Barycentric Lagrange Interplation

As discussed by Jean-Paul Berrut and Lloyd N. Trefethen (2004)

Maximilian Jentzsch, Math 56 Final Project

Instructor: Dr. Alex Barnett

Lagrange interpolation

Given a set D_n of $n+1$ nodes x_n with corresponding values f_n, we aim to contruct the polynomial that satisfies

$$p(x_j) = f_j \; j = 0, \ldots, n$$

This data set can be interpolated by the Lagrange form of the interpolation polynomial [3]

$$p_{01\ldots n}(x) = \sum_{j=0}^{n} l_j(x) f(x_j), \quad (1)$$

where

$$l_j(x) = \frac{\prod_{k=0,\, k\neq j}^{n}(x - x_k)}{\prod_{k=0,\, k\neq j}^{n}(x_j - x_k)} \quad (2)$$

One must note that there are several issues with Lagrange's formula:

1. Each evaluation of $p(x)$ requires $O(n^2)$ additons and multiplications.
2. Adding a new data pair (x_{n+1}, f_{n+1}) requires an entirely new computation of every l_j.
3. The computation can be numerically unstable.

This poster will focus on points one and two.

Improved Lagrange Formula

Note that the numerator of (2) can be written as

$$l(x) = (x - x_0)(x - x_1)\ldots(x - x_n)$$

Define the *barycentric weights* by

$$w_j = \frac{1}{\prod_{k\neq j}(x_j - x_k)}, \; j = 0, \ldots, n \quad (3)$$

Now, l_j can be written as

$$l_j(x) = l(x) \sum_{j=0}^{n} \frac{w_j}{x - x_j} f_j$$

which yields the new form of (1):

$$p(x) = l(x) \sum_{j=0}^{n} \frac{w_j}{x - x_j} f_j \quad (4)$$

This improved formula now requires $O(n^2)$ floating point operations (flops) to calculate quantities independent of x. To evaluate p, only $O(n)$ flops are required.

In addition, this formula can now easily be updated with a new Data set (x_{n+1}, f_{n+1}) just by dividing each w_j from (3) by $(x_j - x_{n+1})$, and then computing w_j using (3). The addition of a new Data set therefore requires a mere $O(n)$ flops, instead of an entire recalculation of every l_j.

The Barycentric Formula

Equation (4) can still be written in an even nicer form. The interpolant of the constant function 1 is itself. Plugging 1 into (4) yields

$$1 = \sum_{j=0}^{n} l_j(x) = l(x) \sum_{j=0}^{n} \frac{w_j}{x - x_j}.$$

Dividing (4) by the interpolation of 1, $l(x)$ cancels and gives the so-called *barycentric formula* for p:

$$p(x) = \frac{\sum_{j=0}^{n} \frac{w_j}{x - x_j} f_j}{\sum_{j=0}^{n} \frac{w_j}{x - x_j}} \quad (5)$$

The barycentric formula is still a Lagrange formula where the weights w_j can be updated with a new data pair (x_{n+1}, f_{n+1}) using $O(n)$ flops. In addition, there exist explicit formulas for the barycentric weights w_j when using equidistant or Chebyshev nodes, facilitating their computation:

1. Equidistant: $w_j = (-1)^j \binom{n}{j}$
2. Chebyshev: $w_j = (-1)^j \delta_j$, where $\delta_j = \frac{1}{2}$ when $j = 0$ or $j = n$ and $\delta_j = 1$ otherwise.

Runge Phenomenon

The use of equidistant nodes in any type of interpolation poses one big problem. For large N, the different weights w_j vary by exponentially large factors [1]. This makes polynomial interpolation with equidistant points *ill-conditioned*. Figures 1 and 2 demonstrate this problem. At the "edges" of the interpolation, the approximations do not converge with increasing N. As can be seen in the Figures , the maximum error eventually grows exponentially. It can be shown that the error can grow as fast as 2^N [4]. This phenomenon, which is more extreme than the Gibbs phenomenon, is called the *Runge phenomenon*.

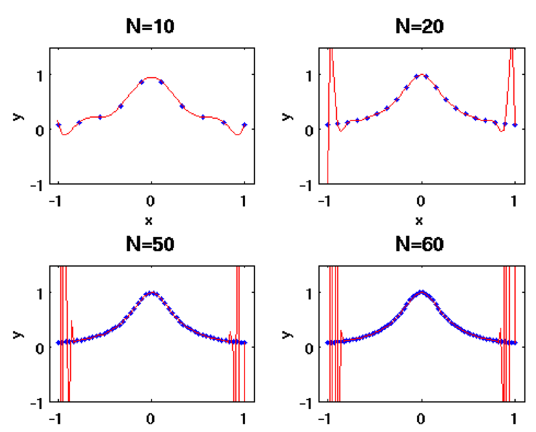

Figure 1: Barycentric Lagrange Interpolation with varying N

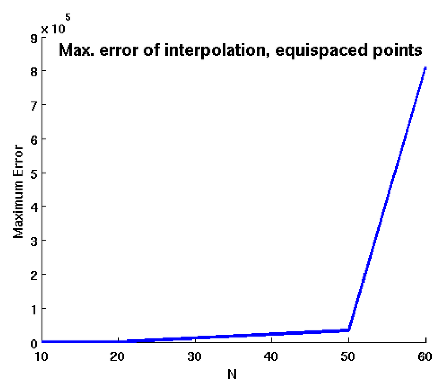

Figure 2: Maximum Error with increasing N

To overcome this problem, unevenly distributed interpolation points must be used. There are different types of such points, but here we shall focus on the most common and simplest kind, the *Chebyshev points*. Figure 1 shows that the interpolation works well in the center, but fails at the edges. Chebyshev points, which have a density $\sim \frac{N}{\pi\sqrt{1-x^2}}$ can fix this issue (See [4] for more information).

Convergence Rates of Smooth Functions

Let f be analytic on an inside an ellipse in the complex plane with foci [-1,1] and axis lenghts 2L and 2l. When Chebyshev points are used, the interpolant converges exponentially as $N \to \infty$ [4]. In addition, the interpolants p_N satisfy the error estimate

$$max_{x\epsilon[-1,1]} |f(x) - p_N(x)| \leq CK^{-N}$$

for some constants C and K¿1 [1, 2, p. 508, p. 173]. If the conditions above are satisfied, then $K = L + l$, and K denotes the convergence rate. Note that the convergence rate depends on the poles of *f*, and that a larger region of analyticity also results in a higher convergence rate [1]. Figure 3 shows the convergence of several functions with increasing N. We will also calculate their convergence rates:

1. $f(x) = \frac{exp(x)}{cos(x)}$: $K = \frac{\pi}{2} + \sqrt{\frac{\pi^2}{4} - 1} \approx 2.7822$.
   Note that there is a mistake in [1], where the term under the square root is $\pi^2 - 1$.
2. $f(x) = \frac{1}{1+12x^2}$: $K = \frac{1}{\sqrt{12}} + \sqrt{\frac{13}{12}} \approx 1.3295$
3. $f(x) = \frac{1}{1+50x^2}$: $K = \frac{1}{\sqrt{50}} + \sqrt{\frac{51}{50}} \approx 1.1514$
4. $f(x) = abs(x)$ is not analytic on [-1,1] because it is not differentiable at $x = 0$. The interpolation does not seem to converge (at least exponentially).
5. $f(x) = tan(x)$: we would expect this function to have a ver similar convergence rate as $\frac{exp(x)}{cos(x)}$, since it also has nodes at ± 1. Indeed, Figure 3 shows this!

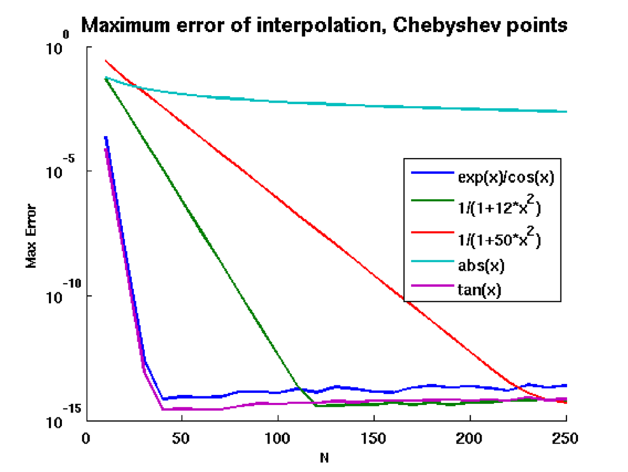

Figure 3: Convergence Rates of Different Functions

References

[1] Jean-Paul Berrut and Lloyd N. Trefethen. Barycentric Lagrange Interpolation. *SIAM REVIEW*, 46(3):501–517, 2004.

[2] Bengt Fornberg. *A Practical Guide to Pseudospectral Methods*. Cambridge University Press, 1996.

[3] Nabil R. Nassif and Dolly K. Fayyad. *Numerical Analysis and Scientific Computing*. CRC Press, 2014.

[4] Lloyd N. Trefethen. *Spectral Methods in Matlab*, chapter 5. Polynomial Interplation and Clustered Grids, pages 41–50. SIAM, 2000.