TD 4 : Séries numériques, Algorithmes de tri

1 Objectifs

- Utilisation de la commande sum pour l'évaluation symbolique de séries.
- Algorithmique élémentaire.
- Familiarisation avec l'algèbre linéaire sous Maple.
- Représentation avec plot de graphiques discrets.

Nous verrons que Maple est *très efficace* pour manipuler les séries numériques, ce qui en fait un outil de choix pour obtenir si ce n'est toujours leur solution, au moins une intuition de la façon dont il faut aborder certains problèmes. Encore une fois, il est important de systématiquement essayer de *prévoir* ce qu'affiche le logiciel, afin de comprendre les phénomènes mathématiques sous-jacents et de mieux apprécier l'apport de l'ordinateur qui, loin de se substituer au raisonnement, doit s'y soumettre. La dernière partie, quant à elle, poursuit l'introduction à l'algorithmique entamée dans les TD précédents.

2 Autour de la fonction ζ de Riemann

1. Calculer avec Maple :
$$\sum_{k=1}^{\infty} \frac{1}{k^2}$$
2. On définit la fonction : $\zeta : z \to \sum_{k=1}^{\infty} \frac{1}{k^z}$. Quel est son domaine de définition ? Tracer au brouillon son allure, puis la représenter graphiquement.
3. (ENSAM 98) Pour $0 \leqslant z \leqslant 1$, la série $\sum_{k=1}^{\infty} \frac{1}{k^z}$ est divergente. Ecrire une fonction nombre qui, pour tout A réel et tout z réel, renvoie le résultat suivant :
   - si z vérifie $0 \leqslant z \leqslant 1$, le plus petit entier N tel que : $\sum_{k=1}^{N} \frac{1}{k^z} > A$,
   - sinon, un message d'erreur.

   Représenter graphiquement sur un même graphique les fonctions $z -> nombre(A, z)$, avec z variant entre 0 et 1 pour A prenant les valeurs 2, 3, 4 et 5.
4. Quel est le comportement asymptotique de la suite définie par $r_n = \sum_{k=1}^{n-1} \frac{1}{k} - ln(n)$? Représenter graphiquement $(r_n)_{n\in\mathbb{N}}$, puis en trouver un développement asymptotique.

3 Différentes formes de séries

1. (oral de l'X) Trouver a et b vérifiant :
$$\sum_{k=1}^{n} \ln(k+n) = n \ln n + an + b + o(1)$$
2. Discuter selon la valeur de $z \in \mathbb{C}$ la valeur de $\sum_{k=0}^{\infty} z^k$, puis de $\sum_{k=0}^{\infty} (-1)^k z^{2k}$. Représenter graphiquement en trois dimensions le module de cette dernière somme en fonction de $z = x + iy$.
3. Soit P la matrice $\begin{pmatrix} 1 & 1 & 0 \\ 1 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$, Di la matrice diagonale de coefficients 1, 2, 3 et $A = PDP^{-1}$.
   Calculer la somme $\sum_{k=0}^{n} \frac{A^k}{k!}$. Que remarque-t-on quand N tend vers l'infini ? Maple a une fonction exponential pour les matrices : à quoi correspond-elle ? Comparer $\exp(A)$ et $P\exp(Di)P^{-1}$: expliquer.

4. On considère la suite $(u_n)_{n\in\mathbb{N}}$ définie par $u_0 \in [0, 2\pi]$ et par $u_{n+1} = sin(u_n)$. Quel est son comportement ?

   Tracer simultanément u_n et $\sqrt{\frac{3}{n}}$. Que peut-on conjecturer ? Essayer de le montrer.

4 Algorithmique

Le but de cette partie est d'écrire deux procédure *tri_insertion* et *tri_fusion* prenant en argument une liste et retournant une liste contenant les mêmes éléments (avec la même multiplicité) dans l'ordre croissant : on appelle cela une procédure de *tri*, et il s'agit d'une des premières notions classiquement étudiées en informatique. On appelle *algorithme de tri* toute méthode permettant d'y parvenir, et il en existe de nombreux plus ou moins efficaces. Pour cette raison, on s'intéressera particulièrement à la complexité de ces procédures, c'est à dire grossièrement à la relation entre la taille d'une liste et le temps qu'elles mettent à la trier.

1. Ecrire une procédure *maxi* qui prend en argument un liste et qui retourne le maximum de cette liste ainsi que la liste privée de ce maximum. En déduire un premier algorithme et une procédure *tri_insertion*. Combien de comparaisons *tri_insertion* effectue-t-elle au maximum pour trier une liste de taille N ?
2. Le second prodédé provient du paradigme "diviser pour régner". On commence par supposer que l'on sait trier toute liste de taille à peu près $n/2$. On en déduit une méthode pour trier les listes de taille n.
   (a) Ecrire une procédure *fusion* prenant en argument deux listes supposées triées et retournant une liste triée contenant les éléments de ces deux listes.
   (b) En déduire la *tri_fusion*.
   (c) Combien de comparaisons *tri_fusion* effectue-t-elle dans le pire des cas pour trier une liste de taille N ?