Отчет

Шилов Максим

Интерполяционный полином в форме Лагранжа

$$P_n(x) = \sum_{k=0}^{n} \frac{w(x)}{(x - x_k)w'(x_k)} f(x_k)$$

где $w(x) = (x - x_0)(x - x_1)\dots(x - x_n)$ - полином степени n+1 (n+1 узел)

Узлы Чебышева

Под узлами Чебышева понимают корни многочлена Чебышева первого рода. Они часто используются в качестве узлов при полиномиальной интерполяции, так как позволяют снизить влияние феномена Рунге.

Для натурального числа n узлы Чебышева на отрезке $[-1, 1]$ задаются формулой

$$x_k = \cos\left(\frac{2k-1}{2n}\pi\right), \quad k = 1, \dots, n.$$

Это корни многочлена Чебышева первого рода степени n. Для получения узлов на произвольном отрезке [a, b] можно применить аффинное преобразование отрезков:

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

Феномен Рунге

В численном анализе эффект нежелательных осцилляций (колебаний), возникающий при интерполяции полиномами высоких степеней.

Результаты численных экспериментов

Для наглядного примера возьмем функцию

$$f(x) = \frac{1}{1+x^2}, \quad [a,b] = [-10,10]$$

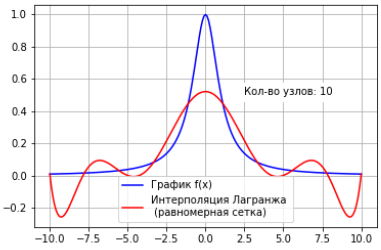
и точку близкую к границе $x = 9.9 \quad \Rightarrow \quad f(9.9) = 0.0100999$

1. Равномерная сетка [1]

а) 10 узлов:

$P_9(9.9)$ = -0.069398

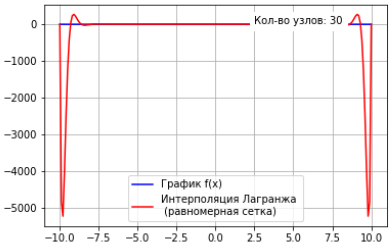
Реальная погрешность: $|P_9(9.9)\text{-}f(9.9)| = 0.079498$



б) 30 узлов:

$P_{29}(9.9)$ = -4849.297227

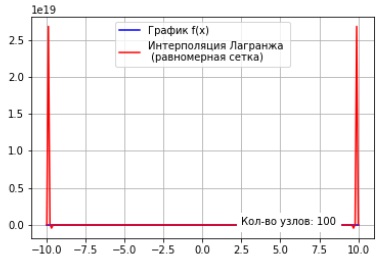
Реальная погрешность: $|P_{29}(9.9)\text{-}f(9.9)| = 4849.307327$



в) 100 узлов:

$P_{99}(9.9)$ = 2.6803e+19

Реальная погрешность: $|P_{99}(9.9)\text{-}f(9.9)| = 2.6803e+19$



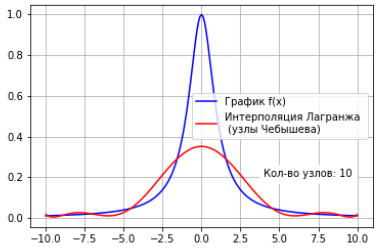
[1] На графиках погрешностей интерполяции на равномерной сетке отчетливо виден феномен Рунге.

Такой эффект роста уклонения при росте степени многочлена зависит как от выбираемой последовательности узлов, так и от интерполируемой функции. А именно, для любой последовательности узлов можно подобрать такую непрерывную функцию, что ошибка ее интерполяции по этим конкретным узлам будет неограниченно расти. С другой стороны, согласно аппроксимационной теореме Вейерштрасса, для любой непрерывной функции на отрезке можно подобрать последовательность полиномов, равномерно сходящихся к этой функции на отрезке. Это теоретически позволяет подобрать (для этой конкретной функции) последовательность узлов без феномена Рунге.

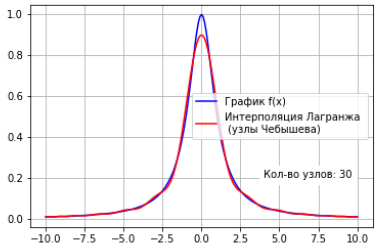
Компромиссом можно считать узлы Чебышёва, погрешность интерполяции по ним равномерно убывает для любой абсолютно непрерывной функции

2. Узлы Чебышева

а) 10 узлов:
$P_9(9.9) = 0.011114$
Реальная погрешность: $|P_9(9.9)\text{-}f(9.9)| = 0.001014$



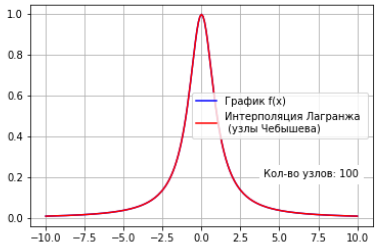
б) 30 узлов:
$P_{29}(9.9) = 0.0096468$
Реальная погрешность: $|P_{29}(9.9)\text{-}f(9.9)| = 0.000453$



в) 100 узлов:

$P_{99}(9.9) = 0.010100005$

Реальная погрешность: $|P_{99}(9.9)\text{-}f(9.9)| = 0.0000000156$



1 Интерполяционный полином в форме Лагранжа (равномерная сетка)

```
import numpy as np

def function(x):
    return 1/(1 + x**2)

x = 1

A = -10
B = 10

num_nods = 100
step = (B - A)/(num_nods-1)

nodes = np.arange(A, B + step, step)
point_function = function(nodes)

n = len(nodes)
l = np.ones(n)

interpolation_polynomial = 0

for i in range(1, n):
    for j in range(i):
        l[i] = l[i] * ((x - nodes[j]) / (nodes[i] - nodes[j]))

for i in range(1, n-1):
    for j in range(i, n-1):
        l[i] = l[i] * ((x - nodes[j + 1]) / (nodes[i] - nodes[j + 1]))

for i in range(n-1):
    l[0] = l[0] * ((x - nodes[i + 1]) / (nodes[0] - nodes[i + 1]))

for i in range(n):
    interpolation_polynomial += point_function[i] * l[i]

print(function(x))
print(interpolation_polynomial)
print('Реальная погрешность: ', np.abs(function(x) - interpolation_polynomial))
```

2 Интерполяционный полином в форме Лагранжа (узлы Чебышева)

```
import numpy as np

def function(x):
    return 1/(1 + x**2)

x = 1

A = -10
B = 10

num_nods = 100

k = np.arange(1, num_nods + 1)

nodes =  (A + B) / 2 + ((B - A) / 2) * np.cos((2*k - 1)*np.pi/(2 * len(k)))
point_function = function(nodes)

n = len(k)
l = np.ones(n)

interpolation_polynomial = 0

for i in range(1, n):
    for j in range(i):
        l[i] = l[i] * ((x - nodes[j]) / (nodes[i] - nodes[j]))

for i in range(1, n-1):
    for j in range(i, n-1):
        l[i] = l[i] * ((x - nodes[j + 1]) / (nodes[i] - nodes[j + 1]))

for i in range(n-1):
    l[0] = l[0] * ((x - nodes[i + 1]) / (nodes[0] - nodes[i + 1]))

for i in range(n):
    interpolation_polynomial += point_function[i] * l[i]

print(function(x))
print(interpolation_polynomial)
print('Реальная погрешность: ', np.abs(function(x) - interpolation_polynomial))
```