

- TP 2 : Calcul d'enveloppes convexes -

Le but de ce TP est de calculer l'enveloppe convexe d'un nuage de points dans le plan en utilisant le parcours de Jarvis puis l'algorithme de Graham.

Le langage à utiliser est laissé libre. Toutefois, des primitives d'affichage et de tri ainsi que des trames de programme sont fournies en C++ à l'adresse :

<http://www.lirmm.fr/~bessy/AlgoGeo/accueil.html>

- Calcul d'enveloppes convexes -

Les points du plan sont repérés par leurs coordonnées supposées entières et appartenant à un domaine $[0, x_{max}] \times [0, y_{max}]$. Pour l'affichage en postscript, on prendra $x_{max} = 612$ et $y_{max} = 792$. Le logiciel *ggv* permet un affichage standard des documents postscript.

- Exercice 1 - Génération de points du plan -

Ecrire une fonction `PointAuHasard` qui génère un ensemble de n points du plan uniformément dans un disque centré en le centre de la zone d'affichage. Pratiquement, on prendra $n = 20$ et pour l'affichage en postscript, un disque de centre $(300, 400)$ et de rayon 250 convient.

- Exercice 2 -

Ecrire une fonction `AnglePolaireInferieur` qui prend en entrée trois points du plan p_0 , p_1 et p_2 (donnés par leurs coordonnées) et qui renvoie *Vrai* si p_2 est strictement à droite de la droite (p_0p_1) orientée de p_0 à p_1 et *Faux* sinon.

- Exercice 3 - Parcours de Jarvis -

Ecrire une fonction `Jarvis`, codant le parcours de Jarvis, qui prend en entrée un ensemble Q de n points du plan numérotés de 0 à $n - 1$, donnés par leur coordonnées, et qui retourne une liste *envconv* contenant la liste cyclique des indices des points formant les sommets de l'enveloppe convexe de Q . On commencera et terminera cette liste par le point d'ordonnée minimum. Une telle liste peut être codée par un tableau de taille $n + 1$, contenant les indices des sommets de l'enveloppe convexe et complétée par les valeurs -1, le gain de place par rapport à une liste chaînée n'étant pas significatif ici. Un exemple est donné Figure 1.

- Exercice 4 - Algorithme de Graham -

Sur le modèle de l'exercice précédent, écrire une fonction `Graham`, codant l'algorithme de Graham, prenant en entrée un ensemble de n points du plan et retournant la liste *envconv* correspondant à leur enveloppe convexe.

Pour cela, il faut utiliser un tri par angle polaire des points du plan, une telle fonction est fournie en C++ mais nécessite l'implémentation de la fonction `AnglePolaireInferieur` de l'exercice 2.

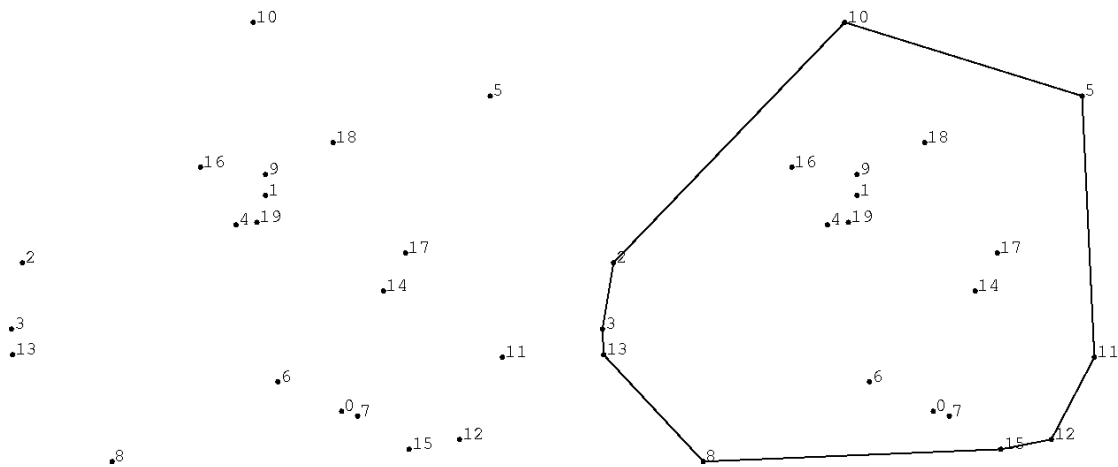


FIGURE 1 – Exemple : Un ensemble Q de points du plan, puis le même ensemble avec son enveloppe convexe. La liste *envconv* correspondante est (8, 15, 12, 11, 5, 10, 2, 3, 13, 8).

- Exercices supplémentaires -

- Exercice 5 - Pelures d'oignon -

Calculer les pelures d'oignon (voir Exercice 10 de la fiche de TD) en utilisant l'algorithme Jarvis, et faire afficher le résultat (en modifiant la fonction qui génère le fichier Postscript).

- Exercice 6 - Points les plus éloignés -

Implémenter une fonction `PointsExtremes` qui détermine en $O(n \log n)$ les deux points les plus éloignés parmi un ensemble de n points du plan.

- Exercice 7 - Maintien dynamique de l'enveloppe convexe -

Etant donné un ensemble Q de n points du plan et d'un point supplémentaire p , connaissant l'enveloppe convexe de Q , on souhaite calculer l'enveloppe convexe de $Q \cup x$ en un temps $O(n)$. Plus précisément, écrire une fonction `AjoutPoint` qui prend en entrée l'ensemble Q , le point x et la suite *envconv* correspondant à l'enveloppe convexe de Q et qui retourne en $O(n)$ la suite *NEWenvconv* correspondant à l'enveloppe convexe de $Q \cup x$.