III	Algebraic and geometric forms, tangents and normal, blending functions, reparametrization, straight lines, conics, cubic splines, Bezier curves and B-spline curves.	4	15%
	Plane surface, ruled surface, surface of revolution, tabulated cylinder, bi-cubic surface, bezier surface, B-spline surfaces and their modeling techniques.	3	
IV	Solid models and representation scheme, boundary representation, constructive solid geometry.	3	15%
	Sweep representation, cell decomposition, spatial occupancy enumeration, coordinate systems for solid modeling.	4	

4.1 Mathematical Models for Curves and Surfaces

We all have an intuitive understanding of curves and surfaces. But can we answer mathematically these basic questions: What is a curve? What is a surface? It turns out that there are several acceptable answers, and that different branches of mathematics use different definitions. We first introduce some of the fundamental notions through the simplest possible examples: the straight line, which is a special case of a curve, and the plane, which is a special case of a surface.

A straight line, or simply a *line*, in Euclidean space is a set of points $\mathbf{p}$ that satisfy

$$\mathbf{p} - \mathbf{p}_0 = u(\mathbf{p}_1 - \mathbf{p}_0), \quad u \in (-\infty, +\infty), \quad \mathbf{p}_0 \neq \mathbf{p}_1.$$

Here $\mathbf{p}_0$ and $\mathbf{p}_1$ are arbitrary but distinct points of the line. The equation above contains a parameter u and is called a *parametric equation*. As the parameter u takes all possible values from minus infinity to plus infinity, the point $\mathbf{p}$ traces the entire line.

The parametric equation of the line can be written in a different format. Algebraic manipulation of the original equation yields

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

This is the fundamental equation of *linear interpolation*. It shows that

$$u = 0 \Rightarrow \mathbf{p} = \mathbf{p}_0$$

$$u = 1 \Rightarrow \mathbf{p} = \mathbf{p}_1$$

$$\mathbf{p} - \mathbf{p}_0 = u(\mathbf{p}_1 - \mathbf{p}_0), \quad u \in [0,1], \quad \mathbf{p}_0 \neq \mathbf{p}_1 \,.$$

Letting

$$a_0 = 1 - u$$
$$a_1 = u$$

the interpolation equation can also be written as

$$\mathbf{p} = a_0\mathbf{p}_0 + a_1\mathbf{p}_1$$
$$a_0 + a_1 = 1$$

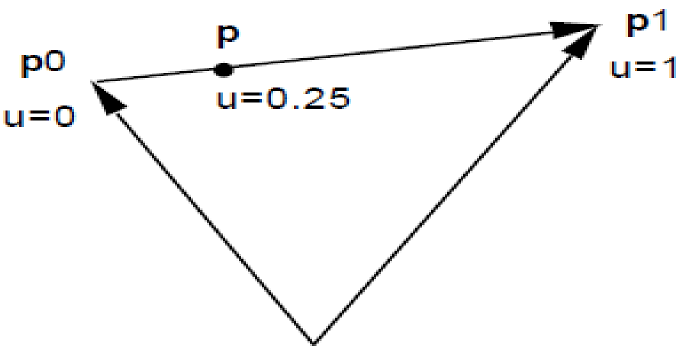


Figure 4.1.1.1 – Linear interpolation

$$\mathbf{p} \leftrightarrow \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad \boldsymbol{n} \leftrightarrow \begin{bmatrix} a \\ b \\ b \end{bmatrix}$$

$$\mathbf{p}_0 . \boldsymbol{n} = -d \ ,$$

the equation of the plane takes the familiar form

$$ax + by + cz + d = 0 \ .$$

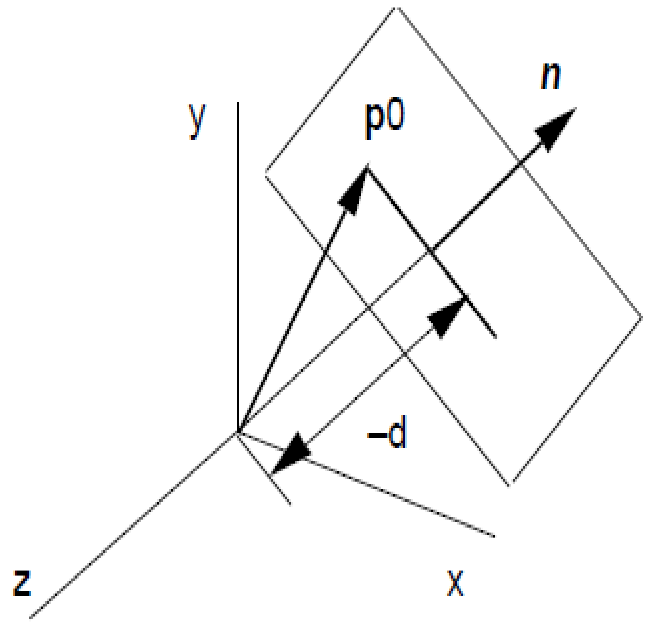


Figure 4.1.1.2 – A plane defined by a point and a normal vector

We defined a line by its parametric equation and a plane by its implicit equation. Lines also have implicit equations and planes parametric equations. For a line we consider two non-parallel planes and write

$$a_1x + b_1y + c_1z + d_1 = 0$$

$$a_2x + b_2y + c_2z + d_2 = 0$$

4.1.2 Curves

Calculus textbooks and most of the literature on curve modeling define a curve as a set of points **p** that satisfy a set of parametric equations:

$$
\begin{aligned}
p_x &= f_x(u) \\
p_y &= f_y(u) \,. \\
p_z &= f_z(u)
\end{aligned}
$$

These can be abbreviated as

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

set of reals we obtain the complete curve. If we restrict u to an interval, we define a *curve segment*.

It is important to understand that the same curve (i.e., set of points) can be defined by different parametric equations. The function **f** defines not only a curve but also a *parameterization* for it, and there are many ways of parameterizing the same curve.

The function **f** and some of its derivatives are usually required to be continuous. If **f** is continuous, the curve is called C^0 continuous. If both the function and its first derivative are continuous, we say the curve is C^1 continuous. In general, a curve is C^i continuous if **f** plus its first i derivatives are continuous.

If a curve is parameterized by its *arc length* s, the derivatives of the generic point of the curve $\mathbf{p}(s)$ are related to the tangent and normal to the curve as follows

$$\frac{d\mathbf{p}}{ds} = \boldsymbol{t}$$

$$\frac{d\boldsymbol{t}}{ds} = \frac{1}{\rho}\boldsymbol{n}$$

$$x = a_3u^3 + a_2u^2 + a_1u + a_0$$

$$y = b_3u^3 + b_2u^2 + b_1u + b_0$$

$$z = c_3u^3 + c_2u^2 + c_1u + c_0$$

If we interpret the coefficients as the coordinates of points

$$\mathbf{p}_3 \leftrightarrow \begin{bmatrix} a_3 \\ b_3 \\ c_3 \end{bmatrix}, \quad \mathbf{p}_2 \leftrightarrow \begin{bmatrix} a_2 \\ b_2 \\ c_2 \end{bmatrix}, \quad \mathbf{p}_1 \leftrightarrow \begin{bmatrix} a_1 \\ b_1 \\ c_1 \end{bmatrix}, \quad \mathbf{p}_0 \leftrightarrow \begin{bmatrix} a_0 \\ b_0 \\ c_0 \end{bmatrix}$$

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

which shows that the generic point of a parametric polynomial curve can be expressed as a linear combination of other points, usually called *control points*.

The parametric definition of a curve has its limitations. For example, space-filling "curves" can be defined by continuous **f** functions. These "curves" contradict our intuition of a curve as a 1-D entity, because they actually correspond to 2-D or 3-D regions of space. In addition, parametric curves can self-intersect. If self-intersections are undesirable, we can model a curve by a different mathematical entity called a manifold.

A *closed n-manifold* is a set that is locally just like an Euclidean space. For example, each point in a closed 1-manifold has a small interval around it that can be elastically deformed into an interval of the real line. An elastic deformation, without cutting or glueing, is called in topology a *homeomorphism*. Two sets are called *topologically equivalent* if they are related by a homeomorphism. Topology, like Euclidean geometry, is the study of properties of objects that remain invariant under certain transformations. For Euclidean geometry the relevant transformations were the isometries, whereas for topology they are the homeomorphisms. The top three 1-D objects of Figure 4.1.2.1 are homeomorphic, and so are the three bottom objects. However, top objects are not homeomorphic to bottom ones, because a top object cannot be elastically deformed into a bottom one without glueing its endpoints.

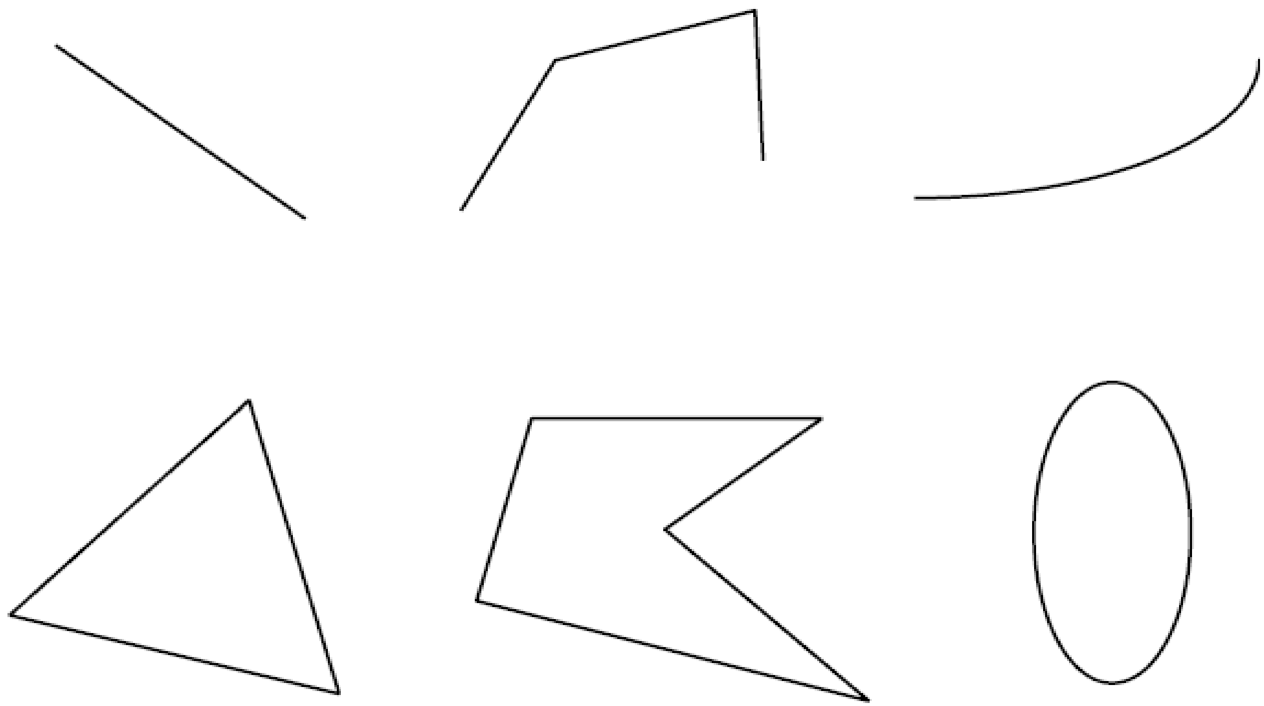


Figure 4.1.2.1 – Compact, connected 1-manifolds

$$f_1(x,y,z) = 0$$

$$f_2(x,y,z) = 0$$

This is a generalization of the implicit equation of a straight line, discussed in the previous section. The most interesting class of curves defined implicitly arises when the functions f are *algebraic*, i.e., they are polynomials in the spatial variables x, y, z.

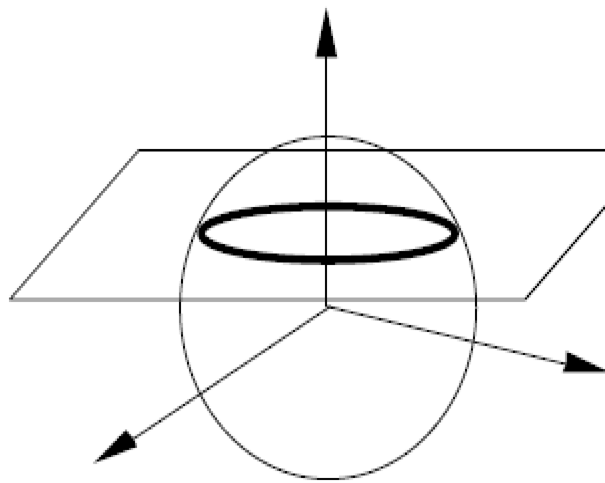


Figure 4.1.2.3 – A circle defined implicitly

$$x = f_x(u,v)$$
$$y = f_y(u,v) \; .$$
$$z = f_z(u,v)$$

Now two parameters are needed, because surfaces are intrinsically two dimensional. These equations may be abbreviated as

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

At each point of a smooth surface there are infinitely many lines tangent to the surface. These lines define the tangent plane at the point. The tangent plane is perpendicular to the normal to the surface.

If we let

$$u = g_u(t)$$

$$v = g_v(t)$$

and substitute in the parametric equation of a surface, we obtain

$$\mathbf{p} = \mathbf{f}[g_u(t), g_v(t)] = \mathbf{h}(t) .$$

derivatives of the **h** functions. Therefore, the tangent vector to a constant-v curve is $\dfrac{\partial \mathbf{p}}{\partial u}$, and the tangent to a constant-u curve is $\dfrac{\partial \mathbf{p}}{\partial v}$. The normal to the surface must be perpendicular to both of these tangents, and therefore (assuming the two vectors are not parallel) the unit normal is given by

$$n = \frac{\dfrac{\partial \mathbf{p}}{\partial u} \times \dfrac{\partial \mathbf{p}}{\partial v}}{\left| \dfrac{\partial \mathbf{p}}{\partial u} \times \dfrac{\partial \mathbf{p}}{\partial v} \right|}.$$

Figure 4.1.3.1 shows that the surface of a solid cube is an example of a piecewise-planar, or *polyhedral*, closed 2-manifold. It is a collection of polygonal faces such that the neighborhoods of vertices, or of points in the interior of an edge, or in the interior of a face, all can be deformed elastically so as to become disks.

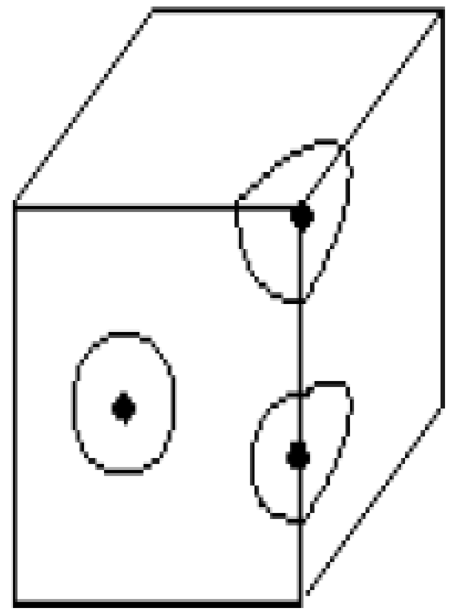


Figure 4.1.3.1 – The surface of a solid cube is a closed 2-manifold

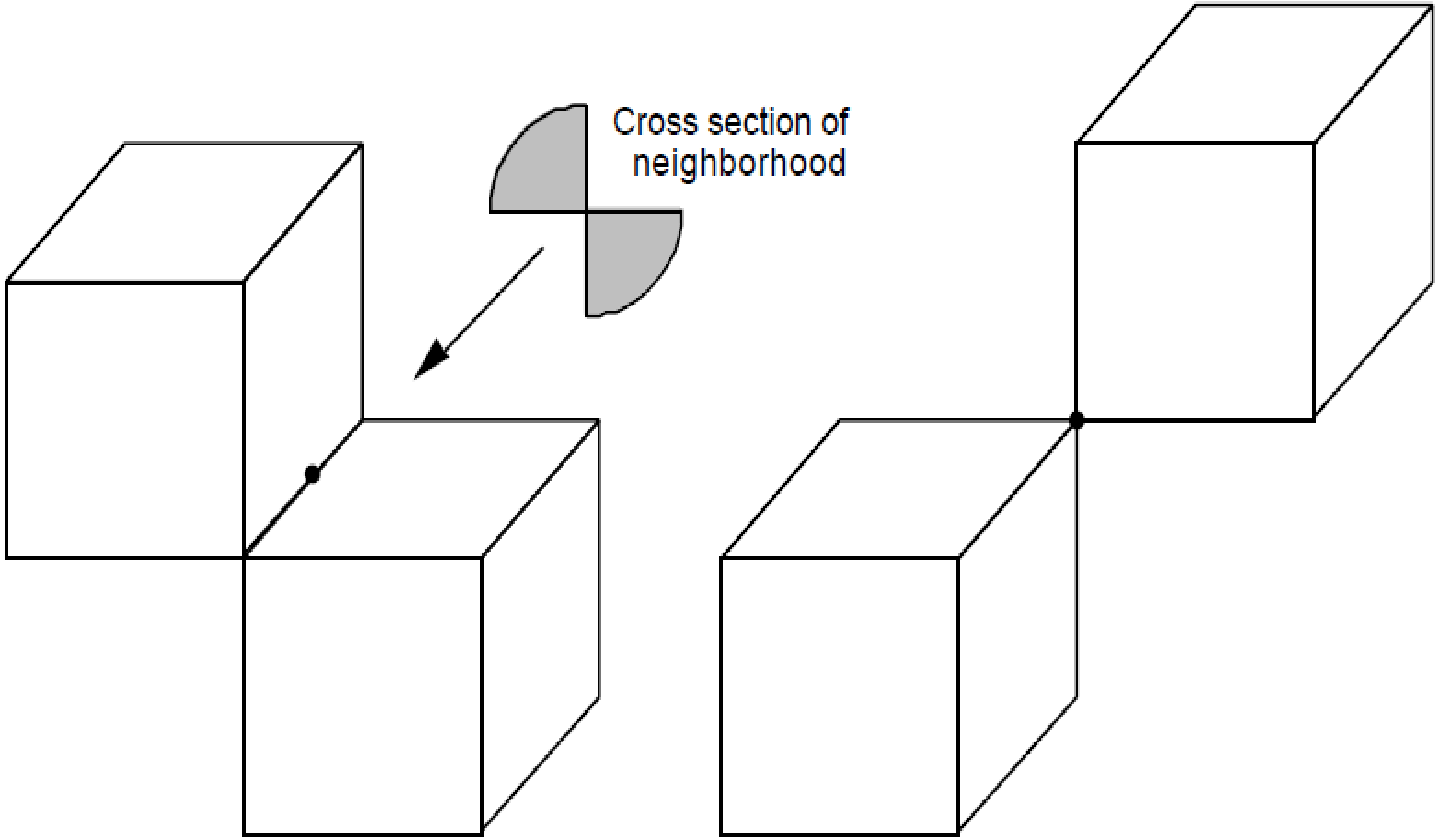


Figure 4.1.3.2 – The boundaries of two cubes glued at an edge or at a vertex are not 2-manifolds

A plane is the simplest example of an algebraic surface, and is defined by a linear implicit equation. Equations of degree higher than one correspond to curved surfaces. For example, a sphere of radius R, centered at the origin, is defined by

$$x^2 + y^2 + z^2 - R^2 = 0 ,$$

and a cylinder of radius R and axis coincident with the z coordinate axis is defined by

$$x^2 + y^2 - R^2 = 0 .$$

The normal to a surface $f(x, y, z) = 0$ is parallel to its *gradient* vector, defined as

$$\nabla f \leftrightarrow \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \\ \frac{\partial f}{\partial z} \end{bmatrix} .$$

Parametric polynomial or rational surfaces can be implicitized, but the resulting algebraic varieties typically have much higher degrees.

where the prime denotes the derivative. The geometric meaning of these constraints is shown in Figure 4.2.1.1.

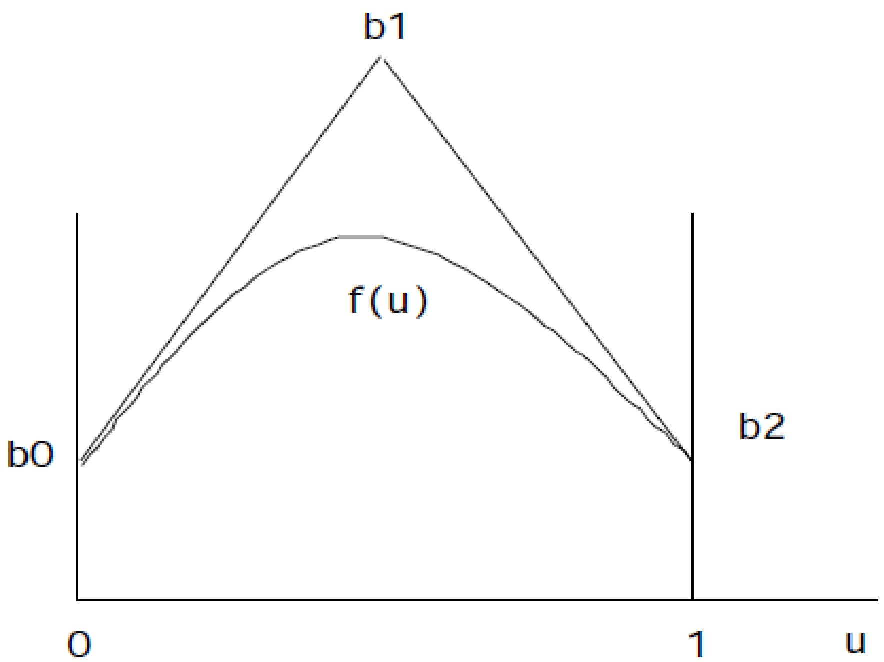


Figure 4.2.1.1 – A quadratic polynomial and its three defining points.

The derivative of *f* is

$$f'(u) = \begin{bmatrix} 0 & 1 & 2u \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}$$

$$f(0) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = b_0$$

$$f(0) + \frac{1}{2} f'(0) = \begin{bmatrix} 1 & \frac{1}{2} & 0 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = b_1$$

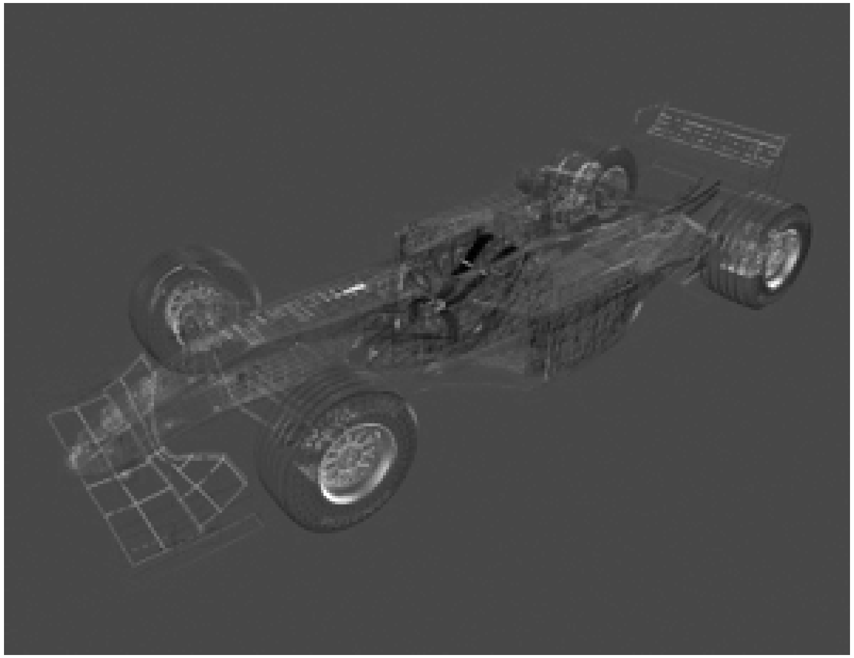
$$f(1) = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = b_2$$

These three equations can be written together in matrix form as

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & ½ & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix}.$$

Modeling Complex Shapes

- We want to build models of very complicated objects
- Complexity is achieved using simple pieces
  - polygons,
  - parametric curves and surfaces, or
  - implicit curves and surfaces
- This lecture: **parametric curves**



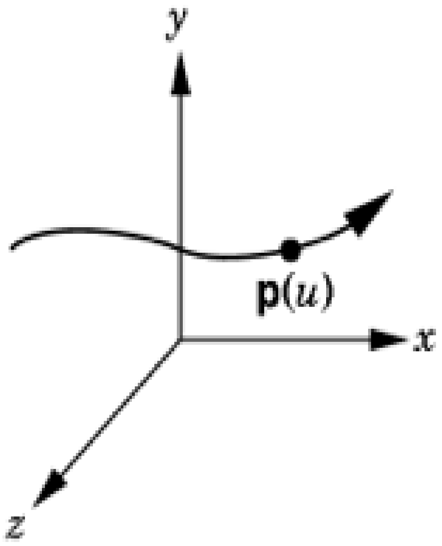
What Do We Need From Curves in Computer Graphics?

- Local control of shape
(so that easy to build and modify)
- Stability
- Smoothness and continuity
- Ability to evaluate derivatives
- Ease of rendering

Curve Representations

- Explicit: y = f(x) $y = x^2$ $y = mx + b$
  - Must be a function (single-valued)
  - Big limitation—vertical lines?
- Parametric: (x,y) = (f(u),g(u))
  + Easy to specify, modify, control
  - Extra "hidden" variable u, the *parameter*

$$(x, y) = (\cos u, \sin u)$$

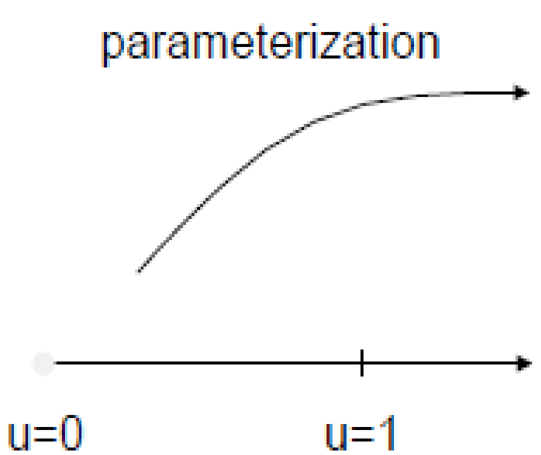


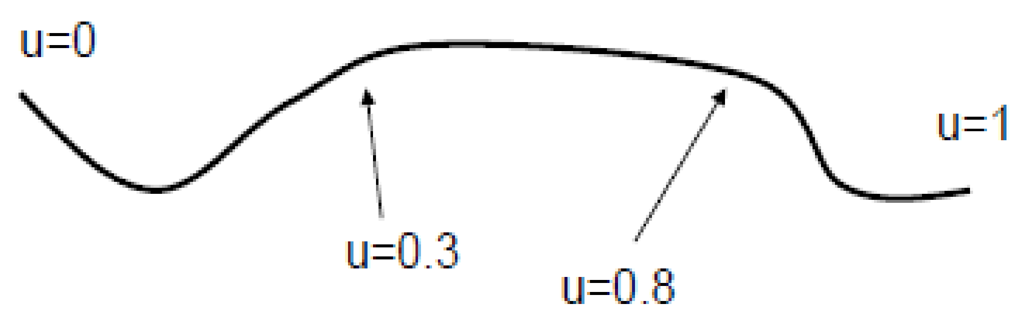
- Implicit: f(x,y) = 0
  + y can be a multiple valued function of x
  - Hard to specify, modify, control

$$x^2 + y^2 - r^2 = 0$$

Parameterization of a Curve

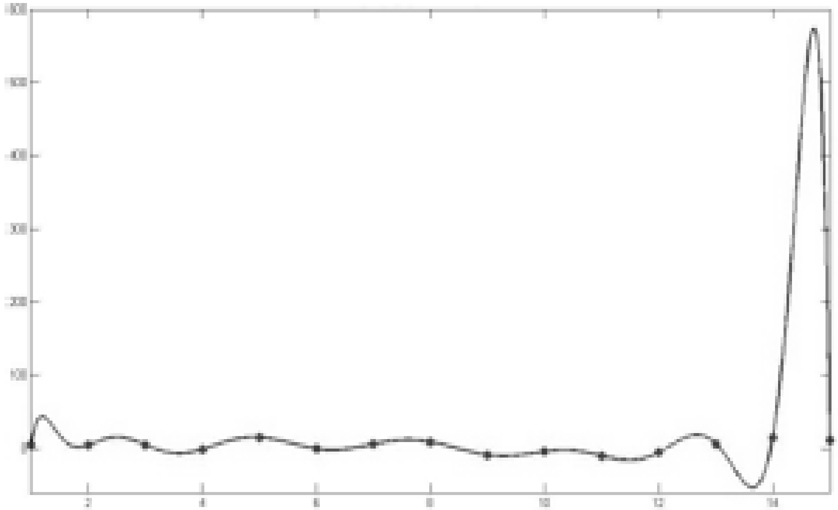
- *Parameterization* of a curve: how a change in u moves you along a given curve in xyz space.
- Parameterization is not unique. It can be slow, fast, with continuous / discontinuous speed, clockwise (CW) or CCW…





Polynomial Interpolation

- An *n*-th degree polynomial fits a curve to n+1 points
  - called Lagrange Interpolation
  - result is a curve that is too wiggly, change to any control point affects entire curve (non-local)
  - *this method is poor*
- We usually want the curve to be as smooth as possible
  - minimize the wiggles
  - high-degree polynomials are bad

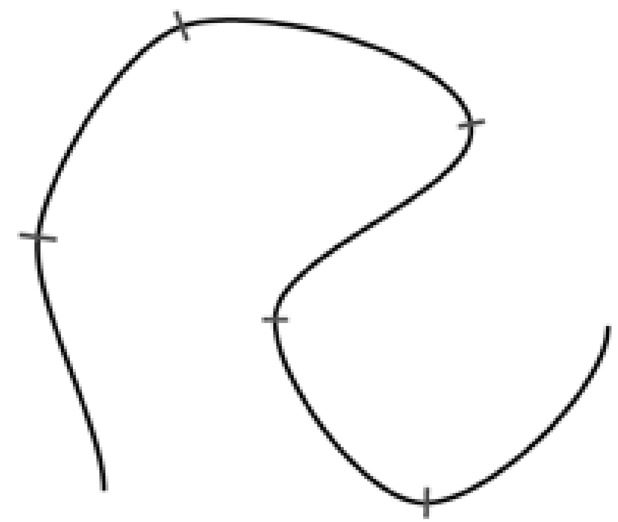


source: Wikipedia

Lagrange interpolation, degree=15

Splines: Piecewise Polynomials

- A spline is a *piecewise polynomial:* Curve is broken into consecutive segments, each of which is a low-degree polynomial interpolating (passing through) the control points

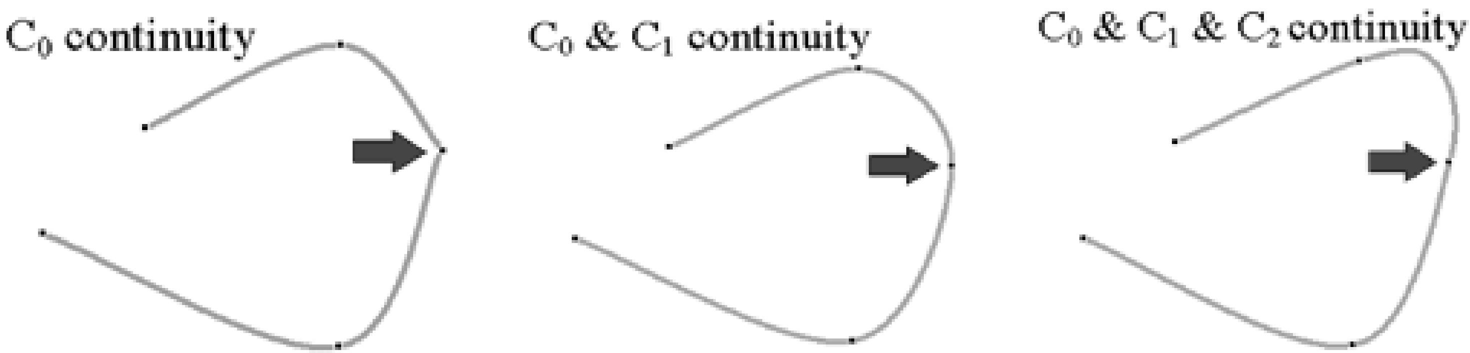


a spline

- *Cubic* piecewise polynomials are the most common:
  - They are the lowest order polynomials that
    1. interpolate two points and
    2. allow the gradient at each point to be defined (C^1 continuity is possible).
  - Piecewise definition gives local control.
  - Higher or lower degrees are possible, of course.

Piecewise Polynomials

- Spline: many polynomials pieced together
- Want to make sure they fit together nicely



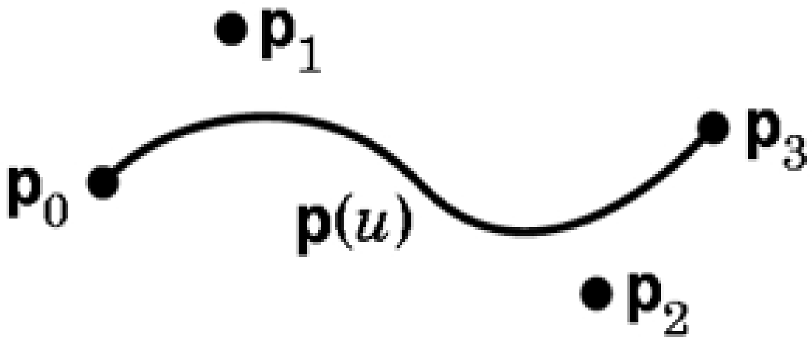
Continuous in position

Continuous in position and tangent vector

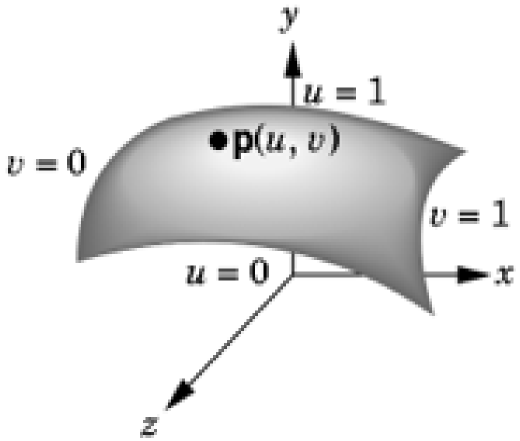
Continuous in position, tangent, and curvature

Splines

- Types of splines:
  - Hermite Splines
  - Bezier Splines
  - Catmull-Rom Splines
  - Natural Cubic Splines
  - B-Splines
  - NURBS



- Splines can be used to model both curves and surfaces



Cubic Curves in 3D

- Cubic polynomial:
  - $p(u) = au^3+bu^2+cu+d = [u^3 \;\; u^2 \;\; u \;\; 1]\,[a \;\; b \;\; c \;\; d]^T$
  - a,b,c,d are 3-vectors, u is a scalar
- Three cubic polynomials, one for each coordinate:
  - $x(u) = a_x u^3 + b_x u^2 + c_x u + d_x$
  - $y(u) = a_y u^3 + b_y u^2 + c_y u + d_y$
  - $z(u) = a_z u^3 + b_z u^2 + c_z u + d_z$
- In matrix notation:

$$\begin{bmatrix} x(u) & y(u) & z(u) \end{bmatrix} = \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix} \begin{bmatrix} a_x & a_y & a_z \\ b_x & b_y & b_z \\ c_x & c_y & c_z \\ d_x & d_y & d_z \end{bmatrix}$$

- Or simply: $p = [u^3 \; u^2 \; u \; 1]\, A$

Cubic Hermite Splines

Hermite Specification

We want a way to specify the end points and the slope at the end points!

Deriving Hermite Splines

- Four constraints: value and slope (in 3-D, position and tangent vector) at beginning and end of interval [0,1] :

$$p(0) = p_1 = (x_1 , y_1 , z_1)$$
$$p(1) = p_2 = (x_2 , y_2 , z_2)$$
$$p'(0) = \overline{p}_1 = (\overline{x}_1 , \overline{y}_1 , \overline{z}_1)$$
$$p'(1) = \overline{p}_2 = (\overline{x}_2 , \overline{y}_2 , \overline{z}_2)$$

the user constraints

- Assume cubic form: $p(u) = au^3 + bu^2 + cu + d$
- Four unknowns: a, b, c, d

Deriving Hermite Splines

- Assume cubic form: $p(u) = au^3 + bu^2 + cu + d$

$$p_1 = p(0) = d$$

$$p_2 = p(1) = a + b + c + d$$

$$\overline{p_1} = p'(0) = c$$

$$\overline{p_2} = p'(1) = 3a + 2b + c$$

- Linear system: 12 equations for 12 unknowns
(however, can be simplified to 4 equations for 4 unknowns)

- Unknowns: *a*, *b*, *c*, *d* (each of *a*, *b*, *c*, *d* is a 3-vector)

Deriving Hermite Splines

$$\boxed{\begin{aligned} d &= p_1 \\ a + b + c + d &= p_2 \\ c &= \overline{p}_1 \\ 3a + 2b + c &= \overline{p}_2 \end{aligned}}$$

Rewrite this 12x12 system as a 4x4 system:

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix} \begin{bmatrix} a_x & a_y & a_z \\ b_x & b_y & b_z \\ c_x & c_y & c_z \\ d_x & d_y & d_z \end{bmatrix} = \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \overline{x}_1 & \overline{y}_1 & \overline{z}_1 \\ \overline{x}_2 & \overline{y}_2 & \overline{z}_2 \end{bmatrix}$$

The Cubic Hermite Spline Equation

- After inverting the 4x4 matrix, we obtain:

$$\underbrace{\begin{bmatrix} x & y & z \end{bmatrix}}_{\text{point on the spline}} = \underbrace{\begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix}}_{\text{parameter vector}} \underbrace{\begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}}_{\text{basis}} \underbrace{\begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \bar{x}_1 & \bar{y}_1 & \bar{z}_1 \\ \bar{x}_2 & \bar{y}_2 & \bar{z}_2 \end{bmatrix}}_{\substack{\text{control matrix} \\ \text{(what the user gets to pick)}}}$$

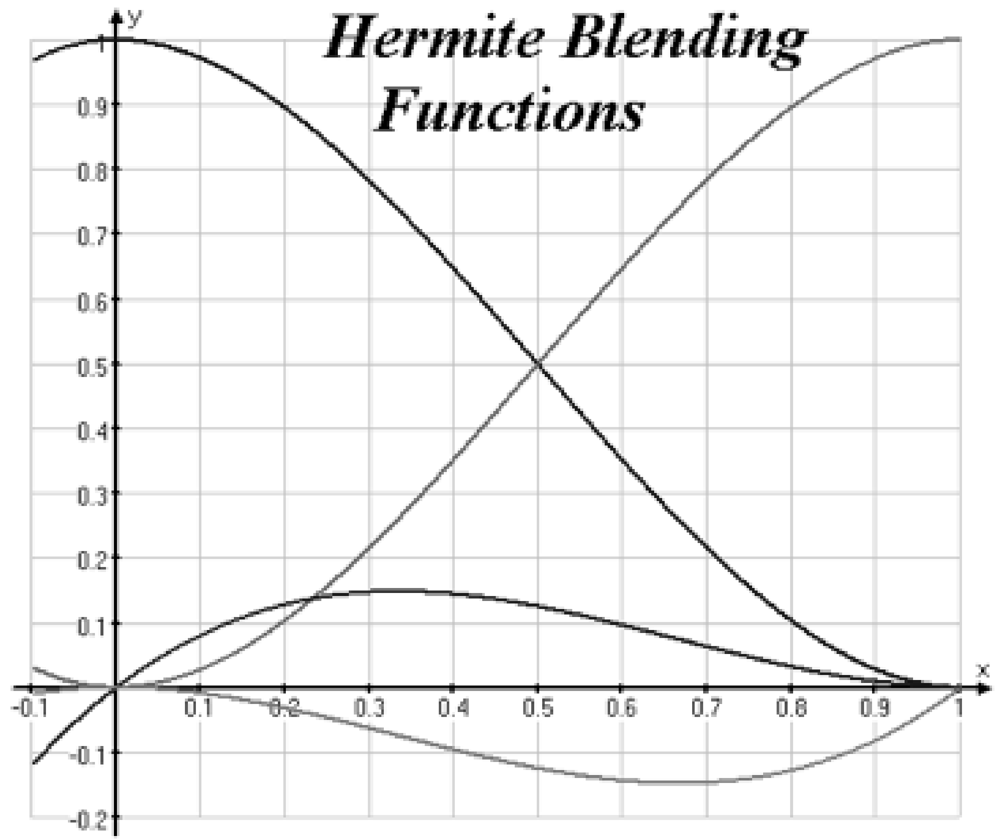
- This form is typical for splines
  - basis matrix and meaning of control matrix change with the spline type

Four Basis Functions for Hermite Splines

transpose

$$p(u) = \begin{bmatrix} 2u^3 - 3u^2 + 1 \\ -2u^3 + 3u^2 \\ u^3 - 2u^2 + u \\ u^3 - u^2 \end{bmatrix}^T \begin{bmatrix} p_1 \\ p_2 \\ \overline{p}_1 \\ \overline{p}_2 \end{bmatrix}$$

4 Basis Functions

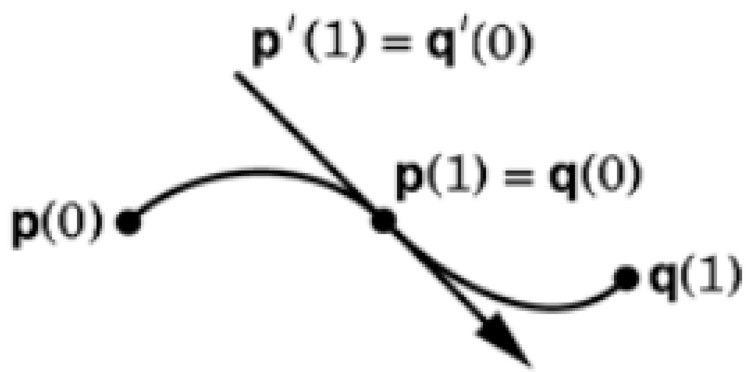


Every cubic Hermite spline is a linear combination (blend) of these 4 functions.

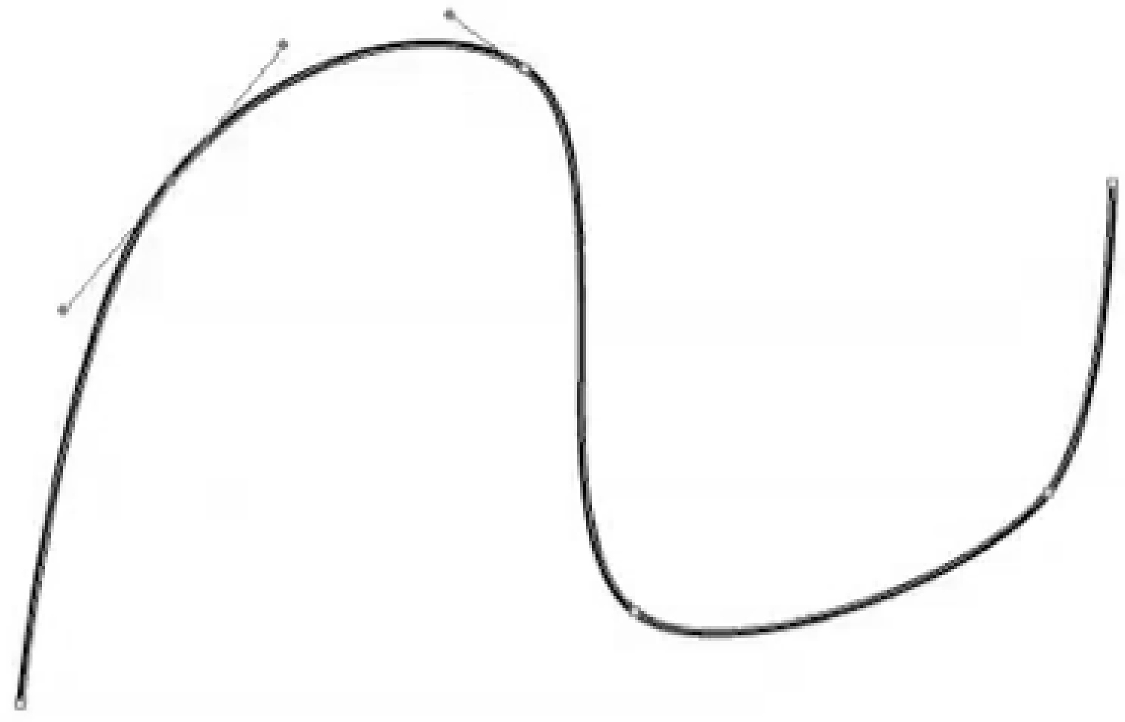
Piecing together Hermite Splines

It's easy to make a multi-segment Hermite spline:

- each segment is specified by a cubic Hermite curve
- just specify the position and tangent at each "joint" (called *knot*)
- the pieces fit together with matched positions and first derivatives
- gives C1 continuity

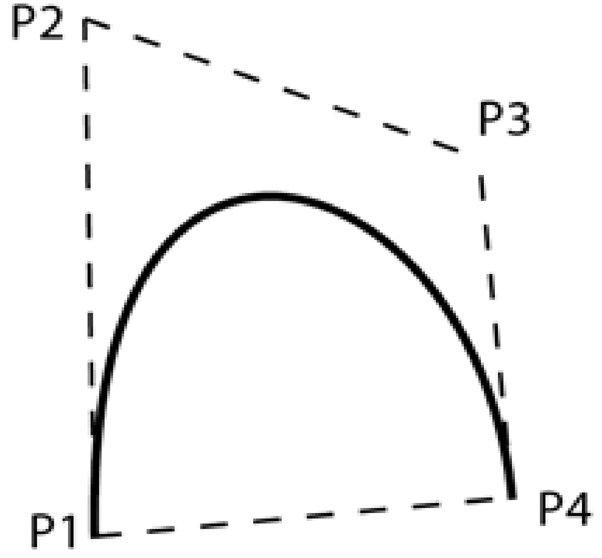


Hermite Splines in Adobe Illustrator



Bezier Splines

- Variant of the Hermite spline
- Instead of endpoints and tangents, four control points
  - points P1 and P4 are on the curve
  - points P2 and P3 are off the curve
  - p(0) = P1, p(1) = P4,
  - p'(0) = 3(P2-P1), p'(1) = 3(P4 - P3)



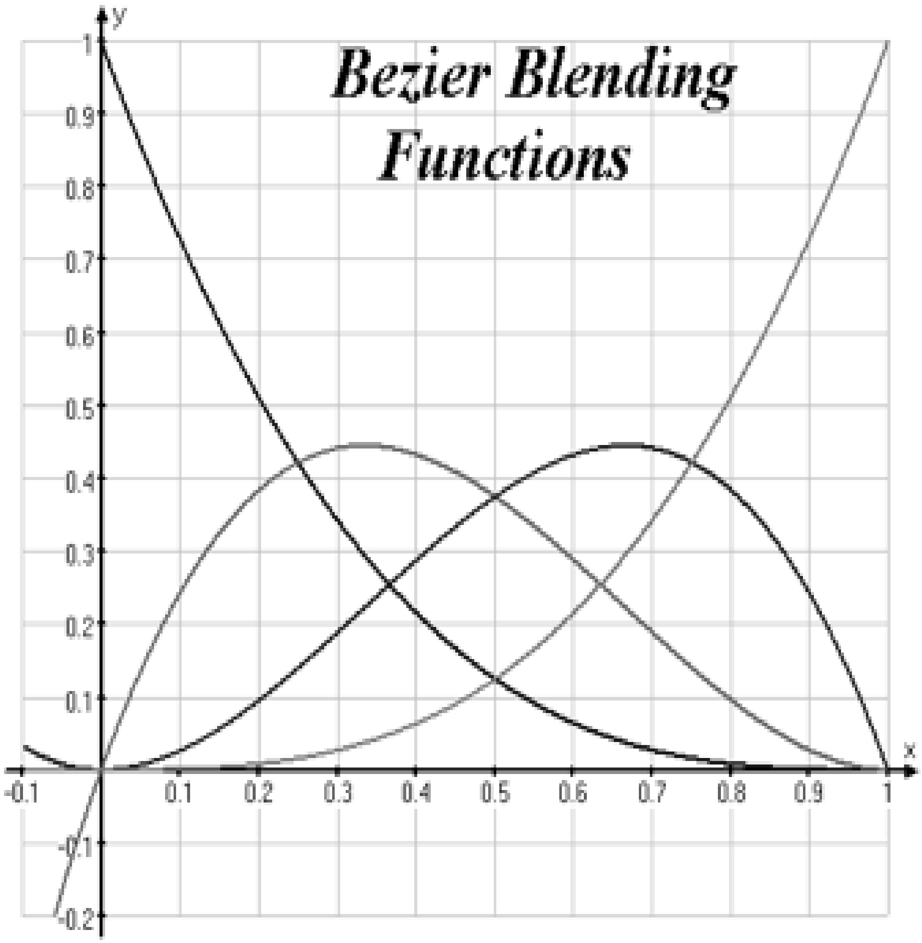
- Basis matrix is derived from the Hermite basis (or from scratch)
- Convex Hull property: curve contained within the convex hull of control points
- Scale factor "3" is chosen to make "velocity" approximately constant

The Bezier Spline Matrix

$$[x \ \ y \ \ z] = [u^3 \ \ u^2 \ \ u \ \ 1] \underbrace{\begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}}_{\text{Hermite basis}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -3 & 3 & 0 & 0 \\ 0 & 0 & -3 & 3 \end{bmatrix}}_{\text{Bezier to Hermite}} \underbrace{\begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \\ x_4 & y_4 & z_4 \end{bmatrix}}_{\substack{\text{Bezier} \\ \text{control matrix}}}$$

$$= [u^3 \ \ u^2 \ \ u \ \ 1] \underbrace{\begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}}_{\text{Bezier basis}} \underbrace{\begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \\ x_4 & y_4 & z_4 \end{bmatrix}}_{\substack{\text{Bezier} \\ \text{control matrix}}}$$

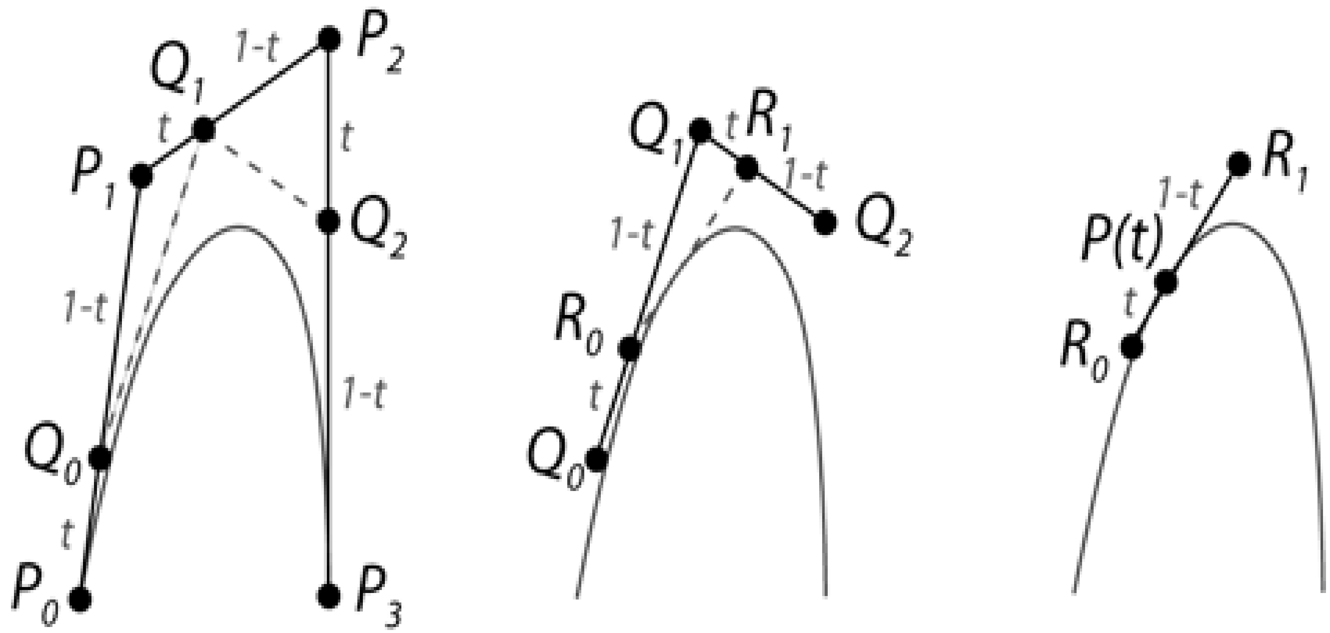
Bezier Blending Functions



$$p(t) = \begin{bmatrix} (1-t)^3 \\ 3t(1-t)^2 \\ 3t^2(1-t) \\ t^3 \end{bmatrix}^{\mathbf{T}} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix}$$

- Also known as the order 4, degree 3 Bernstein polynomials
- Nonnegative, sum to 1
- The entire curve lies inside the polyhedron bounded by the control points

DeCasteljau Construction



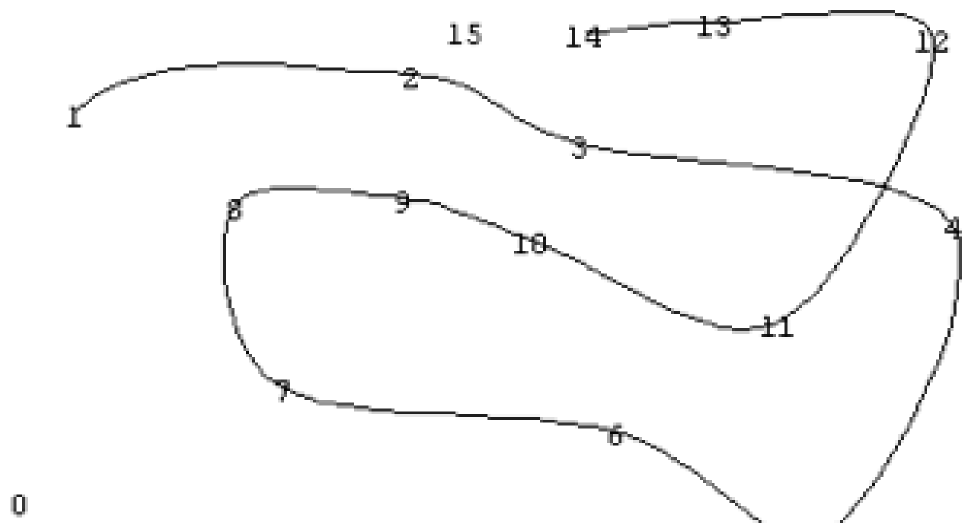
Efficient algorithm to evaluate Bezier splines.

Similar to Horner rule for polynomials.

Can be extended to interpolations of 3D rotations.

Catmull-Rom Splines

- Roller-coaster (next programming assignment)
- With Hermite splines, the designer must arrange for consecutive tangents to be collinear, to get C^1 continuity. Similar for Bezier. This gets tedious.
- Catmull-Rom: an interpolating cubic spline with *built-in C^1 continuity*.
- Compared to Hermite/Bezier: fewer control points required, but less freedom.



Constructing the Catmull-Rom Spline

Suppose we are given n control points in 3-D: p_1, p_2, …, p_n.

For a Catmull-Rom spline, we set the tangent at p_i to $s^*(p_{i+1} – p_{i-1})$ for i=2, ..., n-1, for some s (often s=0.5)

s is *tension parameter*: determines the magnitude (but not direction!) of the tangent vector at point p_i

What about endpoint tangents? Use extra control points p_0, p_{n+1} .

Now we have positions and tangents at each knot. This is a Hermite specification. Now, just use Hermite formulas to derive the spline.

Note: curve between p_i and p_{i+1} is completely determined by p_{i-1}, p_i, p_{i+1}, p_{i+2} .

Catmull-Rom Spline Matrix

$$[x \;\; y \;\; z] = [u^3 \;\; u^2 \;\; u \;\; 1] \begin{bmatrix} -s & 2-s & s-2 & s \\ 2s & s-3 & 3-2s & -s \\ -s & 0 & s & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \\ x_4 & y_4 & z_4 \end{bmatrix}$$

basis control matrix

- Derived in way similar to Hermite and Bezier
- Parameter s is typically set to s=1/2.

Splines with More Continuity?

- So far, only C^1 continuity.
- How could we get C^2 continuity at control points?

- Possible answers:
  - Use higher degree polynomials

    degree 4 = quartic, degree 5 = quintic, … but these get computationally expensive, and sometimes wiggly
  - Give up local control → natural cubic splines

    A change to any control point affects the entire curve
  - Give up interpolation → cubic B-splines

    Curve goes near, but not through, the control points

Comparison of Basic Cubic Splines

Type	Local Control	Continuity	Interpolation
Hermite	YES	C1	YES
Bezier	YES	C1	YES
Catmull-Rom	YES	C1	YES
Natural	NO	C2	YES
B-Splines	YES	C2	NO

Summary:

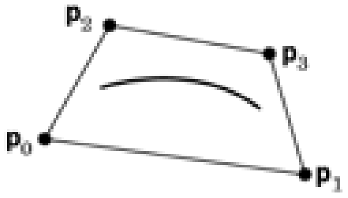
Cannot get C2, interpolation and local control with cubics

Natural Cubic Splines

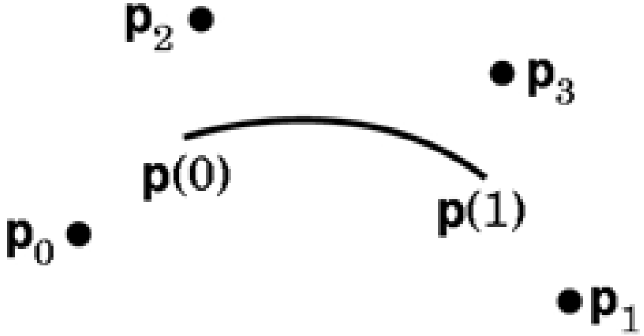
- If you want 2nd derivatives at joints to match up, the resulting curves are called *natural cubic splines*
- It's a simple computation to solve for the cubics' coefficients. (*See Numerical Recipes in C* book for code.)
- Finding all the right weights is a *global* calculation (solve tridiagonal linear system)

B-Splines

- Give up interpolation
  - the curve passes *near* the control points
  - best generated with interactive placement (because it's hard to guess where the curve will go)



- Curve obeys the convex hull property
- C2 continuity and local control are good compensation for loss of interpolation

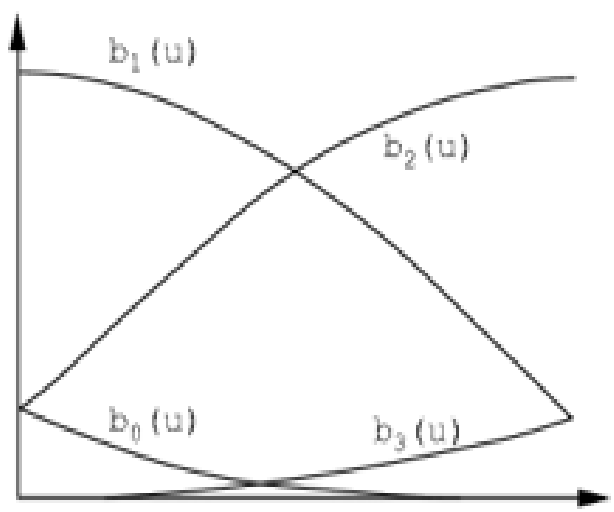


B-Spline Basis

- We always need 3 more control points than the number of spline segments

$$M_{Bs} = \frac{1}{6}\begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix}$$

$$G_{Bsi} = \begin{bmatrix} P_{i-3} \\ P_{i-2} \\ P_{i-1} \\ P_i \end{bmatrix}$$



Other Common Types of Splines

- Non-uniform Splines
- Non-Uniform Rational Cubic curves (NURBS)
- NURBS are very popular and used in many commercial packages

How to Draw Spline Curves

- Basis matrix equation allows same code to draw any spline type
- <u>Method 1</u>: brute force
  - Calculate the coefficients
  - For each cubic segment, vary *u* from *0* to *1* (fixed step size)
  - Plug in *u* value, matrix multiply to compute position on curve
  - Draw line segment from last position to current position
- What's wrong with this approach?
  - Draws in even steps of u
  - Even steps of u does not mean even steps of x
  - Line length will vary over the curve
  - Want to bound line length
    - » too long: curve looks jagged
    - » too short: curve is slow to draw

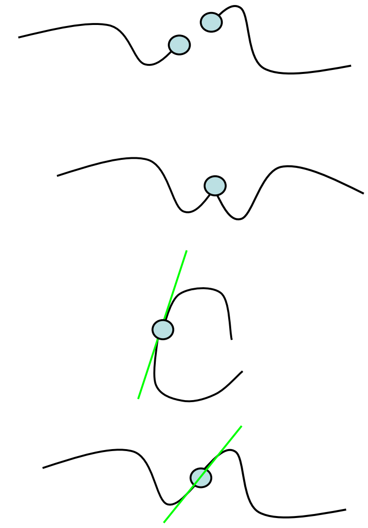
Drawing Splines, 2

- Method 2: recursive subdivision - vary step size to draw short lines

```
Subdivide(u0,u1,maxlinelength)
    umid = (u0 + u1)/2
    x0 = F(u0)
    x1 = F(u1)
    if |x1 - x0| > maxlinelength
       Subdivide(u0,umid,maxlinelength)
       Subdivide(umid,u1,maxlinelength)
    else drawline(x0,x1)
```

- Variant on Method 2 - subdivide based on curvature
  - replace condition in “if” statement with straightness criterion
  - draws fewer lines in flatter regions of the curve

Continuity at Join Points (from Lecture 2)



- Discontinuous: physical separation
- Parametric Continuity
  - Positional (C^0): no physical separation
  - $C^1 : C^0$ and matching first derivatives
  - $C^2 : C^1$ and matching second derivatives
- Geometric Continuity
  - Positional (G^0) $= C^0$
  - Tangential (G^1) : G^0 and tangents are proportional, point in same direction, but magnitudes may differ
  - Curvature (G^2) : G^1 and tangent lengths are the same and rate of length change is the same

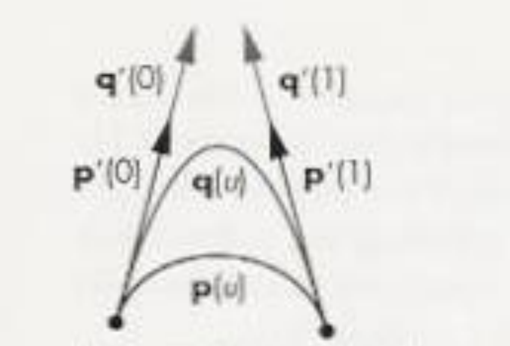


Figure 9.15 Change of magnitude in G^1 continuity.

source: Mortenson, Angel (Ch 9), Wiki

Continuity at Join Points

- Hermite curves provide C^1 continuity at curve segment join points.
  - matching parametric 1st derivatives
- Bezier curves provide C^0 continuity at curve segment join points.
  - Can provide G^1 continuity given collinearity of some control points (see next slide)
- Cubic B-splines can provide C^2 continuity at curve segment join points.
  - matching parametric 2nd derivatives

Composite Bezier Curves

Joining adjacent curve segments is an alternative to degree elevation.

Collinearity of cubic Bezier control points produces G^1 continuity at join point:

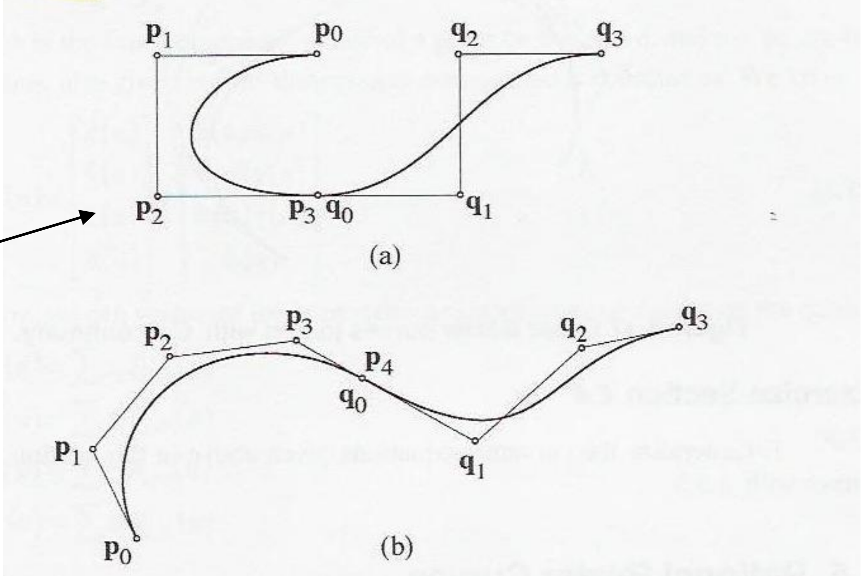


Figure 4.16 Composite Bézier Curves.

Evaluate at u=0 and u=1 to show tangents related to first and last control polygon line segment.

$$\mathbf{p}^u(0) = 3(\mathbf{p}_1 - \mathbf{p}_0) \qquad \mathbf{p}^u(1) = 3(\mathbf{p}_3 - \mathbf{p}_2)$$

For G^2 continuity at join point in cubic case, 5 vertices must be coplanar.

(this needs further explanation – see later slide)

Composite Bezier Surface

- Bezier surface patches can provide G^1 continuity at patch boundary curves.
- For common boundary curve defined by control points $\mathbf{p}_{14}$, $\mathbf{p}_{24}$, $\mathbf{p}_{34}$, $\mathbf{p}_{44}$, need collinearity of: $\{\mathbf{p}_{i,3}, \mathbf{p}_{i,4}, \mathbf{p}_{i,5}\}, \quad i \in [1:4]$
- Two adjacent patches are C^r across their common boundary iff all rows of control net vertices are interpretable as polygons of C^r piecewise Bezier curves.

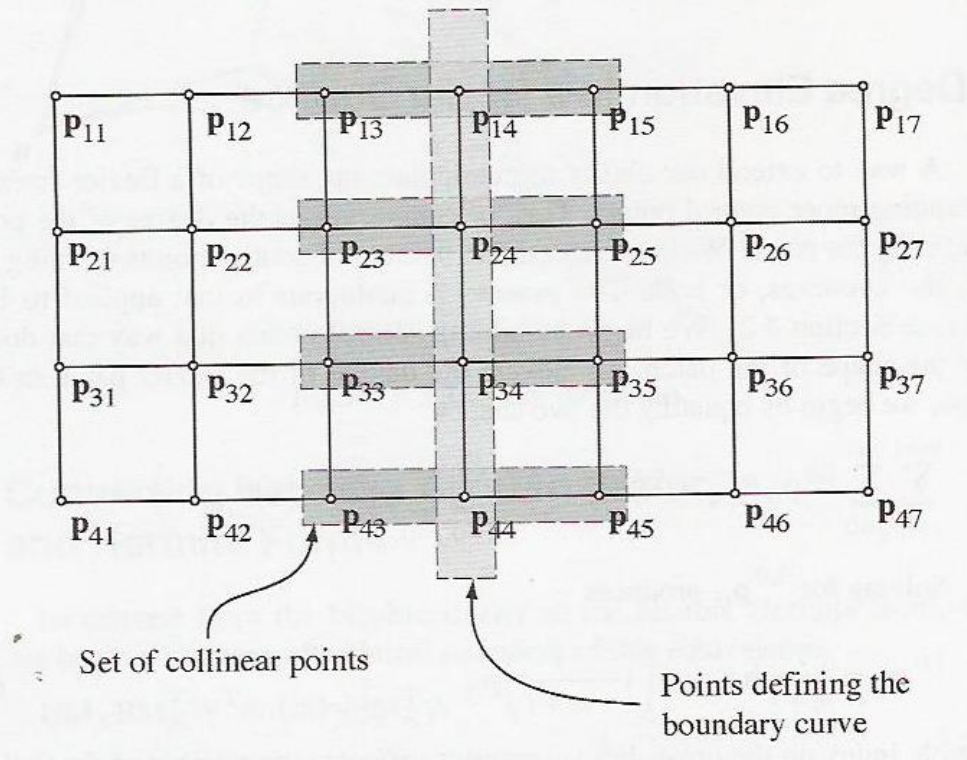


Figure 8.4 G[1] continuity across two Bézier patches.

•Cubic B-splines can provide C^2 continuity at surface patch boundary curves.

source: Mortenson, Farin

Continuity within a (Single) Curve Segment

- **Parametric C^k Continuity**:
  - Refers to the *parametric* curve representation and *parametric* derivatives
  - Smoothness of *motion along the parametric curve*
  - "A curve $P(t)$ has kth-order parametric continuity everywhere in the t-interval $[a,b]$ if all derivatives of the curve, up to the kth, exist and are continuous at all points inside $[a,b]$."
  - A curve with continuous parametric velocity and acceleration has 2nd-order parametric continuity.

Note that C^k continuity implies C^i continuity for $i < k$.

$$x(\theta) = Ke^{b\theta}\cos\theta \qquad y(\theta) = Ke^{b\theta}\sin\theta$$

apply product rule

$$x'(\theta) = (Ke^{b\theta})(-\sin\theta) + (\cos\theta)(Ke^{b\theta})(be^{b\theta})$$

$$y'(\theta) = (Ke^{b\theta})(\cos\theta) + (\sin\theta)(Ke^{b\theta})(be^{b\theta})$$

1st derivatives of parametric expression are continuous, so spiral has 1st-order (C^1) parametric continuity.

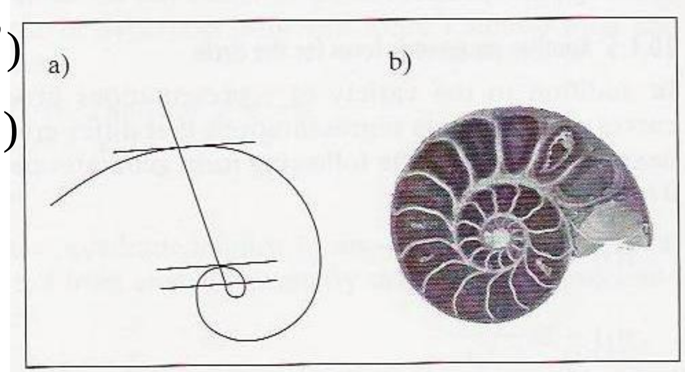


FIGURE 10.5 A logarithmic spiral and chambered nautilus.

Example

source: Hill, Ch 10

Continuity within a (Single) Curve Segment (continued)

- **Geometric G^k Continuity in interval $[a,b]$** (assume P is curve):
  - "Geometric continuity requires that various derivative vectors have a continuous direction even though they might have discontinuity in speed."
  - $G^0 = C^0$
  - G^1: $P'(c-) = k\,P'(c+)$ for some constant k for every c in $[a,b]$.
    - Velocity vector may jump in size, but its direction is continuous.
  - G^2: $P'(c-) = k\,P'(c+)$ for some constant k and $P''(c-) = m\,P''(c+)$ for some constants k and m for every c in $[a,b]$.
    - Both 1st and 2nd derivative directions are continuous.

Note that, for these definitions, G^k continuity implies G^i continuity for $i < k$.

These definitions suffice for that textbook's treatment, but there is more to the story…

source: Hill, Ch 10

Reparameterization Relationship

- Curve has G^r continuity if an arc-length reparameterization exists after which it has C^r continuity. *source: Farin, Ch 10*
- "Two curve segments are G^k geometric continuous at the joining point if and only if there exist two parameterizations, one for each curve segment, such that all ith derivatives, $i \le k$, computed with these new parameterizations agree at the joining point." *source: cs.mtu.edu*

Additional Perspective

- “Parametric continuity of order n implies geometric continuity of order n, but not vice-versa.”

Continuity at Join Point

Parametric Continuity

- Defined using parametric differential properties of curve or surface
- C^k more restrictive than G^k

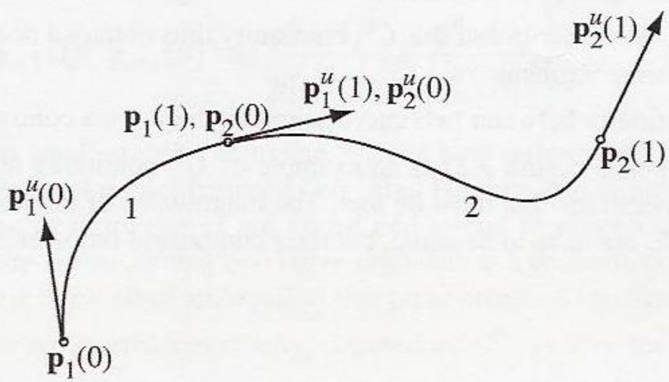


Figure 3.27 Conditions required for G^1 continuity.

Geometric Continuity

- Defined using intrinsic differential properties of curve or surface (e.g. unit tangent vector, curvature), independent of parameterization.
- G^1: common tangent line
- G^2: same curvature, requiring conditions from Hill (Ch 10) & (see differential geometry slides)
  - *Osculating planes* coincide or
  - *Binormals* are collinear.

source: Mortenson Ch 3, p. 100-102

Parametric Cross-Plot

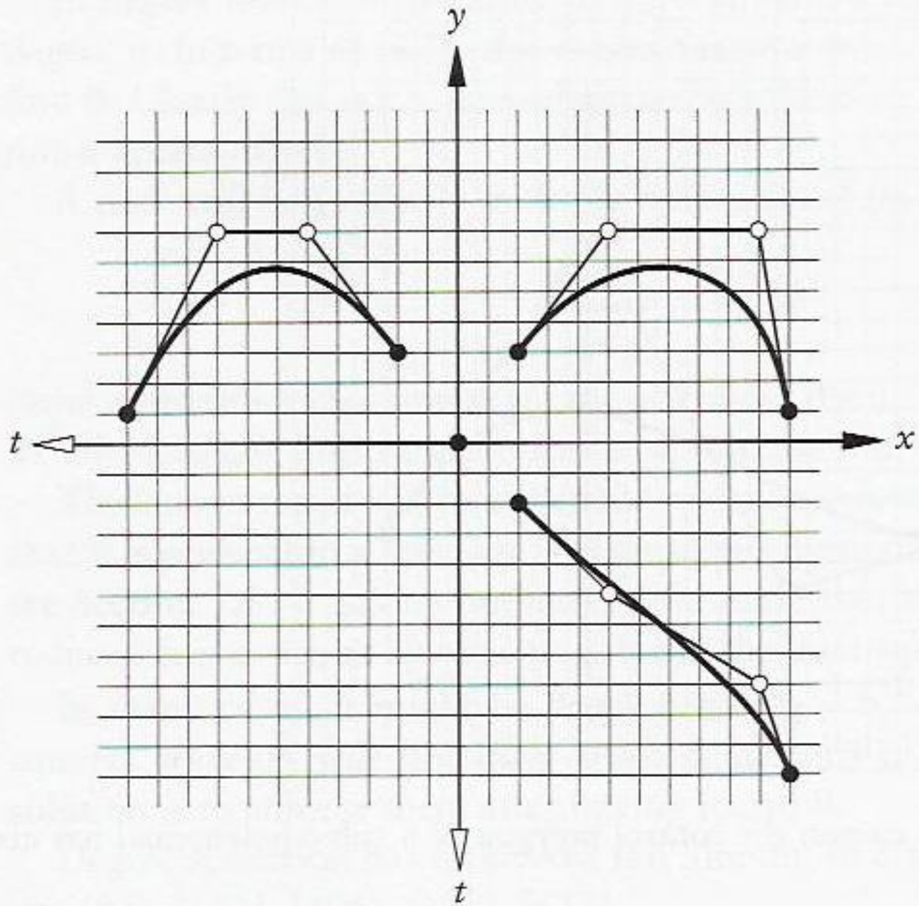


Figure 6.4 Cross plots: a two-dimensional Bézier curve together with its two coordinate functions.

For Farin's discussion of C^1 continuity at join point, cross-plot notion is useful.

source: Farin, Ch 6

Composite Cubic Bezier Curves (continued)

source: Farin, Ch 5

Domain violates (5.30) for y component.

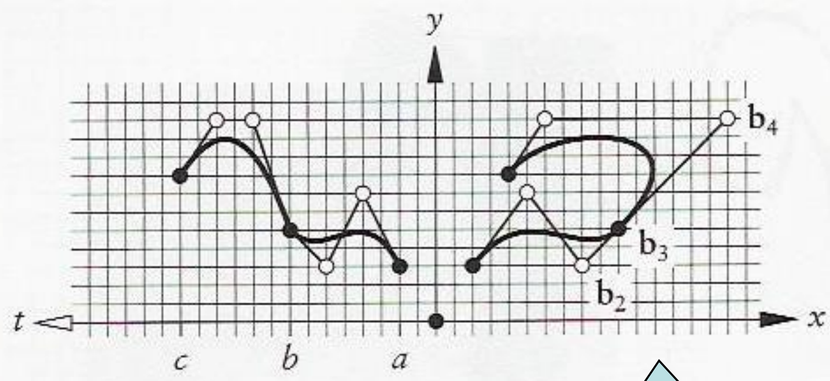


Figure 5.12 Composite curves: a C^0 example.

curves are identical in x,y space

Domain satisfies (5.30) for y component.

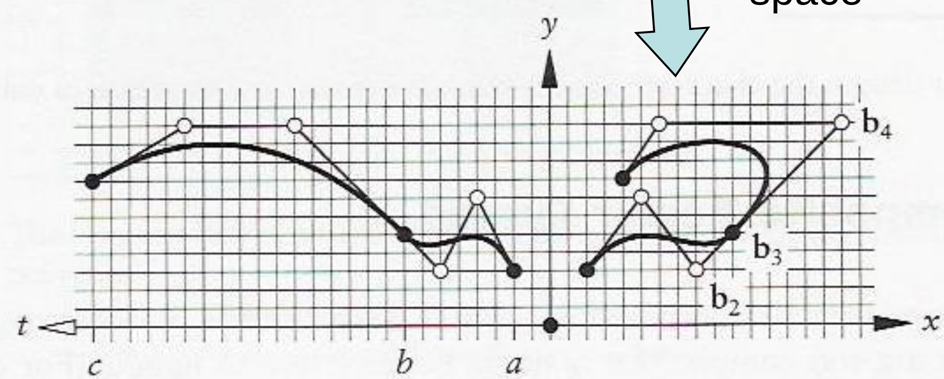


Figure 5.13 Composite curves: a C^1 example.

Parametric C^1 continuity, with parametric domains considered, requires (for x and y components):

$$\frac{3}{(b-a)}\left[b_3 - b_2\right] = \frac{3}{(c-b)}\left[b_4 - b_3\right] \quad (5.30)$$

Composite Bezier Curves

For G^2 continuity at join point in cubic case, 5 vertices

$$\mathbf{p}_{m-2}, \mathbf{p}_{m-1}, \mathbf{p}_m = \mathbf{q}_0, \mathbf{q}_1, \mathbf{q}_2$$

must be coplanar.

(follow-up from prior slide)

Achieving this might require adding control points (degree elevation).

$$\kappa_0 = \frac{2\left|(\mathbf{p}_1 - \mathbf{p}_0) \times (\mathbf{p}_2 - \mathbf{p}_1)\right|}{3\left|\mathbf{p}_1 - \mathbf{p}_0\right|^3} \qquad \kappa_1 = \frac{2\left|(\mathbf{p}_2 - \mathbf{p}_1) \times (\mathbf{p}_3 - \mathbf{p}_2)\right|}{3\left|\mathbf{p}_3 - \mathbf{p}_2\right|^3}$$

curvature at endpoints of curve segment

consistent with: $\kappa_i = \dfrac{\left|\mathbf{p}_i^u \times \mathbf{p}_i^{uu}\right|}{\left|\mathbf{p}_i^u\right|^3}$

source: Mortenson, Ch 4, p. 142-143

C^2 Continuity at Curve Join Point

- "Full" C^2 continuity at join point requires:
  - Same radius of curvature*
  - Same osculating plane*
  - These conditions hold for curves $\mathbf{p}(u)$ and $\mathbf{r}(u)$ if:

$$\mathbf{p}_i = \mathbf{r}_i$$

$$\mathbf{p}_i^u = \mathbf{r}_i^u$$

$$\mathbf{p}_i^{uu} = \mathbf{r}_i^{uu}$$

* see later slides on topics in differential geometry

source: Mortenson, Ch 12

Piecewise Cubic B-Spline Curve Smoothness at Joint

The effects of multipli-coincident control points and multiple knot values on the continuity at segment joints are worth some further discussion. Here is a summary of the continuity conditions:

1. Control point multiplicity $= 1$: C^2 and G^2 ← familiar situation

2. Control point multiplicity $= 2$: C^2 and G^1, with knots restricted to a reduced convex hull. ← looks incorrect

3. Control point multiplicity $= 3$: C^2 and G^0. The curve interpolates the triple control point, and the segments at each side of the joint are straight lines. ← looks incorrect

4. Control point multiplicity $= 4$: G^2 and G^0. The curve segments on both sides of the joint are straight lines and interpolate the control points on both sides. ← looks incorrect

5. Knot multiplicity $= 1$: C^2 and G^2 ← familiar situation

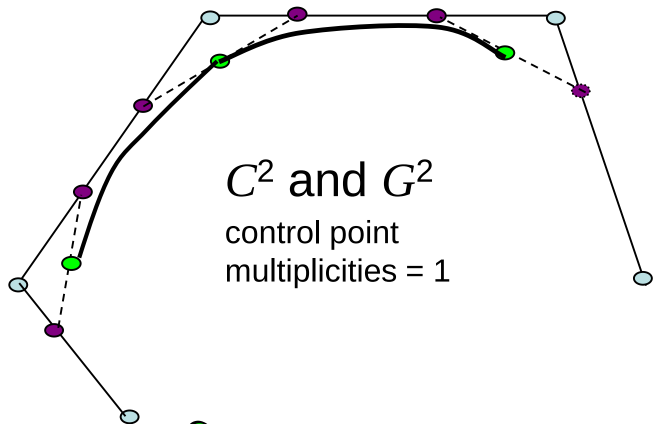
6. Knot multiplicity $= 2$: C^1 and G^1, with knots restricted to a reduced convex hull. ← curvature discontinuity

7. Knot multiplicity $= 3$: C^0 and G^0. The curve interpolates the control point. Curve segment shapes at the joint are free and not constrained to straight lines.

8. Knot multiplicity $= 4$: The curve is discontinuous, ending on one control point and resuming at the next. The shapes of the curve segments adjacent to the discontinuity are unconstrained.

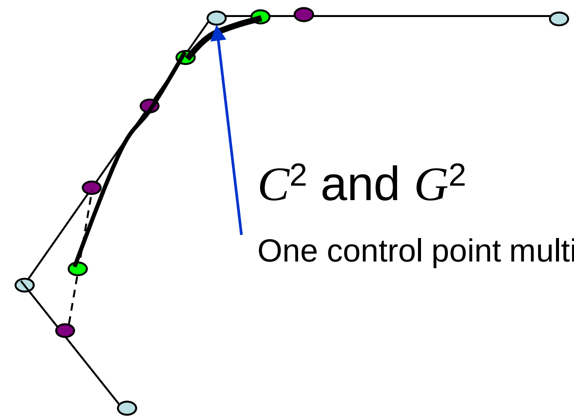
source: Mortenson, Ch 5

Control Point Multiplicity Effect on Uniform Cubic B-Spline Joint



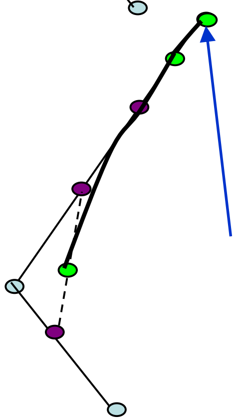
C^2 and G^2

control point multiplicities = 1



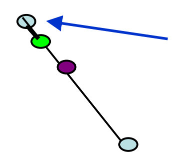
C^2 and G^2

One control point multiplicity = 2



C^2 and G^2

One control point multiplicity = 3



C^0 and G^0

One control point multiplicity = 4

One curve segment degenerates into a single point. Other curve segment is a straight line. First derivatives at join point are equal but vanish. Second derivatives at join point are equal but vanish.

$$\mathbf{p}^u(u) = \frac{1}{2}(-u^2 + 2u - 1)\mathbf{p}_0 + \frac{1}{2}(3u^2 - 4u)\mathbf{p}_1 + \frac{1}{2}(-3u^2 + 2u + 1)\mathbf{p}_2 + \frac{1}{2}u^2\mathbf{p}_3$$

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

Knot Multiplicity Effect on Non-uniform B-Spline

- If a knot has multiplicity r, then the B-spline curve of degree n has smoothness C^{n-r} at that knot.

source: Farin, Ch 8

A Few Differential Geometry Topics Related to Continuity

Local Curve Topics

- Principal Vectors
  - Tangent
  - Normal
  - Binormal
- Osculating Plane and Circle
- Frenet Frame
- Curvature
- Torsion
- Revisiting the Definition of Geometric Continuity

source: Ch 12 Mortenson

Intrinsic Definition
(adapted from earlier lecture)

- No reliance on external frame of reference
- Requires 2 equations as functions of arc length* s:

**length measured along the curve*

1) Curvature: $\frac{1}{\rho} = f(s)$
2) Torsion: $\tau = g(s)$

Torsion (in 3D) measures how much curve deviates from a plane curve.

- For plane curves, alternatively:

$$\frac{1}{\rho} = \frac{d\theta}{ds}$$

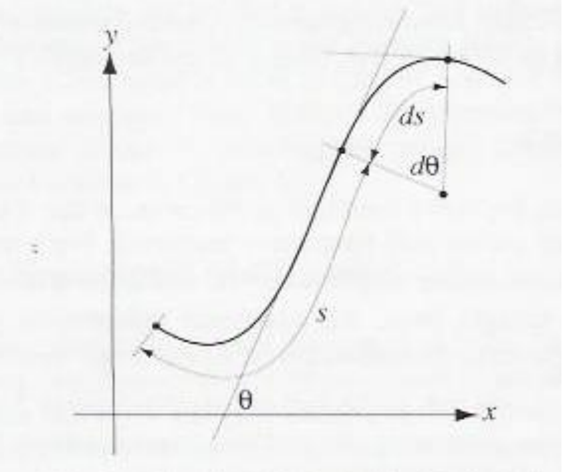


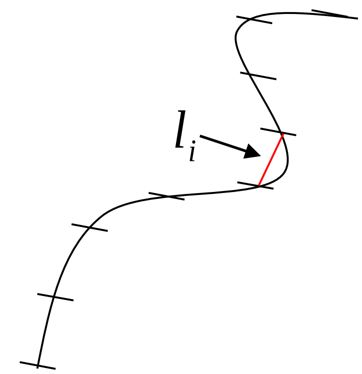
Figure 2.1 Intrinsic definition of a curve.

Treated in more detail in Chapter 12 of Mortenson and Chapter 10 of Farin.

source: Mortenson

Calculating Arc Length

- Approximation: For parametric interval u_1 to u_2, subdivide curve segment into n equal pieces.



$$L = \sum_{i=1}^{n} l_i \qquad \text{where} \qquad l_i = \sqrt{(\mathbf{p}_i - \mathbf{p}_{i-1}) \bullet (\mathbf{p}_i - \mathbf{p}_{i-1})}$$

$$\text{using} \qquad \mathbf{p} \bullet \mathbf{p} = |\mathbf{p}|^2$$

$$L = \int_{u_1}^{u_2} \sqrt{\mathbf{p}^u \bullet \mathbf{p}^u}\, du$$ is more accurate.

source: Mortenson, p. 401

Tangent

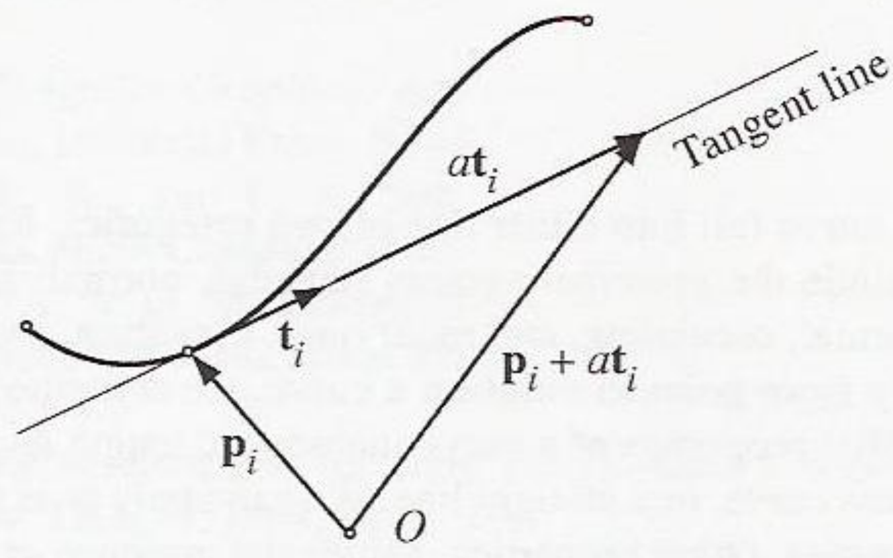


Figure 12.1 Tangent vector and line.

unit tangent vector: $$\mathbf{t}_i = \frac{\mathbf{p}_i^u}{\left|\mathbf{p}_i^u\right|}$$

source: Mortenson, p. 388

Normal Plane

- Plane through $\mathbf{p}_i$ perpendicular to $\mathbf{t}_i$

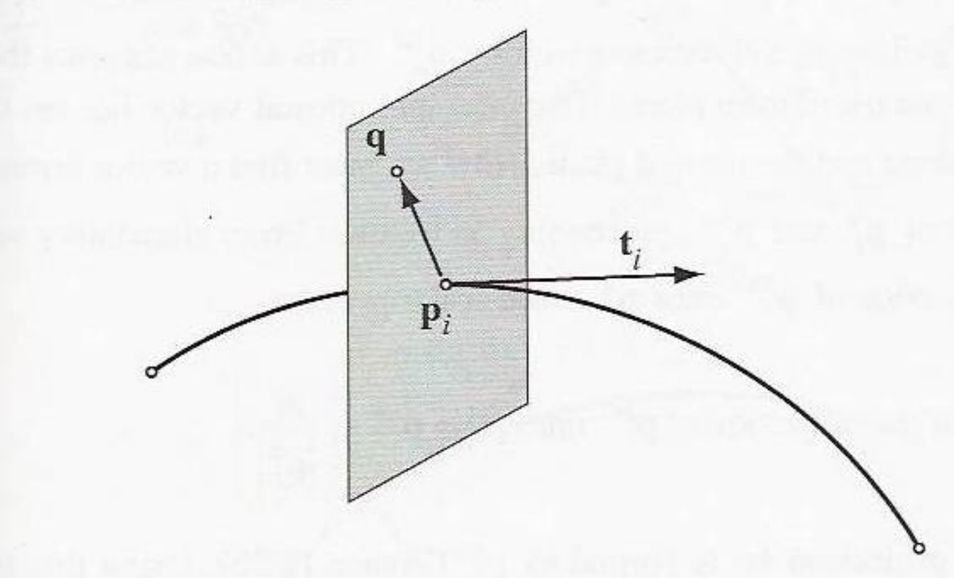


Figure 12.2 Normal plane.

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

$$x_i^u x + y_i^u y + z_i^u z - (x_i x_i^u + y_i y_i^u + z_i z_i^u) = 0$$

source: Mortenson, p. 388-389

Principal Normal Vector and Line

Moving slightly along curve in neighborhood of $\mathbf{p}_i$ causes tangent vector to move in direction specified by: $\mathbf{p}_i^{uu}$

Principal normal vector is on intersection of normal plane with (osculating) plane shown in (a).

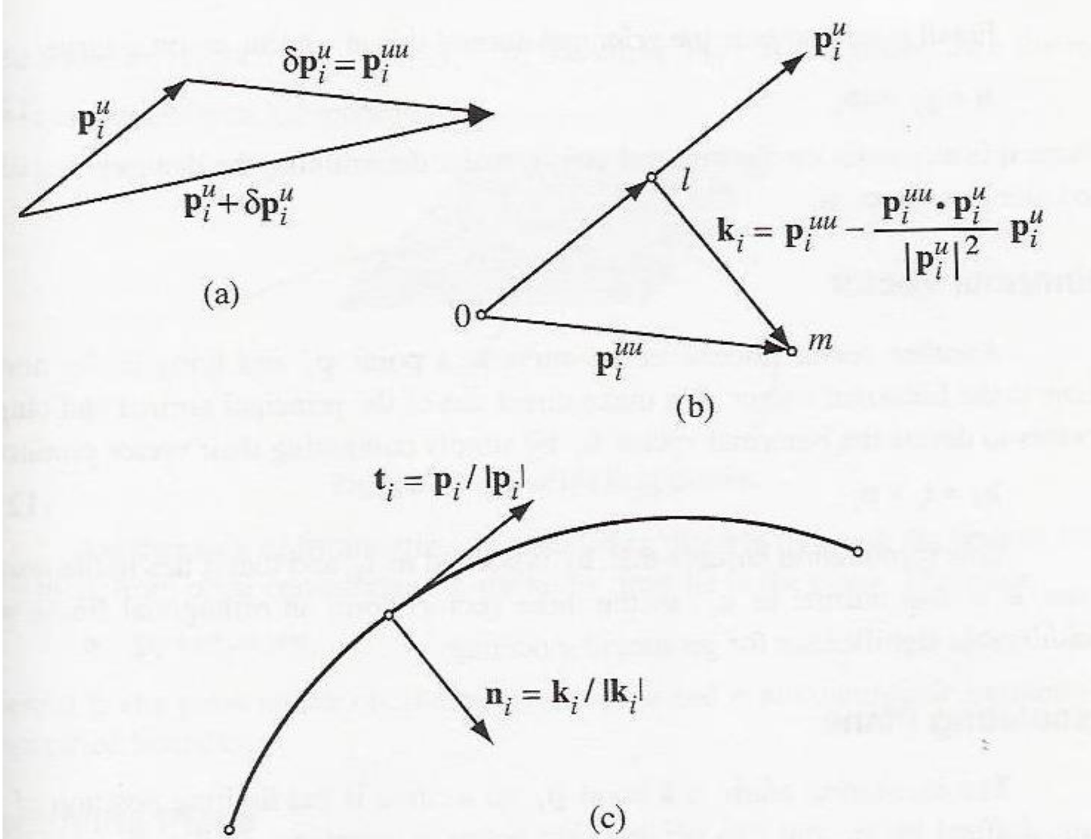


Figure 12.3 Principal normal vector and line.

Use dot product to find projection of $\mathbf{p}_i^{uu}$ onto $\mathbf{p}_i^u$

Binormal vector

$$\mathbf{b}_i = \mathbf{t}_i \times \mathbf{n}_i$$

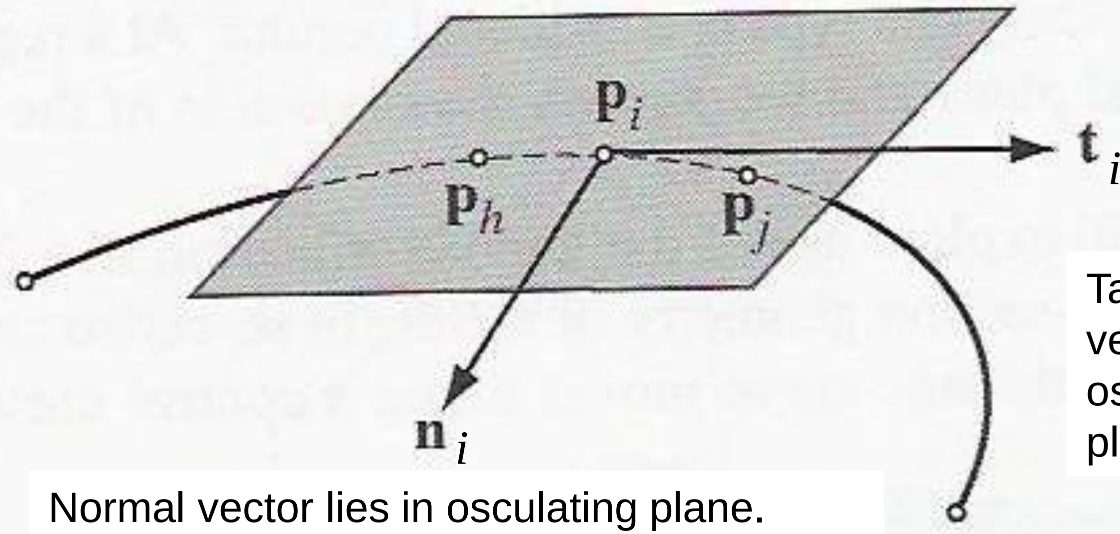
lies in normal plane.

source: Mortenson, p. 389-391

Osculating Plane

Limiting position of plane defined by $\mathbf{p}_i$ and two neighboring points $\mathbf{p}_j$ and $\mathbf{p}_h$ on the curve as these neighboring points independently approach $\mathbf{p}_i$.

Note: $\mathbf{p}_i$, $\mathbf{p}_j$ and $\mathbf{p}_h$ cannot be collinear.



Tangent vector lies in osculating plane.

Normal vector lies in osculating plane.

Figure 12.4 Osculating plane.

$$\begin{vmatrix} x - x_i & x_i^u & x_i^{uu} \\ y - y_i & y_i^u & y_i^{uu} \\ z - z_i & z_i^u & z_i^{uu} \end{vmatrix} = 0$$

source: Mortenson, p. 392-393

Frenet Frame

Rectifying plane at $\mathbf{p}_i$ is the plane through $\mathbf{p}_i$ and perpendicular to the principal normal $\mathbf{n}_i$:

$$(\mathbf{q}-\mathbf{p}_i)\bullet\mathbf{n}_i = 0$$

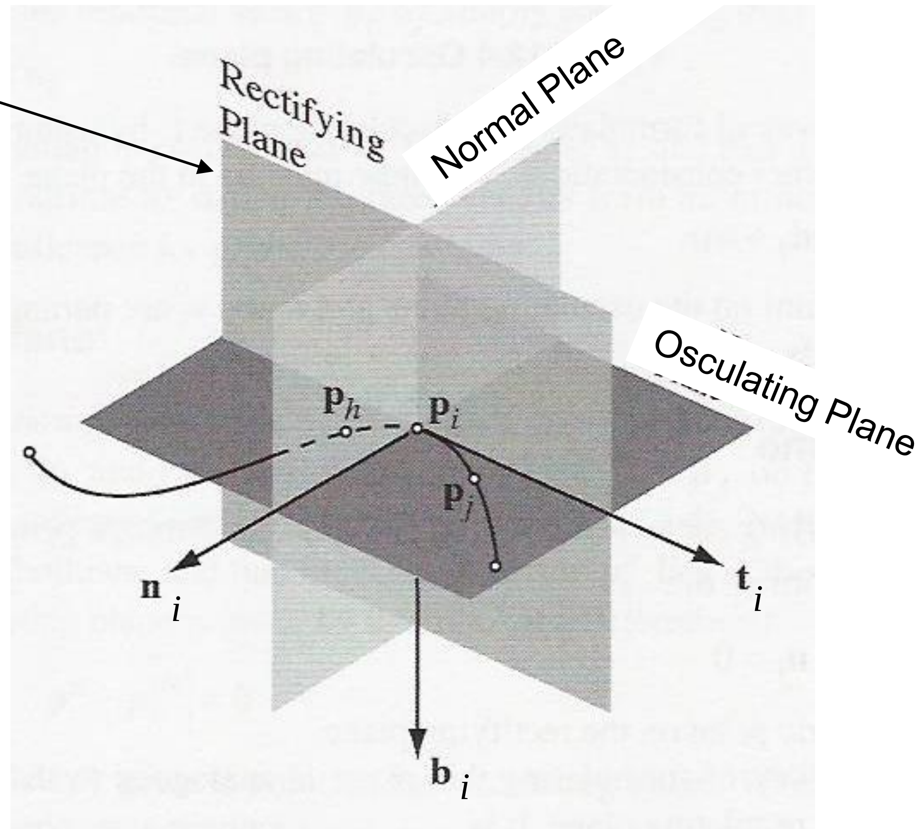


Figure 12.5 The moving trihedron.

Note changes to Mortenson's figure 12.5.

source: Mortenson, p. 393-394

Curvature

- Radius of curvature is ρ_i and curvature at point $\mathbf{p}_i$ on a curve is:

$$\kappa_i = \frac{1}{\rho_i} = \frac{\left|\mathbf{p}_i^u \times \mathbf{p}_i^{uu}\right|}{\left|\mathbf{p}_i^u\right|^3}$$

Recall that vector $\mathbf{p}_i^{uu}$ lies in the osculating plane.

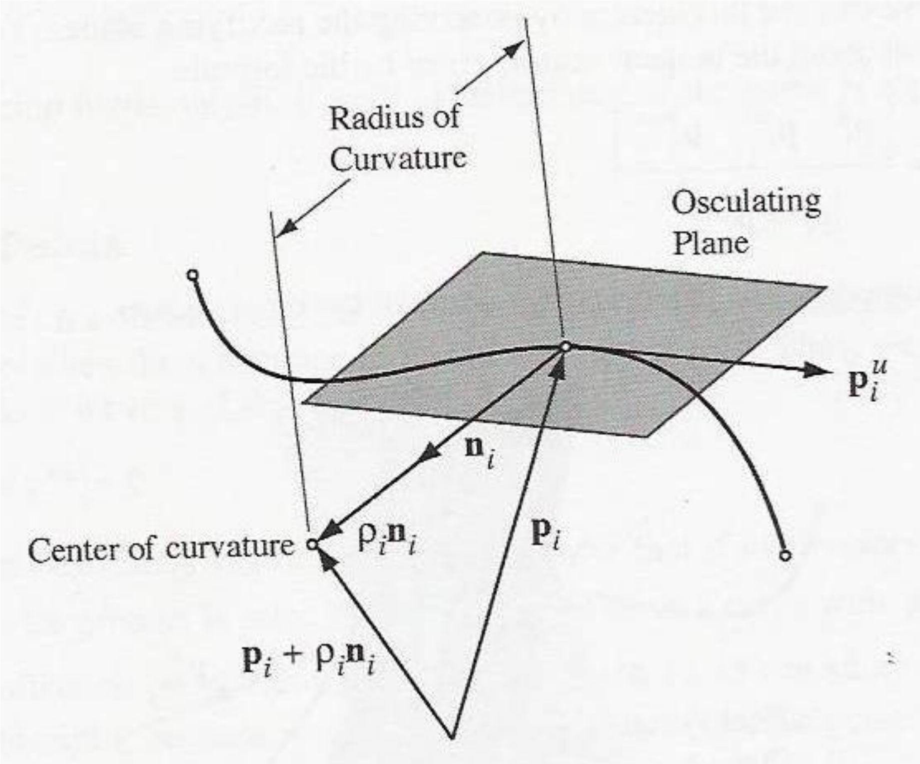


Figure 12.6 Curvature.

Curvature of a planar curve in x, y plane:

$$\frac{1}{\rho} = \frac{d^2y/dx^2}{\left[1+(dy/dx)^2\right]^{3/2}}$$

Curvature is *intrinsic* and does not change with a change of parameterization.

source: Mortenson, p. 394-397

Torsion

- Torsion at $\mathbf{p}_i$ is limit of ratio of angle between binormal at $\mathbf{p}_i$ and binormal at neighboring point $\mathbf{p}_h$ to arc-length of curve between $\mathbf{p}_h$ and $\mathbf{p}_i$, as $\mathbf{p}_h$ approaches $\mathbf{p}_i$ along the curve.

$$\tau_i = \frac{\left[\mathbf{p}_i^u \quad \mathbf{p}_i^{uu} \quad \mathbf{p}_i^{uuu}\right]}{\left|\mathbf{p}_i^u \times \mathbf{p}_i^{uu}\right|^2} = \frac{\mathbf{p}_i^u \bullet \left(\mathbf{p}_i^{uu} \times \mathbf{p}_i^{uuu}\right)}{\left|\mathbf{p}_i^u \times \mathbf{p}_i^{uu}\right|^2}$$

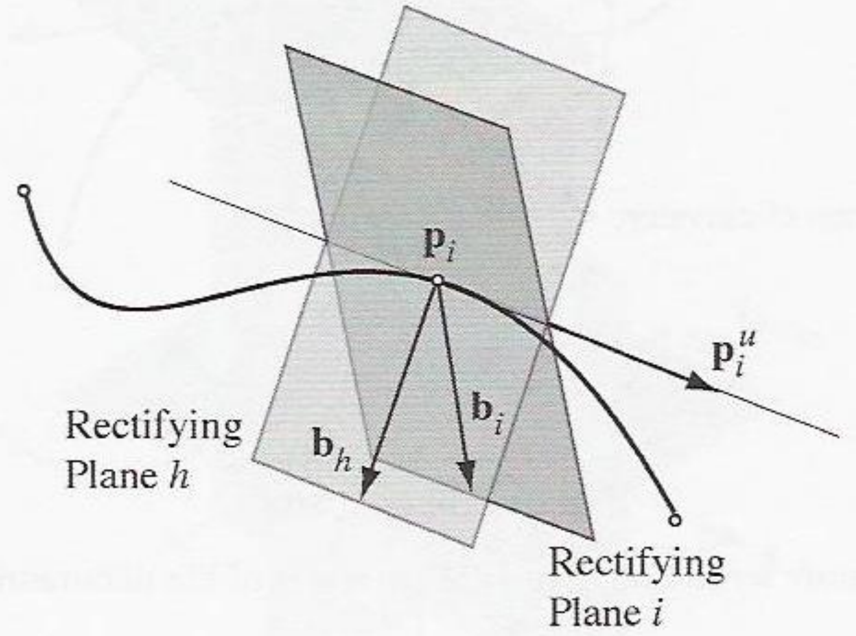


Figure 12.7 Torsion.

<u>Torsion</u> is *intrinsic* and does not change with a change of parameterization.

source: Mortenson, p. 394-397

Reparameterization Relationship

- Curve has G^r continuity if an arc-length reparameterization exists after which it has C^r continuity.
- This is equivalent to these 2 conditions:
  - C^{r-2} continuity of curvature
  - C^{r-3} continuity of torsion

Local properties <u>torsion</u> and <u>curvature</u> are *intrinsic* and *uniquely* determine a curve.

source: Farin, Ch 10, p.189 & Ch 11, p. 200

Local Surface Topics

- Fundamental Forms
- Tangent Plane
- Principal Curvature
- Osculating Paraboloid

source: Ch 12 Mortenson

Local Properties of a Surface
Fundamental Forms

- Given parametric surface $\mathbf{p}(u,w)$
- Form I: $$d\mathbf{p} \bullet d\mathbf{p} = E du^2 + 2F du dw + G dw^2$$
$$E = \mathbf{p}^u \bullet \mathbf{p}^u \qquad F = \mathbf{p}^u \bullet \mathbf{p}^w \qquad G = \mathbf{p}^w \bullet \mathbf{p}^w$$
- Form II: $$-d\mathbf{p}(u,w) \bullet d\mathbf{n}(u,w) = L du^2 + 2M du dw + N dw^2$$
$$L = \mathbf{p}^{uu} \bullet \mathbf{n} \qquad M = \mathbf{p}^{uw} \bullet \mathbf{n} \qquad N = \mathbf{p}^{ww} \bullet \mathbf{n} \qquad \mathbf{n} = \frac{\mathbf{p}^u \times \mathbf{p}^w}{\left|\mathbf{p}^u \times \mathbf{p}^w\right|}$$
- Useful for calculating arc length of a curve on a surface, surface area, curvature, etc.

Local properties first and second fundamental forms are *intrinsic* and *uniquely* determine a surface.

source: Mortenson, p. 404-405

Local Properties of a Surface
Tangent Plane

$$\mathbf{p}^u = \partial \mathbf{p}(u,w) / \partial u$$

$$\mathbf{p}^w = \partial \mathbf{p}(u,w) / \partial w$$

$$(\mathbf{q} - \mathbf{p}) \bullet (\mathbf{p}^u \times \mathbf{p}^w) = 0$$

$$\begin{vmatrix} x - x_i & x_i^u & x_i^w \\ y - y_i & y_i^u & y_i^w \\ z - z_i & z_i^u & z_i^w \end{vmatrix} = 0$$

q $\mathbf{p}(u_i,w_i)$ components of parametric tangent vectors $\mathbf{p}^u(u_i,w_i)$ and $\mathbf{p}^w(u_i,w_i)$

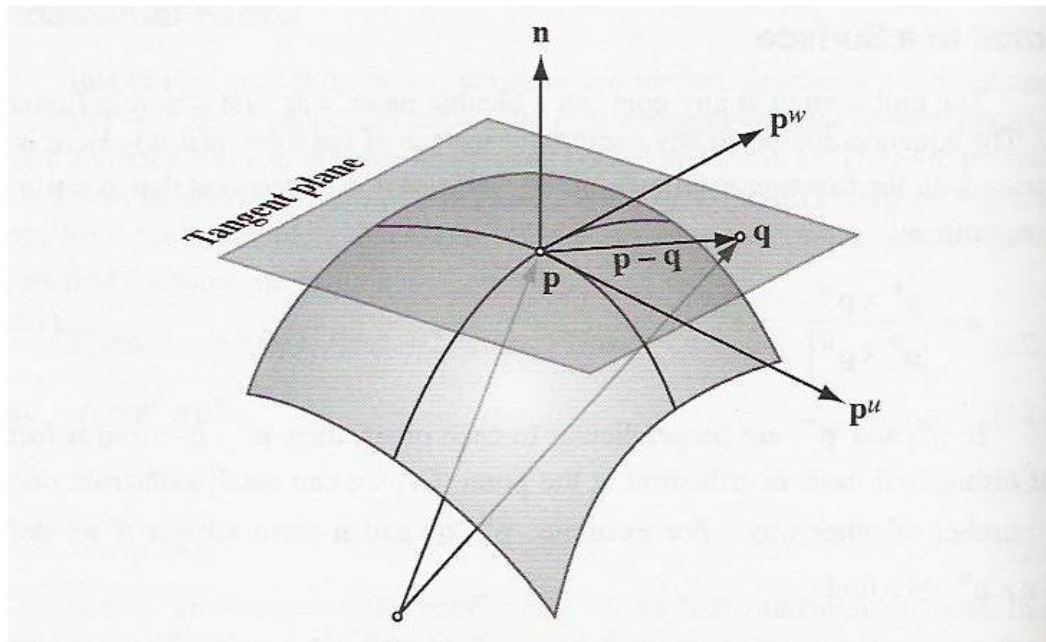


Figure 12.9 Tangent plane.

source: Mortenson, p. 406

Local Properties of a Surface
Principal Curvature

- Derive <u>curvature</u> of all parametric curves C on parametric surface S passing through point **p** with same tangent line l at **p**.

normal curvature vector $\mathbf{k}_n$ = projection of curvature vector **k** onto **n** at **p**

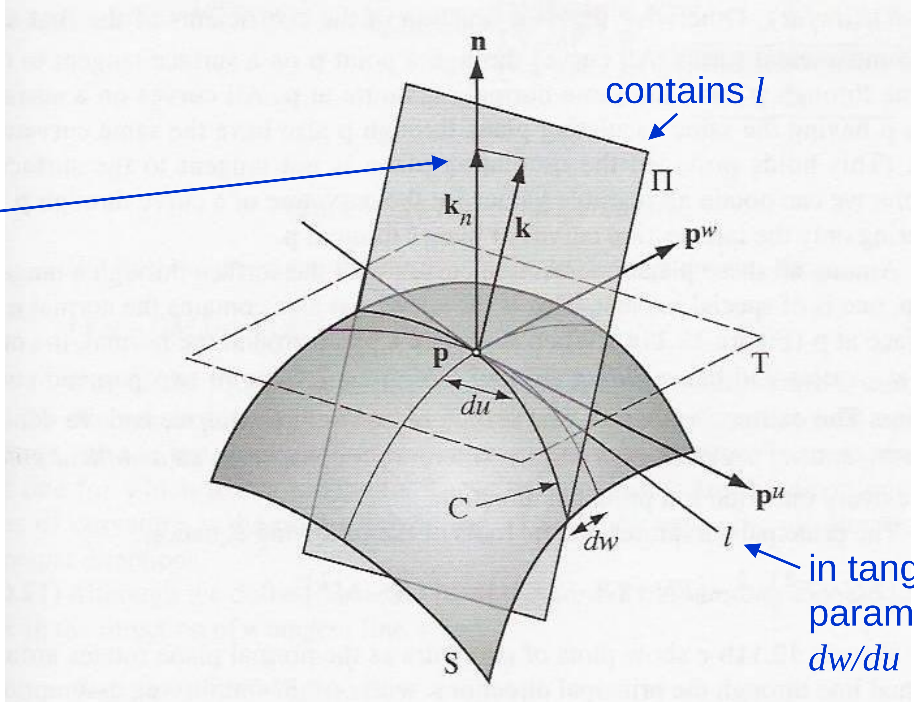


Figure 12.10 Normal curvature.

in tangent plane with parametric direction dw/du

$$\mathbf{k}_n = (\mathbf{k} \bullet \mathbf{n})\mathbf{n}$$

normal curvature: $\kappa_n = \mathbf{k} \bullet \mathbf{n}$

$$\kappa_n = \frac{L(du/dt)^2 + 2M(du/dt)(dw/dt) + N(dw/dt)^2}{E(du/dt)^2 + 2F(du/dt)(dw/dt) + G(dw/dt)^2}$$

source: Mortenson, p. 407-410

Local Properties of a Surface
Principal Curvature (continued)

Rotating a plane around the normal changes the curvature κ_n.

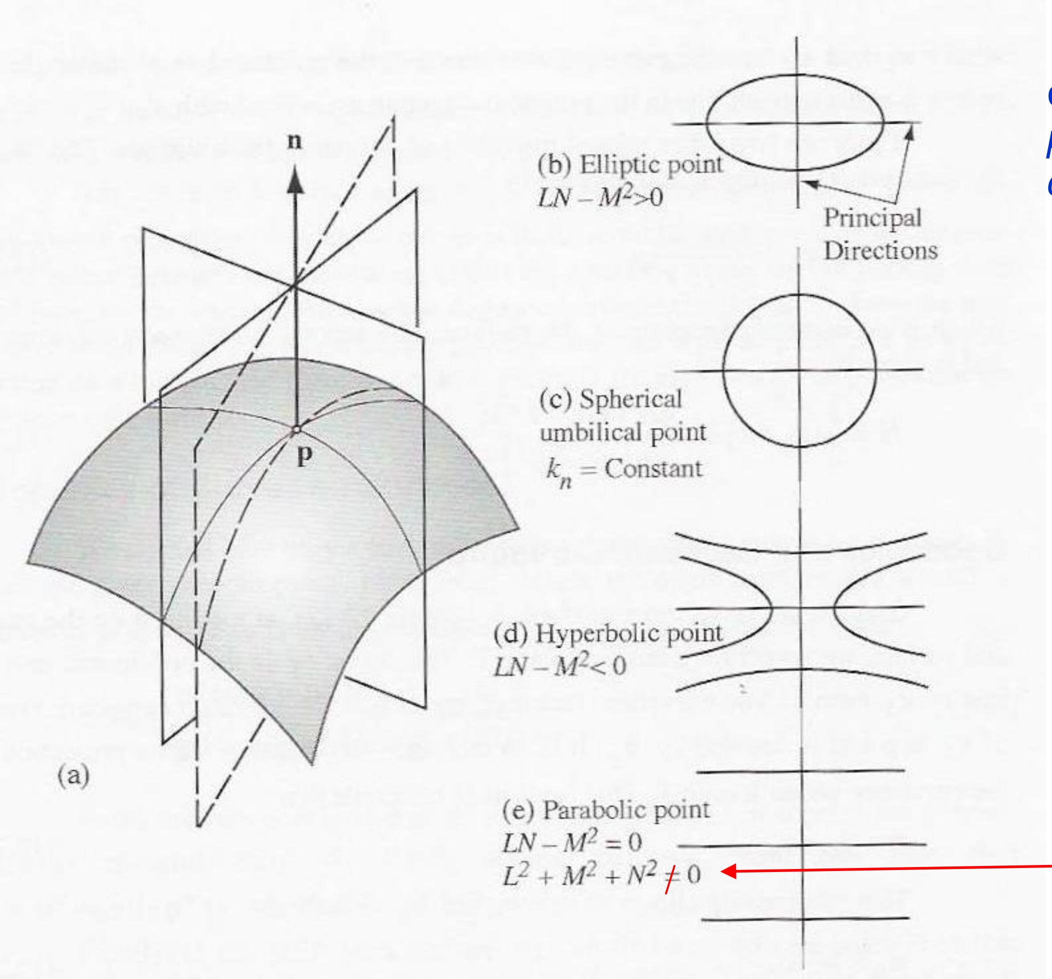


Figure 12.11 Principal curvature.

curvature extrema: *principal normal curvatures*

typographical error?

source: Mortenson, p. 407-410

Local Properties of a Surface

Osculating Paraboloid

Second fundamental form helps to measure distance of surface from tangent plane.

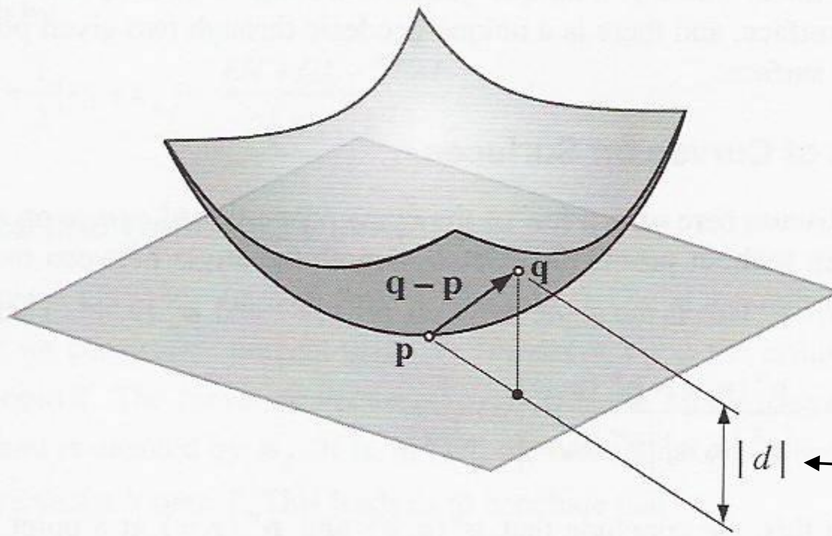


Figure 12.13 Osculating paraboloid at a point on a surface.

$$|d| = (\mathbf{q} - \mathbf{p}) \bullet \mathbf{n}$$

As **q** approaches **p**: $$d = f\underbrace{\left[\frac{1}{2}\left(L\,du^2 + 2M\,du\,dw + N\,dw^2\right)\right]}_{\text{Osculating Paraboloid}}$$

source: Mortenson, p. 412

Local Properties of a Surface
Local Surface Characterization

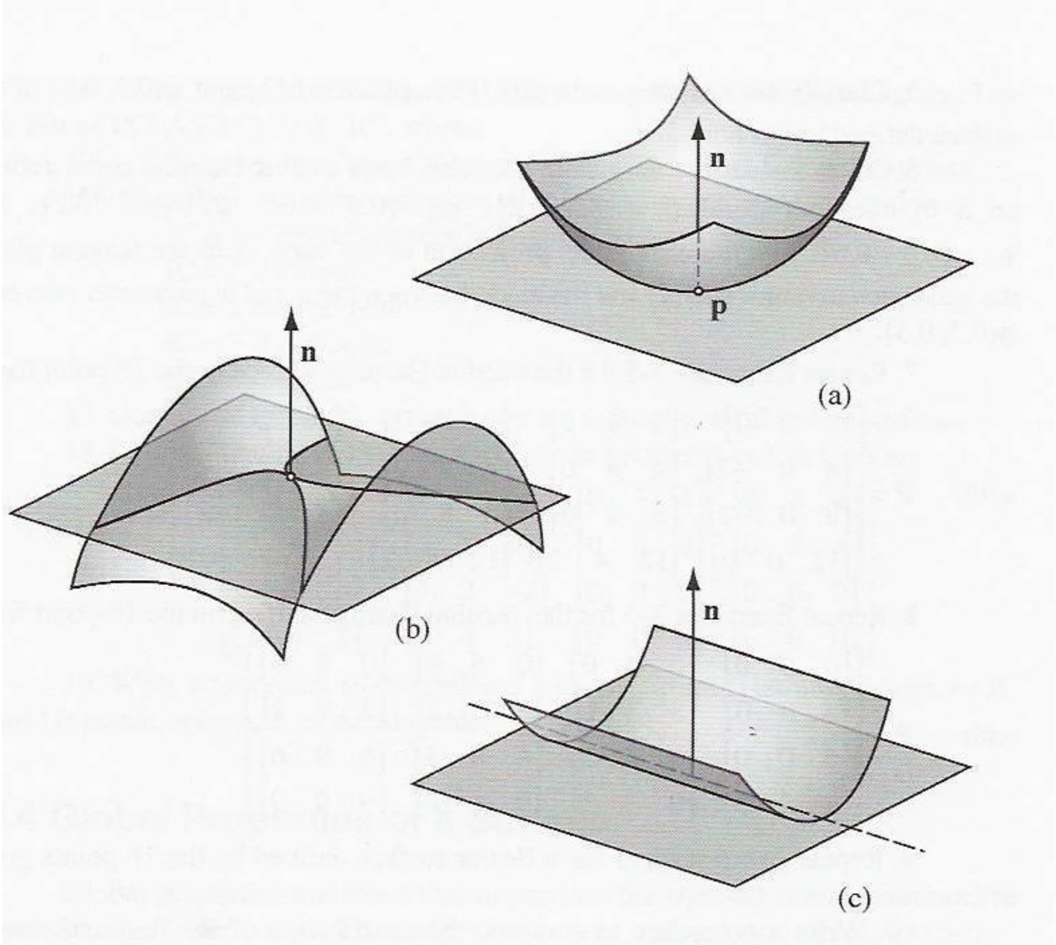
source: Mortenson, p. 412-413

$b) LN - M^2 < 0$

Hyperbolic Point:
“saddle point”

$L = M = N = 0$

Planar Point
(not shown)



$a) LN - M^2 > 0$

Elliptic Point:
locally convex

$c) LN - M^2 = 0$

$L^2 + M^2 + N^2 \neq 0$

typographical
error?

Parabolic Point:
single line in
tangent plane along
which d =0