

NUMP2 - Projet d'Informatique
2025 - 2026

Yannick Bornat - yannick.bornat@enseirb-matmeca.fr
Guillaume Bourmaud - guillaume.bourmaud@enseirb-matmeca.fr
Rémi Giraud - remi.giraud@enseirb-matmeca.fr
Daoud Karakolah - daoud.karakolah@enseirb-matmeca.fr

Projet : Représentation de la fougère de Barnsley

1 Introduction

Le projet de cette année consiste à construire une image représentant la fougère de Barnsley, une figure fractale décrite géométriquement. La Figure 1 donne une représentation de l'image attendue/souhaitée/espérée.

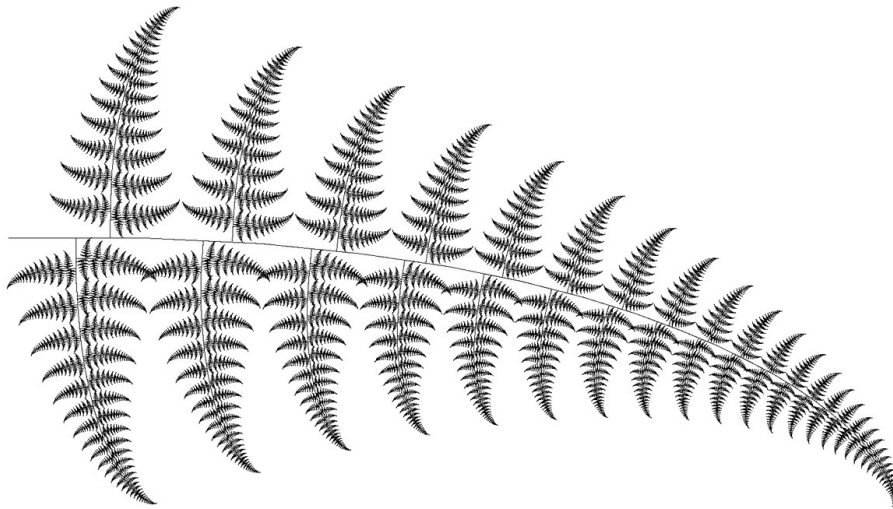


Figure 1: Représentation de la fougère de Barnsley.

1.1 Évaluation

L'évaluation finale aura trois composantes :

- Votre travail durant et entre les séances de TPs
- Votre travail durant les séances de projet
- La qualité technique de votre projet

Aucun rapport ne vous est demandé, mais vous êtes fortement encouragés à commenter votre code, voire à résumer les subtilités que présentent les fonctions créées.

1.2 Consignes générales

Nous sommes dans l'esprit d'un projet, vous avez donc toute liberté quant à la méthode à utiliser pour produire le résultat attendu, moyennant les restrictions suivantes :

- Le résultat que vous présenterez doit être le fruit d'un travail personnel (à toutes fins utiles, précisons que l'intérêt du projet n'est pas tant le résultat final que les compétences que vous aurez mises en œuvre pour y parvenir).
- Gérez votre espace disque : ne pas garder trop de versions haute résolution d'images de test.

Quelques conseils :

- Le calcul d'une image haute résolution nécessitera beaucoup d'itérations, choisissez judicieusement à quelles conditions continuer ou arrêter le calcul pour que le temps de tracé reste faible.
- Pour obtenir une image de qualité, selon la technique que vous aurez choisi, le temps de calcul sera potentiellement long, vous pouvez développer sur des versions en faible résolution. La meilleure technique consiste à définir les paramètres de taille d'image dans un `#define` au début du programme, ce qui permet de les modifier facilement selon le besoin.
- Le résultat demandé est attendu en fin de projet. Il vous appartient de fixer des objectifs intermédiaires pour aider le débogage pendant le développement et continuer sur des bases saines. Partir directement vers le résultat final est la meilleure technique pour ne pas y parvenir.
- Il existe (au moins) un algorithme récursif et un algorithme itératif, vous êtes libre de choisir celui qui vous convient le mieux.
- Aidez-vous du débogueur *GDB* et de l'outil d'analyse mémoire *Valgrind* lors du développement.
- Documentez (commentaires, fichier `readme`, ...) l'utilisation de votre programme.
- Placez vos fonctions et organisez vos fichiers de manière pertinente (`.c`, `.h` correspondants).
- Testez votre programme, pour vous assurer qu'il soit robuste (mauvaises entrées, tailles d'images non équivalentes, etc).
- Vos encadrants AUSSI, savent utiliser une connexion internet et un moteur de recherche...

2 Règles de construction de la fougère

Chaque élément de la figure se construit à partir de (au moins) quatre paramètres :

- la position de départ (coordonnées de la base)
- l'orientation
- la taille approximative (la taille doit plutôt être considérée comme un paramètre d'échelle, elle correspond approximativement à la plus grande dimension de la figure finale, elle est représentée par le segment OH sur la figure 2).

2.1 Approximation

Une approximation très grossière de la figure est proposée sous forme d'un hexagone décrit par la Figure 2 et le Tableau 1. Cette approximation n'est pas nécessaire au tracé de la figure finale, mais elle est particulièrement utile pour les premières phases de développement et pour le débogage.

À terme, votre code devra toutefois laisser le choix de tracer uniquement l'enveloppe hexagonale de la fougère.

2.2 Règle de récurrence

La fougère étant une figure fractale, chaque représentation se retrouve à l'identique dans ses sous-parties. L'élément de base est dessiné à partir de trois copies plus petites ainsi que d'un segment auquel les copies sont reliées. La figure finale (obtenue pour un nombre potentiellement infini d'itérations) n'est techniquement constituée que de ces segments. La Figure 3 et le Tableau 2 indiquent les règles de répétition de la figure. La figure de départ est tracée en gris, les trois sous-figures sont tracées en utilisant les couleurs vert, rouge et bleu. Les angles sont en radians. Les points Og, Or et Ob ne sont pas représentés sur la figure par souci de clarté. Ils sont respectivement situés aux bases des sous-figures verte rouge et bleue. Le segment à tracer est le segment [O Og] représenté en noir sur la figure.

A noter que la sous-figure bleue est inversée par rapport aux autres. Dans un premier temps, et pour simplifier vos premiers tests, vous pouvez considérer que cette inversion n'existe pas.

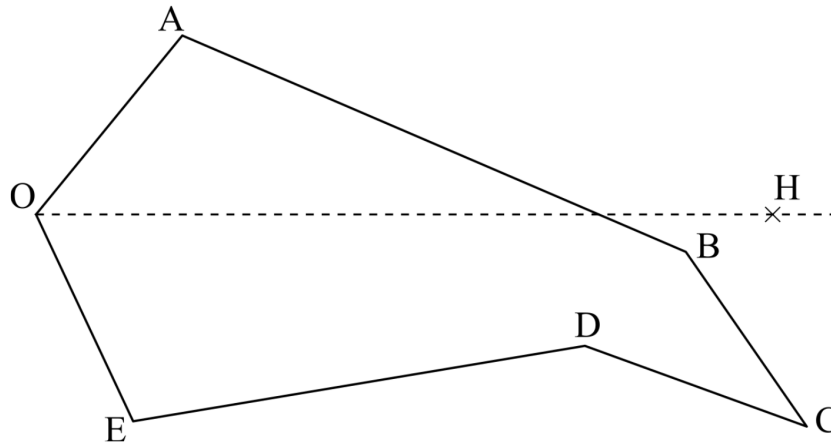


Figure 2: Représentation schématique de la fougère.

Table 1: Données chiffrées pour l'hexagone d'approximation de la fougère

Longueurs	OH =	1 (taille de référence)
	OA / OH =	0.335
	OB / OH =	0.877
	OC / OH =	1.086
	OD / OH =	0.765
	OE / OH =	0.339
Angles (radians)	HOA =	0.9421
	HOB =	-0.0647
	HOC =	-0.3
	HOD =	-0.263
	HOE =	-1.175

2.3 Conditions initiales

Pour le résultat final, votre programme doit créer une image de dimensions 1600 x 900 pixels. La fougère doit être représentée horizontalement : le premier segment de la tige de base part horizontalement, vers la droite, depuis les coordonnées $x = 10$, $y = 424$ (le point de référence est le pixel en haut à gauche). Le facteur d'échelle sera fixé à 1500. Si vous désirez produire une image de taille différente, veillez à respecter les proportions pour éviter qu'une partie du tracé ne sorte de votre image.

3 Pour aller plus loin

3.1 Tracé par liste chaînée

Enregistrez les positions des segments dans un fichier texte selon la même convention que celle utilisée au TP 3. Avec les fonctions de chargement et de gestion des segments sous la forme d'une liste chaînée, effectuez le tracé de la fougère depuis le fichier texte enregistré.

3.2 Options utilisateur

Facilitez l'utilisation du programme en ligne de commande avec des options gérées via `getopts`. Par exemple des options qui permettent de gérer la résolution, le nombre de récurrences, la couleur utilisée ou le nom de l'image de sortie.

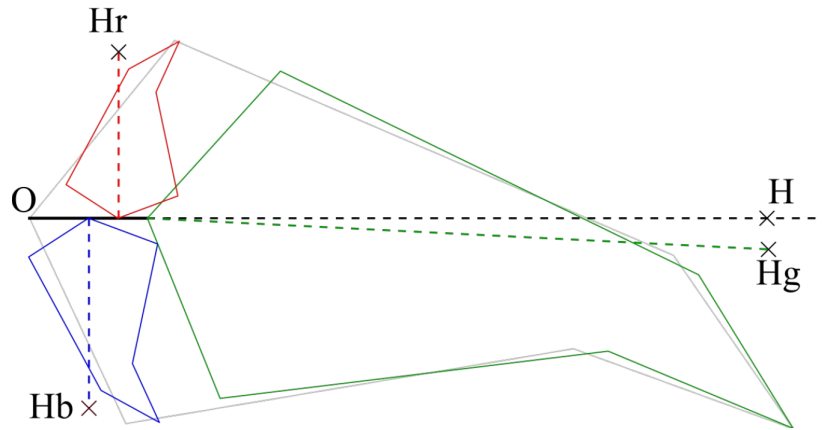


Figure 3: Relations d'héritages entre la figure d'origine (gris) est les sous-composantes (rouge, verte et bleue).

Table 2: Données d'héritage chiffrées pour les sous-figures. Les longueurs sont données par rapport à OH, taille de la figure d'origine

Positions	O Og / OH	0.16
	O Or / OH	0.12
	O Ob / OH	0.08
Angles (radians)	H Og Hg	-0.05
	H Or Hr	$\pi/2$
	H Ob Hb	$-\pi/2$
Tailles	OgHg / OH	0.86
	OrHr / OH	0.27
	ObHb / OH	0.30

3.3 Mesure du temps de calcul

Vous pouvez mesurer le temps de calcul de votre programme pour effectuer le rendu. Il s'agit d'un bon moyen pour tester des optimisations.

3.4 Bonus

Vous êtes libres d'ajouter des fonctionnalités à votre programme, sous réserve de les documenter.

Toute remarque de votre encadrant est prioritaire sur ce sujet