Dessin de fractales

Nicolas Gast - nicolas.gast@ens.fr

6 novembre 2006

1 Introduction

1.1 Rappel : comment faire une procédure

En Maple, ce que nous appelons une procédure est une fonction. Elle s'écrit de la façon suivante :

```
ma_fonction := proc(arg1, arg2, ... , argn)
  # Des lignes de Maple
end;
```

`arg1, arg2, ... argn` sont les arguments de la procédure et le résultat rendu est le dernier résultat calculé.

Exemple de procédure `fac` prenant un argument `n` et rendant la factorielle de `n` :

```
fac := proc(n)
   if n = 0
     then 1
     else n*fac(n-1)
   fi;
end;
fac(10);
                                      3628800
```

1.2 Séquences et listes

En informatique, on est souvent amener à manipuler un grand nombre de variables. Si on dispose des notes des 48 élèves d'une classe, on peut les stocker "bêtement" :

```
note1 := 18; note2 := 5, note3 := 9.5; ... note48 := 20;
```

Mais il existe des manières bien plus pratique en Maple : les listes et les séquences. Une séquence est une suite de chiffres, une liste est en gros une séquence entourée de crochets. On peut accéder au i$^{\text{ième}}$ élément de la liste `note` en tapant `note[i]`. On peut construire des séquences très facilement à l'aide de la commande `seq`. Dans la vie de tous les jours, la différence entre une liste et la séquence est la possibilité de faire des listes de listes.

Exemple :

```
note := 18,5,9.5,20: note[1];
                                        18

liste_des_ds := [ [note], [5,6,1,20], [6,11,8.4,14] ]:
liste_des_ds[2][3];
                                         1

#Utilisation de seq:
nombres := [seq( i^2, i=1..10)];
                       1, 4, 9, 16, 25, 36, 49, 64, 81, 100

#Pour appliquer une fonction à l'ensemble des éléments d'une liste:
traitement_nombre := seq( f(nombres[i]), i=1..nops(nombres));
   traitement_nombre :=
    f(1), f(4), f(9), f(16), f(25), f(36), f(49), f(64), f(81), f(100)

seq( [seq(i,i=1..j)], j=1..5);
               [1], [1, 2], [1, 2, 3], [1, 2, 3, 4], [1, 2, 3, 4, 5]
```

Exercice 1. Mise en jambe

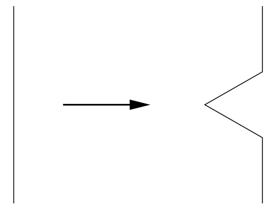
a. Réécrire les différents exemples ci-dessus afin de comprendre le fonctionnement des listes et suites.

b. Construire la listes des carrés des 100 premiers nombres entiers.

c. Construire une liste à partir de la précédante formée des restes de la division euclidienne des carrés des 100 premiers nombres par 7 (cette liste commencera donc par `0,1,4,2,...`).

Exercice 2. le Flocon de Von Koch

Le flocon de Von Koch est une des premières courbes fractales connues. Pour le construire, on part d'un segment que l'on partage en trois parties égales ; puis on construit le triangle équilatéral basé sur la partie centrale. On recommence ensuite la même opération sur les quatre nouveaux sous-segments obtenus, le flocon est la limite à l'infini de cette construction.



Le but de cet exercice est de faire un programme Maple permettant de construire les premières itérations. Afin de simplifier les calculs, nous allons calculer toutes les coordonnées à l'aide de complexes.

Dans la suite de l'énoncé, un point désignera une pair de coordonné `[x,y]`.

a. Écrire une fonction `affixe := proc(p)` prenant en entrée un point en rendant son affixe.

b. Écrire une fonction `pointaffixe := proc(a)` prenant en entrée une liste d'affixe de points et rendant la liste des points correspondants (utiliser la commande `seq` et regardez l'aide de `Re` et `Im`).

c. Écrire une fonction `rotation := proc(a,b,`θ`)` prenant en entrée deux affixes a et b et un angle θ et rendant l'affixe de l'image de b par la rotation de centre a et d'angle θ

d. Écrire une fonction `transfo := proc(a,b)` prenant en entrée deux affixes de points et rendant une suite de 3 affixes correspondant aux trois points supplémentaires créés à une étape de la transformé de Von Koch.

e. Écrire une fonction récursive `flocon := proc( a,b, n)` et rendant une suite de points correspondant à `''n''` transformations de Von Koch.

f. Tester cette fonction sur le triangle formé des trois points $(0, 1, e^{i\frac{\pi}{3}})$

g. BONUS

g1. Calculer (sur le papier) le périmètre du flocon de Von Koch à l'itération n.

g2. Que peut-on dire de cette distance lorsque n tend vers l'infini ?

Exercice 3. Encore une fractale

Le tapis de Sierpiński est une fractale obtenue à partir d'un carré. Le tapis se fabrique en découpant le carré en neuf carrés égaux avec une grille de trois par trois, et en supprimant la pièce centrale, et en appliquant cette procédure indéfiniment aux huit carrés restants.

C'est une généralisation de l'ensemble de Cantor en deux dimensions ; Des généralisations en dimensions supérieures sont possibles.

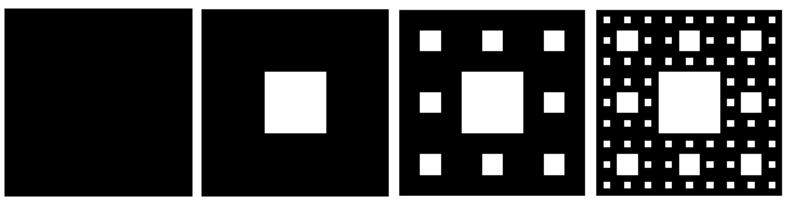


Fig. 1 – Les 4 premières étapes de la construction du tapis

Un carré peut-être défini par un coin, une taille et une orientation. Dans notre cas, tous les carrés ont la même taille et la même orientation, nous assimilerons donc les carrés à le coin inférieur gauche.

Le tapis de Sierpiński à l'étape n sera représenté par une liste de complexes et un nombre représentant la taille des carrés. Si on part d'un carré de coté 1 et dont l'origine est en zéro, les premières étapes sont donc :

1. `( [0], 1)`
2. `( [0,1/3,2/3, i/3,2/3+i/3,2i/3,2i/3+1/3,2i/3+2/3], 1/3)`

a. Faites un programme qui prend comme arguments une liste de carré et affiche le dessin correspondant.

b. Faites un programme qui prend comme arguments un complexe et un réel (*ie* un carré) et qui renvoie le tapis basé sur ce carré après une itération.

c. Faites un programme qui affiche la n$^{\text{ième}}$ étape de la construction du tapis de Sierpiński.