TP 4 : Récurrence et récursivité

Exercice 1. Formules pour les suites récurrentes

1. Maple est capable de trouver des formules exactes pour certaines suites définies par récurrence. On définit une suite récurrente ainsi : `rec1 := u(n+1)=a*u(n)+b` est la représentation Maple de la suite définie par $u_{n+1} = au_n + b$. Définissez de même la récurrence 2 : $u_{n+2} = 3u_n - u_{n+1}$, la récurrence 3 : $u_{n+3} = -u_{n+2} + 7u_{n+1} + 2u_n + n^2$, la récurrence 4 : $u_{n+1} = \frac{n^2-3n+1}{2-n^2} u_n$ et la récurrence 5 : $u_{n+2} = \frac{1}{n^2+3n+2}\big[3(n+1)\,u_{n+1} + (n^2+2n-8)\,u_n + n\big]$.
2. On résout ensuite une récurrence ainsi : `sol1 := rsolve(rec1, u(n))`. La solution donnée par Maple est-elle valable pour toutes valeurs de a, b ?
3. Résolvez la récurrence 2. En utilisant la commande `eval`, calculez une solution vérifiant $u_0 = u_1 = 1$.
4. Résolvez la récurrence 3, commentez la forme de la solution. On peut forcer Maple à donner une formule explicite avec la commande `convert(`*expression*`, radical)`.
5. Résolvez la récurrence 4. La fonction Γ qui apparait est une fonction qui vérifie $\Gamma(x+1) = x\Gamma(x)$ et $\Gamma(1) = 1$. Que vaut alors $\Gamma(n+1)$ pour n entier ? D'où viennent les nombres α qui apparaissent dans les termes $\Gamma(n-\alpha)$ de la solution ?
6. Résolvez la récurrence 5, commentez la forme de la solution.

Exercice 2. Programmation de suites récurrentes

1. Pour programmer une suite récurrente, le plus naturel est d'utiliser une procédure *récursive*, c'est-à-dire qui s'appelle elle-même. Par exemple, essayez la procédure :
```
s := proc(n)
if n=0 then
 return 0
else
 return 2*n-1+s(n-1)
end if
end;
```
Décrivez ce qui se passe lorsqu'on appelle `s(2)`. Que se passe-t-il si on appelle `s(-1)` ? et `s(1/2)` ?
2. On rappelle que la suite de Fibonacci est la suite (F_n) définie par $F_0 = F_1 = 1$ et $F_{n+2} = F_{n+1} + F_n$. Écrivez une procédure récursive `fibo` qui prend en entrée un entier n et envoie F_n en sortie. Calculez toutes les valeurs de F_n pour n entre 0 et 10. Calculez F_{20}, F_{30}. Pourrez-vous calculer F_{100} ? Pourquoi ?
3. Pour améliorer cela, on peut s'arranger pour que la procédure ne s'appelle qu'une seule fois. Écrivez une procédure `fibo2` qui prend en entrée un entier n et qui renvoie la liste des deux valeurs valeur $[F_n, F_{n+1}]$. Calculez les valeurs de F_n pour n entre 1 et 10, comparez à celles obtenues précédemment. Calculez les valeurs approchées de F_{2^m} pour m entre 0 et 15. Pourrez-vous calculer $F_{2^{30}}$? Pourquoi ?

4. Une autre solution est de demander à Maple de retenir tous les résultats déjà calculés de la procédure, afin de ne jamais les calculer plusieurs fois. Pour cela, ajoutez la commande « `option remember:` » au début du corps de la procédure `fibo`. Testez à nouveau.
5. Dans ce cas particulier, on a beaucoup de chance car on peut même obtenir une formule pour F_n. Résolvez la récurrence de la suite de Fibonacci à l'aide de Maple, puis écrivez une procédure `fibo3` qui prend en entrée un entier n et renvoie F_n en utilisant la formule. Vérifiez les valeurs F_n déjà calculées, ainsi que les valeurs approchées de F_{2^m}. Calculez des valeurs approchées de $F_{2^{20}}$ et $F_{2^{30}}$.

Exercice 3. Programmation récursive

La programmation récursive consiste à utiliser comme précédemment une procédure récursive pour résoudre un problème. Pour cela, il faut exprimer une solution de ce problème en fonction de solutions plus simples de ce même problème.

1. Le but est de calculer x^n récursivement, en utilisant seulement des multiplications. Exprimer x^n en fonction de x et d'une puissance plus petite de x. Écrivez une procédure `puissance` qui prend en entrée x et n et renvoie x^n. Calculez les valeurs de 1.0000001^n pour n entre 0 et 20, puis $n = 10^m$ pour m entre 1 et 4. Pourrez-vous calculer la valeur pour $n = 10^6$?
2. Nous allons voir une version plus rapide. Pour cela, exprimez x^n comme une puissance de x^2 lorsque n est pair, et en fonction de x et d'une puissance de x^2 lorsque n est impair. Utilisez cela pour écrire une nouvelle procédure `puissance2` qui prend en entrée x et n et renvoie x^n. Testez-la.
3. Combien de multiplications la première procédure utilise-t-elle pour calculer x^n ? Soit M_n le nombre de multiplications utilisées par la seconde procédure pour calculer x^n. Montrez qu'on a l'inégalité $M_n \leq 2 + M_{[\frac{n}{2}]}$, en déduire que $M_n = O(\ln n)$.
4. On rappelle que si $a > b$ sont deux entiers tels que $b \neq 0$ et que r est le reste dans la division euclidienne de a par b, alors le PGCD de a et b est le même que celui de b et r. Écrivez une procédure récursive `pgcd` qui prend en entrée deux entiers a, b et renvoie leur PGCD. Comparez les valeurs obtenues avec la fonction Maple `igcd`.
5. Pour $x \in \mathbb{R}$, on peut écrire $x = [x] + \{x\}$ où $a_0 = [x] \in \mathbb{Z}$ est la partie entière et $\{x\} \in [0, 1[$ est la partie fractionnaire. Si $\{x\} \neq 0$ on peut écrire $\{x\} = \frac{1}{y}$ avec $y \notin [0, 1[$, on obtient $x = a_0 + \frac{1}{a_1 + \frac{1}{z}}$ avec $y = a_1 + \{y\}$, et on peut appliquer à nouveau le procédé. Le réel x est rationnel si et seulement si le procédé se termine (une partie fractionnaire vaut 0). Sinon, on obtient une suite d'entiers a_n, et l'écriture

$$x = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \dots}}}$$

est appelée la *fraction continue* associée à x (on peut montrer que la suite de fractions obtenue converge vers x).

   a. Écrivez une procédure `FractionContinue` qui prend en entrée un réel x et un entier n, et qui renvoie la séquence des termes $a_0, \dots, a_n$ (ou seulement ceux qui sont définis si x est rationnel).
   b. Calculez 20 termes de la fraction continue de π. Calculez 20 termes de la fraction continue de $\sqrt{7}$ et de celle de $\sqrt{13}$. Remarque ? Et si on remplace par des racines cubiques ?
   c. Définissez un nombre rationnel (au hasard), quotient de deux grands nombres entiers (une dizaine de chiffres). Calculez les 40 premiers termes de sa fraction continue, puis les 40 premiers termes de la fraction continue d'une valeur approchée de ce nombre. Commentez.
   d. Modifiez votre procédure `pgcd` pour qu'elle affiche, lorsque $b \neq 0$, le quotient de a par b. Calculez le PGCD de deux nombres entiers a, b, puis la fraction continue de $\frac{a}{b}$. Commentez.