CPSC 424
Review I (curves)

© Wolfgang Heidrich & Alla Sheffer

Curves: Review

Curves in 2D and 3D

- Implicit vs. Explicit vs. Parametric curves
- Bézier curves, de Casteljau algorithm
- Continuity

© Wolfgang Heidrich & Alla Sheffer

Curves & Surfaces as Explicit Functions

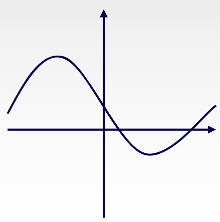
Curves:

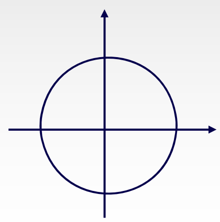
$$y = F(x)$$

Surfaces:

$$z = F(x, y)$$

Examples:





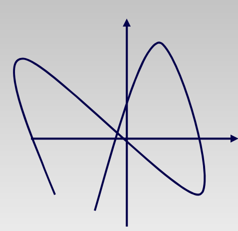
Not a function in Cartesian coord.,

$$y = \pm\sqrt{1 - x^2}$$

© Wolfgang Heidrich & Alla Sheffer

Curves & Surfaces as Explicit Functions

Not representable as a function:



Limitations of explicit functions:

- Cannot model every curve in 2D
- No true 3D curves possible
  - *All curves confined to a plane*

© Wolfgang Heidrich & Alla Sheffer

Curves & Surfaces as Implicit Functions

Curves

$$F(x, y) = 0$$

Surfaces

$$F(x, y, z) = 0$$

Interpretation for curves:

- Iso-lines (contours) in a terrain

Property:

- If F is continuous, implicit curves and surfaces are always closed or extend to infinity

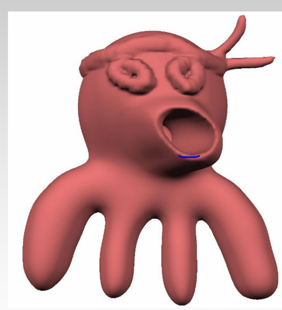
© Wolfgang Heidrich & Alla Sheffer

Curves & Surfaces as Implicit Functions

Conversion:

- Explicit to implicit: trivial $\quad y - F(x) = 0$
- Implicit to explicit: hard
  - *Solving for y involves root finding!*

Limitations of implicit curves:



- Curves only in 2D
  - *Every smooth implicit function in 3D describes a surface!*
- Often unintuitive
- (Difficult to render (display))
- But: useful for many tasks, including modeling, ML,medical imaging

© Wolfgang Heidrich & Alla Sheffer

Curves & Surfaces as Parametric Functions

Concept:

- Curve as function of artificial "time" parameter t

2D curve:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} F_x(t) \\ F_y(t) \end{pmatrix} =: F(t); F : \mathcal{R} \mapsto \mathcal{R}^2$$

3D curve:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} F_x(t) \\ F_y(t) \\ F_z(t) \end{pmatrix} =: F(t); F : \mathcal{R} \mapsto \mathcal{R}^3$$

© Wolfgang Heidrich & Alla Sheffer

Curves & Surfaces as Parametric Functions

Curve example:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \cos t \\ \sin t \\ t \end{pmatrix}$$

Surfaces (in 3D):

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} F_x(s,t) \\ F_y(s,t) \\ F_z(s,t) \end{pmatrix} = F(s,t); F : \mathcal{R}^2 \mapsto \mathcal{R}^3$$

© Wolfgang Heidrich & Alla Sheffer

Curves & Surfaces as Parametric Functions

This works in arbitrary dimensions!

- Curves:

$$\mathbf{x} = F(t); F : \mathcal{R} \mapsto \mathcal{R}^d$$

- Surfaces:

$$\mathbf{x} = F(s,t); F : \mathcal{R}^2 \mapsto \mathcal{R}^d$$

- Hypersurfaces:

$$\mathbf{x} = F(\mathbf{t}); F : \mathcal{R}^n \mapsto \mathcal{R}^d ; n < d$$

Notation:

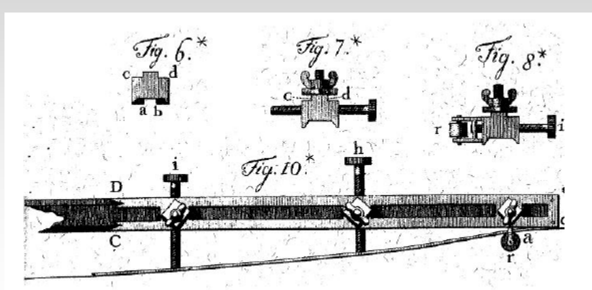
- Bold variables ($\mathbf{t}, \mathbf{x}$) denote vectors, while italics denote scalars (t, d).

© Wolfgang Heidrich & Alla Sheffer

Splines: parametric curves over geometric base

Geometric meaning of coefficients (base)

- Approximate/interpolate set of positions, derivatives, etc..



© Wolfgang Heidrich & Alla Sheffer

Splines

Description = basis functions + coefficients

$$F(t) = \sum_{i=0}^{n} P_i B_i(t) = (x(t), y(t))$$

$$x(t) = \sum_{i=0}^{n} P_i^x B_i(t)$$

$$y(t) = \sum_{i=0}^{n} P_i^y B_i(t)$$

- Same basis functions for all coordinates

© Wolfgang Heidrich & Alla Sheffer

Parametric Spline Curves

Commonly used classes:

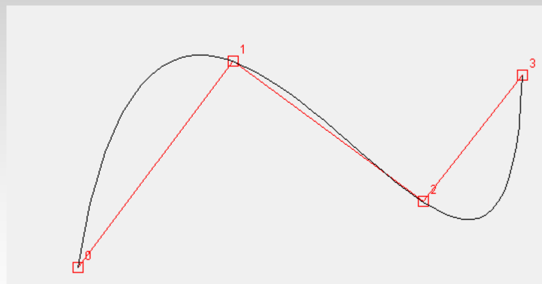
- Polynomials
  - *Lagrange, Bézier, Hermite*
- Piecewise polynomials
  - *B-splines*
- (Rational and piecewise-rational curves)
  - *Rational Bézier curves, rational B-splines (NURBS)*

© Wolfgang Heidrich & Alla Sheffer

Interpolate “Control” Points: Lagrange Polynomials

Use points we want to interpolate as basis

- Polynomial degree = number of input points – 1



© Wolfgang Heidrich & Alla Sheffer

Basis Functions: Lagrange Polynomials

- Given: m+1 parameter values $t_0 \ldots t_m$
- Define

$$L_i^m(t) := \prod_{j=0..m,\, j\neq i} \frac{t - t_j}{t_i - t_j}; i = 0 \ldots m$$

$$L_i^m(t_k) = \begin{cases} 1; i = k \\ 0; else \end{cases}$$

- Lagrange spline

$$F(t) = \sum_{i=0}^{m} L_i^m(t) \cdot b_i$$

© Wolfgang Heidrich & Alla Sheffer

Properties?

© Wolfgang Heidrich & Alla Sheffer

Other Option: Hermite Cubic Basis

Geometrically-oriented coefficients

- 2 positions + 2 tangents

Require $F(0)=P_0$, $F(1)=P_1$, $F'(0)=T_0$, $F'(1)=T_1$

Define basis function per requirement

$$F(t) = P_0 h_{00}(t) + P_1 h_{01}(t) + T_0 h_{10}(t) + T_1 h_{11}(t)$$

© Wolfgang Heidrich & Alla Sheffer

Hermite Cubic Basis

Can satisfy with cubic polynomials as basis

$$h_{ij}(t) = a_3 t^3 + a_2 t^2 + a_1 t + a_0$$

Obtain - solve 4 linear equations in 4 unknowns for each basis function

$$h_{ij}(t): i, j = 0,1, t \in [0,1]$$

curve	$F(0)$	$F(1)$	$F'(0)$	$F'(1)$
$h_{00}(t)$	1	0	0	0
$h_{01}(t)$	0	1	0	0
$h_{10}(t)$	0	0	1	0
$h_{11}(t)$	0	0	0	1

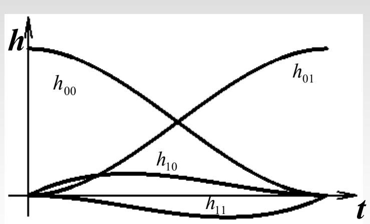
© Wolfgang Heidrich & Alla Sheffer

Hermite Cubic Basis

Four polynomials that satisfy the conditions

$$h_{00}(t) = t^2(2t-3)+1 \qquad h_{01}(t) = -t^2(2t-3)$$

$$h_{10}(t) = t(t-1)^2 \qquad h_{11}(t) = t^2(t-1)$$



© Wolfgang Heidrich & Alla Sheffer

Properties?

© Wolfgang Heidrich & Alla Sheffer

Bézier Curves

Definition:

- Bézier curve is a polynomial curve that uses **Bernstein polynomials** as basis

$$F(t) = \sum_{i=0}^{m} \mathbf{b}_i B_i^m(t)$$

- b_i are called control points of Bézier curve
- Control polygon obtained by connecting control points with line segments

© Wolfgang Heidrich & Alla Sheffer

Bernstein Polynomials

$$B_i^m(t) := \binom{m}{i} t^i (1-t)^{m-i}; i = 0..m; t \in [0,1],$$

$$\binom{m}{i} = \frac{m!}{(m-i)!i!}$$

© Wolfgang Heidrich & Alla Sheffer

Properties?

© Wolfgang Heidrich & Alla Sheffer

Properties of Bézier Curves

- Endpoints b_0 and b_m of control polygon interpolated & corresponding parameter values are t=0 and t=1
- Bézier curve is tangential to control polygon at endpoints
- Curve lies within convex hull of control points
- Curve is *affine invariant*
- There is a fast, recursive evaluation algorithm – de Casteljau algorithm

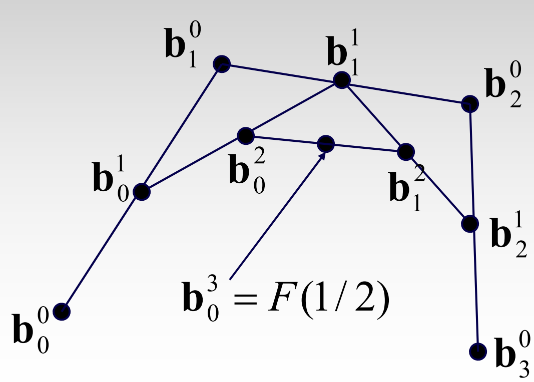
- Which of these apply to: Hermite, Lagrange?

© Wolfgang Heidrich & Alla Sheffer

De Casteljau Algorithm

Graphical Interpretation:

- Determine point $F(1/2)$ for the cubic Bézier curve given by the following four points:



© Wolfgang Heidrich & Alla Sheffer

Bezier Subdivision

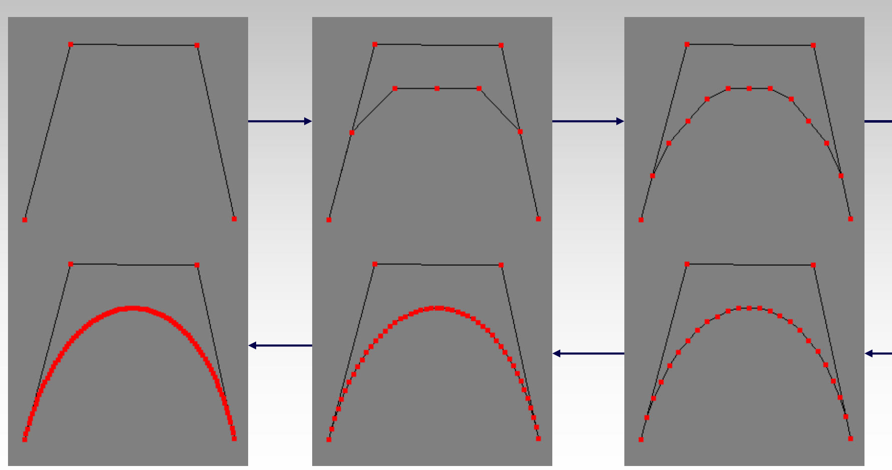
De Casteljau generates 2 new control polygons!

- For parameter interval [0,1/2], and [1/2,1]
- Can be used to recursively subdivide control polygon & approximate actual curve
- Useful for drawing, etc..

© Wolfgang Heidrich & Alla Sheffer

Example

Cubic case:



© Wolfgang Heidrich & Alla Sheffer

Derivatives of Bézier Curves

Theorem:

- The derivative of a Bézier curve

$$F(t) := \sum_{i=0}^{m} B_i^m(t) \cdot \mathbf{b}_i$$

is given as

$$F'(t) := m \cdot \sum_{i=0}^{m-1} B_i^{m-1}(t) \cdot (\mathbf{b}_{i+1} - \mathbf{b}_i)$$

© Wolfgang Heidrich & Alla Sheffer

Continuity

Def:

- A curve $F(t)$ is called C^k-continuous if its k^{th} derivative $F^{(k)}(t)$ exists (i.e. is continuous) everywhere

Note:

- Polynomial curves are infinitely continuous

Def:

- Two curve segments $F(t)$ defined over $[t,t_0]$ and $G(t)$ defined over $[t_0,t']$ are called C^k-continuous at t_0 if their first k derivatives match at t_0
  - *Definition extends to cases with "shifted" parameter intervals $F(t)$ $[t,t_0]$ and $G(t)$ $[t_1,t']$ are called C^k-continuous if at if first k derivatives of F(t) at t_0 match first k derivatives of G(t) at t_1*

© Wolfgang Heidrich & Alla Sheffer

Bezier Continuity

- ***C^0* :** share end control points $b_m = b'_0$
- ***C^1* :** $b_m - b_{m-1} = b'_1 - b'_0$
- ***G^1* :** $b_m - b_{m-1}$ collinear to $b'_1 - b'_0$

© Wolfgang Heidrich & Alla Sheffer