

Lecture 12

Interpolation/Splines

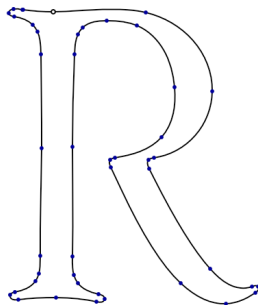
T. Gambill

Department of Computer Science
University of Illinois at Urbana-Champaign

April 12, 2011

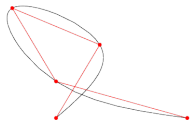
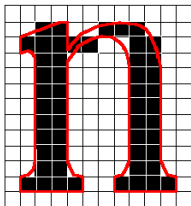


Fonts == interpolation



- how do we "contain" our interpolation?
- splines
- Postscript (Adobe): rasterization on-the-fly. Fonts, etc are defined as cubic Bézier curves (linear interpolation between lower order Bézier curves)
- TrueType (Apple): similar, quadratic Bézier curves, thus cannot convert from TrueType to PS (Type1) losslessly

Why Splines?



- truetype fonts, postscript, metafonts
- graphics surfaces
- smooth surfaces are needed
- how do we interpolate smoothly a set of data?
- keywords: Bezier Curves, splines, B-splines, NURBS
- basic tool: piecewise interpolation

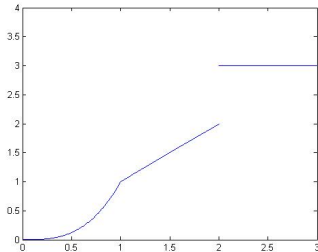


Piecewise Polynomial

A function $f(x)$ is considered a piecewise polynomial on $[a, b]$ if there exists a (finite) partition P of $[a, b]$ such that $f(x)$ is a polynomial on each $(t_i, t_{i+1}) \in P$.

Example

$$f(x) = \begin{cases} x^3 & x \in [0, 1] \\ x & x \in (1, 2) \\ 3 & x \in [2, 3] \end{cases}$$



What do we want?

- we would like the piecewise polynomial to do two things
 - 1 interpolate (or be close to) some set of data points
 - 2 look nice (smooth)
- one option is called a *spline*

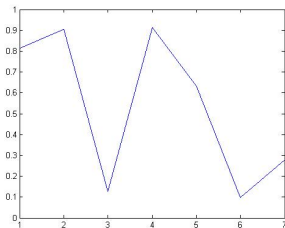


Splines

- A *spline* is a piecewise polynomial with a certain level of smoothness.
- take Matlab:

`plot(1 : 7, rand(1,7))`

- this is linear and continuous, but not very smooth
- the function changes behavior at *knots* $x = 1, x = 2 \dots, x = 7$



Degree 1 Splines

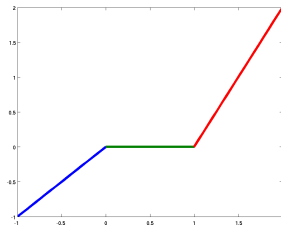
definition

A function $S(x)$ is a spline of degree 1 if:

- 1 The domain of $S(x)$ is an interval $[a, b]$
- 2 $S(x)$ is continuous on $[a, b]$
- 3 There is a partition $a = t_0 < t_1 < \dots < t_n = b$ such that $S(x)$ is linear on each subinterval $[t_i, t_{i+1}]$.

Example

$$S(x) = \begin{cases} x & x \in [-1, 0] \\ 0 & x \in (0, 1) \\ 2x - 2 & x \in [1, 2] \end{cases}$$



Degree 1 Splines

Given data $t_0, \dots, t_n$ and $y_0, \dots, y_n$, how do we form a spline?

We need two things to hold in the interval $[a, b] = [t_0, t_n]$:

- ❶ $S(t_i) = y_i$ for $i = 0, \dots, n$
- ❷ $S(x) = S_i(x) = a_i x + b_i$ for $x \in [t_i, t_{i+1}]$ and $i = 0, \dots, n - 1$

Write $S_i(x)$ in point-slope form

$$\begin{aligned} S_i(x) &= y_i + m_i(x - t_i) \\ &= y_i + \frac{y_{i+1} - y_i}{t_{i+1} - t_i}(x - t_i) \end{aligned}$$



Evaluation of a Degree 1 Splines: Computing $S(x)$

```
1 input  $t, y$  vectors of data
2 input evaluation location  $x$ 
3 find interval  $i$  with  $x \in [t_i, t_{i+1}]$ 
4  $S(x) = y_i + (x - t_i) ((y_{i+1} - y_i) / (t_{i+1} - t_i))$ 
```



Determining the coefficients of $S_i(x)$ for a Degree 1 Spline

Input $n + 1$ data points $t_0, \dots, t_n, y_0, \dots, y_n$

$S_i(x)$ view:

- in each interval we have $S(x) = S_i(x) = a_i x + b_i$ for $x \in [t_i, t_{i+1}]$, and $i = 0, \dots, n - 1$
- 2 unknowns a_i, b_i per interval $[t_i, t_{i+1}]$
- we have $S_i(t_i) = y_i$ and $S_i(t_{i+1}) = y_{i+1}$, for $i = 1, \dots, n - 1$ and $S_0(t_0) = y_0$ and $S_{n-1}(t_n) = y_n$.

$S(x)$ view:

- $2n$ total unknowns
- 2 constraints (equations) per interval gives $2n$ total constraints



Degree 2 Splines

definition

A function $S(x)$ is a spline of degree 2 if:

- 1 The domain of $S(x)$ is an interval $[a, b]$
- 2 $S(x)$ is continuous on $[a, b]$
- 3 $S'(x)$ is continuous on $[a, b]$
- 4 There is a partition $a = t_0 < t_1 < \dots < t_n = b$ such that $S(x)$ is quadratic on each subinterval $[t_i, t_{i+1}]$.



Degree 2 Splines

$$S(x) = \begin{cases} S_0(x) & x \in [t_0, t_1] \\ S_1(x) & x \in [t_1, t_2] \\ \vdots & \vdots \\ S_{n-1}(x) & x \in [t_{n-1}, t_n] \end{cases}$$

for each $i = 0, 1, \dots, n-1$ we have

$$S_i(x) = a_i x^2 + b_i x + c_i$$

What are a_i, b_i, c_i ?



Degree 2 Splines

- 3 unknowns a_i, b_i, c_i in each interval $[t_i, t_{i+1}]$ for $i = 0, 1, \dots, n - 1$
- $3n$ total unknowns
- $2n$ constraints (equations) for matching up the input data (continuity of $S(x)$):
$$S_i(t_i) = y_i, \quad S_i(t_{i+1}) = y_{i+1}$$
- $n - 1$ interior points require continuity of $S'(x)$:
$$S'_i(t_{i+1}) = S'_{i+1}(t_{i+1}) \text{ for } i = 0, 2, \dots, n - 2$$
- but this is just $n - 1$ constraints
- total of $3n - 1$ constraints
- extra constraint: $S'(t_0) = \text{given}$, for example.



Degree 3 Spline: Cubic splines

definition

A function $S(x)$ is a spline of degree 3 if:

- 1 The domain of $S(x)$ is an interval $[a, b]$
- 2 $S(x)$ is continuous on $[a, b]$
- 3 $S'(x)$ is continuous on $[a, b]$
- 4 $S''(x)$ is continuous on $[a, b]$
- 5 There is a partition $a = t_0 < t_1 < \dots < t_n = b$ such that $S(x)$ is cubic on each subinterval $[t_i, t_{i+1}]$.



Degree 3 Splines: Cubic Splines $4n$ Unknowns

In each interval $[t_i, t_{i+1}]$ for $i = 0, 1, \dots, n-1$, $S(x)$ looks like

$$S_i(x) = a_{0,i} + a_{1,i}x + a_{2,i}x^2 + a_{3,i}x^3$$

- n intervals, 4 unknowns per interval
- $4n$ unknowns



Degree 3 Splines: Cubic Splines $4n - 2$ Constraints

In each interval $[t_i, t_{i+1}]$ for $i = 0, 1, \dots, n - 1$, $S(x)$ looks like

$$S_i(x) = a_{0,i} + a_{1,i}x + a_{2,i}x^2 + a_{3,i}x^3$$

- n intervals, 4 unknowns per interval
- $4n$ unknowns



Degree 3 Splines: Cubic Splines $4n - 2$ Constraints

In each interval $[t_i, t_{i+1}]$ for $i = 0, 1, \dots, n - 1$, $S(x)$ looks like

$$S_i(x) = a_{0,i} + a_{1,i}x + a_{2,i}x^2 + a_{3,i}x^3$$

- n intervals, 4 unknowns per interval
- $4n$ unknowns
- $2n$ constraints by continuity: $S_i(t_i) = y_i$ and $S_i(t_{i+1}) = y_{i+1}$ for $i = 0, 1, \dots, n - 1$



Degree 3 Splines: Cubic Splines $4n - 2$ Constraints

In each interval $[t_i, t_{i+1}]$ for $i = 0, 1, \dots, n - 1$, $S(x)$ looks like

$$S_i(x) = a_{0,i} + a_{1,i}x + a_{2,i}x^2 + a_{3,i}x^3$$

- n intervals, 4 unknowns per interval
- $4n$ unknowns
- $2n$ constraints by continuity: $S_i(t_i) = y_i$ and $S_i(t_{i+1}) = y_{i+1}$ for $i = 0, 1, \dots, n - 1$
- $n - 1$ constraints by continuity of $S'(x)$: $S'_i(t_{i+1}) = S'_{i+1}(t_{i+1})$ for $i = 0, 1, \dots, n - 2$



Degree 3 Splines: Cubic Splines $4n - 2$ Constraints

In each interval $[t_i, t_{i+1}]$ for $i = 0, 1, \dots, n - 1$, $S(x)$ looks like

$$S_i(x) = a_{0,i} + a_{1,i}x + a_{2,i}x^2 + a_{3,i}x^3$$

- n intervals, 4 unknowns per interval
- $4n$ unknowns
- $2n$ constraints by continuity: $S_i(t_i) = y_i$ and $S_i(t_{i+1}) = y_{i+1}$ for $i = 0, 1, \dots, n - 1$
- $n - 1$ constraints by continuity of $S'(x)$: $S'_i(t_{i+1}) = S'_{i+1}(t_{i+1})$ for $i = 0, 1, \dots, n - 2$
- $n - 1$ constraints by continuity of $S''(x)$: $S''_i(t_{i+1}) = S''_{i+1}(t_{i+1})$ for $i = 0, 1, \dots, n - 2$



Degree 3 Splines: Cubic Splines $4n - 2$ Constraints

In each interval $[t_i, t_{i+1}]$ for $i = 0, 1, \dots, n-1$, $S(x)$ looks like

$$S_i(x) = a_{0,i} + a_{1,i}x + a_{2,i}x^2 + a_{3,i}x^3$$

- n intervals, 4 unknowns per interval
- $4n$ unknowns
- $2n$ constraints by continuity: $S_i(t_i) = y_i$ and $S_i(t_{i+1}) = y_{i+1}$ for $i = 0, 1, \dots, n-1$
- $n-1$ constraints by continuity of $S'(x)$: $S'_i(t_{i+1}) = S'_{i+1}(t_{i+1})$ for $i = 0, 1, \dots, n-2$
- $n-1$ constraints by continuity of $S''(x)$: $S''_i(t_{i+1}) = S''_{i+1}(t_{i+1})$ for $i = 0, 1, \dots, n-2$
- $4n - 2$ total constraints



Degree 3 Splines: Cubic Splines $4n - 2$ Constraints

In each interval $[t_i, t_{i+1}]$ for $i = 0, 1, \dots, n - 1$, $S(x)$ looks like

$$S_i(x) = a_{0,i} + a_{1,i}x + a_{2,i}x^2 + a_{3,i}x^3$$

- n intervals, 4 unknowns per interval
- $4n$ unknowns
- $2n$ constraints by continuity: $S_i(t_i) = y_i$ and $S_i(t_{i+1}) = y_{i+1}$ for $i = 0, 1, \dots, n - 1$
- $n - 1$ constraints by continuity of $S'(x)$: $S'_i(t_{i+1}) = S'_{i+1}(t_{i+1})$ for $i = 0, 1, \dots, n - 2$
- $n - 1$ constraints by continuity of $S''(x)$: $S''_i(t_{i+1}) = S''_{i+1}(t_{i+1})$ for $i = 0, 1, \dots, n - 2$
- $4n - 2$ total constraints
- This leaves 2 extra degrees of freedom. The cubic spline is not yet unique!



Degree 3 Spline: Cubic Spline

Some options:

- natural cubic spline: $S''(t_0) = S''(t_n) = 0$
- fixed-slope: $S'(t_0) = a, S'(t_n) = b$
- not-a-knot: $S'''(x)$ continuous at t_1 and t_{n-1}
- periodic: S' and S'' are periodic at the ends: $S'(t_0) = S'(t_n)$ and $S''(t_0) = S''(t_n)$



Natural Cubic Spline

How do we find $a_{0,i}, a_{1,i}, a_{2,i}, a_{3,i}$ for each i ?

Consider knots $t_0, \dots, t_n$. Follow our example with the following steps:

- 1 Assume we knew $S''(t_i)$ for each i
- 2 $S''_i(x)$ is linear, so construct it
- 3 Get $S_i(x)$ by integrating $S''_i(x)$ twice
- 4 Impose continuity
- 5 Differentiate $S_i(x)$ to impose continuity on $S'(x)$



Natural Cubic Spline: Step 1

Assume we knew $S''(t_i)$ for each i

We know $S''(x)$ is continuous. So assume

$$z_i = S''(t_i) \text{ for } i = 1, \dots, n-1$$

$$z_0 = z_n = 0 \text{ conditions for natural cubic spline}$$

(we don't actually know z_i , not yet at least)



Natural Cubic Spline: Step 2

$S_i''(x)$ is linear, so construct it

Since $S_i''(x)$ is linear, and

$$\begin{aligned}S_i''(t_i) &= z_i \\S_i''(t_{i+1}) &= z_{i+1}\end{aligned}$$

we can write $S_i''(x)$ as

$$\begin{aligned}S_i''(x) &= \frac{z_{i+1} - z_i}{t_{i+1} - t_i} (x - t_i) + z_i \\&= \frac{z_{i+1}(x - t_i)}{t_{i+1} - t_i} - \frac{z_i(x - t_i)}{t_{i+1} - t_i} + z_i \\&= \frac{z_{i+1}(x - t_i)}{t_{i+1} - t_i} + \frac{z_i(t_{i+1} - x)}{t_{i+1} - t_i} \\&= z_i \frac{t_{i+1} - x}{t_{i+1} - t_i} + z_{i+1} \frac{x - t_i}{t_{i+1} - t_i} \\&= \frac{z_i}{h_i} (t_{i+1} - x) + \frac{z_{i+1}}{h_i} (x - t_i)\end{aligned}$$

where $h_i = t_{i+1} - t_i$



Natural Cubic Spline: Step 3

Get $S_i(x)$ by integrating $S_i''(x)$ twice

Take

$$S_i''(x) = \frac{z_i}{h_i}(t_{i+1} - x) + \frac{z_{i+1}}{h_i}(x - t_i)$$

and integrate once:

$$S_i'(x) = -\frac{z_i}{2h_i}(t_{i+1} - x)^2 + \frac{z_{i+1}}{2h_i}(x - t_i)^2 + \hat{C}_i$$

twice:

$$S_i(x) = \frac{z_i}{6h_i}(t_{i+1} - x)^3 + \frac{z_{i+1}}{6h_i}(x - t_i)^3 + \hat{C}_i x + \hat{D}_i$$

adjust:

$$S_i(x) = \frac{z_i}{6h_i}(t_{i+1} - x)^3 + \frac{z_{i+1}}{6h_i}(x - t_i)^3 + C_i(x - t_i) + D_i(t_{i+1} - x)$$



Natural Cubic Spline: Step 4

Impose continuity

For each interval $[t_i, t_{i+1}]$, we require $S_i(t_i) = y_i$ and $S_i(t_{i+1}) = y_{i+1}$:

$$y_i = S_i(t_i) = \frac{z_i}{6h_i}(t_{i+1} - t_i)^3 + \frac{z_{i+1}}{6h_i}(t_i - t_i)^3 + C_i(t_i - t_i) + D_i(t_{i+1} - t_i)$$

$$= \frac{z_i}{6}h_i^2 + D_i h_i$$

$$D_i = \frac{y_i}{h_i} - \frac{h_i}{6}z_i$$

and

$$y_{i+1} = S_i(t_{i+1}) = \frac{z_i}{6h_i}(t_{i+1} - t_{i+1})^3 + \frac{z_{i+1}}{6h_i}(t_{i+1} - t_i)^3 + C_i(t_{i+1} - t_i) + D_i(t_{i+1} - t_{i+1})$$

$$= \frac{z_{i+1}}{6}(h_i)^2 + C_i h_i$$

$$C_i = \frac{y_{i+1}}{h_i} - \frac{h_i}{6}z_{i+1}$$



Natural Cubic Spline: Step 4

Impose continuity

So far we have

$$S_i(x) = \frac{z_i}{6h_i}(t_{i+1}-x)^3 + \frac{z_{i+1}}{6h_i}(x-t_i)^3 + \left(\frac{y_{i+1}}{h_i} - \frac{h_i}{6}z_{i+1}\right)(x-t_i) + \left(\frac{y_i}{h_i} - \frac{h_i}{6}z_i\right)(t_{i+1}-x)$$

Where the values $t_i, y_i, h_i = t_{i+1} - t_i$ are given as data and only the z_i remain unknown.



Natural Cubic Spline: Step 5

Differentiate $S_i(x)$ to impose continuity on $S'(x)$

$$S'_i(x) = -\frac{z_i}{2h_i}(t_{i+1} - x)^2 + \frac{z_{i+1}}{2h_i}(x - t_i)^2 + \frac{y_{i+1}}{h_i} - \frac{h_i}{6}z_{i+1} - \frac{y_i}{h_i} + \frac{h_i}{6}z_i$$

We need $S'_i(t_i) = S'_{i-1}(t_i)$ for $i = 1, \dots, n-1$:

$$S'_i(t_i) = -\frac{h_i}{6}z_{i+1} - \frac{h_i}{3}z_i + \underbrace{\frac{1}{h_i}(y_{i+1} - y_i)}_{b_i}$$

$$S'_{i-1}(t_i) = \frac{h_{i-1}}{6}z_{i-1} + \frac{h_{i-1}}{3}z_i + \underbrace{\frac{1}{h_{i-1}}(y_i - y_{i-1})}_{b_{i-1}}$$

Thus z_i is defined by

$$h_{i-1}z_{i-1} + 2(h_i + h_{i-1})z_i + h_iz_{i+1} = 6(b_i - b_{i-1})$$



Natural Cubic Spline: Step 6

solve

z_i is defined by

$$h_{i-1}z_{i-1} + 2(h_i + h_{i-1})z_i + h_i z_{i+1} = 6(b_i - b_{i-1})$$

- This is $n + 1$ equations, $n + 1$ unknowns (though $z_0 = z_n = 0$ already)
- an $(n + 1) \times (n + 1)$ tridiagonal system

$$\begin{bmatrix} 1 & & & & & & & & \\ h_0 & u_1 & h_1 & & & & & & \\ & h_1 & u_2 & h_2 & & & & & \\ & & h_2 & u_3 & h_3 & & & & \\ & & & \ddots & \ddots & \ddots & & & \\ & & & & h_{n-3} & u_{n-2} & h_{n-2} & & \\ & & & & & h_{n-2} & u_{n-1} & h_{n-1} & \\ & & & & & & & 1 & \end{bmatrix} \begin{bmatrix} z_0 \\ z_1 \\ z_2 \\ z_3 \\ \vdots \\ z_{n-2} \\ z_{n-1} \\ z_n \end{bmatrix} = \begin{bmatrix} 0 \\ v_1 \\ v_2 \\ v_3 \\ \vdots \\ v_{n-2} \\ v_{n-1} \\ 0 \end{bmatrix}$$

$$u_i = 2(h_i + h_{i-1})$$

$$v_i = 6(b_i - b_{i-1})$$



Example

Find the natural cubic spline for $\begin{array}{c|ccc} x & -1 & 0 & 1 \\ \hline y & 1 & 2 & -1 \end{array}$

- 1 Determine $h_i = t_{i+1} - t_i$, $b_i = \frac{y_{i+1} - y_i}{h_i}$, $u_i = 2(h_i + h_{i-1})$, $v_i = 6(b_i - b_{i-1})$

$$h = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad b = \begin{bmatrix} 1 \\ -3 \end{bmatrix} \quad u = [4] \quad v = [-24]$$

- 2 Solve

$$\begin{bmatrix} 1 & & \\ 1 & 4 & 1 \\ & & 1 \end{bmatrix} \begin{bmatrix} z_0 \\ z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} 0 \\ -24 \\ 0 \end{bmatrix}$$

- 3 Result:

$$\begin{bmatrix} z_0 \\ z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} 0 \\ -6 \\ 0 \end{bmatrix}$$

example

Find the natural cubic spline for $\begin{array}{c|ccc} x & -1 & 0 & 1 \\ \hline y & 1 & 2 & -1 \end{array}$

1 Plug z_i into

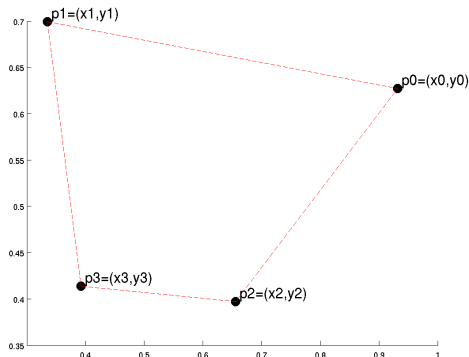
$$\begin{aligned} S_i(x) = & \frac{z_i}{6h_i}(t_{i+1} - x)^3 + \frac{z_{i+1}}{6h_i}(x - t_i)^3 + \left(\frac{y_{i+1}}{h_i} - \frac{h_i}{6}z_{i+1} \right) (x - t_i) \\ & + \left(\frac{y_i}{h_i} - \frac{h_i}{6}z_i \right) (t_{i+1} - x) \end{aligned}$$

$$S(x) = \begin{cases} -(x+1)^3 + 3(x+1) - x & -1 \leq x < 0 \\ -(1-x)^3 - x + 3(1-x) & 0 \leq x < 1 \end{cases}$$



Bézier Curves

- Different than splines
- Similar process
- Does not require interpolation, only that the curve stay within the *convex hull* of the control points
- Can move one point with only local effect



Parametric Form

A function $y = f(x)$ can be expressed in parametric form. The parametric form represents a relationship between x and y through a parameter t :

$$x = F_1(t) \quad y = F_2(t)$$

Example

The equation for a circle can be written in parametric form as

$$x = r \cos(\theta)$$

$$y = r \sin(\theta)$$

(x, y) is now expressed as $(x(t), y(t))$. We will use $0 \leq t \leq 1$.



Bézier Points

Consider a set of *control* points:

$$p_i = (x_i, y_i), \quad i = 0, \dots, n$$

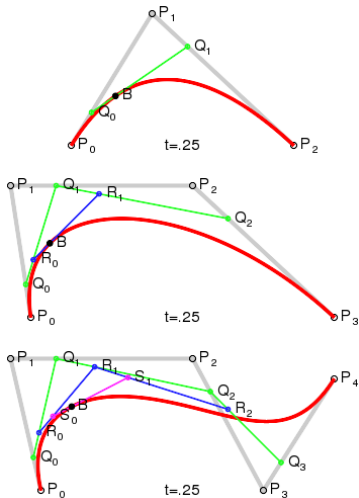
These may be in any order.

So $p_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix}$ or in parametric form the set of points is expressed as

$$P(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix}$$



Bézier Curves



points Q_0 and Q_1 vary linearly
from $P_0 \rightarrow P_1$ and $P_1 \rightarrow P_2$

Q 's vary linearly, R 's vary
quadratically

all within the hull of the control
points

Bernstein Polynomial

The polynomials

$$q(t) = (1-t)^{n-i}t^i$$

have the nice property that for $0 < i < n$, $q(0) = q(1) = 0$.

If we scale them with

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

we have the *Bernstein* polynomials:

$$b_{i,n}(t) = \binom{n}{i} (1-t)^{n-i}t^i$$

Among the interesting properties is that

$$\sum_{i=0}^n b_{i,n}(t) = (x + (1-x))^n = 1$$

(hint: binomial theorem)



Bernstein Polynomial

The n th-degree Bézier Polynomial through the $n + 1$ points is given by

$$p(t) = \sum_{i=0}^n \binom{n}{i} (1-t)^{n-i} t^i p_i$$

where

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

For $n = 3$ (cubic) we have

$$x(t) = (1-t)^3 x_0 + 3(1-t)^2 t x_1 + 3(1-t) t^2 x_2 + t^3 x_3$$

$$y(t) = (1-t)^3 y_0 + 3(1-t)^2 t y_1 + 3(1-t) t^2 y_2 + t^3 y_3$$

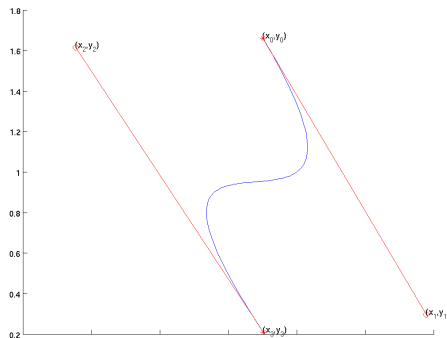
plot_bernstein.m

Cubic Bézier Curve

$$x(t) = (1-t)^3x_0 + 3(1-t)^2tx_1 + 3(1-t)t^2x_2 + t^3x_3$$

$$y(t) = (1-t)^3y_0 + 3(1-t)^2ty_1 + 3(1-t)t^2y_2 + t^3y_3$$

Notice that $(x(0), y(0)) = p_0$ and $(x(1), y(1)) = p_3$. So the Bézier curve interpolates the endpoints but not the interior points.



Bézier Curves

$$x(t) = (1-t)^3x_0 + 3(1-t)^2tx_1 + 3(1-t)t^2x_2 + t^3x_3$$

$$y(t) = (1-t)^3y_0 + 3(1-t)^2ty_1 + 3(1-t)t^2y_2 + t^3y_3$$

Notice:

- 1 $P(0) = p_0$ and $P(1) = p_3$
- 2 The slope of the curve at $t = 0$ is a secant:

$$\frac{dy}{dx} = \frac{dy}{dt} \frac{dt}{dx} = \frac{3(y_1 - y_0)}{3(x_1 - x_0)} = \frac{y_1 - y_0}{x_1 - x_0}$$

- 3 The slope of the curve at $t = 1$ is a secant between the last two control points.
- 4 The curve is contained in the convex hull of the control points



Bézier Curves

$$x(t) = (1-t)^3x_0 + 3(1-t)^2tx_1 + 3(1-t)t^2x_2 + t^3x_3$$

$$y(t) = (1-t)^3y_0 + 3(1-t)^2ty_1 + 3(1-t)t^2y_2 + t^3y_3$$

Easier construction given points $p_0, \dots, p_3$:

$$P(t) = \begin{bmatrix} t^3 & t^2 & t^1 & t^0 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix}$$

see `bezier_demo.m` from Mathworks File Exchange

<http://www.math.psu.edu/dlittle/java/parametricequations/beziercurves/index.html>

Vector Graphics, Fonts, Adobe

Vector Graphics include primitives like

- lines, polygons
- circles
- **Bézier curves**
- **Bézier splines** or Bezigons
- text (letters created from **Bézier** curves)

Flash Animation

- Use **Bézier curves** to construct animation path

Microsoft Paint, Gimp, etc

- Use **Bézier curves** to draw curves
- <http://msdn2.microsoft.com/en-us/library/ms534244.aspx>

Graphics

- Use **Bézier surfaces** to draw smooth objects



Bézier Surfaces

Take (n, m) . That is, $(n + 1, m + 1)$ control points $p_{i,j}$ in 2d. Then let

$$\mathbf{P}(t, s) = \sum_{i=0}^n \sum_{j=0}^m \phi_{ni}(t) \phi_{mj}(s) \mathbf{p}_{ij}$$

Where, again, ϕ_{ni} are the Bernstein polynomials:

$$\phi_{ni}(t) = \binom{n}{i} (1-t)^{n-i} t^i$$

- again, all within the convex hull of control points
- <http://www.math.psu.edu/dlitttle/java/parametricequations/beziersurfaces/index.html>

