Méthodes itératives : Annales 2015 à 2020

I. Suites de type $u_{n+1} = f(u_n)$

ECRICOME – 2015

- On considère la fonction F définie par :

$$F(x) = \begin{cases} 0 & \text{si } x < 0 \\ 1 - e^{-x} & \text{si } x \geqslant 0 \end{cases}$$

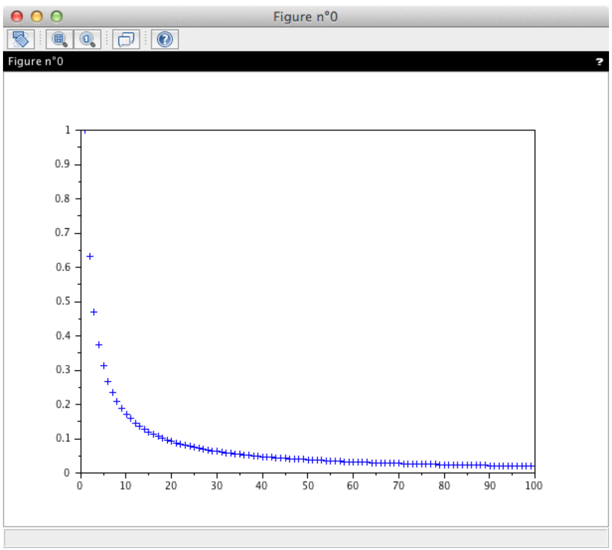
(fonction de répartition de d'une v.a.r. X telle que $X \hookrightarrow \mathcal{E}(1)$)
et on définit la suite $(u_n)_{n\in\mathbb{N}^*}$ définie par :

$$\begin{cases} u_1 = 1 \\ \forall n \in \mathbb{N}^*, \ u_{n+1} = F(u_n) \end{cases}$$

a) Recopier et compléter le programme **Scilab** suivant qui permet de représenter les cent premiers termes de la suite $(u_n)_{n\in\mathbb{N}^*}$:

```
U = zeros(1,100)
U(1) = 1
for n = 1 : 99
    U(n+1) = ------
end
plot(U,"+")
```

b) Le programme complété permet d'obtenir la représentation graphique suivante.

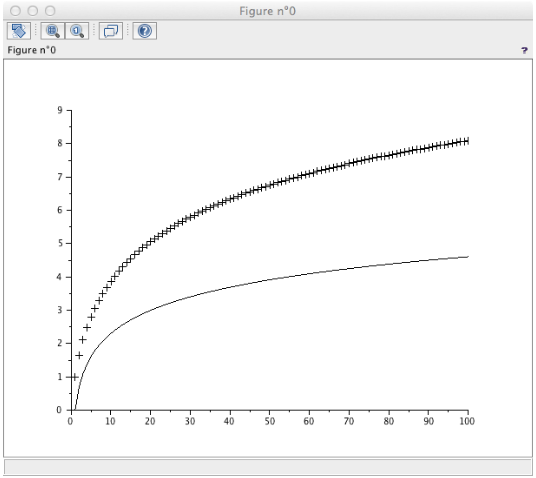


Quelle conjecture pouvez-vous émettre sur la monotonie et la limite de la suite $(u_n)_{n\in\mathbb{N}^*}$?

c) On modifie le programme écrit en question ***a)*** en remplaçant la dernière ligne par :

```
X = 1:100
S = cumsum(U)
Y = log(X)
plot2d(X, S, -1)
plot2d(X, Y)
```

Le programme ci-dessus permet d'obtenir la représentation graphique suivante :



Que représente le vecteur-ligne `S` ?
Quelle conjecture pouvez-vous émettre sur la nature de la série de terme général u_n ?

`ECRICOME – 2018`

- On considère les matrices : $A = \begin{pmatrix} 2 & 1 & -2 \\ 0 & 3 & 0 \\ 1 & -1 & 5 \end{pmatrix}$ et $B = \begin{pmatrix} 1 & -1 & -1 \\ -3 & 3 & -3 \\ -1 & 1 & 1 \end{pmatrix}$.

  On pose $X_0 = \begin{pmatrix} 3 \\ 0 \\ -1 \end{pmatrix}$, $X_1 = \begin{pmatrix} 3 \\ 0 \\ -2 \end{pmatrix}$, et pour tout $n \in \mathbb{N}$: $X_{n+2} = \dfrac{1}{6}\, A\, X_{n+1} + \dfrac{1}{6}\, B\, X_n$.

a) Compléter la fonction ci-dessous qui prend en argument un entier n supérieur ou égal à 2 et qui renvoie la matrice X_n :

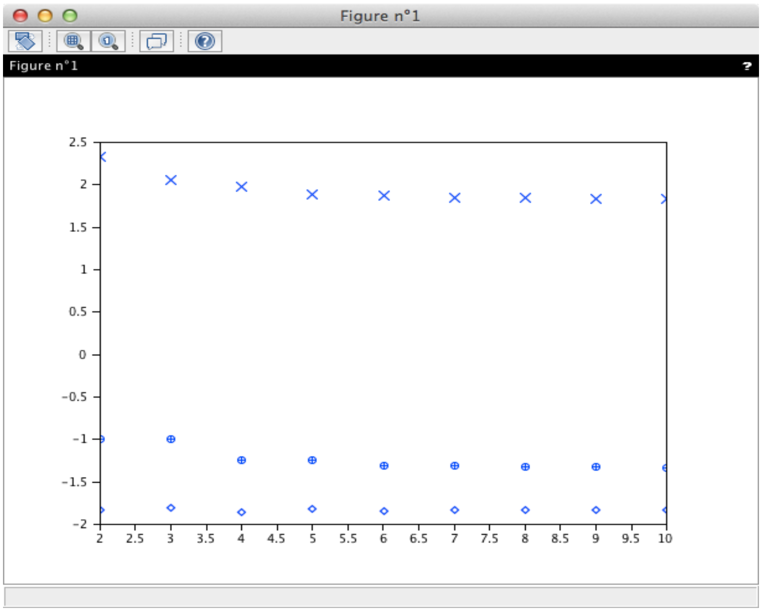
```
function res = X(n)
    Xold = [3; 0; -1]
    Xnew = [3; 0; -2]
    A = [2,1,-2; 0,3,0; 1,-1,5]
    B = [1,-1,-1; -3,3,-3; -1,1,1]
    for i = 2:n
        Aux = .........
        Xold = ........
        Xnew = ........
    end
    res = ........
endfunction
```

- Dans l'exercice, il était noté $X_n = \begin{pmatrix} \alpha_n \\ \beta_n \\ \gamma_n \end{pmatrix}$ et on démontrait, pour tout $n \in \mathbb{N}$:

$$\alpha_n = \frac{11}{6} + \frac{2}{3}\left(-\frac{1}{2}\right)^n + \left(\frac{1}{2}\right)^{n-1} - \frac{3}{2}\left(-\frac{1}{3}\right)^n$$

$$\beta_n = \left(\frac{1}{2}\right)^{n-1} - \frac{2}{3}\left(-\frac{1}{2}\right)^n - \frac{4}{3} \quad \text{et} \quad \gamma_n = -\frac{11}{6} - \frac{2}{3}\left(-\frac{1}{2}\right)^n + \frac{3}{2}\left(-\frac{1}{3}\right)^n$$

b) La fonction précédente a été utilisée dans un script permettant d'obtenir graphiquement (voir figure 1) les valeurs de α_n, β_n et γ_n en fonction de n.
Associer chacune des trois représentations graphiques à chacune des suites $(\alpha_n)_{n\in\mathbb{N}}$, $(\beta_n)_{n\in\mathbb{N}}$, $(\gamma_n)_{n\in\mathbb{N}}$ en justifiant votre réponse.



EDHEC – 2019

- On définit la suite (u_n) par : $\forall n \in \mathbb{N}$, $u_n = \dfrac{4^n\,(n!)^2}{(2n+1)!}$.

  On admet que, si `t` est un vecteur, la commande `prod(t)` renvoie le produit des éléments de `t`.
  Compléter le script **Scilab** suivant afin qu'il permette de calculer et d'afficher la valeur de u_n pour une valeur de n entrée par l'utilisateur.

```
n = input('entrez une valeur pour n : ')
x = 1:n
m = 2 * n + 1
y = 1:m
v = ..........
w = ..........
u = .......... * v^2 / w
disp(u)
```

II. Calcul du premier entier n qui vérifie une condition donnée

EDHEC – 2016

- On considère une suite $(u_n)_{n\in\mathbb{N}}$ qui vérifie :

$$\forall n \in \mathbb{N},\ \mathrm{e}^{-\sqrt{u_n}} \leqslant u_n - n \leqslant \mathrm{e}^{-\sqrt{n}}$$

a) Compléter les commandes **Scilab** suivantes afin qu'elles permettent d'afficher un entier n pour lequel $u_n - n$ est inférieur ou égal à 10^{-4}.

```
n = 0
while ------
    n = ------
end
disp(n)
```

b) Le script ci-dessous affiche l'une des trois valeurs $n = 55$, $n = 70$ et $n = 85$.
Préciser laquelle en prenant $2,3$ comme valeur approchée de $\ln(10)$.

EML – 2016

- On considère la fonction f définie par : $f(x) = \begin{cases} x^2 - x\ln(x) & \text{si } x \neq 0 \\ 0 & \text{si } x = 0 \end{cases}$

  et on définit la suite $(u_n)_{n\in\mathbb{N}^*}$ par : $\begin{cases} u_0 = \frac{1}{2} \\ \forall n \in \mathbb{N},\ u_{n+1} = f(u_n) \end{cases}$

  (on démontre que (u_n) est croissante et qu'elle converge vers 1)

a) Écrire un programme en **Scilab** qui calcule et affiche un entier naturel N tel que :

$$1 - u_N < 10^{-4}$$

EML – 2017

- On considère la fonction $f :]0, +\infty[\to \mathbb{R}$ définie par $f : x \mapsto \mathrm{e}^x - \mathrm{e}\ln(x)$.
- On considère la suite réelle $(u_n)_{n\in\mathbb{N}}$ définie par : $\begin{cases} u_0 = 2 \\ \forall n \in \mathbb{N},\ u_{n+1} = f(u_n) \end{cases}$

a) Écrire un programme en **Scilab** qui, étant donné un réel A, renvoie un entier naturel N tel que $u_N \geqslant A$.

III. Calcul de la valeur approchée de la limite d'une suite

EML – 2015

- On considère l'application $f : \mathbb{R} \to \mathbb{R},\ x \mapsto f(x) = x^3\mathrm{e}^x$.

a) Montrer que la série $\sum\limits_{n\geqslant 1} \dfrac{1}{f(n)}$ converge. On note $S = \sum\limits_{n=1}^{+\infty} \dfrac{1}{f(n)}$.

On démontre alors que : $\forall n \in \mathbb{N}^*,\ \left| S - \sum\limits_{k=1}^{n} \dfrac{1}{f(k)} \right| \leqslant \dfrac{1}{(\mathrm{e}-1)\ \mathrm{e}^n}$.

b) En déduire une fonction **Scilab** qui calcule une valeur approchée de S à 10^{-4} près.

EML – 2018

- On pose : $u_0 = 4$ et $\forall n \in \mathbb{N},\ u_{n+1} = \ln(u_n) + 2$.

On devait démontrer les propriétés suivantes.

- $\forall n \in \mathbb{N},\ u_n \geqslant b$.
- $\forall n \in \mathbb{N},\ u_{n+1} - b \leqslant \dfrac{1}{2}\,(u_n - b)$.
- $\forall n \in \mathbb{N},\ 0 \leqslant u_n - b \leqslant \dfrac{1}{2^{n-1}}$.

a) Écrire une fonction **Scilab** d'en-tête `function u = suite(n)` qui, prenant en argument un entier n de $\mathbb{N}$, renvoie la valeur de u_n.

b) Recopier et compléter la ligne 3 de la fonction **Scilab** suivante afin que, prenant en argument un réel `epsilon` strictement positif, elle renvoie une valeur approchée de b à `epsilon` près.

```
1 function b = valeur_approchee(epsilon)
2     n = 0
3     while ..........
4         n = n + 1
5     end
6     b = suite(n)
7 endfunction
```

EML – 2019

- On considère la fonction f définie sur $]0, +\infty[$ par :

$$\forall t \in\ ]0, +\infty[,\ \ f(t) = t + \frac{1}{t}$$

- On introduit la suite $(u_n)_{n \in \mathbb{N}^*}$ définie par :

$$u_1 = 1 \quad \text{et} \quad \forall n \in \mathbb{N}^*,\ \ u_{n+1} = u_n + \frac{1}{n^2\, u_n} = \frac{1}{n}\, f(n\, u_n)$$

- Dans l'exercice, on devait démontrer la convergence de la suite (u_n) vers un réel ℓ (qu'il ne fallait pas déterminer). De plus, on devait démontrer :

$$\forall p \geqslant 2,\ 0 \ \leqslant\ \ell - u_p \ \leqslant\ \frac{1}{p-1} \qquad (*)$$

a) Recopier et compléter les lignes 3 et 4 de la fonction **Scilab** suivante afin que, prenant en argument un entier n de $\mathbb{N}^*$, elle renvoie la valeur de u_n.

```
1 function u=suite(n)
2   u = 1
3   for k = ..................
4     u = .................
5   end
6 endfunction
```

b) Déduire de la propriété $(*)$ une fonction **Scilab** qui renvoie une valeur approchée de ℓ à 10^{-4} près.

ECRICOME – 2018

- Pour tout entier naturel n non nul, on pose : $u_n = \sum\limits_{k=1}^{n} \frac{1}{k} - \ln(n)$.
- On démontrait que la suite (u_n) était convergente, vers une limite notée $\gamma \in \mathbb{R}$ puis :

$$\forall n \in \mathbb{N}^*, \ |u_n - \gamma| \leqslant \frac{1}{n}$$

a) Écrire une fonction d'en-tête : `function y = u(n)` qui prend en argument un entier naturel n non nul et qui renvoie la valeur de u_n.

b) On rappelle que l'instruction `floor(x)` renvoie la partie entière d'un réel x et on suppose que la fonction `u` a été correctement programmée. Expliquer l'intérêt et le fonctionnement du script ci-dessous :

```
eps = input('Entrer un réel strictement positif : ')
n = floor(1/eps) + 1
disp(u(n))
```

IV. Calcul de la valeur approchée des éléments d'une suite

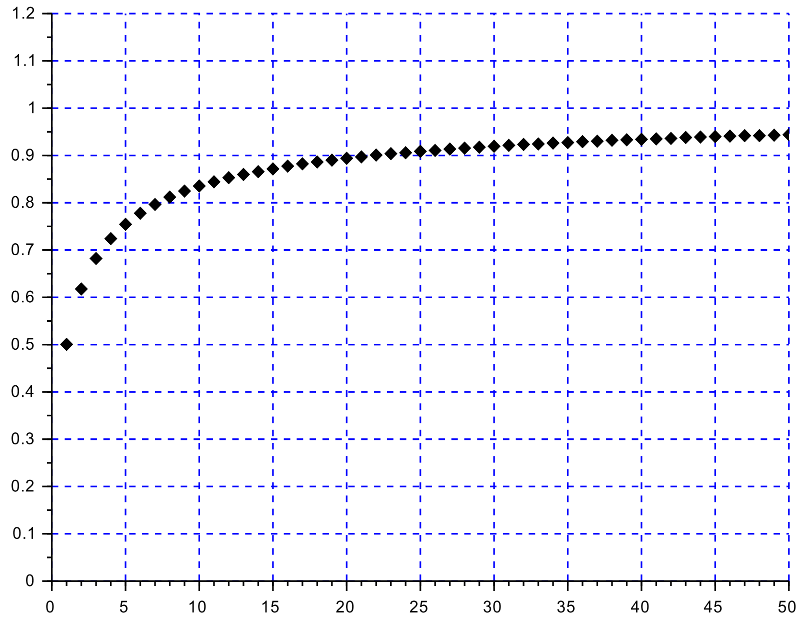
EML – 2020

- On note, pour tout n de $\mathbb{N}^*$, (E_n) l'équation : $x^n + x - 1 = 0$.
- On devait démontrer les propriétés suivantes :
  - × pour tout $n \in \mathbb{N}^*$, l'équation (E_n) admet une unique solution sur $\mathbb{R}_+$ que l'on note u_n (via l'étude de la fonction $x \mapsto x^n + x - 1$).
  - × pour tout n de $\mathbb{N}^*$, u_n appartient à l'intervalle $]0,1[$.

a) Recopier et compléter la fonction **Scilab** suivante afin que, prenant en argument un entier n de $\mathbb{N}^*$, elle renvoie une valeur approchée de u_n à 10^{-3} près, obtenue à l'aide de la méthode par dichotomie.

```
function u = valeur_approchee(n)
    a = 0
    b = 1
    while ...
        c = (a + b) / 2
        if (c ^ n + c - 1) > 0 then
            ...
        else
            ...
        end
        u = ...
    end
endfunction
```

b) On représente alors les premiers termes de la suite $(u_n)_{n\in\mathbb{N}^*}$ et on obtient le graphe suivant. Quelles conjectures peut-on faire sur la suite $(u_n)_{n\in\mathbb{N}^*}$ concernant sa monotonie, sa convergence et son éventuelle limite ?



`ECRICOME – 2019`

- Pour tout entier n non nul, on note h_n la fonction définie sur $\mathbb{R}_+^*$ par :

$$\forall x > 0, \quad h_n(x) \ = \ x^n + 1 + \frac{1}{x^n}$$

- On devait démontrer les propriétés suivantes :
  - × pour tout entier naturel n non nul, la fonction h_n est strictement décroissante sur $]0,1[$ et strictement croissante sur $[1,+\infty[$.
  - × pour tout entier n non nul, l'équation $h_n(x) = 4$ admet exactement deux solutions, notées u_n et v_n et vérifiant : $0 < u_n < 1 < v_n$.

a) Écrire une fonction **Scilab** d'en-tête `function y = h(n,x)` qui renvoie la valeur de $h_n(x)$ lorsqu'on lui fournit un entier naturel n non nul et un réel $x \in \mathbb{R}_+^*$ en entrée.

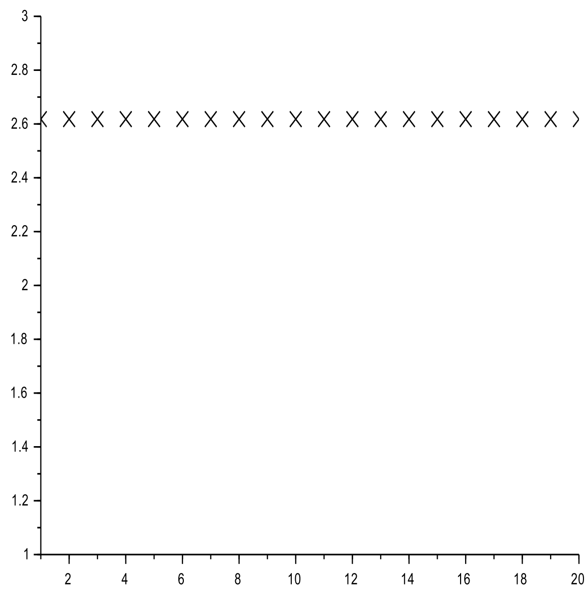
b) Compléter la fonction suivante pour qu'elle renvoie une valeur approchée à 10^{-5} près de v_n par la méthode de dichotomie lorsqu'on lui fournit un entier $n \geqslant 1$ en entrée :

```
function res=v(n)
  a = 1
  b = 3
  while (b-a) > 10 ^ (-5)
    c = (a+b)/2
    if h(n,c) < 4 then
      ........
    else
      ........
    end
  end
  ..........
endfunction
```

c) À la suite de la fonction v, on écrit le code suivant :

```
X = 1:20
Y = zeros(1,20)
for k = 1:20
  Y(k) = v(k)^k
end
plot2d(X, Y, style=-2, rect=[1,1,20,3])
```

À l'exécution du programme, on obtient la sortie graphique suivante :



Expliquer ce qui est affiché sur le graphique ci-dessus.
Que peut-on conjecturer ?