Barycentric Lagrange Interpolation

Brian Caravantes

Course: Math 4401 Numerical Methods

Faculty: Barry McQuarrie

University of Minnesota, Morris

Fall 2016

ABSTRACT

This paper will dive into what exactly is Barycentric Lagrange Interpolation and how the method works. The derivation of the Barycentric Lagrange Interpolation will be included. Additionally examples using the Barycentric method is applied and compared to that of other techniques learned in class such as the Lagrange Interpolation and Chebyshev Interpolation. Other factors will be discussed in this paper such as strengths and weaknesses of this method as well as possible improvements.

1. INTRODUCTION

Barycentric Lagrange Interpolation is a variation of that of Lagrange Polynomial interpolation, only that Barycentric Interpolation is a more reliable and quicker method. The following paper will describe why and how Barycentric Interpolation is a better counterpart to that of Lagrange Interpolation and a better option than that of Newton Interpolation when it comes to computations. After discussing how Barycentric Interpolation functions, the formula itself will be derived. Then the aspect of variation of nodes as well as its importance will be explained. Also an important aspect of any method is that of application, Barycentric interpolation is no different. Although there a multitude of applications, this paper focuses on applying it to functions and functions of data sets. This paper will touch upon other theoretical applications. Lastly this paper touches upon the strengths as well as weakness and possible improvements to that of Barycentric Lagrange Interpolation.

2. THE METHOD

2.1 Lagrange Interpolation

What will be discussed first is the method of Lagrange Interpolation to give some background as what how Barycentric Interpolation came to be. Lagrange interpolation is a method for dealing with polynomial interpolants. The interpolation starts by letting there exist n+1 distinct interpolation nodes (points) x_j where j = 0… n. The assumption can made that all nodes will be real. Also in the Lagrange Interpolation there will exist corresponding numbers that of f_j..Where f_j can be a data set or a function of a data set. To derive the Lagrangian formula there must exist a vector space $\prod_n$ such that it will help find a polynomial p Є $\prod_n$ that will interpolate f_j at the nodes where

$$p(x_j) = f_j, \qquad j = 0 \ldots n \text{ [6]}.$$

As we figured in class and as explained in the Numerical Analysis 2nd edition [4] the solution will be unique and will depend mostly on the data set. The Lagrangian form can then be written as

$$(1)\ p(x) = \sum_{j=0}^{n} f_j l_j(x), \qquad l_j(x) = \frac{\prod_{k=0,k\neq j}^{n}(x-x_k)}{\prod_{k=0,k\neq j}^{n}(x_j-x_k)}.$$

Where the Lagrange polynomial $l_j(x)$ within this interpolation has the parameters,

$$l_j(x_k) = \begin{cases} 1, & j = k \\ 0, & otherwise \end{cases} \qquad j, k = 0, \ldots, n \text{ [6].}$$

Now we will see how Newton's interpolation works and compare it to Lagrange Interpolation.

2.2 Newton's Method

The Lagrangian Interpolation method has some shortcomings that will be discussed later in this paper, but for now we can assume that Lagrange Interpolation is regarded as a theoretical tool rather than a computational one. For computations we will choose Newton interpolation or Newton's method of Divided Differences. To approach Newton's method one must first formulate Newton's Tableau of Divided differences. The figure below shows how one might compute the divided differences.

$$\begin{array}{cccc}
x_0 & f(x_0) & & \\
 & & \frac{f(x_1)-f(x_0)}{x_1-x_0} & \\
 & & & \frac{\frac{f(x_2)-f(x_1)}{x_2-x_1} - \frac{f(x_1)-f(x_0)}{x_1-x_0}}{x_2-x_0} \\
x_1 & f(x_1) & & \\
 & & \frac{f(x_2)-f(x_1)}{x_2-x_1} & \\
x_2 & f(x_2) & & \vdots \\
 & & \vdots & \\
\vdots & & & \vdots \\
 & & \vdots & \\
x_n & f(x_n) & &
\end{array}$$

Figure 1: Newton's Divided Differences Tableau [4]

After refining the figure into a recursive formula of Newton's method

$$(2) \quad f\left[x_j, x_{j+1}, \ldots x_{k-1,} x_k\right] = \frac{f[\, x_{j+1}, \ldots x_{k,}] - f[\, x_j, \ldots x_{k-1,}]}{x_k - x_j},$$

with the initial condition $f[x_j] = f_j$ in order for the tableau equation to work. After all the calculations have been achieved, then using what is known as the Newton Interpolation formula,

$$(3)\quad p(x) = f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) + \cdots + f[x_0, x_1, \dots, x_n](x - x_0)(x - x_1) \dots (x - x_{n-1}),$$

then for each x, one can solve the corresponding polynomial $p(x)$ [6].

2.3 Shortcomings of the Lagrange Interpolation

Since this paper wants to implement Barycentric Lagrange Interpolation as the method of choice for evaluating polynomial interpolants one must look at what is so wrong with Lagrange interpolation. One can look at its shortcomings in two manners, computational speed and reliability. Regarding computational speed, we will say that with each method listed previous, each has a period in which the process can be completed. This process period can be defined in Big O notation, where $O(n)$ operations means that the process was completed in O time depending in n time, while $O(n^2)$ is completed in n squared time. With this knowledge, we know that the regular Lagrange Interpolation is completed in $O(n^2)$ operations while Newton's method is done in $O(n)$ operations.

If we consider the fact that because Newton's method can be done in $O(n)$ operations then why can't standard Lagrange Interpolation formula? This is what motivates an improvement of the standard interpolation formula. Now in terms of reliability the Lagrange Interpolation method cannot be trusted because of the Runge Phenomenon it produces with larger degree polynomials [7][5]. This means that as n gets large there will huge oscillations toward the edges of the interval where interpolation occurs.

2.4 How Barycentric Lagrange Interpolation works

Now that we have the preliminary information behind Barycentric Interpolation we can define and derive Barycentric Interpolation.

Now let

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

where $l(x)$ isn't a Lagrange Polynomial but rather the numerator of the polynomial. To continue improving this formula, we define what will be known as the Barycentric weight as

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

where $w_j = \frac{1}{l'(x_j)}$. Taking a closer look at equation () we can see this is the denominator of the Lagrange Polynomial. Now manipulating $l(x)$ by dividing by $x - x_j$ and solving for $l_j(x)$, we obtain the Lagrange Polynomial to be,

$$l_j(x) = l(x)\frac{w_j}{x - x_j}.$$

Using equation (1) to evaluate *p(x)*, thus

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

making Lagrange Interpolation compute in $O(n)$ operations when the Barycentric weights i.e. the numbers independent of x are known. It is important to note that if these numbers w_j are not identified beforehand than the Lagrange Formula continues to process in $O(n^2)$ operations when solving for $p(x)$ [1] [7]. The equation above is sometimes known as the first Barycentric Formula or the Modified Lagrange Formula. Equation (4) is not to be confused with a variation of the true form of the Barycentric Formula but rather a prerequisite equation needed to derive it. This equation thus can be changed and derived more finely to achieve another equation.

Suppose we interpolate on some data set $f_{j\,,}$ where we our goal is to achieve some sort of interpolation where the interpolant returns itself. This can be achieved by using the constant function of 1 as f_j such that

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

Now we can divided the equation attained as the Modified Lagrange Formula and divide it by the constant function, then cancelling $l(x)$ we achieve the Barycentric Formula,

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

Next we will see how the Barycentric Lagrange Interpolation changes depending on the certain nodes chosen when interpolating.

3. VARIATION OF NODES

Using the Barycentric formula with different node points will give explicit formulas for the Barycentric Weights w_j these explicit formulas all derive out to be

$$(6)\quad w_j = (-1)^j \binom{n}{j},$$

where the $\binom{n}{j}$ is the binomial coefficient that expands as $\frac{n!}{(n-j)!\,j!}$. To get to this we will start the derivation by choosing equidistant nodes on the interval [-1, 1] with spacing of h = 2/n where we can manipulate the weight to be

$$(w_j = \frac{(-1)^{n-j}\binom{n}{j}}{(h^n n!)}.$$

Then canceling j terms we get equation (6). Now if we want to get an equation for any interval

[a, b] then computation would take place by multiplying by $2^n(b-a)^{-n}$ where after simplification equation (6) is returned. Now we must consider the fact that like with any polynomial interpolation, unless the degree of polynomial is low then using equidistant nodes can cause great uncertainty as mentioned before with Lagrangian Interpolation. In the case of Barycentric Interpolation, the weight will vary exponentially by a factor of 2^n when n is large. The effect this produces is that data near the edges of interpolation interval will have huge oscillations (Runge Phenomenon). One way to fix the oscillation like with any other polynomial interpolation one should use sets of clustered nodes such as Chebyshev points [6]. The derivation of changing the Barycentric weight in terms of Chebyshev points is derived in [7] and [6], where the weight w_j turns out to be

$$(7)\quad w_j = (-1)^j \delta_j,\ \delta_j = \begin{cases} \frac{1}{2} & j = 0\ or\ j = n \\ 1 & otherwise \end{cases}.$$

Thus to get the most from Barycentric Lagrange Interpolation one must use clustered points like that of Chebyshev points. We will see how the variation of nodes takes a role in Barycentric Lagrange Interpolation when applying to data sets and functions.

4. APPLICATIONS

This Section will touch upon different examples analyzed using Barycentric Lagrange Interpolation and compared to that of other interpolation methods. For most examples listed the result, process, and explanation of solving them can be found in Appendix A. While the examples will be listed below. The examples come from either [2] or [3].

3.3 Polynomial use with Chebyshev Points

The data function and question come from [3] and [2].

We will analyze the function $f(x)$ on the interval such that

$$f(x) = \frac{Cos(x)*e^{x}}{1+5x^{2}},$$

where we will compare Lagrange Interpolation, Chebyshev Interpolation and Barycentric Interpolation as well as comparing their errors, $|f(x) - P(x)|$. This is one of the most significant examples because when compared to Barycentric Lagrange Interpolation not only beats Lagrange Interpolation but also Chebyshev Interpolation when comparing errors.

3.2 Data Sets

Data Set 1

The data and question for this data set come from [3] and [2].

Year	1940	1950	1960	1970	1980	1990
Population (Thousands)	132,165	151,326	179,323	203,302	226,542	249,633

Table 1:

Find the 5th degree Lagrange Polynomial fitting this data. We will analyze this data and compare using the Barycentric Lagrange Interpolation.

Data Set 2

The data and question for this data set come from [3] and [2].

Temp[k]	50	100	150	200	250	300	500	800	1200
Adiabatic Compressibility$[TPA]^{-1}$	7.12	7.19	7.27	7.34	7.44	7.54	7.95	8.49	8.89

Table 2:

Find the 8th degree Lagrange Polynomial fitting this data. We will analyze this data and compare using the Barycentric Lagrange Interpolation.

Data Set 3 Oscillations (wiggles)

The data and question for this data set come from [2].

We are given a data set with significant oscillations inside the region we want to interpolate and will fix the oscillations using Barycentric Lagrange Interpolation.

3.3 Continuous function with Equidistant Nodes

The data function comes from [2]. This example started by using a module in Mathematica for Langrangian Interpolation of equally spaced nodes where oscillations again occur. We will see that with Barycentric Lagrange Interpolation, the oscillations diminish.

3.4 Other Applications

Barycentric Lagrange Interpolation not only can be applied to computation but additionally used with theoretical tools such as Differentiation of Polynomial Interpolants, Rational Interpolation and Fast Multipole Methods. For more detailed explanation of the use of Barycentric Interpolation in these methods as well as other see [7].

5. CONCLUSION

Moreover Barycentric Lagrange Interpolation is the reliable and quick tool for approximating polynomial interpolants. The Barycentric formula is derived from that of Lagrange Interpolation and builds upon it to improve the method. Barycentric Lagrange Interpolation is in terms of computational speed, and operations faster than that of its counter parts. Barycentric Interpolation computes in $O(n)$ operations. This speed is quicker than that of Lagrangian Interpolation and just as fast as Newton Interpolation, if not faster when computing $p(x)$. Additionally Barycentric is more reliable as seen in Applications section and Appendix A where it approximates a better polynomial with a lower error when compared to Chebyshev Interpolation. Although Barycentric Interpolation is a highly reliable interpolating tool it does have its weaknesses such as when $x = x_j$ where an indeterminate form appears which nothing can really be done about it. The only fix is that since most interpolation is done with computer programs, an “if” statement should be included to catch this problem and just return the data. Since Barycentric Lagrange Interpolation should be the method of choice the only improvement that should be considered has already been iterated. For improvement sets of clustered nodes should be used with Barycentric Interpolation for best results. Furthermore Barycentric Lagrange Interpolation is the method to choose when it comes to approximating polynomial interpolants.

REFERENCES

1. Higham, Nicholas. J. "The Numerical Stability of Barycentric Lagrange Interpolation." *IMA Journal of Numerical Analysis (2004) 24, 547–556* (2004): n. pag. *Http://www.maths.manchester.ac.uk/~higham/narep/narep440.pdf*. IMA Journal of Numerical Analysis. Web. 11 Nov. 2016.
2. McQuarrie, Barry. LectureLagrangeInterpolation, LectureChebyshevInterpolation, Assignment3 Solutions. Mathematica. Sept. 2016.
3. McQuarrie, Barry. Numerical Methods Assignment3, PDF. Sept. 2016.
4. "Newton Polynomial." *Wikipedia*. Wikimedia Foundation, n.d. Web. 20 Nov. 2016.
5. " Sauer, Timothy. "Chapter 3:Data and Interpolating Functions." *Numerical Analysis*. 2nd ed. Boston: Pearson, 2012. 139-66. Print.
6. Trefethen, Lloyd N. *PDF*. N.p.: n.p., n.d. Approximation Theory and Approximation Practice. Web. 11 Nov. 2016.
7. Trefethen, Lloyd N., and Jean-Paul Berrut. *PDF*. N.p.: 2004 Society for Industrial and Applied Mathematics, 2004. Web. 11 Nov. 2016.