Curve Fitting

Why do we want to curve fit? In general, we fit data points to produce a smooth representation of the system whose response generated the data points. We do this for a variety of reasons

1. to interpolate or extrapolate
2. to understand the behavior of the system
3. to integrate or differentiate the response. In some cases the derivative may represent physical characteristics of the system, i.e., if the response is fluid velocity, the derivatives reflect the shear stresses
4. to create curves of pleasing shapes, e.g., fonts

1 Interpolation

1.1 Fitting \( f(x) \)
Fit all points = interpolation. Since the \( p_n(x) \) polynomial has \( n - 1 \) max and min extremes and an error proportional to \( f^{(n+1)}(\xi) \) for large \( n \) the fit may oscillate wildly.

1.2 Hermite interpolation
Problem here is the need for the derivatives, \( df/dx \). These are unlikely to be known, and if estimates are used, the fit may be badly degraded.

2 Composite Fitting

Consider using a low order polynomial to fit the curve in a limited region. Then if we adjusted the polynomial coefficients to ensure continuity of derivatives at the edges of the regions, we could get a pretty good fit. We let the region be defined as having a span \( x_i \leq x \leq x_{i+m} \), having \( m + 1 \) points (this is the same as the panels of composite integrations) and choose different continuity requirements.

2.1 Parabolic and Cubic Splines
The most popular splines are cubic splines that ensure that \( df/dx \) and \( d^2f/dx^2 \) are continuous. Sometimes parabolic splines are used, but then we can only enforce continuity of \( df/dx \). Cubic splines require specifying the end conditions and these will impact the fit. They often ripple too much and cannot handle vertical tangents. Sometimes parabolic and cubic splines are mixed with some panels using cubic functions and other parabolic.
2.2 Ripples and Vertical Tangents

Ripples can be reduced by applying tension to the spline. The problem is knowing how much tension is needed. Too much tension reduces the spline to a series of straight line segments in each span. The Monder method ensures that the first derivative does not change sign in any span. Not as extreme as a highly tensioned spline and particularly valuable when the spline fit must be differentiated to evaluate a physical property.

Composite splines require solving a set of linear equations and changing any one data point affects the entire fitted curve.

2.3 B-Splines

B-splines, also called fundamental splines, are very popular. The idea is to fit a curve over 4 spans encompassing the points, \(x_{i-4}, x_{i-3}, x_{i-2}, x_{i-1}\), and \(x_i\) with the curve having \(f = df/dx = d^2f/dx^2 = 0\) at \(x_{i-4}\) and \(x_i\) and being non-zero at points \(x_{i-4} < x < x_i\). The B-spline extends on either side, \(x < x_{i-4}\) and \(x > x_i\) with zero values. Any one data point affects only 3 B-spline segments and thus one can locally modify the fit. B-splines still require solving a set of equations.

3 Bezier Curves

In this case the fitted curve does not pass through the data points but the data points exert a pull on the curve. The degree of pull can be adjusted by the placement of the data points. Bezier curves are parametric curves and can be used to represent multivalued functions. The evaluation of a single Bezier function for a large number of data points involves very ill conditioned calculations. Consequently most Bezier fits are done using composite fitting.

4 Least Squares

The fitted curve is defined by minimizing some function of the difference between the data points and the fitted curve. Let \(\epsilon_i = f(x_i) - y(x_i)\). The choices of minimizing a) \(\sum |\epsilon_i|\) (called Laplace’s method) or \(max(|\epsilon_i|)\) leads to difficulties so most fitting is done by minimizing \(\sum \epsilon^2\), an approach attributed to Gauss.

4.1 Problems

While many assume that the noise (error) is in \(y\) and fit \(y\) vs \(x\), why not fit \(x\) vs \(y\)? This often leads to very different fitted curves. A recommended approach is orthogonal fitting in which the orthogonal distance between the data points and the curve is minimized. This requires that one knows \(\sigma(y)/\sigma(x)\) where \(\sigma\) is the standard deviation of the noise. Often one assumes that \(\sigma(y)/\sigma(x) = 1\) but the results are usually strong functions of this ratio.
If one fits a polynomial, the coefficients are found from solving a set of linear algebraic equations. If the data points are equally spaced the matrix will be very ill conditioned for relatively small number of data points. The solution is to use orthogonal polynomials. These depend upon the data point spacing and must be calculated for each set of data.
Introduction

Suppose that a system has a constant response, $\alpha$, which we attempt to determine by measuring the response $n$ times. Let these responses be $x_i$ and let them be chosen from a population with a mean of $\alpha$ and a standard deviation of $\sigma$. Then let us estimate $\alpha$ by requiring that

$$S = \sum_{i=1}^{n} (x_i - \alpha)^2 = \text{minimum}$$  \hspace{1cm} (1)

Since $\alpha$ is unknown, we require that the variation of $S$ with respect to $\alpha$ be zero, or

$$\delta S = -2 \sum_{i=1}^{n} (x_i - \alpha) \delta \alpha = 0$$  \hspace{1cm} (2)

which leads to the result

$$\alpha = \bar{x} = \frac{\sum x_i}{n}$$  \hspace{1cm} (3)

and we see that the arithmetic mean is the Least Squares estimator of the true system response $\alpha$.

Weights

Now suppose that instead of using Eq 1 as it stands, we group the readings, $x_i$ into $m$ sets, each of $n_j$ readings. That is

$$\alpha = \frac{\sum \sum x_i}{n} = \frac{\sum n_j \bar{x}_j}{n}$$  \hspace{1cm} (4)

where $\bar{x}_j$ represents the average of each set of $n_j$ readings. Suppose that we consider the values of $n_j$ as weights which act upon the readings $\bar{x}_j$ and we revise our approach to consider a weighted sum in Equation 1. That is we write

$$S = \sum_{j=1}^{m} w_j (\bar{x}_j - \alpha)^2 = \text{minimum}$$  \hspace{1cm} (5)

Following the same derivation as before we would obtain the result for $\alpha$ of

$$\alpha = \bar{x} = \frac{\sum w_j \bar{x}_j}{\sum w_j}$$  \hspace{1cm} (6)

By comparing Equations 4 and 6, we see that the appropriate definition of the weights is $w_j = n_j$.

The fundamental interpretation of a weight is that it represents a number of repeated readings.
To clarify the role of the weights, let us define \( w_j = \frac{k}{\sigma^2(\bar{x}_j)} \), that is the weight is inversely proportional to the variance of the set of readings which make up the \( j^{th} \) average, \( \bar{x}_j \). What is the appropriate value of the constant \( k \)? We write
\[
 w_j = n_j = \frac{k}{\sigma^2(\bar{x}_j)} = \frac{kn_j}{\sigma^2} \tag{7}
\]

Where we have used the fact that the variance of a mean is related to the variance of the population by \( \sigma^2(\bar{x}_j) = \sigma^2/n_j \). From Eq. 7, we find that \( k = \sigma^2 \). Thus we finally obtain that the appropriate weight for any quantity \( f \) is given by
\[
 w_f = \frac{\sigma^2}{\sigma_f^2} \tag{8}
\]

Sometimes the weight functions in the example are expressed as \( w_j = n_j/n \), i.e., normalized. However this obscures the fundamental interpretation of the weight and also destroys some of the symmetry in the development of weights for generalized quantities, \( f \).

Clearly, \( w_f \) is a measure of the accuracy by which a set of readings of \( f \) represents the true function. For example, if \( x_i \) is known exactly, \( w_{x_i} \) is infinite, i.e., \( \sigma_{x_i} = 0 \).

Fitting Data
Consider a set of \( n \) values \( f_i \) which we wish to approximate by a set of functions \( \phi_j \) in the form
\[
 f(x) = \sum_j^M a_j \phi_j(x) \tag{9}
\]
where \( \phi_j \) are user define functions. These functions are chosen to fit the data \( f_i \) in the best way possible. The coefficients \( a_j \) are determined by requiring that the error, \( E \), is minimized.
\[
 E = \sum_{k=1}^n \left[ f_k - \sum_{j=1}^M a_j \phi_j(x_k) \right]^2 \tag{10}
\]
If the \( \phi(x) \) do not adequately represent the data, the approximation will be poor. If \( \phi_j(x) = x^j \), then when \( M = n \), the method reduces to the polynomial approximation. If \( \phi_j(x) = \cos(jx) \) and \( M = n \), the result is the cosine Fourier series.

We minimize \( E \) by differentiating it with respect to the unknown coefficients \( a_j \) and setting the derivatives to zero.
\[
 \frac{\partial E}{\partial a_l} = -2 \sum_{i=1}^n \left[ f_k - \sum_{j=1}^M a_j \phi_j(x_k) \right] \phi_l(x_k) = 0 \tag{11a}
\]
Ignoring the $-2$, and dropping the summation over $k$ sign for convenience, this equation reduces to

$$
\sum_{j=1}^{M} a_j \phi_j(x_k) \phi_l(x_k) = f \phi_l(x_k)
$$

(11b)

Expressing this in matrix form we have,

$$
\begin{pmatrix}
\phi_1 \phi_1 & \phi_2 \phi_1 & \phi_3 \phi_1 & \ldots & \phi_M \phi_1 \\
\phi_1 \phi_2 & \phi_2 \phi_2 & \phi_3 \phi_2 & \ldots & \phi_M \phi_2 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\phi_1 \phi_M & \phi_2 \phi_M & \phi_3 \phi_M & \ldots & \phi_M \phi_M
\end{pmatrix}
\begin{pmatrix}
a_1 \\
a_2 \\
\vdots \\
a_M
\end{pmatrix}
= 
\begin{pmatrix}
f_k \phi_1 \\
f_k \phi_2 \\
\vdots \\
f_k \phi_M
\end{pmatrix}
$$

(12)

Although the set of equations appears to be very simple to solve (particularly since it is symmetric), if the data points are equally spaced and the $\phi_j = x^j$, the determinant is extremely small for values of $M$ as small as 7.

**CONSTRAINTS**

It is common to require that the fitting function pass through specific points, or that it satisfy some constraints expressed in terms of integrals or derivatives. The usual way to force the least squares fit to pass through a point, is to include a weighting function of the form

$$
E = \sum_{k=1}^{n} w_k [f_k - \sum_{j=1}^{M} a_j \phi_j(x_k)]^2
$$

(13)

where the weights $w_k$ are adjusted to have a large value where you wish the error to be the smallest. This method, although widely used, and reasonably appropriate when emphasizing the importance of some points with respect to other points, is totally incorrect when forcing the curve pass through specific points.

Let us consider the augmented error form

$$
E = \sum_{k=1}^{n} [f_k - \sum_{j=1}^{M} a_j \phi_j(x_k)]^2 + 2\lambda G(x_p)
$$

(14)

where $G(x_p)$ represents a constraint upon the fitting function. For example, if we required that the fitting function pass through the point $x_p$, then $g(x_p)$ would be given by

$$
G(x_p) = f_p - \sum_{j=1}^{M} a_j \phi_j(x_p) = 0
$$

(15)

Note that adding $G(x_p)$ does not affect the value of $E$ since $G = 0$. Now differentiating $E$ with respect to $a_l$ and to $\lambda$ will give
\[ \frac{\partial E}{\partial x_l} = -2 \sum_{i=1}^{n} [f_k - \sum_{j=1}^{M} a_j \phi_j(x_k)] \phi_l(x_k) - 2\lambda \phi_l(x_p) = 0 \]

\[ \frac{\partial E}{\partial \lambda} = f_p - \sum_{j=1}^{M} a_j \phi_j(x_p) = 0 \]  

(16a)

The matrix form then becomes

\[
\begin{pmatrix}
\phi_1 \phi_1 & \phi_2 \phi_1 & \phi_3 \phi_1 & \cdots & \phi_M \phi_1 & \phi_1(x_p) \\
\phi_1 \phi_2 & \phi_2 \phi_2 & \phi_3 \phi_2 & \cdots & \phi_M \phi_2 & \phi_2(x_p) \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
\phi_1 \phi_M & \phi_2 \phi_M & \phi_3 \phi_M & \cdots & \phi_M \phi_M & \phi_M(x_p) \\
\phi_1(x_p) & \phi_2(x_p) & \phi_3(x_p) & \cdots & \phi_M(x_p) & 0 \\
\end{pmatrix}
\begin{pmatrix}
a_1 \\
a_2 \\
\vdots \\
a_M \\
\lambda \\
\end{pmatrix}
= \begin{pmatrix}
f_k \phi_1 \\
f_k \phi_2 \\
\vdots \\
f_k \phi_M \\
f_p \\
\end{pmatrix}
\]  

(16b)

where terms of the form \( \phi_i \phi_j \) and \( f_k \phi_j \) represent summations over all of the data point, while terms of the form \( f_p \) or \( \phi_j(x_p) \) are to be evaluated at the point \( p \). If there are several points the curve must pass through, then there must be a \( \lambda \) for each point. If the constraint is of the form

\[ \int \left[ \sum_{j=1}^{M} a_j \phi_j(x) \right] = V \]  

(17a)

then the constraint would be given by

\[ G = \int \left[ \sum_{j=1}^{M} a_j \phi_j(x) \right] - V = 0 \]  

(17b)

Note that the matrix involving \( \lambda \), while symmetric, has a zero diagonal element. This suggests that one must be careful in solving using Gaussian elimination to be sure that the constraints are not the first equations in the set unless pivoting is implemented in your solver.

**Statistics of the Fit**

One of the most common questions in curve fitting data is whether the fit is good and if the parameters are different from 0. For example, if we fit using a polynomial, \( P_n(x) \), what is the best order to use. It might be thought that \( S \) where

\[ S = \sum_{j=1}^{m} w_j (y_j - P_j)^2 = \text{minimum} \]  

(18a)

is a good measure since the closer the fit is to the points, the smaller \( S \) is. However, if there are \( n \) points and we fit with a curve which has \( n \) free parameters, we are not fitting
but interpolating and the curve will pass exactly through each point, yielding $S = 0$. A better estimate of the goodness of the fit is the variance of the fit, $\sigma^2$. As the fit improves, $\sigma^2$ will diminish. Since we do not know this variance, we must estimate it by

$$\sigma^2(\text{est}) = S/(n - p) \quad (18b)$$

where $n$ is the number of points and $p$ is the number of parameters plus additional constraints, that is $n - p =$ number of degrees of freedom.

The approach is to fit with an ever increasing number of parameters (i.e., increasing the complexity of the fitting function) and to check the variance of the fit. As the complexity of the fit is increased, $\sigma(\text{est})$ should decrease. Whenever it begins to increase, we have exceeded the best fit.

**Orthogonal Functions**

If the functions $\phi_j(x)$ in Eq. 9 are orthogonal, then the off diagonal terms in the matrix, Eq. (12) would be zero and we wouldn’t have to worry about the ill conditioning of the matrix. A suitable set of orthogonal polynomials is given by

$$p_{j+1}(x) = (x - \alpha_{j+1})p_j(x) - \beta_j p_{j-1}(x), \quad j = 0, 1, 2, \ldots, n \quad (19a)$$

with

$$p_{-1}(x) = 0, p_0(x) = 1 \quad (19b)$$

where

$$\alpha_{k+1} = \frac{\sum_{i=1}^{N} w_i x_i p_k^2(x_i)}{\sum_{i=1}^{N} w_i p_k^2(x_i)} \quad (19c)$$

$$\beta_k = \frac{\sum_{i=1}^{N} w_i x_i p_k^2(x_i)}{\sum_{i=1}^{N} w_i p_{k-1}^2(x_i)} \quad (19d)$$

If $w_i = 1$ and the data are equally spaced, these polynomials are known as the Gram polynomials.
Consider a function $f$ of $x$, $y$, and $z$ for which we wish to find the maximum or minimum value, i.e., the extremal value. At the extremum the derivative of $f$ must be zero, that is

$$df(x, y, z) = \frac{\partial f}{\partial x}dx + \frac{\partial f}{\partial y}dy + \frac{\partial f}{\partial z}dz = 0$$ (1)

If $x$, $y$, and $z$ are independent, then the only way for this to be true is for the coefficients of $dx$, $dy$, and $dz$ to be zero.

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial y} = \frac{\partial f}{\partial z} = 0$$ (2)

Now suppose that $x$, $y$, and $z$ are related through the function $\Phi(x, y, z) = C$. Then only two of these can be varied at will, the other must follow according to the function $\Phi$. One way of accounting for this constraint would be to solve for $z$ from $\Phi$, substitute for $z$ in $f$ and then require that the partials of $f$ with respect to $x$ and $y$ be zero. Unfortunately we may not be able to solve for $z$. The better way is to use Lagrange multipliers.

Let us find the extremum of

$$F(x, y, z) = f(x, y, z) + \lambda(\Phi - C)$$ (3)

We do this by finding the values of $x$, $y$, and $z$ which make $dF = 0$.

$$dF(x, y, z) = \frac{\partial F}{\partial x}dx + \frac{\partial F}{\partial y}dy + \frac{\partial F}{\partial z}dz = 0$$ (4)

or

$$dF(x, y, z) = (\frac{\partial f}{\partial x} + \lambda \frac{\partial \Phi}{\partial x})dx + (\frac{\partial f}{\partial y} + \lambda \frac{\partial \Phi}{\partial y})dy + (\frac{\partial f}{\partial z} + \lambda \frac{\partial \Phi}{\partial z})dz = 0$$ (5)

Now suppose that $x$ and $y$ are independent and $z$ is given in terms of $\Phi$. Let us choose $\lambda$ such that the coefficient of $dz$ is zero. Then since $x$ and $y$ are independent, the only way that $dF$ can be zero is for the coefficients of $dx$ and $dy$ to be zero. Thus we arrive at the conditions

$$\frac{\partial f}{\partial x} + \lambda \frac{\partial \Phi}{\partial x} = 0$$  \hspace{1cm} (6a)

$$\frac{\partial f}{\partial y} + \lambda \frac{\partial \Phi}{\partial y} = 0$$  \hspace{1cm} (6b)

$$\frac{\partial f}{\partial z} + \lambda \frac{\partial \Phi}{\partial z} = 0$$  \hspace{1cm} (6c)

$$\Phi(x, y, z) - C = 0$$  \hspace{1cm} (6d)

These equations are just what we would have gotten if we differentiated $F$ in terms of $x$, $y$, $z$, and $\lambda$.

**Note** that

$$\frac{\partial F}{\partial C} = \lambda$$ (7)

and thus the Lagrangian multipliers give us the sensitivity of $F$ to changes in the constraints.
Spline Fitting

A.F. Emery

Consider the set of data points, \( x_i, y_i \). Let us fit each interval \( x_i \) to \( x_{i+1} \) with a cubic. To ensure smoothness we will require continuity of the function and of different degrees of derivatives from one interval to the next. Normally we would use the expression

\[
y(x) = a_i + b_i x + c_i x^2 + d_i x^3 \quad \text{for} \quad x_i < x < x_{i+1}
\]

However, this is a little inconvenient when we try to evaluate the coefficients, \( a_i, b_i, c_i, d_i \). Instead, let us write the cubic in the form

\[
y(x) = s(\bar{x}) = a_i + b_i \bar{x} + c_i \bar{x}^2 + d_i \bar{x}^3 \quad \text{where} \quad \bar{x} = x - x_i
\]

Let us further define

\[
h_i = x_{i+1} - x_i \quad w = \bar{x}/h_i \quad \overline{w} = 1 - w
\]

Since the second derivative of a cubic is linear, if we write

\[
\frac{d^2 s}{dx^2} = \frac{\sigma_{i-1}(x_i - x) + \sigma_i(x - x_{i-1})}{h_i}
\]

and then integrate twice we will obtain

\[
s(\bar{x}) = w y_{i+1} + \overline{w} y_i + h_i^2 \left[(w^3 - w)\sigma_{i+1} + (\overline{w}^3 - \overline{w})\sigma_i \right]
\]

The first two terms represent the linear fit of the 2nd derivative while the third term is a correction term which varies as \( x^3 \). Note that at the points \( x_i \) and \( x_{i+1} \) we have

\[
x_i: \quad w = 0, \overline{w} = 1, s(x_i) = y_i
\]

\[
x_{i+1}: \quad w = 1, \overline{w} = 0, s(x_{i+1}) = y_{i+1}
\]

The different derivatives of \( s_i(x) \) are given by

\[
s'_i(x) = \frac{y_{i+1} - y_i}{h_i} + h_i [(3w^2 - 1)\sigma_{i+1} - (3\overline{w}^2 - 1)\sigma_i]
\]

\[
s''_i(x) = 6w\sigma_{i+1} + 6\overline{w} \sigma_i
\]

\[
s'''_i(x) = \frac{6(\sigma_{i+1} - \sigma_i)}{h_i}
\]

\[
s''''_i(x) = 0
\]

From this we see that at the point \( x_{i+1} \), the second derivative is given in terms of the functions \( s_i = s(x), x_i < x < x_{i+1} \) and \( s_{i+1} = s(x), x_{i+1} < x < x_{i+2} \) as

\[
s''_i(x_{i+1}) = 6\sigma_{i+1}
\]

\[
s''_{i+1}(x_{i+1}) = 6\sigma_{i+1}
\]

Thus the second derivatives are given uniquely in terms of \( \sigma \) and they are automatically continuous. From the equation for \( s(x) \), we see that \( s(x) \) is also continuous and defined in terms of \( y_i \) and \( \sigma_i \). Thus all that is left is to ensure that \( s' \) is continuous.
Consider two intervals, \( x_{i-1} \) to \( x_i \) and \( x_i \) to \( x_{i+1} \). Let us evaluate \( s'(x_i) \) from the left using \( s_{i-1} \) and \( s'(x_i) \) from the right, using \( s_i \), so that

\[
s'_{i-1}(x_i) = \frac{y_i - y_{i-1}}{h_{i-1}} + h_{i-1}[2\sigma_i + \sigma_{i-1}]
\]

\[
s'_i(x_i) = \frac{y_{i+1} - y_i}{h_i} + h_i[-\sigma_{i+1} - 2\sigma_i]
\]

Letting \( \Delta_i = \frac{y_{i+1} - y_i}{h_i} \) we have

\[
\Delta_{i-1} + h_{i-1}(2\sigma_i + \sigma_{i-1}) = \Delta_i - h_i(2\sigma_i + \sigma_{i+1})
\]

or

\[
h_{i-1}\sigma_{i-1} + 2(h_{i-1} + h_i)\sigma_i + h_i\sigma_{i+1} = \Delta_i - \Delta_{i-1}
\]

If there are \( n \) data points, there are \( n - 2 \) of these equations to determine the \( n \) values of \( \sigma_i \). Thus we need to obtain 2 extra equations to complete the problem.

**END CONDITIONS**

We usually use one of two different end conditions.

A **Natural Spline** for which \( s''(x_1) = s''(x_n) = 0 \). In this case

\[
\sigma_1 = \sigma_n = 0
\]

B **Constrained Spline** in which we require that the spline be a cubic which fits the first 4 and last 4 points exactly. If we define

\[
\Delta^{(1)}_i = \frac{y_{i+1} - y_i}{x_{i+1} - x_i}
\]

\[
\Delta^{(2)}_i = \frac{\Delta^{(1)}_{i+1} - \Delta^{(1)}_i}{x_{i+2} - x_i}
\]

\[
\Delta^{(3)}_i = \frac{\Delta^{(2)}_{i+1} - \Delta^{(2)}_i}{x_{i+3} - x_i}
\]

we find that

\[
\frac{\sigma_2 - \sigma_1}{h_1} = \Delta^{(3)}_1
\]

\[
\frac{\sigma_n - \sigma_{n-1}}{h_{n-1}} = \Delta^{(3)}_{n-3}
\]

For both (A) and (B), we have sufficient equations to determine all of the \( \sigma_i \) needed to evaluate the cubic spline.

The final set of equations is of the form

\[
\begin{pmatrix}
-h_1 & h_1 & 0 & 0 & 0 \\
h_1 & 2(h_1 + h_2) & h_2 & 0 & 0 \\
0 & h_2 & 2(h_2 + h_3) & h_3 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & h_n & -h_n
\end{pmatrix}
\begin{pmatrix}
\sigma_1 \\
\sigma_2 \\
\sigma_3 \\
\vdots \\
\sigma_n
\end{pmatrix} =
\begin{pmatrix}
h_1^2\Delta^{(3)}_1 \\
\Delta_2 - \Delta_1 \\
\Delta_3 - \Delta_2 \\
\vdots \\
-h_{n-1}^2\Delta^{(3)}_{n-3}
\end{pmatrix}
\]
Spline Fitting

This matrix is tridiagonal and can be solved easily using Gaussian elimination and reduction.

Note that after the problem has been solved to obtain the values of $\sigma_i$, it is generally more convenient to determine the coefficients, $a, b, c$ and $d$ and then to evaluate the cubic rather than to find the value of $x$, then $\overline{x}$, then $w$ and $\overline{w}$ and then to evaluate $s(x)$.

Some disadvantages of splines are:

1. local modification involves resolving the entire problem
2. if there is a discontinuity in the second derivative, e.g., when a circular arc is continued as a straight line, oscillations may occur.
3. the curvature of the spline sometimes varies in a non-uniform manner causing reflections when used to create surfaces.

A good reference is *Computational Geometry for Design and Manufacture* by Faux, I. D. and Pratt, M. J. Ellis Horwood Publishers