CAD: Curves and Surfaces: Part III of III

In the last part, we looked at B-spline curves and how we can use the tensor product of B-spline curves to define surfaces. Now we shall look at the most popular format of storing/manipulating curve and surface data of the last decade – Non-Uniform Rational B-Splines (NURBS). First, a brief introduction to the terms:

B-Splines: because NURBS are based on the basic B-Spline definition (see below).

Rational: because the definition takes a rational form (*a/b*), where *a* and *b* are polynomials. An advantage of using rational forms is that it allows the exact representation of some popular shapes (e.g. circular arcs) that cannot be represented by B-splines (remember the parametric form for circles that we derived earlier).

Non-Uniform: because the knot vector contains a series of knots that may not be distributed uniformly over the knot interval.

First, the definition of a ***NURBS curve***:

$$C(u) = \frac{\sum_{i=0}^{n} N_{i,p}(u)\, w_i\, P_i}{\sum_{i=0}^{n} N_{i,p}(u)\, w_i} \qquad a \le u \le b,$$

where $\boldsymbol{P_i}$ are the control points, $\{w_i\}$ are the weights, and $\{ N_{i,p}(u) \}$ are the *p*-th degree B-Spline basis functions defined over the non-uniform knot vector $U = \{ a, \ldots p+1 \text{ times} \ldots, a, u_{p+1}, \ldots, u_{m-p-1}, b, \ldots p+1 \text{ times} \ldots, b\}$.

Usually, the domain of *u* is [0,1], and each weight is positive.

We can write the equation in terms of rational (function) coefficients of the control points:

$$R_{i,p}(u) = \frac{N_{i,p}(u)\, w_i}{\sum_{j=0}^{n} N_{j,p}(u)\, w_j} \qquad giving: \quad C(u) = \sum_{i=0}^{n} R_{i,p}(u) P_i$$

Just as in the case of Bezier and B-Splines, it is useful to become familiar with the properties of the (rational) basis functions of NURBS, $R_{i,p}(u)$.

NBP1. Non-negativity:

$R_{i,p}(u) \geq 0$, for all i, p and all u in [0,1].

NBP2. Partition of unity:

$\sum_{i=0}^{n} R_{i,p}(u) P_i = 1$, for all u in [0,1].

NBP3. Local Support.

$R_{i,p}(u) = 0$ except in the range $u_i \leq u < u_{i+p+1}$

Recall that the big advantage of B-splines (and also of NURBS) is that they represent piecewise (rational) polynomials – that is, however complex the curve shape may be, at any point, the curve is represented by low degree polynomials. This guarantees that changing a given control point will only change the shape of the curve in its neighborhood – not globally.

NBP4. Continuity and differentiability:

All derivatives of $R_{i,p}(u)$ exist in the interior of a knot span (since it is a rational polynomial function with non-zero denominator). At a knot-point, $R_{i,p}(u)$ is continuously differentiable p-k times, where k is the *multiplicity* (number of times the knot value is repeated) of the knot.

NBP5. general case of B-splines:

If all the weights, $w_i = 1$, then the NURBS becomes the usual B-spline curves.

Using the definition and the above properties, we can easily derive some useful properties of NURBS:

NP1. $\boldsymbol{C}(0) = \boldsymbol{P}_0$, and $\boldsymbol{C}(1) = P_n$

That is, the curve begins at the first control point, and ends at the last control point.

NP2. Affine invariance

Since the curve depends linearly on each control point – hence we can apply any affine transformation (especially rotation, translation) to the curve by just applying it to the control points.

NP3. Convex Hull Property

The curve lies entirely within the convex hull of the control points. This follows from the basis function properties of *partition of unity* and *non-negativity*, and the definition of the convex hull.

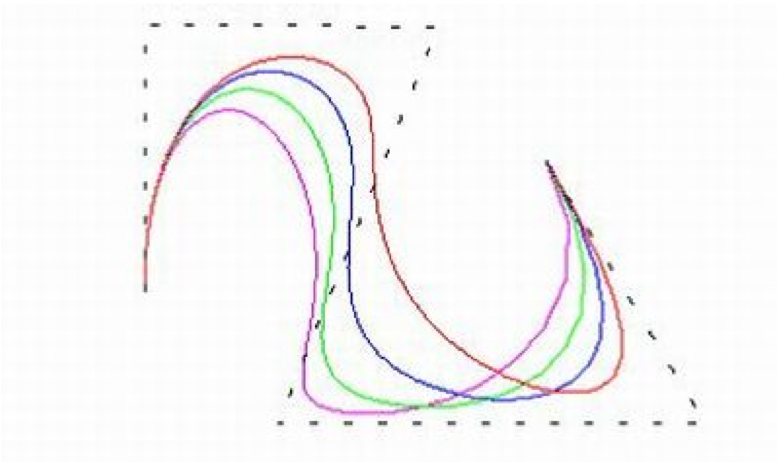
In fact, for NURBS, it can be shown that for u in [u_i, u_{i+1}), $\boldsymbol{C}(u)$ lies entirely inside the convex hull of the control points $P_{i-p}, \ldots, P_i$. To prove this, we also need the property NBP3 above.

NP4. Special cases:
A NURBS curve with no interior knots is a rational Bezier curve; further, if each weight = 1, it is a Bezier curve.
If each weight = 1, but there exist interior knots, then the NURBS is a B-spline curve.

NP5. Local Control:
A change in position or weight of the *i*-th control point, $\boldsymbol{P_i}$, will only change the curve in the interval $[u_i, u_{i+p+1})$.
Also note that increasing the weight tends to 'pull' the curve towards the corresponding control point; decreasing the weight 'pushes' the curve away from the control point.

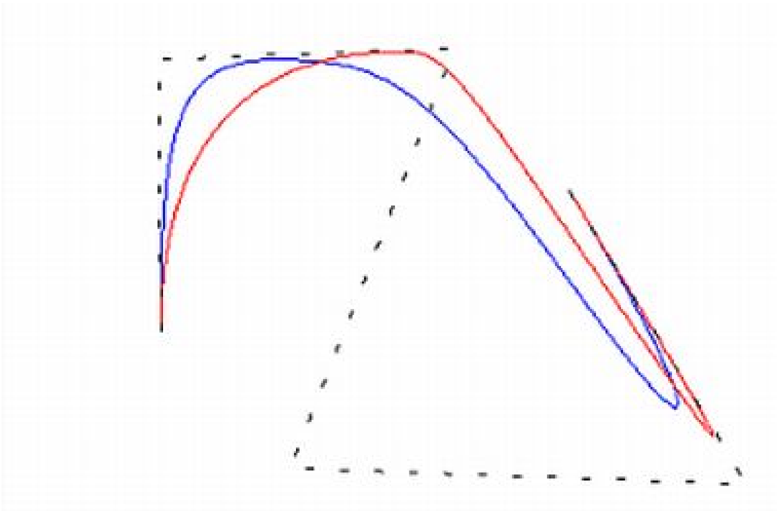
We now look at a few examples to gain familiarity with the definitions and the properties of NURBS curves.

Example 1. The effect of weight.
The figure below shows a NURBS curve with the following data:
n = 5, p = 3, U = {0, 0, 0, 0, 0.5, 0.75, 1, 1, 1, 1}, P[i] =
{(0.0 0.0 0.0), (0.0 1.0 0.0), (1.1 1.0 0.0), (0.5 -0.5 0.0), (2.1 -0.5 0.0), (1.5 0.5 0.0) }
Four NURBS curves are drawn, with the weight for P[3] changing as follows:
red curve = 0.5, blue curve = 1.0, green curve = 2.0, cyan curve = 4.0.



Example 2. The effect of low weights.
Using the same NURBS description as above, we let weight of P[3] = 0.01. The figure below shows the resulting curve (red color).
To reduce the sharp turn near P[4], we attempt decreasing the weights of neighboring points as follows: P[2] = 0.2, P[4] = 0.2, to get the curve in blue.

The changing of the shape by change of weights requires a lot of practice before we can develop a good intuition about how the curve will behave.



NURBS Surfaces

We saw earlier that B-spline surfaces may be defined as a mapping using tensor products of two B-spline curves and a rectangular mesh of control points. Likewise, NURBS surfaces are defined using the tensor product of two NURBS curves.
As before, this is a big limitation from all possible real-valued vector functions of two parameters – yet it is a superset of B-Spline surfaces. In fact, for most practical designs, NURBS are considered to be fairly powerful – they can be used to represent complex shapes, and allow easy interface for several important CAD functional requirements. These requirements include ease/efficiency of computing the surface coordinates, computing derivatives, changing shape of the surface, local control on the shape using either the control mesh, or the weights, or the degree of the polynomial etc.
Since NURBS are piecewise rational polynomials, they allow us exact representations of some important surface/curve types (e.g. sphere/circle). Why is that useful ? Because we can *potentially* use the same representation scheme for all surfaces of a solid model, without having to write separate “case-based” functions to handle each different surface type in a CAD program.

We first look at the definition, and then the properties of NURBS surfaces.

A NURBS surface of degree *p* in the *u*-direction and q in the *v*-direction is a piecewise vector function, defined as:

$$S(u,v) = \frac{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)\, N_{j,q}(v)\, w_{i,j}\, P_{i,j}}{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)\, N_{j,q}(v)\, w_{i,j}}, \qquad 0 \le u,v \le 1$$

As for the B-Splines, the $P_{i,j}$ form a $(n+1)x(m+1)$ mesh of control points; each control point is associated with a positive, real weight, $w_{i,j}$; and the $N_{i,p}$ and $N_{j,q}$ are the usual B-spline basis functions defined over the knot spans:
$U = \{ 0,\ldots p+1 \text{ times}\ldots, 0, u_{p+1}, \ldots, u_{r-p-1}, 1, \ldots p+1 \text{ times} \ldots, 1\}$
$V = \{ 0,\ldots q+1 \text{ times}\ldots, 0, u_{q+1}, \ldots, u_{s-q-1}, 1, \ldots q+1 \text{ times} \ldots, 1\}$
where $r = n+p+1$ and $s = m+q+1$

For notational convenience, we sometimes write the equation in terms of the (rational) coefficient functions:

$$R_{i,j}(u,v) = \frac{N_{i,p}(u)\ N_{j,q}(v)\ w_{i,j}}{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)\ N_{j,q}(v)\ w_{i,j}}$$, and:

$$S(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} R_{i,j}(u,v)\ P_{i,j}$$

By and large, all the (useful) properties of B-spline surfaces are carried over to NURBS surfaces. In fact, it is easy to see that when each $w_{i,j} = 1$, the summation in the denominator is also equal to 1 (partition of unity property of B-spline basis functions), and therefore the NURBS will become a B-Spline surface. Further, if the knot vectors are restricted to the form { 0, …, 0, 1, …, 1}, then the basis functions become the Bezier basis functions of degree n, and so $\boldsymbol{S}(u,v)$ is a Bezier surface.

Let us list the basic useful properties of NURBS surfaces, and then take a look at some simple examples.

NSP1. Non-negativity
The rational basis functions, $R_{i,j}(u,v) \geq 0$ for all i, j, u and v.

NSP2. Partition of Unity
$\sum_{i=0}^{n}\sum_{j=0}^{m} R_{i,j}(u,v) = 1$ for all (u,v) in [0,1]x[0,1]

NSP3. Local support
$R_{i,j}(u,v) = 0$ whenever u is outside $[u_i, u_{i+p+1})$ and v is outside $[v_j, v_{j+q+1})$

NSP4. A useful observation for computations of surface points is the following:
In the parameter-space rectange, $[u_r, u_{r+1})x[v_s, v_{s+1})$, at most $(p+1)(q+1)$ basis functions are non-zero. Only the $R_{i,j}(u,v)$ corresponding to $r\text{-}p \leq i \leq r$ and $s\text{-}q \leq j \leq s$ are non-zero.

NSP5. Convex Hull Property
As a result of the above three properties of the NURBS basis functions, we can derive the convex hull property – the entire NURBS surface lies inside the convex hull of the control mesh.

NSP6. Corner point interpolations
$R_{0,0}(0, 0) = R_{n,0}(1, 0) = R_{0,m}(0, 1) = R_{n,m}(1, 1) = 1$
Hence the NURBS surface coincides with the four corner points of the control mesh.

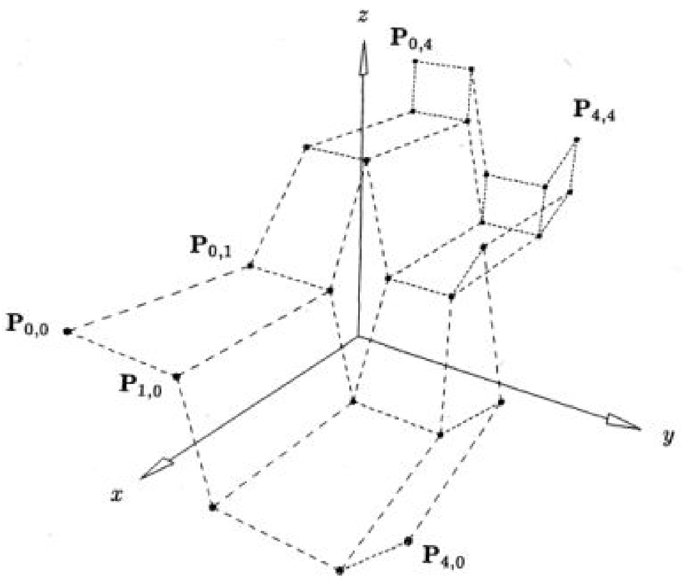
NSP7. Affine Invariance
The surface can be subject to any affine transformation by merely applying the transformation to (each point) of the control mesh.

NSP8. Shape modifications
The shape of a NURBS surface can be modified by several methods.
(a) Decreasing the degree of the NURBS surface causes the surface to move closer to the control mesh. It also becomes less 'smooth' (see NSP8).
(b) Increasing (decreasing) the weight of $P_{i,j}$ will pull the surface closer to (push the surface further from) $P_{i,j}$.
(c) Moving the control point(s)

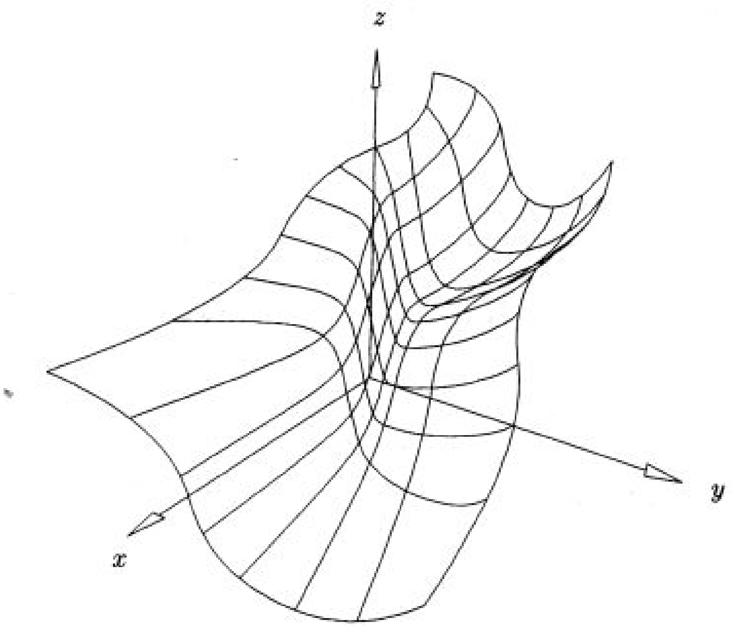
Note that due to the local control properties, moving a control point $P_{i,j}$, or changing the weight $w_{i,j}$ will only change the surface shape within the region mapped from $[u_i, u_{i+p+1})x[v_j, v_{j+q+1})$. This is the Local support property.

NSP9. Differentiability
In the interior of the parameter space rectangles formed by the *u*- and *v*-knot lines, all partial derivatives of $R_{i,j}(u,v)$ exist.
At a u-knot of multiplicity k, the basis function is *p*-*k* times differentiable with respect to *u*; at a *v*-knot,it is *q*-*k* times differentiable with respect to *v*.

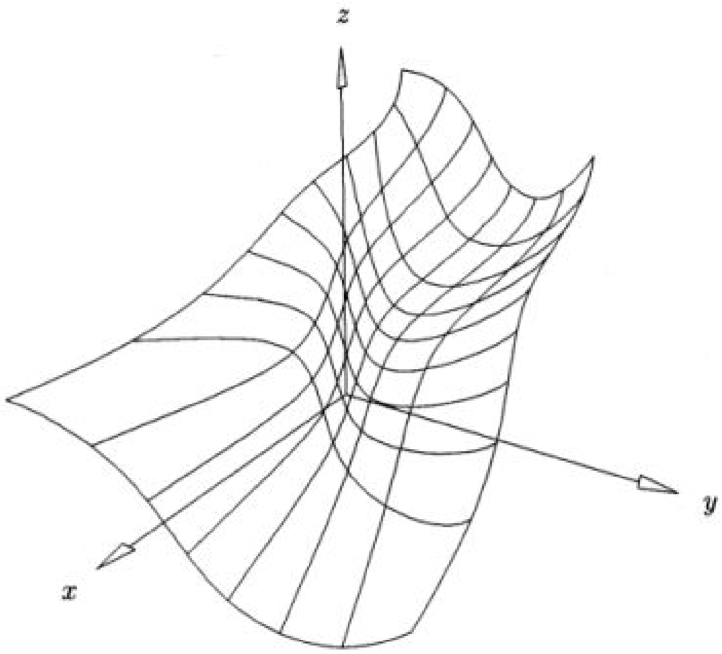
Example 1. Same control mesh ,different degree
The following three figures show the effect of changing the degree of the basis functions used to define the surface: For the same control mesh, the bi-cubic surface is much ‘smoother’ than the bi-quadratic. The bi-quadratic follows the control mesh much closer.



(a) The control mesh



(b) Bi-quadratic (degree 2 in *u* and *v*); $w_{11} = w_{12} = w_{21} = w_{22} = 10$, all other $w_{ij} = 1$;
$U = V = \{ 0, 0, 0, 0.33, 0.67, 1, 1, 1\}$

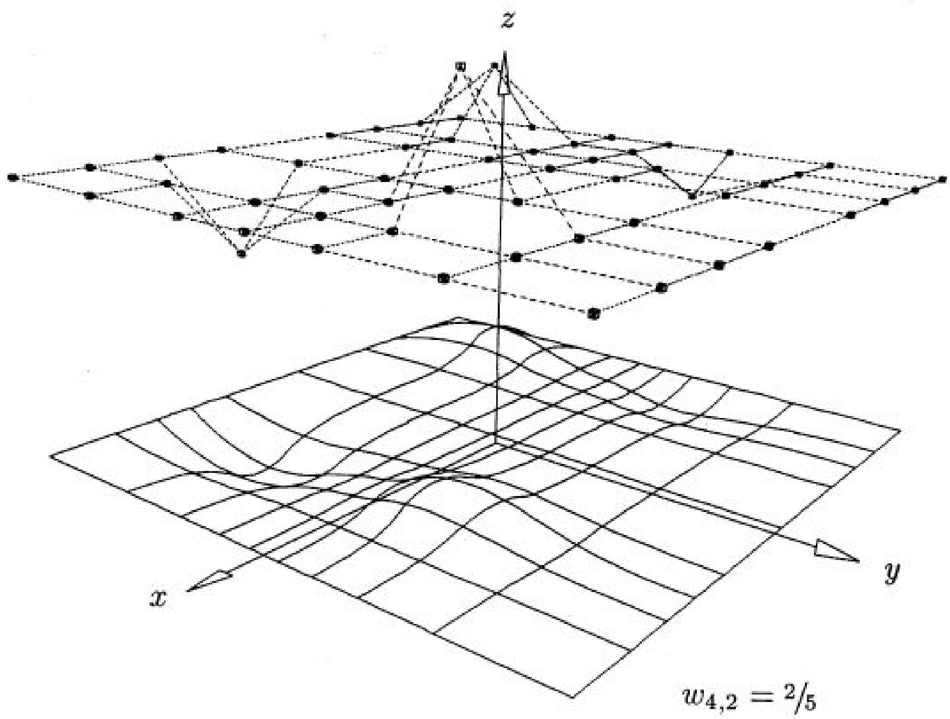


(c) Bi-cubic (degree 3 in *u* and *v*); $w_{11} = w_{12} = w_{21} = w_{22} = 10$, all other weights = 1;
$U = V = \{ 0, 0, 0, 0, 0.5, 1, 1, 1, 1\}$

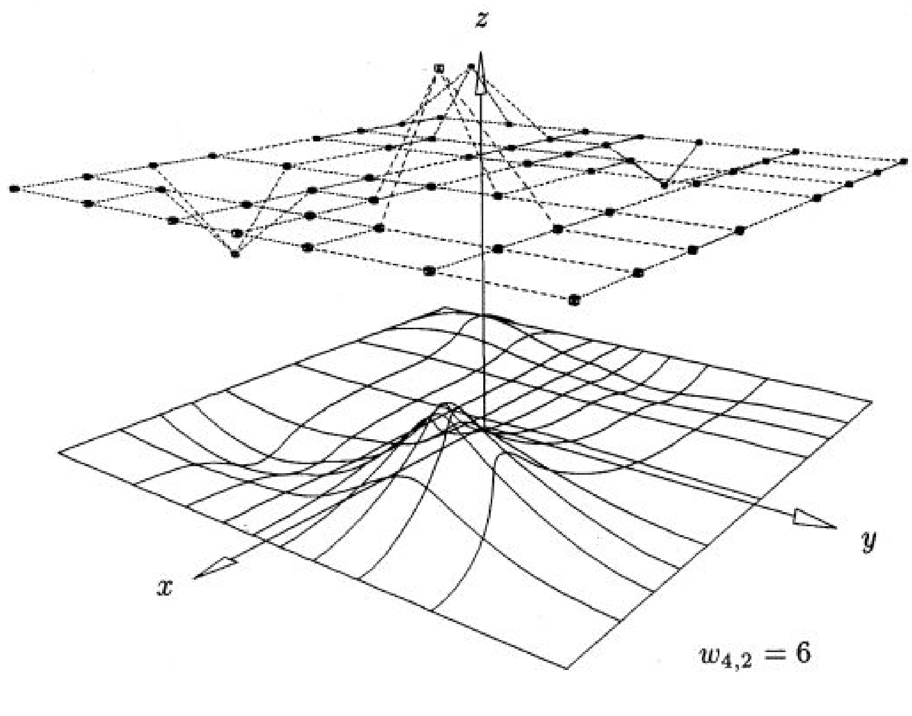
Example 2. Changing the weight, and see local shape change effect.

This example shows the effect of changing the weight of a control point, in this case, P[4][2]. Notice how the shape of the surface in regions far from the affected point stays unchanged (see the little bulge due to P[1][7] is unchanged due the changing weights at P[4][2]).

[Note: in the following two figures, the control mesh is drawn offset in the z-direction from the surface to show the surfaces clearly.]



(a) Cubic x quadratic surfaces; w_{42} = 2/5, all other w_{ij} = 1;
U = { 0, 0, 0, 0, .25, .5, .75, 1, 1, 1, 1}, V = {0, 0, 0, .2, .4, .6, .6, .8, 1, 1, 1}



(b) Cubic x quadratic surfaces; w_{42} = 6, all other w_{ij} = 1;
U = { 0, 0, 0, 0, .25, .5, .75, 1, 1, 1, 1}, V = {0, 0, 0, .2, .4, .6, .6, .8, 1, 1, 1}