Geometric representation → UNIT-5

→ mathematical elem... of computer graphics (Rogers)

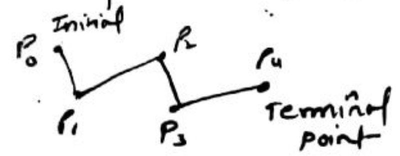
Major concept in computer graphics is modeling of objects.

↓

numerical description of the objects in terms of their geometric property (size, shape) and how they interact with light (reflect, transmit)

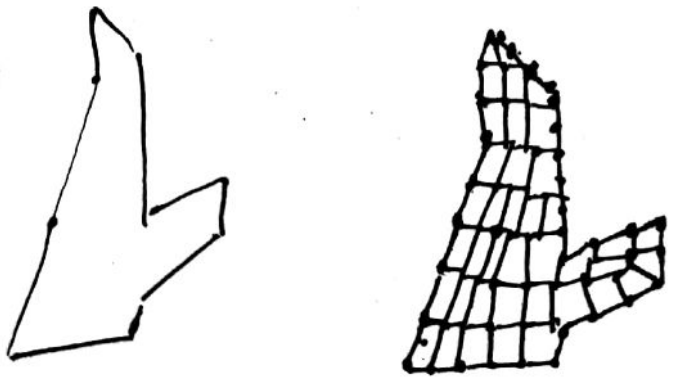
Simple geometric forms

- Points, lines
- Polylines → chain of connected line segments



Polygone → polygon is a closed polyline, that is, one in which the initial and terminal points coincide.

~~#||||~~ wireframe model → also called polygon mesh.



Polygon mesh or wireframe model

but drawback in wireframe model or mesh model is that it doesn't provide smooth representation.

For images in which most of the portion is curved, we will use curve to represent that image or object.

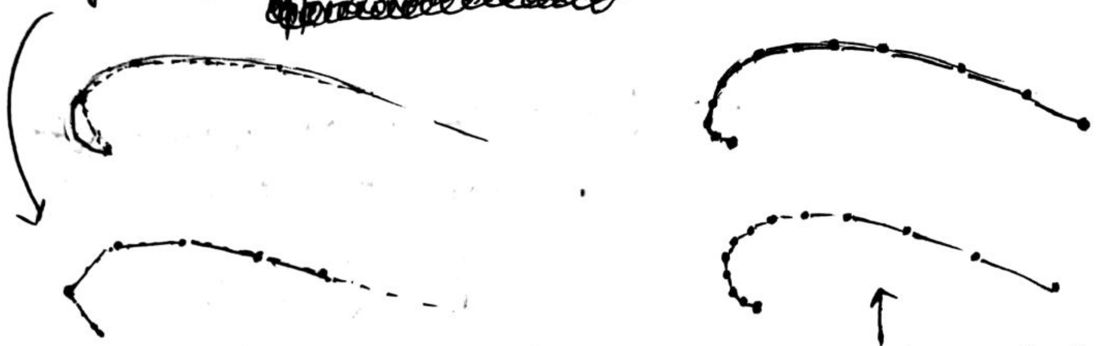
Plane curves → curves in 2D

Space curves → curves in 3D

Curve representation

→ A curve may be represented as a collection of points.

Polylines:



curve will be look like this
(a)

curve will be look like this
(b)

→ (b) representation will look better for curve representation

i.e In (a) equal point density along the curve.

(b) point density increases with decreasing radius of curvature.

Intermediate points must be obtained using interpolation.

interpolation is a method of constructing new data points within the range of discrete set of known points

Name → SR
B. Tech CSE
SAP ID → 50
Roll No →

Date 1

Experiment – 5

Title: Stack

Objective: The student will be able to (1) Implement the Stack using Link List and Arrays, (2) Implement different operations on Stack (3) Realize the applications of Stack.

Programs:

1.write a C program for implementing stack using array and perform FUSH and POP operations using functions.

```
#include <stdio.h>
#include<conio.h>
#define MAX 5
int top, status;
void push (int stack[], int item)
{   if (top == (MAX-1))
status = 0;
else
{   status = 1;
++top;
stack [top] = item;
}
}
int pop (int stack[j)
{
int ret;
if (top == -1)
{   ret = 0;
status = 0;
}
else
{   status = 1;
ret = stack [top];
--top;
}
return ret;
}
void display (int stack[])
{   int i ;
printf ("\n The Stack is: ");
if (top == -1)
printf ("empty");
else
{   for (i=top; i>=0; --i)
```

→ parametric or a nonparametric form is used to represent a curve.

→ A non parametric representation is either explicit or implicit.

→ for a plane curve, an explicit nonparametric form is given by $y = f(x)$

ex. eq. of straight line
$y = mx + c$
(for each x-value only one y value is obtained)

→ for closed or multiple value curves, we cannot represent explicit form.

eg $\boxed{\text{a circle cannot be represented explicitly}}$.

$x^2 + y^2 - R^2 = 0 \Rightarrow f(x,y) = 0$

Scanned by CamScanner

high degree approximation

Unit-5

curve representation → why? → to draw models like real object (If polygon mesh used, they are not real).

Nonparametric form	Parametric form	Implicit form	explicit form
$y = f(x)$ *	$x = x(t)$ $y = y(t)$	$f(x,y) = 0$ $x^2 + y^2 - R^2 = 0$	$y = mx + c$ *

→ most of the curve representation follow parametric form.

→ eq. of different ~~curves~~ conic section – parabola, hyperbola, ellipse, circle ← is a special case of ellipse.

we can generate different types of curves by varying the parameters of these curves (if we know the equations of these curves)

They are ~~generated~~ derived with the help of generalized curve representation called conic sections.

⇓

A conic section is the intersection of a plane and a cone.

In 2D, intersection of two lines gives a point

In 3D, " " " planes or surfaces gives a curve.

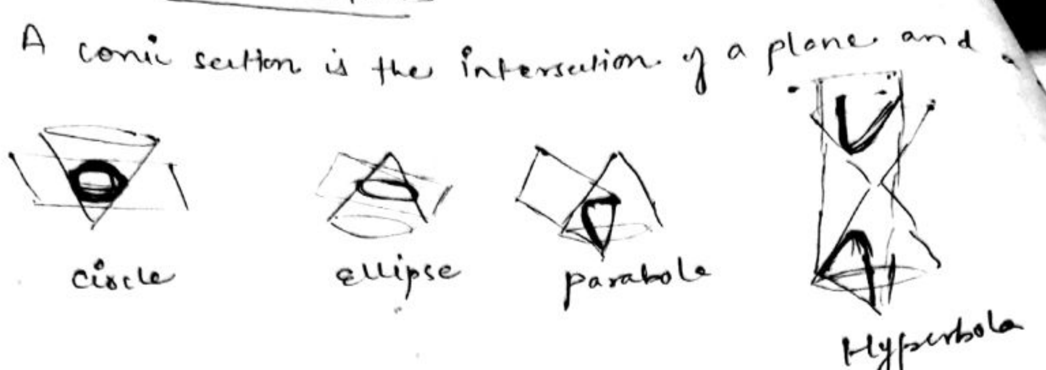
A general second-degree implicit equation written as

$$ax^2 + 2bxy + cy^2 + 2dx + 2ey + f = 0$$

provides a wide variety of 2D curve forms called conic sections.

Conic sections

A conic section is the intersection of a plane and a



circle ellipse parabola Hyperbola

By changing the angle and location of intersection, we can produce a circle, ellipse, parabola or hyperbola.

or the special case when the plane touches the vertex:

a point, line or 2 intersecting lines.

point line Double line

General equation of conic section (Implicit form)

This form of the expression, provide a wide variety of 2D curve forms called Conic sections.

$$Ax^2 + 2Bxy + Cy^2 + 2Dx + 2Ey + F = 0$$

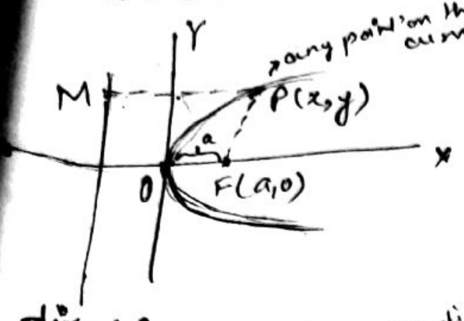
$\boxed{\text{Why 2?}}$

because of symmetry

By varying the coefficients A, B, G, D, E, F, we can derive equations of different curves. (in 2D)

3 special classes of curves under conic section.

Parabola.



$y^2 = 4ax$ $\quad a>0$

focus $(a,0)$

directrix $= -a$

eccentricity, $e=1$

directrix → a line perpendicular to the axis of symmetry
$x=-a$

parametric form

$$\boxed{x = at^2, \quad y = \pm 2at}$$

or $\boxed{x = \tan^2\phi, \quad y = \pm 2\sqrt{a\tan\phi}}$

eccentricity $= \dfrac{PF}{PM}$

↓

is the ratio of the distance of point on the curve from the focus to the distance of point on the curve from the directrix

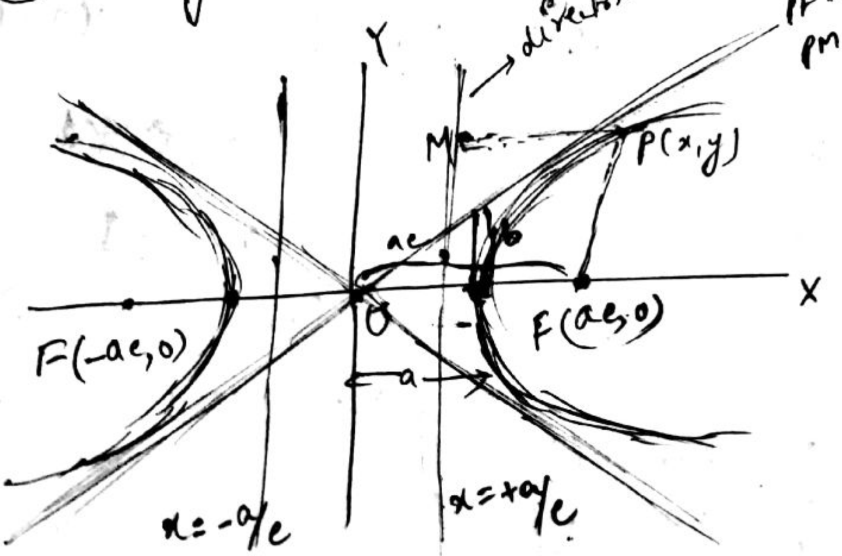
for parabola, $PF = PM$

$\therefore \boxed{e=1}$

[e cannot be -ve because it is a ratio of two distances and distance is always a positive no.]

(2). hyperbola → (a, b) ask ?

we know
$e = \dfrac{PF}{PM}$
$PF = (x-ae)^2 + (y-0)^2$
$PM = (x-\frac{a}{e})^2 + (y-y)^2$



$$\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1$$

where, $b^2 = a^2(e^2-1)$

foci: $(\pm ae, 0)$

Directrices: $x = \pm a/e$

$e>1$ $\quad \boxed{PF > PM}$

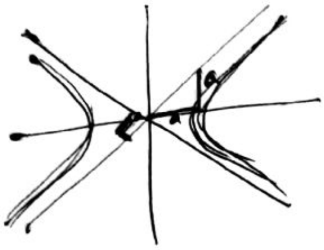
Parametric form

$x = a\sec t$

$y = b\tan t$

$\boxed{-\frac{\pi}{2} < t < \frac{\pi}{2}}$

rectangular hyperbola or equilateral hyperbola or right hyperbola.



when $a = b$

it becomes rectangular hyperbola

$\therefore$ two asymptotes are $\perp$ to each other (Perpendicular)

tangents on the curve

$\therefore$ eq. $x^2 - y^2 = a^2$

we know $b^2 = a^2(e^2-1)$

$(a=b)$ $\quad a^2 = a^2(e^2-1)$

$1 = e^2 - 1$

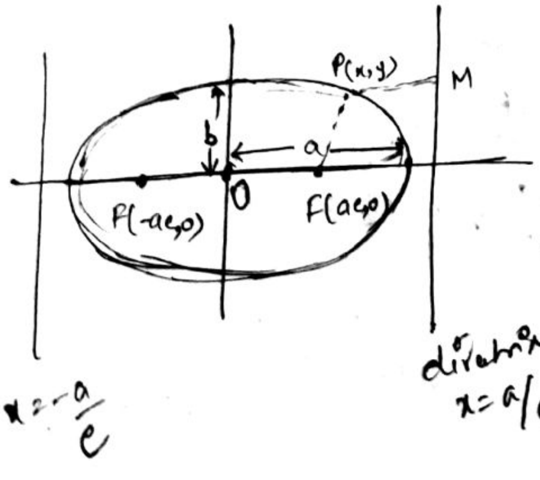
$e^2 = 2$

$\boxed{e = \sqrt{2}}$

③ Ellipse

$\boxed{e<1}$

$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$



$e = \frac{PF}{PM}$

$\therefore \boxed{PM > PF}$

$a \geq b > 0$

$b^2 = a^2(1-e^2)$

$0 < e < 1$

foci : $(\pm ae, 0)$

directrices: $x = \pm a/e$

Parametric form

$x = a\cos t$, $y = b\sin t$ $\quad t \in [-\pi, \pi]$

ow, work on general equation of conic

$$Ax^2 + 2Bxy + Cy^2 + 2Dx + 2Ey + F = 0$$

if the conic passes through the origin : $F = 0$

(i.e. $x=0, y=0$, we can calculate $F=0$)

→ remaining 5 coefficients, may be obtained using 5 geometric conditions :

say :

Boundary conditions -

- two (2) end points .
- slope of the curves at two (2) end points
- one (1) intermediate point .

using these $2+2+1=5$ conditions we can calculate A, B, C, D, E

Matrix form of the general equation of conic

$$XSX^T = 0 \quad , S \text{ is symmetric matrix}$$

$$\begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} A & B & D \\ B & C & E \\ D & E & F \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0$$

After multiplication we will get -

$$\boxed{Ax^2 + 2Bxy + Cy^2 + 2Dx + 2Ey + F = 0}$$

Special conditions

$B^2 = AC$ $(ac-b^2=0)$ (the eq. represents a Parabola) Rob

(R

$B^2 < AC$ $\boxed{AC-B^2>0}$ (" " " an Ellipse)

$B^2 > AC$ $\boxed{AC-B^2<0}$ (" " " hyperbola)

to Problems ①

→ D, E, F represents the linear part of the curve only A, B, C coefficients decide the form of curve.

Proof Suppose $Ax^2 + 2Bxy + Cy^2 = 1$

Ask?

matrix form

$$\begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} A & B \\ B & C \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$S = \begin{bmatrix} A & B \\ B & C \end{bmatrix}$$

eigen Values ~~Discriminant~~

$$\begin{vmatrix} A-d & B \\ B & C-d \end{vmatrix} = 0$$

$$(A-d)(C-d) - B^2 = 0$$

$$AC - Ad - Cd + d^2 - B^2 = 0$$

$$d^2 - d(A+C) + (AC - B^2) = 0$$

$\therefore$ roots $\dfrac{+(A+C) \pm \sqrt{[-(A+C)]^2 - 4\times 1 \times (AC-B^2)}}{2\times 1}$ roots $\dfrac{-b \pm \sqrt{b^2-4ac}}{2a}$

Shreedharacharya formula

if $AC = B^2$, we will get $0, (A+C)$ two ~~roots~~ values

$\therefore$ one zero value & one positive value

* Problems in non parametric form.
(Reason to switch from non-parametric form to parametric form).

Problems ① → Both explicit & implicit non-parametric curve representations are axis dependent.

$f(x,y)=0$

$y=f(x)$

(to calculate y value, we have to substitute x)

but in parametric curve rep., it is axis independent.

$\left.\begin{array}{l} x = x(t) \\ y = y(t) \end{array}\right]$ depends on third variable

→ Thus the choice of coordinate system affects the ease of use.

② when points on axis dependent non parametric curve are calculated at equal increments in x or y, they are not evenly distributed along the curve length. This unequal distribution of points affects the quality and accuracy of graphical representations.

eg. $x^2+y^2=1$

$y=\sqrt{1-x^2}$

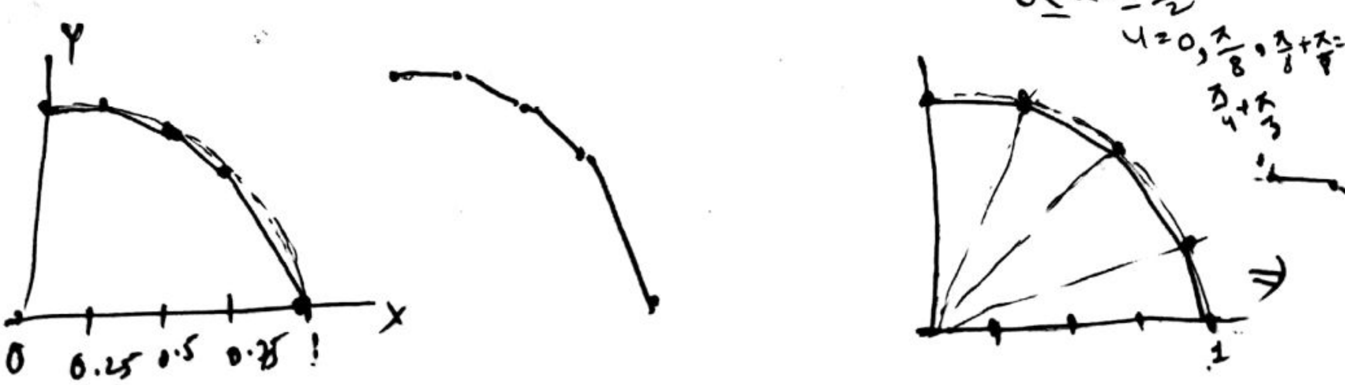
$x=\cos u$

$y=\sin u$

→ difference $\frac{\pi}{8}$

$0 \le u \le \frac{\pi}{2}$

$u=0, \frac{\pi}{8}, \frac{\pi}{8}+\frac{\pi}{8}=\frac{\pi}{4}, \frac{\pi}{4}+\frac{\pi}{8}$



$AC > B^2$, we will get both +ve value value

$\boxed{AC - B^2 > 0}$ ellipse. (eg. $A=2, B=1, C=3$)

If $AC < B^2$, we will get one +ve & one −ve value

hyperbola. (eg $A=1, B=2, C=3$)

(to explain eigen value → eg. girl (eigenvector) increase in heartbeat (eigenvalue)))

In conic sections, we have studied the intersection of plane & cone.

but the intersection of two quaradic (or non linear) surfaces produces a curve.

eg. circle & cone intersects produces a curve.

Parametric representation of cubic curves in 3D

Parametric rep. of curve

$X = x_1 + t\Delta x$

$Y = Y_1 + t\Delta y$

par eq of helix

$x = r\cos(t)$

$y = r\sin(t)$

$z = bt$

$b \neq 0$, $-\infty < t < \infty$

b gives the direction.

change in z is called pitch of the helix.

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

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

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

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

$$= T \cdot C$$

where $T = [\,t^3 \;\; t^2 \;\; t \;\; 1\,]$ and $C = \begin{bmatrix} a_x & a_y & a_z \\ b_x & b_y & b_z \\ c_x & c_y & c_z \\ d_x & d_y & d_z \end{bmatrix}$

$$Q(t) = [T][C]_{4\times 3}$$

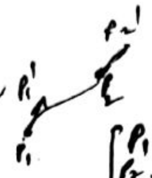
$$\Downarrow$$

$$T\cdot M\cdot G \longrightarrow [\ \ ]_{1\times 4}[\ \ ]_{4\times 4}[\ \ ]_{4\times 3}$$

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

$M = [m_{ij}]_{4\times 4}$ and $G = [g_1\ g_2\ g_3\ g_4]^T$

$$= \underbrace{\begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ g_4 \end{bmatrix}}_{4\times 3} \rightarrow \text{these are points}$$



$$\begin{bmatrix} P_1 \\ P_2 \\ P_1' \\ P_2' \end{bmatrix}$$

$M \Rightarrow 4\times4$ basis matrix

$G \Rightarrow$ four element column vector of geometric constants, called geometric vector.

The curve is a weighted sum of the elements of the geometry matrix.

The weights are each cubic polynomials of t, and are called blending function

$$B = T\cdot M$$

Parametric curves

Parametric form each coordinate of a point on a curve represented as a function of a single parameter.

The position vector of a point on the curve is fixed by the value of the parameter.

$$x = x(t)$$
$$y = y(t)$$

Position vector of a point on the curve is

$$P(t) = [\,x(t) \;\; y(t)\,]$$

As the point moves, the position vector will change in length or in direction or in both length & direction.

$$P'(t) = [\,x'(t) \;\; y'(t)\,]$$

derivative or tangent vector on a parametric curve (differentiation with respect to the parameter).

Slope of the curve $\Rightarrow$ $\dfrac{dy}{dx} = \dfrac{dy}{dt} * \dfrac{dt}{dx}$

$$\boxed{\frac{dy}{dx} = \frac{y'(t)}{x'(t)}}$$

When $x'(t) = 0$

slope is infinite

Q) Determine the type of conic:

$$2x^2 - 72xy + 23y^2 + 140x - 20y + 50 = 0$$

$A = 2$, $2B = -72$ $C = 23$

$B = -36$

B^2 AC

$(-36)^2 > 2 \times 23 = 46$

$\boxed{B^2 > AC}$ i.e hyperbola.

Curve fitting

Curve fairing

→ A curve that passes through all known data points is said to fit the data.

Cubic spline & parabolically blended curves → curve fitting techniques

They are characterized by the fact that the derived mathematical curve passes through each & every data point.

→ Alternatively, the mathematical description of a space curve is generated without any prior knowledge of the curve shape or form.

Bezier curves & B-spline curves are curve fairing techniques

they are characterized by the fact that few if any points on the curve pass through the control points used to define the curve.

→ Bezier curve $\xrightarrow{\text{follows}}$ approximation

B-spline →

→ cubic spline $\xrightarrow{\text{follow}}$ interpolation

```
char x;
int op1,op2,val;
init(&s);
printf("Enter the expression:");

while((x=getchar())!='\n')
{
  if(isdigit(x))
    push(&s,x-48);
  else
  {
    op2=pop(&s);
    op1=pop(&s);
    val=evaluate(x,op1,op2);
    push(&s,val);
  }
}

val=pop(&s);
printf("\nValue of expression=%d",val);

return 0;
}

int evaluate(char x,int op1,int op2)
{
  if(x=='+')
    return(op1+op2);
  if(x=='-')
    return(op1-op2);
  if(x=='*')
    return(op1*op2);
  if(x=='/')
    return(op1/op2);
  if(x=='%')
    return(op1%op2);
}

void init(stack *s)

s->top=-1;

empty(stack *s)

(s->top==-1)
return(1);

urn(0);
```

Interpolation spline → when the curve passes through all the control points.

Approximation spline → when the curve does not passes through the control points.

cubic splines → representation used to represent curves

spline → flexible strip used to produce a smooth curve through a designated set of points.

$$\boxed{P(t) = \sum_{i=1}^{4} B_i t^{i-1} \; ;} \quad t_1 \le t \le t_2 \quad — (1)$$

($i = 4$, because of cubic polynomial)

⇒ $P(t)$ is the position vector of any point on the cubic spline segment.

B_1, B_2, B_3, B_4 → control points that will control the shape of the curve.

$$P(t) = [x(t), y(t), z(t)]$$ → cartesian coordinate

break the elements in terms of x, y, z

$$x(t) = \sum_{i=1}^{4} B_{ix} t^{i-1}$$

$$y(t) = \sum_{i=1}^{4} B_{iy} t^{i-1} \qquad t_1 \le t \le t_2$$

$$z(t) = \sum_{i=1}^{4} B_{iz} t^{i-1}$$

From equation (1)

$$P(t) = B_1 t^0 + B_2 t^1 + B_3 t^2 + B_4 t^3$$

$$= B_1 + B_2 t + B_3 t^2 + B_4 t^3 \qquad t_1 \le t \le t_2$$

∴ B_i's will control the shape of the cubic splines

Scanned by CamScanner

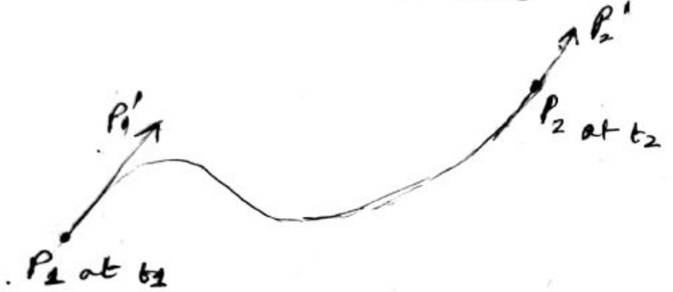
Calculations of B_i's (use boundary conditions to coefficients)

Take the derivative of eq (I)

$$P'(t) = \sum_{i=2}^{4} (i-1)\, B_i\, t^{(i-2)}$$

$$\boxed{P'(t) = B_2 + 2B_3 t + 3B_4 t^2} \quad \text{—— (II)}$$

Suppose there is a curve



P_1' & P_2' are the derivative (or tangent) at starting and endpoints.

let's assume $t_1 = 0$

at point P_1, $t_1 = 0$

$$P(0) = P_1 \qquad P(t_2) = P_2$$
$$P'(0) = P_1' \qquad P'(t_2) = P_2'$$

∴ we have 4 boundary conditions, we can calculate B_1, B_2, B_3, B_4 with the help of these boundary conditions.

$$P(0) = \boxed{P_1 = B_1}$$
$$P'(0) = \boxed{P_1' = B_2}$$

From eq (I) and (II)

$$P(t_2) = P_2 = B_1 + B_2 t_2 + B_3 t_2^2 + B_4 t_2^3$$
$$P'(t_2) = P_2' = B_2 + 2B_3 t_2 + 3B_4 t_2^2$$

(III) & (IV)

ulate B_3, B_4

$$B_1 + B_2 t_2 + B_3 t_2^2 + B_4 t_2^3 = P_2$$ [$B_1 = P_1$]

$$P_1 + B_2 t_2 + B_3 t_2^2 + B_4 t_2^3 = P_2$$ [$B_2 = P_1'$]

$$P_1 + P_1' t_2 + B_3 t_2^2 + B_4 t_2^3 = P_2 \quad \text{— (V)}$$

eq (IV)

$$B_2 + 2B_3 t_2 + 3B_4 t_2^2 = P_2'$$

$$P_1' + 2B_3 t_2 + 3B_4 t_2^2 = P_2' \quad \text{— (VI)}$$

Subtract eq (V) and (VI)

$$P_1 + P_1' t_2 + B_3 t_2^2 + B_4 t_2^3 = P_2 \quad * 2$$

$$P_1' + 2B_3 t_2 + 3B_4 t_2^2 = P_2' \quad * t_2$$

$$2P_1 + 2P_1' t_2 + \cancel{2B_3 t_2^2} + 2B_4 t_2^3 = 2P_2$$

$$P_1' t_2 + \cancel{2B_3 t_2^2} + 3B_4 t_2^3 = P_2' t_2$$

$$- \quad - \quad \quad - \quad \quad -$$

$$2P_1 + P_1' t_2 - B_4 t_2^3 = 2P_2 - P_2' t_2$$

$$2P_1 - 2P_2 + P_1' t_2 + P_2' t_2 = B_4 t_2^3$$

$$\boxed{B_4 = \frac{2(P_1 - P_2)}{t_2^3} + \frac{P_1'}{t_2^2} + \frac{P_2'}{t_2^2}} \quad \text{— (VII)}$$

Take eqn (V) and (VI)

$$P_1 + P_1' t_2 + B_3 t_2^2 + B_4 t_2^3 = P_2 \quad * 3$$

$$P_1' + 2B_3 t_2 + 3B_4 t_2^2 = P_2' \quad * t_2$$

eq (9) is for → To represent spline seg

$$3P_1 + 3P_1' t_2 + 3B_3 t_2^2 + \cancel{3B_4 t_2^3} = 3P_2$$

$$P_1' t_2 + 2B_3 t_2^2 + \cancel{3B_4 t_2^3} = P_2' t_2$$

$$- \quad - \quad - \quad -$$

$$3P_1 + 2P_1' t_2 + B_3 t_2^2 = 3P_2 - P_2' t_2$$

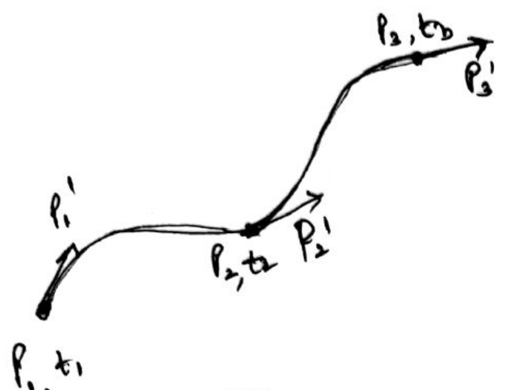
$$B_3 t_2^2 = 3P_2 - P_2' t_2 - 3P_1 - 2P_1' t_2$$

$$\boxed{B_3 = \frac{3(P_2 - P_1)}{t_2^2} - \frac{P_2'}{t_2} - \frac{2P_1'}{t_2}} \quad \text{(VIII)}$$

$$\boxed{P(t) = P_1 + P_1' t + \left[\frac{3(P_2 - P_1)}{t_2^2} - \frac{P_2'}{t_2} - \frac{2P_1'}{t_2}\right] t^2 + \left[\frac{2(P_1 - P_2)}{t_2^3} + \frac{P_1'}{t_2^2} + \frac{P_2'}{t_2^2}\right] t^3} \quad (9)$$

eq (9) is for single cubic spline segment

→ To represent a complete curve, multiple cubic spline segments are joined together.



Two piecewise cubic spline segments

→ This point should be continuous.

⇒ a piecewise spline of degree K has continuity of order K-1 at the internal points.

∴ cubic spline has second order continuity at the internal points.

this means that the second derivative $P_2''(t)$ is continuous across the point.

Other approaches for continuity are →
- Normalized cubic splines
- Blending functions
- Weighting functions

① a space without shape or form. eg. skin of car.

m.

Scanned by CamScanner

From eq (11) $\longrightarrow$ $P''(t) = \sum_{i=3}^{4} (i-1)(i-2) B_i \, t^{i-3}$

$$P''(t) = 2B_3 + 6B_4 t$$

(1)

i ? a space e without curve shape or form : eg. skin of car.

It is continuous at t_2

$\therefore t = t_2$

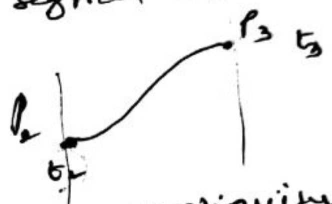
$$P''(t_2) = 2B_3 + 6B_4 t_2$$

For second cubic spline segment parameter range is ren.

$0 \le t \le t_3$.

$\therefore t_2 = 0$

this segment is for



$$P'' = 2B_3$$

equate both the above conditions to ensure the continuity, i.e end point of first curve is the starting point of the second curve.

$$\underbrace{2B_3 + 6B_4 t_2}_{\text{from 1st curve}} = \underbrace{2B_3}_{\text{from 2nd curve}}$$

$$2 * \left[\frac{3(P_2 - P_1)}{t_2^2} - \frac{2P_1'}{t_2} - \frac{P_2'}{t_2}\right] + 6t_2\left[\frac{2(P_1 - P_2)}{t_2^3} + \frac{P_1'}{t_2^2} + \frac{P_2'}{t_2^2}\right]$$

$$= 2\left[\frac{3(P_3 - P_2)}{t_3^2} - \frac{2P_2'}{t_3} - \frac{P_3'}{t_3}\right] \quad \text{(10)}$$

$$\cancel{6t_2\left[\frac{2(P_1 - P_2)}{t_2^3} + \frac{P_1'}{t_2^2} + \frac{P_2'}{t_2^2}\right] + 2\left[3(P_2\right.}$$

multiply the eq (10) by $t_2 t_3$

$$t_2 t_3\left[\frac{3(P_2-P_1)}{t_2^2}-\frac{2P_1'}{t_2}-\frac{P_2'}{t_2}\right]+6t_2^2 t_3\left[\frac{2(P_1-P_2)}{t_2^3}+\frac{P_1'}{t_2^2}+\frac{P_2'}{t_2^2}\right]$$

$$=2t_2\left[\frac{3(P_3-P_2)}{t_3^2}-\frac{2P_2'}{t_3}-\frac{P_3'}{t_3}\right]$$

$$\frac{6t_3}{t_2}(P_2-P_1)-4P_1't_3-2P_2't_3+\frac{12t_3}{t_2}(P_1-P_2)+6P_1't_3$$

$$+6t_3P_2'=\frac{6t_2}{t_3}(P_3-P_2)-4P_2't_2-2P_3't_2$$

$$-\frac{\cancel{6}^{3}t_3}{t_2}(P_2-P_1)+\cancel{4}^{2}P_1't_3+\cancel{4}^{2}P_2't_3+\cancel{4}^{2}P_2't_2+\cancel{2}P_3't_2$$

$$=\frac{\cancel{6}^{3}t_2(P_3-P_2)}{t_3}$$

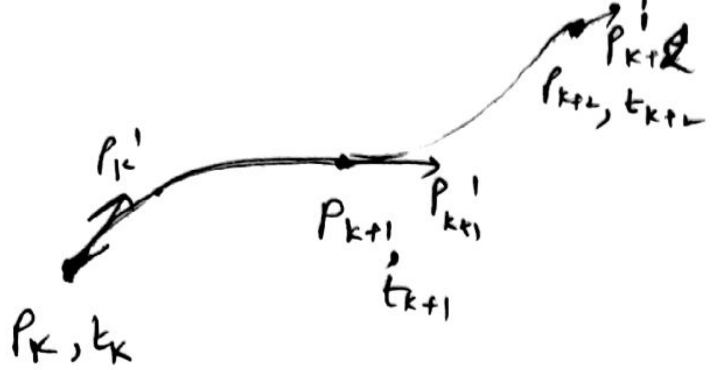
$$2P_2'(t_3+t_2)+t_2P_3'+t_3P_1'=\frac{3t_2(P_3-P_2)}{t_3}+\frac{3t_3(P_2-P_1)}{t_2}$$

$$\boxed{2P_2'(t_3+t_2)+t_2P_3'+t_3P_1'=\frac{3}{t_2t_3}\left[t_2^2(P_3-P_2)+t_3^2(P_2-P_1)\right]}$$

(11)

from eq (11), we can solve P_2', the unknown tangent vector.

→ if we have multiple cubic spline segments.



generalized eqn for any two adjacent cubic spline segments, $P_k(t)$ and $P_{k+1}(t)$ (refer from eq 9)

for first segment

$$\boxed{P_k(t) = P_k + P_k' t + \left[\frac{3(P_{k+1}-P_k)}{t_{k+1}^2} - \frac{P_{k+1}'}{t_{k+1}} - \frac{2P_k'}{t_{k+1}}\right]t^2 + \left[\frac{2(P_k-P_{k+1})}{t_{k+1}^3} + \frac{P_k'}{t_{k+1}^2} + \frac{P_{k+1}'}{t_{k+1}^2}\right]t^3} \quad \text{— (12)}$$

for second segment

$$\boxed{P_{k+1}(t) = P_{k+1} + P_{k+1}' t + \left[\frac{3(P_{k+2}-P_{k+1})}{t_{k+2}^2} - \frac{P_{k+2}'}{t_{k+2}} - \frac{2P_{k+1}'}{t_{k+2}}\right]t^2 + \left[\frac{2(P_{k+1}-P_{k+2})}{t_{k+2}^3} + \frac{P_{k+1}'}{t_{k+2}^2} + \frac{P_{k+2}'}{t_{k+2}^2}\right]t^3} \quad \text{— (13)}$$

Scanned by CamScanner

$$
\begin{bmatrix}
t_3 & 2(t_2+t_3) & t_2 & 0 & \cdots & & & \\
0 & t_4 & 2(t_3+t_4) & t_3 & 0 & \cdots & & \\
0 & 0 & t_5 & 2(t_4+t_5) & t_4 & 0 & \cdots & \\
 & & & \vdots & \vdots & & & \\
 & & & 0 & t_n & 2(t_n+t_{n-1}) & t_{n-1} &
\end{bmatrix}_{(n-2)\times n} \times
$$

n-2 rows
n columns

$$
\begin{bmatrix} P_1' \\ P_2' \\ P_3' \\ \vdots \\ P_n' \end{bmatrix}_{(n\times 1)} = \begin{bmatrix} \frac{3}{t_2 t_3}\{t_2^2(P_3-P_2)+t_3^2(P_2-P_1)\} \\ \frac{3}{t_3 t_4}\{t_3^2(P_4-P_3)+t_4^2(P_3-P_2)\} \\ \vdots \\ \frac{3}{t_{n-1} t_n}\{t_{n-1}^2(P_n-P_{n-1})+t_n^2(P_{n-1}-P_{n-2})\} \end{bmatrix}_{n-2\times}
$$

$$[M^*][P'] = R$$

$\downarrow$

is not square matrix (we cannot find inverse of M^*

to calculate $\boxed{P' = M^{*-1} R}$

we need inverse of M^{*-1}

So, Assume that end tangent vectors P_1' & P_n' are known, then the problem becomes determinant.

So, the matrix will become —

Recall that parameter range begins at zero for each segment,
for the first segment $0 \le t \le t_{k+1}$ and for the second segment $0 \le t \le t_{k+2}$.

For any two adjacent spline segments, equating the second derivatives at the common internal joint

letting $P_k''(t_k) = P_{k+1}''(0)$

From eq (11)

$$\boxed{2P_{k+1}'(t_{k+2}+t_{k+1}) + t_{k+1}P_{k+2}' + P_k' t_{k+2} = \frac{3}{t_{k+1}t_{k+2}}\left[t_{k+1}^2(P_{k+2}-P_{k+1}) + t_{k+2}^2(P_{k+1}-P_k)\right]} \quad \text{— (14)}$$

$$1 \le k \le \underbrace{n}_{\text{no. of points}} - 2$$

eq (14) is for calculating the tangent vector at the internal joint b/w any two spline segments P_k & P_{k+1}

eq (14) in matrix form

$$\begin{bmatrix} t_3 & 2(t_3+t_2) & t_2 \\ & & \\ & & \end{bmatrix} \begin{bmatrix} P_1' \\ P_2' \\ P_3' \end{bmatrix} = \begin{bmatrix} \frac{3}{t_2 t_3}\left[t_2^2(P_3-P_2) + t_3^2(P_2-P_1)\right] \end{bmatrix}$$

write this matrix in recursive form for all spline segments.

$$
\begin{bmatrix}
0 & \cdots & & & & & \\
t_3 & 2(t_2+t_3) & t_2 & 0 & \cdots & & \\
0 & t_4 & 2(t_3+t_4) & t_3 & 0 & \cdots & \\
0 & 0 & t_5 & 2(t_4+t_5) & t_4 & 0 & \cdots \\
 & & & \vdots & \vdots & & \\
 & & & 0 & t_n & 2(t_n+t_{n-1}) & t_{n-1} \\
 & & & 0 & 0 & 0 & 1
\end{bmatrix}_{n \times n}
\times
$$

$$
\begin{bmatrix} P_1' \\ P_2' \\ P_3' \\ \vdots \\ P_n' \end{bmatrix}_{n \times 1}
=
\begin{bmatrix}
P_1' \\
\frac{3}{t_2 t_3}\left\{ t_2^2 (P_3 - P_2) + t_3^2 (P_2 - P_1) \right\} \\
\frac{3}{t_3 t_4}\left\{ t_3^2 (P_4 - P_3) + t_4^2 (P_3 - P_2) \right\} \\
\vdots \\
\frac{3}{t_{n-1} t_n}\left\{ t_{n-1}^2 (P_n - P_{n-1}) + t_n^2 (P_{n-1} - P_{n-2}) \right\} \\
P_n'
\end{bmatrix}_{n \times 1}
\qquad (15)
$$

Once the P_k's are known, the B_i coefficients for each spline segments can be calculated –

$$B_{1k} = P_k$$

$$B_{2k} = P_k'$$

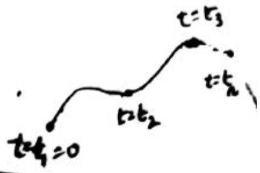
$$B_{3k} = \frac{3(P_{k+1} - P_k)}{t_{k+1}^2} - \frac{2P_k'}{t_{k+1}} - \frac{P_{k+1}'}{t_{k+1}}$$

$$B_{4k} = \frac{2(P_k - P_{k+1})}{t_{k+1}^3} + \frac{P_k'}{t_{k+1}^2} + \frac{P_{k+1}'}{t_{k+1}^2}$$

In matrix form

$$[B] \Rightarrow \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \frac{-3}{t_{k+1}^2} & \frac{-2}{t_{k+1}} & \frac{3}{t_{k+1}^2} & \frac{-1}{t_{k+1}} \\ \frac{2}{t_{k+1}^3} & \frac{1}{t_{k+1}^2} & \frac{-2}{t_{k+1}^3} & \frac{1}{t_{k+1}^2} \end{bmatrix} \begin{bmatrix} P_k \\ P_k' \\ P_{k+1} \\ P_{k+1}' \end{bmatrix} = \begin{bmatrix} B_{1k} \\ B_{2k} \\ B_{3k} \\ B_{4k} \end{bmatrix} \quad \text{--- (16)}$$

∴ generalized form of eq (1) is



$$\boxed{P_k(t) = \sum_{i=1}^{4} B_{ik}\, t^{i-1} \qquad 0 \le t \le t_{k+1}, \quad 1 \le k \le n-1} \quad \text{--- (17)}$$

In matrix form eq (17) becomes

$$P_k(t) = \begin{bmatrix} 1 & t & t^2 & t^3 \end{bmatrix}_{1\times4} \begin{bmatrix} B_{1k} \\ B_{2k} \\ B_{3k} \\ B_{4k} \end{bmatrix}_{4\times1} \quad \text{--- (18)}$$

T.C.

Substituting eq (16) in (18)

$$P_k(t) = \begin{bmatrix} 1 & t & t^2 & t^3 \end{bmatrix} \underbrace{\underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \frac{-3}{t_{k+1}^2} & \frac{-2}{t_{k+1}} & \frac{3}{t_{k+1}^2} & \frac{-1}{t_{k+1}} \\ \frac{2}{t_{k+1}^3} & \frac{1}{t_{k+1}^2} & \frac{-2}{t_{k+1}^3} & \frac{1}{t_{k+1}^2} \end{bmatrix}_{4\times 4}}_{\text{basis matrix}}}_{\text{blending function matrix}} \underbrace{\begin{bmatrix} P_k \\ P_k' \\ P_{k+1} \\ P_{k+1}' \end{bmatrix}_{4\times 1}}_{\text{Geometric vector matrix}}$$

$$P_k(t) = \begin{bmatrix} \underbrace{1 + t^2 * \frac{-3}{t_{k+1}^2} + t^3 * \frac{2}{t_{k+1}^3}} & \underbrace{t + \frac{-2}{t_{k+1}} * t^2 + t^3 * \frac{1}{t_{k+1}^2}} & \underbrace{\frac{3}{t_{k+1}^2} * t^2 + \frac{-2}{t_{k+1}^3} * t^3} & \frac{-1}{t_{k+1}} * t^2 + \frac{1}{t_{k+1}^2} * t^3 \end{bmatrix} \times \begin{bmatrix} P_k \\ P_k' \\ P_{k+1} \\ P_{k+1}' \end{bmatrix}$$

~~Suppose~~ $\cancel{\tau = \frac{t}{t_{k+1}}}$

$\cancel{P_k(\tau) = \Big[\, 1 - 3\tau^2 + 2\tau^3}$

$\cancel{P_k\left(\frac{t}{t_{k+1}}\right) = \Big[\, 1 + \left(\frac{t}{t_{k+1}}\right)^2 * \frac{-3}{t_{k+1}^2}}$

Suppose $\tau = \frac{t}{t_{k+1}}$

$$P_k(\tau) = \begin{bmatrix} 1 - 3\tau^2 + 2\tau^3 & t(1 - 2\tau + \tau^2) & 3\tau^2 - 2\tau^3 & t(-\tau + \tau^2) \end{bmatrix} \times \begin{bmatrix} P_k \\ P_k' \\ P_{k+1} \\ P_{k+1}' \end{bmatrix}$$

Substitute value of t and arrange

$$P_k(\tau) = \begin{bmatrix} 1-3\tau^2+2\tau^3 & 3\tau^2-2\tau^3 & \tau t_{k+1}(1-2\tau+\tau)^2 & \tau t_{k+1}(\tau^2-\tau) \end{bmatrix} \begin{bmatrix} P_k \\ P_{k+1} \\ P_k' \\ P_{k+1}' \end{bmatrix}$$

i.e.

$$P_k(\tau) = \begin{bmatrix} F_{1k}(\tau) & F_{2k}(\tau) & F_{3k}(\tau) & F_{4k}(\tau) \end{bmatrix} \begin{bmatrix} P_k \\ P_{k+1} \\ P_k' \\ P_{k+1}' \end{bmatrix}$$

where

$$\left.\begin{aligned} F_{1k}[\tau] &= 2\tau^3-3\tau^2+1, \\ F_{2k}[\tau] &= 3\tau^2-2\tau^3 \\ F_{3k}[\tau] &= \tau(\tau^2-2\tau+1)\, t_{k+1} \\ F_{4k}[\tau] &= \tau(\tau^2-\tau)\, t_{k+1} \end{aligned}\right\}$$ Are called blending functions or weighting functions.

In matrix form)

$$\boxed{P_k(\tau) = [F]\,[G]}$$

To calculate G we have to find tangent vectors of intermediate points.

Normalized cubic splines → computationally inexpensive in comparison to cubic splines

→ Normalize the t value for each segment

$0 \le t \le 1$

Here
$t_{k+1} = 1$
$T = t$

Now blending functions will become:

$$
\left.\begin{aligned}
F_1(t) &= 2t^3 - 3t^2 + 1 \\
F_2(t) &= 3t^2 - 2t^3 \\
F_3(t) &= t(t^2 - 2t + 1) \\
&= t^3 - 2t^2 + t \\
F_4(t) &= t^3 - t^2
\end{aligned}\right\}
$$

These ~~blending~~ functions are called cubic Hermite polynomial blending functions

For the normalized cubic spline the blending function matrix will become -

$$[F] = [T][N]$$

$$= \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

The matrix eq. for cubic spline segment can now be written as:

$$\boxed{P(t) = [F][G]}$$

$$= \underbrace{[T][N]}_{\text{Blending function matrix}}[G]$$

normalized base matrix

geometric vector matrix

[redacted] ([redacted]).

Consider the four 2D position vectors $P_1[0\ 0]$, $P_2[1\ 1]$, $P_3[2\ -1]$ and $P_4[3\ 0]$ with tangent vectors $P_1'[1\ 1]$ and $P_4'[1\ 1]$. Determine the normalized piecewise cubic spline curve through them.

Sol:- Here the t_k parameter values are $t_2 = t_3 = t_4 = 1.0$

The internal tangent vectors are obtained using eq -

$$\begin{bmatrix} 1 & 0 & \cdots & & \\ 1 & 4 & 1 & 0 & \cdots \\ 0 & 1 & 4 & 1 & 0 \cdots \\ & & \vdots & & \\ & 0 & 1 & 4 & 1 \\ & & & 0 & 1 \end{bmatrix} \begin{bmatrix} P_1' \\ P_2' \\ \vdots \\ P_n' \end{bmatrix} = \begin{bmatrix} P_1' \\ 3\{(P_3-P_2)+(P_2-P_1)\} \\ 3\{(P_4-P_3)+(P_3-P_2)\} \\ \vdots \\ P_n' \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 4 & 1 & 0 \\ 0 & 1 & 4 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_1' \\ P_2' \\ P_3' \\ P_4' \end{bmatrix} = \begin{bmatrix} [1 \quad 1] \\ 3\{[2\ -1]-[1\ 1]+[1\ 1]-[0\ 0]\} \\ 3\{[3\ 0]-[2\ -1]+[2\ -1]-[1\ 1]\} \\ [1 \quad 1] \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 1 \\ 6 & -3 \\ 6 & -3 \\ 1 & 1 \end{bmatrix}$$

using row reduction method

Inverting and premultiplying yields:

$$\begin{bmatrix} P_1' \\ P_2' \\ P_3' \\ P_4' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -0.267 & 0.267 & -0.067 & 0.067 \\ 0.067 & -0.067 & 0.267 & -0.267 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 6 & -3 \\ 6 & -3 \\ 1 & 1 \end{bmatrix}$$

$$\begin{bmatrix} P_1' \\ P_2' \\ P_3' \\ P_4' \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -0.8 \\ 1 & -0.8 \\ 1 & 1 \end{bmatrix}$$

$t = \frac{1}{3}$

$F_1(t) = 2t^3 - 3t^2 + 1 = 2\left(\frac{1}{3}\right)^3 - 3\left(\frac{1}{3}\right)^2 + 1$

$= \frac{20}{27}$

$F_2(t) = -2t^3 + 3t^2 = -2\left(\frac{1}{3}\right)^3 + 3\left(\frac{1}{3}\right)^2$

$= \frac{7}{27}$

$F_3(t) = t^3 - 2t^2 + t = \left(\frac{1}{3}\right)^3 - 2\left(\frac{1}{3}\right)^2 + \left(\frac{1}{3}\right)$

$= \frac{4}{27}$

$F_4(t) = t^3 - t^2 = \left(\frac{1}{3}\right)^3 - \left(\frac{1}{3}\right)^2$

$= -\frac{2}{27}$

At $t = \frac{2}{3}$

$F_1(t) = \frac{7}{27}$ $\quad F_2(t) = \frac{20}{27}$ $\quad F_3(t) = \frac{2}{27}$ $\quad F_4(t) = -\frac{4}{27}$

The point on the first spline segment at $t = \frac{1}{3}$ is:

$P(t) = [F][G]$

$= \begin{bmatrix} 20/27 & 7/27 & 4/27 & -2/27 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 1 & 1 \\ 1 & -0.8 \end{bmatrix}$

$= \begin{bmatrix} \frac{1}{3} & \frac{63}{135} \end{bmatrix}$

$= \begin{bmatrix} 0.333 & 0.467 \end{bmatrix}$

The point on the first spline segment at $t = \frac{2}{3}$

$$P(t) = [F][G] = \begin{bmatrix} 7/27 & 20/27 & 2/27 & -4/27 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 1 & 1 \\ 1 & -0.8 \end{bmatrix}$$

$$= \begin{bmatrix} \frac{2}{3} & \frac{126}{135} \end{bmatrix}$$

$$= \begin{bmatrix} 0.667 & 0.933 \end{bmatrix}$$

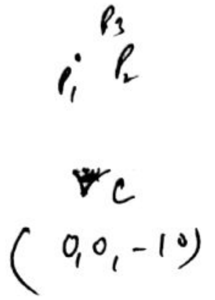
Similarly we can solve for other points also for each segment.

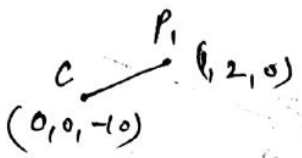
$\therefore$

Segment	t	$P_x(t)$	$P_y(t)$
1	1/3	0.333	0.467
	2/3	0.667	0.933
2	1/3	1.333	0.422
	2/3	1.667	-0.422
3	1/3	2.333	-0.933
	2/3	2.667	-0.467

why are hidden surface algorithms needed? Given points $P_1(1,2,0)$, $P_2(3,6,20)$ and $P_3(2,4,6)$ and a viewpoint $C(0,0,-10)$, determine which points obscure the others when viewed from C.

Sol:→ Visible surface detection algorithms are needed to determine which objects & surface will obscure those objects and surfaces that are in back of them, thus rendering a more realistic image.





Suppose a line b/w C & P_1 then equation of line:

$x = 0 + (1-0)t$ $\boxed{x = t}$

$y = 0 + (2-0)t$ $\boxed{y = 2t}$

$z = -10 + (0-(-10))t$

$\boxed{z = -10 + 10t}$

if $t = 3$, $x = 3, y = 6, z = 20$

i.e P_2 lies on this line (i.e P_2 lies on the projection line through C and P_1).

Next, we determine which point is in front with respect to C.

Now C occurs on the line when $t=0$

P_1 occurs at $t=1$

occurs $= 3$

$\therefore \boxed{P_1 \text{ obscures } P_2}$

hen t = 2

$x = 2, y = 4, z = 10$

but P_3 is $(2,4,6)$ ~~that~~ means P_3 doesn't lie on projecting line

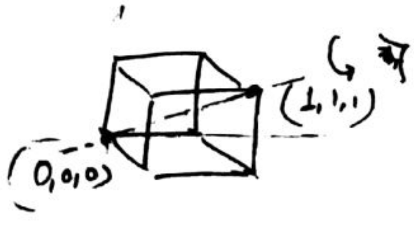
$\therefore$ $\boxed{P_3 \text{ neither obscures nor is obscured by } P_1 \,\&\, P_2.}$

Q 3 Derive transformation matrix for rotation about an arbitrary axis.

Sol: Refer book (Mathematical elements of computer graphics).

Q-1 Given a unit cube with one corner at (0,0,0) and the opposite corner at (1,1,1), derive the transformations necessary to rotate the cube by θ degrees about the main diagonal (from (0,0,0) to (1,1,1)) in the counterclockwise direction when looking along the diagonal toward the origin.

Sol:



→ when we are performing rotation that means rotation will be clockwise

$\therefore$ rotation is done by $(-\theta)$.

To rotate the diagonal to the z-axis, we have to perform rotation about x axis and y-axis

unit cube
coordinates will be ⇒
$$\begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

diagonal passes from $(0,0,0)$ to $(1,1,1)$

$$C_x = \frac{1-0}{\sqrt{3}} = \frac{1}{\sqrt{3}}$$

$$C_y = \frac{1}{\sqrt{3}}$$

$$C_z = \frac{1}{\sqrt{3}}$$

$|V|$ → distance b/w two points

$$|V| = \sqrt{(1-0)^2 + (1-0)^2 + (1-0)^2} = \sqrt{3}$$

$$d = \sqrt{C_y^2 + C_z^2} = \sqrt{\left(\frac{1}{\sqrt{3}}\right)^2 + \left(\frac{1}{\sqrt{3}}\right)^2} = \sqrt{\frac{2}{3}}$$

<u>Sequence of operations will be like this</u>:

$$\boxed{T = R_y^{-1}(\beta) R_x^{-1}(\alpha) R_z(-\theta) R_y(\beta) R_x(\alpha)}$$

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

$$\cos\alpha = C_z/d = \frac{1}{\sqrt{3}} \Big/ \frac{\sqrt{2}}{\sqrt{3}} = \frac{1}{\sqrt{2}}$$

$$\sin\alpha = C_y/d = \frac{1}{\sqrt{3}} \Big/ \frac{\sqrt{2}}{\sqrt{3}} = \frac{1}{\sqrt{2}}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1/\sqrt{2} & -1/\sqrt{2} & 0 \\ 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

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

$$\cos\beta = d$$

$$\sin\beta = C_x$$

Scanned by CamScanner

Inverse using row reduction

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 4 & 1 & 0 \\ 0 & 1 & 4 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$R_3 \Rightarrow R_3 - R_4$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 4 & 1 & 0 \\ 0 & 1 & 4 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$R_2 \Rightarrow R_2 - R_1$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 4 & 1 & 0 \\ 0 & 1 & 4 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$\left\{ \begin{array}{l} R_3 \to 4R_3 \\ R_3 \to R_3 - R_2 \end{array} \right.$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 4 & 1 & 0 \\ 0 & 0 & 15 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 1 & -1 & 4 & -4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$R_3 \to \frac{R_3}{15}$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 4 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ \frac{1}{15} & \frac{-1}{15} & \frac{4}{15} & \frac{-4}{15} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$R_2 \to R_2 - R_3$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{-16}{15} & \frac{16}{15} & \frac{-4}{15} & \frac{4}{15} \\ \frac{1}{15} & \frac{-1}{15} & \frac{4}{15} & \frac{-4}{15} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$R_2 \to \frac{R_2}{4}$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{-4}{15} & \frac{4}{15} & \frac{-1}{15} & \frac{1}{15} \\ \frac{1}{15} & \frac{-1}{15} & \frac{4}{15} & \frac{-4}{15} \\ 0 & 0 & 0 & 1 \end{bmatrix} \longrightarrow \text{this is inverse matrix.}$$

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

$$= \begin{bmatrix} d & 0 & -c_x & 0 \\ 0 & 1 & 0 & 0 \\ +c_x & 0 & d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \sqrt{2/3} & 0 & -\frac{1}{\sqrt{3}} & 0 \\ 0 & 1 & 0 & 0 \\ +\frac{1}{\sqrt{3}} & 0 & \sqrt{2/3} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

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

$$\cancel{R_y(\beta) R_x(\alpha) = \begin{bmatrix} \sqrt{\frac{2}{3}} & 0 & \frac{1}{\sqrt{3}} & 0 \\ 0 & 1 & 0 & 0 \\ \frac{1}{\sqrt{3}} & 0 & \sqrt{\frac{2}{3}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}$$

$\therefore$ Resultant sequence of transformations :

$$P' = T \cdot P$$

T ↓ composition matrix

P ↓ is coordinate matrix

$$\boxed{T = \begin{bmatrix} \sqrt{\frac{2}{3}} & 0 & \frac{1}{\sqrt{3}} & 0 \\ 0 & 1 & 0 & 0 \\ -\frac{1}{3} & 0 & \sqrt{\frac{2}{3}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \sqrt{\frac{2}{3}} & 0 & -\frac{1}{\sqrt{3}} & 0 \\ 0 & 1 & 0 & 0 \\ \frac{1}{\sqrt{3}} & 0 & \sqrt{\frac{2}{3}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}$$

7. Write 3 dimensional homogeneous matrix to rotate by π degrees about the line passing through the point $(0,0,0)$ and $(1,0,1)$ ①

a space without shape or form. eg. skin year.

$$c_x = \frac{1-0}{\sqrt{(1-0)^2+(0-0)^2+(1-0)^2}} = \frac{1}{\sqrt{2}}$$

$c_y = 0$

$c_z = \frac{1}{\sqrt{2}}$

2.)

$$d = \sqrt{c_y^2+c_z^2} = \sqrt{0+\frac{1}{2}} = \frac{1}{\sqrt{2}}$$

$$R_y^{-1}(-\beta)\, R_x^{-1}(\alpha)\, R_z(\pi)\, R_y(-\beta)\, R_x(\alpha)$$

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

$\cos\alpha = c_z/d$

$= \frac{\frac{1}{\sqrt{2}}}{\frac{1}{\sqrt{2}}} = 1$

$\sin\alpha = c_y/d = 0$

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

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

$\cos\beta = d$

$\sin\beta = +c_x$

$$= \begin{bmatrix} d & 0 & -c_x & 0 \\ 0 & 1 & 0 & 0 \\ c_x & 0 & d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} & 0 \\ 0 & 1 & 0 & 0 \\ \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

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

∴ 3 dimensional homogeneous matrix will be

$$R_y^{-1}(\beta)\, R_x^{-1}(\alpha)\, R_z(\pi)\, R_y(\beta)\, R_x(\alpha)$$

$$\begin{bmatrix} \frac{1}{\sqrt2} & 0 & \frac{1}{\sqrt2} & 0 \\ 0 & 1 & 0 & 0 \\ -\frac{1}{\sqrt2} & 0 & \frac{1}{\sqrt2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} -1 & -1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} \frac{1}{\sqrt2} & 0 & -\frac{1}{\sqrt2} & 0 \\ 0 & 1 & 0 & 0 \\ \frac{1}{\sqrt2} & 0 & \frac{1}{\sqrt2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} \frac{1}{\sqrt2} & 0 & \frac{1}{\sqrt2} & 0 \\ 0 & 1 & 0 & 0 \\ -\frac{1}{\sqrt2} & 0 & \frac{1}{\sqrt2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} -1 & -1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} \frac{1}{\sqrt2} & 0 & -\frac{1}{\sqrt2} & 0 \\ 0 & 1 & 0 & 0 \\ \frac{1}{\sqrt2} & 0 & \frac{1}{\sqrt2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} \frac{1}{\sqrt2} & 0 & \frac{1}{\sqrt2} & 0 \\ 0 & 1 & 0 & 0 \\ \frac{1}{\sqrt2} & 0 & \frac{1}{\sqrt2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} -\frac{1}{\sqrt2} & -1 & +\frac{1}{\sqrt2} & 0 \\ \frac{1}{\sqrt2} & -1 & -\frac{1}{\sqrt2} & 0 \\ \frac{1}{\sqrt2} & 0 & \frac{1}{\sqrt2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} \frac{1}{\sqrt2} & 0 & \frac{1}{\sqrt2} & 0 \\ 0 & 1 & 0 & 0 \\ -\frac{1}{\sqrt2} & 0 & \frac{1}{\sqrt2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} -\frac{1}{\sqrt2} & -1 & \frac{1}{\sqrt2} & 0 \\ \frac{1}{\sqrt2} & -1 & -\frac{1}{\sqrt2} & 0 \\ \frac{1}{\sqrt2} & 0 & \frac{1}{\sqrt2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 0 & -\frac{1}{\sqrt2} & 1 & 0 \\ \frac{1}{\sqrt2} & -1 & -\frac{1}{\sqrt2} & 0 \\ 1 & 1/\sqrt2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Note that [T] & [N] are constant for all cubic spline segments. Only the geometry matrix [G] changes from segment to segment.

Now the matrix to calculate internal tangent vectors required in [G], now becomes — eq (15) will become

$$\underbrace{\begin{bmatrix} 1 & 0 & \cdots & & & \\ 1 & 4 & 1 & 0 & \cdots & \\ 0 & 1 & 4 & 1 & 0 & \cdots \\ & & \vdots & & & \\ & & 0 & 1 & 4 & 1 \\ & & & & 0 & 1 \end{bmatrix}}_{M} \underbrace{\begin{bmatrix} P_1' \\ P_2' \\ P_3' \\ \vdots \\ P_n' \end{bmatrix}}_{P} = \underbrace{\begin{bmatrix} P_1' \\ 3\{(P_3-P_2)+P_2-P_1)\} \\ 3\{(P_4-P_3)+(P_3-P_2)\} \\ \vdots \\ 3\{(P_n-P_{n-1})+(P_{n-1}-P_{n-2})\} \end{bmatrix}}_{R}$$

$$[P] = [M]^{-1} \times [R]$$

∴ Chord length approximation provides a smoother, more rounded representation of curves.

Unit-5 (7) Bezier curves → follows curve fitting technique
• approximation curve.
↳ need? Some mathematical description of a space curve is generated ab initio i.e. without prior knowledge of the curve shape or form: eg. skin of car.

Properties

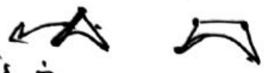
(i) Basis functions are real.

(ii) Degree of polynomial is one less than the no. of points.

i.e 3 point → quadratic curve
4 point → cubic curve

i.e n degree polynomial with (n+1) control point.

(iii) curve generally follows the shape of the defining polygon.

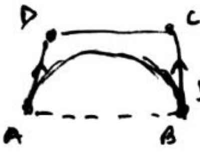


this is defining polygon or (convex hull)

(iv) First and last points on the curve are coincident with the first and last points of the polygon.

(v) Tangent vectors at the ends of the curve have the same directions as the respective spans.

(vi) The curve is contained within the convex hull of the defining polygon.



ABCD is convex hull

when t=0

if (VII) The curve is invariant under an affine transformation.

$J_{n,0}$ (if we perform rotation, translation, scaling etc on curve, it will not change the nature of the curve.

$J_{n,i}$ (i.e if a curve is cubic, then it will remain cubic after transformations.)

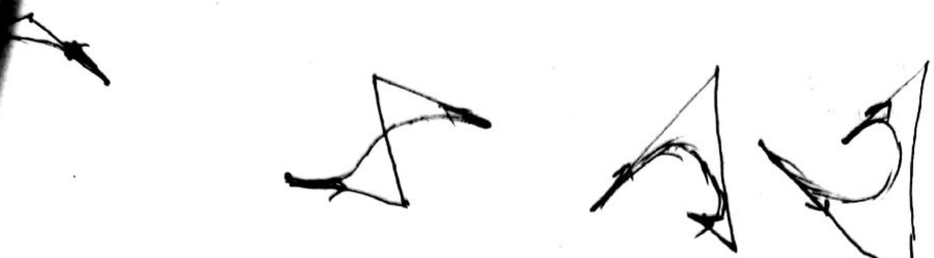
Scanned by CamScanner

NURBS → Superset of curves.

→ a weight is assigned

(2)

examples of cubic polynomials for Bezier



Equation of parametric Bezier curve

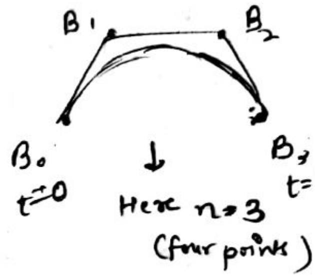
$$\boxed{P(t) = \sum_{i=0}^{n} B_i \, J_{n,i}(t)}$$

$0^0 = 1$, $0! = 1$

$x^0 = 1$

$0 \le t \le 1$

$J_{n,i}(t)$ → ith nth order Bernstein basis or blending function.



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

where , n → degree of the defining Bernstein basis function and thus of the polynomial curve segment.

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

(i.e quadratic curve $n=2$
cubic " $n=3$
i.e n is one less than the no. of points given to draw the curve.)

B_i's are called control points.

$\boxed{\text{when } t=0}$ starting case.

if $i=0$, $t=0$ (initially)

$$J_{n,0}(0) = \frac{n!}{0!\,n!} 0^0 (1-0)^{n-0} = \frac{\cancel{n!}}{1 \times \cancel{n!}} * 1 * 1^n \Rightarrow 1$$

$$J_{n,i}(0) = \frac{n!}{i(n-i)!} 0^i (1-0)^{n-i} \Rightarrow 0$$

Scanned by CamScanner

$\boxed{\text{when } t=1}$ finishing case

$i=n$

$$J_{n,n}(1) = \frac{\cancel{n!}}{\cancel{n!}(n-n)!} 1^n (1-1)^{n-n} = \frac{1}{0!} 1^n \times 0^0 \Rightarrow \frac{1}{1} \times 1^n$$

further, it ... the ...

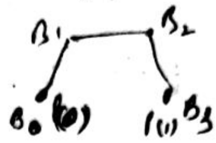
$$\text{if } i \neq n \quad J_{n,i}(1) = \frac{n!}{i!(n-i)!} 1^i (1-1)^{n-i} \Rightarrow \frac{n!}{i!(n-i)!} 1^i \times 0^{n-i} = 0$$

first point

$P(0) = B_0 J_{n,0}(0) = B_0 \times 1 = B_0$

$P(1) = B_n J_{n,n}(1) = B_n \times 1 = B_n$

$\therefore$ we can say that starting & finishing point of curve coincide with the Bezier polynomial



Further, it can be shown that for any given value of the parameter t, the summation of the basis functions is precisely one.

$$\boxed{\sum_{i=0}^{n} J_{n,i}(t) = 1}$$

Also,

$$\boxed{J_{n,i}(t) = (1-t)\, J_{(n-1),i}(t) + t \cdot J_{(n-1),(i-1)}(t)} \quad \boxed{n > i \geq 1}$$

i.e. this eq. is used when we want to derive higher order curve from lower order curves.

$\therefore$ cubic curve can be derived from two quadratic curves

erivation of this

Let $n = 3$

$\sum_{i=0}^{3} J_{3,i}(t)$

$J_{3,0}(t) + J_{3,1}(t) + J_{3,2}(t) + J_{3,3}(t)$

${}^3C_0\, t^0 (1-t)^{3-0} + {}^3C_1\, t^1 (1-t)^{3-1} + {}^3C_2\, t^2 (1-t)^{3-2} + {}^3C_3\, t^3 (1-t)^{3-3}$

$(1-t)^3 + 3t(1-t)^2 + 3t^2(1-t) + t^3$

$$\cancel{t^3} - 3\cancel{t} + 3\cancel{t^2} + 3\cancel{t} + 3\cancel{t^3} - 6\cancel{t^2} + \cancel{3t^2} - 3\cancel{t^3} + \cancel{t^3}$$

$$1$$

<u>proved</u>

∴ Bezier cubic curve can be written as :-
here $n=3$, four points will be given (B_0, B_1, B_2, B_3)

$$P(t) = \sum_{i=0}^{3} B_i J_{3,i}(t)$$

$$= B_0 J_{3,0}(t) + B_1 J_{3,1}(t) + B_2 J_{3,2}(t) + B_3 J_{3,3}(t)$$

$$\boxed{P(t) = B_0 (1-t)^3 + B_1 * 3t(1-t)^2 + B_2 * 3t^2(1-t) + B_3 t^3}$$

<u>In matrix form</u> $(n=3)$

$$= \underbrace{\underbrace{\begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix}}_{T} \cdot \underbrace{\begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}}_{N}}_{F} \underbrace{\begin{bmatrix} B_0 \\ B_1 \\ B_2 \\ B_3 \end{bmatrix}}_{G}$$

not needed to explain

$$F \quad G$$

$$\underbrace{\begin{bmatrix} (1-t)^3 & 3t(1-t)^2 & 3t^2(1-t) & t^3 \end{bmatrix}}_{f\text{ (blending function)}} \begin{bmatrix} B_0 \\ B_1 \\ B_2 \\ B_3 \end{bmatrix}$$

for $n=4$

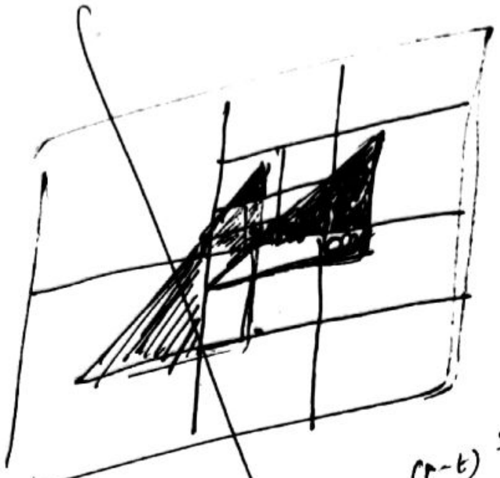
$$P(t) = \begin{bmatrix} t^4 & t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} 1 & -4 & 6 & -4 & 1 \\ -4 & 12 & -12 & 4 & 0 \\ 6 & -12 & 6 & 0 & 0 \\ -4 & 4 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} B_0 \\ B_1 \\ B_2 \\ B_3 \\ B_4 \end{bmatrix}$$

$$F = \begin{bmatrix} J_{n,0}(t) & J_{n,1}(t) & \cdots & J_{n,n}(t) \end{bmatrix}$$

$$N = [\lambda_{ij}]$$

$$\boxed{\lambda_{ij} = \begin{cases} {}^{n}C_{j}\; {}^{n-j}C_{n-i-j}\,(-1)^{n-i-j} & 0 \le (i+j) \le n \\ 0 & \text{otherwise} \end{cases}}$$

Cohen & Reisenfeld gave this generalised representation.



Q.

$B_0 = [1\ 1]$ $B_1 [2\ 3]$
$B_2 [4\ 3]$ $B_3 [3\ 1]$ $t = 0.15$

$(1-t)^3 = (1-0.15)^3 = .614125$

$3t(1-t)^2 = 3 * 0.15(1-0.15)^2$
$= .325125$

$3t^2(1-t) = 3 * 0.15^2(1-0.15)$
$= 0.057375$

$t^3 = (.15)^3 = 0.003375$

$$\begin{bmatrix}(1-t)^3 & 3t(1-t)^2 & 3t^2(1-t) & t^3\end{bmatrix}\begin{bmatrix}1 & 1\\ 2 & 3\\ 4 & 3\\ 3 & 1\end{bmatrix}$$

$$\begin{bmatrix}.614125 & .325125 & .057375 & 0.003375\end{bmatrix}\begin{bmatrix}1 & 1\\ 2 & 3\\ 4 & 3\\ 3 & 1\end{bmatrix}$$

$$= \begin{bmatrix}1.504 & 1.765\end{bmatrix}$$

eg. Bezier curve

B spline

change will be only in some part (i.e in one segment)

B-spline curves → Basis spline → Generalisation of Bezier curve

Drawbacks in Bezier or Bernstein basis

Two characteristics of Bernstein basis, however, limit the flexibility of the resulting curves.

1) no. of specified polygon vertices fixes the order of the resulting polynomial which defines the curve.
The only way to reduce the degree of the curve is to reduce the no. of vertices, and conversely the only way to increase the degree of the curve is to increase the no. of vertices.

2) Global nature of the bernstein basis. Due to global nature, a change in one vertex is felt throughout the entire curve. (change in one control point will change the entire shape of the polygon because each point in the curve is made up of all control points).

Properties of B-spline → order of the B-spline curves is independent of the control points.

→ B spline curve is made up of $(n+1)$ control points and order of the curve K where $2 \le K \le n+1$

* degree is one less than order.

- $K=2$, degree of curve is 1 (linear)
- $K=3$, degree of curve is 2 (quadratic)
- $K=4$, degree of curve is 3 (cubic)

→ The curve has an advantage that it has local propagation
(i.e. the curve is made up of diff. segments, so change in one control point will affect only some portions of the curve)

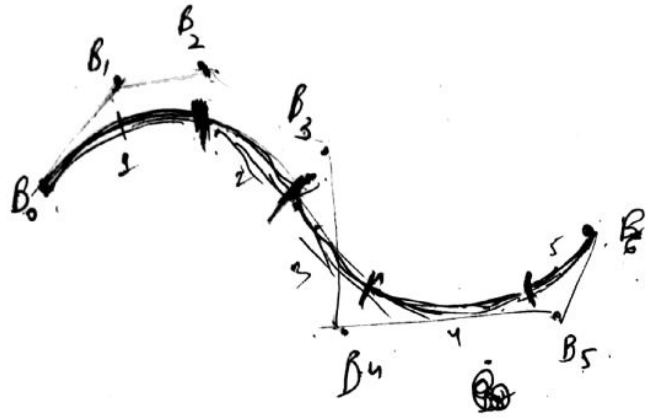
→ The curve can be used to define both open & closed curves (if starting & ending control points are same) has $n-k+2$ segments

$n = 6$ ∴ control points $= n+1$
$= 7$

degree

Let $k = 3$ → order
that is quadratic curve will be drawn

segments $= n-k+2$
$= 6-3+2$
$= 5$



$0 \le t \le$ no. of segments
$0 \le t \le 5$

1st seg. $0 \le t < 1$
$1 \le t < 2$
$2 \le t < 3$
$3 \le t < 4$
$4 \le t < 5$

} values of t.

$k=3$ means one segment $B_0 B_1 B_2$
$B_1 B_2 B_3$
?

→ we can add or subtract control points without changing the degree of the polynomial.

∴ $n = 5$ and again $k = 3$

↓ then it will also produce quadratic curve.

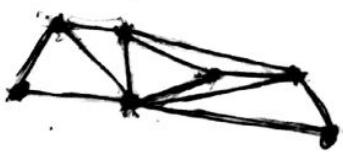
∴ degree of curve doesn't depend on no. of control points.

→ convex hull for B spline curves

$K=2$

$k=3$

$k=4$



eq. of B-spline curve

$$\boxed{P(t) = \sum_{i=0}^{n} B_i N_{i,k}(t)} \quad \text{— (1)}$$

$0 \le t \le n-k+2$

$2 \le k \le n+1$

→ K represents how many points will affect one segment

Suppose $k=3$

if we are changing B_0 or B_1 or B_2 then curve will change only in one segment.

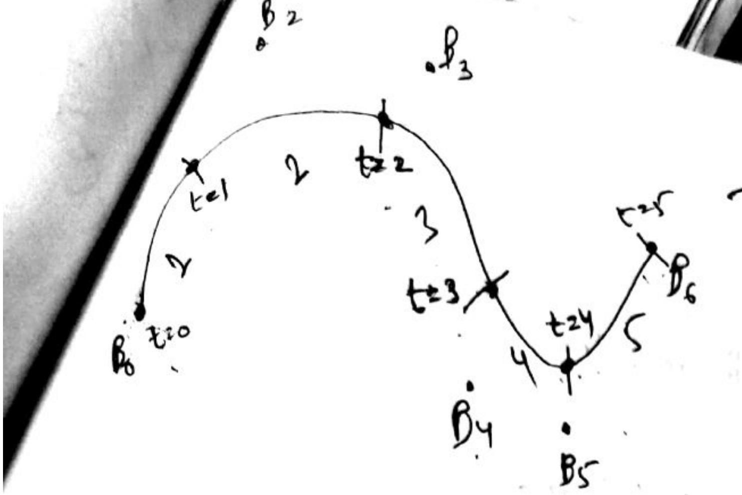
→ B-spline can be represented by
- uniformly
- Non-Uniformly.

→ Most commonly used is Non-Uniform B-spline curve.

→ if we will use B-spline curve then the starting & ending points of curve will not lie on the first and last control point.

→ what is the meaning of Non-Uniform here?

Scanned by CamScanner



∴ for 4th segment

non-uniform knot values → $[0, 0, 0, 1, 2, 3, 4, 5, 5, 5]$

↳ first k and last k knot values are same

$N(t) = B_3 * 0.5(4-t) \ldots + B_5 \ldots$

Here
$n = 6,\ k = 3$

∴ $n+1 = 7$ control points

$k = 3$ means 3 control points will affect the change in a segment

$n - k + 2 = 6 - 3 + 2$
$= 5$ segments

→ control points which are effecting segment 1 → $B_0\ B_1\ B_2$

" 2 → $B_1\ B_2\ B_3$

" 3 → $B_2\ B_3\ B_4$

" 4 → $B_3\ B_4\ B_5$

" 5 → $B_4\ B_5\ B_6$

we know that every segment is affected by 3 (i.e. k) control points but if we are looking inversely

how many segments a control point influences?

this defines the non-uniformity

$B_0 \xrightarrow{\text{influences}}$ only 1st segment

$B_1 \longrightarrow$ 2 segments (1 & 2)

$B_2 \longrightarrow$ 3 segments (1 & 2 & 3)

$B_3 \longrightarrow$ 3 " (2, 3, 4)

$B_4 \longrightarrow$ 3 " (3, 4, 5)

$B_5 \longrightarrow$ 2 segm (4, 5)

$B_6 \longrightarrow$ 1 " (5)

not same for all control points

non uniform B-spline curve

$n=5,\ k=3$

$$\boxed{P(t) = B_0 N_{0,3}(t) + B_1 N_{1,3}(t) + B_2 N_{2,3}(t) + B_3 N_{3,3}(t) + B_4 N_{4,3}(t) + B_5 N_{5,3}(t)} \quad (1)$$

$\therefore \boxed{0 \le t \le 4}$

$0 \le t \le n-k+2$

$5-3+2$
$= 4$

where

this is called cox-deBoor recursion formula

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

this is recursive expression

→ to terminate from this recursive expression we have

$x_i \rightarrow$ are called knot values

$0 \le i \le n+k$

$x_i = 0 \quad$ if $0 \le i < k$

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

$x_i = n-k+2 \quad$ if $n < i \le n+k$

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

$$* \quad \boxed{\sum_{i=0}^{n} N_{i,k}(t) = 1}$$

~~aaetkel~~

B $N_{1,2}(t) = \frac{(t-x_1)\,N_{1,1}(t)}{x_2 - x_1} + \frac{(x_3-t)\,N_{2,1}(t)}{x_3 - x_2}$

$= \frac{t * N_{1,1}(t)}{0} + \frac{(1-t)\,N_{2,1}(t)}{1}$

$= (1-t)\,\underbrace{N_{2,1}(t)}_{=1}$

if $(x_i \le t < x_{i+1})$

Here in 1st seg $0 \le t < 1$

$i = 2$

$x_2 \le t < x_3$

$(0 \le t < 1)$

$\therefore \boxed{N_{0,3}(t) = (1-t)^2}$

$\cancel{N_{2,1}(t) = \frac{(t-x_2)\,N_{2,0}(t)}{x_2 - x_2}}$

$i\ k$

$N_{(i,k)}(t) = x_1 \le t < x_2$

$i\ k$

$N_{(0,3)}(t) = x_0 \le t < x_1$

```
}
printf("Queue Underflow \n");
return;
}
else
{
printf("Element deleted from queue is : %d\n", queue_array[front]);
front = front + 1;
}
} /* End of delete() */
display()
{
int i;
if (front == - 1)
printf("Queue is empty \n");
else
{
printf("Queue is : \n");
for (i = front; i <= rear; i++)
printf("%d ", queue_array[i]);
printf("\n");
```

$n=5$ and $K=3$

$x_i \; (0 \le i \le n+K)$

$(0 \le i \le 8)$

$x_i = \{0,0,0,1,2,3,4,4,4\}$ → first K are same & last K of values are same.

($x_0\; x_1\; x_2\; x_3\; x_4\; x_5\; x_6\; x_7\; x_8$)

Here 0/0 = 0 is accepted

$$N_{0,3}(t) = \frac{(t-x_0)\, N_{0,2}(t)}{x_2 - x_0} + \frac{(x_3-t)\, N_{1,2}(t)}{x_3 - x_1}$$

$$= \frac{t * N_{0,2}(t)}{0} + \frac{(1-t)\, N_{1,2}(t)}{1}$$

$$= 0 + (1-t)\, N_{1,2}(t) \leftarrow (x_1 \le t < x_2,\; 0 \le t < 0)$$

$$N_{0,3}(t) = (1-t)\, N_{1,2}(t) \rightarrow$$

$$= (1-t)(1-t)\, N_{2,1}(t) \Rightarrow (1-t)^2 N_{2,1}(t)$$

From eq (1) $\quad n=5,\; K=3$

n-1 Seg. = $n-k+2$ = $5-3+2$ = 4

for 1st segment $\quad 0 \le t < 1$

$$P(t) = B_0(1-t)^2 + B_1 * 0.5t(4-3t) + B_2 * 0.5t^2$$

for 2nd seg. $\quad 1 \le t < 2$

$$P(t) = B_1 * 0.5(2-t)^2 + B_2 * 0.5t(-2t^2+6t-3) + B_3 * 0.5(t-1)^2$$

for 3rd seg. $\quad 2 \le t < 3$

$$P(t) = B_2 * 0.5(3-t)^2 + B_3 * 0.5t(-2t^2+10t-11) + B_4 * 0.5(t-2)^2$$

- for 4th segment

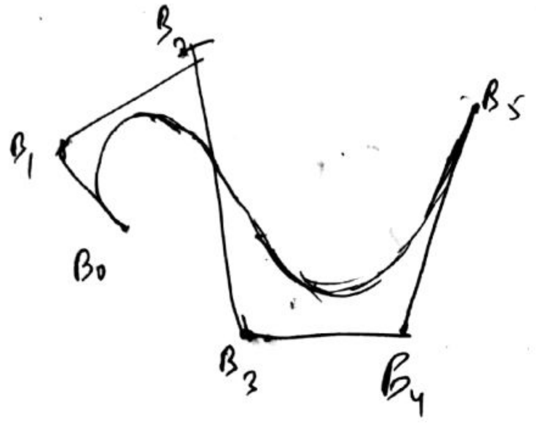
$$P(t) = B_3 * 0.5(4-t)^2 + B_4 * 0.5t(-3t^2+20t-32) + B_5 * 0.5(t-3)^2$$

NURBS → Superset of all the curves.

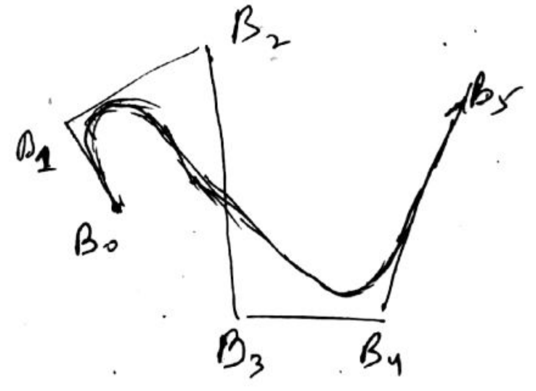
→ a weight is assigned to each control point.

→ All the curves like circles ellipse, hyperbola, parabola can be drawn with the help of NURBS.

→ If all weights are equal to 1, a NURBS curve reduces to a B-spline curve.



$w_i = 1$



$w_0 = w_2 = w_3 = w_5 = 1$

$w_1 = w_4 = 4$