

# Le modèle proie-prédateur

## Énoncé

Lorsque deux populations animales interagissent, il est fréquent que leurs taux d'accroissement respectifs soient imbriqués. C'est le cas notamment avec les lynx (prédateur) et les petits rongeurs (proies) sur un territoire clos du Canada.

### A) Le modèle discret

On note  $L_n$  l'effectif des lynx après  $n$  mois et  $r_n$  celui des rongeurs ; on suppose alors que :

$$(E) : \begin{cases} L_{n+1} = 0,96 \times L_n + 0,000\,05 \times L_n \times r_n \\ r_{n+1} = 1,05 \times r_n - 0,002 \times L_n \times r_n \end{cases}$$

a) En l'absence de prédateurs, comment évolue l'effectif des proies ?

Dans ce cas, exprimer  $r_n$  en fonction de  $n$  et  $r_0$ .

.....  
 .....  
 .....

b) En l'absence de proie, comment évolue l'effectif des prédateurs ?

Dans ce cas, exprimer  $L_n$  en fonction de  $n$  et  $L_0$ .

.....  
 .....  
 .....

### B) Simulation du modèle discret à l'aide du tableur

Ouvrir une feuille de calcul sur tableur.

1) Placer les valeurs de  $n$ , de 0 à 1 000 dans la colonne **A**, les valeurs  $r_0 = 2\,000$  et  $L_0 = 40$  en **B2** et **C2**.

2) Introduire les formules (voir (E)) dans **B3** et **C3** permettant de calculer  $r_1$  et  $L_1$ .

3) Étirer les formules jusqu'à la ligne 1 002

4) Faire représenter les deux graphiques (deux graphiques différents) suivants :

- ★ les points de coordonnées  $(n ; r_n)$  ;
- ★ les points de coordonnées  $(n ; L_n)$ .

5) Faire représenter la courbe reliant les points de coordonnées  $(r_n ; L_n)$  sur un autre graphique.

	A	B	C
1	$n$	$r_n$	$L_n$
2	0	2000	40
3	1		
4	2		
5	3		
6	4		

## C) Simulation du modèle discret à l'aide d'un logiciel de programmation

Charger le logiciel « EduPython ». Ouvrir un nouveau fichier (modèle : "lycée").

### 1) Taper les commandes

```
from lycee import *
# On crée deux listes nulles contenant 1001 termes chacune
L=[0]*1001
r=[0]*1001
# Initialisation
L[0]=40
r[0]=2000
# On calcule tous les termes des deux listes
for n in range(1000): # de 0 à 999 soit 1000 termes
    L[n+1],r[n+1]=0.96*L[n]+0.00005*L[n]*r[n],1.05*r[n]-0.002*L[n]*r[n]
```

### 2) Pour représenter les courbes représentant $(n ; r_n)$ , $(n ; L_n)$ et $(r_n ; L_n)$ , taper les commandes

```
# On construit les nuages de points (n;r_n); (n;L_n) et (r_n;L_n)
n=np.linspace(0,1000,1001) # Liste des valeurs de n de 0 à 1000, soit 1001 termes
repere.subplot(231)
repere.xlabel(r'$n$')
repere.ylabel('Proie '+r'$r_n$')
repere.title('Evolution de la population des proies',fontsize=10)
repere.plot(n,r,'r.')

repere.subplot(233)
repere.xlabel(r'$n$')
repere.ylabel('Prédateur '+r'$L_n$')
repere.title('Evolution de la population des prédateurs',fontsize=10)
repere.plot(n,L,'g.')

repere.subplot(235)
repere.xlabel('Proie '+r'$r_n$')
repere.ylabel('Prédateur '+r'$L_n$')
repere.title('Evolution de la population des proies et des prédateurs',fontsize=10)
repere.plot(r,L,'b.')
repere.show()
```

Exécuter en sélectionnant la flèche verte dans la barre des outils.

