CURVES Basics

Prof. Janakarajan Ramkumar
Professor
Department of Mechanical & Design Program
IIT Kanpur, India.
Free Form Surfaces

• Planar surface
  • A planar surface is a flat 2D surface.

• Curved surface
  • Single curved surface: It is a simple curved surface.
  • Double curved surface: It is a complex surface generated not by a straight line but a curved surface.
  • Ruled surface: Ruled surface is a surface constructed by transitioning between two or more curves by using linear blending between each section of the surface.
Free Form Surfaces

Model generated using the surfaces

https://www.pinterest.co.uk/pin/465207836504089607/
Free Form Surfaces

Types of surfaces used in Geometric Modelling

Classification of surfaces

- Planar surfaces
  - Plane
    - Polygon Polyhedra
  - Single Curved
    - Cylinder Cone
- Curved surfaces
  - Double curved
    - Sphere Ellipsoid Paraboloid Torus
- Free-form surfaces
  - Coons surface
    - B-spline Bezier surface Nurbs Fractals
  - Ruled surfaces
  - Lofted surfaces
Free Form Surfaces

Ruled surface on the left is shown the curves from which the ruled surface on the right is formed.
Free Form Surfaces

Two different ruled surfaces formed between two space curves
“Computers can’t draw curves.”

The more points/line segments that are used, the smoother the curve.
Why have curves?

- Representation of “irregular surfaces”
- Example: Auto industry (car body design)
  - Artist’s representation
  - Clay / wood models
  - Digitizing
  - Surface modeling (“body in white”)
  - Scaling and smoothening
  - Tool and die Manufacturing
Curve representation

- Problem: How to represent a curve easily and efficiently
- “Brute force and ignorance” approaches:
  - storing a curve as many small straight line segments
  - doesn’t work well when scaled
  - inconvenient to have to specify so many points
  - need lots of points to make the curve look smooth
  - working out the equation that represents the curve
  - difficult for complex curves
  - moving an individual point requires re-calculation of the entire curve
Solution - Interpolation

• Define a small number of points

• Use a technique called “interpolation” to invent the extra points for us.

• Join the points with a series of (short) straight lines
The need for smoothness

• So far, mostly polygons can approximate any geometry, but Only approximate

• Need lots of polygons to hide discontinuities Storage problems

• Math problems

• Not very convenient as modeling tool

• Gets even worse in animation
Requirements

• Want mathematical smoothness

• Some number of continuous derivatives of P

• Local control

• Local data changes have local effect

• Continuous with respect to the data

• No wiggling if data changes slightly

• Low computational effort
A solution

- Use SEVERAL polynomials
- Complete curve consists of several pieces
- All pieces are of low order
  - Third order is the most common
- Pieces join smoothly
- This is the idea of spline curves
  - or just “splines”
Continuity

- **Parametric continuity Cx**
  - Only P is continuous: C0
    - Positional continuity
  - P and first derivative dP/du are continuous: C1
    - Tangential continuity
  - P + first + second: C2
    - Curvature continuity

- **Geometric continuity Gx**
  - Only directions have to match
Order of continuity

\[ G^{(0)} \] 0th order continuity

\[ C^{(0)} \]

\[ G^{(1)} \] 1st order continuity

\[ C^{(1)} \]

\[ G^{(2)} \] 2nd order continuity

\[ C^{(2)} \]

Rao, CAD/CAM Principles and Applications, 2010, TMH
Often designers will have to deal with information for a given object in the form of coordinate data rather than any geometric equation.

In such cases it becomes necessary for the designers to use mathematical techniques of curve fitting to generate the necessary smooth curve that satisfies the requirements.

Airfoil Section Curve fitted with Data Points
## Comparison of curve fitting methods

<table>
<thead>
<tr>
<th>Interpolation methods</th>
<th>Best fit methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>It is necessary that the curve produced will have to go through all the data points</td>
<td>The curve will not pass through all the points, but will result in a curve that will be closest to as many points as possible.</td>
</tr>
<tr>
<td>Cubic Splines, and Lagrange interpolation methods are used.</td>
<td>Regression and least square methods are used for the purpose. Bezier curves also fall in this category.</td>
</tr>
<tr>
<td>Shape of the curve is affected to a great extent by manipulating a single data point.</td>
<td>It is possible to have a local modification easily by tweaking a single point where the behaviour is more predictable.</td>
</tr>
</tbody>
</table>
Synthetic curves

- Easy to enter the data and easy to control the continuity of the curves to be designed.

- Requires much less computer storage for the data representing the curve.

- Having no computational problems and faster in computing time.
  - Bezier curves
  - Hermite cubic spline
  - B-spline curves
  - Rational B-splines (including Non-uniform rational B-splines – NURBS)
Beziers curves

- A Bezier curve is a **mathematically defined curve** used in two-dimensional graphic applications.
- The curve is defined by points: the **initial position** and the **terminating position** (which are called "anchors") and **middle points** (which are called "handles or control points").
- The **shape** of a Bezier curve can be **altered** by moving the **control points**.

https://en.wikipedia.org/wiki/File:B%C3%A9zier_3_big.gif
Bezier curves

Bezier curves and the Associated Control Polygon

https://plus.maths.org/content/bridges-string-art-and-bezier-curves

Rao, CAD/CAM Principles and Applications, 2010, TMH
Bézier curves

Simple Bezier Curve

Quadratic Bazier Curve

Cubic Bazier Curve
Bézier curves

A linear Bézier curve is a line segment joining two control points $b_0(p_0, q_0)$ and $b_1(p_1, q_1)$, and parametrized by

$$(x(t), y(t)) = (1 - t)(p_0, q_0) + t(p_1, q_1), \quad \text{for } t \in [0, 1],$$

so that $x(t) = (1 - t)p_0 + tp_1$, and $y(t) = (1 - t)q_0 + tq_1$. Letting $B(t) = (x(t), y(t))$, the curve can be written in the vector form

$$B(t) = (1 - t)b_0 + tb_1.$$

Example

The Bézier form for the linear segment passing through points $b_0(1, 2)$ and $b_1(3, 4)$ is $B(t) = (1 - t)b_0 + tb_1 = (1 - t)(1, 2) + t(3, 4)$. Hence $x(t) = (1 - t) + 3t = 1 + 2t$ and $y(t) = 2(1 - t) + 4t = 2 + 2t$. 
Bézier curves

Then the *quadratic Bézier curve* is defined to be

\[ B(t) = (1 - t)^2(p_0, q_0) + 2(1 - t)t(p_1, q_1) + t^2(p_2, q_2), \quad \text{for } t \in [0, 1]. \]

The starting point of the curve is \( B(0) = b_0 \) and the finishing point is \( B(1) = b_2 \). The curve can be expressed in the parametric form \((x(t), y(t))\) where

\[
\begin{align*}
x(t) &= (1 - t)^2p_0 + 2(1 - t)t p_1 + t^2p_2, \quad \text{and} \\
y(t) &= (1 - t)^2q_0 + 2(1 - t)t q_1 + t^2q_2.
\end{align*}
\]

The triangle \( b_0b_1b_2 \) obtained by joining the control points with line segments, in their prescribed order, is called the *control polygon*. 
Bézier curves

Example

The parametric form of the quadratic Bézier curve $B(t)$ with control points $b_0(1, 2)$, $b_1(4, -1)$, and $b_2(8, 6)$ is $(x(t), y(t))$ where

$$
x(t) = (1-t)^2(1) + 2(1-t)t(4) + t^2(8) = 1 + 6t + t^2, \text{ and}
$$

$$
y(t) = (1-t)^2(2) + 2(1-t)t(-1) + t^2(6) = 2 - 6t + 10t^2.
$$

The point $B(0.5)$ is obtained by substituting $t = 0.5$ into the equations to give $x(0.5) = 4.25$ and $y(0.5) = 1.5$, that is, $B(0.5) = (4.25, 1.5)$. Alternatively, the coordinates of the point $B(0.5)$ can be evaluated using the vector form of the curve

$$
B(t) = (1-0.5)^2(1,2) + 2(1-0.5)(0.5)(4,-1) + (0.5)^2
= 0.25(1,2) + 0.5(4,-1) + 0.25(8,6) = (4.25, 1.5).
$$

Quadratic Bézier curve with control points $b_0(1, 2)$, $b_1(4, -1)$,
Bézier curves

Suppose four control points $b_0$, $b_1$, $b_2$, and $b_3$ are specified, then the cubic Bézier curve is defined to be

$$B(t) = (1 - t)^3b_0 + 3(1 - t)^2tb_1 + 3(1 - t)t^2b_2 + t^3b_3, \quad t \in [0, 1].$$

As in the quadratic case, the polygon obtained by joining the control points in the specified order is called the control polygon.
Bézier curves

Example

The parametric form of the quadratic Bézier curve $B(t)$ with control points $b_0(1,2)$, $b_1(4,-1)$, and $b_2(8,6)$ is $(x(t), y(t))$ where

$$x(t) = (1-t)^2(1) + 2(1-t)t(4) + t^2(8) = 1 + 6t + t^2,$$
$$y(t) = (1-t)^2(2) + 2(1-t)t(-1) + t^2(6) = 2 - 6t + 10t^2.$$

The point $B(0.5)$ is obtained by substituting $t = 0.5$ into the equations to give $x(0.5) = 4.25$ and $y(0.5) = 1.5$, that is, $B(0.5) = (4.25, 1.5)$. Alternatively, the coordinates of the point $B(0.5)$ can be evaluated using the vector form of the curve

$$B(t) = (1-0.5)^2(1,2) + 2(1-0.5)(0.5)(4,-1) + (0.5)^2(8,6)$$
$$= 0.25(1,2) + 0.5(4,-1) + 0.25(8,6) = (4.25, 1.5).$$
In some of the literature the nomenclature generally used is

\[ \mathbf{B} \rightarrow \mathbf{p}, \quad t \rightarrow u, \text{ and } \mathbf{b}_i \rightarrow \mathbf{p}_i \]

Bézier chose Bernstein polynomials as the basis functions for the curves.

\[ p(u) = \sum_{i=0}^{n} p_i B_{i,n}(u) \quad u \in [0, 1] \]

Based on these basis functions, the equation for the Bézier curve is given by:

\[ p(u) = (1-u)^3 \mathbf{p}_0 + 3 u (1-u)^2 \mathbf{p}_1 + 3 u^2 (1-u) \mathbf{p}_2 + u^3 \mathbf{p}_3 \]
Bézier curves

- This can be written in matrix form as

\[
p(u) = \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix}
\]

- \( p(u) = \{U\} [M_B] [P] \)
Bézier curves

Two cubic Bézier curves joined at P3

Rao, CAD/CAM Principles and Applications, 2010, TMH
Bézier curves

\[ p(u) = \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix} \]
Bézier curve properties

- The Bézier curve passes through the first and last control points while it maintains proximity to the intermediate control points.

- As such the entire Bézier curve lies in the interior of the convex hull of the control points.

- If a control point is moved the entire curve moves.

- Being polynomial functions, Bézier curves are easily computed, and infinitely differentiable.

- If the control points of the Bézier curve are transformed, the curve moves to the corresponding new coordinate frame without changing its shape.
Example: A cubic Bezier curve is described by the four control points: (0,0), (2,1), (5,2), (6,1). Find the tangent to the curve at t = 0.5.

\[ P(t) = [t^3 \quad t^2 \quad t \quad 1] \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} V_0 \\ V_1 \\ V_2 \\ V_3 \end{pmatrix} \]

where, \( V_0 = (0,0) \)
\( V_1 = (2,1) \)
\( V_2 = (5,2) \)
\( V_3 = (6,1) \)

The tangent is given by the derivative of the general equation above,

\[ P'(t) = [3t^2 \quad 2t \quad 1 \quad 0] \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} V_0 \\ V_1 \\ V_2 \\ V_3 \end{pmatrix} \]
At $t = 0.5$, we get,

$$P'(t) = \begin{bmatrix} 3(.5)^2 & 2(.5) & 1 & 0 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} V_0 \\ V_1 \\ V_2 \\ V_3 \end{bmatrix}$$

$$= \begin{bmatrix} 6.75 & 1.5 & 0 & 1 \end{bmatrix}$$
Hermite cubic spline

- Hermite cubic splines are the more general form of curves that can be defined through a set of vertices (points).

- A spline is a piecewise parametric representation of the geometry of a curve with a specified level of parametric continuity.

- Each segment of a Hermite cubic spline is approximated by a parametric cubic polynomial to maintain the $C^2$ continuity.
Hermite cubic spline

We want curves that fit together smoothly. To accomplish this, we would like to specify a curve by providing:

• The endpoints
• The 1st derivatives at the endpoints

The result is called a *Hermite Curve*. 
Hermite cubic spline

Hermite Cubic Spline Curve
The parametric equation of a Hermite cubic spline is given by

\[ p(u) = \sum_{i=0}^{3} C_i u^i \quad u \in [0, 1] \]

In an expanded form it can be written as

\[ p(u) = C_0 + C_1 u + C_2 u^2 + C_3 u^3 \]

Where \( u \) is a parameter, and \( C_i \) are the polynomial coefficients.
Hermite cubic spline

- In matrix form

\[ p(u) = \begin{bmatrix} u^3 & u^2 & u^1 & 1 \end{bmatrix} \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ p_0' \\ p_1' \end{bmatrix} \]

- \( p(u) = \{U\} [M] [P] \)
Hermite cubic spline

\[ p(u) = \begin{bmatrix} 2u^3 - 3u^2 + 1 \\ -2u^3 + 3u^2 \\ u^3 - 2u^2 + u \\ u^3 - u^2 \end{bmatrix}^T \begin{bmatrix} p_1 \\ p_2 \\ \nabla p_1 \\ \nabla p_2 \end{bmatrix} \]

4 Basis Functions

Hermite Blending Functions
Examples

Example 5: A parametric cubic curve passes through the points (0,0), (2,4), (4,3), (5, -2) which are parametrized at \( t = 0, \frac{1}{4}, \frac{3}{4}, \) and 1, respectively. Determine the geometric coefficient matrix and the slope of the curve when \( t = 0.5 \).

Solution: The points on the curve are

\[
\begin{align*}
(0,0) & \text{ at } t = 0 \\
(2,4) & \text{ at } t = \frac{1}{4} \\
(4,3) & \text{ at } t = \frac{3}{4} \\
(5,-2) & \text{ at } t = 1
\end{align*}
\]

Substituting in equation (4.15), we get,

\[
\begin{pmatrix}
0 & 0 \\
2 & 4 \\
4 & 3 \\
5 & -2
\end{pmatrix}
= \begin{pmatrix}
0 & 0 & 0 & 1 \\
0.0156 & 0.0625 & 0.25 & 1 \\
0.4218 & 0.5625 & 0.75 & 1 \\
1 & 1 & 1 & 1
\end{pmatrix}
\begin{pmatrix}
2 & -2 & 1 & 1 \\
-3 & 3 & -2 & -1 \\
0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0
\end{pmatrix}
\begin{pmatrix}
P(0) \\
P(1) \\
P'(0) \\
P'(1)
\end{pmatrix}
\]
Solving, we get,

\[
\begin{pmatrix}
P(0) \\
P(1) \\
P'(0) \\
P'(1)
\end{pmatrix} = \begin{pmatrix}
0 & 0 \\
5 & -2 \\
10.33 & 22 \\
4.99 & -26
\end{pmatrix}
\]

The slope at \( t = 0.5 \) is found by taking the first derivative of the equation (4.15), as follows,

\[
P'(t) = [3t^3 \ 2t \ 1 \ 0] \begin{pmatrix}
2 & -2 & 1 & 1 \\
-3 & 3 & -2 & -1 \\
0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0
\end{pmatrix} \begin{pmatrix}
0 & 0 \\
5 & -2 \\
10.33 & 22 \\
4.99 & -26
\end{pmatrix}
\]

Therefore,

\[
P'(0.5) = [3.67 \ -2.0], \text{ or}
\]

Slope = \( \Delta x/\Delta y = -2.0/3.67 = -0.545 \)
B-splines

- In the case of Bezier curve, it is a single curve controlled by all the control points.
  - With an increase in the number of control points, the order of the polynomial representing the curve increases.

- B-spline generates a single piecewise parametric polynomial curve through any number of control points with the degree of the polynomial selected by the designer.
B-splines

- B-spline curves have the flexibility of choosing the degree of the curve irrespective of the number of control points.

- With four control points, it is possible to get a cubic Bézier curve, while with B-spline curve one can get a linear, quadratic or cubic curve.

- B-spline also uses the basis (blending) functions and the equation is of the form

\[ 0 \leq u \leq u_{\text{max}} \]

\[ p(u) = \sum_{i=0}^{n} p_i \ N_{i,k}(u) \]
B-splines

- Where $N_{i,k}(u)$ are the basis functions for B-splines.

$$N_{i,1}(u) = \begin{cases} 
1 & \text{if } u_i \leq u \leq u_{i+1} \\
0 & \text{otherwise}
\end{cases}$$

where $k$ controls the degree $(k-1)$ of the resulting polynomial in $u$ and also the continuity of the curve. The $u_i$ are the knot values, which relate the parametric variable $u$ to the $p_i$ control points.

$$N_{i,k}(u) = \frac{(u - u_i) N_{i,k-1}(u)}{u_{i+k-1} - u_i} + \frac{(u_{i+k} - u) N_{i+1,k-1}(u)}{u_{i+k} - u_{i+1}}$$
The plotting of B-spline curve is done by varying the parameter $u$ over the range of knot values $(u_{k-1}, u_{n+1})$.

The knot vector adds flexibility to the curve and provides better control of its shape.

Partition of Unity: For any knot span, $[u_i, u_{i+1}]$,

Positivity:

\[ \sum_{i=0}^{n} N_{i,k}(u) = 1 \]

$N_{i,k}(u) \geq 0$ for all $i$, $k$ and $u$. 
B-spline properties

- Local Support Property:

\[ N_{i, k}(u) = 0 \text{ if } u \notin [u_i, u_{i+k+1}] \]

This property can be deduced from the observation that \( N_{i, k}(u) \) is a linear combination of \( N_{i, k-1}(u) \) and \( N_{i+1, k-1}(u) \).

- Continuity:

\( N_{i, k}(u) \) is \((k-2)\) times continuously differentiable, being a polynomial.
B-spline properties

- The curve follows the shape of the control points and lie in the convex hull of the control points.

- The entire B-spline curve can be affinely transformed by transforming the control points and redrawing the curve from the transformed points.

- B-splines exhibit local control, i.e., when a control point is moved only that segment is influenced.
B-spline properties for CAD

- Local control can be achieved by changing the position of the control points.
- B-spline curve tightens by increasing its degree. As the degree of the B-spline is lowered, it comes closer to the control polygon.
- If $k = n+1$, then the resulting B-spline curve is a Bézier curve.
- The B-spline curve is contained in the convex hull of its control points.
- Affine transformations of the coordinate system do not change the shape of the B-spline curve.
- Increasing the degree of the curve makes it more difficult to control, and hence a cubic B-spline is sufficient for a majority of applications.
Rational Curves

- A rational curve utilises the algebraic ratio of two polynomials.
- They are important in CAD because of their invariance when geometric transformations are applied.
Rational Curves

- A rational curve defined by \((n+1)\) points is given by

\[
p(u) = \sum_{i=0}^{n} p_i R_{i,k}(u) \quad 0 \leq u \leq u_{\text{max}}
\]

- Where \(R_{i,k}(u)\) is the rational B-spline basis function and is given by

\[
R_{i,k}(u) = \frac{h_i N_{i,k}(u)}{\sum_{i=0}^{n} h_i N_{i,k}(u)}
\]
Non-uniform rational basis spline (NURBS) is a mathematical model commonly used in computer graphics for generating and representing curves and surfaces.

- It offers great flexibility and precision for handling both analytic (surfaces defined by common mathematical formulae) and modeled shapes.

- NURBS is flexible for designing a large variety of shapes by manipulating the control points and weights.

- Weights in the NURBS data structure determine the amount of surface deflection toward or away from its control point.
Evaluation of NURBS is reasonably fast and numerically stable.
NURBS

- Uniform cubic B-splines are the curves with the parametric intervals defined at equal lengths.

- The most common scheme used in all the CAD system is the non-uniform rational B-spline (commonly known as NURB), allowing a non-uniform knot vector.

- It includes both the Bézier and B-spline curves.
NURBS

- Rational form of the B-splines can be written as

\[ p(u) = \frac{\sum_{i=0}^{n} w_i \ p_i \ N_{i,k}(u)}{\sum_{i=0}^{n} w_i \ N_{i,k}(u)} \]

- where \( w_i \) is the weighing factor for each of the vertex.
NURBS

- They have all of B-spline surface abilities. In addition they overcome the limitation of B-spline surfaces by associating each control point with a weight.

- Uniform representation for a large variety of curves and surfaces. This helps with the storage of geometric data.

- NURBS are invariant during geometric transformations as well as projections.
NURBS

- NURBS is flexible for designing a large variety of shapes by manipulating the control points and weights.

- Weights in the NURBS data structure determine the amount of surface deflection toward or away from its control point.

- It makes it possible to create curves that are true conic sections.

- Surfaces based on conics, arcs or spheres can be precisely represented by a NURBS surface.

- Evaluation of NURBS is reasonably fast and numerically stable.
NURBS

- Number facilities available in NURBS such as knot insertion/ refinement/ removal, degree elevation, splitting, etc. makes them ideal to be used throughout the design process.

- NURBS surfaces can be incorporated into an existing solid model by "stitching" the NURBS surface to the solid model.

- Reverse engineering is heavily dependent on NURBS surfaces to capture digitized points into surfaces.
Problems with NURBS

- Analytical curves and surfaces require additional storage.

- NURBS parameterization can often be affected by improper application of the weights, which may lead to subsequent problems in surface constructions.

- Not all geometric interrogation techniques work well with NURBS.
Thank You