SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC | Approved by AICTE

COMPUTER GRAPHICS & MULTIMEDIA SYSTEMS
SCS1302

UNIT III – PART II

Syllabus

SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY FACULTY OF COMPUTING

		L	T	P	Credits	Total Marks
SCS1302	COMPUTER GRAPHICS AND MULTIMEDIA SYSTEMS	3	0	0	3	100

COURSE OBJECTIVES

- To gain knowledge to develop, design and implement two and three dimensional graphical structures.
- To enable students to acquire knowledge of Multimedia compression and animations.
- To learn creation, Management and Transmission of Multimedia objects.

UNIT 1 BASICS OF COMPUTER GRAPHICS 9 Hrs.

Output Primitives: Survey of computer graphics - Overview of graphics systems - Line drawing algorithm - Circle drawing algorithm - Curve drawing algorithm - Attributes of output primitives - Anti-aliasing.

UNIT 2 2D TRANSFORMATIONS AND VIEWING 8 Hrs.

Basic two dimensional transformations - Other transformations - 2D and 3D viewing - Line clipping - Polygon clipping - Logical classification - Input functions - Interactive picture construction techniques.

UNIT 3 3D CONCEPTS AND CURVES 10 Hrs.

3D object representation methods - B-REP , sweep representations, Three dimensional transformations. Curve generation - cubic splines, Beziers, blending of curves- other interpolation techniques, Displaying Curves and Surfaces, Shape description requirement, parametric function. Three dimensional concepts – Introduction - Fractals and self similarity- Successive refinement of curves, Koch curve and peano curves.

Syllabus

UNIT 4 METHODS AND MODELS **8 Hrs.**

Visible surface detection methods - Illumination models - Halftone patterns - Dithering techniques - Polygon rendering methods - Ray tracing methods - Color models and color applications.

UNIT 5 MULTIMEDIA BASICS AND TOOLS **10 Hrs.**

Introduction to multimedia - Compression & Decompression - Data & File Format standards - Digital voice and audio - Video image and animation. Introduction to Photoshop - Workplace - Tools - Navigating window - Importing and exporting images - Operations on Images - resize, crop, and rotate - Introduction to Flash - Elements of flash document - Drawing tools - Flash animations - Importing and exporting - Adding sounds - Publishing flash movies - Basic action scripts - GoTo, Play, Stop, Tell Target

Max. 45 Hours

TEXT / REFERENCE BOOKS

1. Donald Hearn, Pauline Baker M., "Computer Graphics", 2nd Edition, Prentice Hall, 1994.
2. Tay Vaughan ,"Multimedia", 5th Edition, Tata McGraw Hill, 2001.
3. Ze-Nian Li, Mark S. Drew ,"Fundamentals of Multimedia", Prentice Hall of India, 2004.
4. D. McClelland, L.U.Fuller ,"Photoshop CS2 Bible", Wiley Publishing, 2005.
5. James D. Foley, Andries van Dam, Steven K Feiner, John F. Hughes, "Computer Graphics Principles and Practice, 2nd
6. Edition in C, Audison Wesley, ISBN - 981 -235-974-5
7. William M. Newman, Roberet F. Sproull, " Principles of Interactive Computer Graphics", Second Edition, Tata McGraw-Hill Edition.

Course Objective(CO)

CO1: Construct lines and circles for the given input.

CO2: Apply 2D transformation techniques to transform the shapes to fit them as per the picture definition.

CO3: Construct splines, curves and perform 3D transformations

CO4: Apply colour and transformation techniques for various applications.

CO5: Analyse the fundamentals of animation, virtual reality, and underlying technologies.

CO6: Develop photo shop applications

Three dimensional transformations

Basic Transformations:
1.Translation
2.Rotation
3.Scaling

Other Transformations:
1.Reflection
2.Shearing

Translation

A translation in space is described by tx , ty and tz. It is easy to see that this matrix realizes the equations:

x2=x1+tx
y2=y1+ty
z2=z1+tz

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ t_x & t_y & t_z & 1 \end{bmatrix}$$

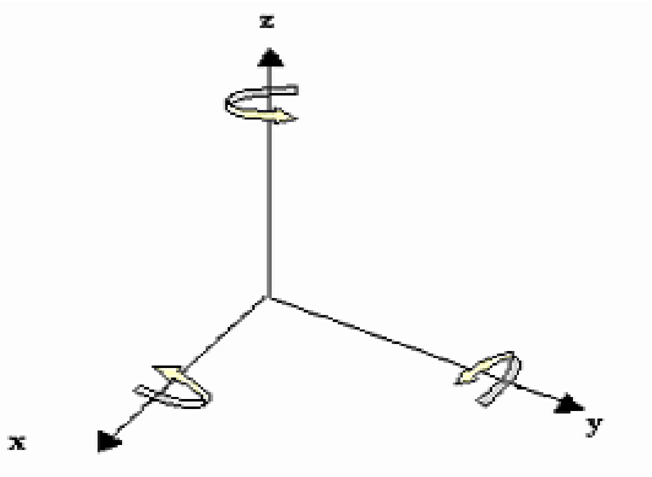
Translation Matrix

Rotation

3D rotation is not same as 2D rotation. In 3D rotation, we have to specify the angle of rotation along with the axis of rotation. We can perform 3D rotation about X, Y, and Z axes. They are represented in the matrix form as below :

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos\theta & -sin\theta & 0 \\ 0 & sin\theta & cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} R_y(\theta) = \begin{bmatrix} cos\theta & 0 & sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -sin\theta & 0 & cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_z(\theta) = \begin{bmatrix} cos\theta & -sin\theta & 0 & 0 \\ sin\theta & cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Three dimensional transformations

Scaling

In 3D scaling operation, three coordinates are used. Let us assume that the original coordinates are (X, Y, Z), scaling factors are (S_x, S_y, S_z) respectively, and the produced coordinates are (X', Y', Z'). This can be mathematically represented as shown below :

$$S = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$P' = P \cdot S$$

$$[X' \;\; Y' \;\; Z' \;\; 1] = [X \;\; Y \;\; Z \;\; 1] \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= [X.S_x \;\; Y.S_y \;\; Z.S_z \;\; 1]$$

Reflection

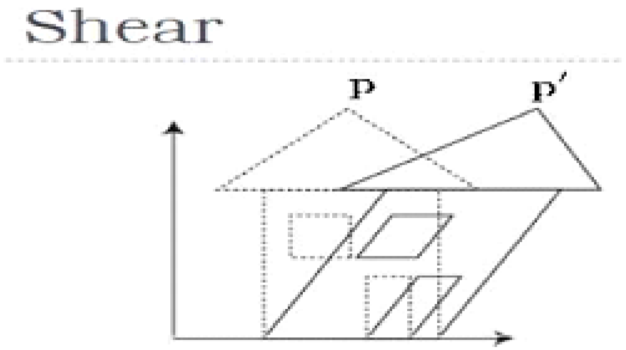
A transformation that gives the mirror image of the object.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

We can mirror the different planes by using scaling factor -1 on the axis that is placed normally on the plane. Notice the matrix to the left. It mirrors around the xy-plane, and changes the coordinates from a right hand system to a left hand system.

Shear

A transformation that slants the shape of an object is called the shear transformation. Like in 2D shear, we can shear an object along the X-axis, Y-axis, or Z-axis in 3D.



As shown in the above figure, there is a coordinate P. You can shear it to get a new coordinate P', which can be represented in 3D matrix form as below

$$Sh = \begin{bmatrix} 1 & sh_x^y & sh_x^z & 0 \\ sh_y^x & 1 & sh_y^z & 0 \\ sh_z^x & sh_z^y & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$P' = P \cdot Sh$$

$$X' = X + Sh_x^y Y + Sh_x^z Z$$

$$Y' = Sh_y^x X + Y + sh_y^z Z$$

$$Z' = Sh_z^x X + Sh_z^y Y + Z$$

Translation - Example

- Given a 3D object with coordinate points A(0, 3, 1), B(3, 3, 2), C(3, 0, 0), D(0, 0, 0). Apply the translation with the distance 1 towards X axis, 1 towards Y axis and 2 towards Z axis and obtain the new coordinates of the object.
- Given-

  Old coordinates of the object = A (0, 3, 1), B(3, 3, 2), C(3, 0, 0), D(0, 0, 0)

  Translation vector = $(T_x, T_y, T_z) = (1, 1, 2)$
- **<u>For Coordinates A(0, 3, 1)</u>**

  Let the new coordinates of A = (X', Y', Z').

Applying the translation equations, we have-

$X' = X + T_x = 0 + 1 = 1$

$Y' = Y + T_y = 3 + 1 = 4$

$Z' = Z + T_z = 1 + 2 = 3$

Thus, New coordinates of A = (1, 4, 3).

Translation - Example

For Coordinates B(3, 3, 2)

Let the new coordinates of B = (X', Y', Z').

Applying the translation equations, we have-

$X' = X + T_x = 3 + 1 = 4$

$Y' = Y + T_y = 3 + 1 = 4$

$Z' = Z + T_z = 2 + 2 = 4$

Thus, New coordinates of B = (4, 4, 4).

For Coordinates C(3, 0, 0)

Let the new coordinates of C = ((X', Y', Z').

Applying the translation equations, we have-

$X' = X + T_x = 3 + 1 = 4$

$Y' = Y + T_y = 0 + 1 = 1$

$Z' = Z + T_z = 0 + 2 = 2$

Thus, New coordinates of C = (4, 1, 2).

Translation - Example

<u>For Coordinates D(0, 0, 0)</u>

Let the new coordinates of D = (X’, Y’, Z’).

Applying the translation equations, we have-

$X' = X + T_x = 0 + 1 = 1$

$Y' = Y + T_y = 0 + 1 = 1$

$Z' = Z + T_z = 0 + 2 = 2$

Thus, New coordinates of D = (1, 1, 2).

<u>Thus, New coordinates of the object = A (1, 4, 3), B(4, 4, 4), C(4, 1, 2), D(1, 1, 2).</u>

Rotation - Example

- Given a homogeneous point (1, 2, 3). Apply rotation 90 degree towards X, Y and Z axis and find out the new coordinate points.

Solution-

Given-

Old coordinates = (X, Y, Z) = (1, 2, 3)

Rotation angle = θ = 90º

For X-Axis Rotation-

This rotation is achieved by using the following rotation equations-

$X' = X$

$Y' = Y \cos\theta - Z \sin\theta$

$Z' = Y \sin\theta + Z \cos\theta$

For Y-Axis Rotation-

This rotation is achieved by using the following rotation equations-

$X' = Z \sin\theta + X \cos\theta$

$Y' = Y$

$Z' = Y \cos\theta - X \sin\theta$

For Z-Axis Rotation-

This rotation is achieved by using the following rotation equations-

$X' = X \cos\theta - Y \sin\theta$

$Y' = X \sin\theta + Y \cos\theta$

$Z' = Z$

Rotation - Example

For X-Axis Rotation-

Let the new coordinates after rotation = (X', Y', Z').

Applying the rotation equations, we have-

$X' = X = 1$

$Y' = Y \cos\theta - Z \sin\theta$

$= 2 \cos 90° - 3 \sin 90°$

$= 2 \times 0 - 3 \times 1 = -3$

$Z' = Y \sin\theta + Z \cos\theta$

$= 2 \sin 90° + 3 \cos 90°$

$= 2 \times 1 + 3 \times 0 = 2$

Thus, New coordinates after rotation = (1, -3, 2).

For Y-Axis Rotation-

Let the new coordinates after rotation = (X', Y', Z').

Applying the rotation equations, we have-

$X' = Z \sin\theta + X \cos\theta$

$= 3 \sin 90° + 1 \cos 90°$

$= 3 \times 1 + 1 \times 0 = 3$

$Y' = Y = 2$

$Z' = Y \cos\theta - X \sin\theta$

$= 2 \times \cos 90° - 1 \times \sin 90°$

$= 2 \times 0 - 1 \times 1 = -1$

Thus, New coordinates after rotation = (3, 2, -1).

Rotation - Example

For Z-Axis Rotation-

Let the new coordinates after rotation = (X', Y', Z').

Applying the rotation equations, we have-

$X' = X \cos\theta - Y \sin\theta$

$= 1 \times \cos 90° - 2 \times \sin 90°$

$= 1 \times 0 - 2 \times 1 = -2$

$Y' = X \sin\theta + Y \cos\theta$

$= 1 \times \sin 90° + 2 \times \cos 90°$

$= 1 \times 1 + 2 \times 0 = 1$

$Z' = Z = 3$

Thus, New coordinates after rotation = (-2, 1, 3).

Scaling

Consider a point object O has to be scaled in a 3D plane.

Let-

Initial coordinates of the object O = (X, Y,Z)

Scaling factor for X-axis = S_x

Scaling factor for Y-axis = S_y

Scaling factor for Z-axis = S_z

New coordinates of the object O after scaling = (X', Y', Z')

This scaling is achieved by using the following scaling equations-

$X' = X \times S_x$

$Y' = Y \times S_y$

$Z' = Z \times S_z$

Scaling - Example

- Given a 3D object with coordinate points A(0, 3, 3), B(3, 3, 6), C(3, 0, 1), D(0, 0, 0). Apply the scaling parameter 2 towards X axis, 3 towards Y axis and 3 towards Z axis and obtain the new coordinates of the object.

Solution-

Given-

Old coordinates of the object = A (0, 3, 3), B(3, 3, 6), C(3, 0, 1), D(0, 0, 0)

Scaling factor along X axis = 2

Scaling factor along Y axis = 3

Scaling factor along Z axis = 3

Scaling - Example

- **For Coordinates A(0, 3, 3)**

Let the new coordinates of A after scaling = (X', Y', Z').

Applying the scaling equations, we have-

$X' = X \times S_x = 0 \times 2 = 0$

$Y' = Y \times S_y = 3 \times 3 = 9$

$Z' = Z \times S_z = 3 \times 3 = 9$

Thus, New coordinates of corner A after scaling = (0, 9, 9).

- **For Coordinates B(3, 3, 6)**

Let the new coordinates of B after scaling = (X', Y', Z').

Applying the scaling equations, we have-

$X' = X \times S_x = 3 \times 2 = 6$

$Y' = Y \times S_y = 3 \times 3 = 9$

$Z' = Z \times S_z = 6 \times 3 = 18$

Thus, New coordinates of corner B after scaling = (6, 9, 18).

Scaling - Example

- **For Coordinates C(3, 0, 1)**

Let the new coordinates of C after scaling = (X', Y', Z').

Applying the scaling equations, we have-

$X' = X \times S_x = 3 \times 2 = 6$

$Y' = Y \times S_y = 0 \times 3 = 0$

$Z' = Z \times S_z = 1 \times 3 = 3$

Thus, New coordinates of corner C after scaling = (6, 0, 3).

- **For Coordinates D(0, 0, 0)**

Let the new coordinates of D after scaling = (X', Y', Z').

Applying the scaling equations, we have-

$X' = X \times S_x = 0 \times 2 = 0$

$Y' = Y \times S_y = 0 \times 3 = 0$

$Z' = Z \times S_z = 0 \times 3 = 0$

Thus, New coordinates of corner D after scaling = (0, 0, 0).

Reflection - Example

- Reflection relative to XY plane
- Reflection relative to YZ plane
- Reflection relative to XZ plane

<u>Reflection Relative to XY Plane:</u>

This reflection is achieved by using the following reflection equations-

X’ = X

Y’ = Y

Z’ = -Z

<u>Reflection Relative to YZ Plane:</u>

This reflection is achieved by using the following reflection equations-

X’ = -X

Y’ = Y

Z’ = Z

<u>Reflection Relative to XZ Plane:</u>

This reflection is achieved by using the following reflection equations-

X’ = X

Y’ = -Y

Z’ = Z

Reflection - Example

Given a 3D triangle with coordinate points A(3, 4, 1), B(6, 4, 2), C(5, 6, 3). Apply the reflection on the XY plane and find out the new coordinates of the object.

Solution

Given:

Coordinates of the triangle = A (3, 4, 1), B(6, 4, 2), C(5, 6, 3)

Reflection has to be taken on the XY plane

For Coordinates A(3, 4, 1)

Let the new coordinates of corner A after reflection = (X', Y', Z').

Applying the reflection equations, we have-

X' = X = 3

Y' = Y = 4

Z' = -Z = -1

Thus, New coordinates of corner A after reflection = (3, 4, -1).

Reflection - Example

For Coordinates B(6, 4, 2)

Let the new coordinates of corner B after reflection = (X’, Y’, Z’).

Applying the reflection equations, we have-

X’ = X = 6

Y’ = Y = 4

Z’ = -Z = -2

Thus, New coordinates of corner B after reflection = (6, 4, -2).

For Coordinates C(5, 6, 3)

Let the new coordinates of corner C after reflection = (X’, Y’, Z’).

Applying the reflection equations, we have-

X’ = X = 5

Y’ = Y = 6

Z’ = -Z = -3

Thus, New coordinates of corner C after reflection = (5, 6, -3).

Thus, New coordinates of the triangle after reflection = A (3, 4, -1), B(6, 4, -2), C(5, 6, -3).

Shearing - Example

- Shearing parameter towards X direction = Sh_x
- Shearing parameter towards Y direction = Sh_y
- Shearing parameter towards Z direction = Sh_z

<u>Shearing in X Axis-</u>

Shearing in X axis is achieved by using the following shearing equations-

$X' = X$

$Y' = Y + Sh_y \times X$

$Z' = Z + Sh_z \times X$

<u>Shearing in Y Axis-</u>

Shearing in Y axis is achieved by using the following shearing equations-

$X' = X + Sh_x \times Y$

$Y' = Y$

$Z' = Z + Sh_z \times Y$

<u>Shearing in Z Axis-</u>

Shearing in Z axis is achieved by using the following shearing equations-

$X' = X + Sh_x \times Z$

$Y' = Y + Sh_y \times Z$

$Z' = Z$

Shearing - Example

- Given a 3D triangle with points (0, 0, 0), (1, 1, 2) and (1, 1, 3). Apply shear parameter 2 on X axis, 2 on Y axis and 3 on Z axis and find out the new coordinates of the object.

Solution-

Given-

Old corner coordinates of the triangle = A (0, 0, 0), B(1, 1, 2), C(1, 1, 3)

$Sh_x = 2$

$Sh_y = 2$

$Sh_z = 3$

Shearing in X Axis-

For Coordinates A(0, 0, 0)

Let the new coordinates of corner A after shearing = (X', Y', Z').

Applying the shearing equations, we have-

X' = X = 0

$Y' = Y + Sh_y \times X = 0 + 2 \times 0 = 0$

$Z = Z + Sh_z \times X = 0 + 3 \times 0 = 0$

Thus, New coordinates of corner A after shearing = (0, 0, 0).

Shearing - Example

<u>For Coordinates B(1, 1, 2)</u>

Let the new coordinates of corner B after shearing = (X’, Y’, Z’).

Applying the shearing equations, we have-

X’ = X = 1

Y’ = Y + Sh_y x X = 1 + 2 x 1 = 3

Z’ = Z + Sh_z x X = 2 + 3 x 1 = 5

Thus, New coordinates of corner B after shearing = (1, 3, 5).

<u>For Coordinates C(1, 1, 3)</u>

Let the new coordinates of corner C after shearing = (X’, Y’, Z’).

Applying the shearing equations, we have-

X’ = X = 1

Y’ = Y + Sh_y x X = 1 + 2 x 1 = 3

Z’ = Z + Sh_z x X = 3 + 3 x 1 = 6

Thus, New coordinates of corner C after shearing = (1, 3, 6).

Thus, New coordinates of the triangle after shearing in X axis = A (0, 0, 0), B(1, 3, 5), C(1, 3, 6).

Curve generation – Beziers Curves– B Spline Curves

Types of Curves

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**.

1. **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
   - It can represent multivalued curves (multiple y values for an x value). A common example is the circle, whose implicit representation is
   - $\mathbf{x^2 + y^2 - R^2 = 0}$

Types of Curves

2. Explicit Curves

- A mathematical function y = f(x) can be plotted as a curve. Such a function is the explicit representation of the curve. The explicit representation is not general, since it cannot represent vertical lines and is also single-valued. For each value of x, only a single value of y is normally computed by the function.

3. Parametric Curves

- Curves having parametric form are called parametric curves. The explicit and implicit curve representations can be used only when the function is known. In practice the parametric curves are used. A two-dimensional parametric curve has the following form −

$$P(t) = f(t), g(t) \text{ or } P(t) = x(t), y(t)$$

- The functions f and g become the (x, y) coordinates of any point on the curve, and the points are obtained when the parameter t is varied over a certain interval [a, b], normally [0, 1].

Types of Curves

Explicit: $y = f(x)$. Cannot represent closed or multiple valued curves.

Implicit: $f(x, y) = 0$. Represent larger class. But, difficult to compute points on the curve.

Parametric: Both explicit and implicit forms are axis dependent. It is difficult to compute points using these forms. Parametric forms are widely used to avoid these diffculties.

$x(u) = f_1(u)$ and $y(u) = f_2(u)$ for planar curves. Generally, the parametric range is normalized to $[0, 1]$ interval.

It is difficult to check membership using parametric forms.

Examples: $x = a + \ell u$, $y = b + mu$, $z = c + nu$ are parametric equations for a line in 3D. For $u = [0, 1]$, this is the line segment from $\mathbf{p}(0) = (a, b, c)$ to $\mathbf{p}(1) = ((a + \ell), (b + m), (c + n))$.

$x = u$, $y = u^2$, $z = u^3$ is a cubic curve in 3D...twisted parabola.

BEZIER CURVES

- A **polynomial** is a sum of variables raised to powers and multiplied by coefficients, e.g.,

$$\sum_{i=0}^{n} a_i x^n = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n.$$

- The **degree** (or **order**) of a polynomial is the highest power of the variables.

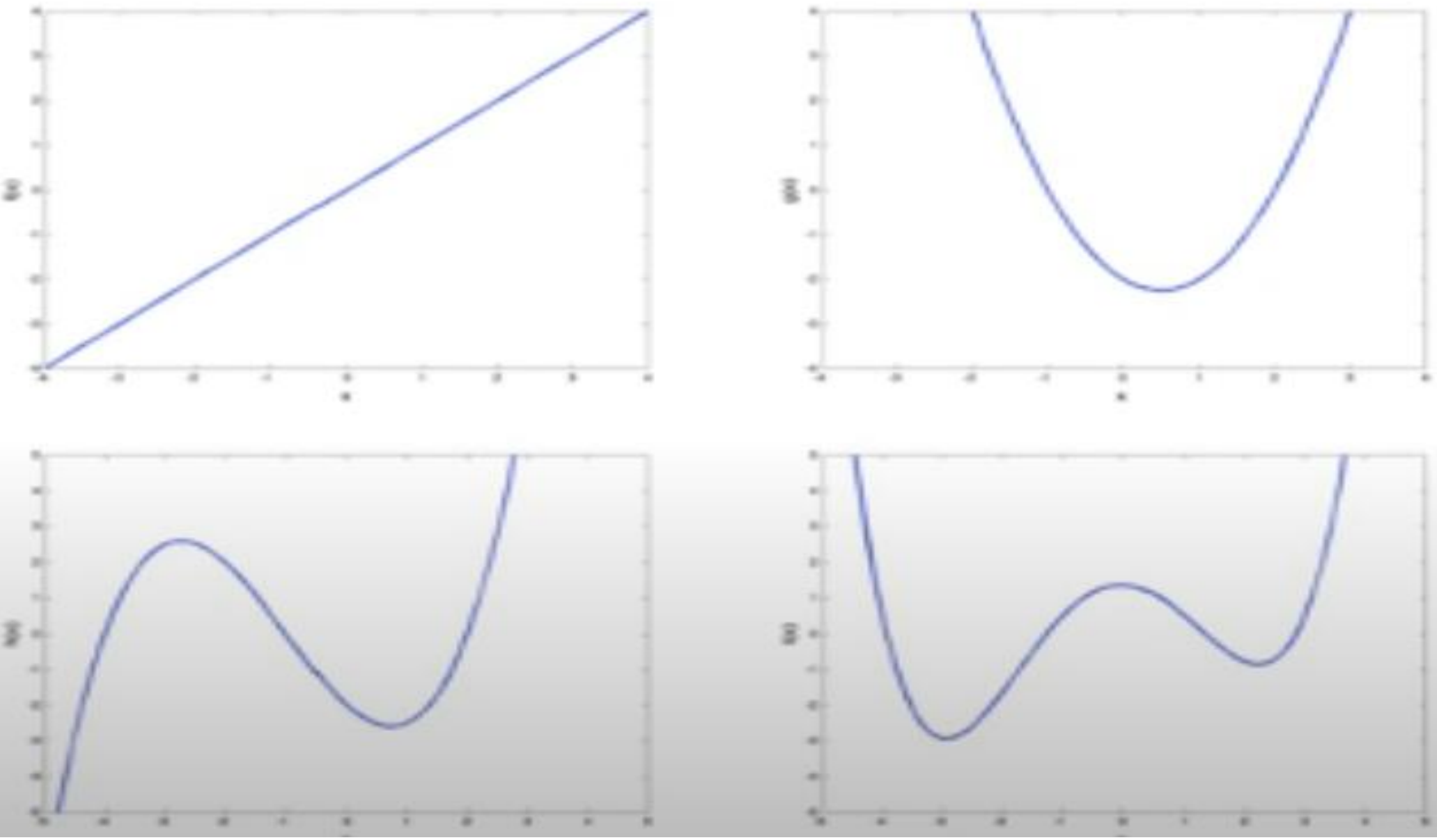
- Polynomials of different degrees are given names

$f(x) = x + 1,$	linear
$g(x) = x^2 + x + 1,$	quadratic
$h(x) = x^3 + x^2 + x + 1,$	cubic
$i(x) = x^4 + x^3 + x^2 + x + 1.$	quartic

BEZIER CURVES

Polynomials

- The higher the degree, the more changes of direction the polynomial can have



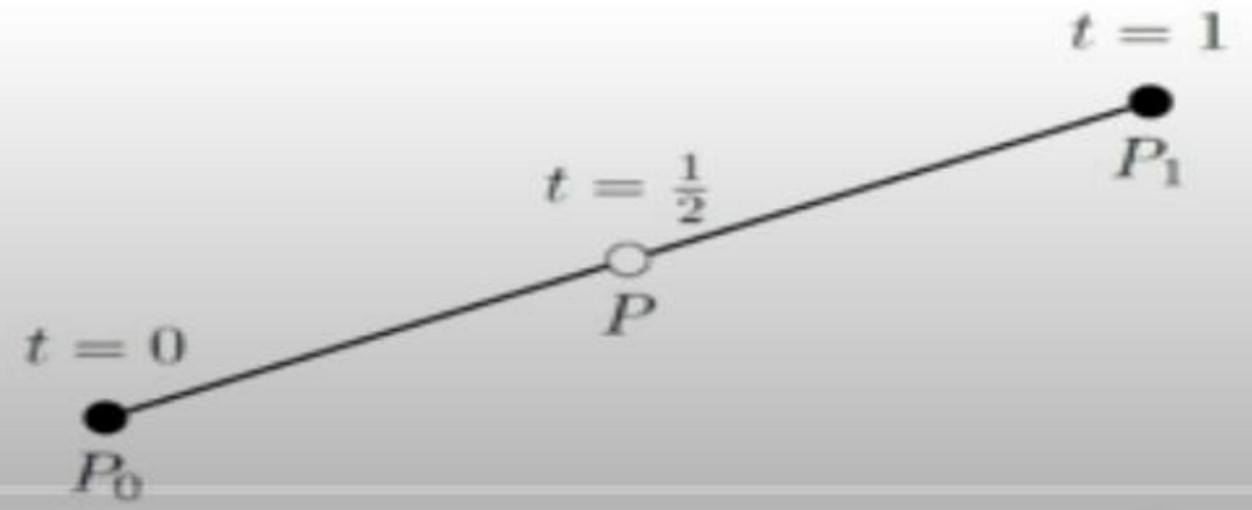
Parametric equations

- Bézier curves are expressed as **parametric equations**
- A parameter, t, is used to determine the value of the variables, e.g.,

$$x(t) = (1-t)x_0 + tx_1,$$
$$y(t) = (1-t)y_0 + ty_1,$$

where $0 \leq t \leq 1$. Let $P_0 = (x_0, y_0)$, $P_1 = (x_1, y_1)$ and $P = (x, y)$ then

$$P(t) = (1-t)P_0 + tP_1.$$



Bézier curves

- A Bézier curve is a parametric curve that is used to draw smooth lines
- Named after Pierre Bézier who used them for designing cars at Renault, actually invented by Paul de Casteljau 3 years earlier whilst working for Citroën
- Common applications include CAD software, 3D modelling and typefaces
- An n degree Bézier curve is defined using $n+1$ **control points**
- Translations can be easily applied to the control points

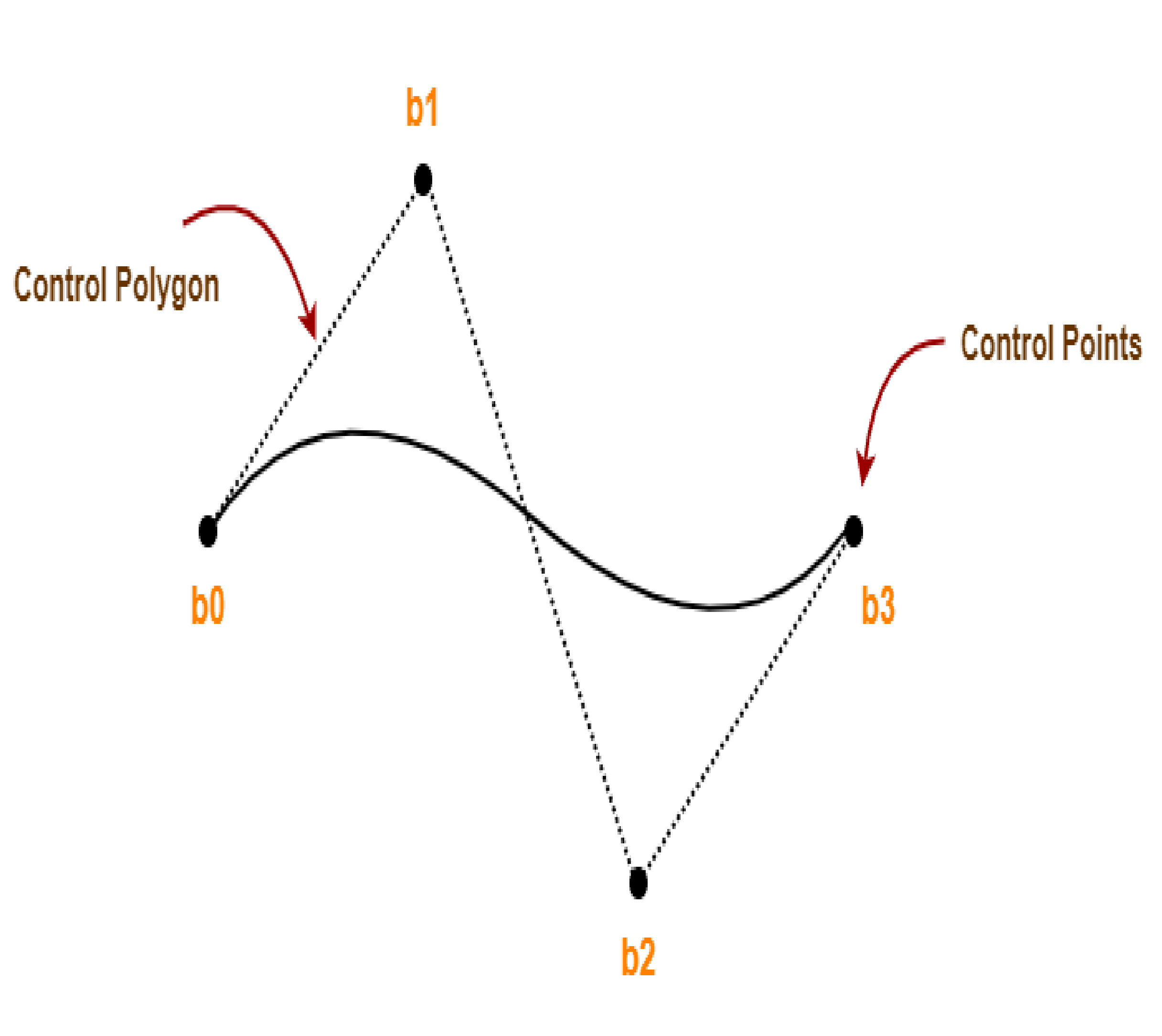
Bezier Curve

The concept of bezier curves was given by Pierre Bezier.

Bezier Curve may be defined as-

- Bezier are ends of the curve.
- Other Curve is parametric curve defined by a set of control points.
- Two points points determine the shape of the curve.

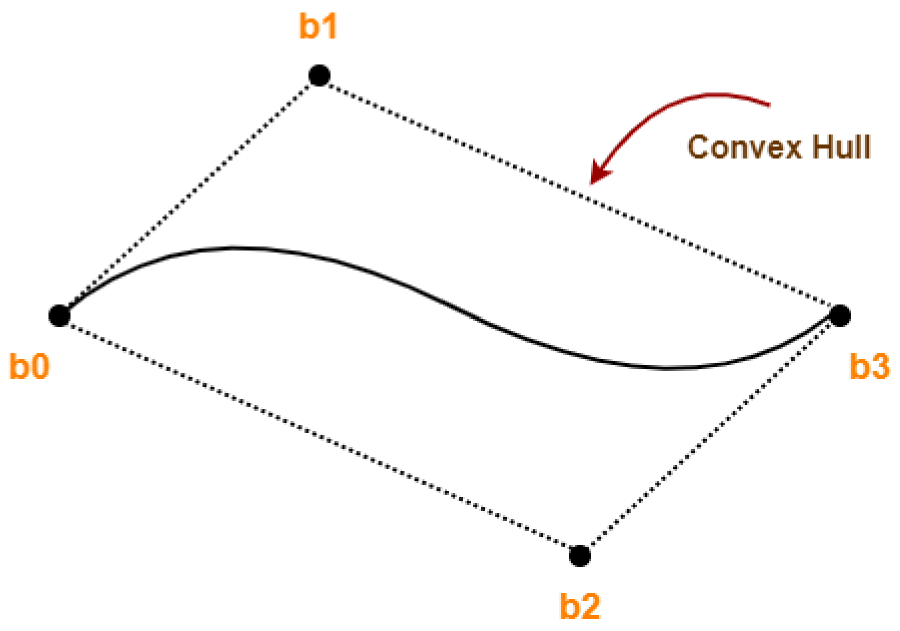
Example of a Bezier Curve



- Here,
- This bezier curve is defined by a set of control points b_0, b_1, b_2 and b_3.
- Points b_0 and b_3 are ends of the curve.
- Points b_1 and b_2 determine the shape of the curve.

Bezier Curve Properties

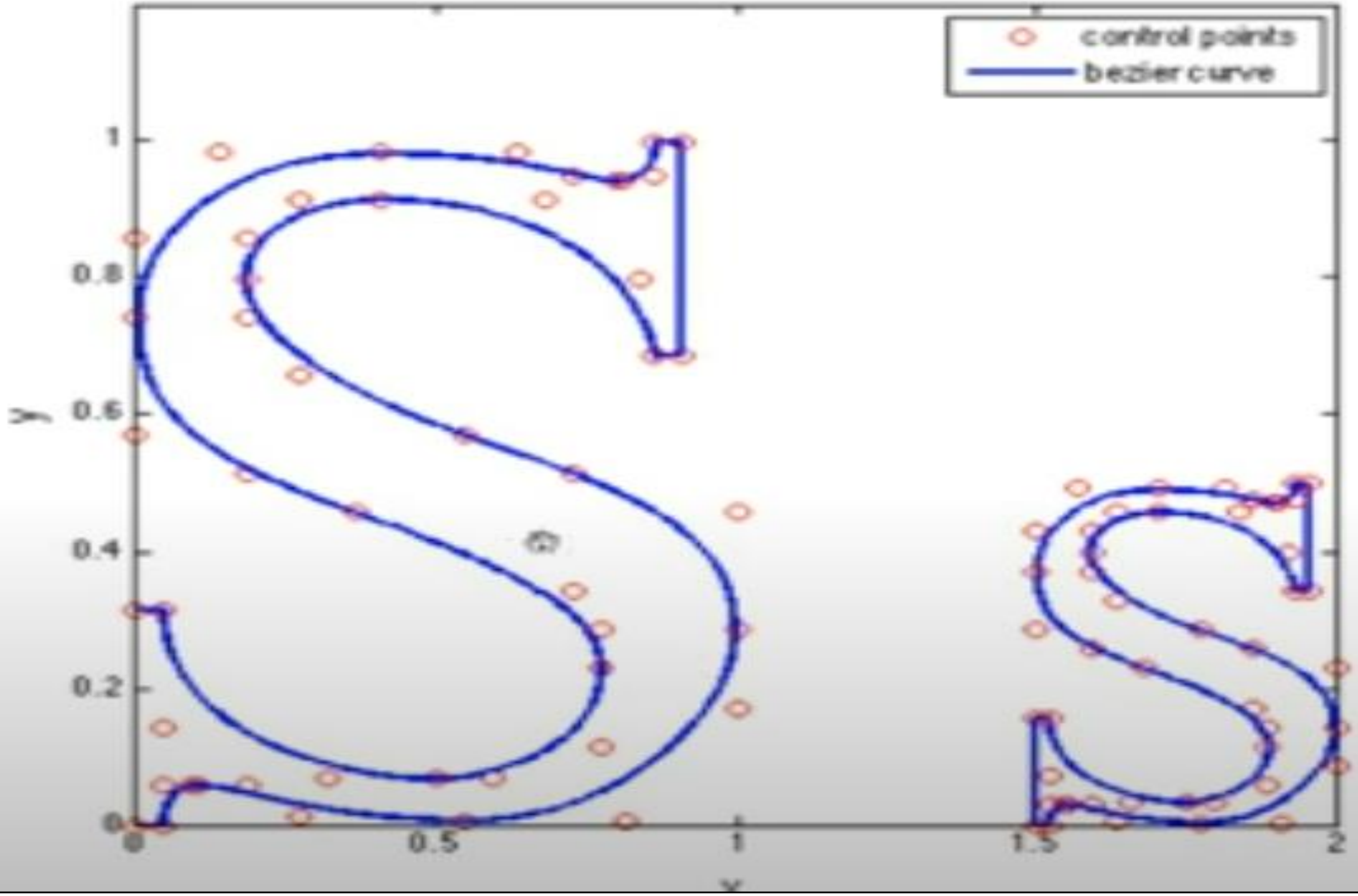
- Bezier curve is always contained within a polygon called as convex hull of its control points.



- Bezier curve generally follows the shape of its defining polygon.
- The first and last points of the curve are coincident with the first and last points of the defining polygon.
- The degree of the polynomial defining the curve segment is one less than the total number of control points.

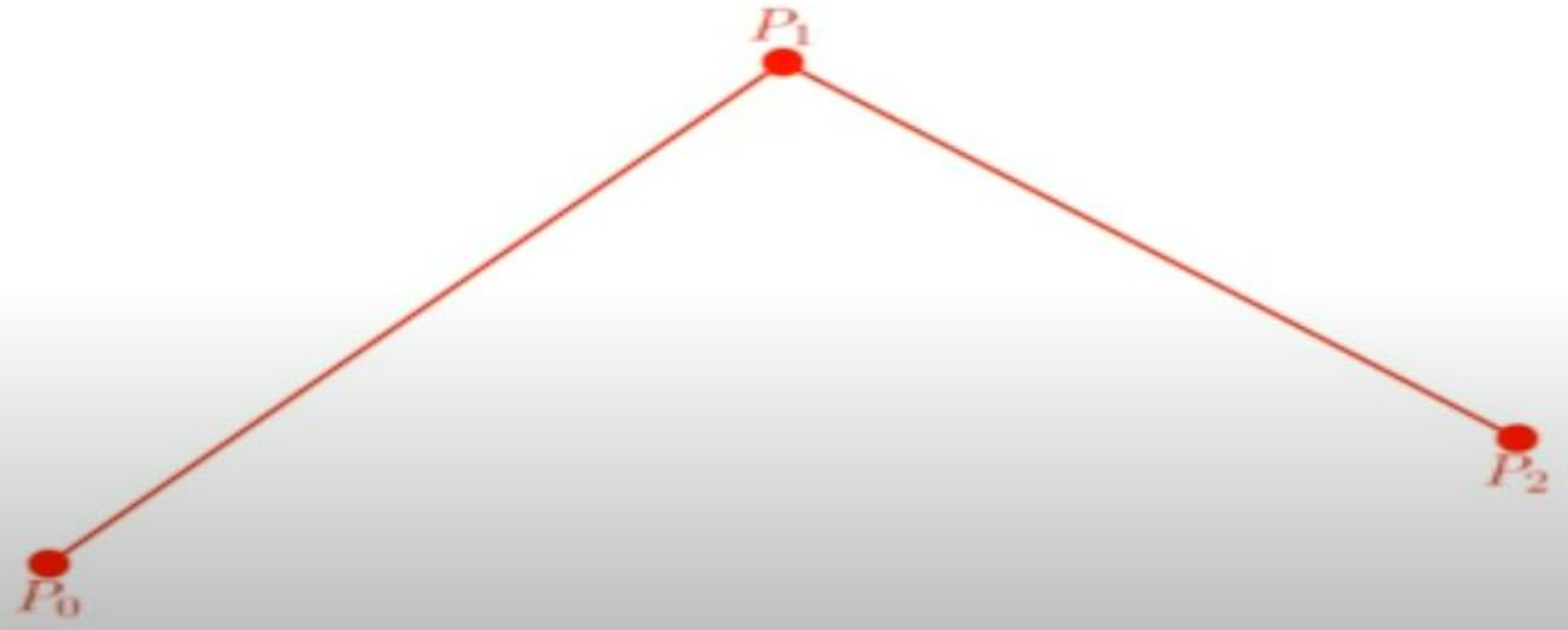
Degree = Num of Control Points – 1

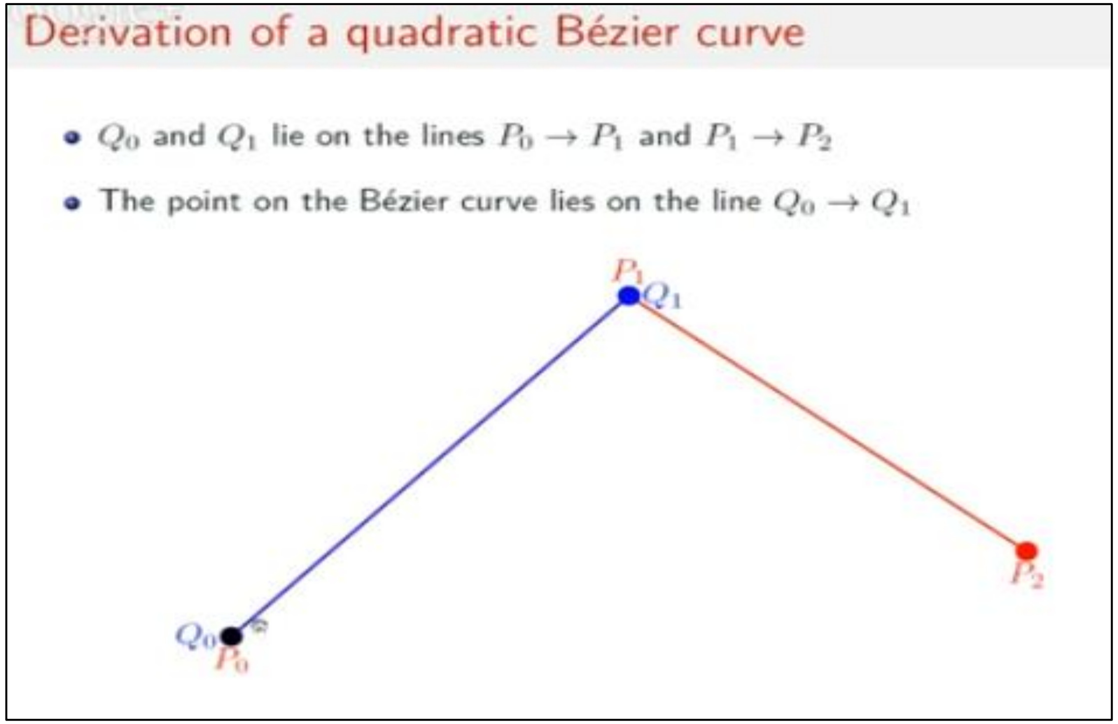
Bézier curves used to draw a typeface

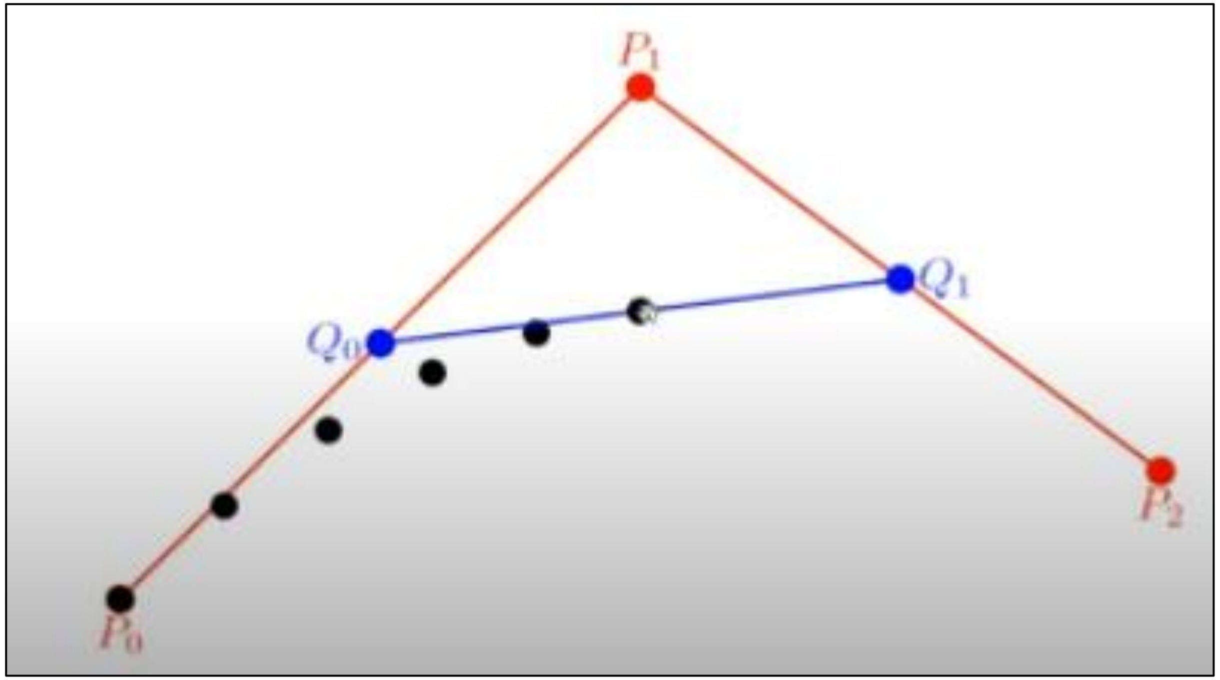


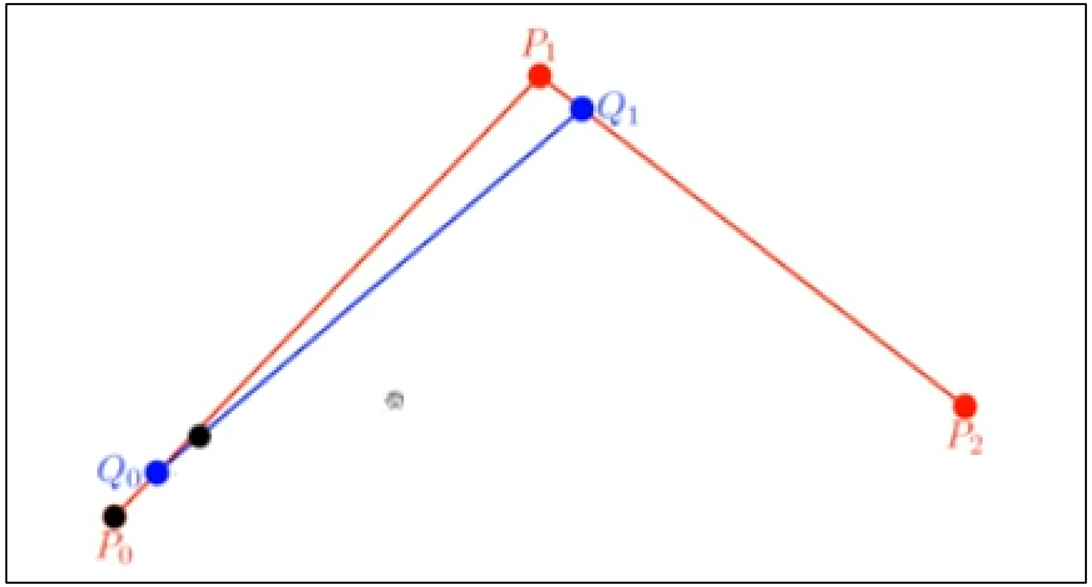
Derivation of a quadratic Bézier curve

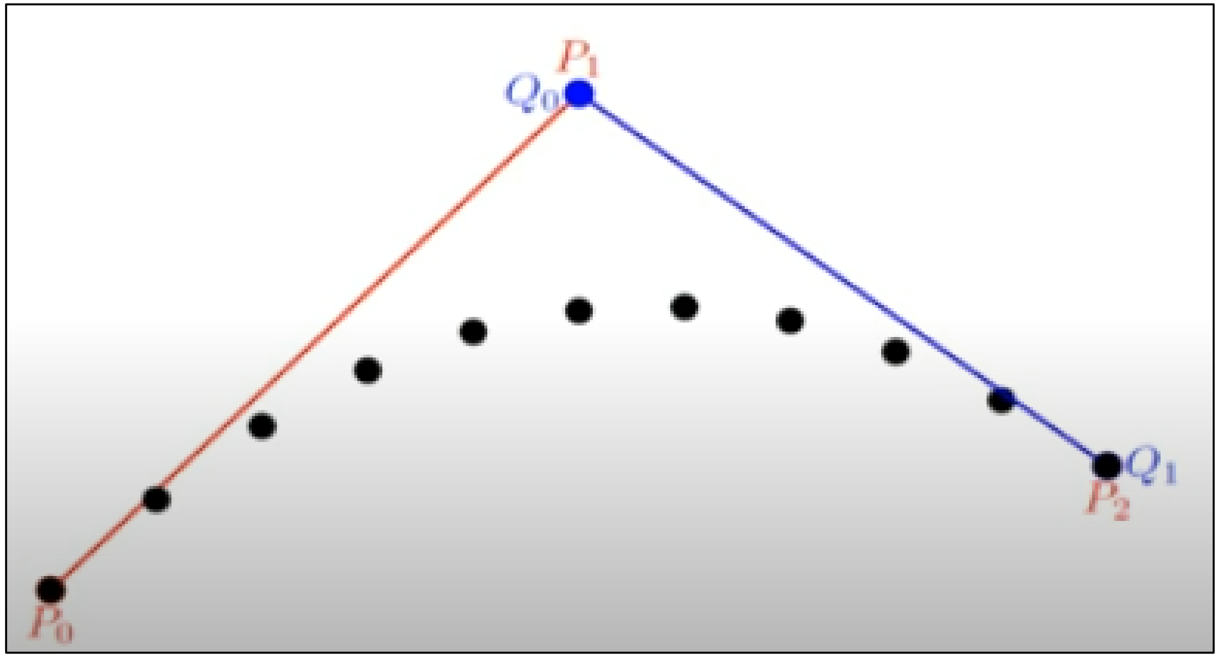
- Q_0 and Q_1 lie on the lines $P_0 \to P_1$ and $P_1 \to P_2$
- The point on the Bézier curve lies on the line $Q_0 \to Q_1$




Derivation of a quadratic Bézier curve
Q_0 and Q_1 lie on the lines $P_0 \to P_1$ and $P_1 \to P_2$
The point on the Bézier curve lies on the line $Q_0 \to Q_1$
P_1
Q_1
Q_0
P_0
P_2


P_1
Q_1
Q_0
P_2
P_0


P_1
Q_1
Q_0
P_0
P_2


P_1
Q_0
Q_1
P_2
P_0

Derivation of a quadratic Bézier curve

- Q_0 and Q_1 are points on the lines $P_0 \to P_1$ and $P_1 \to P_2$

$$Q_0 = (1-t)P_0 + tP_1,$$
$$Q_1 = (1-t)P_1 + tP_2.$$

- $C(t)$ is a point on the Bézier curve on the line $Q_0 \to Q_1$

$$C(t) = (1-t)Q_0 + tQ_1.$$

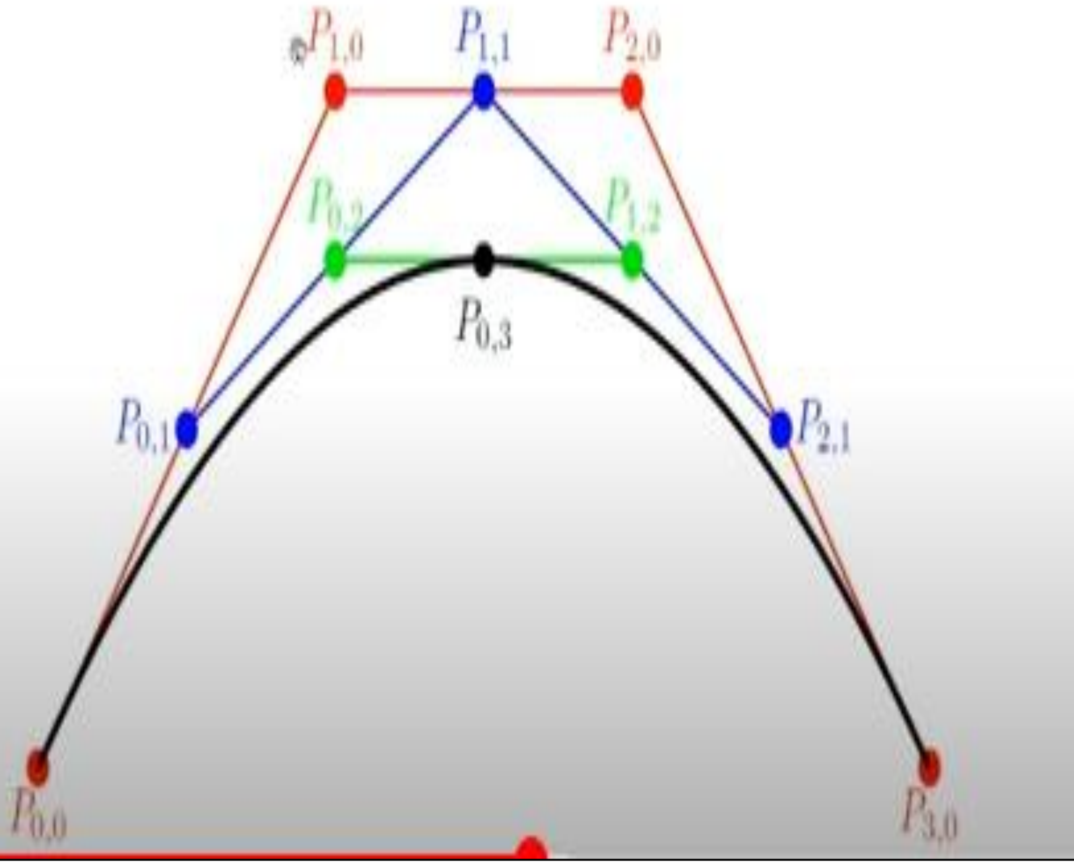
- Combining gives

$$C(t) = (1-t)^2 P_0 + 2t(1-t)P_1 + t^2 P_2.$$

Cubic Bézier curve

- A cubic Bézier curve is defined by 4 control points: $P_{0,0}$, $P_{1,0}$, $P_{2,0}$ and $P_{3,0}$

$$P_{0,3} = (1-t)^3 P_{0,0} + 3t(1-t)^2 P_{1,0} + 3t^2(1-t)P_{2,0} + t^3 P_{3,0}.$$



Degree n Bézier curves

- The general form of a degree n Bézier curve defined by the control points P_i (where $i = 0, 1, \ldots, n$) is

$$C(t) = \sum_{i=0}^{n} b_{i,n}(t) P_i,$$

where $b_{i,n}(t)$ are called **Bernstein polynomials** that are defined using

$$b_{i,n}(t) = \binom{n}{i} t^i (1-t)^{n-i},$$

and $\binom{n}{i}$ is the Binomial coefficient.

The binomial coefficient

- The **Binomial coefficient** is written using $\binom{n}{i}$ and is read as "n choose i" since it gives the number of ways of choosing i items from a set of n items

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

where $n!$ denotes the factorial of n

Cubic Bernstein polynomials

$$b_{i,n}(t) = \binom{n}{i} t^i (1-t)^{n-i},$$

- For example, the Bernstein polynomials for a cubic Bézier curve are

$$b_{0,3}(t) = \binom{3}{0} t^0 (1-t)^{3-0} = (1-t)^3,$$

$$b_{1,3}(t) = \binom{3}{1} t^1 (1-t)^{3-1} = 3t(1-t)^2,$$

$$b_{2,3}(t) = \binom{3}{2} t^2 (1-t)^{3-2} = 3t^2(1-t),$$

$$b_{3,3}(t) = \binom{3}{3} t^3 (1-t)^{3-3} = t^3,$$

Matrix form of a quadratic Bézier curve

- Consider the quadratic Bézier curve with the brackets expanded out

$$C(t) = (t^2 - 2t + 1)P_0 + (-2t^2 + 2t)P_1 + t^2 P_2,$$

this can be expressed in matrix form as

$$C(t) = \begin{pmatrix} P_0 & P_1 & P_2 \end{pmatrix} \begin{pmatrix} 1 & -2 & 1 \\ -2 & 2 & 0 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} t^2 \\ t \\ 1 \end{pmatrix}.$$

- Similarly a cubic Bézier curve can be expressed using

$$C(t) = \begin{pmatrix} P_0 & P_1 & P_2 & P_3 \end{pmatrix} \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} t^3 \\ t^2 \\ t \\ 1 \end{pmatrix}.$$

Properties of Bezier Curves

- They generally follow the shape of the control polygon, which consists of the segments joining the control points.
- They always pass through the first and last control points.
- They are contained in the convex hull of their defining control points.
- The degree of the polynomial defining the curve segment is one less that the number of defining polygon point. Therefore, for 4 control points, the degree of the polynomial is 3, i.e. cubic polynomial.
- A Bezier curve generally follows the shape of the defining polygon.
- The direction of the tangent vector at the end points is same as that of the vector determined by first and last segments.
- The convex hull property for a Bezier curve ensures that the polynomial smoothly follows the control points.
- No straight line intersects a Bezier curve more times than it intersects its control polygon.
- They are invariant under an affine transformation.
- Bezier curves exhibit global control means moving a control point alters the shape of the whole curve.
- A given Bezier curve can be subdivided at a point t=t0 into two Bezier segments which join together at the point corresponding to the parameter value t=t0.

Bezier Curves

Bezier curve is discovered by the French engineer **Pierre Bézier**. These curves can be generated under the control of other points. Approximate tangents by using control points are used to generate curve. The Bezier curve can be represented mathematically as –

$$\sum_{k=0}^{n} P_i B_i^n(t)$$

Where p_i is the set of points and $B_i^n(t)$

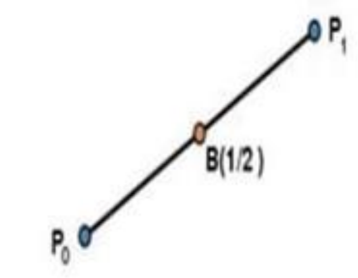
represents the Bernstein polynomials which are given by –

Blending function – $$B_i^n(t) = \binom{n}{i}(1-t)^{n-i}t^i$$

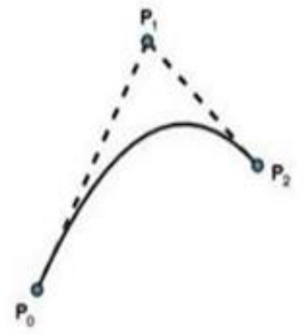
Where $$\binom{n}{i} = \frac{n!}{i!(n-i)!},$$

Where **n** is the polynomial degree, **i** is the index, and **t** is the variable.

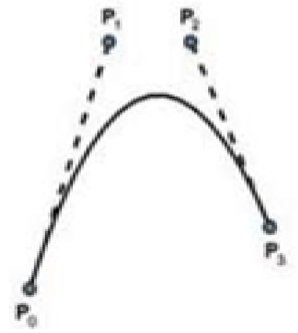
The simplest Bézier curve is the straight line from the point P_0 to P_1. A quadratic Bezier curve is determined by three control points. A cubic Bezier curve is determined by four control points.



Simple Bezier Curve



Quadratic Bazier Curve



Cubic Bazier Curve

BEZIER CURVE EXAMPLE

Refer the BEZIER CURVE - PDF

(separately uploaded in LMS)

Applications of Bezier Curves

1. Computer Graphics

- Bezier curves are widely used in computer graphics to model smooth curves.
- The curve is completely contained in the convex hull of its control points.
- So, the points can be graphically displayed & used to manipulate the curve intuitively.

2. Animation

- Bezier curves are used to outline movement in animation applications such as Adobe Flash and synfig.
- Users outline the wanted path in bezier curves.
- The application creates the needed frames for the object to move along the path.
- For 3D animation, bezier curves are often used to define 3D paths as well as 2D curves.

3. Fonts

- True type fonts use composite bezier curves composed of quadratic bezier curves.
- Modern imaging systems like postscript, asymptote etc use composite bezier curves composed of cubic bezier curves for drawing curved shapes.

Spline Representations

A spline is a smooth curve defined mathematically using a set of constraints.

- Splines have many uses:
    - 2D illustration
    - Fonts
    - 3D Modelling
    - Animation

Interpolation Vs Approximation

A spline curve is specified using a set of control points. There are two ways to fit a curve to these points:

- Interpolation - the curve passes through all of the control points
- Approximation - the curve does not pass through all of the control points

- Interpolation

“Polynomial sections are fitted so that the curve passes through each control point, Resulting curve is said to interpolate the set of control points.”

Interpolation curves are commonly used to digitize drawings or to specify animation paths.

A set of six control points interpolated with piecewise continuous polynomial sections.

- Approximation:

“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.”
Approximation curves are primarily used as design tools to structure object surfaces

Approximation spline surface created for a design application. Straight lines connect the control-point positions above the surface.

A set of six control points approximated with piecewise continuous polynomial sections

B-Spline Curves

The Bezier-curve produced by the Bernstein basis function has limited flexibility.

- First, the number of specified polygon vertices fixes the order of the resulting polynomial which defines the curve.
- The second limiting characteristic is that the value of the blending function is nonzero for all parameter values over the entire curve.

The B-spline basis contains the Bernstein basis as the special case. The B-spline basis is non-global. A B-spline curve is defined as a linear combination of control points Pi and B-spline basis function *Ni*, k (t) given by,

$$C(t) = \sum_{i=0}^{n} P_i N_{i,k}(t), \quad n \geq k-1, \quad t \in [tk-1, tn+1]$$

B-Spline Curves

$$C(t) = \sum_{i=0}^{n} P_i N_{i,k}(t), \qquad n \geq k-1, \qquad t \epsilon [tk-1, tn+1]$$

Where,

- $\{p_i$: i=0, 1, 2….n$\}$ are the control points
- k is the order of the polynomial segments of the B-spline curve. Order k means that the curve is made up of piecewise polynomial segments of degree k - 1,
- the $N_i, k(t)$ are the "normalized B-spline blending functions". They are described by the order k and by a non-decreasing sequence of real numbers normally called the "knot sequence".

$$t_i: i=0,\ldots n+K$$

The N_i, k functions are described as follows −

$$N_{i,1}(t) = \begin{cases} 1, & if\ u \epsilon [t_i, t_{i+1}) \\ 0, & Otherwise \end{cases}$$

and if k > 1,

$$N_{i,k}(t) = \frac{t-t_i}{t_{i+k-1}} N_{i,k-1}(t) + \frac{t_{i+k}-t}{t_{i+k}-t_{i+1}} N_{i+1,k-1}(t)$$

and

$$t \epsilon [t_{k-1}, t_{n+1})$$

Properties of B-spline Curve

- The sum of the B-spline basis functions for any parameter value is 1.
- Each basis function is positive or zero for all parameter values.
- Each basis function has precisely one maximum value, except for k=1.
- 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 number of vertices of defining polygon.
- B-spline allows the local control over the curve surface because each vertex affects the shape of a curve only over a range of parameter values where its associated basis function is nonzero.
- The curve exhibits the variation diminishing property.
- The curve generally follows the shape of defining polygon.
- Any affine transformation can be applied to the curve by applying it to the vertices of defining polygon.
- The curve line within the convex hull of its defining polygon.

Shape description requirements

The following are some of the important properties that are used to design the curves.

1. Control Points: The shape of the curve can be controlled easily with the help of a set of control points, means the points will be marked first and curve will be drawn that intersects each of these points one by one in a particular sequence. The more number of control points makes the curve smoother. The following figure shows the curve with control points.

Fig : Control points (Indicated by dots) govern the shape of the curve

Shape description requirements

2. Multiple values: In general any curve is not a graph of single valued function of a coordinate, irrespective the choice of coordinate system. Generally single valued functions of a coordinate make the curves or graphs that are dependent on axis. The following figure shows multivalued curve with respect to all coordinate systems.

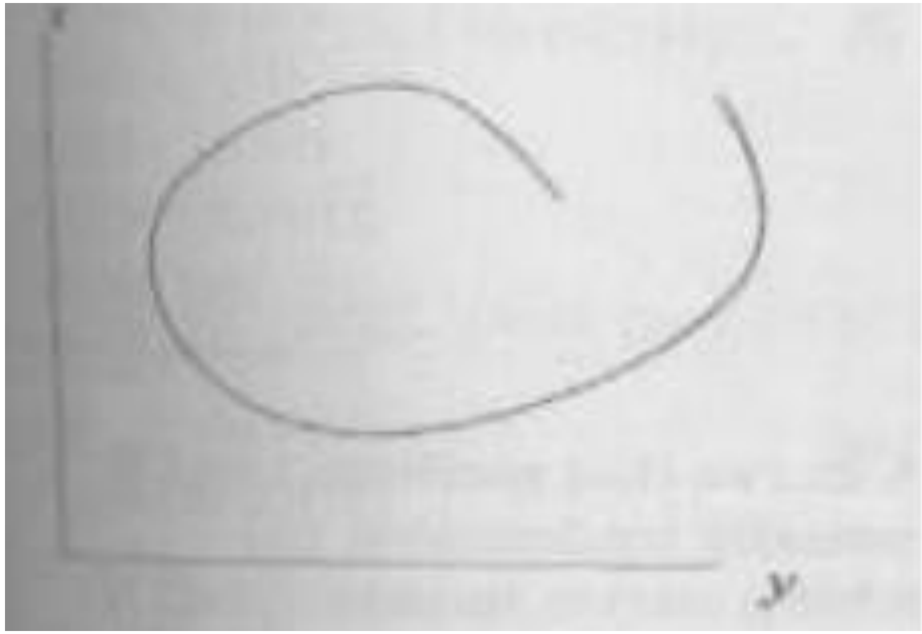


Fig: A curve can be multi-valued with respect to all coordinate systems.

Shape description requirements

- **Axis independence:** The shape of an object should not change when the control points are measured in different coordinate systems, that means when an object is rotated to certain angle in any direction (clockwise or anti-clockwise) the shape of the curve should not be affected.
- **Global or local control:** The control points of a curve must be controlled globally from any function of the same program or it can also be controlled locally by the particular function used to design that curve by calculating the desired control points.
- **Variation-diminishing property:** Some of the mathematical functions may diminish the curve at particular points and in some other points it may amplify the points. This leads to certain problems for curves appearance at the time of animations, (just as a vehicle looks curved when it is taking turn). This effect must be avoided with the selection of proper mathematical equations with multiple valued functions.
- **Versatility:** The functions that define the shape of the curve should not be limited to only few varieties of shapes, instead they must provide vide varieties for the designers to make the curves according to their interest.
- Order of continuity: For any complex shapes or curves or surfaces it is essential to maintain continuity in calculating control points. When we are not maintaining the proper continuity of control points it makes a mesh while marking the curve and the complex object.

Parametric functions

- In mathematics, a **parametric** equation defines a group of quantities as **functions** of one or more independent variables called parameters. Parameterizations are non-unique; more than one set of **parametric** equations can specify the same **curve**.
- The dominant form used to model curves and surfaces is the parametric or vector valued function. A point on a curve is represented as a vector: P(u)=[x(u) y(u) z(u)].
- For surfaces, two parametric are required: P(u, v)=[(x(u, v) y(u, v) z(u, v)].
- As the parametric u and v take on values in a specified range, usually 0 to 1, the parametric functions x, y and z trace out the location of the curve or surface. The parametric functions can themselves take many forms. A single curve be approximated in sever different ways as given below:

$$P(u) = [\cos u \ \sin u]$$

$$P(u) = [(1 - u^2)/(1 + u^2) \ 2u/(1 + u^2)]$$

$$P(u) = [u \ (1 - u^2)^{1/2}]$$

- By using simple parametric functions, we cannot expect the designer to achieve a desirable curve by changing coefficients of parametric polynomial functions or of any other functional form. Instead we must find ways to determine the parametric function from the location of control points that are manipulated by the designer.

References

- https://en.wikipedia.org/wiki/Gallery_of_curves