

Rapport préliminaire de projet APD

Calcul de figures fractales

Consigne et objectif

Sujet n°6 : Écrire un programme MPI pour calculer deux figures fractales IFS (Iterated Function System), le flocon de Koch et le dragon de Lévy. Le programme utilisera la bibliothèque Cairo pour dessiner des lignes et permettra de spécifier la profondeur d'itération de la fonction de transformation. La taille du "papier" sera calculée en fonction de la profondeur d'itération.

On demandera à l'utilisateur un nombre d'itérations. La création de la figure fractale, avec cairo, sera répartie entre les différents processeurs avec MPI. Afin de conserver la lisibilité de la figure quelle que soit l'itération, le format de l'image s'agrandira en conséquence.

Librairie cairo

Après avoir rapidement regardé les deux figures fractales en question pour savoir de quoi nous aurions besoin, nous nous sommes penchés sur la doc de cairo. Le seul point à retenir ici est qu'il existe 2 méthodes de dessin : en utilisant les coordonnées ou en orientant le curseur et en avançant d'une certaine longueur.

Algorithmique

Ces figures fractales se dessinent très bien avec des algos récursifs, ce qui ne correspond pas à notre besoin pour paralléliser correctement le programme.

Nous avons donc élaboré un premier algorithme itératif qui fonctionne pour les 2 figures :

- On enregistre en mémoire les segments initiaux de la figure.
- Une fonction élémentaire permet ensuite, à partir d'un segment, de calculer et enregistrer ses sous-segments de l'itération suivante. Par exemple pour Koch : un segment en devient 4, avec un triangle sur son ancienne partie centrale.
- On relance notre fonction sur les nouveaux segments jusqu'à obtenir la figure finale, le nombre total de segments d'une l'itération donnée étant connu.

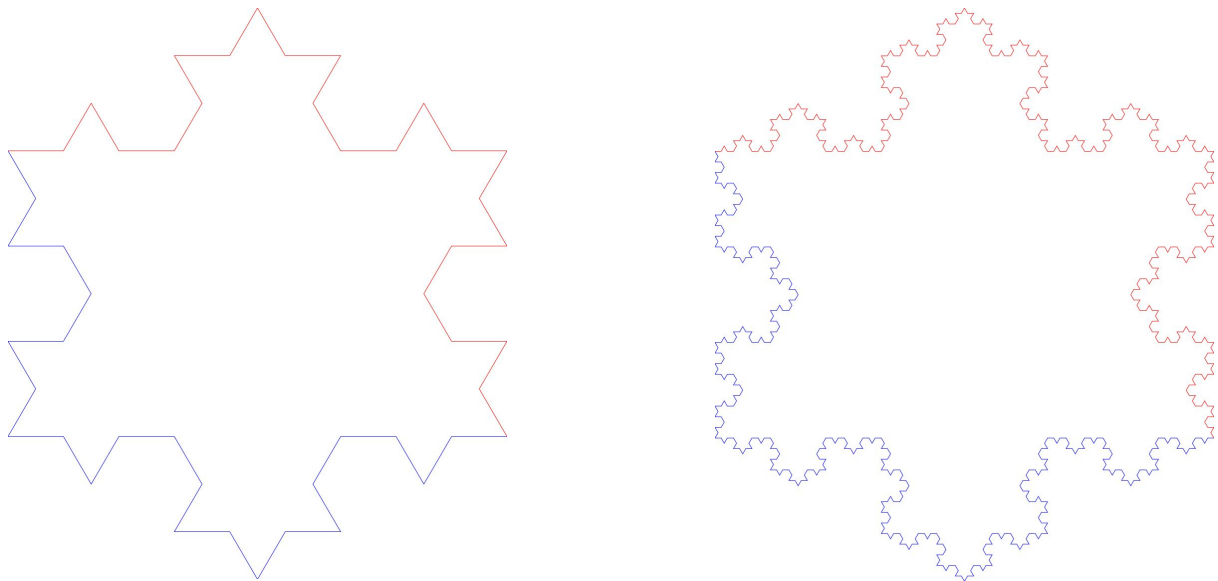
Certes cela fonctionne, mais cette solution est coûteuse en temps car on calcule toutes les itérations une à une, et coûteuse en espace pour enregistrer tous les segments.

Nous avons réussi à mettre au point un algorithme efficace pour le flocon de Koch, mais le dragon de Lévy nous résiste encore. Le nouvel algorithme repose sur l'étude de la suite

d'angles qui constituent la figure. On est capable de déterminer cette suite et, la longueur et le nombre de segments finaux étant connus, on peut tracer un segment puis s'orienter et recommencer. Cette solution nécessite uniquement un tableau de taille du nombre d'itérations en mémoire.

Et concrètement ?

Pour l'instant, nous avons réalisé le flocon de Koch de façon efficace en local.



Ci-dessus le flocon de Koch que nous avons produit pour les itérations 2 et 4.

Celui-ci est bicolore car nous avons utilisé deux curseurs différents pour commencer nos tests sur la parallélisation.

En passant à une vraie parallélisation avec MPI, la mémoire ne sera pas partagée. Il faudra donc que chaque processeur dessine sa partie sur une surface distincte transmise ensuite au processeur "père" qui fera la fusion.

Cairo nous permet de gérer ces fusions et cela ne posera pas de problème. En revanche, la transmission de surface entre processeurs sera plus (très ?) compliquée.

Limites

En omettant nos difficultés à mettre au point un algorithme efficace pour le dragon de Lévy, nous nous heurtons à 2 problèmes.

D'abord, concernant la parallélisation. Chaque processus sera chargé de dessiner une partie de la figure. Ceux-ci doivent être disposés à des points précis de la figure, des points que nous devons contrôler pour attribuer correctement le travail. De plus, si nous ne sommes pas maîtres du nombre de processeurs, il sera compliqué de paralléliser de façon efficace.

On propose donc de fixer le nombre de processeurs disponibles. On pourrait au mieux mettre en place plusieurs scénarii pour des nombres de processeurs spécifiques.

Enfin, un problème concerne la visualisation de la figure. La taille du papier doit croître avec le nombre d'itérations pour que le résultat soit visible. Or, si la taille de l'image est trop importante, cairo ne la génère plus correctement (sans compter qu'il peut être compliqué d'ouvrir le fichier, au moins sur les machines de l'école).

Nous avons pensé au format SVG qui pourrait être idéal dans notre cas. Nous avons trouvé une librairie permettant de générer des images SVG, on espère que cela résoudra bien ce problème.

Bibliographie

https://fr.wikipedia.org/wiki/Flocon_de_Koch

https://fr.wikipedia.org/wiki/Courbe_de_L%C3%A9vy

(Ainsi que leur équivalent en anglais) pour avoir un aperçu global de ces deux figures fractales et des méthodes pour les construire.

<https://www.geeksforgeeks.org/koch-curve-koch-snowflake/>

Présentation détaillée du flocon de Koch et d'un algorithme récursif le réalisant.

<http://ecademy.agnesscott.edu/~lriddle/ifs/levy/levy.htm>

Présentation détaillée des deux méthodes les plus courantes pour réaliser le dragon de Lévy.

<https://www.cairographics.org/documentation/>

Documentation de la librairie cairo et exemples de code.

<http://lists.cairographics.org/archives/cairo/attachments/20120914/6ac47ff8/attachment-0001.c>

Exemple simple utilisant la librairie cairo_svg pour générer une image au format svg.

<https://www.open-mpi.org>

Documentation de openMPI.