ChebyshevInterpolation

September 4, 2023

1 Chebyshev interpolation

The Chebyshev polynomials of the first kind are defined by

$$T_n(x) = \cos(n \cos^{-1} x), \quad -1 \leq x \leq 1$$

where n is a non-negative integer. In this form, it is not obvious that T_n are polynomials. You can prove by induction that

$$\cos(nx)$$

can be written as a polynomial in $\cos x$.

https://en.wikipedia.org/wiki/Chebyshev_polynomials

The idea behind Chebyshev interpolation is that one expands an arbitrary function $f(x)$ (although restricted to $-1 \leq x \leq 1$) as a truncated series of the for

$$f(x) \approx \sum_{n=0}^{N-1} c_n T_n(x).$$

To obtain c_n, we can use the roots of $T_N(x)$. Let

$$x_k = \cos\left(\pi \frac{2k+1}{2N}\right), \quad k = 0, 1, 2, \cdots, N-1,$$

where

$$T_N(x_k) = \cos\left(\pi \frac{2k+1}{2}\right) = 0.$$

The Chebyshev T polynomials obey the following orthogonality condition

$$\sum_{k=0}^{N-1} T_n(x_k) T_m(x_k) = \begin{cases} 0 & n \neq m \\ N & n = m = 0 \\ N/2 & n = m \neq 0 \end{cases},$$

where both n and m are less than N.

Hence

$$c_0 = \frac{1}{N} \sum_{k=0}^{N-1} f(x_k) T_0(x_k)$$

and

$$c_n = \frac{2}{N}\sum_{k=0}^{N-1} f(x_k)T_n(x_k)$$

In the example below we will interpolate Runge's function

$$f(x) = \frac{1}{1+25x^2}$$

using Lagrange's formula and Chebyshev interpolation. The former is unstable, while Chebyshev interpolation rapidly converges to very small errors.

[1]:
```
import numpy as np
import matplotlib.pyplot as plt
from numba import jit
```

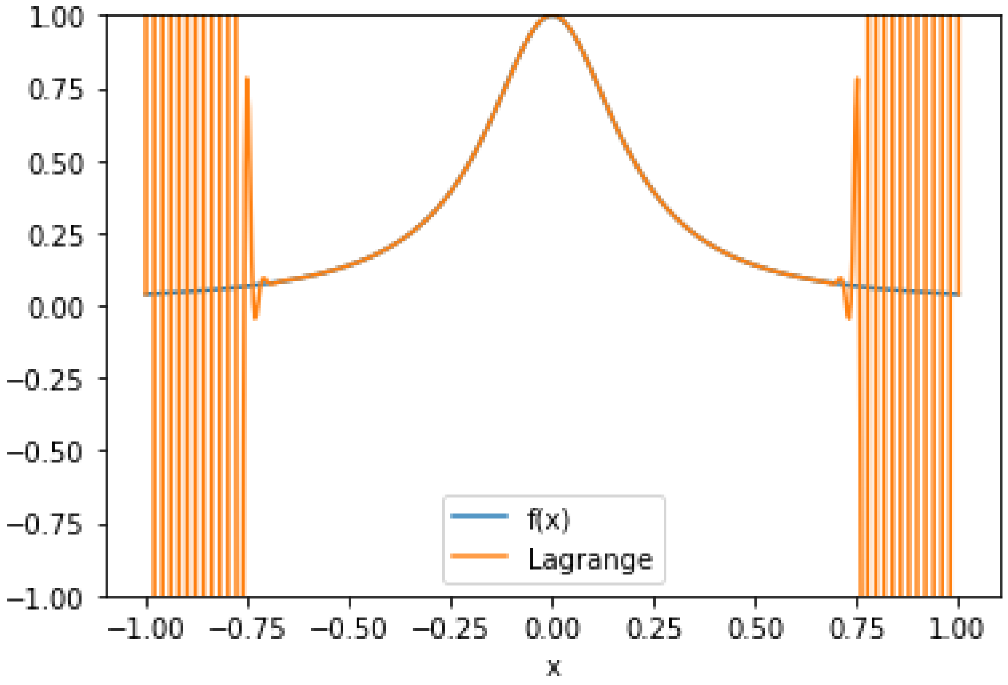
[2]:
```
@jit(nopython=True)
def lagrange_interp(x_val, x_data, y_data):
    assert len(x_data) == len(y_data)

    allresult = np.zeros_like(x_val)
    for k in range(len(x_val)):
        x = x_val[k]

        result = 0
        for i in range(len(y_data)):
            sub_result = y_data[i]
            for j in range(len(x_data)):
                if i == j:
                    continue
                sub_result *= (x - x_data[j]) / (x_data[i] - x_data[j])
            result += sub_result
        allresult[k] = result
    return allresult
```

[3]:
```
xarray = np.linspace(-1, 1, 100)
yarray = 1.0 / (1 + 25 * xarray ** 2)
```

[4]:
```
xvalues = np.linspace(-1, 1, 500)
yvalues = lagrange_interp(xvalues, xarray, yarray)
# yvalues = lagrange_interp(xarray, xarray, yarray)
```

[5]:
```
plt.plot(xarray, yarray, label="f(x)")
plt.plot(xvalues, yvalues, label="Lagrange")
plt.xlabel('x')
plt.ylim(-1, 1)
plt.legend()
```

[5]:
```
<matplotlib.legend.Legend at 0x7fd4ea7e33d0>
```



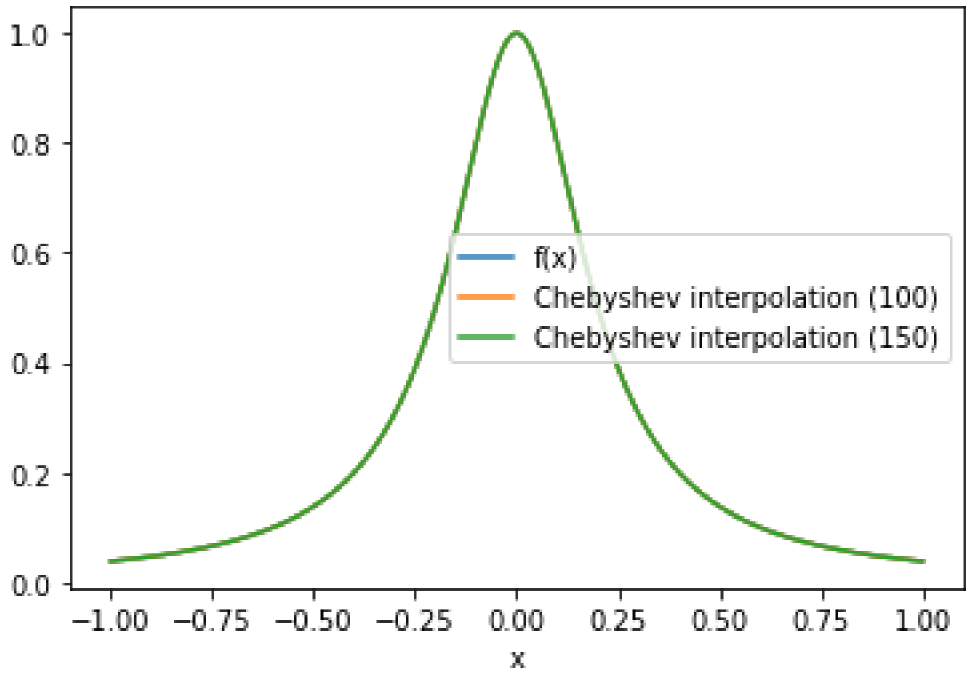
[6]:
```python
def ChebyshevT_roots(n):
    return np.array([np.cos(np.pi * (2 * k + 1) / 2 / n) for k in range(n)])


def ChebyshevT(n, x):
    return np.cos(n * np.arccos(x))


def ChebyshevT_coefficients(function, n):
    x_roots = ChebyshevT_roots(n)
    coefs = np.zeros(n)
    for xk in x_roots:
        cheby_values = np.array([ChebyshevT(i, xk) for i in range(n)])
        coefs += function(xk) * cheby_values
    coefs = 2 * coefs / n
    coefs[0] /= 2
    return coefs


def Evaluate_ChebyshevT_expansion_at(x, coeffs, n):
    cheby_values = np.array([ChebyshevT(i, x) for i in range(n)])
    return np.sum(cheby_values * coeffs)
```

[18]:
```
cfs_100 = ChebyshevT_coefficients(lambda x: 1.0 / (1 + 25 * x ** 2), 100)
cfs_150 = ChebyshevT_coefficients(lambda x: 1.0 / (1 + 25 * x ** 2), 150)
```

[19]:
```
xarray = np.linspace(-1,1, 1000)
vals_100 = np.array([Evaluate_ChebyshevT_expansion_at(x, cfs_100, 100) for x in␣
 ↪xarray])
vals_150 = np.array([Evaluate_ChebyshevT_expansion_at(x, cfs_150, 150) for x in␣
 ↪xarray])
```

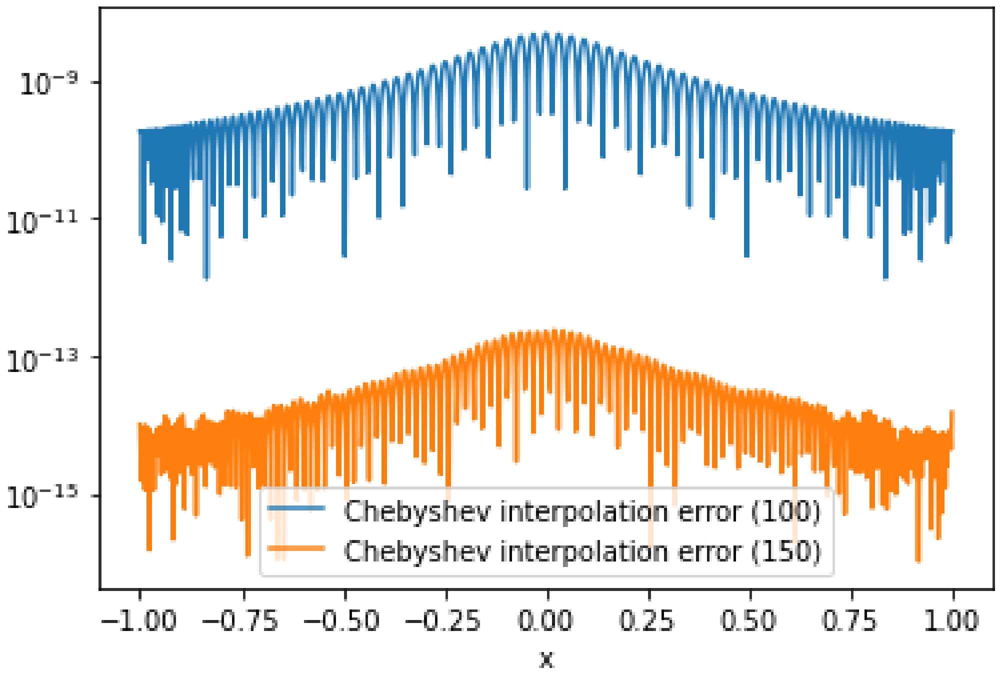
[20]:
```
plt.plot(xarray, 1 / (1+25 * xarray**2), label="f(x)")
plt.plot(xarray, vals_100, label="Chebyshev interpolation (100)")
plt.plot(xarray, vals_150, label="Chebyshev interpolation (150)")
plt.xlabel('x')
plt.legend()
```

[20]:
```
<matplotlib.legend.Legend at 0x7fd4ea427af0>
```



[21]:
```
plt.plot(xarray, np.abs(vals_100 - 1 / (1+25 * xarray**2)), label="Chebyshev␣
 ↪interpolation error (100)")
plt.plot(xarray, np.abs(vals_150 - 1 / (1+25 * xarray**2)), label="Chebyshev␣
 ↪interpolation error (150)")
plt.xlabel('x')
plt.yscale('log')
plt.legend()
```

```
[21]: <matplotlib.legend.Legend at 0x7fd4ea414f70>
```



10⁻⁹
10⁻¹¹
10⁻¹³
10⁻¹⁵
Chebyshev interpolation error (100)
Chebyshev interpolation error (150)
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
x

```
[ ]:
```