Computer Graphics - Week 9

Questions about Last Week?

Bengt-Olaf Schneider
IBM T.J. Watson Research Center
Overview of Week 9

- The second half of the course will be spent on studying a variety of advanced topics in computer graphics.

- Freeform curves and surfaces
  - Hermite curves and surfaces
  - Bézier curves
  - B-splines curves
Freeform Curves and Surfaces

- Polygons and polyhedra usually provide only crude approximations of real objects
  - Real objects have smooth surfaces not faceted surfaces
  - Such surfaces are better described by higher-order functions

- Freeform curves and surfaces
  - Represent non-linear functions
  - More compact representation than polygons
  - Better control over shape, tangent direction and slope

- We will look at various ways to represent freeform curves and surfaces

Explicit Functions

- Explicit functions
  - Curves: \( z = f(x) \)
  - Surfaces: \( z = f(x, y) \)

- Explicit functions ...
  - do not allow for multiple values for a given argument, e.g. circles or folding surfaces
  - cannot describe vertical tangents, as infinite slopes are hard to represent
  - are not rotation-invariant, i.e. the rotated version of a curve may require different description. E.g. rotated half-circle.
Implicit Functions

- Implicit functions
  - Use a function that states which points are on and off the curve
    \[ f(x, y, z) = 0 \]
  - We have used implicit functions to define lines and planes

- Implicit functions ...
  - do not have some of the problems of explicit functions
  - are hard to find for many shapes and curves
  - provide no control over tangents at connection points when joining several implicit functions

Parametric Functions

- Curves and Surfaces are defined by independent functions in a common parameter
  \[ x = x(t) ; y = y(t) ; z = z(t) \]

- Parametric functions ...
  - avoid the problems of implicit and explicit functions
  - can provide control over tangent direction at the end points
Cubic Polynomials

The coordinates can be defined by arbitrary functions

Here we will discuss polynomials of order $n$

- In many cases we will only look at $n=3$ (cubics)
- Cubic polynomials allow definition of end points and tangent direction at both endpoints

$$Q(t) = \begin{bmatrix} x(t) = a_xt^3 + b_xt^2 + c_xt + d_x \\ y(t) = a_yt^3 + b_yt^2 + c_yt + d_y \\ z(t) = a_zt^3 + b_zt^2 + c_zt + d_z \end{bmatrix}, \quad \text{for } 0 \leq t \leq 1$$

$$Q(t) = T \cdot C = \begin{bmatrix} a_x & a_y & a_z \\ b_x & b_y & b_z \\ c_x & c_y & c_z \\ d_x & d_y & d_z \end{bmatrix} \cdot \begin{bmatrix} t^3 \\ t^2 \\ t \end{bmatrix}$$

Tangent Vectors

Tangent vectors are the derivatives at a given point

$$\frac{\partial}{\partial t} Q(t) = Q'(t) = \frac{\partial}{\partial t} T \cdot C = \begin{bmatrix} 3t^2 \\ 2t \\ 1 \end{bmatrix} \cdot \begin{bmatrix} a_x & a_y & a_z \\ b_x & b_y & b_z \\ c_x & c_y & c_z \\ d_x & d_y & d_z \end{bmatrix} = \begin{bmatrix} 3a_xt^2 + 2b_xt + c_x \\ 3a_yt^2 + 2b_yt + c_y \\ 3a_zt^2 + 2b_zt + c_z \end{bmatrix}^T$$

- $Q'(t)$ is the velocity of $t$ as it moves along the curve
- $Q''(t)$ is the acceleration of $t$ along the curve
Joining Parametric Curves (1)

- Curves can be defined piecewise using several parametric curve segments
  - At the joints, we often desire continuity

- **Geometric continuity**
  - $G^0$: Curve segments meet at the joint
  - $G^1$: Tangents at the joint have same direction
  - $G^n$: N-th derivatives have same direction

\[
\frac{\partial^n}{\partial t^n} Q(1) = k \cdot \frac{\partial^n}{\partial t^n} P(0)
\]

- **Parametric continuity**
  - $C^n$: N-th derivatives have same magnitude

\[
\frac{\partial^n}{\partial t^n} Q(1) = \frac{\partial^n}{\partial t^n} P(0)
\]

Joining Parametric Curves (2)

- **Parametric continuity implies geometric continuity**
  - The reverse is not generally true
  - Visually geometric and parametric continuity differ only slightly
Computing Cubic Curves

- Each cubic polynomial is described by four parameters.

- Therefore, we need four conditions to specify those parameters.

- There are different sets of constraints to provide those four conditions:
  - Hermite Curves: Start and end point, start and end tangent vector
  - Bezier Curves: Start and end point, 2 intermediate points
  - B-splines: 4 arbitrary points

- We will look at all of these now ...

Formulating the Constraints (1)

- We split the coefficient matrix to express the four required conditions:

  \[ Q(t) = \begin{pmatrix} t^3 & t^2 & t & 1 \end{pmatrix} \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{pmatrix} \begin{pmatrix} G_1 \\ G_2 \\ G_3 \\ G_4 \end{pmatrix} = T \cdot M \cdot G \]

- The coefficients of G are vectors with x, y, z components, i.e.

  \[ G = \begin{pmatrix} G_1 \\ G_2 \\ G_3 \\ G_4 \end{pmatrix} = \begin{pmatrix} g_{1X} & g_{1Y} & g_{1Z} \\ g_{2X} & g_{2Y} & g_{2Z} \\ g_{3X} & g_{3Y} & g_{3Z} \\ g_{4X} & g_{4Y} & g_{4Z} \end{pmatrix} \]
Formulating the Constraints (2)

For instance, for $x(t)$ we get:

$$x(t) = \left( m_{11}t^3 + m_{21}t^2 + m_{31}t + m_{41} \right) g_{1X} +$$
$$\left( m_{12}t^3 + m_{22}t^2 + m_{32}t + m_{42} \right) g_{2X} +$$
$$\left( m_{13}t^3 + m_{23}t^2 + m_{33}t + m_{43} \right) g_{3X} +$$
$$\left( m_{14}t^3 + m_{24}t^2 + m_{34}t + m_{44} \right) g_{4X}$$

Each of the terms in parentheses are cubic polynomial

The constraints $g_{ix}$ compute a weighted sum of those four cubics.
Therefore, the functions $G_i$ are also known as blending functions.

Formulating the Constraints (3)

Now, all that’s left is to determine the blending functions $G$ and the basis matrix $M$.

For that we substitute points or tangent vectors for the blending functions $G_i$ and then determine the coefficients of $M$.

The actual choice of those points and vectors determines the type of curve.
Hermite Curves (1)

Hermite Curves are defined by start and end point of the curve and the tangent vectors at those points.

- \( G_H = \begin{pmatrix} P_1 & P_4 & R_1 & R_4 \end{pmatrix}^T \)
  - (Indices are numbered 1 and 4 to be consistent with other curves that use 4 points to define a curve.)

Recall, the expressions for \( x(t) \) and \( x'(t) \):

\[
x(t) = T \cdot M_H \cdot G_{HX} = \begin{pmatrix} t^3 & t^2 & t & 1 \end{pmatrix} \cdot M_H \cdot G_{HX}
\]

\[
x'(t) = T' \cdot M_H \cdot G_{HX} = \begin{pmatrix} 3t^2 & 2t & 1 & 0 \end{pmatrix} \cdot M_H \cdot G_{HX}
\]

Hermite Curves (2)

Substituting the start point (t=0) and end point (t=1) into this equation gives:

\[
x(0) = P_{1x} = \begin{pmatrix} 0 & 0 & 0 & 1 \end{pmatrix} \cdot M_H \cdot G_{HX}
\]

\[
x(1) = P_{4x} = \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix} \cdot M_H \cdot G_{HX}
\]

Substituting the tangent vectors at the start and end points gives:

\[
x'(0) = R_{1x} = \begin{pmatrix} 0 & 0 & 1 & 0 \end{pmatrix} \cdot M_H \cdot G_{HX}
\]

\[
x'(1) = R_{4x} = \begin{pmatrix} 3 & 2 & 1 & 0 \end{pmatrix} \cdot M_H \cdot G_{HX}
\]
Hermite Curves (3)

Pulling it all together:

\[ \begin{pmatrix} P_{1x} \\ P_{4x} \\ R_{1x} \\ R_{4x} \end{pmatrix} = G_{HX} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{pmatrix} \cdot M_H \cdot G_{HX} \]

Therefore:

\[ M_H = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{pmatrix}^{-1} = \begin{pmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \]

Hermite Curves (4)

\[ M_H \] is the basis matrix for Hermite curves.

- It describes the blending functions for the geometric constraints \( G \).
- We find the blending function \( B_i \) as:

\[
Q(t) = T \cdot M_H \cdot G_H = B_H \cdot G_H
\]

\[
= (2t^3 - 3t^2 + 1) \cdot P_1 + (-2t^3 + 3t^2) \cdot P_4 +
(t^3 - 2t^2 + t) \cdot R_1 + (t^3 - t^2) \cdot R_4
\]

- All blending functions have support over the entire interval [0,1].
- A change in any of the constraints affects the entire curve.
- The blending functions interpolate the starting and ending points.
Hermite Curves (5)

The blending functions $B_i^x$ have the following shape.

Example of Hermite Curves

Curves shown for $k = 4, 6, 8, 10$.
Joining Hermite Curves

- Continuity at the joints between Hermite curves can be easily ensured by specifying tangent vectors with the same direction.

- For geometric continuity:

\[
\begin{pmatrix}
P_1 \\
P_4 \\
R_1 \\
R_4
\end{pmatrix}
\quad \text{and} \quad
\begin{pmatrix}
P_4 \\
P_7 \\
k \cdot R_4 \\
R_7
\end{pmatrix}
\quad \text{with } k > 0
\]

- For parametric continuity \( k = 1 \)

Bicubic Patches (1)

- Moving a curve through space can create a surface.
- If each of the control points itself moves along a cubic curve as the curve itself is moved, a bicubic patch is generated.
Bicubic Patches (2)

To describe such patches we start from the expression for parametric, cubic curves ...

\[ Q(t) = (t^3 \ t^2 \ t \ 1) \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{pmatrix} \begin{pmatrix} G_1 \\ G_2 \\ G_3 \\ G_4 \end{pmatrix} = T \cdot M \cdot G \]

... and rename the parameter \( t \) to \( s \):

\[ Q(s) = (s^3 \ s^2 \ s \ 1) \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{pmatrix} \begin{pmatrix} G_1 \\ G_2 \\ G_3 \\ G_4 \end{pmatrix} = S \cdot M \cdot G \]

Bicubic Patches (3)

Each of the geometric constraints \( G_i \) is itself described by a cubic polynomial

\[ Q(s,t) = S \cdot M \cdot G(t) = S \cdot M \cdot \begin{pmatrix} G_1(t) \\ G_2(t) \\ G_3(t) \\ G_4(t) \end{pmatrix} \]

\[ G_i(t) = T \cdot M \cdot g_i = T \cdot M \cdot \begin{pmatrix} g_{i1} \\ g_{i2} \\ g_{i3} \\ g_{i4} \end{pmatrix} \]
Bicubic Patches (4)

- Transposing the expressions for the polynomials $g_i$:

$$G_i(t) = T \cdot M \cdot g_i = g_i^T \cdot M^T \cdot T^T$$

$$= \begin{pmatrix} g_{i1} & g_{i2} & g_{i3} & g_{i4} \end{pmatrix} \cdot M^T \cdot T^T$$

- Combining this with expression for the patch $Q(s,t)$:

$$Q(s,t) = S \cdot M \cdot G(t) = S \cdot M \cdot g \cdot M^T \cdot T^T$$

$$= S \cdot M \cdot \begin{pmatrix} g_{11} & g_{12} & g_{13} & g_{14} \\ g_{21} & g_{22} & g_{23} & g_{24} \\ g_{31} & g_{32} & g_{33} & g_{34} \\ g_{41} & g_{42} & g_{43} & g_{44} \end{pmatrix} \cdot M^T \cdot T^T$$

Bicubic Patches (5)

- This is the generic description of bicubic patches

- By choosing a concrete geometry matrix $g$, different patch types can be realized
  
  - We will first explore Hermite patches.
  - The mechanism applies similarly to Bézier and B-spline patches
Hermite Patches (1)

Hermite patches are formed by using the Hermite basis matrix $M_H$ and a cubic Hermite geometry vector $g_H$:

$$Q_H(s,t) = S \cdot M_H \cdot g_H \cdot M_H^T \cdot T^T$$

$$g_H = \begin{pmatrix}
    P(0,0) & P(0,1) & \frac{\partial}{\partial t} P(0,0) & \frac{\partial}{\partial t} P(0,1) \\
    P(1,0) & P(1,1) & \frac{\partial}{\partial t} P(1,0) & \frac{\partial}{\partial t} P(1,1) \\
    \frac{\partial}{\partial s} P(0,0) & \frac{\partial}{\partial s} P(0,1) & \frac{\partial^2}{\partial s \partial t} P(0,0) & \frac{\partial^2}{\partial s \partial t} P(0,1) \\
    \frac{\partial}{\partial s} P(1,0) & \frac{\partial}{\partial s} P(1,1) & \frac{\partial^2}{\partial s \partial t} P(1,0) & \frac{\partial^2}{\partial s \partial t} P(1,1)
\end{pmatrix}$$

Hermite Patches (2)

Each row or column of the geometry matrix $g_H$ describes a cubic polynomial.

- These polynomials can be interpreted as how the control points change across the patch, e.g. boundary curve for $s=0$
Hermite Patches (3)

- Control points at the four patch corners

\[ g_H = \begin{pmatrix} P(0,0) & P(0,1) \\ P(1,0) & P(1,1) \\ \frac{\partial}{\partial s} P(0,0) & \frac{\partial}{\partial s} P(0,1) \\ \frac{\partial}{\partial t} P(0,0) & \frac{\partial}{\partial t} P(0,1) \\ \frac{\partial^2}{\partial s \partial t} P(0,0) & \frac{\partial^2}{\partial s \partial t} P(0,1) \end{pmatrix} \]

Hermite Patches (4)

- Starting tangent vectors in both directions at the four patch corners

\[ g_H = \begin{pmatrix} P(0,0) & P(0,1) \\ P(1,0) & P(1,1) \\ \frac{\partial}{\partial s} P(0,0) & \frac{\partial}{\partial s} P(0,1) \\ \frac{\partial}{\partial t} P(0,0) & \frac{\partial}{\partial t} P(0,1) \\ \frac{\partial^2}{\partial s \partial t} P(0,0) & \frac{\partial^2}{\partial s \partial t} P(0,1) \end{pmatrix} \]
Hermite Patches (5)

Starting tangent vectors of the tangent vectors at the four patch corners, a.k.a. twists

- The twists describe how much the corner is "twisted", i.e. how much the tangents are rotated around the corner.

\[
g_H = \begin{pmatrix}
  P(0,0) & P(0,1) & \frac{\partial}{\partial t} P(0,0) & \frac{\partial}{\partial t} P(0,1) \\
  P(1,0) & P(1,1) & \frac{\partial}{\partial t} P(1,0) & \frac{\partial}{\partial t} P(1,1) \\
  \frac{\partial}{\partial s} P(0,0) & \frac{\partial}{\partial s} P(0,1) & \frac{\partial^2}{\partial s^2} P(0,0) & \frac{\partial^2}{\partial s^2} P(0,1) \\
  \frac{\partial}{\partial s} P(1,0) & \frac{\partial}{\partial s} P(1,1) & \frac{\partial^2}{\partial s^2} P(1,0) & \frac{\partial^2}{\partial s^2} P(1,1)
\end{pmatrix}
\]

Joining Hermite Patches (1)

- Similar to joining Hermite curves, patches can be joined with C^1 or G^1 continuity.

- For smooth joint, the following conditions must be met:
  - Assuming boundary along constant t.
  - Control points along the common boundary must be the same.
  - Tangent vectors across the common boundary must match.
Joining Hermite Patches (2)

\[
\begin{pmatrix}
P_1(1,0) & P_1(1,1) & \frac{\partial}{\partial t} P_1(1,0) & \frac{\partial}{\partial t} P_1(1,1) \\
\frac{\partial}{\partial s} P_1(1,0) & \frac{\partial}{\partial s} P_1(1,1) & \frac{\partial^2}{\partial s \partial t} P_1(1,0) & \frac{\partial^2}{\partial s \partial t} P_1(1,1)
\end{pmatrix}
\begin{pmatrix}
P_2(1,0) & P_2(1,1) & \frac{\partial}{\partial t} P_2(1,0) & \frac{\partial}{\partial t} P_2(1,1) \\
\frac{\partial}{\partial s} P_2(1,0) & \frac{\partial}{\partial s} P_2(1,1) & \frac{\partial^2}{\partial s \partial t} P_2(1,0) & \frac{\partial^2}{\partial s \partial t} P_2(1,1)
\end{pmatrix}
\]

Bézier Curves (1)

- Bézier specify the start and end point directly
- The tangent vectors are specified indirectly using two additional points

\[
R_1 = Q'(0) = 3(P_2 - P_1)
\]
\[
R_4 = Q'(1) = 3(P_4 - P_3)
\]

- Bézier curves interpolate first + last control point
- The other two control points are approximated
Bézier Curves (2)

We will first derive the basis matrix $M_B$ using the basis matrix $M_H$ for Hermite curves.

- The geometry vector for Bézier curves contains four points:
  \[ G_B = \begin{pmatrix} P_1 & P_2 & P_3 & P_4 \end{pmatrix}^T \]

- Using the definition of tangent vectors for Bézier curves we obtain the geometry vector for Hermite curves:
  \[ G_H = \begin{pmatrix} P_1 \\ P_2 \\ R_1 \\ R_4 \end{pmatrix} = \begin{pmatrix} P_1 \\ P_4 \\ 3(P_2 - P_1) \\ 3(P_4 - P_3) \end{pmatrix} = M_H \cdot G_B \]

Bézier Curves (3)

By substituting this expression into the formula for Hermite curves we obtain:

\[ Q(t) = T \cdot M_H \cdot G_H = T \cdot M_H \cdot M_{HB} \cdot G_B = T \cdot M_B \cdot G_B \]

\[ M_B = M_H \cdot M_{HB} = \begin{pmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -3 & 3 & 0 & 0 \\ 0 & 0 & -3 & 3 \end{pmatrix} = \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \]

This gives the following expression for $Q(t)$:

\[ Q(t) = (1-t)^3 P_1 + 3t(1-t)^2 P_2 + 3t^2(1-t)P_3 + t^3 P_4 \]
Bernstein Polynomials (1)

- The blending functions are known as the Bernstein polynomials.
- In general, the Bernstein polynomials are defined as:
  
  \[ B_{n,i}(t) = \binom{n}{i} t^i (1-t)^{n-i} \]

- Bézier curves with \( n \) control points are defined as:
  
  \[ Q(t) = \sum_{i=0}^{n} P_i \cdot B_{n,i}(t) \quad \text{for} \quad 0 \leq t \leq 1 \]

Bernstein Polynomials (2): Degree 2

![Graph showing Bernstein polynomials of degree 2](image_url)
Bernstein Polynomials (2): Degree 3

Bernstein Polynomials (2): Degree 4
Bernstein Polynomials (3)

- Positive over the range $t$ in $[0,1]$:
  - The Bézier curve is contained in the convex hull of the control points $P_i$.
  - All control points influence the overall shape of the curve, no local support.

- The sum of all Bernstein functions is always 1 in $[0,1]$
  - Hence, a Bézier curve is simply a weighted average of the $P_i$.

Bézier Curve: Properties (1)

- The degree of the polynomial defining the curve segment is one less than the number of control points.
  - The more control points, the higher the degree of the polynomial
  - The curve becomes "wigglier" as more control points are added

- The curve follows the shape of the control polygon
  - This makes Bézier curves intuitive to work with
  - With a little practice, one can predict the shape of the curve given the control polygon

- The first and last points on the curve interpolate the first and last point of the control polygon
  - Makes it easy to join several Bézier segments
Bézier Curve: Properties (2)

- The tangent vectors at the curve ends have the same direction as first + last segment of the control polygon
  - Provides intuitive control over the tangent direction at start and end points
  - Makes it easy to join several Bézier segments

- The curve is contained in the convex hull of the control polygon
  - Facilitates clipping of Bézier curves. Perform trivial accept/reject on the control polygon before actual clipping of the curve.
  - Makes the shape and extent of the curve predictable

Bézier Curve: Properties (3)

- The curve exhibits the variation diminishing property, i.e. curve does not oscillate about any straight line more often than the defining polygon
  - The curve follows the shape of the control polygon and does not oscillate more than indicated by the control polygon

- The curve is invariant under an affine transformation
  - The curve can be transformed by by transforming the control polygon
  - The curve is not invariant under perspective transformations!
Joining Bézier Curves

- Bézier curves can be easily joined with $C^1$ or $G^1$ continuity
  - The tangent vectors $(P_4-P_3)$ and $(P_5-P_4)$ must be collinear
  - For parametric continuity they must have the same length

Bézier Curves: Summary of Properties

- The degree of the polynomial defining the curve segment is one less than the number of control points.
- The curve follows the shape of the control polygon
- The first and last points on the curve interpolate the first and last point of the control polygon
- The tangent vectors at the curve ends have the same direction as first + last segment of the control polygon
- The curve is contained in the convex hull of the control polygon
- The curve exhibits the variation diminishing property, i.e. curve does not oscillate about any straight line more often than the defining polygon
- The curve is invariant under an affine transformation.
Bézier Curves: Demo

- DesignMentor from Michigan Technological University

- Bézier Curves
  - Influence of control points
  - Tangent vectors
  - Joining Bézier curves

Bézier Curves: Problems

- Bézier curves have limited utility mainly because of two reasons:
  - The number of control points determines the degree of the polynomial defining the curve
    - Adding points will automatically raise the degree of the polynomial
    - The only way to reduce the degree of the polynomial is to delete control points.
  - The Bernstein polynomials have global support
    - Changing one control point affects the entire curve
    - Difficult to introduce local changes to the curve

- Both problems are a result of the choice of the basis functions
- Another set of basis functions can eliminate these problems: B-Splines
Spline Curves

Splines were used to create curves in drafting
- Splines are long, metal strips that are bent and held into shape at certain points by "ducks"
- The "ducks" define the control points of the control polygon
- The metal strip implements the spline function
- The original splines interpolate the control points
- The shape of the curve depends on all control points

We will look at B-splines, a different class of splines, where control points have only local influence.
- The curve does not interpolate, but only approximate, the control points.

B-spline Curves: Concepts (1)

B-spline curves are controled by an arbitrary number of control points
- B-spline curves are defined in several segments, each of which is controlled by 4 control points (for cubic B-splines)
- B-splines are described by $m+1$ control points $P_0, P_1, P_2, \ldots P_m$
- Each curve consists of $m-2$ curve segments $Q_0, Q_1, \ldots Q_m$
- The parameter $t$ sequentially covers a range of 1 for every segment
- The parameter $t$ normally ranges from 0 to $m-2$
- The segments are connected with $C^2$ continuity (again: for cubic B-splines). This makes B-splines smoother than Hermite or Bézier curves.
B-spline Curves: Concepts (2)

B-spline Curves: Concepts (3)

- A segment is controlled by 4 control points
  - \( Q_5: \) \( P_2 \ldots P_5 \)
  - \( Q_i: \) \( P_{i-3} \ldots P_i \)

- Each segment covers parameter interval of 1
  - \( Q_5: \) \( t_5 \ldots t_6 \)
  - \( Q_i: \) \( t \ldots t_i+1 \)

- Each control point influences 4 segments
  - \( P_4: \) \( Q_4 \ldots Q_7 \)
  - \( P_n: \) \( Q_n \ldots Q_{n+3} \)
B-Spline Curves: Concepts (4)

- The joints between curve segments are called *knots*
- The *knot vector* specifies the parameter values at the knots
- Generally, the only condition for knots is that there values must increase monotonically, i.e. \( t_i \leq t_{i+1} \)

  - So far we have only considered uniform knot vectors, i.e. \( t_i - t_{i-1} = 1 \), e.g. \( (0, 1, 2, 3, 4, 5) \)
  - However, knot vectors can contain repeated knot values and knot values that are not spaced at equal intervals, e.g. \( (0, 0, 0, 0, 1, 2, 3, 3, 4, 4, 4, 4) \)

B-Splines Curves: Definition (1)

- Similar to Bézier curves, B-spline curves are defined by computing the weighted sum of the control points:

\[
Q(t) = \sum_{i=0}^{m} P_i \cdot B_{i,k}(t)
\]

  - The \( P_i \) are the control points
  - The functions \( B_{i,k} \) are the B-spline basis functions of order \( k \)
  - B-splines of order \( k \) are generating polynomials of degree \( (k-1) \)

  - The order of the B-splines can be selected as needed
  - The maximum order of a B-spline curve is equal to the number of control points, i.e. \( k \leq m \)
**B-Splines Curves: Definition (2)**

The B-spline basis functions are recursively defined as

\[
B_{i,1}(t) = \begin{cases} 
1 & t_i \leq t < t_{i+1} \\
0 & \text{otherwise}
\end{cases}
\]

\[
B_{i,k}(t) = \frac{t - t_i}{t_{i+k-1} - t_i} B_{i,k-1}(t) + \frac{t_{i+k} - t}{t_{i+k} - t_{i+1}} B_{i+1,k-1}(t)
\]

- By definition: \( 0 / 0 = 0 \)

**B-Spline Curves: Properties (1)**

Recursive computation of the basis functions results in a triangular dependence pattern

\[
\begin{array}{cccc}
B_{i,k} & B_{i+1,k-1} \\
B_{i,k-1} & B_{i+1,k-2} \\
B_{i,k-2} & B_{i+1,k-2} & B_{i+2,k-2} \\
\vdots & \ddots \\
B_{i,1} & B_{i+1,1} & B_{i+2,1} & \cdots & B_{i+k-1,1}
\end{array}
\]

- Basis functions of order \( k \) are non-zero for \( k \) segments
  - In each of the \( k \) recursion, the order is reduced by 1.
  - Each recursion level adds one segment because \( N_{i,k} = f(N_{i,k-1}, N_{i+1,k-1}) \)
  - Therefore, for \( k \) recursion levels, \( k-1 \) segments are added
B-Spline Curves: Properties (2)

- Basis functions of order \( k \) are non-zero for \( k \) segments
- Therefore, each control point has influence over only \( k \) curve segments

- The basis functions are non-negative
- Therefore, each segment is contained in the convex hull of its control points. Obviously, then the entire curve is also contained in the convex hull of the entire control polygon.
- This is a stronger property than for Bézier curves.

Periodic B-Spline Curves (1)

- Periodic B-splines have a uniform knot vector, i.e. knot values are spaced 1 apart, e.g. \((0, 1, 2, 3, 4, 5)\)

- Explicite computation of the basis functions for \( k=4 \), i.e. cubic B-splines
- Recall the dependency pattern for computing the basis functions:
**Periodic B-Spline Curves (2)**

- Expression for a B-spline curve of order 4:
  \[ Q(t - t_i) = \sum_{i=0}^{m} P_i \cdot B_{i,t}(t) \]

- Expanding this expression gives:
  \[ Q(t - t_i) = T \cdot M_{Bs} \cdot G_{Bs} = B_{Bs} \cdot G_{Bs} \]

\[ = (t^3 \quad t^2 \quad t \quad 1) \cdot \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 4 & 0 \end{pmatrix} \cdot \begin{pmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{pmatrix} \]

\[ = \frac{(1-t)^3}{6} \cdot P_0 + \frac{3t^3 - 6t^2 + 4}{6} \cdot P_1 + \frac{3t^3 + 3t^2 + 3t + 1}{6} \cdot P_2 + \frac{t^3}{6} \cdot P_3 \]

**Periodic B-Spline Curves (3)**

- The matrix \( M_{Bs} \) is the B-spline basis matrix

- The vector \( B_{Bs} \) contains the B-spline basis functions
Periodic B-Spline Curves (3)

- B-spline basis functions for \( k=4 \)

Open B-Spline Curves (1)

- Periodic B-spline curves do not interpolate the first and last control point

- Multiple control points, e.g. \( P_i = P_{i+1} = \ldots = P_{i+n} \), pull the curve closer to the control polygon.
  - The curve interpolates the control point for \( n = k-1 \).
  - In those cases the curve reduces to a line segment.

- Non-uniform B-splines offer another degree of flexibility in controlling the shape of the curve
  - For non-uniform B-splines knots are not equally spaced apart
  - Multiple knots, e.g. \( t_i = t_{i+1} = \ldots = t_{i+n} \), provide control over the continuity of the curve and how close the curve comes to the control polygon
Open B-Spline Curves (2)

- Open B-splines have $k$-fold knot at both ends of the knot vector,
  - For instance (0, 0, 0, 1, 2, 3, 4, 4, 4, 4) for a cubic B-spline
  - The curve interpolates the first and last control point
  - The basis functions are not periodic any more, i.e. the control points are weighted differently and have different support.

B-Spline Curves: Demo

- Local control over the curve shape
  - Uniform knot vector
    - Periodic B-spline
  - Non-uniform knot vectors
    - Called clamped knot vector in DesignMentor
    - Open B-spline
  - Multiple vertices
Things we have not talked about ...

- Rational Bézier and B-splines curves
  - Allows exact modeling of conic sections, e.g. circle or parabola
  - Even more flexibility for shape control
- Other basis functions
- Subdivision of curves and surfaces
  - Provides a tool to introduce local changes without complicating the entire shape
- Elevation of degree
  - Describe the same curve/surface with a higher-degree polynomial
  - Another means to provide more control over the shape
- Techniques to efficiently evaluate and draw curves and surfaces
- Read Foley et al. and more advanced text books

To find out more ...

- Study the text book
- David F. Rogers and J. Alan Adams
  *Mathematical Elements for Computer Graphics*
Summary

- Freeform Curves and Surfaces
  - Explicit, implicit and parametric shapes
  - Geometric and Parametric continuity
  - Hermite, Bézier and B-spline curves

Homework

- Study freeform curves and surfaces (chapter 11)
- Prepare Foley et al. Chapter 12, "Solid Modeling"
Next Week ...

- Solid Modeling
- Introduction to Ray Tracing