Specifying Curves from Points

• Want to modulate curves via “control” points.

• Strategy depends on application. Possibilities:
  – Force a polynomial of degree N-1 through N points (Lagrange interpolate)
  – Specify a combination of “anchor” points and derivatives (Hermite interpolate)
  – Other “blends” (Bézier, B-splines) -- more useful than Lagrange/Hermite
Specifying Curves from Points-II

• Issues:
  – Continuity of curve and derivatives (geometric, parametric)
  – Local versus global control
  – Polynomials verses other forms
  – Higher polynomial degree versus stitching lower order polynomials together
  – Polynomial degree (usually 3--fewer is not flexible enough, and higher gives hard to control wiggles).
Parametric vs Geometric Continuity

- **Parametric continuity:**
  - The curve and derivatives up to $k$ are continuous *as a function of parameter value*
  - $C^k$
  - Useful for (for example) animation

- **Geometric continuity**
  - curve, derivatives up to $k$’th are the same for equivalent parameter values
  - $D^k$
  - i.e. there exists a reparametrisation that would achieve parametric continuity
  - Useful, because we often don’t require parametric continuity,
Lagrange Interpolate

- Construct a parametric curve that passes through (interpolates) a set of points.
- Lagrange interpolate:
  - give parameter values associated with each point
  - use Lagrange polynomials (one at the relevant point, zero at all others) to construct curve
  - curve is:

\[
\sum_{i \in \text{points}} p_i \phi_i^{(l)}(t)
\]

- Degree is (#pts-1)
  - e.g. line through two points
  - quadratic through three.
- Functions phi are known as “blending functions”
Hermite curves

• Hermite interpolate
  – Curve passes through specified points and has specified derivatives at those points.
  – Curve is:

\[ \sum_{i \in \text{points}} p_i \phi_i^{(h)}(t) + \sum_{i \in \text{points}} v_i \phi_i^{(hd)}(t) \]
Blended curves (§9.2)

- Assume degree 3
- Includes Hermite (§9.2.1), Bézier (§9.2.2), and others

\[
Q(t) = \begin{bmatrix} x(t) \\ y(t) \\ z(t) \end{bmatrix} = CT = \begin{bmatrix} a_x & b_x & c_x & d_x \\ a_y & b_y & c_y & d_y \\ a_z & b_z & c_z & d_z \end{bmatrix} \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}
\]
Blended curves (§9.2)

- Assume degree 3
- Includes Hermite (§9.2.1), Bézier (§9.2.2), and others

\[
Q(t) = \begin{bmatrix}
x(t) \\
y(t) \\
z(t)
\end{bmatrix} = [G_1 \quad G_2 \quad G_3 \quad G_4] \begin{bmatrix}
m_{11} & m_{12} & m_{13} & m_{14} \\
m_{21} & m_{22} & m_{23} & m_{24} \\
m_{31} & m_{32} & m_{33} & m_{34} \\
m_{41} & m_{42} & m_{43} & m_{44}
\end{bmatrix} \begin{bmatrix}
t^3 \\
t^2 \\
t \\
1
\end{bmatrix}
\]
Hermite (§9.2.2)

- Geometry matrix
  - First two columns are endpoints
  - Next two columns are derivatives at those points

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

Derivation?
See page 333
Bézier (§9.2.3)

• Curve goes through two control points
• Curve is adjusted by moving two (cubic case) other control points
• Tangent at endpoints is in direction of adjacent control point
• Curve lies in convex hull of all 4 (cubic case) control points.
  – First two columns are endpoints
  – Next two columns are derivatives at those points
Example Bézier Curves
Bézier (§9.2.2)

• Geometry matrix
  – First two columns are endpoints
  – Next two are like derivatives from the Hermite case, but are now defined by
    \[ R_1 = 3(P_2 - P_1) \]
    \[ R_2 = 3(P_4 - P_3) \]
  – Note that this gives our condition on endpoint tangents
  – Factor of 3 gives good “balance” in control point effect (see book exercise 9.9), and is needed to be consistent with other derivations (e.g., Bernstein polynomials, subdivision, etc).
Bézier in standard form (summary)

\[ Q(t) = \begin{bmatrix} x(t) \\ y(t) \\ z(t) \end{bmatrix} = \begin{bmatrix} P_1 & P_4 & R_1 & R_2 \end{bmatrix} M_B \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix} \]

\[ M_B = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \]
Bézier curves - II

- Blending functions are the Bernstein polynomials

\[ c(t) = \sum_{i=0}^{n} p_i B_i^n(t) \]

\[ B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i} \]
Bézier curves - III

• Bernstein polynomials have several important properties
  – they sum to 1, hence curve lies within convex hull of control points
  – curve interpolates its endpoints
  – curve’s tangent at start lies along the vector from \( p_0 \) to \( p_1 \)
  – tangent at end lies along vector from \( p_{n-1} \) to \( p_n \)
Bézier curve tricks - I

- “Pull” a curve toward a control point by doubling the control point
Bézier curve tricks-II

- Close the curve by making last point and first point coincident
  - curve has continuous tangent if first segment and last segment are collinear