TD 1 : Etude de suites

Ce premier TD se présente sous la forme d'une succession de petits exercices qui présentent chacun un aspect particulier de l'étude, avec Maple, d'une suite.
Entre parenthèses, en `Courier`, à la fin des questions, on trouvera des commandes Maple utiles pour répondre à la question.

Ex 1 : Outils élémentaires

On étudie une suite simple quelconque ($u_n = \frac{1}{n^2}$ par exemple).

1. Définir une fonction $u(n)$ permettant de calculer le n-ième terme de la suite (`->`).
2. Calculer ses premiers termes (`seq`).
3. Représenter graphiquement les premières valeurs de la suite (`seq`, `plot`).
4. Regarder si la suite est monotone, bornée (`assume`, `is`, `simplify`).
5. Calculer la limite de la suite, si elle existe (`limit`).

Ex 2 : Convergence d'une série

A. Calcul du logarithme népérien de 2

1. Calculez des valeurs approchées de $\ln(2)$ en utilisant le développement en série entière de $\ln(1+x)$ (`evalf`, `add`, `seq`).
2. Comparez avec la valeur fournie par Maple (`evalf`).

B. Idem, plus rapide

1. Idem, en utilisant le développement en série entière de $\ln\left(\frac{1+x}{1-x}\right)$.
2. Cette méthode est-elle plus efficace ? Pourquoi ?

Ex 3 : Récurrence linéaire

On étudie une récurrence linéaire simple ($u_n = 10 \cdot u_{n-1} - 16 \cdot u_{n-2}$ avec $u_1 = 20$ et $u_2 = -8$ par exemple).

1. Définir une fonction récursive permettant de calculer (si possible efficacement) le n-ième terme de la suite (`proc`, `option remember`).
2. Calculer ses premiers termes.
3. Résoudre la récurrence (`rsolve`).
4. Définir une fonction non récursive permettant de calculer le n-ième terme de la suite.
5. Déterminer la limite de la suite ainsi qu'un équivalent (`asympt`).

Ex 4 : Récurrences du type $u_{n+1} = \mathrm{f}(u_n)$

Exemple 1

On étudie la suite définie par : $u_{n+1} = \sqrt{3 \cdot u_n - 2}$ et $u_0 = \frac{3}{2}$.

1. Définir la fonction f qui à u_n associe u_{n+1}.
2. Définir une fonction permettant de calculer le n-ième terme de la suite.
3. Calculer les premiers termes de la suite et tracer sur un même graphe la courbe de f, celle de l'application identité, et l'escalier représentant les premiers termes de la suite (`plots[display]`).

On pourra s'aider de la fonction :
`marche := i -> ([u(i),u(i)],[u(i),u(i+1)])`

4. Déterminer les points attractifs et répulsifs de f (`solve, D` ou `diff`).
5. Reconstruire l'escalier pour $u_0 = 0{,}95$.

Exemple 2 (Plus long)

Idem avec la suite $u_{n+1} = R \cdot u_n \cdot \left(1 - u_n\right)$:

1. Expérimenter diverses valeurs de *R* (1.5, 2.6, 3.25, ...). Que remarque-t-on ?
2. Déterminer les points attractifs/répulsifs de la fonction qui a u_n fait correspondre u_{n+1} (`solve, D` ou `diff`). Quelle valeur particulière de *R* apparaît ?
3. Que se passe-t-il pour *R*>3 ? *R*>3,5 ? et *R*>4 ? Peut-on déterminer les valeurs de *R* pour lesquelles le comportement de la suite change ?

Exemple 3 (Plus difficile)

On étudiera la suite $u_{n+1} = \frac{1}{2} + \frac{1}{2} {u_n}^2$.

(limites possibles, monotonie, convergence en fonction de u_0, etc.)

Ex 5 : Récurrence à solution polynômiale

On étudie la suite définie par $u_{n+2} = \frac{5}{n} u_{n+1} + \frac{n+1}{n} u_n$.

Chercher une solution sous forme d'un polynôme en *n* de degré 3.
Méthode :

1. Commencer par exprimer la définition de la suite sous forme d'une expression devant être nulle pour tout n.
2. Créer un polynôme dont les coefficients sont *a0* .. *a3* (`add`, `.`).
3. Le transformer en une fonction *p* (`unapply`).
4. Remplacer *u* par *p* dans l'équation initiale (`subs`).
5. Exprimer le résultat obtenu sous forme de polynôme en *n* (`collect`).
6. En extraire les coefficients (`coeffs`).
7. Résoudre (tous les coefficients doivent être nuls) (`solve`).
8. Substituer les solutions dans le polynôme initial.
9. Vérifier que la solution obtenue convient.

Ex 6 : Suite de fonctions

On étudie la suite de fonctions : $f_n(x) = \frac{n \cdot \left(x^3 + x\right) \cdot e^{-x}}{n \cdot x + 1}$.

1. Etudier la convergence simple sur [0,1].
2. Tracer les premiers termes de la suite et sa limite simple.
3. Etudier la convergence uniforme sur [*a*,1] où 0<*a*<1 (`maximize`, `minimize`).
4. Puis la non convergence uniforme sur [0,1] (et]0,1]).