Génie Mathématique et Modélisation

Projet COMORE

Modélisation et commande optimale de l'interaction Coccinelle/Puceron dans le cadre de la lutte biologique

Caroline Bidot
3^o année du cycle ingénieur
DEA Mathématiques Appliquées
Stage effectué de Février à Juin 2001
Responsable du stage: Mr J.L. Gouzé

Remerciements

Je remercie Jean-Luc Gouzé pour m'avoir proposé ce sujet, pour ses explications et ses idées.

Je voudrais remercier toute l'équipe du projet COMORE pour leur accueil et leur gentillesse.

Je remercie également Mr Ferran et son équipe qui m'ont montré tout le processus des expérimentations, de l'élevage aux notations en serres.

Cadre du stage

L'Institut National de Recherche en Informatique et en Automatique (INRIA) est un Etablissement Public à caractère Scientifique et Technologique (EPST), placé sous la double tutelle du Ministère de la recherche et du Ministère de l'économie et des finances.

Ces objectifs essentiels sont : conduire des actions de recherche, développer la liaison entre recherche et industrie, favoriser les actions de transfert et de valorisation, participer à la formation par la recherche, jouer un rôle moteur dans le cadre européen.

Les recherches menées à l'INRIA sont tournées vers l'informatique, l'automatique et leurs applications. Elles associent mathématiciens appliqués, automaticiens, et informaticiens dans le cadre de quatre thèmes, toutes unités de recherche confondues :

- Thème 1 : Réseaux et Systèmes
- Thème 2 : Génie logiciel et Calcul symbolique
- Thème 3 : Interaction homme-machine, Images, Données, Connaissances
- Thème 4 : Simulation et Optimisation de systèmes complexes

Ce stage s'est effectué au sein du projet COMORE (Contrôle et Modélisation de Ressources Renouvelables) qui modélise des biosystèmes. COMORE, élément du Thème 4, est un projet commun avec le CNRS, Equipe Analyse et Simulation du Fonctionnement des Ecosystèmes (Station Zoologique, Villefranche sur Mer).

L'objectif global du projet est d'appliquer des méthodes de l'automatique (régulation, estimation, identification, contrôle optimal, théorie des jeux) et de la théorie des systèmes dynamiques à la modélisation mathématique et à la gestion des ressources vivantes exploitées. Le responsable de ce projet est Mr Jean-Luc Gouzé.

Les thèmes développés à ce jour sont essentiellement :

- Dynamique et contrôle des pêches et de l'aquaculture,
- Modélisation et contrôle de bioréacteurs,
- Modélisation et contrôle de la croissance du phytoplancton,
- Modélisation de la dynamique de la forêt,
- Mathématiques de la modélisation en biologie,
- Logiciels à usage des modélisateurs et des biologistes.

Le sujet du stage découle d'une collaboration avec l'Equipe Entomologie et Lutte Biologique, conduite par Mr Ferran, de l'INRA Antibes.

Table des matières

Introduction

L'immense majorité des pucerons connus sont des espèces monophages ou associées étroitement à une série de plantes de la même famille botanique. Ils forment un exemple de ravageurs hautement évolués, singulièrement spécialisés et avec une forte capacité d'adaptation au milieu. Ainsi le puceron Aphis gossypii Glover est un ravageur des cultures de cucurbitacées sous serre (concombre et melon). Ravageur car les dégâts qu'il cause entraînent une baisse du rendement des plantes.

La coccinelle Harmonia axyridis Pallas (Coleoptera : Coccinellidae) présente des caractéristiques biologiques qui en font un auxiliaire de lutte biologique intéressant contre les pucerons. On a sélectionné à partir de la souche originaire de Chine, une souche "flightless", incapable de voler. Cette souche présente deux intérêts majeurs : elle est sédentarisée sur les cultures où elle exerce son activité prédatrice et se reproduit, et elle ne peut pas se disperser dans l'environnement. On travaille, au travers d'un réseau d'expérimentation, à la mise au point des stratégies optimales de lâcher sur cultures. Ces essais sont effectués actuellement de façon empirique, en procédant par essai/erreur. La modélisation peut aider, sur le plan pratique, en termes de prédiction, mais également sur un plan plus fondamental, en termes de compréhension des interactions proie/prédateur.

Des expériences en serres sont actuellement menées par Mr Ferran et l'équipe entomologie et lutte biologique de l'INRA d'Antibes. Elles consistent en un lâcher de pucerons Aphis gossypii Glover sur des plants de concombres, suivi, quelques jours plus tard, d'un lâcher de coccinelles Harmonia axyridis Pallas. Des évaluations de la quantité d'individus de chacune des deux espèces sont effectuées à intervalle régulier pendant toute la durée de ces expériences (environ 3 mois).

L'objectif de ce stage est d'améliorer l'efficacité du traitement biologique, en utilisant les outils de l'automatique et des systèmes dynamiques. Il faudra tout d'abord écrire un modèle décrivant le système formé par les 2 espèces. Ensuite, plusieurs techniques de contrôle optimal seront testées pour chercher le meilleur compromis entre le coût (nombre de coccinelles lâchées, sachant qu'une coccinelle adulte est vendue 3F50) et l'efficacité (le nombre de pucerons doit rester en dessous d'un seuil économiquement acceptable).

Chapitre 1

Modélisation et le problème de contrôle

Notre but est de rechercher combien de coccinelles il nous faut introduire de manière à minimiser les dégâts causés par les pucerons.
Pour cela il nous tout d'abord avoir un modèle qui reproduise l'évolution des 2 populations.

1.1 Le modèle

On choisit un modèle "simple" non linéaire, de type proie-prédateur (Duffau, 2000) :

$$(S_1)\begin{cases} \frac{dx}{dt} = \dot{x} = ax(1-\frac{x}{K}) - b\frac{x}{x+c}y \\ \frac{dy}{dt} = \dot{y} = e\frac{x}{x+c}y - dy \end{cases}$$

x représentant la biomasse de la population de pucerons et y, la biomasse de la population de coccinelles.
Les paramètres de ce modèle sont a, K, b, c, e et d :

- a : taux de croissance intrinsèque des proies,
- K : capacité de soutien du milieu, $x \in [0\ ,\ K]$,
- b : masse maximale de proies attaquées par unité de temps,
- c : défini tel que $f(c) = \frac{b}{2}$ avec $f(x) = \frac{bx}{x+c}$,
  où la fonction f est le taux de prédation,
  relation entre la masse de proies attaquées par un prédateur
  par unité de temps et la masse de proies disponibles,
- e : $e = bc_u$ où c_u est le coefficient d'utilisation,
  c'est-à-dire le pourcentage de proies ingérées par les prédateurs,
- d : taux de disparition des prédateurs (mortalité et fuite).

On effectue une mise à l'échelle afin que les valeurs de x et y ne deviennent pas

trop grandes. Pour ce faire on applique un changement de variables : $x_e = \frac{x}{K}$ et $y_e = \frac{y}{y_m}$.
Comme la variable x varie entre 0 et K, on choisit pour le changement de diviser par K, et ainsi, la nouvelle variable x_e varie entre 0 et 1. On cherche à faire de même pour la variable y, or on ne connaît pas la valeur maximale que peut prendre cette variable, néanmoins on prendra y_m assez grand (proche de K) afin d'avoir y_e le plus souvent possible dans $[0\ ,\ 1]$.
Le système devient :

$$(S)\left\{\begin{array}{l} \dot{x}_e = ax_e(1-x_e) - b\frac{x_e}{Kx_e+c}y_m y_e \\ \dot{y}_e = e\frac{x_e}{x_e+\frac{c}{K}}y_e - dy_e \end{array}\right.$$

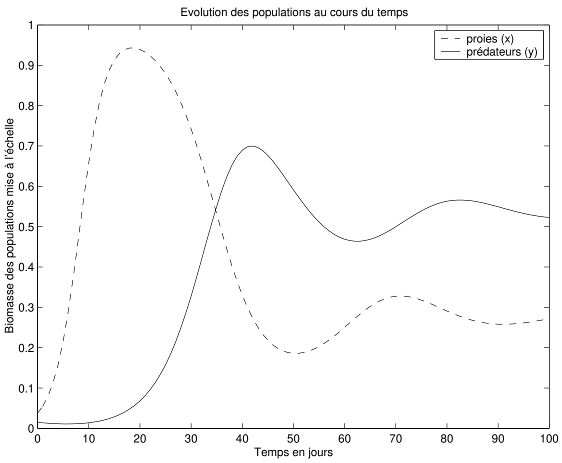
Dans la suite, par souci de simplification, on notera (x, y) pour (x_e, y_e).

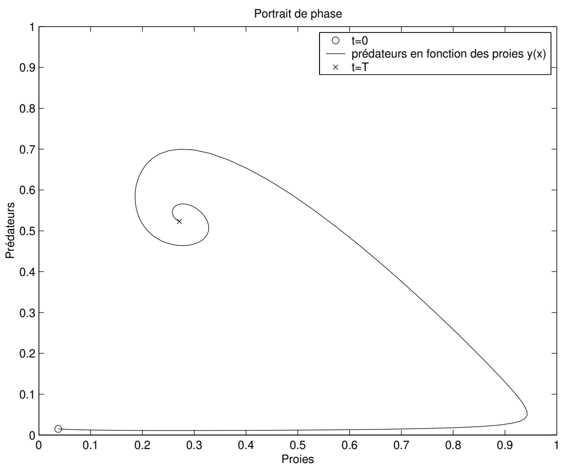
Ce modèle sera étudié sur un intervalle de temps $[0\ ,\ T]$, le temps 0 étant le moment où apparaissent les pucerons, et la variable T indiquant le temps final considéré, le moment de la récolte.

Exemple de comportement du système : les graphes suivants ont été obtenus pour les paramètres

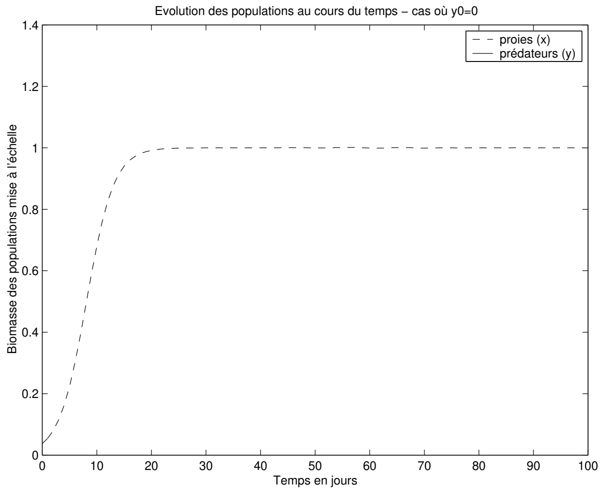
a	K	b	c	e	d	y_m
0,4	40000	2	100000	1	0,1	$\frac{3}{4}K = 30000$

pour les conditions initiales $(x_0, y_0) = (\frac{1500}{K} = 0,0375\ ,\ \frac{450}{y_m} = 0,015)$ et un intervalle de temps $[0\ ,\ 100]$.





Avec les mêmes paramètres et le même intervalle de temps, en changeant seulement la condition initiale en y_0, soit $y_0 = 0$, on observe la croissance logistique des proies en absence de prédateurs :



Etude du système : les points d'équilibre du système sont les couples (x_i, y_i) qui sont solution de (S) et de

$$\begin{cases} \dot{x} = 0 \\ \dot{y} = 0 \end{cases}$$

Le système (S) admet 3 points d'équilibre : $P_1 = (0\ ,\ 0)$, $P_2 = (1\ ,\ 0)$ et $P_3 = (x_3\ ,\ y_3)$, où

$$\begin{cases} x_3 = \frac{cd}{K(e-d)} \\ y_3 = \frac{ace}{b(e-d)y_m}(1 - \frac{cd}{K(e-d)}) \end{cases}$$

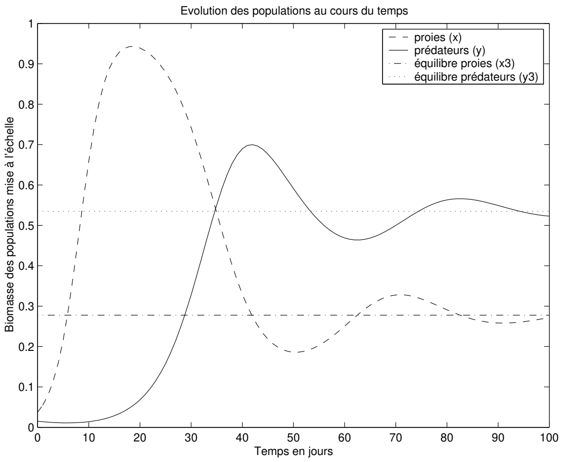
Ces points d'équilibre agissent sur le comportement du système :
P_1 est instable,
P_2 est stable sous la condition $\dfrac{e}{1+\frac{c}{K}} - d < 0$, or cette condition n'est jamais vérifiée, en effet $(\dfrac{e}{1+\frac{c}{K}} - d < 0) \Leftrightarrow (\dfrac{cd}{K(e-d)} = x_3 > 1)$, et $x_3 \in [0\ ,\ 1]$ car (x_3, y_3) est une solution de (S). Donc P_2 est instable,
et P_3 est stable sous la condition $\dfrac{K-c}{2} \leq \dfrac{cd}{e-d}$.

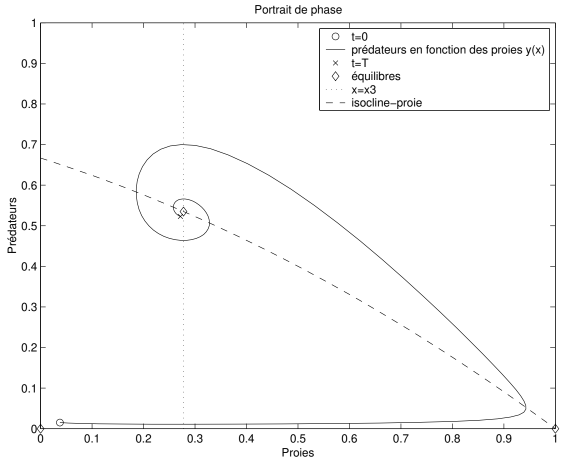
Les paramètres du système (S) sont choisis tels que cette dernière condition soit vérifiée.

Les variations des solutions de (S), $x(t)$ et $y(t)$, sont définies par les courbes isoclines. Ces courbes sont déterminées par $\dot{x} = 0$ pour les isoclines liées à x, et par $\dot{y} = 0$ pour celles liées à y. Les points d'intersection des isoclines sont les points d'équilibre.
Ainsi, $(\dot{y} = 0) \Leftrightarrow (y = 0 \text{ ou } x = \dfrac{cd}{K(e-d)} = x_3)$, d'où l'équation de l'isocline nulle par rapport à y est $x = x_3$, c'est-à-dire quand $x < x_3$, $y(t)$ est décroissante, et quand $x > x_3$, $y(t)$ est croissante.

De même pour $x(t)$, l'équation d'une isocline nulle est $y_{iso}(x) = \dfrac{aK}{b}(1-x)(x+\dfrac{c}{K})$. Pour $y < y_{iso}(x)$, $x(t)$ est croissante, et pour $y > y_{iso}(x)$, $x(t)$ est décroissante.

Illustration de ces variations sur le portrait de phase ci-dessous :





Pour les mêmes valeurs de paramètres que les graphes précédents, la valeur numérique obtenue pour le point d'équilibre P_3 est $(0,278\ ,\ 0,535)$.

En résumé : le modèle (S) peut décrire l'évolution au cours du temps de notre système (pucerons, coccinelles).

$$(S)\begin{cases} \dot{x} = ax(1-x) - b\frac{x}{Kx+c}y_m y \\ \dot{y} = e\frac{x}{x+\frac{c}{K}}y - dy \end{cases}$$

Il possède trois équilibres dont un qui peut être éventuellement stable, selon les paramètres, et on se placera toujours dans ce cas.

1.2 Contrôle et critère

Dans notre problème, $x(t)$ représente l'évolution de la population des pucerons, et $y(t)$ celle de la population des coccinelles. Notre but est d'agir sur la population de pucerons (de la faire diminuer) en introduisant des coccinelles.

Ce qui correspond dans notre système (S), à agir sur la condition initiale y_0, ou à intervenir sur $y(t)$ ponctuellement, ou encore à introduire un terme de contrôle. Dans ce dernier cas, le système devient :

$$(S_u)\begin{cases} \dot{x} = ax(1-x) - b\frac{x}{Kx+c}y_m y \\ \dot{y} = e\frac{x}{x+\frac{c}{K}}y - dy + u \end{cases}$$

Ici, u est une fonction du temps et représente une vitesse d'introduction de coccinelles (ou plutôt d'une biomasse mise à l'échelle).

De même qu'il nous faut choisir quel type de contrôle adopter, nous devons déterminer un critère qui tienne compte du coût de l'introduction des coccinelles, et de la limitation du nombre de pucerons. Ce critère, qui sera le plus souvent à minimiser, servira d'élément de comparaison.

Pour ce qui est du coût des coccinelles, il suffit, suivant le contrôle choisi, d'opter pour un terme de la forme αy_0 ou $\alpha \int_0^T u(t)dt$.
Cela correspond à la quantité de coccinelles apportées pondérée par une variable ($\alpha > 0$) de coût par unité. Dans le cas du modèle (S), la quantité en question est y_0, dans le cas de (S_u), c'est $\int_0^T u(t)dt$.
La variable T désigne le temps final de l'expérience.

Quant à la population de pucerons, la première idée est de fixer un seuil à ne pas dépasser et de regarder la dernière valeur : $x(T) \leq x_{max}$. Mais regarder seulement la population à l'instant T ne permet pas de tenir compte des fluctuations de la population au cours du temps, il faudrait plutôt traiter $\forall t \in [0, T], x(t) \leq x_{max}$. Néanmoins pour évaluer les dégâts causés sur les cultures, il faut prendre en compte la somme des pucerons. Ainsi, pour considérer la population au cours du temps, on peut opter pour un critère de la forme $\int_0^T x(t)dt$ ou $\int_0^T x^2(t)dt$.

Afin de se faire une idée des ordres de grandeur, la valeur numérique de ce dernier a été calculée pour les cas correspondants aux premiers graphes présentés :

x_0	y_0	$\int_0^T x(t)dt$	$\int_0^T x^2(t)dt$
0,0375	0	91,79	89,39
0,0375	0,015	41,27	23,64

Les critères possibles sont donc $\alpha y_0 + \displaystyle\int_0^T x(t)dt$ ou $\displaystyle\int_0^T (x(t) + \alpha u(t))dt$ ou encore de type quadratique, $\displaystyle\int_0^T (x^2(t) + \alpha u^2(t))dt$.

La difficulté est alors de quantifier α, de comprendre sa signification et de pouvoir évaluer sa valeur. A priori, cette variable représente le rapport entre le coût des coccinelles et le coût des dégâts causés par les pucerons.

1.3 Etat de l'art

Des travaux sur le contrôle d'un système de plusieurs populations ont déjà été effectués.

B. S. Goh, G. Leitmann, et T. L. Vincent ([6]) traitent du contrôle optimal d'un système proie-prédateur.
Ils appliquent la théorie du contrôle optimal au modèle de Lotka-Volterra :

$$\begin{cases} \frac{dN_1}{dt} = (\alpha_1 - \beta_1 N_2)N_1, \\ \frac{dN_2}{dt} = (\beta_2 N_1 - \alpha_2)N_2, \end{cases}$$

où, à l'instant t, $N_1(t)$ désigne le nombre de proies, et $N_2(t)$ celui de prédateurs.

Deux types de variables de contrôle sont utilisées. Une de ces variables est le taux d'introduction de prédateurs (ou de proies), de la forme $v(t)$ (ou $u(t)$). L'autre variable représente le taux d'application d'un insecticide, de la forme $-b_1u(t)N_1(t)$ (ou $-b_2u(t)N_2(t)$).

Leur critère est une combinaison du coût des dégats sur les cultures causés par les proies et du coût de l'introduction d'insectes supplémentaires ou de l'insecticide utilisé. Il est, par exemple, de la forme $J = \int_0^T (c_1N_1 + c_2u)dt$.

Dans certains cas, en imposant $0 \leq u \leq u_{max}$, le contrôle optimal trouvé est une combinaison de $u = 0$ et $u = u_{max}$ (solution dite "bang-bang"). Leurs résultats montrent entre autre qu'il est possible de controler la population de proies en utilisant un insecticide qui tue seulement les prédateurs, ou en introduisant des proies.

W. M. Getz ([5]), W. Silvert et W. R. Smith ([9]) se sont également intéressés à ce type de problème.

Chapitre 2

Recherche de la condition initiale optimale

Cette première étape consiste à rechercher, pour le système (S), la condition initiale y_0, pour un x_0 donné, qui va minimiser la fonction coût :

$$J(y_0) = \alpha y_0 + \int_0^T x(t)dt.$$

Tout d'abord, simplifions le problème.

2.1 Cas d'un système linéaire

Regardons ce qu'il se passe dans le cadre d'un système linéaire $\dot{X} = AX + B$ où $X = (x, y)^T$, $B = (B_1, B_2)^T$ et A est une matrice carrée.

Dans ce cas, minimiser par rapport à X_0 le critère $J(X_0) = CX_0 + \displaystyle\int_0^T pX(t)dt$, où $C = (C_1, C_2)$ et $p = (p_1, p_2)$, revient à choisir pour X_0 l'une des bornes de son intervalle de définition.
En effet, la solution de $\dot{X} = AX + B$ est donnée par la formule :

$$X(t) = e^{At}X_0 + \int_0^t e^{A(t-\tau)}Bd\tau.$$

Si on remplace X par son expression dans celle du critère, on obtient :

$$J(X_0) = (C + pI_T)X_0 + \int_0^T p \int_0^t e^{A(t-\tau)}Bd\tau dt,$$

avec $I_T = \displaystyle\int_0^T e^{At}dt$. Le terme $\int_0^T p \int_0^t e^{A(t-\tau)}Bd\tau dt$ est constant par rapport à X_0, de même que le vecteur $C + pI_T$.
La fonction J est donc linéaire en X_0, et par conséquent en y_0.

Donc si y_0 est imposé entre 0 et 1, le minimum de J sera atteint pour $y_0 = 0$ dans le cas où $(C + pI_T).(0,1)^T > 0$, et il sera atteint pour $y_0 = 1$ dans le cas contraire $((C + pI_T).(0,1)^T < 0)$.

2.2 Cas d'un système non linéaire plus simple

Revenons à un système non linéaire, mais toutefois moins compliqué que celui qui nous intéresse :

$$(S_2) \begin{cases} \dot{x} = a_2 x(1-x) - b_2 xy \\ \dot{y} = e_2 xy - d_2 y \end{cases}$$

Le taux de prédation, qui était de la forme $f(x) = \frac{bx}{x+c}$ dans le modèle (S), est ici de la forme $f(x) = b_2 x$. Ceci constitue la seule différence dans l'écriture de ces 2 modèles.

2.2.1 Etude de (S_2)

Les points d'équilibre de ce système sont : $P_1 = (0\ ,\ 0)$, $P_2 = (1\ ,\ 0)$ et $P_3 = (x_3 = \frac{d_2}{e_2}\ ,\ y_3 = \frac{a_2}{b_2}(1 - \frac{d_2}{e_2}))$.

Regardons la stabilité de ces points :
P_1 est instable,
P_2 est stable sous la condition $e_2 - d_2 < 0$, or $x_3 = \frac{d_2}{e_2} < 1$ donc la condition n'est pas vérifiée et P_2 est instable,
et P_3 est stable sans condition.

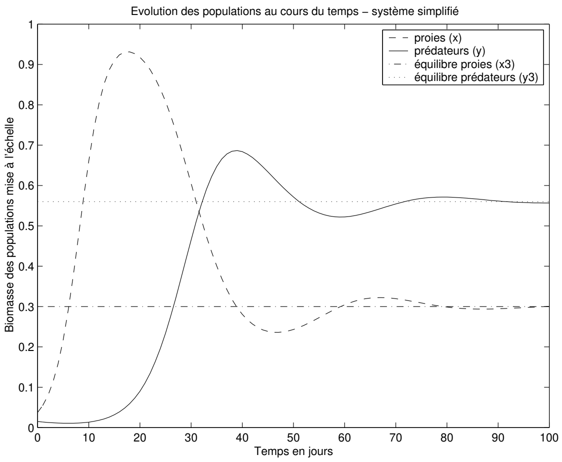
Pour ce modèle, les équations des isoclines nulles sont :

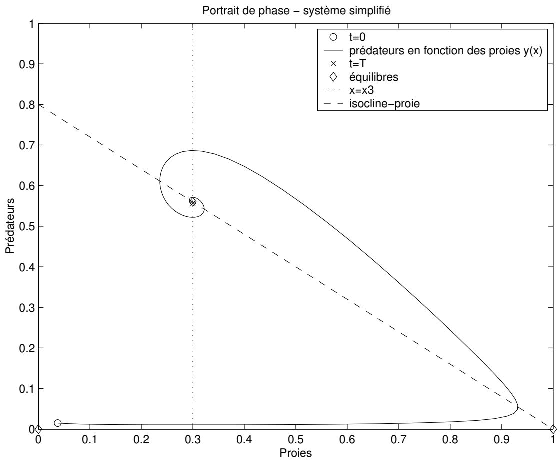
$$\begin{cases} y_{iso}(x) = \frac{a_2}{b_2}(1-x) \\ x = x_3 \end{cases}$$

Les graphes suivants ont été obtenus, en choisissant pour les paramètres

a_2	b_2	e_2	d_2
$a = 0,4$	$\frac{by_m}{0,5K+c} = 0,5$	$\frac{e}{0,5+\frac{c}{K}} = 0,33$	$d = 0,1$

pour les conditions initiales $(x_0, y_0) = (0,0375\ ,\ 0,015)$ et un intervalle de temps $[0\ ,\ 100]$.





La valeur numérique du point d'équilibre P_3 est ici $(0,3\ ,\ 0,56)$.

Dans ce cas, les valeurs obtenues pour les critères correspondant à la population de pucerons sont :

x_0	y_0	$\int_0^T x(t)dt$	$\int_0^T x^2(t)dt$
0,0375	0,015	40,84	21,96

En choisissant bien les paramètres a_2, b_2, e_2, et d_2, le système (S_2) donne des résultats proches de ceux obtenus avec le système (S).

2.2.2 Minimisation du critère

Le critère que l'on désire minimiser est :

$$J(y_0) = \alpha y_0 + \int_0^T x(t)dt.$$

Or l'expression de $\dot{y}$ ($\dot{y} = e_2xy - d_2y$) amène, pour $y_0 \neq 0$, $x = \dfrac{1}{e_2}\dfrac{\dot{y}}{y} + \dfrac{d_2}{e_2}$. En intégrant sur $[0, T]$, on obtient :

$$\int_0^T x(t)dt = \frac{1}{e_2}(\ln(y(T)) - \ln(y_0)) + \frac{d_2}{e_2}T.$$

D'où

$$J(y_0) = \alpha y_0 - \frac{1}{e_2}\ln(y_0) + \frac{1}{e_2}(d_2T + \ln(y(T))).$$

En supposant T assez grand, $y(T)$ tend vers l'équilibre qui est stable, quelle que soit la condition initiale, on peut considérer $y(T)$ comme indépendant de y_0. Alors, en dérivant J par rapport à y_0, on obtient $\alpha - \frac{1}{e_2y_0}$.

Pour trouver y_0 qui minimise J, on cherche y_0 qui annule $\dfrac{dJ}{dy_0}(y_0) = \alpha - \dfrac{1}{e_2y_0}$. Ce qui donne $y_0^* = \dfrac{1}{e_2\alpha}$.

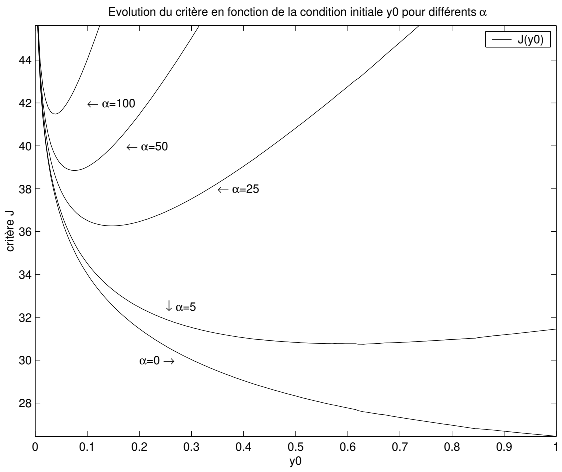
Pour ce modèle (S_2), le choix de la condition initiale pour minimiser le critère va être dépendant du coût (inversement proportionnelle).

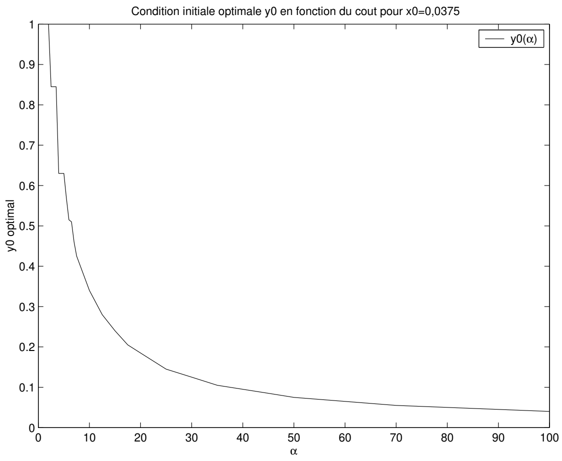
2.3 Cas du modèle (S)

On cherche la valeur de y_0 de façon à minimiser le critère $J(y_0) = \alpha y_0 + \int_0^T x(t)dt$ pour x_0 fixé, pour le système (S).

Pour ce faire, on fixe la valeur de α, on calcule $J(y_0)$ pour plusieurs y_0 dans $[0\ ,\ 1]$, et on choisit celui pour lequel la valeur calculée est la plus petite.

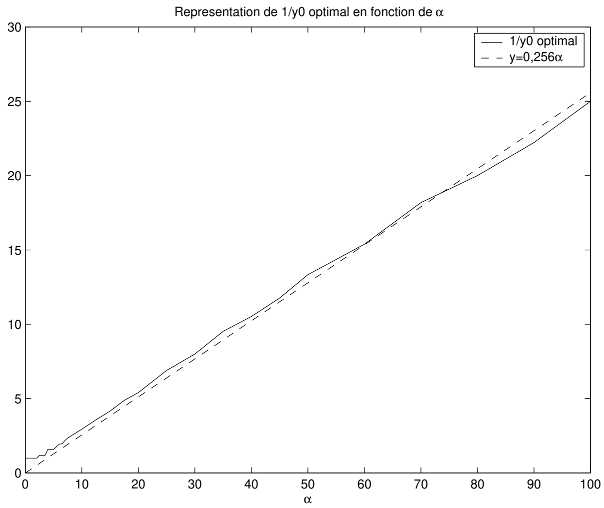
Pour différentes valeurs de α comprises entre 0 et 100, on a obtenu les résultats présentés sur les graphes suivants :





D'après le second graphe, il semble que l'on ait une relation proche de $y_0^* = \frac{k_1}{\alpha}$.

Sur le graphe suivant, on a représenté $\frac{1}{y_0^*}$ en fonction de α ainsi que la droite $y = k_1\alpha$ avec $k_1 = 0,256$ calculé à l'aide de la fonction *regress* de Matlab (cette fonction fait les calculs pour une régression linéaire).



La courbe obtenue, est effectivement assez proche d'une droite.

Ainsi, la condition initiale optimale pourrait se déduire du poids α utilisé dans le critère.

Sur le tableau ci-dessous, se trouvent les valeurs obtenues pour différents α, pour le système (S) défini par les mêmes paramètres et une condition initiale $x_0 = 0,0375$:

α	0	0,5	1	3,5	5
y_0^*	1,0	1,0	1,0	0,845	0,63
$J(y_0^*)$	26,45	26,95	27,45	29,76	30,74

α	10	20	25	40	50	100
y_0^*	0,34	0,185	0,145	0,095	0,075	0,04
$J(y_0^*)$	32,99	35,45	36,26	38,01	38,84	41,48

La valeur du critère est fortement réduite par rapport à la valeur obtenue en considérant une population de proies en l'absence de prédateurs.

Si la condition initiale est bien choisie, il est donc possible d'avoir un coût réduit, sans avoir à agir sur le système au cours du temps.

Pour le modèle (S), la condition initiale optimale peut être approchée en utilisant une valeur inversement proportionnelle au coût.

Chapitre 3

Répartition de la condition initiale optimale

Maintenant que l'on a déterminé la condition initiale y_0^* optimale pour le critère $J(y_0) = \alpha y_0 + \int_0^T x(t)dt$, on regarde si l'on peut améliorer la valeur du critère en répartissant la quantité y_0^* de façon à introduire $(y_0^* - n_1)$ à $t = 0$ et n_1 à $t = t_1$, avec $0 \leq n_1 \leq y_0^*$ et $0 \leq t_1 \leq T$. Il s'agit alors de déterminer n_1 et t_1.
Dans ce cas le critère devient :

$$J(n_1, t_1) = \alpha(y_0^* - n_1) + \int_0^{t_1} x(t)dt + \alpha n_1 + \int_{t_1}^{T} x(t)dt = \alpha y_0^* + \int_0^T x(t)dt.$$

3.1 Cas d'un système linéaire

Revenons au système linéaire $\dot{X} = AX + B$. Pour la répartition de la condition initiale optimale ($X_0^* = (x_0, y_0^*)^T$), le problème s'écrit :

$$\begin{cases} \dot{Y} = AY + B, t \in [0\ ,\ t_1], & Y(0) = X_0^* - N_1, \\ \dot{Z} = AZ + B, t \in [t_1\ ,\ T], & Z(t_1) = Y(t_1) + N_1, \end{cases}$$

où $N_1 = (0\ ,\ n_1)^T$, et la solution X vaut Y sur $[0\ ,\ t_1[$ et Z sur $[t_1\ ,\ T]$.

Sur ces deux intervalles, la solution s'écrit :

$$\begin{cases} Y(t) = e^{At}(X_0^* - N_1) + \int_0^t e^{A(t-\tau)}Bd\tau, & t \in [0\ ,\ t_1], \\ Z(t) = e^{A(t-t_1)}(Y(t_1) + N_1) + \int_{t_1}^t e^{A(t-\tau)}Bd\tau, & t \in [t_1\ ,\ T]. \end{cases}$$

Le critère à minimiser est ici $J(n_1, t_1) = CX_0^* + \displaystyle\int_0^{t_1} pY(t)dt + \int_{t_1}^{T} pZ(t)dt$, avec $C = (C_1, C_2)$ et $p = (p_1, p_2)$.

En utilisant l'écriture de Y et Z, on obtient :

$$\begin{cases} \int_0^{t_1} pY(t)dt = pI_{t_1}(X_0^* - N_1) + p\int_0^{t_1}\int_0^t e^{A(t-\tau)}Bd\tau dt, \\ \int_{t_1}^{T} pZ(t)dt = pI_{(T-t_1)}(Y(t_1) + N_1) + p\int_{t_1}^{T}\int_{t_1}^t e^{A(t-\tau)}Bd\tau dt, \end{cases}$$

où $I_t = \int_0^t e^{A\tau} d\tau$.

Si on regarde la dérivée partielle de J par rapport à n_1, on a

$$\frac{\partial J}{\partial n_1}(n_1, t_1) = \frac{\partial J}{\partial N_1}(n_1, t_1) \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

et

$$\frac{\partial J}{\partial N_1}(n_1, t_1) = p(I_{(T-t_1)}(1 - e^{At_1}) - I_{t_1}).$$

Or $I_{(T-t_1)} e^{At_1} = \displaystyle\int_0^{T-t_1} e^{A(t+t_1)} dt = \int_{t_1}^{T} e^{At} dt.$

D'où :

$$\begin{aligned}
\frac{\partial J}{\partial N_1}(n_1, t_1) &= p(I_{(T-t_1)} - I_{t_1} - \textstyle\int_{t_1}^T e^{At} dt) \\
&= p(\textstyle\int_0^{T-t_1} e^{At} dt - \int_0^{t_1} e^{At} dt - \int_{t_1}^T e^{At} dt) \\
&= p(\textstyle\int_0^{T-t_1} e^{At} dt - \int_0^{T} e^{At} dt) \\
\frac{\partial J}{\partial N_1}(n_1, t_1) &= -p \textstyle\int_{T-t_1}^{T} e^{At} dt
\end{aligned}$$

Cette dérivée ne dépend que de t_1.
La fonction J est linéaire en N_1, donc en n_1. La minimisation de J par rapport à cette variable dépendra du signe de $\frac{\partial J}{\partial n_1}$. Si $\frac{\partial J}{\partial n_1}(t_1) > 0$, alors il faudra poser $n_1 = 0$, et si $\frac{\partial J}{\partial n_1}(t_1) < 0$, il faudra poser $n_1 = y_0^*$ pour atteindre le minimum de J.

3.2 Cas d'un système non linéaire plus simple

Revenons maintenant au cas du système (S_2). De même que dans le chapitre précédent, on utilise l'expression de $\dot{y}$: pour $y_0 \neq 0$, $x = \dfrac{1}{e_2}\dfrac{\dot{y}}{y} + \dfrac{d_2}{e_2}$.

En intégrant sur $[0\ ,\ t_1]$ et sur $[t_1\ ,\ T]$, on obtient :

$$\left\{ \begin{array}{l} \int_0^{t_1} x(t)dt = \frac{1}{e_2}(\ln(y_1) - \ln(y_0^* - n_1)) + \frac{d_2}{e_2} t_1 \\ \int_{t_1}^{T} x(t)dt = \frac{1}{e_2}(\ln(y(T)) - \ln(y_1 + n_1)) + \frac{d_2}{e_2}(T - t_1) \end{array} \right.$$

où y_1 désigne la valeur de $y(t)$ à l'instant t_1 avant que l'on ajoute la quantité n_1.

Ainsi le critère $\alpha y_0^* + \int_0^T x(t)dt$ devient

$$J(n_1, t_1) = \frac{1}{e_2}(\ln(y_1) - \ln(y_1 + n_1) - \ln(y_0^* - n_1)) + \alpha y_0^* + \frac{1}{e_2}(d_2 T + \ln(y(T))),$$

en considérant T assez grand pour que $y(T)$ soit à l'équilibre, et soit ainsi indépendant de la condition initiale.

On peut alors regarder la dérivée partielle de la fonction J par rapport à t_1 :

$$\begin{aligned}
\frac{\partial J}{\partial t_1}(n_1\ ,\ t_1) &= \frac{1}{e_2}\left(\frac{\dot{y}(t_1)}{y_1} - \frac{\dot{y}(t_1)}{y_1 + n_1}\right) \\
&= \frac{e_2 x(t_1) y_1 - d_2 y_1}{e_2}\left(\frac{1}{y_1} - \frac{1}{y_1 + n_1}\right) \\
\frac{\partial J}{\partial t_1}(n_1\ ,\ t_1) &= \frac{n_1(e_2 x(t_1) - d_2)}{e_2(y_1 + n_1)}
\end{aligned}$$

Le couple (n_1, t_1) qui minimise la fonction J doit vérifier $\frac{\partial J}{\partial n_1}(n_1\ ,\ t_1) = 0$ et $\frac{\partial J}{\partial t_1}(n_1\ ,\ t_1) = 0$. La seconde condition donne $n_1 = 0$ ou t_1 tel que $x(t_1) = \frac{d_2}{e_2}$. Ce dernier cas correspond à un instant où $x(t)$ atteint la valeur de l'équilibre, ou plutôt coupe l'isocline nulle d'équation $x = x_3$.

3.3 Cas du modèle (S)

Pour résoudre numériquement ce problème avec le système (S), une fonction de 2 paramètres (n_1 et t_1) qui retourne la valeur $\alpha y_0^* + \int_0^T x(t)dt$, a été créée telle que la condition initiale à $t = 0$ soit $y_0 = y_0^* - n_1$, une première intégration de (S) soit faite de 0 à t_1, la quantité n_1 soit ajoutée à $y(t_1)$ suivie d'une seconde integration de (S) de t_1 à T.

Il faut alors minimiser cette fonction par rapport à ses 2 paramètres. Pour cela, la fonction *fmincon* de Matlab a été utilisée. Cette fonction permet de trouver un minimum d'une fonction de plusieurs variables, tout en respectant des contraintes imposées pour ces variables. En effet dans notre cas, n_1 doit être compris entre 0 et y_0^*, et t_1 entre 0 et T.

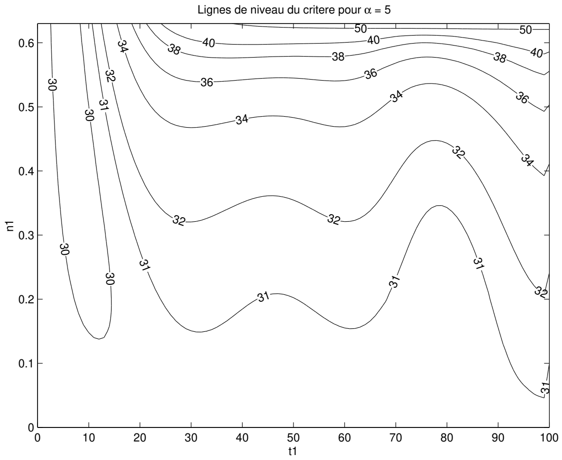
D'après les résultats obtenus, on remarque que distribuer la quantité y_0^* en deux temps améliore la valeur du critère.

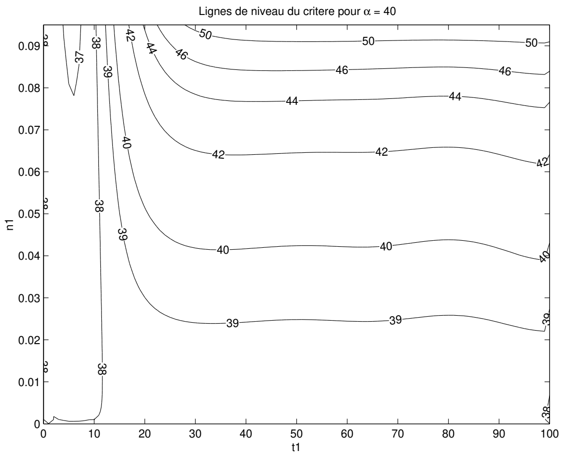
α	0	0,5	1	3,5	5
$y_0^* - n_1^*$	0,548	0,548	0,548	0,478	0,258
n_1^*	0,452	0,452	0,452	0,367	0,372
t_1^*	12	12	12	12	9
$J(n_1^*, t_1^*)$	24,81	25,31	25,81	28,42	29,64

α	10	20	25	40	50	100
$y_0^* - n_1^*$	0,0058	0	0	0	0	0
n_1^*	0,334	0,185	0,145	0,095	0,075	0,04
t_1^*	7	5	5	3	3	3
$J(n_1^*, t_1^*)$	31,89	34,29	35,11	37,16	37,99	40,63

Ces résultats proviennent de calculs faits sous les mêmes conditions (initiales et paramètres) que dans le chapitre précédent.

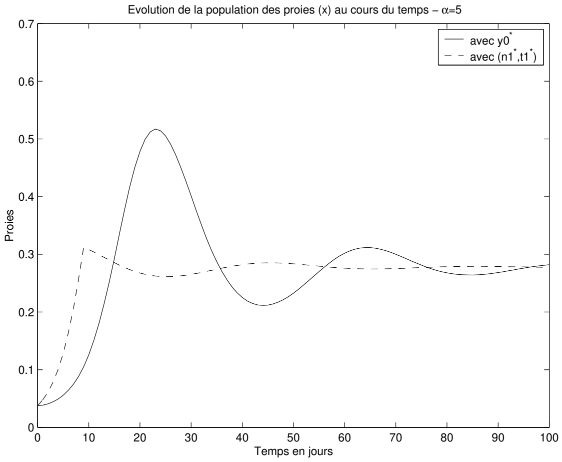
Quand α est petit (y_0^* grand), un partage de la quantité y_0^* est fait, néanmoins le temps au bout duquel on introduit la deuxième part est court comparé à l'intervalle de temps considéré (100 jours).
Quand α est grand (y_0^* petit), n_1 atteint son maximum, c'est-à-dire qu'à $t = 0$, rien n'est introduit ($y_0 = 0$). Dans ce cas, il semble que cela soit plus intéressant d'attendre un peu et d'introduire toute la quantité disponible au même moment.

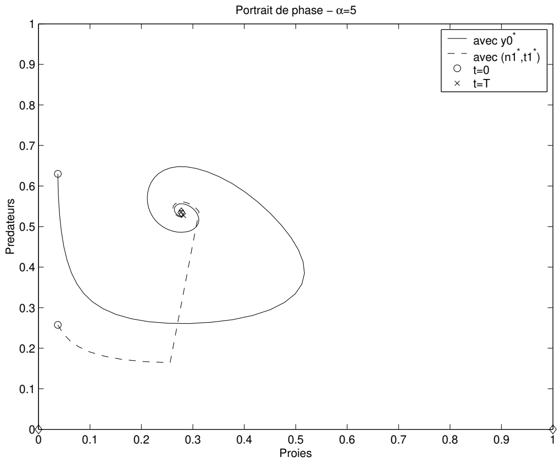




Sur les graphes précédents, le critère $J(n_1, t_1)$ a été représenté pour 2 valeurs de α (5 et 40). On peut ainsi remarquer que les résultats obtenus avec *fmincon* sont proches du minimum.

Les courbes suivantes sont celles obtenues, pour $\alpha = 5$, avec la condition initiale optimale y_0^* et avec le couple de paramètres optimaux calculés $(n_1^*\ ,\ t_1^*)$:





Le premier graphe montre uniquement l'évolution de $x(t)$ car c'est la fonction qui nous intéresse pour le critère. Dans le cas de la répartition de y_0^*, la courbe est plus aplatie autour de l'équilibre.
Le portrait de phase montre que l'ajout au cours au temps t_1^* de la quantité n_1^* permet d'atteindre plus vite l'équilibre (x_3, y_3). D'ailleurs, il semble que le changement se fait au moment où $x(t)$ dépasse un peu la valeur x_3, ce qui est cohérent avec le résultat théorique obtenu pour le modèle plus simple (S_2).

En conclusion : la répartition de la condition initiale optimale y_0^* en deux temps améliore la valeur du critère $J(y_0^*)$, en amenant les fonctions $x(t)$ et $y(t)$ plus rapidement autour de leur équilibre.

Cette répartition peut se faire de deux façons suivant la valeur de α. Quand α est petit, un partage est fait, tandis que pour α grand, $n_1^* = y_0^*$, c'est-à-dire toute la quantité disponible est introduite d'un coup mais pas au début de l'expérience.

Par ailleurs, dans le cas d'un équilibre stable, l'amélioration du critère va certainement être liée à la vitesse à laquelle le système tendra vers l'équilibre.

Chapitre 4

Résolution du problème linéarisé

Le critère peut être minimisé en choisissant correctement la condition initiale, et il peut être amélioré en répartissant cette dernière en deux temps.

On voudrait maintenant trouver une fonction de contrôle optimale $u(t)$ continue, utilisée dans le système :

$$(S_u) \begin{cases} \dot{x} = ax(1-x) - b\frac{x}{Kx+c}y_m y \\ \dot{y} = e\frac{x}{x+\frac{c}{K}}y - dy + u \end{cases}$$

Pour simplifier le problème, on peut le linéariser pour appliquer les techniques utilisant l'équation de Riccati (D'Andréa-Novel et Cohen de Lara, 1993).

4.1 Optimisation du problème linéarisé

Le système (S) peut s'écrire $\dot{X} = F(X)$ avec $X = (x,y)^T$ et $F(X) = \begin{pmatrix} f_1(x,y) \\ f_2(x,y) \end{pmatrix}$, f_1 et f_2 étant définies par le second membre de (S).
Le système (S_u), quant à lui, peut s'écrire $\dot{X} = F(X) + Bu$ avec $B = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$.

Les paramètres du système (S) sont choisis tels que P_3 soit stable.

On peut alors linéariser (S) autour de ce point P_3. Le problème devient $\dot{Z} = J_3 Z + Bu$ avec J_3 la jacobienne de $F(X)$ au point P_3.

En effet, pour linéariser le système, on se place autour du point P_3 : $X = P_3^T + Z$ avec X vérifiant le système $\dot{X} = F(X) + Bu$ et P_3 vérifiant $F(P_3^T) = 0$. On obtient alors $\dot{X} = \dot{Z}$ et $\dot{X} = F(P_3^T + Z) + Bu$, or $F(P_3^T + Z) = F(P_3^T) + DF(P_3^T)Z = J_3 Z$. On a donc le système linéaire $\dot{Z} = J_3 Z + Bu$.

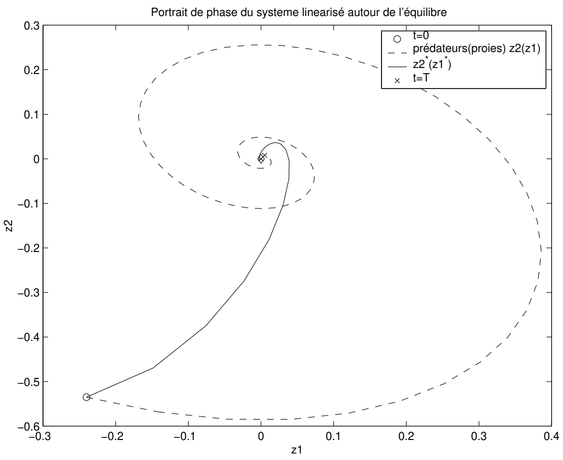
Ainsi pour minimiser le critère $\displaystyle\int_0^T (z_1^2(t) + \alpha^2 u^2(t))dt$, il suffit de résoudre l'équation de Riccati différentielle :

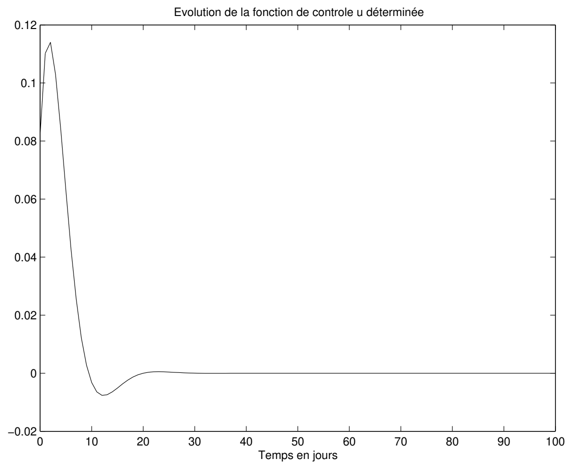
$$\begin{cases} \dot{R}(t) + J_3^T R(t) + R(t) J_3 - R(t) B \alpha^{-1} B^T R(t) + Q = 0 \\ R(T) = 0 \end{cases}$$

avec Q définie par le critère : $\displaystyle\int_0^T (Z^T(t) Q Z(t) + \alpha^2 u^2(t))dt$, d'où $Q = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$.
$R(t)$, la solution de l'équation de Riccati différentielle, est une matrice symétrique.

La commande pour obtenir le coût minimum est $u^*(t) = -\alpha^{-2} B^T R(t) Z^*(t)$, et le coût minimum vaut $Z_0^T R(0) Z_0$. $Z^*(t)$ est la trajectoire optimale correspondant à l'application de la commande précédente, soit $\dot{Z}^* = J_3 Z^* + B u^* = (J_3 - B\alpha^{-2} B^T R(t)) Z^*$ avec $Z_0^* = Z_0$.

Exemple de résultat obtenu pour $\alpha = 1$ et $X_0 = (0,0375\ ,\ 0)^T$:
ce qui donne $Z_0 = P_3^T - X_0 = (-0,2403\ ,\ 0,535)^T$, et le coût minimum égal à 0,1135,





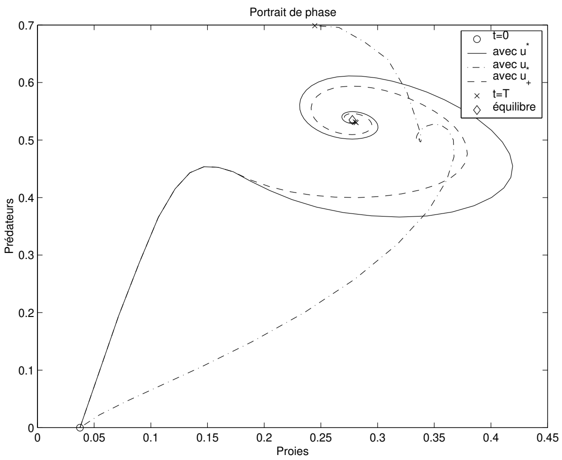
4.2 Application au problème non linéaire

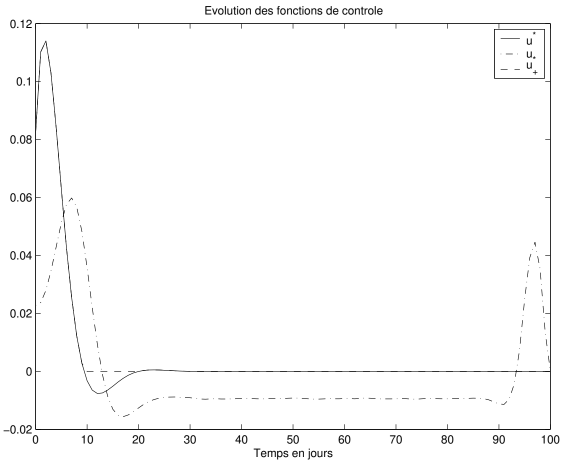
On revient au système (S_u), et on désire lui appliquer les résultats obtenus par l'optimisation précédente dans les mêmes conditions (mêmes valeurs pour α et les paramètres et mêmes conditions initiales pour les 2 problèmes).

Néanmoins il existe trois possibilités pour passer de l'un à l'autre :

- $u^*(t)$, commande optimale déterminée pour l'optimisation quadratique du système linéaire, est le contrôle appliqué au système non linéaire,
- $u^*(t)$ étant de la forme $k_1(t)z_1(t) + k_2(t)z_2(t)$, on conserve k_1 et k_2, et on applique le contrôle $u(t) = k_1(t)x(t) + k_2(t)y(t)$,
- $u^*(t)$ ayant souvent des valeurs négatives, on choisit comme contrôle $u_+^*(t) = max(0, u^*(t))$.

Pour $\alpha = 1$ et $X_0 = (0,0375\ ,\ 0)^T$, on obtenu le résultat suivant :





Sur ces graphes, il semble que la deuxième possibilité (contrôle $u(t) = k_1(t)x(t) + k_2(t)y(t)$) ne soit pas intéressante.

Contrôle	$\int_0^T (x(t) + \alpha u(t))dt$	$\int_0^T (x^2(t) + \alpha u^2(t))dt$
$u^*(t)$	27,80	7,98
$u(t)$	32,06	10,71
$u_+^*(t)$	27,45	7,65

Effectivement, en regardant les résultats numériques, la seconde possibilité donne un critère supérieur à ceux des deux autres. Les valeurs pour les commandes $u^*(t)$ et $u_+^*(t)$ sont sensiblement les mêmes. Néanmoins celle que l'on préfèrera est $u_+^*(t)$ car on obtient des coûts légèrement inférieurs, et de plus, cette commande est positive par définition. En effet un contrôle négatif à un instant donné pose un problème de signification et d'application.

Toutefois, malgré l'amélioration qu'apporte cette méthode (pour la même condition initiale, le système sans commande a un coût $\int_0^T x(t)dt = 91,79$), le résultat obtenu est moins bon que celui donné en utilisant un partage de la condition initiale optimale (chapitre précédent).

L'allure de $u_+^*(t)$ rappelle ce partage, en effet elle forme deux pics distincts, et proche dans le temps.

En conclusion : le passage par le système linéarisé permet d'obtenir une commande peu coûteuse (car souvent nulle) et donne une valeur pour le critère de l'ordre de celle obtenue pour la répartition de la condition initiale optimale, mais supérieure.

Chapitre 5

Résolution du problème non linéaire

On s'intéresse à présent au système (S_u) :

$$(S_u)\begin{cases} \dot{x} = ax(1-x) - b\frac{x}{Kx+c}y_m y \\ \dot{y} = e\frac{x}{x+\frac{c}{K}}y - dy + u \end{cases}$$

Le but est toujours de trouver la commande optimale $u(t)$ qui va minimiser le critère choisi.

5.1 Théorie du contrôle optimal

On présente ici quelques rappels concernant le contôle optimal (Hocking, 1997,Luenberger, 1979).

Le problème est de la forme :

$$\begin{cases} \dot{x}(t) = f(x(t), u(t)) \\ x(0) = x_0 \\ u(t) \in U \subset \mathbb{R}^m \\ J = \psi(x(T)) + \int_0^T l(x(t), u(t))dt \end{cases}$$

où T désigne le temps final, les fonctions x et f sont à valeurs dans $\mathbb{R}^n$, et J dans $\mathbb{R}$.

La fonction u est la fonction de contrôle, et J la fonction objectif qui est à minimiser.

Pour tenir compte du modèle, J est modifiée par l'introduction de la variable adjointe λ de $\mathbb{R}^n$, et devient :

$$\bar{J} = J - \int_0^T \lambda^T(t)(\dot{x}(t) - f(x(t), u(t)))dt.$$

Ainsi pour tout x et u, solutions de $\dot{x} = f(x, u)$, on a $\bar{J} = J$.

Soit H, la fonction Hamiltonienne définie par

$$H(\lambda, x, u) = \lambda^T f(x, u) + l(x, u).$$

Ceci nous permet d'écrire :

$$\bar{J} = \psi(x(T)) + \int_0^T (H(\lambda(t), x(t), u(t)) - \lambda^T(t)\dot{x}(t))dt.$$

On applique à notre problème, un contrôle $v(t) \in U$ tel que $\int_0^T |u_i(t)-v_i(t)|dt < \varepsilon$, pour chaque composante u_i de u, avec ε petit. L'effet de cette petite variation sur u va créer une nouvelle trajectoire : $x + \delta x$ et une variation sur $\bar{J}$:

$$\begin{aligned} \delta\bar{J} = \ & (\tfrac{\partial\psi}{\partial x}(x(T)) - \lambda^T(T))\delta x(T) + \textstyle\int_0^T (\tfrac{\partial H}{\partial x}(\lambda, x, u) + \dot{\lambda}^T)\delta x dt \\ & + \textstyle\int_0^T (H(\lambda, x, v) - H(\lambda, x, u))dt + o(\varepsilon). \end{aligned}$$

Les variables adjointes λ doivent être choisie de façon à faire disparaître tous les termes de $\delta\bar{J}$ sauf la dernière intégrale.

Ainsi, λ est solution de l'état adjoint :

$$\dot{\lambda}^T(t) = -\lambda^T(t)\frac{\partial f}{\partial x}(x(t), u(t)) - \frac{\partial l}{\partial x}(x(t), u(t)),$$

avec la condition finale $\lambda^T(T) = \frac{\partial\psi}{\partial x}(x(T))$.

On déduit alors de ces calculs le principe du minimum.

Principe du minimum (Pontryagin) :

Soit $u(t) \in U$ et $x(t)$, contrôle optimal et trajectoire pour le problème. Alors il existe une trajectoire adjointe $\lambda(t)$ telle que $u(t)$, $x(t)$ et $\lambda(t)$ vérifient :

$$\left\{ \begin{array}{ll} \dot{x}(t) = f(x(t), u(t)) & x(0) = x_0 \\ \dot{\lambda}^T(t) = -\lambda^T(t)\frac{\partial f}{\partial x}(x(t), u(t)) - \frac{\partial l}{\partial x}(x(t), u(t)) & \lambda^T(T) = \frac{\partial\psi}{\partial x}(x(T)) \\ H(\lambda(t), x(t), u(t)) \leq H(\lambda(t), x(t), v) & \forall v \in U \end{array} \right.$$

avec $H(\lambda, x, u) = \lambda^T f(x, u) + l(x, u)$.

Et u optimale est déterminé par

$$\frac{\partial H}{\partial u} = 0 \text{ ou } \lambda^T\frac{\partial f}{\partial u} + \frac{\partial l}{\partial u} = 0$$

Ces derniers points décrivent des conditions nécessaires pour obtenir le contrôle optimal du problème.

5.2 Critère linéaire

On cherche la commande $u(t)$, tel que pour tout $t \in [0\ ,\ T]$, $0 \leq u(t) \leq u_{max}$, et qui minimise le critère $J = \int_0^T (x(t) + \alpha u(t))dt$.

Le modèle (S_u) s'écrit également :

$$(S_u)\left\{\begin{array}{l}\dot{x} = f_1(x, y) \\ \dot{y} = f_2(x, y) + u\end{array}\right.$$

5.2.1 Etude théorique

L'Hamiltonien s'écrit alors

$$H(\lambda, x, u) = \lambda_1 f_1(x, y) + \lambda_2 f_2(x, y) + \lambda_2 u + x + \alpha u.$$

Le problème est alors de minimiser cette fonction par rapport à u. Pour avoir une indication sur u, il faut regarder la dérivée partielle de H :

$$\frac{\partial H}{\partial u} = \lambda_2 + \alpha.$$

Elle est indépendante de u, on ne peut donc pas déterminer u directement de cette expression.

La fonction H est donc linéaire en u. La solution est alors dite de type "bang-bang". En effet, la fonction prendra alternativement la valeur des bornes de son ensemble de définition. Dans notre cas, quand $\lambda_2 + \alpha < 0$, on posera $u = u_{max}$ et quand $\lambda_2 + \alpha > 0$, on posera $u = 0$.

Il faut toutefois faire attention au cas où $\lambda_2 + \alpha = 0$, qui peut donner des trajectoires singulières.

Dans notre cas, quand $\alpha = 0$, il n'existe pas de trajectoire singulière.
En effet, supposons $\lambda_2 + \alpha = 0$, cela implique que pour tout $t \in [0\ ,\ T]$, $\lambda_2(t) = 0$ et donc $\dot{\lambda}_2(t) = 0$.
Or

$$\dot{\lambda}_2 = -\lambda_1 \frac{\partial f_1}{\partial y}(x, y) - \lambda_2 \frac{\partial f_2}{\partial y}(x, y) = -\lambda_1 \frac{\partial f_1}{\partial y}(x, y),$$

pour tout (x, y), donc cela donne, pour tout $t \in [0\ ,\ T]$, $\lambda_1(t) = 0$. Mais les variables adjointes ne peuvent pas, par définition, être nulles en même temps. Donc il n'y a pas de trajectoires singulières pour $\alpha = 0$.

Pour $\alpha > 0$, les calculs deviennent vite compliqués.

5.2.2 Algorithme utilisé

Afin de déterminer numériquement la commande $u(t)$ optimale pour notre problème, un algorithme utilisant une matrice de transition a été programmé (Bryson et Ho, 1975).

Cette méthode ressemble à celle des algorithmes dits "de tir" : il s'agit, en partant d'une condition initiale arbitraire, de se rapprocher petit à petit par des modifications sur cette condition, du point final que l'on désire atteindre. Ici ce point final, est la valeur des variables adjointes (λ_1, λ_2) au temps T. Cette valeur, définie par le choix du critère, doit être nulle.

Description de l'algorithme :

- *Etape 1*: Choix des conditions initiales pour les variables adjointes : $\lambda_1(0)$ et $\lambda_2(0)$.
- *Etape 2*: Intégration sur $[0\ ,\ T]$ du système :

$$\begin{cases} \dot{x} = f_1(x,y) \\ \dot{y} = f_2(x,y) + u \\ \dot{\lambda}_1 = -\lambda_1 \frac{\partial f_1}{\partial x}(x,y) - \lambda_2 \frac{\partial f_2}{\partial x}(x,y) - 1 \\ \dot{\lambda}_2 = -\lambda_1 \frac{\partial f_1}{\partial y}(x,y) - \lambda_2 \frac{\partial f_2}{\partial y}(x,y) \end{cases}$$

  en utilisant $u = 0$ si $\lambda_2 + \alpha > 0$, $u = u_{max}$ sinon.
- *Etape 3*: Stockage des valeurs finales des variables adjointes : $\mu(T) = (\lambda_1(T)\ ,\ \lambda_2(T))^T$.
- *Etape 4*: Calcul de la matrice de transition $M_t = \frac{\partial \mu(T)}{\partial \lambda(0)}$, définie par $\delta\mu(T) = M_t \delta\lambda(0)$.
  Ce calcul se fait par différenciation numérique directe : on applique une petite variation sur $\lambda_i(0)$:$\lambda_i(0) + \delta\lambda_i(0)$, on intègre le système, et on calcule $\delta\mu_i(T)$ correspondant. On obtient M_t par division: la première colonne par $\frac{1}{\delta\lambda_1(0)}\delta\mu_1(T)$, et la seconde par $\frac{1}{\delta\lambda_2(0)}\delta\mu_2(T)$.
- *Etape 5*: Choix de la valeur de $\delta\mu(T)$ pour se rapprocher de la valeur désirée $((0\ ,\ 0)^T)$: $\delta\mu(T) = -\varepsilon\mu(T)$ avec $0 < \varepsilon \leq 1$.
- *Etape 6*: Calcul de la variation à appliquer aux conditions initiales des variables adjointes : $\delta\lambda(0) = M_t^{-1}\delta\mu(T)$. Et ainsi : $\lambda_i(0) = \lambda_i(0) + \delta\lambda_i(0)$.
  Retour à l'étape 2.
- *Test d'arrêt*: sur la valeur finale des variables adjointes : $\mu(T)$ doit être proche de 0. On définit pour cela une variable de tolérance petite à laquelle $\mu(T)$ sera comparé.

Néanmoins, cet algorithme peut être parfois confronté à un problème de conditionnement pour la matrice de transition.

5.2.3 Résultats

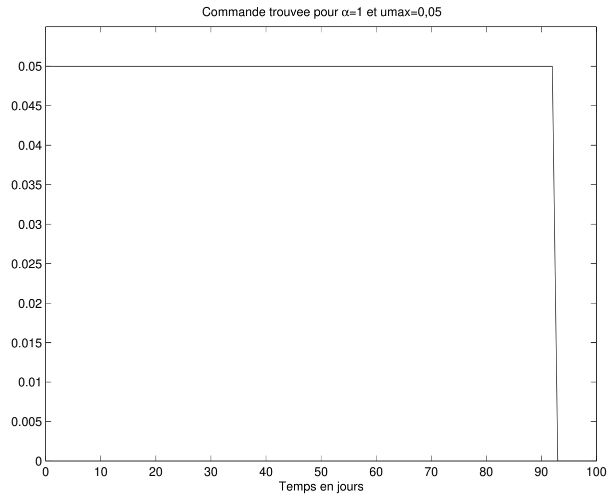
Les calculs ont été effectués avec toujours les mêmes valeurs pour les paramètres, et la condition initiale x_0. L'intervalle de temps considéré est $[0\ ,\ 100]$. La condition initiale y_0 est posée égale à 0.

Dans le tableau ci-dessous, sont présentés quelques résultats obtenus à l'aide de l'algorithme précédent :

u_{max}	α	0	0,5	1	3,5
0,01	J	27,57		28,54	30,83
0,025	J	18,58	19,80	21,02	26,79
0,05	J	8,32	10,77	13,03	

Le coût du critère est fortement lié à la valeur de u_{max}. Or cette valeur est difficile à définir.

De plus, les commandes optimales correspondant à ces résultats sont toutes de la forme : $u(t) = u_{max}, t \in [0\ ,\ t_1]$ et $u(t) = 0, t \in]t_1\ ,\ T]$, avec t_1 plus ou moins proche de 90 jours quand T en vaut 100 :



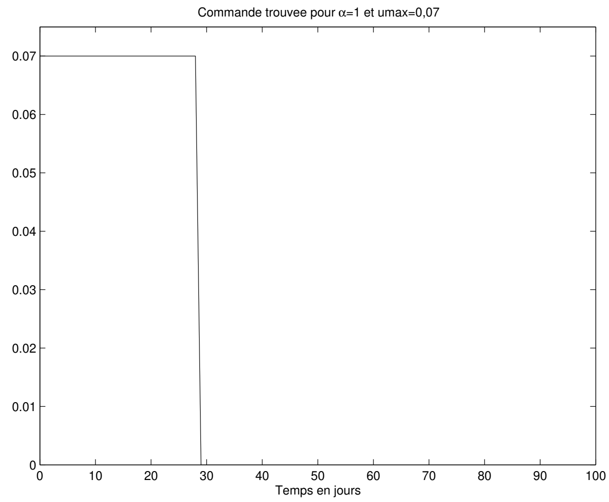
D'ailleurs l'algorithme utilisé semble avoir du mal à converger pour α ou u_{max} grands.

Une autre méthode, empirique, a été testée. Il s'agit, en partant d'une commande nulle, de résoudre le système (S_u), puis l'état adjoint alternativement jusqu'à ce que la commande u, déduite du signe de $\lambda_2(t) + \alpha$, ne varie plus.

Les résultats obtenus ainsi, sont cohérents avec les précédents pour les mêmes valeurs de α et u_{max}. Et pour certaines valeurs, il arrive que cette méthode ne converge pas mais oscille entre 2 solutions.

Par exemple, en prenant $\alpha = 1$ et $u_{max} = 0,07$, on obtient un coût $J = 24,23$

pour une commande de la forme :



Certes, le coût est plus élevée que pour $u_{max} = 0,05$, mais la commande à appliquer est plus raisonnable.

5.3 Critère quadratique

On cherche la commande $u(t)$ qui minimise le critère $J(u) = \int_0^T (x^2(t)+\alpha u^2(t))dt$, toujours pour notre modèle (S_u).

5.3.1 Etude théorique

L'Hamiltonien s'écrit ici

$$H(\lambda, x, u) = \lambda_1 f_1(x, y) + \lambda_2 f_2(x, y) + \lambda_2 u + x^2 + \alpha u^2.$$

Pour minimiser cette fonction, il faut regarder la dérivée partielle de H par rapport à u :

$$\frac{\partial H}{\partial u} = \lambda_2 + 2\alpha u.$$

Ainsi la commande optimale est donné par : $u(t) = -\frac{\lambda_2(t)}{2\alpha}$.

5.3.2 Algorithme utilisé

Dans ce cas, nous avons opté pour un algorithme du gradient (Bryson et Ho, 1975).

Précédemment, pour une petite variation sur u, on a obtenu :

$$\delta \bar{J} = \int_0^T (H(\lambda, x, v) - H(\lambda, x, u))dt = \int_0^T \frac{\partial H}{\partial u}(\lambda(t), x(t), u(t))\delta u(t)dt.$$

Le gradient du critère J est donc lié à $\frac{\partial H}{\partial u}$.

L'algorithme consiste à résoudre successivement le problème et son adjoint, en appliquant sur la fonction de contrôle, à chaque boucle, la variation $-\frac{1}{W}\frac{\partial H}{\partial u}$, $\frac{1}{W}$ désignant le poids accordé à cette variation qui est en fait une direction de descente. La valeur choisie au départ pour W est $\varepsilon\frac{\partial^2 H}{\partial u^2} = 2\alpha\varepsilon$, avec ε choisi entre 0 et 1. Néanmoins, elle peut changer au fil des itérations.

Description de l'algorithme :

- *Etape 1* : Choix d'une fonction de contrôle $u(t)$ initiale.
- *Etape 2* : Intégration sur $[0\ ,\ T]$ du système :

$$\begin{cases} \dot{x} = f_1(x,y) \\ \dot{y} = f_2(x,y) + u \end{cases}$$

- *Etape 3* : Intégration sur $[T\ ,\ 0]$ du système :

$$\begin{cases} \dot{\lambda_1} = -\lambda_1\frac{\partial f_1}{\partial x}(x,y) - \lambda_2\frac{\partial f_2}{\partial x}(x,y) - 2x \\ \dot{\lambda_2} = -\lambda_1\frac{\partial f_1}{\partial y}(x,y) - \lambda_2\frac{\partial f_2}{\partial y}(x,y) \end{cases}$$

  avec pour conditions finales : $\lambda_1(T) = \lambda_2(T) = 0$.
- *Etape 4* : Calcul de $I_J = \int_0^T \frac{1}{W}(\lambda_2(t) + 2\alpha u(t))^2 dt$.
- *Etape 5* : Calcul de la variation à appliquer à la fonction de contrôle : $\delta u(t) = -\frac{1}{W}(\lambda_2(t) + 2\alpha u(t))$.
  Retour à l'étape 2.
- *Test d'arrêt* : sur I_J, sa valeur doit être proche de 0. On définit pour cela une variable de tolérance petite à laquelle I_J sera comparé.

5.3.3 Résultats

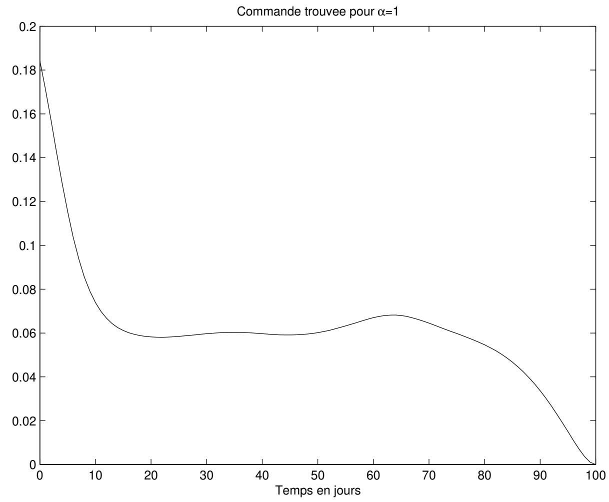
Les calculs ont été effectués avec toujours les mêmes valeurs pour les paramètres, et la condition initiale x_0. L'intervalle de temps considéré est $[0\ ,\ 100]$. La condition initiale y_0 est posée égale à 0.

α	0	0,5	1	3,5	5
$max(u^*)$	13,55	0,23	0,18	0,12	0,11
J	0,0007	0,31	0,55	1,47	1,89

α	10	20	25	40	50	100
$max(u^*)$	0,082	0,064	0,06	0,051	0,048	0,038
J	2,97	4,34	4,85	5,91	6,45	8,13

Dans ce tableau, on a reporté pour différentes valeurs de α, le maximum de la fonction de contrôle optimale u^* déterminée, et la valeur du critère. Il faut faire attention car ici le critère est de la forme $\int_0^T (x^2(t) + \alpha u^2(t))dt$, contrairement à la plupart des résultats présentés dans les cas précédents.

Pour comparaison, pour $\alpha = 1$, on a obtenu $\int_0^T (x^2(t) + \alpha u^2(t))dt = 0,55$ et si on effectue le calcul pour le critère linéaire : $\int_0^T (x(t) + \alpha u(t))dt = 8,73$. La commande optimale déterminée dans ce cas est représentée ci-dessous :



Pour toutes les valeurs de α testées, la commande optimale déterminée est de cette forme, c'est-à-dire, fortement décroissante, puis légèrement stationnaire, et enfin décroissante jusqu'à arriver à 0.

En conclusion : ces commandes minimisent assez bien le critère mais elles ne sont pas très intéressantes du fait qu'elles ne sont jamais nulles. En effet, une commande de la sorte voudrait dire qu'il faut introduire des coccinelles en permanence, ce qui n'est pas souhaitable sur un plan pratique.

Conclusion

Après avoir défini un modèle proie-prédateur pouvant décrire le système formé par les pucerons et les coccinelles, et un critère permettant d'évaluer l'efficacité du traitement biologique, nous avons testé plusieurs méthodes d'optimisation. Par ailleurs, l'utilisation d'un modèle simplifié a permis une étude théorique, décrivant assez bien les résultats numériques obtenus avec le modèle choisi.

Les meilleurs résultats, du point de vue du critère, ont été obtenus en résolvant le problème de contrôle optimal avec un critère de type quadratique. Néanmoins, la commande optimale donnant ces résultats ne peut pas être appliquée concrètement. La commande la plus intéressante à appliquer pourrait être de type “bang-bang”, c'est-à-dire soit on rajoute de nouveaux individus, toujours la même quantité, soit on n'en rajoute pas, pourvu que la période pendant laquelle on en rajoute ne soit pas trop grande.
D'un point de vue pratique, la meilleure méthode serait de déterminer théoriquement la quantité initiale optimale à introduire dans le système, et de la distribuer en deux temps, ce qui améliore le coût. Les résultats obtenus avec cette méthode sont satisfaisants par rapport à ceux des autres méthodes.

L'étude a plutôt porté sur une comparaison par rapport au coût des coccinelles (α). Mais le nombre de pucerons présents au début de l'expérience (x_0) doit également influencer sur la quantité de coccinelles à introduire et la façon de le faire. Il serait donc intéressant de se pencher sur cet aspect du problème.

D'autre part, il est difficile de quantifier ce paramètre α, qui doit faire le rapport entre le coût d'introduction des coccinelles et le coût des dégâts causés par les pucerons. Comme il est difficile d'interpreter la signification de la variable de contrôle u, qui serait une vitesse d'introduction de coccinelles, et de savoir comment l'appliquer dans les expériences.

L'étude sur le modèle proie-prédateur choisi permet de voir quelle méthode peut être la plus intéressante d'un point de vue pratique. Toutefois, ce modèle ne décrit pas parfaitement l'évolution des populations de pucerons et de coccinelles. Un modèle structuré en classes, afin de reproduire l'évolution en stade des coccinelles, s'adapte mieux aux données (Duffau, 2000). Ainsi il serait également intéressant d'essayer des techniques d'optimisation sur ce modèle.

Bibliographie

[1] **D'Andréa-Novel B. & Cohen De Lara M.** *Commande linéaire des systèmes dynamiques.* Masson. (1993)

[2] **Bryson A.E. & Ho Y.C.** *Applied optimal control.* Hemisphere. (1975)

[3] **Duffau M.** *Modélisation d'un système proie-prédateur dans le cadre de la lutte biologique.* Rapport de stage Insa Toulouse. (2000)

[4] **Faurre P. & Robin M.** *Eléments d'automatique.* Dunod. (1986)

[5] **Getz W.M.** On harvesting two competing populations. *Journal of optimization theory and applications* 28, p. 585-602. (1979)

[6] **Goh B.S., Leitmann G. & Vincent T.L.** Optimal control of a prey-predator system. *Mathematical Biosciences* 19, p. 263-286. (1974)

[7] **Hocking L.M.** *Optimal Control.* Clarendon Press. (1997)

[8] **Luenberger D.G.** *Introduction to dynamic systems.* J. Wiley and sons. (1979)

[9] **Silvert W. & Smith W.R.** Optimal Exploitation of a multi-species community. *Mathematical Biosciences* 33, p. 121-134. (1977)

[10] **Teo K.L., Goh C.J. & Wong K.H.** *A unified computationnal approach to optimal control problems.* (1991)

[11] **Vincent T.L.** Pest management programs via optimal control theory. *Biometrics* 31, p. 1-10. (1975)

Annexes

Annexe A

Listings des principaux programmes

Les calculs présentés dans ce rapport ont tous été faits sous Matlab.

A.1 Recherche de la condition initiale optimale

```
function res=jy0(y0)
% Le systeme est :
%  dx/dt = a.x.(1-x) - b.ymax.(x/(x+c/K)).y/K
%  dy/dt = e.(x/(x+c/K)).y - d.y
% Le resultat est J(y0)= integrale sur [0,T] de x(t)dt + alpha.y0
% Utilisation de la fonction exple1bis pour resoudre le systeme

global a;
global b;
global c;
global d;
global e;
global K;
global ymax;
global T;
global X;
global X0;
global alpha;

X01=X0;
X01(2)=y0;
% Resolution :
[t,X]=ode45('exple1bis',T,X01);
% x = X(:,1) et y = X(:,2)
criter1=trapz(T,X(:,1));
res=criter1 + alpha*y0;
```

A.2 Répartition de la condition initiale optimale

```
function res=jn1t1(n1t1)
% Le systeme est :
%  dx/dt = a.x.(1-x) - b.ymax.(x/(x+c/K)).y/K
%  dy/dt = e.(x/(x+c/K)).y - d.y
% On connait x0 et on dispose de y0 a repartir en y0-n1 a t0, et n1 a t1
% Le resultat est J(n1,t1)= integrale sur [0,T] de x(t)dt + alpha.y0
% Utilisation de la fonction exple1bis pour resoudre le systeme

global a;
global b;
global c;
global d;
global e;
global K;
global ymax;
global T;
global X;
global X0;
global alpha;

n1=n1t1(1);
t1=fix(n1t1(2));

y0=X0(2);
X01=X0;
if (n1==0.)
  % Resolution
  [t,X]=ode45('exple1bis',T,X01);
else
  X01(2)=y0-n1;
  % Resolution :
  if (t1>T(1))
    [t,X0t1]=ode45('exple1bis',T(T<=t1),X01);
    X02=X0t1(end,:);
  else
    X0t1=[];
    X02=X01;
  end;
  if (t1<T(end))
    X02(2)=X02(2)+n1;
    [t,Xt1T]=ode45('exple1bis',T(T>=t1),X02);
  else
    Xt1T=[];
  end;
```

```
  X=[X0t1(1:end-1,:);Xt1T];
end;
% x = X(:,1) et y = X(:,2)
criter1=trapz(T,X(:,1));
res=criter1 + alpha*y0;
```

A.3 Résolution du problème linéarisé

```
clear all;
% Le systeme est :
%  dx/dt = a.x.(1-x) - b.ymax.(x/(x+c/K)).y/K
%  dy/dt = e.(x/(x+c/K)).y - d.y + u
% On linearise autour du point d'equilibre X3, le pb devient dZ/dt=J3.Z + B.u,
% et on calcule commande de facon a minimiser critere quadratique
% integrale(Z(1)^2 + alpha.u^2) avec utilisation ricatti differentielle
%        -> optimisation en temps fini

% Initialisation des parametres :
global a;
global b;
global c;
global d;
global e;
global K;
global ymax;
global J3;
global S;
global T;
global M;
global Q;

a=0.4;b=2.;c=100000.;d=0.1;e=1.;K=40000.;ymax=K*3/4;
alpha=3.5;

% Calcul du pt d'equilibre (3), les autres pts etant (0,0) et (1,0) :
X3=[c*d/(K*(e-d)) 0];
X3(2)=a*c*e*(1-X3(1))/(b*ymax*(e-d));
X3
% stabilite :
stab= (0.5*(K-c) < c*d/(e-d) );

if(stab)
  % Duree et conditions initiales
  T=[0:1:100];
  X0=[1500./K 0.];
```

```
Z0=X0-X3

% Jacobienne au point X3
J3=zeros(2,2);
J3(1,1)=a*d*(1-c*(e+d)/(K*(e-d)))/e;
J3(1,2)=-b*d*ymax/(e*K);
J3(2,1)=a*(K*(e-d)-c*d)/(b*ymax);

% Matrices pour l'equation de riccati :
Q=zeros(2,2); Q(1,1)=1;
R=alpha*alpha;
B=zeros(2,1); B(2,1)=1;
M=B*(1/R)*B';

% Resolution systeme linearise sans commande:
S=zeros(2,2);
[t,Zavt]=ode45('slin',T,Z0);
% z1 = Z(:,1) et z2 = Z(:,2)

% Resolution de l'equation de riccati differentielle
% dS/dt + J3'.S + S.J3 - S.B.(1/R).B'.S + Q = 0 avec S(end)=0
T2=[T(end):T(1)-T(2):T(1)];
ST=zeros(3,1);
[t,coeffS]=ode45('riccadiff',T2,ST);

S=zeros(2,2,max(size(T)));
S(1,1,:)=coeffS(end:-1:1,1);
S(1,2,:)=coeffS(end:-1:1,2);
S(2,1,:)=S(1,2,:);
S(2,2,:)=coeffS(end:-1:1,3);
clear('coeffS')

% Resolution systeme linearise avec controle:
[t,Z]=ode45('slin2',T,Z0);
% z1 = Z(:,1) et z2 = Z(:,2)

u=zeros(size(T));
for i=T
  u(T==i)=-(1/R)*B'*S(:,:,T==i)*Z(T==i,:)';
end;

echo on
minimum=Z0*S(:,:,1)*Z0'
%Critere du systeme linearise sans controle
c1=trapz(T,Zavt(:,1).^2)
c2=trapz(T,(Zavt(:,1)+X3(1)).^2)
```

```
  %Critere du systeme linearise + controle
  c1b=trapz(T,Z(:,1).^2)
  c2b=trapz(T,(Z(:,1)+X3(1)).^2)
  c1c=trapz(T,Z(:,1).^2 + R.*u(:).^2)
  c2c=trapz(T,(Z(:,1)+X3(1)).^2 + R.*u(:).^2)
  echo off;
else
  disp 'pt d''equilibre X3 pas stable'
end;
```

A.4 Résolution du problème non linéaire

A.4.1 Critère linéaire

Méthode avec matrice de transition

```
% Le systeme est
%     dx/dt  = a.x.(1-x) - b.(x/(x+c/K)).ymax.y/K
%     dy/dt  = e.(x/(x+c/K)).y - d.y + u
% Le critere a minimiser est integrale de 0 a T de (L = x + alpha.u)
% Utilisation d'un algo de la matrice de transition
%     ref : A Transition-Matrix Algorithm - p.216 - Applied Optimal Control

% Variables globales :
global a;
global b;
global c;
global d;
global e;
global K;
global ymax;
global umax;
global alpha;
global T;
global u;

% Initialisation des parametres :
a=0.4;b=2.;c=100000.;d=0.1;e=1.0;K=40000.;
ymax=K*3/4;
umax=0.09;
alpha=3.5;
eps=0.9; %rand
tol=5e-5;
dl1=0.1; %*rand
dl2=0.1; %*rand
```

```
stock=[];compteur=0.;qwe=[];

% Duree et conditions initiales
T=[0:1:100];
XL0=zeros(4,1);
XL0(1:2)=[1500./K 0]; % conditions initiales pour x et y

% Etape 1 : choix de lambda1(0) et lambda2(0)
XL0(3:4)=[20.1 -9.1];

tps=cputime;

% Etape 2 : resolution du systeme complet (x,y,lambda1,lambda2)
[t,XL]=ode45('scomplet',T,XL0);

% Etape 3 : stockage des valeurs lambda1(T) et lambda2(T) dans muT
muT=XL(end,3:4)';
stock=[stock, muT];

while(~all(abs(muT)<=tol))
  % Etape 4 : determination de la matrice de transition
  %           ici par differentiation numerique directe
  trans=zeros(2,2);

  XL0i=XL0;
  XL0i(3)=XL0i(3)+dl1;
  [t,XL]=ode45('scomplet',T,XL0i);
  muTi=XL(end,3:4)';
  dmuT=muTi-muT;
  trans(:,1)=dmuT/dl1;

  XL0i=XL0;
  XL0i(4)=XL0i(4)+dl2;
  [t,XL]=ode45('scomplet',T,XL0i);
  muTi=XL(end,3:4)';
  dmuT=muTi-muT;
  trans(:,2)=dmuT/dl2;

  % Etape 5 : choix de delta_mu(T)=-eps.(mu(T)-muT_voulu), ici muT_voulu=0
  dmuT= - eps*muT;

  % Etape 6 : calcul de la variation a appliquer a lambda1(0) et lambda2(0)
  dl0=inv(trans)*dmuT;

  % Etape 7 : changement des valeurs initiales lambda1(0) et lambda2(0)
```

```
  XL0(3:4)=XL0(3:4)+dl0(:);

  % Etape 2 : resolution du systeme complet (x,y,lambda1,lambda2)
  [t,XL]=ode45('scomplet',T,XL0);

  % Etape 3 : stockage des valeurs lambda1(T) et lambda2(T) dans muT
  muT=XL(end,3:4)';

  ver=[abs(stock(1,:)-muT(1));abs(stock(2,:)-muT(2))]<=1e-7;
  [q,w1]=max(ver(1,end:-1:1));
  [q,w2]=max(ver(2,end:-1:1));
  if(any(ver(1,:)) & any(ver(2,:)) & w1==w2)
    compteur=compteur+1;
    qwe=[qwe,muT];
    if(compteur==5)
      disp 'boucle',
      break;
    end;
  end;
  stock=[stock, muT];

end;

tps=cputime-tps

Hu=XL(:,4)+alpha;
u=max(0,-sign(Hu(:))*umax);
[t,Xu]=ode45('exple1abis',T,XL0(1:2));
criter1=trapz(T,Xu(:,1) + alpha*u(:))
figure(1),plot(T,Xu(:,1),T,Xu(:,2)),legend('x','y')
figure(2),plot(T,XL(:,3),T,XL(:,4)),legend('lambda1','lambda2')
figure(3),plot(T,u),legend('u')
```

Méthode empirique

```
clear all;
% Le systeme est :
%  x' = a.x.(1-x) - b.ymax.(x/(x+c/K)).y/K
%  y' = e.(x/(x+c/K)).y - d.y + u
% Resolution et graphes (en fonction du temps et portrait de phase)
% resolution du systeme puis utilisation pour resoudre etat adjoint :
%  l1' = -1 - l1.[a.(1-2.x) - (b.c/(K.x+c)^2).ymax.y] - l2.(e.c/K(x+c/K)^2).y
%  l2' = l1.b.ymax.(x/(K.x+c)) - l2.[e.(x/(x+c/K)) - d]
% puis calcul commande correspondante...
```

```
% Initialisation des parametres :
global a;
global b;
global c;
global d;
global e;
global K;
global ymax;

a=0.4;b=2.;c=100000.;d=0.1;e=1.0;K=40000.;
ymax=K*3/4;
alpha=3.5;
umax=0.5;

% Duree et conditions initiales
global T;
T=[0:1:100];
X0=[1500./K 0];

% Calcul du pt d'equilibre (3) :
X3=[c*d/(K*(e-d)) 0];
X3(2)=a*c*e*(1-X3(1))/(b*ymax*(e-d));
X3;
% stabilite :
stab= (0.5*(K-c) < c*d/(e-d) )

global X1;
global u;
u=zeros(size(T));
u2=ones(size(T));
criter2=100;
T2=[100:-1:0];
LT=[0. 0.];
% Resolution :
[t,X1]=ode45('exple1abis',T,X0);
% x = X(:,1) et y = X(:,2)
criter1=trapz(T,X1(:,1) + alpha*u(:))
temp=criter1;
i=0;
while(any(u(:)~=u2(:)))
  i=i+1
  criter2=criter1;
  %X2=X1;
  u3=u2;
  u2=u;
  %Puis etat adjoint :
```

```
  [t,L]=ode45('lambda1bis',T2,LT);
  %Commande :
  u=max(0,-sign(L(end:-1:1,2)+alpha)*umax);
  if(all(u(:)==u3(:))) disp 'alterne entre deux val', break; end;
  % Resolution :
  X2=X1;
  [t,X1]=ode45('exple1abis',T,X0);
  % x = X(:,1) et y = X(:,2)
  criter1=trapz(T,X1(:,1) + alpha*u(:))

end;
```

A.4.2 Critère quadratique

```
% Le systeme est
%     dx/dt  = a.x.(1-x) - b.(x/(x+c/K)).ymax.y/K
%     dy/dt  = e.(x/(x+c/K)).y - d.y + u
% Le critere a minimiser est integrale de 0 a T de (L = x^2 + alpha.u^2)
% Utilisation d'une methode du gradient
%     ref : A first-order gradient Algorithm - p.222 - Applied Optimal Control

% Variables globales :
global a;
global b;
global c;
global d;
global e;
global K;
global ymax;
global alpha;
global u;
global T;
global X1;

% Initialisation des parametres :
a=0.4;b=2.;c=100000.;d=0.1;e=1.0;K=40000.;
ymax=K*3/4;
alpha=3.5;
eps=0.9; %rand
tol=5e-5;
umax=0.07;
if alpha<=0.5/eps
  W=2*(eps+0.5);
else
```

```
  W=eps*2*alpha;
end;
pw=1;pwavt=1;disp([W,pw])
% Duree et conditions initiales
T=[0:1:100];
T2=[T(end):-1:T(1)];
u=zeros(size(T));
deltaU=zeros(size(T));
l2plus2au=zeros(size(T));
LT=zeros(2,1); % conditions finales pour lambda1 et lambda2
Ijj=1;
X0=zeros(2,1);
X0(1)=1500./K; % conditions initiales pour x (et y)

% Etape 1 : choix de u(t)
u(:)=max(0,umax*cos(T*5*pi/100).*exp(-T/20));

while(~(abs(Ijj)<=tol))
  % Etape 5 : calcul de la variation a appliquer a u
  deltaU=-(l2plus2au(:))./(W^pw);

  % Etape 6 : changement des valeurs de u
  uavt=u;pwavt=pw;
  u=u(:)+deltaU(:);
  while(any(u<0))
    deltaU=deltaU/W;pw=pw+1;
    u=uavt(:)+deltaU(:);
  end;
  if (pwavt~=pw) disp([W,pw]), end;
  figure(3),plot(T,uavt,T,u)
  legend('uavt','u')

  % Etape 2 : resolution du systeme dX1/dt=f(X1,u) (x,y)
  [t,X1]=ode45('exple1abis',T,X0);
  if ~(max(size(t))==max(size(T))) disp 'pb integration etape 2', end;
  criter1=trapz(t,X1(:,1).^2 + alpha*u(:).^2);

  % Etape 3 : determination des variables adjointes :
  [t,L]=ode45('lambda2bis',T2,LT);
  if ~(max(size(t))==max(size(T))) disp 'pb integration etape 3', end;

  % Etape 4 : calcul de Ijj = integrale sur [0,T] de (1/W).(l2+2.alpha.u)^2
  l2plus2au=L(end:-1:1,2)+2*alpha*u(:);
  Ijj=trapz(T,l2plus2au(:).^2);
  Ijj=Ijj/(W^pw)
```

```
end;
disp([W,pw,W^pw])
criter1=trapz(T,X1(:,1).^2 + alpha*u(:).^2)
figure(1),plot(T,X1(:,1),T,X1(:,2)),legend('x','y')
figure(2),plot(T2,L(:,1),T2,L(:,2)),legend('lambda1','lambda2')
figure(3),plot(T,u),legend('u')
```