Computer Graphics

Unit 5 – Part 2

–By Manjula. S

Animation

Introduction

•**Computer animation** is the process used for generating animated images (moving images) using computer graphics.

• **Animators** are artists who specialize in the creation of animation.

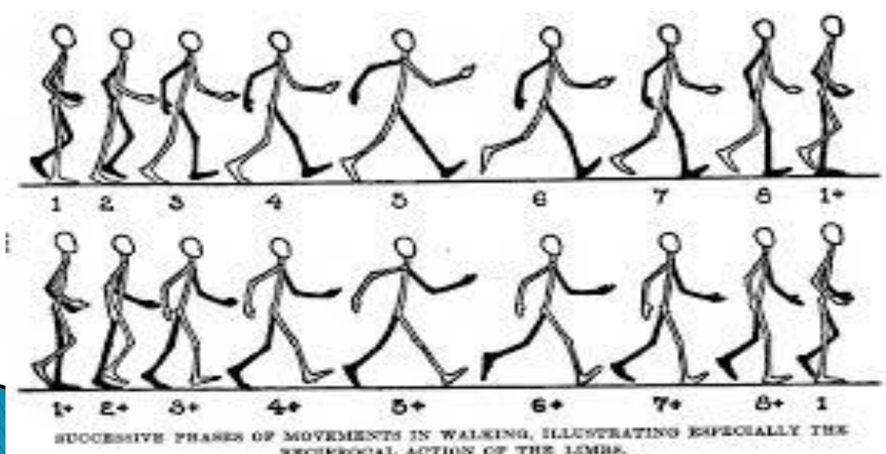
•From Latin **animātiō**, "the act of bringing to life"; from *animō* ("to animate" or "give life to") and *-ātiō* ("the act of").

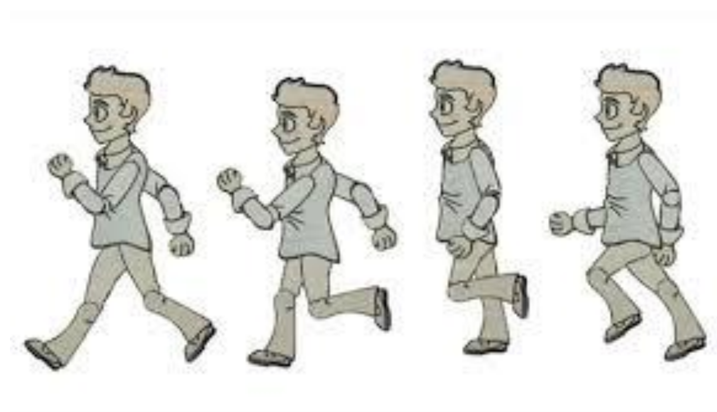
2D ANIMATION

3D ANIMATION

Animation

- It is defined as the act of making something came alive.
- In animation, a series of images rapidly changed to create an illusion of movement.
- Usage of animation: Artistic purpose, story telling, displaying data





Types of Animation

- Cel Animation
- Computer Animation
- Kinematics
- Morphing

Types of Animation

- Cel Animation:
  -A traditional form of animation used in the production of cartoons or animated movies where each frame of the scene is drawn by hand.
  -A full-length feature film produced using Cel animation would often require a million or more drawings to complete.

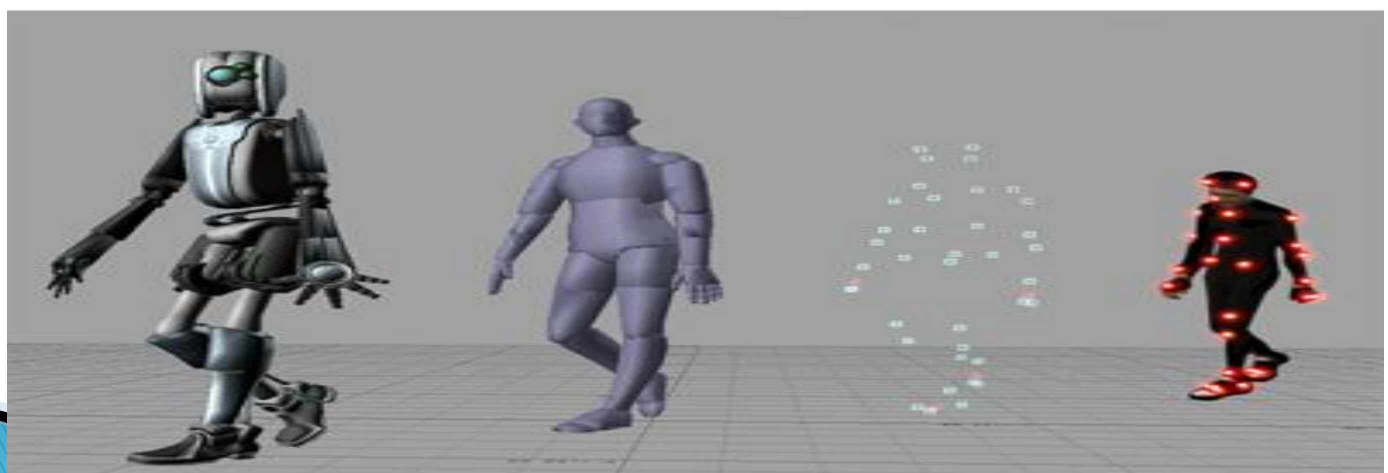
- Computer Animation:
  -Subset of both computer graphics and animation technologies.
  - It is the creation of moving images(animation) using computer technology.

Types of Animation

- <u>Kinematics</u>

–It is the study of the movement and motion of structures that have joints, such as walking man.

–such type of animation are usually used in the areas like mechanics etc



Types of Animation

- Morphing

  –It is popular effect in which one image transforms into another.

  –the morphed images were built at a rate of 8 frames per second with each transition taking a total of 4 seconds.

Example of Image Morphing (a) Original image of Hillary Clinton (b) Morphed image (c) Original image of Donald Trump

Animation File Formats

- Director *.dir
- Animation Pro *.fli, *.fle
- 3D studio Max *.max
- Supercard & Director *.pics
- Compuserve *.gif
- Flash *.fla, *.swf

Software Used

3D Studio Max, Flash, AnimationPro

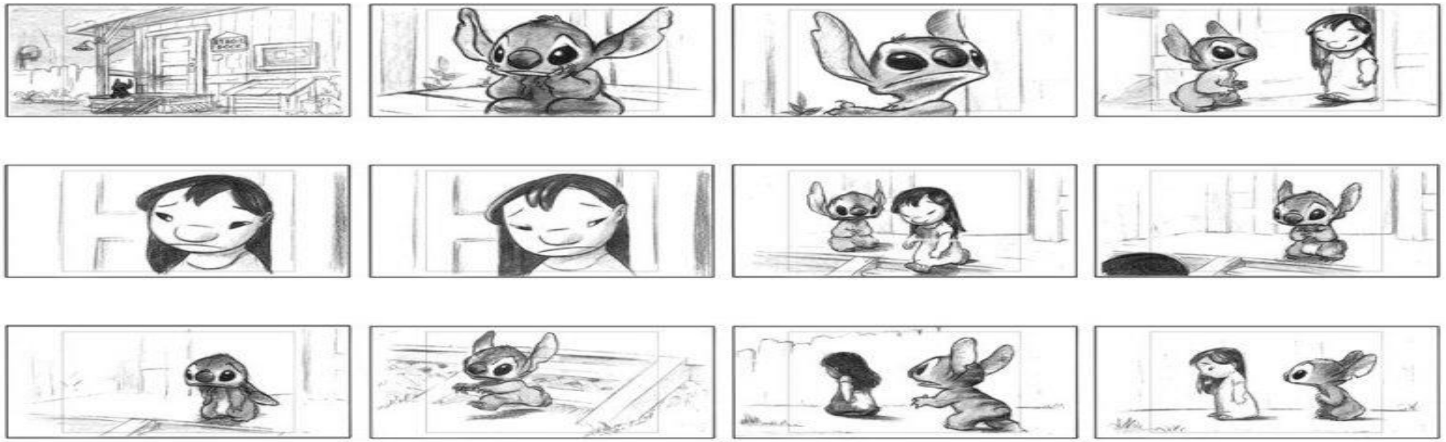
Designing an Animation

- There are 4 steps in designing an animation sequence:

1. Storyboard layout
2. Object definition
3. Key frame specification
4. Generation of in-between frames

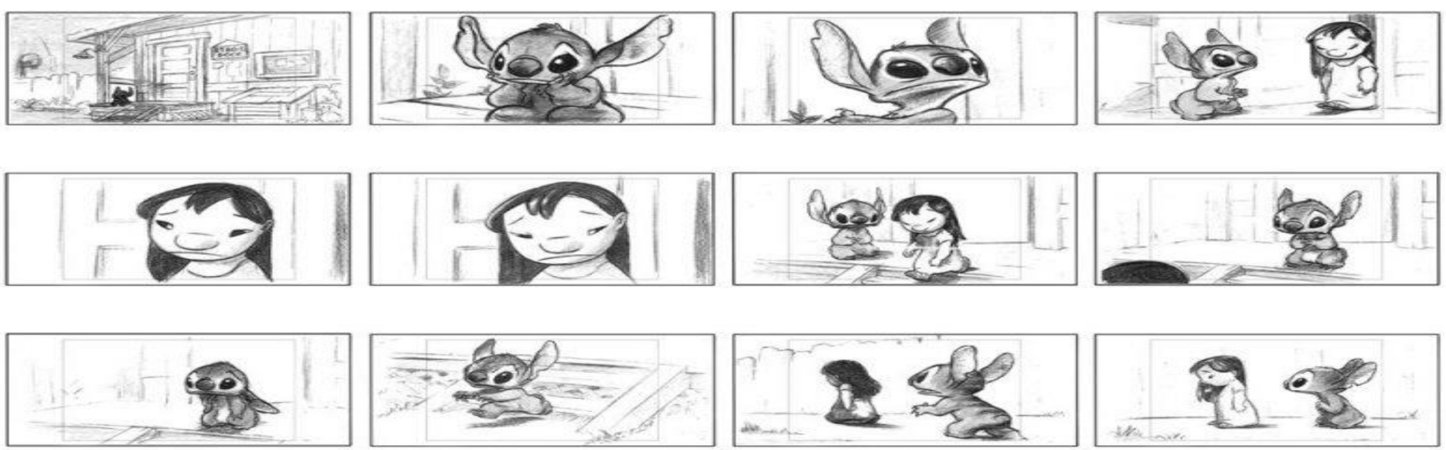
1. Story Board layout

- It is the outline of an action . It defines the motion sequences as a set of basic that are to take place.
- Depending on the type of animation to be produced, the storyboard could be consist of a set of rough sketches or it could be a list of basic ideas for motion.



2. object Definition

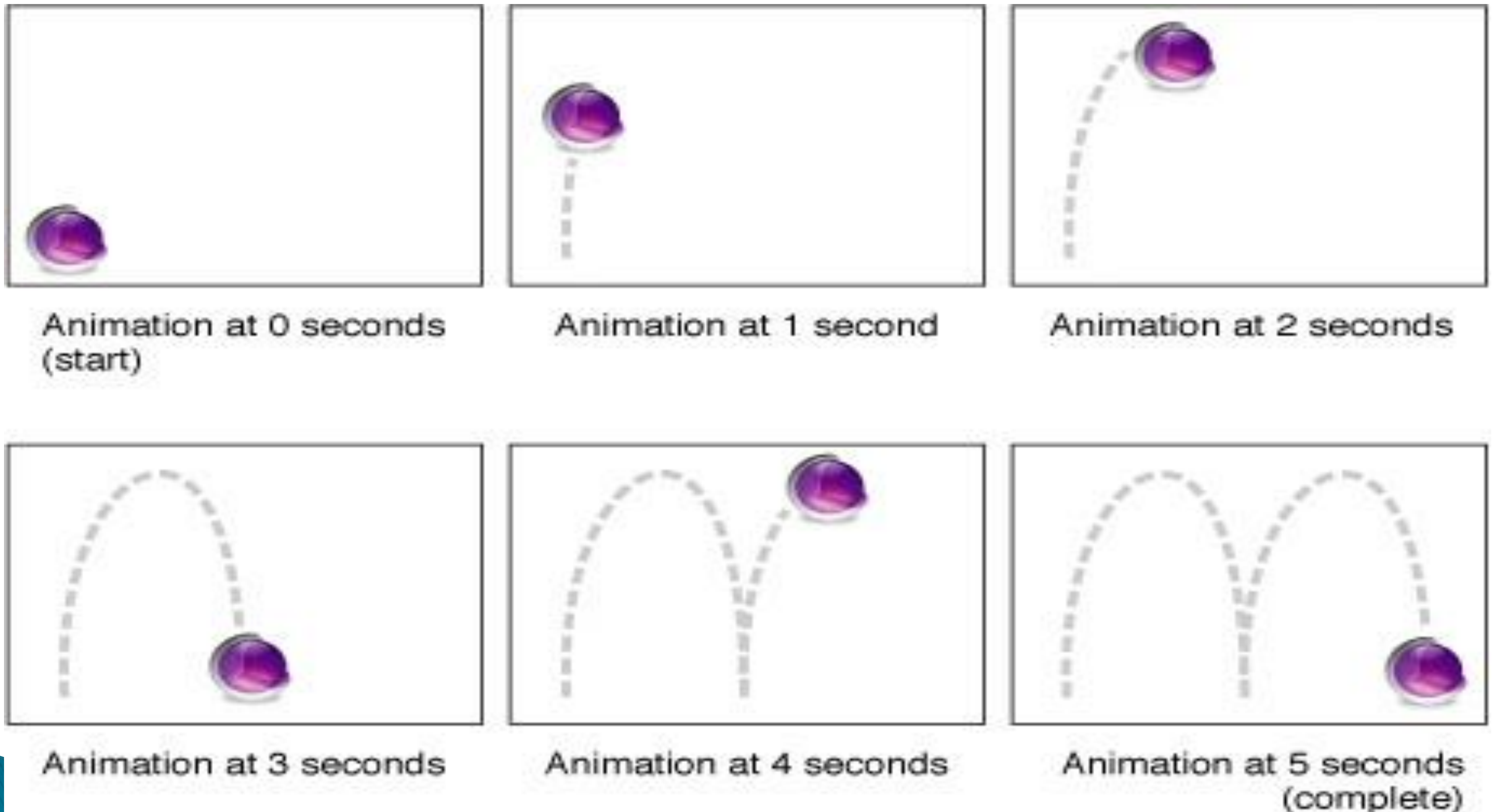
- Each object participating in the action is given object definition, such as terms of basic shapes, such as polygon or shapes.



3. key frame specification

- A key frame in animation and film making is a drawing that defines the starting and ending points of any smooth transition.
- A sequence of key frames which movement the spectator will see, but the position of the key fames on the film, defines the timimng of the movement. 2 or 3 can be present for a span of a second.

Key Frames

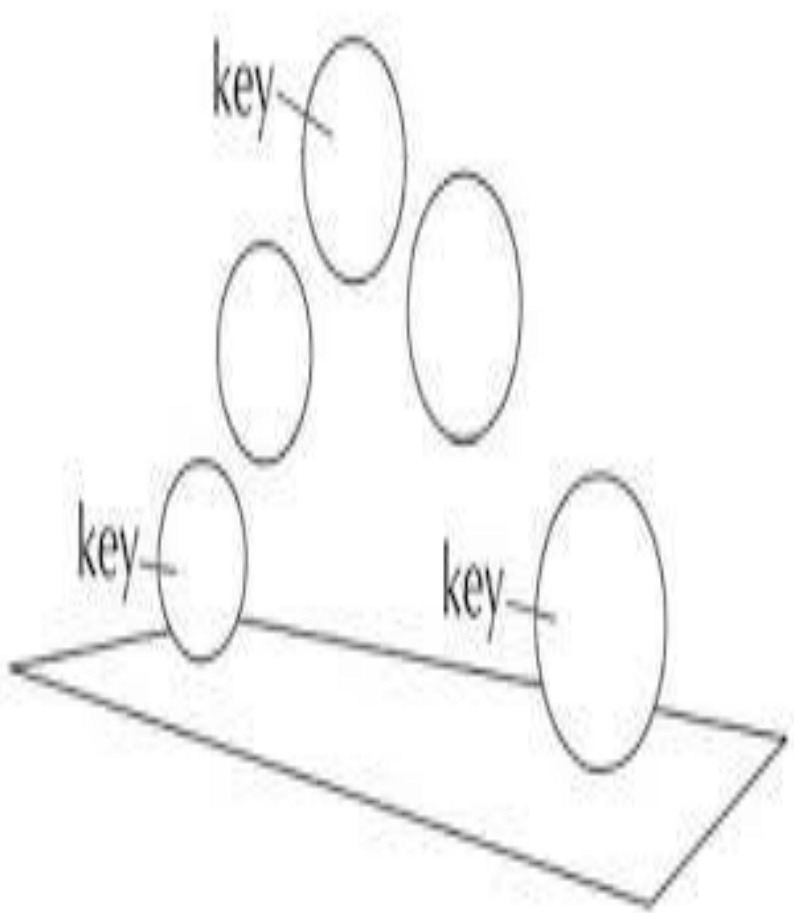


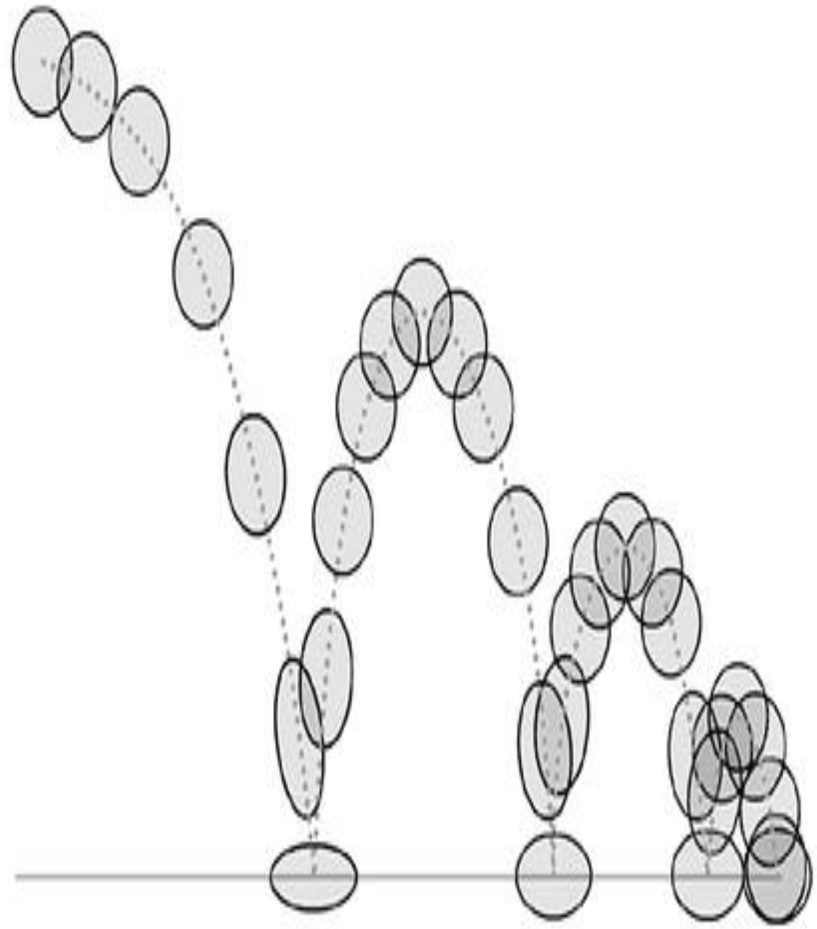
Animation at 0 seconds (start)
Animation at 1 second
Animation at 2 seconds
Animation at 3 seconds
Animation at 4 seconds
Animation at 5 seconds (complete)

4. Generation of In between Frames

- It is a process of generating intermediate frames between 2 images to give appearance that the 1^{st} image evolves smoothly into the second images. In-betweens are the drawing between the key frames which help to create the illusion of motion.
- Filmfilm requires 24 frames per second and graphic terminals are refreshed at a rate of 30 to 60 frames per second.

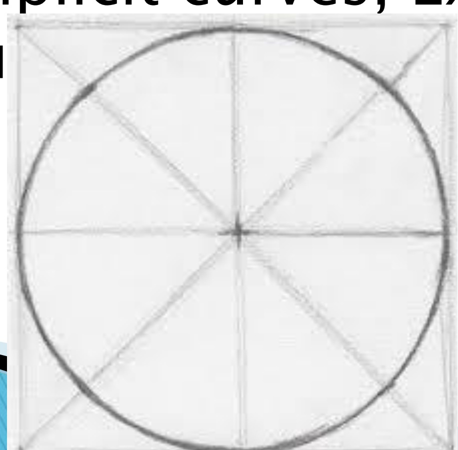
In Between Frames

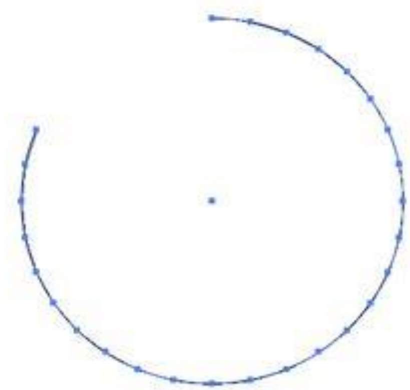




Computer Graphics Curves

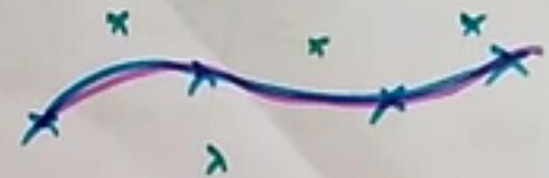
- In CG, we often need to draw different types of object onto the screen.
- Object are not flat all the time and we need to draw curves many times to draw an object.
- Types of curves
  - Curve is Infinitely large set of points.
  - Each point has two neighbors except end points.
  - Implicit curves, Explicit curves and parametric cu





Types of Curve

Implicit Curves:



$F(x, y) = 0$

$\longrightarrow$ scalar func $\Rightarrow$ returns a single real no:

Explicit Curves:

$y = F(x)$ $\Rightarrow$ for each values of x a single value y is normally computed by the function

Parametric Curves:

$P(t) = F(t), g(t)$ (or)

$P(t) = x(t), y(t)$

} 2D

Bezier Curve

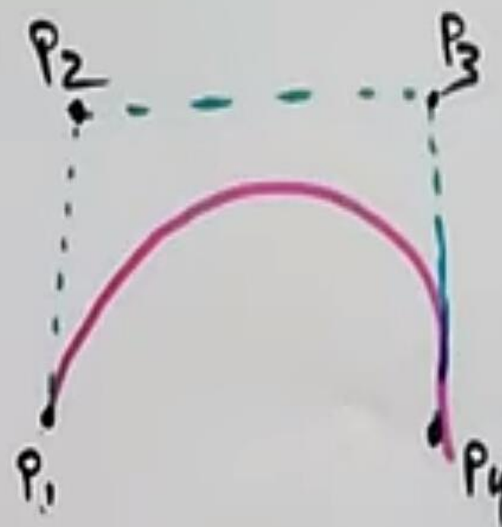
⇓

defined by {control points}

P_1, P_2, P_3, P_4

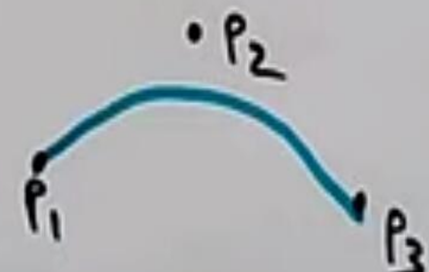
⇓

control points

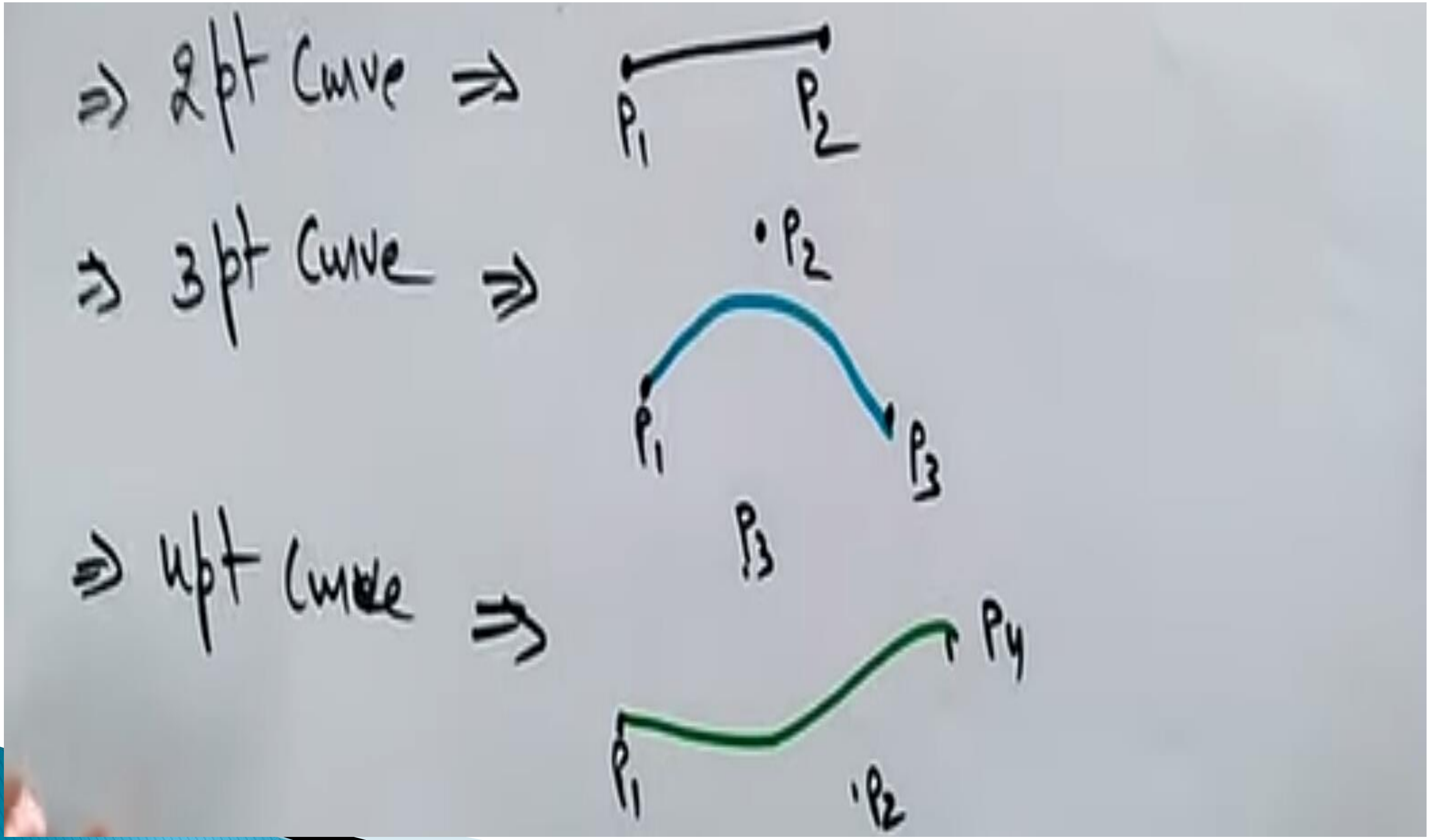


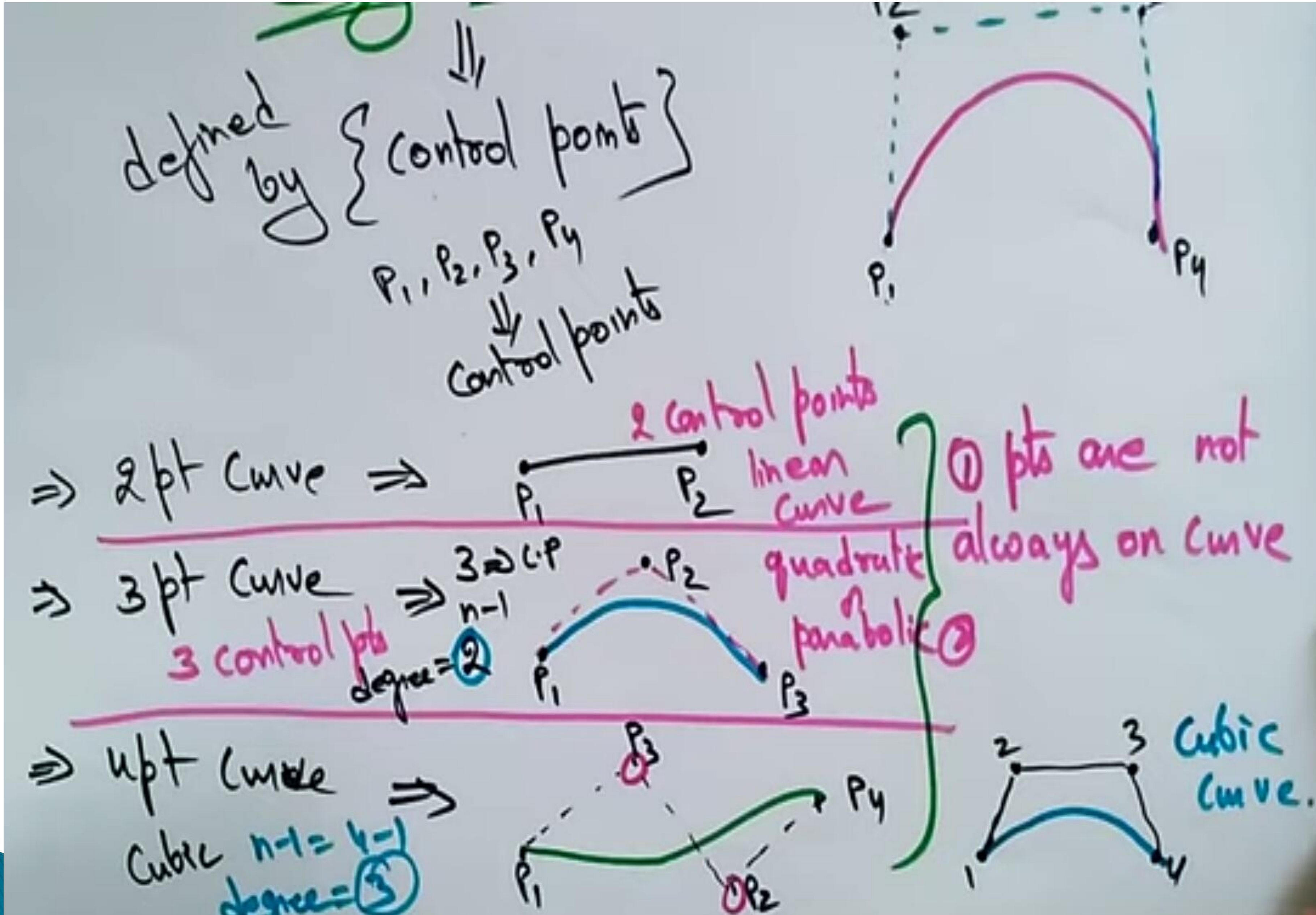
⇒ 2pt Curve ⇒

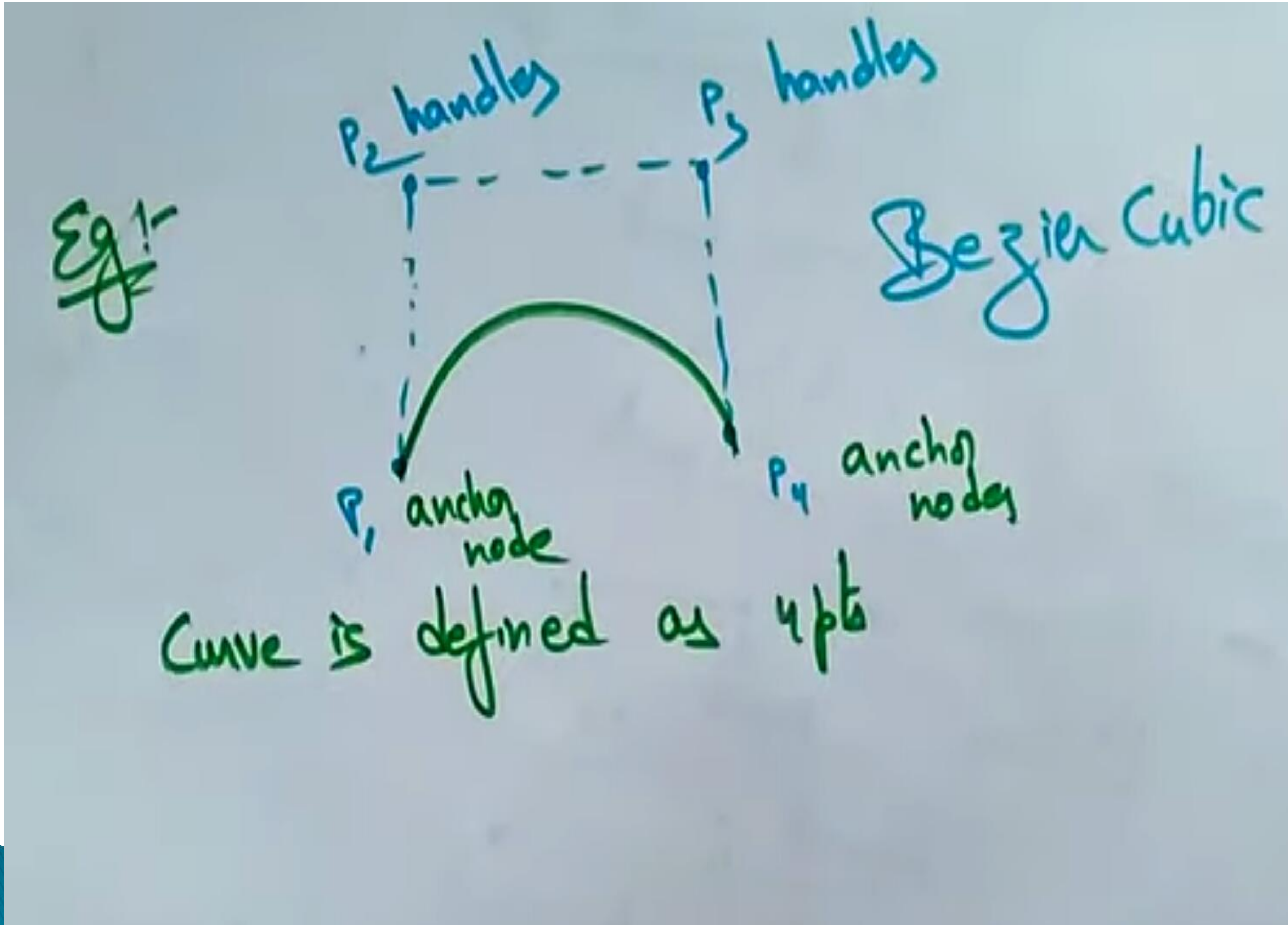
⇒ 3pt Curve ⇒



Types of Brezier curves




defined by {control points}
P1, P2, P3, P4
control points
P1
P4
2 control points
linear curve
⇒ 2pt curve ⇒
P1
P2
① pts are not always on curve
⇒ 3pt curve ⇒ 3 ⇒ C.P
n−1
3 control pts
degree = 2
P1
P2
P3
quadratic
or
parabolic ②
⇒ 4pt curve ⇒
P3
P4
P1
P2
cubic n−1 = 4−1
degree = 3
1
2
3
4
cubic curve.


Eg 1:-
P2 handles
P3 handles
Bezier Cubic
P1 anchor node
P4 anchor nodes
Curve is defined as 4 pts

Properties of Breizer Curves

properties

→ They always pass through first & last control pts.

→ They are contained in the convex hull of their defining control points

→ convex hull

→ The degree of polynomial defining the curve segment is one less than the no of defining polygon pt.

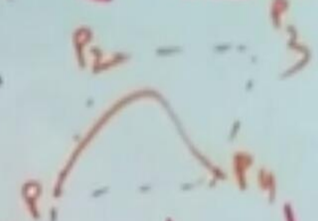
→ Bezier curves are tangent to their first & last edges of control polyline

→ Bezier curves exhibit global control pts means moving a control pts alters the slope of the whole curve.

→ It is a spline curve [smooth in nature used to draw CAD diagram)

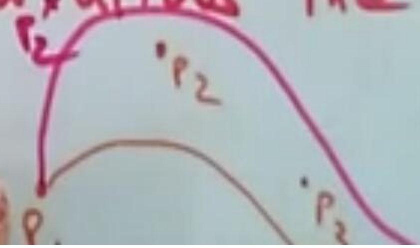
Properties of Breizer Curves

⇒ The shape of defining polygon is usually followed by Bezier Curve



⇒ Bezier Curves are tangent to their first & last Edges of control polyline.

⇒ Bezier Curves exhibit global control point ~~alters the~~ means moving a control points alters the shape of the whole curve.



Derivation of Bezier Curve

⇒ Blending / Basis fun

$$p(u) = \sum_{i=0}^{n} P_i \, B_{i,n}(u), \quad 0 \le u \le 1$$

where

$n+1 \Rightarrow$ control pts

$n \Rightarrow$ degree

$B_{i,n}(u) \Rightarrow C(n,i)\, u^i (1-u)^{n-i}$

$$C(n,i) = \frac{n!}{i!(n-i)!}$$

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

Cubic Bezier

$$P(u) = \sum_{i=0}^{n} P_i \, B_{i,n}(u)$$

$$P(u) = P_0 B_{0,3}(u) + P_1 B_{1,3}(u) + P_2 B_{2,3}(u) + P_3 B_{3,3}(u)$$

Control pts

P_0, P_1, P_2, P_3

$i=0$

$$P(u) = P_0 B_{0,3}(u) + P_1 B_{1,3}(u) + P_2 B_{2,3}(u) + P_3 B_{3,3}(u)$$

Control pt
P_0, P_1, P_2, P_3

$$B_{0,3}(u) = \frac{3!}{0!\,(3-0)!} u^0 \cdot (1-u)^{3-0} = (1-u)^3$$

$$B_{1,3}(u) = \frac{3!}{1!\,2!} u^1 (1-u)^{3-1} \Rightarrow 3u \cdot (1-u)^2$$

$$B_{2,3}(u) = 3u^2(1-u)$$

$$B_{3,3}(u) = u^3.$$

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

So, blending function,

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

$$\boxed{P(u_x) = P_0 x_0 (1-u)^3 + Px_1 \cdot 3u \cdot (1-u)^2 + Px_2 \cdot 3u^2 \cdot (1-u) + Px_3 \cdot u^3}$$

$$\boxed{P(u_y) = Py_0 (1-u)^3 + Py_1 \cdot 3u \cdot (1-u)^2 + Py_2 \cdot 3u^2 \cdot (1-u) + Py_3 \cdot u^3}$$

$$\boxed{P(u_z) = Pz_0 (1-u)^3 + Pz_1 \cdot 3u \cdot (1-u)^2 + Pz_2 \cdot 3u^2 \cdot (1-u) + Pz_3 \cdot u}$$

$$P(u) = P_0 B_{0,3}(u) + P_1 B_{1,3}(u) + P_2 B_{2,3}(u) + P_3 B_{3,3}(u)$$

$$\Rightarrow B_{0,3}(u) = C(3,0) \cdot u^0 \cdot (1-u)^{3-0}$$

$$= \frac{3!}{0!\,3!}(1-u)^3 \Rightarrow (1-u)^3 \quad \text{—— } ①$$

$$\Rightarrow B_{1,3}(u) = C(3,1) \cdot u^1 \cdot (1-u)^{3-1}$$

$$= 3u \cdot (1-u)^2 \quad \text{—— } ②$$

$$\Rightarrow B_{2,3}(u) = C(3,2) \cdot u^2 \cdot (1-u)^{3-2}$$

Example: Design a Bezier Curve Controlled by points A(1,1) B(2,3) C(4,3) D(6,4)

Solution:-

Control points = 4

$n = 4 - 1 = 3$

$A(1,1) = P_0$ $\quad$ $C(4,3) = P_2$

$B(2,3) = P_1$ $\quad$ $D(6,4) = P_3$

Solution:- Control points = 4

$n = 4-1 = 3$

$A(1,1) = P_0$ $C(4,3) = P_2$

$B(2,3) = P_1$ $D(6,4) = P_3$

Eqn of bezier Curve,

$$P(u) = \sum_{i=0}^{n} P_i B_{i,n}(u)$$

$$\boxed{B_{i,n}(u) = \frac{n!}{i!(n-i)!} \cdot u^i \cdot (1-u)^{n-i}}$$

$$P(u) = P_0 B_{0,3}(u) + \ldots$$

$$= \boxed{P_0(1-u)^3 + P_1 \cdot 3u(1-u)^2 + P_2\, 3u^2(1-u) + P_3 u^3}$$

$P_0 = (1, 1)$ $P_1 = (2, 3)$

$P_2 = (4, 3)$ $P_4 = (6, 4)$

To find $x(u)$ & $y(u)$

$$x(u) = x_0(1-u)^3 + 6u(1-u)^2 + 12u^2(1-u) + 6u^3$$

$$y(u) = (1-u)^3 + 9u(1-u)^2 + 9u^2(1-u) + 4u^3$$

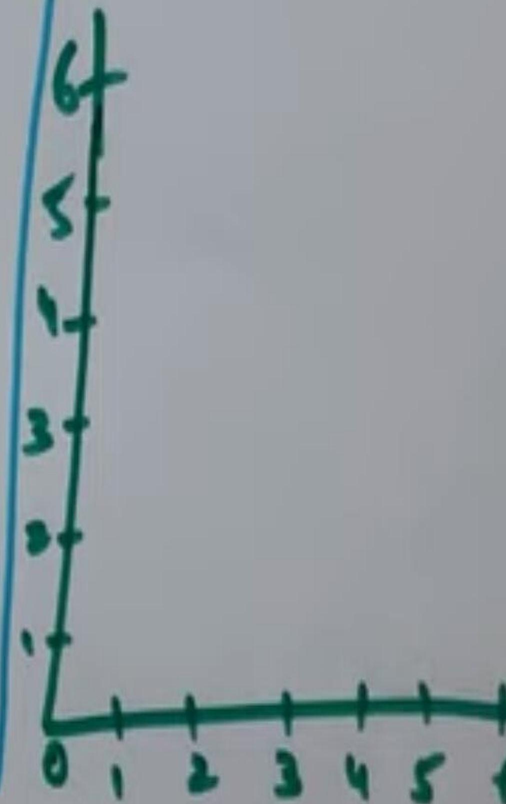
$\le u \le 1$

	u	x(u)	y(u)
	0	1	1
0.2	0.2	1.712	0.984
0.2	0.4	2.616	2.632
0.2	0.6	3.664	3.088
0.2	0.8	4.808	3.496
0.2	1	6	4

u<1

u	x(u)	y(u)
0	1	1
0.2	1.712	0.984
0.4	2.616	2.632
0.6	3.664	3.088
0.8	4.808	3.496
1	6	4

0.2
0.2
0.2
0.2
0.2

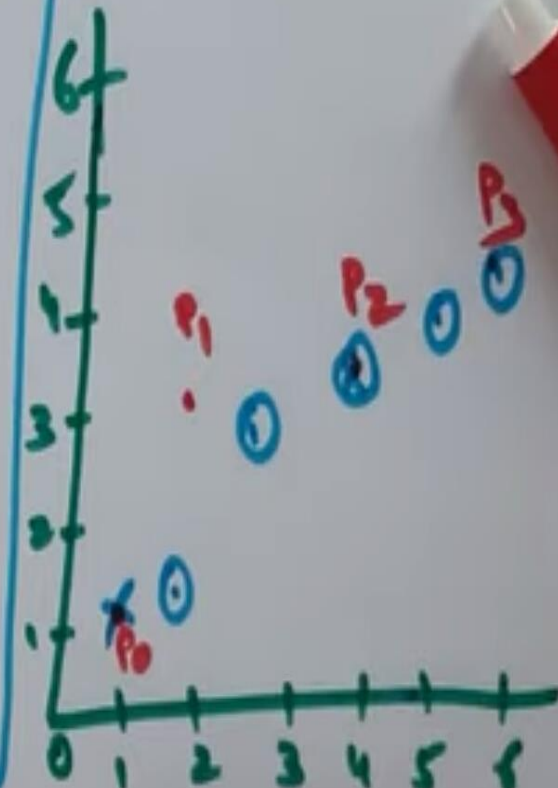


$P_0(1,1)$ $P_2(4,3)$

$P_1(2,3)$ $P_3(6,4)$

u	x(u)	y(u)
0	1	1
0.2	1.712	1.984
0.4	2.616	2.632
0.6	3.664	3.088
0.8	4.808	3.496
1	6	4

0.2 0.2 0.2 0.2 0.2



$P_0(1,1)$ $P_2(4,3)$

$P_1(2,3)$ $P_3(6,4)$

B-Spline Curve

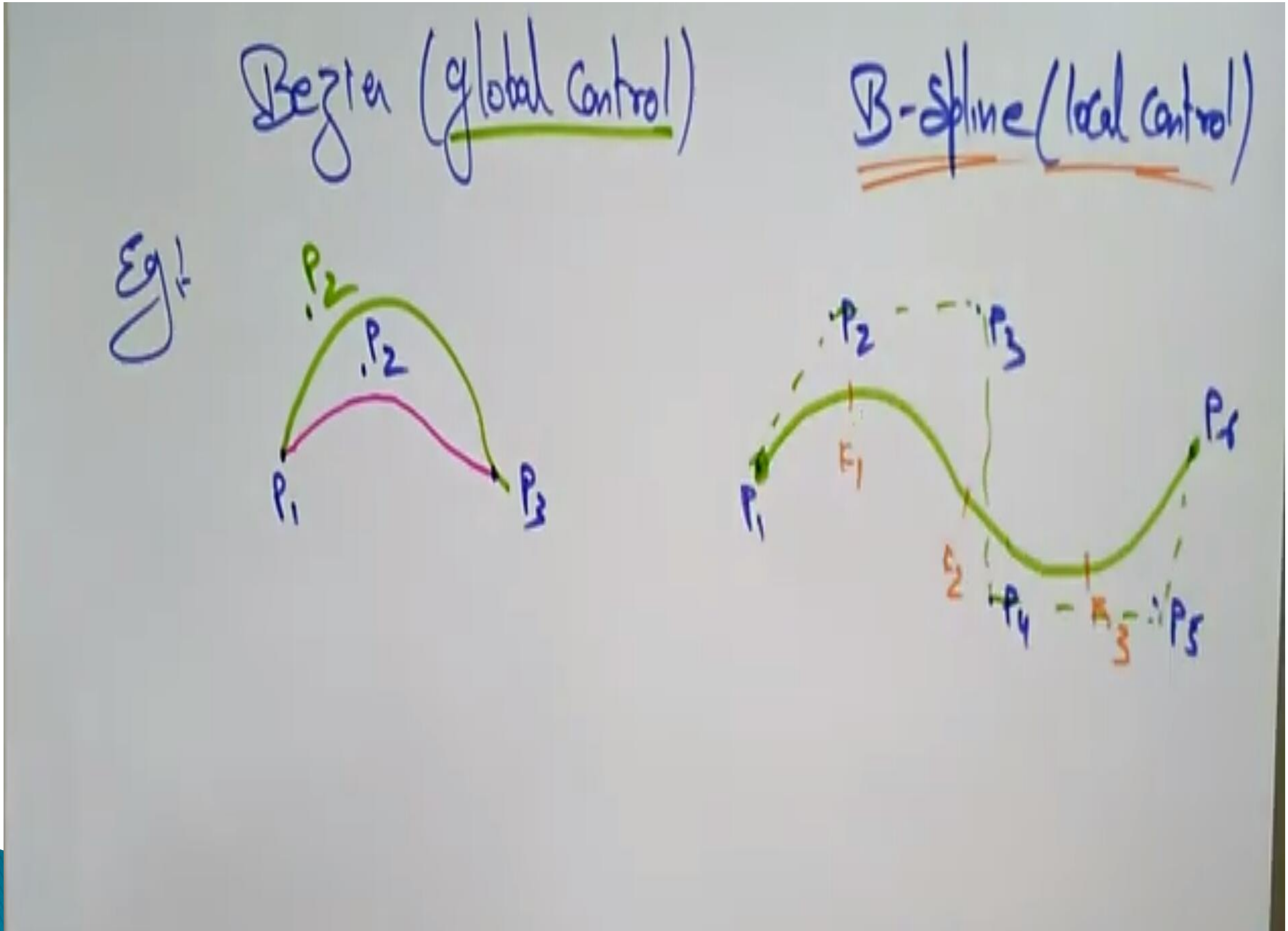
⇒ Bezier Curve (global control) ⇒ Control points

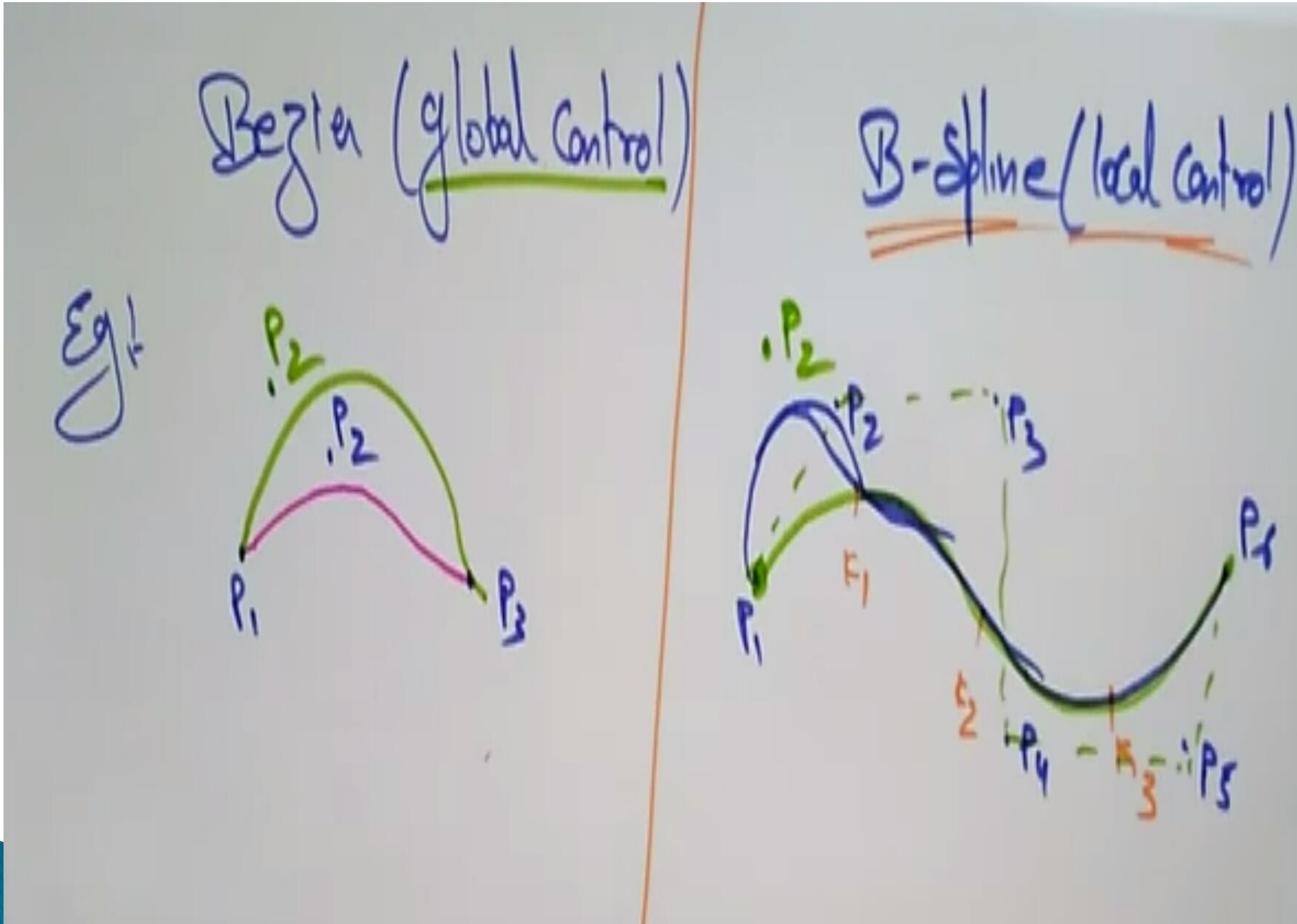
⇒ B-spline Curve (local Control)

⇓

degree of poly Does not depends upon ~~con~~ no: of Control points

⇒ It depends upon the order of Polynomial.


Bezier (global control)
B-spline (local control)
Eg:
P_2
P_2
P_1
P_3
P_2
P_3
P_1
P_4
P_5



Bezier (global control)
B-spline (local control)
Eg:
P2
P2
P1
P3
P2
P2
P3
P1
k1
k2
P4
k3
P5

Properties of B-spline Curve

→ The sum of B-spline basis function at any parameter 'u' equal to 1 i.e

$$\sum_{i=1}^{n+1} N_{i,k}(u) = 1$$

n+1 = no: of control point

k = order of B-spline curve

→ The basis function is +ve or zero for all parameter values i.e $N_{i,k}(u) \geq 0$. Except for $k=1$ each basis function has one maximum value

→ The maximum order of the curve is equal to the number of vertices of defining polygon.

→ The degree of B-spline polynomial is independent on the no: of ~~verti~~ vertices of defining polygon

→ B-spline allow the control (ie local control) over the curve surface.

→ The curve lies within the convex hull of its defining polygon

→ The curve generally follow the ~~spo~~ shape of defining polygon.

→ The B-spline curve generally represented as,

$$P(u) = \sum_{i=1}^{n+1} P_i \, N_{i,k}(u), \quad \left\{ \begin{array}{l} \text{Blending function} \\ \text{for B-spline curve} \end{array} \right\}$$

where $u_{min} \le u \le u_{max}$

$$2 \le k \le n+1$$

$$N_{i,k}(u) = \frac{(u - x_i) \cdot N_{i,k-1}(u)}{x_{i+k-1} - x_i} + \frac{(x_{i+k} - u) \cdot N_{i+1,k-1}(u)}{x_{i+k} - x_{i+1}}$$

$$N_{i,k}(u) = \begin{cases} 1 & x_i \le u \le x_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

$x_i = 0$ if $i < k$

$x_i = i - k + 1$ if $k \le i \le n$

$x_i = n - k + 2$ if $i > n$

When we designing the B-spline curve we have to Evaluate knot, vector based on the no: of control points & order of curve.

⇒ knot vector can be evaluated by using

i) uniform
ii) Non-uniform
iii) open-uniform

OpenGL Curve Functions

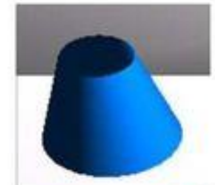
- **GLU** (OpenGL Utility) functions:

  3D Quadrics (Spheres, Cylinders)

  Rational B-Splines (circles, ellipse, Bezier curve)

  (有理B樣條) gluSphere/gluCylinder/gluDisk…

  NURBS (non-uniform rational B-splines)



(By Mark Kilgard)

- **GLUT** (OpenGL Utility Toolkit) functions:

  3D Quadrics (Spheres, Cones etc.) glutSolidSphere/glutWireSphere, glutSolidCone, glutSolidTorus, …

- Approximate a simple curve using a polyline
  - The more line sections, the smoother appearance of the curve.
- Write your own curve-generation algorithms

4

Computer Graphics with Open GL, Third Edition, by Donald Hearn and M.Pauline Baker.
ISBN 0-13-0-15390-7

Thank You