Explorations with the Barycentric Formula for Polynomial Interpolation

Dennis Pence

Westerm Michigan University, Kalamazoo, Michigan, USA

Abstract

We will explore graphing calculator and other implementations of the barycentric formula for polynomial interpolations. These are related to some of the `chebfun` package implementations of these ideas for Matlab provided by L. N. Trefethen. Interestingly, Peter Henrici provides some of this in his 1982 book, *Essentials of Numerical Analysis with Pocket Calculator Demonstrations,* but it seems to have been little noticed.

There are many ways to express polynomials. Some of the ways have advantages for theoretical work (say for interpolation), and some of the ways have advantages for computation, particularly on computers and calculators. I have been teaching for over 30 years that the *Lagrange form*

$$p(x) = \frac{y_0(x-x_1)(x-x_2)\cdots(x-x_n)}{(x_0-x_1)(x_0-x_2)\cdots(x_0-x_n)} + \frac{y_1(x-x_0)(x-x_2)\cdots(x-x_n)}{(x_1-x_0)(x_1-x_2)\cdots(x_1-x_n)} + \cdots + \frac{y_n(x-x_0)(x-x_1)\cdots(x-x_{n-1})}{(x_n-x_0)(x_n-x_1)\cdots(x_n-x_{n-1})}$$

is very nice for theoretical work with interpolation, but it does not seem all that practical for computer implementation. Further, following the example of Runge, I have also warned my student against the practice of trying polynomial interpolation with high degree polynomials. Runge's example using equally-spaced nodes is still standard in almost all numerical analysis textbooks. But several recent papers show the way, both to use the Lagrange form and to use high-degree polynomial interpolation for practical computation.

The first step is to consider the barycentric formula for this Lagrange form. Obviously we can precomute parts of the Lagrange form. Let the *weights* be

$$w_i = \frac{1}{(x_i-x_0)(x_i-x_1)\cdots(x_i-x_{i-1})(x_i-x_{i+1})\cdots(x_i-x_n)},\ i = 0, 1, \ldots, n$$

and

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

which is usually called the *nodal polynomial.* The *first barycentric formula* is then

$$p(x) = l(x)\sum_{i=0}^{n} \frac{w_i y_i}{(x-x_i)}.$$

Trefethen claims that this may go back to Jacobi's thesis. Notice that this (now rational expression) is technically not defined at the exact nodes x_i, but we already know the value of the polynomial here, namely the interpolation values y_i. Next we know that polynomial interpolants are unique, so the Lagrange form that interpolates the constant function 1 will give

$$1 = l(x)\sum_{i=0}^{n} \frac{w_i}{(x-x_i)}.$$

Dividing the last two formulas, we get the second barycentric formula

$$p(x) = \frac{\sum_{i=0}^{n} \frac{w_i y_i}{(x-x_i)}}{\sum_{i=0}^{n} \frac{w_i}{(x-x_i)}}.$$

This is the formual that we can nicely compute in a simple loop for any x, where we accumulate the sums in the numerator and denominator separately, and then divide after the loop is completed.

We will briefly review what is know about the weights for certain choices of nodes. Then we will consider implementations of these ideas in certain environments, including graphing calculators. This will be based on some of the ideas in the `chebfun` package for Matlab, given to us by L. N. Trefethen and his colleagues at Oxford University. Interestingly, these barycentric formulas can be found in the 1982 book, *Essentials of Numerical Analysis with Pocket Calculator Demonstrations,* by the Swiss mathematican Peter Henrici. Unfortunately this later book did not receive as much attention as his earlier *Elements of Numerical Analysis* where the barycentric formula does not appear. If you check on the *Mathematical Geneology Project*, you will find that I am one of the 270 decendants of Peter Henrici.