Aula 6

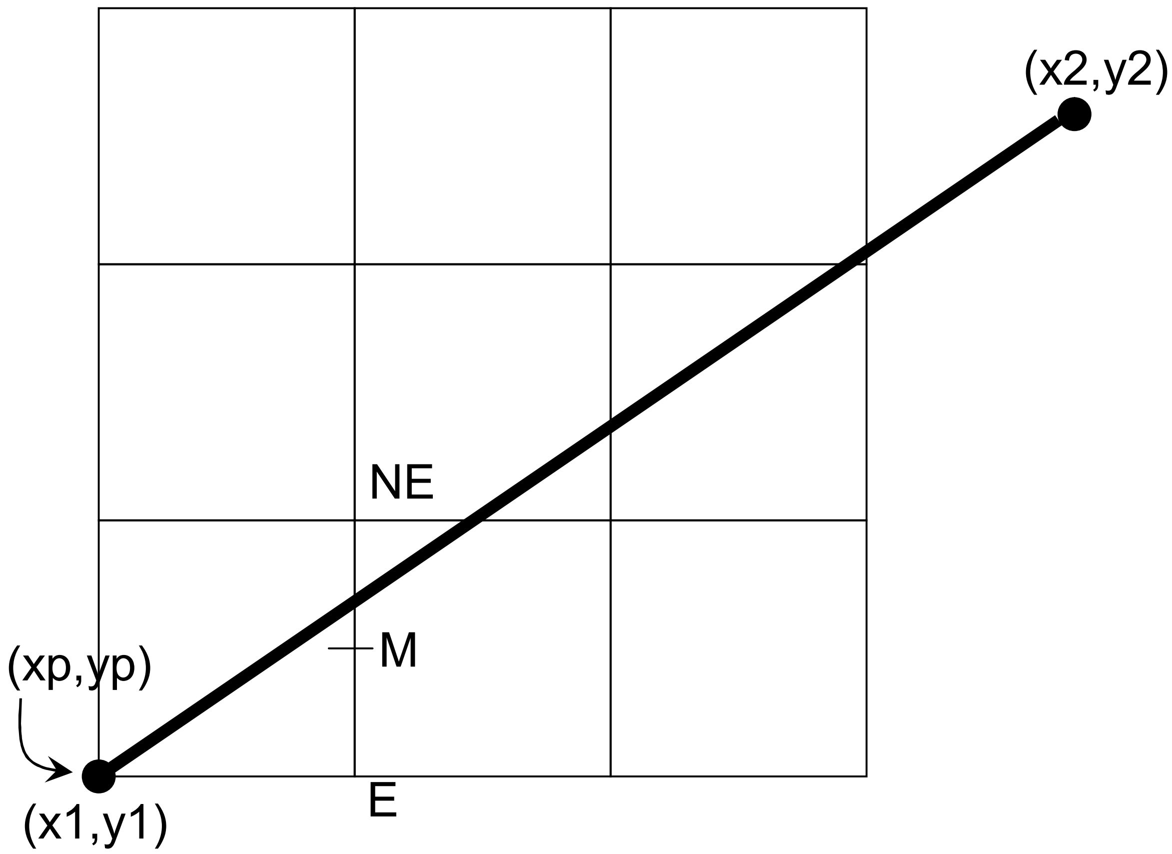
Algoritmo de Bresenham
(ponto médio)
para o traçado de linhas retas

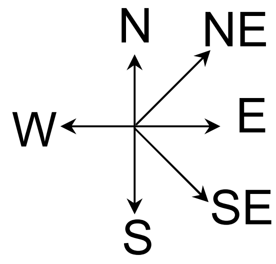
Algoritmo de Bresenham - Traçado de Retas

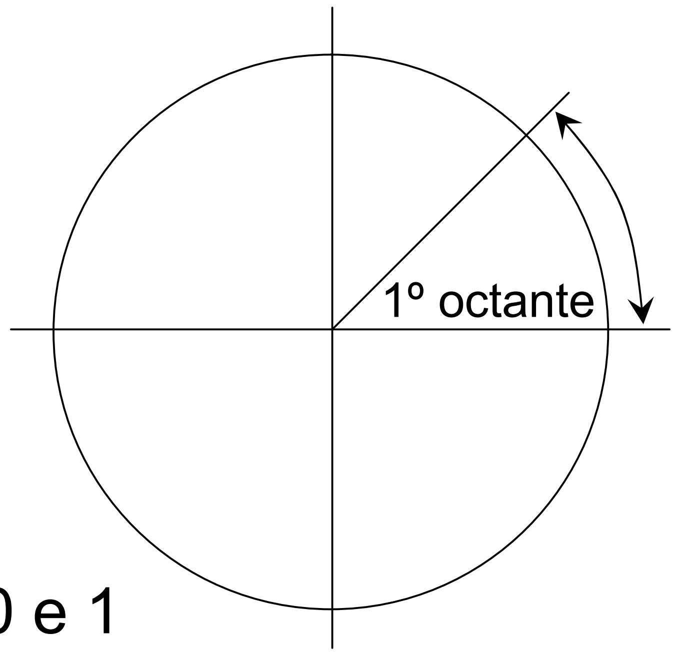
Este algoritmo, de 1965, permite o traçado de linhas retas com importantes vantagens:

- Não usa operações em ponto flutuante, tornando mais rápida a sua execução
- Usa apenas operações de soma e subtração e multiplicações por dois, que podem ser substituídas por deslocamentos de bits
- Pode ser implementado em hardware simples ou em assembly de processadores mais básicos (anos 70)

Algoritmo de Bresenham - Traçado de Retas

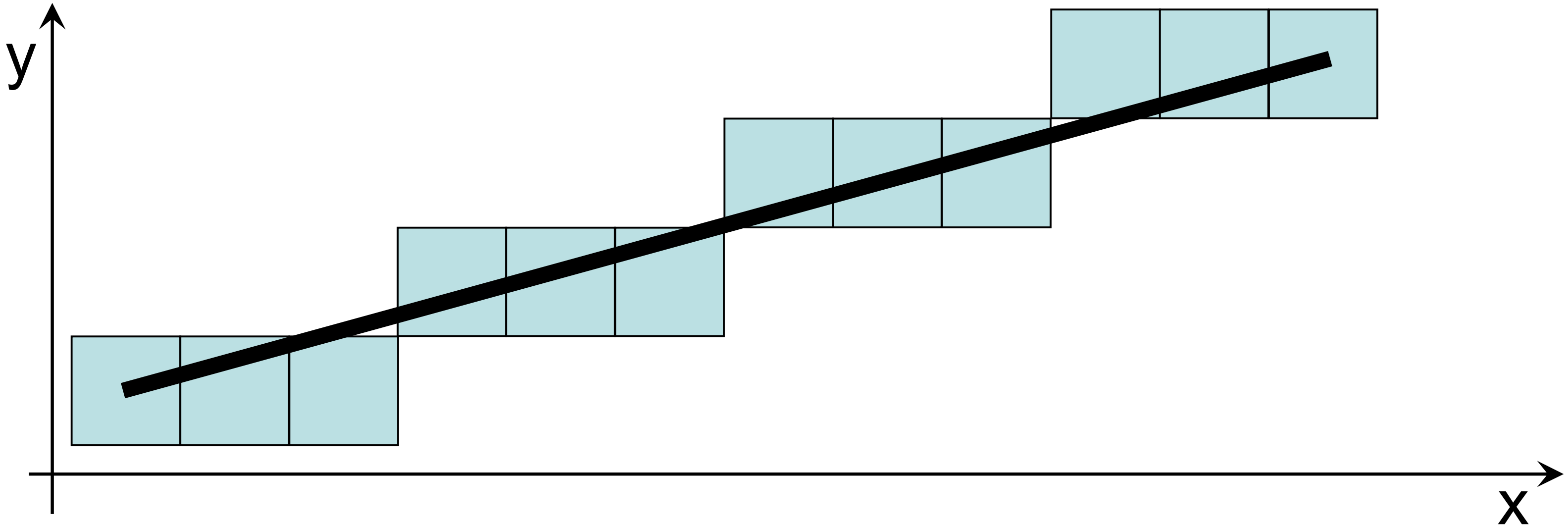






Assume que a linha está no primeiro octante, ou seja, tem inclinação entre 0 e 1

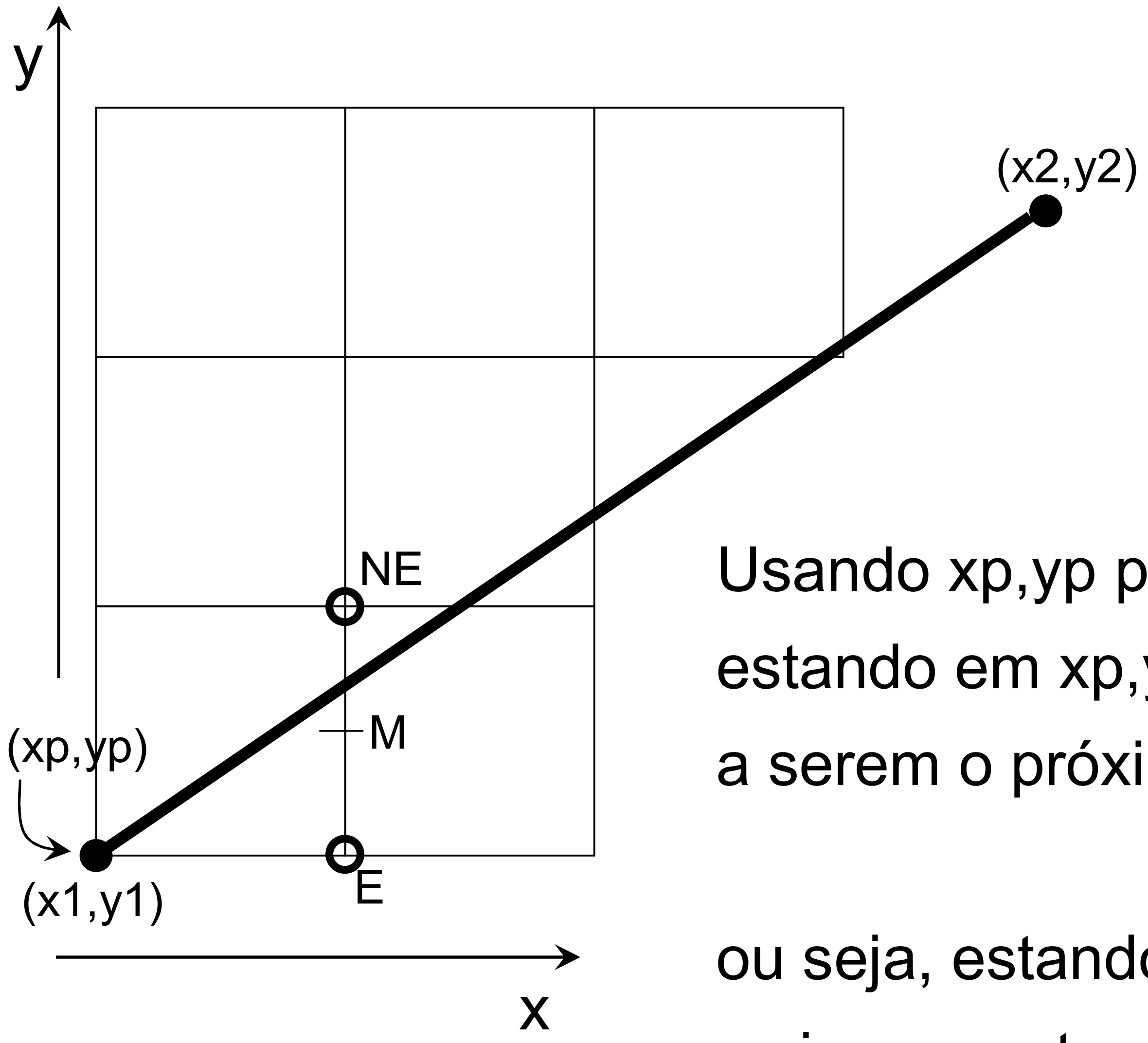
Algoritmo de Bresenham - Traçado de Retas



Neste octante, x sempre deve ser incrementado de um em um e, o y será calculado

Em resumo, x sempre é incrementado e,
y é ou não é incrementado

Algoritmo de Bresenham - Traçado de Retas



Usando xp,yp para ficar genérico,

estando em xp,yp, os dois candidatos

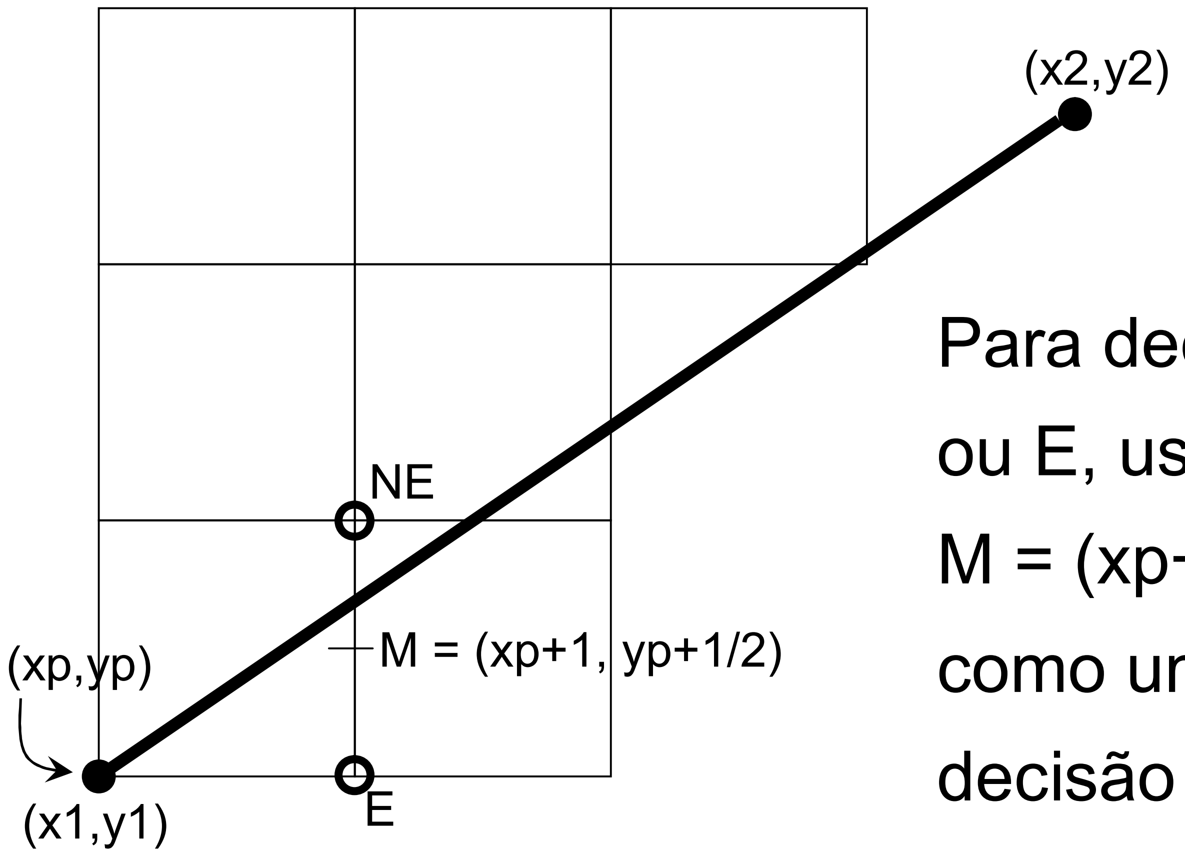
a serem o próximo pixel são NE e E

ou seja, estando em xp,yp:

se incrementar x e y irá para NE

se incrementar x e manter y, irá para E

Algoritmo de Bresenham - Traçado de Retas



Para decidir se vai para NE ou E, usa-se o ponto médio M = (xp+1, yp+1/2) como uma variável de decisão

Algoritmo de Bresenham - Traçado de Retas

Se M está abaixo da reta (figura 1), a reta está mais próxima de NE

Se M está acima da reta(figura 2), a reta está mais próxima de E

Figura 1

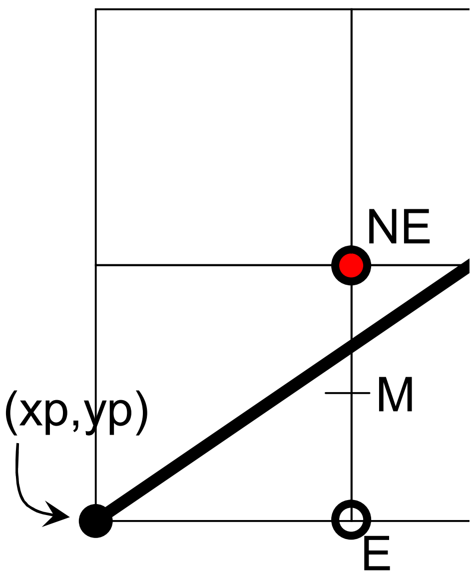
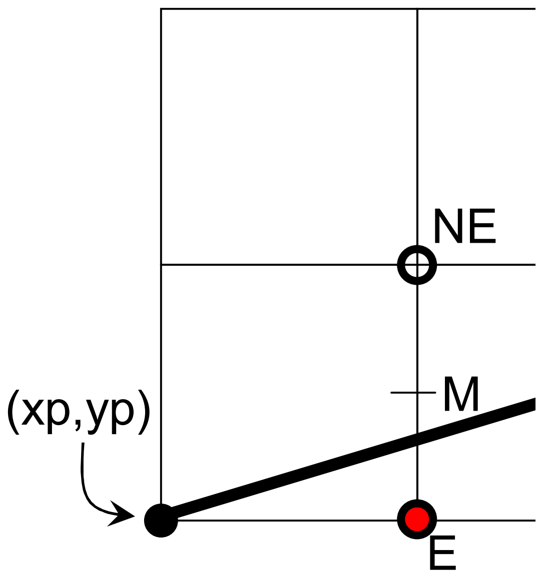


Figura 2



Algoritmo de Bresenham - Traçado de Retas

A equação implícita da reta é **f(x,y) = ax + by + c = 0**

A equação da reta em termos de sua inclinação é:

y = A.x + B

e a inclinação da reta é dada por A = dy/dx

onde: dx = x2 – x1
e dy = y2 – y1

assim: $$y = \frac{dy}{dx}.x + B$$

$$y = \frac{dy.x + B.dx}{dx}$$

$$y.dx = dy.x + B.dx$$

$$f(x, y) = \underbrace{dy}_{a}.x \underbrace{- dx}_{b}.y + \underbrace{B.dx}_{c} = 0$$

$\boxed{\begin{array}{l}\textbf{Portanto:}\\ \mathbf{a = dy}\\ \mathbf{b = -dx}\\ \mathbf{c = B.dx}\end{array}}$

Algoritmo de Bresenham - Traçado de Retas

Sabe-se que:

- para pontos acima da reta, $f(x,y) < 0$

- para pontos na reta, $f(x,y) = 0$

- para pontos abaixo da reta, $f(x,y) > 0$

Para testar se M está acima ou abaixo da reta, toma-se uma variável de decisão d

assim, faz-se $d = f(M)$

Supondo que se inicia no ponto xp,yp,

a tomada de decisão se dará no ponto $M=(xp+1,yp+1/2)$

ou seja, calcula-se $d = f(M) = f(xp+1,yp+1/2)$

Algoritmo de Bresenham - Traçado de Retas

aplicando a função *f* em $f(xp+1, yp+1/2)$

tem-se: $d = f(M) = a.(xp+1) + b(yp+1/2) + c$

e, se $d > 0$, significa que M está abaixo da reta (como na figura 1), logo, a reta está mais próxima de NE

caso contrário, se $d \leq 0$, então M está acima ou na reta (como na figura 2), assim, deve-se escolher E

Figura 1

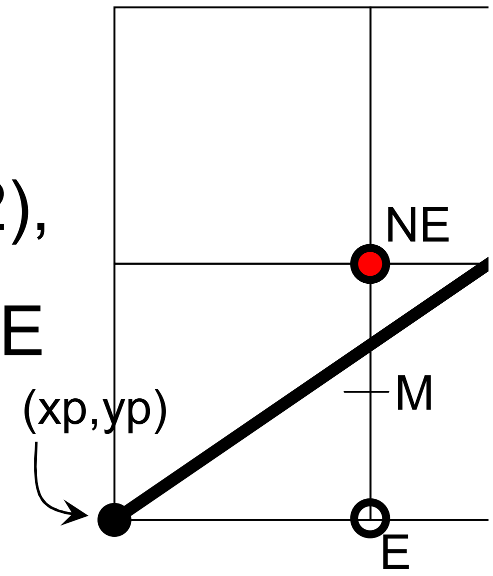
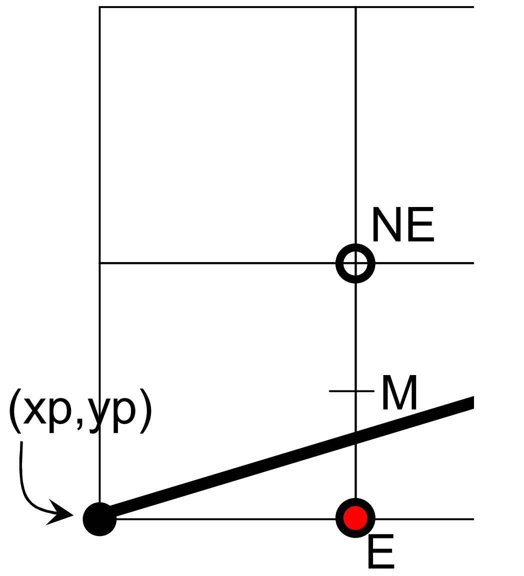


Figura 2



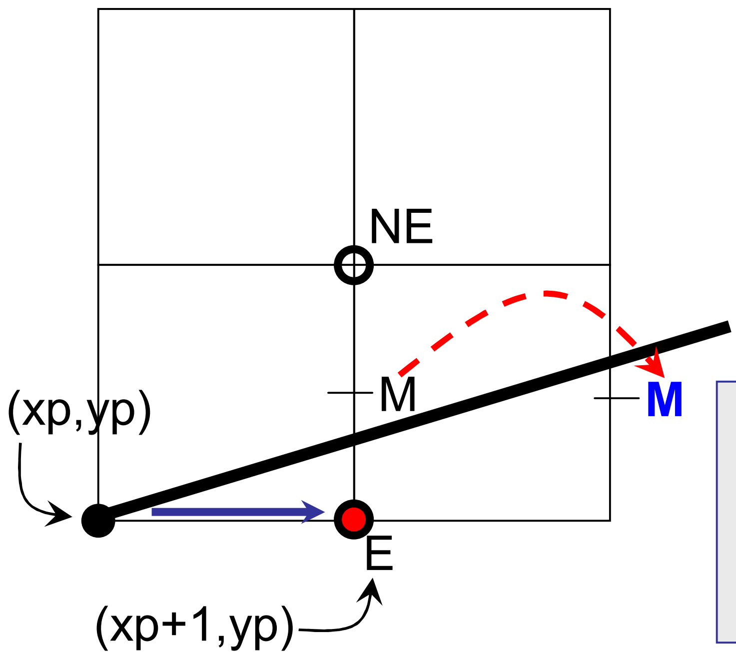
Algoritmo de Bresenham - Traçado de Retas

Qual o valor de *d* no próximo ponto ?

isto depende da escolha feita anteriormente, pois:

Se escolheu E, apenas o x será incrementado, ou seja,

estaremos em E = (xp+1, yp) e, assim, o próximo ponto M



será   M = xp+**2**, yp+1/2

assim, d_{novo} = f(M) = f(xp+2, yp+1/2)

logo, d_{novo} = a(xp+2)+b(yp+1/2)+c

Para reduzir o número de cálculos, Bresenham faz uso do calculo incremental

Algoritmo de Bresenham - Traçado de Retas

assim, para calcular $d_{novo} = a(\cancel{xp}+2) + b(\cancel{yp+1/2}) + \cancel{c}$

lembra-se que $\underline{d = a(\cancel{xp}+1) + b(\cancel{yp+1/2}) + \cancel{c}}$ $(-)$

assim, $d_{novo} - d = a$

portanto, $d_{novo} = d + a$

e, recordando o quadro abaixo, tem-se que quando

se escolhe E, dnovo = d + dy

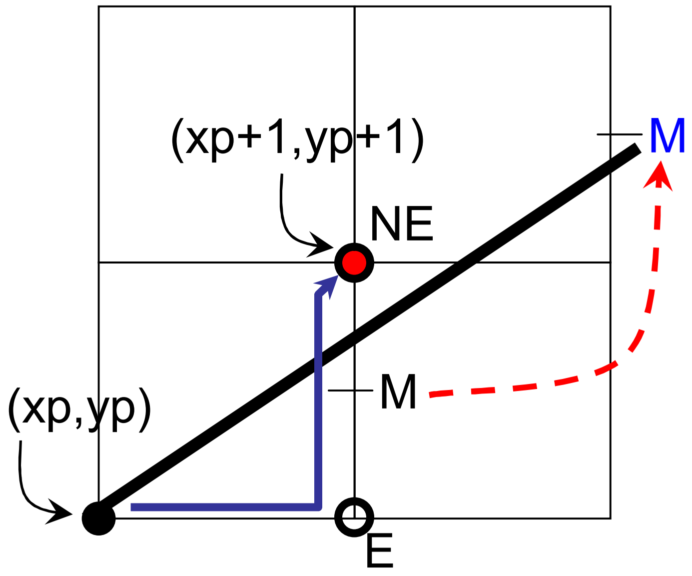
este acréscimo dy é chamado ΔE

<u>é o incremento após escolher E</u>

Portanto:
a = dy
b = -dx
c = B.dx

Algoritmo de Bresenham - Traçado de Retas

Por outro lado, se escolheu NE, ambos x e y serão incrementados, ou seja, estaremos em NE = (xp+1,yp+1) e, assim, o próximo ponto M



será M = xp+**2**, yp+**3/2**

assim, d_{novo} = f(M) = f(xp+**2**, yp+**3/2**)

logo, d_{novo} = a(xp+2)+b(yp+3/2)+c

Algoritmo de Bresenham - Traçado de Retas

assim, para calcular $d_{novo} = a(\cancel{xp}+2) + b(\cancel{yp}+3/2) + \cancel{c}$

(–)

lembra-se que $d = a(\cancel{xp}+1) + b(\cancel{yp}+1/2) + \cancel{c}$

assim, $d_{novo} - d = \quad a \quad + \quad b$

portanto, $d_{novo} = d + a + b$

e, recordamos o quadro abaixo

Portanto:
a = dy
b = -dx
c = B.dx

então, quando se escolhe NE

dnovo = d + dy - dx

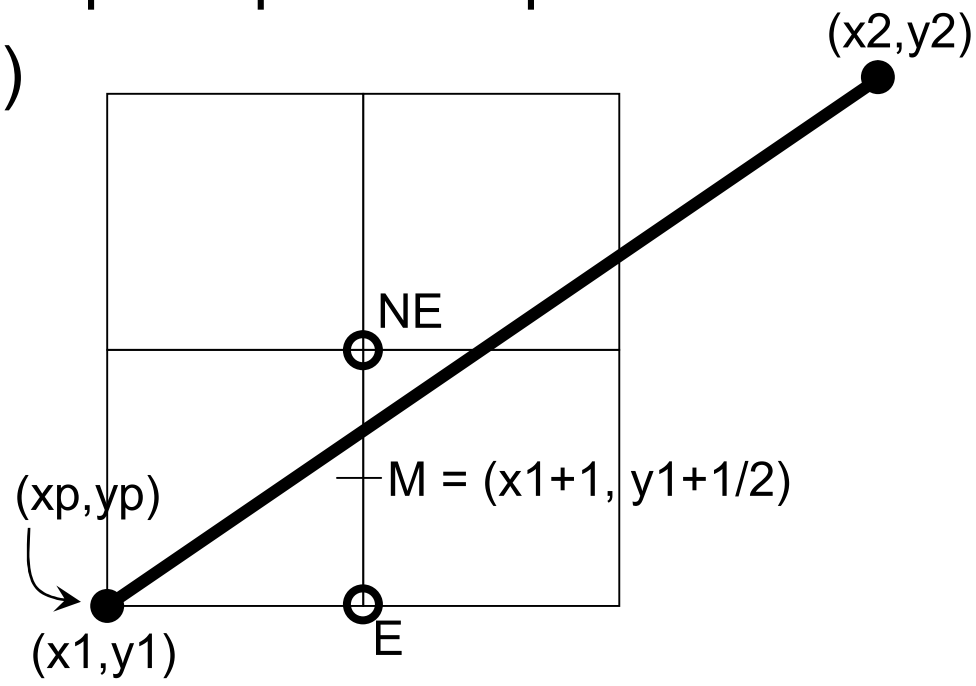
este acréscimo dy-dx é chamado ΔNE

é o incremento após escolher NE

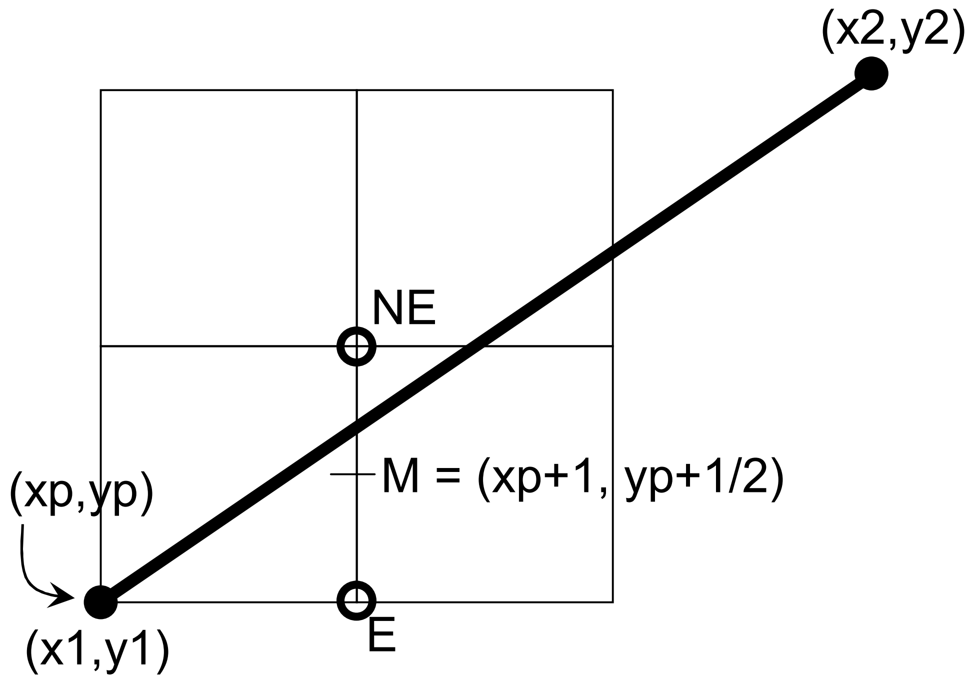
Algoritmo de Bresenham - Traçado de Retas

Usando o cálculo incremental, Bresenham reduz muito a quantidade de operações, pois o valor de ***d*** é obtido a partir do ***d*** anterior, mas ainda resta resolver o problema de definir o primeiro ***d***

Para isto, deve-se observar que o primeiro ponto médio é M = (x1+1,y1+1/2)



Algoritmo de Bresenham - Traçado de Retas



assim, deve-se calcular

$f(M) = f(x1+1, y1+1/2)$

$d = a.(x1+1)+b.(y1+1/2)+c$

$d = a.x1+a + b.y1+b/2 + c$

$d = \underbrace{a.x1 + b.y1 + c}_{=0} + a + b/2$

como o ponto (x1,y1) está na reta, **f(x1,y1)=0**

$d = a + b/2$ e, recordando o quadro

$\boxed{\begin{array}{l} \mathbf{a = dy} \\ \mathbf{b = -dx} \\ \mathbf{c = B.dx} \end{array}}$

$d = dy - dx/2$

Algoritmo de Bresenham - Traçado de Retas

Como se deseja saber apenas se $d > 0$, para escolher NE pode-se evitar a fração $dx/2$ em

$d = dy - dx/2$ multiplicando tudo por 2, ficando:

$d = 2.dy - dx$

Mas, então, é preciso fazer o mesmo com ΔNE e ΔE, ou seja, $\Delta NE = 2.(dy - dx)$

e $\Delta E = 2.dy$

Algoritmo de Bresenham - Traçado de Retas

Algoritmo:

```
{  int x1,y1,x2,y2,cor;
   dx = x2 – x1
   dy = y2 – y1
   d = 2 * dy – dx
   dE = 2 * dy
   dNE = 2 * (dy – dx)
   x = x1
   y = y1
   pixel[x,y] = cor
   while (x < x2)
   {  if (d < 0)              // escolhe E
        {  d = d + dE
           x = x+1
        }
      else
        {  d = d + dNE     // escolhe NE
           x = x+1
           y = y+1
        }
      pixel[x,y] = cor
   }
}
```

Usa apenas operações com inteiros
Usa apenas somas e subtrações
As multiplicações podem ser implementadas usando deslocamento de bits

Algoritmo de Bresenham - Traçado de Retas

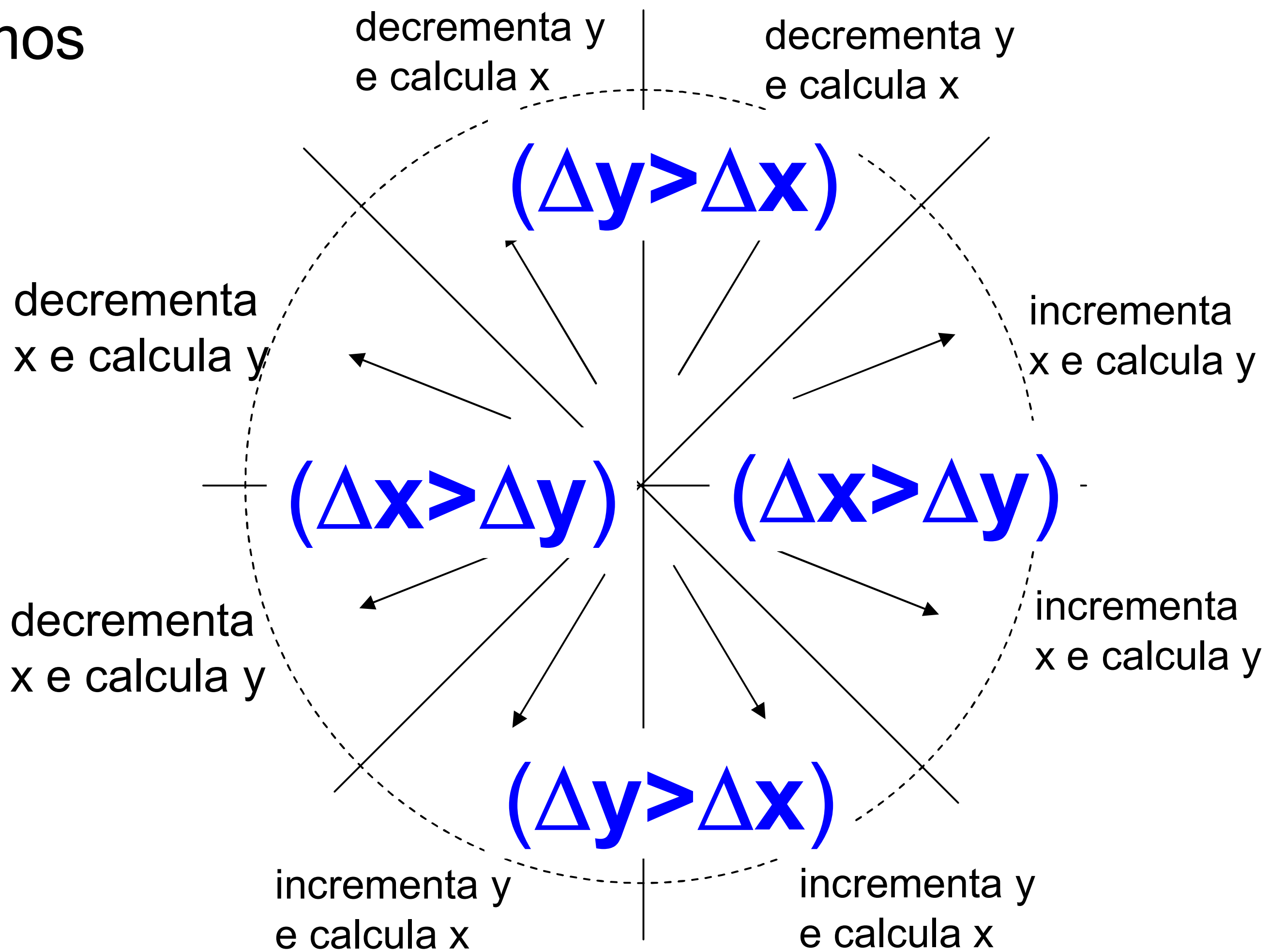
Este algoritmo funciona apenas no **1º octante**, mas pode ser adaptado para operar em todos os octantes

fazendo acréscimos

como

incrementando x
if (x1 < x2)
then inc =1
else inc = -1

incrementando y
if (y1 < y2)
then inc =1
else inc = -1



Algoritmo de Bresenham - Traçado de Retas

Prática: **(para entregar)**

Implementar o algoritmo de Bresenham para o traçado de linhas retas, para operar em todos os octantes.

Existem várias implementações na Internet, retiradas dos livros.

Pode-se usar estes códigos, mas o aluno precisa ver que, aquilo que viu nesta aula, está lá no código.