Лабораторная работа №

Ларин Егор. 4 группа 2 курс

22 марта 2022 г.

Теория

Корни многочлена Чебышева на отрезке $[a, b]$

$$x_k^{ch} = \frac{(a+b)}{2} + \frac{(b-a)}{2}\cos\left(\frac{2k+1}{2n+2}\pi\right), a = -2, b = 2$$

Равностоящие узлы

$$x_k = a + \frac{k(b-a)}{n}, a = -2, b = 2$$

Интерполяционный многочлен Ньютона

$P_n(x) = f(x_0)+(x-x_0)f(x_0;x_1)+(x-x_0)(x-x_1)f(x_0;x1;x_2)+\dots+(x-x_0)\dots(x-x_{n-1})f(x_0;\dots x_n)$,

где $f(x_0;\dots;x_k)$ – разделенная разность k-ого порядка.

Листинг кода

Многочлен представлен матрицей размера $n+1\times n$. Первый столбец содержит интерполяционные узлы. Второй столбец содержит значения интерполируемой функции в узлах. Оставшиеся столбцы содержат разделенные разности соответствующих порядков.

```
from math import exp, cos, pi
import matplotlib.pyplot as plt

a = -2
b =  2

f_1 = lambda x: exp(cos(x))
f_2 = lambda x: abs(x * abs(x) - 1)

x_ch = lambda k, n: (a+b)/2  + (b-a) / 2 * cos((2*k + 1)*pi / (2*n
x_ = lambda i, n: a + i*(b-a)/n
```

```python
def newton(x, y, n):
    xss = [x(i, n) for i in range(n)]
    yss = [y(xss[i]) for i in range(n)]
    coefficients = [xss, yss]
    for i in range(1, n):
        prev = coefficients[i]
        diff = []
        for j in range(0, n-i):
            d = (prev[j] - prev[j+1]) / (xss[j] - xss[j+i])
            diff.append(d)
        coefficients.append(diff)
    return coefficients

def calc(x, c):
    ans = 0
    factor = 1
    n = len(c[0])
    for i in range(n):
        ans = ans + c[i+1][0] * factor
        factor = factor * (x-c[0][i])
    return ans

def compare(f, n):
    c = newton(x_, f, n)
    c_ch = newton(x_ch, f, n)
    m = 100
    xss = [x_(i, m) for i in range(m)]
    print(n)
    print(f"Ordinary error: {max(f, c)}")
    print(f"Chebyshev error: {max(f, c_ch)}")
    print()

    plt.plot(
            xss, [calc(x,c) for x in xss], "b",
            xss, [calc(x,c_ch) for x in xss],  "r",
            xss, [f(x) for x in xss], "g",

            c_ch[0], [f(x) for x in c_ch[0]], "ro",
            c[0], [f(x) for x in c[0]], "bo")
    plt.show()

def max(f, c):
    n = 101
    m = abs(calc(x_(0, n), c)  - f(x_(0, n)))
```

```python
    for i in range(n):
        temp = abs(calc(x_(i, n), c)  - f(x_(i, n)))
        if temp > m:
            m = temp
    return m

print("f_1")
for i in [5, 10, 15, 20]:
    compare(f_1, i)

print("f_2")
for i in [5, 10, 15, 20]:
    compare(f_2, i)
```

Результаты вычислительного эксперимента

Рис. 1:

n	5	10	15	20
$\max\limits_{i=0,100}\lvert P_n(\bar{x}_i)-f_1(\bar{x}_i)\rvert$	0.10720127143956915	0.37370293297995616	0.0002514595332558933	0.0007981181921262737
$\max\limits_{i=0,100}\lvert P_n(\bar{x}_i)-f_1(\bar{x}_i)\rvert$	0.07862578176350299	0.09228630146680383	5.581088999839778e-06	1.2644319992927144e-05

n	5	10	15	20
$\max\limits_{i=0,100}\lvert P_n(\bar{x}_i)-f_2(\bar{x}_i)\rvert$	4.45392602321415	10.950849326348116	218.70809595875286	1276.8293403380942
$\max\limits_{i=0,100}\lvert P_n^{ch}(\bar{x}_i)-f_2(\bar{x}_i)\rvert$	1.82784050476298	2.75145530108653	2.11848506072225	0.1478943958439275

Выводы

Узлы Чебышева действительно являются оптимальными. При этом при интерполировании функций, производная которых не является непрерывной, возникает значительная ошибка, которая, справедливости ради, все еще меньше, чем при интерполировании по равностоящим узлам.

Графики

Зеленым обозначена исходная функция. Красным обозначен график полинома интерполяции, полученный при использовании узлов Чебышева. Зеленым обозначена исходная функция. Синим обозначен график полинома интерполяции, полученный при использовании равностоящих узлов.

Графики для f_1

Рис. 2: $n = 5$

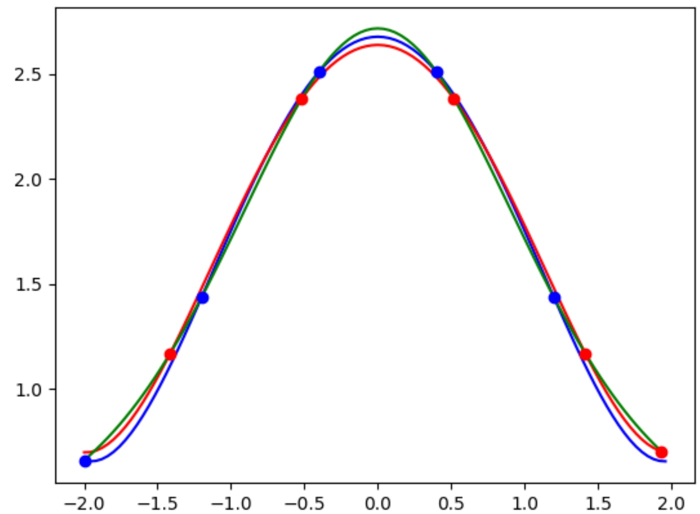


Рис. 3: $n = 10$

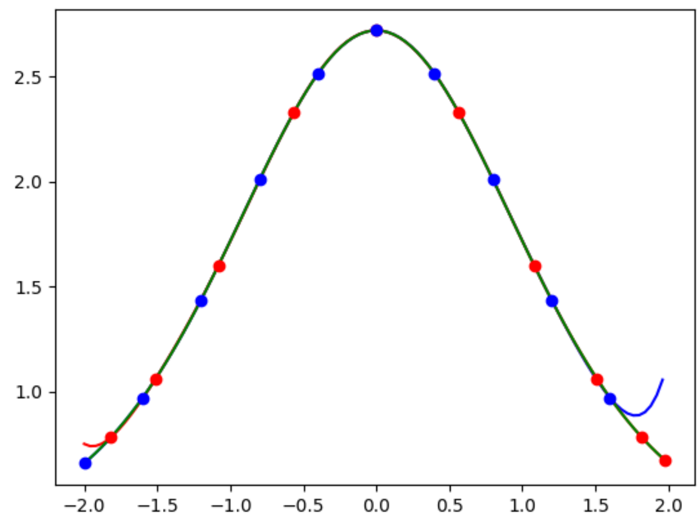


Рис. 4: $n = 15$

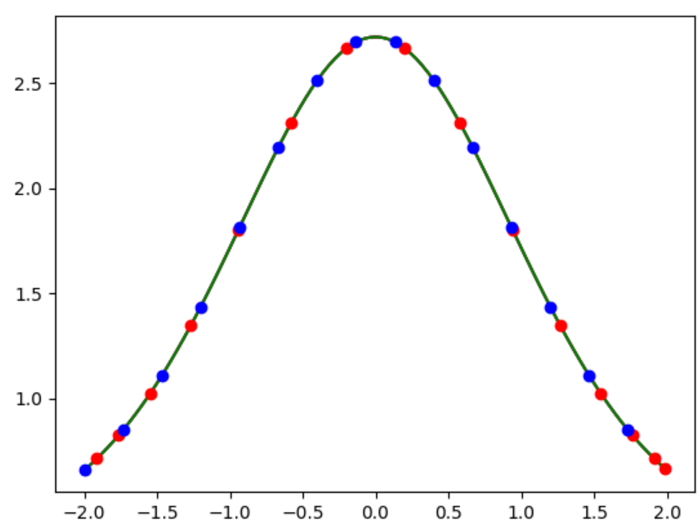
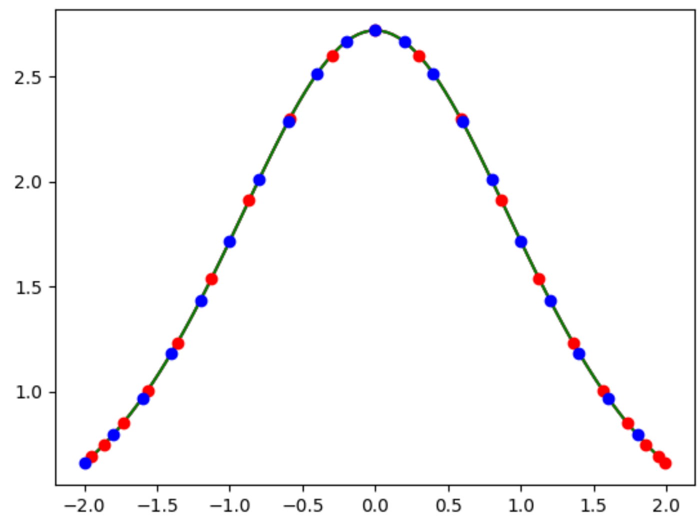


Рис. 5: $n = 20$



Графики для f_2

Рис. 6: $n = 5$

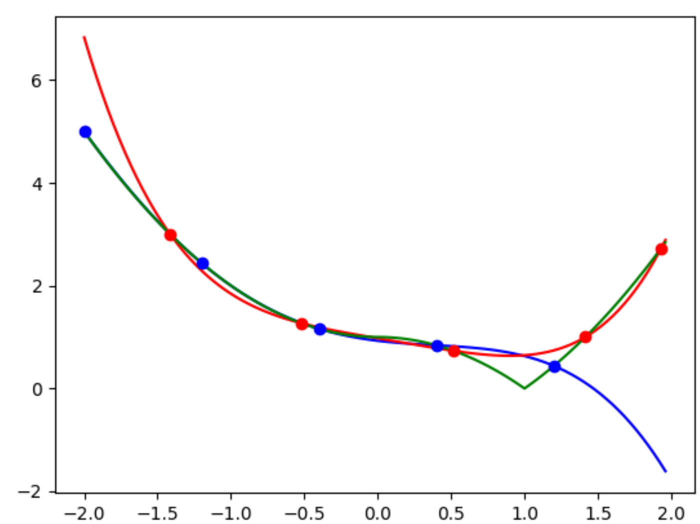


Рис. 7: $n = 10$

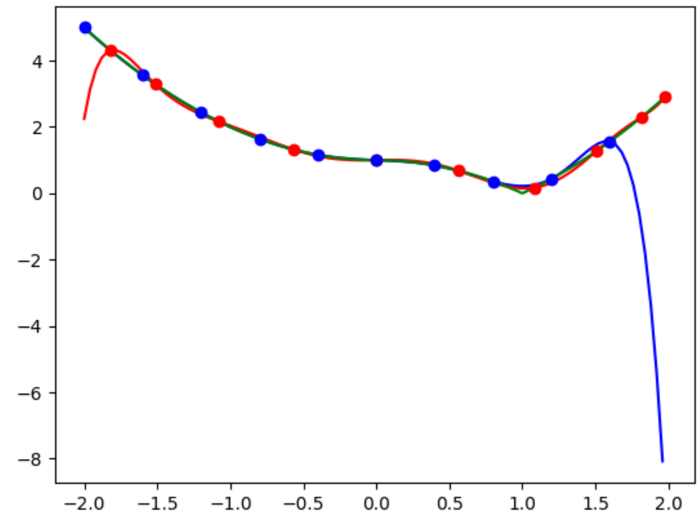


Рис. 8: $n = 15$

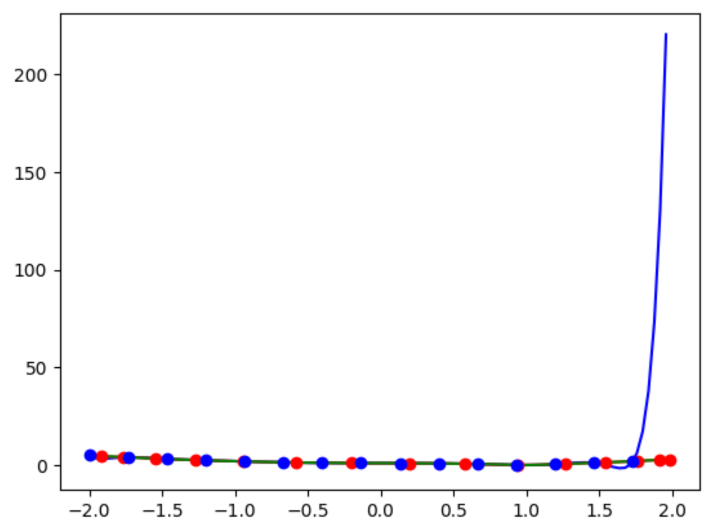


Рис. 9: $n = 20$