Chauffe ENSAM – 1/2

Mercredi 31 mai 2017

1 Proies et prédateurs (Laure André)

On s'intéresse à un système « proies vs prédateurs », modélisé par :

$$\begin{cases} x'(t) = ax(t) - bx(t)y(t) \\ y'(t) = -cy(t) + dx(t)y(t) \end{cases}$$

avec x et y les nombres respectifs de proies et prédateurs. On prendra comme constantes :

$$(a, b, c, d) = (2, 0.3, 1.5, 0.5)$$

1. Déterminer F pour que le système devienne, en notant $X = (x, y)$: $X'(t) = F(X(t), t)$. Écrire en Python une telle fonction.
2. Pour différentes conditions initiales, calculer avec `odeint` puis représenter la solution du système sur $[0, 2]$ puis $[0, 5]$.
   On pourra prendre par exemple :

$$(x_0, y_0) \in \{(10, 10); (7, 7); (5, 7; (3, 7)\}$$

2 Décomposition en base 2 (récolte 2015)

Pour $n \in \mathbb{N}$, $T(n)$ désigne la somme modulo 2 des chiffres dans la représentation binaire de n.

1. Écrire une fonction calculant la représentation en base 2 d'un entier donné en paramètre.
2. En déduire une fonction calculant $T(n)$, avec n donné en paramètre.
3. Écrire une fonction récursive calculant $T(n)$.

3 Des probabilités (RMS 1371)

1. Préciser ce que font ces commandes :

```
>>> from numpy.random import rand
>>> LR = rand(6)
>>> LR < 0.5
>>> 1*(LR < 0.5)
```

2. Créer une fonction `tirage` prenant en argument un entier $n > 0$ et renvoyant une liste de n tirages successifs indépendants suivant chacun une loi de Bernoulli de paramètre 1/2.
3. Calculer numériquement l'espérance du nombre de tirages ayant donné 1.
4. Créer une fonction prenant en entrée un motif et une suite de tirages (l'un et l'autre sont des listes de 0/1) et renvoyant le temps d'attente du motif dans la suite de tirages, c'est-à-dire le premier indice de la suite de tirages où on a observé le motif.
   Par exemple, `attente([1,0,1,1], [0,0,1,0,0,1,0,1,1])` doit renvoyer 8.

4 Sous-matrices (Alice Tincrès, 2015)

On rappelle qu'on peut récupérer les dimensions d'une matrice (`array` de `numpy`) `M` grâce à l'attribut `A.shape` (qui est un couple). Par ailleurs, on dispose du slicing permettant de récupérer une sous-matrice comme pour les listes : `A[ a:b , c:d]` extrait les lignes de a à b (exclu) et colonnes de c à d (exclu). Enfin, l'égalité de deux matrices peut être testée via la fonction `array_equal` de `numpy`

1. Créer une matrice aléatoire A_0 d'entiers de taille $(9, 9)$ à l'aide de `randint` de `numpy`. Même chose avec $B_0 \in \mathcal{M}_2(\mathbb{N})$.
2. Créer une fonction prenant deux matrices A et B en entrée et testant si B est ou non une sous-matrice de A (au sens qui plaira au candidat !).
3. Écrire une fonction prenant en entrée deux matrices C et D de même taille et retournant la somme des $(c_{i,j} - d_{i,j})^2$.
4. Écrire une fonction prenant en entrée deux matrices A et B et retournant la sous-matrice de A la plus proche de B au sens précédent.

5 Un joli dessin

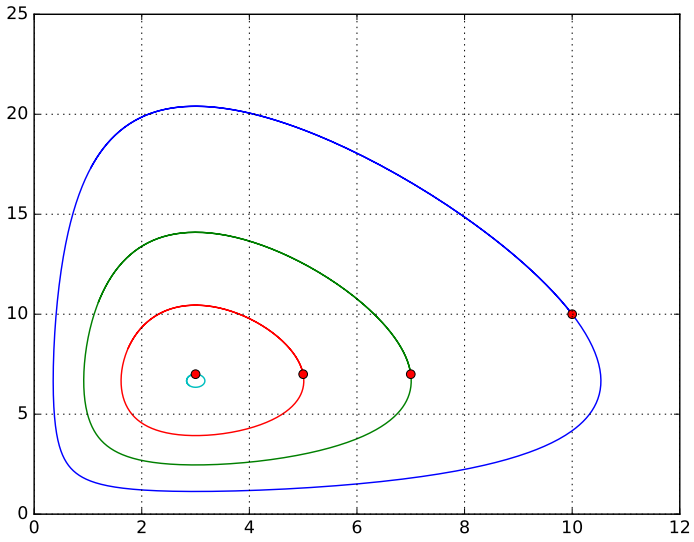

FIGURE 1 – Quelques trajectoires