

2016

Exercice 1

On considère l'ensemble U des suites réelles $u = (u_n)_{n \in \mathbb{N}}$ telles que :

$$\forall n \in \mathbb{N}, \quad u_{n+2} = u_{n+1} + 2u_n.$$

Pour un $u \in U$ les deux premiers termes u_0 et u_1 sont donnés. La suite nulle définie par $u_0 = 0$ et $u_1 = 0$ appartient à U .

1. Montrer que si $a \in U$, $b \in U$ et $\alpha \in \mathbb{R}$, on a $a + b \in U$ et $\alpha a \in U$.
En déduire que U est un espace vectoriel sur \mathbb{R} .
2. Soient c et d les deux suites appartenant à U telles que $c_0 = 1$, $c_1 = 0$, $d_0 = 0$, $d_1 = 1$.
 - (a) Montrer que (c, d) est une base de U .
 - (b) Quelle est la dimension de l'espace vectoriel U ?
3. (a) Montrer qu'il existe deux réels distincts et non nuls ρ et σ que l'on calculera, avec $\rho < 0 < \sigma$, tels que les suites géométriques $(\rho^n)_{n \in \mathbb{N}}$ et $(\sigma^n)_{n \in \mathbb{N}}$ appartiennent à U .
On notera r et s les suites telles que $\forall n \in \mathbb{N}$, $r_n = \rho^n$ et $s_n = \sigma^n$.
 - (b) Montrer que (r, s) est une autre base de U .
4. (a) Si v est la suite de U telle que $v_0 = x$, $v_1 = y$, donner en fonction de x et y les composantes de v dans la base (r, s) .
 - (b) En déduire une formule générale de v_n en fonction de n (sans utiliser la formule de récurrence).

Les deux questions d'informatique suivantes peuvent être rédigées au choix en pseudo-code ou en Scilab.

5. (a) Écrire une procédure informatique F ayant pour variables d'entrée deux réels x , y et un entier naturel n et qui renvoie le terme v_n de la suite $v \in U$ telle que $v_0 = x$ et $v_1 = y$.
La fonction F utilisera la relation de récurrence $v_{n+2} = v_{n+1} + 2v_n$ dans une boucle ou de manière récursive.
- (b) Écrire une procédure informatique G ayant également pour variable d'entrée deux réels x , y et un entier naturel n et qui renvoie aussi le terme v_n de la suite $v \in U$ telle que $v_0 = x$ et $v_1 = y$, mais sans faire de boucle (en utilisant la formule établie à la question 4.b).
- (c) Expliquez quelle est, entre F et G , la fonction la plus efficace pour calculer v_n ?

```

function nb=f(t)
    nb=0
    n=length(t)
    for i=1:(n-1)
        if t(i)<t(i+1)
            nb=nb+1
        end
    end
endfunction

```

ATS

2017

On se donne un tableau à une dimension, de longueur n dont les entrées sont indexées de 1 à n , ne contenant que des 0 et des 1, commençant et se terminant par un 0. On supposera qu'il ne contient pas que des 0.

Dans un tel tableau on appelle **séquence1** une suite de 1 consécutifs précédés et suivis d'au moins un 0. Ci-dessous, on a un exemple d'un tel tableau de longueur 16, comportant 4 **séquence1**, dont une de longueur 4 entre les numéros 8 et 11, et une en 15 de longueur 1. On se propose d'écrire des fonctions en métalangage ou en Scilab permettant de calculer le nombre, la longueur et la position de telles **séquence1**.

Index i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Tableau1(i)	0	0	1	1	1	1	0	1	1	1	1	0	1	0	1	0

1. Soit la fonction f écrite en Scilab, avec en entrée un tableau t défini précédemment.

```

function nb=f(t)
    nb=0
    n=length(t)
    for i=1:(n-1)
        if t(i)<t(i+1)
            nb=nb+1
        end
    end
endfunction

```

Expliquer ce que renvoie $f(\text{Tableau1})$ et préciser le fonctionnement de f .

2. Écrire en métalangage ou en Scilab une fonction g qui détermine et renvoie la position et la longueur [début,fin,longueur] de la plus longue séquence1 du tableau (en cas d'égalité de longueur celle de la première rencontrée). Par exemple $g(\text{Tableau1})$ doit renvoyer la liste [3,6,4].

INFO 2018

On admet que $\sum_{n=1}^{\infty} \frac{1}{n^6} = \frac{\pi^6}{945}$ et $\sum_{n=N+1}^{+\infty} \frac{1}{n^6} \leq \frac{1}{5N^5}$.

1. Ecrire en métalangage ou en scilab une fonction *fin* prenant comme argument un réel *eps* avec $10^{-10} \leq eps \leq 1$, et renvoyant le plus petit entier naturel non nul *N* tel que $\frac{1}{N^5} \leq eps$.
2. Ecrire en métalangage ou en scilab une fonction *somme* prenant comme argument un réel *N* strictement positif et renvoyant une approximation décimale de $\sum_{n=1}^N \frac{1}{n^6}$.
3. En utilisant les deux fonctions précédentes, écrire en métalangage ou en Scilab une fonction *approxPi6* prenant comme argument un réel *eps* strictement positif et renvoyant une approximation de π^6 à *eps* près.

INFO 2019

les deux questions sont indépendantes

1. On souhaite obtenir un encadrement du réel α , solution de l'équation $\text{sh}(\alpha)-1=0$, en appliquant un procédé de dichotomie. Recopier en la complétant la fonction Scilab suivante, qui prend en argument un réel strictement positif $\varepsilon > 0$, et renvoie deux réels *a* et *b* vérifiant $a \leq \alpha \leq b$ et $b - a \leq \varepsilon$. On sait que α est compris entre 0 et 1.

```
function [a,b]=dicho(eps)
```

```
    a = ...
```

```
    b = ...
```

```
    while .....
```

```
        c = (a+b)/2
```

```
        if ..... then a = c
```

```
            else b = c
```

```
        end
```

```
    end
```

```
endfunction
```

Note : la fonction *sh* s'écrit *sinh* en Scilab

2. En utilisant la fonction précédente sur machine, on trouve 0,881 comme valeur approchée de α . On rappelle de plus que la suite $(I_n)_{n \in \mathbb{N}}$ vérifie :

$$I_0 = \alpha, \text{ et pour tout } n \in \mathbb{N}, I_{n+1} = \frac{\sqrt{2}}{2n+2} - \left(\frac{2n+1}{2n+2}\right) I_n.$$

Ecrire une fonction en Scilab ou bien en pseudo code, qui prend en entrée un entier naturel *n* et renvoie une valeur approchée de I_n .

ÉNONCÉ 2020

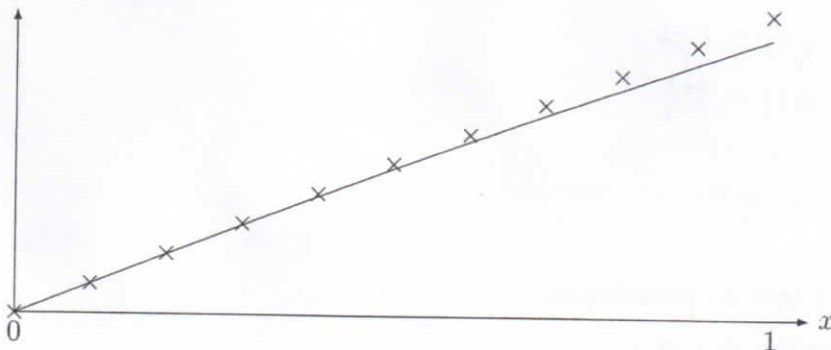
Partie C – Résolution approchée d'un problème de Cauchy

Soit N un entier naturel non nul. Dans cette partie, on cherche à résoudre l'équation (E_0) , avec les conditions initiales $y(0) = 0$ et $y'(0) = 1$ par la méthode d'Euler, en prenant un pas égal à $1/N$. On admet que cela revient à calculer les termes de la suite $(u_n)_{n \in \mathbb{N}}$ définie par $u_0 = 0$, $u_1 = 1/N$ et la relation de récurrence

$$\forall n \in \mathbb{N}, \quad N^2(u_{n+2} - 2u_{n+1} + u_n) + u_n = \frac{n^2}{N^2}.$$

1. Pour tout $n \in \mathbb{N}$, exprimer u_{n+2} en fonction de N , n , u_n et u_{n+1} .
2. Écrire une fonction **cauchy**, en *Scilab* ou bien en pseudo-code, qui prend en entrée un entier naturel non nul N et renvoie le vecteur $[u_0, u_1, \dots, u_N]$.

Sur la figure suivante, on représente le graphe de la solution théorique du problème de Cauchy sur l'intervalle $[0, 1]$, ainsi que les points de coordonnées $(k/N, u_k)$ avec $k \in \{0, 1, \dots, N\}$ (ici, on a choisi $N = 10$).



3. Comment agir sur le paramètre N pour améliorer la solution approchée ? Quel est l'impact sur le temps de calcul ?

ATS 2021

Partie C – Calcul approché

On considère la fonction $f: \mathbb{R} \rightarrow \mathbb{R}$ définie par

$$f(x) = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n}}{n!^2}.$$

Pour tout réel a , l'unique entier $p \in \mathbb{Z}$ tel que $p - 1 < a \leq p$ est appelé la *partie entière supérieure* de a , et on le note $\lceil a \rceil$. De manière équivalente, $\lceil a \rceil$ est le plus petit entier qui soit supérieur ou égal à a .

1. Soit $x \in \mathbb{R}^*$. On note $n_x = \lceil |x| - 1 \rceil$. Montrer que la série $\sum_{n=n_x}^{\infty} \frac{(-1)^n x^{2n}}{n!^2}$ vérifie la règle spéciale des séries alternées.

Dans la suite, on admet que le résultat de la question précédente implique que

$$\left| f(x) - \sum_{n=0}^N \frac{(-1)^n x^{2n}}{n!^2} \right| \leq \frac{x^{2(N+1)}}{(N+1)!^2}$$

pour tout entier $N \geq n_x$.

2. La fonction `fapprox` ci-dessous prend en entrée un réel x non nul, un réel strictement positif ε , et renvoie une approximation de $f(x)$ à ε près, c'est-à-dire un réel y tel que $|f(x) - y| \leq \varepsilon$. Recopier la fonction `fapprox` (en version *Scilab* ou bien pseudo-code) et la compléter.

*Note : les fonctions « valeur absolue » et « partie entière supérieure » s'écrivent respectivement `abs` et `ceil` en *Scilab*.*

```
function y = fapprox(x, eps)
  nx = ceil(abs(x) - 1)
  y = 0
  t = ...
  n = ...
  while ... | ...
    y = y + t
    n = n + 1
    t = -t * x^2 / n^2
  end
end
```

```
fonction FAPPROX(x, eps)
  nx ← [|x| - 1]
  y ← 0
  t ← ...
  n ← ...
  tant que ... ou ...
    y ← y + t
    n ← n + 1
    t ← -tx^2/n^2
  fin tant que
  renvoyer y
fin fonction
```

ATS 2022

$$s_n(x) = \sum_{k=0}^{n-1} \frac{(-1)^k x^{2k+1}}{(2k+1)^2}$$

On a montré à la question 4 que $|s_n(1) - I| \leq \frac{1}{(2n+1)^2}$

5. Déterminer un entier naturel $N \in \mathbb{N}^*$ tel que $s_n(1)$ soit une valeur approchée de I à 10^{-4} près.

6. Ecrire une fonction nommée « s », en scilab ou bien en pseudo-code qui prend en entrée un entier naturel non nul n et renvoie le nombre $s_n(1)$.

7. Ecrire une fonction nommée « trouve_N », en scilab ou bien en pseudo-code, qui prend en entrée un réel strictement positif ε et renvoie le plus petit entier N tel que $\frac{1}{(2n+1)^2} \leq \varepsilon$.

8. En utilisant les fonctions s et trouve_n , écrire en scilab ou bien en pseudo-code les instructions calculant et affichant une valeur approchée de l'intégrale I à 10^{-10} près.