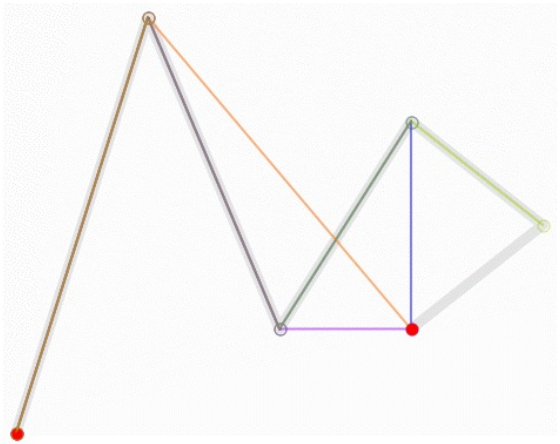
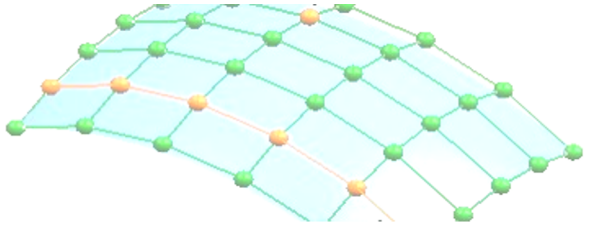
FACULTY OF ENGINEERING AND ARCHITECTURE

E016712: Computer Graphics

Curves and Surfaces





Lecturers: Aleksandra Pizurica and Danilo Babin

GHENT UNIVERSITY

Why modeling curves and curved surfaces?

We worked so far with flat entities such as lines and flat polygons

- Graphics systems render flat (3D) objects at high rates
- Efficient hidden surface removal, shading, texture mapping…
- Mathematically simple

The world is not composed of flat entities

- We can provide the means to work with curved objects
- This can be done at the application level
- Implementation can render these objects approximately with flat primitives

Overview

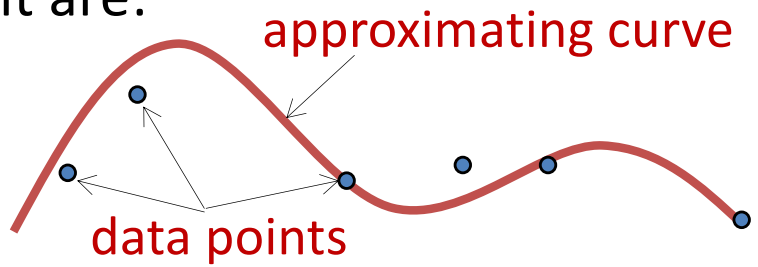
- Representation of curves and surfaces
- Polynomial forms
- Hermite curves
- Bézier curves

Based on:

E. Angel and D. Shreiner: *Interactive Computer* Graphics – *A Top Down Approach with Shader-Based OpenGL* (6th ed). Chapter 10: Curves and Surfaces

On object representation

There are many ways to represent curves and surfaces. Some of the considerations to take into account are:



- Stability
- Smoothness
- Ease of evaluation
- Do we need accurate interpolation or can we just come close to data?
- Do we need derivatives?

In general, there are three major types of object representation

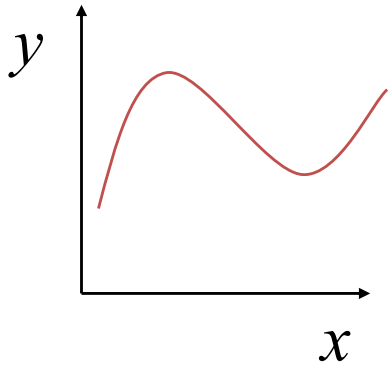
- Explicit
- Implicit
- Parametric

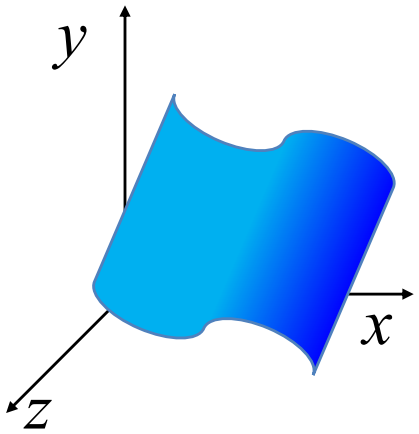
Each of these has certain advantages and disadvantages

Explicit Representation

Most familiar form of curve in 2D

$$y = f(x)$$





Cannot represent all curves

- Vertical lines
- Circles

Extension to 3D

- $y = f(x), z = g(x)$ defines a curve in 3D
  - e.g. equations $y = ax + b;\ z = cx + d$ describe a line in 3D, but cannot represent a line in a plane with constant x
- The form $z = f(x,y)$ defines a surface (but cannot represent a sphere)

Implicit Representation

Two dimensional curve in implicit form:

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

Most curves and surfaces that we work with have implicit forms:

- All lines $ax + by + c = 0$
- Circles $x^2 + y^2 - r^2 = 0$

In three dimensions $f(x,y,z) = 0$ defines a surface

- e.g., any plane: $ax + by + cz + d = 0$, with constants a, b, c and d
- a sphere centered at origin, with radius r: $x^2 + y^2 + z^2 - r^2 = 0$

Curves in 3D are not so easily represented in implicit form.

- Possibly as intersection of two surfaces $f(x,y,z) = 0$ and $g(x,y,z) = 0$

In general, we cannot solve for points that satisfy the implicit form

Implicit Representation, contd.

Algebraic surface

- $f(x,y,z)$ is the sum of polynomials in the three variables

$$f(x,y,z) = \sum_i \sum_j \sum_k x^i y^j z^k = 0$$

Quadric surface

- a special type of algebraic surface where each term in $f(x,y,z)$ can have degree up to 2 (i.e., $i + j + k \leq 2$)
- of interest because they include objects like spheres, disks and cones and they generate at most two intersection points with lines

  → the problem of finding the intersection with a ray reduces to solving a quadratic equation

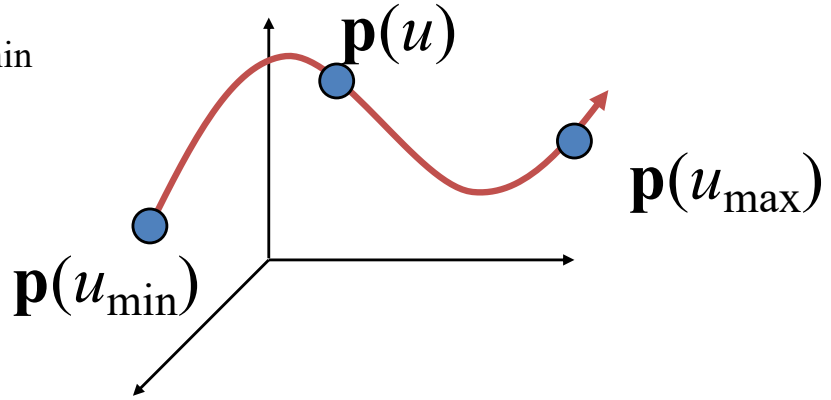
Parametric Representation

Separate equation for each spatial variable, expressed in terms of an independent variable, called the parameter:

$$\left.\begin{aligned} x &= x(u), \\ y &= y(u), \\ z &= z(u) \end{aligned}\right\}$$

Visualize a curve for $u_{\max} \geq u \geq u_{\min}$

$$\mathbf{p}(u)=[x(u),\ y(u),\ z(u)]^{\mathrm{T}}$$



$$\frac{d\mathbf{p}(u)}{du} = \left[\frac{dx(u)}{du} \quad \frac{dy(u)}{du} \quad \frac{dz(u)}{du}\right]^{T}$$

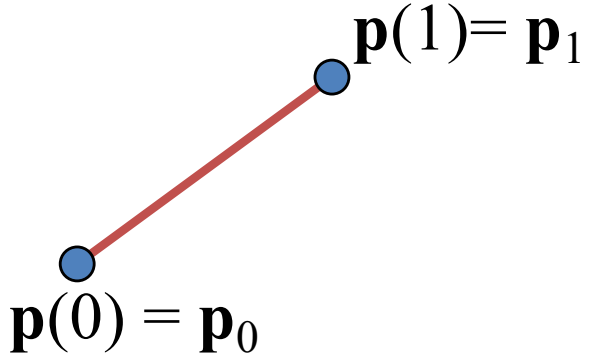
⇨ Gives the velocity with which the curve is traced out and points in the direction tangent to the curve

Parametric Lines

We can normalize u to be over the interval (0,1)

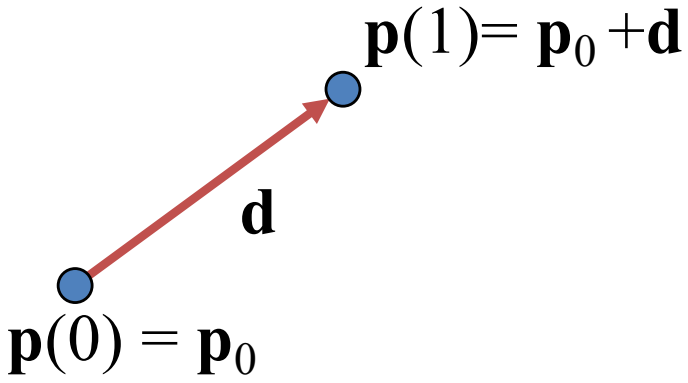
Line connecting two points $\mathbf{p}_0$ and $\mathbf{p}_1$

$$\mathbf{p}(u)=(1-u)\mathbf{p}_0+u\mathbf{p}_1$$



Ray from $\mathbf{p}_0$ in the direction $\mathbf{d}$

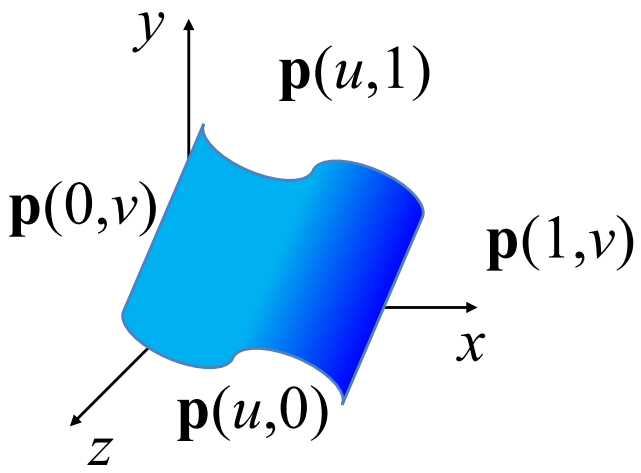
$$\mathbf{p}(u)=\mathbf{p}_0+u\mathbf{d}$$



Parametric Surfaces

Surfaces require 2 parameters

$$x = x(u,v),$$
$$y = y(u,v),$$
$$z = z(u,v)$$



As u and v vary, we generate all the points on the surface

$$\mathbf{p}(u,v) = [x(u,v),\ y(u,v),\ z(u,v)]^{\mathrm{T}}$$

Both for curves and surfaces we choose the functions according to some desired properties:

- Smoothness
- Differentiability
- Ease of evaluation

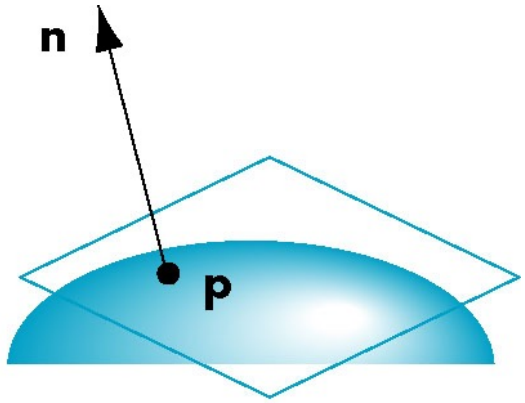
Parametric Representation

The partial derivatives determine the tangent plane at each point of the surface

$$\frac{\partial \mathbf{p}(u,v)}{\partial u} = \begin{bmatrix} \partial x(u,v)/\partial u \\ \partial y(u,v)/\partial u \\ \partial z(u,v)/\partial u \end{bmatrix}; \qquad \frac{\partial \mathbf{p}(u,v)}{\partial v} = \begin{bmatrix} \partial x(u,v)/\partial v \\ \partial y(u,v)/\partial v \\ \partial z(u,v)/\partial v \end{bmatrix}$$

Moreover, we can obtain the normal at any point $\mathbf{p}$ as the cross product of these vectors (as long as they are not parallel):

$$\mathbf{n} = \frac{\partial \mathbf{p}(u,v)}{\partial u} \times \frac{\partial \mathbf{p}(u,v)}{\partial v}$$



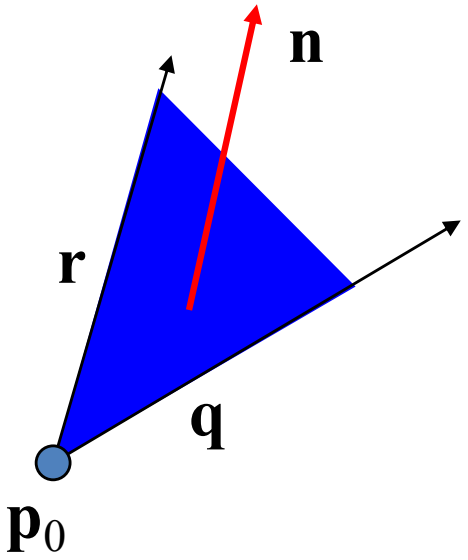
The parametric form is the most flexible and robust for computer graphics

Parametric Planes

Point-vector form

$$\mathbf{p}(u,v)=\mathbf{p}_0+u\,\mathbf{q}+v\,\mathbf{r}$$

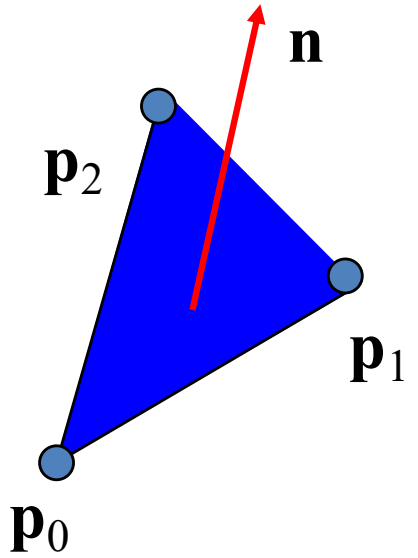
$$\mathbf{n}=\mathbf{q}\times\mathbf{r}$$



Three-point form

$$\mathbf{q}=\mathbf{p}_1-\mathbf{p}_0$$
$$\mathbf{r}=\mathbf{p}_2-\mathbf{p}_0$$

$$\mathbf{n}=\mathbf{q}\times\mathbf{r}$$

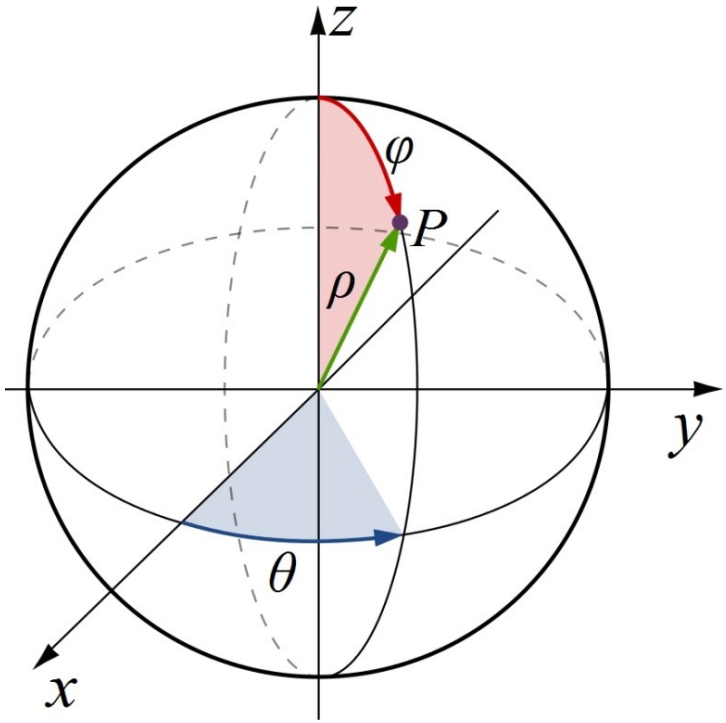


Parametric Sphere

The points on a sphere with radius ρ

$$x(\theta, \varphi) = \rho \cos\theta \sin\varphi$$
$$y(\theta, \varphi) = \rho \sin\theta \sin\varphi$$
$$z(\theta, \varphi) = \rho \cos\varphi$$

$$360 \geq \theta \geq 0$$
$$180 \geq \varphi \geq 0$$



Constant θ : circles of constant longitude

Constant φ : circles of constant latitude

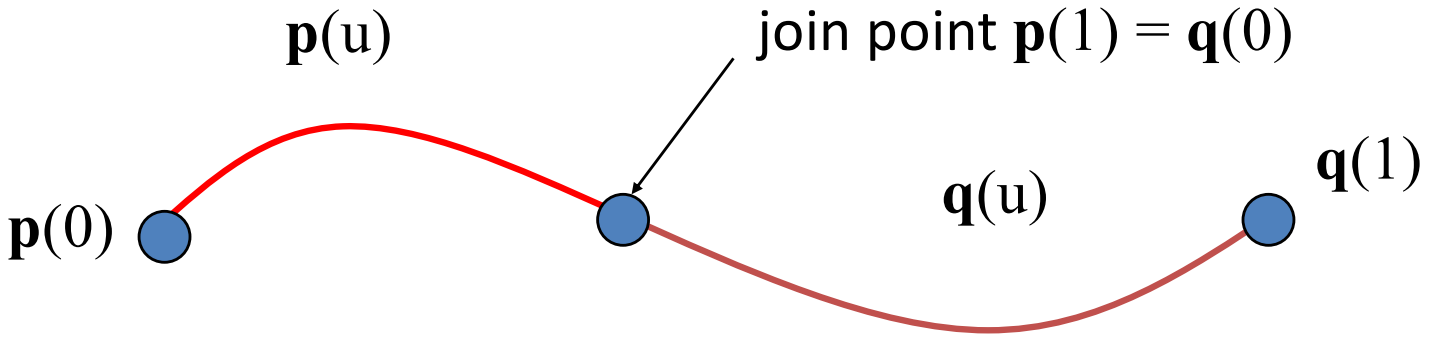
Differentiate to show $\mathbf{n} = \mathbf{p}$

Curve Segments

After normalizing u, each curve is written

$$\mathbf{p}(u)=[x(u), y(u), z(u)]^{\mathrm{T}}, \quad 1 \geq u \geq 0$$

While in classical numerical methods we design a single global curve, in computer graphics it is better to design small connected curve segments



Advantages of designing each segment individually

- working interactively; affecting the shape only where we want
- local control implies stability: small changes in parameters (independent var.) cause small changes in dependent variables

Polynomial representation

Parametric Polynomial Curves

$$x(u)=\sum_{i=0}^{N} c_{xi}\,u^{i}, \qquad y(u)=\sum_{j=0}^{M} c_{yj}\,u^{j}, \qquad z(u)=\sum_{k=0}^{K} c_{zk}\,u^{k}$$

If $N=M=K$, we need to determine $3(N+1)$ coefficients. Equivalently we need $3(N+1)$ independent conditions

Noting that the curves for x, y and z are independent, we can define each independently in an identical manner. We can use the form

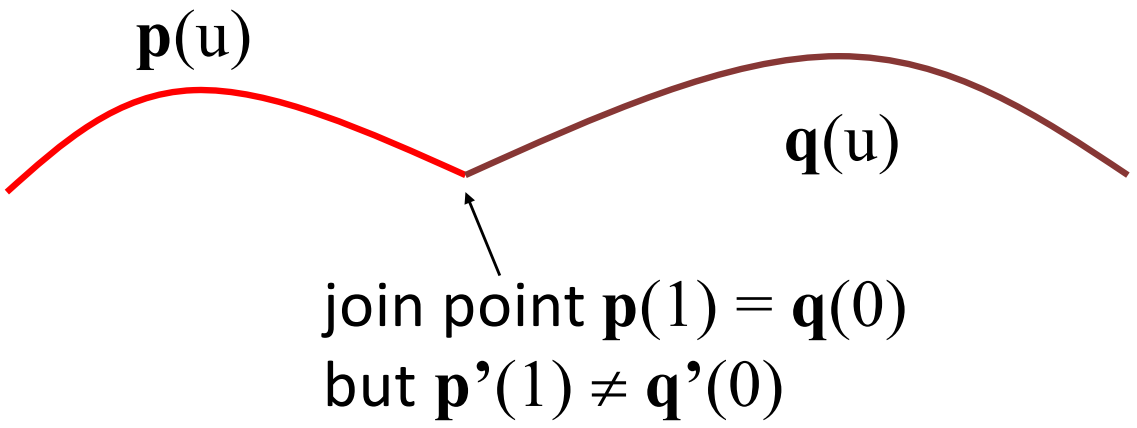
$$p(u)=\sum_{k=0}^{L} c_k\,u^{k}$$

where p can be any of x, y, z. Or we can write in a vector form

$$\mathbf{p}(u)=[x(u)\;\; y(u)\;\; z(u)]^{T}, \qquad \mathbf{p}(u)=\sum_{k=0}^{L}\mathbf{c}_k u^{k}, \qquad \mathbf{c}_k=[c_{xk}\;\; c_{yk}\;\; c_{zk}]^{T}$$

Why Polynomials?

- Easy to evaluate
- Continuous and differentiable everywhere
  - Remember that we work with curve segments. It is important to ensure continuity at join points including continuity of derivatives
  - For a polynomial curve all derivatives exist and can be computed analytically



Cubic Parametric Polynomials

- Choosing the degree of the polynomial curve involves a trade-off
  - higher degree = more parameters to set → better ability to form the shape
  - however, more costly and more danger that the curve will become rougher
- $N = M = L = 3$, is considered as a good balance between ease of evaluation and flexibility in design

$$p(u) = \sum_{k=0}^{3} c_k u^k$$

- Four coefficients to determine for each of x, y and z

  → Four independent conditions needed for various values of u resulting in 4 equations in 4 unknowns for each of x, y and z
- The conditions are a mixture of continuity requirements at the join points and conditions for fitting the data

Cubic Polynomial Surfaces

$\mathbf{p}(u,v) = [x(u,v), y(u,v), z(u,v)]^{\mathrm{T}}$, where

$$p(u,v) = \sum_{i=0}^{3} \sum_{j=0}^{3} c_{ij} u^i v^j$$

p is any of x, y and z.

Requires 48 coefficients (3 independent sets of 16) to determine a surface patch.

Designing parametric curves

- Introduce the types of curves
  - Interpolating
  - Hermite
  - Bézier
  - B-spline
- Analyze their performance

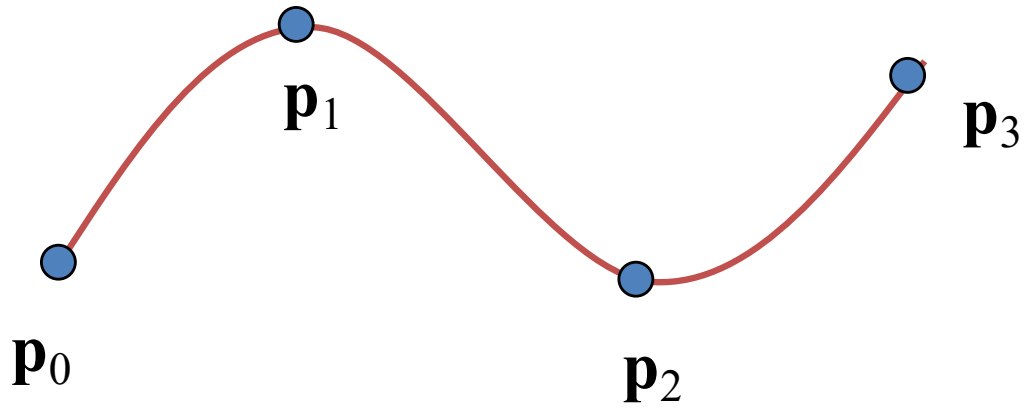
Matrix-Vector Form

$$\mathbf{p}(u) = \mathbf{c}_0 + \mathbf{c}_1 u + \mathbf{c}_2 u^2 + \mathbf{c}_3 u^3 = \sum_{k=0}^{3} \mathbf{c}_k u^k$$

define $\mathbf{u} = \begin{bmatrix} 1 \\ u \\ u^2 \\ u^3 \end{bmatrix}$, $\mathbf{c} = \begin{bmatrix} \mathbf{c}_0 \\ \mathbf{c}_1 \\ \mathbf{c}_2 \\ \mathbf{c}_3 \end{bmatrix}$, $\mathbf{c}_k = \begin{bmatrix} c_{kx} \\ c_{ky} \\ c_{kz} \end{bmatrix}$

then $$\mathbf{p}(u) = \mathbf{u}^T \mathbf{c} = \mathbf{c}^T \mathbf{u}$$

Interpolating Curve



$$\mathbf{p}_k = \begin{bmatrix} x_k \\ y_k \\ z_k \end{bmatrix}$$

Given four data (control) points $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$
determine cubic $\mathbf{p}(u)$ which passes through them

→ We must find $\mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3$

Interpolation Equations

First, we need to decide at which values of u the interpolation takes place. Lacking any other information, we can choose these values to be equally spaced. Since we have chosen u to vary over [0,1], we apply interpolating conditions at $u = 0, 1/3, 2/3, 1$

$$\mathbf{p}_0=\mathbf{p}(0)=\mathbf{c}_0$$
$$\mathbf{p}_1=\mathbf{p}(1/3)=\mathbf{c}_0+(1/3)\mathbf{c}_1+(1/3)^2\mathbf{c}_2+(1/3)^3\mathbf{c}_3$$
$$\mathbf{p}_2=\mathbf{p}(2/3)=\mathbf{c}_0+(2/3)\mathbf{c}_1+(2/3)^2\mathbf{c}_2+(2/3)^3\mathbf{c}_3$$
$$\mathbf{p}_3=\mathbf{p}(1)=\mathbf{c}_0+\mathbf{c}_1+\mathbf{c}_2+\mathbf{c}_3$$

or in matrix form with $\mathbf{p} = [\mathbf{p}_0\ \mathbf{p}_1\ \mathbf{p}_2\ \mathbf{p}_3]^T$ we have $\mathbf{p}=\mathbf{A}\mathbf{c}$ with

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & \left(\frac{1}{3}\right) & \left(\frac{1}{3}\right)^2 & \left(\frac{1}{3}\right)^3 \\ 1 & \left(\frac{2}{3}\right) & \left(\frac{2}{3}\right)^2 & \left(\frac{2}{3}\right)^3 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Interpolation Matrix

Solving for $\mathbf{c}$ we find the interpolation matrix

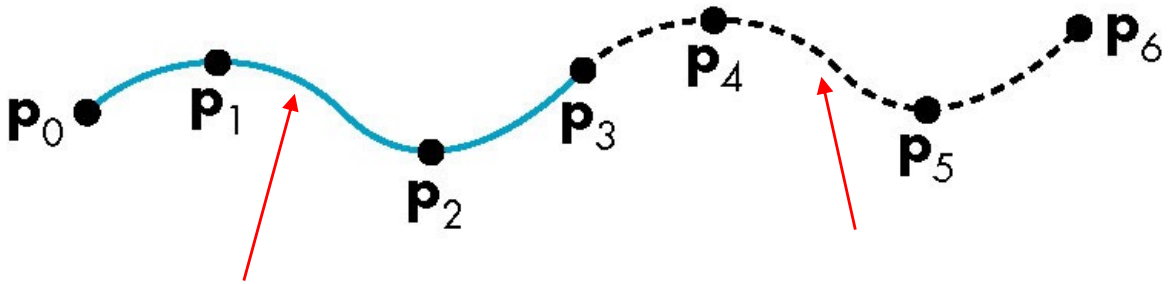
$$\mathbf{M}_I = \mathbf{A}^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -5.5 & 9 & -4.5 & 1 \\ 9 & -22.5 & 18 & -4.5 \\ -4.5 & 13.5 & -13.5 & 4.5 \end{bmatrix}$$

and we obtain the coefficients $\mathbf{c}$ as

$$\mathbf{c} = \mathbf{M}_I \mathbf{p}$$

Note that $\mathbf{M}_I$ does not depend on input data and can be used for each segment in x, y and z.

Interpolating Multiple Segments



use $\mathbf{p} = [\mathbf{p}_0\ \mathbf{p}_1\ \mathbf{p}_2\ \mathbf{p}_3]^T$

use $\mathbf{p} = [\mathbf{p}_3\ \mathbf{p}_4\ \mathbf{p}_5\ \mathbf{p}_6]^T$

We achieve continuity at join points but not continuity of derivatives

Blending Functions

We can rewrite the equation for $\mathbf{p}$ as

$$\mathbf{p}(u)=\mathbf{u}^{\mathrm{T}}\mathbf{c}=\mathbf{u}^{\mathrm{T}}\mathbf{M}_I\mathbf{p} = \mathbf{b}(u)^{\mathrm{T}}\mathbf{p}$$

where $\mathbf{b}(u) = [b_0(u)\ \ b_1(u)\ \ b_2(u)\ \ b_3(u)]^T = \mathbf{M}_I^T\mathbf{u}$ is a column matrix of four blending polynomials, such that

$$\mathbf{p}(u) \ = \ b_0(u)\mathbf{p}_0 + b_1(u)\mathbf{p}_1 + b_2(u)\mathbf{p}_2 + b_3(u)\mathbf{p}_3 = \sum_{i=0}^{3} b_i(u)\mathbf{p}_i$$

These blending functions for the cubic interpolating polynomial are

$$\mathrm{b}_0(u) = \text{-}4.5(u\text{-}1/3)(u\text{-}2/3)(u\text{-}1)$$

$$\mathrm{b}_1(u) = 13.5\mathrm{u}\ (u\text{-}2/3)(u\text{-}1)$$

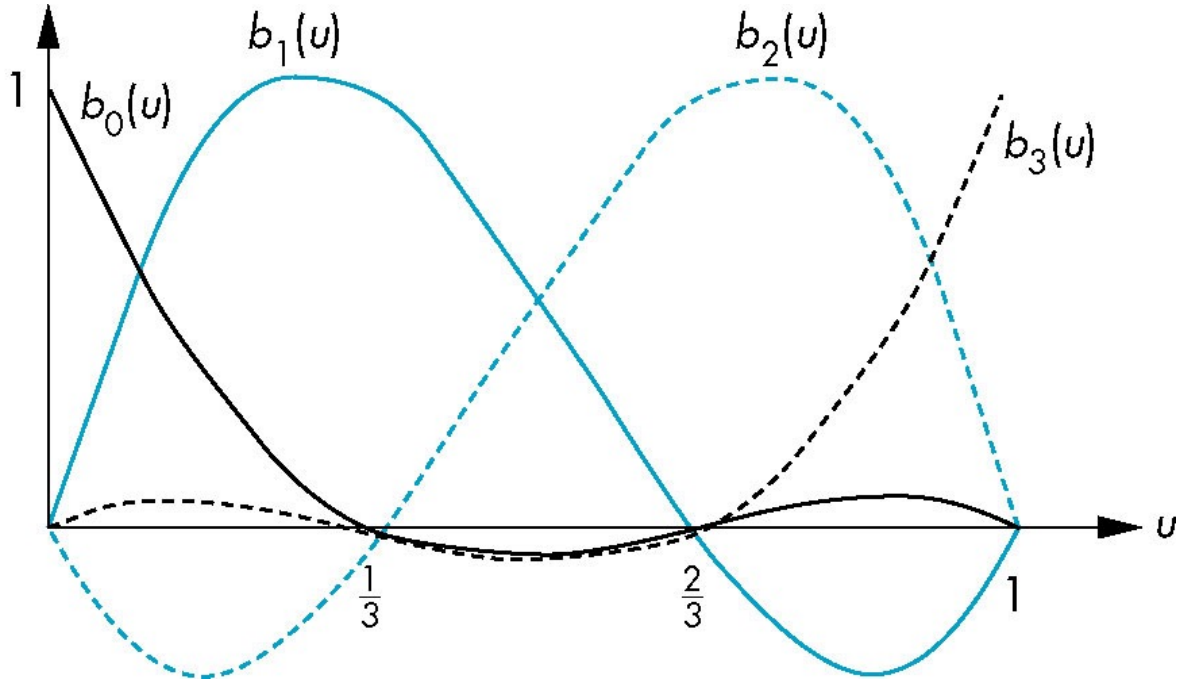
$$\mathrm{b}_2(u) = \text{-}13.5u\ (u\text{-}1/3)(u\text{-}1)$$

$$\mathrm{b}_3(u) = 4.5u\ (u\text{-}1/3)(u\text{-}2/3)$$

Blending Functions, contd.

These functions are not smooth

- Hence the interpolation polynomial is not smooth



The lack of smoothness is the consequence of the interpolating requirement that the curve must pass through the control points (rather than just come close to them)

Interpolating Patch

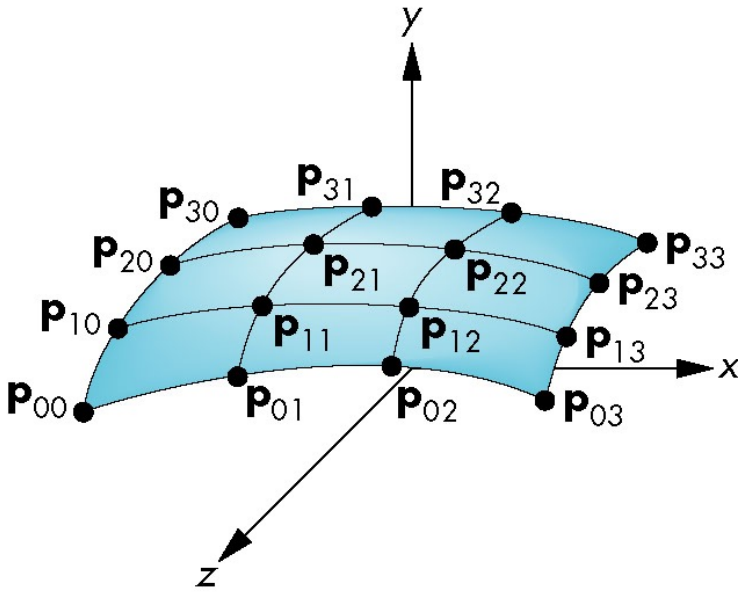
Interpolating patch is a natural extension of the interpolating curve.
A bicubic surface patch can be written as

$$\mathbf{p}(u,v) = \sum_{i=0}^{3} \sum_{j=0}^{3} u^i v^j \mathbf{c}_{ij}$$

Where $\mathbf{c}_{ij}$ is a three-element column matrix of the x, y and z coefficients for the ij-th term in the polynomial.

We need 16 conditions to determine the 16 coefficients $\mathbf{c}_{ij}$. Choose at $u,v = 0,\ 1/3,\ 2/3,\ 1$

A particular bicubic polynomial patch is defined by 48 elements of $\mathbf{C}$, i.e. by 16 three-element vectors



Matrix Form

Define: $\mathbf{v} = [1 \; v \; v^2 \; v^3]^T$, $\mathbf{C} = [\mathbf{c}_{ij}]$ $\mathbf{P} = [\mathbf{p}_{ij}]$

3D control points

We can rewrite the expression for $\mathbf{p}(u,v)$ as

$$\mathbf{p}(u,v) = \mathbf{u}^T\mathbf{C}\mathbf{v}$$

If we observe that for constant u (v), we obtain interpolating curve in v (u), we can show that

$$\mathbf{C} = \mathbf{M}_I\mathbf{P}\mathbf{M}_I^T$$

$$\mathbf{p}(\mathrm{u},\mathrm{v}) = \mathbf{u}^T\mathbf{M}_I\mathbf{P}\mathbf{M}_I^T\mathbf{v}$$

Blending Patches

$$\mathbf{p}(u,v) = \sum_{i=0}^{3} \sum_{j=0}^{3} b_i(u)\, b_j(v)\, \mathbf{p}_{ij}$$

Each $b_i(u)b_j(v)$ is a blending patch

We can build and analyze surfaces from our knowledge of curves

Surfaces formed from curves by this technique are known as tensor product surfaces. They are an example of separable surfaces, which can be written as

$$\mathbf{p}(u,v) = \mathbf{f}(u)\mathbf{g}(v)$$

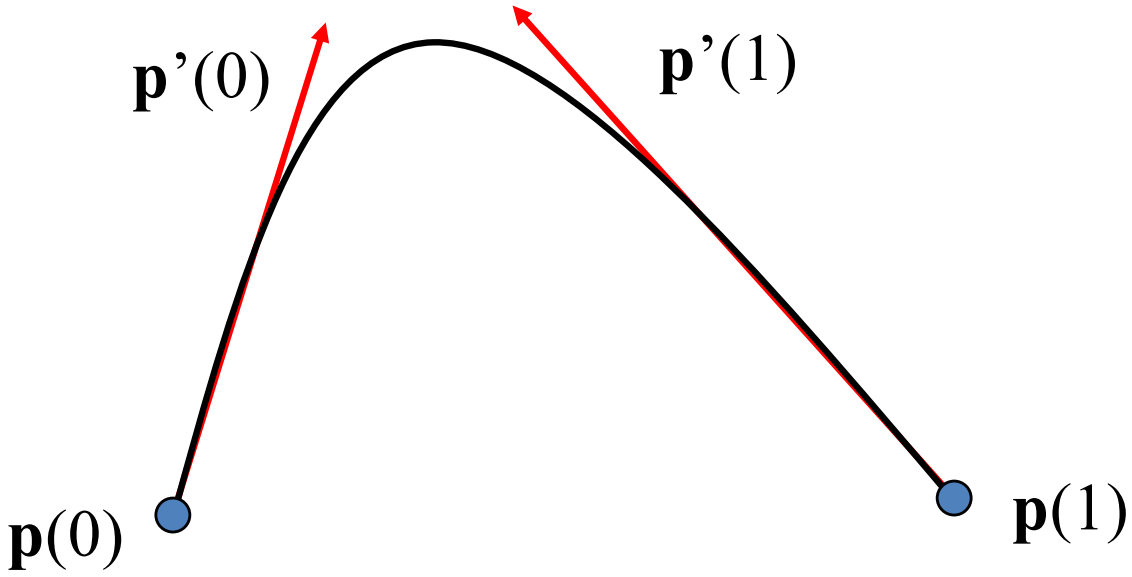
Where $\mathbf{f}$ and $\mathbf{g}$ are suitably chosen matrices. The advantage is that we work with functions of u and v independently.

Other Types of Curves and Surfaces

- How can we get around the limitations of the interpolating form
  - Lack of smoothness
  - Discontinuous derivatives at join points
- We have four conditions (for cubics) that we can apply to each segment
  - Use them other than for interpolation
  - Relax the “data fit” - sufficient to come close to the data (no need to really “go through” the data points)

Hermite form

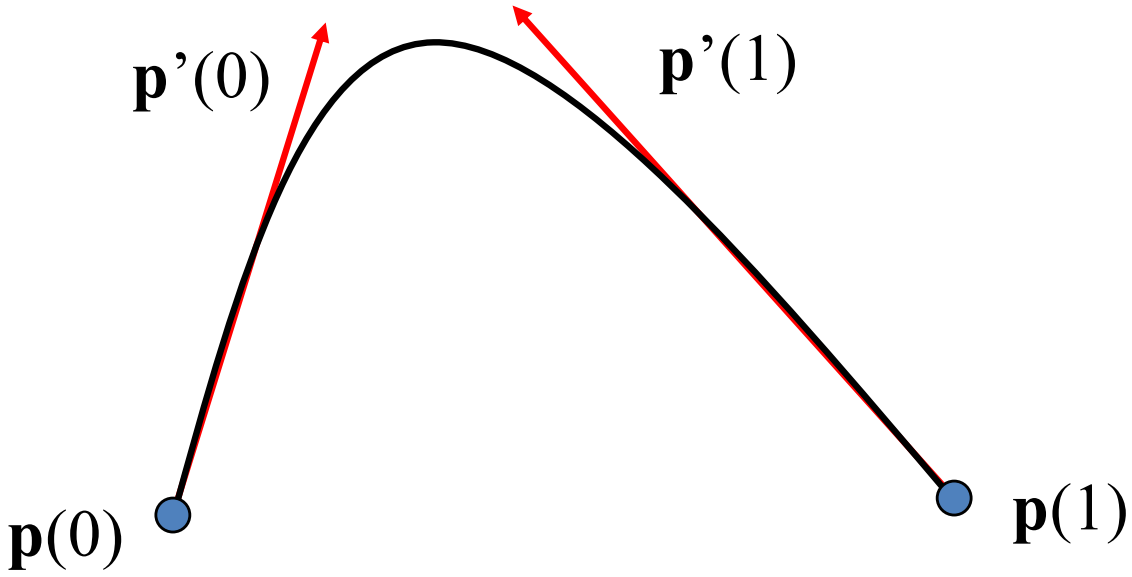
Hermite Form



The curve is forced to pass through only 2 control points (so, only 2 “interpolating” conditions)

Two extra conditions remain and these will be used to meet the requirement on the derivatives.

Hermite Form, contd.



So, use two interpolating conditions and two derivative conditions per segment

This ensures continuity and first derivative continuity between segments

Hermite Form, contd.

Interpolation conditions are the same as before at the end points:

$$\mathbf{p}_0=\mathbf{p}(0)=\mathbf{c}_0$$
$$\mathbf{p}_3=\mathbf{p}(1)=\mathbf{c}_0+\mathbf{c}_1+\mathbf{c}_2+\mathbf{c}_3$$

The derivative of $\mathbf{p}(u)=\mathbf{c}_0+\mathbf{c}_1u+\mathbf{c}_2u^2+\mathbf{c}_3u^3$ is simply

$$\mathbf{p}'(u)=\left[\frac{dx(u)}{du}\quad\frac{dy(u)}{du}\quad\frac{dz(u)}{du}\right]^{\mathrm{T}}=\mathbf{c}_1+2u\mathbf{c}_2+3u^2\mathbf{c}_3$$

Evaluating at end points yields two other conditions:

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

Hermite Form, contd.

$$\mathbf{q} = \begin{bmatrix} \mathbf{p}_0 \\ \mathbf{p}_3 \\ \mathbf{p}'_0 \\ \mathbf{p}'_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & 3 \end{bmatrix} \mathbf{c}$$

Solving, we find $\mathbf{c}=\mathbf{M}_H\mathbf{q}$ where $\mathbf{M}_H$ is the Hermite matrix

$$\mathbf{M}_H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -3 & 3 & -2 & -1 \\ 2 & -2 & 1 & 1 \end{bmatrix}$$

Blending Polynomials for the Hermite form

$$\mathbf{p}(u) = \mathbf{b}(u)^T \mathbf{q}$$

$$\mathbf{b}(u) = \begin{bmatrix} 2u^3 - 3u^2 + 1 \\ -2u^3 + 3u^2 \\ u^3 - 2u^2 + u \\ u^3 - u^2 \end{bmatrix}$$

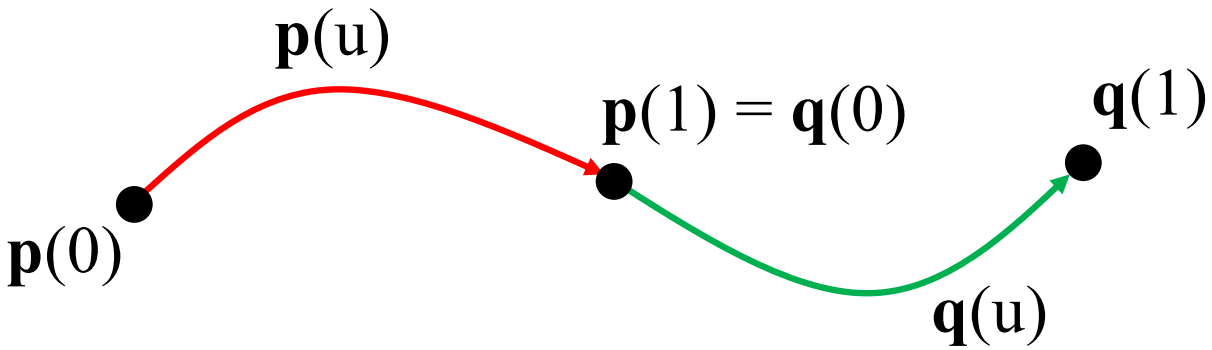
Although these functions are smooth, the Hermite form is not used directly in Computer Graphics and CAD because we usually have control points but not derivatives

However, the Hermite form is the basis of the Bezier form

Parametric and Geometric Continuity

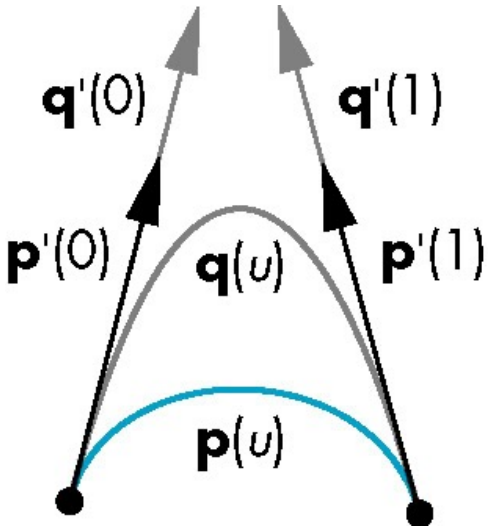
- We can require the derivatives of x, y and z to each be continuous at join points - parametric continuity
- Alternately, we can only require that the tangents of the resulting curve be continuous - geometry continuity
- The latter gives more flexibility as we have to satisfy only two conditions instead of three at each join point

Parametric and Geometric Continuity



- We can enforce various continuity conditions by matching polynomials and their derivatives at $\mathbf{p}(1)$ with $\mathbf{q}(0)$.
- Parametric continuity - requires that the derivatives of x, y and z are continuous at join points $\mathbf{p}'(1) = \mathbf{q}'(0)$.
- Geometric continuity - requires only that the tangents of the resulting curve are continuous $\mathbf{p}'(1) = \alpha\mathbf{q}'(0)$, $\alpha > 0$ (derivatives proportional)
- The latter gives more flexibility as we have to satisfy only two conditions instead of three at each join point

Example



- Here the $\mathbf{p}$ and $\mathbf{q}$ have the same tangents at the ends of the segment but different derivatives
- Generate different Hermite curves
- This techniques is used in drawing applications

Higher Dimensional Approximations

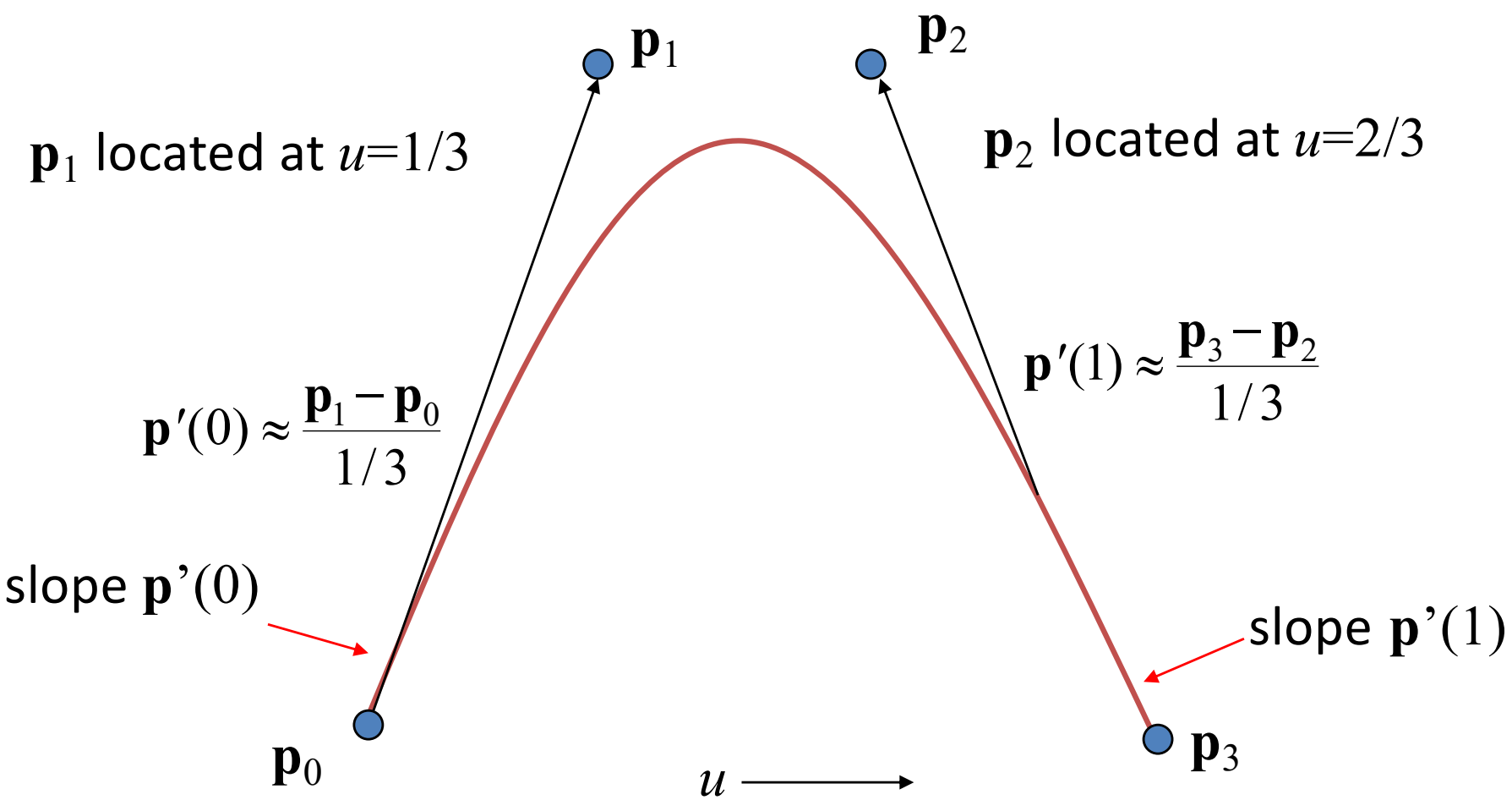
- The techniques for both interpolating and Hermite curves can be used with higher dimensional parametric polynomials
- For interpolating form, the resulting matrix becomes increasingly more ill-conditioned and the resulting curves less smooth and more prone to numerical errors
- In both cases, there is more work in rendering the resulting polynomial curves and surfaces

Bézier representation

Bézier's Idea

- In graphics and CAD, we do not usually have derivative data
- Bezier suggested using the same 4 data points as with the cubic interpolating curve to approximate the derivatives in the Hermite form

Approximating Derivatives



Bézier form

Interpolating conditions are the same as for the Hermite form

$$\mathbf{p}_0=\mathbf{p}(0)=\mathbf{c}_0$$
$$\mathbf{p}_3=\mathbf{p}(1)=\mathbf{c}_0+\mathbf{c}_1+\mathbf{c}_2+\mathbf{c}_3$$

Approximating derivative conditions

$$\mathbf{p'}_0= 3(\mathbf{p}_1-\mathbf{p}_0)=\mathbf{c}_1$$
$$\mathbf{p'}_3= 3(\mathbf{p}_3-\mathbf{p}_2) =\mathbf{c}_1+2\mathbf{c}_2+3\mathbf{c}_3$$

Solve four linear equations for $\mathbf{c}=\mathbf{M}_B\mathbf{p}$

Bézier Matrix

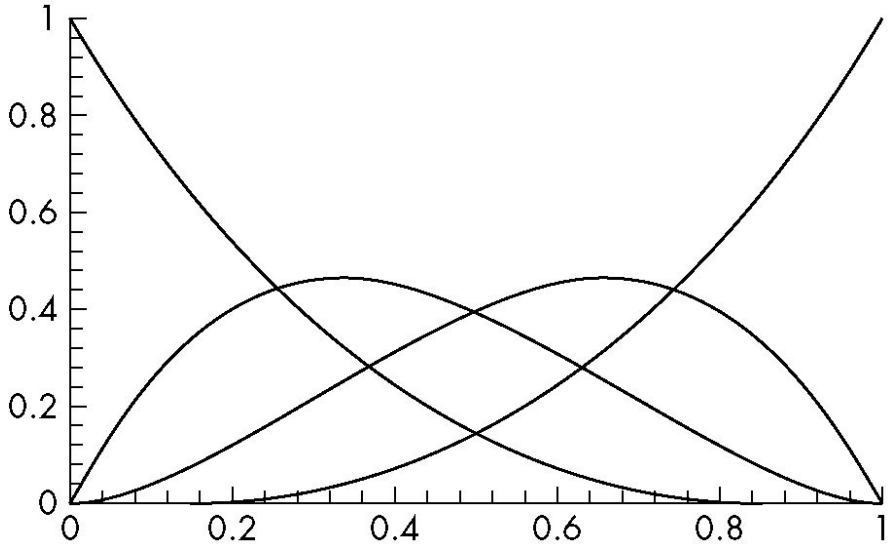
$$\mathbf{M}_B = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix}$$

$$\mathbf{p}(u) = \mathbf{u}^{\mathrm{T}}\mathbf{M}_B\mathbf{p} = \mathbf{b}(u)^{\mathrm{T}}\mathbf{p}$$

blending functions

Blending Functions

$$\mathbf{b}(u) = \begin{bmatrix} (1-u)^3 \\ 3u(1-u)^2 \\ 3u^2(1-u) \\ u^3 \end{bmatrix}$$



Note that all zeros are at 0 and 1 which forces the functions to be smooth over (0,1)

Bernstein Polynomials

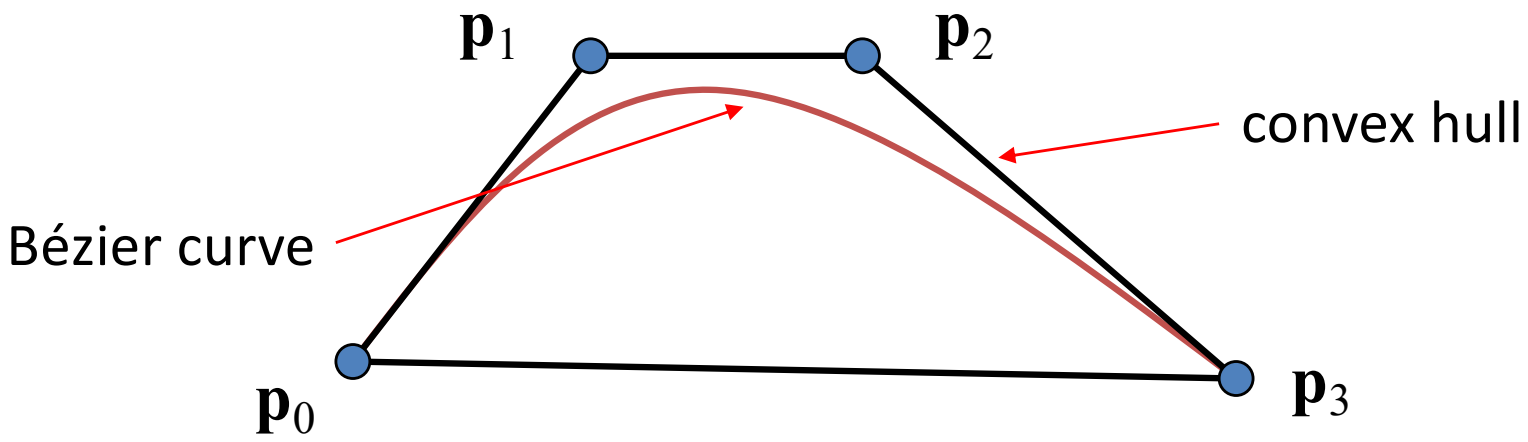
- The blending functions are a special case of the Bernstein polynomials

$$b_{kd}(u) = \frac{d!}{k!(d-k)!} u^k (1-u)^{d-k}$$

- These polynomials give the blending polynomials for any degree Bezier form
  - All zeros at 0 and 1
  - For any degree they all sum to 1
  - They are all between 0 and 1 inside (0,1)

Convex Hull Property

- The properties of the Bernstein polynomials ensure that all Bezier curves lie in the convex hull of their control points
- Hence, even though we do not interpolate all the data, we cannot be too far away

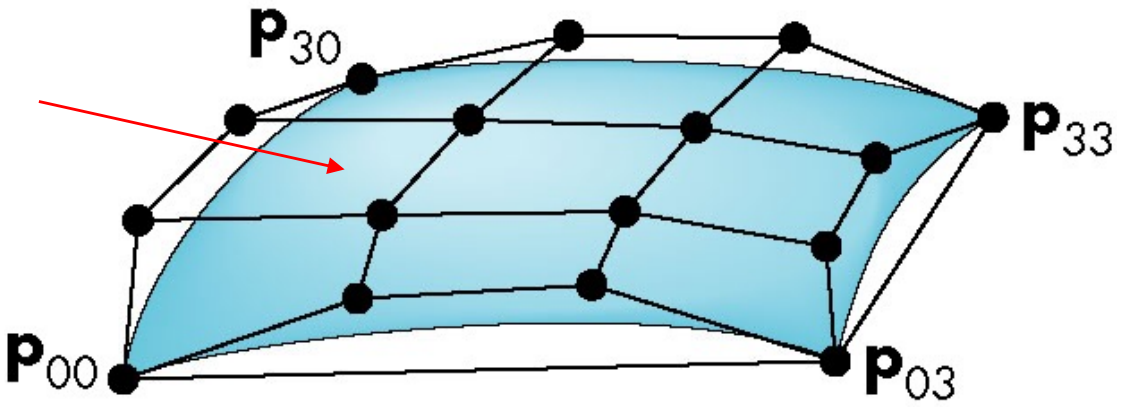


Bézier Patches

Using same data array $\mathbf{P}=[\mathbf{p}_{ij}]$ as with interpolating form

$$\mathbf{p}(u,v)=\sum_{i=0}^{3}\sum_{j=0}^{3}b_i(u)\,b_j(v)\,\mathbf{p}_{ij}=u^T\,\mathbf{M}_B\,\mathbf{P}\,\mathbf{M}_B^T\,v$$

Patch lies in convex hull



Analysis

- Although the Bezier form is much better than the interpolating form, the derivatives are not continuous at join points
- Can we do better?
  - Go to higher order Bezier
    - More work
    - Derivative continuity still only approximate
    - Supported by OpenGL
  - Apply different conditions
    - Tricky without letting order increase

Summary

- Representation of curves and surfaces
- Polynomial forms
- Hermite curves
- Bézier curves

For more details, see

E. Angel and D. Shreiner: *Interactive Computer* Graphics – *A Top Down Approach with Shader-Based OpenGL* (6th ed). Chapter 10: Curves and Surfaces