Digital Differential Algorithm(DDA)

- **Scan conversion algorithm based on calculating either Δx or Δy.**
- **Sample the line at unit intervals in one coordinate and determine the corresponding integer values nearest the line path for the other coordinate.**

Case 1

Line with positive slope less than or equal to 1, we sample at unit x intervals (Δx =1) and determine the successive y value as

$$Y_{k+1} = Y_k + m$$

$$\Delta y = m\,\Delta x$$
$$Y_{k+1} - Y_k = m\,\Delta x$$
$$Y_{k+1} - Y_k = m \quad (\Delta x = 1)$$

- k takes integer values starting from 1 and increases by 1 until the final point is reached.
- m can be any number between 0 and 1.
- Calculated y values must be rounded off to the nearest integer.
- For lines with positive slope greater than 1, sample at unit y intervals ($\Delta y = 1$) and calculate each successive x value as
- $$x_{k+1} = x_k + 1/m$$
- The above eqns are based on the assumption that lines are processed from left to right. If the processing is reversed then
-
- $\Delta x = -1$ & $y_{k+1} = y_k - m$ ⟶ Slope >1

  $\Delta y = -1$ & $x_{k+1} = x_k - 1/m$ ⟶ Slope <1

Steps

P1 (xa,ya) and P2 (xb,yb) are the two end points.

2. Horizontal and vertical differences between the endpoint positions are computed & assigned to two parameters namely dx and dy.

3. The difference with greater magnitude determines the 'value' of increments to be done. That means the number of times sampling has to be done. This value is assigned to a parameter called 'steps'.

- 4. Starting from the 1st pixel ,we determine the offset needed at each step to generate the next pixel position along the line path. Call the offset value as $x_{increment}$ and $y_{increment}$.
- The starting points are xa and ya.
- Assign x =xa and y=ya
- $x = x + x_{incr}$ & $y = y + y_{incr}$

- 5. Loop through the process steps times, till the last point xb, yb is reached.

P1 (xa,ya)

P2 (xb,yb)

Steps for DDA Algorithm

Step 1: Accept as input the 2 end point pixel position.

P1(xa,ya) P2 (xb, yb)

Step 2: Horizontal and vertical differences between the endpoint positions are computed & assigned to two parameters namely dx and dy.

dx= xb – xa dy = yb - ya

- Step 3: If abs(dx) >abs(dy)
- steps =abs(dx)
- else steps =abs(dy)
- Step 4: x_{incr} = dx/steps
- (if steps = abs(dx) , xincr = 1)
- y_{incr} =dy/steps
- (ie dy/dx = m)

- Assign x= xa and y=ya
- x= x+ x_{incr} & y= y+ y_{incr}

- Step 5: Loop through the process steps times.

DDA Example

- Suppose we want to draw a line starting at pixel (2,3) and ending at pixel (12,8).
- What are the values of the variables x and y at each timestep?
- What are the pixels colored, according to the DDA algorithm?

numsteps = 12 – 2 = 10
xinc = 10/10 = 1.0
yinc = 5/10 = 0.5

t	x	y	R(x)	R(y)
0	2	3	2	3
1	3	3.5	3	4
2	4	4	4	4
3	5	4.5	5	5
4	6	5	6	5
5	7	5.5	7	6
6	8	6	8	6
7	9	6.5	9	7
8	10	7	10	7
9	11	7.5	11	8
10	12	8	12	8