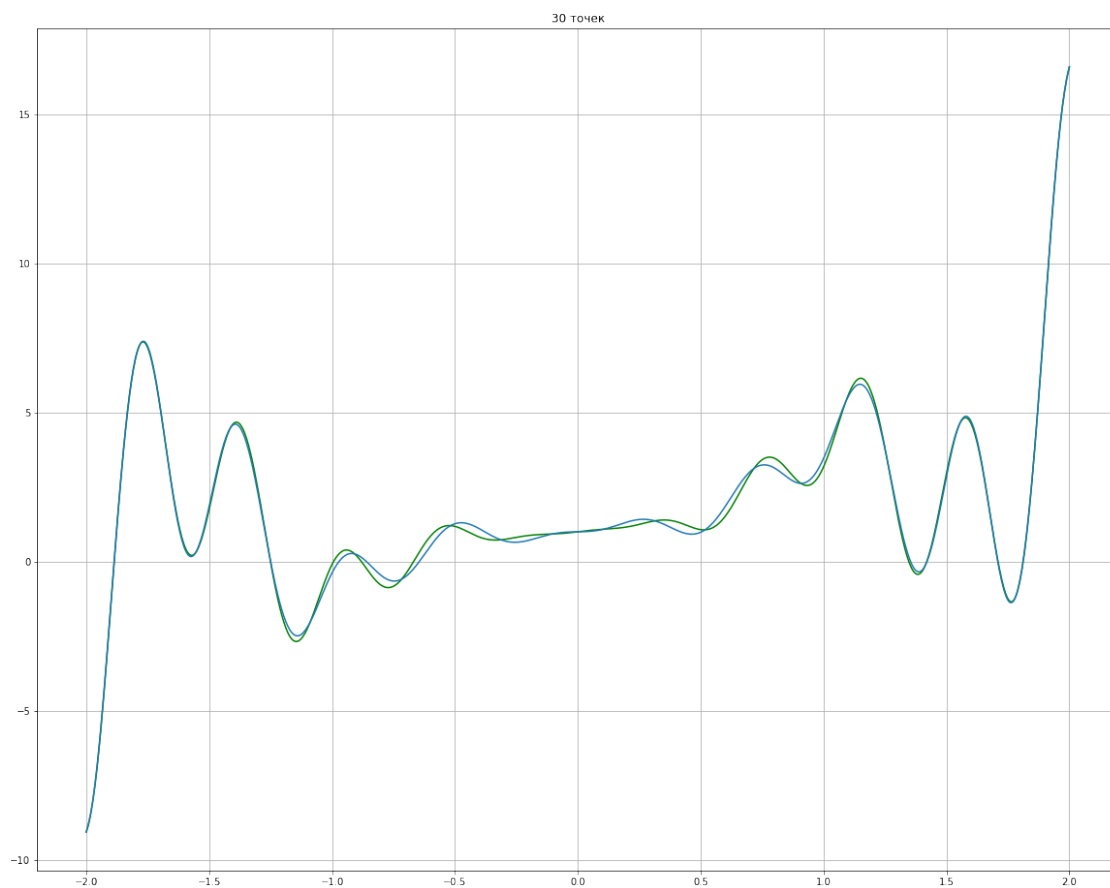
plt.figure(figsize=(20,16))
plt.title(str(10 *(i+1)) + ' ')
```

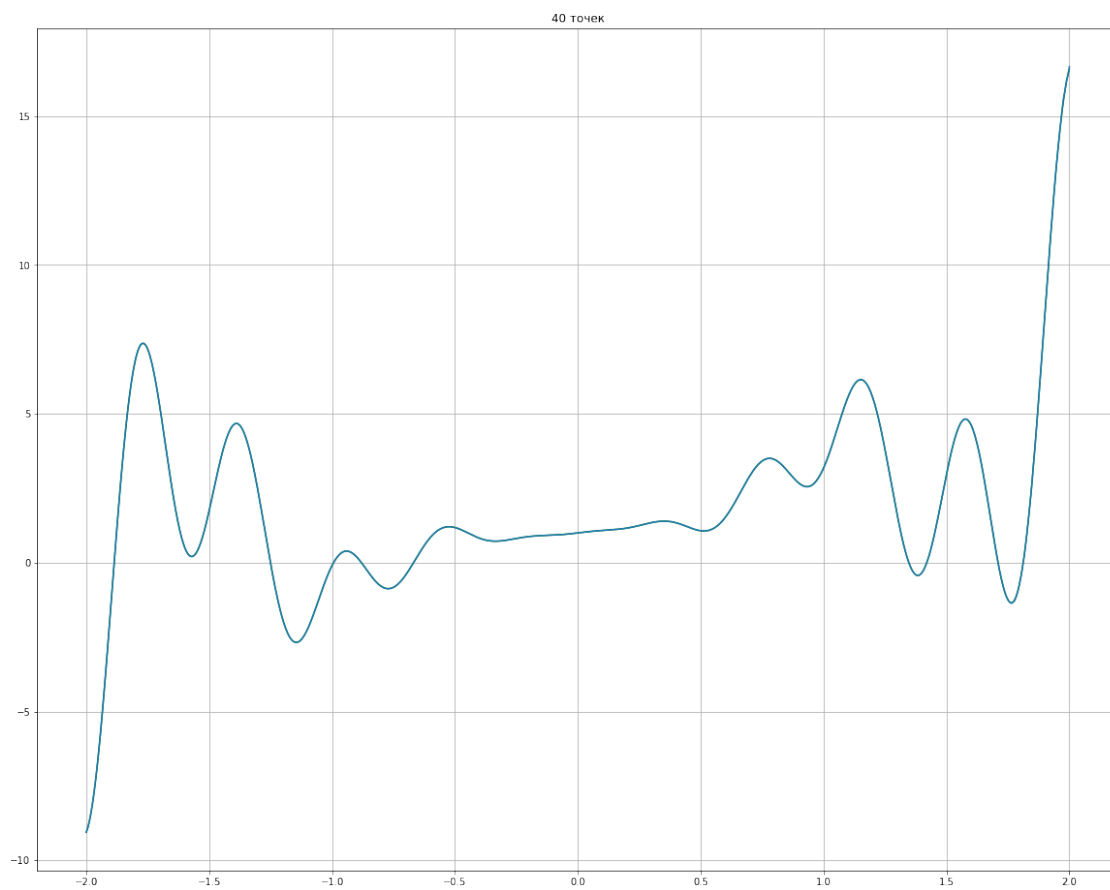
```
plt.plot(rr, f_np(rr), color='green')
plt.plot(rr, list(map(lambda i : eval(arr_i_nwt_cheb, xs_i_nwt_cheb, i), rr)))
plt.grid()
plt.show()
```

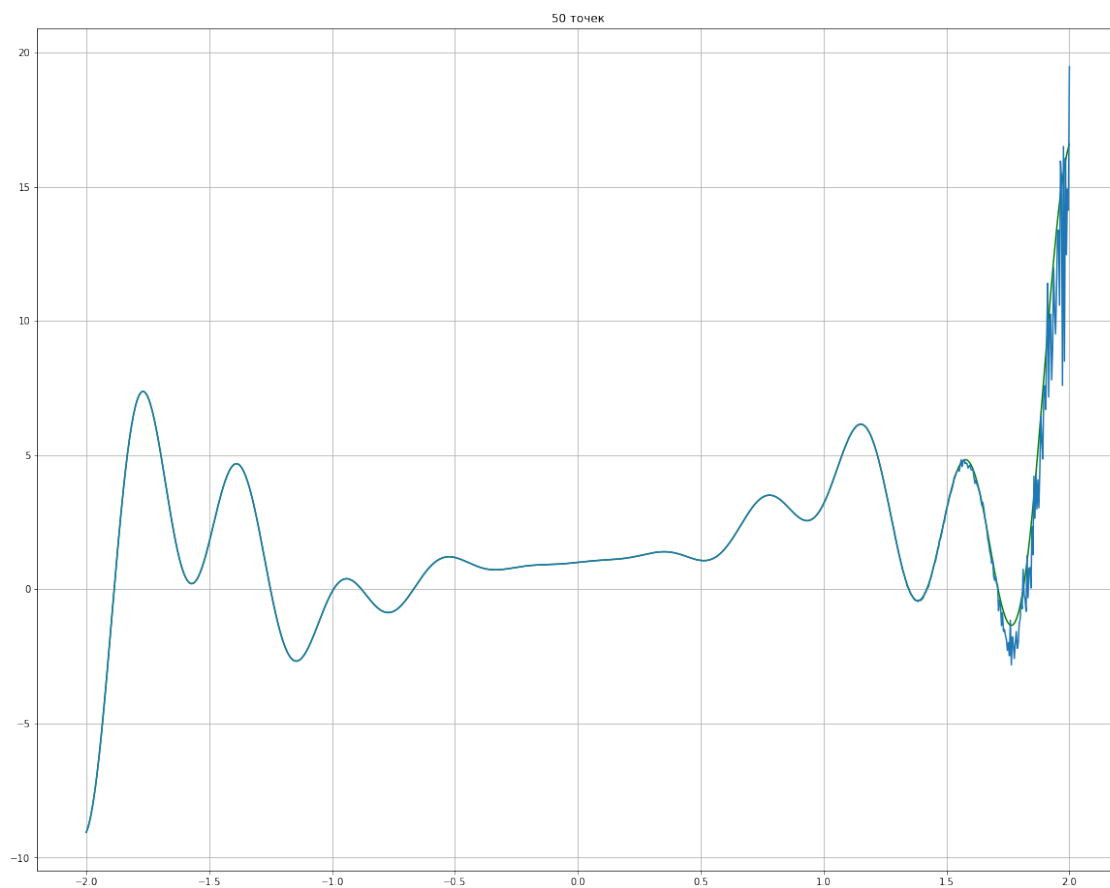
```
table[i][3]=max(list(map(lambda i : abs(f(i)-eval(arr_i_nwt_cheb, xs_i_nwt_cheb, i))
```

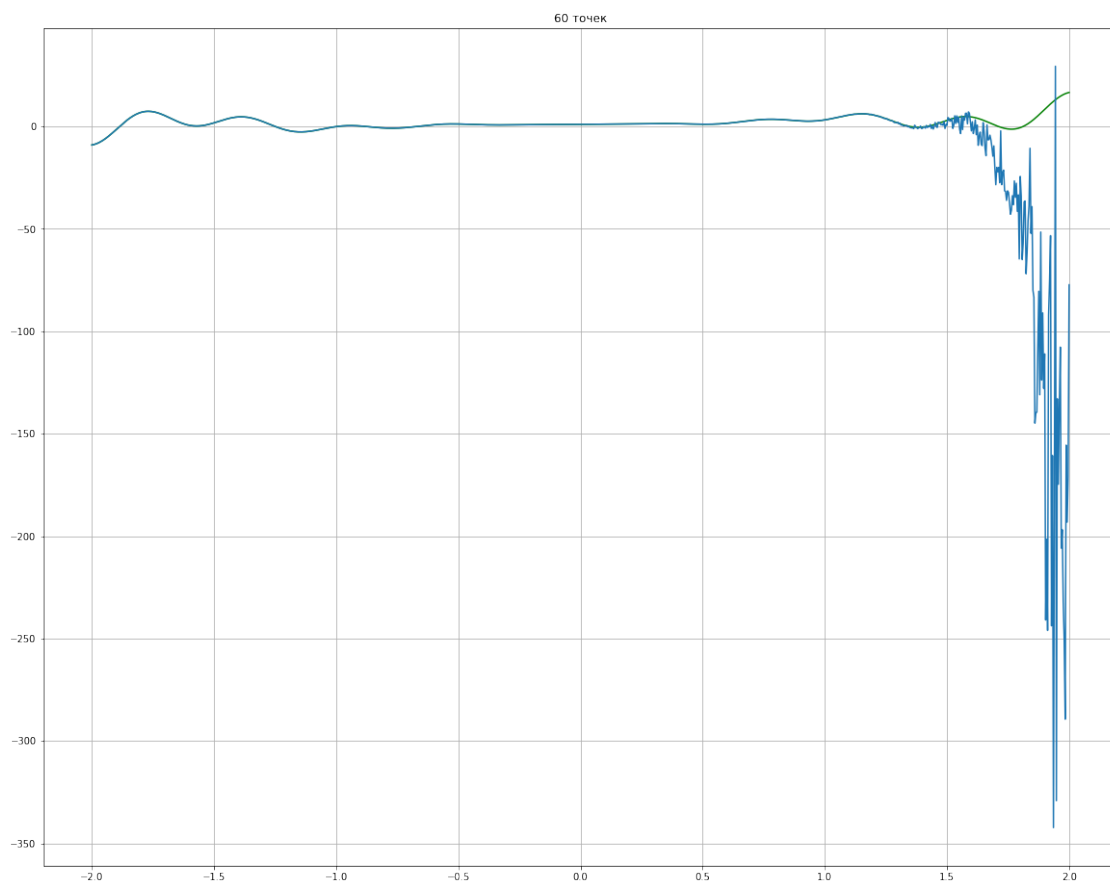


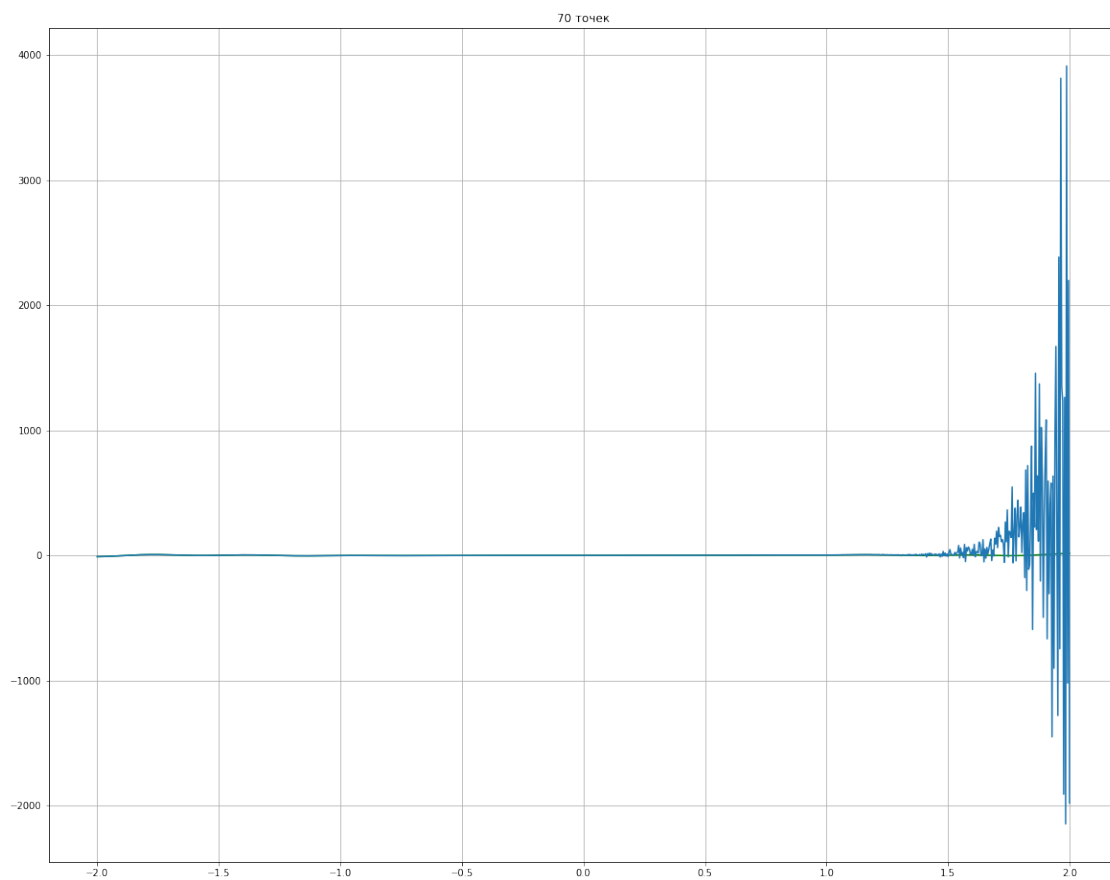


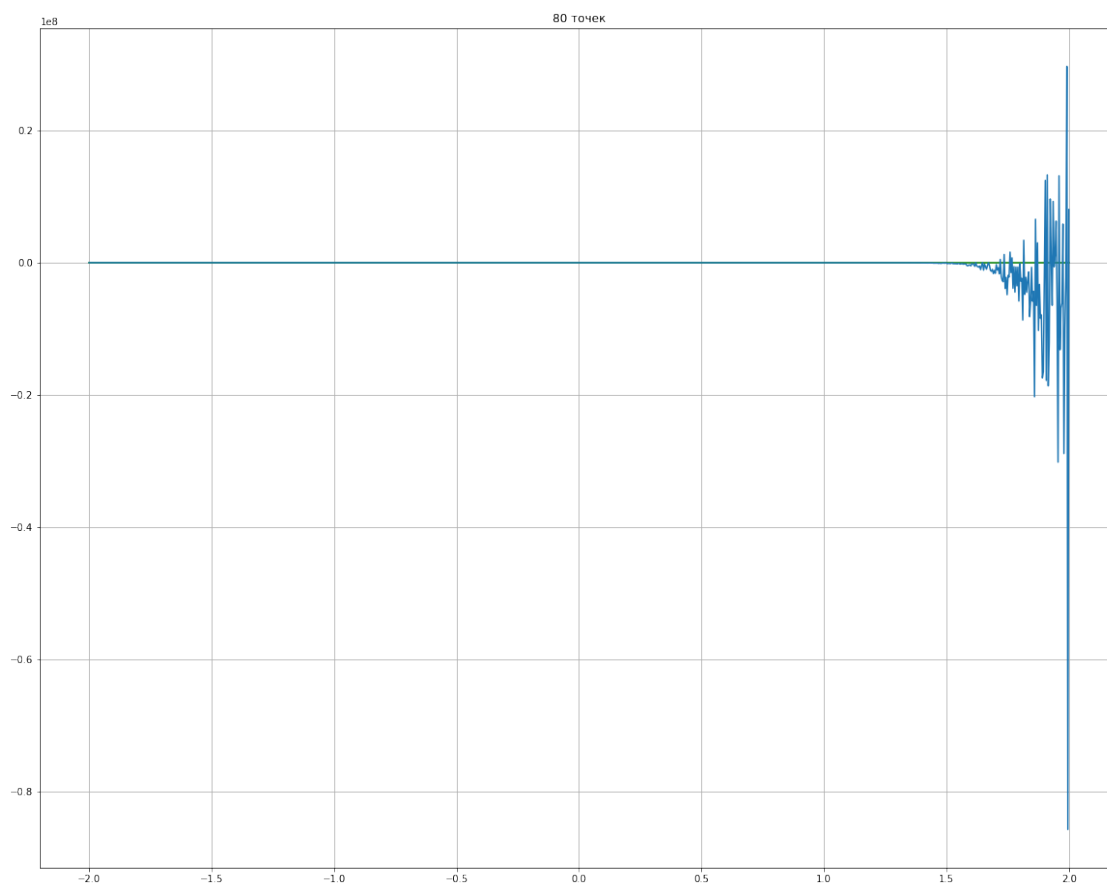


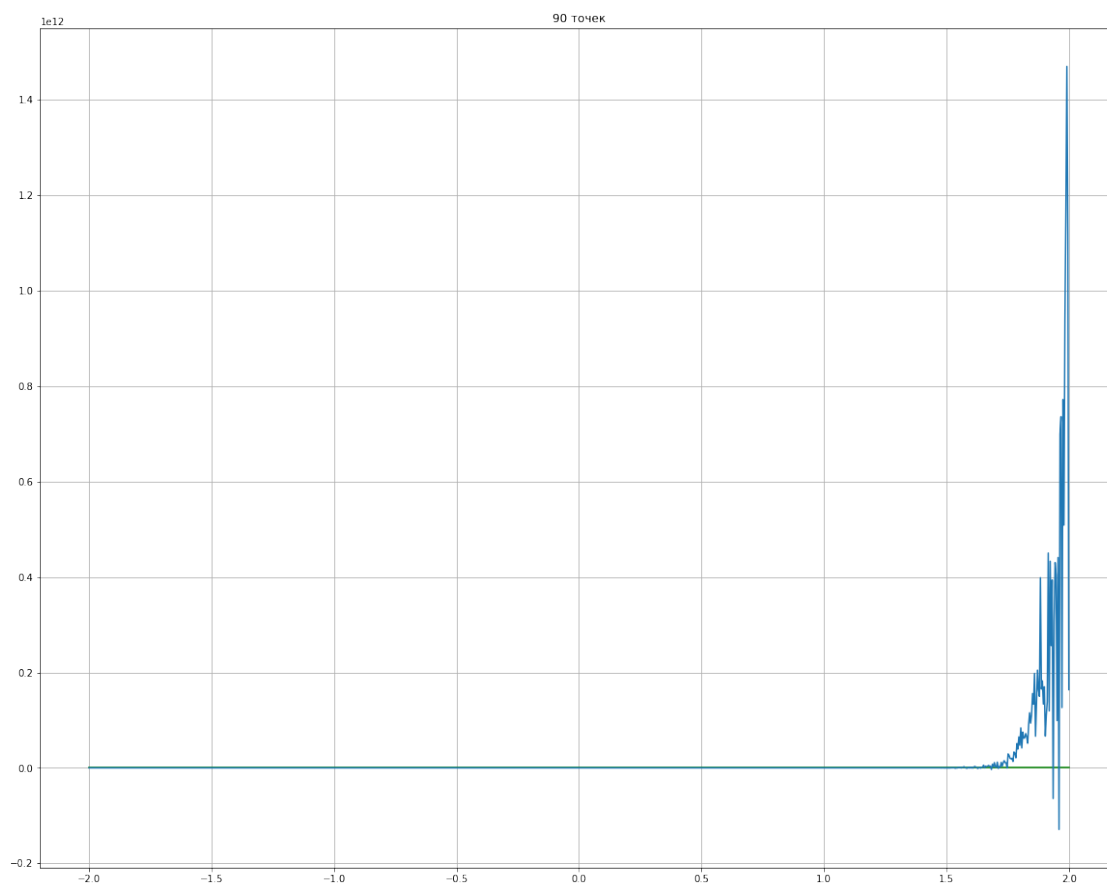


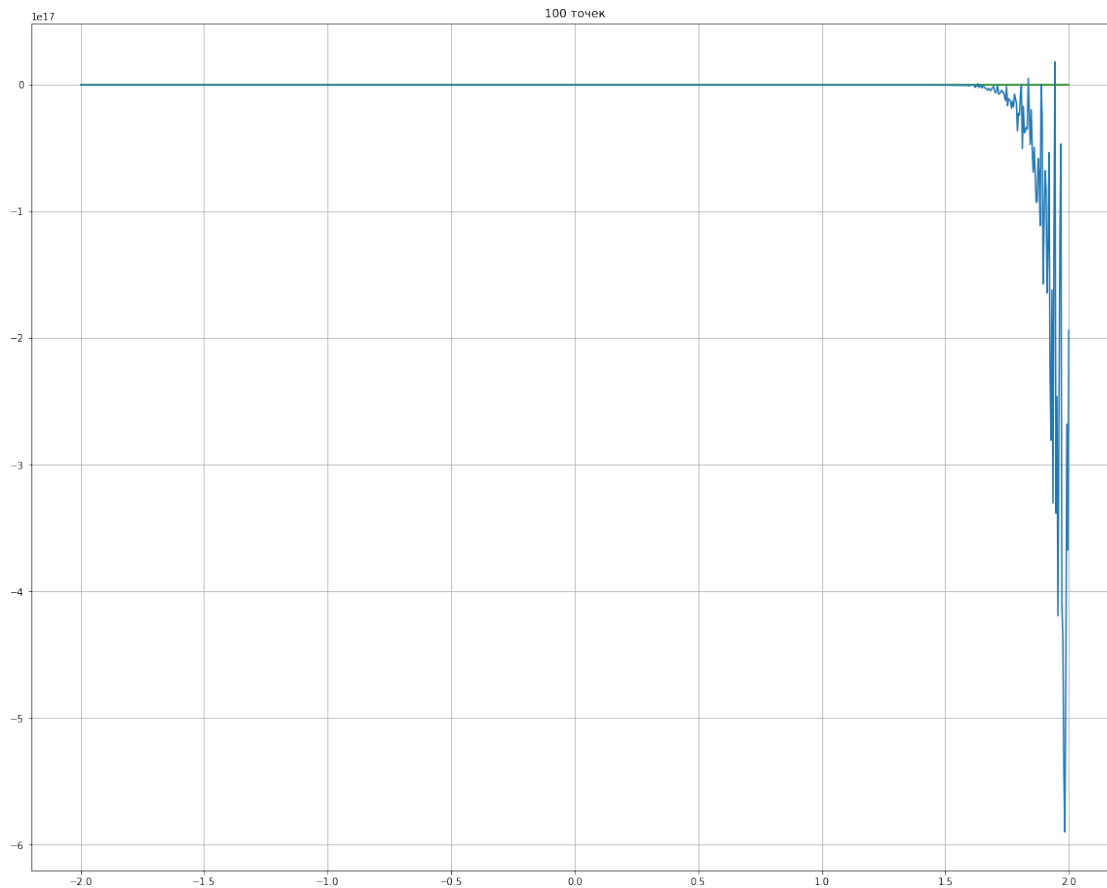












: ,

1.0.3 3:

```
In [11]: def cubic_interp1d(x0, x, y):
    size = len(x)
    xdiff = []
    ydiff = []
    for i in range (len(x)-1):
        xdiff.append(x[i+1]-x[i])
        ydiff.append(y[i+1]-y[i])
    Li = np.empty(size)
    Li_1 = np.empty(size-1)
    z = np.empty(size)
    Li[0] = sqrt(2*xdiff[0])
    Li_1[0] = 0.0
    B0 = 0.0
    z[0] = B0 / Li[0]
    for i in range(1, size-1, 1):
        Li_1[i] = xdiff[i-1] / Li[i-1]
```

```

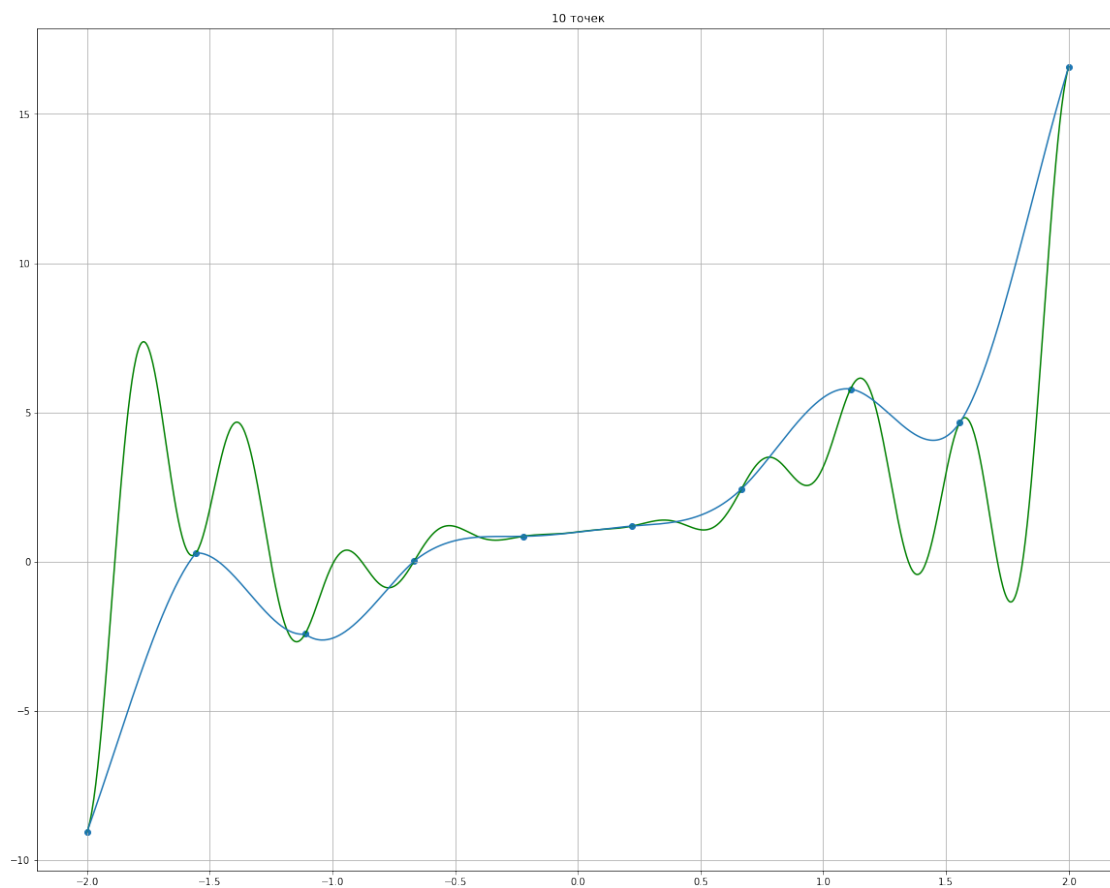
        Li[i] = sqrt(2*(xdiff[i-1]+xdiff[i]) - Li_1[i-1] * Li_1[i-1])
        Bi = 6*(ydiff[i]/xdiff[i] - ydiff[i-1]/xdiff[i-1])
        z[i] = (Bi - Li_1[i-1]*z[i-1])/Li[i]
    i = size - 1
    Li_1[i-1] = xdiff[-1] / Li[i-1]
    Li[i] = sqrt(2*xdiff[-1] - Li_1[i-1] * Li_1[i-1])
    Bi = 0.0
    z[i] = (Bi - Li_1[i-1]*z[i-1])/Li[i]
    i = size-1
    z[i] = z[i] / Li[i]
    for i in range(size-2, -1, -1):
        z[i] = (z[i] - Li_1[i-1]*z[i+1])/Li[i]
    index = x.searchsorted(x0)
    np.clip(index, 1, size-1, index)
    xi1, xi0 = x[index], x[index-1]
    yi1, yi0 = y[index], y[index-1]
    zi1, zi0 = z[index], z[index-1]
    hi1 = xi1 - xi0
    f0 = zi0/(6*hi1)*(xi1-x0)**3 + zi1/(6*hi1)*(x0-xi0)**3 + \
        (yi1/hi1 - zi1*hi1/6)*(x0-xi0) + (yi0/hi1 - zi0*hi1/6)*(xi1-x0)
    return f0

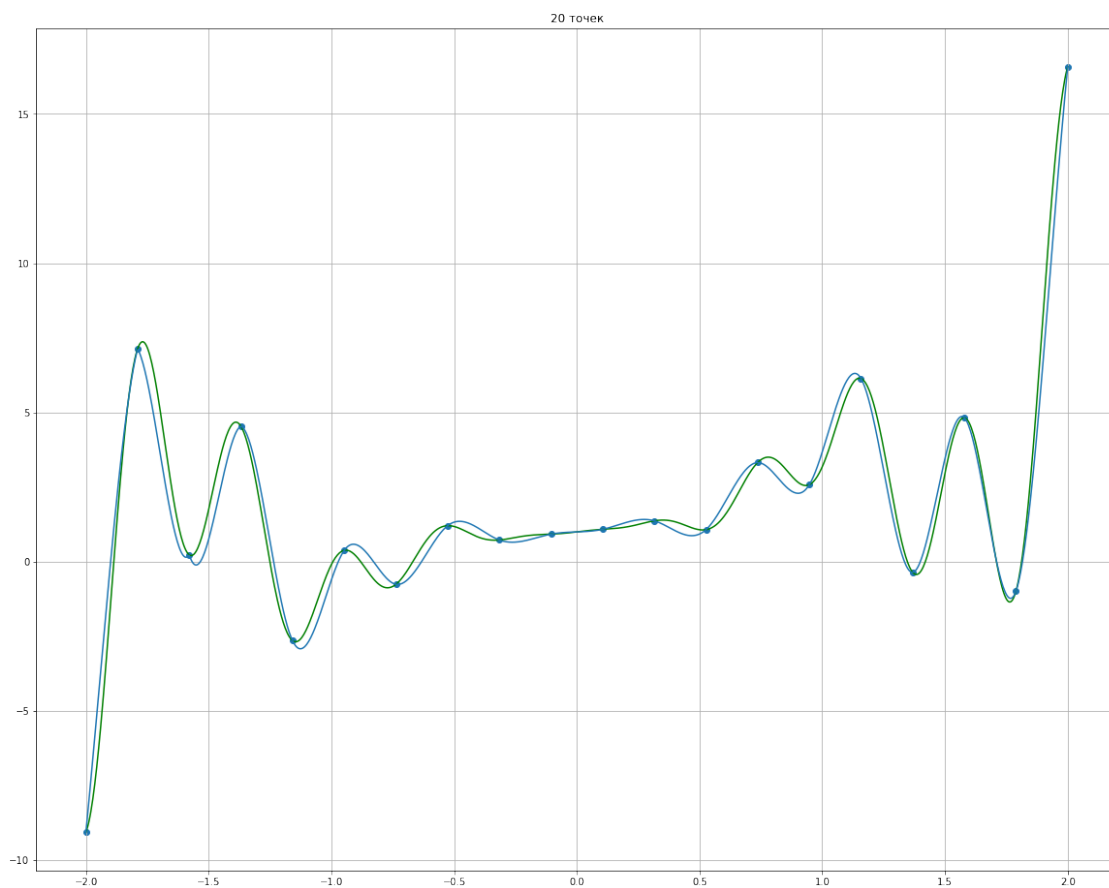
#10i
for i in range (0, 10, 1):
    start_time=time.time()
    x = np.linspace(-2, 2, 10*(i+1))
    y = f_np(x)
    f0 = cubic_interp1d(rr, x, y)
    table[i][6]=(time.time() - start_time)*10e3

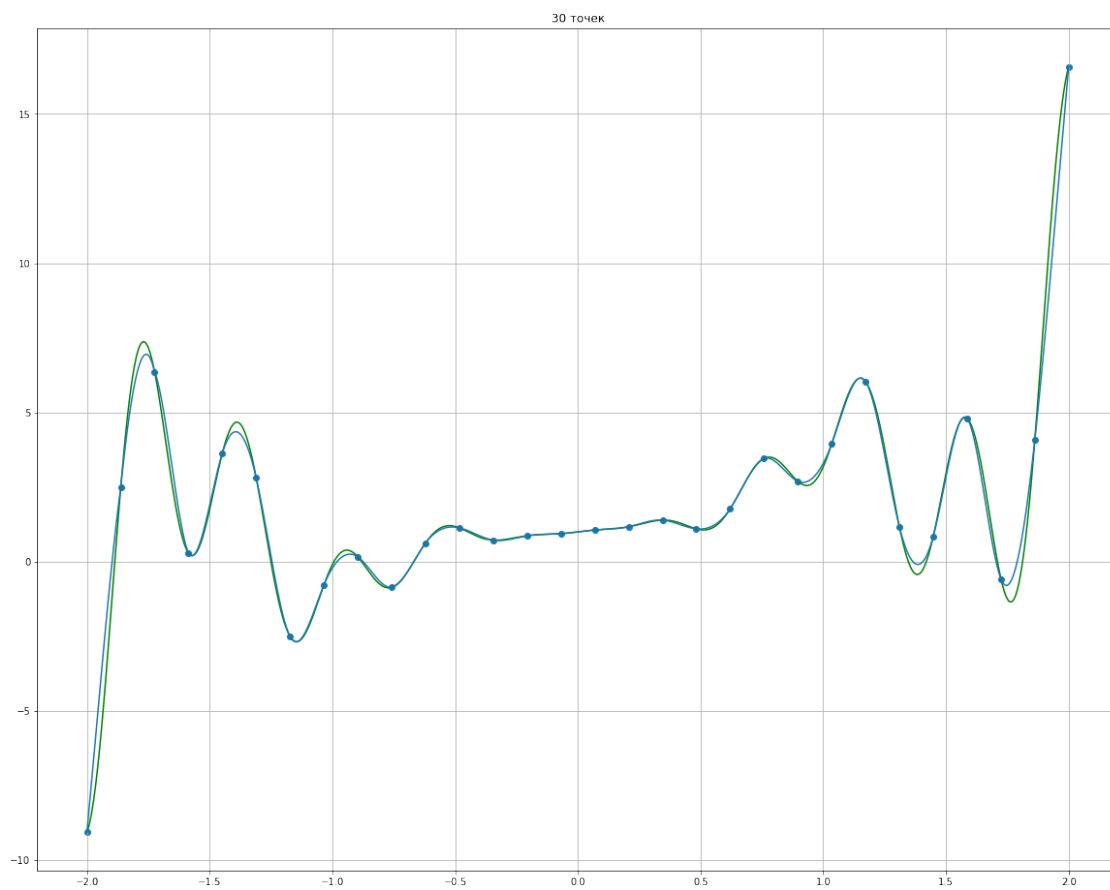
    plt.figure(figsize=(20,16))
    plt.title(str(10*(i+1)) + ' ')
    plt.scatter(x, y)
    plt.plot(rr, f_np(rr), color='green')
    plt.plot(rr, f0)
    plt.grid()
    plt.show()

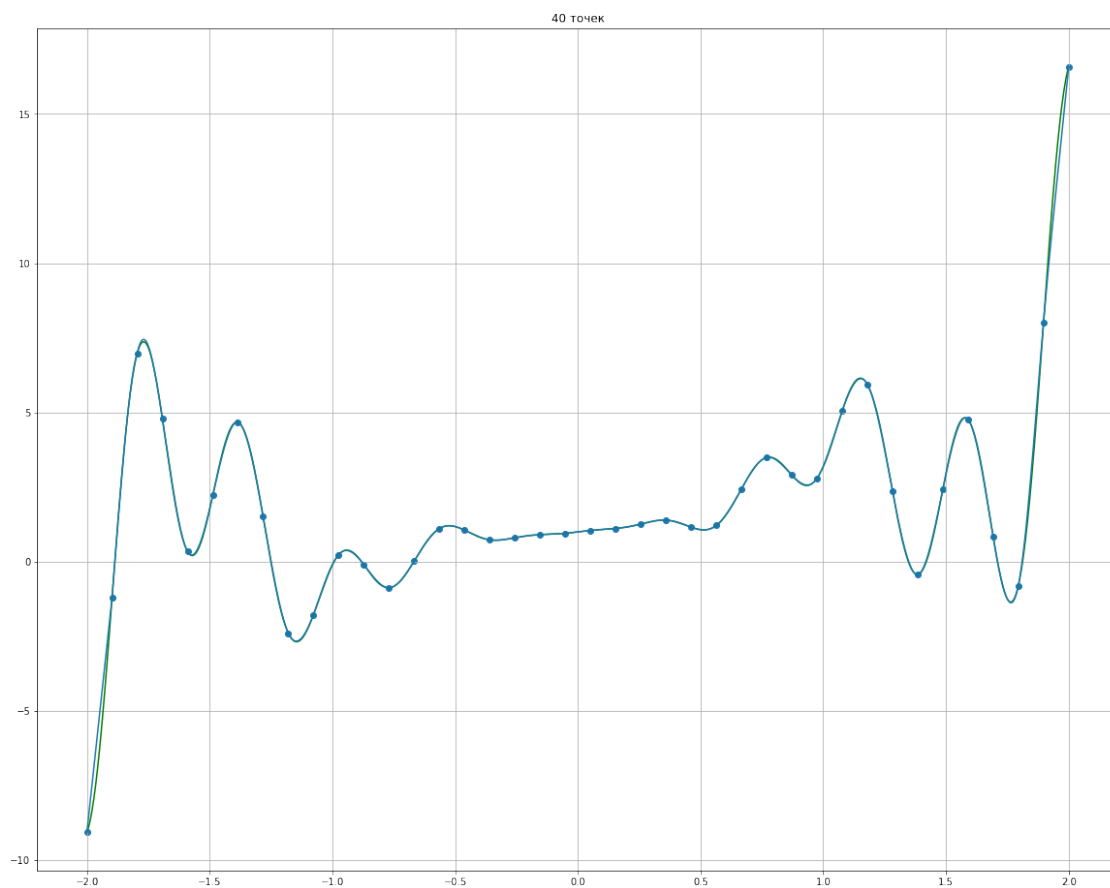
    func = f_np(rr)
    table[i][5]=max(list(map(lambda i : abs(func[i]-f0[i]), range (0, rr.size, 1))))

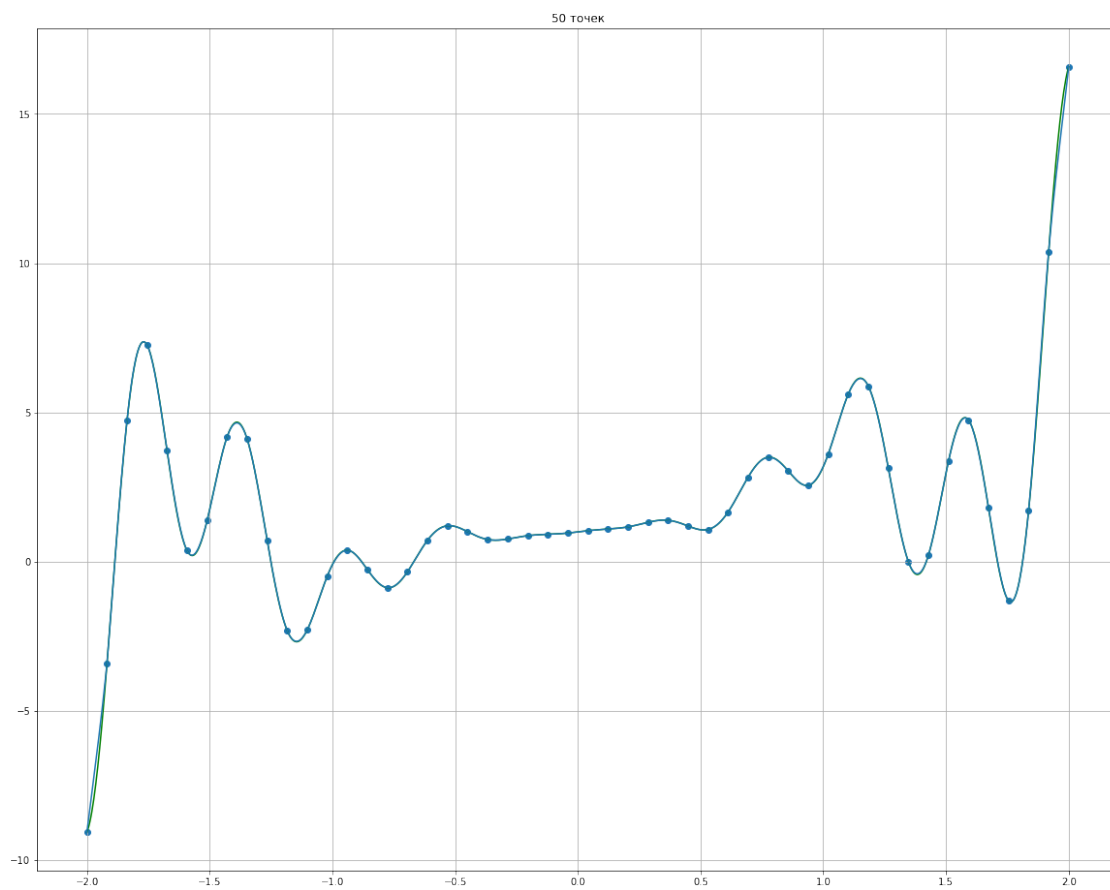
```

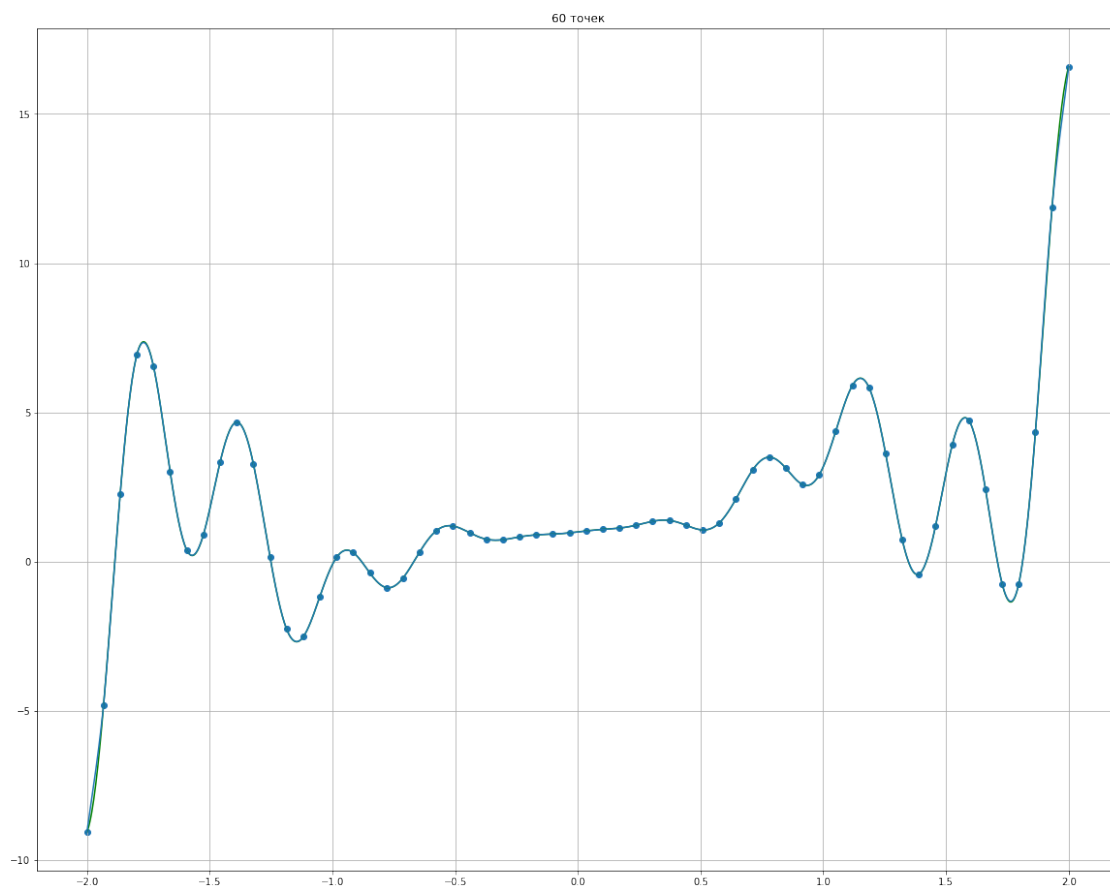


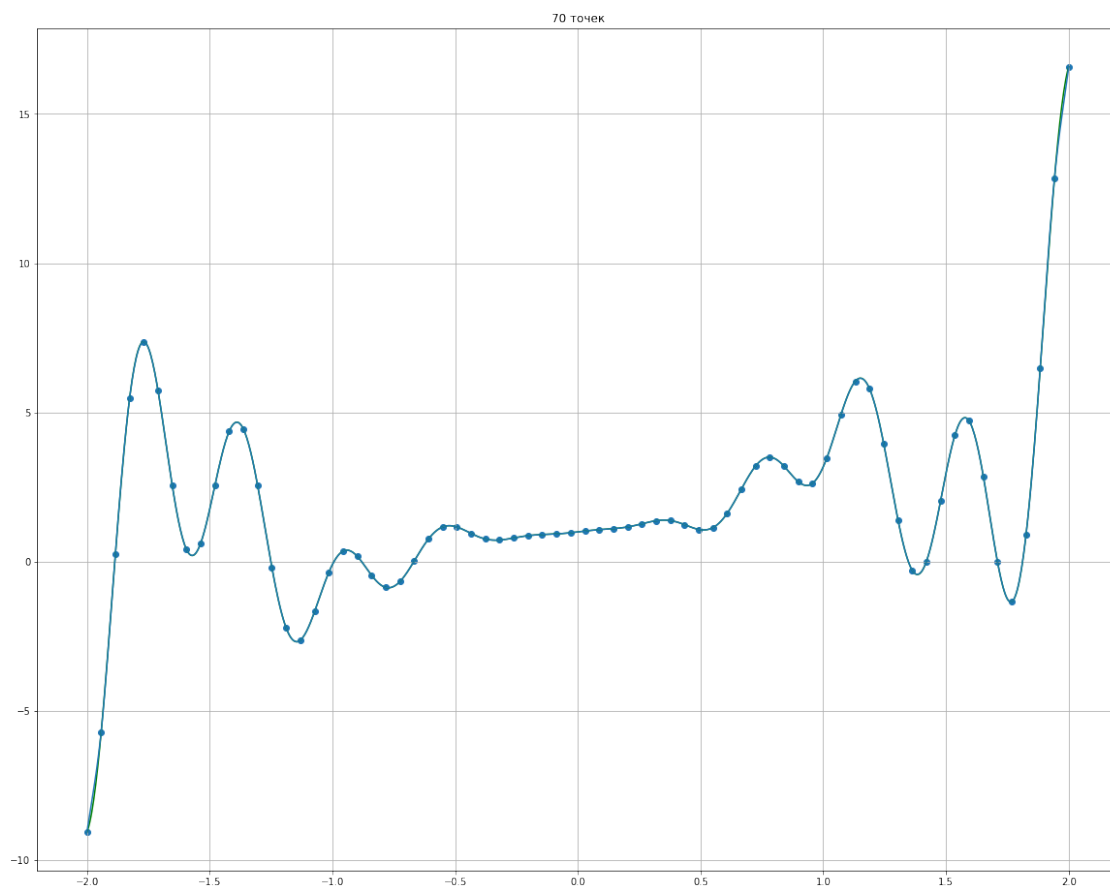


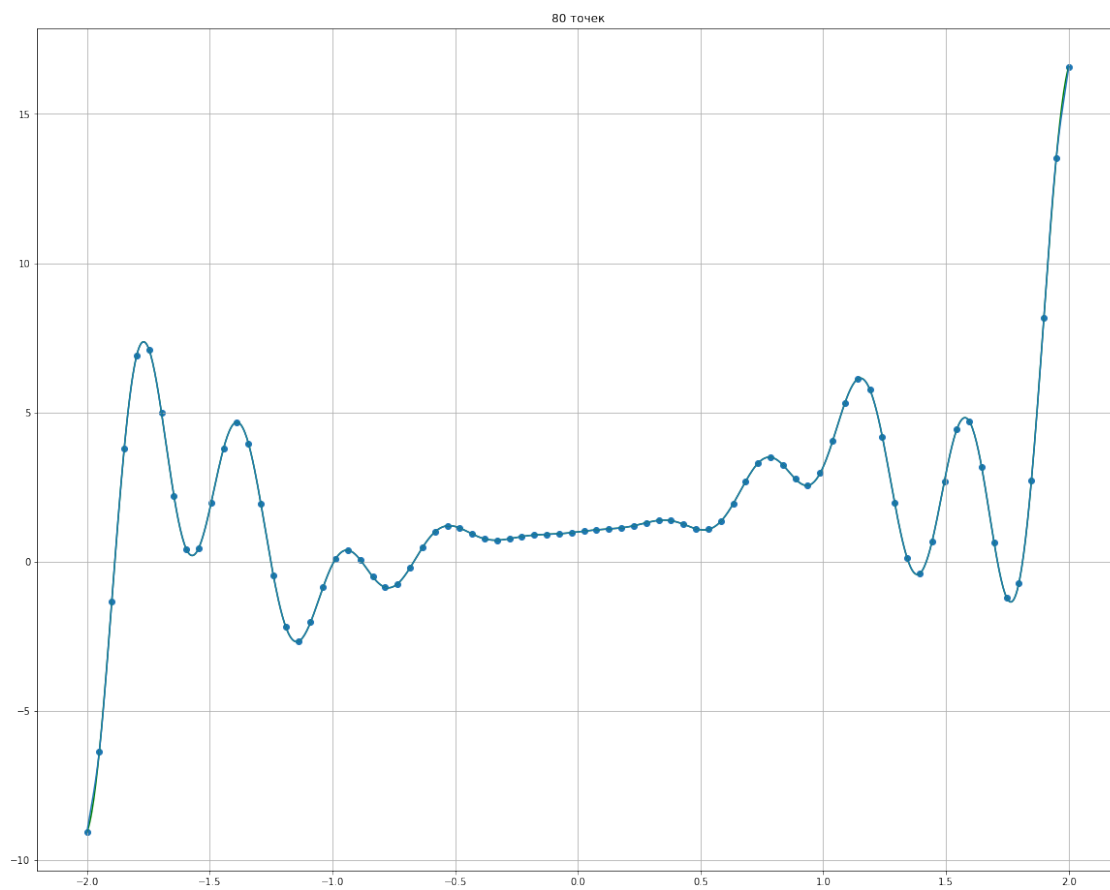


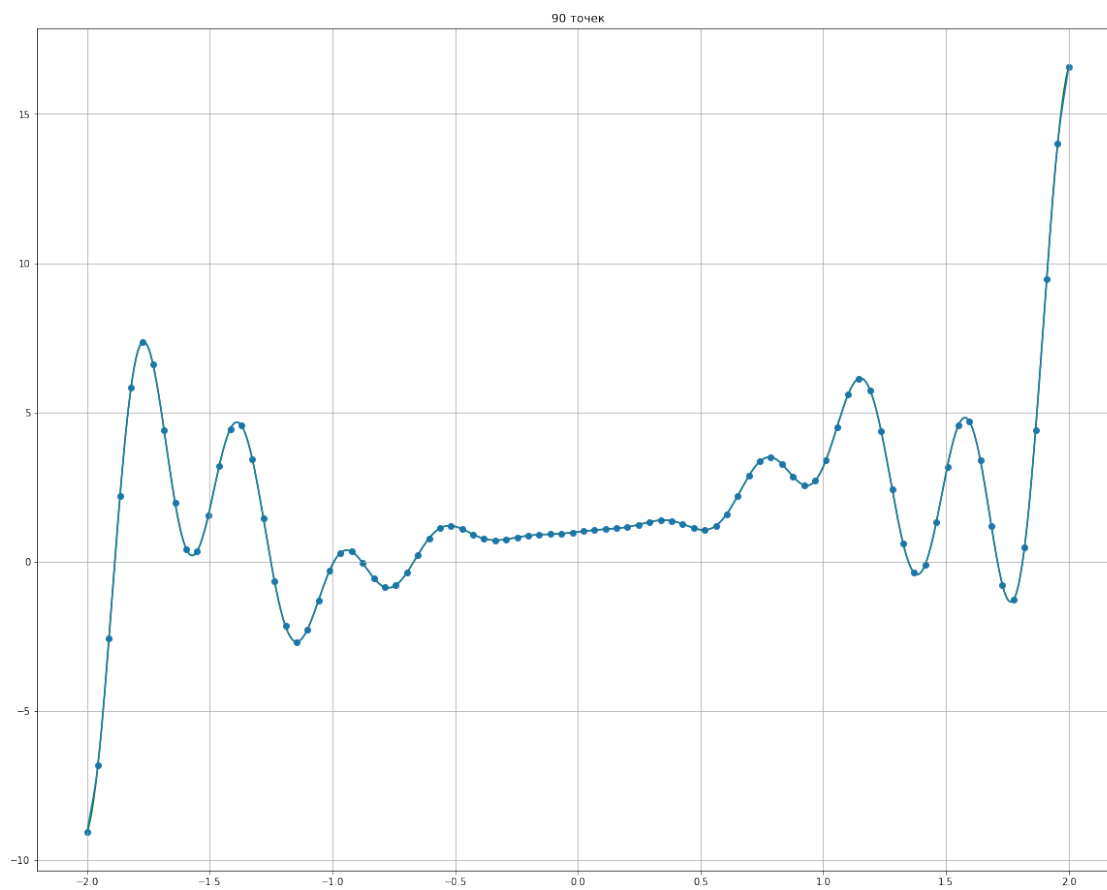


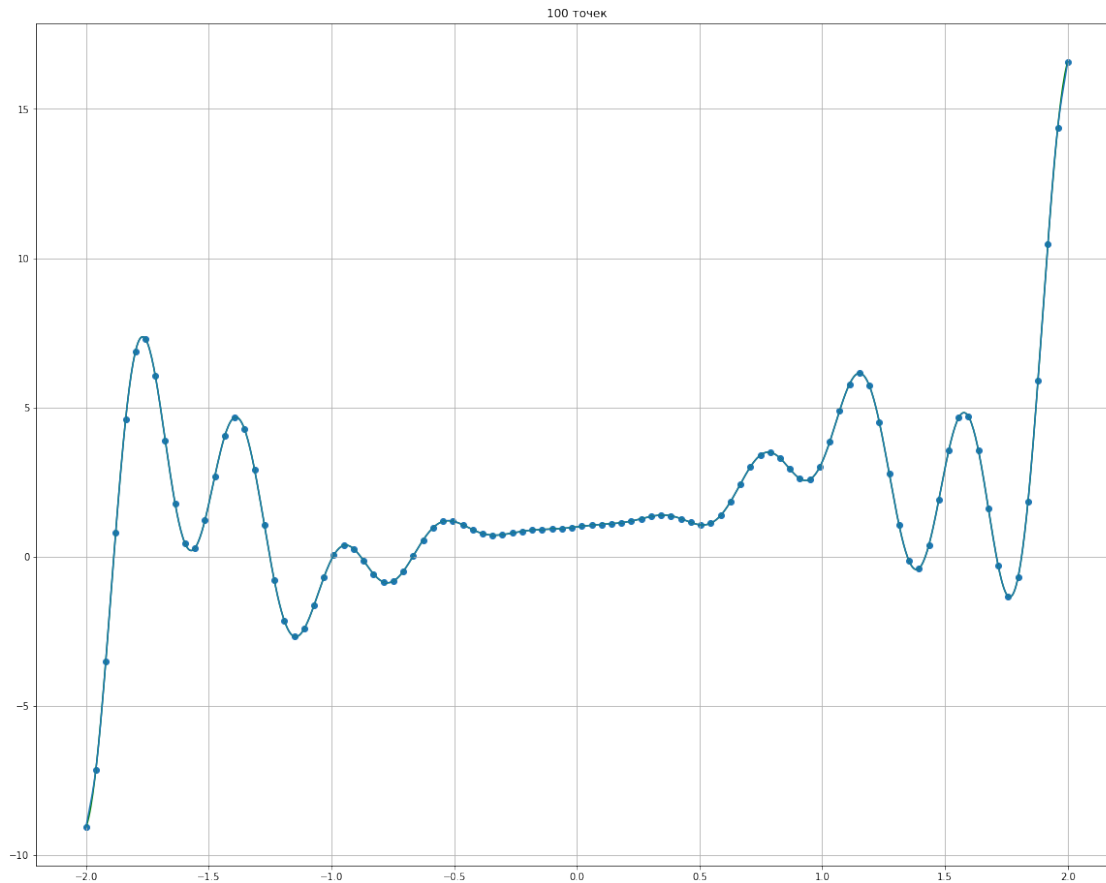












: ,

1.0.4 4:

In [12]: `import pandas as pd`

```
text = np.empty((10, 4), dtype=object)
for i in range(0, 10, 1):
    text[i][0] = str(table[i][0])
    text[i][1] = "{:.0e}".format(table[i][1]) + '/' + str(round(table[i][2])) + ''
    text[i][2] = "{:.0e}".format(table[i][3]) + '/' + str(round(table[i][4])) + ''
    text[i][3] = "{:.0e}".format(table[i][5]) + '/' + str(round(table[i][6])) + ''
collabel=("N", "()", "()", "\")
pd.DataFrame(data=text, columns=collabel)
```

Out[12]:

	N	()	()	\
0	10			7e+00/10
1	20			2e+03/10
2	30			7e+04/20
3	40			5e+03/30
4	50			1e+01/40

5	60	4e+01/40	4e+02/30
6	70	3e+02/80	4e+03/30
7	80	9e+04/70	9e+07/50
8	90	2e+09/80	1e+12/90
9	100	2e+13/150	6e+17/100

0	1e+01/10
1	2e+00/20
2	2e+00/20
3	1e+00/30
4	7e-01/20
5	5e-01/20
6	4e-01/20
7	3e-01/30
8	2e-01/30
9	2e-01/30

: ,