Lagrange Polynomial by MATLAB

أ.اسماء ابوبكر علي عون

Abstract

Completing an idea consisting of representing numerical data with an appropriate polynomial, then calculating the value of the dependent variable from the polynomial corresponding to any given value of the independent variable leads to the necessity of having a formula to represent a certain set of numerical data on a pair of variables through the appropriate polynomial. One such formula was developed in this study. The formula is derived from Lagrange's formula. The formula obtained to represent the numerical data, for the total population of India since 1971, has been applied by an appropriate polynomial

Introduction

Calculus provides many tools that can be used to understand the behaviour of functions, but mostly It is essential that these functions be continuous or differentiable. This is a problem in most "real" applications, where functions are used to model relationships between quantities, but our only knowledge of these functions consists of a set of discrete data points, where the data is It is obtained from the measurements. Therefore, we need to be able to build continuous functionality based on separate data.

The problem of creating such a continuous function is called data fit. In this research we discuss a special case of data fit known as interpolation, where the goal is to find a file a linear combination of n functions known to fit a set of data that imposes n constraints, and thus ensuring a unique solution that fits the data perfectly, not nearly. the widest the term "constraints" is used, rather than just "data points", because the description of data may be include additional information such as rates of change or requirements for which job is appropriate it has a certain number of continuous derivatives.

Lagrange Polynomial:

The Lagrange interpolation formula is a way to find a polynomial which takes on certain values at arbitrary points.

We will try to generalize the concept of linear interpolation by constructing a limit to a greater degree of estimation:

$$(x_0, f(x_0)), (x_1, f(x_1)), \dots\dots\dots, (x_n, f(x_n))$$

As:

$$p_2(x) = \frac{(x - x_1)}{(x_0 - x_1)} f(x_0) + \frac{(x - x_0)}{(x_1 - x_0)} f(x_1)$$

$$= l_0 f(x_0) + l_1 f(x_1)$$

$$l_0(x) = \frac{(x - x_1)}{(x_0 - x_1)} = f(x) = \begin{cases} 1, & x = x_0 \\ 0, & x \neq x_0 \end{cases}$$

$$l_1(x) = \frac{(x - x_0)}{(x_1 - x_0)} = f(x) = \begin{cases} 1, & x = x_1 \\ 0, & x \neq x_1 \end{cases}$$

$$p_n(x) = l_0(x_0) f(x_0) + l_1(x_1) f(x_1) + l_2(x_2) f(x_2) + \cdots + l_n(x_n) f(x_n)$$

Whereas $l_0(x), l_1(x), \dots\dots\dots, l_n(x)$, are polynomials of degree at most **n.**

In this case:

$$l_0(x_0) = \frac{(x - x_1)(x - x_2) \dots\dots (x - x_n)}{(x_0 - x_1)(x_0 - x_2) \dots\dots (x_0 - x_n)}$$

The same thing : $l_1(x_1), l_2(x_2), \dots\dots\dots, l_n(x_n)$

$$l_1(x_1) = \frac{(x - x_0)(x - x_2) \dots\dots (x - x_n)}{(x_1 - x_0)(x_1 - x_2) \dots\dots (x_1 - x_n)}$$

In general is :

$$l_i(x) = \frac{(x - x_0)(x - x_1) \dots\dots (x - x_{i-1})(x - x_{i+1}) \dots\dots (x - x_n)}{(x_i - x_0)(x_i - x_1) \dots\dots (x_i - x_{i-1})(x_i - x_{i+1}) \dots\dots (x_i - x_n)}$$

In this method , given $(x_0, y_0), \dots\dots\dots\dots, (x_n, y_n)$, one can fit a n^{th}order Lagrangian polynomial given by :

$$p_n(x) = \sum_{i=0}^{n} L_i(x) f(x_i)$$

Where n in $f_n(x)$ stands for the n^{th} order polynomial that approximates the function $= f(x)$, and

$$L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_i}{x_i - x_j}$$

$L_i(x)$ is a weighting function that includes a product of $n - 1$ terms with terms of $j = i$ omitted.

Then to find the first derivative, one can differentiate $f_n(x)$ once , and so on for other derivatives .

Example

We will use Lagrange interpolation to find the unique polynomial $p_3(x)$, of degree 3 or less , that agrees with the following data:

X	1	2	4	5
F(x)	1	0.3	0.25	0.2

And we will find $f(3)$

The answer:

In other word , we must have $p_3(1) = 1, p_3(2) = 0.3, p_3(4) = 0.25, p_3(5) = 0.2.$

First , we construct the Lagrange polynomials , using the formula:

$$L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{3} \frac{x - x_i}{x_i - x_j}$$

This yields

$$l_0(x_0) = \frac{(x-x_1)(x-x_2)(x-x_3)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)}$$

$$= \frac{(x-2)(x-4)(x-5)}{(1-2)(1-4)(1-5)}$$

$$= \frac{-1}{12}(x^3 - 11x^2 + 38x - 40)$$

$$l_1(x_1) = \frac{(x-x_0)(x-x_2)(x-x_3)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)}$$

$$= \frac{(x-1)(x-4)(x-5)}{(2-1)(2-4)(2-5)}$$

$$= \frac{1}{6}(x^3 - 10x^2 + 29x - 20)$$

$$l_2(x_2) = \frac{(x-x_0)(x-x_1)(x-x_3)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)}$$

$$= \frac{(x-1)(x-2)(x-5)}{(4-1)(4-2)(4-5)}$$

$$= \frac{-1}{6}(x^3 - 8x^2 + 17x - 10)$$

$$l_3(x_3) = \frac{(x-x_0)(x-x_1)(x-x_2)}{(x_3-x_0)(x_3-x_1)(x_3-x_2)}$$

$$= \frac{(x-1)(x-2)(x-4)}{(5-1)(5-2)(5-4)}$$

$$= \frac{1}{12}(x^3 - 7x^2 + 14x - 8)$$

By substituting, in follows that the Lagrange interpolating polynomial $p_3(x)$ is given by:

$$p_3(x) = \sum_{i=0}^{3} L_i(x) f(x_i)$$

$$p_3(x) = l_0(x_0)f(x_0) + l_1(x_1)f(x_1) + l_2(x_2)f(x_2) + l_3(x_3)f(x_3)$$

$$p_3(x) = \frac{-1}{12}(x^3 - 11x^2 + 38x - 40) + \frac{1}{3}(x^3 - 10x^2 + 29x - 20)$$
$$-\frac{2}{3}(x^3 - 8x^2 + 17x - 10) + \frac{5}{12}(x^3 - 7x^2 + 14x - 8)$$

$$p_3(x) = 1.95 - 1.225x + 0.3x^2 - 0.025x^3$$

$$f(3) \approx p_3(x) = 1.95 - 1.225(3) + 0.3(3^2) - 0.025(3^3) = 0.3$$

```
EDU x=[1 2 4 5];
y=[1 0.5 0.25 0.2];
n=length(x)-1;
xp=3; %approximate y when x=3
sm=0;
for i=1:n+1
    pr=1;
    for j=1:n+1
        if j~=i
            pr=pr*(xp-x(j))/(x(i)-x(j));
        end
    end
    sm=sm+y(i)*pr;
end
yp=sm
```

```
yp =

   0.3000
```

While Lagrange polynomials are easy to calculate, they are difficult to work with. Furthermore, if new interpolation points are added, all Lagrange polynomials must be recalculated.

Unfortunately, it is not uncommon, in practice, to add to an existing set of interpolation points. It can be determined after calculating a polynomial of degree m,$p_m(x)$ of a function $f(x)$ that $p_m(x)$ is not a sufficiently accurate approximation of $f(x)$ on some domain. Therefore, an interpolation polynomial of a higher degree must be calculated, which requires additional interpolation points.

To address these issues, we consider the problem of calculating an internal polynomial

Frequently. More precisely, let $m > 0$, and let $p_m(x)$ be the polynomial of degree m that interpolates the function $f(x)$ at the points $x_0, x_1, \dots\dots, x_m$. Ideally, we would like to be able to have it $p_m(x)$ from polynomials of degree $m-1$ that interpolate $f(x)$ at points chosen from among $x_0, x_1, \dots\dots, x_m$. The following result shows that this is possible.

Theorem :

let n be a positive integer , and let $f(x)$ be a function defined on a domain containing the $n+1$ distinct points $x_0\ x_1, \dots\dots, x_n$,and let $p_n(x)$ be the polynomial of degree n that interpolates $f(x)$ at the point $x_0\ x_1, \dots\dots, x_n$ .For each $i = 0,1, \dots\dots, n$, we define $p_{n-1,i}(x)$ to be the polynomial of degree $n-1$ that interpolates $f(x)$at the points $x_0, x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n$.

If i and j are distinct nonnegative integers not exceeding n, then:

$$p_n(x) = \frac{(x-x_i)p_{n-1,j}(x)-(x-x_i)p_{n-1,i}(x)}{x_i-x_j}.$$

This result leads to an algorithm called the *Neville's method* that computes the value of $p_n(x)$

At a given point using the values of low-degree polynomials at x. We now describe this algorithm in detail.

Algorithm: let $x_0\ x_1, \dots\dots, x_n$. distinct numbers, and let $f(x)$ be a function defined in the domain of contain these numbers. Given a number x^*, the following algorithm computes $y^* = p_n(x^*)$,where $p_n(x)$ is the n[th] polynomial interpolation of $f(x)$ that interpolates $f(x)$ at the points $x_0\ x_1, \dots\dots, x_n$.

$for\ j = 0\ to\ n\ d$

$Q_j = f(x_j)$

end

$for\ \ j = 1\ to\ n\ do$

$for\ \ m = n\ to\ j\ do$

$$Q_m = \left[(x-x_m)Q_{m-1} - \left(x-x_{m-j}\right)Q_m\right]/(x_{m-j}-x_m)$$

end

end

$y^* = Q_n$

At the j[th] repeat the outer loop , the number Q_m , for $m = n, n-1, \dots\dots, j$, represents the value at x of the polynomial that interpolates $f(x)$ at the points $x_m, x_{m-1}, \dots\dots, x_{m-j}$.

The previous theorem can be used to calculate the polynomial $p_n(x)$ itself, instead of that

Value at a given point. This results in an alternative method for constructing an interpolation polynomial, called Newton's interpolation, which is more suitable for tasks such as additional inclusion interpolation points.

In some applications, polynomial interpolation $p_n(x)$ is used to fit a known function $f(x)$in

Points $x_0, \dots\dots, x_n$ usually because $f(x)$ is not possible for tasks such as calculus or integration which is easy for a polynomial, or because it is not easy to evaluate $f(x)$ at points other than interpolation points. In such an application, it is possible to specify how good the $p_n(x)$ approximation $f(x)$.

To this end, we assume that not one of the interpolation points $x_0, \dots\dots, x_n$ and we define

$$\varphi(t) = f(t) - p_n(t) - \frac{f(x) - p_n(x)}{\pi_{n+1}(x)}\pi_{n+1}(t).$$

Where

$$\pi_{n+1}(x) = (x - x_0)(x - x_1) \dots . (x - x_n)$$

is a polynomial of degree n + 1. Since x is not an interpolation point, it follows that

$\varphi(t)$has at least n + 2 zeros: x, and interpolation points $x_0, \dots\dots, x_n$. In addition to,

$\pi_{n+1}(x) \neq 0$, so $\varphi(t)$is well defined.

If it is differentiable $n + 1$ time continuously over an interval $[a, b]$ contains interpolation

The points and x, then, by Generalized Rolle's theorem, $\varphi^{(n+1)}$ must contain at least one zero in $[a, b]$.So, at some point $\xi(x)$ in $[a, b]$, that depends on x, we have:

$$0 = \varphi^{(n+1)}\big(\xi(x)\big) = f^{(n+1)}(t) - \frac{f(x) - p_n(x)}{\pi_{n+1}(x)}(n+1)!$$

which yields the following result.

Example: Suppose that we wish to approximate the function $f(x) = 1/(1 + x^2)^2$ on the interval [−6, 6] with a Twelfth -degree interpolating polynomial that agrees with $f(x)$ at 13 equally-spaced points $x_0, x_1, x_2, \ldots, x_{12}$in [−6, 6], where $x_j = -6 + j$, for j = 0, 1, . . . , 12. Figure 1 shows that the resulting polynomial is not a good approximation of $f(x)$ on this interval, even though it agrees with $f(x)$at the interpolation points. The following MATLAB session shows how the plot in the figure can be created.

```
%creat vector of 11 equally spaced points in [-5,5]
x=linspace (-6,6,13);
%compute corresponding y-values
y=1./(1+x.^2).^2;
%compute 10th-degree interpolating polynomial
p=polyfit(x,y,12);
%for plotting , create vector of 100 equally spaced
points
xx=linspace(-6,6);
% compute corresponding y-values to plot function
yy=1./(1+xx.^2).^2;
% plot function
plot(xx,yy)
%tell MATLAB that next plot should be superimposed on
%current one
hold on
%plot polynomial , using polyval to compute values
%and a red dashed curve
plot(xx,polyval(p,xx),'r--')
%indicate interpolation points on plot using circles
plot(x,y,'o')
```

```
% label axes
xlabel('x')
ylabel('y')
% set caption
title('runge''s example')
```

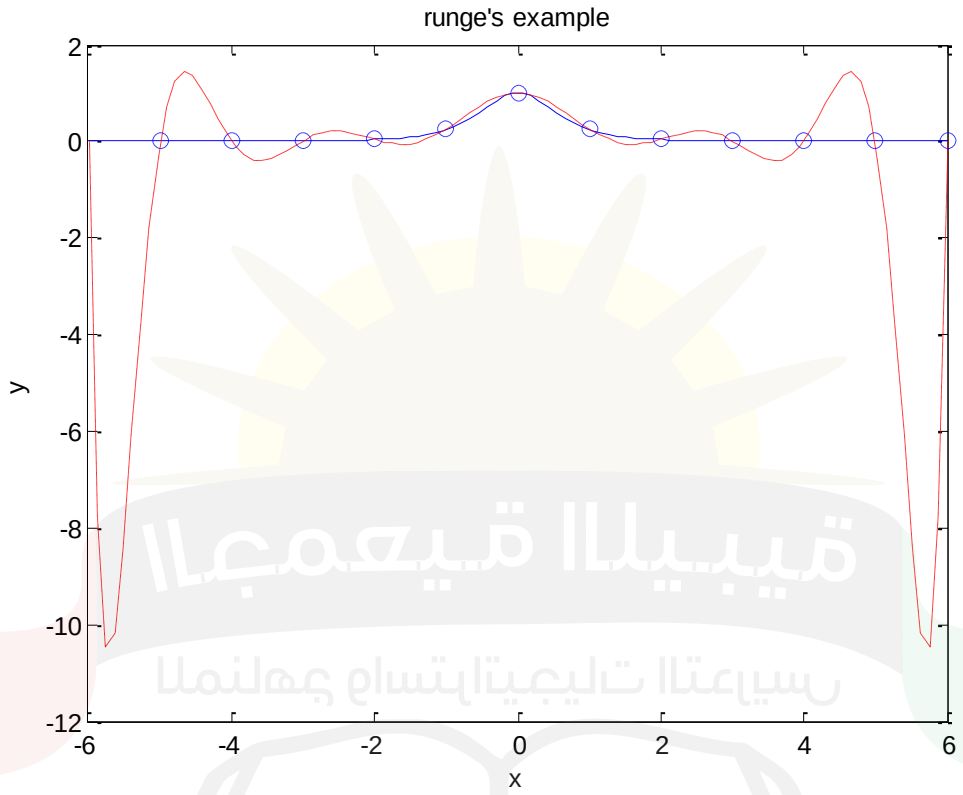

Figure 1: The function $f(x) = 1/(1+x^2)^2$ (solid curve) cannot be interpolated accurately on [−6, 6] using a Twelfth -degree polynomial (dashed curve) with equally-spaced interpolation points. This example that illustrates the difficulty that one can generally expect with high-degree polynomial interpolation with equally-spaced points is known as Runge's example.

Conclusion:

In this research , work was on the Lagrange Polynomial , and we tried to the answer to the main question,. On the other hand, we did a lot of examples to show it.

By proofing the Lagrange Polynomial by a using MATLAB as we have done as a part of examples and theorems.

Finally, these examples and theorems useful to understand concourse using the MATLAB ,. in the end, the Lagrange Polynomial very useful , thought out, comeback ranking for page ,but it is very important to keep in mind that it can be developed further.

References

1.Prof.Dr.Adel Nasim Adib ,Analytical Geometry , Publishing house Dar El Nasher for Universities , Egypt,2008,ISBN:977-316,232 P.(Arabic Language).

2. https://ar.unionpedia.org/%D9%85%D8%AA%D8%B9%D8%AF%D8%AF_%D8%AD%D8%AF%D9%88%D8%AF_%D9%84%D8%A7%D8%BA%D8%B1%D8%A7%D9%86%D8%AC

3.Bell,W.W.;(1968)Special Functions for Scientists and Engineers, Van Nos-trand,London

4.Powell,M.J.D.:(1981)Approximation theory and Methods,Cambridge University Press,Cambridge.

5.كوارتروني ، ألفيو. ساليري ، فاوستو (2003). الحوسبة العلمية مع MATLAB . نصوص في العلوم والهندسة الحاسوبية. المجلد. 2. سبرينغر. ص. 66. ردمك 978-3-6-44363-540..

6. https://en.wikipedia.org/wiki/Lagrange_polynomial#References