6.837 Computer Graphics
Bézier Curves and Splines

Fredo Durand
MIT CSAIL
Before We Begin

• Anything on your mind concerning Assignment 0?
• Any questions about the course?

• Assignment 1 (Curves & Surfaces) posted tomorrow
  – Due Wednesday September 29
• Linear Algebra review session
  – Wednesday Sep 22, 7:30pm
Today

• Curves in 2D
  – Useful in their own right
  – Provides basis for surface editing (Thursday)
Modeling 1D Curves in 2D

- **Polyline**
  - Sequence of vertices connected by straight line segments
  - Useful, but not for smooth curves
  - Very easy!
  - This is the representation that usually gets drawn in the end (smooth curves converted into these)

- **Smooth curves**
  - How do we specify them?
  - A little harder (but not too much)
Splines

- A type of smooth curve in the plane or in 3D
- Many uses
  - 2D Illustration (e.g. Adobe Illustrator)
  - Fonts
  - 3D Modeling
  - Animation: Trajectories
- In general: Interpolation and approximation
How Many Dimensions?
How Many Dimensions?

This curve lies on the 2D plane, but is itself 1D.

6.837 – Durand
How Many Dimensions?

This curve lies on the 2D plane, but is itself 1D.

You can just as well define 1D curves in 3D space.
Two Definitions of a Curve

• A continuous 1D point set on the plane or space
• A mapping from an interval $S$ onto the plane
  – That is, $P(t)$ is the point of the curve at parameter $t$

$$P : \mathbb{R} \ni S \mapsto \mathbb{R}^2, \quad P(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix}$$

• Big differences
  – It’s easy to generate points on the curve from the 2nd
  – The second definition can describe trajectories, the speed at which we move on the curve
General Principle of Splines

- User specifies **control points**
- We’ll interpolate the control points by a smooth curve
  - The curve is completely determined by the control points.
The ducks and spline are used to make tighter curves.
Two Points of View

- Approximation/interpolation
  - We have “data points”, how can we interpolate?
  - Important in lots of applications, both graphics and non-graphics

- User interface/modeling
  - What is an easy way to specify a smooth curve?
  - Our main perspective today.
Questions?
Splines

- Specified by a few control points
  - Good for UI
  - Good for storage

- Results in a smooth parametric curve $P(t)$
  - Just means that we specify $x(t)$ and $y(t)$
  - In practice: Low-order polynomials, chained together
  - Convenient for animation, where $t$ is time
  - Convenient for *tessellation* because we can discretize $t$ and approximate the curve with a polyline
Tessellation

- It's easy to rasterize mathematical line segments into pixels
  - OpenGL and the graphics hardware can do it for you
- But polynomials and other parametric functions are harder
To display $P(t)$, discretize it at discrete $t$s
It’s clear that adding more points will get us closer to the curve.
It’s clear that adding more points will get us closer to the curve.
Interpolation vs. Approximation

• Interpolation
  – Goes through all specified points
  – Sounds more logical

• Approximation
  – Does not go through all points
Interpolation vs. Approximation

• Interpolation
  – Goes through all specified points
  – Sounds more logical
  – But can be more unstable, “ringing”

• Approximation
  – Does not go through all points
  – Turns out to be convenient

• In practice, we’ll do something in between.

Tuesday, September 14, 2010
Questions?
Cubic Bézier Curve

- User specifies 4 control points $P_1 \ldots P_4$
- Curve goes through (interpolates) the ends $P_1, P_4$
- Approximates the two other ones
- Cubic polynomial
Cubic Bézier Curve

That is,

\[ x(t) = (1 - t)^3 x_1 + 3t(1-t)^2 x_2 + 3t^2(1-t) x_3 + t^3 x_4 \]

\[ y(t) = (1 - t)^3 y_1 + 3t(1-t)^2 y_2 + 3t^2(1-t) y_3 + t^3 y_4 \]
Cubic Bézier Curve

- \( P(t) = (1-t)^3 P_1 + 3t(1-t)^2 P_2 + 3t^2(1-t) P_3 + t^3 P_4 \)

Verify what happens for \( t=0 \) and \( t=1 \)
Cubic Bézier Curve

- 4 control points
- Curve passes through first & last control point
- Curve is tangent at $P_1$ to $(P_1 - P_2)$ and at $P_4$ to $(P_4 - P_3)$

A Bézier curve is bounded by the convex hull of its control points.
Cubic Bézier Curve

- 4 control points
- Curve passes through first & last control point
- Curve is tangent at $P_1$ to $(P_1 - P_2)$ and at $P_4$ to $(P_4 - P_3)$

A Bézier curve is bounded by the convex hull of its control points.
Questions?
What’s with the Formula?

• Explanation 1:
  – Magic!

• Explanation 2:
  – These are smart weights that describe the influence of each control point

• Explanation 3:
  – It’s a linear combination of basis polynomials.
Weights

- $P(t)$ is a weighted combination of the 4 control points with weights:
  - $B_1(t) = (1-t)^3$
  - $B_2(t) = 3t(1-t)^2$
  - $B_3(t) = 3t^2(1-t)$
  - $B_4(t) = t^3$

- First, $P_1$ is the most influential point, then $P_2$, $P_3$, and $P_4$
Weights

- $P_2$ and $P_3$ never have full influence
  - Not interpolated!

\[
P(t) = (1-t)^3 P_1 + 3t(1-t)^2 P_2 + 3t^2(1-t) P_3 + t^3 P_4
\]
Questions
What’s with the Formula?

• Explanation 1:
  – Magic!

• Explanation 2:
  – These are smart weights that describe the influence of each control point

• Explanation 3:
  – It’s a linear combination of basis polynomials.
  – The opposite perspective: control points are the weights of polynomials!!
  – Let’s study this in 1D using curves $y=f(t)$
Why study splines as vector space

- Understand relationships between types of splines
  - Conversion
- Express what happens when a spline curve is transformed by an affine transform (rotation, translation, etc.)
- In general, linearity is a useful property
  - Fall back to known case with lots of good properties
- Cool simple example of non-trivial vector space
- Important to understand for advanced methods such as finite elements
Usual Vector Spaces

• In 3D, each vector has three components $x, y, z$
• But geometrically, each vector is actually the sum

$$\mathbf{v} = x \mathbf{i} + y \mathbf{j} + z \mathbf{k}$$

• $\mathbf{i}, \mathbf{j}, \mathbf{k}$ are basis vectors

• Vector addition: just add components
• Scalar multiplication: just multiply components
Polynomials as a Vector Space

- Polynomials: \( y(t) = a_0 + a_1 t + a_2 t^2 + \ldots + a_n t^n \)
- Can be added: just add the coefficients
  \[
  (y + z)(t) = (a_0 + b_0) + (a_1 + b_1)t + (a_2 + b_2)t^2 + \ldots + (a_n + b_n)t^n
  \]
- Can be multiplied by a scalar: multiply the coefficients
  \[
  s \cdot y(t) = (s \cdot a_0) + (s \cdot a_1)t + (s \cdot a_2)t^2 + \ldots + (s \cdot a_n)t^n
  \]
Polynomials as a Vector Space

- In 3D, each vector has three components $x, y, z$
- But geometrically, each vector is actually the sum
  \[ \mathbf{v} = x \mathbf{i} + y \mathbf{j} + z \mathbf{k} \]
  \[ \mathbf{j} \]
  \[ \mathbf{k} \]
  \[ \mathbf{i} \]
- $\mathbf{i}, \mathbf{j}, \mathbf{k}$ are basis vectors
- In the polynomial vector space, \{1, $t$, ..., $t^n$\} are the basis vectors, $a_0, a_1, ..., a_n$ are the components
  \[ y(t) = a_0 + a_1 t + a_2 t^2 + \ldots + a_n t^n \]
Subset of Polynomials: Cubic

\[ y(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \]

- Closed under addition & scalar multiplication
  - Means the result is still a cubic polynomial (verify!)
- This means it is also a vector space, a 4D subspace of the full space of polynomials
  - How many dimensions does the full space have?
- The \( x \) and \( y \) coordinates of cubic Bézier curves belong to this subspace as functions of \( t \).
Basis for Cubic Polynomials

More precisely:

What’s a basis?

- A set of “atomic” vectors
  - Called basis vectors
  - Linear combinations of basis vectors span the space
    - i.e. any cubic polynomial is a sum of those basis cubics

- Linearly independent
  - Means that no basis vector can be obtained from the others by linear combination
    - Example: $\mathbf{i}$, $\mathbf{j}$, $\mathbf{i}+\mathbf{j}$ is not a basis (missing $\mathbf{k}$ direction!)

\[ \mathbf{v} = x \mathbf{i} + y \mathbf{j} + z \mathbf{k} \]
Canonical Basis for Cubics

\{1, t, t^2, t^3\}

• Any cubic polynomial is a linear combination of these
  \[a_0 + a_1 t + a_2 t^2 + a_3 t^3 = a_0 \cdot 1 + a_1 \cdot t + a_2 \cdot t^2 + a_3 \cdot t^3\]

• They are linearly independent
  – Means you can’t write any of the four monomials as a linear combination of the others. (You can try.)
Different basis

- For example:
  - \{1, 1+t, 1+t+t^2, 1+t-t^2+t^3\}
  - \{t^3, t^3+t^2, t^3+t, t^3+1\}

- These can all be obtained from \(1, t, t^2, t^3\) by linear combination

- Infinite number of possibilities, just like you have an infinite number of bases to span \(\mathbb{R}^2\)

- For Bézier curves, the basis polynomials/vectors are Bernstein polynomials
Matrix-Vector Notation

• For example:
  - $1$, $1+t$, $1+t+t^2$, $1+t-t^2+t^3$
  - $t^3$, $t^3+t^2$, $t^3+t$, $t^3+1$

These relationships hold for each value of $t$

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

\[
\begin{pmatrix}
t^3 \\
t^3 + t^2 \\
t^3 + t \\
t^3 + 1
\end{pmatrix} =
\begin{pmatrix}
0 & 0 & 0 & 1 \\
0 & 0 & 1 & 1 \\
0 & 1 & 0 & 1 \\
1 & 0 & 0 & 1
\end{pmatrix}
\begin{pmatrix}
t \\
t \\
t^2 \\
t^3
\end{pmatrix}
\]
Matrix-Vector Notation

- For example:
  - $1, 1+t, 1+t+t^2, 1+t-t^2+t^3$
  - $t^3, t^3+t^2, t^3+t, t^3+1$

$$
\begin{pmatrix}
1 \\
1+t \\
1+t+t^2 \\
1+t-t^2+t^3
\end{pmatrix}
=\begin{pmatrix}
1 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 \\
1 & 1 & 1 & 0 \\
1 & 1 & -1 & 1
\end{pmatrix}
\begin{pmatrix}
t \\
t^2 \\
t^3
\end{pmatrix}
$$

- Not any matrix will do! If it’s singular, the basis set will be linearly dependent, i.e., redundant and incomplete.
Bernstein Polynomials

For cubic:
- $B_1(t) = (1-t)^3$
- $B_2(t) = 3t(1-t)^2$
- $B_3(t) = 3t^2(1-t)$
- $B_4(t) = t^3$

- (careful with indices, many authors start at 0)
- But defined for any degree
Properties of Bernstein polynomials

- $\geq 0$ for all $0 \leq t \leq 1$
- Sum to 1 for every $t$
  - called *partition of unity*
- These two together are the reason why Bézier curves lie within convex hull
- Only $B_1$ is non-zero at 0
  - Bezier interpolates $P_1$
  - Same for $B_4$ and $P_4$ for $t=1$
Bézier in Bernstein basis

- \( P(t) = P_1 B_1(t) + P_2 B_2(t) + P_3 B_3(t) + P_4 B_4(t) \)
  - \( P_i \) are 2D points \((x_i, y_i)\)

- \( P(t) \) is a linear combination of the control points with weights the Bernstein polynomials at \( t \)

- But at the same time, the control points \((P_1, P_2, P_3, P_4)\) are the “coordinates” of the curve in the Bernstein basis
  - In this sense, specifying a Bézier curve with control points is exactly like specifying a 2D point with its \( x \) and \( y \) coordinates.
Two different vector spaces!!!

- The plane where the curve lies, a 2D vector space
- The space of cubic polynomials, a 4D space
- Don’t be confused!
- The 2D control points can be replaced by 3D points – this yields space curves.
  - The math stays the same, just add $z(t)$.
- The cubic basis can be extended to higher-order polynomials
  - More control points
  - Higher-dimensional vector space
Questions?
How do we go from Bernstein basis to the canonical monomial basis $1, t, t^2, t^3$ and back?

– With a matrix!

$$
\begin{pmatrix}
B_1(t) \\
B_2(t) \\
B_3(t) \\
B_4(t)
\end{pmatrix}
= \begin{pmatrix}
1 & -3 & 3 & -1 \\
0 & 3 & -6 & 3 \\
0 & 0 & 3 & -3 \\
0 & 0 & 0 & 1
\end{pmatrix}
\begin{pmatrix}
1 \\
t \\
t^2 \\
t^3
\end{pmatrix}
$$

- $B_1(t) = (1-t)^3$
- $B_2(t) = 3t(1-t)^2$
- $B_3(t) = 3t^2(1-t)$
- $B_4(t) = t^3$
Cubic Bernstein:
- $B_1(t) = (1-t)^3$
- $B_2(t) = 3t(1-t)^2$
- $B_3(t) = 3t^2(1-t)$
- $B_4(t) = t^3$

Expand these out and collect powers of $t$. The coefficients are the entries in the matrix $B$!

$$
\begin{pmatrix}
B_1(t) \\
B_2(t) \\
B_3(t) \\
B_4(t)
\end{pmatrix}
= 
\begin{pmatrix}
1 & -3 & 3 & -1 \\
0 & 3 & -6 & 3 \\
0 & 0 & 3 & -3 \\
0 & 0 & 0 & 1
\end{pmatrix}
\begin{pmatrix}
1 \\
t \\
t^2 \\
t^3
\end{pmatrix}
$$
Change of Basis, Other Direction

- Given \( B_1 \ldots B_4 \), how to get back to canonical \( 1, t, t^2, t^3 \)?

\[
\begin{pmatrix}
B_1(t) \\
B_2(t) \\
B_3(t) \\
B_4(t)
\end{pmatrix} = 
\begin{pmatrix}
1 & -3 & 3 & -1 \\
0 & 3 & -6 & 3 \\
0 & 0 & 3 & -3 \\
0 & 0 & 0 & 1
\end{pmatrix} 
\begin{pmatrix}
1 \\
t \\
t^2 \\
t^3
\end{pmatrix}
\]

6.837 – Durand
Change of Basis, Other Direction

That’s right, with the inverse matrix!

- Given $B_1...B_4$, how to get back to canonical $1, t, t^2, t^3$?

\[
\begin{pmatrix}
1 \\
t \\
t^2 \\
t^3
\end{pmatrix}
= \begin{pmatrix}
1 & 1 & 1 & 1 \\
0 & 1/3 & 2/3 & 1 \\
0 & 0 & 1/3 & 1 \\
0 & 0 & 0 & 1
\end{pmatrix}
\begin{pmatrix}
B_1(t) \\
B_2(t) \\
B_3(t) \\
B_4(t)
\end{pmatrix}
\]

6.837 – Durand
Recap

• Cubic polynomials form a vector space.
• Bernstein basis is canonical for Bézier.
  – Can be seen as influence function of data points
  – Or data points are coordinates of the curve in the Bernstein basis
• We can change between basis with matrices.
More Matrix-Vector Notation

• Remember:

\[ P(t) = \sum_{i=1}^{4} P_i B_i(t) = \sum_{i=1}^{4} \begin{pmatrix} x_i \\ y_i \end{pmatrix} B_i(t) \]

• or, in matrix-vector notation

\[ \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \begin{pmatrix} x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \end{pmatrix} \begin{pmatrix} B_1(t) \\ B_2(t) \\ B_3(t) \\ B_4(t) \end{pmatrix} \]

point on curve (2x1 vector)  
matrix of control points (2 x 4)  
Bernstein polynomials (4x1 vector)
Flashback

\[
\begin{pmatrix}
B_1(t) \\
B_2(t) \\
B_3(t) \\
B_4(t)
\end{pmatrix}
= \begin{pmatrix}
1 & -3 & 3 & -1 \\
0 & 3 & -6 & 3 \\
0 & 0 & 3 & -3 \\
0 & 0 & 0 & 1
\end{pmatrix}
\begin{pmatrix}
1 \\
t \\
t^2 \\
t^3
\end{pmatrix}
\]
Phase 3: Profit (again)

- Cubic Bézier in matrix notation

\[
P(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \begin{pmatrix} x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \end{pmatrix} \begin{pmatrix} 1 & -3 & 3 & -1 \\ 0 & 3 & -6 & 3 \\ 0 & 0 & 3 & -3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ t \\ t^2 \\ t^3 \end{pmatrix}
\]

Point on curve
(2x1 vector)

"Geometry matrix"
of control points P₁..P₄
(2 x 4)

"Spline matrix"
(Bernstein)

Canonical monomial basis

Tuesday, September 14, 2010
General Spline Formulation

\[ Q(t) = GBT(t) = \text{Geometry } G \cdot \text{Spline Basis } B \cdot \text{Power Basis } T(t) \]

• Geometry: control points coordinates assembled into a matrix \((P_1, P_2, \ldots, P_{n+1})\)
• Spline matrix: defines the type of spline
  – Bernstein for Bézier
• Power basis: the monomials \((1, t, \ldots, t^n)\)
• Advantage of general formulation
  – Compact expression
  – Easy to convert between types of splines
  – Dimensionality (plane or space) doesn’t really matter
Question?
A Cubic Only Gets You So Far

• What if you want more control?
Higher-Order Bézier Curves

• > 4 control points
• Bernstein Polynomials as the basis functions
  – For polynomial of order n, the i\(^{th}\) basis function is

\[
B_i^n(t) = \frac{n!}{i!(n-i)!} t^i (1-t)^{n-i}
\]

• Every control point affects the entire curve
  – Not simply a local effect
  – More difficult to control for modeling

• You will not need this in this class
Subdivision of a Bézier curve

• Can we split a Bezier curve into two in the middle, using two new Bézier curves?
  – Would be useful for adding detail, as a single cubic doesn’t get you very far, and higher-order curves are nasty.
Subdivision of a Bezier curve

• Can we split a Bezier curve into two in the middle, using two Bézier curves?
  – The resulting curves are again a cubic (Why? A cubic in $t$ is also a cubic in $2t$)
  – Hence it must be representable using the Bernstein basis. So yes, we can!
De Casteljau Construction

- Take the middle point of each of the 3 segments
- Construct the two segments joining them
- Take the middle of those two new segments
- Join them
- Take the middle point P'''

\[ P' \]
\[ P''_1 \]
\[ P''_2 \]
\[ P'_{1} \]
\[ P'_{3} \]
\[ P''' \]
The two new curves are defined by
- \( P_1, P'_1, P''_1, \) and \( P''' \)
- \( P''', P''_2, P'_3, \) and \( P_4 \)

Together they exactly replicate the original curve!
- Originally 4 control points, now 7 (more control)

Result of Split in Middle
Sanity Check

• Do we get the middle point?
• $B_1(t) = (1-t)^3$
• $B_2(t) = 3t(1-t)^2$
• $B_3(t) = 3t^2(1-t)$
• $B_4(t) = t^3$

\[ P'_1 = 0.5(P_1 + P_2) \]
\[ P'_2 = 0.5(P_2 + P_3) \]
\[ P'_3 = 0.5(P_3 + P_4) \]

\[ P''_1 = 0.5(P'_1 + P'_2) \]
\[ P''_2 = 0.5(P'_2 + P'_3) \]
\[ P''_3 = 0.5(P'_3 + P'_4) \]

\[ P''' = 0.5(P''_1 + P''_2) \]
\[ = 0.5 \left( 0.5(P'_1 + P'_2) + 0.5(P'_2 + P'_3) \right) \]
\[ = 0.5 \left( 0.5 \left[ 0.5(P_1 + P_2) + 0.5(P_2 + P_3) \right] + 0.5 \left[ 0.5(P_2 + P_3) + 0.5(P_3 + P_4) \right] \right) \]
\[ = 1/8P_1 + 3/8P_2 + 3/8P_3 + 1/8P_4 \]
De Casteljau Construction

- Actually works to construct a point at any $t$, not just 0.5
- Just subdivide the segments with ratio $(1-t)$, $t$ (not in the middle)
Recap

- Bezier curves: Piecewise polynomials
- Linear combination of basis functions
  - Coefficient = control point
- Bernstein basis
- All linear, matrix algebra
- Subdivision by de Casteljau algorithm
- Be careful with the ordering of basis functions, there is no single convention in the literature!
  - Spline matrices may be transposed, reordered etc.
That’s All for Today, Folks

• Further reading
  – Buss, Chapters 7 and 8

  – Fun stuff to know about function/vector spaces

• **Inkscape** is an open source vector drawing program for Mac/Windows. Try it out!
Questions?
• What if we want to transform each point on the curve with a linear transformation $M$?

\[
P'(t) = M \begin{pmatrix} P_{1,x} & P_{2,x} & P_{3,x} & P_{4,x} \\ P_{1,y} & P_{2,y} & P_{3,y} & P_{4,y} \end{pmatrix} \begin{pmatrix} 1 & -3 & 3 & -1 \\ 0 & 3 & -6 & 3 \\ 0 & 0 & 3 & -3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ t \\ t^2 \\ t^3 \end{pmatrix}
\]
Linear Transformations & Cubics

What if we want to transform each point on the curve with a linear transformation $M$?

- Because everything is linear, it’s the same as transforming the only the control points!

$$P'(t) = M \left( \begin{array}{cccc} P_{1,x} & P_{2,x} & P_{3,x} & P_{4,x} \\ P_{1,y} & P_{2,y} & P_{3,y} & P_{4,y} \end{array} \right) \left( \begin{array}{cccc} 1 & -3 & 3 & -1 \\ 0 & 3 & -6 & 3 \\ 0 & 0 & 3 & -3 \\ 0 & 0 & 0 & 1 \end{array} \right) \left( \begin{array}{c} 1 \\ t \\ t^2 \\ t^3 \end{array} \right)$$

$$= M \left( \begin{array}{cccc} P_{1,x} & P_{2,x} & P_{3,x} & P_{4,x} \\ P_{1,y} & P_{2,y} & P_{3,y} & P_{4,y} \end{array} \right) \left( \begin{array}{cccc} 1 & -3 & 3 & -1 \\ 0 & 3 & -6 & 3 \\ 0 & 0 & 3 & -3 \\ 0 & 0 & 0 & 1 \end{array} \right) \left( \begin{array}{c} 1 \\ t \\ t^2 \\ t^3 \end{array} \right)$$
Linear Transformations & Cubics

• Homogeneous coordinates also work!
  – Means you can translate, rotate, shear, etc.
  – You can do even perspective transformations!

  • Note though that you need to normalize $P'$ by $1/w$

$$P'(t) = \begin{pmatrix} P_{1,x} & P_{2,x} & P_{3,x} & P_{4,x} \\ P_{1,y} & P_{2,y} & P_{3,y} & P_{4,y} \end{pmatrix} \begin{pmatrix} 1 & -3 & 3 & -1 \\ 0 & 3 & -6 & 3 \\ 0 & 0 & 3 & -3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ t \\ t^2 \\ t^3 \end{pmatrix}$$

$$= \begin{pmatrix} P_{1,x} & P_{2,x} & P_{3,x} & P_{4,x} \\ P_{1,y} & P_{2,y} & P_{3,y} & P_{4,y} \end{pmatrix} \begin{pmatrix} 1 & -3 & 3 & -1 \\ 0 & 3 & -6 & 3 \\ 0 & 0 & 3 & -3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ t \\ t^2 \\ t^3 \end{pmatrix}$$
Questions?
Orders of Continuity

- $C^0 = \text{continuous}$
  - The seam can be a sharp kink
- $G^1 = \text{geometric continuity}$
  - Tangents **point to the same direction** at the seam
- $C^1 = \text{parametric continuity}$
  - Tangents **are the same** at the seam, implies $G^1$
- $C^2 = \text{curvature continuity}$
  - Tangents and their derivatives are the same
Connecting Cubic Bézier Curves

- How can we guarantee $C^0$ continuity?
- How can we guarantee $G^1$ continuity?
- How can we guarantee $C^1$ continuity?
- $C^2$ and higher gets difficult, no real solutions.
Connecting Cubic Bézier Curves

- Where is this curve
  - $C^0$ continuous?
  - $G^1$ continuous?
  - $C^1$ continuous?

- What’s the relationship between:
  - the # of control points, and the # of cubic Bézier subcurves?
Cubic BSplines

- $\geq 4$ control points
- Locally cubic
  - Cubics chained together, again.
Cubic BSplines

- \( \geq 4 \) control points
- Locally cubic
  - Cubics chained together, again.
Cubic BSplines

- \( \geq 4 \) control points
- Locally cubic
  - Cubics chained together, again.
Cubic BSplines

- $\geq 4$ control points
- Locally cubic
  - Cubics chained together, again.
Cubic BSplines

- \( \geq 4 \) control points
- Locally cubic
  - Cubics chained together, again.
- Curve is not constrained to pass through any control points

A BSpline curve is also bounded by the convex hull of its control points.
Cubic BSplines: Basis

\[ B_1(t) = \frac{1}{6}(1 - t)^3 \]

\[ B_3(t) = 16(-3t^3 + 3t^2 + 3t + 1) \]

\[ B_2(t) = \frac{1}{6}(3t^3 - 6t^2 + 4) \]

\[ B_4(t) = \frac{1}{6}t^3 \]

6.837 – Durand

Tuesday, September 14, 2010
Cubic BSplines: Basis

\[ Q(t) = \left( \frac{1-t}{6} \right)^3 P_{i-3} + \left( \frac{3t^3 - 6t^2 + 4}{6} \right) P_{i-2} + \left( \frac{-3t^3 + 3t^2 + 3t + 1}{6} \right) P_{i-1} + \left( \frac{t^3}{6} \right) P_i \]

\[ Q(t) = \text{GBT}(t) \]

Basis BSpline:

\[ B_{B-Spline} = \frac{1}{6} \begin{pmatrix} 1 & -3 & 3 & -1 \\ 4 & 0 & -6 & 3 \\ 1 & 3 & 3 & -3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \]
Cubic BSplines

- Local control (windowing)
- Automatically $C^2$, no need to match up tangents!
BSpline Curve Control Points

Default BSpline

BSpline with derivative discontinuity

Repeat interior control point

BSpline which passes through end points

Repeat end points

6.837 – Durand

87
Bézier is not the same as BSpline

- Relationship to the control points is different

Bézier

BSpline
Bézier is not the same as BSpline
Converting between Bézier & BSpline

- Simple with the basis matrices!

\[ B_{\text{Bez}} = \begin{pmatrix} 1 & -3 & 3 & -1 \\ 0 & 3 & -6 & 3 \\ 0 & 0 & 3 & -3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \]

\[ B_{\text{Bspline}} = \frac{1}{6} \begin{pmatrix} 1 & -3 & 3 & -1 \\ 4 & 0 & -6 & 3 \\ 1 & 3 & 3 & -3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \]

- \( G B_1 B_2^{-1} \) are the control points in new basis.

\[ Q(t) = GBT(t) = \text{Geometry } G \cdot \text{Spline Basis } B \cdot \text{Power Basis } T(t) \]
Why Bother with B-Splines?

• Isn’t Bézier good enough?

• Think of specifying a trajectory of an object over time. If $t$ ranged originally from 0 to 1, inserting the control point makes it range from 0 to 2. I.e., the edit affects the whole subsequent trajectory!
Why Bother with B-Splines?

• B-Splines can be split into segments of non-uniform length without affecting the global parametrization.
  – “Non-uniform B-Splines”
  – We’ll not do this, but just so you know.

• Also, automatic $C^2$ is nice!
NURBS (generalized BSplines)

• BSpline: uniform cubic BSpline
• Rational Bezier/cubic
  – Use homogeneous coordinates
• NURBS: Non-Uniform Rational BSpline
  – non-uniform =
    different spacing between the blending functions,
    a.k.a. “knots”
  – rational =
    ratio of cubic polynomials (instead of just cubic)
Brain Teaser

• Can you represent circles using cubics?
  – If yes, how?
  – If no, can you tell what more is needed?
That’s All for Today, Folks

• Further reading
  – Buss, Chapters 7 and 8
  – Fun stuff to know about function/vector spaces

• **Inkscape** is an open source vector drawing program for Mac/Windows. Try it out!