

TP4. TP final

Lancer `xmaple`.

Si vous êtes perdu, utiliser l'aide de maple : `?` suivi du nom de la commande.

Écrivez vos réponses dans les encadrés de cette feuille. Vous pouvez aussi m'envoyer votre fichier maple par mail (francois.chapon@math.univ-toulouse.fr), au format `L3TP2-NOM-PRENOM.mw` si vous mettez des commentaires pour numéroter les questions et les exercices à l'aide de `#`.

On rappelle pour tout le TP que pour évaluer numériquement des opérations sur les nombres (et notamment la racine carrée), on utilise la commande `evalf`.

Exercice 1 : Méthode de Monte-Carlo pour le calcul approché de π .

La méthode de Monte-Carlo est une méthode probabiliste de calcul d'intégrale. Nous allons nous en servir pour approximer le nombre π , vu comme l'aire d'un disque de rayon 1.

Soit un carré d'aire 1 dont le coin inférieur gauche se situe à l'origine du plan. On considère un cercle de rayon 1 centré à l'origine, et donc un quart de cercle est inscrit dans le carré. On tire aléatoirement N points du plan à l'intérieur du carré en question. La probabilité qu'un point appartienne à l'intérieur du quart de disque, donnée par le ratio du nombre de points à l'intérieur du quart de disque sur le nombre de points dans le carré, converge, quand N tend vers l'infini, vers le ratio de l'aire du quart de disque sur l'aire du carré, c'est-à-dire $\pi/4$.

- a) La fonction `rand` de maple permet de tirer des nombres aléatoirement. Elle est définie elle-même comme une procédure. Par défaut, la commande `rand()` renvoie un entier composé de 12 chiffres. Pour tirer aléatoirement un entier entre 1 et 6, on fait alors

```
> alea:=rand(1..6);  
> alea();
```

Tirer un nombre réel aléatoirement entre 0 et 1, composé de 5 chiffres après la virgule.

```
alea:=
```

- b) À quelle condition un point de coordonnées (x, y) est-il situé dans le carré? dans le quart de cercle?

- c) Écrire une procédure prenant en argument le nombre de points tirés et approxinant le nombre π via la méthode de Monte-Carlo.

```
approxPi:=
```

- d) Tester l'approximation de π pour un nombre de points tirés de 10^6 .

Exercice 2 : Des jolies fractales.

Une fractale est une courbe irrégulière qui se crée suivant des règles déterministes ou stochastiques impliquant une homothétie interne. C'est donc un objet dont la structure est invariante par changement d'échelle (le même motif se répète à toutes les échelles). Les fractales s'observent régulièrement dans la nature, pensez par exemple au chou romanesco...

Une procédure récursive est une procédure qui s'appelle elle-même. On va alors se servir de procédures récursives pour construire de jolies fractales.

- a) Un exemple. Écrire une procédure récursive `facto` calculant la factorielle d'un nombre.

- b) La courbe de Koch est une fractale construite récursivement de la façon suivante :

On part d'un segment de droite, on divise le segment de droite en trois segments de longueurs égales, on construit un triangle équilatéral ayant pour base le segment du milieu, puis on supprime le segment qui était la base du triangle équilatéral en question.

On répète alors indéfiniment le même procédé sur chacun des nouveaux segments.

Les deux premières itérations de l'algorithme de Koch donnent par exemple :



Pour tracer des lignes brisées en maple, on utilise la commande `plot` :

```
plot([[x1,y1],[x2,y2],...,[xn,yn]],axes=None,scaling=constrained)
```

trace une ligne brisée composée des segments d'extrémités de coordonnées $(x_1, y_1), \dots, (x_n, y_n)$ (l'option `axes` permet d'enlever les axes, et `scaling` d'avoir un repère orthonormé). Par exemple, tester les commandes suivantes :

```
plot([[0,0],[0,1]],axes=None,scaling=constrained)
plot([[0,0],[1,0],[1,1],[2,1]],axes=None,scaling=constrained)
```

Tracer la première itération de la courbe de Koch en partant d'un segment de longueur 1, et écrire votre commande ci-dessous.

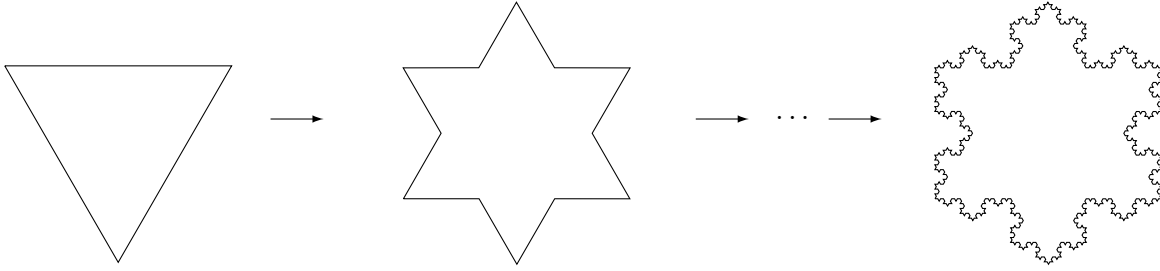
- c) Soient (x_a, y_a) et (x_b, y_b) les coordonnées des extrémités de l'un des segments de la courbe de Koch. Donner les coordonnées des extrémités des 4 nouveaux segments obtenus lors de la prochaine itération de la courbe de Koch appliquée à ce segment. *On pourra remarquer qu'une rotation d'angle θ est donnée par la matrice $\begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}$.*

- d) Écrire une procédure récursive `koch` calculant les coordonnées des extrémités des segments de la courbe de Koch, prenant en argument le nombre d'itérations et les coordonnées des extrémités du segment de départ.

```
koch:=proc(n,xa,ya,xb,yb)
```

- e) Tracer la courbe de Koch pour $n = 1, 2, 7$ itérations.
f) Si la longueur du segment de départ est $l_0 = 1$, quelle est la longueur l_n de la courbe de Koch à l'étape n ? Quand n tend vers l'infini?

- g) Au lieu de partir d'un segment de droite, on peut aussi partir d'un triangle équilatéral et appliquer l'algorithme de Koch à chacun des segments du triangle équilatéral. On obtient ainsi le flocon de Koch :



Le périmètre du flocon de Koch à l'itération n est donc $3l_n$.

Donner une formule de récurrence pour l'aire du flocon de Koch au bout de n itérations, et en déduire l'aire du flocon quand n tend vers l'infini. Que remarque-t-on ?

Exercice 3 : Conjecture de Syracuse.

La suite de Syracuse est une suite d'entiers $(u_n)_{n \geq 0}$ définie de la façon suivante : u_0 est un entier, et pour tout $n \geq 0$,

$$u_{n+1} = \begin{cases} \frac{u_n}{2}, & \text{si } u_n \text{ est pair} \\ 3u_n + 1, & \text{si } u_n \text{ est impair.} \end{cases}$$

- a) Écrire une procédure `syra` qui à un entier N associe $\frac{N}{2}$ si N est pair, et $3N + 1$ si N est impair.

- b) Écrire une procédure `suitesyra` qui à n et u affiche tous les termes de la suite de Syracuse de terme initial $u_0 = u$ jusqu'au terme u_n .

- c) Tester votre procédure pour $u_0 = 4$ et $n = 20$, et pour $u_0 = 15$ et $n = 20$. Que remarque-t-on ?

- d) Écrire une procédure **syracuse** qui prend en argument le terme initial u et qui affiche tous les termes de la suite de Syracuse jusqu'à ce que 1 soit atteint.

- e) Le temps de vol de la suite de Syracuse est défini comme le plus petit indice n tel que $u_n = 1$. Calculer le temps de vol des suites de Syracuse de terme initial 129, et 1207.

- f) A votre avis, l'algorithme converge-t-il ? (C'est-à-dire 1 est-t-il atteint pour n'importe quel terme initial ?) Ceci est en fait un problème non résolu. . .