Chapter 4

Curve Modeling and Surface Modeling

Curve Modeling

Introduction

- A curve is an infinitely large set of points. Each point has two neighbors except endpoints. Curves can be broadly classified into three categories: **explicit, implicit,** and **parametric curves**.
- ***Implicit Curves***
  - Implicit curve representations ***define the set of points on a curve by employing a procedure*** that can test to see if a point in on the curve.
  - Usually, an implicit curve is defined by an implicit function of the form $f(x, y)=0$; in two dimensions and $f(x, y, z)=0$; in three dimensions
- ***Explicit Curves***
  - ***Explicit function:*** to express the idea that we have one dependent variable on the left-hand side of an equation, and all the independent variables and constants on the right-hand side of the equation.
  - For example, ***the equation of a line is: y=mx+b***
  - A mathematical function ***y = f(x) can be plotted as a curve***. Such a function is the explicit representation of the curve.
  - For each value of x, only a single value of y is normally computed by the function.

Introduction

- **Parametric *Curves***
  - Parametric equations ***are generators of paths through space.***
  - Curve descriptions ***provide a mapping from a free parameter to the set of points on the curve.*** The parametric form of a curve defines a function that assigns positions to values of the free parameter.

  •We write the coordinate ***pair (x, y) as a pair of functions (r cos(t), r sin(t)).***

  • ***Separate equation for each spatial variable***

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

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

Introduction

Why Parametric Cubic Curves?

- Parametric representations are the most common in computer graphics.
- A curve is approximated by a piecewise polynomial curve.
- ***Cubic polynomials are most often used because:***

  1) Lower-degree polynomials offer too little flexibility in controlling the shape of the curve.

  2) Higher-degree polynomials can introduce unwanted wiggles and also require more computation.

Polynomial functions

- ***Linear:*** $$f(t) = at + b$$

- ***Quadratic:*** $$f(t) = at^2 + bt + c$$

- ***Cubic:*** $$f(t) = at^3 + bt^2 + ct + d$$

Parametric Representation of a curve

The cubic polynomial that define a curve segment is

$Q(t) = [x(t) \;\; y(t) \;\; z(t)]$

Where

$$x(t) = a_x t^3 + b_x t^2 + c_x t + d_x$$

$$y(t) = a_y t^3 + b_y t^2 + c_y t + d_y$$

$$z(t) = a_z t^3 + b_z t^2 + c_z t + d_z$$

$$0 \leq t \leq 1$$

To represent this in matrix form, we have

$$T = [t^3 \;\; t^2 \;\; t \;\; 1]$$

$$C = \begin{pmatrix} a_x & a_y & a_z \\ b_x & b_y & b_z \\ c_x & c_y & c_z \\ d_x & d_y & d_z \end{pmatrix}$$

So, we can write

$$Q(t) = T.C$$

Spline Representation

- A spline is a ***flexible strip used to produce a smooth curve*** through a designated **set** of points.
- In computer graphics, the term spline curve refers to any composite curve formed with polynomial section satisfying specified continuity conditions at the boundary of the pieces.
- Splines are used in graphics applications ***to design curve and surface shapes***, to digitize drawings for computer storage, ***and to specify animation paths for the objects or the camera in a scene.***
- Typical CAD applications for splines include the design of automobile bodies, aircraft and spacecraft surfaces, and ship hulls.
- We specify a ***spline curve by giving a set of coordinate positions***, called control points, which indicates the general shape of the curve These, ***control points are then fitted with piecewise continuous parametric polynomial functions***.

Spline Representation

When polynomial sections ***are fitted so that the curve passes through each control point***, as in Figure 1**,** the resulting curve is ***said to interpolate the set of control points.***

▪ On the other hand, when the ***polynomials are fitted to the general control-point path without necessarily passing through any control point***, the resulting curve is ***said to approximate the set of control points*** (Figure 2**).**

Figure 1: A set of six control points interpolated with piecewise Continuous polynomial sections.

Figure 2: A set of six control points approximated with piecewise Continuous polynomial sections.

Parametric Continuity Conditions

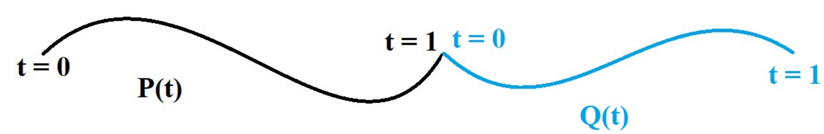
- To ensure a smooth transition from one section **of** a piecewise parametric **curve** to the next, can impose various **continuity conditions** at the connection points.
- If each ***section of a spline is described with a set of parametric coordinate*** functions of the form

$$x=x(u),\ y=y(u),\ z=z(u)\ u_1 \le u \le u_2$$

- Set parametric continuity ***by matching the parametric derivatives of adjoining curve sections*** at their common boundary.

Zero-order Parametric Continuity (C^0): If two curve ***segments join together.***

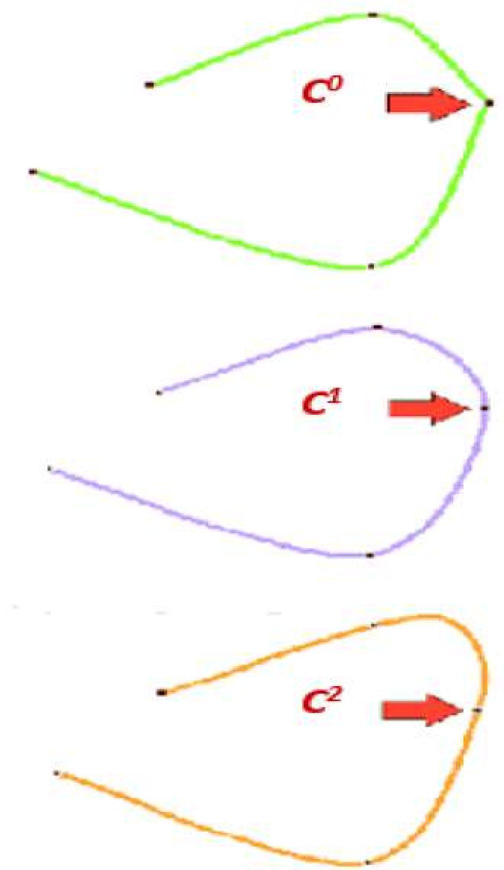
P(1) = Q(0)



First-order Parametric Continuity (C^1): First derivatives equal. If the ***directions and magnitudes*** of the two segments' tangent vectors are ***equal at a join point.*** ***P'(1) = Q'(0)***

Second-order Parametric Continuity (C2):First and second derivatives are equal. The first and second parametric derivatives of the two curve sections are the same at the intersection. If t is taken to be time, this implies that the acceleration is continuous. ***P"(1) = Q"(0)***

Parametric Continuity Conditions



Geometric Continuity Conditions

- An alternative methods for joining ***two successive curve sections*** is to specify conditions ***for geometric continuity:*** only require ***parametric derivatives of the two sections to be proportional to each other at their common boundary*** instead of equal to each other.
-
  - **Zero order geometric (G^0):** Same as C^0

    $$C_1(1) = C_2(0)$$

  - **First order geometric (G^1):** The parametric first derivatives are proportional at the intersection of two successive sections.

    $$C'_1(1) = \alpha C'_2(0)$$

  - **Second order geometric (G^2):** Both the first and second parametric derivatives of the two curve sections are proportional at their boundary,

    $$C''_1(1) = \alpha C''_2(0)$$

Spline Specification

There are three equivalent ***methods for specifying a particular spline representation:***

1) We can state the ***set of boundary conditions that are imposed on the spline***; or

2) We can ***state the matrix that characterizes the spline***; or

3) We can ***state the set of blending functions (or basis functions)*** that determine how specified geometric constraints on the curve are combined to calculate positions along the curve path.

We have the following ***parametric cubic polynomial*** representation for the ***x coordinate along the path of a spline section:***

$$x(t) = \sum_{i=0}^{n} a_i t^n \qquad ; 0 \leq t \leq 1$$

y(t) and z(t) are similar and each is handled independently.

i.e.

$$\begin{aligned} x(t) &= a_3t^3 + a_2t^2 + a_1t + a_0 \\ y(t) &= b_3t^3 + b_2t^2 + b_1t + b_0 \\ z(t) &= c_3t^3 + c_2t^2 + c_1t + c_0 \end{aligned}$$

AND

$$\begin{aligned} x'(t) &= 3a_3t^2 + 2a_2t + a_1 \\ y(t) &= 3b_3t^2 + 2b_2t + b_1 \\ z(t) &= 3c_3t^2 + 2c_2t + c_1 \end{aligned}$$

Spline Specification

A ***compact version of the parametric equations*** can be

$$x(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix}$$

$$x(t) = T \cdot A$$

Similarly, we can write

$y(t) = T . B$

$z(t) = T . C$

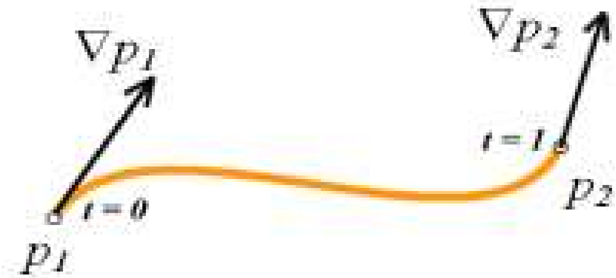
Each dimension is treated independently, so we can deal with curves in any number of dimensions.

Hermite Spline Curve

- Hermite is an interpolating ***piecewise cubic polynomial with a specified tangent at each control point.***
- If **P(u)** represents a ***parametric cubic point function*** for the curve section between control ***points*** P_k ***and*** P_{k+1}, then the ***boundary conditions*** that define ***this Hermite curve*** section are

$$P(0) = P_k$$
$$P(1) = P_{k+1}$$
$$P'(0) = DP_k$$
$$P'(1) = DP_{k+1} \text{ ....eqn(1)}$$

with DP_k and DP_{k+1} specifying the values for the parametric derivatives (slope of the curve) a t control points P_k and P_{k+1} respectively.



Hermite Specification

- We can write the Hermite curve section as

$$P(t) = at^3 + bt^2 + ct + d; \; 0 \le t \le 1 \text{ ....eqn.(2)}$$

where the **x** component of P is $x(t) = a_x t^3 + b_x t^2 + c_x t + d_x$, and similarly for the y and z components.

- The ***Matrix equivalent of eqn(2) is***

Hermite Spline Curve

- The ***Matrix equivalent of eqn(2) is***

$$P(t) = [\,t^3 \;\; t^2 \; t \; 1\,]\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} \text{.....eqn(3)}$$

- The ***derivative of the point function*** can be expressed as

$$P'(t) = [\,3t^2 \;\; 2t \; 1 \; 0\,]\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}$$

- Substituting ***endpoint values 0 and 1 for parameter t*** Into the previous two equations, we can express the ***Hermite boundary conditions eqn(1)***

$P_k = P(0) = d$ $\quad\quad DP_k = P'(0) = c$

$P_{k+1} = P(1) = a + b + c + d$ $\quad\quad DP_{k+1} = P'(1) = 3a + 2b + c$

Hermite Spline Curve

- *In the matrix form:*

$$\begin{bmatrix} P_k \\ P_{k+1} \\ DP_k \\ DP_{k+1} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}$$

- *And the solution is:*

$d = P_k$

$c = DP_k$

$b = -3\,P_k - 2\,DP_k + 3\,P_{k+1} - DP_{k+1}$

$a = 2\,P_k + DP_k - 2\,P_{k+1} + DP_{k+1}$

Hermite Spline Curve

$$\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = M_H \cdot \begin{bmatrix} P_k \\ P_{k+1} \\ DP_k \\ DP_{k+1} \end{bmatrix} \quad \text{Where } M_H \text{ is Hermite matrix.}$$

- ***The eqn(3) can be written as:*** $\mathbf{P(t) = [\, t^3 \;\; t^2 \; t \; 1]}\; M_H \cdot \begin{bmatrix} P_k \\ P_{k+1} \\ DP_k \\ DP_{k+1} \end{bmatrix}$

$$\mathbf{P(t) = [\, t^3 \;\; t^2 \; t \; 1]} \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_k \\ P_{k+1} \\ DP_k \\ DP_{k+1} \end{bmatrix}$$

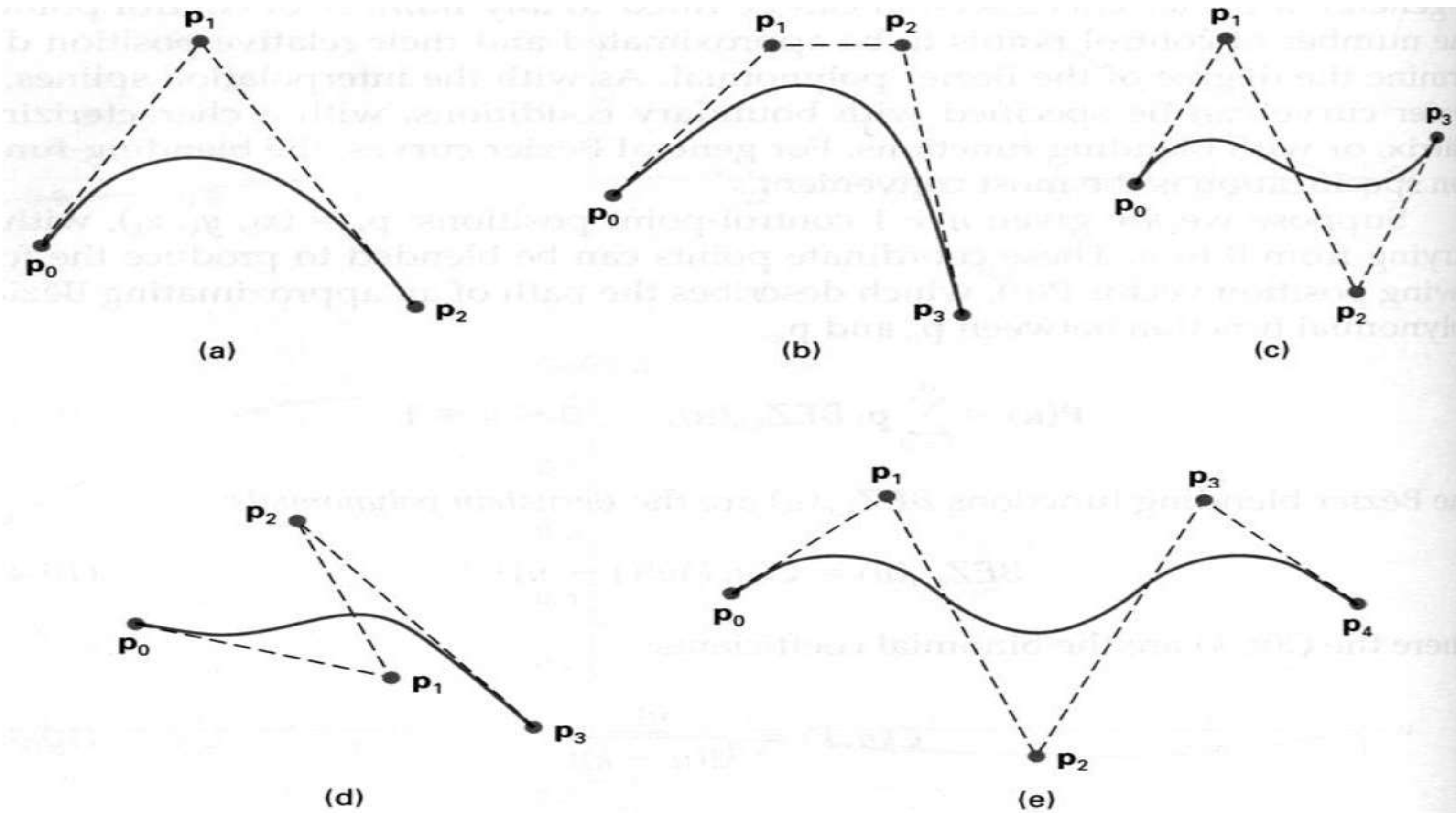
Bezier Curve

- This spline approximation method was developed by the French engineer Pierre Bezier for use in the design of Renault automobile bodies.

- Bezier splines have a number of properties that make them highly useful and convenient for curve and surface design and they are also easy to implement. For these reasons, Bezier splines are widely available in various CAD systems.
- In general, a Bezier curve section can be fitted to any number of control points.
- The number of control points to be approximated and their relative position determine the degree of the Bezier polynomial.

- ***A Bezier curve is a polynomial of degree one less than the designated number of control points.***

Bezier Curve



- ***Figure:*** ***Examples of two-dimensional Bezier curves generated from three, four, and five control points. Dashed lines connect the control-point positions.***

Bezier Curve

- ***Given n+1 control point positions:*** $\mathbf{p}_k = (x_k, y_y, z_k) \qquad 0 \le k \le n$

- These coordinate points can be blended ***to produced the following position vector p(u),*** which describes the path of an approximating Bezier polynomial function between $\mathbf{P_0}$ **and** $\mathbf{P_n}$.

$$\boxed{p(u) = \sum_{k=0}^{n} \mathbf{p}_k B_{k,n}(u), \quad 0 \le u \le 1}$$

- ***The Bezier blending functions are the Bernstein polynomials:***

$$\boxed{B_{k,n}(u) = C(n,k)u^k(1-u)^{n-k}} \qquad \boxed{C(n,k) = \frac{n!}{k!(n-k)!}}$$

Bezier Curve

- Cubic Bezier curves are ***generated with four control points***. ***Bezier polynomial function between P_0 and P_3***

$$\boxed{p(u) = \sum_{k=0}^{3} \mathbf{p}_k B_{k,3}(u), \quad 0 \le u \le 1}$$

- **Where**

$$\boxed{B_{k,3}(u) = C(3,k)u^k(1-u)^{3-k}}$$

$$\boxed{\begin{aligned} B_{0,3}(u) &= (1-u)^3 \\ B_{1,3}(u) &= 3u(1-u)^2 \\ B_{2,3}(u) &= 3u^2(1-u) \\ B_{3,3}(u) &= u^3 \end{aligned}}$$

Bezier Curve

$$p(u) = \sum_{k=0}^{3} \mathbf{p}_k B_{k,3}(u), \quad 0 \le u \le 1$$

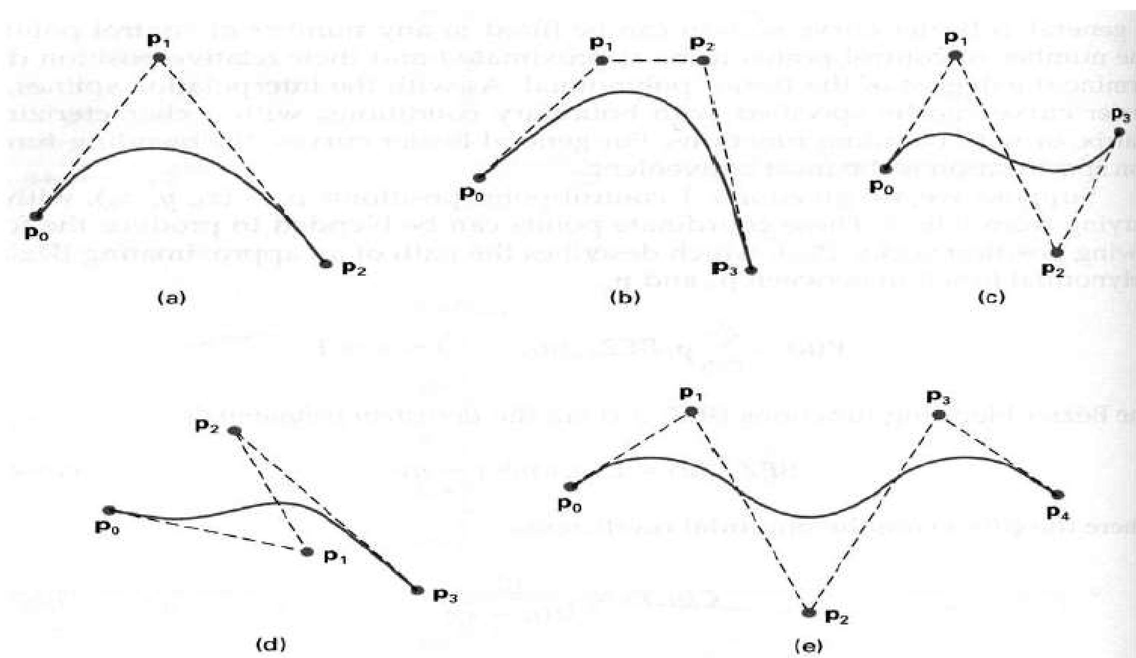
$$P(u) = (1\text{-}u)^3 P_0 + 3u(1\text{-}u)^2 P_1 + 3u^2(1\text{-}u) P_2 + u^3 P_3$$

$$p(u) = \begin{bmatrix} (1-\mathrm{u})^3 & 3\mathrm{u}(1-\mathrm{u})^2 & 3\mathrm{u}^2(1-\mathrm{u}) & u^3 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix}$$

$$P(u) = \begin{bmatrix} 1 & u & u^2 & u^3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix}$$

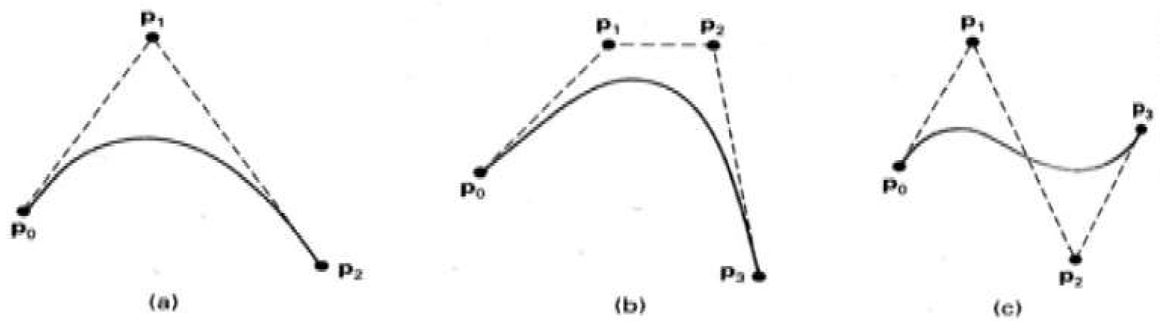
Properties Bezier Curve

1) The ***degree of a Bezier curve*** defined by ***n+1 control points is n***.
2) Bezier curve is that it always passes ***through the first P_0*** *and* ***last control points P_n***; this is the so-called ***endpoint interpolation property.***



Properties Bezier Curve

3) The Bezier curve lies completely in the **convex hull** of the given control points.



4) All basis functions are positive and their sum is always 1.

$$\boxed{\sum_{k=0}^{n} B_{k,n}(u) = 1}$$

1) Find the coordinates at U=0.25, 0.5, and 0.75 with respect to the control points (10, 10), (15, 25), (20, 30), and (25, 5) using Bezier function. Draw your curve with given control points.

5. Mention two important properties of Bezier Curve and find the Bezier Curve which passes through (0,0,0) and (-2,1,1) and is controlled by (7,5,2) and (2,0,1).

Surface Modeling

Introduction

- Graphics scenes ***can contain many different kinds of objects***: trees , flowers, clouds, rocks, water, bricks, wood paneling, rubber, paper, marble, steel, glass, plastic, and cloth, just to mention a few.
- ***No single method can use to describe objects*** that will include all characteristics of these different materials.
- And to produce realistic displays of scenes, we need to use representations that accurately model object characteristics.

Polygon and quadric surfaces provide precise descriptions for simple Euclidean objects ***such as polyhedrons and ellipsoids***;

Spline surfaces are useful for ***designing aircraft wings, gears, and other engineering*** structures with curved surfaces;

Procedural methods, such as fractal constructions and particle systems, allow us to give accurate representations ***for clouds, clumps of grass, and other natural objects;***

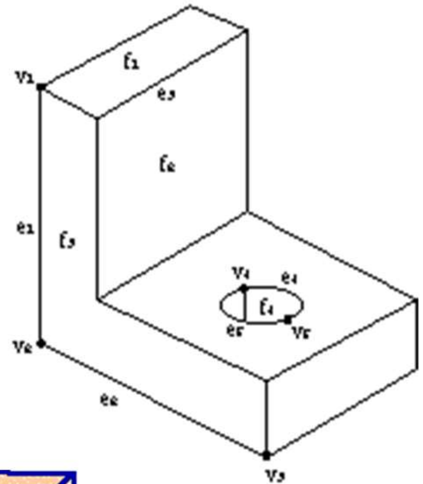
Physically based modeling methods using systems of interacting forces can be used to describe the nonrigid ***behavior of a piece of cloth;***

Octree encodings are used to represent internal features of objects, such as those obtained from ***medical CT images***;

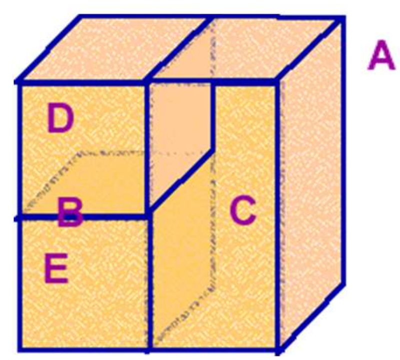
Polygon Surfaces

Objects are represented as a collection of surfaces. 3D object representation is divided into two categories.

- **Boundary Representations (B−reps)** − It describes a 3D object as a set of surfaces that separates the object interior from the environment.

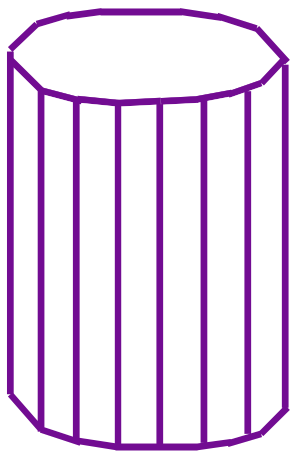


- **Space–partitioning representations** − It is used to describe interior properties, by partitioning the spatial region containing an object into a set of small, non-overlapping, contiguous solids usually cubes



Polygon Surfaces

- The most commonly used ***boundary representation for a 3D graphics object*** is a set of surface polygons that enclose the object interior. ▪Many graphics systems store all object descriptions ***as sets of surface polygons.***
- This ***simplifies and speeds up the surface rendering*** and display of objects, since all surfaces ***are described with linear equations***.
- Realistic ***renderings are produced by interpolating shading patterns*** across the polygon surfaces ***to eliminate or reduce the presence of polygon edge boundaries.***
- Polygon descriptions are often referred to as "***standard graphics objects."***



Polygon Mesh

- A polygon mesh is ***collection of edges, vertices and polygons*** connected ***such that each edge is shared by at most two polygons***.
- An edge connects two vertices and a polygon is a closed sequence of edges. An edge can be ***shared by two polygons*** and a ***vertex is shared by at least two edges.***
- ***High-quality graphics systems*** typically model objects with polygon meshes and ***set up a database of geometric and attribute information*** to facilitate processing of the polygon facets.
- The ***<u>quadrilateral mesh</u>*** **,** which generates a ***mesh of (n -1) by (m - 1)*** quadrilaterals, given the coordinates ***for an n by m array of vertices***.
- **A** quadrilateral mesh ***containing 12 quadrilaterals*** constructed ***from a 5 by 4 input vertex*** array.

Polygon Mesh

- ***Advantages***
  - It can be used to model almost any object.
  - They are easy to represent as a collection of vertices.
  - They are easy to transform.
  - They are easy to draw on computer screen.
- ***Disadvantages***
  - Curved surfaces can only be approximately described.
  - It is difficult to simulate some type of objects like hair or liquid.
- Other type of polygon mesh ***is the triangle strip***. This function ***produces n - 2 connected triangles*** , given the coordinates ***for n vertices.***
- ***Example:*** A triangle strip formed with 11 triangles connecting 13 vertices

Polygon Data Tables

- ***Polygon data tables*** stores information for each polygon (set of vertex coordinates and associated attribute parameters.) in the subsequent processing, display, and manipulation of the objects in a scene.

- Polygon data tables can ***be organized into two groups:***

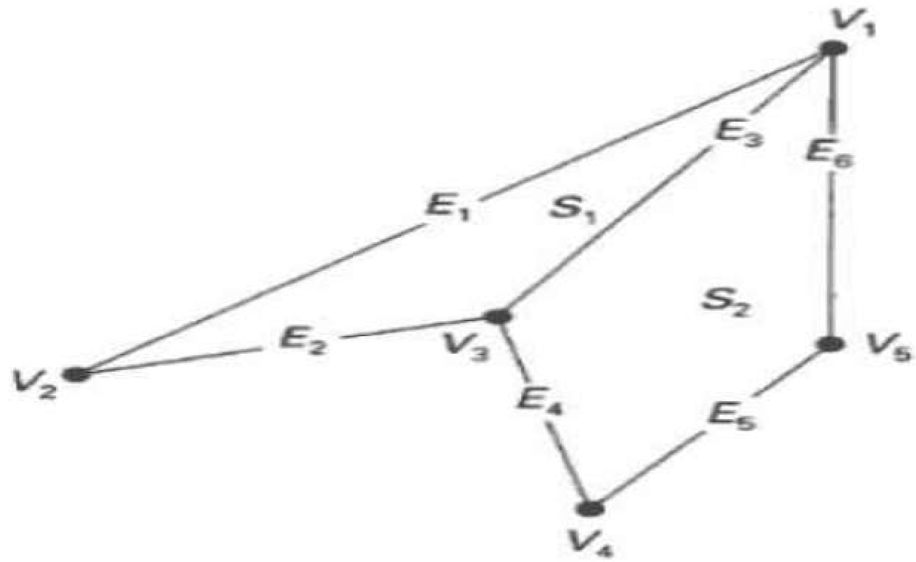
  - ***Geometric data tables:*** It contains vertex coordinates and parameters to identify the spatial orientation of the polygon surfaces.

  - ***Attribute data table:*** it includes parameters specifying the degree of transparency of the object and its surface reflectivity and texture characteristics.

- A geometric data further ***divided into three lists:*** *a vertex table, an edge table, and a polygon table.*

  - ***Vertex table:*** it contains coordinate values for each vertex in the object.
  - ***Edge table:*** it contains pointers back into the vertex table to identify the vertices for each polygon edge.
  - ***Polygon table:*** it contains pointers back into the edge table to identify the edges for each polygon

Polygon Data Tables



VERTEX TABLE	
V_1:	x_1, y_1, z_1
V_2:	x_2, y_2, z_2
V_3:	x_3, y_3, z_3
V_4:	x_4, y_4, z_4
V_5:	x_5, y_5, z_5

EDGE TABLE	
E_1:	V_1, V_2
E_2:	V_2, V_3
E_3:	V_3, V_1
E_4:	V_3, V_4
E_5:	V_4, V_5
E_6:	V_5, V_1

POLYGON-SURFACE TABLE	
S_1:	E_1, E_2, E_3
S_2:	E_3, E_4, E_5, E_6

Polygon Data Tables

- Guidelines ***to generate Error Free tables:***
  1) Every vertex is listed as an endpoint for at least two edges,
  2) Every edge is part of at least one polygon,
  3) Every polygon is closed,
  4) Each polygon has at least one shared edge, and
  5) If the edge table contains pointers to polygons, every edge referenced by a polygon pointer has a reciprocal pointer back to the polygon.
-

Plane Equation

- Plane equation method is another method for representation the polygon surface for 3D object. The information about the spatial orientation of object is described by its individual surface, which is obtained by the vertex co-ordinates and the equation of each surface.
- The ***equation for a plane surface*** can be expressed in the form: $Ax + By + Cz + D = 0$

  Where (x, y, *z) is any point on* the plane, and the *coefficients A, B, C, and D* are constants describing, spatial properties of the plane.

- The values of ***A, B, C, D can be obtained by solving*** a set of ***three plane equations*** using co-ordinate values of 3 non collinear points on the plane.
- Let ***(x1, y1, z1), (x2, y3, z2) and (x3, y3, z3) are three such points on the plane***, then,
- ***Plane Equation***

$$Ax1 + By1 + Cz1 + D = 0$$

$$Ax2 + By2 + Cz2 + D = 0$$

$$Ax3 + By3 + Cz3 + D = 0$$

Plane Equation

- The solution of these equations can be obtained ***in determinant from using Cramer's rule as:***

$$A = \begin{vmatrix} 1 & y_1 & z_1 \\ 1 & y_2 & z_2 \\ 1 & y_3 & z_3 \end{vmatrix} \qquad B = \begin{vmatrix} x_1 & 1 & z_1 \\ x_2 & 1 & z_2 \\ x_3 & 1 & z_3 \end{vmatrix} \qquad C = \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} \qquad D = -\begin{vmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{vmatrix}$$

- For any points (x, y, z)

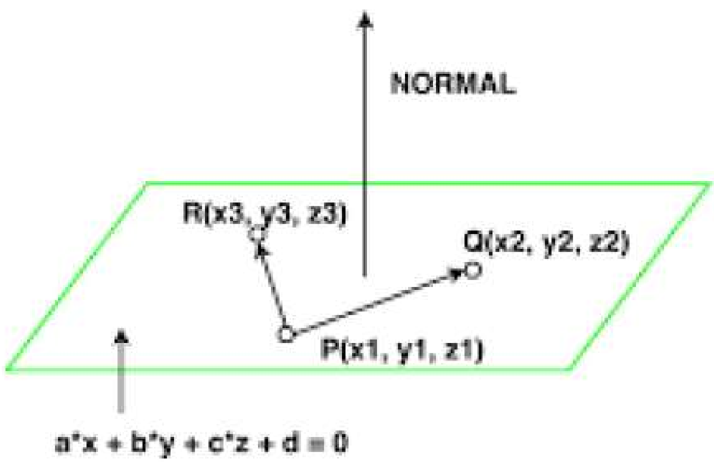
  If $Ax + By + Cz + D \neq 0$, then (x, y, z) is not on the plane.

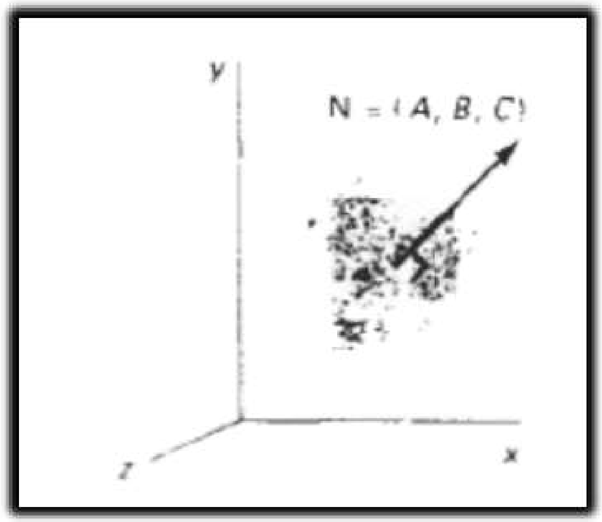
  If $Ax + By + Cz + D < 0$, then (x, y, z) is inside the plane i. e. invisible side

  If $Ax + By + Cz + D > 0$, then (x, y, z) is lies outside the plane

Normal Vector of a Plane

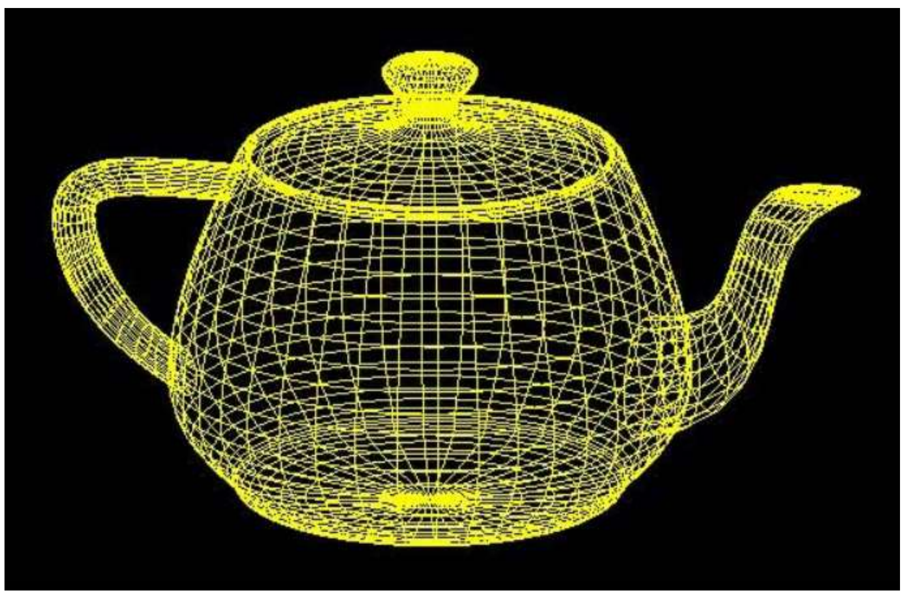
- Orientation of a plane surface in space can be described with the normal vector to the plane
- This surface normal vector has Cartesian components ***(A, B,*** C), where parameters ***A, B,*** and ***C*** are the plane coefficients.





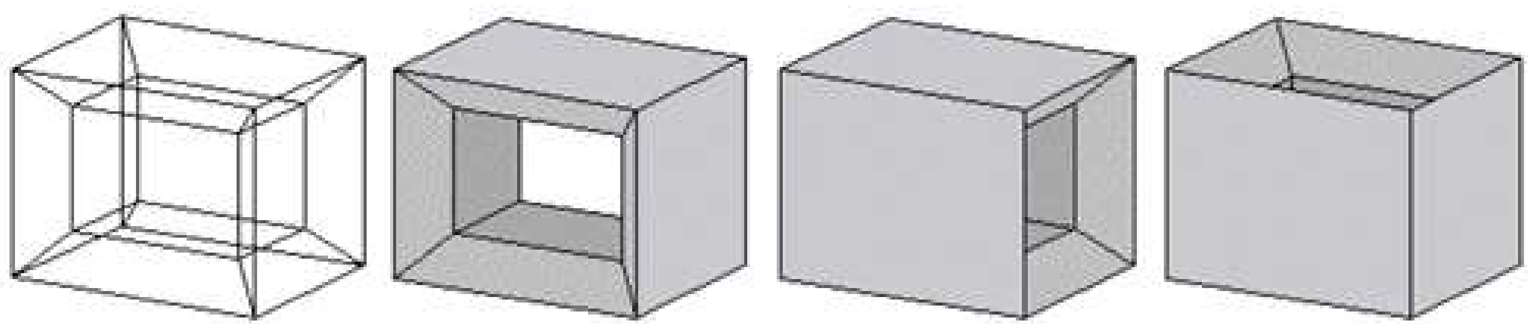
Wireframe Representation

- A **wire-frame model**, is a visual representation of a three dimensional (3D) physical object used in 3D computer graphics.
- It is created ***by specifying each edge of the physical object*** where two mathematically continuous smooth surfaces meet, or by connecting an object's constituent vertices ***using (straight) lines or curves.***
- A wireframe representation is a 3-D ***line drawing of an object showing only the edges*** without any side surface in between. A frame constructed from thin wires ***representing the edges and projected lines and curves***.



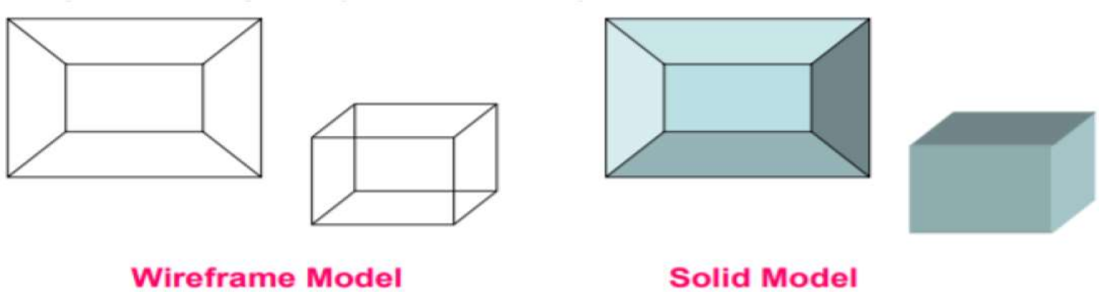
Wireframe Representation

- Furthermore, the wire-frame representation is an ambiguous technique for representing an object, as it does not define explicitly the enclosed surfaces.
- Usually, there may be more than one possible interpretation of the same wireframe. Wire frames can often be interpreted as different solid objects or as different orientations of the same object.



Solid Modeling

- ▪ A ***wireframe representation*** of an object is done ***using edges (lines curves) and vertices.*** Surface representation then is the logical evolution using faces (surfaces), edges and vertices.
- ▪ In this sequence of developments, ***the solid modeling uses topological information in addition to the geometrical information*** to represent the object unambiguously and completely.
- Solid modeling is based on *complete, valid and unambiguous* geometric representation of physical object.
    - ***Complete:*** points in space can be classified.(inside/ outside)
    - ***Valid:*** vertices, edges, faces are connected properly.
    - ***Unambiguous:*** there can only be one interpretation of object
- Solid model consist of ***geometric and topological data***.
    - ***Geometry:*** The graphical information of dimension, length, angle, area and transformations.
    - ***Topology:*** The invisible information about the connectivity, neighborhood, associatively etc.
-



6. Represent the following surfaces by polygon table method and find the normal of surface S1. [2+5]