Université de Picardie Jules Verne — L2 Mathématiques : Langage de calcul numérique

TP2 : Arithmétique de l'ordinateur

Exercice 1 : Suite de Muller

On s'intéresse à la suite $(u_n)_{n\in\mathbb{N}}$ définie par la relation de récurrence suivante :

$$\forall n \in \mathbb{N}^*, \quad u_{n+1} = 111 - \frac{1130}{u_n} + \frac{3000}{u_n u_{n-1}} \tag{1}$$

et initialisée par ses deux premiers termes u_0 et u_1. On peut montrer qu'une telle suite peut s'écrire sous la forme :

$$\forall n \in \mathbb{N}, \quad u_n = \frac{\alpha 100^{n+1} + \beta 6^{n+1} + \gamma 5^{n+1}}{\alpha 100^n + \beta 6^n + \gamma 5^n}, \tag{2}$$

avec α, β et γ des réels dépendant de u_0 et u_1 et tels que $\alpha + \beta + \gamma = 1$. On fixera dans tout cet exercice $u_0 = 2$ et $u_1 = -4$.

1. Déterminer (sur le papier) la limite de la suite $(u_n)_{n\in\mathbb{N}}$ dans les cas suivants :

- **a.** $\alpha = 0$,
- **b.** $\alpha \neq 0$.

On pourra pour cela utiliser la forme (2).

2. Écrire un script permettant :

- **a.** de calculer les 30 premiers termes de la suite $(u_n)_{n\in\mathbb{N}}$ en utilisant la définition par récurrence (1) ;
- **b.** de tracer le graphe de ces termes en fonction de n.

Que peut-on observer ?

3.
- **a.** Calculer les coefficients α, β et γ correspondant à cette suite.
- **b.** Écrire un script permettant de calculer les 30 premiers termes de la suite $(u_n)_{n\in\mathbb{N}}$ en utilisant la définition (2). On tracer le graphe de ces termes en fonction de n.
- **c.** Que peut-on observer ?

4. Reprendre la question précédente, mais en simulant une petite erreur dans le terme α. On pourra par exemple considérer $\alpha + \epsilon$ avec ϵ la précision machine (obtenable dans Scilab avec `%eps`).

5. Conclure.

Exercice 2 : Accumulation d'erreurs

On considère un exemple de calcul inutile : obtenir 1 en sommant n fois la valeur $\frac{1}{n}$, pour $n \in \mathbb{N}^*$. Écrire un script implémentant l'Algorithme 1.

Algorithme 1 : Somme inutile

$n = 10^5$;
$s = 0$;
pour $k = 1, \ldots, n$
  $s = s + 1/n$;

Que vaut $1 - s$ à l'issue de l'exéution de ce script ? Expliquer pourquoi l'Algorithme 2 ne nous dira jamais bonjour.

Algorithme 2 : Bonjour

$n = 10^5$;
$s = 0$;
tant que $1 - s \neq 0$
  $s = s + 1/n$;
Afficher 'Bonjour';

Exercice 3 : De l'importance de la factorisation
On souhaite écrire une fonction calculant les valeurs de la fonction polynomiale suivante :

$$\forall x \in \mathbb{R}, \quad f(x) = (x-2)^9, \tag{3}$$

qui peut également être écrite sous la forme :

$$\forall x \in \mathbb{R}, \quad f(x) = \sum_{k=0}^{9} \binom{9}{k} (-2)^{(9-k)} x^k. \tag{4}$$

Écrire deux fonctions Scilab permettant de calculer les valeurs de f, l'une avec la formule (3), l'autre avec la formule (4). Tracer les graphes des valeurs obtenues en fonction de x, pour x variant dans l'intervalle $[2 - 10^{-5}, 2 + 10^{-5}]$. Que peut-on observer ?

Exercice 4 : Algorithme de Malcolm
On considère l'algorithme suivant.

Algorithme 3 : Algorithme de Malcolm

$a = 1$;
$b = 1$;
tant que $((a+1) - a) - 1 = 0$
  $a = 2a$;
tant que $((a+b) - a) - b \neq 0$
  $b = b + 1$;

Écrire un script implémentant cet algorithme et afficher la valeur de b obtenue. Que fait cet algorithme ?