

Préparation à l'agrégation interne TP Scilab

B. Boutin - benjamin.boutin@univ-rennes1.fr

Dans ce sujet, nous aborderons plusieurs thèmes du programme de l'agrégation interne qui peuvent chacun faire l'objet d'illustrations numériques avec le logiciel Scilab. Pour votre usage personnel, sachez que vous pouvez télécharger et installer librement le logiciel depuis <http://www.scilab.org>. Chaque thème débutera par un **échauffement** visant à prendre en main quelques fonctionnalités utiles, suivi par quelques **exercices** d'illustration. Enfin une partie **approfondissement** propose quelques problèmes qui pourront être traités à l'issue de la séance. Deux annexes constituent la fin du document, comprenant quelques généralités sur la représentation des nombres en machine d'une part et un **aide-mémoire Scilab** d'autre part.

Concrètement le logiciel permet au choix d'exécuter les commandes au fur et à mesure dans la Console Scilab ou bien, pour des programmes plus élaborés, de travailler dans l'éditeur SciNotes. Les instructions sont alors structurées et sauvegardées dans des fichiers portant l'extension `.sci`. Les commandes de menu « exécuter fichier avec écho » ou « évaluer la sélection avec écho » sont alors des outils précieux !

Approximation de réels par des suites

Échauffement

Quelques commandes utilisant l'outil de régression linéaire et des fonctionnalités de tracés.

```
--> x=linspace(1,5,50); y=3*x.^2+2*rand(x);
--> [a,b,sig]=reglin(log(x),log(y))
--> scf(1); clf; plot(x,y,x,exp(b)*x.^a)
--> [a2,b2,sig]=reglin(log(x($-10:$)),log(y($-10:$)))
--> scf(2); clf; plot(log(x),log(y),'+-r',log(x),a*log(x)+b,'-b',log(x),a2*log(x)+b2,'-g')
--> title('Regression lineaire');xlabel('logx');ylabel('logy');legend('data','reg1','reg2')
```

Exercices

Vitesse de convergence

Définition. Soit $(u_n)_{n \in \mathbb{N}}$ une suite de réels strictement positifs, convergeant vers 0. On suppose que le rapport $\left| \frac{u_{n+1}}{u_n} \right|$ converge, de limite notée ρ , qui est alors nécessairement un élément de $[0, 1]$.

On dit que la convergence de la suite (u_n) est

- **sous-linéaire** si $\rho = 1$
- **linéaire** (ou d'ordre 1 exactement) si $\rho \in]0, 1[$
- **sur-linéaire** (ou super-linéaire) si $\rho = 0$.

Dans ce dernier cas, on appelle **ordre de convergence** le plus grand réel $\alpha \geq 1$ tel que la suite $\frac{|u_{n+1}|}{|u_n|^\alpha}$ est bornée.

Pour le cas d'une suite $(v_n)_{n \in \mathbb{N}}$ convergente de limite $\ell \in \mathbb{R}$, on étend les dénominations précédentes en considérant la vitesse de convergence de la suite $u_n = |v_n - \ell|$.

Remarque : En pratique, pour observer numériquement l'ordre de convergence, on peut utiliser la caractérisation équivalente suivante :

$$\exists \beta \in \mathbb{R}, \forall n \in \mathbb{N}, \ln |u_{n+1}| \leq \alpha \ln |u_n| + \beta.$$

- Déterminer numériquement la vitesse et l'ordre de convergence des suites :

$$u_n = 1/n, \quad v_n = 1/n^3, \quad w_n = 1/(n^2 n!), \quad z_n = 1/2^n, \quad x_n = 0.9999^{3^n},$$

$$y_0 = 0.4, \quad y_{n+1} = y_n - \frac{y_n(y_n - 1)}{2y_n - 1}.$$

Convergence locale d'itérées

Théorème. Soient $f \in \mathcal{C}^1([0, 1])$ et $r \in]0, 1[$ tels que $f(r) = r$ et $|f'(r)| < 1$. Alors il existe $\delta > 0$ tel que pour tout $u_0 \in [0, 1]$ vérifiant $|u_0 - r| \leq \delta$, la suite $u_{n+1} = f(u_n)$ est bien définie et à valeurs dans $[0, 1]$, et qu'elle converge vers r .

- Dans un fichier `iterations.sce` à l'aide de l'éditeur `SciNotes`, écrire le code suivant permettant de calculer les premières itérées $(u_n)_{0 \leq n \leq 30}$ de la suite définie par $u_0 = 0$ et par la récurrence pour $n \geq 0$: $u_{n+1} = \cos(u_n)$. Le résultat sera stocké dans un vecteur ligne `U`.

```
u=0;
U=u;
for i=1:30 do
    u=cos(u);
    U=[U,u];
end
```

Vérifier la bonne exécution depuis la `Console Scilab` à l'aide des commandes suivantes. Attention à rendre le fichier accessible dans le navigateur de fichiers.

```
--> exec iterations.sce
--> plot(U)
```

- À la suite du fichier `iterations.sce`, définir la fonction suivante, qui permet de structurer les itérées `U` en vue de tracés du type « escaliers » ou « escargots ».

```
function [X,Y]=BeforePlot(U)
    // Construit les abscisses et ordonnées pour le trace d'iterations
    // U : vecteur ligne de taille n+1
    // X,Y : vecteur ligne de taille 2n
    X=[U(1:$-1);U(1:$-1)];
    Y=[U(1:$-1);U(2:$)];
    X=X(:);
    Y=Y(:);
endfunction
```

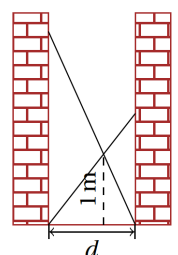
```
--> exec iterations.sce
--> [XU,YU]=BeforePlot(U);
--> scf(2); clf; x=linspace(0,1,100)'; plot(x,[cos(x),x],XU,YU)
```

- Superposer au tracé précédent un second tracé pour l'initialisation $u_0 = 1/2$.
- Par une regression linéaire, déterminer expérimentalement deux constantes $C > 0$ et $\rho > 0$ telles que, ayant noté $\ell = \lim_{+\infty} u_n$, on ait l'estimation asymptotique suivante : $|u_n - \ell| \sim C\rho^n$. Pour déterminer la limite ℓ , on pourra retenir l'approximation que constitue la dernière itération, avec quelques précautions, ou s'aider de la fonction `fsolve`.

Approfondissement

Méthode de Newton

- À deux murs verticaux parallèles deux échelles sont adossées en les croisant. La première échelle mesure 3m de long et la seconde 2m. Il se trouve que les échelles se croisent à une hauteur de 1m. Écrire la méthode de Newton (rappelée ci-dessous) pour déterminer une approximation de la distance d entre les deux murs.



Théorème. Soient $f \in \mathcal{C}^2([a, b])$ et $r \in]a, b[$ tels que $f(r) = 0$ et $f'(r) \neq 0$. Alors il existe $\delta > 0$ tel que pour tout $x_0 \in [a, b]$ vérifiant $|x_0 - r| \leq \delta$, la suite définie par la récurrence :

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad n \geq 0,$$

est bien définie et à valeurs dans $[a, b]$, et telle qu'elle converge vers r . De plus, il existe une constante $C > 0$ telle que pour tout $x_0 \in [a, b]$ vérifiant $|x_0 - r| \leq \delta$ et tout $n \in \mathbb{N}$:

$$|x_{n+1} - r| \leq C|x_n - r|^2.$$

Diagramme fractal de Feigenbaum

Étant donné un paramètre $\mu \in [0, 4]$, on considère les suites récurrentes de la forme :

$$\begin{aligned} x_0 &\in]0, 1[, \\ x_{n+1} &= \mu x_n(1 - x_n), \quad n \in \mathbb{N}. \end{aligned}$$

- Pour les valeurs de μ successivement $\{2.9, 3.3, 3.5\}$, tracer les itérations issues de $x_0 = 0.1$, à la manière des tracés en « escargots » précédents. On pourra dans le même temps tracer les courbes représentatives de f , $f \circ f$ et $f \circ f \circ f \circ f$ et exploiter les résultats graphiques obtenus pour commenter la dynamique de la récurrence.

- Tracer en fonction de μ , l'ensemble des valeurs d'adhérences de la suite considérée. Pour ce faire, on pourra tracer un pixel dans le plan (μ, x_n) pour chaque valeurs $(x_n)_{N \leq n \leq M}$ où N et M sont des entiers suffisamment grands par une commande de la forme :

```
| plot(MU, XN, '.', 'markersize', 0);
```

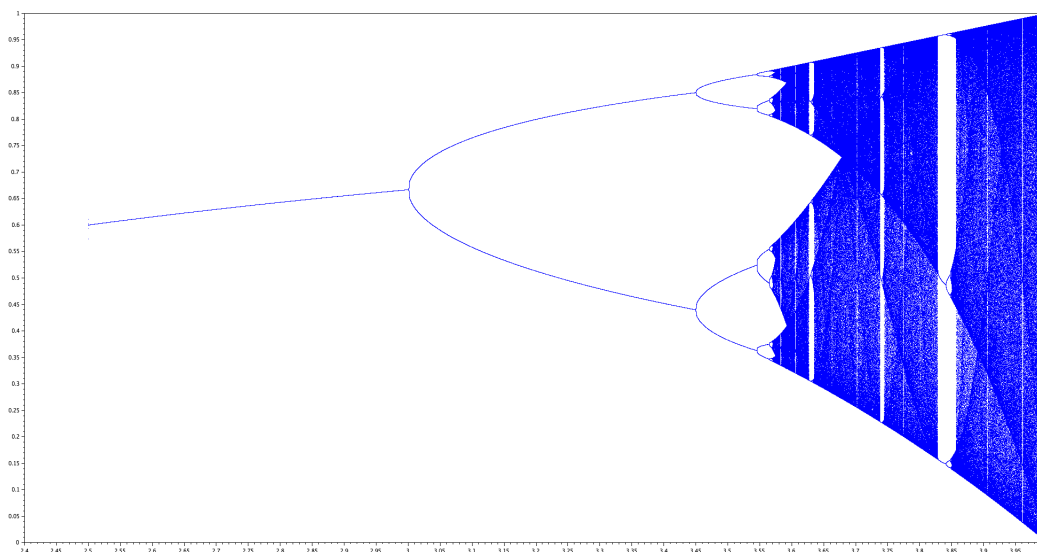


FIGURE 1 – Diagramme de bifurcation pour la suite logistique.

Résolution d'EDO

Échauffement

- Fonction `ode`. Consulter l'aide de Scilab et adapter le premier exemple proposé de manière à résoudre numériquement l'équation différentielle ordinaire suivante.

On commencera par définir la fonction $f(t, y)$ du problème de Cauchy.

$$\begin{cases} y(0) = 0 \\ y'(t) = \frac{1}{2}(1 - y(t)) \left(\cos^2 t + \frac{1}{1 + 5y(t)^2} \right), \quad t \geq 0. \end{cases}$$

On tracera la trajectoire solution $y(t)$ sur l'intervalle $t \in [0, 15]$ et on superposera sur une même figure les trajectoires obtenues pour les données initiales $y(0) = 1$ et $y(0) = 2$ respectivement.

- Programmer la résolution du problème précédent en utilisant la méthode d'Euler explicite de pas $h = 0.5$

$$\begin{cases} y_0 = 0 \\ t_{n+1} = t_n + h \\ y_{n+1} = y_n + hf(t_n, y_n). \end{cases}$$

Superposer aux précédentes la trajectoire approchée ainsi obtenue. Faire varier le pas h , observer la convergence de la méthode d'Euler vers la trajectoire solution, lorsque h tend vers 0.

Exercices

Le pendule pesant et l'oscillateur harmonique

On souhaite examiner comparativement les comportements qualitatifs des solutions du problème du pendule pesant :

$$\ddot{\theta} + \omega^2 \sin \theta = 0, \quad (\text{PP})$$

et de sa version linéarisée au voisinage de $(\theta, \dot{\theta}) = (0, 0)$, l'oscillateur harmonique :

$$\ddot{\theta} + \omega^2 \theta = 0. \quad (\text{OH})$$

Par commodité, on a ici noté $\dot{\theta}$ au lieu de $\frac{d\theta}{dt}$ et $\ddot{\theta}$ au lieu de $\frac{d^2\theta}{dt^2}$.

Pour la suite, on supposera que $\omega = 1$ (ce qui est possible en dilatant l'échelle de temps convenablement).

- Reformuler chacun de ces deux équations différentielles d'ordre 2 sous la forme d'une équation différentielle d'ordre 1 dont l'inconnue est la fonction y à valeurs dans \mathbb{R}^2 avec $t \mapsto y(t) = (\theta(t), \dot{\theta}(t)) \in \mathbb{R}^2$. Programmer les fonctions PP et OH telles que l'équation du pendule pesant se réécrive $y'(t) = \text{PP}(t, y(t))$ et celle de l'oscillateur harmonique $y'(t) = \text{OH}(t, y(t))$.

- À l'aide du solveur `ode` résoudre les deux problèmes (PP) et (OH) pour la donnée initiale $y(0) = (\pi/2, 0)$. Tracer sur une même figure l'évolution de la quantité θ en fonction du temps, pour chacune des deux équations. Qu'observe-t-on ?

- Sur une nouvelle figure tracer les champs de vecteur dans le plan de phase en suivant la syntaxe suivante :

```
xf = linspace(-2*pi, 2*pi, 20);
yf = linspace(-5, 5, 20);
scf(2); clf
subplot(1, 2, 1)
fchamp(PP, 0, xf, yf, arfact=0.5)
subplot(1, 2, 2)
fchamp(OH, 0, xf, yf, arfact=0.5)
```

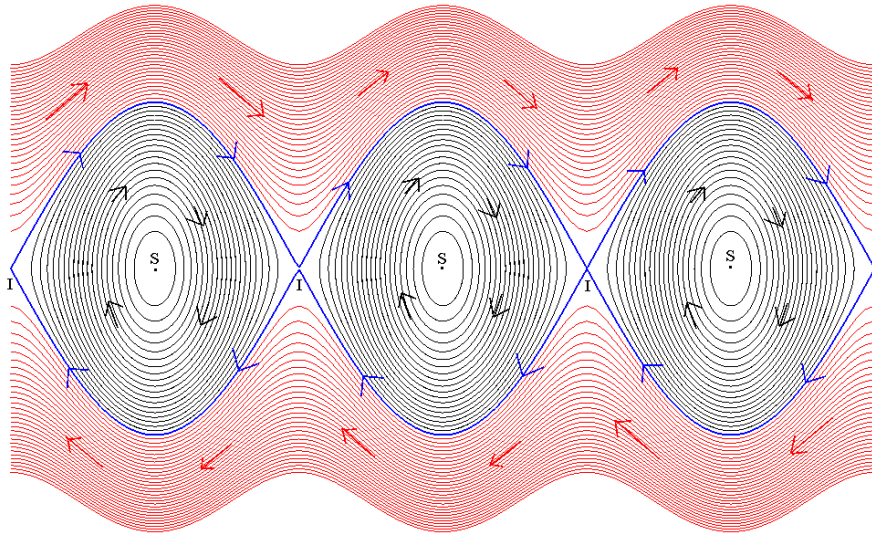
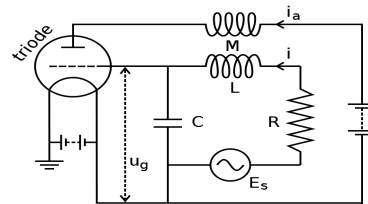


FIGURE 2 – Portrait de phase du pendule (OH) (extrait de Wikipedia)

- Superposer à la figure précédente quelques trajectoires pour chaque problème considéré, obtenues avec ode.
- Étudier de même le comportement qualitatif du système de Van der Pol modélisant un circuit électrique comportant une triode oscillante :

$$\begin{cases} x' = y - x^3/3 + x \\ y' = -x. \end{cases}$$



Quelques liens et références :

Pour l'usage de scilab

<https://www.scilab.org>

<https://cermics.enpc.fr/~bl/PS/pinson.pdf>

http://gradus-ad-mathematicam.fr/documents/300_Directeur.pdf

https://old.i2m.univ-amu.fr/agreg-interne-maths/doku.php?id=f._hubert

Calcul scientifique

Introduction à Scilab - Exercices pratiques corrigés d'algèbre linéaire, Allaire Grégoire, Kaber Sidi Mahmoud, Ellipses, Mathématiques Université, 2002.

Méthodes d'approximation - Équations différentielles - Applications Scilab - Niveau L3, Guerre-Delabrière Sylvie, Postel Marie, Ellipses, Mathématiques Université, 2004.

Calcul scientifique, Cours exercices corrigé et illustrations en Matlab et Octave A.Quarteroni, F.Saleri, P.Gervasio, Springer, 2010.