Rapport MA316

Florian Luce

L'objectif de ce projet est d'écrire une fonction qui, à la donnée d'un point de $\mathbb{R}^2$, indique s'il appartient à l'intérieur d'un polygone P. Dans ce projet, le lancement du programme affichera une fenêtre gnuplot affichant le polygone à partir d'un fichier DAT. Lorsque l'utilisateur fera un clic gauche sur le graphique, il aura alors sélectionné le point dont les coordonnées correspondent à la position du curseur au moment du clic.

1 Polygone non auto-intersecté (fonctionne avec ou sans trous)

Des fichiers DAT contiennent une série de points en deux dimensions dont une ligne correspond à un point. On forme alors un polygone en reliant chaque point à celui de la ligne suivante. Le dernier point doit être le même que le premier afin que la figure soit fermée.

L'idée de l'algorithme est la suivante :
— On trace une droite horizontale passant le point à tester
— On compte le nombre d'intersections se situant à droite de ce point
— Si il est impair, le point est dans le polygone, sinon, il ne l'est pas

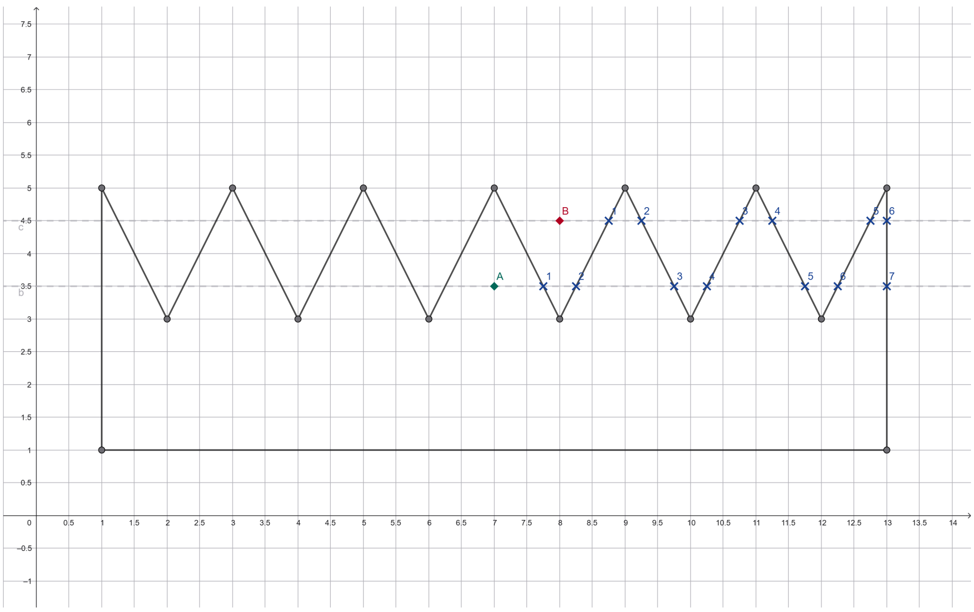

Figure 1 – Exemple d'application de l'algorithme sur un polygone en couronne, on peut voir ici que le point A appartient au polygone, contrairement à B

Dans la pratique, je regarde pour chaque segment si l'ordonnée du point testé est compris entre celles des points composant le segment, cela permet de savoir si le segment est coupé par la droite horizontale passant le point.

Ensuite, j'ai d'abord pensé qu'il suffirait de comparer les abscisses du point testé avec les points composant le segment, cela aurait été correcte si les segments étaient verticaux. Mais voici un contre exemple :

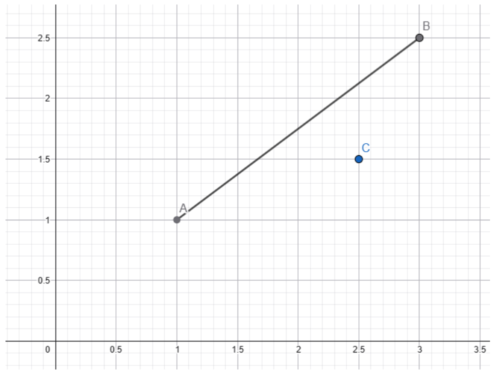

Figure 2 – On peut clairement voir que les coordonnées de C sont bornées par celle de A et de B, mais que C n'appartient pas au segment

Cela montre clairement la nécessité de calculer les points d'intersections, afin de voir si ils sont bien à droite du point.

Rappel :
Soit $A = \begin{pmatrix} x_A \\ y_A \end{pmatrix}$ et $B = \begin{pmatrix} x_B \\ y_B \end{pmatrix}$ deux points distincts de $\mathbb{R}^2$. Une droite passant par A et B signifie qu'il existe une fonction $f : \mathbb{R} \to \mathbb{R}$ et $(m, p) \in \mathbb{R}^2$ tels que :

$$\begin{cases} f(x_A) = mx_A + p = y_A \\ f(x_B) = mx_B + p = y_B \end{cases}$$

On a donc $m = \frac{y_B - y_A}{x_B - x_A}$, et $p = y_A - mx_A$, on trouve alors :

$$f(x) = \frac{y_B - y_A}{x_B - x_A}(x - x_A) + y_A$$

Si une fonction constante g intersecte f, alors il existe $c \in \mathbb{R}$ tel que :

$$\forall x \in \mathbb{R},\ g(x) = c$$

Trouver l'abscisse d'intersection revient donc à résoudre $\frac{y_B - y_A}{x_B - x_A}(x - x_A) + y_A = c$.
On obtient alors $x = x_A + \frac{(c - y_A)(x_B - x_A)}{y_B - y_A}$

Je sais que ce rappel peut sembler rudimentaire mais je tenais à le faire afin de justifier la formule utilisée dans la fonction `Dans_polygone_sans_auto_intersection`.

Ensuite, sans raison, j'ai décidé d'inclure les frontières.

Un point C appartient à un segment [AB] du polygone si les points A, B et C sont alignés et si C est entre A et B. C'est à dire s'il existe un réel $t \in [0, 1]$ tel que $\overrightarrow{AB} = t\overrightarrow{AC}$.

Rappel : Soient $A = \begin{pmatrix} x_A \\ y_A \end{pmatrix}$, $B = \begin{pmatrix} x_B \\ y_B \end{pmatrix}$ et $C = \begin{pmatrix} x_C \\ y_C \end{pmatrix}$ trois points de $\mathbb{R}^2$ avec $A \neq B$.
Montrons que $\overrightarrow{AB}$ est colinéaire à $\overrightarrow{AC}$ revient à montrer qu'il existe $t \in \mathbb{R}$ tel que :

$$\overrightarrow{AC} = t\overrightarrow{AB}$$
$$\begin{pmatrix} x_C - x_A \\ y_C - y_A \end{pmatrix} = \begin{pmatrix} t \times (x_B - x_A) \\ t \times (y_B - y_A) \end{pmatrix}$$

C'est à dire :

$$\begin{cases} x_C - x_A = t \times (x_B - x_A) \\ y_C - y_A = t \times (y_B - y_A) \end{cases}$$
$$\begin{cases} (x_C - x_A)(y_B - y_A) = t \times (x_B - x_A)(y_B - y_A) \\ (y_C - y_A)(x_B - x_A) = t \times (y_B - y_A)(x_B - x_A) \end{cases}$$

Et donc un tel t existe si on a : $(x_C - x_A)(y_B - y_A) = (y_C - y_A)(x_B - x_A)$, et on aurait $\overrightarrow{AB}$ colinéaire à $\overrightarrow{AC}$.

Rappel :
Soit $\overrightarrow{AB} = \begin{pmatrix} x_B - x_A \\ y_B - y_A \end{pmatrix}$ et $\overrightarrow{AC} = \begin{pmatrix} x_C - x_A \\ y_C - y_A \end{pmatrix}$ deux vecteurs colinéaires avec $A \neq B$ tel qu'il existe $t \in \mathbb{R}$ tel qu'on ait $\overrightarrow{AC} = t\overrightarrow{AB}$.

Le point C appartient à $[AB]$ si le vecteur $\overrightarrow{AC}$ est une fraction du vecteur $\overrightarrow{AB}$, c'est à dire si $t \in [0, 1]$.

Comme on a supposé $A \neq B$, on a soit $(x_B - x_A) \neq 0$, soit $(y_B - y_A) \neq 0$ (ou inclusif), on a :

$$t = \frac{x_C - x_A}{(x_B - x_A)} \quad \text{ou(inclusif)} \quad t = \frac{y_C - y_A}{(y_B - y_A)}$$

2 Polygone auto-intersecté

Je rappelle que le polygone est représenté par une suite ordonnée de points dont chaque point forme un segment avec le suivant.

Soient $A = \begin{pmatrix} x_A \\ y_A \end{pmatrix}$, $B = \begin{pmatrix} x_B \\ y_B \end{pmatrix}$ et $C = \begin{pmatrix} x_C \\ y_C \end{pmatrix}$ trois points de $\mathbb{R}^2$.

L'idée de l'algorithme est la suivante :
- On trace une droite horizontale f passant par C.
- Pour chaque arête $[AB]$ du polygone, on vérifie si elle intersecte f.
- Vérifier si l'arête $[AB]$ traverse f de haut en bas ou de bas en haut. ($y_A > y_B$ ou $y_B > y_A$?)
- Calculer le sinus de l'angle orienté $(\overrightarrow{CA}, \overrightarrow{CB})$. (L'angle est il direct ou in direct ?)
- Si $y_A > y_B$ (resp. $y_B > y_A$) et $\sin((\overrightarrow{CA}, \overrightarrow{CB}))<0$ (resp. $\sin((\overrightarrow{CA}, \overrightarrow{CB}))>0$) alors on aura fait un tour autour du point C dans le sens horaire (resp. anti-horaire), on incrémente (resp. décrémente) une variable.
- Si la valeur finale de notre variable est 0 alors notre point C n'est entouré d'aucune arête, il sera donc à l'extérieur de notre polygone.

Problème notable, si C est à l'intérieur d'un trou, le polygone ferait un tour complet autour du point, et donc l'algorithme nous dira que C est à l'intérieur du polygone.

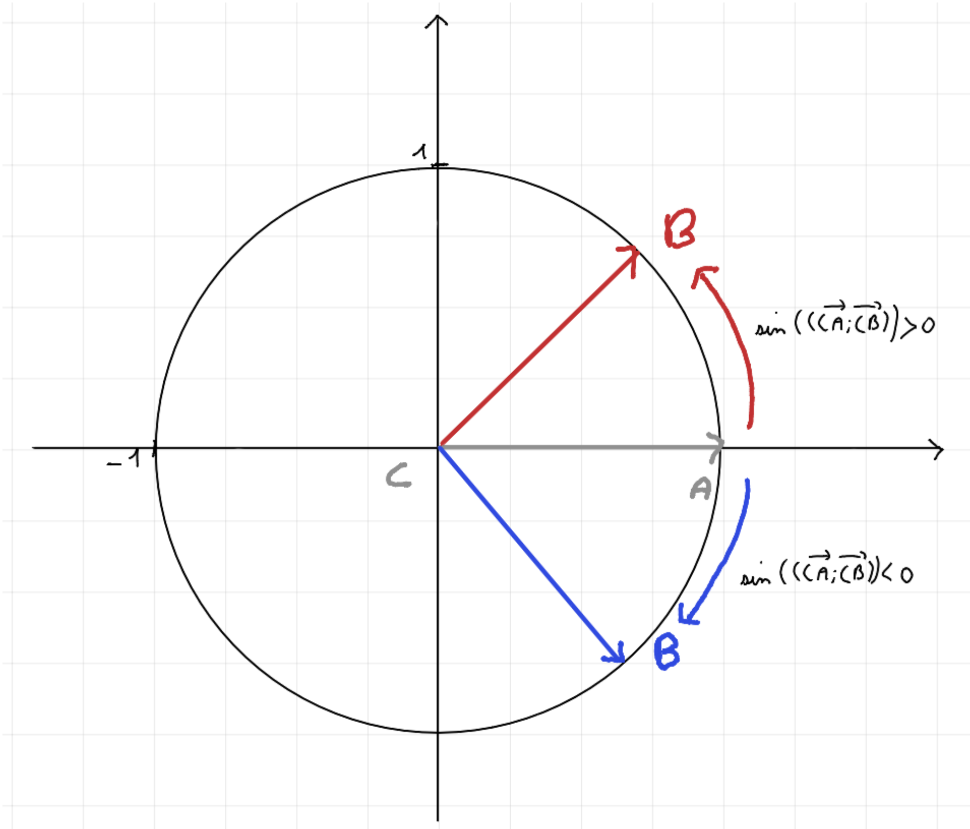

FIGURE 3 – On observe que si B est vers la gauche (resp. droite) par rapport à $\overrightarrow{CA}$, alors $\sin((\overrightarrow{CA}, \overrightarrow{CB})) > 0$ (resp. < 0).

Rappel : Soit $\overrightarrow{CA} = \begin{pmatrix} x_A - x_C \\ y_A - y_C \end{pmatrix}$ et $\overrightarrow{CB} \begin{pmatrix} x_B - x_C \\ y_B - y_C \end{pmatrix}$ deux vecteurs de $\mathbb{R}^2$, le produit vectoriel peut se calculer de deux manières :

$$\begin{cases} \overrightarrow{CA} \wedge \overrightarrow{CB} = (x_A - x_C)(y_B - y_C) - (y_A - y_C)(x_B - x_C) \\ \qquad\qquad \text{(calcul des normes avec le thm de pythagore)} \\ \overrightarrow{CA} \wedge \overrightarrow{CB} = \sqrt{(x_A - x_C)^2 + (y_A - y_C)^2}\sqrt{(x_B - x_C)^2 + (y_B - y_C)^2} \times \sin(\overrightarrow{CA}, \overrightarrow{CB}) \end{cases}$$

Ainsi on peut déduire le signe du $\sin((\overrightarrow{CA}, \overrightarrow{CB}))$ à partir de $(x_A - x_C)(y_B - y_C) - (y_A - y_C)(x_B - x_C)$.