# Computação Gráfica

## Rasterização de Circunferências

*Professor: Luciano Ferreira Silva, Dr.*

*Aula 7*

# Rasterização de Circunferência



- **Cálculo de pontos ao longo da curva**

  $(x-x_c)^2 + (y-y_c)^2 = r^2$ ou $y = y_c \quad \sqrt{r^2 - (x-xc)^2}$

- **Aritmética de pontos flutuantes;**
  - ✓ Consome bastante tempo;
  - ✓ Provoca falhas na curva;
- **Cálculos na forma paramétrica evita falhas;**

  $x = x_c + r \cdot \cos(t)$

  $y = y_c + r \cdot \text{sen}(t)$

- **Computação ainda em pontos flutuantes;**

# Circunferências: Eq. Paramétrica



- **Algoritmo:**

  $x = x_c + r.\cos(t)$

  $y = y_c + r.\operatorname{sen}(t)$

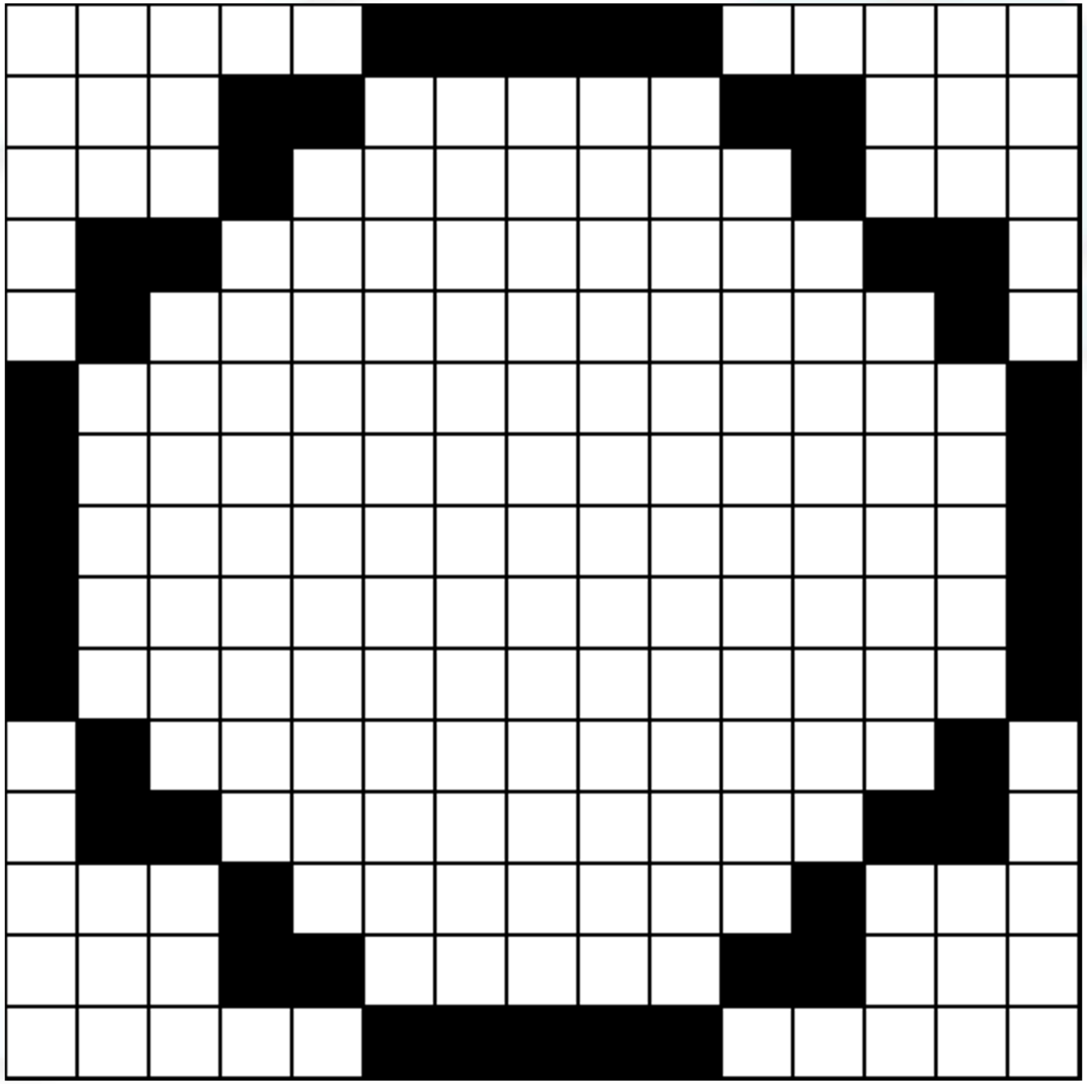
**$x = x_c$ + raio** $\quad$ **$y = y_c$**

**para t de 1 até 360 com passo "t"**

> **pixel (x , y , cor)**
>
> $$x = x_c + r \cdot \cos(\tfrac{\pi \cdot t}{180})$$
>
> $$y = y_c + r \cdot sen(\tfrac{\pi \cdot t}{180})$$

# Circunferências: Eq. Paramétrica

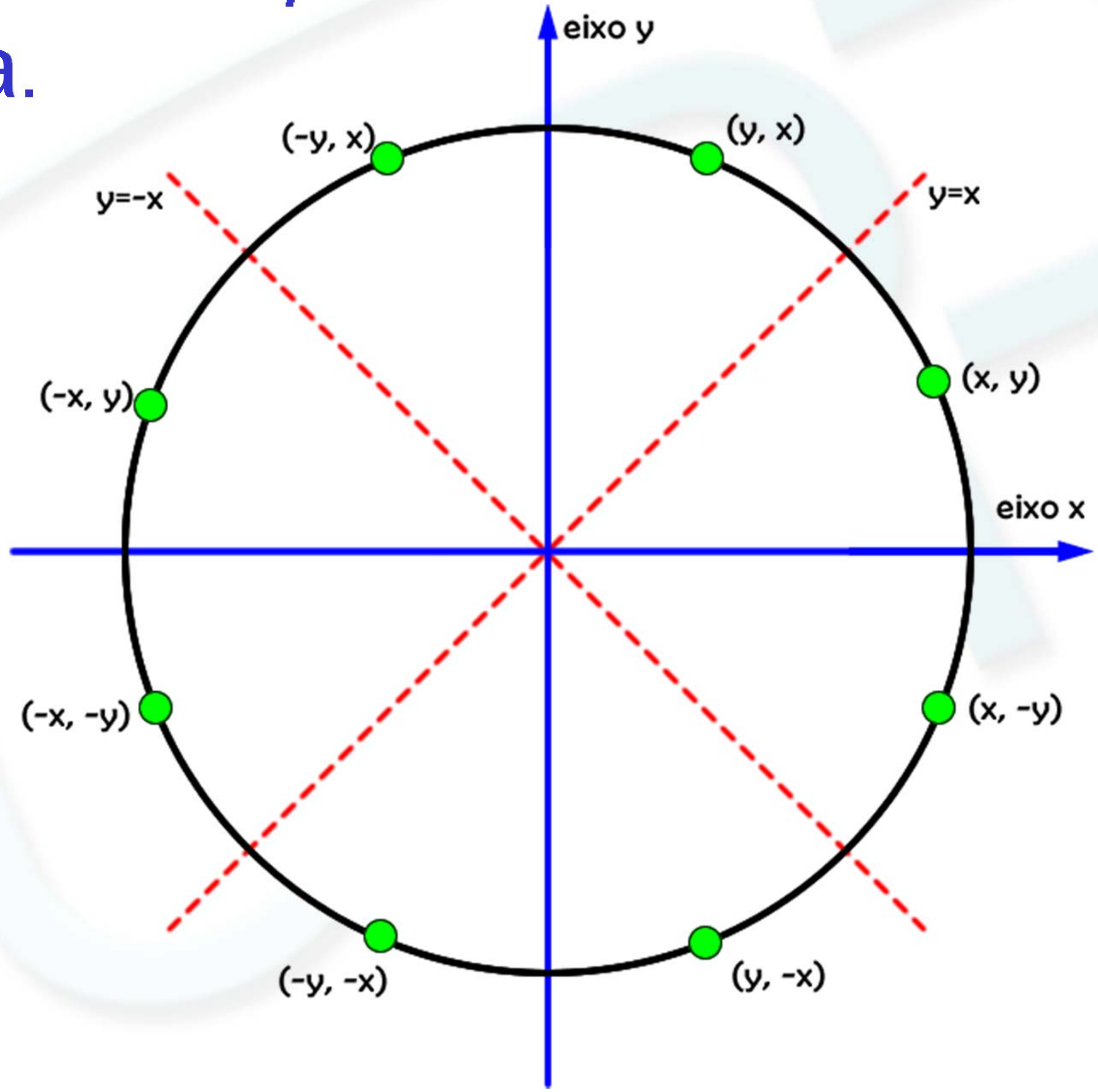
- Exemplo:



# Alg. Incremental com Simetria

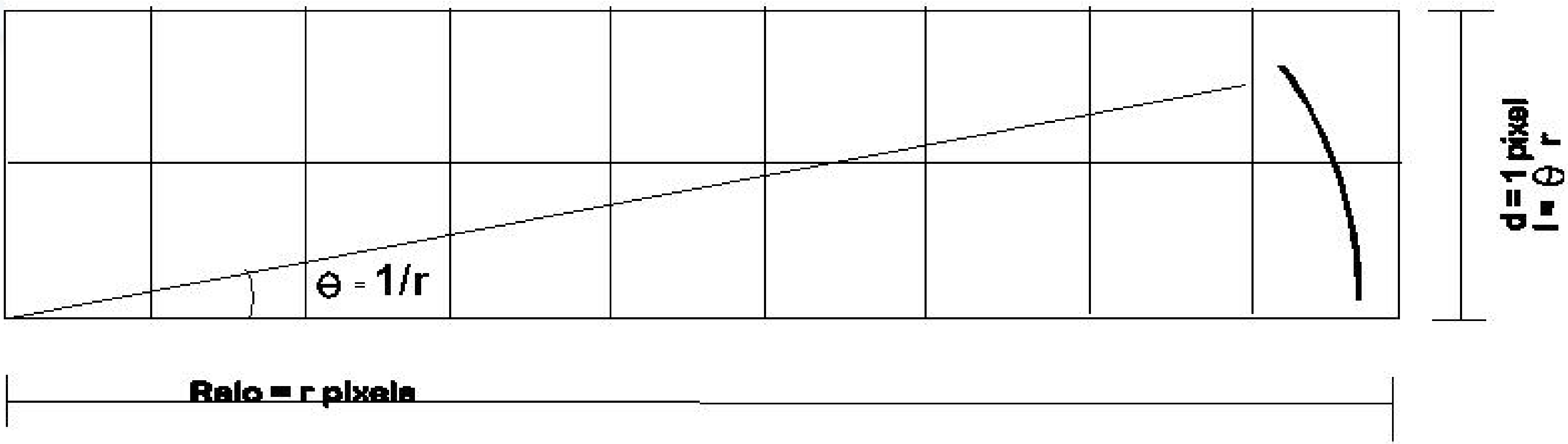


- Propriedade importante: a circunferência é simétrica.



# Alg. Incremental com Simetria

- **Fundamento: deslocamento angular com incremento de uma unidade de pixel;**
  - ✓Deslocamento em radianos = 1/r (r = raio);
  - ✓Calcular as coordenadas de 1/8 da circunferência;
  - ✓Obter os demais pontos por simetria;



# Alg. Incremental com Simetria

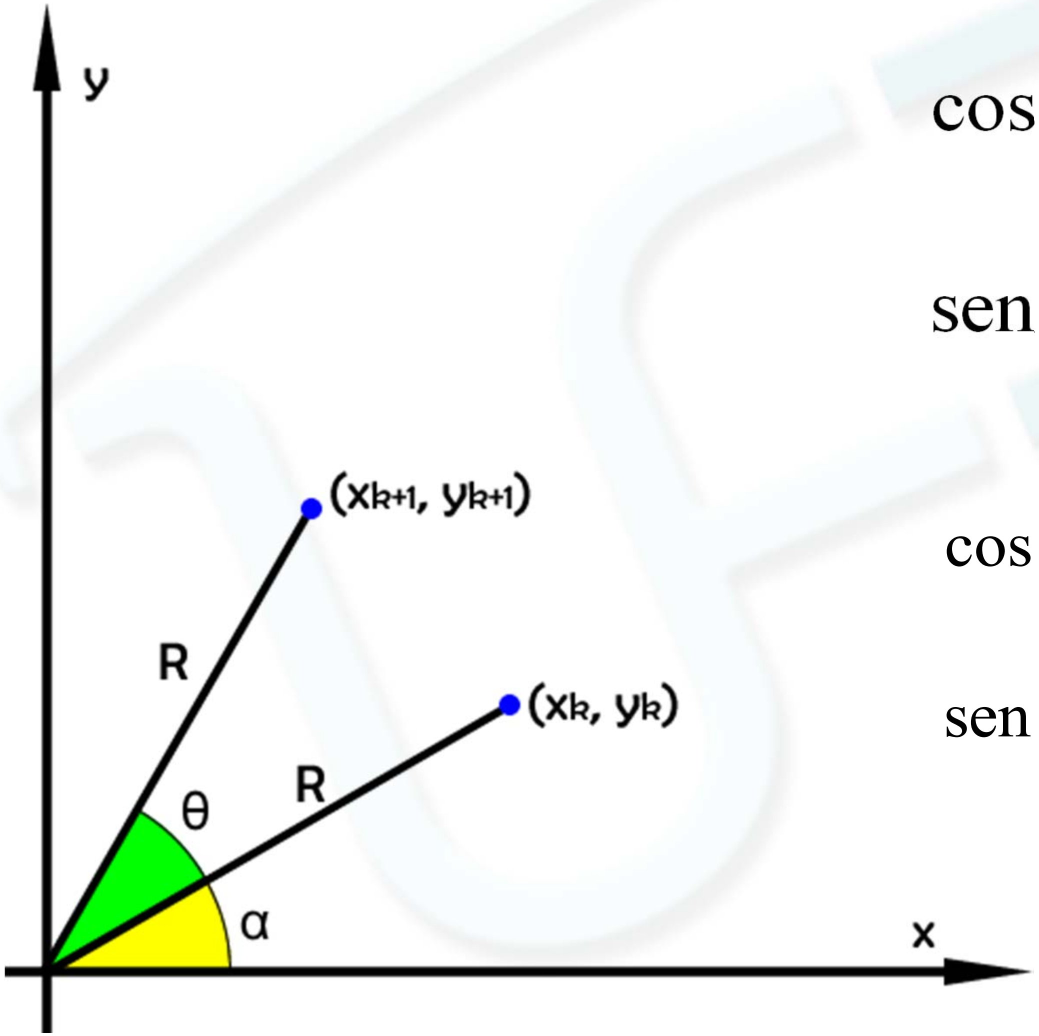
- <u>Solução proposta:</u> algoritmo incremental com deslocamento angular constante e pequeno, com a rotação a partir de um ponto inicial

$$\begin{cases} x_{n+1} = x_n \cos\theta - y_n \sin\theta \\ y_{n+1} = y_n \cos\theta + x_n \sin\theta \end{cases}$$

- Perceba *cos θ* e *sen θ* são valores fixo;
- <u>Problemas:</u>
  - ✓ Erros acumulativos, em função do uso de $x_n$ e $y_n$ nas iterações seguintes;
  - ✓ Uso de números reais - necessidade do arredondamento, para cada pixel;

# Alg. Incremental com Simetria

- **Demonstração:**




$$R = raio;$$

$$\cos\alpha = \frac{x_k}{R} \Rightarrow R.\cos\alpha = x_k\ ;$$

$$\text{sen}\ \alpha = \frac{y_k}{R} \Rightarrow R.\text{sen}\ \alpha = y_k\ ;$$

$$\cos(\theta+\alpha) = \frac{\mathrm{x}_{\mathrm{k+1}}}{\mathrm{R}}$$

$$\text{sen}\ (\theta+\alpha) = \frac{\mathrm{y}_{\mathrm{k+1}}}{\mathrm{R}}$$

# Alg. Incremental com Simetria



- **Temos**

Como: $$\begin{cases} \cos(a+b) = \cos a \cdot \cos b - \operatorname{sen} a \cdot \operatorname{sen} b \\ \operatorname{sen}(a+b) = \operatorname{sen} a \cdot \cos b + \operatorname{sen} b \cdot \cos a \end{cases}$$

Logo:

$$\begin{cases} \dfrac{x_{k+1}}{R} = \cos\theta \cdot \cos\alpha - sen\theta \cdot sen\alpha \Rightarrow x_{k+1} = R.\cos\alpha \cdot \cos\theta - R.sen\alpha \cdot sen\theta \\ \dfrac{y_{k+1}}{R} = sen\theta \cdot \cos\alpha + sen\alpha \cdot \cos\theta \Rightarrow y_{k+1} = R.\cos\alpha \cdot sen\theta + R.sen\alpha \cdot \cos\theta \end{cases}$$

Como: $R.\cos\alpha = x_k$ ; $R.\operatorname{sen}\alpha = y_k$ ;

Daí: $$x_{k+1} = x_k \cdot \cos\theta - y_k \cdot sen\theta$$ $$y_{k+1} = y_k \cdot \cos\theta + x_k \cdot sen\theta$$

# Circunferência Alg. de Bresenham

- **Também utiliza a simetria da circunferência;**
  - ✓ Gera o primeiro quadrante e os demais por simetria;
- **Evita utilizar raízes, potências e funções trigonométricas;**
- **Pode-se começar: no ponto (0, R) → construção horária; ou no ponto (R, 0) → construção anti-horária;**
- **A escolha recai sobre três pixels, na tentativa de selecionar o que está mais próximo da curva ideal;**
- **O critério de seleção entre tais pontos leva em conta a distância relativa entre os mesmos e a circunferência ideal;**

# Circunferência Alg. de Bresenham

- **Utiliza-se a função de circunferência:**

$$f_{circle}(x,y) = x^2 + y^2 - R^2$$

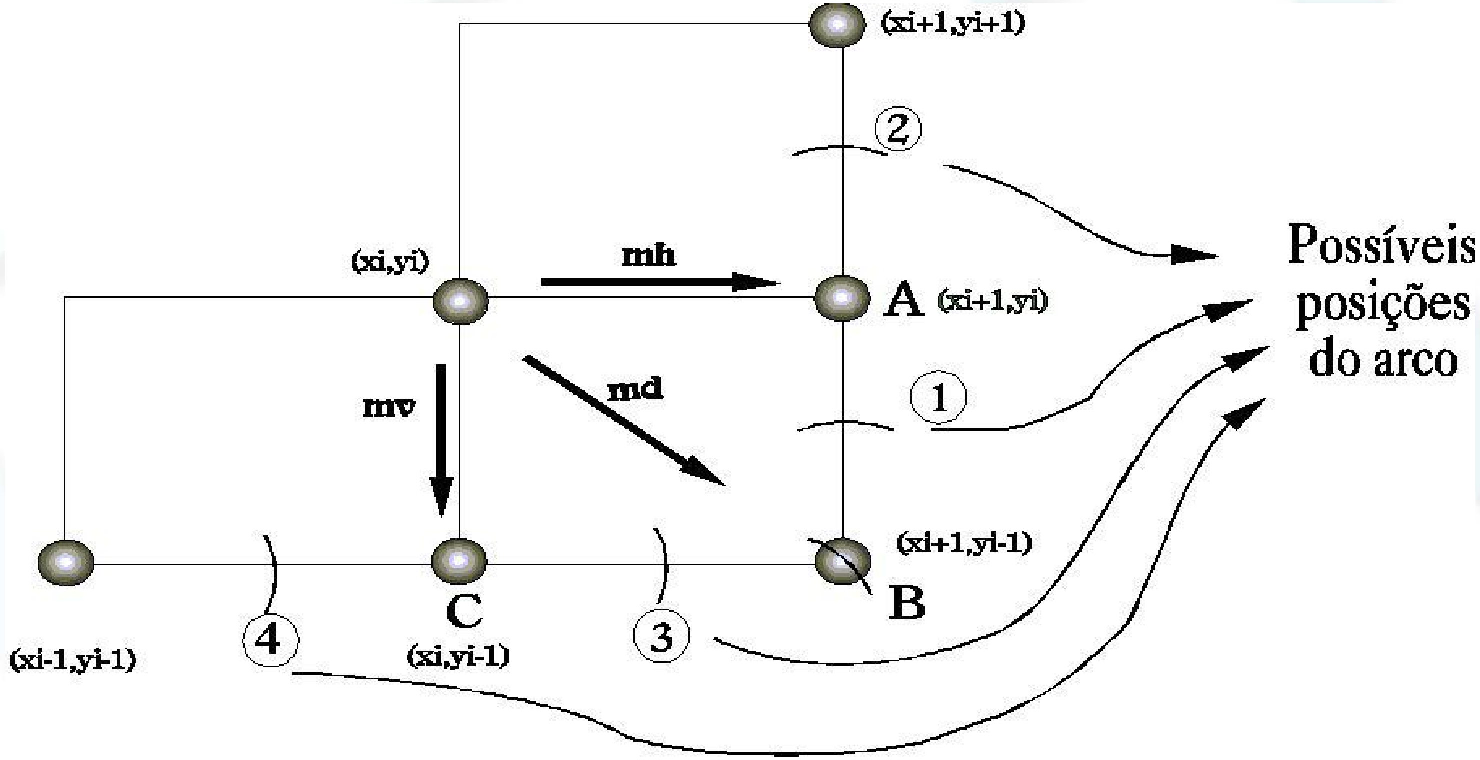
- ***$f_{circle}(x,y) = 0 \rightarrow (x,y)$ está no limite do circunferência.***

- ***$f_{circle}(x,y) < 0 \rightarrow (x,y)$ está no interior da circunferência.***

- ***$f_{circle}(x,y) > 0 \rightarrow (x,y)$ está fora da circunferência.***

# Circunferência Alg. de Bresenham

- Começando no ponto (0, R) → construção horária;




$$\begin{cases} mh = \left| (x_i + 1)^2 + (y_i)^2 - R^2 \right| \\ md = \left| (x_i + 1)^2 + (y_i - 1)^2 - R^2 \right| \\ mv = \left| (x_i)^2 + (y_i - 1)^2 - R^2 \right| \end{cases}$$

# Circunferência Alg. de Bresenham

- **O algoritmo escolhe o pixel que minimize o quadrado da distância entre um destes 3 pixels e o círculo verdadeiro.**

- **O critério de seleção entre os pontos será dado pela distância relativa entre eles:**
  - ✓ Horizontalmente para direita [$mh = |(x_i+1)^2 + (y_i)^2 - R^2|$ ];
  - ✓ Diagonalmente para baixo à direita [$md = |(x_i+1)^2 + (y_i-1)^2 - R^2|$ ];
  - ✓ Verticalmente para baixo [$mv = |(x_i)^2 + (y_i-1)^2 - R^2|$];

# Circunferência Alg. de Bresenham



- **Analisando:**

- **Seja $\Delta i = (x_i+1)^2 + (y_i-1)^2 - R^2$**

**1° caso: Se $\Delta i = 0 \rightarrow$ o ponto B deve ser escolhido.**

**2° caso: Se $\Delta i < 0 \rightarrow$ O ponto B está no interior do circunferência, então deve-se escolher o melhor ponto pelo valor da variável:**

**$\delta = mh - md$;** [Posição (1) ou (2)]

a) **Se $\delta \leq 0$ o ponto escolhido é o A;**

b) **Se $\delta > 0$ o ponto escolhido é o B;**

**3° caso: Se $\Delta i > 0 \rightarrow$ O ponto B está fora da circunferência, então deve-se escolher o melhor ponto pelo valor da variável:**

**$\beta = md - mv$;** [Posição (3) ou (4)]

a) **Se $\beta \leq 0$ o ponto escolhido é o B;**

b) **Se $\beta > 0$ o ponto escolhido é o C;**

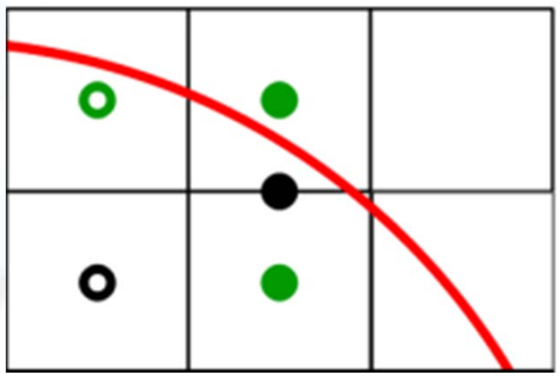
# Circunferência Alg. de Bresenham



- **Outro forma:**
  - ✓Adota-se o parâmetro de decisão

$$f_{circle}(x,y) = x^2 + y^2 - R^2$$

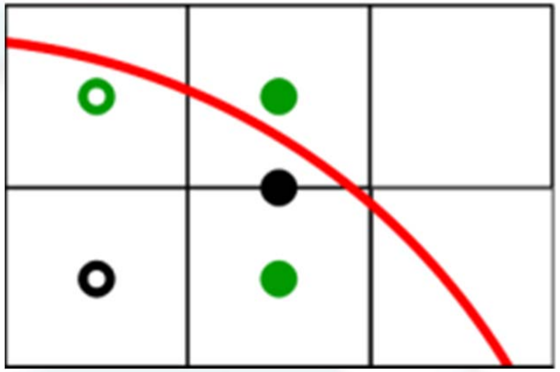
  - ✓Gera-se um octante da circunferência, o restante usa-se simetria;
  - ✓A escolha recai em dois pixels;



# Circunferência Alg. de Bresenham



- **Próximo pixel depende do sinal de:**



$$p_k = f(x_k+1, y_k-1/2) = (x_k + 1)^2 + (y_k - 1/2)^2 - r^2$$

$$p_k = x_k^2 + 2x_k + y_k^2 - y_k + 5/4 - r^2$$

- **Parâmetro de decisão calculado incrementalmente:**

$$p_{k+1} = x_{k+1}^2 + 2x_{k+1} + y_{k+1}^2 - y_{k+1} + 5/4 - r^2$$

- **Considerando** $x_{k+1} = x_k+1$

$$p_{k+1} = (x_k+1)^2 + 2(x_k+1) + y_{k+1}^2 - y_{k+1} + 5/4 - r^2$$

$$p_{k+1} = x_k^2 + 2x_k + 1 + 2x_k + 2 + y_{k+1}^2 - y_{k+1} + 5/4 - r^2$$

$$p_{k+1} = x_k^2 + 4x_k + y_{k+1}^2 - y_{k+1} + 17/4 - r^2$$

# Circunferência Alg. de Bresenham



- **Dois casos possíveis:**

1. **Se $p_k < 0$**

✓ A circunferência passa entre ponto médio e pixel superior

✓ Logo: $y_{k+1} = y_k$

$$p_{k+1} = x_k^2 + 4x_k + y_{k+1}^2 - y_{k+1} + 17/4 - r^2$$

$$p_{k+1} = x_k^2 + 4x_k + y_k^2 - y_k + 17/4 - r^2$$

$$p_{k+1} - p_k = 2x_k + 3$$

$$\boxed{p_{k+1} = p_k + 2x_k + 3}$$

# Circunferência Alg. de Bresenham



## 2. Se $p_k \geq 0$

- ✓ A circunferência passa entre ponto médio e pixel inferior
- ✓ Logo: $y_{k+1} = y_k - 1$

$$p_{k+1} = x_k^2 + 4x_k + y_{k+1}^2 - y_{k+1} + 17/4 - r^2$$

$$p_{k+1} = x_k^2 + 4x_k + (y_k - 1)^2 - (y_k - 1) + 17/4 - r^2$$

$$p_{k+1} - p_k = 2x_k - 2y_k + 5$$

$$\boxed{p_{k+1} = p_k + 2x_k - 2y_k + 5}$$

# Circunferência Alg. de Bresenham

- No ponto incial (0, r)

$$p_k = x_k^2 + 2x_k + y_k^2 - y_k + 5/4 - r^2$$

$$p_0 = x_0^2 + 2x_0 + y_0^2 - y_0 + 5/4 - r^2$$

$$p_0 = 0^2 + 2*0 + r^2 - r + 5/4 - r^2$$

$$\boxed{p_0 = 5/4 - r}$$

- Quando r for inteiro, defina $p_0 = 1 - r$

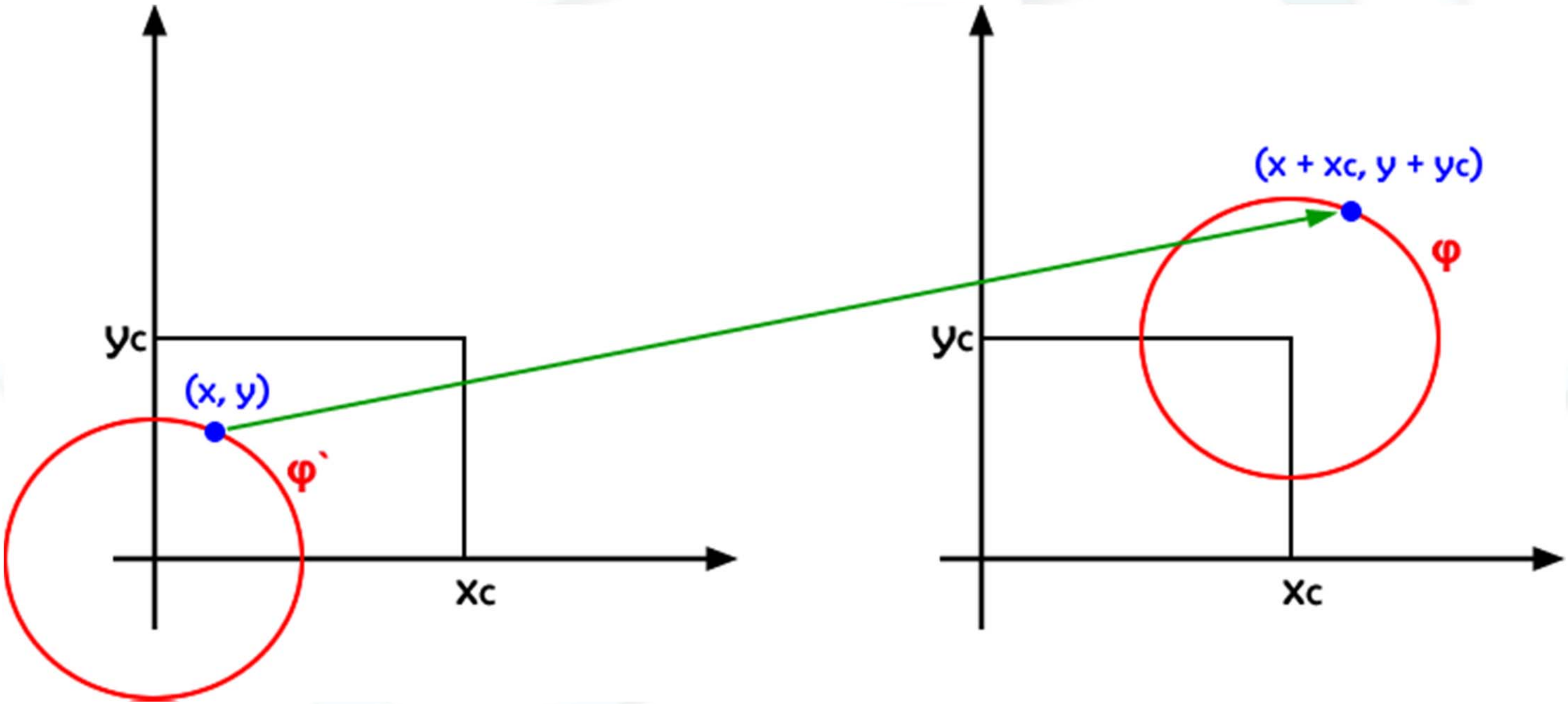
# Circunferência Alg. de Bresenham

UFRR – Departamento de Ciência da Computação
Computação Gráfica – Prof. Dr. Luciano F. Silva

- **Observação: perceba que a circunferência anterior está centrada na origem;**
- **Caso se queira gerar uma circunferência (φ) centra em ($x_c$, $y_c$) e raio R:**
  - ✓ Obtêm-se os pontos (x, y) para a curva (φ\`) centrada na origem e raio R,
  - ✓ A circunferência φ desejada terá pontos no formato ($x + x_c$, $y + y_c$);
  - ✓ Tecnicamente diz-se que φ é uma translação de φ\`.

# Circunferência Alg. de Bresenham





# Circunferência Alg. de Bresenham

$x = 0$ e $y = r$

Parâmetro $= p = 5/4 – r$ ou $1 - r$

Enquanto $x \neq y$ faça:

liga pixel (x, y, cor)

$p \geq 0$

| Sim | Não |
|---|---|
| $y = y-1$<br>$p = p + 2x – 2y + 5$<br>$x = x + 1$ | não altera y<br>$p = p + 2x +3$<br>$x = x + 1$ |

Desenhar os demais octantes

Transladar a circunferência para o centro (xc, yc)

UFRR

# Terceiro Trabalho

- Construir um programa que permita desenhar circunferências pelos algoritmos: Eq. Paramétrica, Incremental com simetria e Bresenham.
- Descreva os resultados de maneira comparativa.