B SPLINES
& 3D MODELING

Eric Rowe

Don Hatori

INTRODUCTION

- Bezier Curves

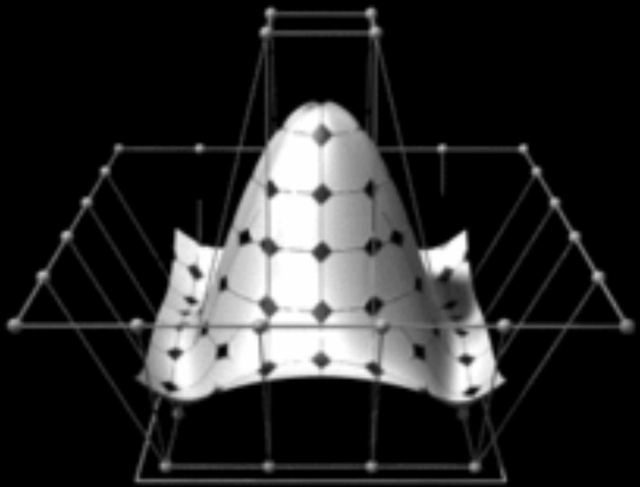
  are parametric curves used in modeling

- B-Splines

  are a special case of Bezier curves

- NURBS

  are a special use of B-Splines and Bezier curves



BEZIER CURVES

$$B_{i,n}(u) = \binom{n}{i} u^i (1-u)^{n-i}$$

- Bezier curves use control points to interpolate
- Bernstein polynomials give the basis function. Here, i is control point that can be weighted, n is the degree of the curve, and u the parameter that goes from 0 to 1
- The de Casteljau algorithm solves for any point of a Bezier curve. We subdivide the control points to get “new” control points

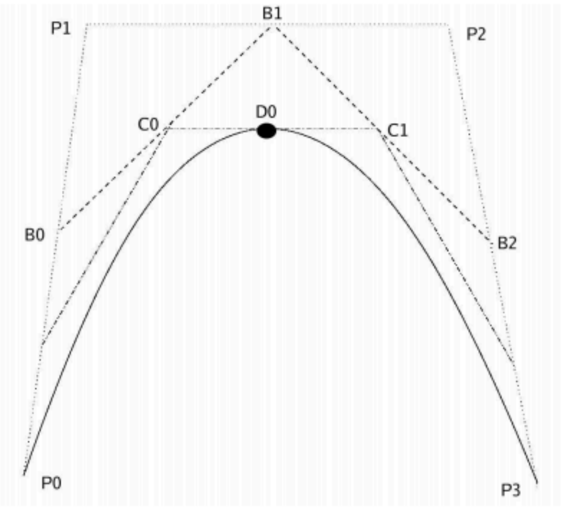


Figure 2.7: Illustration of de Casteljau's algorithm

Let $P_{0,j}$ be P_j for $j = 0, 1, \ldots, n$ where n is the degree of the curve, $i = 1, 2, \ldots, n$ and $j = 0, 1, \ldots, n-i$. Point $P_{i,j}$ is then computed as

$$P_{i,j} = (1-u)P_{i-1,j} + uP_{i-1,j+1} \quad (2.8)$$

BEZIER CURVES

- The Bezier curve is defined by a sum. To evaluate the curve from 0 to 1, we iterate n + 1 control points and compute
- Rational Bezier curves extend each control point by one coordinate to represent the scalar weight
- Bezier surfaces are just an extension of Bezier curves in two parametric directions. The surface can also be rationalized.

curve definitions

$$C(u) = \sum_{i=0}^{n} P_i B_{i,n}(u)$$

$$C(u) = U M_B G_B$$

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

rationality

$$C(u) = \frac{\sum_{i=0}^{n} w_i P_i B_{i,n}(u)}{\sum_{i=0}^{n} w_i B_{i,n}(u)}$$

$$C(u,v) = \frac{\sum_{i=0}^{n} \sum_{j=0}^{m} w_{i,j} P_{i,j} B_{i,n}(u) B_{j,m}(u)}{\sum_{i=0}^{n} \sum_{j=0}^{m} w_{i,j} B_{i,n}(u) B_{j,m}(u)}$$

B-SPLINES

- A B-Spline curve of degree m with n control points consists of n – m Bezier segments
- Knots are the join points between segments. In general, there are n + m + 1 knots. Uniform splines have evenly spaced knots; non-uniform splines do not
- Cox de Boor algorithm recursively computes the basis function to any uniform or non-uniform B-spline curve of degree n
- B-Spline is defined by a sum

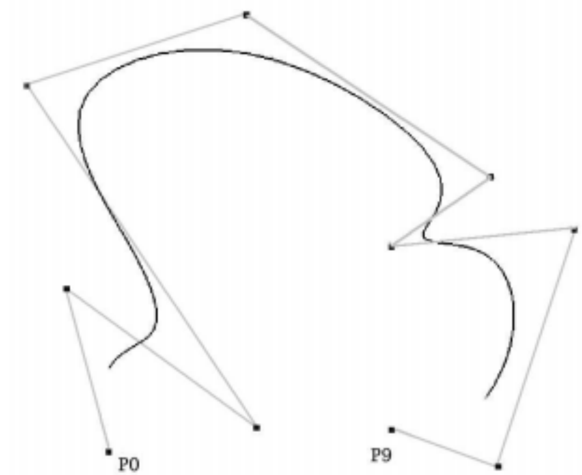


Figure 3.1: Uniform B-spline curve

$$B_{i,1}(t) = 1.0 \text{ if } t_i \leq t \leq t_{i+1}, \text{ else } 0.0$$

$$B_{i,2}(t) = \frac{t-t_i}{t_{i+1}-t_i} B_{i,1}(t) + \frac{t_{i+2}-t}{t_{i+2}-t_{i+1}} B_{i+1,1}(t)$$

$$B_{i,n}(t) = \frac{t-t_i}{t_{i+n-1}-t_i} B_{i,n-1}(t) + \frac{t_{i+n}-t}{t_{i+n}-t_{i+1}} B_{i+1,n-1}(t)$$

$$W^m(u) = \sum_{i=0}^{n} P_i B_{i,m}(u)$$

MATLAB DEMONSTRATION

BEZIER VS B-SPLINE

- B-Spline is more computational
- B-Splines offer more control and flexibility than Bezier curves
- B-splines have local control while Bezier curves do not

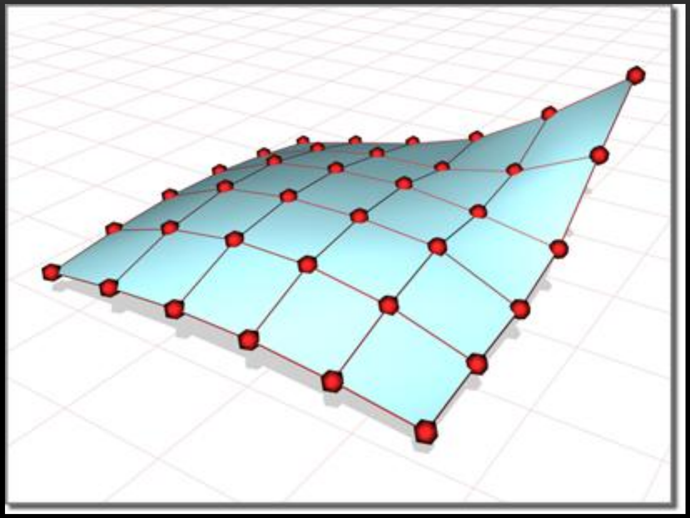
3D Cubic B–Spline Curves

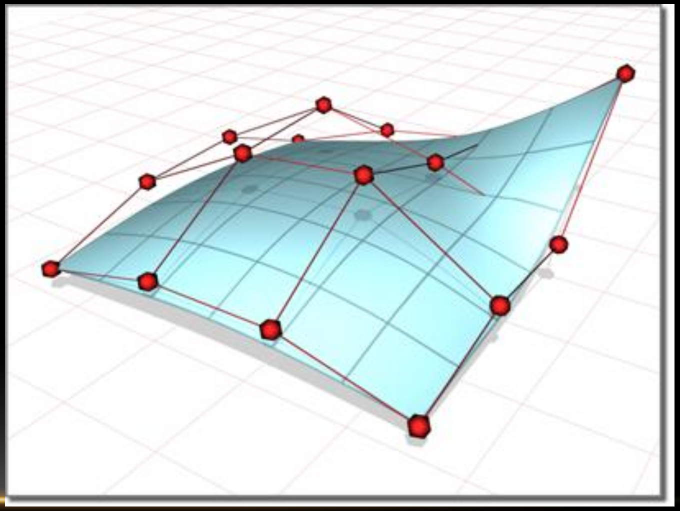
Jeff Bryant

From the Wolfram Demonstrations Project

WHAT IS NURBS?

- Non Uniform Rational B Spline: Allows control of curvatures and smoothness; defined by control points and weights
- Non Uniform: Some sections of a defined shape can be shortened or elongated relative to other sections.
- Rational: The ability to give more weight to some points in the shape than other points
- B-Spline: Based on 4 local functions or control points that lie outside of the curve itself





WHY IS THE USE OF NURBS SIGNIFICANT?

- NURBS provides local support like B-Splines, and a soft look using control points like Bezier curves.
- NURBS uses weighted control points, which makes the NURBS curves rational
- Generally, more control points gives better approximations, but only certain classes of curves can be represented exactly with finite number of control points
- The fact that NURBS features a scalar weight for each control point allows for more control over the shape of the curve without increasing the number of control points

THE NURBS EQUATIONS

General form of a NURBS curve

$$C(u) = \sum_{i=1}^{k} \frac{N_{i,n} w_i}{\sum_{j=1}^{k} N_{j,n} w_j} \mathbf{P}_i = \frac{\sum_{i=1}^{k} N_{i,n} w_i \mathbf{P}_i}{\sum_{i=1}^{k} N_{i,n} w_i}$$

Which may be written as:

$$C(u) = \sum_{i=1}^{k} R_{i,n}(u) \mathbf{P}_i$$

where:

$$R_{i,n}(u) = \frac{N_{i,n}(u) w_i}{\sum_{j=1}^{k} N_{j,n}(u) w_j}$$

are the rational basis functions.

General form of a NURBS surface

A NURBS surface is obtained as the tensor product of two NURBS curves:

$$S(u,v) = \sum_{i=1}^{k} \sum_{j=1}^{l} R_{i,j}(u,v) \mathbf{P}_{i,j}$$

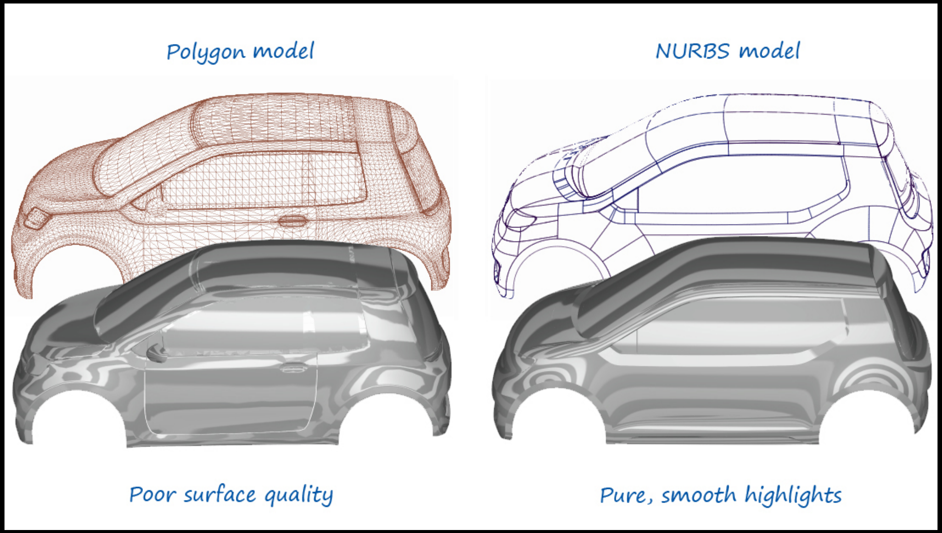
with

$$R_{i,j}(u,v) = \frac{N_{i,n}(u) N_{j,m}(v) w_{i,j}}{\sum_{p=1}^{k} \sum_{q=1}^{l} N_{p,n}(u) N_{q,m}(v) w_{p,q}}$$

as rational basis functions.

WHY AND HOW ARE NURBS USEFUL?

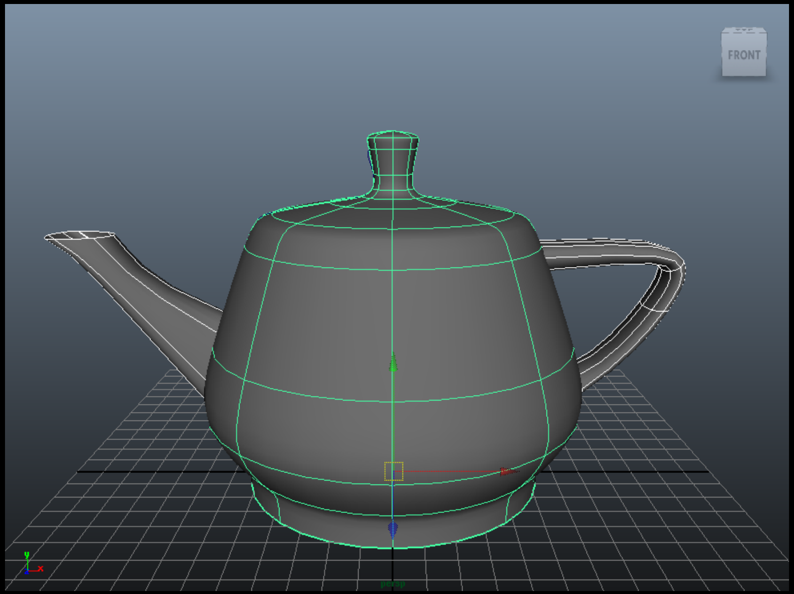
- One common mathematical form for standard and free form shapes
- Flexible
- Reduced memory consumption for storage
- Quick evaluation by numerically stable algorithms
- Can apply operations to the NURBS curve through their control points
- Moving a control point only changes the model slightly
- Used in engineering, product design, and depictions of mathematically exact objects



APPLICATIONS

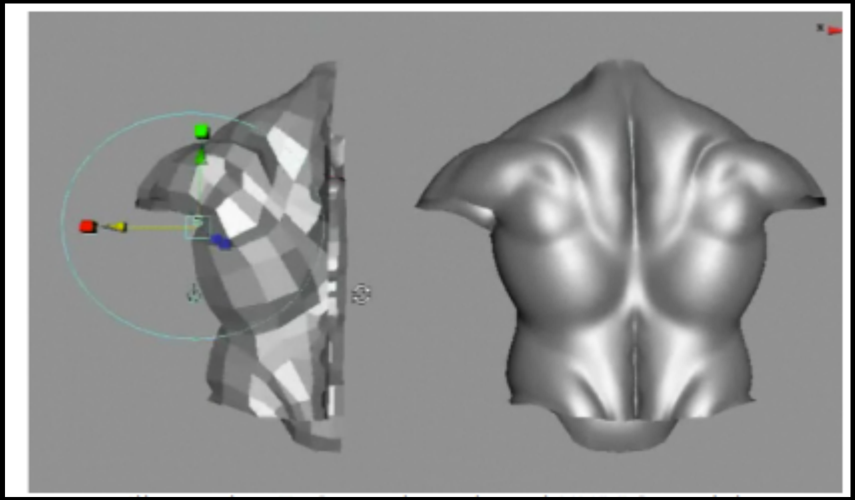
- Camera movement through space uses splines to avoid jerky movements
- NURBS provides excellent control for smooth and sharp surfaces in topology
- Object modeling

  Utah teapot



SUMMARY/CONCLUSION

- Bezier curves are helpful for modeling but have issues with local control
- B-Splines have local control and help alleviate issues Bezier curves have
- NURBS combines Bezier and B-Spline traits to make smooth curves with local control



REFERENCES

- http://www.doc.ic.ac.uk/~dfg/AndysSplineTutorial/BSplines.html
- https://www-vs.informatik.uni-ulm.de/teach/ss10/tc/pub/3D_Modeling_Paper.pdf
- http://www.uio.no/studier/emner/matnat/ifi/INF-MAT5340/v07/undervisningsmateriale/kap2.pdf
- http://fei.edu.br/~psergio/CG_arquivos/Superficies.pdf