2.3.1 Vector Generation/Digital Differential Analyzer (DDA) Algorithm

- The vector generation algorithms which step along the line to determine the pixel which should be turned on are sometimes called **Digital Differential Analyzer (DDA).**
- The slope of a straight line is given as,

$$m = \frac{\Delta y}{\Delta x} = \frac{y_2 - y_1}{x_2 - x_1} \quad \text{... (2.3.1)}$$

The above differential equation can be used to obtain a rasterized straight line. For any given x interval along a line, we can compute the corresponding y interval Δy from equation (2.3.1) as,

$$\Delta y = \frac{y_2 - y_1}{x_2 - x_1} \Delta x \quad \text{... (2.3.2)}$$

Similarly, we can obtain the x interval Δx corresponding to a specified Δy as

$$\Delta x = \frac{x_2 - x_1}{y_2 - y_1} \Delta y \quad \text{... (2.3.3)}$$

- Once the intervals are known the values for next x and next y on the straight line can be obtained as follows,

$$x_{i+1} = x_i + \Delta x$$

$$= x_i + \frac{x_2 - x_1}{y_2 - y_1} \Delta y \quad \text{... (2.3.4)}$$

and

$$y_{i+1} = y_i + \Delta y$$

$$= y_i + \frac{y_2 - y_1}{x_2 - x_1} \Delta x \quad \text{... (2.3.5)}$$

- The equations (2.3.4) and (2.3.5) represent a recursion relation for successive values of x and y along the required line. Such a way of rasterizing a line is called a **digital differential analyzer (DDA).** For simple DDA either Δx or Δy , whichever is larger, is chosen as one raster unit, i.e.

if $|\Delta x| \geq |\Delta y|$ then

$\Delta x = 1$

else

$\Delta y = 1$

With this simplification, if $\Delta x = 1$ then

we have $y_{i+1} = y_i + \frac{y_2 - y_1}{x_2 - x_1}$ and

$x_{i+1} = x_i + 1$

If $\Delta y = 1$ then

we have $y_{i+1} = y_i + 1$ and

$$x_{i+1} = x_i + \frac{x_2 - x_1}{y_2 - y_1}$$

Let us see the vector generation/digital differential analyzer (DDA) routine for rasterizing a line.

Vector Generation/DDA Line Algorithm

13m

1. Read the line end points (x_1, y_1) and (x_2, y_2) such that they are not equal.

   [if equal then plot that point and exit]

2. $\Delta x = |x_2 - x_1|$ and $\Delta y = |y_2 - y_1|$

3. If $(\Delta x \geq \Delta y)$ then

```
length = Δx
    else
        length = Δy
    end if
```

4. $\Delta x = (x_2 - x_1)$ / length

   $\Delta y = (y_2 - y_1)$ / length

[This makes either Δx or Δy equal to 1 because length is either $|x_2 - x_1|$ or $|y_2 - y_1|$. Therefore, the incremental value for either x or y is one.]

5. $x = x_1 + 0.5 * \text{Sign}(\Delta x)$

   $y = y_1 + 0.5 * \text{Sign}(\Delta y)$

   Here, Sign function makes the algorithm work in all quadrant. It returns – 1, 0, 1 depending on whether its argument is < 0, = 0, > 0 respectively. The factor 0.5 makes it possible to round the values in the integer function rather than truncating them.

```
plot (Integer (x), Integer (y))
```

6. i = 1

```
[Begins the loop, in this loop points are plotted]
While ( i ≤ length)
{
    x = x + Δx
    y = y + Δy
    plot (Integer (x), Integer (y))
    i = i + 1
}
```

7. Stop

Let us see few examples to illustrate this algorithm.

18 m.

Example 2.3.1 *Consider the line from (0, 0) to (4, 6). Use the simple DDA algorithm to rasterize this line.*

Solution : Evaluating steps 1 to 5 in the DDA algorithm we have

$x_1 = 0 \quad y_1 = 0 \quad x_2 = 4 \quad y_2 = 6$

$$\therefore \quad \text{Length} = |y_2 - y_1| = 6$$

$$\therefore \quad \Delta x = |x_2 - x_1| / \text{length} = \frac{4}{6}$$

and $$\Delta y = |y_2 - y_1| / \text{length} = 6 / 6 = 1$$

Initial value for

$$x = 0 + 0.5 * \text{Sign}\left(\frac{4}{6}\right) = 0.5$$

$$y = 0 + 0.5 * \text{Sign}(1) = 0.5$$

Tabulating the results of each iteration in the step 6 we get,

i	x	y	Plot
1	0.50	0.50	(0, 0)
2	1.17	1.50	(1, 1)
3	1.83	2.50	(1, 2)
4	2.50	3.50	(2, 3)
5	3.17	4.50	(3, 4)
6	3.83	5.50	(3, 5)
7	4.50	6.50	(4, 6)

Table 2.3.1

The results are plotted as shown in the Fig. 2.3.2. It shows that the rasterized line lies to both sides of the actual line, i.e. the algorithm is **orientation dependent**.

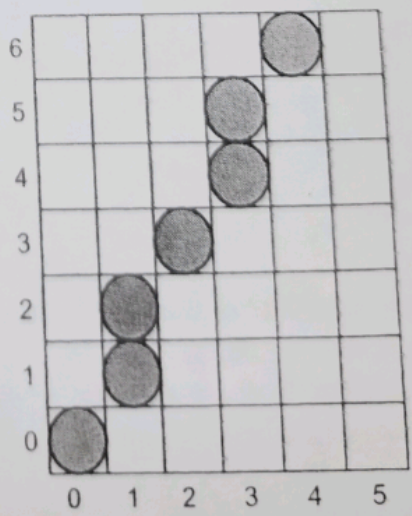

Fig. 2.3.2 Result for a simple DDA