“Calcul formel” Maple
L1/MK1 2007-2008

Pour le lundi
10 décembre

Devoir Maison Groupe C3

Groupe A : les exercices 1, 2, 3, 4 sont obligatoires, le 5 est facultatif.
Groupe B : les exercices 3, 4, 5 sont obligatoires, les 1 et 2 sont facultatifs.
Les questions précédées d'une étoile (*) sont à faire sur feuille. Tout le reste est à faire sur machine. On rendra une copie double (avec le nom) les réponses aux questions (*), ainsi que les impressions des feuilles de travail Maple (le nom sur chaque feuille). La date limite de retour est **le 10 décembre**. Tout DM rendu en retard se verra attribuer la note 0 (et ne sera pas corrigé). Si vous avez des questions, n'hésitez pas à me demander par mail ou directement en TP.
Le temps passe toujours trop vite, et le DM peut paraître long. **Ne vous y prenez pas au dernier moment !** Vous avez accès à Maple dans les salles libre accès du SCRIPT. Pensez à consulter les horaires suffisamment à l'avance !

Une attention particulière sera apportée à la syntaxe de Maple et à la rédaction.

Exercice 1 (Gr. A)
Soit f la fonction définie par:

$$\forall x \in \mathbb{R},\ f(x) = \frac{x^3+6x}{3x^2+2}$$

1. Définir la fonction f dans Maple.
2. Vérifier (avec Maple) que f est continue sur son ensemble de définition.
3. Tracer sur un même graphe la fonction f et la fonction $x \mapsto x$ (la première en vert, la seconde en rouge), sur l'intervalle (en abscisse) $[-3; 3]$.
4. Trouver avec Maple l(es) abscisse(s) du(des) point(s) d'intersection des deux courbes, en utilisant la commande de résolution d'équation.
5. Soit u la suite récurrente définie par:

$$\begin{cases} u_1 = -1/57 & ; \\ u_n = f(u_{n-1}) & \text{si } n \geq 2 \end{cases}$$

   En utilisant la fonction f, définir la suite u dans une procédure qui prend en entrée n. Cette procédure devra retourner un message d'erreur si $n \leq 0$ (et la valeur exacte de u_n sinon).
6. Afficher la séquence des valeurs approchées des 10 premières valeurs de la suite. (*) Que constatez-vous?
7. Donner une valeur approchée de $u_{10} - (-\sqrt{2})$ à 10^{-213} près.
8. À l'aide d'une boucle *while*, trouver le plus petit entier m tel que $|u_m + \sqrt{2}| \leq 10^{-1000}$.

Exercice 2 (Gr. A)
Écrire une procédure *transformer* qui prend en entrée une matrice M et qui renvoie la matrice M à laquelle on a :

- divisé par 7 tous les coefficients de la colonne du milieu, si le nombre de colonnes de M est impair ;
- mis au carré tous les coefficients de la dernière ligne, sinon.

Utiliser l'aide si besoin (impair = *odd* en anglais). Tester votre procédure sur des exemples bien choisis.

Exercice 3 (Une conjecture. (Gr. A et B))

I) Différences successives

1. Pour cet exercice vous aurez besoin d'utiliser un nombre particulier, votre **prénombre**. Pour le calculer, transformez en nombre chaque lettre de votre prénom (selon leur rang dans l'alphabet), et ajoutez tous ces nombres. Exemple : Vivien $\mapsto 22+9+22+9+5+14=81$.

   Dans toute la suite on notera a le prénombre ; c'est le nombre que vous venez de calculer qu'il faudra alors utiliser.

2. Écrire une procédure (ou une fonction) appelée *differences* qui prend en entrée une liste L de réels, et qui retourne la liste M des différences successives des nombres de la liste L. Exemple : $L = [2\ 7\ 10\ 23\ 27]$ doit donner $M = [5\ 3\ 13\ 4]$. Le k-ième élément de M doit être la différence entre les $(k+1)$-ième et k-ième éléments de L.

3. Écrire une procédure *itere* qui prend en entrée une liste L et un entier naturel n, et qui retourne le n-ième itéré de la procédure *differences* appliquée à L. Par exemple, si L est comme dans la question précédente, on a $\mathit{itere}(L,0) = L$, $\mathit{itere}(L,1) = \mathit{differences}(L) = M$, et $\mathit{itere}(L,2) = \mathit{differences}(M) = [-2\ 10\ \ -9]$. On pourra utiliser une procédure récursive, ou une boucle *for*.

   Tester cette procédure sur des exemples simples.

4. Calculer *itere(L,n)* dans les cas suivants (rappelons que a est votre prénombre) :

   - L est la liste des carrés des entiers de a à $a+30$, et $n=2$;
   - L est la liste des cubes des entiers de a à $a+30$, et $n=3$;
   - L est la liste des puissances quatrièmes des entiers de a à $a+30$, et $n=4$;
   - L est la liste des puissances dixièmes des entiers de a à $a+30$, et $n=10$.

5. (*) Que se passe-t-il ? Formulez une conjecture précise sur la situation. (indication : $3628800 = 10!$). Vous pouvez tester d'autres exemples du même type pour vous faire un idée.

II) Polynômes et dérivée discrète

Rappels (vocabulaire sur les polynômes). Considérons un polynôme (en une variable X) P. Par exemple, $P = 3X^7 - 2X^4 + 5X - 3$. Son **degré** est 7, son **coefficient dominant** est 3, son **terme dominant** est $3X^7$.

Dans Maple, on définira les polynômes comme des expressions (pas des fonctions !) en X (on utilisera toujours la même variable X).

1. Écrire une procédure (ou une fonction) *Delta* qui prend en entrée un polynôme P et renvoie le polynôme $P(X+1) - P(X)$. [Par exemple, $\mathit{Delta}(X^2)$ doit donner $(X+1)^2 - X^2$ ie $2X+1$]. On pourra utiliser la commande *subs*, et demander dans la procédure de simplifier le résultat obtenu (*simplify* ou *expand*).

   Par la suite on notera $\Delta(P)$ ou ΔP ce polynôme (appelé dérivée discrète de P : (*) pourquoi?)

2. Écrire une procédure *Deltaitere* qui prend en entrée un polynôme P et un entier naturel n et retourne le n-ième itéré de *Delta* appliqué à P, que l'on notera par la suite $\Delta^n P$. On doit avoir : $\Delta^0 P = P$, $\Delta^1 P = \Delta P$, $\Delta^2 P = \Delta(\Delta(P))$, etc... On pourra utiliser une procédure récursive, ou une boucle *for*.

3. Calculer $\Delta^n P$ dans les cas suivants :

   - $P = X^2$, et $n = 2$;

- $P = X^3$, et $n = 3$;
- $P = X^4$, et $n = 4$;
- $P = X^{10}$, et $n = 10$.

(*) Que remarquez-vous ? Conjecture ? Lien avec la partie précédente ?

Nous allons tenter de démontrer cette conjecture. Les questions suivantes sont exclusivement mathématiques, mais vous pouvez vous aider de Maple sur des exemples particuliers.

4. (*)Rappeler la formule du binôme de Newton (développement de $(x+y)^n$).

5. (*)L'utiliser pour montrer que si $P = X^n$ (n quelconque), le terme dominant de ΔP est nX^{n-1}.

6. (*)De façon plus générale, montrer que si P est un polynôme de degré n et de coefficient dominant c, alors ΔP est un polynôme de degré $n-1$, de coefficient dominant $n \cdot c$.

7. (*) Par récurrence sur n, montrer que si P est un polynôme de degré n et de coefficient dominant c, alors $\Delta^n P$ est une constante (ie un polynôme de degré 0), qui vaut $n! \cdot c$. Conclure sur l'exercice.

Exercice 4 (Suites récurrentes. (Gr. A et B))

On va s'intéresser aux suites de la forme $u_{n+1} = f(u_n)$. Dans cet exercice on notera un point du plan sous la forme de la liste à deux éléments [abscisse, ordonnée].
Les 3 premières questions sont là pour aider à faire la suite.

1. Regarder ce qu'affiche Maple avec la commande:

   plot([[1,2],[3,4],[4,4],[4,3]], color=magenta);

2. Aller voir l'aide sur la commande *display* de la bibliothèque *plots*. L'utiliser pour afficher sur un même graphe la structure précédente et la courbe de $\sqrt{x}$ entre 0 et 5, l'une en magenta, l'autre en bleu.

3. Rappeler comment on ajoute un élément x à une liste L (en tant que dernier élément).

4. Écrire une procédure *liste*, qui prend trois arguments en entrée: une condition initiale u_0, une fonction f, et le nombre d'itérations n, et qui affiche en sortie la liste des points du plan suivants: $[u_0, 0], [u_0, u_1], [u_1, u_1], [u_1, u_2], [u_2, u_2], [u_2, u_3], \ldots, [u_{n-1}, u_n], [u_n, u_n]$.

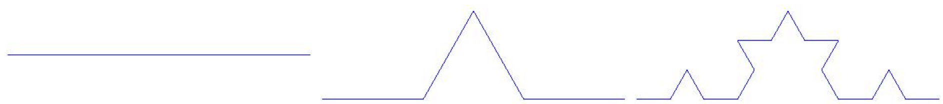
   On pourra utiliser une boucle *for* et la question précédente.

5. Écrire une procédure *spirale*, qui prend les mêmes entrée que ci-dessus, et qui affiche en sortie sur un même graphe, la fonction f, la fonction $x \to x$, et la ligne polygonale qui joint les points de la liste ci-dessus.

   On pourra se servir de la procédure *liste* et de la question 2.

6. Tester la fonction avec $f(x) = cos(x), u_0 = 0.3, n = 10$ puis avec $f(x) = \sqrt{x+2}, u_0 = 0.5, n = 6$, et enfin avec $f(x) = \frac{1}{2}(x + \frac{2}{x}), u_0 = 2, n = 4$. Déterminer la limite éventuelle de la suite dans le dernier cas.

7. Ajouter à la fonction *spirale*, une manière d'obtenir une taille de fenêtre automatiquement adaptée, en utilisant les fonctions *max* et *min*.



(a) Segment de départ (b) Première étape (c) Deuxième étape

Figure 1: L'évolution du flocon de Von Koch

Exercice 5 (Le flocon de Von Koch. (Gr. B))

Il s'agit d'une des premières courbes fractales découvertes. On la construit par itération: on part d'un segment que l'on partage en 3 parties égales, puis l'on remplace la partie centrale par deux cotés du triangle équilatéral correspondant: voir figure ci-dessus.
N.B. : dans cet exercice on notera un point du plan (ou "point cartésien") sous la forme de la liste à deux éléments [abscisse, ordonnée].

1. Créer une fonction *cart2complex*, qui prend en entrée un point M du plan (ie une liste à 2 éléments) et qui renvoie le nombre complexe correspondant, ie l'affixe de M.

2. Créer une fonction *complex2cart*, qui prend en entrée un nombre complexe et qui renvoie le point correspondant.

3. Créer une fonction *rotation*, qui prend en argument des nombres complexes a, b, et un angle (en radians!) θ et qui renvoie l'image de a par la rotation d'angle θ centrée en b. Conseil: faire un dessin!

4. Créer une fonction *transformation*, qui prend en entrée une liste de deux points cartésiens A et B et qui renvoie la **séquence** des 4 segments obtenus de la même manière que l'on passe de la figure (a) à la figure (b).

   Remarque: On notera un segment sous forme de liste de deux points cartésiens.

5. Créer une fonction qui prend en entrée une liste de points cartésiens, et qui renvoit la n-ième itération de la courbe de Von Koch. On utilisera les fonctions *map*, et *transformation*. Indication : créer une liste de segments à partir de la liste des points donnée, et appliquer *transformation* à chacun des segments.

   Tester avec 1,2,3,4 itérations quand on part avec $[0,0],[0,1]$; puis $[1,0],[\frac{-1}{2},\frac{\sqrt{3}}{2}],[\frac{-1}{2},\frac{-\sqrt{3}}{2}],[1,0]$; puis $[1,0],[\frac{-1}{2},\frac{-\sqrt{3}}{2}],[\frac{-1}{2},\frac{\sqrt{3}}{2}],[1,0]$.

6. Fractales aléatoires: modifier la fonction *transformation*, de manière à ce que le sens de rotation soit cette fois aléatoire (triangle en haut ou triangle en bas). On pourra d'abord construire un moyen de tirer à pile ou face dans $\{-1,1\}$, en utilisant la fonction *rand*. Utiliser ensuite cette fonction pour créer une fractale de Von Koch aléatoire.