

# Le jeu de la vie

Le jeu de la vie est une simulation sensée reproduire de façon simple l'évolution d'un groupe d'organismes vivants.

Chaque cellule représentée par sa position dans une matrice est soit vivante, soit morte. On considère la matrice grande mais non infinie. On se propose donc d'adopter des conditions périodiques aux limites pour limiter les effets de bords.

Entre deux instants, l'état d'une cellule évolue en fonction du nombre de voisins vivants (parmi ses 8 voisins):

- Toute cellule ayant moins de deux voisins meurt d'isolement
- Toute cellule ayant 2 ou 3 voisins survit jusqu'à l'instant suivant
- Toute cellule ayant plus de 3 voisins meurt de surpopulation

Par ailleurs, les cellules peuvent se reproduire. On peut considérer que :

- Toute cellule morte ayant exactement 3 cellules vivantes a une probabilité  $p$  de devenir vivante.

Dans un premier temps on prendra  $p=1$ .

Implémenter cette simulation.

Pour ce faire, on souhaite implémenter une classe Simulation.

Toutes les règles seront implémentées dans une classe Cellules qui sera appelée par la classe simulation. On se propose de représenter la simulation avec matplotlib.

On peut utiliser ce code pour faire varier un plot dans le temps (un clic = avancer d'un pas de temps)

```
def onclick(event):
    #clear frame
    plt.clf()
    plt.plot(np.random.rand(10)) #inform matplotlib of the new data
    plt.draw() #redraw

fig,ax=plt.subplots()
ax.plot(np.random.rand(10))
fig.canvas.mpl_connect('key_press_event',onclick)
plt.show()
plt.draw()

def onclick(event):
    #clear frame
    plt.clf()
    plt.plot(np.random.rand(10)) #inform matplotlib of the new data
    plt.draw() #redraw
    fig,ax=plt.subplots()
    ax.plot(np.random.rand(10))
    fig.canvas.mpl_connect('key_press_event',onclick)
    plt.show()
    plt.draw()
```

Cette simulation présente déjà des résultats d'évolution intéressants. Il y a cependant deux limites évidentes :

- Il n'y a qu'une espèce, ce qui empêche fatallement tout mécanisme de concurrence entre espèces
- L'énergie ne se conserve pas, ou en tous cas rien n'explique la variation de l'énergie totale du système.

# Modélisation simple d'un écosystème

On souhaite donc améliorer cette simulation pour rendre compte de l'évolution d'un écosystème. Les mécanismes sont bien plus complexes. Nous partirons de mécanismes très simples que nous ferons évoluer.

Considérons un écosystème constitué de loups, de renards, de lapins, et de carottes

:

- On associe à chaque être vivant et à chaque instant une quantité d'énergie.
- Tous les animaux se déplacent. Un déplacement consomme de l'énergie. Considérons pour le moment que les animaux se déplacent (ou non) aléatoirement sur une des 4 cases voisines.
- Un prédateur a une probabilité(prédateur, proie) de manger sa proie s'ils sont sur la même case. Manger une proie revient à ingurgiter un ratio  $r_{digestion}(pred, proie)$  de l'énergie de la proie. A chaque instant un prédateur ne peut ingurgiter qu'une quantité de nourriture inférieure à  $\text{max\_digestion}(pred, proie)$  :
  - Les loups mangent les renards et les lapins
  - Les renards mangent les lapins
  - Les lapins mangent les carottes
- Un animal qui n'a plus d'énergie meurt
- A chaque instant, un animal qui a au moins un voisin de la même espèce a une probabilité  $p_{repro}$  de donner naissance à un enfant
- A chaque instant, les carottes poussent ce qui fait croître leur énergie d'une certaine quantité. On suppose donc pour le moment que les carottes poussent en consommant un stock illimité d'énergie.

Implémenter cette nouvelle simulation en créant les classes pertinentes et les héritages adaptés.

Faire varier tous les paramètres afin d'arriver à un point de pseudo stabilité après un régime transitoire.

# Pour aller plus loin

On pourra implémenter plus tard les règles suivantes :

- Le déplacement d'un animal n'est plus aléatoire mais un prédateur poursuit sa proie
- Les animaux ont des vitesses de déplacement différentes
- Un animal fatigué se déplace moins vite
- Un animal fatigué chasse moins bien
- Un animal fatigué se reproduit moins bien

Par ailleurs, la simulation ne prend pas en compte le vieillissement des populations. Certains critères pourraient être ajoutés.

Enfin, la simulation sus-citée ne prend pas en compte la conservation de l'énergie. Cela signifie que les carottes ont une ressource infinie en nutriment pour pousser. Si cela est vrai pour un court temps de simulation, cette hypothèse rencontre vite des limites.

Une solution à ce problème pour les grandes échelles de temps pourrait être de prendre en compte la décomposition des matières organiques (carcasses et déjections), en introduisant une nouvelle espèce : les champignons.

- Les carottes ont désormais besoin de champignons pour pousser. Plus un terrain est riche en champignons, plus les carottes poussent vite. Cependant, les carottes produisent plus d'énergie qu'elles n'en prélève dans le sol car la photosynthèse est à prendre en compte.
- Les champignons décomposent les matières organiques avec un certain rythme.
- Enfin, il pourrait sembler pertinent d'appliquer un noyau gaussien sur la distribution des matières organiques à chaque instant, pour modéliser la diffusion des nutriments dans le sol.