Travaux Pratiques n°6
Algorithmique

A. MARNAT
antoine.marnat@ens-cachan.org

MPSI.2 Lycée Champollion 2011/12

Exercice 1. Après avoir ouvert sa session et ouvert la présente feuille de TP située sur le réseau, lancer Maple et enregistrer la feuille de calcul (vierge) dans son répertoire personnel avec un titre explicite. Sortir feuilles de papier et stylos. Taper `restart:`.

L'objectif de ce TP est de se familiariser à l'écriture de procédures.

1 Premiers exemples

Exercice 2. Écrire une procédure `mult` qui prend en argument deux nombres et calcule leur produit.

Exercice 3. Écrire une procédure `PGCD` qui prend en argument deux entiers relatifs et retourne leur plus grand commun diviseur en utilisant l'algorithme d'Euclide.

Exercice 4. Écrire, en utilisant la relation du triangle de Pascal, un algorithme qui renvoie un renvoie un tableau de dimension $N \times N$ dont le coefficient (p, q) est le coefficient binomial.

Remarque : on pourra vérifier la correction de la procédure avec `binomial`.

Exercice 5. Écrire une procédure qui prend en argument un réel x et un entier n et calcule naïvement x^n. Écrire ensuite une procédure qui prend les mêmes arguments et calcule le résultat en utilisant une méthode dite d'exponentiation rapide. Comparer les deux procédures à l'aide de la fonction `time`.

Méthode d'exponentiation rapide : on écrit $n = \sum a_i 2^i$, *et on calcule récursivement* $x^n = x^{a_0}(x^2)^{a_1} \dots (x^{2^d})^{a_d}$

Exercice 6. Écrire une procédure qui prend en argument un indice n, les termes initiaux u_0 et u_1 et des coefficients a et b, et qui renvoie le n^e terme de la suite définie par $u_0, u_1, u_{n+2} = au_{n+1} + bu_n$.

Exercice 7. La suite de Syracuse, présentation d'une conjecture

Soit n_0 un entier strictement positif. On forme la suite n_k définie par :

$$n_{k+1} = \begin{cases} 3n_k + 1 & \text{si } n_k \text{ impair} \\ n_k/2 & \text{si } n_k \text{ pair} \end{cases}$$

- Construire une procédure qui forme les 100 premiers termes la suite de Syracuse d'un nombre n_0 donné en argument.
- Tester la procédure sur plusieurs premiers termes, que remarquez vous ?
- Il a été conjecturé que la suite de Syracuse converge pour tout premier terme vers le cycle $4 - 2 - 1$. Le résultat a été montré empiriquement pour tout entier inférieur à 2^{62}. Construire une procédure qui forme la suite de Syracuse d'un nombre n_0 donné en argument. La suite se terminera par la valeur 1.
- Tracer sur deux représentations graphiques différentes les trajectoires de $n_0 = 26$ et $n_0 = 27$ en traçant une suite de segments reliant les poids d'ordonnées (k, n_k) et $(k+1, n_{k+1})$.
- On appelle *altitude maximale* la valeur maximale prise par la suite. Construire une procédure donnant l'altitude maximale en fonction de n_0.
- On appelle *temps de vol* le nombre de termes d'une suite. Construire une procédure T qui donne l'indice du premier indice tel que la suite prend la valeur 1 -appelé *temps de vol* selon le premier terme.
- Tracer le graphe du temps de vol en fonction du premier terme pour $n_0 \in [1, 1000]$.

2 Méthodes numériques

Exercice 8. Cet exercice propose une implémentation de la méthode de Newton pour l'approximation de zéros de fonctions à valeurs réelles. Cette méthode est basée sur le théorème suivant :

Théorème. *Soit $\varepsilon > 0$, $f \in \mathcal{C}^2(\mathbb{R})$, $a \in \mathbb{R}$ tels que $f(a) = 0$, $f'(a) \neq 0$ et $\forall x \in I :=]a - \varepsilon; a + \varepsilon[\setminus\{a\}$, $f(x) \neq 0$. On définit alors la suite :*

$$x_0 \in I, x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

Alors la suite $x_n \to a$ lorsque $n \to +\infty$.

- Écrire une procédure `MethNewton` := `proc(f,x0},n)` qui prend en entrée une fonction f, un réel x_0 et un entier naturel n, et renvoie un vecteur à $n+1$ éléments dont le $i^{\text{è}}$ coefficient est le réel x_{i-1}.
- Appliquer cet algorithme à la fonction $x \in \mathbb{R} \mapsto x^3 - x - 1$ pour $x_0 = 0$ et $n = 100$. Représenter graphiquement l'ensemble des points $(n, x_n)_{n \in [1;100]}$.
- Modifier la procédure `MethNewton` := `proc(f,x0, pre)` pour que l'algorithme s'arrête dès que $|x_n - x_{n-1}| < 10^{-pre}$ pour un certain entier n. Comparer votre procédure avec la procédure `fsolve` de Maple à l'aide de la commande `time`.

Exercice 9. On s'intéresse maintenant au schéma d'Euler explicite d'ordre 1 pour approcher la solution d'une équation différentielle d'ordre 1. On rapelle le principe de cette méthode à l'aide du théorème suivant :

Théorème. *Soit $f \in \mathcal{C}^1(\mathbb{R})$. Soit l'équation différentielle $y'(t) = f(y(t))$ sur $[0, T]$ et $y(0) = y_0 \in \mathbb{R}$. On définit la suite $(x_i)_{i \in [0;N]}$:*

$$x_0 = y_0; x_{i+1} = x_i + hf(x_i)$$

où $h = \frac{T}{N}$. La fonction affine par morceaux définie par le couples $(hi, x_i)_{i \in [0;N]}$ et une approximation de la solution théorique y.

- Écrire une procédure `SchEuler := proc(f,T,y0,N)` qui prend en entrée une fonction f, un temps d'observation T, un état initial y_0 et un nombre de points d'approximation N qui trace la fonction affine par morceaux définie par le théorème notée x
- Appliquer la procédure `SchEuler` à la fonction $f : x \in \mathbb{R} \mapsto 2x+3$, $T = 10$ et $y_0 = 2$. Déterminer puis implémenter dans l'environnement Maple la fonction g solution de l'équation différentielle $g'(t) = f(g(t))$ et $g(0) = 2$. Ajouter le graphe de g à celui de x.
- Idem avec $f_1 : x \in \mathbb{R} \mapsto x^2, T = 10, y_0 = 1$.
- Idem avec $f_2 : x \in \mathbb{R} \mapsto \sin(x), T = 100, y_0 = 1$.

Exercice 10. Tester les procédures `MethNewton` et `SchEuler` dans d'autres configurations. Comparer avec les procédures déjà implémentées dans l'environnement Maple.