Ch. 8 Interpolation

You will frequently have occasions to estimate values between precise data points. The most common method used for this purpose is polynomial interpolation.

\[ f(x) = a_0 + a_1x + a_2x^2 + \ldots + a_nx^n \]  

(1)

For \( n+1 \) data points there is one and only one polynomial of order \( n \) or less that passes through all points. For example there is only one straight line (that is, a first-order polynomial) that connects two points.

First order (linear)  
Second order (parabolic)  
Third order (cubic)

Polynomial interpolation consists of determining the unique \( n \)-th order polynomial that fits \( n+1 \) data points.

There are a variety of mathematical forms in which the polynomial can be expressed. We are interested in only one representation but we will describe two alternatives that are well-suited for computer implementation. There are the Newton and Lagrange polynomials, and we will study the Lagrange polynomials.

**Newton’s Divided-Difference Interpolating Polynomials**

This is among the most popular and useful forms.
**Linear Interpolations**

The simplest form of interpolation is to connect two data points with a straight line. This technique called linear interpolation is depicted graphically. Using similar triangles

\[
\frac{f_i(x) - f(x_0)}{x - x_0} = \frac{f(x_i) - f(x_0)}{x_i - x_0}
\]

which can be rearranged to yield

\[
f_i(x) = f(x_0) + \frac{f(x_i) - f(x_0)}{x_i - x_0}(x - x_0)
\]

which is a linear interpolation formula.

The notation \(f_i(x)\) designates that this is a first-order interpolation polynomial. The term \([f(x_i) - f(x_0)]/[x_i - x_0]\) is a finite divided-difference approximation of the first derivative. In general, the smaller the interval between the data points, the better the approximation.

**Quadratic Interpolation**

Suppose three data points are available. A particularly convenient form for this purpose is

\[
f_2(x) = b_0 + b_1(x - x_0) + b_2(x - x_0)(x - x_1)
\]

This form although looks different from that given in Eq 1, These are equivalent. It can be show as follows from Eq 4

\[
f_2(x) = b_0 + b_1x - b_1x_0 + b_2x - 2b_2x_0 + b_2xx_0 - b_2xx_1
\]

or collecting terms

\[
f_2(x) = a_0 + a_1x + a_2x^2
\]

where
\[ a_0 = b_0 - b_1 x_0 + b_2 x_0 x_1, \quad a_1 = b_1 - b_2 x_0 - b_2 x_1, \quad a_2 = b_2 \]

Eqs 1 and 4 are alternative equivalent formulations of the unique second-order polynomial joining the three points.

A simple procedure can be used to determine the values of the coefficients. For \( b_0 \) Eq 4 with \( x = x_0 \) can be used to compute

\[ b_0 = f(x_0), \quad b_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0} \]  

(5,6)

by substituting Eqs (5,6) into Eq 4, which can be evaluated at \( x = x_2 \) and solved (after some algebraic manipulations for

\[ b_2 = \frac{f(x_2) - f(x_1)}{x_2 - x_1} \frac{f(x_1) - f(x_0)}{x_1 - x_0} \]  

(7)

The last term in Eq 4 introduces the second-order curvature into the formula

**Lagrange Interpolating Polynomials**

Given a set of data \( (x_i, y_i), i = 1, 2, \ldots, n \), find a smooth curve that passes through the data.

From the problem statement, we must have

\[ f(x_i) = y_i, \quad i = 1, 2, \ldots, n \]  

(8)

- The function should be easy to evaluate.
- It should be easy to integrate and differentiate.
- It should be linear in the adjustable parameters.
In Lagrange interpolation we pass a polynomial of lowest possible degree through \( n \) given data points. Since \( n \) parameters are needed, the degree required is \( (n-1) \) so that

\[
\ell_{(n-1)}(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_{(n-2)} x^{(n-2)} + a_{(n-1)} x^{(n-1)}
\]  

(9)

The straightforward approach of finding the coefficients is to substitute Eq. (9) in Eq. (8). We obtain

\[
y_i = a_0 + a_1 x_i + a_2 x_i^2 + \ldots + a_{(n-2)} x_i^{(n-2)} + a_{(n-1)} x_i^{(n-1)} \quad i = 1, 2, 3, \ldots n
\]  

(10)

which represents a set of \( n \) linear algebraic equations in \( n \) unknowns \( a_0, a_1, \ldots, a_{(n-1)} \) since the \( x_i \) and \( y_i \) are known.

This set can be solved by standard linear equation solvers, but this is not a good way to proceed, because Eq. (10) needs a computer if \( n \) is larger than 4 or 5.

- Eq. (10) becomes ill-conditioned for \( n \) larger than 4-5
- And finally it is better to have a closed form expression in any case.

By ill-conditioned we mean that the solution of the system of equations is very sensitive to small changes in the data. When such a system is solved, small errors are magnified and result may contain large errors.

Thus we seek an alternative approach.

From Eq. (10) we see that the coefficients \( a_0, a_1, \ldots, a_{(n-1)} \) must be linear combination of the \( y_i \).

The most general expression that is linear in each of the \( y_i \) and a polynomial of degree \( (n-1) \) in \( x \) is
\[ f_{n-1}(x) = \sum_{i=1}^{n} L_i(x)y_i \] (11)

where the \( L_i(x) \) are polynomials of degree \((n-1)\). Thus \( f_{(n-1)}(x) \) must have this form.

The problem is to find \( L_i(x) \)

Let \( L_i(x_j) = \delta_{ij} \quad j = 1, 2, \ldots n \) (12)

where \( \delta_i = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \) (13)

\[ L_i(x_j) = a_j (x - x_1)(x - x_2) \ldots (x - x_{(n-1)}) (x - x_n) \] (14)

or can be abbreviated as

\[ L_i(x_j) = a_j \prod_{j \neq i}^{n} (x - x_j) \] (15)

and

\[ a_j = \frac{1}{\prod_{j \neq i}^{n} (x_i - x_j)} \] (16)

Then the Lagrange polynomials are

\[ L_i(x) = \prod_{j \neq i}^{n} \frac{x - x_j}{x_i - x_j} \] (17)

and the interpolating polynomial becomes

\[ f_{n-1}(x) = \sum_{i=1}^{n} L_i(x)f(x_i) \] (18)

**Example:** Find a quadratic polynomial using the three given points

\((x_i, y_i) \quad i = 1, 2, 3\) \quad \(x = [-2, 0, 2]\) \quad \(y = [4, 2, 8]a\)

We then have \(n = 3\)

\[ f_2(x) = \sum_{i=1}^{3} L_i(x)y_i \]
\[ = L_1 y_1 + L_2 y_2 + L_3 y_3 \]

Now

\[ L_1 (x) = \left( \frac{x - x_2}{x_1 - x_2} \right) \left( \frac{x - x_3}{x_1 - x_3} \right) = \left( \frac{x - 0}{2 - 0} \right) \left( \frac{x - 2}{2 - 2} \right) \]

\[ L_2 (x) = \left( \frac{x - x_1}{x_2 - x_1} \right) \left( \frac{x - x_3}{x_2 - x_3} \right) = \left( \frac{x - (-2)}{0 - (-2)} \right) \left( \frac{x - 2}{0 - 2} \right) \]

\[ L_3 (x) = \left( \frac{x - x_2}{x_3 - x_2} \right) \left( \frac{x - x_1}{x_3 - x_1} \right) = \left( \frac{x - (-2)}{2 - (-2)} \right) \left( \frac{x - 0}{2 - 0} \right) \]

\[ f(x) = \frac{x(x - 2)}{8} + \frac{(x + 2)(x - 2)}{-4} + \frac{x(x + 2)}{8} - x^2 + x + 2 \]
function c = Lagrange_coef(x, y)
% Calculate coefficients of Lagrange Functions
n = length(x);
for k = 1:n
    d(k) = 1;
    for i = 1:n
        if i ~= k
            d(k) = d(k) * (x(k) - x(i));
        end
        c(k) = y(k) / d(k);
    end
end

function p = Lagrange_Eval(t, x, c)
% Evaluate Lagrange interpolation polynomial at x = t
m = length(x);
for i = 1:length(t),
    p(i) = 0;
    for j = 1:m
        N(j) = 1;
        for k = 1:m
            if j ~= k
                N(j) = N(j) * (t(i) - x(k));
            end
        end
        p(i) = p(i) + N(j) * c(j);
    end
end