
Transformation de Arnold

(1) Commande input

La commande `input` permet d'afficher un message à l'écran et d'attendre que l'utilisateur réponde quelque chose, par exemple tapez:

```
toto=input('Quel est votre nom','s')
```

Répondez à la question **sans mettre d'accents ou de caractères spéciaux** et vérifiez le contenu de la variable `toto`. Vous pouvez utiliser cette variable en tapant par exemple:

```
'Bonjour '+toto+', comment allez vous ?'
```

Les apostrophes servent à prévenir scilab que l'on va utiliser du texte, le signe `+` sert à **concatener** du texte et le symbole `'s'` dans la fonction `input` précise que le résultat sera du texte.

(2) Nous utiliserons souvent cette commande `input` pour arrêter un programme et demander à l'utilisateur si il veut continuer. Par exemple le programme suivant :

```
function r=idiot()  
r='y'  
i=0  
while r=='y'  
r=input(string(i)+'eme fois, On continue ? repondre y','s')  
i=i+1  
end  
endfunction
```

(3) Commandes `save` et `load`

Ces commandes servent à sauver et à recharger des variables. Nous les utiliserons lors des examens pour contrôler vos résultats. Essayez:

```
X=1  
save('td4',X)  
clear  
X  
load('td4')  
X
```

Le programme a enregistré la variable demandée dans le fichier `td4`. La commande `clear` permet d'effacer les variables. Remarquez que la commande `load` choisit toute seule le nom de la variable, si vous ne savez pas quel est le nom de la variable, utilisez la commande `who`.

- (4) Dans le répertoire au-dessus de votre répertoire par défaut se trouve un fichier `chat`, chargez le. Il contient une variable `X` qui est une liste de points. Les abscisses sont dans la première ligne et les ordonnées dans la deuxième. Pour la visualiser utilisez la commande:

```
plot2d(X(1,:),X(2,:),style=-4,rect=[0,0,1,1])
```

- (5) On étudie la transformation de Arnold. C'est la transformation du carré $[0, 1] \times [0, 1]$ donnée par:

$$f(x, y) = (x + y \text{ mod}(1), x + 2y \text{ mod}(1))$$

Où *mod* est le "modulo" (le reste de la division euclidienne). Sous Scilab, $a \text{ mod}(b)$ est donné par la commande `pmodulo(a,b)`.

- (6) Soit g la transformation $g(x, y) = (x + y, x + 2y)$. Montrer que g est la composition de 2 "transformations de cisaillement"

$$h_1(x, y) = (x + y, y) \text{ et } h_2(x, y) = (x, x + y).$$

Tracer l'image du carré $[0, 1] \times [0, 1]$ par h_1, h_2, g en déduire l'action de f sur ce carré.

- (7) On travaillera avec des matrices X de taille $2, k$ qui représente k points, l'abscisse du k^{eme} point étant le coefficient $X(1, k)$ et l'ordonnée le coefficient $X(2, k)$. Ecrire une fonction `henon0(X)` qui prend une telle liste de points et qui rend la liste de points `Y`, transformés une fois par l'application de Hénon.

- (8) Ecrire une fonction `henon(X)` qui applique plusieurs fois la fonction `henon0` et qui affiche le résultat. On s'arrêtera après chaque itération en demandant à l'utilisateur si il veut continuer.

- (9) Utilisez `henon(X)` en faisant 15 itérations, que se passe-t-il? Et avec `henon(Y)` où

```
Y=X+rand(X)/124
```