IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 1 Issue 10, December 2014.
www.ijiset.com

ISSN 2348 – 7968

Analysis of Lagrange Interpolation Formula

Vijay Dahiya

[1]Department of MathematicsMaharaja Surajmal Institute

Abstract: This work presents a theoretical analysis of Lagrange Interpolation Formula. In order to analyze the method, power series, basis function and quadratic interpolation using basis function and cubic interpolation are chosen. Also to check the performance of the considered method an error associated with Lagrange interpolation has considered. Errors are analyzed by comparing the actual sampled values with the values obtained by Lagrange's interpolation formula.

Keywords: Interpolation, cubic interpolation, basis function quadratic interpolation, unit step function.

1. Introduction:

From very ancient time interpolation is being used for various purposes. Sir Edmund Whittaker, a professor of numerical mathematics at the University of Edinburgh from 1913 to 1923, observed "The most common form of interpolation occurs when we seek data from a table which does not have the exact values we want". Liu Zhuo used the equivalent of second order Gregory-Newton interpolation to construct an "Imperial Standard Calendar". In 625 AD, Indian astronomer and mathematician Brahmagupta introduced a method for second order interpolation of the sine function and, later on, a method for interpolation of unequal-interval data. Numerous researchers study the possibility of interpolation based on the Fourier transformer, the Hartley transformer and the discrete cosine transform. In 1983, Parker et al. published a first comparison of interpolation technique in medical image processing. They failed, however to implement cubic B-spline interpolation correctly and arrive at erroneous conclusion concerning this technique[1]. In present days, several algorithms are used for image resizing [2] based on Lagrange's Interpolation Formula[3].

In this paper, Lagrange's interpolation formula is used for reconstructing power series fitting to analyze its performance. This paper is organized as follows: in section 2, we explain the mathematical principal of Lagrange's Interpolation method. The implementation is devised in section 3. In section 4 the experimental results are given. The conclusion is summarized in section 5.

2. Theory of Interpolation and considered functions

2.1 Lagrange Interpolation:

We consider the problem of approximating a given function by a class of simpler functions, mainly polynomials. There are two main uses of interpolation or interpolating polynomials. The first use is reconstructing the function f(x) when it is not given explicitly and only the values of f(x) and/or its certain order derivatives at a set of points, called nodes, tabular points or arguments are known. The second use is to replace the function f(x) by an interpolating polynomial p(x) so that many common operations such as determination of roots, differentiation and integration etc. which are intended for the function f(x) may be performed using p(x). A polynomial p(x) is called an interpolating polynomial if the value of p(x) and/or its certain order derivatives coincides with those of f(x) and/or its same order derivatives at one or more tabular points. In general, if there are N+1 distinct points $a \le x_0 < x_1 < x_2 < x_3 < ... < x_n \le b,$ then the problem of interpolation is to obtain p(x) satisfying the conditions

$$p(x_i) = f(x_i), i = 1, 2, ..., n \tag{1}$$

Substituting the conditions, we obtain the system of equations

$$\begin{array}{l} a_0 + a_1 x_0 + a_2 x_0^2 + ... + a_n x_0^n = f(x_0) \\ a_0 + a_1 x_1 + a_2 x_1^2 + ... + a_n x_1^n = f(x_1) \\ ------------------ \\ ------------------ \\ a_0 + a_1 x_n + a_2 x_n^2 + ... + a_n x_n^n = f(x_n) \end{array} \tag{2}$$

This system of equation has a unique solution. The interpolation points or nodes are given as:

$$\begin{array}{ll} x_0 & f(x_0) = f_0 \\ x_1 & f(x_1) = f_1 \\ -------- & \\ -------- & \\ x_N & f(x_N) = f_N \end{array} \tag{3}$$

There exists only N^{th} degree polynomial that passes through a given set of N+1 points. Its form is expressed as a power series:

$$g(x) = a_0 + a_1 x + a_2 x^2 + ... + a_N x^N \tag{4}$$

Where a_i = unknown coefficients, i = 1,2,3,…,N.

It does not matter how we define the N^{th} degree polynomial whether by Fitting power series, Lagrange Interpolation functions or by Newton forward or backward interpolation. The resulting polynomial will always be the same.

2.2 Power Series Fitting to Define Lagrange Interpolation:

g(x) must match f(x) at the selected data points.

$$\begin{aligned} g(x_0) = f_0 &\Rightarrow a_0 + a_1 x_0 + a_2 x_0^2 + ... + a_n x_0^n = f(x_0) \\ g(x_1) = f_1 &\Rightarrow a_0 + a_1 x_1 + a_2 x_1^2 + ... + a_n x_1^n = f(x_1) \\ &------------------ \\ &------------------ \\ g(x_N) = f_N &\Rightarrow a_0 + a_1 x_n + a_2 x_n^2 + ... + a_n x_n^n = f(x_n) \end{aligned} \tag{5}$$

Solve the simultaneous equation we get:

$$\begin{bmatrix} 1 & x_0 & x_0^2 & . & . & . & x_0^N \\ 1 & x_1 & x_1^2 & . & . & . & x_1^N \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ 1 & x_N & x_N^2 & . & . & . & x_N^N \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ . \\ . \\ a_N \end{bmatrix} = \begin{bmatrix} f_0 \\ f_1 \\ . \\ . \\ f_N \end{bmatrix} \tag{6}$$

It is relatively computationally costly to solve the coefficients of the interpolating function g(x).

2.3 Lagrange Interpolation Using Basis Functions:

We note that in general $g(x_i) = f_i$.

Let $$g(x) = \sum_{i=0}^{N} f_i v_i(x) \tag{7}$$

ISSN 2348 – 7968

Where $v_i(x)$ = polynomial of degree N associated with each node *i* such that

$$v_i(x_j) = \begin{pmatrix} 0 & i \neq j \\ 1 & i = j \end{pmatrix} \tag{8}$$

For example we have 5 interpolation points or nodes, then

$$g(x_3) = f_0 v_0(x_3) + f_1 v_1(x_3) + f_2 v_2(x_3) + f_3 v_3(x_3) + f_4 v_4(x_3) \tag{9}$$

Using the definition for $v_i(x_j)$, we have $g(x_3) = f_3$.

i.e the sum of polynomial of degree N is also a polynomial of degree N. g(x) is equivalent to fitting the power series and computing coefficient $a_0, a_1, a_2, ..., a_N$.

3. Analysis and Implementation

3.1 Lagrange Linear Interpolation Using Basis Functions:

Linear Lagrange (N = 1) is the simplest form of Lagrange interpolation:

$$g(x) = \sum_{i=0}^{1} f_i v_i(x)$$

That is $g(x) = f_0 v_0(x) + f_1 v_1(x)$ (10)

Where $v_0(x) = \frac{(x - x_1)}{(x_0 - x_1)}$ and $v_1(x) = \frac{(x - x_0)}{(x_1 - x_0)}$.

Example: Given the following data:

$x_0 = 2$, $f_0 = 1.5$

$x_1 = 5$, $f_1 = 4.0$

find the linear interpolating function g(x).

Solution: Lagrange basis functions are:

$$v_0(x) = \frac{(5 - x)}{3} \quad \text{and} \quad v_1(x) = \frac{(x - 2)}{3}$$

The interpolating function $g(x) = 1.5 v_0(x) + 4.0 v_1(x)$.

ISSN 2348 – 7968

3.2 Lagrange Quadratic Interpolation Using Basis Functions:

For Quadratic Lagrange interpolation, N = 2

$$g(x) = \sum_{i=0}^{2} f_i v_i(x)$$

That is $g(x) = f_0 v_0(x) + f_1 v_1(x) + f_2 v_2(x)$ (11)

$$v_0(x) = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)},$$

Where $v_1(x) = \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)},$.

$$v_2(x) = \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}$$

Note that location of the roots of $v_0(x), v_1(x)$ and $v_2(x)$ are defined such that the basic premise of interpolation is satisfied, namely that $g(x_i) = f_i$.

4. Errors Associated with Lagrange Interpolation

Using Taylor series analysis, the error can be shown to be given by:

$$E(x) = f(x) - g(x)$$

$$E(x) = L(x) f^{(N+1)}(\xi),\ x_0 \le \xi \le x_N \quad (12)$$

Where $f^{(N+1)}(\xi) = N+1$ derivatives of *f* w.r.t. *x* evaluated at ξ.

and $L(x) = \frac{(x-x_0)(x-x_1)\dots(x-x_N)}{(N+1)!}$ = an N+1th degree polynomial.

Now if f(x) is a polynomial of degree $M \le N$, then

$$f^{(N+1)}(x) = 0 \Rightarrow E(x) = 0 \text{ for all } x.$$

Therefore *g(x)* will be exact representation of *f(x)*.

5. Conclusion

In this paper, according to the analysis the performance of Lagrange interpolation formula on different types of function is presented. Experimental results show that it works better for a function whose value will increase or remain constant with the independent variable.

6. References

[1]. Meijering, Erik. “A Chronology of Interpolation: From Ancient Astronomy to Modern Signal and Image Processing.” Proceedings of the IEEE. vol. 90, no. 3, pp. 319-42. March 2002

[2]. Jianping Xiao, Xuecheng Zou, Zhenglin Liu, Xu Guo, "Adaptive Interpolation Algorithm for Real-time Image Resizing," icicic, vol. 2, pp.221-224, First International Conference on Innovative Computing, Information and Control - Volume II (ICICIC'06), 2006

[3]. Numerical Methods for Engineers, by Steven C. Chapra and Raymond P. Canale, Forth Edition, Chapter 18, Page no: 474.