REPRÉSENTATION DE COURBES

Deux formes possibles

- ***paramétrique***:

$$x = x(t) \textbf{ et } y = y(t) \tag{1}$$

- ***implicite***:

$$f(x, y) = 0 \tag{2}$$

Trois problèmes à résoudre

- **Choix des points de la courbe**
- **Calcul d'une valeur approchée de leurs coordonnées (du domaine du calcul scientifique)**
- **Tracé effectif sur une trame (*scan conversion*) c'est-à-dire du choix des pixels à allumer**

FORME PARAMÉTRIQUE

- **Un intervalle $[a, b]$, $P(t)$ la position du point pour la valeur $a \leq t \leq b$.**
- **Problème: déterminer les valeurs $t_0 = a, t_1, \ldots, t_n = b$ de façon à ce que les points représentés ne soient ni trop rapprochés ni trop éloignés.**

On fixe d et on prend $\Delta t_i = t_{i+1} - t_i$, avec

$$d = \Delta t_i \sqrt{x'(t_i)^2 + y'(t_i)^2}$$

Exemple — Pour la parabole d'équation paramétrique $x = t$ et $y = t^2$, avec $\Delta t_i = t_{i+1} - t_i$, on a

$$d = \Delta t_i \sqrt{1 + 4t_i^2}$$

Ici Δt doit être inversement proportionnel à t.

FORME IMPLICITE

On dérive $f(x, y) = 0$ et on obtient l'équation différentielle:

$$\frac{\partial f}{\partial x} dx + \frac{\partial f}{\partial y} dy = 0 \tag{3}$$

On pose $F_x(x, y) = \frac{\partial f}{\partial x}$ et $F_y(x, y) = \frac{\partial f}{\partial y}$ d'où l'équation différentielle:

$$\frac{dy}{dx} = -\frac{F_x(x, y)}{F_y(x, y)}$$

donnant lieu à l'approximation discrète:

$$\begin{cases} x_{k+1} & = x_k + cF_y(x_k, y_k) \\ y_{k+1} & = y_k - cF_x(x_k, y_k) \end{cases} \tag{4}$$

où c est une constante arbitraire dont la valeur détermine la densité de la courbe.

Attention: problème d'instabilité numérique car (4) n'est qu'une approximation de (3).

EXEMPLE

Cercle centré à l'origine:

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

Il est solution de l'équation différentielle

$$\frac{dy}{dx} = -\frac{x}{y}$$

Alors (4) donne la suite linéaire récurrente:

$$\begin{cases} x_{k+1} & = x_k + cy_k \\ y_{k+1} & = y_k - cx_k \end{cases} \tag{5}$$

En fait (5) définit une spirale. Pour que la courbe se referme on considérera la suite récurrente dont la solution est en fait une ellipse:

$$\begin{cases} x_{k+1} & = x_k + cy_k \\ y_{k+1} & = y_k - cx_{k+1} \end{cases} \tag{6}$$

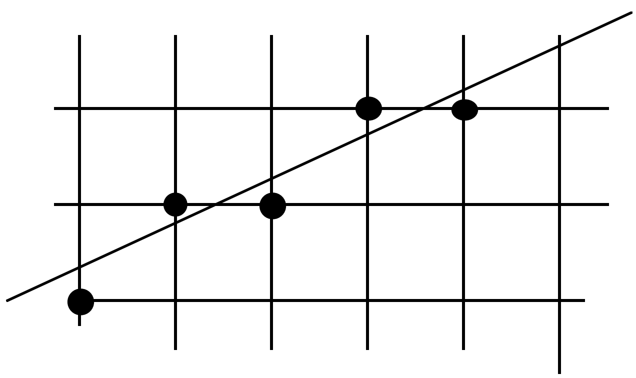
Pour obtenir un tracé de cercle correct on considérera la recurrence suivante pour θ petit:

$$\begin{cases} x_{k+1} & = x_k cos\theta \quad + \ y_k sin\theta \\ y_{k+1} & = -x_k sin\theta \ + \ y_k cos\theta \end{cases} \tag{7}$$

ANALYSEUR INCRÉMENTAL DE TRACÉ DE DROITE (DDA)

(Digital Differential Analyser)

- **Convention: on considère les centres des pixels placés sur les points de coordonnées entières.**
- **But: éviter les calculs en nombres réels. Coder "au plus près":**



De l'équation différentielle de la droite :

$$\frac{dy}{dx} = m$$

on déduit la suite réccurrente:

$$\begin{cases} x_{i+1} = x_i + e \\ y_{i+1} = y_i + me \end{cases} \tag{8}$$

On remplace l'opération (coûteuse) d'arrondi par une troncature en observant

$$\textbf{arrondi}(x) = \lfloor x + 0.5 \rfloor$$

L'analyseur incrémental simple privilégie une des deux coordonnées. On choisit $e = 1$ lorsque $|m| < 1$ (calculer y pour les valeurs entières de x). On inverse les rôles de x et y lorsque $|m| > 1$.

procedure DDA;

{*tracé d'un segment donné par ses deux extremités*
P1 $= (x1, y1)$ *et* P2 $= (x2, y2)$.
Les coordonnées sont des réels et la pente de la droite est $0 \le |m| \le 1$}

var

  $XInit$, $XFinal$, $XIncrement$, x: **integer;**
  $YIncrement$, y: **real;**

début

{*choix du rôle des coordonnées suivant la valeur de* m}

  $XIncrement := 1$;
  $YIncrement := m$;

{*initialisation*}

  $XInit$ **:=** **trunc**$(x1 + 0.5)$;
  $XFinal$ **:=** **trunc**$(x2 + 0.5)$;
  $x := XInit$;
  $y := y1 + 0.5$;

{*tracé effectif* }

  tantque $x <= XFinal$ **faire**
  début
    point$(x$, **trunc**$(y))$;
    $x := x + XIncrement$;
    $y := y + YIncrement$
  fin

fin;

Tracer un segment de droite arbitraire

procedure DDAComplet;

{*tracé d'un segment quelconque donné par ses deux extremités* P1 =
et P2 = $(x2, y2)$. *Les coordonnées sont des réels*}

début

$deltax := x2 - x1$;

$deltay := y2 - y1$;

si $abs(deltax) >= abs(deltay)$ **alors**

début

si $deltax < 0$ **alors** $Swap(P1, P2)$;

{*recopier le corps de la procedure antérieure* }

fin

sinon

début

si $deltay < 0$ **alors** $Swap(P1, P2)$

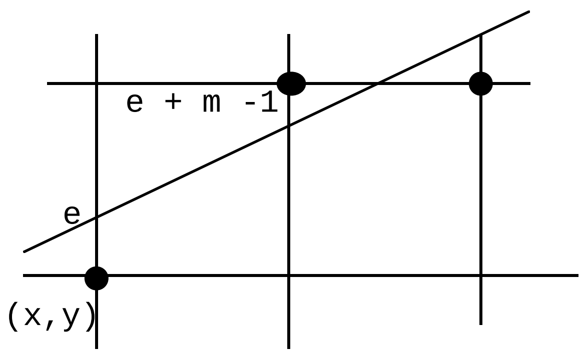
{*recopier le corps de la procedure antérieure en échangeant x et y*}

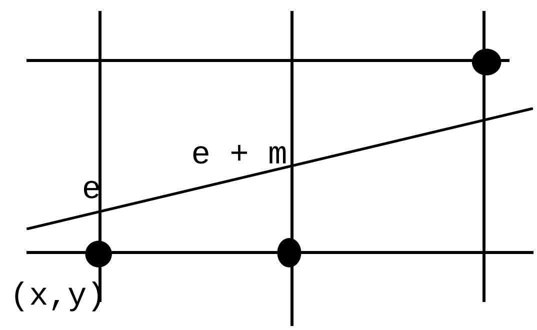
fin

fin;

MÉTHODE DE BRESENHAM DE TRACÉ DE DROITE

- **Hypothèse: la pente de la droite vérifie** $0 < m < 1$
- **But: à éviter, autant que possible, les opérations en réels flottants**
- **Méthode: retenir l'erreur** $-0.5 < e < 0.5$





On choisit le pixel d'abscisse $x + 1$ **ainsi:**

$$\begin{cases} (x+1, y+1) & \textbf{si } e + m \geq 0.5 \quad \{\ e := e + m - 1\} \\ (x+1, y) & \textbf{sinon} \qquad\qquad\ \ \{\ e := e + m\} \end{cases} \tag{9}$$

De la sorte, e **vérifie toujours les inégalités ci-dessus.**

```
procedure DroiteBresenham;
{0n suppose que le segment est dans le premier octant
Les coordonnées sont des entiers.}
début
    deltax := x2 - x1;
    deltay := y2 - y1;
    m := deltay/deltax;
(1) Erreur := 0;
    x := x1;
    y := y1;
    tantque x <= x2 faire début
        point(x, y);
        x := x + 1;
(2)     si Erreur + m >= 0.5 alors début
            y := y + 1;
(3)         Erreur := Erreur + m - 1
        fin
        sinon
(4)         Erreur := Erreur + m
    fin
fin;
```

Pour travailler avec des nombres entiers et pour remplacer le test à 0.5 par un test à 0 dans l'instruction (2), on fait le changement de variable:

$$e' = 2\Delta x * e + 2\Delta y - \Delta x$$

On modifie les lignes (1), (2), (3) ,(4) respectivement par:

(1) $Erreur := 2deltay - deltax$

(2) si $Erreur >= 0$ **alors**

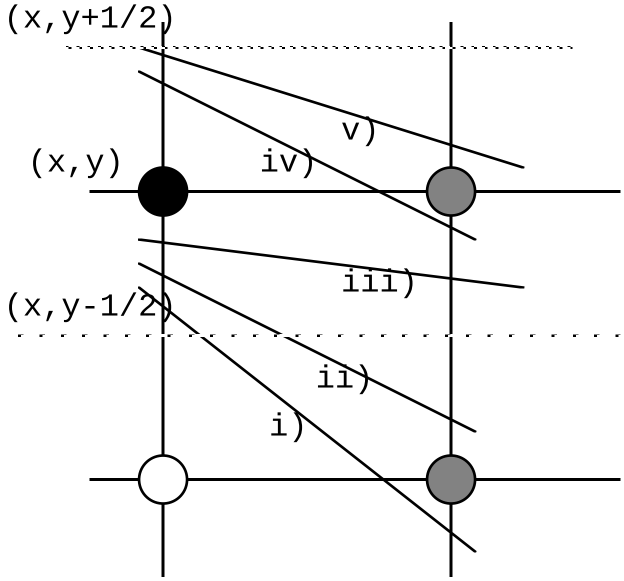
(3) $Erreur := Erreur + 2 * (deltay - deltax)$

(4) $Erreur := Erreur + 2 * deltay$

MÉTHODE DE BRESENHAM DE TRACÉ DE CERCLES

- **Données: un cercle C de rayon R supposé centré à l'origine.**
- **But: travailler en nombres entiers (R lui-même est supposé entier)**
- **Méthode: tracer la partie du cercle dans le deuxième octant et compléter par symétries**
- **Remarque: la tangente a une pente comprise entre -1 et 0.**

5 positions distinctes possibles du point courant



P$= (x, y)$ **le point courant,** $\mathbf{P}_h = (x+1, y)$ **le point "à droite" et** $\mathbf{P}_d = (x+1, y-1)$ **le point "diagonal" Le point suivant P doit être choisi entre** $\mathbf{P}_h$ **et** $\mathbf{P}_d$**. On pose:**

$$\begin{cases} \Delta_d &= (x+1)^2 + (y-1)^2 - R^2 \quad \textbf{et} \quad m_d &= |\Delta_d| \\ \Delta_h &= (x+1)^2 + y^2 - R^2 \quad \textbf{et} \quad m_h &= |\Delta_h| \end{cases} \tag{10}$$

Cas1: $\Delta_d \geq 0$ **donc** $\mathbf{P}_d$ **est à l'extérieur de C et a fortiori** $\mathbf{P}_h$ **: retenir** $\mathbf{P}_d$**.**

Cas 2: $\Delta_d < 0$ **et** $\Delta_h \leq 0$ **alors les deux points sont à l'intérieur du cercle : retenir** $\mathbf{P}_h$**.**

Si $\Delta_d < 0$ **et** $\Delta_h > 0$ **alors** $\mathbf{P}_d$ **est à l'intérieur et** $\mathbf{P}_h$ **à l'extérieur du cercle. On pose** $\Delta = m_d - m_h$**.**

Cas 3: $\Delta > 0$ **: retenir** $\mathbf{P}_d$**.**

Cas 4: sinon retenir $\mathbf{P}_h$**.**

```
procedure ArcBresenham;
début
    moveto(0, R);
    y := R;
    x := 0;
    DeltaDiag := -2 * R + 2;
    DeltaHor := 1;
    tantque (y > x) faire début
        {tant que l'on est dans le deuxième octant}
        Cas1 := (DeltaDiag >= 0);
        Cas3:= ((DeltaDiag < 0) et (DeltaHor > 0)
            et ((DeltaDiag + DeltaHor) > 0);
        CasDiag := Cas1 ou Cas3;
        si CasDiag alors début
            {choix du point diagonal}
            line(1, -1);
            DeltaDiag := DeltaDiag + 2 * (x - y) + 6;
            DeltaHor := DeltaHor + 2 * (x - y) + 4; {mise à jour de DeltaH
            x := x + 1;
            y := y - 1
            fin
            sinon début
                {choix du point horizontal}
                line(1, 0);
                DeltaDiag := DeltaDiag + 2 * x + 3;
                DeltaHor := DeltaHor + 2 * x + 3;
                x := x + 1
                fin
        fin
    fin
```

References

[1] Peroche B. *La synthèse d'image.* Hermes, 1988.

[2] Rogers D.F. *Procedural Elements for Computer Graphics.* International Student Edition, 1985. (bon ouvrage, mais recouvre seulement une partie du cours, essentiellemnt la partie algorithmique).

[3] Hégron G. *Synthèse d'image: algorithmes élémentaires.* Dunod, 1985. (ouvrage français).

[4] Foley J.D., Van Dam A, Feiner S. K, and Hughes J.F. *Computer Graphics.* Addison-Wesley, 1990. (très complèt).

[5] Ammeraal L. *Programming Principles in Computer Graphics.* John Wiley, 1986. (clair et didactique. Programmation en C des algorithmes de l'espace en 3 dimensions).

[6] Salmon R. and M. Slater. *Computer Graphics, Systems and Concepts.* Addislon-Wesley, 1987. (basé sur GKS et le langage ML).