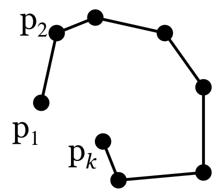
Piecewise Linear Plane Curves

This is the curve representation we've started with

- sequence of vertices $p_1, p_2, \ldots, p_k$
- line segments connect each pair of consecutive vertices



Today, we'll look at piecewise-polynomial curves

- our goal being a more compact description of smooth curves

Representations of Smooth Curves

Explicit curves

$$\begin{pmatrix} x \\ y(x) \end{pmatrix}$$

- single-valued for each value of x
- rotations completely change representation

Implicit curves

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

- supports multiple values for each x
- but often hard to construct & control

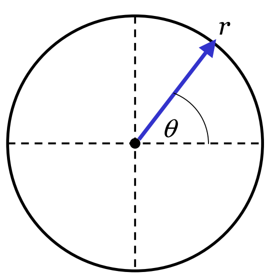
Parametric curves

$$\begin{pmatrix} x(u) \\ y(u) \end{pmatrix}$$

- supports multiple values for each x
- fairly easy to construct & control
- but introduces an extra variable: the parameter u

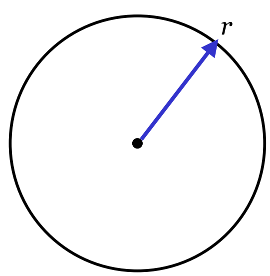
Parametric vs. Implicit Curves

Parametric Circle



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

Implicit Circle



$$F(x,y) = x^2 + y^2 - r^2 = 0$$

inside: $F < 0$

outside: $F > 0$

Parametric Space Curves

Described in general by a vector-valued function

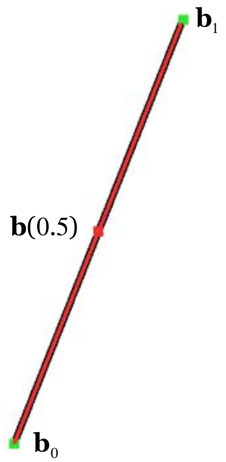
$$\mathbf{p}(u) = \begin{pmatrix} x(u) \\ y(u) \\ z(u) \end{pmatrix}$$

We can define unit tangent and normal vectors as follows

$$\mathbf{t}(u) = \frac{\mathbf{p}'(u)}{\|\mathbf{p}'(u)\|} \qquad \mathbf{n}(u) = \frac{\mathbf{p}''(u)}{\|\mathbf{p}''(u)\|}$$

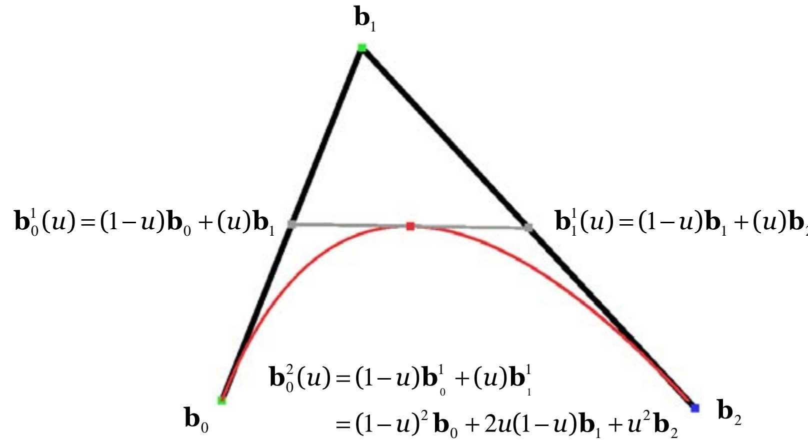
$$\mathbf{p}'(u) = \frac{d}{du}\mathbf{p}(u) = [x'(u) \quad y'(u) \quad z'(u)]$$

Linear Interpolation

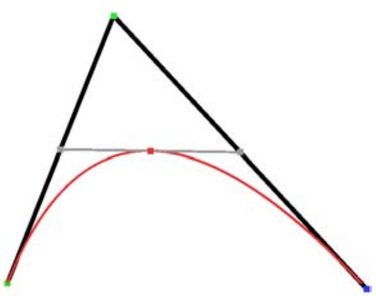


$\mathbf{b}(u) = (1-u)\mathbf{b}_0 + (u)\mathbf{b}_1$ where $0 \leq u \leq 1$

"Doubled" Linear Interpolation



Result: A Second Degree Curve



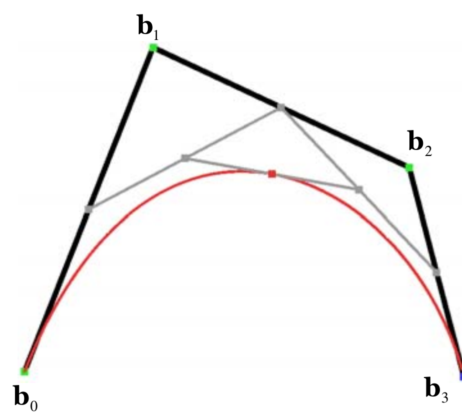
$$\mathbf{b}_0^2(u) = (1-u)\mathbf{b}_0^1 + (u)\mathbf{b}_1^1$$
$$= (1-u)^2\mathbf{b}_0 + 2u(1-u)\mathbf{b}_1 + u^2\mathbf{b}_2$$

Curve is a quadratic polynomial

- weighted combination of original points
- each weight is a quadratic polynomial in u

This is a quadratic Bézier curve

"Tripled" Linear Interpolation



Get a cubic polynomial curve

$$\mathbf{b}_0^3(u) = (1-u)^3\mathbf{b}_0 + 3(1-u)^2(u)\mathbf{b}_1 + 3(1-u)(u)^2\mathbf{b}_2 + (u)^3\mathbf{b}_3$$

This is a cubic Bézier curve

"Tripled" Linear Interpolation

Repeated averaging in tableau form:

Input points

$$
\begin{array}{llll}
\mathbf{b}_0 & & & \\
\mathbf{b}_1 & \mathbf{b}_0^1 & & \\
\mathbf{b}_2 & \mathbf{b}_1^1 & \mathbf{b}_0^2 & \\
\mathbf{b}_3 & \mathbf{b}_2^1 & \mathbf{b}_1^2 & \mathbf{b}_0^3
\end{array}
$$

Point on curve

This clearly suggests a recursive procedure …

General Bézier Curves

Given $n+1$ **control points**

$$\mathbf{b}_0, \mathbf{b}_1, \ldots, \mathbf{b}_n \in R^3$$

We can define a Bézier curve

$$\mathbf{b}(u) = \mathbf{b}^n(u) = \mathbf{b}_0^n(u)$$

via the recursive construction

$$\mathbf{b}_i^r(u) = (1-u)\,\mathbf{b}_i^{r-1}(u) + (u)\,\mathbf{b}_{i+1}^{r-1}(u)$$
$$\mathbf{b}_i^0(u) = \mathbf{b}_i$$

This is the de Casteljau Algorithm

Bernstein Polynomials

We can also write Bézier curves as explicit polynomials

$$\mathbf{b}^n(u) = \sum_{i=0}^{n} \mathbf{b}_i B_i^n(u) \qquad 0 \le u \le 1$$

These weighting functions are the Bernstein polynomials

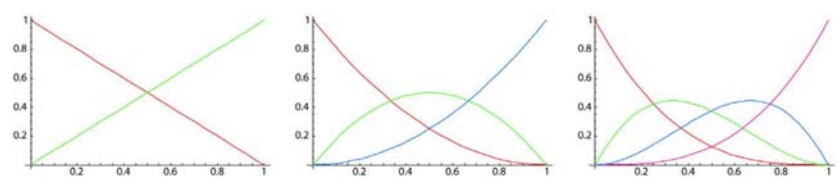
$$B_i^n(u) = \binom{n}{i} u^i (1-u)^{n-i} \qquad \binom{n}{i} = \frac{n!}{i!(n-i)!}$$

Common Bernstein Polynomials

$$B_0^1 = 1-u \qquad B_1^1 = u$$

$$B_0^2 = (1-u)^2 \qquad B_1^2 = 2(1-u)(u) \qquad B_2^2 = u^2$$

$$B_0^3 = (1-u)^3 \qquad B_1^3 = 3(1-u)^2(u) \qquad B_2^3 = 3(1-u)(u)^2 \qquad B_3^3 = u^3$$



Handy Facts about Bernstein Polynomials

Sum of all polynomials of order n is 1

$$\sum_{i=0}^{n} B_i^n(u) = 1$$

- implies Bézier curve always within convex hull of control points

Can also be defined recursively

$$B_i^n(u) = (1-u)B_i^{n-1}(u) + (u)B_{i-1}^{n-1}(u)$$

$$B_0^0(u) = 1$$

Cubic Space Curves

Consider coordinate functions that are cubic polynomials

$$x(u) = a_3u^3 + a_2u^2 + a_1u + a_0$$
$$y(u) = b_3u^3 + b_2u^2 + b_1u + b_0 \qquad \text{where } 0 \le u \le 1$$
$$z(u) = c_3u^3 + c_2u^2 + c_1u + c_0$$

Each is a linear combination of monomial terms

$$x(u) = \sum_{i=0}^{3} a_i u^i$$
$$y(u) = \sum_{i=0}^{3} b_i u^i$$
$$z(u) = \sum_{i=0}^{3} c_i u^i$$

Vector/Matrix Forms for Cubic Curves

For convenience, we can rewrite in vector form

$$x(u) = \sum_{i=0}^{3} a_i u^i \qquad y(u) = \sum_{i=0}^{3} b_i u^i \qquad z(u) = \sum_{i=0}^{3} c_i u^i \qquad \mathbf{a}_i = \begin{pmatrix} a_i \\ b_i \\ c_i \end{pmatrix}$$

$$\mathbf{p}(u) = \mathbf{a}_3u^3 + \mathbf{a}_2u^2 + \mathbf{a}_1u + \mathbf{a}_0$$
$$= \sum_{i=0}^{3} \mathbf{a}_i u^i$$

And in an even more condensed matrix form

$$\mathbf{p}(u) = \begin{pmatrix} 1 & u & u^2 & u^3 \end{pmatrix} \begin{pmatrix} \mathbf{a}_0 \\ \mathbf{a}_1 \\ \mathbf{a}_2 \\ \mathbf{a}_3 \end{pmatrix} = \mathbf{u}^\mathsf{T}\mathbf{A} \qquad \mathbf{u} = \begin{pmatrix} 1 \\ u \\ u^2 \\ u^3 \end{pmatrix}$$

Rewriting with Geometric Constraints

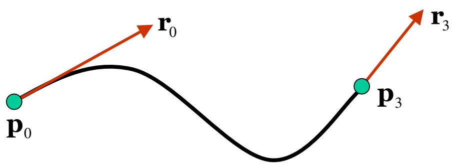
Recall that a cubic is determined by 4 constraint points

- want to rewrite spline formulas in terms of these constraints
- *not* the coefficients of the monomial terms

$$\mathbf{p}(u) = \begin{pmatrix} 1 & u & u^2 & u^3 \end{pmatrix} \underbrace{\begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{pmatrix}}_{\text{basis matrix}} \overbrace{\begin{pmatrix} \mathbf{g}_1 \\ \mathbf{g}_2 \\ \mathbf{g}_3 \\ \mathbf{g}_4 \end{pmatrix}}^{\text{geometry matrix}} = \mathbf{u}^\mathsf{T}\mathbf{M}\mathbf{G}$$

$$\mathbf{p}(u) = \mathbf{u}^\mathsf{T}\mathbf{A} = \mathbf{u}^\mathsf{T}\mathbf{M}\mathbf{G} \quad ♥ \quad \mathbf{A} = \mathbf{M}\mathbf{G}$$

Hermite Curves



$$\mathbf{p}_0 = \mathbf{p}(0)$$
$$\mathbf{p}_3 = \mathbf{p}(1)$$
$$\mathbf{r}_0 = \mathbf{p}'(0)$$
$$\mathbf{r}_3 = \mathbf{p}'(1)$$

$$\mathbf{G} = \begin{pmatrix} \mathbf{p}_0 \\ \mathbf{p}_3 \\ \mathbf{r}_0 \\ \mathbf{r}_3 \end{pmatrix}$$

Specify 4 geometry constraints

- endpoints of the curve segment
- tangent vectors at each of the endpoints

Easy to paste Hermite segments together

- specify coincident endpoints and identical tangents
- guarantees tangents are continuous — C^1 continuity

Deriving the Hermite Basis Matrix

These are the constraints that we want:

$$\mathbf{p}_0 = \mathbf{p}(0) = \mathbf{a}_0$$
$$\mathbf{p}_3 = \mathbf{p}(1) = \mathbf{a}_3 + \mathbf{a}_2 + \mathbf{a}_1 + \mathbf{a}_0$$
$$\mathbf{r}_0 = \mathbf{p}'(0) = \mathbf{a}_1$$
$$\mathbf{r}_3 = \mathbf{p}'(1) = 3\mathbf{a}_3 + 2\mathbf{a}_1 + \mathbf{a}_1$$

$$\mathbf{p}(u) = \mathbf{a}_3 u^3 + \mathbf{a}_2 u^2 + \mathbf{a}_1 u + \mathbf{a}_0$$

We can rewrite this as:

$$\begin{pmatrix} \mathbf{p}_0 \\ \mathbf{p}_3 \\ \mathbf{r}_0 \\ \mathbf{r}_3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & 3 \end{pmatrix} \begin{pmatrix} \mathbf{a}_0 \\ \mathbf{a}_1 \\ \mathbf{a}_2 \\ \mathbf{a}_3 \end{pmatrix}$$

Deriving the Hermite Basis Matrix

Starting from here, we invert the coefficient matrix …

$$\begin{pmatrix} \mathbf{p}_0 \\ \mathbf{p}_3 \\ \mathbf{r}_0 \\ \mathbf{r}_3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & 3 \end{pmatrix} \begin{pmatrix} \mathbf{a}_0 \\ \mathbf{a}_1 \\ \mathbf{a}_2 \\ \mathbf{a}_3 \end{pmatrix}$$

… and solve for the spline coefficient matrix C

$$\begin{pmatrix} \mathbf{a}_0 \\ \mathbf{a}_1 \\ \mathbf{a}_2 \\ \mathbf{a}_3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & 3 \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{p}_0 \\ \mathbf{p}_3 \\ \mathbf{r}_0 \\ \mathbf{r}_3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -3 & 3 & -2 & -1 \\ 2 & -2 & 1 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{p}_0 \\ \mathbf{p}_3 \\ \mathbf{r}_0 \\ \mathbf{r}_3 \end{pmatrix}$$

The Equation for a Hermite Curve

We're done! The curve, in terms of the constraints is

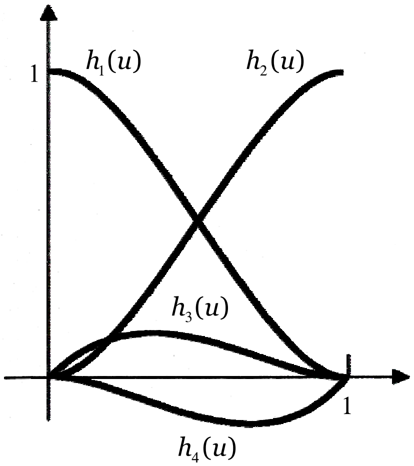
$$\mathbf{p}(u) = \begin{pmatrix} 1 & u & u^2 & u^3 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -3 & 3 & -2 & -1 \\ 2 & -2 & 1 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{p}_0 \\ \mathbf{p}_3 \\ \mathbf{r}_0 \\ \mathbf{r}_3 \end{pmatrix}$$

We can also look at it as a weighted sum of the constraints

$$\mathbf{p}(u) = (2u^3 - 3u^2 + 1)\mathbf{p}_0 + (-2u^3 + 3u^2)\mathbf{p}_3 + (u^3 - 2u^2 + u)\mathbf{r}_0 + (u^3 - u^2)\mathbf{r}_3$$
$$= h_1\mathbf{p}_0 + h_2\mathbf{p}_3 + h_3\mathbf{r}_0 + h_4\mathbf{r}_3$$

- each is weighted by a blending function
- whose coefficients are the columns of the basis matrix

Hermite Blending Polynomials



$$h_1(u) = 2u^3 - 3u^2 + 1$$
$$h_2(u) = -2u^3 + 3u^2$$
$$h_3(u) = u^3 - 2u^2 + u$$
$$h_4(u) = u^3 - u^2$$