

CHAPTER-V CURVE FITTING

5.1 Introduction

Data is often given for discrete values along a continuum. However estimates of points between these discrete values may be required. One way to do this is to formulate a function to fit these values approximately. This application is called curve fitting. There are two general approaches to curve fitting.

The first is to derive a single curve that represents the general trend of the data. One method of this nature is the *least-squares regression*. The second approach is *interpolation* which is a more precise one. The basic idea is to fit a curve or a series of curves that pass directly through each of the points.

5.2 Least squares regression

5.2.1 Linear regression

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

$$y = a_0 + a_1x + e$$

where a_0 , and a_1 are coefficients representing the y-intercept and the slope of the line respectively while e is the *error* or *residual* between the model and the observations, which can be represented as:

$$e = y - a_0 - a_1x$$

Thus the error is the discrepancy between the true value of y (observed value) and the approximate value $a_0 + a_1x$, predicted by the linear equation. Any strategy of approximating a set of data by a linear equation (best fit) should minimize the sum of residuals. The least squares fit of straight line minimizes the sum of the squares of the residuals.

$$S_r = \sum_{i=1}^n e_i^2 = \sum_{i=1}^n (y_{i,measured} - y_{i,model})^2 = \sum_{i=1}^n (y_i - a_0 - a_1x_i)^2 \quad (4.1)$$

To determine the values of a_0 and a_1 , differentiate (4.1) with respect to each coefficient

$$\begin{aligned} \frac{\partial S_r}{\partial a_0} &= -2 \sum (y_i - a_0 - a_1x_i) \\ \frac{\partial S_r}{\partial a_1} &= -2 \sum [(y_i - a_0 - a_1x_i)x_i] \end{aligned} \quad (4.2)$$

Setting the derivatives equal to zero will result in a minimum S_r . The equations can then be expressed as

$$\begin{aligned}\sum y_i - \sum a_o - \sum a_1 x_i &= 0 \\ \sum y_i x_i - \sum a_o x_i - \sum a_1 x_i^2 &= 0\end{aligned}\quad (4.3)$$

Solving for a_o and a_1 simultaneously

$$a_1 = \frac{n \sum x_i y_i - \sum x_i \sum y_i}{n \sum x_i^2 - (\sum x_i)^2} \quad (4.4)$$

$$a_o = \bar{y} - a_1 \bar{x} \quad (4.5)$$

where \bar{y} and \bar{x} are the means of y and x , respectively

Test of goodness of fit

Any line other than the one derived above gives larger sum of the squares of the residuals. The square of the residual represents the square of the vertical distance between the data and the straight line. The standard deviation of the regression line can be determined by *the standard error of the estimate*

$$S_{y/x} = \sqrt{\frac{S_r}{n-2}} \quad (4.6)$$

$S_{x/y}$ quantifies the spread around the regression line. This concept can be used to quantify the goodness of fit. This is particularly necessary when comparing several regressions. To do this, we return to the original data and determine *the total sum of squares* around the mean of the dependent variable i.e., y . This quantity is designated S_t and represents the magnitude of the residual error associated with the dependent variable prior to the regression. Recall that S_r characterizes the residual error that remains after the regression.

The difference between the two quantities, $S_t - S_r$, quantifies the improvement or error reduction due to describing the data in terms of a straight line rather than an average value. The normalized difference yields

$$r^2 = \frac{S_t - S_r}{S_t} \quad (4.7)$$

where r^2 is called the coefficient of determination and r is the correlation coefficient. For linear regression r is given by

$$r = \frac{n \sum x_i y_i - (\sum x_i)(\sum y_i)}{\sqrt{n \sum x_i^2 - (\sum x_i)^2} \sqrt{n \sum y_i^2 - (\sum y_i)^2}} \quad (4.8)$$

For a perfect fit $r=1$ and r should be close to 1 for good fit.

Caution!

Although the correlation coefficient provides a handy measure of goodness of fit, care should be taken not to take for granted that the value of r close to one necessarily mean "good fit". It is always advised to plot the data along with the regression curve because it is possible to obtain relatively high value of r when the relationship between y and x is not linear.

Linearization of non-linear relationships

Linear regression provides a powerful technique for fitting a "best" line to a data. However it is predicated on the fact that the relationship between the independent and dependent variables is linear. But usually this is not the case. Visual inspection of the plot of the data will provide useful information whether linear regression is acceptable. In situations where linear regression is inadequate other methods such as polynomial regression are appropriate. For others, transformations can be used to express the data in a form that is compatible with linear regression.

The followings are examples of functions which can be linearized

i Exponential functions

$$y = a_1 e^{b_1 x} \quad \text{where } a_1 \text{ and } b_1 \text{ are constants}$$

This function can be linearized by taking the natural logarithm of both sides of the equation

$$\ln y = \ln a_1 + b_1 x \ln e$$

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

the plot of $\ln y$ versus $\ln x$ will yield a straight line with a slope of b_1 and an intercept of $\ln a_1$.

ii Power functions

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

where a_2 and b_2 are constant coefficients.

This equation can be linearized by taking its base 10 logarithm to give

$$\log y = b_2 \log x + \log a_2$$

the plot of $\log y$ versus $\log x$ will yield a straight line with a slope of b_2 and an intercept of $\log a_2$.

iii Saturation growth rate equation

$$y = a_3 \frac{x}{b_3 + x}$$

this equation can be linearized by inverting to give

$$\frac{1}{y} = \frac{b_3}{a_3} \frac{1}{x} + \frac{1}{a_3}$$

the plot of $1/y$ versus $1/x$ will be linear, with a slope of b_3/a_3 and an intercept of $1/a_3$.

In their transformed forms, these models are fit using linear regression in order to evaluate the constant coefficients. Then they can be transformed back to their original state and used for predictive purposes.

Example Find the standard "least squares line" $Y = aX + b$ for the data points $(-1, 10), (0, 9), (1, 7), (2, 5), (3, 4), (4, 3), (5, 0), (6, -1)$.

$$\{X_k\} = \{-1, 0, 1, 2, 3, 4, 5, 6\}$$

$$\{Y_k\} = \{10, 9, 7, 5, 4, 3, 0, -1\}$$

$$\sum_{k=1}^n X_{[k]} = 20$$

$$\sum_{k=1}^n Y_{[k]} = 37$$

$$\sum_{k=1}^n (X_{[k]})^2 = 92$$

$$\sum_{k=1}^n X_{[k]} Y_{[k]} = 25$$

$$Y = b + aX$$

The normal equations for finding the coefficients a and b are:

$$\begin{pmatrix} 92 & 20 \\ 20 & 8 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 25 \\ 37 \end{pmatrix}$$

The solution is

$$\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} -\frac{45}{28} \\ \frac{121}{14} \end{pmatrix}$$

$$a = -\frac{45}{28}$$

$$b = \frac{121}{14}$$

The least squares line is

$$Y = \frac{121}{14} - \frac{45}{28}X = 8.64286 - 1.60714x$$

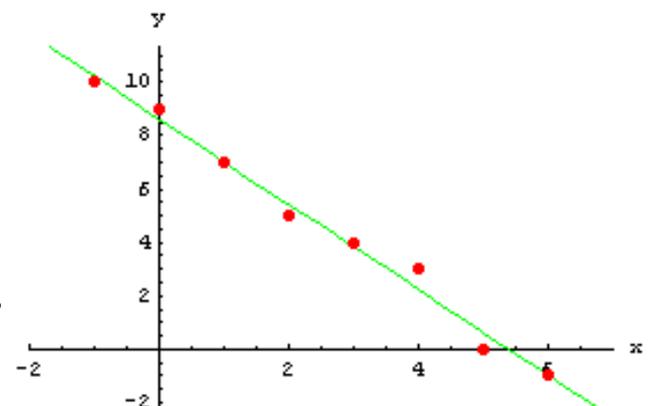
And the graph will look like:

Points = $\{(-1, 10), (0, 9), (1, 7), (2, 5), (3, 4), (4, 3), (5, 0), (6, -1)\}$.

The 'least squares line' is

$$Y = \frac{121}{14} - \frac{45}{28}X = 8.64286 - 1.60714x$$

Error (the sum of residual's squared)



$$\sum_{k=1}^n (Y_{[k]} - aX_{[k]} - b)^2 = \frac{39}{28} = 1.39286$$

4.2.2 Polynomial regression

Some engineering data, although exhibiting marked pattern, is poorly represented by straight line. For these cases a curve would be better suited to fit the data. One of the possible ways of solving this kind of problems is to fit the data by a polynomial function. This is called polynomial regression. The least squares method can be extended to fit data to a higher-order polynomial. Suppose we want to fit a second order polynomial:

$$y = a_0 + a_1x + a_2x^2 + e$$

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

$$S_r = \sum (y_i - a_0 - a_1x_i - a_2x_i^2)^2$$

Taking the derivative of the above equation with respect to each unknown coefficients of the polynomial gives

$$\frac{\partial S_r}{\partial a_0} = -2 \sum (y_i - a_0 - a_1x_i - a_2x_i^2)$$

$$\frac{\partial S_r}{\partial a_1} = -2 \sum x_i (y_i - a_0 - a_1x_i - a_2x_i^2)$$

$$\frac{\partial S_r}{\partial a_2} = -2 \sum x_i^2 (y_i - a_0 - a_1x_i - a_2x_i^2)$$

Setting these equations equal to zero and rearranging to develop the following set of equations

$$\begin{aligned} na_0 + (\sum x_i)a_1 + (\sum x_i^2)a_2 &= \sum y_i \\ (\sum x_i)a_0 + (\sum x_i^2)a_1 + (\sum x_i^3)a_2 &= \sum x_i y_i \\ (\sum x_i^2)a_0 + (\sum x_i^3)a_1 + (\sum x_i^4)a_2 &= \sum x_i^2 y_i \end{aligned}$$

Solving for the coefficients of the quadratic regression is equivalent to solving three simultaneous linear equations. The techniques for solving these problems are discussed in chapter two.

This discussion can easily be extended to an m^{th} order polynomial as

$$y = a_0 + a_1x + a_2x^2 + \dots + a_mx^m + e$$

Thus determination of the coefficients of an m^{th} order polynomial is equivalent to solving a system of $m+1$ simultaneous linear equations. For this case the standard error is formulated as

$$S_{y/x} = \sqrt{\frac{S_r}{n - (m + 1)}}$$

Example Find the standard "least squares parabola" $a + bx + cx^2$ for the data points

$(-1, 10), (0, 9), (1, 7), (2, 5), (3, 4), (4, 3), (5, 0), (6, -1)$.

$$y = a + bx + cx^2$$

The normal equations for finding the coefficients a and b are:

$$\begin{pmatrix} 8 & 20 & 92 \\ 20 & 92 & 440 \\ 92 & 440 & 2276 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} 32 \\ 64 \\ 400 \end{pmatrix}$$

The solution is

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} \frac{118}{21} \\ -\frac{26}{7} \\ \frac{2}{3} \end{pmatrix}$$

$$a = \frac{118}{21}$$

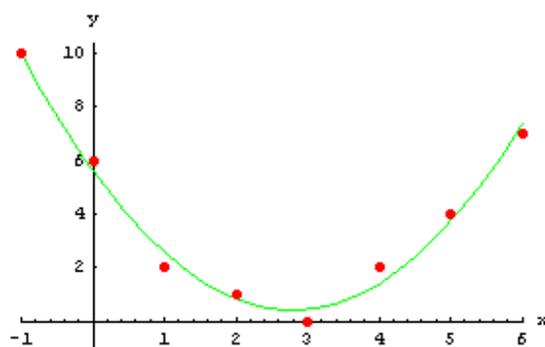
$$b = -\frac{26}{7}$$

$$c = \frac{2}{3}$$

The 'least squares parabola' is

$$y = \frac{118}{21} - \frac{26x}{7} + \frac{2x^2}{3} = 5.61905 - 3.71429x + 0.666667x^2$$

Points = $\{(-1, 10), (0, 6), (1, 2), (2, 1), (3, 0), (4, 2), (5, 4), (6, 7)\}$



4.2.3 Multiple linear regression

A useful extension of linear regression is the case where y is a linear function of more than one variable, say x_1 and x_2 ,

$$y = a_o + a_1x_1 + a_2x_2 + e$$

Such an equation is useful when fitting experimental data where the variable being studied is a function of two other variables.

In the same manner as the previous cases the best values of the coefficients are determined by setting the sum of the squares of the residuals to a minimum.

$$S_r = \sum (y_i - a_o - a_1x_{1i} - a_2x_{2i})^2$$

Differentiating with respect to the unknown coefficients, we have

$$\frac{\partial S_r}{\partial a_o} = -2 \sum (y_i - a_o - a_1x_{1i} - a_2x_{2i})$$

$$\frac{\partial S_r}{\partial a_1} = -2 \sum x_{1i} (y_i - a_o - a_1x_{1i} - a_2x_{2i})$$

$$\frac{\partial S_r}{\partial a_2} = -2 \sum x_{2i} (y_i - a_o - a_1x_{1i} - a_2x_{2i})$$

The coefficients yielding the minimum sum of the residuals are obtained by setting the partial derivatives equal to zero and expressing the result in a matrix form as

$$\begin{bmatrix} n & \sum x_{1i} & \sum x_{2i} \\ \sum x_{1i} & \sum x_{1i}^2 & \sum x_{1i}x_{2i} \\ \sum x_{2i} & \sum x_{1i}x_{2i} & \sum x_{2i}^2 \end{bmatrix} \begin{Bmatrix} a_o \\ a_1 \\ a_2 \end{Bmatrix} = \begin{Bmatrix} \sum y_i \\ \sum x_{1i}y_i \\ \sum x_{2i}y_i \end{Bmatrix}$$

The above case can be extended to m dimension,

$$y = a_o + a_1x_1 + a_2x_2 + \dots + a_mx_m + e$$

where the standard error is formulated as

$$S_{y/x} = \sqrt{\frac{S_r}{n - (m - 1)}}$$

The coefficient of determination is computed as in equation (4.7).

Multiple linear regression has utility in the derivation of power equations of the general form

$$y = a_o x_1^{a_1} x_2^{a_2} \dots x_m^{a_m}$$

Transformation of this form of equation can be achieved by taking the logarithm of the equation.

4.2.4 General linear least squares

In the preceding discussions we have seen three types of regression: linear, polynomial and multiple linear. All these belong to the general linear least squares model given by

$$y = a_0 z_0 + a_1 z_1 + a_2 z_2 + \dots + a_m z_m + e \quad (4.9)$$

where $z_0, z_1, z_2, \dots, z_m$ are $m+1$ different functions. For multiple linear regression $z_0 = 1, z_1 = x_1, z_2 = x_2, \dots, z_m = x_m$. For polynomial regression, the z 's are simple monomials as in $z_0 = 1, z_1 = x, z_2 = x^2, \dots, z_m = x^m$.

The terminology linear refers only to the model's dependence on its parameters i.e., the a 's. Equation (4.9) can be expressed in a matrix form as

$$\{y\} = [z]\{A\} + \{E\}$$

where $[z]$ is a matrix of calculated values of the z functions at the measured values of the independent variables.

$$\begin{bmatrix} z_{01} & z_{11} & \dots & \dots & z_{m1} \\ z_{02} & z_{12} & \dots & \dots & z_{m2} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ z_{0n} & z_{1n} & \cdot & \cdot & z_{mn} \end{bmatrix}$$

where m is the number of variables in the model and n is the number of data points. Because $n \geq m+1$, most of the time $[z]$ is not a square matrix.

The column vector $\{y\}$ contains the observed values of the dependent variable.

$$\{y\}^T = [y_1, y_2, \dots, y_n]$$

The column vector $\{A\}$ contains the unknown coefficients

$$\{A\}^T = [a_0, a_1, \dots, a_m]$$

the column vector $\{E\}$ contains the residuals

$$\{E\}^T = [e_1, e_2, \dots, e_n]$$

The sum of the squares of the residuals can be defined as

$$S_r = \sum_{i=1}^n \left(y_i - \sum_{j=0}^m a_j z_{ji} \right)^2$$

This quantity can be minimized by taking its partial derivative with respect to each of the coefficients and setting the resulting equation equal to zero. The outcome of this process can be expressed in matrix form as

$$[[Z]^T [Z]]\{A\} = \{[Z]^T \{y\}\}$$

4.3 Interpolation

In many engineering applications it is required to estimate intermediate values between precise data points. The most common method used is the polynomial interpolation. For $n+1$ data points, there is a unique polynomial of order n that passes through all the points.

$$f(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$$

Polynomial interpolation involves the determination of that unique n^{th} order polynomial that fits $n+1$ data points. This formula can then be used to estimate intermediate values. Although there is only one n^{th} order polynomial that fits $n+1$ points, there are a variety of mathematical formats in which this polynomial can be expressed. The most widely used alternatives are the Newton and Lagrange interpolating polynomials.

4.3.1 Newton's Divided Difference Interpolating Polynomials

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

$$\frac{f_1(x) - f(x_0)}{x - x_0} = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

which can be rearranged to yield

$$f_1(x) = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0)$$

This is the linear interpolation formula. The notation $f_1(x)$ indicates that this is a first order polynomial. The term $[f(x_1) - f(x_0)] / (x_1 - x_0)$ is the finite difference approximation of the first derivative. The smaller the interval between the data points, the better the approximation.

Approximation of a curve with a straight line can result in significant errors. A better approximation can be achieved if some curvature is introduced into the line connecting the points. If three points are available, a second order polynomial can be plotted through the data points. This is called a quadratic interpolating polynomial. A convenient form for this purpose is

$$f_2(x) = b_0 + b_1(x - x_0) + b_2(x - x_0)(x - x_1)$$

b_0 can be solved for by setting $x = x_0$, in which case $b_0 = f_2(x_0)$

then $f_2(x)$ can be evaluated at $x = x_1$ to solve for b_1

$$b_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

evaluating $f_2(x)$ at $x = x_2$ gives

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

Note that b_1 represents the slope of the line between x_0 and x_1 and b_2 is similar to the finite difference approximation of the second derivative.

4.3.2 General form of Newton's Interpolating Polynomials

The preceding analysis can be generalized to fit an n^{th} order polynomial to $n+1$ data points. The n^{th} order polynomial is

$$f_n(x) = b_0 + b_1(x-x_0) + b_2(x-x_0)(x-x_1) + \dots + b_n(x-x_0)(x-x_1)\dots(x-x_{n-1})$$

The data points can be used to evaluate the coefficients $b_0, b_1, b_2, \dots, b_n$.

$$b_0 = f(x_0)$$

$$b_1 = f[x_1, x_0]$$

$$b_2 = f[x_2, x_1, x_0]$$

.

.

$$b_n = f[x_n, x_{n-1}, \dots, x_1, x_0]$$

where the bracketed function evaluations are finite divided differences. For example the first finite divided difference is given by

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

The second finite divided difference is expressed by

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

Similarly the n -th finite divided difference is

$$f[x_n, x_{n-1}, \dots, x_1, x_0] = \frac{f[x_n, x_{n-1}, \dots, x_1] - f[x_{n-1}, x_{n-2}, \dots, x_0]}{x_n - x_0}$$

These differences can be used to evaluate the coefficients and then can be substituted to yield the interpolating polynomial

$$f_n(x) = f(x_0) + (x-x_0)f[x_1, x_0] + (x-x_0)(x-x_1)f[x_2, x_1, x_0] + \dots + (x-x_0)(x-x_1)\dots(x-x_{n-1})f[x_n, x_{n-1}, \dots, x_0] \quad (4.10)$$

This is called the Newton's divided difference interpolating polynomial. It does not require that the data be put in ascending order or the interval between data points to be equal.

Example 1

Form the Newton polynomials of degree $n = 1, 2, 3, 4,$ and 5 for the function $f[x] = \cos[x]$ over the interval $[x_0, x_n]$ using equally spaced nodes selected from the following list

$$\{(x_k, y_k)_{k=0}^5\} = \left\{ \left(0, 1\right), \left\{\frac{1}{5}, \cos\left[\frac{1}{5}\right]\right\}, \left\{\frac{2}{5}, \cos\left[\frac{2}{5}\right]\right\}, \left\{\frac{3}{5}, \cos\left[\frac{3}{5}\right]\right\}, \left\{\frac{4}{5}, \cos\left[\frac{4}{5}\right]\right\}, \left(1, \cos[1]\right) \right\}$$

a) Nodes = $\{(0, 1), (0.2, 0.980067)\}$

The interval for interpolation is $[0.0, 0.2]$.

$$p_1[x] = 1 - 0.0996671x$$

b) Nodes = $\{(0, 1), (0.2, 0.980067), (0.4, 0.921061)\}$

The interval for interpolation is $[0.0, 0.4]$.

$$p_2[x] = 1 - 0.0996671x - 0.488402(-0.2+x)x$$

$$p_2[x] = 1 - 0.00198671x - 0.488402x^2$$

c) Nodes = {{0, 1}, {0.2, 0.980067}, {0.4, 0.921061}, {0.6, 0.825336}}

The interval for interpolation is [0.0,0.6].

$$p_3[x] = 1 - 0.0996671x - 0.488402(-0.2+x)x + 0.0490076(-0.4+x)(-0.2+x)x$$

$$p_3[x] = 1 + 0.0019339x - 0.517807x^2 + 0.0490076x^3$$

d) Nodes = {{0, 1}, {0.2, 0.980067}, {0.4, 0.921061}, {0.6, 0.825336}, {0.8, 0.696707}}

The interval for interpolation is [0.0,0.8].

$$p_4[x] = 1 - 0.0996671x - 0.488402(-0.2+x)x + 0.0490076(-0.4+x)(-0.2+x)x + 0.0381225(-0.6+x)(-0.4+x)(-0.2+x)x$$

$$p_4[x] = 1 + 0.000104026x - 0.501033x^2 + 0.00326069x^3 + 0.0381225x^4$$

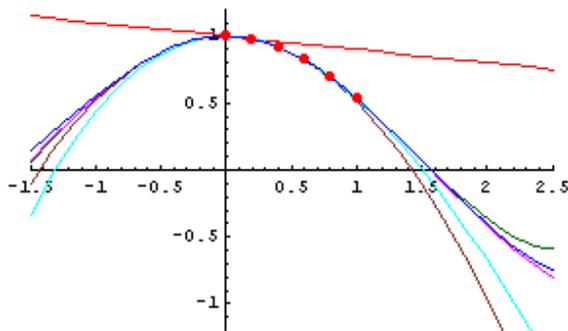
e)

Nodes = {{0, 1}, {0.2, 0.980067}, {0.4, 0.921061}, {0.6, 0.825336}, {0.8, 0.696707}, {1., 0.540302}}

The interval for interpolation is [0.0,0.8].

After simplification

$$p_5[x] = 1 - 0.0000481164x - 0.499448x^2 - 0.00228618x^3 + 0.0460466x^4 - 0.00396205x^5$$



$$f[x] = \text{Cos}[x]$$

$$p_1[x] = 1 - 0.0996671x$$

$$p_2[x] = 1 - 0.0996671x - 0.488402(-0.2+x)x$$

$$p_3[x] = 1 - 0.0996671x - 0.488402(-0.2+x)x + 0.0490076(-0.4+x)(-0.2+x)x$$

$$p_4[x] = 1 + 0.000104026x - 0.501033x^2 + 0.00326069x^3 + 0.0381225x^4$$

$$p_5[x] = 1 - 0.0000481164x - 0.499448x^2 - 0.00228618x^3 + 0.0460466x^4 - 0.00396205x^5$$

Errors in Newton's Interpolating polynomials

The structure of eq(4.10) is similar to the Taylor series expansion in the sense that terms are added sequentially to capture the higher order behavior of the underlying function. These terms are finite difference approximations of the higher order derivatives. Thus if the true underlying function is n^{th} order polynomial, the n^{th} order interpolating polynomial based on $n+1$ data points will yield exact results.

As in the Taylor series, a formulation of the truncation error can be obtained. Recall that the truncation error in Taylor series is generally expressed as

$$R_n = \frac{f^{(n+1)}(\xi)}{(n+1)!} (x_{i+1} - x_i)^{n+1}$$

where ξ is somewhere in the interval x_i to x_{i+1} . For the n^{th} order interpolating polynomial, an analogous relationship for error is

$$R_n = \frac{f^{(n+1)}(\xi)}{(n+1)!} (x - x_0)(x - x_1)\dots(x - x_n)$$

where ξ is somewhere in the interval containing the unknown and the data. An alternative formulation which uses a finite difference to approximate the $(n+1)^{\text{th}}$ derivative is

$$R_n = f[x, x_n, x_{n-1}, \dots, x_0](x - x_0)(x - x_1)\dots(x - x_n)$$

where $f[x, x_n, x_{n-1}, \dots, x_0]$ is the $(n+1)^{\text{th}}$ finite divided difference. The above equation can be solved if an additional point $f(x_{n+1})$ is available in which case

$$R_n \cong f[x_{n+1}, x_n, x_{n-1}, \dots, x_0](x - x_0)(x - x_1)\dots(x - x_n)$$

The error estimate for the n^{th} order polynomial is equivalent to the difference between the $(n+1)^{\text{th}}$ order and the n -th order prediction.

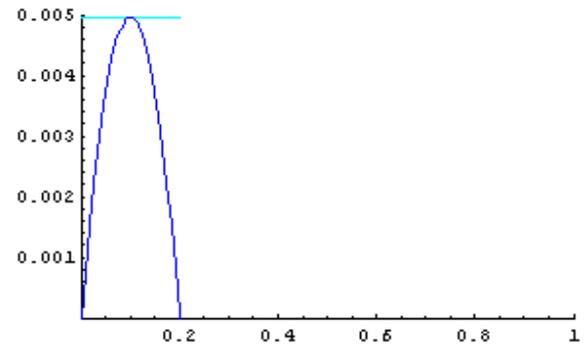
$$R_n = f_{n+1}(x) - f_n(x)$$

The validity of this approach lies on the fact that the series is strongly convergent. For such situations the $(n+1)^{\text{th}}$ order prediction would be closer to the true value than the n^{th} order prediction.

Example 2 Investigate the error for the Newton polynomial approximations Example 1.

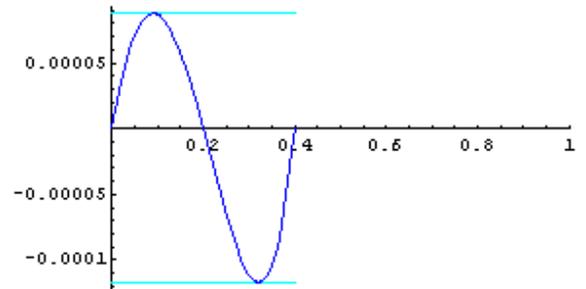
a)

$f[x] = \text{Cos}[x]$
 The interval for interpolation is $[0.0, 0.2]$.
 Graph of the error $e_1[x] = f[x] - p_1[x]$
 Extrema for $e_1[x]$ is $\{0.00497089\}$
 $|e_1[x]| \leq 0.00497089$

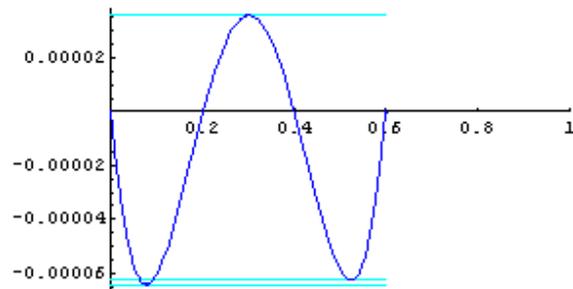


b)

$f[x] = \text{Cos}[x]$
 The interval for interpolation is $[0.0, 0.4]$.
 Graph of the error $e_2[x] = f[x] - p_2[x]$
 Extrema for $e_2[x]$ are $\{0.0000876038, -0.000116469\}$
 $|e_2[x]| \leq 0.000116469$

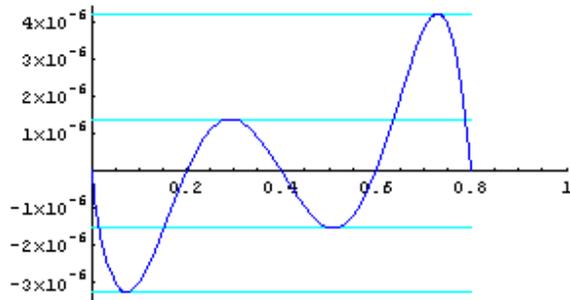


c)



$f[x] = \text{Cos}[x]$
 The interval for interpolation is $[0.0, 0.6]$.
 Graph of the error $e_3[x] = f[x] - p_3[x]$
 Extrema for $e_3[x]$ are $\{-0.0000642481, 0.0000357062, -0.0000624928\}$
 $|e_3[x]| \leq 0.0000642481$

d)



$$f[x] = \text{Cos}[x]$$

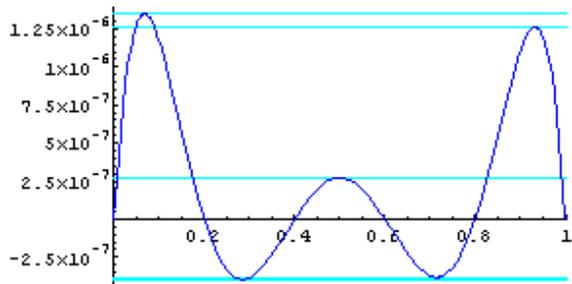
The interval for interpolation is $[0.0, 0.8]$.

Graph of the error $e_4[x] = f[x] - p_4[x]$

Extrema for $e_4[x]$ are $\{-3.25851 \times 10^{-6}, 1.40342 \times 10^{-6}, -1.52922 \times 10^{-6}, 4.23054 \times 10^{-6}\}$

$$|e_4[x]| \leq 4.23054 \times 10^{-6}$$

e)



$$f[x] = \text{Cos}[x]$$

The interval for interpolation is $[0.0, 1.0]$.

Graph of the error $e_5[x] = f[x] - p_5[x]$

Extrema for $e_5[x]$ are $\{1.34999 \times 10^{-6}, -3.97654 \times 10^{-7}, 2.72539 \times 10^{-7}, -3.84527 \times 10^{-7}, 1.26163 \times 10^{-6}\}$

$$|e_5[x]| \leq 1.34999 \times 10^{-6}$$

4.3.3 Lagrange Interpolation Polynomials

The Lagrange interpolating polynomial is a reformulation of the Newton polynomial that avoids the computation of divided differences. It can be represented as

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

where

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

where Π designates the "product of."
for linear interpolation

$$f_1(x) = \frac{x - x_1}{x_0 - x_1} f(x_0) + \frac{x - x_0}{x_1 - x_0} f(x_1)$$

For second order interpolating polynomials

$$f_2(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} f(x_0) + \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} f(x_1) + \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} f(x_2)$$

As in the Newton's method the Lagrange version has an estimated error of

$$R_n = f[x, x_n, x_{n-1}, \dots, x_0] \prod_{i=0}^n (x - x_i)$$

Thus if an additional point at $x=x+1$ is available an error estimate can be obtained. But because the finite differences are not employed as part of the Lagrange operation, this is rarely done. For cases where the order of the polynomial is not determined prior to interpolation, the Newton method has advantages because of the insight it provides into the behavior of the different order formulas.

Higher order polynomials tend to be ill conditioned; i.e. they are highly sensitive to round-off error. The same order goes for higher order polynomial regression.

Special care should be taken when interpolating polynomials are to be used for extrapolation. This arises because the point lies outside the base points used of the interpolation. The best results of interpolation are obtained when the unknown lies near the center of the base points. Subsequently when the point lies outside the range of the data set, the error in extrapolation can be very large.

Example Construct three interpolating polynomials of degree $n=1$ for the function

$$f[x] = \cos[x] \quad \text{over } [0,1]. \text{ Use the following sets of interpolation nodes.}$$

- Use the nodes $\{0.0, f[0.0]\}$ and $\{1.0, f[1.0]\}$.
- Use the nodes $\{0.1, f[0.1]\}$ and $\{0.9, f[0.9]\}$.
- Use the nodes $\{0.146447, f[0.146447]\}$ and $\{0.853553, f[0.853553]\}$.

a)

$$f[x] = \cos[x]$$

$$\text{Nodes} = \{(0., 1.), \{1., 0.540302\}\}$$

$$p_1[x] = -1. (-1. + x) + 0.540302 (0. + x)$$

b)

$$f[x] = \cos[x]$$

$$\text{Nodes} = \{(0.1, 0.995004), (0.9, 0.62161)\}$$

$$q_1[x] = -1.24376(-0.9 + x) + 0.777012(-0.1 + x)$$

c)

$$f[x] = \cos[x]$$

$$\text{Nodes} = \{(0.146447, 0.989296), (0.853553, 0.65731)\}$$

$$t_1[x] = -1.39908(-0.853553 + x) + 0.929577(-0.146447 + x)$$

Note that the three polynomials of degree $n = 1$ are different. The error in approximating $f[x]$ will also be different.

Exercise 1. Form several Lagrange polynomials of degree $n = 2, 3, 4,$ and 5 for the function $f[x] = \cos[x]$ over the interval $[0, 1]$ using $n+1$ equally spaced nodes. Then compare the four Lagrange polynomials.

Example 2. Investigate the error for the Lagrange polynomial approximations of degree $n = 2, 3, 4,$ and 5 in Exercise 1.

4.3.4 Spline Interpolation

Employing high-order polynomial functions can lead to erroneous results because of round-off error and overshoot. An alternative to this approach is to apply lower order polynomials to subsets of data points. Such connecting polynomials are called spline functions. The functions are so constructed that the connections between adjacent equations are visually smooth.

i. Linear splines

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,

$$\begin{aligned} f(x) &= f(x_0) + m_0(x-x_0) & x_0 \leq x \leq x_1 \\ f(x) &= f(x_1) + m_1(x-x_1) & x_1 \leq x \leq x_2 \\ &\vdots \\ f(x) &= f(x_{n-1}) + m_{n-1}(x-x_{n-1}) & x_{n-1} \leq x \leq x_n \end{aligned}$$

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

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

The disadvantage of first order splines is that they are not smooth. The slope changes abruptly at the knots i.e., the first derivative of the function is discontinuous at these points.

ii. Quadratic Splines

To ensure the m^{th} order derivatives are continuous at the knots, a spline of at least $m+1$ order must be used. Cubic splines ensure that the first and second derivatives are continuous at the knots. Third and higher derivatives may be discontinuous but cannot be detected visually and consequently are ignored.

The derivation of cubic splines can be facilitated by understanding the concepts of second order interpolating polynomial (quadratic spline).

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

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

for $n+1$ data points there are n intervals and consequently $3n$ unknown coefficients to evaluate. There fore, $3n$ equations are required to solve the problem. These equations are obtained from:

1. The function values of adjacent polynomials must be equal at the interior knots:

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

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

for $i = 2$ to n . Because only interior points are used, this provides $2n-2$ conditions.

2. The first and last functions must pass through the end points. This gives two additional equations:

$$a_1x_0^2 + b_1x_0 + c_1 = f(x_0)$$

$$a_nx_n^2 + b_nx_n + c_n = f(x_n)$$

3. The first derivatives at the interior knots must be equal

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

for $i = 2$ to n this provides $n-1$ conditions.

4. Assume that the second derivative is zero at the first point. Because the second derivative the quadratic equation is $2a_i$ this condition is mathematically expressed as

$$a_1 = 0$$

This means that the first two points are connected by a straight line.

iii. Cubic Splines

The objective in cubic splines is to derive a third order polynomial for each interval between knots, as in

$$f_i(x) = a_ix^3 + b_ix^2 + c_ix + d_i$$

For $n+1$ data points there are n intervals and therefore $4n$ unknowns. The required $4n$ conditions are obtained from:

1. the function values at the interior knots must be equal ($2n-2$ conditions)
2. the first and last functions must pass through the end points (2 conditions)
3. The third derivatives at the interior knots must be equal ($n-1$ conditions)
4. the second derivatives at the interior knots must be equal ($n-1$ conditions)
5. The second derivatives at the end knots are zero (2 conditions)

Alternative technique for cubic spline

The second derivatives of the cubic curves in each interval between the knots are straight lines. These lines can be represented by a first order Lagrange interpolating polynomial:

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

This equation can be integrated twice to yield an expression for $f_i(x)$. By applying the condition that the function values at the interior knots must be the same the following cubic equation results:

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

This equation contains only two unknowns—the second derivatives at the end of each interval. The second derivatives can be evaluated by invoking the condition that the first derivatives at the knots must be continuous. The above equation can be differentiated to give an expression for the first derivative. Setting the first derivatives for the (i-1)th and the ith intervals equal, we have

$$(x_i - x_{i-1})f''(x_{i-1}) + 2(x_{i+1} - x_{i-1})f''(x_i) + (x_{i+1} - x_i)f''(x_{i+1}) = \frac{6}{x_{i+1} - x_i}[f(x_{i+1}) - f(x_i)] + \frac{6}{x_i - x_{i-1}}[f(x_i) - f(x_{i-1})]$$

If this equation is written for all the interior knots, n-1 simultaneous equations result with n-1 unknowns.

Example Construct the natural cubic spline for the points

(0, 1), (1, 0), (2, 0), (3, 1), (4, 2), (5, 2), (6, 1), that has the endpoint constraints $S'''(0) = 0$ and $S'''(6) = 0$.

data points = {(0., 1.), {1., 0.), {2., 0.), {3., 1.), {4., 2.), {5., 2.), {6., 1.)}

The spline coefficients are

$$\begin{pmatrix} 1. & -1.2 & 0 & 0.2 \\ 0 & -0.6 & 0.6 & 0 \\ 0 & 0.6 & 0.6 & -0.2 \\ 1. & 1.2 & 0 & -0.2 \\ 2. & 0.6 & -0.6 & 0 \\ 2. & -0.6 & -0.6 & 0.2 \end{pmatrix}$$

