1 Introduction

Data is often given in discrete values along a continuum. However, you may require estimates at points between the discrete values. This chapter describes techniques to fit curves to such data to obtain intermediate estimates.

There are two general approaches for curve fitting that are distinguished from each other on the basis of the amount of *error* associated with the data.

- First, where the data exhibits a significant degree of error or noise, the strategy is to derive a single curve that represents the general trend of the data. **Because any individual data point may be incorrect, we make no effort to intersect every point. Rather, the curve is designed to follow the pattern of the points taken as a group**. One approach of this nature is called **least-squares regression** (fig.1).

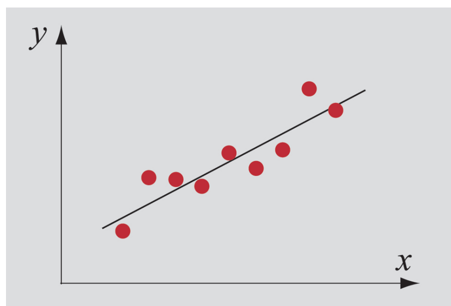


Fig.1

- Second, where the data is known to be very precise, the basic approach is to fit a curve or a series of curves that pass directly through each of the points. The estimation of values between well-known discrete points is called **interpolation** (fig.2).

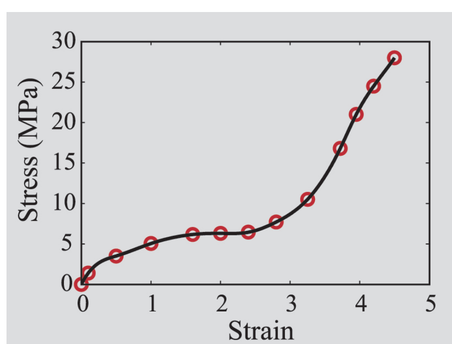


Fig.2

2 Least square regression

This section is devoted to least-squares regression. We will first learn how to fit the best straight line through a set of *uncertain* data points. This technique is called **linear regression**. We will learn how to calculate the slope and intercept of this straight line. The second half of this section presents a general technique for fitting a best polynomial. Thus, you will learn to derive a parabolic, cubic, or higher-order polynomial that optimally fits *uncertain* data. Linear regression is a subset of this more general approach, which is called **polynomial regression**.

2.1 Linear regression

The simplest example of a least squares approximation is fitting a straight line to a set of paired observations: (x_1, y_1), (x_2, y_2), ..., (x_n, y_n). The mathematical expression for the straight line is

$$y = a_o + a_1 x + e$$

where a_o and a_1 are coefficients representing the intercept and the slope, respectively, and e is the error, or residual, between the model and the observations, which can be represented by rearranging the previous equation as

$$e = y - a_o - a_1 x$$

Thus, the error, or residual, is the discrepancy between the true value of y and the approximate value, $a_o + a_1 x$, predicted by the linear equation.

- **Criteria for a "Best" fit**

  One strategy for fitting a best line through the data would be to minimize the sum of the squares of residual errors for all the available data, as in

  $$E_r = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - a_o - a_1 x_i)^2$$

- **Linear least square regression**

  Linear least-squares regression is a procedure in which the coefficients a_o and a_1 of a linear function ($y = a_o + a_1 x$) are determined such that the function has the best fit to a given set of data points. The best fit is defined as the smallest possible total error that is calculated by adding the squares of the residuals.

  To determine a_o and a_1, the previous equation is differentiated with respect to each coefficient.

  $$\begin{cases} \dfrac{\partial E_r}{\partial a_o} = -2 \sum (y_i - a_o - a_1 x_i) \\[2ex] \dfrac{\partial E_r}{\partial a_1} = -2 \sum [(y_i - a_o - a_1 x_i) x_i] \end{cases}$$

  Setting these derivatives equal to zero will result in a minimum E_r. When solved for a_o and a_1 it gives

  $$\begin{cases} a_1 = \dfrac{n \sum_{i=1}^{n} x_i y_i - \left(\sum_{i=1}^{n} x_i \right) \left(\sum_{i=1}^{n} y_i \right)}{n \sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i \right)^2} \\[4ex] a_o = \dfrac{\left(\sum_{i=1}^{n} x_i^2 \right) \left(\sum_{i=1}^{n} y_i \right) - \left(\sum_{i=1}^{n} x_i y_i \right) \left(\sum_{i=1}^{n} x_i \right)}{n \sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i \right)^2} \end{cases}$$

Any line other than the one computed results in a larger sum of the squares of the residuals. Thus, the line is unique and in terms of our chosen criterion is a best line through the points.

2.2 Linearization of nonlinear relationships

Linear regression provides a powerful technique for fitting a best line to data. However, it is predicated on the fact that the relationship between the dependent and independent variables is linear. This is not always the case, and the first step in any regression analysis should be to plot and visually inspect the data to ascertain whether a linear model applies. For example, fig.3 shows some data that is obviously curvilinear. In some cases, techniques such as polynomial regression, which is described in the next section, are appropriate. For others, transformations can be used to express the data in a form that is compatible with linear regression.

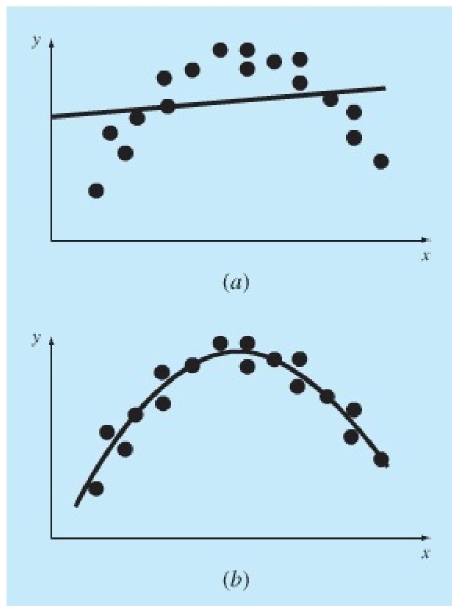


Fig.3

One example is the exponential model

$$y = a_1 e^{b_1 x}$$

where a_1 and b_1 are constants. This model is used in many fields of engineering to characterize quantities that increase (positive b_1) or decrease (negative b_1) at a rate that is directly proportional to their own magnitude. Another example of a nonlinear model is the simple power equation

$$y = a_2 x^{b_2}$$

where a_2 and b_2 are constant coefficients. This model has wide applicability in all fields of engineering.

Nonlinear regression techniques (not discussed in this chapter) are available to fit these equations to experimental data directly. However, a simpler alternative is to use mathematical manipulations to transform the equations into a linear form. Then, simple linear regression can be employed to fit the equations to data.

For example the first equation can be linearized by taking its natural logarithm to yield

$$\ln(y) = \ln(a_1) + b_1 x$$

Thus, a plot of $\ln(y)$ versus x will yield a straight line with a slope of b_1 and an intercept of $\ln(a_1)$. The second equation is linearized by taking its logarithm to give

$$\ln(y) = b_2 \ln(x) + \ln(a_2)$$

Thus, a plot of $\ln(y)$ versus $\ln(x)$ will yield a straight line with a slope of b_2 and an intercept of $\ln(a_2)$.

In their transformed forms, these models can use linear regression to evaluate the constant coefficients. They could then be transformed back to their original state and used for predictive purposes.

Many other nonlinear equations can be transformed into linear form in a similar way. The next table lists several such equations.

Nonlinear equation	Linear form	Relationship to $Y = a_1X + a_0$	Values for linear least-squares regression	Plot where data points appear to fit a straight line
$y = bx^m$	$\ln(y) = m\ln(x) + \ln(b)$	$Y = \ln(y)$, $X = \ln(x)$ $a_1 = m$, $a_0 = \ln(b)$	$\ln(x_i)$ and $\ln(y_i)$	y vs. x plot on logarithmic y and x axes. $\ln(y)$ vs. $\ln(x)$ plot on linear x and y axes.
$y = be^{mx}$	$\ln(y) = mx + \ln(b)$	$Y = \ln(y)$, $X = x$ $a_1 = m$, $a_0 = \ln(b)$	x_i and $\ln(y_i)$	y vs. x plot on logarithmic y and linear x axes. $\ln(y)$ vs. x plot on linear x and y axes.
$y = b10^{mx}$	$\log(y) = mx + \log(b)$	$Y = \log(y)$, $X = x$ $a_1 = m$, $a_0 = \log(b)$	x_i and $\ln(y_i)$	y vs. x plot on logarithmic y and linear x axes. $\ln(y)$ vs. x plot on linear x and y axes.
$y = \frac{1}{mx + b}$	$\frac{1}{y} = mx + b$	$Y = \frac{1}{y}$, $X = x$ $a_1 = m$, $a_0 = b$	x_i and $1/y_i$	$1/y$ vs. x plot on linear x and y axes.
$y = \frac{mx}{b + x}$	$\frac{1}{y} = \frac{b}{m}\frac{1}{x} + \frac{1}{m}$	$Y = \frac{1}{y}$, $X = \frac{1}{x}$ $a_1 = \frac{b}{m}$, $a_0 = \frac{1}{m}$	$1/x_i$ and $1/y_i$	$1/y$ vs. $1/x$ plot on linear x and y axes.

Fig.4

2.2.1 How to choose an appropriate nonlinear function

A plot of the given data points can give an indication as to the relationship between the quantities. Whether the relationship is linear or nonlinear can be determined by plotting the points in a figure with linear axes. If in such a plot the points appear to line up along a straight line, then the relationship between the plotted quantities is linear. A plot with linear axes in which the data points appear to line up along a curve indicates a nonlinear relationship between the plotted quantities. The question then is which nonlinear function to use for the curve fitting. Many times in engineering and science there is knowledge from a guiding theory of the physical phenomena and the form of the mathematical equation associated with the data points. For example, the process of charging a capacitor is modeled with an exponential function. If there is no knowledge of a possible form of the equation, choosing the most appropriate nonlinear function to curve-fit given data may be more difficult.

For given data points it is possible to foresee, to some extent, if a proposed nonlinear function has a potential for providing a good fit. This is done by plotting the data points in a specific way and examining whether the points appear to fit a straight line. For the functions listed in the previous table this is shown in the fifth (last) column of the table. For power and exponential functions, this can be done by plotting the data using different combinations of linear and logarithmic axes. For all functions it can be done by plotting the transformed values of the data points in plots with linear axes.

2.3 Polynomial regression

Previously, a procedure was developed to derive the equation of a straight line using the least-squares criterion. Some engineering data, although exhibiting a marked pattern such as seen in fig.3(a), is poorly represented by a straight line. For these cases, a curve would be better suited to fit the data. As discussed in the previous section, one method to accomplish this objective is to use transformations. Another alternative is to fit polynomials to the data using **polynomial regression**. The least-squares procedure can be readily extended to fit the data to a higher-order polynomial. For example, suppose that we fit a second-order polynomial or quadratic:

$$y = a_o + a_1 x + a_2 x^2 + e$$

For this case the sum of the squares of the residuals is

$$E_r = \sum_{i=1}^{n} (y_i - a_o - a_1 x_i - a_2 x_i^2)^2$$

Following the procedure of the previous section, we take the derivative of the above equation with respect to each of the unknown coefficients of the polynomial, as in

$$\begin{cases} \dfrac{\partial E_r}{\partial a_o} = -2 \sum (y_i - a_o - a_1 x_i - a_2 x_i^2) \\[2ex] \dfrac{\partial E_r}{\partial a_1} = -2 \sum [(y_i - a_o - a_1 x_i - a_2 x_i^2) x_i] \\[2ex] \dfrac{\partial E_r}{\partial a_2} = -2 \sum [(y_i - a_o - a_1 x_i - a_2 x_i^2) x_i^2] \end{cases}$$

These equations can be set equal to zero and arranged to develop the following set of linear equations

$$\begin{cases} n a_o + \left(\sum x_i\right) a_1 + \left(\sum x_i^2\right) a_2 = \sum y_i \\[2ex] \left(\sum x_i\right) a_o + \left(\sum x_i^2\right) a_1 + \left(\sum x_i^3\right) a_2 = \sum x_i y_i \\[2ex] \left(\sum x_i^2\right) a_o + \left(\sum x_i^3\right) a_1 + \left(\sum x_i^4\right) a_2 = \sum x_i^2 y_i \end{cases}$$

where all summations are from $i = 1$ to $i = n$. Note that the above three equations are linear and have three unknowns: a_o, a_1, and a_2. The coefficients of the unknowns can be calculated directly from the observed data. For this case, we see that the problem of determining a least-squares second-order polynomial is equivalent to solving a system of three simultaneous linear equations. Techniques to solve such equations were discussed in the previous chapter.

The two-dimensional case can be easily extended to an mth-order polynomial as

$$y = a_o + a_1 x + a_2 x^2 + \; + a_m x^m + e$$

The foregoing analysis can be easily extended to this more general case. Thus, we can recognize that determining the coefficients of an mth-order polynomial is equivalent to solving a system of $(m + 1)$ simultaneous linear equations.

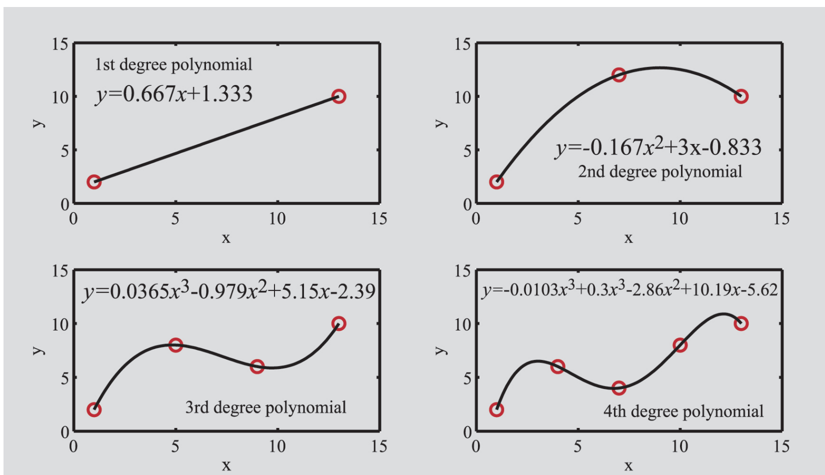
3 Interpolation

You will frequently have occasions to estimate intermediate values between precise data points. The most common method used for this purpose is **polynomial interpolation**. Recall that the general formula for an nth-order polynomial is

$$f(x) = a_o + a_1 x + a_2 x^2 + \quad + a_n x^n$$

For $n+1$ data points, there is one and only one polynomial of order n that passes through all the points. For example, there is only one straight line (that is, a first-order polynomial) that connects two points. Similarly, only one parabola connects a set of three points. Polynomial interpolation consists of determining the unique nth-order polynomial that fits $n+1$ data points. This polynomial then provides a formula to compute intermediate values.

Although there is one and only one nth-order polynomial that fits $n+1$ points, there are a variety of mathematical formats in which this polynomial can be expressed. In this chapter, we will describe three alternatives: the **standard**, **Newton** and **Lagrange** polynomials.



3.1 Standard interpolating polynomials

The standard form of an mth order polynomial is

$$f(x) = a_o + a_1 x + a_2 x^2 + \quad + a_m x^m$$

The coefficient $a_i's$ are determined by solving a system of $m+1$ linear equations that are obtained by writing the polynomial explicitly for each point. The resulting system is of the form shown below and can be solved using one of the techniques discussed in the previous chapter.

$$\begin{bmatrix} 1 & x_1 & x_1^2 .... & x_1^m \\ 1 & x_2 & x_2^2 .... & x_2^m \\ \\ 1 & x_{m+1} & x_{m+1}^2 .... & x_{m+1}^m \end{bmatrix} \begin{bmatrix} a_o \\ a_1 \\ \\ a_m \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \\ y_{m+1} \end{bmatrix}$$

3.2 Lagrange interpolating polynomials

Lagrange interpolating polynomials are a particular form of polynomials that can be written to fit a given set of data points by using the values at the points. The polynomials are in form

$$f(x) = \sum_{i=1}^{n} a_i \prod_{j=1, j \neq i}^{n} (x - x_j)$$

for n point (x_1, y_1)......, (x_n, y_n) one can show that the coefficients a_i stistfy

$$a_i = y_i \prod_{j=1, j\neq i}^{n} \frac{1}{(x_i - x_j)}$$

and

$$f(x) = \sum_{i=1}^{n} y_i \prod_{j=1, j\neq i}^{n} \frac{(x - x_j)}{(x_i - x_j)} = \sum_{i=1}^{n} y_i L_i(x)$$

where $L_i(x) = \prod_{j=1, j\neq i}^{n} \frac{(x - x_j)}{(x_i - x_j)}$ are called the **Lagrange functions**

- For a first order polynomial

$$f(x) = a_1(x - x_2) + a_2(x - x_1)$$

  where

$$a_1 = \frac{y_1}{x_1 - x_2} \qquad and \qquad a_2 = \frac{y_2}{x_2 - x_1}$$

- The spacing between the data points does not have to be equal.
- If an interpolated value is calculated for a given set of data points, and then the data set is enlarged to include additional points, all the terms of the Lagrange polynomial have to be calculated again.

3.3 Newton's interpolating polynomials

As stated above, there are a variety of alternative forms for expressing an interpolating polynomial. Newton's interpolating polynomial is among the most popular and useful forms. Before presenting the general equation, we will introduce the first and second order versions because of their simple visual interpretation.

The general form of an nth order Newton's interpolating polynomials is

$$f(x) = a_o + a_1(x - x_1) + a_2(x - x_1)(x - x_2) + ... + a_n(x - x_1)(x - x_2)....(x - x_n)$$

3.3.1 First order Newton's polynomial

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

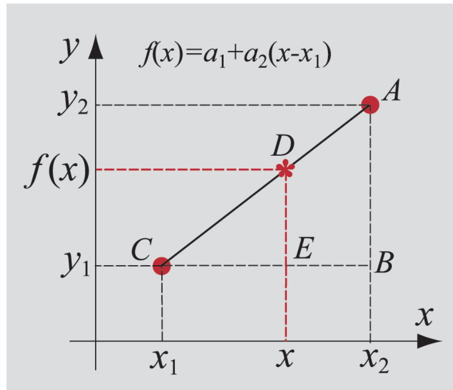
$$\frac{f(x) - y_1}{x - x_1} = \frac{y_2 - y_1}{x_2 - x_1}$$

which can be rearranged to yield

$$f(x) = y_1 + \frac{y_2 - y_1}{x_2 - x_1}(x - x_1) = a_o + a_1(x - x_1)$$

where

$$a_o = y_1 \qquad a_1 = \frac{y_2 - y_1}{x_2 - x_1}$$



1. In general, the smaller the interval between the data points, the better the approximation. This is due to the fact that, as the interval decreases, a continuous function will be better approximated by a straight line. This characteristic is demonstrated in the following example. Two linear interpolations to estimate ln(2). Note how the smaller interval provides a better estimate.

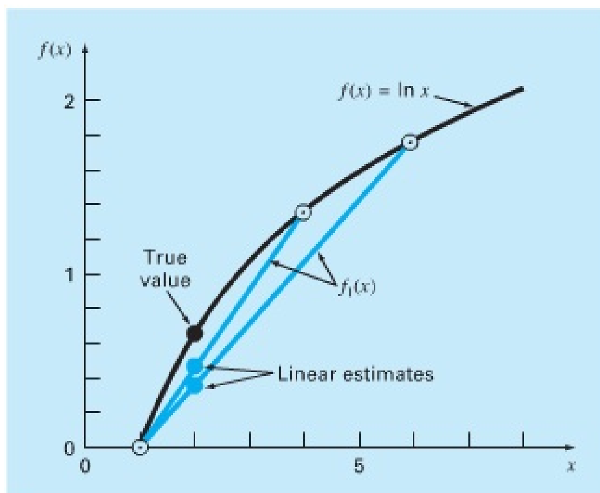


Fig.9

3.3.2 Second order Newton's polynomial

The error in the previous example resulted from our approximating a curve with a straight line. Consequently, a strategy for improving the estimate is to introduce some curvature into the line connecting the points. If three data points are available, this can be accomplished with a second-order polynomial (also called a quadratic polynomial or a parabola). A particularly convenient form for this purpose is

$$f_2(x) = a_o + a_1(x - x_1) + a_2(x - x_1)(x - x_2)$$

Note that although the above equation might seem to differ from the general polynomial introduced at the beginning of this section, the two equations are equivalent. This can be shown by multiplying the terms in the above equation to yield

$$f(x) = (a_o - a_1x_1 + a_2x_1x_2) + (a_1 - a_2x_1 - a_2x_2)\, x + (a_2)\, x^2$$

a simple procedure can be used to determine the values of the coefficients, and it results in

$$\begin{cases} a_o = y_1 \\ a_1 = \dfrac{y_2 - y_1}{x_2 - x_1} \\ a_2 = \dfrac{\dfrac{y_3 - y_2}{x_3 - x_2} - \dfrac{y_2 - y_1}{x_2 - x_1}}{x_3 - x_1} \end{cases}$$

Notice that, as was the case with linear interpolation, b_1 still represents the slope of the line connecting points x_1 and x_2. Thus, the first two terms are equivalent to linear interpolation from x_1 and x_2. The last term, $b_2(x - x_1)(x - x_2)$, introduces the second-order curvature into the formula.

3.3.3 Third order Newton's polynomial

In the same manner a fourth order polynomial can be devised from four given points $(x_1, y_1$, $(x_2, y_2$, $(x_3, y_3$, $(x_4, y_4$

The first three coefficient are identical to the one from the second polynomial

$$\begin{cases} a_o = y_1 \\ a_1 = \dfrac{y_2 - y_1}{x_2 - x_1} \\ a_2 = \dfrac{\dfrac{y_3 - y_2}{x_3 - x_2} - \dfrac{y_2 - y_1}{x_2 - x_1}}{x_3 - x_1} \\ a_3 = \dfrac{\dfrac{\left(\dfrac{y_4 - y_3}{x_4 - x_3} - \dfrac{y_3 - y_2}{x_3 - x_2}\right)}{x_4 - x_2} - \dfrac{\left(\dfrac{y_3 - y_2}{x_3 - x_2} - \dfrac{y_2 - y_1}{x_2 - x_1}\right)}{x_3 - x_1}}{x_4 - x_1} \end{cases}$$

3.3.4 General form of Newton's interpolating polynomials

The preceding analysis can be generalized to fit an nth-order polynomial to $n+1$ data points. The nth-order polynomial is

$$f_n(x) = a_o + a_1(x - x_1) + ... + a_n(x - x_1)(x - x_2)(x - x_n)$$

As was done previously with the linear and quadratic interpolations, data points can be used to evaluate the coefficients a_o, $a_1, ...,$ a_n. For an nth-order polynomial, $n + 1$ data points are required: $[x_1, f(x_1)]$, $[x_2, f(x_2)]...$, $[x_n, f(x_n)]$. We use these data points and the following equations to evaluate the coefficients:

$$\begin{cases} a_o = f(x_1) \\ a_1 = f[x_2, x_1] \\ \\ a_n = f[x_{n+1}, x_m, ..., \ x_1] \end{cases}$$

where the bracketed function evaluations are finite divided differences. For example, the first finite divided difference is represented generally as

$$f[x_i, x_j] = \frac{f(x_i) - f(x_j)}{x_i - x_j}$$

The second finite divided difference, which represents the difference of two first divided differences, is expressed generally as

$$f[x_i, x_j, x_k] = \frac{f[x_i, x_j] - f[x_j, x_k]}{x_i - x_k}$$

Similarly, the nth finite divided difference is

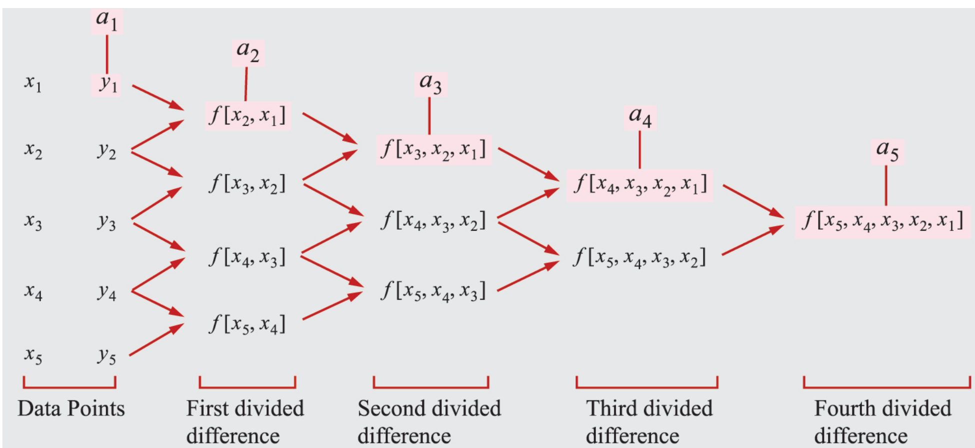
$$f[x_{n+1}, x_n, ..., x_1] = \frac{f[x_{n+1}, x_n, ..., x_2] - f[x_n, ..., x_1]}{x_{n+1} - x_1}$$

These differences can be used to evaluate the coefficients a_i to yield the interpolating polynomial

$$f_n(x) = f(x_1)+(x-x_1)f[x_2, x_1]+(x-x_1)(x-x_2)f[x_3, x_2, x_1]+.....+(x-x_1)(x-x_2)(x-x_n)f[x_{n+1},\ x_n, ...,\ x_1]$$

which is called Newton's 'interpolating polynomial. It should be noted that it is not necessary that the data points used be equally spaced or that the abscissa values necessarily be in ascending order.

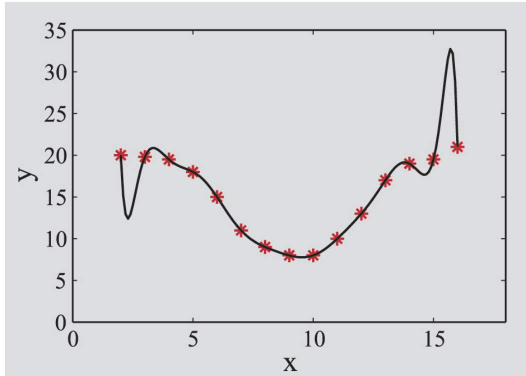
The procedure for finding the coefficients by using divided differences can be followed in a divided difference table as the one shown in the figure below for $n = 5$



- The spacings between the data points do not have to be the same.
- For a given set of n points, once the coefficients a_1 through a_n are determined, they can be used for interpolation at any point between the data points.
- After the coefficients a_1 through a_n are determined (for a given set of n points), additional data points can be added (they do not have to be in order), and only the additional coefficients have to be determined.

3.4 Spline interpolation

In the previous sections, nth-order polynomials were used to interpolate between $n + l$ data points. For example, for eight points, we can derive a perfect seventh-order polynomial. This curve would give exact values at the points and estimated values between the points. When the number of points is small such that the order of the polynomial is low, typically the interpolated values are reasonably accurate. However, large errors might occur when a high-order polynomial is used for interpolation involving a large number of points. This is shown in the figure below where a polynomial of 15th order is used for interpolation with a set of 16 data points. An alternative approach is to apply lower-order polynomials to subsets of data points. Such connecting polynomials are called **spline functions**.



In this section, simple linear functions will first be used to introduce some basic concepts and problems associated with spline interpolation. Then we derive an algorithm for fitting quadratic splines to data. Finally, we present material on the cubic spline, which is the most common and useful version in engineering practice.

3.4.1 Linear spline

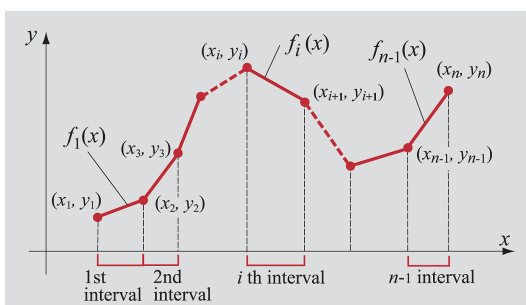
The simplest connection between two points is a straight line. The first-order splines for a group of ordered data points can be defined as a set of linear functions,

$$\begin{cases} f(x) = f(x_o) + m_o(x - x_o) & x_o \le x \le x_1 \\ f(x) = f(x_1) + m_1(x - x_1) & x_1 \le x \le x_2 \\ \\ \\ f(x) = f(x_{n-1}) + m_{n-1}(x - x_{n-1}) & x_{n-1} \le x \le x_n \end{cases}$$

where m_i is the slope of the straight line connecting the points

$$m_i = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}$$

These equations can be used to evaluate the function at any point between x_o and x_n by first locating the interval within which the point lies. Then the appropriate equation is used to determine the function value within the interval. The method is obviously identical to linear interpolation.



Visual inspection the primary disadvantage of first-order splines is that they are not smooth. In essence, at the data points where two splines meet (called a **knot**), the slope changes abruptly. In formal terms, the first derivative of the function is discontinuous at these points. This deficiency is overcome by using higher-order polynomial splines that ensure smoothness at the knots by equating derivatives at these points, as discussed in the next section.

3.4.2 Quadratic spline

To ensure that the mth derivatives are continuous at the knots, a spline of at least $m+1$ order must be used. Quadratic splines that ensure continuous first derivative at the knots and third-order polynomials or cubic splines that ensure continuous first and second derivatives are most frequently used in practice. We first illustrate the concept of spline interpolation using second-order polynomials. Although quadratic splines do not ensure equal second derivatives at the knots, they serve nicely to demonstrate the general procedure for developing higher-order splines.

The objective in quadratic splines is to derive a second-order polynomial for each interval between data points. The polynomial for each interval can be represented generally as

$$f_i(x) = a_i x^2 + b_i x + c_i$$

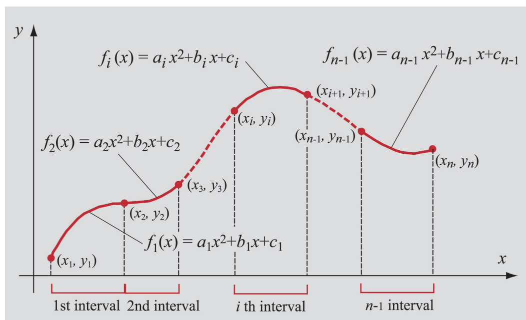


Fig.13

For n data points (i = 1, 2, . . . , n), there are $n-1$ intervals and, consequently, $3n-3$ unknown constants (the as, bs, and cs) to evaluate. Therefore, $3n-3$ equations or conditions are required to evaluate the unknowns. These are:

- At the end points

$$a_1 x_1^2 + b_1 x_1 + c_1 = f(x_1)$$

$$a_{n-1} x_n^2 + b_{n-1} x_n + c_{n-1} = f(x_n)$$

- For $i = 1$ to $i = n-2$

$$a_i x_{i+1}^2 + b_i x_{i+1} + c_i = f(x_{i+1})$$

$$a_{i+1} x_{i+1}^2 + b_{i+1} x_{i+1} + c_{i+1} = f(x_{i+1})$$

  These equations provide $2n-2$ conditions.

- The first derivatives at the interior knots must be equal. The condition can be represented generally as

$$2a_i x_{i+1} + b_i = 2a_{i+1} x_{i+1} + b_{i+1}$$

  for $i = 1$ to $n-2$. This provides another $n-2$ conditions for a total of $2n-2+n2 = 3n-4$. Because we have 3n-3 unknowns, we are one condition short. Unless we have some additional information regarding the functions or their derivatives, we must make an arbitrary choice to successfully compute the constants. Although there are a number of different choices that can be made, we select the following: Assume that the second derivative is zero at the first point. Because the second derivative is $2a_1$, this condition can be expressed mathematically as

$$a_1 = 0$$

The visual interpretation of this condition is that the first two points will be connected by a straight line.

3.4.3 Cubic spline based on Standard form polynomials

Cubic spline are third order polynomials connecting the end points of each interval. Cubic splines are continuous at the joints between intervals and so are the first and second derivatives. For n points we have $n-1$ intervals. In each interval $1 < i < n-1$ the polynomial id of the form

$$f_i(x) = a_i x^3 + b_i x^2 + c_i x + d_i$$

The conditions to be satisfied are

1. for $1 \le i \le n-1$ $f_i(x_i) = y_i$ and $f_i(x_{i+1}) = y_{i+1}$. That is a total of $2n-2$ equations
2. for the internal points ($n-2$ to be exact), the first derivative is continuous $f_i'(x_{i+1}) = f_{i+1}'(x_{i+1})$. That is a total of $n-2$ equations
3. for the internal points ($n-2$ to be exact), the second derivative is continuous $f_i''(x_{i+1}) = f_{i+1}''(x_{i+1})$. That is a total of $n-2$ equations
4. all together now we have $4n-6$ equations for $4n-4$ unknowns. Two more equations can be added by requiring the teh second derivative be zero at $f_1''(x_1) = 0$ and $f_{n-1}''(x_n) = 0$.

3.4.4 Cubic spline based on Lagrange form polynomials

Recall that when choosing cubic splines we enforce continuity of the first and second derivatives at the knots. The first derivative ($f'(x)$), which is a polynomial of second order satisfies $f_i'(x_{i+1}) = f_{i+1}'(x_{i+1})$. The second derivative, which is a first order polynomial satisfies $f_i''(x_{i+1}) = f_{i+1}''(x_{i+1})$. In this regard if we choose to write the second derivative in the lagrange form we would have

$$f_i''(x) = f_i''(x_i)\left(\frac{x - x_{i+1}}{x_i - x_{i+1}}\right) + f_i''(x_{i+1})\left(\frac{x - x_i}{x_{i+1} - x_i}\right)$$

Integrating this expression twice and using the fact that $f_i(x_i) = y_i$ and $f_i(x_{i+1}) = y_{i+1}$ reduces to for $x_i \le x \le x_{i+1}$ and $i = 1, 2, 3, ..., n-1$

$$f_i(x) = f_i''(x_i)\frac{(x_{i+1} - x)^3}{6\ (x_{i+1} - x_i)} + f_i''(x_{i+1})\frac{(x - x_i)^3}{6\ (x_{i+1} - x_i)} +$$

$$\left[\frac{y_i}{x_{i+1} - x_i} - \frac{f_i''(x_i)(x_{i+1} - x_i}{6}\right](x_{i+1} - x) +$$

$$\left[\frac{y_{i+1}}{x_{i+1} - x_i} - \frac{f_i''(x_{i+1})(x_{i+1} - x_i}{6}\right](x - x_i)$$

The number of unknowns in each interval is two $f_i''(x_i)$ and $f_i''(x_{i+1})$. A total of $2(n-1) = 2n-2$ unknowns. However since

$$f_i''(x_{i+1}) = f_{i+1}''(x_{i+1})$$

This is a set of $n-2$ equation for n points. the number reduces to n unknowns.
Using the continuity of $f'(x)$ at the knots produces a set of $n-2$ additional equations of the form

$$f_i'(x_{i+1}) = f_{i+1}'(x_{i+1})$$

$$(x_{i+1} - x_i)f_i''(x_i) + 2(x_{i+2} - x_i)f_i''(x_{i+1}) + (x_{i+2} - x_{i+1})f_{i+1}''(x_{i+2})$$
$$= 6\left[\frac{y_{i+2} - y_{i+1}}{x_{i+2} - x_{i+1}} - \frac{y_{i+1} - y_i}{x_{i+1} - x_i}\right]$$

The last two additional equations are obtained by setting

$$f_1''(x_1) = 0 \quad \textit{and} \quad f_{n-1}''(x_n) = 0$$