

Algorithme de Bentley-Ottmann

I. Présentation

Ce projet propose une implémentation du célèbre algorithme de Bentley-Ottman permettant de trouver les intersections entre des segments par "balayage" horizontal du plan.

Le répertoire de travail contient plusieurs fichiers/répertoires:

- bo.py → module contenant la plupart des fonctions de l'algorithme de Bentley-Ottman
- lacer_tous_les_tests.sh → lance tous les tests
- /geo → répertoire contenant des modules de géométrie
- /sortedcontainers → répertoire contenant les modules permettant l'utilisation des sortedcontainers
- /tests → répertoire contenant les tests

II. Résultats et performances

Remarque: précision = n signifie une précision de $10^{(-n)}$

Remarque2: les temps d'exécutions prennent en compte le temps nécessaire pour afficher les images avec tycat

On constate que l'on obtient à peu près tous les points d'intersections à partir d'une précision de 10. En revanche, lorsque la précision dépasse 16, on assiste à quelques incohérences (beaucoup de points non détectés dans certains cas, programme qui prend énormément de temps pour random_200.bo avec precision=16, ...).

La précision optimale se trouve donc entre ces 2 valeurs.

Or dans [10, 15], le rapport (moyenne du nombre de points d'intersections détecté) / (moyenne du temps d'exécution du programme) est le plus élevé pour une précision égale à 14.

Nous prenons donc **PRECISION = 14**

Les tableaux et graphes contenant les résultats de nos tests et les valeurs sur lesquelles nous sommes appuyés pour choisir la précision sont dans **annexes.pdf**.