Feuille de Td numéro 3

Exercice 1

On va dans ce TP revenir sur les boucles grâce à de *simulations de systèmes discrets.*

On considère le système dynamique discret suivant :

$$x_{n+1} = 3\,x_n + 1 \quad (x_n \; impair)$$

$$x_{n+1} = \frac{x_n}{2} \quad (x_n \; pair)$$

On simulera ce système en partant d'une valeur x_1 donnée par l'opérateur (voir fonction input de MATLAB). Le calcul de x_{n+1} à partir de x_n sera fait dans une fonction que vous programmerez et qui sera appelée par le script principal.
Dans une première version du programme, vous fixerez au préalable le nombre total itérations (structure « for »). Dans une deuxième version, vous fixerez vous-mêmes un test d'arrêt (structure « while ») en fonction des observations que vous aurez faites avec la première version.

Exercice 2

Les *automates cellulaires* sont des systèmes dynamiques discrets à la fois dans l'espace et dans le temps. Ils sont composés d'un réseau carré de cellules (à 1D ou à 2D), chaque cellule pouvant prendre la valeur 0 ou 1.
On part d'une configuration initiale de cellules, et à chaque pas de temps on calcule la valeur suivante de toutes les cellules en appliquant une *règle* : la valeur de chaque cellule au temps suivant dépend de la valeur de la cellule elle-même et de la valeur de ses 2 (à 1D) ou 8 (à 2D) voisines.

Commençons par un exemple à 1D.
On commencera par tirer au hasard un tableau de N valeurs de 0 et de 1 à l'aide de la fonction rand (attention cette fonction génère des nombres aléatoires *réels* entre 0 et 1, il faut donc manipuler le résultat pour trouver soit 0 soit 1 uniquement!).

On programmera ensuite l'évolution pour la *règle* suivante :

$$0[0]0 \rightarrow [1]$$
$$1[0]0 \rightarrow [0]$$
$$0[0]1 \rightarrow [0]$$
$$1[0]1 \rightarrow [1]$$
$$0[1]0 \rightarrow [1]$$
$$1[1]0 \rightarrow [0]$$
$$0[1]1 \rightarrow [1]$$
$$1[1]1 \rightarrow [0]$$

Quel problème se pose aux bords ?
On le règlera en ne faisant *pas* évoluer les deux cellules du bord (1ère version), puis en appliquant des conditions au bord périodiques (i.e. la voisine de gauche de la cellule la plus à gauche sera la cellule la plus à droite, et la voisine de droite de la cellule la plus à droite est la cellule la plus à gauche).
L'évolution sera programmée à l'aide d'une *fonction*.
On fixera a priori le nombre d'itérations.

Le résultat sera stocké dans une matrice, dont les ligne successives représenteront les états successifs de notre tableau 1D.
Pour finir on visualisera le résultat à l'aide de la fonction imshow.

Exercice 3

Automate cellulaire à 2D. On se donne une matrice carrée initiale de taille *PxP*, remplie de 0 et de 1 (même problème que précédemment). On fait évoluer ce système à l'aide de la règle suivante. Si *q* est le nombres de voisines valant 1, on fera :

$$q=3:([0]\rightarrow[1],[1]\rightarrow[1])$$
$$q=2:([0]\rightarrow[0],[1]\rightarrow[1])$$
$$q\neq 2,3:([0]\rightarrow[0],[1]\rightarrow[0])$$

Programmer (toujours à l'aide d'une fonction appelée par un script) l'évolution de ce système. On fixera à nouveau a priori le nombre total itérations. On représentera par imshow l'image de la matrice à chaque pas de temps. Regardez le film.