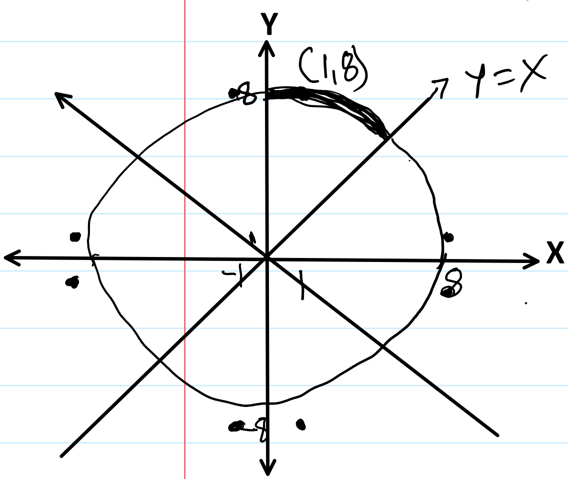
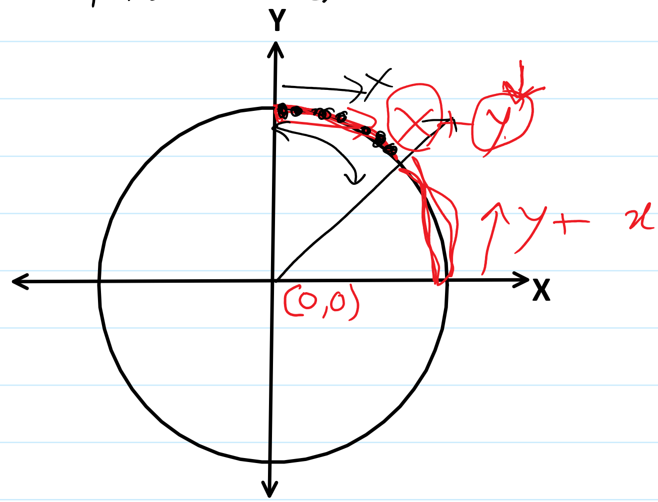
DDA circle drawing algorithm

Thursday, September 9, 2021 10:01 AM

Circle:- Euclidean geometry defines a circle as a set of all points in a plane at a fixed distance (radius) from a given point (centre)





Equation of circle

$$x^2+y^2=r^2$$

$x = x+1$ $\quad r =$ $\quad 0 - r\sqrt{2}$

$y = ?$

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

(2) $x = r\cos\theta$

$y = r\sin\theta \leftarrow$

Digital Differential Analyzer (DDA)

$$x^2+y^2=r^2 \longrightarrow \text{eq}^n \text{ of circle}$$

$$2x\,dx+2y\,dy=0$$

$$x\,dx+y\,dy=0$$

$$y\,dy=-x\,dx$$

$$\frac{dy}{dx}=\frac{-x}{y} \quad \text{——— (1)}$$

increment x value by $\Delta x = \varepsilon y \leftarrow$

& increment in y value $\Delta y = -\varepsilon x \leftarrow$

ε – calculated from the radius of circle

$$2^{n-1} \le r < 2^n$$

$$\varepsilon = 2^{-n}$$

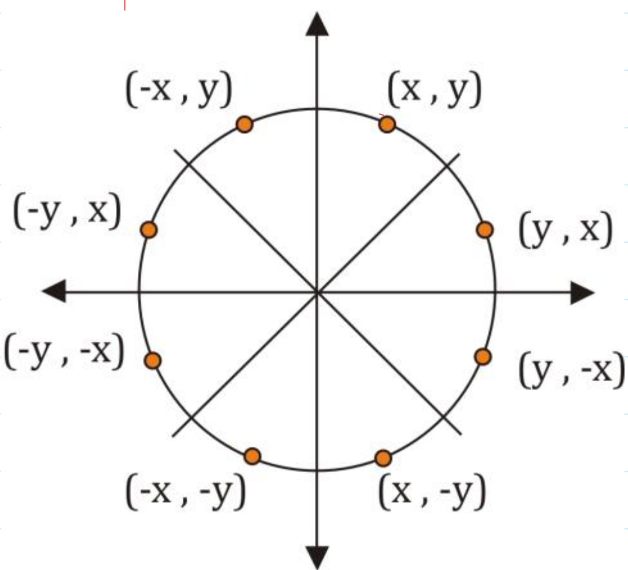
$$\varepsilon = 2^{-n}$$

$r = 50$ $\quad\quad \varepsilon = ?$

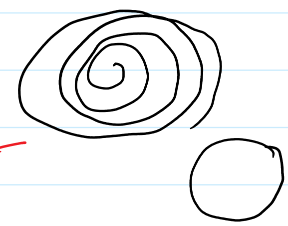
$32 \leq 50 < 64 \quad$ ie $2^5 \leq 50 < 2^6$

$n = 6$

$\therefore \varepsilon = 2^{-6} = 0.0156$



$$\boxed{\begin{aligned} x_{n+1} &= x_n + \varepsilon y_n \\ y_{n+1} &= y_n - \varepsilon x_n \end{aligned}}$$



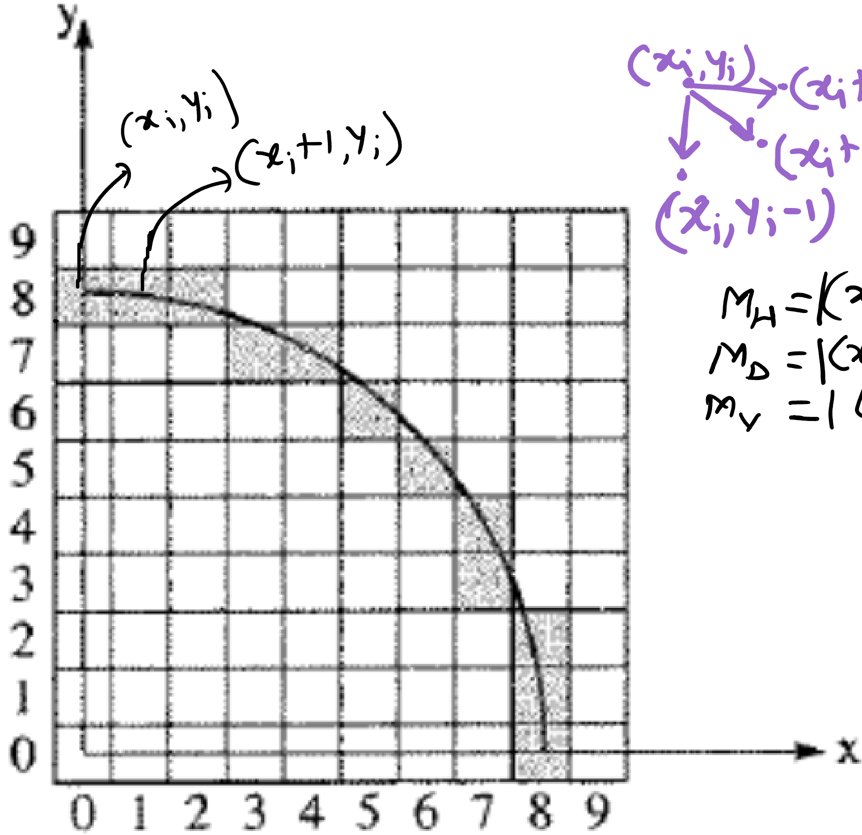
$$\boxed{\begin{aligned} x_{n+1} &= x_n + \varepsilon y_n \\ y_{n+1} &= y_n - \varepsilon x_{n+1} \end{aligned}}$$

DDA algorithm

1) Read the radius (r) of the circle and calculate value of ε
2) $x = 0$
   $y = r$
3) $x_1 = x$
   $y_1 = y$
4) do
   $x_2 = x_1 + \varepsilon y_1$
   $y_2 = y_1 - \varepsilon x_2$
   plot (int(x_2), int(y_2))
   $x_1 = x_2$
   $y_1 = y_2$
   while(($y_1 - y$) < ε or ($x - x_1$) > ε)
5) Finish

Bresenham's Circle drawing algorithm

Thursday, September 9, 2021 12:20 PM



$(x_i, y_i) \to (x_i+1, y_i)$

(x_i+1, y_i-1)

(x_i, y_i-1)

$$M_H = |(x_i+1)^2 + (y_i)^2 - R^2|$$
$$M_D = |(x_i+1)^2 + (y_i-1)^2 - R^2|$$
$$M_V = |(x_i)^2 + (y_i-1)^2 - R^2|$$

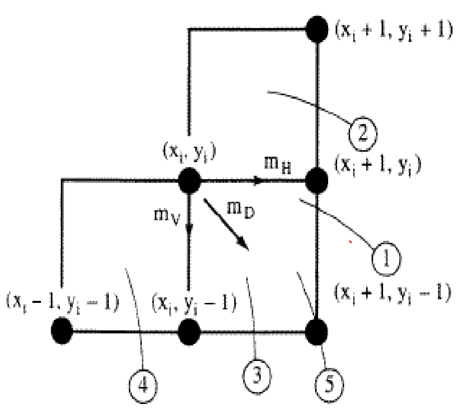


Figure 2–13 Intersection of a circle and the raster grid.

Error

$$\Delta_i = (x_i+1)^2 + (y_i-1)^2 - R^2$$

If $\Delta_i < 0$

m_H m_D

$$\delta = |(x_i+1)^2 + (y_i)^2 - R| - |(x_i+1)^2 + (y_i-1)^2 - R^2|$$

if $\delta \le 0 \leftarrow m_H\ (x_i+1, y_i)$

$\delta > 0 \leftarrow m_D\ (x_i+1, y_i-1)$

$\delta = 0$

Case 1:

$\underline{(x_i+1)^2 + (y_i)^2} - \underline{R}^2 \ge 0$

$(x_i+1)^2 + (y_i-1)^2 - R^2 < 0$

$$\delta = (x_i+1)^2 + (y_i)^2 - R^2 + (x_i+1)^2 + (y_i-1)^2 - R^2$$
$$\delta = 2[(x_i+1)^2 + (y_i-1)^2 - R^2] + 2y_i - 1$$
$$\delta = \underline{2\Delta_i + 2y_i - 1}$$
$$\delta = 2(\Delta_i + y_i) - 1$$

$\delta = \underline{2(\Delta_i + y_i) - 1}$

Case 2:

$$(x_i+1)^2 + (y_i)^2 - R^2 < 0$$

$$(x_i+1)^2 + (y_i-1)^2 - R^2 < 0$$

$\Delta_i > 0$

$$\delta' = |(x_i+1)^2 + (y_i-1)^2 - R^2| - |(x_i)^2 + (y_i-1)^2 - R^2|$$

$\delta' < 0 \leftarrow m_D\,(x_i+1, y_i-1)$

$\delta' > 0 \leftarrow m_V\,(x_i, y_i-1)$

Case 3:

$\delta' = 2(\Delta_i - x_i) - 1$

Summarizing these results yields

$\Delta_i < 0$				
$\delta \le 0$	choose the pixel at (x_i+1, y_i)	$\longrightarrow$	m_H	
$\delta > 0$	choose the pixel at (x_i+1, y_i-1)	$\longrightarrow$	m_D	
$\Delta_i > 0$				
$\delta' \le 0$	choose the pixel at (x_i+1, y_i-1)	$\longrightarrow$	m_D	
$\delta' > 0$	choose the pixel at (x_i, y_i-1)	$\longrightarrow$	m_V	
$\Delta_i = 0$	choose the pixel at (x_i+1, y_i-1)	$\longrightarrow$	m_D	

$$(x_i, y_i) \rightarrow (x_i+1, y_i)$$

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

$$y_{i+1} = y_i$$

$$\Delta_{i+1} = (x_{i+1}+1)^2 + (y_{i+1}-1)^2 - R^2$$

$$= (x_i+1+1)^2 + (y_i-1)^2 - R^2$$

$$= (x_i+1)^2 + (y_i-1)^2 - R^2 + 2x_{i+1} + 1$$

$$\Delta_{i+1} = \Delta_i + 2x_{i+1} + 1$$

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

$$y_{i+1} = y_i - 1$$

$$\Delta_{i+1} = \Delta_i + 2x_{i+1} - 2y_{i+1} + 2$$

$x_{i+1} = x_i \quad , \quad y_{i+1} = y_i - 1$

$\Delta_{i+1} = \Delta_i - 2y_{i+1} + 1$

Bresenham's incremental circle algorithm *for the first quadrant*

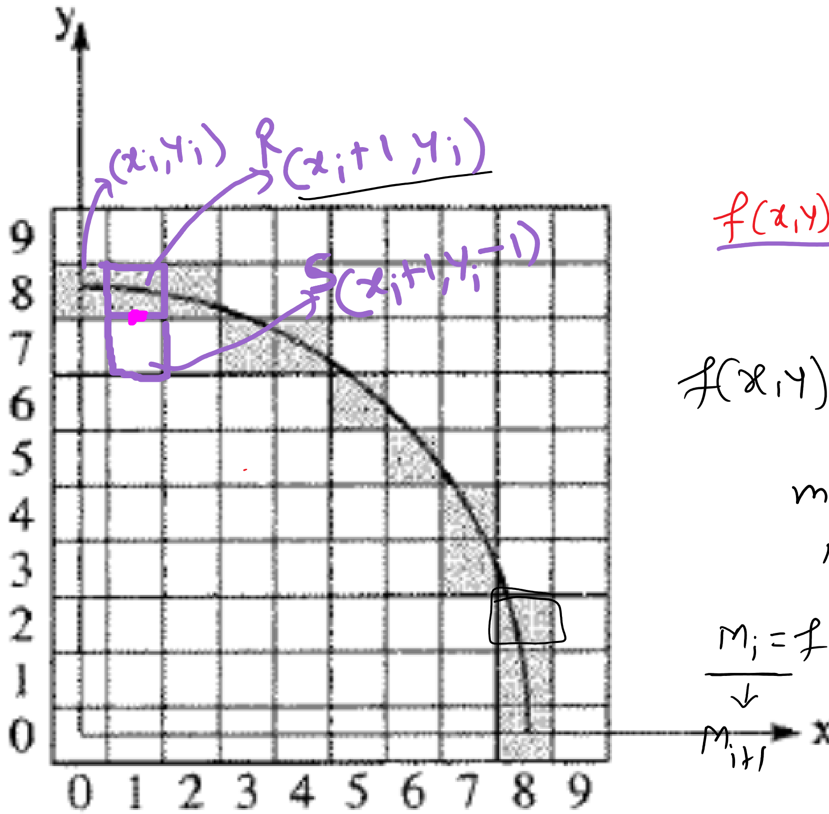
all variables are assumed integer

```
    initialize the variables
    x_i = 0
    y_i = R
    Δ_i = 2(1 − R)
    Limit = 0
    while y_i ≥ Limit
        call setpixel (x_i, y_i)
        determine if case 1 or 2, 4 or 5, or 3
        if Δ_i < 0 then
            δ = 2Δ_i + 2y_i − 1
            determine whether case 1 or 2
            if δ ≤ 0 then
                call mh(x_i, y_i, Δ_i)
            else
                call md(x_i, y_i, Δ_i)
            end if
        else if Δ_i > 0 then
            δ' = 2Δ_i − 2x_i − 1
            determine whether case 4 or 5
            if δ' ≤ 0 then
                call md(x_i, y_i, Δ_i)
            else
                call mv(x_i, y_i, Δ_i)
            end if
        else if Δ_i = 0 then
            call md(x_i, y_i, Δ_i)
        end if
    end while
finish

move horizontally
subroutine mh(x_i, y_i, Δ_i)
    x_i = x_i + 1
    Δ_i = Δ_i + 2x_i + 1
end sub

move diagonally
subroutine md(x_i, y_i, Δ_i)
    x_i = x_i + 1
    y_i = y_i − 1
    Δ_i = Δ_i + 2x_i − 2y_i + 2
end sub

move vertically
subroutine mv(x_i, y_i, Δ_i)
    y_i = y_i − 1
    Δ_i = Δ_i − 2y_i + 1
end sub
```

Midpoint Circle algorithm

Wednesday, September 15, 2021 9:49 AM

midpoint subdivision circle drawing algorithm.



$f(x,y) = x^2 + y^2 = r^2$

$$f(x,y) = x^2 + y^2 - r^2 \begin{cases} < 0 & (x,y) \text{ inside} \\ = 0 & (x,y) \text{ on circle} \\ > 0 & (x,y) \text{ outside} \end{cases}$$

midpoint m_i

$m_i(x_i+1, y_i - \frac{1}{2})$

$m_i = f(x_i+1, y_i - \frac{1}{2}) = (x_i+1)^2 + (y_i - \frac{1}{2})^2 - r^2$

m_{i+1} x_{i+1}

$m_i < 0 \rightarrow R$ y_{i+1}

$m_i = 0 \leftarrow R$

$m_i > 0 \rightarrow S$

m_i

$m_{i+1} = f(x_{i+1}+1, y_{i+1} - \frac{1}{2}) = (x_{i+1}+1)^2 + (y_{i+1} - \frac{1}{2})^2 - r^2$

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

$m_{i+1} = [(x_i + \cancel{1}+1)]^2 + (y_{i+1} - \frac{1}{2})^2 - r^2$

$m_{i+1} = m_i + 2(x_i+1) + (y_{i+1}^2 - y_i^2) - (y_{i+1} - y_i) + 1$

$y_{i+1} =$

$m_i < 0 \qquad y_{i+1} = y_i$

$m_i > 0 \qquad y_{i+1} = y_i - 1$

$$m_{i+1} = \begin{cases} m_i + 2x_i + 3 & \text{when } m_i \le 0 \\ m_i + 2(x_i - y_i) + 5 & \text{when } m_i \ge 0 \end{cases}$$

$(x_0, y_0) = (0, r)$

$$m_0 = (0+1)^2 + (r - 1/2)^2 - r^2$$
$$= 1 + (r - 1/2)^2 - r^2$$
$$\boxed{m_0 = 5/4 - r}$$

Step 1: Input the radius of the circle. Note that radius is the distance from the origin.

Step 2: Initialize the start pixel on the circle

$x = 0$ and $y = r$

Step 3: [Compute the initial decision factor]

$M_i = M = (5/4) - r$

Step 4: [Compute the decision factor and determine the next pixel coordinates on the circle]

```
While (x ≤ y)
    Plot (x, y)
    if (M < 0) then
        M = M + 2x + 3
    Else
        M = M + 2(x − y) + 5
        y = y − 1
    End if
        x = x + 1
End while
```

Step 5: Finish