



## Dessin d'une fractale

### Objectifs

- Utilisation d'un module déjà implanté
- Définition des modules `file` et `pile`
- Parcours en profondeur et largeur

Récupérez le fichier `TP9.tar` et dé-archivé le dans votre répertoire de programmation fonctionnelle.

### 1 Compilation d'un module

Pour compiler le module `pile`, il suffit de lancer les commandes suivantes dans un SHELL (*Terminal*)<sup>1</sup> :

```
> ocamlc -c pile.mli
```

```
> ocamlc -c pile.ml
```

La première compilation crée un fichier `pile.cmi`. La deuxième crée le fichier objet `pile.cmo` et vérifie qu'il implémente effectivement les fonctions de l'interface `pile.mli`.

#### 1.1 Interpréteur et utilisation d'un module compilé

Pour pouvoir utiliser les fonctions du module `pile` dans l'interpréteur, il vous faut lancer OCAML en lui rajoutant le fichier objet en paramètre :

```
> ocaml pile.cmo
```

Dans l'interpréteur, vous pouvez accéder aux fonctions en tapant le nom de la fonction préfixé du nom du module, par exemple<sup>2</sup>

```
# let p = Pile.creer_vide ();;
```

ou directement (sans mettre `Pile.`), en tapant la commande :

```
# open Pile;;
```

```
# let p = creer_vide ();;
```

Les fichiers `pile.mli` et `pile.ml` sont fournis.

▷ **Exercice 1** *Compilez les fichiers du module `pile`.*

▷ **Exercice 2** *Testez le module `Pile` à l'aide de l'interpréteur.*

---

1. le caractère `'>'` représente le prompt du SHELL, à ne pas taper

2. le caractère `'#'` représente le prompt de l'interpréteur, à ne pas taper

## 1.2 Exécutable et utilisation d'un module compilé

Pour pouvoir utiliser les fonctions du module `Pile` dans un fichier que vous voulez exécuter, vous devez compiler le fichier comme suit :

```
> ocamlc pile.cmo <nom_fichier>.ml -o <nom_executable>
```

Les fichiers `file.mli` et `testfile.ml` sont fournis.

▷ **Exercice 3** *Implémentez le module `File` (créez et complétez le fichier `file.ml`).*

▷ **Exercice 4** *Compilez le module `File`.*

▷ **Exercice 5** *Compilez le fichier `testfile.ml` et exécutez le programme généré.*

## 2 Dessin du flocon de Von Koch

### 2.1 Le module graphique de OCaml

Un certain nombre de bibliothèques (libraries en anglais) de fonctions sont à disposition du programmeur OCaml (par exemple : *The core library*, *The standard library*, *The unix library*, *The graphics library*...).

Chacune de ces bibliothèques se présente sous la forme de 2 fichiers : un fichier `*.cma` et un fichier `*.cmi` dans le répertoire `/usr/lib/ocaml`.

On peut utiliser les fonctions d'une bibliothèque depuis l'interpréteur. Par exemple pour ce TP, nous allons utiliser les fonctions de la bibliothèque graphique, soit en appelant l'interpréteur avec la commande :

```
> ocaml graphics.cma
```

soit en incluant ce fichier bibliothèque `graphics.cma` dans la commande de compilation :

```
> ocamlc graphics.cma <nom_fichier>.ml -o <nom_executable>
```

### 2.2 La courbe de Von Koch

Une courbe fractale est une courbe auto-similaire qui peut être définie récursivement très naturellement.

Un segment de la **courbe de Von Koch** est la courbe (fractale) plane limite obtenue à partir du processus suivant et d'un segment initial quelconque dans le plan :

- Transformation récursive d'un segment quelconque de la courbe (appelé ici  $[P_1, P_5]$ ) en une suite de 4 segments  $[P_1, P_2]$ ,  $[P_2, P_3]$ ,  $[P_3, P_4]$  et  $[P_4, P_5]$ , avec  $P_2$ ,  $P_3$  et  $P_4$  disposés selon la Figure 1 (où le segment  $[P_1, P_5]$  est représenté horizontalement).

La courbe de Von Koch peut être constituée de segments qui s'appuient sur plusieurs segments initiaux. Si ces segments initiaux sont les côtés d'une figure régulière comme un triangle équilatéral, un carré, on parlera de **flocon de Von Koch**.

Pour créer et afficher un flocon, vous utiliserez les fonctions du module `Dessin`. L'interface est dans `dessin.mli`, et le code compilé (non visible donc) dans `dessin.cmo`.

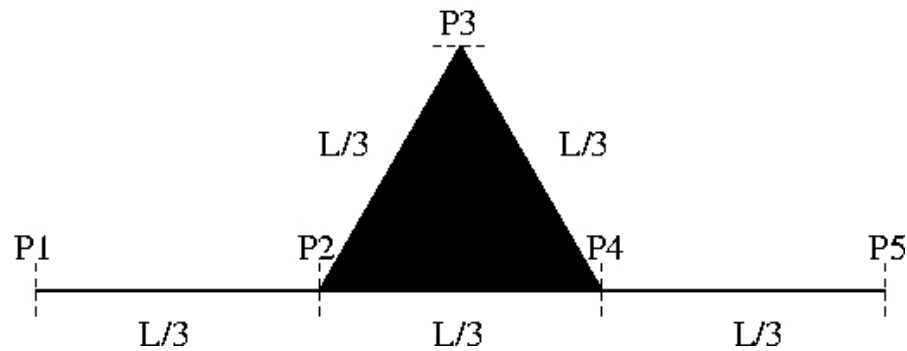


FIGURE 1 – Une étape de la construction de la courbe de Von Koch

## 2.3 Utilisation de la pile d'appel

Dans le fichier `vonkoch1.ml`, nous avons défini une première fonction récursive pure définissant un côté du flocon de Von Koch. Elle rend une liste de segments composant la courbe de Von Koch. un segment dont la longueur est supérieure à un certain seuil est subdivisé en 4 segments, sinon le processus est arrêté (en suivant la définition).

### ▷ Exercice 6

Compilez le fichier `dessin.mli`

```
> ocamlc -c dessin.mli
```

Compilez le fichier `vonkoch1.ml` et générez l'exécutable `vonkoch1` :

```
> ocamlc graphics.cma unix.cma dessin.cmo vonkoch1.ml -o vonkoch1
```

et affichez un côté du flocon de Von Koch :

```
> ./vonkoch1
```

Dans le fichier `vonkoch2.ml`, nous avons défini une seconde fonction récursive terminale dessinant le  $n$ ième niveau d'un tiers de flocon.

### ▷ Exercice 7 Compilez le fichier `vonkoch2.ml` et générez l'exécutable `vonkoch2` :

```
> ocamlc graphics.cma unix.cma dessin.cmo vonkoch2.ml -o vonkoch2
```

et affichez le 3ième niveau d'un côté du flocon de Von Koch<sup>3</sup> :

```
> ./vonkoch2
```

### ▷ Exercice 8 Modifiez les 2 fichiers `vonkoch1.ml` et `vonkoch2.ml` pour dessiner le flocon complet s'appuyant sur un triangle équilatéral.

---

3. vous pouvez jouer avec la valeur du niveau en modifiant cette valeur dans l'appel à la fonction `dessin_fractal`; **attention**, il vous faudra recompiler l'exécutable à chaque modification.

## 2.4 Utilisation explicite des files et des piles

Dans cette deuxième partie, on souhaite dessiner des niveaux successifs de la courbe de Von Koch en choisissant de réaliser un parcours en profondeur ou en largeur. Pour cela, on ne souhaite plus s'appuyer sur la pile d'appel d'OCaml (qui réalise un parcours en profondeur) mais nous voulons stocker dans une structure de données explicite (soit dans une `file`, soit dans une `pile`) les segments intermédiaires à traiter. Ces segments intermédiaires devront également être affichés.

### 2.4.1 File (parcours en largeur)

- ▷ **Exercice 9** *Écrivez une fonction qui construit les  $n$  générations de la courbe de Von Koch en réalisant un parcours en largeur à l'aide d'une file.*

### 2.4.2 Pile (parcours en profondeur)

- ▷ **Exercice 10** *Écrivez une fonction qui construit les  $n$  générations de la courbe de Von Koch en réalisant un parcours en profondeur à l'aide d'une pile.*

## 3 ...d'autres fractales (optionnel)

Si vous avez envie de dessiner autre chose que la courbe de Von Koch, vous avez aussi la courbe de Peano (voir Figure 2) ou la courbe du dragon (les trois premières étapes apparaissent sur la Figure 3).

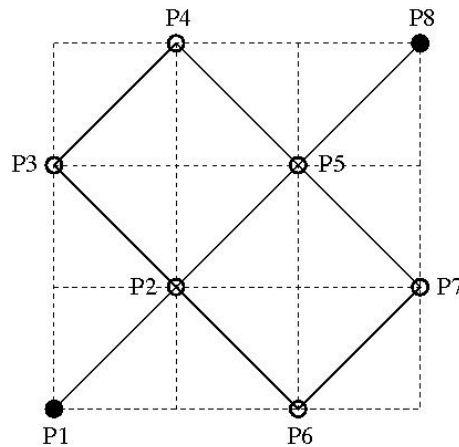


FIGURE 2 – Une étape de la construction de la courbe de Peano : le segment  $[P_1, P_8]$  est remplacé par les segments successifs formés par  $[P_1, P_2, P_3, P_4, P_5, P_2, P_6, P_7, P_5, P_8]$ .

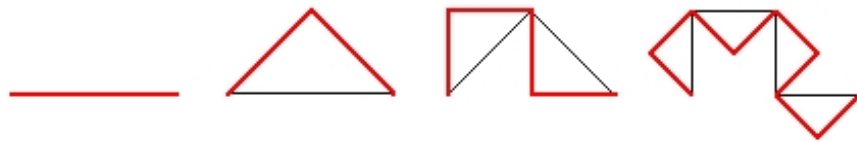


FIGURE 3 – Les trois premiers niveaux d’une courbe du dragon (en rouge). Attention au sens de parcours des segments...