



Computação Gráfica

Rasterização de Circunferências

Professor: Luciano Ferreira Silva, Dr.



Rasterização de Circunferência

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

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

- **Aritmética de pontos flutuantes;**

- ✓ Consome bastante tempo;
- ✓ Provoca falhas na curva;

- **Cálculos na forma paramétrica reduz a falhas;**

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

$$y = y_c + r \cdot \sin(t)$$

- **Computação ainda em pontos flutuantes;**



Circunferências: Eq. Paramétrica

■ Algoritmo:

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

$$y = y_c + r \cdot \sin(t)$$

$$x = x_c + \text{raio} \quad y = y_c$$

para t de 1 até 360 com passo “t”

pixel (x , y , cor)

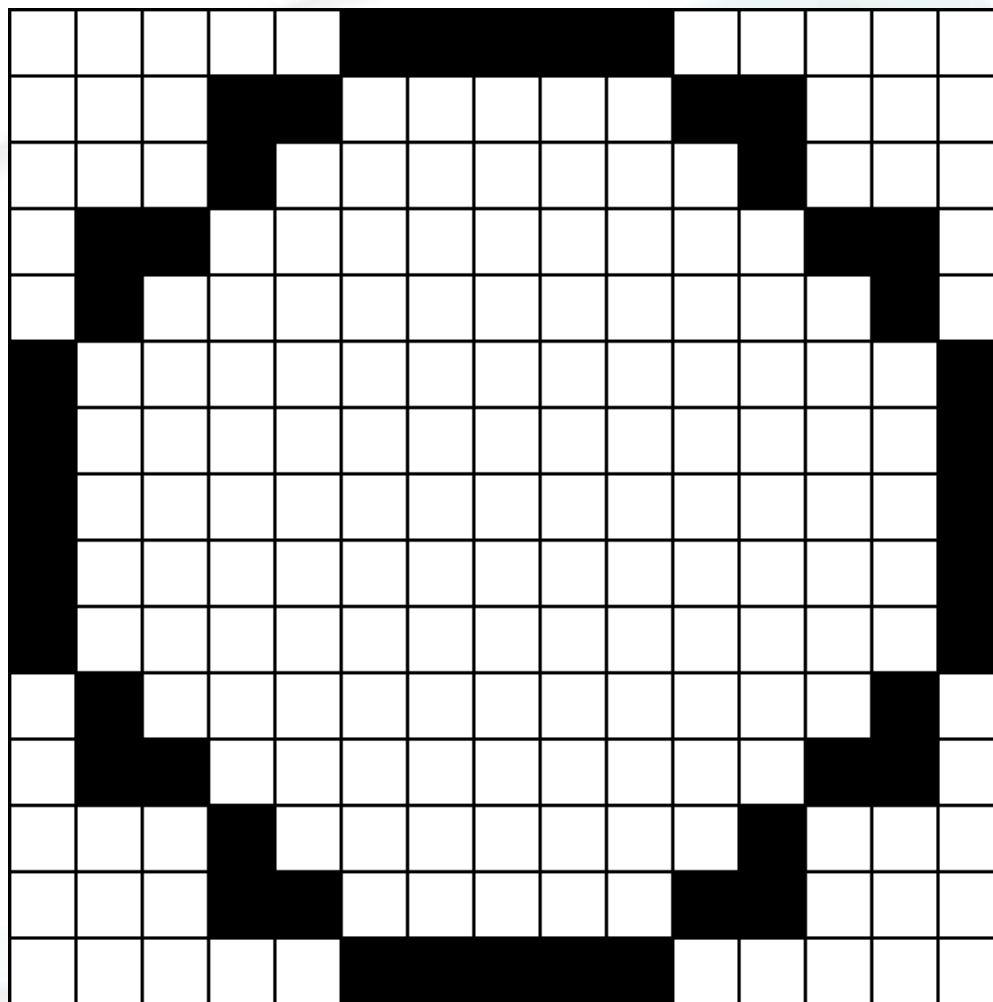
$$x = x_c + r \cdot \cos\left(\frac{\pi \cdot t}{180}\right)$$

$$y = y_c + r \cdot \sin\left(\frac{\pi \cdot t}{180}\right)$$



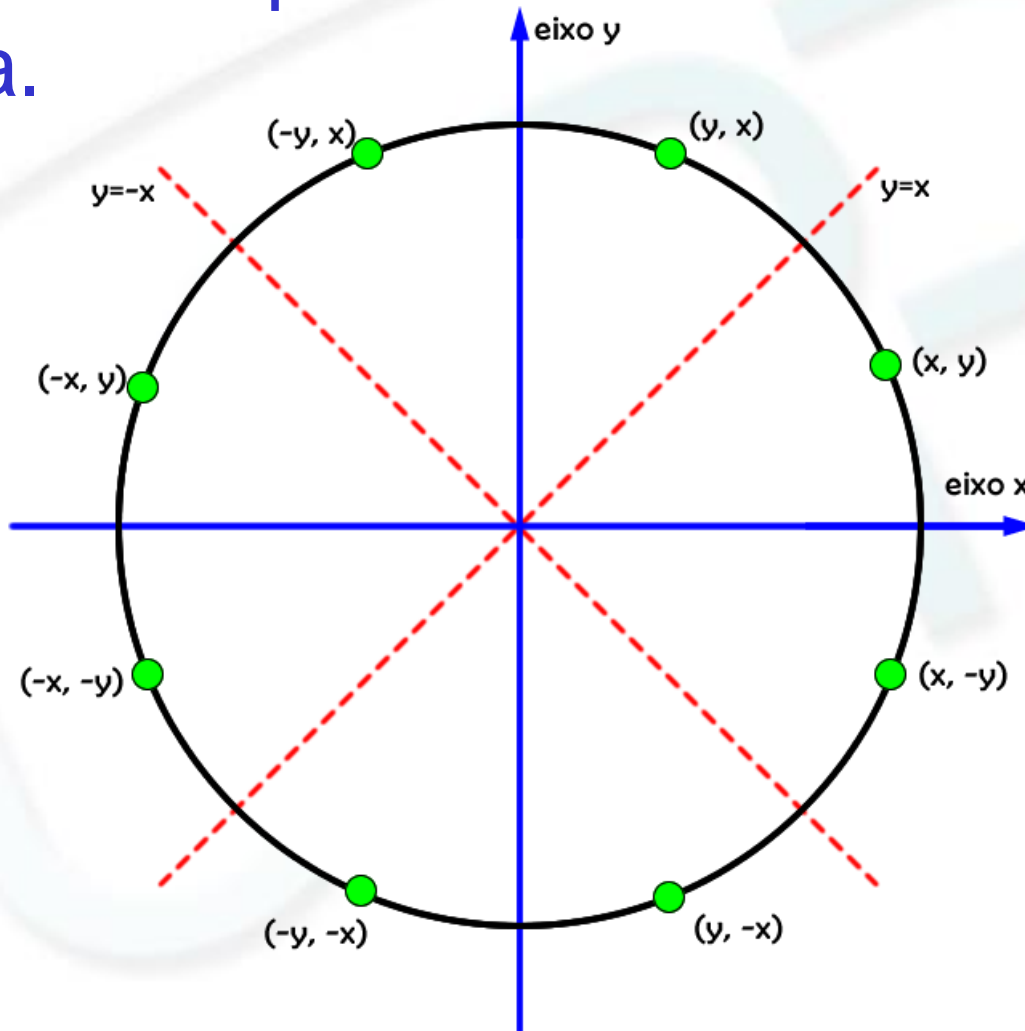
Circunferências: Eq. Paramétrica

■ Exemplo:



Alg. Incremental com Simetria

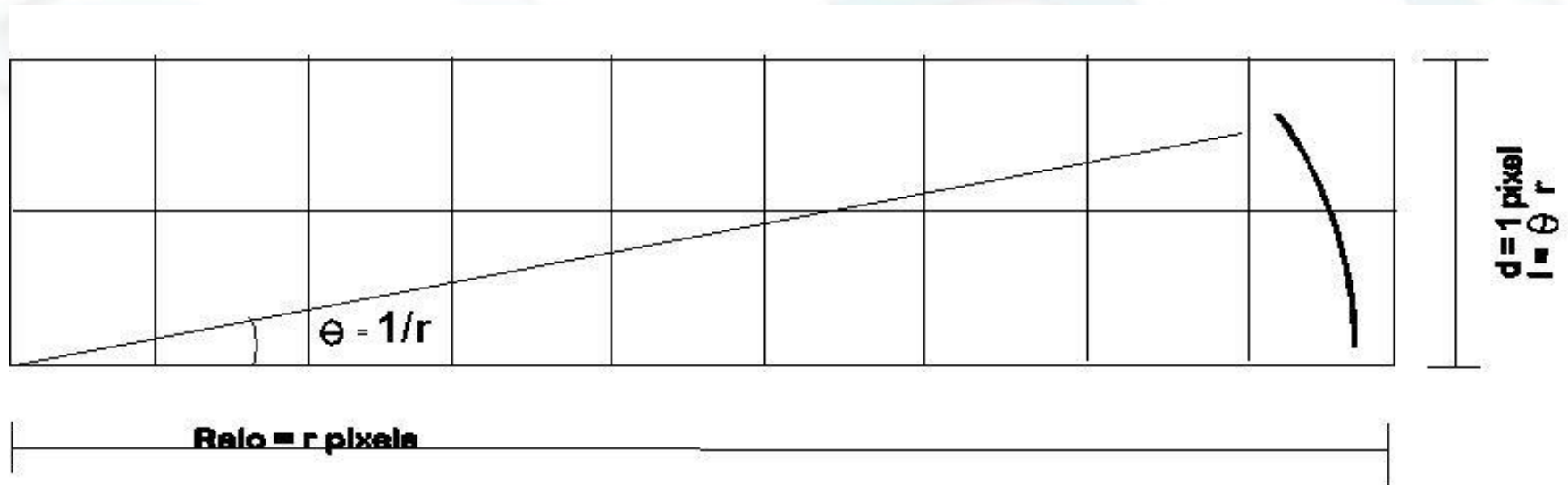
- Propriedade importante: a circunferência é simétrica.





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

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

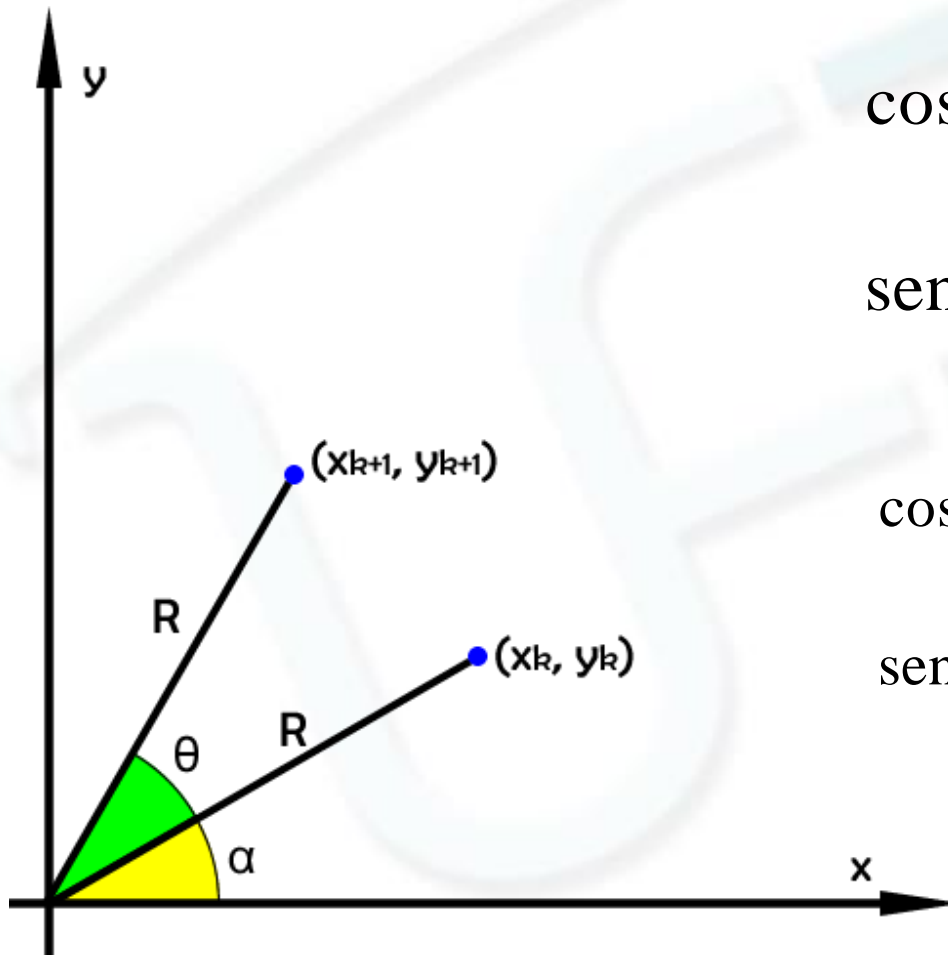
$$\begin{cases} x_{n+1} = x_n \cos \theta - y_n \sin \theta \\ y_{n+1} = y_n \cos \theta + x_n \sin \theta \end{cases}$$

- Perceba $\cos \theta$ e $\sin \theta$ são valores fixo;
- Problemas:
 - ✓ 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:



$R = \text{raio};$

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

$$\sin \alpha = \frac{y_k}{R} \Rightarrow R \cdot \sin \alpha = y_k ;$$

$$\cos (\theta + \alpha) = \frac{x_{k+1}}{R}$$

$$\sin (\theta + \alpha) = \frac{y_{k+1}}{R}$$



Alg. Incremental com Simetria

■ Temos

Como:

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

Logo:

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

Como:

$$\begin{aligned} R \cdot \cos \alpha &= x_k ; \\ R \cdot \text{sen } \alpha &= y_k ; \end{aligned}$$

Daí:

$$\begin{aligned} x_{k+1} &= x_k \cdot \cos \theta - y_k \cdot \text{sen } \theta \\ y_{k+1} &= y_k \cdot \cos \theta + x_k \cdot \text{sen } \theta \end{aligned}$$



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

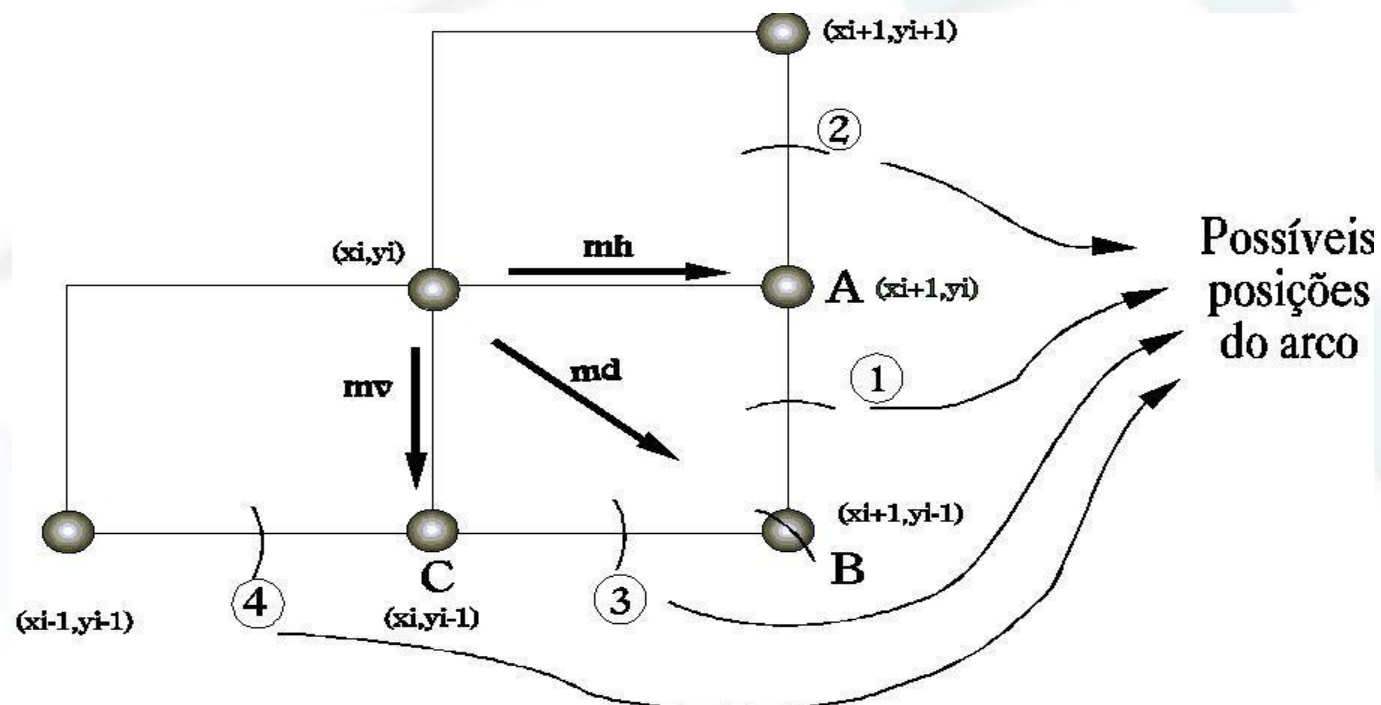
- Utiliza-se a função de circunferência:

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

- $f_{circle}(x,y) = 0 \rightarrow (x,y)$ 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 $(0, R) \rightarrow$ 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 $\Delta i = (x_i+1)^2 + (y_i-1)^2 - R^2$

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

2º caso: Se $\Delta i < 0 \rightarrow$ O ponto B está no interior da 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;

3º 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;



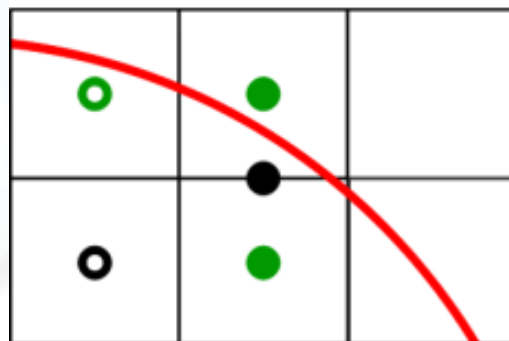
Circunferência Alg. de Bresenham

■ 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;



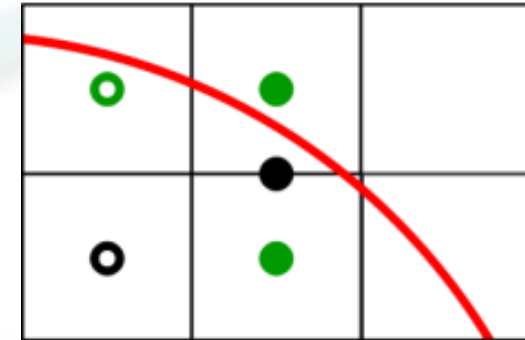


Circunferência Alg. de Bresenham

- 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: $y_{k+1} = y_k$

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

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

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

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



Circunferência Alg. de Bresenham

2. Se $p_k \geq 0$

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

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

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

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

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



Circunferência Alg. de Bresenham

- **No ponto inicial (0, r)**

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

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

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

$$p_0 = 5/4 - r$$

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

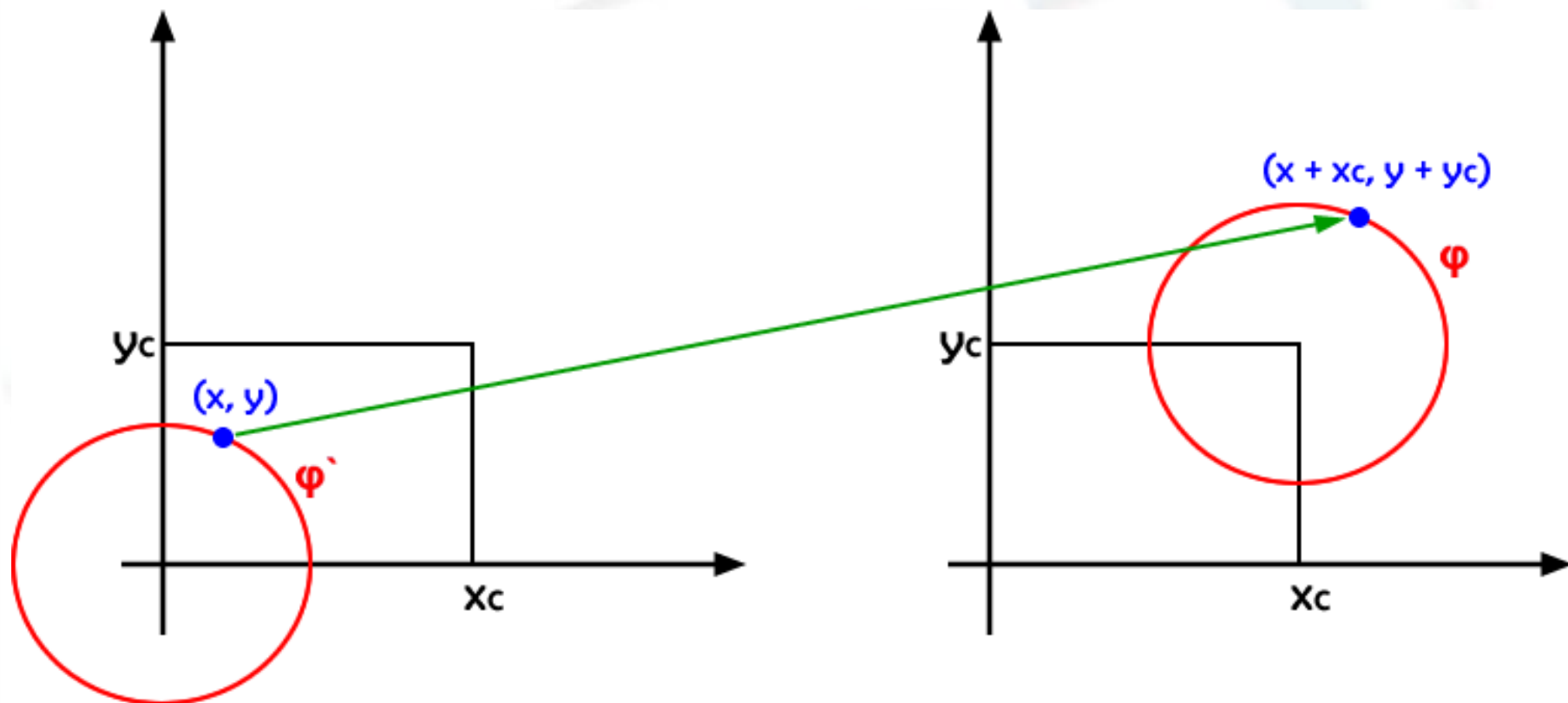


Circunferência Alg. de Bresenham

- **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	
$y = y - 1$ $p = p + 2x - 2y + 5$ $x = x + 1$	não altera y $p = p + 2x + 3$ $x = x + 1$	
Desenhar os demais octantes		
Transladar a circunferência para o centro (xc, yc)		



Segundo Trabalho

- **Neste trabalho você deve (INDIVIDUALMENTE):**
 1. Desenvolver um programa que permita desenhar circunferências por meio dos algoritmos: **Equação paramétrica, Incremental com Simetria e Bresenham.**
 2. Construir um relatório que descreva a construção e os resultados de maneira comparativa.
 3. Apresentar o programa desenvolvido e entregar o relatório digital na sala do professor;
 - Obviamente você será arguido nesse momento.



Segundo Trabalho

■ OBSERVAÇÕES:

- ✓ Objetivo do trabalho é que você veja os algoritmos trabalhando e os compare, desse modo, implemente de forma que isso aconteça;
- ✓ Trabalhos entregues sem defesa não serão aceitos (receberão nota zero (0)).
 - Dessa forma, **NÃO** adianta apenas fazer o trabalho e enviar por e-mail;
 - **Muita atenção:** não sou uma PJ ou PF precisando de software de rasterização, com isso em mente apresente com o intuito de:
 - Provar que você entende os algoritmos;
 - Provar que você realmente é o autor do programa.



Segundo Trabalho

■ OBSERVAÇÕES (Cont...):

- ✓ Trabalhos entregues sem o relatório serão avaliados em cinquenta por cento (50%) da nota;
- ✓ Trabalhos entregues após a data final estabelecida não serão aceitos, a não ser com a apresentação de atestado médico ou declaração de serviço militar;

■ DICAS:

1. Escolha a linguagem de programação que você mais domina e que você entende que **não** será um fator limitante no desenvolvimento do trabalho;
2. Comece a desenvolver o trabalho hoje e não deixe para apresentar no último dia.



Dúvidas

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

