Note on Equidistant Polynomial Interpolation

Gao Shang*
School of Computer Science and Engineering
Jiangsu University of Science and Technology
Zhenjiang, China
e-mail: gao_shang@just.edu.cn

Qian qiang
School of Computer Science and Engineering
Jiangsu University of Science and Technology
Zhenjiang, China
e-mail: qianqiang_just@163.com

Abstract—In the mathematical field of numerical analysis, interpolation is a method of constructing new data points within the range of a discrete set of known data points. Based on analysis of basic polynomial interpolation, the equidistant polynomial interpolation problem is studied. Simple divided difference is given and it is proved by mathematical induction. The computation is smaller than the traditional method. At last, this calculation method is illustrated through an example.

Keywords-polynomial interpolation; equidistant interpolation; lagrange interpolation; newton interpolation

I. INTRODUCTION

In numerical analysis, polynomial interpolation is the interpolation of a given data set by a polynomial: given some points, find a polynomial which goes exactly through these points[1-3]. Polynomials [4-10] can be used to approximate more complicated curves, for example, the shapes of letters in typography, given a few points. A relevant application is the evaluation of the natural logarithm and trigonometric functions: pick a few known data points, create a lookup table, and interpolate between those data points. This results in significantly faster computations. Polynomial interpolation also forms the basis for algorithms in numerical quadrature and computations. Polynomial interpolation is studied in this paper.

II. GENERAL METHODS TO SOLVE POLYNOMIAL INTERPOLATION

A. Linear System Method

Definition: Given a set of \( n + 1 \) data points \((x_i, y_i)\) where no two \( x_i \) are the same, one is looking for a polynomial \( P_n(x) \) of degree at most \( n \) with the property

\[
P_n(x) = y_i, \quad i = 0, 1, 2, \ldots, n.
\]

Suppose that the interpolation polynomial is in the form

\[
P_n(x) = a_n + a_{n-1}x + \cdots + a_1x^{n-1} + a_0x^n
\]  

(1)

The statement that \( p \) interpolates the data points means that \( P_n(x_i) = y_i \), \( i = 0, 1, 2, \ldots, n \)

If we substitute equation (1) in here, we get a system of linear equations in the coefficients \( a_k \):

\[
\begin{align*}
a_0 + a_1x_0 + a_2x_0^2 + \cdots + a_nx_0^n &= y_0 \\
a_0 + a_1x_1 + a_2x_1^2 + \cdots + a_nx_1^n &= y_1 \\
&\quad \vdots \\
a_0 + a_1x_n + a_2x_n^2 + \cdots + a_nx_n^n &= y_n
\end{align*}
\]

(2)

We have to solve this system for \( a_k \) to construct the interpolation \( P_n(x) \).

B. Lagrange’s Interpolation Formula

In numerical analysis, Lagrange polynomials are used for polynomial interpolation. The interpolation polynomial in the Lagrange form is a linear combination

\[
P_n(x) = \sum_{k=0}^{n} l_k(x)y_k = \sum_{k=0}^{n} \prod_{j=0, j\neq k}^{n} \frac{x-x_j}{x_k-x_j} y_k
\]

(3)

where \( l_k(x) = \prod_{j=0, j\neq k}^{n} \frac{x-x_j}{x_k-x_j} \).

C. Newton’s Interpolation Formula

The divided differences for a function \( f(x) \) are defined as follows:

\[
\begin{align*}
f(x_0, x_1) &= \frac{f(x_0) - f(x_1)}{x_0 - x_1} = \frac{f(x_0)}{x_0 - x_1} + \frac{f(x_1)}{x_1 - x_0} \\
f(x_0, x_1, x_2) &= \frac{f(x_0, x_1) - f(x_1, x_2)}{x_0 - x_2} \\
&\quad = \frac{f(x_0)}{(x_0 - x_1)(x_0 - x_2)} + \frac{f(x_1)}{(x_1 - x_0)(x_1 - x_2)} \\
&\quad + \frac{f(x_2)}{(x_2 - x_0)(x_2 - x_1)}
\end{align*}
\]

The recursive rule for constructing higher-order
The divided difference formulae are used to construct the divided difference table 1.

<table>
<thead>
<tr>
<th>$x_k$</th>
<th>$f(x_k)$</th>
<th>First divided difference</th>
<th>Second divided difference</th>
<th>Third divided difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_0$</td>
<td>$f(x_0)$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$x_1$</td>
<td>$f(x_1)$</td>
<td>$f(x_0, x_1)$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$x_2$</td>
<td>$f(x_2)$</td>
<td>$f(x_1, x_2)$</td>
<td>$f(x_0, x_1, x_2)$</td>
<td></td>
</tr>
<tr>
<td>$x_3$</td>
<td>$f(x_3)$</td>
<td>$f(x_2, x_3)$</td>
<td>$f(x_1, x_2, x_3)$</td>
<td></td>
</tr>
</tbody>
</table>

The Newton polynomial of degree $\leq n$ is

$$P_n(x) = f(x_0) + f(x_0, x_1)(x-x_0) + f(x_0, x_1, x_2)(x-x_0)(x-x_1) + \cdots + f(x_0, \ldots, x_n)(x-x_0)(x-x_1)\cdots(x-x_{n-1})$$

**III. Equidistant Interpolation**

A. Equidistant Lagrange’s Interpolation Formula

The equidistant polynomial interpolation is that the interpolation nodes $x_i$ are equal intervals. That is $x_i = x_{i-1} + h$ ($i = 1, 2, \ldots, n$).

$$p(x) = \sum_{i=0}^{n} \frac{y_i}{\prod_{j=i}^{n} (x-x_j)(x-x_{j+1})(x-x_{j+2})\cdots(x-x_{j+k})} \times y_k$$

where $E_k = \frac{y_k}{(-1)^k k!(n-k)! h^n}$.

B. Equidistant Newton’s Interpolation Formula

The first divided differences are calculated as follows:

$$f(x_0, x_1) = \frac{y_1 - y_0}{h}$$
$$f(x_1, x_2) = \frac{y_2 - y_1}{h}$$
$$f(x_{n-1}, x_n) = \frac{y_n - y_{n-1}}{h}$$

The second divided differences are calculated as follows:

$$f(x_0, x_1, x_2) = \frac{y_2 - 2y_1 + y_0}{2h^2}$$
$$f(x_1, x_2, x_3) = \frac{y_3 - 2y_2 + y_1}{2h^2}$$
$$f(x_{n-2}, x_{n-1}, x_n) = \frac{y_n - 2y_{n-1} + y_{n-2}}{2h^2}$$

The third divided differences are calculated as follows:

$$f(x_0, x_1, x_2, x_3) = \frac{y_3 - 3y_2 + 3y_1 + y_0}{3!h^3}$$
$$f(x_1, x_2, x_3, x_4) = \frac{y_4 - 3y_3 + 3y_2 + y_1}{3!h^3}$$
$$f(x_{n-1}, x_{n-2}, x_{n-1}, x_n) = \frac{y_n - 3y_{n-1} + 3y_{n-2} + y_{n-3}}{3!h^3}$$

The $k$th divided difference is calculated as follows:

$$f(x_0, x_1, \ldots, x_k) = \frac{y_k - C_1 y_{k-1} + C_2 y_{k-2} + \cdots + (-1)^k y_0}{k!h^k}$$

(4)

**Theorem:** When $x_0, x_1, \ldots, x_n$ are equal intervals. That is $x_i = x_{i-1} + h$ ($i = 1, 2, \ldots, n$), then

$$f(x_0, x_1, \ldots, x_n) = \frac{y_n - C_1 y_{n-1} + C_2 y_{n-2} + \cdots + (-1)^n y_0}{k!h^n}$$

(5)

**Proof:** (Proof by Mathematical Induction)

When $n = 1$,

$$f(x_0, x_1) = \frac{y_1 - y_0}{h} = \frac{y_1 + (-1)^1 C_1 y_0}{h}$$

Hence, the formula is true.

Assume that the formula is true for $n = k$,

$$f(x_0, x_1, \ldots, x_k) = \frac{y_k - C_1 y_{k-1} + C_2 y_{k-2} + \cdots + (-1)^k y_0}{k!h^k}$$

Then,

$$f(x_1, x_2, \ldots, x_{k+1}) = \frac{y_{k+1} - C_1 y_k + C_2 y_{k-1} + \cdots + (-1)^k y_0}{k!h^k}$$

When $n = k + 1$. 2060
f(x_0, x_1, \ldots, x_k, x_{k+1}) = \frac{f(x_1, \ldots, x_k, x_{k+1}) - f(x_0, x_1, \ldots, x_k)}{(k + 1)h}

= \frac{1}{(k + 1)! h^{k+1}} \left[ y_{k+1} - C_1^1 y_k + C_1^2 y_{k-1} + \cdots + (-1)^k y_1 \right] 

- \left[ y_k - C_1^1 y_{k-1} + C_1^2 y_{k-2} + \cdots + (-1)^k y_0 \right] 

= \frac{1}{(k + 1)! h^{k+1}} \sum_{i=0}^{k+1} (-1)^i C_{k+1}^i y_{i+1}

Hence, the formula is true for \( n = k + 1 \). The formula, therefore, is true for every divided difference.

Because the interpolating polynomial is formed as above using the topmost entries of divided-difference table in each column as coefficients, we only calculate the topmost entries of divided-difference table, needn’t calculate entire divided-difference table. So the computation is smaller than the traditional method.

IV. Numerical Example

Find interpolating polynomials \( P_4(x) \) based on the five points (-4,30),(-2,-42),(0,30),(2,6) and (4,30).

This is an equidistant interpolation problem. The equal width \( h \) is \( h = 2 \).

1) Equidistant Lagrange’s Interpolation

\[
E_0 = \frac{30}{(-1)^4 \cdot 0! \cdot 4! \cdot 2^4} = \frac{5}{64}
\]

\[
E_1 = \frac{-42}{(-1)^3 \cdot 1! \cdot 3! \cdot 2^4} = \frac{7}{16}
\]

\[
E_2 = \frac{30}{(-1)^2 \cdot 2! \cdot 2! \cdot 2^3} = \frac{15}{32}
\]

\[
E_3 = \frac{6}{(-1)^1 \cdot 3! \cdot 1! \cdot 2^2} = \frac{1}{16}
\]

\[
E_4 = \frac{30}{(-1)^0 \cdot 4! \cdot 0! \cdot 2^5} = \frac{5}{64}
\]

\[
P_4(x) = \frac{5}{64} (x + 2) x (x - 2)(x - 4)
\]

+ \frac{7}{16} (x + 4)(x - 2)(x - 4)

+ \frac{15}{32} (x + 4)(x + 2)(x - 2)(x - 4)

- \frac{1}{16} (x + 4)(x + 2)(x)(x - 4)

+ \frac{5}{64} (x + 4)(x + 2)x(x - 2)

= x^4 - x^3 - 16x^2 + 16x + 30

\( P_4(x) \) is shown in figure 1.

(2) Equidistant Newton’s Interpolation

\[
f(-4,-2) = \frac{y_1 - y_0}{h} = \frac{-42 - 30}{2} = -36
\]

\[
f(-4,-2.0) = \frac{y_2 - 2y_1 + y_0}{2h^2} = \frac{30 + 84 + 30}{2} = 18
\]

\[
f(-4,-2.0,2) = \frac{y_3 - 3y_2 + 3y_1 + y_0}{3h^3} = \frac{6 - 90 - 126 - 30}{3 \cdot 48} = -5
\]

\[
f(-4,-2.0,2,4) = \frac{y_4 - 4y_3 + 6y_2 - 4y_1 + y_0}{4!h^4} = \frac{30 - 24 + 180 + 168 + 30}{4 \cdot 24 \times 16} = 1
\]

Therefore,

\[
P_4(x) = 30 - 36(x + 4) + 18(x + 4)(x + 2) - 5(x + 4)(x + 2)x + (x + 4)(x + 2)x(x - 2)
\]

\[
= x^4 - x^3 - 16x^2 + 16x + 30
\]

Figure 1. Graph of \( P_4(x) \)

ACKNOWLEDGMENT

This work was partially supported by Natural Science Foundation of Jiangsu Province of China (No. BK2012700), the Open Project Program of Key Laboratory of Intelligent Computing & Information Processing (Xiangtan University), Ministry of Education (No. 2011ICIP05), Artificial Intelligence of Key Laboratory of Sichuan Province (2012RJ04) and the National Natural Science Foundation of China under Grant No. 61305058.

REFERENCES


