University of Jordan

Faculty of Engineering and Technology
Chemical Engineering Department
0905301 Numerical Methods in Chemical Engineering

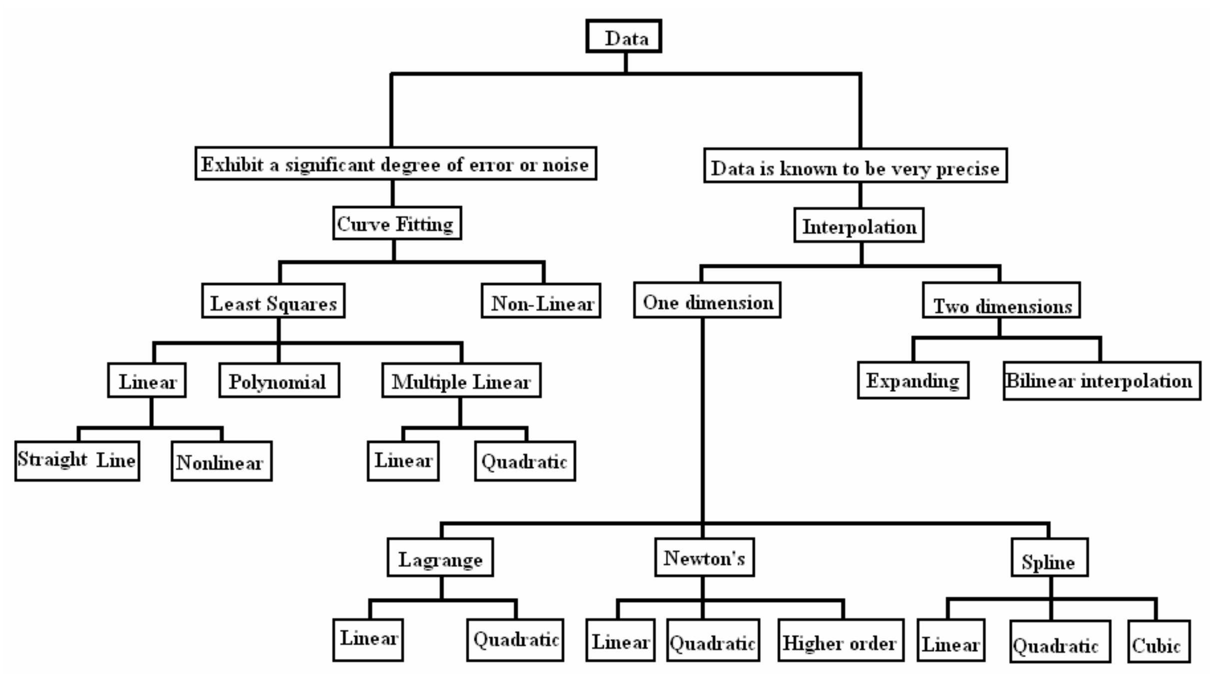
Chapter 4

Interpolation and Curve Fitting

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. This method is called **curve fitting**.

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. Such data usually originated from tables; density of water and heat capacity of gasses as a function of T. This method is called **interpolation**.



CURVE FITTING

♣ Least Squares

1. Linear Regression.
2. Polynomial Regression.
3. Multiple Linear Regression.

1. Linear Regression

Objective is to find a functional relation $y = f(x)$, which best approximate a set of n data points (x_i, y_i).

A) Straight Line $y = a + bx$

The difference between the data value, y_i, and the represented by the equation is:

$$\delta_i = y_i - (a + bx_i)$$

By the principle of least squares, the equation will best fit the data when the sum of the squares of the errors is a minimum.

$$S = \sum_{i=1}^{n} (\delta_i)^2 = \sum_{i=1}^{n} [y_i - (a + bx_i)]^2$$

One condition for S to be minimum is that the partial derivatives of S with respect to a and b must be zero.

$$\frac{\partial S}{\partial a} = \sum_{i=1}^{n} 2[y_i - (a + bx_i)](-1) = 0$$

$$\frac{\partial S}{\partial b} = \sum_{i=1}^{n} 2[y_i - (a + bx_i)](-x_i) = 0$$

Simplifying:

$$\sum_{i=1}^{n} y_i - \sum_{i=1}^{n} a - \sum_{i=1}^{n} bx_i = 0 \Rightarrow \sum_{i=1}^{n} y_i = na + b\sum_{i=1}^{n} x_i = 0$$

$$\sum_{i=1}^{n} y_i x_i - a\sum_{i=1}^{n} x_i - b\sum_{i=1}^{n} x_i^2 = 0 \Rightarrow \sum_{i=1}^{n} y_i x_i = a\sum_{i=1}^{n} x_i + b\sum_{i=1}^{n} x_i^2 = 0$$

Solving for a and b:

$$a = \frac{\sum_{i=1}^{n} y_i - b\sum_{i=1}^{n} x_i}{n}$$

$$b = \frac{n\sum_{i=1}^{n} y_i x_i - \sum_{i=1}^{n} y_i \sum_{i=1}^{n} x_i}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2}$$

B) **Nonlinear equation**.

Sometimes it is possible to transform a nonlinear equation to the linear form by proper substitutions:

1) $y = ab^x$ 2) $y = ax^b$

3) $y = e^{(ax+b)}$ 4) $y = ae^{bx}$

5) $y = \dfrac{1}{a+bx}$

2. <u>Polynomial Regression</u>

We wish to approximate *n* data points (x_i, y_i) by a polynomial of degree *m* ($m<n$)

$$y(x) = C_1 + C_2 x_i + C_3 x_i^2 + C_4 x_i^3 + \dots\dots\dots\dots\dots + C_m x_i^{m-1} + C_{m+1} x_i^m$$

Applying the principle of least squares:

$$\delta_i = y_i - C_1 - C_2 x_i - C_3 x_i^2 - C_4 x_i^3 + \dots\dots\dots\dots\dots - C_m x_i^{m-1} - C_{m+1} x_i^m$$

$$S = \sum_{i=1}^{n} (\delta_i)^2 = \sum_{i=1}^{n} \left[y_i - C_1 - C_2 x_i - C_3 x_i^2 - C_4 x_i^3 + \dots\dots\dots\dots\dots - C_m x_i^{m-1} - C_{m+1} x_i^m \right]^2$$

$$\frac{\partial S}{\partial C_1} = \sum_{i=1}^{n} 2\left[y_i - C_1 - C_2 x_i - C_3 x_i^2 - C_4 x_i^3 + \dots\dots\dots\dots\dots - C_m x_i^{m-1} - C_{m+1} x_i^m \right](-1) = 0$$

$$\frac{\partial S}{\partial C_2} = \sum_{i=1}^{n} 2\left[y_i - C_1 - C_2 x_i - C_3 x_i^2 - C_4 x_i^3 + \dots\dots\dots\dots\dots - C_m x_i^{m-1} - C_{m+1} x_i^m \right](-x_i) = 0$$

$$\Downarrow$$

$$\frac{\partial S}{\partial C_m} = \sum_{i=1}^{n} 2\left[y_i - C_1 - C_2 x_i - C_3 x_i^2 - C_4 x_i^3 + \dots\dots\dots\dots\dots - C_m x_i^{m-1} - C_{m+1} x_i^m \right]\left(-x_i^{m-1}\right) = 0$$

$$\frac{\partial S}{\partial C_{m+1}} = \sum_{i=1}^{n} 2\left[y_i - C_1 - C_2 x_i - C_3 x_i^2 - C_4 x_i^3 + \dots\dots\dots\dots\dots - C_m x_i^{m-1} - C_{m+1} x_i^m \right]\left(-x_i^{m}\right) = 0$$

Simplifying

$$\sum_{i=1}^{n} y_i = nC_1 + C_2 \sum_{i=1}^{n} x_i + C_3 \sum_{i=1}^{n} x_i^2 + C_4 \sum_{i=1}^{n} x_i^3 + \dots\dots\dots\dots\dots + C_m \sum_{i=1}^{n} x_i^{m-1} + C_{m+1} \sum_{i=1}^{n} x_i^m$$

$$\sum_{i=1}^{n} y_i x_i = C_1 \sum_{i=1}^{n} x_i + C_2 \sum_{i=1}^{n} x_i^2 + C_3 \sum_{i=1}^{n} x_i^3 + C_4 \sum_{i=1}^{n} x_i^4 + \dots\dots\dots\dots\dots + C_m \sum_{i=1}^{n} x_i^{m} + C_{m+1} \sum_{i=1}^{n} x_i^{m+1}$$

$$\Downarrow$$

$$\sum_{i=1}^{n} y_i x_i^{m-1} = C_1 \sum_{i=1}^{n} x_i^{m-1} + C_2 \sum_{i=1}^{n} x_i^m + C_3 \sum_{i=1}^{n} x_i^{m+1} + C_4 \sum_{i=1}^{n} x_i^{m+2} + \dots\dots\dots\dots\dots + C_m \sum_{i=1}^{n} x_i^{2m-2} + C_{m+1} \sum_{i=1}^{n} x_i^{2m-1}$$

$$\sum_{i=1}^{n} y_i x_i^{m} = C_1 \sum_{i=1}^{n} x_i^{m} + C_2 \sum_{i=1}^{n} x_i^{m+1} + C_3 \sum_{i=1}^{n} x_i^{m+2} + C_4 \sum_{i=1}^{n} x_i^{m+3} + \dots\dots\dots\dots\dots + C_m \sum_{i=1}^{n} x_i^{2m-1} + C_{m+1} \sum_{i=1}^{n} x_i^{2m}$$

These equations represent a system of linear equation which can be written as:

$$
\begin{bmatrix}
n & \sum x_i & \sum x_i^2 & \sum x_i^3 \ldots\ldots\ldots\ldots & \sum x_i^{m-1} & \sum x_i^m \\
\sum x_i & \sum x_i^2 & \sum x_i^3 & \sum x_i^4 \ldots\ldots\ldots\ldots & \sum x_i^m & \sum x_i^{m+1} \\
\sum x_i^2 & \sum x_i^3 & \sum x_i^4 & \sum x_i^5 \ldots\ldots\ldots\ldots & \sum x_i^{m+1} & \sum x_i^{m+2} \\
\Downarrow & & & & & \\
\sum x_i^{m-1} & \sum x_i^m & \sum x_i^{m+1} & \sum x_i^{m+2} \ldots\ldots\ldots\ldots & \sum x_i^{2m-2} & \sum x_i^{2m-1} \\
\sum x_i^m & \sum x_i^{m+1} & \sum x_i^{m+2} & \sum x_i^{m+3} \ldots\ldots\ldots\ldots & \sum x_i^{2m-1} & \sum x_i^{2m}
\end{bmatrix}
\begin{Bmatrix} C_1 \\ C_2 \\ C_3 \\ \Downarrow \\ C_m \\ C_{m+1} \end{Bmatrix}
=
\begin{Bmatrix} \sum y_i \\ \sum y_i x_i \\ \sum y_i x_i^2 \\ \Downarrow \\ \sum y_i x_i^{m-1} \\ \sum y_i x_i^m \end{Bmatrix}
$$

The above system can be reduced to any degree. For example a 2^{nd} the constants of a second order polynomial can be obtained by solving the following system of linear equations:

$$
\begin{bmatrix}
n & \sum x_i & \sum x_i^2 \\
\sum x_i & \sum x_i^2 & \sum x_i^3 \\
\sum x_i^2 & \sum x_i^3 & \sum x_i^4
\end{bmatrix}
\begin{Bmatrix} C_1 \\ C_2 \\ C_3 \end{Bmatrix}
=
\begin{Bmatrix} \sum y_i \\ \sum y_i x_i \\ \sum y_i x_i^2 \end{Bmatrix}
$$

3. <u>Multiple Linear Regression</u>

Frequently experimental data involve more than two variables.
The function can assume various forms: linear, polynomial, logarithmic, exponential, and trigonometric.

A) Multivariable linear regression

$$
F = C_1 + C_2 x + C_3 y + C_4 z
$$

The least squares fit gives

$$
\begin{bmatrix}
n & \sum x & \sum y & \sum z \\
\sum x & \sum x^2 & \sum xy & \sum xz \\
\sum y & \sum xy & \sum y^2 & \sum yz \\
\sum z & \sum xz & \sum yz & \sum z^2
\end{bmatrix}
\begin{Bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \end{Bmatrix}
=
\begin{Bmatrix} \sum F \\ \sum Fx \\ \sum Fy \\ \sum Fz \end{Bmatrix}
$$

B) Multivariable polynomial approximation

Consider the **quadratic** multivariable polynomial:

$$
z = C_1 + C_2 x + C_3 y + C_4 x^2 + C_5 y^2 + C_6 xy
$$

Using the least squares technique, the following system of linear equations will be obtained:

$$\begin{bmatrix} n & \sum x_i & \sum y_i & \sum x_i^2 & \sum y_i^2 & \sum x_i y_i \\ \sum x_i & \sum x_i^2 & \sum x_i y_i & \sum x_i^3 & \sum x_i y_i^2 & \sum x_i^2 y_i \\ \sum y_i & \sum x_i y_i & \sum y_i^2 & \sum x_i^2 y_i & \sum y_i^3 & \sum x_i y_i^2 \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^2 y_i & \sum x_i^4 & \sum x_i^2 y_i^2 & \sum x_i^3 y_i \\ \sum y_i^2 & \sum x_i y_i^2 & \sum y_i^3 & \sum x_i^2 y_i^2 & \sum y_i^4 & \sum x_i y_i^3 \\ \sum x_i y_i & \sum x_i^2 y_i & \sum x_i y_i^2 & \sum x_i^3 y_i & \sum x_i y_i^3 & \sum x_i^2 y_i^2 \end{bmatrix} \begin{Bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ C_6 \end{Bmatrix} = \begin{Bmatrix} \sum z_i \\ \sum z_i x_i \\ \sum z_i y_i \\ \sum z_i x_i^2 \\ \sum z_i y_i^2 \\ \sum z_i x_i y_i \end{Bmatrix}$$

Coefficient of Determination (r^2)

To find the coefficient of determination follow the following procedure:

1) $S_r = \sum \left(y_i - y_{calculated}\right)^2$

2) $\bar{y} = \dfrac{\sum y_i}{n}$

3) $S = \sqrt{\dfrac{S_r}{n-(m+1)}}$

4) $S_t = \sum \left(y_i - \bar{y}\right)^2$

5) $r^2 = \dfrac{S_t - S_r}{S_t} = \dfrac{\sum \left(y_i - \bar{y}\right)^2 - \sum \left(y_i - y_{calculated}\right)^2}{\sum \left(y_i - \bar{y}\right)^2}$

♣ Nonlinear Regression

- There are many cases in engineering where nonlinear models must be fit the data.
- These models are defined as those which have a nonlinear dependence on their parameters.
- There is no way that these equations can be manipulated so that it conforms to the general form of the linear equations.
- As with linear least squares, nonlinear regression is based on determining the values of the parameters that minimize the sum of the squares of the residuals.
- For nonlinear case, the solution must proceed in an iterative fashion.
- Successful solutions are often highly dependent on good initial guesses for the parameters.

Algorithm:

1) Find [Z_j] the matrix of partial derivatives of the function evaluated at the initial guess, *j*.

$$Z_j = \begin{bmatrix} \frac{\partial f_1}{\partial C_1} & \frac{\partial f_1}{\partial C_2} & \frac{\partial f_1}{\partial C_3} \\ \frac{\partial f_2}{\partial C_1} & \frac{\partial f_2}{\partial C_2} & \frac{\partial f_2}{\partial C_3} \\ \frac{\partial f_3}{\partial C_1} & \frac{\partial f_3}{\partial C_2} & \frac{\partial f_3}{\partial C_3} \\ \Downarrow & & \\ \frac{\partial f_n}{\partial C_1} & \frac{\partial f_n}{\partial C_2} & \frac{\partial f_n}{\partial C_3} \end{bmatrix}$$

Where n is the number of data points

$\frac{\partial f_n}{\partial C_i}$ is the partial derivatives of the function with respect to the C^{th} parameter evaluated at the n^{th} data point.

2) Find vector {D} contains the difference between the measurements and the function values.

$$\{D\} = \begin{Bmatrix} y_1 - f(x_1) \\ y_2 - f(x_2) \\ y_3 - f(x_3) \\ \Downarrow \\ y_n - f(x_n) \end{Bmatrix}$$

3) Find $[Z_j]^T[Z_j]$ and $[Z_j]^T\{D\}$

4) Find $\{\Delta C\} = \begin{Bmatrix} \Delta C_1 \\ \Delta C_2 \\ \Delta C_3 \\ \Downarrow \\ \Delta C_n \end{Bmatrix}$ using the following equation:

$$\left[[Z_j]^T[Z_j]\right]\{\Delta C\} = [Z_j]^T\{D\}$$

5) Find the new values for the parameters using:

$$\begin{aligned} C_{1,j+1} &= C_{1,j} + \Delta C_1 \\ C_{2,j+1} &= C_{2,j} + \Delta C_2 \\ C_{3,j+1} &= C_{3,j} + \Delta C_3 \\ &\Downarrow \\ C_{n,j+1} &= C_{n,j} + \Delta C_n \end{aligned}$$

This procedure is repeated until the solution converges and falls below an acceptable stopping criterion.

$$\varepsilon_a = \left| \frac{C_{i,j+1} - C_{i,j}}{C_{i,j+1}} \right| * 100\%$$

INTERPOLATION

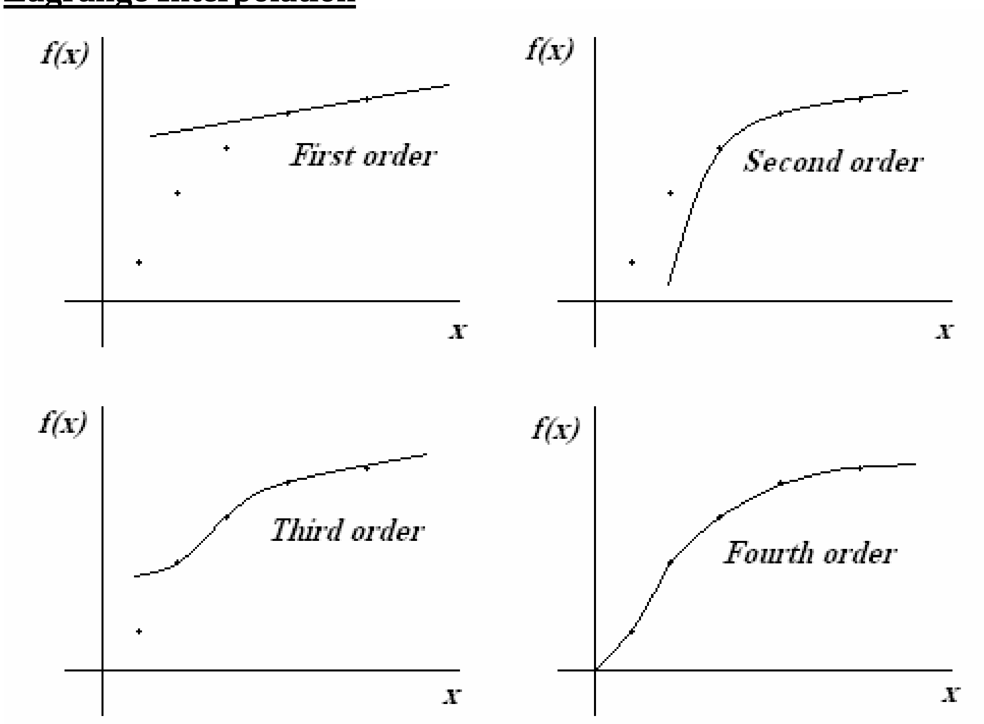
A) Interpolation in one dimension.
B) Interpolation in two dimensions.

A) Interpolation in One Dimension

Polynomial interpolation:

- Lagrange interpolation
- Newton interpolation
- Spline interpolation

1) Lagrange Interpolation



Lagrange interpolating polynomial can be represented by:

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

$$n \text{ is the order of polynomial}$$

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

$$\prod \text{ product of}$$

I) Linear interpolation (*n=1*)

$$f_1(x) = \sum_{i=0}^{1} L_i(x) f(x_i) = L_0(x) f(x_0) + L_1(x) f(x_1)$$

$$L_0(x) = \prod_{j=1}^{1} \frac{x - x_j}{x_0 - x_j} = \frac{x - x_1}{x_0 - x_1}$$

$$L_1(x) = \prod_{j=0}^{1} \frac{x - x_j}{x_1 - x_j} = \frac{x - x_0}{x_1 - x_0}$$

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

II) Quadratic interpolation (*n=2*)

$$f_2(x) = \sum_{i=0}^{2} L_i(x) f(x_i) = L_0(x) f(x_0) + L_1(x) f(x_1) + L_2(x) f(x_2)$$

$$L_0(x) = \prod_{j=1}^{2} \frac{x - x_j}{x_0 - x_j} = \frac{x - x_1}{x_0 - x_1} \frac{x - x_2}{x_0 - x_2}$$

$$L_1(x) = \prod_{j=0}^{2} \frac{x - x_j}{x_1 - x_j} = \frac{x - x_0}{x_1 - x_0} \frac{x - x_2}{x_1 - x_2}$$

$$L_2(x) = \prod_{j=0}^{2} \frac{x - x_j}{x_2 - x_j} = \frac{x - x_0}{x_2 - x_0} \frac{x - x_1}{x_2 - x_1}$$

$$\therefore\ f_1(x) = \frac{x - x_1}{x_0 - x_1} \frac{x - x_2}{x_0 - x_2} f(x_0) + \frac{x - x_0}{x_1 - x_0} \frac{x - x_2}{x_1 - x_2} f(x_1) + \frac{x - x_0}{x_2 - x_0} \frac{x - x_1}{x_2 - x_1} f(x_2)$$

2) Newton Interpolation Polynomials

I) Linear interpolation (*n=1*)

$$f_1(x) = a_0 + a_1(x - x_0)$$

$$at\ x_0 \rightarrow f(x) = f(x_0)$$

$$at\ x_1 \rightarrow f(x) = f(x_1)$$

substitute and solve for the two constants a_0 and a_1

$$a_0 = f(x_0)$$

$$a_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

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

II) Quadratic interpolation (*n=2*)

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

$$at\ x_0 \rightarrow f(x) = f(x_0)$$

$$at\ x_1 \rightarrow f(x) = f(x_1)$$

$$at\ x_2 \rightarrow f(x) = f(x_2)$$

substitute and solve for the three constants b_0, b_1 and b_2

$$b_0 = f(x_0)$$

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

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

$$\therefore f_1(x) = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0) + \frac{\frac{f(x_2) - f(x_1)}{x_2 - x_1} - \frac{f(x_1) - f(x_0)}{x_1 - x_0}}{x_2 - x_0}(x - x_0)(x - x_1)$$

III) Higher order interpolation polynomials

There are two disadvantages for the Lagrange polynomial method for interpolation compared to the divided-difference method:

More arithmetic operations are required

If we desire to add or subtract a point from the set used to construct the polynomial, we essentially have to start over in the computations.

The divided-difference method avoids all of these computations.

Assume that the *x's* are not evenly spaced or even the values are arranged in any particular order.

$$f[x_0, x_1] = \frac{f_1 - f_0}{x_1 - x_0} = f_0^{[1]} \text{ first divided – difference between } x_0 \text{ and } x_1$$

$$f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} = f_0^{[2]} \text{ second divided – difference between } x_0, x_1 \text{ and } x_2$$

$$f[x_0, x_1, \ldots\ldots\ldots\ldots x_n] = \frac{f[x_1, x_2, \ldots\ldots\ldots, x_n] - f[x_0, x_1, \ldots\ldots\ldots, x_{n-1}]}{x_n - x_0} = f_0^{[n]}$$

n^{th} divided – difference between $x_0, x_1, \ldots\ldots\ldots\ldots\ldots,$ and x_n

x_i	f_i	$f[x_i,x_{i+1}]$	$f[x_i,x_{i+1}, x_{i+2}]$	$f[x_i,x_{i+1}, x_{i+2}, x_{i+3}]$	$f[x_i,x_{i+1}, x_{i+2}, x_{i+3}, x_{i+4}]$
x_0	$f_0 = a_0$	$f[x_0,x_1] = a_1$	$f[x_0,x_1, x_2] = a_2$	$f[x_0,x_1, x_2, x_3] = a_3$	$f[x_0,x_1, x_2, x_3, x_4] = a_4$
x_1	f_1	$f[x_1,x_2]$	$f[x_1,x_2, x_3]$	$f[x_1,x_2, x_3, x_4]$	
x_2	f_2	$f[x_2,x_3]$	$f[x_2,x_3, x_4]$		
x_3	f_3	$f[x_3,x_4]$			
x_4	f_4				

First divided-difference:

$$f[x_0,x_1]=\frac{f_1-f_0}{x_1-x_0} \qquad f[x_1,x_2]=\frac{f_2-f_1}{x_2-x_1}$$

$$f[x_2,x_3]=\frac{f_3-f_2}{x_3-x_2} \qquad f[x_3,x_4]=\frac{f_4-f_3}{x_4-x_3}$$

Second divided-difference:

$$f[x_0,x_1,x_2]=\frac{f[x_1,x_2]-f[x_0,x_1]}{x_2-x_0}$$

$$f[x_1,x_2,x_3]=\frac{f[x_2,x_3]-f[x_1,x_2]}{x_3-x_1}$$

$$f[x_2,x_3,x_4]=\frac{f[x_3,x_4]-f[x_2,x_3]}{x_4-x_2}$$

Third divided-difference:

$$f[x_0,x_1,x_2,x_3]=\frac{f[x_1,x_2,x_3]-f[x_0,x_1,x_2]}{x_3-x_0}$$

$$f[x_1,x_2,x_3,x_4]=\frac{f[x_2,x_{34},x_4]-f[x_1,x_2,x_3]}{x_4-x_1}$$

Fourth divided difference:

$$f[x_0,x_1,x_2,x_3,x_4]=\frac{f[x_1,x_2,x_3,x_4]-f[x_0,x_1,x_2,x_3]}{x_4-x_0}$$

Then a fourth order Newton's function can be written as:

$$\begin{aligned}P_4(x)=a_0+a_1(x-x_0)+a_2(x-x_0)(x-x_1)+a_3(x-x_0)(x-x_1)(x-x_2)\\+a_4(x-x_0)(x-x_1)(x-x_2)(x-x\ \ )\end{aligned}$$

3) <u>Spline Interpolation</u>

The disadvantage of using a single polynomial (of high degree) to interpolate a large number of data points can be avoided using piecewise polynomials.

A) Piecewise Linear Interpolation

Suppose that there are four data points:

$$(x_0, f(x_0)), (x_1, f(x_1)), (x_2, f(x_2)), \textit{and } (x_3, f(x_3))$$
$$x_0 \langle x_1 \langle x_2 \langle x_3$$

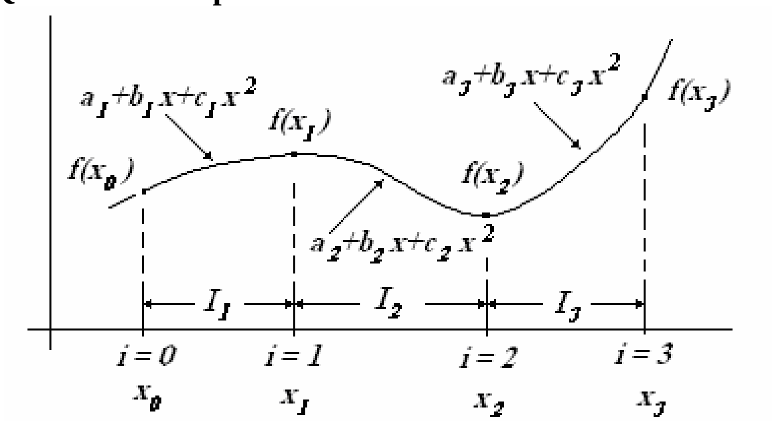
These data points can be split into 3 intervals:

$$I_1=[x_0,x_1], I_1=[x_1,x_2], \text{and } I_1=[x_2,x_3]$$

Three linear interpolating functions can be written one for each interval:

$$P(x)=\begin{cases} f(x_0)+\dfrac{f(x_1)-f(x_0)}{x_1-x_0}(x-x_0) & x_0\le x\le x_1 \\ f(x_1)+\dfrac{f(x_2)-f(x_1)}{x_2-x_1}(x-x_1) & x_1\le x\le x_2 \\ f(x_2)+\dfrac{f(x_3)-f(x_2)}{x_3-x_2}(x-x_2) & x_2\le x\le x_3 \end{cases}$$

B) Piecewise Quadratic Interpolation



Quadratic interpolating functions can be written for each interval with different constant for the second order equation:

$$f_i(x)=a_i+b_ix+c_ix^2$$

- There are 3 constant in this equation.
- For *n+1* data points, there are n intervals
- There are *3* constants for each interval
- Then the number of unknowns is *3n*

To find the values of these unknowns use the following information:

1) The function values must be equal at the interior knots, (this gives *2n-2* equation)
2) The first and last functions must pass through the end points, (this makes the number of equations *2n*)
3) The first derivatives at the interior knots must be equal, (this makes the number of equations *3n-1*)
4) Assume that the second derivative is zero at the first point, (this makes the number of equations *3n*)

C) Cubic Spline

$$f_i(x)=a_i+b_ix+c_ix^2+d_ix^3$$

- There are *4* constant in this equation.
- For *n+1* data points, there are *n* intervals
- There are *4* constants for each interval
- Then the number of unknowns is *4n*

To find the values of these unknowns use the following information:

1) The function values must be equal at the interior knots, (this gives *2n-2* equation)
2) The first and last functions must pass through the end points, (this gives *2* equations, i.e. makes the total number of equations *2n*)
3) The first derivatives at the interior knots must be equal, (this gives *n-1* equation, i.e. makes the total number of equations *3n-1*)

4) The second derivatives at the interior knots must be equal, (this gives *n-1* equation, i.e. makes the total number of equations *4n-2*)
5) Assume that the third derivatives are zero at the end knots, (this gives *2* equations, i.e. makes the total number of equations *4n*)

Interpolation in Two Dimensions

- The general interpolation problem for two (or more) independent variable is much more difficult than for a single variable.
- Unless the function values are known on a rectangular grid of points, it is not easy either to order the data points or to determine which of them should be used to find the interpolated value at any particular point in the region.

1. The simplest form of interpolation in two dimensions expands the data matrix Z by interleaving interpolates between every element.

$$x = [x_1, x_2, ..., x_n] \text{and } y = [y_1, y_2, ..., y_m]$$

Then the interpolated values are found on the grid defined by the vector

$$xx = \left[x_1, \frac{(x_1 + x_2)}{2}, x_2, \frac{(x_2 + x_3)}{2}, ..., \frac{(x_{n-1} + x_n)}{2}, x_n \right]$$
$$yy = \left[y_1, \frac{(y_1 + y_2)}{2}, y_2, \frac{(y_2 + y_3)}{2}, ..., \frac{(y_{m-1} + y_m)}{2}, y_m \right]$$

2. Using bilinear interpolation functions

$$Z = a + bx + cy + dxy$$

Using the data values at the four corners of the region.
Thus, the region R_{ij}, with data values:

$$(x_i, y_i), (x_i, y_{i+1}), (x_{i+1}, y_i), (x_{i+1}, y_{i+1})$$

There are four equations for the four unknowns a_{ij}, b_{ij}, c_{ij}, and d_{ij}.

$$Z(i, j) = a_{ij} + b_{ij} x_i + c_{ij} y_j + d_{ij} x_i y_j$$
$$Z(i, j+1) = a_{ij} + b_{ij} x_i + c_{ij} y_{j+1} + d_{ij} x_i y_{j+1}$$
$$Z(i+1, j) = a_{ij} + b_{ij} x_{i+1} + c_{ij} y_j + d_{ij} x_{i+1} y_j$$
$$Z(i+1, j+1) = a_{ij} + b_{ij} x_{i+1} + c_{ij} y_{j+1} + d_{ij} x_{i+1} y_{j+1}$$