

# Projet Numérique Master Physique Recherche Fondamentale : Modèle proie-prédateur

COLOMBIER Marc-Antoine et FERRY Morgan

6 mai 2024

## Table des matières

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Introduction</b>  | <b>1</b>  |
| <b>2</b> | <b>Une première approche : Les équations de Lotka-Volterra</b> | <b>2</b>  |
| 2.1      | Modèle théorique . . . . .                                     | 2         |
| 2.2      | Analyse Numérique . . . . .                                    | 3         |
| 2.2.1    | Méthode d'intégration . . . . .                                | 3         |
| 2.2.2    | Modèle classique . . . . .                                     | 4         |
| 2.2.3    | Modèle avec influence extérieur . . . . .                      | 5         |
| 2.2.4    | ouverture sur un 3ème modèle : Modèle avec effort . . . . .    | 9         |
| <b>3</b> | <b>Une seconde approche : Modèle 2D</b>                        | <b>10</b> |
| 3.1      | Modélisation physique . . . . .                                | 10        |
| 3.2      | Simulation numérique . . . . .                                 | 11        |
| 3.2.1    | Structure du programme . . . . .                               | 11        |
| 3.2.2    | Résultats . . . . .  | 13        |
| <b>4</b> | <b>Conclusion</b>  | <b>18</b> |

## 1 Introduction

Un modèle proie-prédateur est un modèle simulant la dynamique de deux espèces, une étant la proie et l'autre le prédateur. Le modèle de prédation de Lotka-Volterra, proposé dans les années 1920, est le premier modèle mettant en jeu un couple d'équations différentielles utilisé pour décrire la démographie du lynx et du lièvre des neiges. L'objectif de ce projet est de résoudre numériquement les équations de Lotka-Volterra puis de comparer ces résultats à un modèle 2D simulant l'évolution des deux espèces. Les programmes seront exclusivement écrits en Python. Marc-Antoine s'est occupé de la résolution numérique des équations de Lotka-Volterra et Morgan s'est occupé de la conception et de la résolution numérique du modèle 2D.

## 2 Une première approche : Les équations de Lotka-Volterra

### 2.1 Modèle théorique

Le modèle de Lotka-Volterra est un jeu de deux équations décrivant l'évolution des populations l'une influant sur l'autre. Le modèle le plus simple que l'on puisse imaginer est celui-ci :

$$\begin{cases} \frac{\partial x}{\partial t}(t) = x(t)(\alpha - \beta y(t)) \\ \frac{\partial y}{\partial t}(t) = y(t)(\delta x(t) - \gamma) \end{cases}$$

où :

$\alpha$  est le taux de reproduction intrinsèque de la proie

$\beta$  le taux de mortalité des proies dû aux prédateurs

$\delta$  le taux de reproduction des prédateurs lié aux proies mangées

$\gamma$  le taux de mortalité intrinsèque aux prédateurs

Ces paramètres étant des taux, ils sont tous compris entre 0 et 1. Ce jeu d'équation décrit l'évolution des proies  $x(t)$  et des prédateurs  $y(t)$  au cours du temps. Ce modèle peut être traité analytiquement grâce à une étude de stabilité. Pour cela, on peut écrire la Jacobienne de ce jeu d'équation.

$$J(x(t), y(t)) = \begin{pmatrix} \alpha - \beta y(t) & -\beta x(t) \\ \delta y(t) & \delta x(t) - \gamma \end{pmatrix}$$

Le système admet deux points fixes :  $(x(t), y(t)) = (0, 0)$  et  $(x(t), y(t)) = (\frac{\alpha}{\beta}, \frac{\gamma}{\delta})$ .

- Dans le cas du premier point fixe  $(0, 0)$  on obtient la matrice Jacobienne suivante :

$$J(0, 0) = \begin{pmatrix} \alpha & 0 \\ 0 & -\gamma \end{pmatrix}$$

La matrice est déjà diagonale de valeurs propres  $\{\alpha, -\gamma\}$ .  $\alpha$  et  $\gamma$  étant des réels compris entre 0 et 1, les deux valeurs propres de la Jacobienne sont toujours de signes opposés et le point  $(0, 0)$  n'est donc pas stable. Il est donc presque impossible que les deux espèces s'éteignent.

- Dans le cas du second point fixe  $(\frac{\alpha}{\beta}, \frac{\gamma}{\delta})$ , la Jacobienne s'écrit :

$$J(\frac{\gamma}{\delta}, \frac{\alpha}{\beta}) = \begin{pmatrix} 0 & -\beta \frac{\gamma}{\delta} \\ \delta \frac{\alpha}{\beta} & 0 \end{pmatrix}$$

On cherche alors à la diagonaliser afin d'obtenir les valeurs propres dont le signe nous donnera la stabilité du point. On trouve :

$$J(\frac{\gamma}{\delta}, \frac{\alpha}{\beta}) = \begin{pmatrix} i\sqrt{\alpha\gamma} & 0 \\ 0 & -i\sqrt{\alpha\gamma} \end{pmatrix}$$

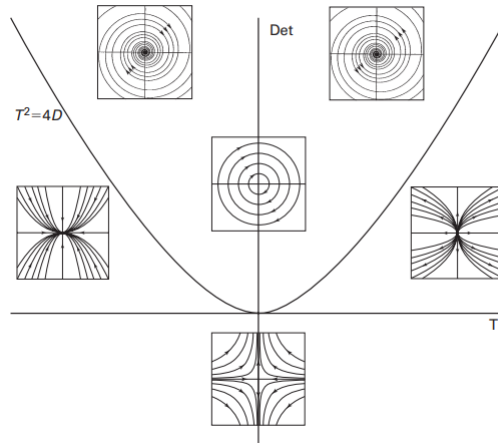


FIGURE 1 – Classification des points fixes d'un système dynamique bi-dimensionnel selon la trace et le déterminant de la Jacobienne.

On peut déduire plusieurs choses de cette matrice.

Tout d'abord, on voit que les valeurs propres sont imaginaires pures. De plus, la trace est nulle et le déterminant  $Det(J(\frac{\gamma}{\delta}, \frac{\alpha}{\gamma})) = \alpha\gamma$  positif. On peut donc dire d'après la figure 1 que l'on a un cycle limite autour de ce point et que l'évolution des populations doit osciller autour de ce point fixe elliptique.

De plus, on peut observer que la trace est toujours nulle ( $\forall\alpha, \forall\beta, \forall\delta, \forall\gamma$ ), mais que le déterminant dépend du produit  $\alpha\delta$ . Ainsi, la taille du cycle limite dépend exclusivement du taux de reproduction (intrinsèque) des proies et du taux de reproduction des prédateurs.

## 2.2 Analyse Numérique

### 2.2.1 Méthode d'intégration

Pour résoudre le système d'équations, nous devons les intégrer. Pour ce faire, nous avons d'abord pensé à la méthode la plus simple en utilisant la méthode d'Euler. Cependant, lorsque nous utilisons cette méthode d'intégration, nous obtenions des cycles divergents alors que nous étions censés tomber sur des cycles stables. Nous avons donc réfléchi à un autre moyen pour résoudre notre système d'équations. Ainsi, nous nous sommes tournés vers la méthode de Runge-Kutta.

```
def RunchKutta(x,y): #Runch-Kutta Method
    x1,y1 = f(x,y)
    x2,y2 = f(x + x1 * dt * 0.5, y + y1 * dt * 0.5)
    x3,y3 = f(x + x2 * dt * 0.5, y + y2 * dt * 0.5)
    x4,y4 = f(x + x3 * dt, y + y3*dt)
    X = x + dt * (x1/6 + x2/3 + x3/3 +x4/6)
    Y = y + dt * (y1/6 + y2/3 + y3/3 +y4/6)
    return X,Y
```

FIGURE 2 – Caption

Cette méthode d'intégration est plus longue que la méthode d'Euler en terme de calcul numérique mais est beaucoup plus précise. Ainsi, tous les codes que nous avons écrits utilisent cette méthode.

### 2.2.2 Modèle classique

Nous avons cherché à résoudre les équations de Lotka-Volterra sur Python. Pour cela, nous avons utilisé une méthode d'intégration, la méthode de Runge-Kutta, et avons joué sur les différents paramètres de notre système.

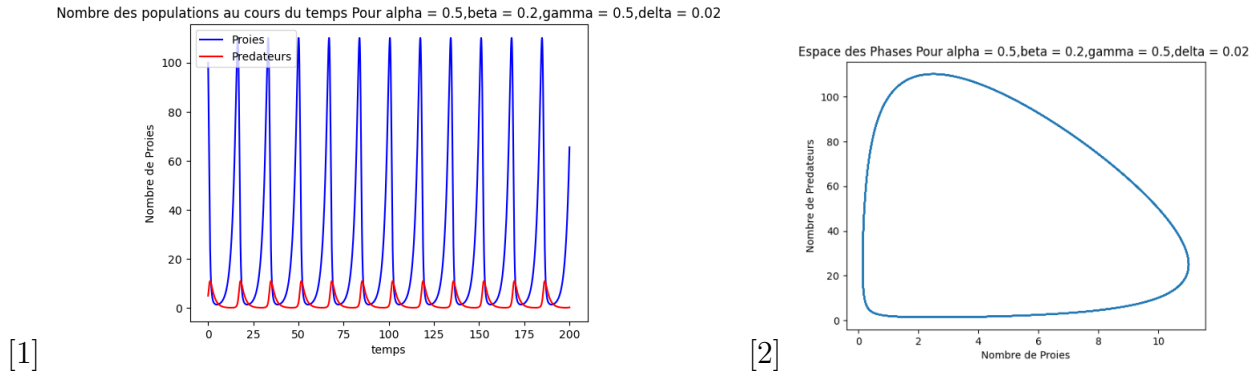


FIGURE 3 – Evolution des populations pour  $\alpha = 0.5, \beta = 0.2, \gamma = 0.5, \delta = 0.02$

La figure 3 nous permet de déterminer une période caractéristique de l'évolution des espèces. Pour ces paramètres, on obtient la même période pour les deux espèces qui est de 16.838 (en unité de temps arbitraire). Nous pouvons modifier les paramètres et obtenir une évolution différente.

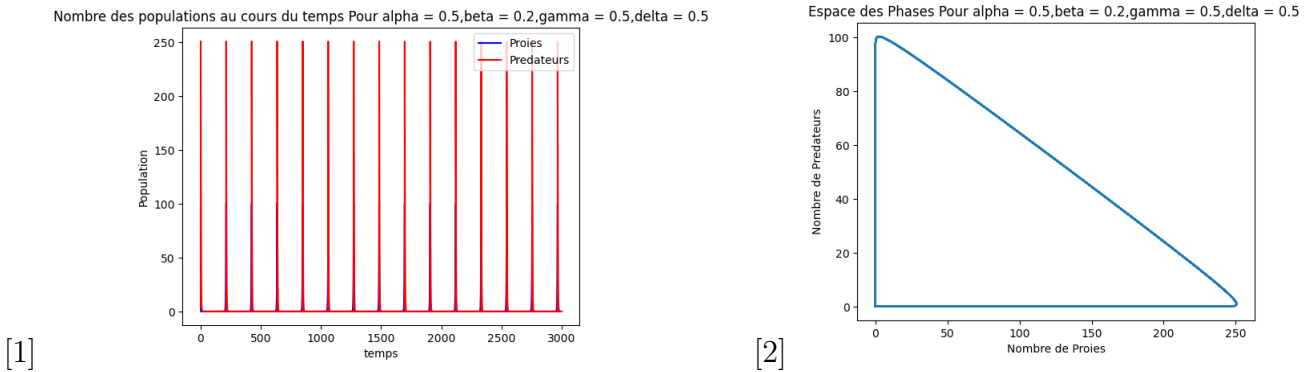


FIGURE 4 – Evolution des populations pour  $\alpha = 0.5, \beta = 0.2, \gamma = 0.5, \delta = 0.5$

On peut remarquer que la période d'oscillation dépend de nos paramètres.

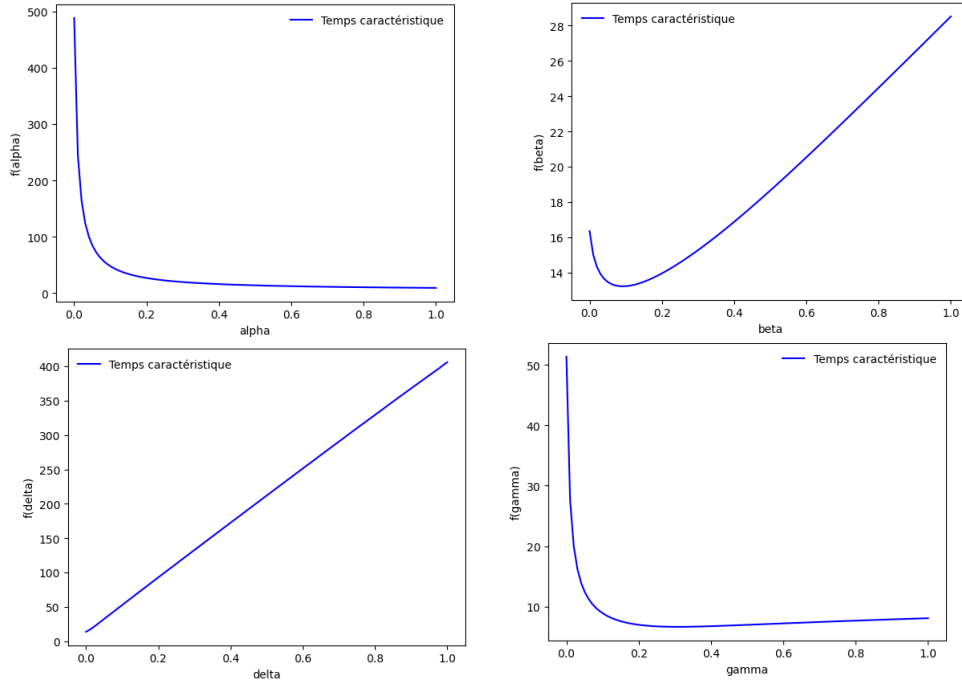


FIGURE 5 – Evolution du temps caractéristique en fonction des différents paramètres qui varient un à un

Comme on peut le voir, la dépendance de la période d'oscillation est différente pour chacun des paramètres, mais si on se place dans le cas où les paramètres sont dans l'intervalle  $[0,3; 1]$ , alors on peut approximer nos dépendances par des droites.

$$d\tau = -59.516 \times d\alpha + 166.787 \times d\beta - 67.071 \times d\gamma + 952.68 \times d\delta \quad (1)$$

### 2.2.3 Modèle avec influence extérieur

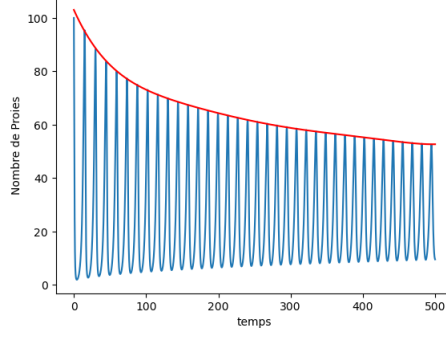
Le modèle précédent décrivait l'évolution d'une population "fermée", c'est à dire comme un système fermé ou isolé. Nous avons voulu savoir s'il était possible de décrire l'évolution d'une population subissant une influence extérieure, ici l'Homme. Pour ce modèle, on a choisi de s'inspirer d'un écologue qui décide de rajouter des proies si leurs nombre est faible et de chasser des prédateurs si le nombre de proies diminue. Il s'agit donc de rajouter une régulation des populations par le biais de l'Homme. Ainsi, on obtient le système d'équation suivant.

$$\begin{cases} \frac{\partial x}{\partial t}(t) = x(t)(\alpha - \beta y(t)) + \omega \times (1/x) \\ \frac{\partial y}{\partial t}(t) = y(t)(\delta x(t) - \gamma) - \phi \times (1/x) \end{cases}$$

La signification physique de  $\alpha$ ,  $\beta$ ,  $\gamma$  et  $\delta$  reste identique à celui du modèle précédent mais nous avons introduit deux nouvelles constantes de "régulation" :  $\omega$  et  $\phi$ .

Alors que  $\omega$  se traduit par la repopulation des proies induites pas l'Homme,  $\phi$  traduit l'efficacité de la chasse. Ainsi, pour des valeurs  $\alpha = 0.5, \beta = 0.2, \gamma = 0.5, \delta = 0.02, \phi = 0.1$  et  $\omega = 0.7$ , on obtient :

Nombre de Proies au cours du temps Pour  $\alpha = 0.5, \beta = 0.2, \gamma = 0.5, \delta = 0.02, \phi = 0.1$  et  $\omega = 0.7$



Nombre de Prédateur au cours du temps Pour  $\alpha = 0.5, \beta = 0.2, \gamma = 0.5, \delta = 0.02, \phi = 0.1$  et  $\omega = 0.7$

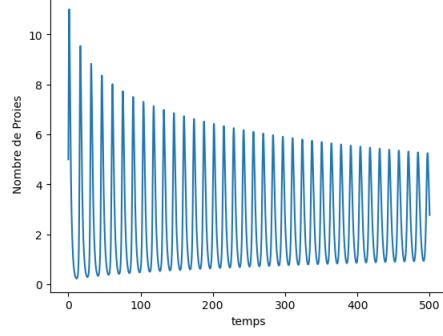


FIGURE 6 – Evolution des populations pour  $\alpha = 0.5, \beta = 0.2, \gamma = 0.5, \delta = 0.5, \omega = 0.7, \phi = 0.1$

Comme on peut le voir, le cycle semble converger vers une valeur finie. Pour voir la valeur final, on peut augmenter le temps d'acquisition (nombre d'itérations).

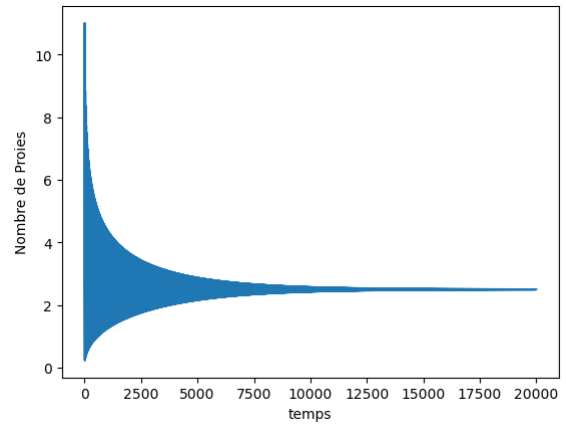
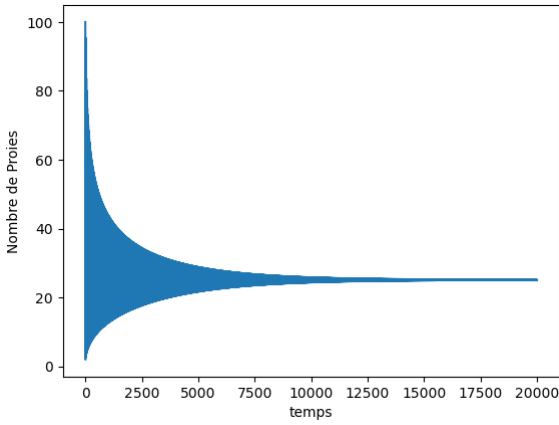
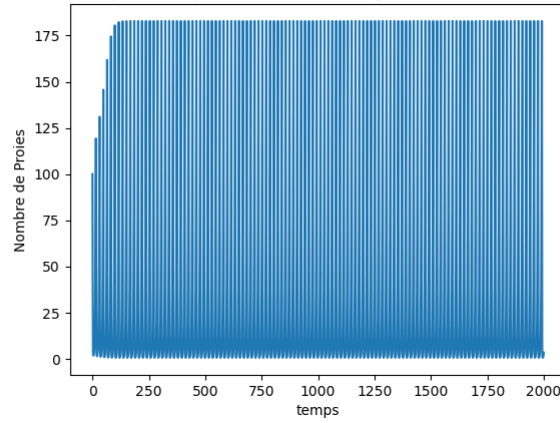


FIGURE 7 – Evolution des populations pour  $\alpha = 0.5, \beta = 0.2, \gamma = 0.5, \delta = 0.5, \omega = 0.7, \phi = 0.1$

Comme on peut le voir, le nombre de Proies tend vers 25 alors que le nombre de Prédateur tend vers 2. Pour autant, il est possible de trouver des paramètres pour lesquels on tombe sur d'autres situations. Si on choisit  $\omega = 0.2$  et  $\phi = 0.2$ , on retombe sur le modèle de Lotka-Volterra sur des temps longs.

Nombre de Proies au cours du temps Pour  $\alpha = 0.5, \beta = 0.2, \gamma = 0.5, \delta = 0.02, \phi = 0.2$  et  $\omega = 0.2$



Nombre de Prédateur au cours du temps Pour  $\alpha = 0.5, \beta = 0.2, \gamma = 0.5, \delta = 0.02, \phi = 0.2$  et  $\omega = 0.2$

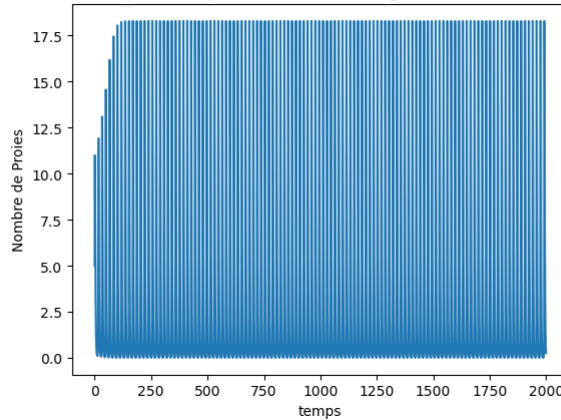


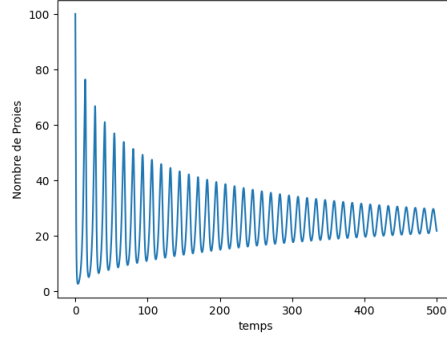
FIGURE 8 – Evolution des populations pour  $\alpha = 0.5, \beta = 0.2, \gamma = 0.5, \delta = 0.5, \omega = 0.2, \phi = 0.2$

Comme on peut le voir, à partir d'un certain temps, on atteint un régime décrit par le modèle précédent. Ainsi, si les valeurs de  $\omega$  et de  $\phi$  sont comprises entre 0 et 1, on a deux situations possibles :

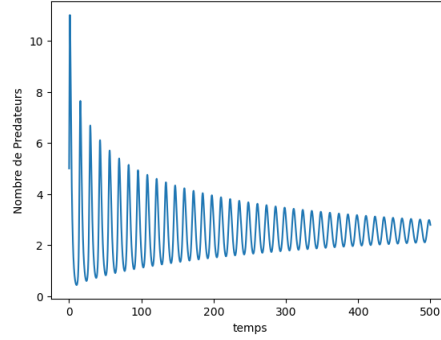
- Soit on revient sur le modèle initial
- Soit on obtient une nouvelle évolution qui converge vers une valeur finie

Or ici,  $\omega$  peut être supérieur à 1. En effet, il représente un taux de proies amené par les Hommes. Prenons alors, comme dans les cas précédent, les mêmes paramètres mais en modifiant la valeur de  $\omega$ .

Nombre de Proies au cours du temps Pour  $\alpha = 0.5, \beta = 0.2, \gamma = 0.5, \delta = 0.02, \phi = 0.2$  et  $\omega = 3$



Nombre de Prédateur au cours du temps Pour  $\alpha = 0.5, \beta = 0.2, \gamma = 0.5, \delta = 0.02, \phi = 0.2$  et  $\omega = 3$



Espace des Phases Pour  $\alpha = 0.5, \beta = 0.2, \gamma = 0.5, \delta = 0.02, \phi = 0.2$  et  $\omega = 3$

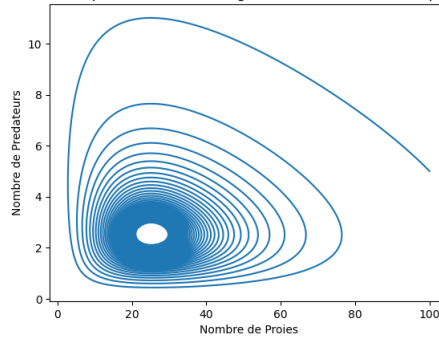


FIGURE 9 – Evolution des populations pour  $\alpha = 0.5, \beta = 0.2, \gamma = 0.5, \delta = 0.5, \omega = 3, \phi = 0.2$

Comme on peut le voir pour des valeurs de  $\omega$  positives, on obtient un cycle qui converge. On retombe sur la situation pour des valeurs de  $\omega = 0.7$  et  $\phi = 0.1$  On peut, comme pour le modèle de Lotka-Volterra, déterminer la contribution d'une légère modification de  $\omega$  sur le temps d'une oscillation. Ainsi, on obtient :

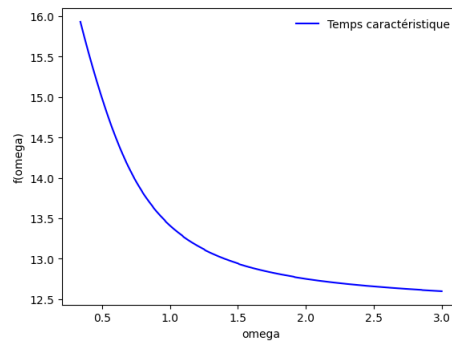


FIGURE 10 – Temps d'une période en fonction de  $\omega$



On remarque que l'on ne peut plus faire d'approximation linéaire de notre caractéristique mais pour autant, le temps caractéristique semble suivre une loi exponentielle en fonction de  $\omega$ . On peut faire la même chose mais maintenant en variant  $\phi$ .

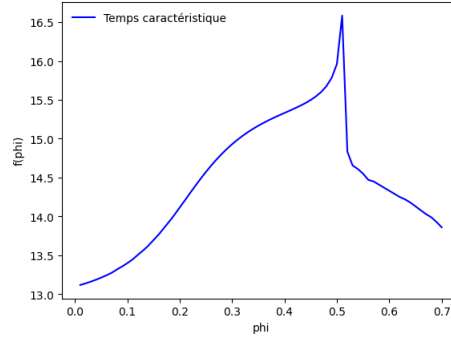


FIGURE 11 – Temps d'une période en fonction de  $\phi$

Au final, on trouve que :

$$d\tau = -59.516 \times d\alpha + 166.787 \times d\beta - 67.071 \times d\gamma + 952.68 \times d\delta + Ce^{-\frac{\omega}{\Omega}} \times d\omega + f(\phi) \times d\phi \quad (2)$$

#### 2.2.4 ouverture sur un 3ème modèle : Modèle avec effort

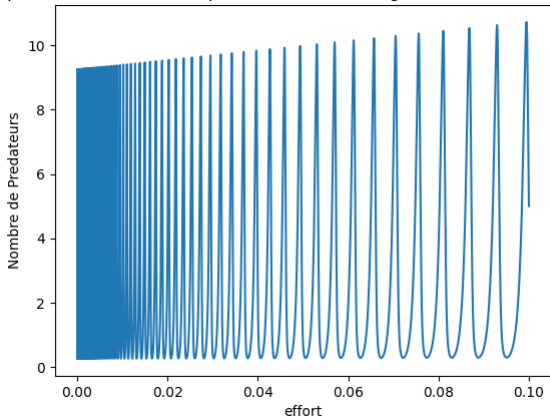
Le principe de ce modèle est simple. Tandis que dans la sous-partie précédente les humains étaient un effet externe au système, ici leurs efforts sont eux aussi assujettis à la variation des populations.

$$\begin{cases} \dot{x}(t) = x(t)(\alpha - \beta y(t) + \omega w) \\ \dot{y}(t) = y(t)(\delta x(t) - \gamma - \omega w) \\ \dot{w}(t) = \sigma w(a - c) \end{cases}$$

Dans ce jeu d'équations, à la différence du modèle précédent, la variation de  $x(t)$  dépend du produit  $x(t)\omega w$  (de même pour  $y(t)$ ).

Ce nouveau modèle permet de mettre en avant un "effort" des Hommes qui est lié à la variation de population. Alors que dans le second modèle l'effort était constant (lié aux valeurs de  $\omega$  et  $\phi$ ), ici il va varier au cours du temps. Ce modèle est une "amélioration" du modèle précédent mais il semble toutefois avoir des défauts comme par exemple la quantification de  $\omega$ .

Espace des Phases Pour  $\alpha = 0.5, \beta = 0.2, \gamma = 0.5, \delta = 0.02$



Espace des Phases Pour  $\alpha = 1, \beta = 1, \gamma = 1, \delta = 1$

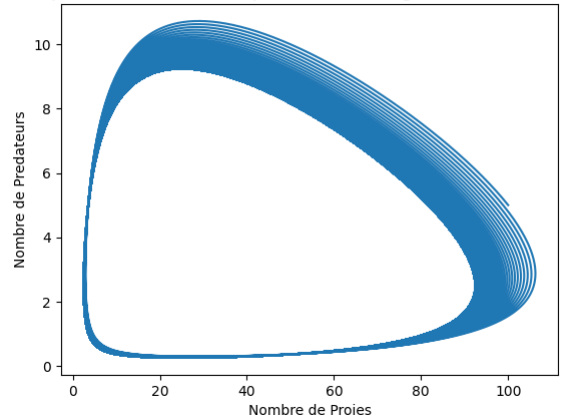


FIGURE 12 – Espace des phases pour  $\alpha = 0.5, \beta = 0.2, \gamma = 0.5, \delta = 0.02, \omega = 0.8, (a - c) = 0.01$

On remarque que l'évolution de l'effort dépend grandement de la valeur de  $(a - c)$ .

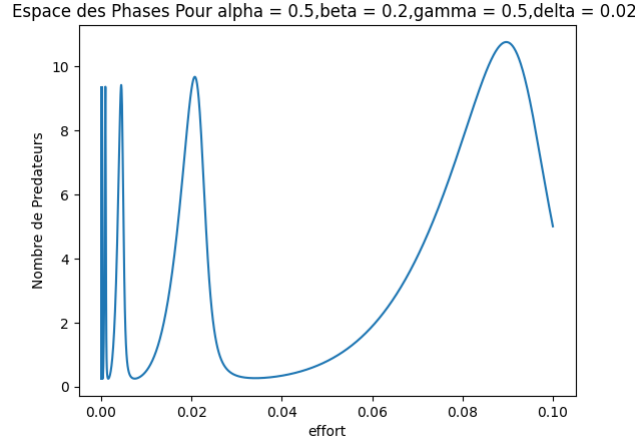


FIGURE 13 – espace des phases de l'évolution des prédateurs et de l'effort pour  $(a - c) = 0.2$

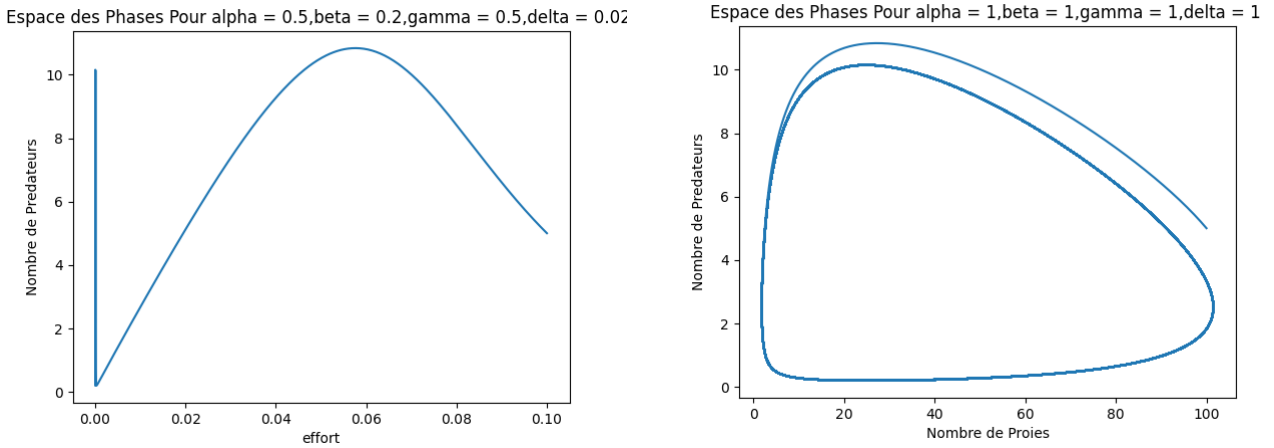


FIGURE 14 – Espace des phases pour  $\alpha = 0.5, \beta = 0.2, \gamma = 0.5, \delta = 0.02, \omega = 0.8, (a - c) = 1$

4

Comme on peut le constater, l'évolution du nombre de Prédateurs et de Proies converge vers un cycle stable que l'on peut décrire avec le premier modèle. Pour autant, l'évolution de l'effort semble être un peu plus compliqué à décrire.

### 3 Une seconde approche : Modèle 2D

L'objectif de cette partie est de proposer un modèle proie-prédateur en 2 dimensions et de le transposer en simulation numérique. On comparera les résultats avec ceux obtenus dans la partie précédente.

#### 3.1 Modélisation physique

Dans ce modèle, les proies et les prédateurs sont disposés sur une grille 2D sur laquelle ils peuvent se déplacer. Ce modèle est discretisé spatialement et temporellement : entre deux instants successifs,

chaque individu se déplace sur une case voisine. Proies et prédateurs peuvent se reproduire et mourir et les prédateurs peuvent chasser les proies. A chaque action est attribuée une probabilité de succès.

Cette méthode présente des différences avec le modèle de Lotka-Volterra. D'abord cette simulation n'est pas déterministe : les actions effectuées (ou non) par les individus (déplacement, reproduction, chasse, décès) sont régies par des probabilités. Cela signifie qu'une situation initiale peut conduire à des évolutions différentes, contrairement au modèle de Lotka-Volterra. De plus, ce modèle simule le déplacement des individus sur une grille et prend donc en compte une dépendance spatiale, ce qui n'est pas le cas dans les équations de Lotka-Volterra. Les conditions initiales spatiales, la taille et la forme de la grille peuvent donc avoir un impact sur l'évolution des populations. Enfin, le modèle de Lotka-Volterra est continu (couple d'équations différentielles) tandis que le modèle de grille 2D est discretisé temporellement et spatialement par les cases de la grille.

Les hypothèses du modèle sont les suivantes :

- On se place dans le cas d'une grille carrée avec  $N^2$  cases ( $N$  étant le nombre de cases par ligne/colonne) et on utilise les conditions aux limites périodiques. Une grille carrée permet de ne pas avoir de direction spatiale privilégiée et les conditions aux limites périodiques permettent de simuler une infinité de grilles collées les unes aux autres ayant les mêmes propriétés moyennes, sans se soucier des effets de bord. Ainsi si un individu sort de la grille, il est ramené dans la grille de l'autre côté. Par exemple, s'il sort de la grille par le haut il est ramené en bas sur la même colonne. On rappelle que le déplacement des individus est restreint à ses 8 cases voisines (en tenant compte des conditions aux limites périodiques).
- Le comportement des proies et des prédateurs est régi par un jeu de quatre nombres réels  $\alpha$ ,  $\beta$ ,  $\gamma$  et  $\delta$  compris entre 0 et 1. Le choix des noms n'est pas anodin et fait référence à ceux choisis dans la partie précédente, les taux étant remplacées ici par des probabilités à chaque itération. Ainsi  $\alpha$  représente la probabilité de reproduction intrinsèque de la proie et  $\gamma$  la probabilité qu'un prédateur meure. Comme dans le modèle de Lotka-Volterra, ces deux paramètres sont intrinsèques, c'est à dire constants. Les paramètres  $\beta$  et  $\delta$  traitent quant à eux des interactions entre proies et prédateurs. En particulier,  $\beta$  représente la probabilité pour une proie d'être tuée par un prédateur. Elle est elle aussi constante mais le taux de "chasse" doit être plus élevé lorsque le nombre de prédateurs et de proies est élevé (terme en  $-x(t)\beta y(t)$  dans Lotka-Volterra). Ce point est bien reproduit avec le modèle 2D. De plus,  $\delta$  représente la probabilité pour un prédateur de se reproduire. Le taux de reproduction des prédateurs doit augmenter avec le nombre de proies rencontrées et chassées. Cela se justifie par le fait qu'un prédateur doit d'abord survenir à ses besoins (dont la nourriture) avant de se reproduire. Dans le modèle précédent, cette dépendance était régie par le terme  $y(t)\delta x(t)$ . Or, le modèle 2D tel quel ne permet pas de satisfaire cette observation. Pour ce faire, on augmente la probabilité  $\delta$  de reproduction d'un prédateur qui vient de manger et on la diminue sinon. Les valeurs de  $\delta$  étant comprises entre 0 et 1, nous avons opté pour la fonction  $x \mapsto x^{\frac{1}{3}}$  pour l'augmentation et  $x \mapsto x^3$  pour la diminution de  $\delta$ . Ainsi, si le prédateur  $i$  a une probabilité de reproduction  $\delta_i(t)$  à l'instant  $t$ , alors la probabilité de reproduction  $\delta_i(t+1)$  à l'instant  $t+1$  est  $\delta_i(t)^{\frac{1}{3}}$  s'il s'est nourri et  $\delta_i(t)^3$  sinon. Ainsi, chaque individu naît avec les mêmes caractéristiques ( $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $\delta$  fixés). Cependant, le paramètre  $\delta_i$  du prédateur  $i$  varie d'une itération à l'autre.

## 3.2 Simulation numérique

### 3.2.1 Structure du programme

Nous avons utilisé Python pour la simulation numérique de notre modèle 2D. Nous avons programmé deux classes *Proie* et *Predateur*. La classe *Proie* permet de créer un individu de type

proie. Ses attributs sont *coord* (couple  $(x, y)$  de coordonnées de la proie) et *reproduction* (probabilité de reproduction  $\alpha$  de la proie, constante pour toutes les proies). La classe *Predateur* permet de créer un individu de type prédateur. Ses attributs sont *coord* (couple  $(x, y)$  de coordonnées du prédateur), *reproduction* (probabilité de reproduction du prédateur  $\delta$ , varie en fonction de sa faim), *mortalite* (probabilité de mortalité  $\gamma$  du prédateur) et *chasse* (probabilité  $\beta$  de tuer une proie située sur une case voisine). Chaque simulation est initialisée en choisissant le nombre de lignes *nb\_lig* ( $= nb\_col$ ), le nombre de prédateurs *nb\_pred*, le nombre de proies *nb\_proie* et la valeur des différents attributs énoncés précédemment. Les individus sont créés et stockés dans deux listes *liste\_pred* et *liste\_proie*. Les individus d'une même classe sont initialisés avec les mêmes attributs, sauf les coordonnées qui sont choisies aléatoirement dans la grille. Le choix des coordonnées ainsi que tous les tests de probabilité dans la suite sont assurés par la fonction *randint* du module *random*.

L'interface graphique est créée sur une fenêtre à part (voir Figure 15). Nous avons utilisé le module *tkinter* (et *tkinter.font*) et en particulier le widget *Canvas* pour créer la fenêtre, la grille et le plot de la démographie des individus des deux espèces en fonction du temps. Le bouton *Evolution* situé en bas à droite de l'interface permet de passer à la frame suivante en faisant évoluer les individus sur la grille comme décrit dans la partie 3.1. L'évolution est régie par la fonction *evolution*.

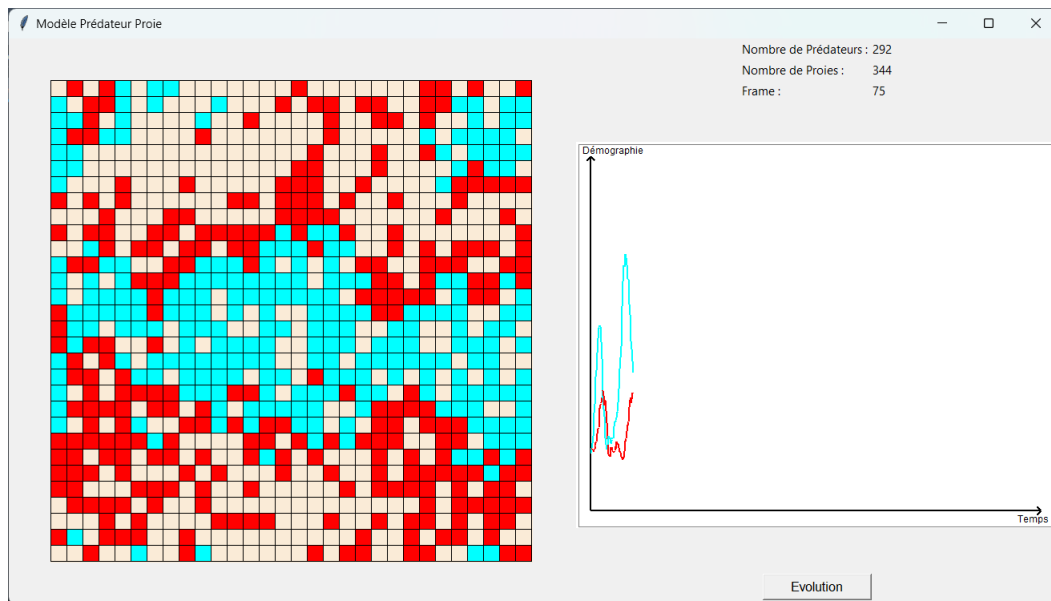


FIGURE 15 – Interface graphique. A gauche, la grille de simulation 2D où sont représentés en bleu les proies et en rouge les prédateurs. A droite, l'interface de tracé de la démographie de proies (en bleu) et de prédateurs (en rouge) en fonction du temps (numéro de frame). En haut à droite, le nombre de prédateurs et de proies à la frame actuelle ainsi que le numéro de cette frame. En bas à droite, le bouton *Evolution* qui permet de passer à la frame suivante.

L'avantage de cette interface graphique est que l'on peut voir en temps réel l'évolution des individus et ainsi apprécier les contraintes spatiales imposées par la taille de la grille et la répartition des individus sur la grille. Cependant, chaque simulation étant différente du fait des probabilités, on ne peut rien conclure d'une simulation unique. Il est donc appréciable de faire tourner plusieurs simulations à la suite pour s'affranchir des fluctuations statistiques et ainsi s'intéresser aux courbes d'évolution moyennes.

### 3.2.2 Résultats

Dans un premier temps, nous fixons  $nb\_lig = nb\_col = 30$ ,  $nb\_pred = nb\_proie = 150$ ,  $\alpha = 0.5$ ,  $\beta = 0.9$ ,  $\delta = 0.24$  et  $\gamma = 0.2$ . Vérifions d'abord que le moyennage sur plusieurs simulations est consistant pour ces paramètres :

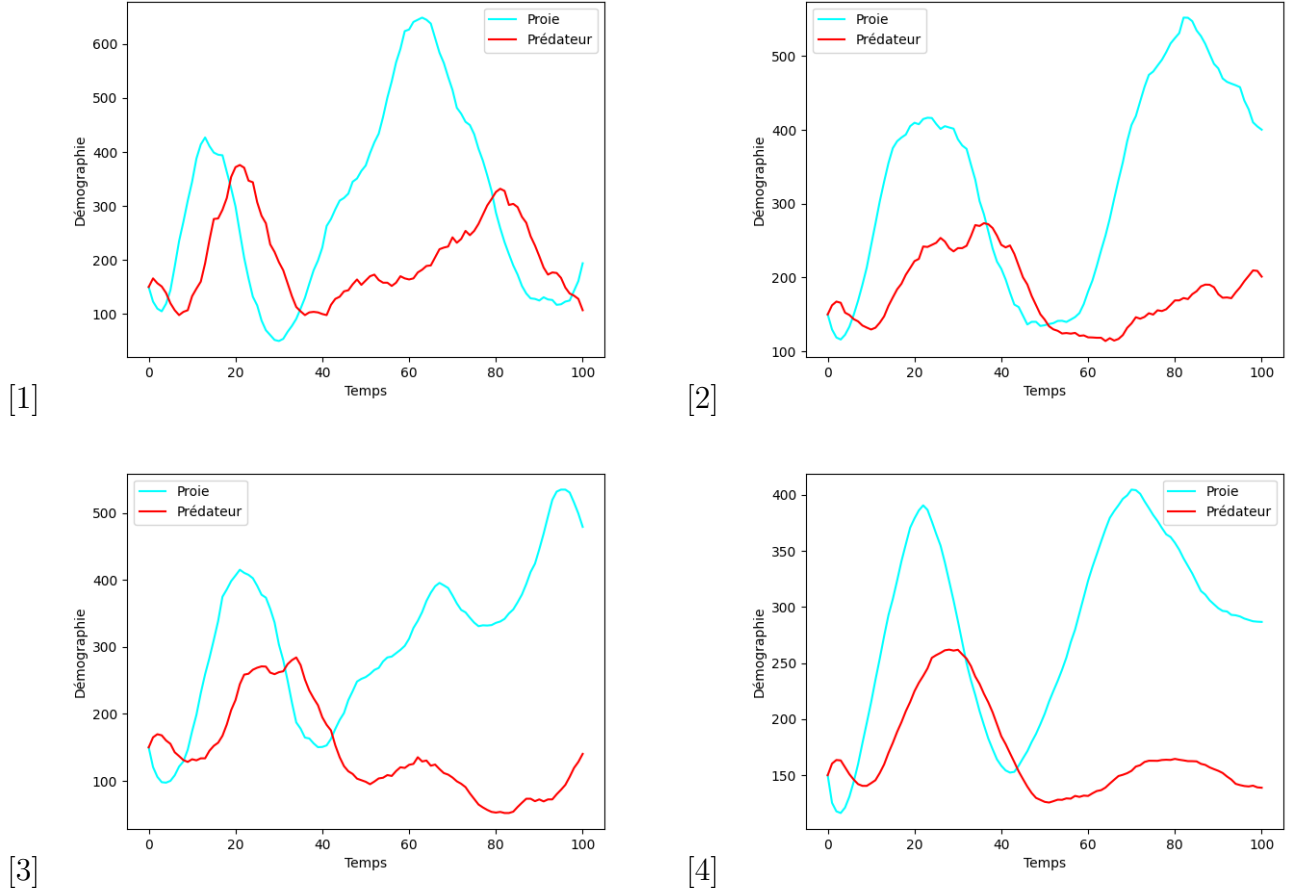


FIGURE 16 – Démographie des populations moyennée sur 1 simulation [1], 5 simulations [2], 20 simulations [3] et 50 simulations [4]. On a fixé le nombre de frame à 100.

On remarque sur la figure 16 que les courbes se lissent lorsqu'on augmente le nombre de simulations pour le moyennage. Cependant, la figure 17 montre que deux moyennages sur le même nombre de simulations peuvent donner des résultats différents si le nombre de simulations est trop faible. Or, les temps de calculs augmentent considérablement avec le nombre de simulations moyennées. On tâchera de choisir dans la suite un nombre de simulations assez important pour ne pas risquer d'étudier un cas particulier mais tout de même raisonnable compte tenu du nombre de frames demandées.

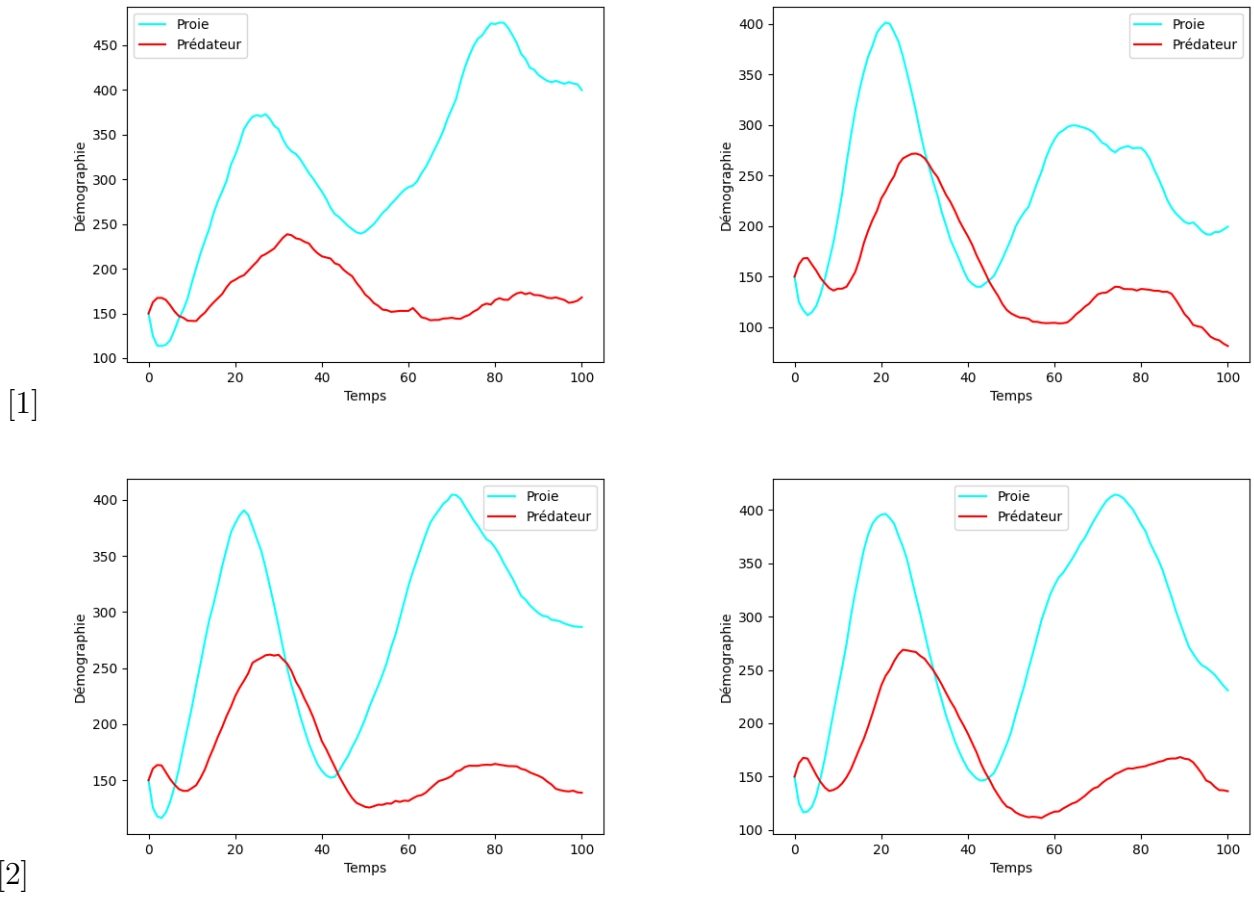


FIGURE 17 – Deux moyennages sur 20 [1] et 50 [2] simulations de la démographie des populations sur 100 frames. On remarque que la robustesse du moyennage se gagne en augmentant le nombre de simulations.

Voyons maintenant l'évolution des populations pour des temps plus longs. Nous allons étudier l'influence des différents paramètres du modèle en les modifiant un à un sans changer les autres. Intéressons-nous d'abord à l'influence du nombre de lignes et de colonnes, c'est à dire à la taille de la grille. Nous considérerons toujours la grille carré et fixons la densité initiale de proies et de prédateurs égales et constantes : 0.17 proies (ou prédateur) par case. Les autres paramètres sont maintenus constants égaux à ceux indiqués en début de sous-partie 3.2.2. Nous pouvons voir sur la figure 18 que le nombre de cases a un impact sur la dynamique des populations. On remarque d'abord que dans tous les cas, les populations semblent atteindre un équilibre se caractérisant. De plus, on peut constater que plus le nombre de case augmente, plus le nombre de frames nécessaires pour atteindre cet équilibre est élevé. Cela peut s'interpréter par le fait que lorsque le nombre de case est élevé, les proies ont plus d'espace pour se reproduire et échapper aux prédateurs. Un nombre plus faible de cases restreint leur possibilités de survivre en se cachant des prédateurs. Avant d'atteindre l'équilibre, on remarque cependant toujours le même paterne : au début, les prédateurs prennent le dessus sur les proies. S'en suit alors une diminution du nombre de proies, qui engendre alors une baisse du nombre de prédateurs. Les oscillations ne sont pas aussi précises que dans le modèle de Lotka-Volterra mais sont tout de même présentes. Cela peut être dû au moyennage (lissage des courbes) ou bien peut être intrinsèque au modèle 2D. Enfin, le fait que l'équilibre est toujours une population nulle de prédateurs et une population non nulle constante de proie est remarquable. Les prédateurs meurent toujours de faim et les proies se maintiennent à un nombre constant, qui dépend de la taille de la grille. Cela n'était pas un cas possible avec le modèle de

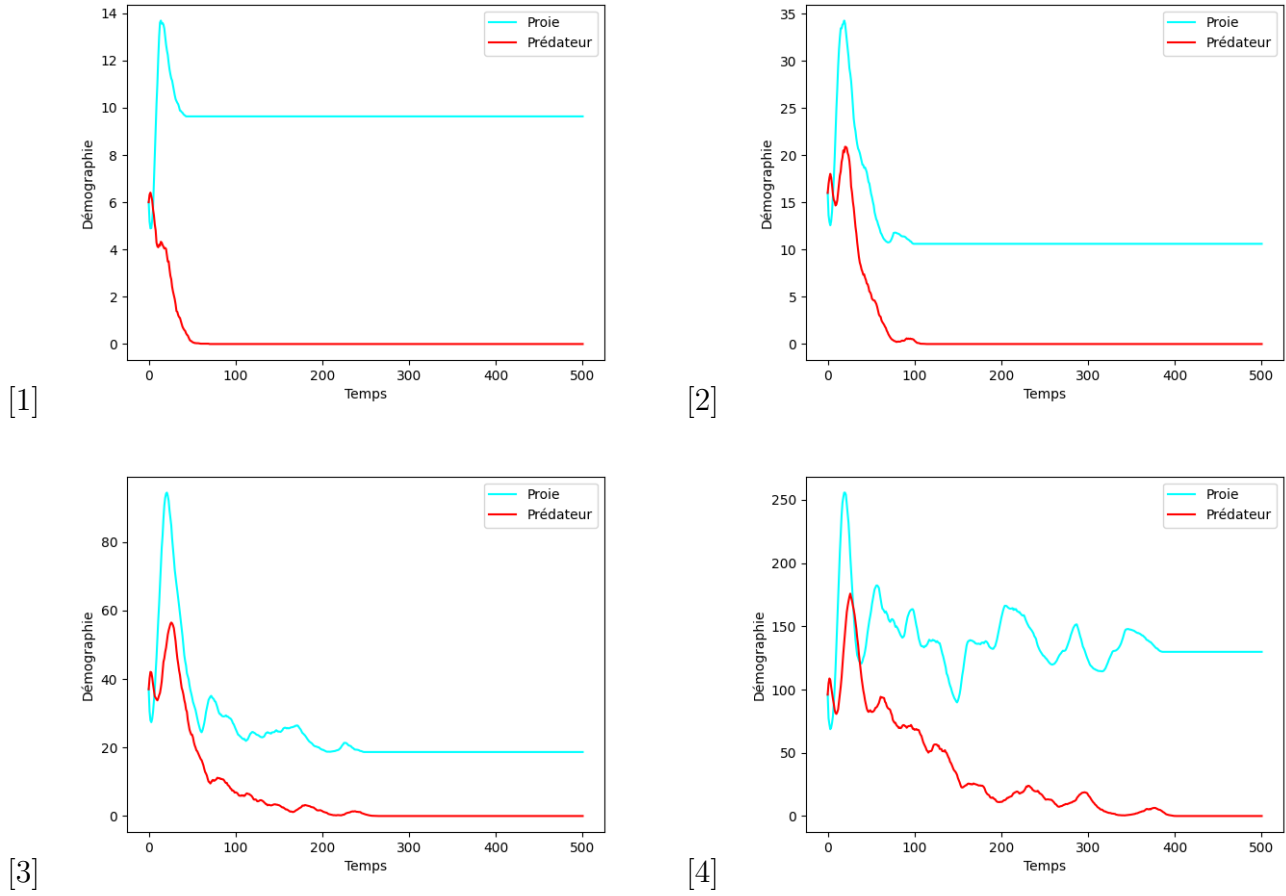
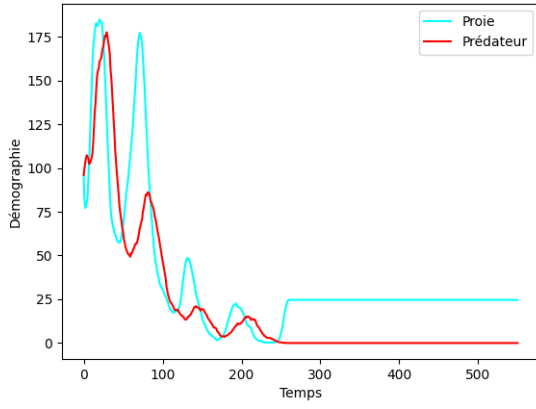
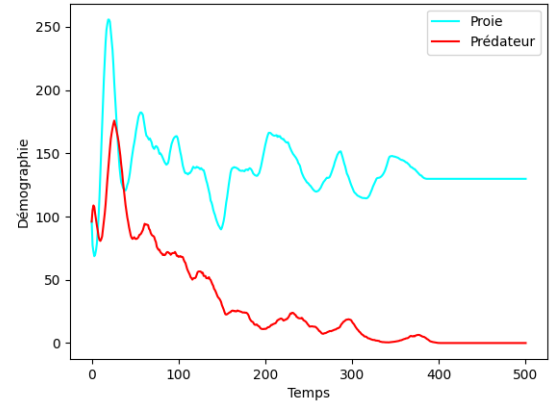


FIGURE 18 – Démographie des populations moyennée sur une grille composée de 6 lignes [1], 10 lignes [2], 15 lignes [3] et 24 lignes [4] sur 500 frames. Le nombre de simulations moyennées est respectivement 150,100,80 et 40. La densité initiale de prédateur est maintenue constante dans toutes les simulations et est égale à celle des proies : 0.17 par case.

Étudions maintenant l'influence du taux de reproduction des prédateurs fixé jusqu'alors à  $\delta = 0.24$ . On fixera le nombre de lignes (et le nombre de colonnes) à 24. On remarque sur la figure 19 que les populations connaissent toujours un équilibre (mort des prédateurs, survie des proies à démographie constante) et que le nombre de frames nécessaire pour l'atteindre croît avec la valeur de  $\delta$ . Cela peut s'expliquer par le fait que lorsque  $\delta$  augmente, les prédateurs se reproduisant plus vite, ils font perdurer d'avantage les oscillations pré-équilibre jusqu'à s'éteindre.



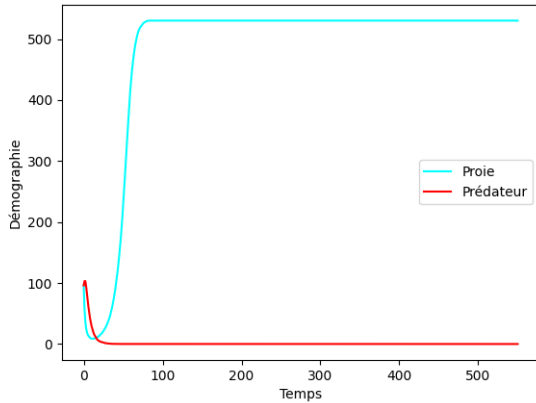
[1]



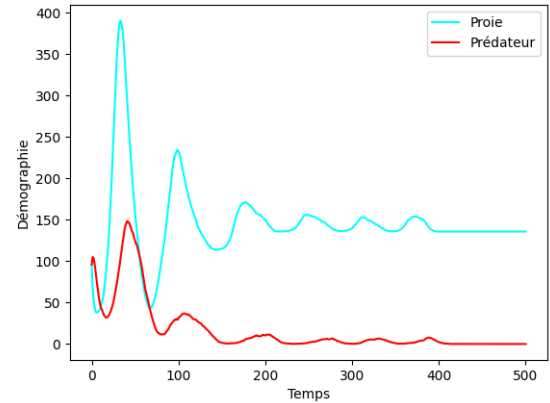
[2]

FIGURE 19 – Démographie des populations moyennée pour  $\delta = 0.1$ [1] et  $\delta = 0.24$ [2]. Le nombre de simulations moyennées est respectivement 30 et 40 et le nombre de lignes (et de colonnes) 24.

Intéressons nous désormais à l'influence du taux de reproduction des proies fixé jusqu'alors à  $\alpha = 0.5$ . La figure 20 montre à nouveau qu'un équilibre s'établit et que le nombre de frames nécessaire pour l'atteindre croît avec  $\alpha$ . Cela peut s'expliquer par le fait que si on augmente le taux de reproduction des proies, les prédateurs peuvent d'avantage se nourrir et faire perdurer les pseudo-oscillations avant de périr. Cela pourrait pourtant paraître contre-intuitif puisqu'on aurait pu s'attendre à ce que les proies s'éteignent lorsque leur taux de reproduction devient plus faible. Or, on remarque sur la figure 20[1] que pour un taux de reproduction de  $\alpha = 0.1$ , le nombre de proies chute en effet drastiquement au début de la simulation, mais cela entraîne une chute brutale de la démographie des prédateurs jusqu'à leur extinction. Les proies peuvent ainsi se reproduire à nouveau et atteindre un équilibre (elles occupent tout l'espace de la grille puisque  $24^2 = 276$ ).



[1]



[2]

FIGURE 20 – Démographie des populations moyennée pour  $\alpha = 0.1$ [1] et  $\alpha = 0.3$ [2]. Le nombre de simulations moyennées est 30 et le nombre de lignes (et de colonnes) 24.

Étudions maintenant l'influence du paramètre de chasse jusqu'alors fixé à  $\delta = 0.9$ . Sur la figure 21, on observe toujours un équilibre des populations aux temps longs. Pour des taux de chasse compris entre 0.1 et 0.9, on observe que cet équilibre est atteint pour des temps plus longs lorsque  $\delta$  augmente (figure 21[1][2][3]). On observe que pour  $\delta \sim 0.1$ , le nombre de prédateurs tend drastiquement vers 0



tandis que le nombre de proies augmente exponentiellement jusqu'à atteindre la limite maximale due à la taille de la grille comme vu précédemment. Cela peut s'expliquer par le fait que les prédateurs ayant plus de difficultés à se nourrir, ils meurent très rapidement ce qui permet aux proies de survivre. Cependant pour des valeurs plus élevées de  $\delta$ , les prédateurs parviennent à lutter pour leur survie mais meurent tout de même pour  $\delta = 0.9$ . On remarque cependant que pour  $\delta = 1$ , les prédateurs ayant une probabilité de 1 de tuer la proie qui se trouve dans les 8 cases voisines, ils parviennent à tuer toutes les proies, entraînant par la même occasion leur extinction.

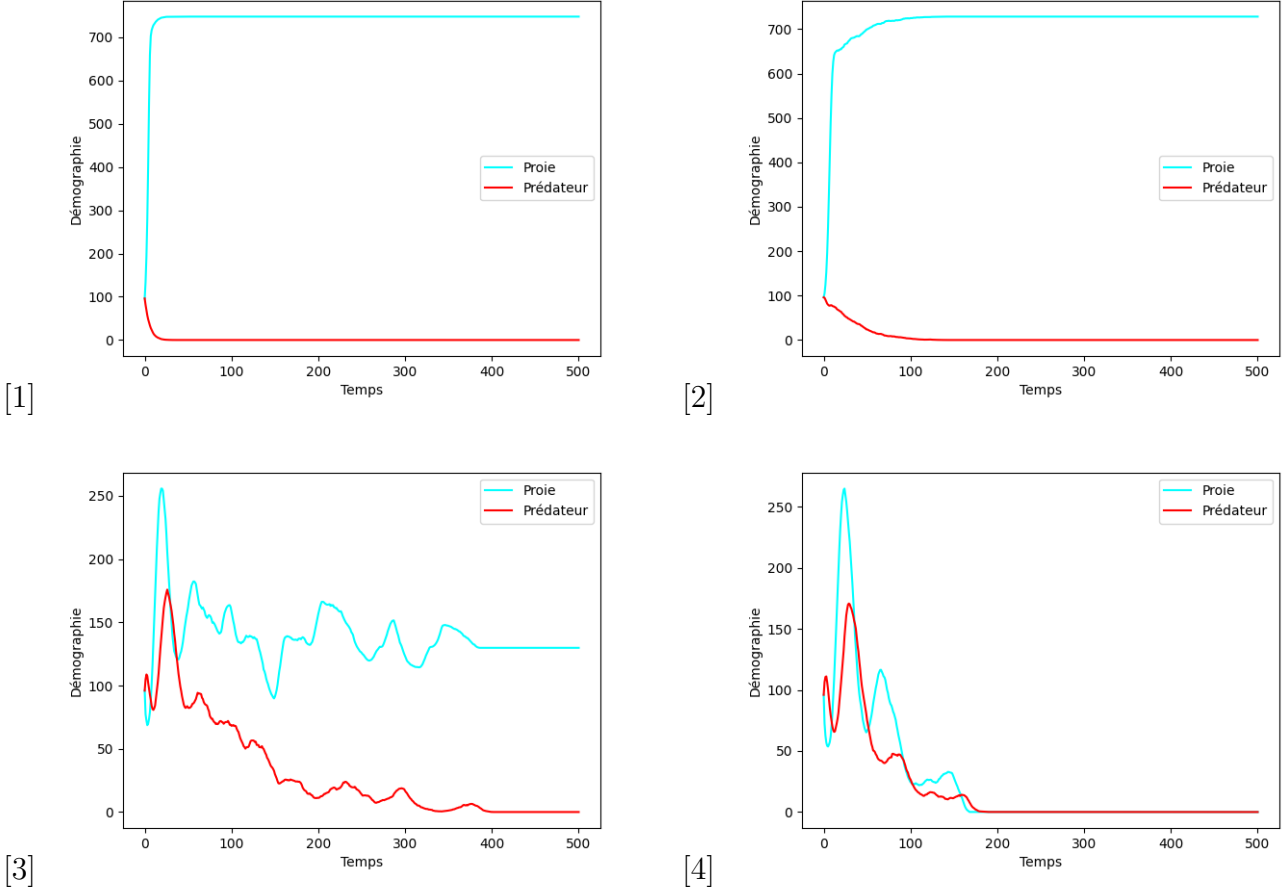


FIGURE 21 – Démographie des populations moyennée pour  $\delta = 0.1$ [1],  $\delta = 0.2$ [2],  $\delta = 0.9$ [3] et  $\delta = 0.1$ [4]. Le nombre de simulations moyennées est 40 et le nombre de lignes (et de colonnes) 24.

Intéressons-nous enfin à l'influence du taux de mortalité intrinsèque des prédateurs fixé jusqu'alors à  $\gamma = 0.2$ . On peut voir sur la figure 22 qu'il existe toujours un équilibre aux temps longs. Pour des petites valeurs de  $\gamma$ , le temps nécessaire pour l'atteindre croît avec  $\gamma$ . Cela peut paraître contradictoire car en augmentant le taux de mortalité intrinsèque des prédateurs, leur durée de vie s'accroît. Or, il faut garder en tête que si les prédateurs meurent moins, alors ils se reproduisent et se nourrissent de plus en plus jusqu'à consommer toutes les proies présentes, ce qui entraîne inévitablement la mort des prédateurs. On remarque que les proies continuent de vivre pour une valeur de  $\delta$  comprise entre 0.15 et 0.16 (voir figure 22[2][3]). Enfin, pour des grandes valeurs de  $\delta$ , presque tous les prédateurs meurent dès les premières itérations et les proies se reproduisent exponentiellement.

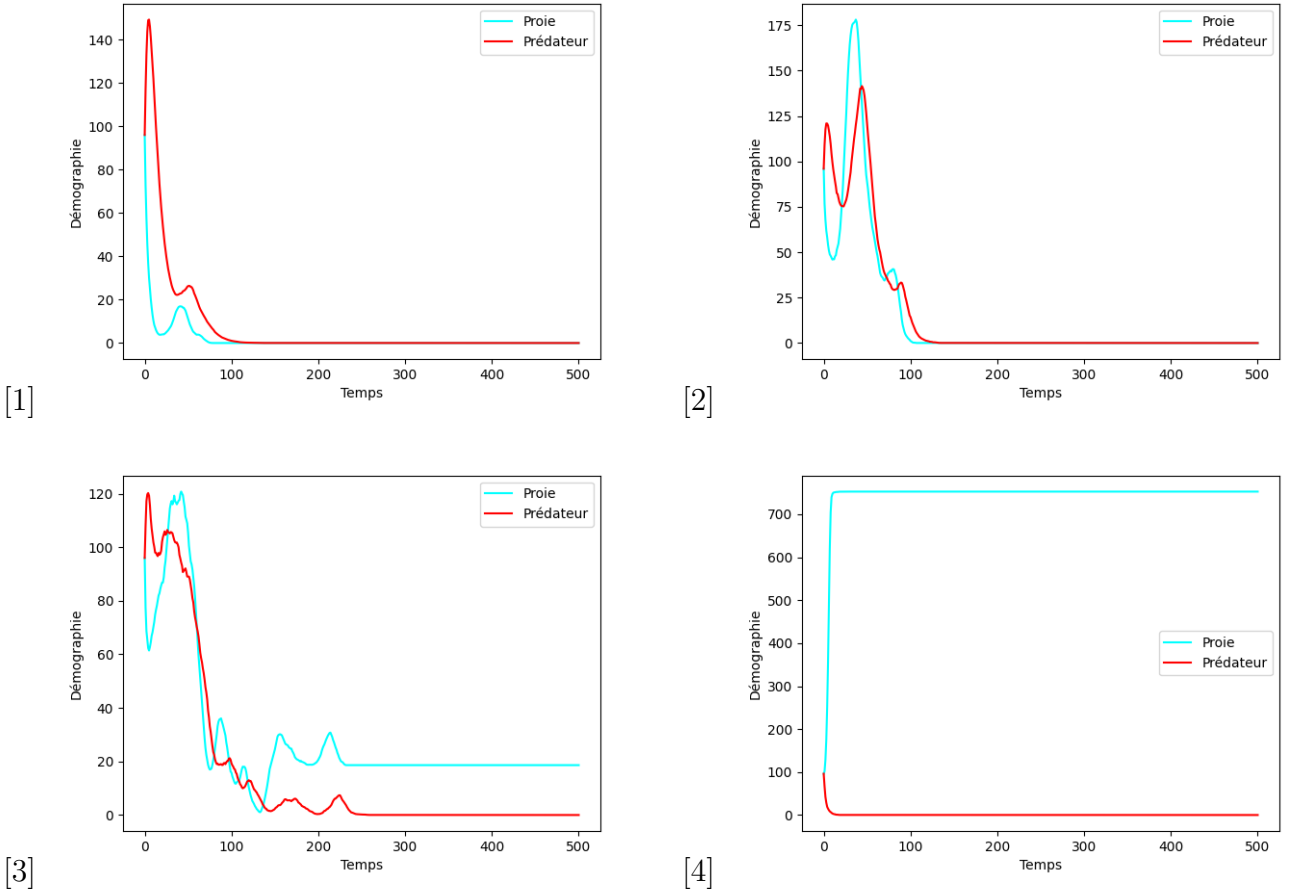


FIGURE 22 – Démographie des populations moyennée pour  $\gamma = 0.1$ [1],  $\gamma = 0.15$ [2],  $\gamma = 0.16$ [3] et  $\gamma = 0.5$ [4]. Le nombre de simulations moyennées est 40 et le nombre de lignes (et de colonnes) 24.

## 4 Conclusion

Pour conclure, les modèles de Lotka-Volterra et celui 2D sont deux approches similaires (taux qui s'assimilent à des probabilités, pseudos-oscillations dans les deux modèles) mais différent par bien des points (déterministe/aléatoire, dépendance spatiale dans le modèle 2D, problème continu/discret) dans l'étude de la démographie des proies et des prédateurs. Ces deux approches se complètent et permettent de mettre en avant des phénomènes différents. Les différents modèles élaborés dans la partie 1 ont permis d'élargir le modèle de Lotka-Volterra en ajoutant des contraintes et nouveaux paramètres. Dans le modèle 2D, la contrainte principale est les conditions de bords choisis ici périodique pour simuler une infinité de grilles collées les unes aux autres. Une même étude pour des conditions aux limites non périodiques pourrait également être intéressante. En particulier, les contraintes liées au nombre de cases seraient différentes puisque les individus pourraient s'amasser au niveau des bords.