# Curves and Surface I

Angel Ch.10

# Representation of Curves and Surfaces

Piece-wise linear representation is inefficient
- line segments to approximate curve
- polygon mesh to approximate surfaces
- can not approximate general curves exactly
- only continous in position (not derivatives)

Modelling of smooth curves and surface
- ‘smooth’ continuous in position and derivatives (1st/2nd…)
- exact representation of non-planar objects

Representation of curves and surfaces:

(1) Explicit: y=g(x)
(2) Implicit: f(x,y)=0
(3) Parametric: x=x(u), y=y(u)

Parametric forms commonly used in computer graphics/CAD

# Explicit Representation of Curves

## Curves in 2D

y=g(x)     is a general curve in x,y space
x independent variable
y dependent variable

Inverse relation: x=h(y)     - inverse may not exist

**Example:** 2D Line   y=ax+b          a-slope
b-intersection with y-axis
Problem with representing a vertical line a=infinity

**Example:** 2D Circle radius r

$$y = \sqrt{r^2 - x^2}$$

- explicit equation only represent half the circle

## Explicit Curves in 3D

The 3D explicit representation of a curve requires 2 equations

y=g(x)

z=f(x) - y,z are both dependent variables

Example: Line in 3D

y=ax+b

z=cx+d

- cannot represent a line in a plane of x=const

## Explicit representation of a surface

z=f(x,y)

- 2 independent variables x,y
- z dependent variable

Cannot represent a full sphere

$$z = \sqrt{r^2 - x^2 - y^2}$$

Explicit representations are co-ordinate system dependent

- can not represent all curves such as lines/circles
- Curves exist independent of any representation
- failures causes serious problems in graphics/CAD

# Implicit Representation

**2D Curves:** f(x,y)=0

- f() is a 'membership' testing function
  (x,y) is on the curve if f(x,y)=0
- In general no analytic way to find points on the curve
  ie to evaluate y value corresponding to given x
- overcomes some of coordinate system dependent problems
- represents all lines and circles

**Example:** 2D Line ax+by+c=0

- represent vertical line a=-1, b=0

**Example:** 2D Circle $x^2 + y^2 - r^2 = 0$

- represents the entire circle

**3D Surfaces:** f(x,y,z)=0

- membership test is (x,y,z) on the curve or surface
- general representation in 3-space
- represents any plane or line in 3-space

**Plane:** ax+by+cz+d=0

- slopes a,b,c wrt x,y,z-axis

**Sphere of radius r:** $x^2 + y^2 + z^2 - r^2 = 0$

**3D Curves:**

- not easily represented in 3D
- intersection of two implicit surfaces

  f(x,y,z)=0

  g(x,y,z)=0

  point (x,y,z) on curve must be a member of both function
- line represented by intersection of 2 planes

Algebraic surfaces

- class of implicit surfaces where f(x,y,z) is a polynomial function of the three variables

$$d + \sum_{i=1}^{n} (a_i x^i + b_i y^i + c_i z^i) = 0$$

is an $n^{th}$ order polynomial

- quadric surfaces n=2 are of particular importance as they represent several common surfaces sphere, cone, torus, ellipsoid

Quadrics have the property that intersection with a line gives at most 2 intersection points (used for rendering)

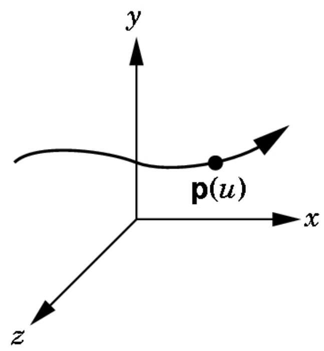
# Parametric Representation

Parametric curve in 3-space is a function of 1-parameter, u

x=x(u)

y=y(u)

z=z(u)




- general representation
- same in 2-space and 3-space
- u is independent variable and x,y,z dependent

Curve can be written as the set (locus) of points:

p(u) = [x(u) y(u) z(u)]

Derivative of curve:

$$\frac{dp(u)}{du} = \begin{bmatrix} \frac{dx(u)}{du} \\ \frac{dy(u)}{du} \\ \frac{dz(u)}{du} \end{bmatrix}$$

- partial derivative wrt each function
- points in the direction tangent to the curve
- magnitude is rate of change wrt to u (velocity)

**Example:** 3D Line

$$p(\alpha) = \alpha p_1 + (1-\alpha) p_2$$

$$= \alpha \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} + (1-\alpha) \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix}$$

$$= \begin{bmatrix} x_2 + \alpha(x_1 - x_2) \\ y_2 + \alpha(y_1 - y_2) \\ z_2 + \alpha(z_1 - z_2) \end{bmatrix}$$

Derivatives:

$$\frac{dp(\alpha)}{d\alpha} = p_\alpha(\alpha) = \begin{bmatrix} (x_1 - x_2) \\ (y_1 - y_2) \\ (z_1 - z_2) \end{bmatrix}$$

**Parametric surface:**

x=x(u,v)

y=y(u,v)

z=z(u,v)

- 2 independent parameters u,v

$$p(u,v) = [\ x(u,v) \quad y(u,v) \quad z(u,v)]^T$$

Partial Derivatives

$$\frac{\partial p(u)}{\partial u} = p_u = \begin{bmatrix} \dfrac{\partial x(u,v)}{\partial u} \\ \dfrac{\partial y(u,v)}{\partial u} \\ \dfrac{\partial z(u,v)}{\partial u} \end{bmatrix} = \begin{bmatrix} x_u(u,v) \\ y_u(u,v) \\ z_u(u,v) \end{bmatrix}$$

$$\frac{\partial p(v)}{d(v)} = p_v$$

- Partial derivatives represent the tangent plane

**Parametric representation**

- widely used in computer graphics for curves and surfaces and CAD
- general representation of many forms.
- simple computation of derivatives
- requires a parameterization (u,v) of the surface ie underlying 2D coordinate system for a 3D surface
- representation depends on 2D coordinate system
- polynomial parametric forms used to represent a wide variety of surfaces.

How can we define the parametric form of a surface?

# Parametric Polynomial Curves

A parametric curve in 3 - space is represented as : $p(u) = \begin{bmatrix} x(u) \\ y(u) \\ z(u) \end{bmatrix}$

A polynomial parametric curve of degree n is given by :

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

where $c_k$ is a column matrix with independent coefficients in x, y, z

$$c_k = \begin{bmatrix} c_{xk} \\ c_{yk} \\ c_{zk} \end{bmatrix}$$

The parametric curve has 3(n + 1) degrees of freedom,

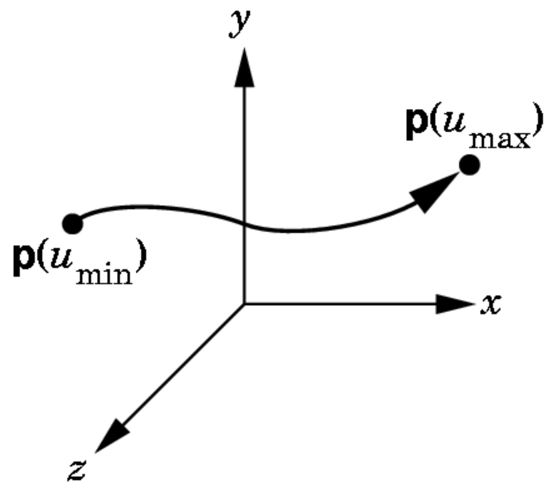
giving 3 independent sets of equations :

$$x(u) = \sum_{k=0}^{n} u^k c_{xk} \quad y(u) = \sum_{k=0}^{n} u^k c_{yk} \quad z(u) = \sum_{k=0}^{n} u^k c_{zk}$$

We can define the curves x(u), y(u), z(u) over a range of u

$$u_{\min} \leq u \leq u_{\max}$$

- p(u) over this range gives a curve segment



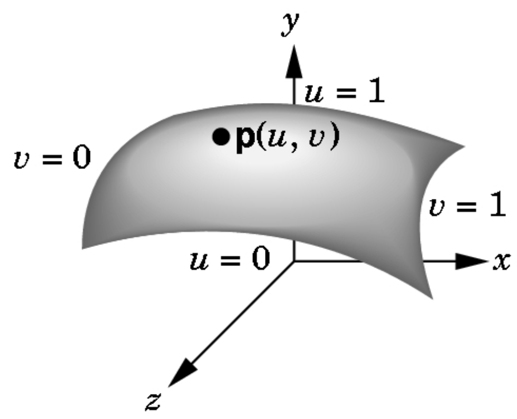

Without loss of generality we can define the curve segment:

$$0 \leq u \leq 1$$

# Parametric Polynomial Surfaces

Define a surface by n and m order polynomials in u and v

$$p(u,v) = \begin{bmatrix} x(u,v) \\ y(u,v) \\ z(u,v) \end{bmatrix} = \sum_{k=0}^{n} \sum_{j=0}^{m} u^k v^j c_{jk}$$




Surface has 3(n+1)(m+1) degrees of freedom

Usually, n=m and we parameterize u,v over the range 0,1 giving a surface patch.

The surface can be viewed in the limit as a collection of curves generated by holding u or v constant

Partial Derivatives

$$p_u(u,v) = \begin{bmatrix} x_u(u,v) \\ y_u(u,v) \\ z_u(u,v) \end{bmatrix} = \sum_{k=0}^{n}\sum_{j=0}^{m} k u^{k-1} v^{j} c_{jk}$$

$$p_v(u,v) = \begin{bmatrix} x_v(u,v) \\ y_v(u,v) \\ z_v(u,v) \end{bmatrix} = \sum_{k=0}^{n}\sum_{j=0}^{m} j u^{k} v^{j-1} c_{jk}$$

2nd Order Partial Derivatives

$$p_{uu}(u,v) = \begin{bmatrix} x_u(u,v) \\ y_u(u,v) \\ z_u(u,v) \end{bmatrix} = \sum_{k=0}^{n}\sum_{j=0}^{m} k(k-1) u^{k-2} v^{j} c_{jk}$$

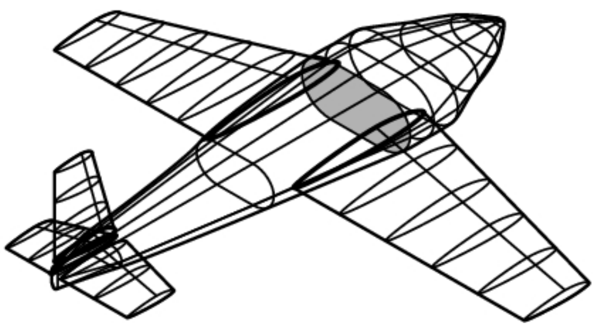
$$p_{uv}(u,v) = p_{vu}(u,v) = \begin{bmatrix} x_v(u,v) \\ y_v(u,v) \\ z_v(u,v) \end{bmatrix} = \sum_{k=0}^{n}\sum_{j=0}^{m} kj u^{k-1} v^{j-1} c_{jk}$$

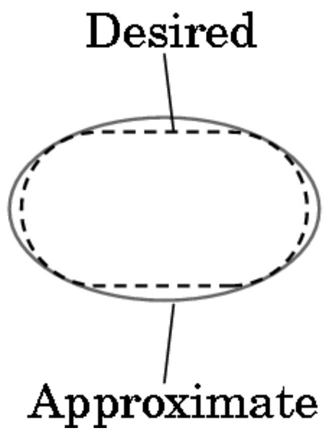
# Design Criteria

Representations of curves and surfaces in computer graphics and CAD should satisfy the following:

- Local control of shape
- Smoothness and continuity
- Direct evaluation of derivatives
- Stability
  (small change in parameters gives a small change in shape)
- Fast Rendering

Use representations based on low-order parametric polynomials to satisfy the above

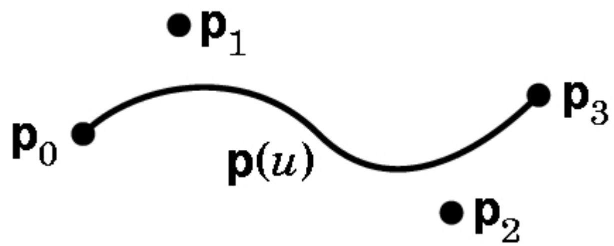




# Local Control of Shape

- ideally control local shape with a set of local parameters rather than global change in shape
- easier to design desired shape

Represent curve shape p(u) by a set of 'control points':




- local curve shape is based on the position of the control points

Interpolating curve: passes through all control point

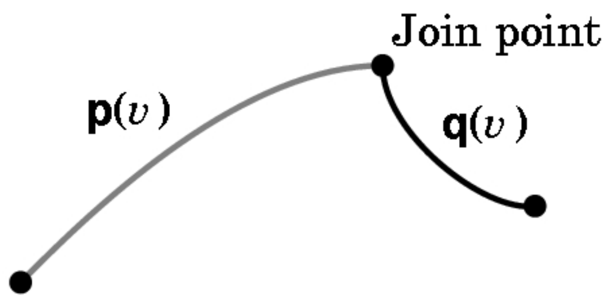
# Smoothness and Continuity

'Smoothness' of a curve refers to the continuity of derivatives of the curve

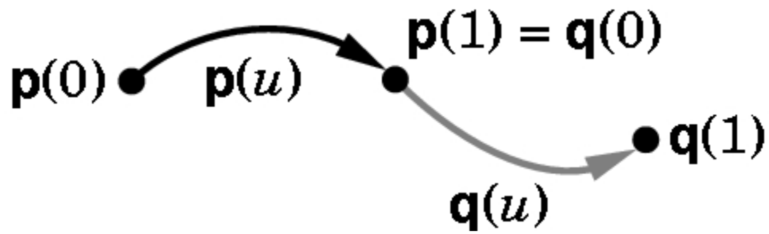
For a parametric curve all derivatives exist and are continuous

$$p(u) = \sum_{k=0}^{n} c_k u^k \qquad p_u(u) = \sum_{k=1}^{n} k c_k u^{k-1} \qquad p_{uu}(u) = \sum_{k=2}^{n} k(k-1) c_k u^{k-2} \qquad .....$$

For two curves meeting at a join point the smoothness is defined by the highest derivative which is continuous




discontinuity in 1st derivative




continuous in 1st derivative

Parametric Continuity $C^n$

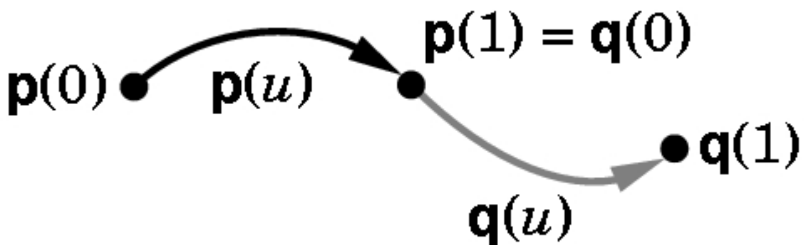
- highest order derivative with respect to parameters which is continous

$C^0 : p(1) = q(0)$

$C^1 : p_u(1) = q_u(0)$

$C^2 : p_{uu}(1) = q_{uu}(0)$

Derivatives are continous in parameter space u and consequently the curve is continous in 3 - space



Geometric Continuity $G^n$

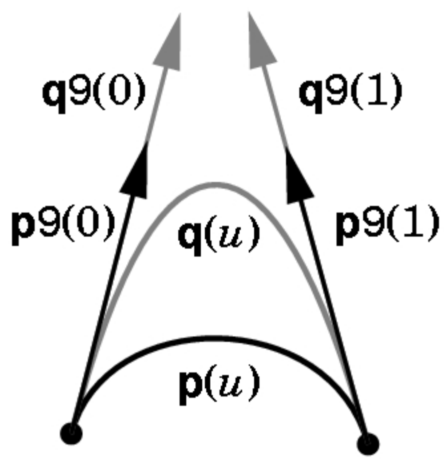
- derivatives in 3 - space are in the same direction but different magnitude

$G^1 : p_u(1) \propto q_u(0)$

- tangents to curve are in the same direction
- requires only 2 constraints on derivates rather than 3 use remaining free parameter to control shape

$G^2 : p_{uu}(1) \propto q_{uu}(0)$

- curvature is in the same direction



# Parametric Cubic Polynomial Curves

What is the correct order of polynomial curve

- low-order: less flexible
- high-order: costly, bumpy (too many degrees of freedom)

Designing curve locally by a set of control points we can use a low-order curve to approximate complex shapes.

Cubic polynomial curves are widely used:

$$p(u) = c_0 + c_1 u + c_2 u^2 + c_3 u^3 = \sum_{k=0}^{3} c_k u^k = u^T c$$

$$c = \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} \qquad u = \begin{bmatrix} 1 \\ u \\ u^2 \\ u^3 \end{bmatrix} \qquad c_k = \begin{bmatrix} c_{kx} \\ c_{ky} \\ c_{kz} \end{bmatrix}$$

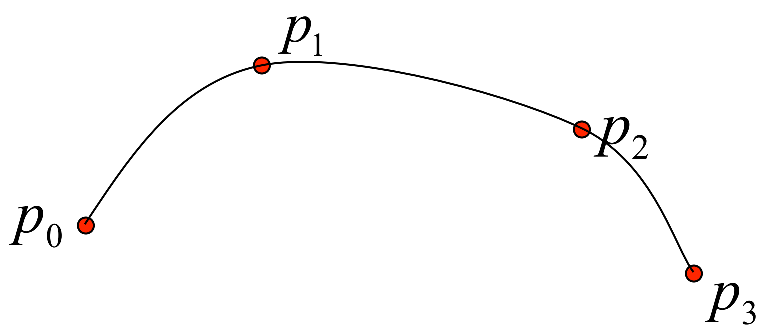
12 equations in 12 unknowns c

Want methods of deriving parameters c for a desired curve!

# Cubic Polynomial Interpolation

Given a set of 4 - points (ie 12dof) derive curve that interpolates between them and exactly passes through them:

$$p_i = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} \quad i = 0...3$$




What are the coefficients c such that the curve $p(u) = u^T c$ interpolates the points $p_i$

Let the points be at equally spaced intervals along the curve $u = 0, \frac{1}{3}, \frac{2}{3}, 1$

This gives the four conditions:

$$p_0 = p(0) = c_0$$

$$p_1 = p(\tfrac{1}{3}) = c_0 + \tfrac{1}{3} c_1 + \left(\tfrac{1}{3}\right)^2 c_2 + \left(\tfrac{1}{3}\right)^3 c_3$$

$$p_2 = p(\tfrac{2}{3}) = c_0 + \tfrac{2}{3} c_1 + \left(\tfrac{2}{3}\right)^2 c_2 + \left(\tfrac{2}{3}\right)^3 c_3$$

$$p_3 = p(1) = c_0 + c_1 + c_2 + c_3$$

In matrix form : $p = Ac$

$$p = \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix} = \begin{bmatrix} x_0 & y_0 & z_0 \\ x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{bmatrix} \qquad A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & \frac{1}{3} & \left(\frac{1}{3}\right)^2 & \left(\frac{1}{3}\right)^3 \\ 1 & \frac{2}{3} & \left(\frac{2}{3}\right)^2 & \left(\frac{2}{3}\right)^3 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$p, c$ are 4x3 element matricies

We know control points p, want unknown curve parameters c

(12 equations in 12 unknowns) :

$$c = A^{-1} p = M_l p$$

$A -$ is non - sigular and can be inverted to obtain the 'interpolating matrix' $M_l$

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

$M_l$ - defines the curve that interpolates a given set of 4 control points

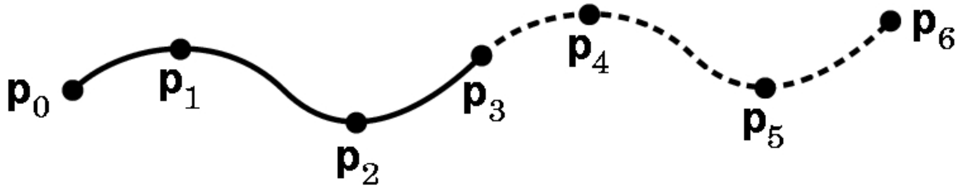
Given a sequence of control points $p_0, p_1....p_m$

we can define a set of cubic interpolating curves each defined by a group of four control points :

$p_0.....p_3$

$p_3.....p_6$

$p_6.....p_9$




......

If we let u = [0,1] for each segment then each segment has same $M_l$

Resulting curve has $C^3$ continuity for each segment but only $C^0$ between segments

# Blending Functions

We can consider the interpolation as the combination of control points $p_i$ according to a set of blending functions $b_i(u)$

$$p(u) = u^T c = u^T M_I p = b(u)^T p$$

$$b(u) = M_I^T u = \begin{bmatrix} b_0(u) \\ b_1(u) \\ b_2(u) \\ b_3(u) \end{bmatrix}$$

Each blending function $b_i(u)$ is a polynomial in u which weights or 'blends' together the individual contributions of each point :

$$p(u) = b_0(\mathrm{u})\,\mathrm{p}_0 + b_1(\mathrm{u})\,\mathrm{p}_1 + b_2(\mathrm{u})\,\mathrm{p}_2 + b_2(\mathrm{u})\,\mathrm{p}_2 = \sum_{i=0}^{3} b_i(u)\,p_i$$

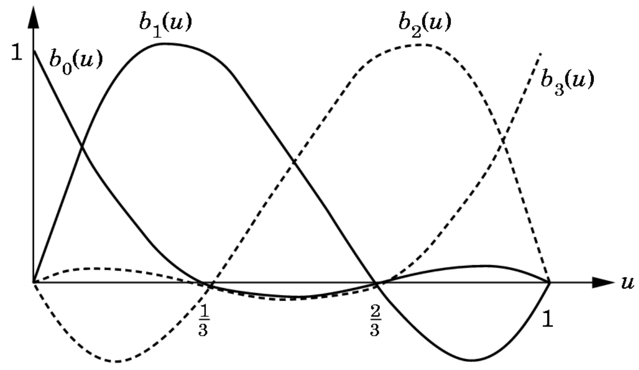
Blending functions for cubic interpolation :

$$b_0(u) = -\tfrac{9}{2}(u-\tfrac{1}{3})(u-\tfrac{2}{3})(u-1)$$

$$b_1(u) = \tfrac{27}{2}u(u-\tfrac{2}{3})(u-1)$$

$$b_2(u) = -\tfrac{27}{2}u(u-\tfrac{1}{3})(u-1)$$

$$b_3(u) = \tfrac{9}{2}u(u-\tfrac{1}{3})(u-\tfrac{2}{3})$$




Note : Blending function are symmetric about $u = \frac{1}{2}$

All zeros of the blending functions are in the interval [0,1]

- blending functions must vary substantially over [0,1] ie rapid changes
- results from requirement that curve pass through all control points

This results in limited usefulness of 'interpolating' cubic polynomial

+ polynomial is discontinuous between sections

Apply same derivation process to polynomials with other constraints

# Cubic Interpolation Patch

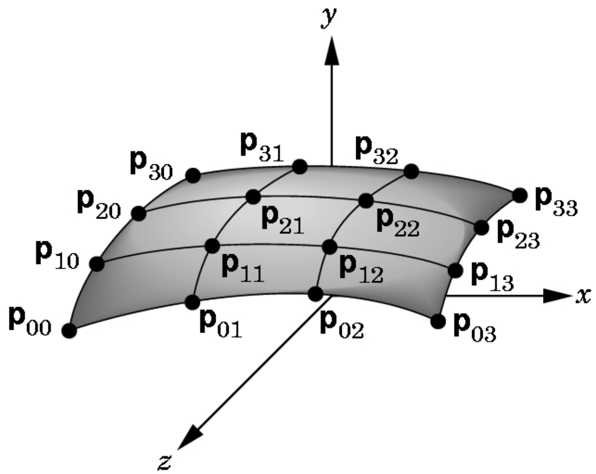
## Extension of interpolating curve to a surface

Bicubic interpolating surface patch is defined by 16 points :

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

$$C = \begin{bmatrix} c_{00} & c_{01} & c_{02} & c_{03} \\ c_{10} & c_{11} & c_{12} & c_{13} \\ c_{20} & c_{21} & c_{22} & c_{23} \\ c_{30} & c_{31} & c_{32} & c_{33} \end{bmatrix} \qquad u = \begin{bmatrix} 1 \\ u \\ u^2 \\ u^3 \end{bmatrix} \quad v = \begin{bmatrix} 1 \\ v \\ v^2 \\ v^3 \end{bmatrix}$$

$$c_{ij} = \begin{bmatrix} c_{xij} & c_{yij} & c_{zij} \end{bmatrix}$$




C - has 48 coefficients which we want to find to evaluate the surface
which interpolates a given set of points

As in the case of a curve let the control points

$p_{ij} = \text{p(u, v) for u, v} = 0, \frac{1}{3}, \frac{2}{3}, 1$

Applying these equations for the individual control points (as in the curve case) we obtain 16 equations relating the known points $p_{ij}$ to unknown coefficients $c_{ij}$

Consider the curve $v = 0$ (for a curve $p(u) = u^T M_l p$)

$$p(u,0) = u^T M_l \begin{bmatrix} p_{00} \\ p_{10} \\ p_{20} \\ p_{30} \end{bmatrix} = u^T C \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

This is the same as we had for the coefficients of a curve defined by 4 points

Likewise, we obtain similar expressions for $v = \frac{1}{3}, \frac{2}{3}, 1$

Putting the curves together in a single equation we obtain :

$$u^T M_l P = u^T C A^T = u^T C M_l^{-T}$$

$M_l^{-T}$ is the transpose of the inverse of $M_l$

Solve this for the unknown coefficients

$$C = M_l P M_l^T$$

$P$ is the matrix of 16 points

Substituting this into the equation for a surface gives

$$p(u,v) = u^T M_l P M_l^T v$$

This defines the bicubic surface patch which interpolates the 16 control points

The bicubic surface patch can be interpreted as the interpolation of a set of curves in u corresponding to each value of v

Alternatively, can consider the surface as the interpolation of a set of blending function in u and v

$$b_i(u) = M_l^T u$$

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

Each term $b_i(u)b_j(v)$ describes a blending patch for point $p_{ij}$

Surface is formed by blending together patches for 16 points

- interpolating functions are separable 'separable surface'

As for bicubic interpolating curves patches are not smooth (all zeros in [0,1])

Example of a 'tensor product' surface - surface formed by curves

# Hermite Curves and Surfaces

Rather than interpolating points we interpolate between endpoints + tangents at end points

- ensures continuity between curve/surface segments

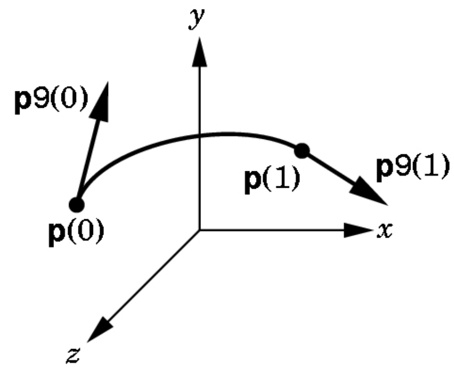
Hermite Form of a Curve define constraints as :

Curve intersects end - points

$$p(0) = p_0 = c_0 \qquad p(1) = p_3 = c_0 + c_1 + c_2 + c_3$$

Constrain the tangent at the end - points

$$p_u(0) = c_1 \qquad p_u(1) = c_1 + 2c_2 + 3c_3$$




In matrix form :

$$q = \begin{bmatrix} \mathrm{p}(0) \\ \mathrm{p}(1) \\ \mathrm{p_u}(0) \\ \mathrm{p_u}(1) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & 3 \end{bmatrix} c$$

Solve equations to find:

$c = M_H q$ Gives 'Hermite geometry' matrix $M_H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -3 & 3 & -2 & 1 \\ 2 & -2 & 1 & 1 \end{bmatrix}$

Resulting polynomial is given by:

$$p(u) = u^T M_H q$$

This can be represented as a set of blending functions on the points:

$$p(u) = b(u)^T q$$

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

The four blending functions have none of their zero's in [0,1]

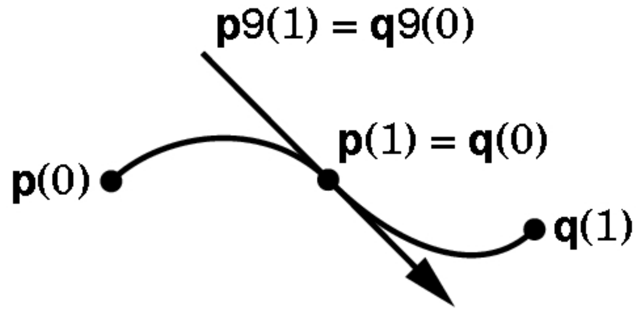
- smoother than interpolating blending function

Hermite polynomials can be used to represent a curve with continuous derivates
- such that the end point of one curve has the same derivative as the start point of the adjacent curve

$$p(1) = q(0)$$

$$p_u(1) = q_u(0)$$

where p(u) and q(u) are adjacent section of the curve with u = [0,1] for both giving a $C^1$ continous curve




This overcomes the problem with interpolating cubics where the end-points were only continuous in position

Hermite surface patch :

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

is defined to interpolate the 4 corner points and their derivatives

At corner (0,0) we define :

$$p(0,0) = c_{00} \quad p_u(0,0) = c_{01} \quad p_v(0,0) = c_{01} \quad p_{uv}(0,0) = c_{11}$$

Solving gives a surface patch which is continous in position and 1st derivative between adjacent patches.

Therefore, Hermite surface patch has advantages over the direct interpolation.

Derivatives can be defined from the input control points

ie $p_u = p_{00} - p_{01}$

## Summary

Surface representation

- explicit
- implicit
- parametric

parametric forms are widely used in computer graphics

Parametric forms

- Interpolating curves and surfaces

Next lecture: other parametric forms of surfaces
Hermite, Bezier, B-Spline, NURBS