TP2 : Calcul du $m^{\text{ème}}$ élément, des m premiers éléments d'une suite

(Révisions sur la structure itérative `for`*)*

Pré-requis : l'objectif des premières séances de TP est de faire le point sur des fonctionnalités importantes de **Scilab** qui ont été vues en première année. Je vous invite à consulter les chapitres de cours correspondants sur ma page : support informatique.
On pourra en particulier se reporter au « CH 7 : Les structures itératives ».

- Dans le dossier `Info_2a` (resp. `Info_3a`) créé au TP précédent, créer le dossier `TP_2`.

I. Calcul du $10^{\text{ème}}$ élément d'une suite

On considère la suite $(u_n)_{n\in\mathbb{N}^*}$ définie par : $\left\{\begin{array}{l} \forall n \in \mathbb{N}^*,\ u_{n+1} \ = \ 2\,u_n + n + 1 \\ u_1 \ = \ 1 \end{array}\right.$

D'aute part, on considère le programme **Scilab** suivant :

```
u = 1
u = 2 * u + 1 + 1
u = 2 * u + 2 + 1
u = 2 * u + 3 + 1
u = 2 * u + 4 + 1
u = 2 * u + 5 + 1
```

- Que réalise ce programme ?

- Comment obtenir la valeur de u_{10} ? De u_{20} ? De u_{250} ?

II. Calcul du $m^{\text{ème}}$ élément d'une suite

► Dans un onglet **SciNotes**, écrire un programme qui :
- × demande initialement à l'utilisateur d'entrer au clavier la valeur d'un entier `m`,
- × initialise une variable `u` à la valeur 1,
- × met à jour `u` dans une structure itérative de sorte à ce que `u` contienne la valeur du $m^{\text{ème}}$ élément de la suite en fin de boucle.
- × affiche la valeur de `u`.

Sauvegarder ce programme sous le nom `emeSuiteU.sce`.

► Calculer u_{12} et u_{20} à l'aide du programme précédent.

► Dans un nouvel onglet **SciNotes**, copier-coller le programme précédent.
Modifier ce programme afin d'obtenir une fonction `emeSuiteU` qui :
- × prend en paramètre une variable `m`,
- × calcule en sortie une variable `u` contenant le `m`$^{\text{ème}}$ élément de la suite (u_n).

► Calculer u_7 et u_{15} à l'aide de la fonction précédente.

► Selon vous, quels sont les avantages de la représentation sous forme de programme avec dialogue utilisateur ? Sous forme de fonction ?

III. Calcul des m premiers éléments d'une suite

- Dans un onglet **SciNotes**, écrire un programme qui :
  - demande initialement à l'utilisateur d'entrer au clavier la valeur d'un entier `m`,
  - crée un vecteur `U` composé initialement de `m` zéros,
  - met à jour les cœfficients de `U` dans une structure itérative de sorte à ce que `U` contienne les valeurs des m premiers éléments de la suite (u_n) en fin de boucle.

  Sauvegarder ce programme sous le nom `premSuiteU.sce`.

- Calculer les 5 premiers éléments de la suite à l'aide du programme précédent.

- Dans un nouvel onglet **SciNotes**, copier-coller le programme précédent.
  Modifier ce programme afin d'obtenir une fonction `premSuiteU` qui :
  - prend en paramètre une variable `m`,
  - calcule en sortie une variable `U` contenant les `m` premiers éléments de la suite (u_n).

- Effectuer le tracé des 30 premières valeurs de la suite (u_n).
  Les points correspondant devront apparaître sous la forme d'un cercle rouge.
  Quelle conjecture peut-on émettre sur la limite de la suite (u_n) ?

IV. Suites $u_{n+1} = f(u_n)$ aux concours

IV.1. ECRICOME - 2015

L'épreuve **ECRICOME - 2015** commençait par l'étude d'une suite récurrente de type $u_{n+1} = F(u_n)$. La fonction F est définie comme suit :

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

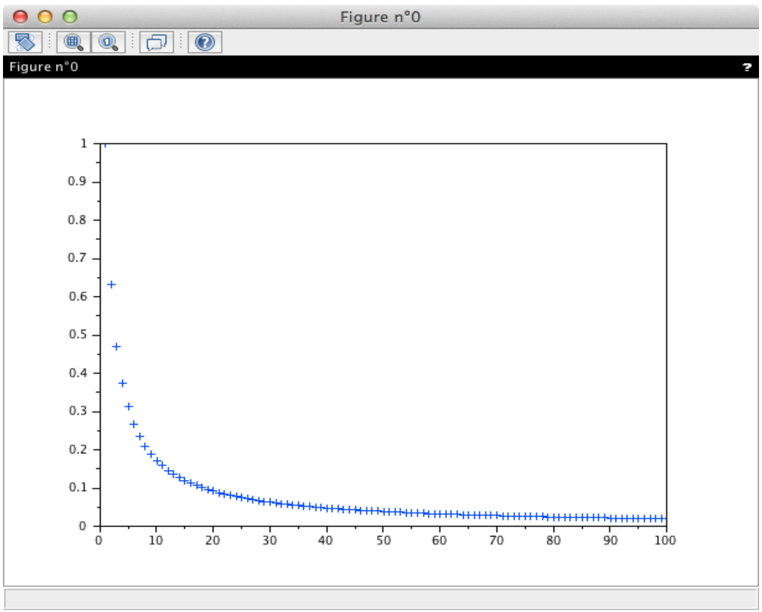
On considère la suite $(u_n)_{n \geqslant 1}$ définie par $u_1 = 1$ et pour tout $n \in \mathbb{N}^*$ par : $u_{n+1} = F(u_n)$.

- *Recopier et compléter le programme **Scilab** suivant qui permet de représenter les cent premiers termes de la suite $(u_n)_{n \geqslant 1}$:*

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

(on demandait de démontrer, dans une question précédente que : $\forall n \geqslant 1,\ u_n > 0$)

- *Le programme précédent complété permet d'obtenir la représentation graphique suivante.*



Quelle conjecture pouvez-vous émettre sur la monotonie et la limite de la suite $(u_n)_{n \geqslant 1}$?

IV.2. ESSEC II - 2016

L'épreuve **ESSEC II - 2016** comportait une unique question d'informatique [1] qui consistait au codage d'une suite récurrente définie comme suit :

$$\begin{cases} u_0 = 1 \\ u_n = u_{n-1}\, p_1 + \ldots + u_0\, p_n \end{cases}$$

▸ *En **Scilab**, soit $P = [p_1, p_2, \ldots, p_n]$ le vecteur ligne tel que $P(j) = p_j$ pour j dans $[\![1, n]\!]$.*
*Écrire un programme en **Scilab** qui calcule u_n à partir de P.*

Remarque

Cette question est bien plus compliquée que ce que laisse entrevoir son énoncé. Détaillons pourquoi.

1) En terme de structures de données :

La suite (u_n) est une suite récurrente dont le $n^{\text{ème}}$ terme dépend de `TOUS` les précédents.
Ce type de suite est bien plus délicate à traiter que les suites récurrentes d'ordre 1 (dont la relation de récurrence s'écrit $u_{n+1} = f(u_n)$) ou d'ordre 2 ($u_{n+1} = g(u_n, u_{n-1})$).

Pour calculer le terme d'indice n, il faut avoir accès aux termes d'indice $0, \ldots, n-1$ de la suite.
↪ il faut donc se servir d'un vecteur pour stocker au fur et à mesure ces valeurs.

2) En terme d'indices :

La suite (u_n) est définie à partir du rang 0 alors que la numérotation des cœfficients matriciels commence à 1 en **Scilab**. De plus, la formule de récurrence mélange ordre croissant pour les éléments de (u_n) et décroissant pour ceux de (p_n).

3) En terme de paramètre :

La fonction consiste à calculer le $n^{\text{ème}}$ terme de la suite (u_n) mais n n'est pas annoncé comme paramètre de la fonction. En effet, le seul paramètre annoncé pour la fonction est le vecteur P. C'est la taille de ce vecteur qui nous fournit la valeur de n.

Pour ce type de questions, il est conseillé de commencer par écrire au brouillon les premières étapes de l'algorithme. Autrement dit, d'effectuer le calcul de u_0, u_1, u_2, u_3, afin de comprendre le mécanisme de calcul.

[1] Ce qui représente une nette amélioration par rapport à l'épreuve 2015.

Afin de répondre à cette question, on pouvait aussi observer que :

$$u_{n-1}\,p_1 + \ldots + u_0\,p_n = \begin{pmatrix} u_0 & u_1 & \ldots & u_{n-1} \end{pmatrix} \times \begin{pmatrix} p_n \\ p_{n-1} \\ \vdots \\ p_1 \end{pmatrix}$$

On pouvait donc se servir des fonctionnalités **Scilab** sur les matrices afin de répondre à cette question.

- Étant donné un vecteur ligne `U`, comment sélectionne-t-on les 5 premiers éléments de `U` ?

- Comment obtenir un vecteur colonne à partir d'un vecteur ligne `P`.

- Comment peut-on obtenir le miroir d'un vecteur `V` comportant `n` éléments (*i.e.* un vecteur contenant les éléments de `V` rangés dans l'ordre inverse).

- Écrire une nouvelle version de la fonction `calcSuiteU` en tirant profit des remarques précédentes.

IV.3. ECRICOME - 2018

On considère les matrices :

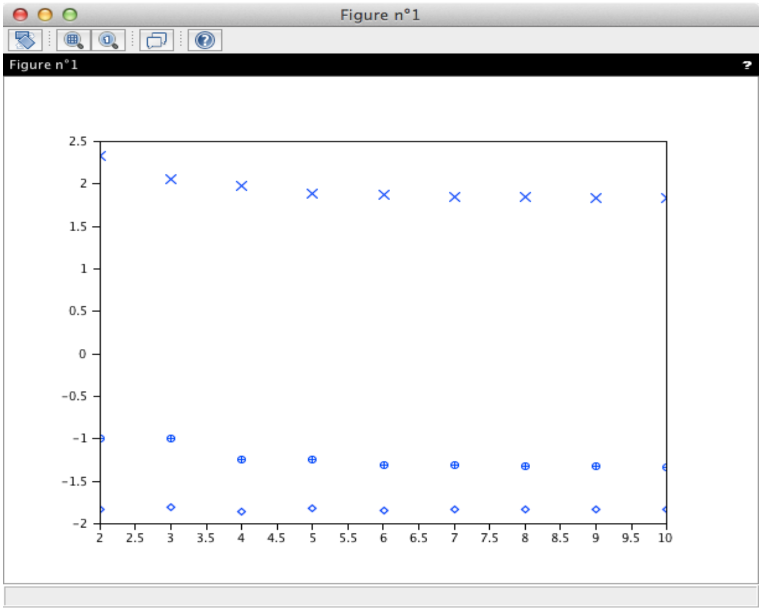
$$A = \begin{pmatrix} 2 & 1 & -2 \\ 0 & 3 & 0 \\ 1 & -1 & 5 \end{pmatrix} \quad \text{et} \quad B = \begin{pmatrix} 1 & -1 & -1 \\ -3 & 3 & -3 \\ -1 & 1 & 1 \end{pmatrix}$$

On pose $X_0 = \begin{pmatrix} 3 \\ 0 \\ -1 \end{pmatrix}$, $X_1 = \begin{pmatrix} 3 \\ 0 \\ -2 \end{pmatrix}$, et pour tout entier naturel 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
```

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.



IV.4. EML - 2018

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

- 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.

Commentaire