

Ракоть Валентин

Код программы:

```
import sys

from matplotlib import pyplot as plt
from numpy import ndarray
from scipy.special import binom
import numpy as np
import pandas as pd

a = 0
b = 0

def getBerns(step, i, x):
    global a
    global b
    x2 = (x-a)/(b-a)
    return binom(step,i)*(x2**i)*((1-x2)**(step-i))

def solve(step, x, y):
    n = len(x)
    G = []
    g = []
    for i in range(0,step):
        G.append([])
        k = 0
        for k in range(0,step):
            G[i].append(0)
    i = 0
    while i < step:
        k = 0
        sum1 = 0
        for h in range(0, n):
            sum1 += y[h]*getBerns(step,i,x[h])
        g.append(sum1)
        while k < step:
            if i > k:
                k += 1
                continue
            sum = 0
            for h in range(0,n):
                sum += getBerns(step,i,x[h])*getBerns(step,k,x[h])
            G[i][k] = sum
            G[k][i] = sum
            k += 1
        i += 1
    matr = np.matrix(G)
    y = np.array(g)
    Q, R = np.linalg.qr(matr)
    prom = np.linalg.solve(Q, y)
    resz = np.linalg.solve(R, prom)
    return resz
```

```

def pogr(x,y):
    n = len(x)
    step = 5
    ns = []
    ys = []
    while step <= 100:
        resz = solve(step, x, y)
        sum = 0
        for h in range(0, n):
            a = 0
            for k in range(0, step):
                a += resz[k] * getBersn(step, k, x[h])
            sum += (a - y[h])**2
        sum = sum**(0.5)
        sum /= (n + 1)
        ns.append(step)
        ys.append(sum)
        step += 5
    plt.semilogy(ns,ys)
    plt.show()

def draw(step, x, y):
    resz = solve(step, x, y)
    n = len(x)
    y2 = []
    for h in range(0, n):
        sum = 0
        for k in range(0, step):
            sum += resz[k] * getBersn(step, k, x[h])
        y2.append(sum)
    plt.scatter(x, y)
    plt.plot(x, y2)
    plt.show()

def main():
    global a
    global b
    res = []
    with open('data.txt', 'r') as file:
        k = 0
        for line in file:
            x = 0
            i = 0
            y = 0
            for word in line.split():
                if i == 0:
                    x = float(word)
                    i += 1
                else:
                    y = float(word)
                    res.append([])
                    res[k].append(x)
                    res[k].append(y)
                    k += 1
    res = sorted(res)
    x = []
    y = []
    for i in range(0, len(res)):
        x.append(res[i][0])
        y.append(res[i][1])
    a = np.min(np.array(x))
    b = np.max(np.array(x))
    pogr(x,y)

```

```

draw(1, x, y)
draw(5, x, y)
draw(20, x, y)
draw(50, x, y)

main()

```

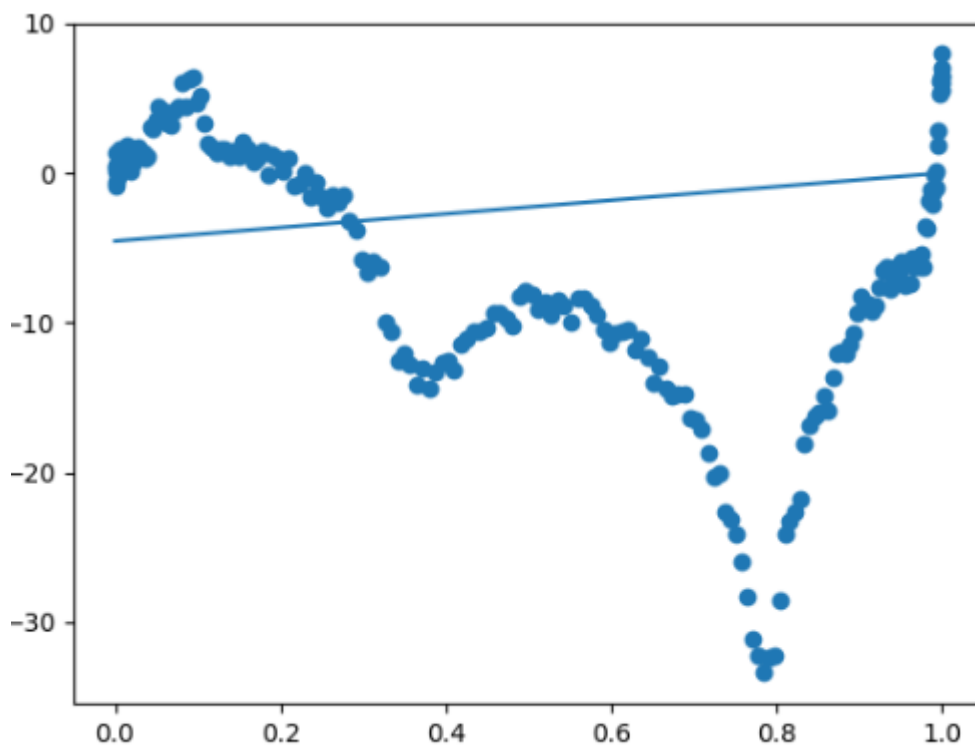
- `getBerns(step,i,x)` – вычисляет значение бернштейна(`step` – n для Бернштейна, i – i для Бернштейна, x – значение); бернштейн принимает значения от 0 до 1, поэтому нормализуем x
- `solve(step, x, y)` – находит коэффициенты для приближения(x и y – массивы входных данных)
- `pogr(x, y)` – строит график погрешностей для разных n с шагом 5
- `draw(step, x, y)` – рисует график приближения и начальные точки для $step$ многочленов и входных данных(x, y)

Графики:

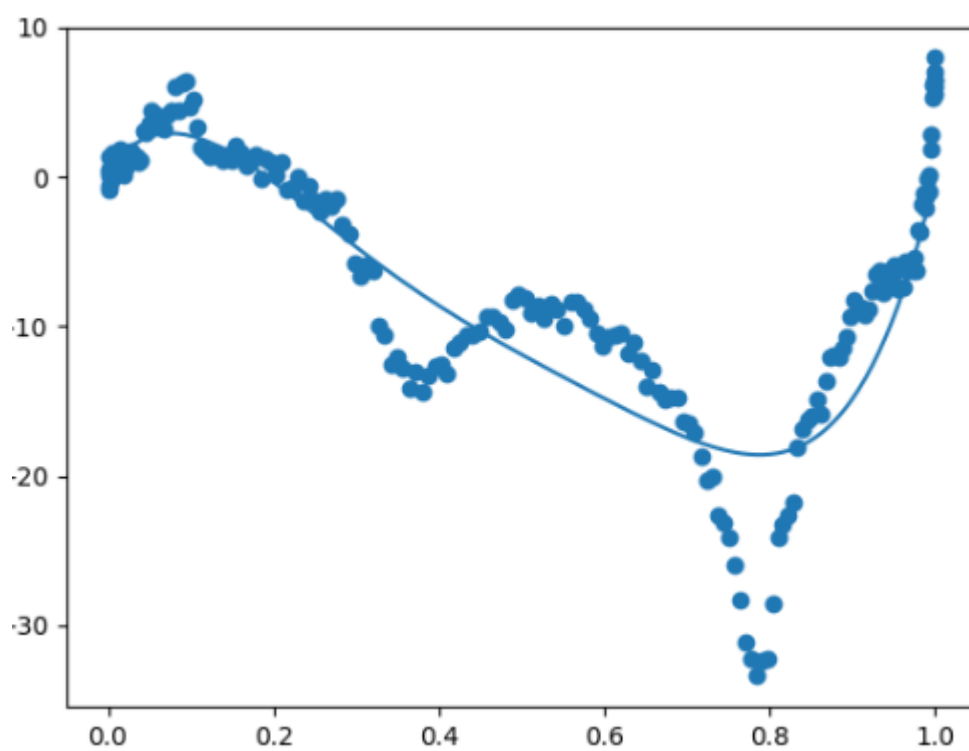
Точки - начальные x и y

График – построен по функции приближения для x

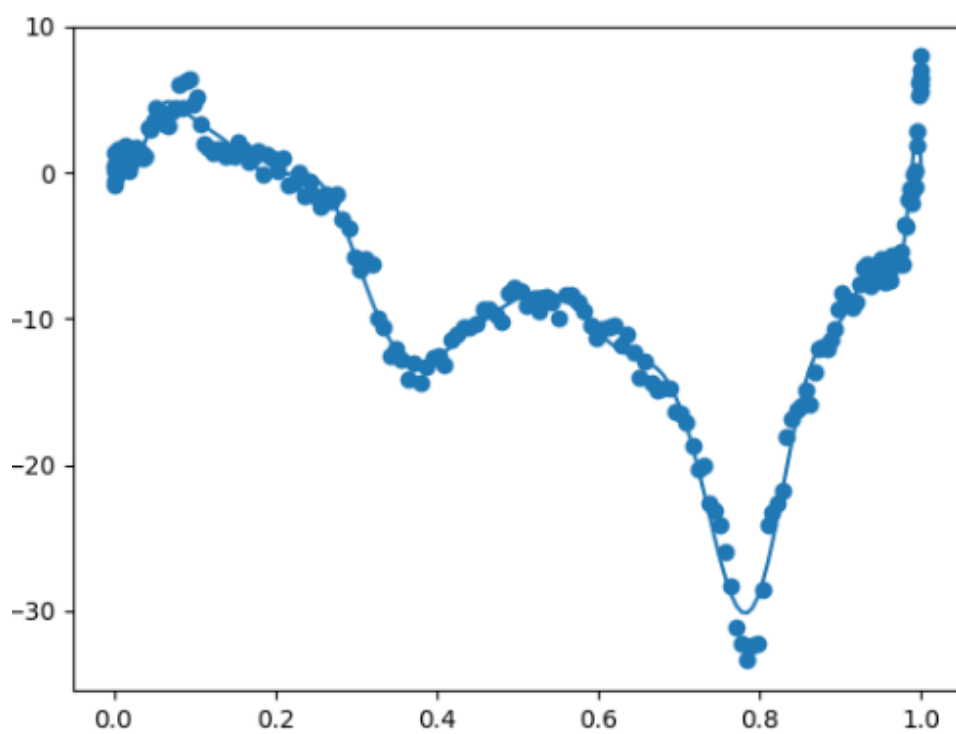
$n = 1$



$n = 5$



$n = 20$



$n = 50$

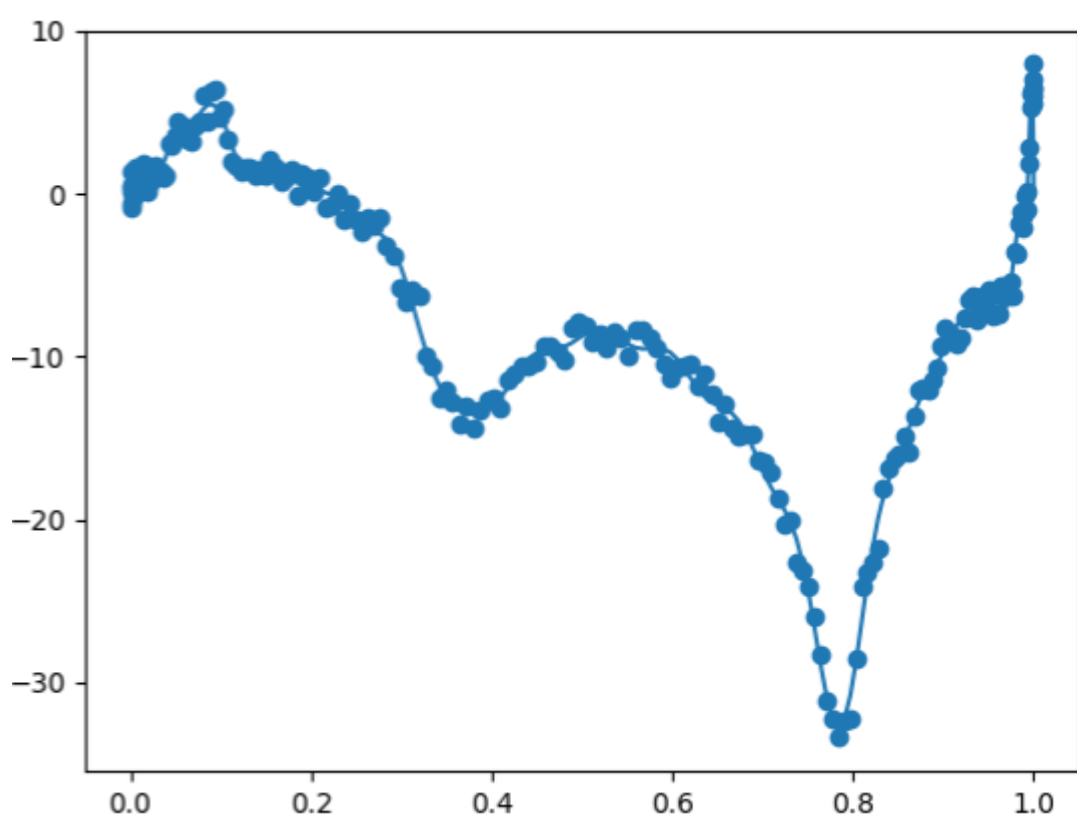
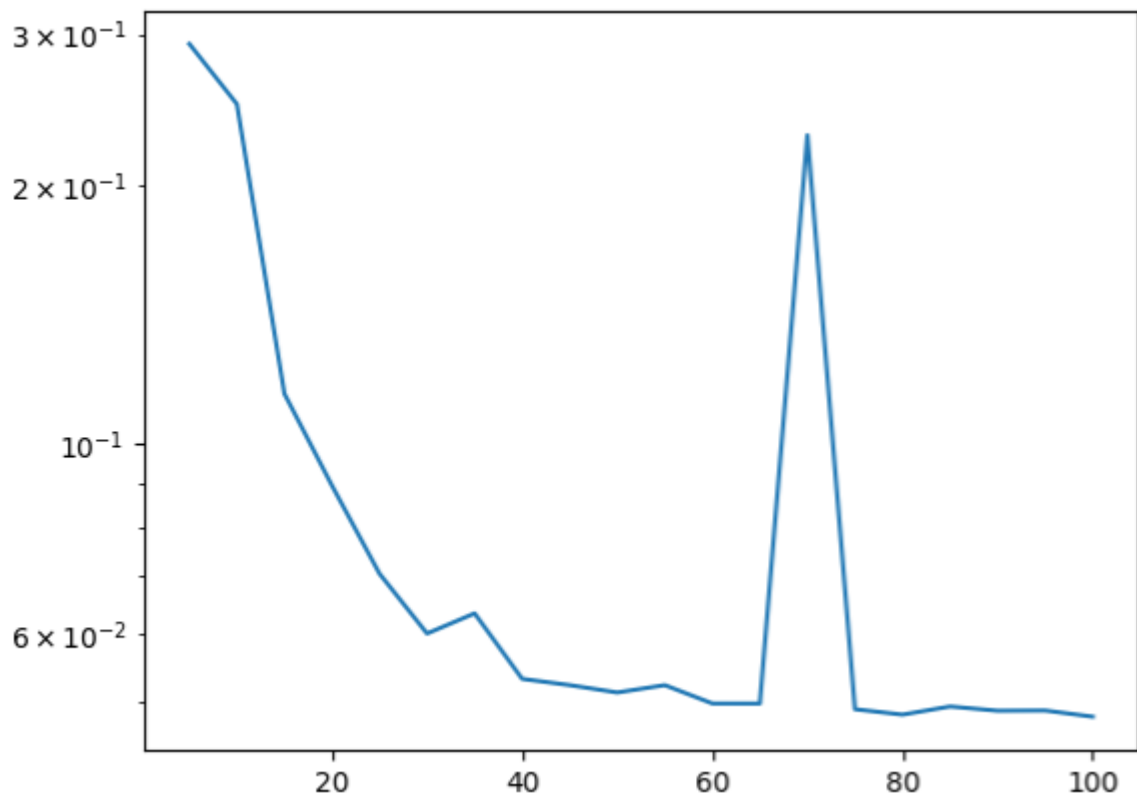


График среднеквадратичного отклонения.

Шаг равен 5.

Ось X – n

Ось Y - погрешность



С увеличением n график убывает(если смотреть глобально), но точки в районе $n = 70$ плохо приближают(резкий скачок погрешности), после $n = 80$ погрешность колеблется.