SUB :- Computer Aided Design
and
Manufacturing.

Dept :- Mechanical Engg.

YEAR :- III -

By :- Dr. A. Buckshumiyan.

Mechanical.

Computer Aided Design & Manufacturing.

UNIT- I

Q1 Compare sequential engineering and Concurrent Engineering. May-19, 17.

For launching a new Product into the market in concurrent Eng. the function of design engineering, manufacturing eng and other department are integrated to reduce the elapsed time, to bring a new Product. Conventional eng. the two functions i.e design eng and manufacturing eng tend to be separated and sequential. it's in figure.

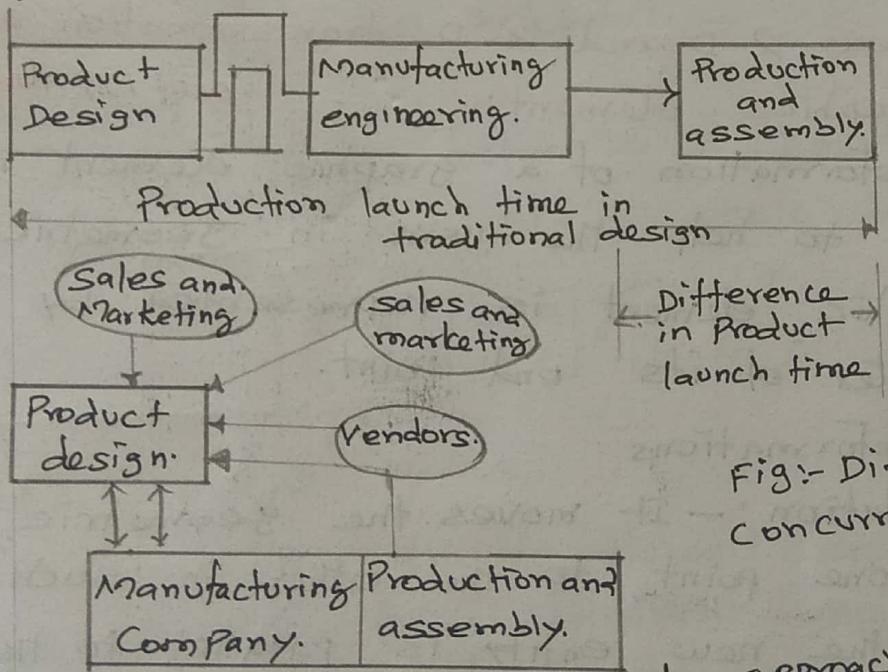


Fig:- Difference between concurrent and sequential Eng.

In this conventional approach there will be little opportunity to the manufacturing engineers to advice on how the design must be altered to make it comfortable in manufacturing.
In this Process, there won't be any interactions between the design ~~the~~ and manufacturing engineers.

In concurrent engineering various department involve in the Product development and give advice on how the product and its components can be designed to facilitate comfort in manufacturing and assembly.

All the dept. such as quality, manufacturing, field services, vendors, Suppliers and customers, involve righ from the initialization of the product. This reduces the time in any change in design, manufacturing and other fields. By doing this the Product development life cycle is reduced.

Q2 Explain how 2-D and 3-D transformation are done on a graphics element: Dec-17, May-18

Transformation of a graphic element is done in order to help the user in Geometric modelling. Any graphic element is represented by the co-ordinates of its end point.

1. 2D Transformations

i) Translation :- It moves the geometric entity from one point to the other in such a way that the new entity is parallel to the old entity.

Consider a point 'P' with coordinates (x, y) translated to a new point 'P'' having coordinates (x', y').

$$x' = x + dx$$

$$y' = y + dy.$$

matrix.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} dx \\ dy \end{bmatrix}$$

$[P'] = [P] + [T]$. $\left(\therefore \text{Translation matrix } [T] = \begin{bmatrix} dx \\ dy \end{bmatrix} \right)$

Ex:- Line with points $P(2,3)$, $Q(5,6)$ translation by 3 units in x direction and 4 units in y direction

$$\begin{bmatrix} 2 & 5 \\ 3 & 6 \end{bmatrix} + \begin{bmatrix} 3 & 3 \\ 4 & 4 \end{bmatrix} = \begin{bmatrix} 5 & 8 \\ 7 & 10 \end{bmatrix}$$

New coordinates are $P(5,7)$ $Q(8,10)$.

ii) Scaling:- Is the transformation used to alter the scale of an entity by enlarging or reducing it with a suitable scaling factor.

$\therefore P' = SP$.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \underbrace{\begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix}}_{\text{scaling transformation matrix}} \begin{bmatrix} x \\ y \end{bmatrix}.$$

S_x, S_y — Scaling factor in x, y - direction.

Ex:- Scaling of a line $P(2,3)$ $Q(5,6)$ by Scaling factor of 2

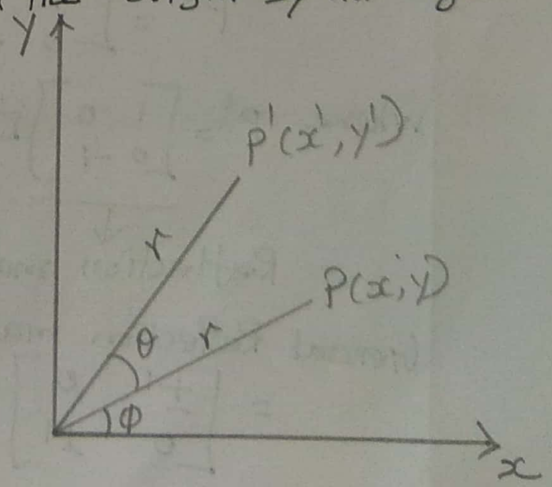
$$\begin{bmatrix} 2 & 5 \\ 3 & 6 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} 4 & 10 \\ 6 & 12 \end{bmatrix}.$$

$P'(4,6)$ and $Q'(10,12)$.

iii) Rotation:- Is a graphic element enables the user to view it from different angles and assists in the creation of circular pattern etc.

- element is rotated about the origin by an angle 'θ'
- Is counter clockwise direction.

$P(x,y)$ — angle ϕ with x-axis. rotated through an angle θ in counter clockwise



$x = r\cos\phi$ $\qquad$ $x' = r\cos(\theta+\phi)$

$y = r\sin\phi$ $\qquad$ $= r\cos\theta\cos\phi - r\sin\theta\sin\phi$

$y' = r\sin(\theta+\phi)$

$= r\sin\theta\cos\phi + r\cos\theta\sin\phi.$

$x' = x\cos\theta - y\sin\theta$

$y' = x\sin\theta + y\cos\theta$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}.$$

Ex :- Line OP will coordinates $O(0,0)$, $P(3,6)$ is rotated through 30° angle in ccw direction.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos 30^\circ & -\sin 30^\circ \\ \sin 30^\circ & \cos 30^\circ \end{bmatrix} \begin{bmatrix} 0 & 3 \\ 0 & 6 \end{bmatrix}.$$

$$= \begin{bmatrix} 0 & -0.4 \\ 0 & 6.69 \end{bmatrix}.$$

New coordinates
$O(0,0)$
$P'(-0.4, 6.69)$.

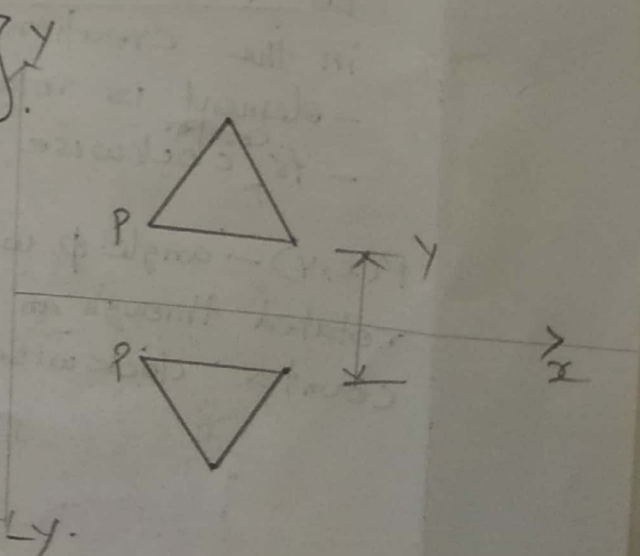
iv) Reflection :- this is displaying the copy of the object while it is reflected about a line or plane.
- used in constructing symmetric models.

To get a reflection of a point 'P' in the graph about x - axis.

$$P' = \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ -y \end{bmatrix}.$$

$$P' = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}.$$

where $P' = \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}}_{\text{Reflection matrix}} [P].$



General Reflection matrix

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

2 3D Transformations.

3D transformations can be obtained in the same way as 2D transformations by adding the Z parameter. the transformation matrix for various transformations are given.

i) Translation

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} dx \\ dy \\ dz \end{bmatrix}$$

where $\begin{bmatrix} dx \\ dy \\ dz \end{bmatrix}$ is translation matrix.

ii) Scaling.

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \underbrace{\begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & S_z \end{bmatrix}}_{\text{is scaling matrix}} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

iii) Rotation.

About Z- axis $R_z = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$

About Y- axis

$$R_y = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix}$$

About X- axis

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

iv) Reflection

General reflection matrix for reflection about the Planes, axes and origin

$$[M] = \begin{bmatrix} m_{11} & 0 & 0 \\ 0 & m_{22} & 0 \\ 0 & 0 & m_{33} \end{bmatrix} = \begin{bmatrix} \pm 1 & 0 & 0 \\ 0 & \pm 1 & 0 \\ 0 & 0 & \pm 1 \end{bmatrix}$$

About x-axis $\quad m_{11} = 1, \; m_{22} = m_{33} = -1$

y-axis $\quad m_{22} = 1, \; m_{11} = m_{33} = -1$

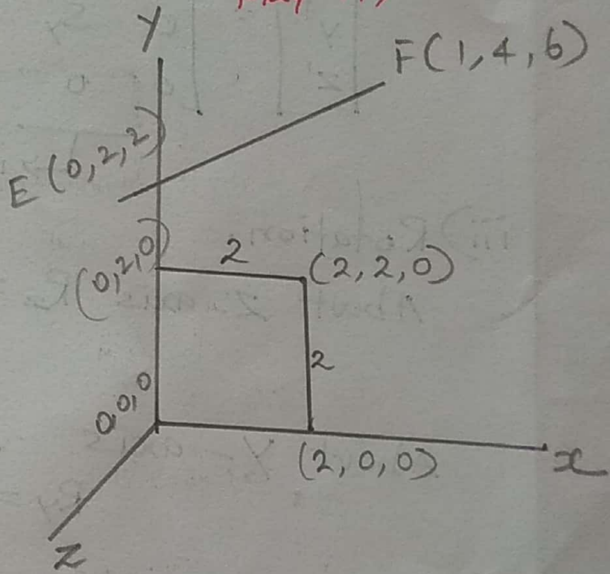
z-axis $\quad m_{33} = 1, \; m_{11} = m_{22} = -1$.

About plane $x = 0, \; m_{11} = -1, \; m_{22} = m_{33} = 1.$

$y = 0, \; m_{22} = -1, \; m_{11} = m_{33} = 1$

$z = 0, \; m_{33} = -1, \; m_{11} = m_{22} = 1$

Q3. Rotate the square shown in fig. 30° counter clockwise about the line EF and find the new coordinates of the rectangle. May-19, Dec-15



Angle of rotation, $\theta = 30°$

coordinate matrix of the square $[P] = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 2 & 0 & 0 & 1 \\ 2 & 2 & 0 & 1 \\ 0 & 2 & 0 & 1 \end{bmatrix}$

Rotation matrix for square about an arbitary line which is not parallel to any of the coordinate axes is obtained by following steps.

1. Transforming rotation axis to any of coordinate axes.
2. Rotating square to desired angle
3. Transforming rotation axis to original position.

Step:-1 Transforming the rotating axis onto the Z-axis

i) Translating rotation axis so that it passes through the origin i.e translating point 'E' to origin

ii) Rotating the rotation axis about x-axis so that it lie in xZ-plane.

iii) Rotating the rotation axis about y-axis so that it will align with Z-axis.

Equation for transformation

$$= [T][R'_x][R_y].$$

$$[T] = \begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix} ; \quad [R'_x] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -\cos\theta_x & \sin\theta_x & 0 \\ 0 & -\sin\theta_x & -\cos\theta_x & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_y = \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Co ordinates of point $E = (x, y, z) = (0, 2, 2)$.

$R = F - E$

$\Rightarrow (1, 4, 6) - (0, 2, 2)$.

$R = (r_1, r_2, r_3) = (1, 2, 4)$.

$$\sin\theta_x = \frac{r_2}{\sqrt{r_2^2 + r_3^2}} = \frac{2}{\sqrt{2^2 + 4^2}} = 0.447.$$

$$\cos\theta_x = \frac{r_3}{\sqrt{r_2^2 + r_3^2}} = \frac{4}{\sqrt{2^2 + 4^2}} = 0.895$$

$$\sin\theta_y = \frac{r_1}{\sqrt{r_1^2+r_2^2+r_3^2}} = \frac{1}{\sqrt{1^2+2^2+4^2}} = 0.218.$$

$$\cos\theta_y = \sqrt{\frac{r_2^2+r_3^2}{\sqrt{r_1^2+r_2^2}^2+r_3^2}} = \sqrt{\frac{2^2+4^2}{1^2+2^2+4^2}} = 0.976.$$

$[T][R_x'][R_y]$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -0.895 & 0.447 & 0 \\ 0 & -0.447 & -0.895 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.976 & 0 & 0.218 & 0 \\ 0 & 1 & 0 & 0 \\ -0.218 & 0 & 0.97 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0.976 & 0 & 0.218 & 0 \\ -0.097 & -0.895 & 0.436 & 2 \\ 0.195 & -0.447 & -0.87 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

step:-2 Rotating the geometry with given angle θ.

$$[P_R] = [T][R_x'][R_y][P][R_z].$$

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

$$[P_R] = \begin{bmatrix} 0.976 & 0 & 0.218 & 0 \\ -0.097 & -0.89 & 0.43 & 2 \\ 0.195 & -0.44 & -0.87 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 1 \\ 2 & 0 & 0 & 1 \\ 2 & 2 & 0 & 1 \\ 0 & 2 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 0.866 & 0.5 & 0 & 0 \\ -0.50 & 0.86 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$[P_R] = \begin{bmatrix} 0.16 & 0.596 & 0 & 1.194 \\ -3.231 & 3.76 & 0 & 1.44 \\ -3.414 & 0.629 & 0 & 0.874 \\ -1.000 & 1.732 & 0 & 1 \end{bmatrix}$$

Step:- 3 Transforming rotating axis to initial Position.

$$[P_R'] = [P_R][R_y'][R_x][T'].$$

$$= [P_R] \times \begin{bmatrix} -\cos\theta_y & 0 & -\sin\theta_y & 0 \\ 0 & 1 & 0 & 0 \\ \sin\theta_y & 0 & -\cos\theta_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x & 0 \\ 0 & \sin\theta_x & \cos\theta_x & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -x \\ 0 & 1 & 0 & -y \\ 0 & 0 & 1 & -z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0.16 & 0.596 & 0 & 1.194 \\ -3.231 & 3.76 & 0 & 1.44 \\ -3.414 & 0.67 & 0 & 0.874 \\ -1.0 & 1.73 & 0 & 1 \end{bmatrix} \begin{bmatrix} -0.976 & 0 & -0.218 & 0 \\ 0 & 1 & 0 & 0 \\ 0.218 & 0 & -0.97 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.895 & -0.447 & 0 \\ 0 & 0.447 & 0.887 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -2 \\ 0 & 0 & 1 & -2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} -0.156 & 0.518 & -0.298 & 0.754 \\ 3.153 & 3.68 & -1.051 & -3.814 \\ 3.332 & 0.896 & 0.385 & -1.688 \\ 0.976 & 1.648 & -0.579 & -1.138 \end{bmatrix}$$

First, second and third.

$A' = (-0.156 \quad 0.518 \quad -0.298)$

$B' = (3.153 \quad 3.680 \quad -1.051)$

$C' = (3.332 \quad 0.896 \quad 0.385)$

$D' = (0.976 \quad 1.648 \quad -0.576)$

Q. Describe and demonstrate DDA line drawing algorithm.

(or)

Explain a line drawing algorithm.

Ans:

One of the methods of line drawing is a Digital Differential Analyzer (DDA)

DDA stands for Digital Differential Analyzer. This is a scan conversion line drawing algorithm in which either the value of Δx or Δy is determined by sampling the line in one co-ordinate (either x (or) y) at unit intervals and then calculating the values in the other co-ordinate (y or x)

Calculating the values of "x" and "y"

1. Lines whose slope is positive and is less than or equal to 1 $(m \leq 1)$

These lines can be sampled at unit 'x' intervals i.e. $\Delta x = 1$ and the corresponding 'y' values can be calculated using.

$$y_{K+1} = y_K + m$$

where "K" takes the integer values starting from 1 and its value is successfully incremented till the last point is reached and the slope "m" can be either "0" or "1".

2. Lines whose slope is positive and is Greater than 1 $(m>1)$

These lines can be sampled at unit "y" intervals i.e, $\Delta y = 1$ and the corresponding "x" values can be determined as,

$$x_{k+1} = x_k + \frac{1}{m}$$

Line processing starts from left to right. If the first end point is at right then,

$$y_{k+1} = y_k - m \text{ and}$$

$$x_{k+1} = x_k - \frac{1}{m}$$

DDA Algorithm

1. Enter the two end points for a line
   i.e, (x_a, y_a) and (x_b, y_b)
2. Horizontal and vertical differences between the end points are assigned to dx and dy.

$$dx = x_b - x_a$$

$$dy = y_b - y_a$$

3. The parameter "steps" is assigned to the larger of "dx" or "dy"

```
if(abs(dx) > abs(dy))
    steps = abs(dx);
else
    steps = abs(dy);
```

4. determine the next pixel position on a line using,

$x_{inc} = dx / steps;$

$y_{inc} = dy / steps;$

5. If magnitude of 'dx' is greater than dy and $x_a < x_b$ then the values of

$x_{inc} = 1$

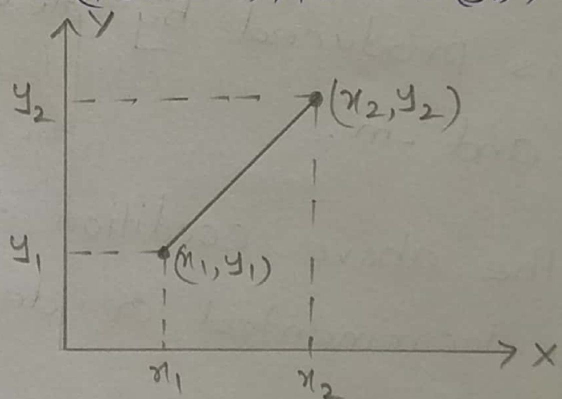
$y_{inc} = m$

6. If the magnitude of 'dx' is greater than 'dy' and $x_a > x_b$ then a new point on a line is produced by decrementing 'x' and 'y' by '-1' and '-m'.

7. If the above condition fails, then 'y' is either incremented or decremented by '1' and 'x' by 1/m

```
#include "device.h"
#define ROUND(r)((int(r+0.5))
void line using DDA (int xa, int ya, int xb, int yb)
    int dx, dy, steps, P;
    float yinc, xinc, x, y;
    dx = xb - xa
    dy = yb - ya;
    x = xa;
    y = ya;
```

```
if(abs(dx) > abs(dy))
    Steps = abs(dx);
else
    steps = abs(dy);
    xinc = dx/(float) steps;
    yinc = dy/(float) steps;
    Setpixel(ROUND(x), ROUND(y));
    for(P=0; P<steps; P=P+1)
        x = x + xinc;
        y = y + yinc;
    Setpixel(ROUND(x), ROUND(y));
```



Advantages.

1. It calculates the pixel positions faster than the calculations performed by using the equation $y = mx + b$

2. Multiplication is eliminated as the x and y increments are used to determine the position of the next pixel on a line.

Disadvantages

1. The rounding and floating point operations are time consuming.
2. The round-off error which results in each successive addition leads to the drift in Pixel Positions, already calculated.

Performance Improvement

The performance of the DDA algorithm can be improved by separating the increments "m" in the integer part and the practical parts, hence the increment "1/m" in the practical parts, hence increment "1/m" in the calculations to integer operations reducing all the

Q5. Define clipping. Also explain the working of a simple line clipping algorithm.

ANS

Clipping: When a picture has been properly scaled and positioned, a method must be considered to remove the lines which are outside the window, such that only the lines within the window are displayed. This process is called clipping.

Cohen - sutherland Line clipping Algorithm: In Cohen - Sutherland line clipping algorithm, each end Point of a line is associated with a four digit Code consisting of 1's and 0's. This Code is Called a region code.

1. This code is used to find the position of a Point with respect to the clipping window.

2. Each bit in the region code corresponds to the four coordinate positions ie, left, right, bottom and top as,

bit 4	bit 3	bit 2	bit 1
Top/above	Bottom/below	Right	Left

3. A Value "1" in the bit Position specifies the presence of a Point in that Position.

4. The region code of a point within the clipping window is 0000.

5. Value of the bits can be calculated by comparing the values (x, y) with the clipping boundaries ie,

   i) Left bit ie, bit 1 is set to 1 if $x < Wx_{min}$

   ii, Right bit ie, bit 2 is set to 1 if $Wx_{max} < x$

   iii, Bottom bit ie, bit 3 is set to 1 if $y < Wy_{min}$

iv, Top bit, ie, bit 4 is set to 1, if $wy_{max} < y$

6. Calculation of the Region codes (Rc) is a two-step procedure consisting of,

i, Finding the difference between the endpoint coordinate values and the boundaries of the clipping window to determine the sign bit as,

a) The left most bit represents the sign bit of $x - wx_{min}$

b) Bit 2 represents the sign bit of $wx_{max} - x$

c) Bit 3 represents the sign bit of $y - wy_{min}$

d) Bit 4 represents the sign bit of $wy_{max} - y$

ii, The sign bit of each difference is used to set the corresponding value in the region code as mentioned above.

After calculating the region codes for all the endpoints, the internal and external lines are determined.

7. If the region codes of both the endpoints are "0000" then that line is completely inside the clipping window and is accepted.

8. If the region codes of the two endpoints of a line contain "1" in the same bit position, then that line is completely outside the clipping window and is not accepted.

9. A logical "AND" operation is performed between the region codes of the two endpoints for a line. If the result is '0000' then the line is completely inside else, it is completely outside the clipping window.

Top left RC = 1001	Top RC = 1000	Top right Rc = 1010
Left RC = 0001	clipping window RC = 0000	Right Rc = 0010
Bottom left RC = 0101	Rc = 0100 Bottom	Bottom right Rc = 0110

Table: Region codes (Rc) for various positions with respect to the clipping window.

For the lines that are not completely inside or completely outside, intersections between the line and the clipping window boundaries are calculated to determine how much portion of the line is accepted or rejected using the formula.

$$\frac{y-y_1}{x-x_1} = \frac{y_2-y_1}{x_2-x_1}$$

$$y-y_1 = \left(\frac{y_2-y_1}{x_2-x_1}\right)(x-x_1)$$

$$y = \left(\frac{y_2-y_1}{x_2-x_1}\right)(x-x_1) + y_1 \quad \left[\because \frac{y_2-y_1}{x_2-x_1} = m\right]$$

$$y = m(x-x_1)+y_1$$

$\therefore$ The y-co-ordinate of the intersection point is

$$y = y_1 + m(x-x_1)$$

where (x_1,y_1) and (x_2,y_2) are the two end points of a line and 'm' is the slope

ie, $m = \frac{y_2-y_1}{x_2-x_1}$ 'x' is either wx_{min} or wx_{max}

The x-coordinate of the intersection point is calculated as

$$y = y_1 + m(x-x_1)$$

$$y-y_1 = m(x-x_1)$$

$$m(x-x_1) = y-y_1$$

$$x-x_1 = \frac{y-y_1}{m}$$

$$\therefore x = x_1 + \frac{y-y_1}{m}$$

Where "y" is either wy_{min} (or) wy_{max}.