4.0 Parametric B-splines

4.1 Spline Curves

Spline curves were first used as a drafting tool for aircraft and ship building industries. A loft man's spline is a flexible strip of material, which can be clamped or weighted so it will pass through any number of points with smooth deformation.

Lobachevsky investigated b-splines as early as the nineteenth century; they were constructed as convolutions of certain probability distributions. In 1946, Schoenburg used B-splines for statistical data smoothing, and his paper started the modern theory of spline approximation. Gordon and Reisenfeld formally introduced B-splines into computer aided design [Fari97].

4.2 The B-spline basis

The underlying core of the B-spline is its basis or basis functions. The original definition of the B-spline basis functions uses the idea of divided differences and is mathematically involved. Carl de boor established in the early 1970's a recursive relationship for the B-spline basis. By applying the Leibniz' theorem, de boor was able to derive the following formula for B-spline basis functions:

$$N_i^k(u) = \frac{u - u_i}{u_{i+k-1} - u_i} N_i^{k-1}(u) + \frac{u_{i+k} - u}{u_{i+k} - u_{i+1}} N_{i+1}^{k-1}(u) \tag{4.1}$$

$$N_i^1(u) = \begin{Bmatrix} 1, u_i \le u < u_{i+1} \\ 0, otherwise \end{Bmatrix}$$

N_i^k = ith B-spline basis function of order k.

u_i = non-decreasing set of real numbers also called as the knot sequence.

u = parameter variable.

This formula shows that the B-spline basis functions of an arbitrary degree can be stably evaluated as linear combinations of basis functions of a degree lower.

The obvious defining feature of the basis function is the knot sequence u_i. The knot sequence is a set of non-decreasing real numbers. The variable u represents the active area of the real number line that defines the B-spline basis. It takes $k+1$ knots or k intervals to define a basis function. Since the basis functions are based on knot differences, the shape of the basis functions is only dependent on the knot spacing and not specific knot values.

Another distinguishing feature of the B-spline is its ability to handle cases where the knot vectors contain coincident knots. Having coincident knots, or forcing knots to be coincident, is an important step in the curve inversion process in order to ensure that the curve meets certain continuity criteria. The curve inversion procedure will be discussed in more detail later.

Figure 4.1 shows the relationship between a cubic basis function and its knot sequence. Some of the properties of the B-spline basis functions are:

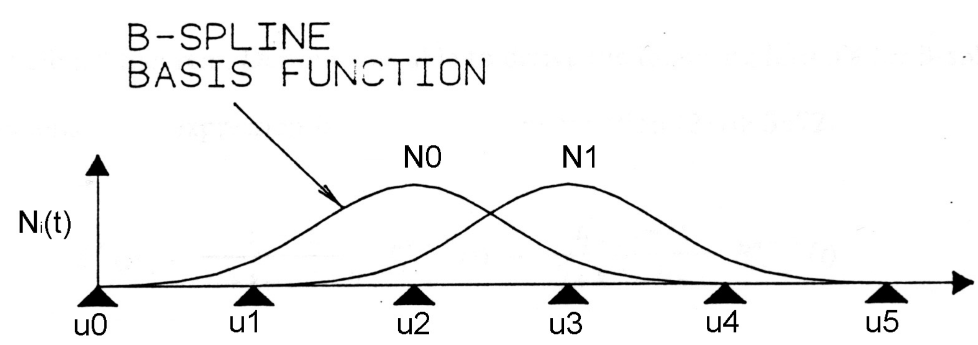

Figure 4.1: Cubic B-spline Basis Functions [Fari97].

- The sum of the B-spline basis functions for any parameter value *u* within a specified interval is always equal to 1; i.e.,

$$\sum_{i=1}^{k} N_i^k(u) \equiv 1 \tag{4.2}$$

- Each basis function is greater or equal to zero for all parameter values.
- Each basis function has only one maximum value.

4.3 The B-spline curve

B-splines are piecewise polynomials of degree n with C^{n-1} continuity at the common points between adjacent segments. B-splines result by mapping the elements of a knot sequence in parametric space into Cartesian space. A spline evaluated at a knot results in a junction point which is the common point shared by two adjacent segments. B-splines are completely specified by the curve's control points, the curve's order and the B-spline basis functions as seen in the Equation (4.3):

$$\boldsymbol{s}(u) = \sum_{i=0}^{n} d_i N_i^k(u) \qquad n \geq k\text{-}1 \tag{4.3}$$

$\boldsymbol{s}(u)$ = Points along the curve as a function of parameter u

d_i = control points also known as the weight or the point coefficients.

N_i^k = *i*th B-spline basis function of order *k*.

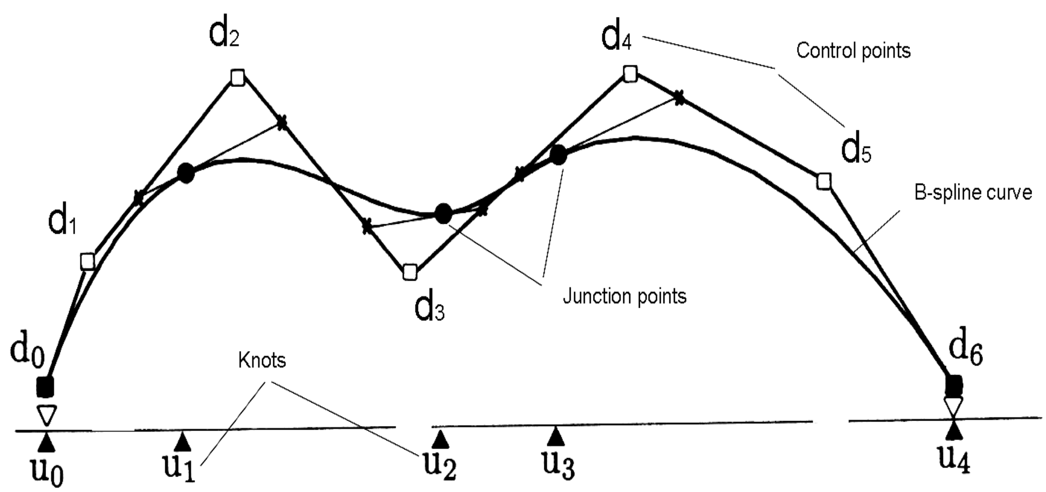

Figure 4.2: A C^2 Cubic B-spline curve with its control polygon [Fari97].

Each point on a B-spline is a weighted combination of the local control points, which form a control polygon enclosing the curve. The number of B-spline basis functions is obviously equal to the number of control points and this number is the dimension of the function space. The number of knots needed to define this function space is equal to the dimension plus its order.

As mentioned earlier, an advantage of parametric representation is that it gives the curve coordinate-system independence. The B-spline *s(u)* can thus be represented as a vector of Cartesian values which are a function of the parametric basis space as shown in Equation (4.4).

$$\boldsymbol{s}(u) = \begin{Bmatrix} x(u) = \sum_{i=0}^{n} dx_i N_i^k(u) \\ y(u) = \sum_{i=0}^{n} dy_i N_i^k(u) \\ z(u) = \sum_{i=0}^{n} dz_i N_i^k(u) \end{Bmatrix} \tag{4.4}$$

B-splines can be represented with respect to their knot sequence as uniform or non-uniform. A curve is uniform if the knot spacing between all the knots is the same. If the curve is uniform, the active portion of all the basis functions forms the same shape over each interval. If each interval is transformed to an interval between 0 and 1, a periodic basis can be used to evaluate each curve segment. Equation (4.5) shows a matrix relationship that is used to evaluate each interval of a periodic cubic curve.

$$\boldsymbol{s}_i(u) = U\,M\,D \tag{4.5}$$

$$U = [u^3 \;\; u^2 \;\; u \;\; 1] \qquad M = 1/6\begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \qquad D = \begin{bmatrix} d_{i-1} \\ d_i \\ d_{i+1} \\ d_{i+2} \end{bmatrix}$$

U = monomial basis

M = constant universal transformation matrix

D = control points involved in the *ith* interval.

If the curve is non-uniform, the knot spacing varies over the knot sequence. The previously discussed recursive algorithm (Equation 4.1) is thus necessary to evaluate the basis functions. Equation (4.6) shows the matrix relationship for each interval of a cubic non-uniform B-spline curve in the fully expanded form.

$$\mathbf{s}_i(u) = N\,D \tag{4.6}$$

u = parameter value; i = knot interval

$$N = \begin{bmatrix} \dfrac{(u_{i+1}-u)^3}{P} \\ \left\{ \dfrac{(u_{i+1}-u)^2(u-u_{i-2})}{P} + \dfrac{(u-u_{i-1})(u_{i+2}-u)(u_{i+1}-u)}{Q} + \dfrac{(u-u_i)(u_{i+2}-u)^2}{R} \right\} \\ \left\{ \dfrac{(u_{i+1}-u)(u-u_{i-1})^2}{Q} + \dfrac{(u-u_i)(u_{i+2}-u)(u-u_{i-1})}{R} + \dfrac{(u-u_i)^2(u_{i+3}-u)}{S} \right\} \\ \dfrac{(u-u_i)^3}{S} \end{bmatrix}^T$$

$P = (u_{i+1}-u_{i-2})(u_{i+1}-u_{i-1})(u_{i+1}-u_i);$ $Q = (u_{i+1}-u_{i-1})(u_{i+2}-u_{i-1})(u_{i+1}-u_i);$

$R = (u_{i+2}-u_i)(u_{i+2}-u_{i-1})(u_{i+1}-u_i);$ $S = (u_{i+2}-u_i)(u_{i+3}-u_i)(u_{i+1}-u_i);$

$$D = \begin{bmatrix} d_{i-1} \\ d_i \\ d_{i+1} \\ d_{i+2} \end{bmatrix}$$

In order to determine the spacing between the adjacent knots in a knot vector, different parametrization techniques are used. Parametrization methods are crucial for the modelling of B-splines since the spacing of the knot sequence influences the basis functions as discussed before. It amounts to defining the length of each parametric interval, which when mapped to modeling space, will define each curve segment.

There are three different methods commonly used to parametrize model curve data; uniform, chord length and centripetal. These methods are discussed below.

- Uniform

  This is the simplest type of parametrization where the knot spacing is chosen to be identical for each interval. Typically, knot values are chosen to be successive integers as shown in Equation (4.7).

  $$u_{i+1} = u_i + 1 \tag{4.7}$$

  For many cases, however, this method is too simplistic and ignores the geometry of the model data points.

- Chord Length

  This parametrization is based on the distance between the data points. The knot spacing is proportional to the distance between the data points. Equation (4.8) reflects this relationship. This parametrization more accurately reflects the geometry of the data points.

$$\frac{u_{i+1} - u_i}{u_{i+2} - u_{i+1}} = \frac{\|d_{i+1} - d_{i+1}\|}{\|d_{i+2} - d_{i+1}\|} \tag{4.8}$$

u_i = *ith* domain knot

d_i = *ith* data point

i = knot interval

- Centripetal

  This parametrization is derived from a physical analogy. It seeks to smooth out variation in the centripetal force acting on a point in motion along the curve. This requires the knot sequence to be proportional to the square root of the distance between the data points as shown in Equation (4.9).

$$\frac{u_{i+1} - u_i}{u_{i+2} - u_{i+1}} = \left[\frac{\|d_{i+1} - d_{i+1}\|}{\|d_{i+2} - d_{i+1}\|}\right]^{1/2} \tag{4.9}$$

Other parametrization methods have been investigated [Fari97]. All these methods have certain circumstantial advantage over the others. There is a trade-off between geometrical representation and computation time. Typically, chord length parametrization results in a very good compromise. In any event, each parametrization results in a different shape of the curve.

Parametric cubic B-spline curves are used in this research. Lower degree polynomials do not provide sufficient control of a curve's shape, and higher degree polynomials are computationally more cumbersome and prone to numerical error.

4.4 B-spline Surfaces

B-spline surfaces are an extension of B-spline curves. The most common kind of a B-spline surface is the tensor product surface. The surface basis functions are products of two univariate (curve) bases. The surface is a weighted sum of surface (two dimensional) basis functions. The weights are a rectangular array of control points. The following Equation (4.10) shows a mathematical description of the tensor product B-spline surface.

$$\boldsymbol{s}(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} d_{ij} N_i^k(u) N_j^l(v) \tag{4.10}$$

where,

$$N_i^k(u) = \frac{u - u_i}{u_{i+k-1} - u_i} N_i^{k-1}(u) + \frac{u_{i+k} - u}{u_{i+k} - u_{i+1}} N_{i+1}^{k-1}(u)$$

$$N_i^1(u) = \begin{Bmatrix} 1, u_i \le u < u_{i+1} \\ 0, otherwise \end{Bmatrix}$$

$$N_j^l(v) = \frac{v - v_j}{v_{j+l-1} - v_j} N_j^{l-1}(v) + \frac{v_{j+1} - v}{v_{j+l} - v_{j+1}} N_{j+1}^{l-1}(v)$$

$$N_j^1(v) = \begin{Bmatrix} 1, v_j \le v < v_{j+1} \\ 0, otherwise \end{Bmatrix}$$

s*(u,v)* = B-spline surface as a function of two variables

$\boldsymbol{d}_{ij}$ = control points

$N_i^k(u)$ = *ith* basis function of order *k* as a function of *u*

$N_j^l(v)$ = *jth* basis function of order *l* as a function of *v*

u_i, v_j = Elements of the knot sequence satisfying the relation

$$u_i \le u_{i+1}, v_i \le v_{i+1}$$

For most computer aided design purposes, as in the case of the curve, ***s****(u,v)* is a vector function of two parametric values *u* and *v*. A mathematical description of this relationship is shown below in Equation (4.11).

$$\boldsymbol{s}(u,v) = \begin{Bmatrix} x(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} d_{x\,ij} N_i^k(u) N_j^l(v) \\ y(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} d_{y\,ij} N_i^k(u) N_j^l(v) \\ z(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} d_{z\,ij} N_i^k(u) N_j^l(v) \end{Bmatrix} \tag{4.11}$$

where *x, y* and *z* are coordinates in model space.

The rectangular array of control points forms what is called a control net. Similar to the B-spline curve, the B-spline surface approximates the shape of the control net. Figure 4.3 shows a bicubic B-spline surface and the corresponding control net.

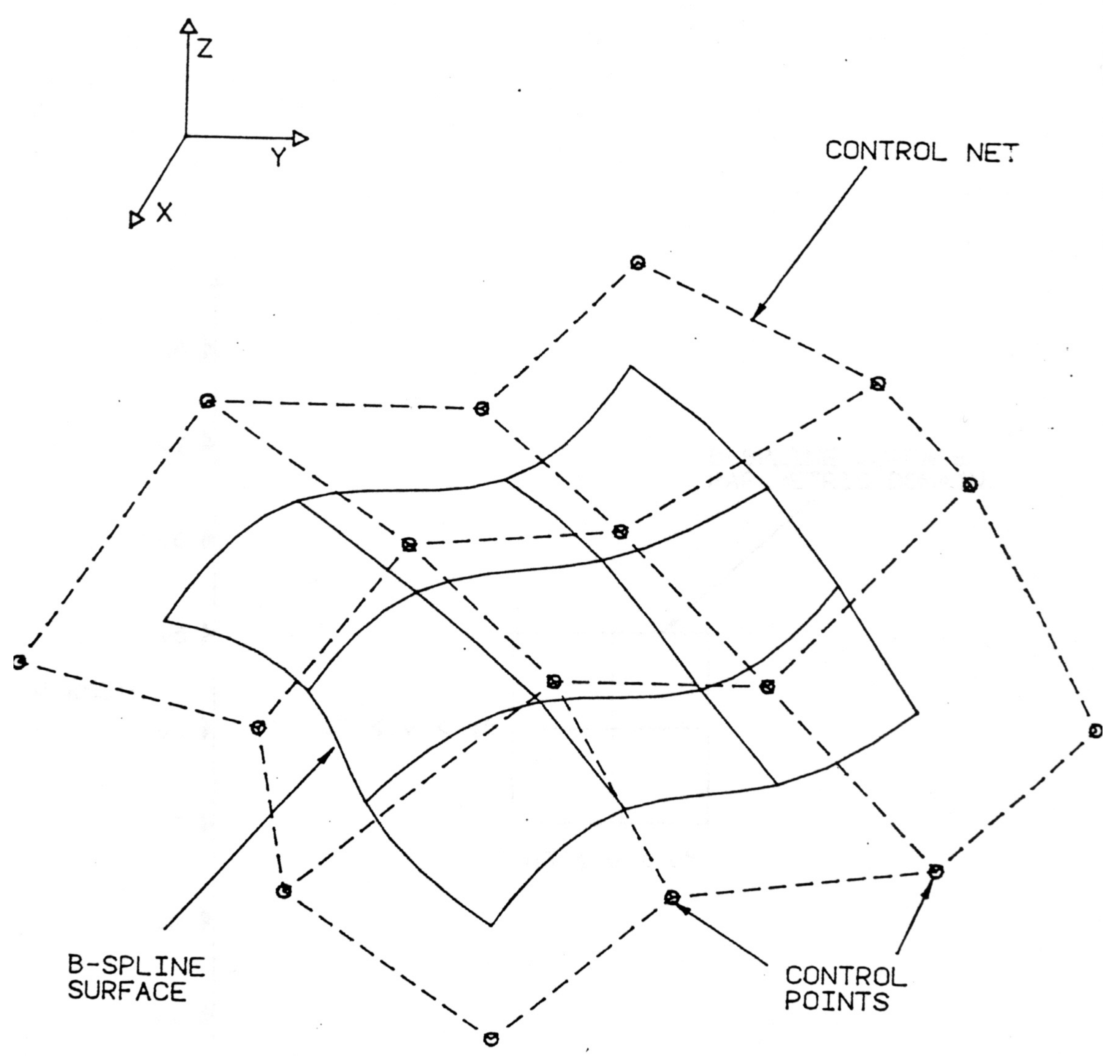

Figure 4.3: A bi-cubic B-spline surface and its control net

Similar to the B-spline curve, the B-spline surface is also a network of polynomial pieces. Each piece of the B-spline surface is a two dimensionally represented part of a surface or patch. As with a B-spline curve, each patch of a B-spline surface may be represented by a periodic relationship provided the knot spacing is uniform in each direction. This is a uniform B-spline surface. The bicubic case is described in matrix form by Equation (4.12).

$$s_{ij}(u,v) = U\,M\,D\,M^T\,V^T \tag{4.12}$$

where,

$U = [u^3 \;\; u^2 \;\; u \;\; 1]$ $\qquad$ $V = [v^3 \;\; v^2 \;\; v \;\; 1]$

$$M = 1/6\begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \qquad D = \begin{bmatrix} \boldsymbol{d}_{i-1,j-1} & \boldsymbol{d}_{i-1,j} & \boldsymbol{d}_{i-1,j+1} & \boldsymbol{d}_{i-1,j+2} \\ \boldsymbol{d}_{i,j-1} & \boldsymbol{d}_{i,j} & \boldsymbol{d}_{i,j+1} & \boldsymbol{d}_{i,j+2} \\ \boldsymbol{d}_{i+1,j-1} & \boldsymbol{d}_{i+1,j} & \boldsymbol{d}_{i+1,j+1} & \boldsymbol{d}_{i+1,j+2} \\ \boldsymbol{d}_{i+2,j-1} & \boldsymbol{d}_{i+2,j} & \boldsymbol{d}_{i+2,j+1} & \boldsymbol{d}_{i+2,j+2} \end{bmatrix}$$

If the knot sequences are not uniformly spaced, then the surface is non-uniform. The basis functions would then have to be evaluated by the recursive relationship. The non-uniform patch Equation can be represented in matrix form. Equation (4.13) shows this relationship for the bicubic case in the compacted form.

$$\boldsymbol{s}_{ij}(u,v) = \begin{bmatrix} N_o^4(u) & N_1^4(u) & N_2^4(u) & N_3^4(u) \end{bmatrix} \begin{bmatrix} \boldsymbol{d}_{i-1,j-1} & \boldsymbol{d}_{i-1,j} & \boldsymbol{d}_{i-1,j+1} & \boldsymbol{d}_{i-1,j+2} \\ \boldsymbol{d}_{i,j-1} & \boldsymbol{d}_{i,j} & \boldsymbol{d}_{i,j+1} & \boldsymbol{d}_{i,j+2} \\ \boldsymbol{d}_{i+1,j-1} & \boldsymbol{d}_{i+1,j} & \boldsymbol{d}_{i+1,j+1} & \boldsymbol{d}_{i+1,j+2} \\ \boldsymbol{d}_{i+2,j-1} & \boldsymbol{d}_{i+2,j} & \boldsymbol{d}_{i+2,j+1} & \boldsymbol{d}_{i+2,j+2} \end{bmatrix} \begin{bmatrix} N_0^4(v) \\ N_1^4(v) \\ N_2^4(v) \\ N_3^4(v) \end{bmatrix} \tag{4.13}$$

As shown in the previous two equations, the bicubic B-spline surface is affected locally by sixteen control points. Generally, a point on any B-spline surface is determined only by a local subset of control points. Also to be noted is the fact that individual isoparametric curves on a B-spline surface are B-spline curves themselves. For instance, for a surface defined by Equation (4.10), lines of constant *v,* i.e., $v = v_f$, are B-spline curves defined by the Equation (4.14):

$$\boldsymbol{s}(u, v_f) = \sum_{i=0}^{n} \boldsymbol{d}_i^f N_i^k(u) \tag{4.14}$$

Here d_i^f are the curve control points. Since this is a parameter curve which lies on the surface, it can be seen from Equation (4.10) that

$$\boldsymbol{s}(u, v_f) = \sum_{i=0}^{n} N_i^k(u) \sum_{j=0}^{m} \boldsymbol{d}_{ij} N_j^k(v_f) \tag{4.15}$$

And therefore

$$\boldsymbol{d}_i^f = \sum_{j=0}^{m} \boldsymbol{d}_{ij} N_j^k(v_f) \tag{4.16}$$

This property is used in the thesis to define isoparametric curves lying adjacent to and normal to the trim boundary for surface interrogation.

Similar to curves, the two knot vectors required to describe a surface have to be determined using one of the parametrization techniques described earlier. However, due to the fact that a B-spline surface is a tensor product and constructed of an array of control points, there are a number of point distances for each individual interval, of each knot sequence. The solution is typically to calculate a single interval distance based on the average of all of the point distances.

4.5 *Differential Geometry*

The motivation for this topic study is to describe the local curve and surface properties like curvature. The main tool used for the development of the results is the local coordinate systems, in terms of which geometric properties are easily described and studied. As discussed earlier for the case of a B-spline, a curve in $\boldsymbol{E}^3$ can be described parametrically as

$$\boldsymbol{s} = \boldsymbol{s}(u) = \begin{bmatrix} x(u) \\ y(u) \\ z(u) \end{bmatrix}, \qquad u \in [a,b] \subset \boldsymbol{R} \tag{4.17}$$

where its Cartesian coordinates *x, y, z,* are differentiable functions of *u* (see Figure 4.4). It is assumed that

$$\dot{\boldsymbol{s}} = \dot{\boldsymbol{s}}(u) = \begin{bmatrix} \dot{x}(u) \\ \dot{y}(u) \\ \dot{z}(u) \end{bmatrix} \neq 0, \qquad u \in [a,b] \subset \boldsymbol{R} \tag{4.18}$$

where the dots denotes derivatives with respect to *u*.

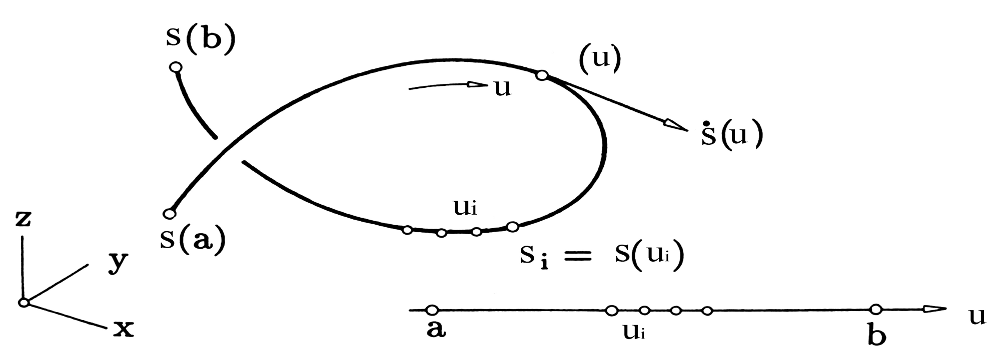

Figure 4.4: Parametric curve in space[Fari97].

An introduction of a special local co-ordinate system called the Frenet frame, linked to a point *s(u)* on the curve will significantly facilitate the description of the local curve properties at the point. The frame (or trihedron) is described by three mutually perpendicular vectors, **t**, **n** and **b** whose orientation varies as *u* traces out the curve. The vector **t** is called as the tangent vector, **n** is called main normal vector, and **b** is called binormal vector. Figure 4.5 depicts the Frenet frame.

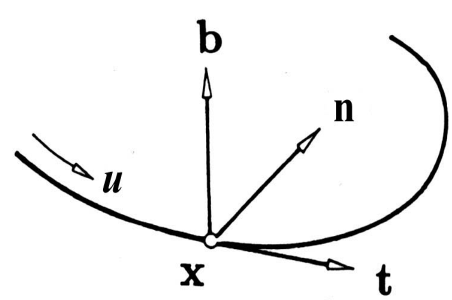

Figure 4.5: The Frenet Frame [Fari97].

The mathematical relationship of the three vectors with respect to the point ***s(u)*** is given by Equation (4.19)

$$\mathbf{t} = \frac{\dot{\boldsymbol{s}}}{\|\dot{\boldsymbol{s}}\|}, \quad \mathbf{n} = \mathbf{b} \wedge \mathbf{t}, \quad \mathbf{b} = \frac{\dot{\boldsymbol{s}} \wedge \ddot{\boldsymbol{s}}}{\|\dot{\boldsymbol{s}} \wedge \ddot{\boldsymbol{s}}\|}, \tag{4.19}$$

where "$\wedge$" denotes the cross product.

The plane spanned by the point $\boldsymbol{s}$ and the two vectors **t**, **n** is called the osculating plane ***O***. Its equation is given by

$$\det\begin{bmatrix} \boldsymbol{r} & \boldsymbol{s} & \dot{\boldsymbol{s}} & \ddot{\boldsymbol{s}} \\ 1 & 1 & 0 & 0 \end{bmatrix} = \det[\boldsymbol{r} - \boldsymbol{s}, \dot{\boldsymbol{s}}, \ddot{\boldsymbol{s}}] = 0,$$

where $\boldsymbol{r}$ denotes any point on ***O.*** Its parametric form is

$$\boldsymbol{O}\,(u,v) = \boldsymbol{s} + u\dot{\boldsymbol{s}} + v\ddot{\boldsymbol{s}} \tag{4.20}$$

Figure 4.6 explains the relation between the osculating plane and the three mutually perpendicular vectors and two other planes called the normal plane and the rectifying plane. The normal and the rectifying planes are perpendicular to each other and each of them is perpendicular to the osculating plane in turn. The normal and the binormal vectors determine the normal plane. It is that plane which passes through a point $\boldsymbol{s}(u)$ on the curve and is perpendicular to the tangent line to the curve at that point. The tangent and the binormal vectors determine the rectifying plane.

Letting the Frenet frame vary with u provides a good idea of the curve's behavior in space. The rate at which the Frenet frame moves with respect to the parameter u gives a measure of the Curvature and the Torsion of the curve. Curvature (κ) is the rate of turning of the tangen*t* vector and is given by the relation

$$\kappa = \kappa(u) = \frac{\|\dot{\mathbf{s}} \wedge \ddot{\mathbf{s}}\|}{\|\dot{\mathbf{s}}\|^3} \tag{4.21}$$

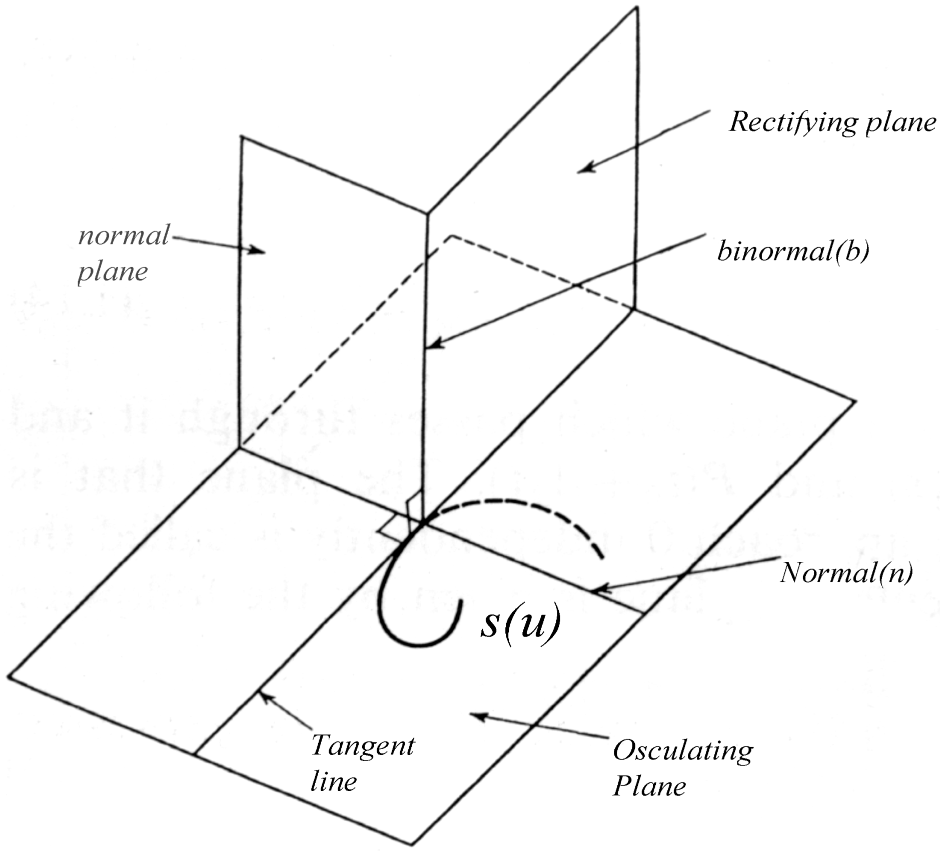

Figure 4.6: Relation between the normal plane, osculating plane and rectifying plane [Fari97].

Torsion (τ) is a measure of the amount of rotation of the osculating plane. In other words torsion is a quantity, which indicates whether the curve is twisting rapidly or slowly. It is expressed mathematically as

$$\tau = \tau(u) = \frac{\det[\dot{\mathbf{s}}, \ddot{\mathbf{s}}, \dddot{\mathbf{s}}]}{\left\| \dot{\mathbf{s}} \wedge \ddot{\mathbf{s}} \right\|^2} \tag{4.22}$$

The curvature, torsion and the three Frenet frame vectors **t**, **n**, **b** are related by the following set of formulas called the Frenet-Serret formulas (Equation 4.23). Figure 4.7 illustrates these formulas

$$\begin{aligned}\mathbf{t}' &= \qquad +\kappa\,\mathbf{n},\\ \mathbf{n}' &= -\kappa\,\mathbf{t} \qquad\qquad +\tau\,\mathbf{b},\\ \mathbf{b}' &= \qquad -\tau\,\mathbf{n}\end{aligned} \tag{4.23}$$

A point on a curve where κ = 0 is called a point of inflection. Since this point is difficult to locate in most practical cases, another measure called the curvature minima is employed in finding a point at which the curvature is close to zero and this point can be considered to be the inflection point, for all practical purposes.

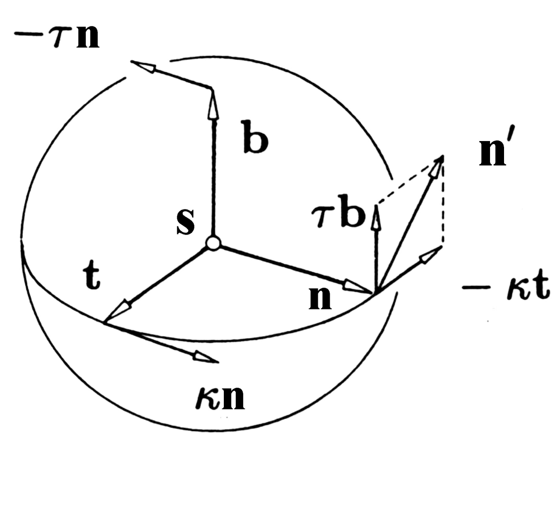

Figure 4.7: The geometric meaning of the Frenet-Serret formulas [Fari97]

A curve is said to have curvature minima at a point where the following condition holds true:

$$\kappa(u_i) - \kappa(u_{i-1}) < 0 \;\wedge\; \kappa(u_{i+1}) - \kappa(u_i) \geq 0 \tag{4.24}$$

Similarly, a curve is said to have curvature maxima at a point where the following holds true:

$$\kappa(u_i) - \kappa(u_{i-1}) > 0 \;\wedge\; \kappa(u_{i+1}) - \kappa(u_i) \leq 0 \tag{4.25}$$

4.6 Constraint-based B-spline Inversion

B-spline theory was originally developed to define a curve or a surface that approximates a set of data points. However, a designer is often more interested in using B-splines for interpolating a set of data points rather than approximating them. The inversion method addresses this issue by finding the control points for a B-spline curve or a surface given a set of data points to be interpolated. This process typically consists of setting up and solving a system of linear equations that are based on interpolation conditions that are required by the model data. Yamaguchi gives a solution for the inversion of uniform cubic splines [Yama88] while Gloudamans [Glou89] extends this to a non-uniform case. Fleming developed an approach to incorporate end constraints to the set of data points to give the curve or surface a desired level of continuity [Flem92a] [Flem92b].

For a cubic curve, the inversion process typically consists of setting up and solving for $n+2$ control points on n interpolating data points. n equations can be generated based on these data point constraints. This is accomplished through a knowledge of the behavior of B-spline basis functions. As previously described, the basis functions are only locally non zero and any cubic B-spline interval is effected by, at most, four basis functions. At each knot value u_i, however, only three basis functions are non-zero. Therefore, the value of each data point s_i can be represented by the following equation.

$$\boldsymbol{s}_i(u_{i+2}) = N^4_{i-1}(u_{i+2})\, d_{i-1} + N^4_i(u_{i+2})\, d_i + N^4_{i+1}(u_{i+2})\, d_{i+1}\ ,\ \ i = 1..n \tag{4.26}$$

This expression, however, is lacking two conditions needed to solve for the n+2 control points. These conditions, known as the curve end conditions, typically are expressed in terms of a zero curvature or a tangency condition.

Hence, the control points can be found by solving the following system of equations as shown in Equation (4.27):

$$\mathbf{S} = \mathbf{N}\ \mathbf{D} \tag{4.27}$$

$$\begin{bmatrix} \text{end} \\ \mathrm{s}_1 \\ \mathrm{s}_2 \\ \bullet \\ \bullet \\ \bullet \\ \mathrm{s}_n \\ \text{end} \end{bmatrix} = \begin{bmatrix} \text{end} & \text{condition} & & & & & 0 \\ N^4_0(u_3) & N^4_1(u_3) & N^4_2(u_3) & & & & \\ & N^4_1(u_4) & N^4_2(u_4) & N^4_3(u_4) & & & \\ & & \bullet & & & & \\ & & & \bullet & & & \\ & & & & \bullet & & \\ & & & & N^4_{n-1}(u_{n+2}) & N^4_n(u_{n+2}) & N^4_{n+1}(u_{n+2}) \\ & & & & & \text{end} & \text{condition} \end{bmatrix} \begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ \bullet \\ \bullet \\ \bullet \\ \bullet \\ d_{n+1} \end{bmatrix}$$

where,

S = Data points

N = basis function values at the knots

D = control points

Surface inversion is slightly more complex than curve inversion and involves solving for the rectangular array of control points that interpolate to model data points with respect to two different bases. This process can be described by the following Equation (4.28):

$$\mathbf{S} = \mathbf{N}(v)\ \mathbf{D}\ \mathbf{N}^{T}(u) \tag{4.28}$$

where

S represents the array of data points and end conditions in both directions.

N(v) represents the basis function matrix of the form in Equation (4.27) as a function of v.

N(u) represents the basis function matrix with respect to u.

D is the array of control points to be solved for.

In Fleming's approach to b-spline inversion, the concept of knot multiplicity is employed. Knot multiplicity amounts to adding extra knots at points where the continuity has to be changed. For instance, in order to ensure tangent continuity, one extra knot is required at the specified point. If instead a sharp corner needs to be modeled, the required knot multiplicity should be three (two extra knots should be added) to provide with point or C^0 continuity at that corner.

Each time a knot is added, the continuity at that point is reduced. However, the dimension of the curve is changed and a new basis function is created. Sine a new basis function is introduced, the number of control points to be solved for also increases which in turn increases the need for an extra constraint in order to solve for the new control points in the inversion process. These extra constraints can be provided in terms of tangency conditions or tangent vectors. For each multiple knot added to the knot sequence, a tangent vector constraint can be added to the linear system in order to solve for the extra control points. Thus, for C^1 continuity, one tangent vector is required, whereas for C^0 continuity, two tangent vectors are required; one on either side of the data point representing the abrupt change in tangency. Figure 4.8 shows the effect adding multiple knots has on the B-spline basis.

When using continuity constraints, a series of steps have to take place. First, the data points are parametrized with one of the available techniques (uniform, chord length, etc.). Next, knot multiplicity is added wherever continuity constraints are required. Once a new set of knots is defined, all basis functions are calculated so that linear systems including continuity constraints can be assembled and solved.

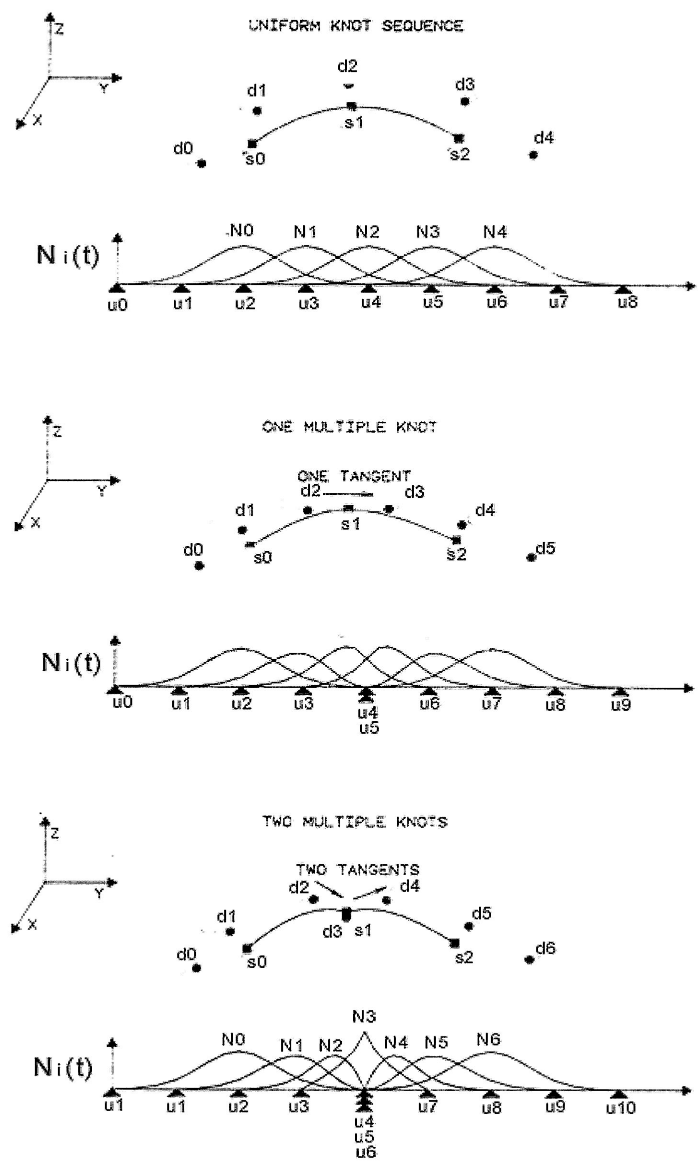

Figure 4.8: Effect of adding multiple knots on the B-spline basis [Flem92a].