

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

\*\*\*



Université Saâd Dahlab - Blida 1

Faculté des Science

Département de Mathématiques



\*\*\*

# MEMOIRE DE MASTER

En MATHEMATIQUES

Spécialité : Recherche Opérationnelle

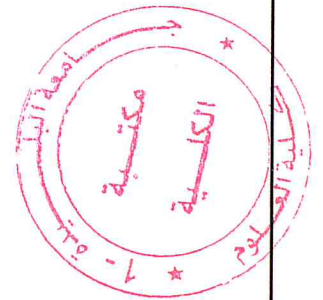
## Thème

*Contrôle optimal de modèle de Lotka-Volterra  
par la méthode combinée Adomian/Alienor*

Réalisé par :

*Azari Allaeddine*

*Kechid Hadj*



MA-510-45-2

Devant le jury composé de :

M. Blidia	Professeur, U. de Blida 1	Président
S. Manseur	Maître de conférences, U. de Blida 1	Examineur
N.A. MESSAOUDI	Chargé de cours, U. de Blida 1	Encadreur

Blida, juillet 2017

## RÉSUMÉ

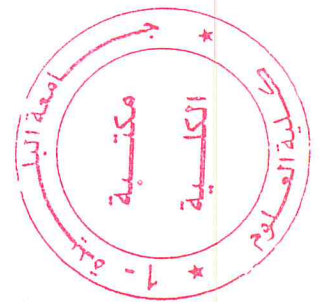
Le but de ce travail est d'utiliser deux méthodes mathématiques : la méthode de décomposition d'Adomian et celle d'Alienor pour la résolution d'un problème de contrôle optimal régi par des systèmes d'équations différentielles non linéaires du premier ordre.

La première méthode permet la résolution analytique d'un système différentiel non linéaire sous forme de série convergente, et dépendant explicitement des paramètres du système, de contrôle et ne nécessite aucune discrétisation ni d'espace ni de temps contrairement aux méthodes numériques.

La deuxième méthode consiste à exprimer les paramètres inconnus par des transformations réductrices, densifiant l'espace des paramètres par une courbe  $\alpha$  - dense et permet de remplacer le problème de minimisation d'une fonction à plusieurs variables par celui d'une fonction à une seule variable.

La combinaison des deux méthodes permet de ramener le problème du contrôle optimal des systèmes dynamiques à des problèmes de minimisation d'une fonction à une seule variable.

Une application du problème de contrôle optimal au modèle dynamique de Lotka-Volterra a été réalisée. Les résultats numériques obtenus par la méthode combinée Adomian/Alienor sont satisfaisants et prouvent l'efficacité de cette approche.



## ABSTRACT

The aim of this work is using two mathematical methods : the Adomian decomposition method and the Alienor method for solving an optimal control problem governed by the nonlinear differentials equations systems.

Firstly, the Adomian method allows to solve a nonlinear differential system. The solution is an analytical function of time and is given in a series form and explicitly dependent on the parameters of the system, control and requires no discretization in space and in time, contrary to numerical methods.

The Alienor method consists of expresse the unknown parameters and control by reduction transformation, in densifying the space of the parameters by  $\alpha - dense$  curve and allows to reduce the problem of minimization of a function with several variables to a function with one variable.

The combination of the two methods allows to approximate the optimal control of a dynamic system to the problem of minimization of a function with single variable.

An application to solve the optimal control problem of Lotka-Volterra model is performed. The results obtained by the combined Adomian/Alienor method are satisfies and prouve the efficiency of this approach.

## ملخص

الهدف من هذا العمل هو استعمال طريقتين: طريقة " ادوميون " وطريقة " الينور " من اجل حل مشكل التحكم الامثل لجمل المعادلات التفاضلية اللاخطية من الدرجة الأولى.

الطريقة الأولى تسمح بالحل التحليلي للجمل التفاضلية الغير الخطية على شكل سلاسل متقاربة و ترتبط بكل وضوح بمعاملات الجملة و دالة التحكم، وهي عكس الطرق العددية فهي لا تتطلب تقطيع المساحات و لا الزمن.

الطريقة الثانية تتمثل في عرض المعاملات الغير معروفة بواسطة التحويلات المختصرة بتكثيف مجال المعاملات مما يسمح باستبدال إشكالية تحسين دالة ذات عدة متغيرات بإشكالية تحسين دالة ذات متغير واحد.

مزاوجة هاتين الطريقتين تسمح بالمرور من إشكالية التحكم الامثل إلى إشكالية خفض دالة ذات متغير واحد.

اختبرنا نموذج تطبيقي لإشكالية التحكم الامثل للجملة الديناميكية لوتكا-فولتيرا. النتائج العددية التي تحصلنا عليها بفضل طريقة المزاوجة بين "ادوميون و الينور" تؤكد و تثبت فعالية الطريقة المتبعة.

## REMERCIEMENTS

Avant tout, nous remercions ALLAH tout puissant de nous avoir donné la foi et la force pour achever ce modeste travail.

Nous tenons à exprimer notre reconnaissance et toute notre gratitude à Madame Messaoudi Nadia Amel chargée de cours à l'université de Blida, nous avoir proposé ce mémoire et dirigé nos travaux, pour ces conseils précieux, ces orientations et surtout sa simplicité et sa patience et ainsi sa disponibilité tout au long de notre travail.

Nous remercions Monsieur BLIDIA Mustapha, professeur à l'université de Blida, d'avoir accepté d'être président du jury.

Nous adressons nos remerciements à Monsieur MANSEUR Salah, Maître de conférences à l'université de Blida qui nous a fait l'honneur d'accepter de juger nos travail.

Nous remercions chaleureusement :  
tous nos familles, surtout nos parents.  
tous(tes) les enseignants(es) qui ont contribué à notre formation.  
tous nos amis(es) en souvenir les plus agréables pour tous les bons moments partagés.

# Table des matières

Résumé	1
Abstract	2
	3
Remerciements	4
Table des Matières	5
Liste des illustrations, GRAPHIQUES et TABLEAUX	7
Introduction Générale	8
<b>Chapitre 1 La méthode de décomposition d'Adomian</b>	<b>11</b>
1.1 Introduction . . . . .	12
1.2 Présentation de la méthode décompositionnelle . . . . .	12
1.2.1 Convergence de la MDA . . . . .	14
1.3 Principe de la méthode décompositionnelle pour des équations différentielles . . . . .	16
1.3.1 Formule de calcul des polynômes d'Adomian pour une fonction à une seule variable . . . . .	17
1.4 Principe de la méthode décompositionnelle d'Adomian pour les systèmes différentielles non linéaires . . . . .	19
1.4.1 Formule de calcul des polynômes d'Adomian pour une fonction à deux variables . . . . .	20
1.5 Conclusion . . . . .	21
<b>Chapitre 2 La méthode d'Alienor</b>	<b>22</b>
2.1 Introduction . . . . .	23
2.2 La méthode Alienor . . . . .	23
2.2.1 Transformation d'Archimède . . . . .	23
2.2.2 Transformation de Cherruault . . . . .	25
2.2.3 Transformation de Mora . . . . .	27
2.3 Application à l'optimisation globale . . . . .	28
2.3.1 Méthode de recherche du minimum global pour une fonction univariante . . . . .	30
2.4 Conclusion . . . . .	33
<b>Chapitre 3 Application de la MDA au modèle de Lotka-Volterra</b>	<b>34</b>

3.1	Introduction . . . . .	35
3.2	Modèle Lotka-Volterra . . . . .	35
3.2.1	Description du modèle Lotka-Volterra . . . . .	35
3.2.2	Signification physique des équations . . . . .	36
3.3	Résolution du modèle Lotka-Volterra par la MDA . . . . .	36
3.3.1	Application numérique . . . . .	39
3.3.2	Le recollement des solutions par la MDA . . . . .	40
3.4	Conclusion . . . . .	41
<b>Chapitre 4 Contrôle optimal du modèle de Lotka-Volterra</b>		<b>42</b>
4.1	Introduction . . . . .	43
4.2	Formulation du problème de contrôle . . . . .	43
4.3	Principe du minimum de Pontryagin . . . . .	44
4.4	Utilisation de la méthode combinée Adomian/Alienor . . . . .	44
4.5	Application au modèle de Lotka-Volterra . . . . .	46
4.6	Discussion . . . . .	48
<b>Conclusion générale</b>		<b>49</b>
<b>Références</b>		<b>50</b>
<b>Annexe 1 : Programation de la MDA en Maple</b>		<b>52</b>
<b>Annexe 2 : Méthodes de Runge-Kutta</b>		<b>53</b>

# Table des figures

Figure 1.1. Courbe de la solution $\phi_4$ . . . . .	16
Figure 1.2. Superposition des courbes des solutions $x_e$ (points) et $\phi_6$ (ligne) . . . . .	19
Figure 2.1. Spirale d'Archimède . . . . .	24
Figure 2.2. Structure d'arbre de la transformation réductrice . . . . .	25
Figure 2.3. Courbe densifiant le pavé $[-1, 1]^2$ . . . . .	26
Figure 2.4. Courbe densifiant le pavé $[0, 40] \times [0, 1]$ . . . . .	28
Figure 2.5. Courbe densifiant le pavé $[0, 1]^2$ . . . . .	28
Figure 2.6. Courbe de $f_1(\theta)$ . . . . .	31
Figure 2.7. Courbe de $f_2(\theta)$ . . . . .	32
Figure 3.1. Superposition des courbes des solutions de $x$ . . . . .	39
Figure 3.2. Superposition des courbes des solutions de $y$ . . . . .	39
Figure 3.3. Courbe de la solution $x$ . . . . .	41
Figure 3.4. Courbe de la solution $y$ . . . . .	41
Figure 3.5. Courbe de la solution $x$ et $y$ . . . . .	41
Figure 4.1. Courbes de contrôle $u^e(+)$ , $u^1$ (points) et $u^2$ (ligne) . . . . .	47
Figure 4.2. Superposition des courbes des solutions $x$ sans contrôle et avec contrôle optimal . . . . .	48
Figure 4.2. Superposition des courbes des solutions $y$ sans contrôle et avec contrôle optimal . . . . .	48

# Liste des tableaux

TABLEAU 4.1. Valeurs des paramètres obtenus par : Adomian/Alienor . . . . .	47
---	----



## Introduction générale

Il existe plusieurs modèles en physique, mécanique, écologie, biologique,... qui sont présentés sous forme d'équations fonctionnelles non linéaires (différentielles, aux dérivées partielles, intégrales, intégro-différentielles, algébriques, etc...).

Souvent il est difficile de résoudre de manière exacte des équation différentielles, pour cela, on fait appel aux méthodes numériques, comme : les méthodes des différences finies, des éléments finis, ou bien la méthode spectrales qui donnent des approximations de la solution en des points discrets, de plus ces méthodes "discrétisent" l'espace et le temps et linéarisent souvent les équations. Tous ces inconvénients n'existent pas dans la méthode décompositionnelle d'Adomian (MDA).

La MDA a été inventée en 1981 par le professeur **George Adomian** aux états unis d'Amérique de manière empirique et sans fondements théoriques ([1], [2]), cette méthode évite la linéarisation de l'équation et la discrétisation de l'espace et du temps. Elle donne les solutions sous forme de séries, de plus cette méthode est basée sur la décomposition de l'opérateur non linéaire en faisant intervenir des polynômes appelés polynômes d'Adomian dont les éléments sont calculés récursivement.

**Karim Abbaoui** a donné dans sa thèse présentée en 1995 [6] un ensemble original et remarquable de résultats théoriques et pratiques sur la méthode d'Adomian. Qui consiste à considérer la méthode décompositionnelle comme une méthode générale pour la résolution des systèmes dynamiques non linéaires. Dans les travaux de Abbaoui et al. ([4], [5]) une nouvelle preuve de la convergence a été proposé.

La méthode d'Adomian (MDA) est utilisée pour la résolution des équations intervenant dans le modèle de **Lotka-Volterra**, du modèle compartimentale de la dégranulation des basophiles, en théorie des microlasers, en mécanique des fluides ([7], [8]), ainsi que l'identification des paramètres et le contrôle optimal des systèmes dynamiques [8], [13], [21].

L'idée de base de la **méthode Alienor (MA)** consiste à exprimer  $n$  variables à l'aide d'une seule, autrement dit à remplir l'espace  $\mathbb{R}^n$  à l'aide d'une courbe. Ici dans le cadre de cette méthode, le problème de minimisation d'une fonction à plusieurs variables continues est ramené, grâce à une transformation réductrice, à un problème de minimisation d'une fonction à une seule variable. Cependant, le bon choix des transformations dépend non seulement de l'expérience d'utilisateur mais aussi du problème à résoudre. L'application de cette méthode aux problèmes concrets prouve son efficacité [8].

Dans notre travail, nous nous intéressons à la résolution d'un problème de contrôle optimale du modèle Lotka-Volterra exprimé sous la forme d'un système d'équations différentielles en général non linéaires comportant des paramètres connus et constants. On pourra faire intervenir les variables de contrôle en vue d'optimiser un certain critère ( maximiser la production, minimiser la quantité de populations, ...) ce qu'on appelle problème de contrôle optimal.

Le problème de contrôle optimal de systèmes de Lotka-Volterra peut être résolu par des méthodes classiques comme le principe du Maximum de Pontryaguine ([8], [25]). Ces méthodes sont difficiles à appliquer sur des exemples non linéaires.

Notre motivation est de chercher le contrôle optimal du système en boucle ouverte qui est une fonction dépendant explicitement des paramètres inconnus et de la variable de temps. Ceci peut être réalisé en minimisant une certaine fonction objective qui dépend explicitement des paramètres de contrôle dont il faudra exprimer les solutions du système en fonction explicite de ces paramètres.

L'objectif de ce mémoire est l'utilisation de la méthode décompositionnelle d'Adomian ([3], [6], [7], [8], [15]) et la méthode Alienor ([16], [8], [7]) pour la résolution du problème de contrôle optimal des systèmes dynamiques. Après avoir exprimé préalablement la fonction de contrôle sous forme de série de fonctions (polynomiales, spline, exponentielles,...) dans le système différentiel, la méthode décompositionnelle d'Adomian est utilisée pour la résolution. La solution donnée dépend explicitement des paramètres de contrôle. Cette solution reportée dans la fonction objective nous conduit à une fonction dépendant de ces paramètres. Il en résulte que le problème de contrôle optimal des systèmes dynamiques devient un problème d'optimisation classique.

La combinaison des deux méthodes Adomian/Alienor permet de ramener le problème du contrôle optimal des systèmes dynamiques à des problèmes de minimisation d'une fonction à une seule variable.

Une application de la méthode combinée Adomian/Alienor pour la résolution du problème de contrôle optimal du modèle de Lotka-Volterra a été réalisée. Les résultats numériques obtenus par la méthode combinée sont satisfaisants et le critère a été minimale.

On a utilisé comme langage de programmation le "*Maple*" (version 6.0) qui est un logiciel de calcul formel.

Le contenu du mémoire est composé de quatre chapitres et d'une conclusion.

Le chapitre un est consacré à la description de la méthode décompositionnelle d'Adomian (MDA) pour résoudre un système différentiel non linéaire. Commençons par présenter la MDA dans le cas général et dans le cas des équations différentielles. Cette méthode donne aux systèmes d'équations différentielles les solutions sous la forme de séries convergentes et permet d'exprimer explicitement des paramètres qui interviennent dans ces équations.

Le deuxième chapitre, aborde la méthode Alienor (MA) pour l'approximation des fonctions à plusieurs variables par des fonctions à une seule variable. Nous rappelons l'idée de base de cette méthode et son utilisation aux problèmes d'optimisations globales de fonctions multivariées.

Dans le chapitre trois, nous décrivons le modèle dynamique de Lotka-Volterra pour deux populations (proies et prédateurs). On montre comment utiliser la méthode d'Adomian à la résolution de ce système. Ensuite, nous comparons les solutions obtenues par la MDA avec ceux données par la méthode de Runge-Kutta.

Le quatrième chapitre est consacré au problème de contrôle optimal régi par le système de Lotka-Volterra. On donne une méthodologie qui utilise la méthode combinée Adomian/Alienor permettant de ramener ce problème à un problème de minimisation d'une fonction à une seule variable. Ensuite, on donne une application numérique pour la résolution du problème.

Nous terminons notre travail par une conclusion et deux annexes.

# CHAPITRE 1

## LA MÉTHODE DE DÉCOMPOSITION D'ADOMIAN

## 1.1 Introduction

Dans ce chapitre, nous allons présenter la MDA dans sa forme initiale ainsi que les résultats de convergence. Nous appliquons la MDA à des équations différentielles et nous donnons quelques exemples illustratifs de la méthode.

## 1.2 Présentation de la méthode décompositionnelle

Nous allons d'abord décrire la méthode sous sa forme initiale présentée par **G.Adomian** ([1], [2]).

Soit l'équation non linéaire :

$$F(x(t)) = g(t) \quad (1.1)$$

où  $F$  représente un opérateur différentiel non linéaire.  $g$  une fonction analytique connue.

L'opérateur  $F$  possède des termes linéaires et non linéaires. Le terme linéaire est décomposé en  $L + R$  où  $L$  est un opérateur linéaire inversible et  $R$  est le reste de l'opérateur linéaire,  $N$  est un terme non linéaire. En pratique on choisit pour  $L$  l'opérateur d'ordre le plus élevé et qui soit facilement inversible.

L'équation (1.1) devient :

$$Lx + Rx + Nx = g \quad (1.2)$$

On applique l'opérateur  $L$  étant inversible, on note  $L^{-1}$  son inverse, sur l'équation (1.2), on obtient :

$$L^{-1}Lx = L^{-1}g - L^{-1}Rx - L^{-1}Nx \quad (1.3)$$

D'où, l'expression équivalente :

$$x = \gamma + L^{-1}g - L^{-1}Rx - L^{-1}Nx \quad (1.4)$$

avec  $\gamma$  fonction vérifiant  $L\gamma = 0$ . Si l'équation (1.4) correspond à un problème de valeur initiale l'opérateur  $L^{-1}$  est l'intégrale de 0 à  $t$ . Si  $L$  est un opérateur d'ordre deux, alors  $L^{-1}$  est l'opérateur d'intégration double défini par :

$$L^{-1}(\cdot) = \int_0^t \left( \int_0^s (\cdot) dv \right) ds$$

et  $L^{-1}Lx = x - x(0) - tx'(0)$ .

A partir des conditions initiales connues, l'équation (1.3) devient :

$$x = x(0) + tx'(0) + L^{-1}g - L^{-1}Rx - L^{-1}Nx \quad (1.5)$$

Pour un opérateur  $L$  d'ordre  $r$  où les conditions initiales sont fixées, le terme  $\gamma$  peut s'écrire sous forme générale :

$$\gamma = \sum_{i=0}^{r-1} \frac{w_i t^i}{(i)!}$$

où  $w_i = \left. \frac{d^i x}{dt^i} \right|_{t=0}$ .

La méthode décompositionnelle d'Adomian (MDA), consiste à chercher la solution sous la forme d'une série :

$$x = \sum_{n=0}^{\infty} x_n \quad (1.6)$$

et à décomposer  $N(x)$  en série :

$$N(x) = \sum_{n=0}^{\infty} A_n \quad (1.7)$$

où les  $A_n$  sont les polynômes d'Adomian qui dépendent uniquement de  $x_0, \dots, x_n$  et sont obtenus à partir de la relation suivante ([6], [4], [9]) :

$$N\left(\sum_{i=0}^{\infty} \lambda^i x_i\right) = \sum_{i=0}^{\infty} \lambda^i A_i \quad (1.8)$$

d'où

$$A_n = \frac{1}{(n)!} \frac{d^n}{d\lambda^n} \left[ N\left(\sum_{i=0}^n \lambda^i x_i\right) \right]_{\lambda=0}, \quad n = 0, 1, 2, \dots \quad (1.9)$$

L'équation (1.9) donne les polynômes d'Adomian :

$$\begin{cases} A_0 = N(x_0) \\ A_1 = x_1 \left( \frac{d}{dx_0} N(x_0) \right) \\ A_2 = x_2 \left( \frac{d}{dx_0} N(x_0) \right) + \frac{x_1^2}{(2)!} \left( \frac{d^2}{dx_0^2} N(x_0) \right) \\ A_3 = x_3 \left( \frac{d}{dx_0} N(x_0) \right) + x_1 x_2 \left( \frac{d^2}{dx_0^2} N(x_0) \right) + \frac{x_1^3}{(3)!} \left( \frac{d^3}{dx_0^3} N(x_0) \right) \\ \cdot \\ \cdot \\ \cdot \end{cases} \quad (1.10)$$

En remplaçant les expressions (1.6) et (1.7) dans (1.5), on obtient :

$$\sum_{n=0}^{\infty} x_n = x(0) + tx'(0) + L^{-1}g - L^{-1}R \sum_{n=0}^{\infty} x_n - L^{-1} \sum_{n=0}^{\infty} A_n \quad (1.11)$$

On pose  $\gamma = x(0) + tx'(0)$ . Le premier terme de la série  $x_0$  est défini par  $\gamma + L^{-1}g$ . Par conséquent, on peut écrire les relations récursives suivantes :

$$\begin{cases} x_0 = \gamma + L^{-1}g \\ x_1 = L^{-1}R x_0 - L^{-1}A_0 \\ x_2 = L^{-1}R x_1 - L^{-1}A_1 \\ x_3 = L^{-1}R x_2 - L^{-1}A_2 \\ \cdot \\ \cdot \\ \cdot \\ x_{n+1} = L^{-1}R x_n - L^{-1}A_n, \quad n \in \mathbb{N} \end{cases} \quad (1.12)$$

qui déterminent sans ambiguïté les termes  $x_n$  de la série solution puisque une fois les  $A_n$  calculés à l'aide de (1.9), les  $x_n$  sont déterminés puisque  $A_{n-1}$  ne dépend que

de  $x_0, x_1, \dots, x_{n-1}$  qui sont calculés aux étapes précédentes.

Une approximation de la solution sous la forme de série tronquée d'ordre  $p$  est donnée par l'expression suivante :

$$\phi_p = \sum_{n=0}^{p-1} x_n. \quad (1.13)$$

La formule suivante a été proposé par **G.Adomian** pour le calcul des polynômes d'Adomian :

$$A_n(x_0, x_1, \dots, x_n) = \sum_{j=0}^n c(n, j) f^{(j)}(x_0) \quad (1.14)$$

où  $c(n, j)$  représente la somme de tous les produits (divisés par  $m!$ ) des  $j$  termes  $x_i$ , dont la somme des indices  $i$  est égale à  $n$ , le nombre  $m$  étant le nombre de répétitions des mêmes termes dans le produit. Par exemple  $c(4, 1)$  correspond seulement à  $x_4$ ,  $c(4, 2)$  est  $x_1x_3 + \frac{x_2^2}{2!}$ ,  $c(4, 3)$  est  $\frac{x_1^2x_2}{2!}$  et  $c(4, 4) = \frac{x_1^4}{4!}$ .

Une comparaison entre la méthode de **Runge-Kutta** et la **MDA** a été présenté dans [10]. **K.Abbaoui** et **Y. Cherruault** [11] ont employé la méthode de décomposition pour résoudre le problème de **Cauchy**. Ils ont également fourni des preuves de convergence en employant une nouvelle formulation des polynômes d'Adomian et ils ont comparé la MDA à la méthode de Picard [6].

### 1.2.1 Convergence de la MDA

Les travaux de **G.Adomian**, jusqu'en 1987, ne contenaient que la définition de la méthode et quelques formules de calcul des polynômes, dérivée de la formule (1.9). Rien n'est justifié quant à la convergence de série  $\sum_{n=0}^{\infty} x_n$  ou, ce qui revient au même, à celle de  $\sum_{n=0}^{\infty} A_n$ .

Les premières preuves de convergence ont été précisées par **Yves Cherruault** et sont basées sur le théorème du point fixe. On peut utiliser les propriétés des séries substituées dans une autre série [8] (dont on suppose que l'opérateur  $N$  de l'Hilbert  $H$  dans lui même est une contraction  $\|N\| \leq \delta < 1$ ).

D'autres résultats de convergence sont donnés dans [6] et [9] qui ne font apparaître des hypothèses que sur  $N(x_0)$  et ses dérivées  $N^{(k)}(x_0)$  au sens de Fréchet.

Pour plus de détails sur la démonstration (voir ([4], [7], [8]) ).

L'exemple suivant illustre l'utilisation de la MDA.

**Exemple 1**

Considérons l'équation suivante [21] :

$$\begin{cases} \frac{d^2x}{dt^2} + \sin x = 0 \\ x(0) = \beta, \quad x'(0) = 0 \end{cases}$$

Posons :  $N(x) = \sin x$  et  $L = \frac{d^2}{dt^2}$ , d'où l'équation :

$$Lx + Nx = 0$$

On cherche la solution  $x$  sous forme suivante :

$$x = \sum_{n=0}^{\infty} x_n$$

En appliquant une double intégration entre 0 et  $t$  de l'équation ci-dessous on obtient :

$$x = \beta - L^{-1}N(x)$$

avec

$$N(x) = \sum_{n=0}^{\infty} A_n.$$

D'où

$$\sum_{n=0}^{\infty} x_n = \beta - L^{-1} \sum_{n=0}^{\infty} A_n.$$

Le calcul des polynômes d'Adomian par l'expression (1.9), nous donne :

$$\begin{cases} A_0 = \sin x_0 \\ A_1 = x_1(\cos x_0) \\ A_2 = x_2(\cos x_0) - \frac{x_1^2}{(2)!}(\sin x_0) \\ A_3 = x_3(\cos x_0) - x_1x_2(\sin x_0) - \frac{x_1^3}{(3)!}(\cos x_0) \\ \cdot \\ \cdot \\ \cdot \end{cases}$$

Le schéma récursif suivant permet de déterminer les  $x_n$  :

$$\begin{cases} x_0 = \beta \\ x_1 = -\frac{t^2}{(2)!} \sin \beta \\ x_2 = \frac{t^4}{(4)!} \sin \beta \cos \beta \\ x_3 = -\frac{t^6}{(6)!} (\sin \beta \cos^2 \beta - 3 \sin^3 \beta) \\ \cdot \\ \cdot \\ \cdot \end{cases}$$



et  $\phi_p = \sum_{n=0}^{p-1} x_n$  est l'approximation de la série solution, tel que  $\lim_{p \rightarrow \infty} \phi_p = x$ .

Une solution approchée de la série solution à l'ordre 4 est donnée par la formule suivante :

$$\phi_4 = \beta - \frac{t^2}{2} \sin \beta + \frac{t^4}{24} \cos \beta \sin \beta + \frac{t^6}{720} (3 \sin^3 \beta - \cos^2 \beta \sin \beta).$$

La figure (1.1) représente la courbe de la solution pour  $\beta = \frac{\pi}{2}$  :

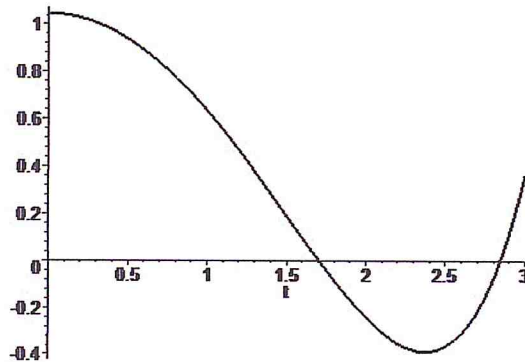


Figure 1.1. Courbe de la solution  $\phi_4$

### 1.3 Principe de la méthode décompositionnelle pour des équations différentielles

Considérons l'équation différentielle suivante :

$$\begin{cases} \frac{dx}{dt} = f(x) + g(t) \\ x(t=0) = x_0^* \end{cases} \quad (1.15)$$

où  $f$  et  $g$  sont deux fonctions connues.

La MDA ([4], [8]) consiste à transformer l'équation (1.15) sous forme canonique.

Pour cela, il suffit de faire une intégration par rapport à  $t$  du fait que  $L$  est l'opérateur différentiel du premier ordre. Posons :

$$L^{-1}(\cdot) = \int_0^t (\cdot) ds$$

On obtient :

$$x(t) = x_0^* + L^{-1}f(x) + L^{-1}g(t). \quad (1.16)$$

On cherche la solution  $x$  sous la forme d'une série :

$$x = \sum_{n=0}^{\infty} x_n \quad (1.17)$$

et on décompose le terme non linéaire  $f(x)$  en série :

$$f(x) = \sum_{n=0}^{\infty} A_n. \quad (1.18)$$

Les  $A_n$  étant les polynômes d'Adomian donnés par la formule suivante :

$$A_n = \frac{1}{(n)!} \frac{d^n}{d\lambda^n} f\left(\sum_{i=0}^n \lambda^i x_i\right) \Big|_{\lambda=0}, \quad n = 0, 1, 2, \dots \quad (1.19)$$

En remplaçant (1.17) et (1.18) dans (1.16) on obtient :

$$\sum_{n=0}^{\infty} x_n = x_0^* + L^{-1}g(t) + L^{-1} \sum_{n=0}^{\infty} A_n \quad (1.20)$$

qui conduit aux expressions déterminants les  $x_n$  :

$$\begin{cases} x_0 = x_0^* + L^{-1}g \\ x_1 = L^{-1}A_0(x_0) \\ x_2 = L^{-1}A_1(x_0, x_1) \\ x_3 = L^{-1}A_2(x_0, x_1, x_2) \\ \cdot \\ \cdot \\ \cdot \\ x_{n+1} = L^{-1}A_n(x_0, x_1, x_2, \dots, x_n), \quad n \in \mathbb{N}. \end{cases} \quad (1.21)$$

En pratique, nous utiliserons une série tronquée d'ordre "p" :

$$\phi_p = \sum_{n=0}^{p-1} x_n$$

### 1.3.1 Formule de calcul des polynômes d'Adomian pour une fonction à une seule variable

Les polynômes  $A_n$  sont donnés par les formules suivantes ([4], [6], [8]) :

$$\begin{cases} A_0(x_0) = f(x_0) \\ A_n(x_0, x_1, \dots, x_n) = \sum_{\alpha=n} f^{(\alpha_1)}(x_0) \frac{x_1^{\alpha_1-\alpha_2}}{(\alpha_1-\alpha_2)!} \cdots \frac{x_{n-1}^{\alpha_{n-1}-\alpha_n}}{(\alpha_{n-1}-\alpha_n)!} \frac{x_n^{\alpha_n}}{(\alpha_n)!} \text{ pour, } n = 1, 2, \dots \\ \text{où } \alpha = \alpha_1 + \dots + \alpha_n \end{cases} \quad (1.22)$$

Où  $(\alpha_i)_{i=1,2,\dots,n}$  est une suite décroissante.

La détermination des  $\alpha_i$  solutions de l'équation suivante :

$$\begin{cases} \alpha_1 + \alpha_2 + \dots + \alpha_n = n \\ \text{avec } \alpha_1 \geq \alpha_2 \geq \dots \geq \alpha_n. \end{cases}$$

Les expressions (1.22), nous permet de calculer facilement les polynômes d'Adomian pour n'importe quel ordre  $n$ . Par exemple, pour  $n = 4$ , les  $\alpha_i$  sont déterminés comme suit :

$\alpha_1$	$\alpha_2$	$\alpha_3$	$\alpha_4$
1	1	1	1
2	1	1	0
2	2	0	0
3	1	0	0
4	0	0	0

D'où, les polynômes d'Adomian pour  $n = 4$  :

$$\begin{cases} A_0 = f(x_0) \\ A_1 = x_1 f^{(1)}(x_0) \\ A_2 = x_2 f^{(1)}(x_0) + \frac{1}{2} x_1^2 f^{(2)}(x_0) \\ A_3 = x_3 f^{(1)}(x_0) + x_1 x_2 f^{(2)}(x_0) + \frac{1}{6} x_1^3 f^{(3)}(x_0) \\ A_4 = x_4 f^{(1)}(x_0) + (x_1 x_3 + \frac{1}{2} x_2^2) f^{(2)}(x_0) + \frac{1}{2} x_1^2 f^{(3)}(x_0) + \frac{1}{24} x_1^4 f^{(4)}(x_0) \end{cases}$$

### Exemple 2

Soit l'équation suivante :

$$\begin{cases} \frac{dx}{dt} = x^3 \\ x(0) = 1 \end{cases}$$

La solution exacte définit sur l'intervalle  $] -\infty, \frac{1}{2}[$  est donnée par l'expression suivante :

$$x_e(t) = \frac{1}{\sqrt{1-2t}}$$

$L$  : désigne l'opérateur différentiel donné par :

$$L = \frac{d}{dt}$$

Le terme non linéaire est :

$$f(x) = x^3$$

Le calcul des polynômes d'Adomian pour  $n = 4$  présentés dans la section (1.3.1), nous donne :

$$\begin{cases} A_0 = x_0^3 \\ A_1 = 3x_0 x_1 \\ A_2 = 3x_0 x_2 + 3x_0 x_1^2 \\ A_3 = 3x_0 x_3 + 6x_0 x_1 x_2 + x_1^3 \\ A_4 = 3x_0 x_4 + x_0(6x_1 x_3 + 3x_2^2) + 3x_1^2 \end{cases}$$

Le premier terme de la série solution est égale à :

$$x_0 = 1$$

Les cinq premiers termes de la série solution sont calculés ainsi :

$$\begin{cases} x_1 = \int_0^t A_0 ds = \int_0^t (1)^3 ds = t \\ x_2 = \int_0^t A_1 ds = \int_0^t 3s ds = \frac{3}{2}t^2 \\ x_3 = \int_0^t A_2 ds = \int_0^t (3 \cdot \frac{3}{2}s^2 + 3s^2) ds = \frac{5}{2}t^3 \\ x_4 = \int_0^t A_3 ds = \int_0^t (3 \cdot \frac{5}{2}s^3 + 6s \cdot \frac{3}{2}s^2 + s^3) ds = \frac{35}{8}t^4 \\ x_5 = \int_0^t A_4 ds = \int_0^t (3 \cdot \frac{35}{8}s^4 + 6s \cdot \frac{5}{2}s^3 + 3 \cdot (\frac{3}{2}s^2s^3)^2 + 3s^2) ds = \frac{63}{8}t^5 \end{cases}$$

Une solution approchée de la série solution à l'ordre 6 est égale à :

$$\phi_6 = 1 + t + \frac{3}{2}t^2 + \frac{5}{2}t^3 + \frac{35}{8}t^4 + \frac{63}{8}t^5.$$

La figure (1.2) représente la superposition des courbes des solutions exacte  $x_e(t)$  et approchée  $\phi_6(t)$  :

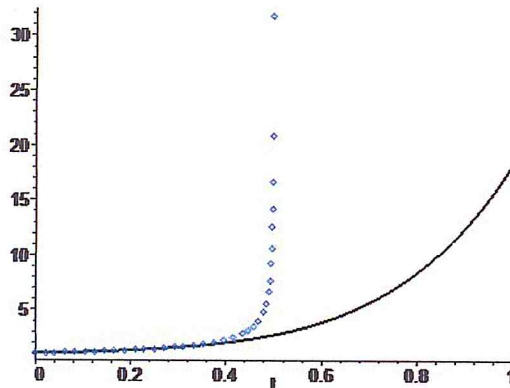


Figure 1.2. Superposition des courbes des solutions  $x_e$  (points) et  $\phi_6$  (ligne)

On remarque une superposition parfaite des solutions sur l'intervalle  $[0, 0.4]$ .

## 1.4 Principe de la méthode décompositionnelle d'Adomian pour les systèmes différentielles non linéaires

Considérons le système différentiel suivant :

$$\begin{cases} \frac{dx_i}{dt} = f_i(x_1, x_2, \dots, x_p) + g_i \\ x_i(t=0) = x_{i0}^* \end{cases} \quad i = 1, \dots, P \quad (1.23)$$

On cherche la solution  $(x_1, x_2, \dots, x_p)$  sous forme de séries :

$$x_i = \sum_{n=0}^{\infty} x_{in}, \quad i = 1, \dots, p \quad (1.24)$$

En décomposant le terme non linéaire  $f_i$  comme suit :

$$f_i(x_1, x_2, \dots, x_p) = \sum_{n=0}^{\infty} A_{in}, \quad i = 1, \dots, p \quad (1.25)$$

Les polynômes d'Adomian  $(A_{in})_{n \geq 0}$  [8] dépendent de  $x_{10}, x_{11}, \dots, x_{1n}, \dots, x_{p0}, x_{p1}, \dots, x_{pn}$ , et qui sont déterminés à partir de la formule suivante :

$$A_{in}(x_{10}, \dots, x_{1n}, x_{20}, \dots, x_{2n}, \dots, x_{p0}, \dots, x_{pn}) = \frac{1}{(n)!} \frac{d^n}{d\lambda^n} f_i(x_{01}, x_{02}, \dots, x_{0p}) |_{\lambda=0} \quad (1.26)$$

où les  $x_i$  sont pris sous la forme :

$$x_1 = \sum_{i=1}^n \lambda^i x_{i1}, \quad x_2 = \sum_{i=1}^n \lambda^i x_{i2}, \quad \dots, \quad x_p = \sum_{i=1}^n \lambda^i x_{ip}. \quad (1.27)$$

On intègre l'équation (1.23) entre 0 et  $t$ , et on remplace les expression (1.24) et (1.25) dans (1.23), on obtient :

$$\sum_{n=0}^{\infty} x_{in} = x_{i0}^* + L^{-1} g_i + L^{-1} \sum_{n=0}^{\infty} A_{in}, \quad i = 1, \dots, p \quad (1.28)$$

qui donnent les  $x_{in}$  des séries solutions :

$$\begin{cases} x_{i0} = x_{i0}^* + L^{-1} g_i \\ x_{i1} = L^{-1} A_{i0} \\ x_{i2} = L^{-1} A_{i1}, & i = 1, \dots, p \text{ et } n = 0, 1, \dots \\ \cdot \\ \cdot \\ x_{in+1} = L^{-1} A_{in}. \end{cases} \quad (1.29)$$

Les  $A_{in}$  sont donnés par les formules suivantes pour une fonction à  $p$  variables, indéfiniment différentiable (voir [6], [8], [12]) :

$$\begin{cases} A_{i0}(x_{10}, x_{20}, \dots, x_{p0}) = f_i(x_{10}, x_{20}, \dots, x_{p0}) \\ A_{in}(x_{10}, \dots, x_{1n}, \dots, x_{p1}, \dots, x_{pn}) = \sum_{\pi=n} \frac{x_{11}^{k_{11}}}{(k_{11})!} \cdots \frac{x_{1n}^{k_{1n}}}{(k_{1n})!} \cdots \frac{x_{p1}^{k_{p1}}}{(k_{p1})!} \cdots \frac{x_{pn}^{k_{pn}}}{(k_{pn})!} \\ \left( \frac{\partial^{k_{11} + \dots + k_{1n} + \dots + k_{p1} + \dots + k_{pn}}}{\partial x_1^{k_{11} + \dots + k_{1n}} \cdots \partial x_p^{k_{p1} + \dots + k_{pn}}} \right) f_i(x_{01} \dots x_{np}) \\ \text{où } \pi = k_{11} + \dots + n k_{1n} + \dots + k_{p1} + \dots + n k_{pn}. \end{cases} \quad (1.30)$$

On ne peut pas calculer tous les termes de la série  $\sum_{n=0}^{\infty} x_{in}$ , alors on se contentera d'une solution approchée sous forme d'une série tronquée d'ordre " $q$ " :

$$\phi_i^q = \sum_{n=0}^{q-1} x_{in}, \quad i = 1, \dots, p \quad (1.31)$$

### 1.4.1 Formule de calcul des polynômes d'Adomian pour une fonction à deux variables

Les polynômes  $A_n$  seront donnés par les expressions suivantes :

$$\begin{cases} A_0(x_{10}, x_{20}) = f(x_{10}, x_{20}) \\ A_n(x_{10}, \dots, x_{1n}, x_{20}, \dots, x_{2n}) = \sum_{\alpha+\beta=n} \frac{x_{11}^{\alpha_1 - \alpha_2}}{(\alpha_1 - \alpha_2)!} \cdots \frac{x_{1n}^{\alpha_n}}{(\alpha_n)!} \frac{x_{21}^{\beta_1 - \beta_2}}{(\beta_1 - \beta_2)!} \cdots \frac{x_{2n}^{\beta_n}}{(\beta_n)!} \\ \frac{\partial^{\alpha_1 + \beta_1}}{\partial x_1^{\alpha_1} \partial x_2^{\beta_1}} f(x_{10}, x_{20}) \\ \text{où } \alpha_1 + \dots + \alpha_n = \alpha, \quad \beta_1 + \dots + \beta_n = \beta \end{cases} \quad (1.32)$$

où les  $(\alpha_i)$  et les  $(\beta_i)$  sont deux suites décroissantes.

Ces formules sont aisément généralisables à  $p$  variables.

Afin de calculer les polynômes d'Adomian pour n'importe quel ordre  $n$ , il suffit de déterminer les deux suites décroissantes les  $(\alpha_i)$  et les  $(\beta_i)$ ,  $i = 1, \dots, n$ , tous les entiers solutions de l'équation :

$$\alpha_1 + \dots + \alpha_n + \beta_1 + \dots + \beta_n = n$$

avec :

$$\begin{cases} \alpha_1 \geq \alpha_2 \geq \dots \geq \alpha_n \\ \beta_1 \geq \beta_2 \geq \dots \geq \beta_n \end{cases}$$

par exemple, pour  $n = 2$ , les solutions des suites  $(\alpha_i)$  et  $(\beta_i)$ ,  $i = 1, 2$  sont données comme suit :

$\alpha_1$	$\alpha_2$	$\beta_1$	$\beta_2$
2	0	0	0
1	1	0	0
0	0	2	0
0	0	1	1
1	0	1	0

D'où, les polynômes d'Adomian pour  $n = 2$  :

$$\begin{cases} A_0 = f_{0.0}(x_0, y_0) \\ A_1 = x_1 f_{1.0}(x_0, y_0) + y_1 f_{0.1}(x_0, y_0) \\ A_2 = x_2 f_{1.0}(x_0, y_0) + y_2 f_{0.1}(x_0, y_0) + \frac{1}{2} x_1^2 f_{2.0}(x_0, y_0) + \frac{1}{2} y_1^2 f_{0.2}(x_0, y_0) + x_1 y_1 f_{1.1}(x_0, y_0) \end{cases}$$

## 1.5 Conclusion

Dans ce chapitre, on a considéré la méthode de décomposition d'Adomian pour résoudre des équations différentielles. Cette méthode est basée sur un schéma récursif permettant de déterminer la solution sous forme d'une série converge.

La MDA fournit des solutions approchée sur des petits intervalles de temps. Afin de remédier à cet inconvénient, on utilise une technique dite "*de recollement des solutions*" dont on réitère la MDA sur des petits intervalles de temps.

Nous allons utiliser dans le chapitre 3, cette technique pour la résolution du modèle de Lotka-Volterra. Dans le chapitre suivant, nous allons présenter la méthode d'optimisation d'Alienor pour résoudre un problème de contrôle optimal du modèle de Lotka-Volterra.

## CHAPITRE 2

### LA MÉTHODE D'ALIENOR

## 2.1 Introduction

De nombreux problèmes issus de la physique, de la mécanique, de la biomathématique ou du monde industriel conduisent à des problèmes d'optimisation globale.

Les méthodes permettant de résoudre cette classe de problèmes se divisent en deux catégories : les méthodes stochastiques et les méthodes déterministes qui convergent soit vers un optimum local soit vers un optimum global.

\* les méthodes stochastiques : sont essentiellement basées sur le hasard (recherche aléatoire de type Monte-Carlo). Ce sont des méthodes souvent heuristiques qui donnent une solution avec une probabilité ( $p \leq 1$ ). Aussi, il arrive qu'on obtienne des solutions locales et non globales.

\* Les méthodes déterministes : quant à elles, sont basées sur des algorithmes de descente. Mais la plupart (méthode du gradient, des variations locales,...) convergent vers des minima locaux. Si l'on veut trouver un optimum global des difficultés numériques (temps de calcul) apparaissent dès que la dimension est supérieure ou égale à 3.

C'est dans ce contexte qu'**Yves Cherruault** et **Arthur Guillez** [16] ont inventé une méthode, appelée **méthode Alienor** en 1982, qui permet de ramener la minimisation d'une fonction multivariable disons  $n$  à celle d'une fonction à une seule variable. Cette méthode est basée sur l'utilisation d'une transformation réductrice permettant de construire des courbes qui " *$\alpha$ -densifient*" l'espace  $\mathbb{R}^n$ .

Dans ce chapitre, nous allons présenter la méthode d'optimisation d'Alienor (MA) et quelques transformations réductrices, puis on résout un problème de minimisation d'une fonction quadratique à trois variables par la méthode.

## 2.2 La méthode Alienor

La méthode Alienor consiste à ramener une fonction multivariables à une fonction d'une seule variable à l'aide d'une transformation réductrice. Cette transformation est basée sur l'introduction d'une courbe qui remplit l'espace au sens de l'alpha densité, ce que nous allons définir.

**Définition 2.1** [7] : *On dit qu'un sous ensemble  $S$  de  $\mathbb{R}^n$  est  $\alpha$ -dense dans  $\mathbb{R}^n$  si  $\forall M \in \mathbb{R}^n$ , il existe au moins un point  $M' \in S$  tel que  $d(M, M') \leq \alpha$ , où  $d$  est la distance euclidienne dans  $\mathbb{R}^n$*

Dans la littérature il existe plusieurs transformations réductrices, on peut citer : la transformation d'Archimède, de Cherruault et de Mora, ... etc.

### 2.2.1 Transformation d'Archimède

La première transformation réductrice proposée par les inventeurs de cette méthode utilise la spirale d'Archimède.

Soit  $(x, y) \in \mathbb{R}^2$ , en coordonnées **polaires** ce point s'écrit :

$$x = r \cos \theta, \quad y = r \sin \theta. \quad (2.1)$$



On peut alors relier  $r$  et  $\theta$  grâce à la spirale d'Archimède d'équation :

$$r = \alpha\theta, \quad \theta \geq 0. \quad (2.2)$$

dont la représentation graphique est donnée par la figure (2.1) pour  $\alpha = 0.02$  et  $\theta \in [0, 50]$ .

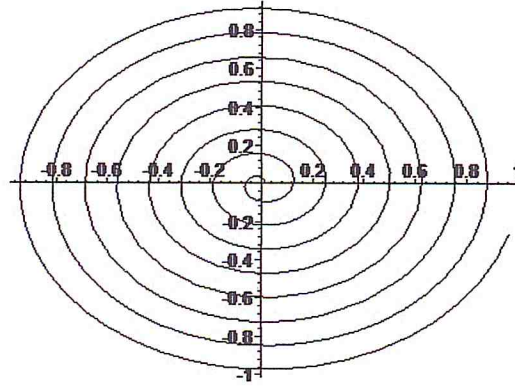


Figure 2.1. Spirale d'Archimède

Remplacer (2.2) dans (2.1) on obtient :

$$x = \alpha\theta \cos \theta = h_1(\theta)$$

$$y = \alpha\theta \sin \theta = h_2(\theta)$$

Nous avons donc exprimé  $(x, y) \in \mathbb{R}^2$  en fonction d'une seule variable  $\theta \in \mathbb{R}$ . Le paramètre  $\alpha$  est destiné tendre vers zéro afin de permettre d'approcher tout point du plan  $\mathbb{R}^2$  par un point de la courbe  $h(\theta) = (h_1(\theta), h_2(\theta))$ .

Pour trois variables  $x_1, x_2, x_3$ , on relie d'abord  $x_1$  et  $x_2$  à l'aide d'une spirale d'angle  $\theta_1$  ce qui donne :

$$x_1 = \alpha\theta_1 \cos \theta_1, \quad x_2 = \alpha\theta_1 \sin \theta_1.$$

puis on relie  $\theta_1$  et  $x_3$  à l'aide d'une spirale d'angle  $\theta$  en posant :

$$\theta_1 = \alpha\theta \cos \theta \text{ et } x_3 = \alpha\theta \sin \theta$$

Ainsi, on obtient la courbe paramétrée  $h(\theta) = (h_1(\theta), h_2(\theta), h_3(\theta))$ , définie par :

$$\begin{cases} h_1(\theta) = \alpha^2\theta \cos \theta \cos(\alpha\theta \cos \theta) \\ h_2(\theta) = \alpha^2\theta \cos \theta \sin(\alpha\theta \cos \theta) \\ h_3(\theta) = \alpha\theta \sin \theta \end{cases}$$

On peut généraliser cette transformation à  $n$  variables  $x_1, x_2, \dots, x_n$  en les reliant deux à deux par des spirales d'angle  $\theta_i$ .

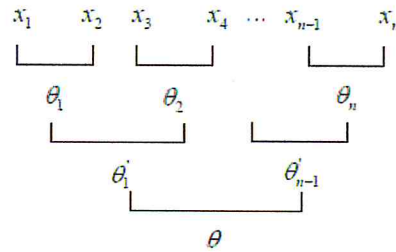


Figure 2.2. Structure d'arbre

A la fin du processus on obtient la variable  $\theta$  qui permet d'exprimer tous les  $x_i$  :

$$x_i = h_i(\theta), \quad i = 1, \dots, n.$$

où les  $h_i(\theta)$  sont des fonctions explicites de  $\theta$  (de classe  $C^\infty$ ), nous avons "approché" l'espace  $\mathbb{R}^n$  par  $\mathbb{R}$  grâce à une transformation réductrice utilisant la spirale d'Archimède.

D'après le théorème suivant la précision de cette approximation dépend du coefficient  $\alpha$  : plus  $\alpha$  est petit la précision est meilleur.

**Théorème 2.1** ([7], [8]) : *Tout point de  $\mathbb{R}^n$  peut être approché, avec la précision que l'on souhaite, par au moins un point de la transformation Alienor. La qualité de l'approximation dépend du choix de  $\alpha$ .*

Pour la démonstration le lecteur peut se référer à l'ouvrage [8].

### 2.2.2 Transformation de Cherruault

Cherruault [17] a proposée deux transformations suivantes :

$$x_i = \cos \alpha_i \theta, \quad i = 1, \dots, n. \tag{2.3}$$

et

$$x_i = \sin \alpha_i \theta, \quad i = 1, \dots, n. \tag{2.4}$$

Ce type de transformation permet d'exprimer en une seule étape toutes les variables en fonction d'une seule variable et donc obtenir plus facilement les expressions de  $h_i(\theta)$ .

En déterminant les  $x_i$  via la transformation (2.3). Nous avons remarqué que  $\cos(\infty)$  n'est pas définit. Pour cela quand  $n$  est grand, les  $\alpha_i$  sont proposés par Cherruault et al. [20] et donnés par les expressions suivantes :

$$\begin{aligned} \alpha_1 &= 1 \\ \alpha_2 &>> \alpha_1 \end{aligned}$$

où  $(\alpha_i)_i$  est une suite lentement croissante ( $i \geq 3$ ).

Pour le choix des  $\alpha_i$  on propose l'utilisation de la suite de Fibonacci définie par [21] :

$$\begin{cases} F_0 = 1, F_1 = 1 \\ F_{j+1} = F_j + F_{j-1}, \quad j = 1, 2, \dots \end{cases}$$

D'où les  $\alpha_i$  sont donnés par :

$$\begin{cases} \alpha_1 = F_1 \\ \alpha_i = \frac{F_i + 0.9}{F_{i+1}}, \quad i = 2, \dots, n. \end{cases}$$

La transformation proposée à l'aide de la suite de Fibonacci est  $\alpha$ -dense dans  $[-1, 1]^n$ .

On prend l'exemple de densification dans  $\mathbb{R}^2$  :

$$\begin{cases} x_1 = \cos \theta \\ x_2 = \cos(0.966\theta); \quad \theta \in [0, 100] \end{cases}$$

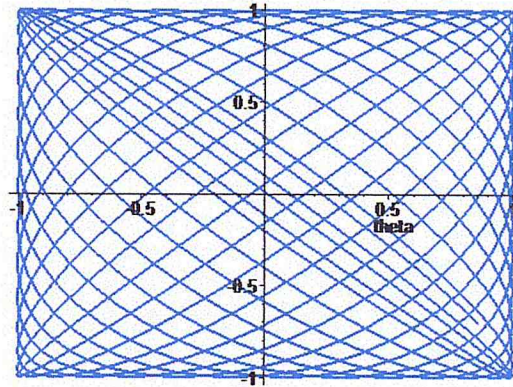


Figure 2.3. Courbe densifiant le pavé  $[-1, 1]^2$

Le théorème suivant donne une formule pour le calcul du paramètre de densification de la transformation (2.3).

**Théorème 2.2** ([17], [18]) : Soient  $\alpha_1, \dots, \alpha_{n-1}$  une suite lentement croissante. La transformation Alienor :

$$x_i = \cos \alpha_i \theta, \quad i = 1, \dots, n.$$

est  $\alpha$ -dense pour :

$$\pi \frac{\alpha_{n-1}}{\alpha_n} \sqrt{(n-1)} \leq \alpha.$$

Le paramètre de densification  $\alpha$  est obtenu pour un choix de :

$$\alpha_n = \pi \frac{\alpha_{n-1}}{\alpha} \sqrt{(n-1)}.$$

**preuve 1** Remarquons d'abord que chaque  $x_i = \cos \alpha_i \theta$  est périodique, de période  $\frac{2\pi}{\alpha_i}$ .

De plus la distance entre deux extrêma successifs vaut  $\frac{\pi}{\alpha_i}$ .

Nous allons montrer que, pour un choix adéquat de la suite  $(\alpha_i)_i$ , la transformation (2.1) remplit bien l'hypercube  $[-1, 1]^n$  avec une alpha densification que nous précisons.

Soit  $P$  un point quelconque fixé dans  $[-1, 1]^n$ . Montrons l'existence d'un point  $P' (x_1^*, x_2^*, \dots, x_n^*)$  appartenant à la courbe réductrice définie par (2.3) et vérifiant :

$$d(P, P') \leq \alpha$$

En choisissant :

$$x_n = x_n^* = \cos \alpha_n \theta^*$$

On obtient :

$$|x_n - x_n^*| = 0$$

Par ailleurs, comme chaque  $\theta_i$  appartient à un intervalle de longueur  $\frac{\pi}{\alpha_n}$ , on déduit :

$$|\theta_i - \theta^*| \leq \pi \frac{\alpha_i}{\alpha_n}, \quad i = 1, 2, \dots, n-1$$

En appliquant le théorème des accroissements finis à la fonction  $\cos(t)$ , on déduit l'inégalité suivante :

$$|x_i - x_i^*| = |\cos \alpha_i \theta_i - \cos \alpha_i \theta_i^*| \leq \pi \frac{\alpha_i}{\alpha_n}, \quad i = 1, 2, \dots, n-1,$$

ce qui implique :

$$d^2(P, P') = \sum_{i=0}^n |x_i - x_i^*|^2 \leq \sum_{i=1}^{n-1} \left(\pi \frac{\alpha_i}{\alpha_n}\right)^2 = \frac{\pi^2}{\alpha_n^2} \sum_{i=1}^{n-1} \alpha_i^2.$$

Donc, le point  $P'(x_1^*, x_2^*, \dots, x_n^*)$  approche le point  $P(x_1, x_2, \dots, x_n)$  avec un paramètre de densification qui vaut  $\frac{\pi^2}{\alpha_n^2} \sum_{i=1}^{n-1} \alpha_i^2$ . Si l'on veut obtenir une densification de paramètre  $\alpha$ , on doit satisfaire :

$$\frac{\pi^2}{\alpha_n^2} \sum_{i=1}^{n-1} \alpha_i^2 \leq \alpha^2. \quad (2.5)$$

Comme la suite  $(\alpha_i)_i$  est choisie croissante, la borne supérieure du premier terme de (2.5) est  $\frac{\pi^2}{\alpha_n^2} \sum_{i=1}^{n-1} \alpha_{i-1}^2$ . On peut alors choisir  $\alpha_n$  et  $\alpha_{n-1}$  tels que :

$$\alpha = \pi \frac{\alpha_{n-1}}{\alpha_n} \sqrt{(n-1)}.$$

La recherche de la valeur de  $\theta_{max}$  pour cette transformation (2.3), sachant que  $x_i \in [-1, 1]$  on a :

$$x_i \in [-1, 1] \Rightarrow \alpha_i \theta \in [2, 2\pi]$$

or  $(\alpha_i)_i$  est une suite lentement croissante est donné par :

$$\theta_{max} = \frac{2\pi}{\alpha_1}$$

### 2.2.3 Transformation de Mora

La transformation suivante ([4], [19]) :

$$\begin{cases} h_1(\theta) = \theta \\ h_2(\theta) = \frac{1}{2}(1 - \cos m\pi\theta), \quad m = 2, 3 \\ \dots \\ h_n(\theta) = \frac{1}{2}(1 - \cos m^{n-1}\pi\theta) \end{cases} \quad (2.6)$$

est  $\alpha$ -dense dans  $[0, 1]^n$ , ce qui montre le théorème suivant.

**Théorème 2.3** [19] : La courbe  $\mathbb{I} \rightarrow \mathbb{R}^n$  définie par les relations (2.6) densifie l'hypercube unité  $\mathbb{I}^n = [0, 1]^n$  avec un coefficient de densification

$$\alpha = \frac{1}{m} \sqrt{(n-1)}.$$

La démonstration de ce théorème se trouve dans [22].

### Exemple de transformation de Mora

Pour  $n = 2$  on a

$$\begin{cases} h_1(\theta) = \theta \\ h_2(\theta) = \frac{1}{2}(1 - \cos 3\pi\theta), \theta \in [0, 40] \end{cases}$$

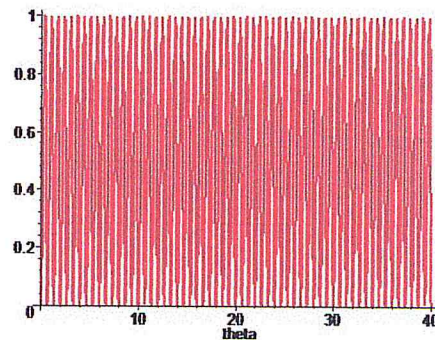


Figure 2.4. Courbe densifiant le pavé  $[0, 40] \times [0, 1]$

On peut ramener cette courbe à densifier le pavé  $[0, 1]^2$ , il suffit de faire la modification suivante :

$$\begin{cases} h_1(\theta) = \frac{\theta}{\theta_{max}}, \theta_{max} = 40 \\ h_2(\theta) = \frac{1}{2}(1 - \cos 3\pi\theta), \theta \in [0, \theta_{max}] \end{cases}$$

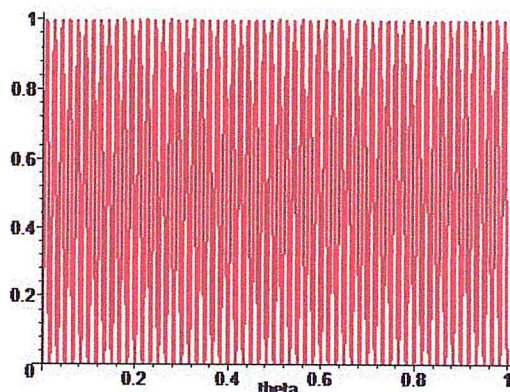


Figure 2.5. Courbe densifiant le pavé  $[0, 1]^2$

## 2.3 Application à l'optimisation globale

Notre objectif consiste à utiliser une transformation réductrice pour résoudre des problèmes d'optimisations globales en se basant sur l'approximation des fonctions de  $n$  variables par des fonctions d'une seule variable.

Soit à résoudre le problème :

$$\underset{x_1, x_2, \dots, x_n}{\text{Min}} f(x_1, x_2, \dots, x_n) \quad (2.7)$$

où  $f$  est une fonction continue sur  $\mathbb{R}^n$ .

On cherche un (ou des) point(s) réalisant le minimum global de  $f$  sachant qu'elle vérifie la condition de croissance à l'infini :

$$\lim_{x_1^2 + x_2^2 + \dots + x_n^2 \rightarrow \infty} f(x_1, x_2, \dots, x_n) = +\infty \quad (2.8)$$

Exprimant  $x_1, x_2, \dots, x_n$  par les expressions suivantes :

$$x_i = h_i(\theta), \quad \theta \geq 0 \text{ pour } i = 1, \dots, n. \quad (2.9)$$

où  $h_i(\theta) \in \mathbb{C}^\infty$  sont des fonctions définies, dites transformation réductrice et  $h(\theta) = (h_1(\theta), h_2(\theta), \dots, h_n(\theta))$  soit une courbe  $\alpha$ -dense.

Nous remplaçons (2.9) dans la fonction  $f(x_1, x_2, \dots, x_n)$ , on obtient la fonction  $f^*(\theta)$  défini par :

$$f^*(\theta) = f(h_1(\theta), h_2(\theta), \dots, h_n(\theta)). \quad (2.10)$$

où  $f^*(\theta)$  est une fonction à une seule variable. Le problème de minimisation (2.7) est alors ramené au problème de minimisation à une seule variable :

$$\underset{\theta \in [0, \theta_{max}]}{\text{Min}} f^*(\theta) \quad (2.11)$$

On ait donc conduit à chercher le ou les minima d'une fonction à une seule variable  $f^*(\theta)$ . On ne pourra le faire que sur un intervalle fermé et borné  $[0, \theta_{max}]$ . Cet intervalle dépendra du fermé et du borné de  $\mathbb{R}^n$  (généralement un pavé  $\prod_{i=1}^n [a_i, b_i]$ ) sur lequel on cherche les minima de  $f(x_1, \dots, x_n)$ .

Le résultat suivant justifie que le problème de minimisation (2.11) est une approximation du problème (2.7).

**Théorème 2.4** [8] : *Tous les minima de  $f(x_1, x_2, \dots, x_n)$  peuvent être approchés par des minima de  $f^*(\theta)$ . La réciproque est fautive en ce sens que tous les minima de  $f^*(\theta)$  ne sont pas des approximations de minima de  $f$ .*

**Remarque 2.1** : *Ce résultat signifie, en particulier que  $f^*(\theta)$  introduit des minima parasites dont il faudra se débarrasser, car c'est la restriction de  $f$  à la courbe  $h$ . Tout minimum relativement à  $h$  n'est pas forcément un minimum relativement à  $\mathbb{R}^n$ , du fait que  $h$  est strictement contenue dans  $\mathbb{R}^n$ . Ce théorème montre que toute solution du problème (2.7) peut être approchée par une solution du problème (2.11), puisque la courbe  $h \subset \mathbb{R}^n$  et la meilleure approximation sera donnée pour un  $\alpha$  petit. Pour que le minimum du problème (2.7) soit approché par le minimum du problème (2.11), il suffit de déterminer le minimum global de  $f^*(\theta)$  par des méthodes de minimisation d'une fonction à une seule variable.*

Il est possible de voir que si  $\theta^*$  est une solution du problème (2.11), alors la solution

$$x_i^* = h_i(\theta^*), \quad i = 1, \dots, n$$

est une solution approximative du problème (2.7).

**Remarque 2.2** Afin de s'assurer que la méthode ALIENOR donne le minimum on peut utiliser le point obtenu par Alienor comme point de départ de la méthode des variations locales. Si la méthode des variations locales converge au voisinage de ce point, c'est que la méthode Alienor nous a fournie une approximation d'un minimum. Sinon on s'éloigne du point donné par Alienor, c'est que l'on a obtenu un point "parasite" ne correspondant pas à un minimum.

### 2.3.1 Méthode de recherche du minimum global pour une fonction univariable

La recherche du minimum global de  $f^*(\theta)$  sur  $[0, \theta_{max}]$  s'effectue de la façon suivante :

- Le  $\theta_{max}$  borne supérieur de l'intervalle d'exploration.
- On choisit un pas  $\Delta\theta$  (qui peut être amener à varier en fonction de  $\theta$ ) pour balayer l'intervalle  $[0, \theta_{max}]$ .
- On construit les points de discrétisation  $k.\Delta\theta$ ,  $k = 0, \dots, N$  avec  $N.\Delta\theta = \theta_{max}$ .

Le minimum global (ou au moins une approximation) est trouvé en résolvant le problème :

$$\underset{k}{\text{Min}} f^*(k.\Delta\theta), \quad k = 0, \dots, N \quad (2.12)$$

On trouvera ainsi un ou plusieurs  $k^*$  réalisant le minimum de  $f$ . Ce dernier problème peut être résolu par un programme informatique.

Dans le cadre de cette méthode, la difficulté vient des points suivants :

- Le pas  $\Delta\theta$  doit être choisi comme fonction décroissante de  $\theta$  (ce qui va augmenter le nombre de points à considérer).
- L'intervalle  $[0, \theta_{max}]$  sur lequel on doit chercher les minima globaux de  $f^*(\theta)$  tel que  $\theta_{max}$  est une fonction croissante de  $n$  que l'on sait trouver.

Le théorème suivant donne une formule du calcul pour  $\theta_{max}$  de la transformation d'Archimède :

**Théorème 2.5** [8] : Pour  $n$  quelconque  $\theta_{max} = \frac{\sqrt{\sum_{i=1}^n (a_i - b_i)^2}}{2n^s}$  où  $s$  est le nombre d'étages de la transformation, défini par les inégalités :  $2^{s-1} < n \leq 2^s$ .

Le paramètre  $\theta_{max}$  dépendra de la transformation réductrice et il existe d'autres formules permettant de calculer ce paramètre (voir [8]).

#### Exemple 1

Considérons le problème de minimisation d'une fonction quadratique à trois variables ( $n = 3$ ) :

$$f_q(x, y, z) = 2x^2 + y^2 + 3z^2 - 2xy - 2xz - x - z \quad (2.13)$$

On cherche  $X^* = (x^*, y^*, z^*)$  qui minimise la fonction  $f_q$ .

La solution analytique vaut  $X^* = (1, 1, \frac{1}{2})^t$  et la valeur de la fonction objective est  $f_q(X^*) = -\frac{3}{4} = -0.75$ .

En appliquant la méthode Alienor à ce problème d'optimisation, pour ce faire il faut définir La transformation réductrice de Mora :

$$\begin{cases} x = \theta \\ y = \frac{1}{2}(1 - \cos 3\pi\theta) \\ z = \frac{1}{2}(1 - \cos 3^2\pi\theta) \end{cases} \quad (2.14)$$

puis remplacer (2.14) dans (2.13), on obtient la fonction  $f_1(\theta)$  à une seule variable  $\theta$  :

$$f_1(\theta) = 2\theta^2 + \left(\frac{1}{2}(1 - \cos 3\pi\theta)\right)^2 + 3\left(\frac{1}{2}(1 - \cos 9\pi\theta)\right)^2 - 2\theta\left(\frac{1}{2}(1 - \cos 3\pi\theta)\right) - 2\theta\left(\frac{1}{2}(1 - \cos 3\pi\theta)\right) - \theta - \left(\frac{1}{2}(1 - \cos 3\pi\theta)\right)$$

La courbe représentative de  $f_1(\theta)$  donnée par la figure (2.6) montre qu'elle admet plusieurs minima :

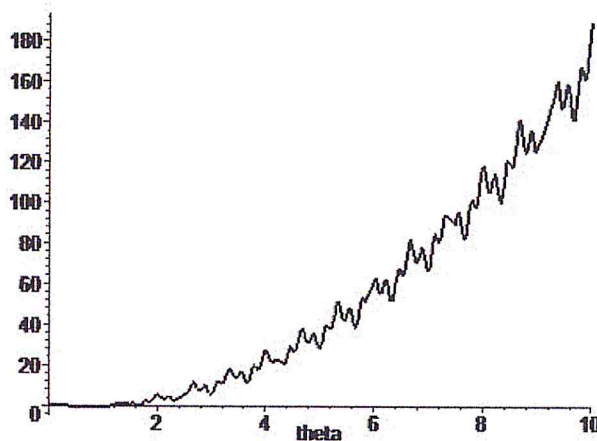
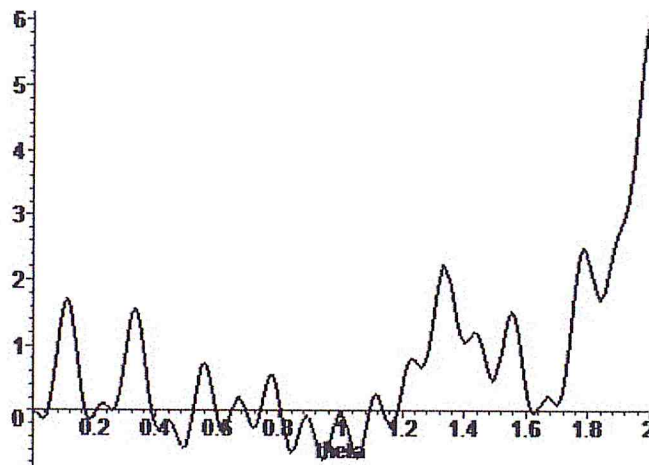


Figure 2.6. Courbe de  $f_1(\theta)$

On remarque que le minimum global se situe sur l'intervalle  $[0, 2]$  puisque la valeur de la fonction objective est minimale.



En appliquant une procédure de minimisation d'une fonction à une seule variable, le minimum obtenu est égale à :

$$x^* = 0.94321$$

$$y^* = 0.9289212254$$

$$z^* = 0.4757979829$$

et la valeur de  $f_q(X^*) = -0.7475623799$ .



## Exemple 2

Considérons la fonction suivante :

$$f_q(x, y, z) = 4(x^2 + y^2 + z^2) - 2(xy + xz + yz) - 6(x + y + z) \quad (2.15)$$

Le minimum global est atteint pour  $(x^*, y^*, z^*) = (\frac{3}{2}, \frac{3}{2}, \frac{3}{2})^t$  et la valeur de la fonction :  $f_q^*(x^*, y^*) = -\frac{27}{2} = -13.5$ .

En utilisant la transformation (2.14), on obtient la fonction  $f_2(\theta)$  à une seule variable  $\theta$  :

$$f_2(\theta) = 4\theta^2 + 4\left(\frac{1}{2}(1 - \cos 3\pi\theta)\right)^2 + 4\left(\frac{1}{2}(1 - \cos 3^2\pi\theta)\right)^2 - \theta(2 - \cos 3\pi\theta - \cos 3^2\pi\theta) - \frac{1}{2}(1 - \cos 3\pi\theta)(1 - \cos 3^2\pi\theta) - 6\theta - 3(1 - \cos 3\pi\theta) - 3(1 - \cos 3^2\pi\theta)$$

La figure (2.7) montre que  $f_2(\theta)$  admet plusieurs minima :

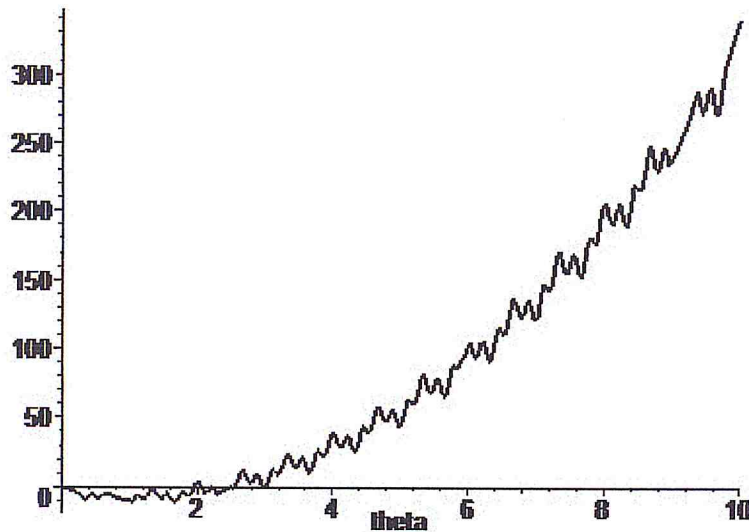


Figure 2.7. Courbe de  $f_2(\theta)$

En appliquant une procédure de minimisation d'une fonction à une seule variable, le minimum obtenu est égale à :

$$\theta^* = 1.00222 \text{ et } f_2(\theta^*) = -12.00222352$$

Remplacer  $\theta^*$  dans (3.14), on obtient :

$$(x^*, y^*, z^*) = (1.00222, 0.9998905609, 0.9990153356)$$

et la valeur de  $f_q(x^*, y^*, z^*) = -12.00222352$

**Remarque 2.3** : On remarque que la solution  $(1.00222, 0.9998905609, 0.9990153356)^t$  obtenue par la (MA) n'est pas identique à la solution analytique  $(\frac{3}{2}, \frac{3}{2}, \frac{3}{2})^t$ , pour pallier à cet inconvénient on propose l'utilisation d'une autre transformation. Au préalable il n'y a pas une étude théorique qui précise le choix de la transformation pour un problème d'optimisation.

## 2.4 Conclusion

Nous avons considéré dans ce chapitre la méthode d'Alienor qui est basée sur l'utilisation des transformations réductrices. Ces transformations sont des fonctions indéfiniment dérivables qui permettent d'approcher des fonctions à  $n$  variables par des fonctions d'une seule variable.

Nous allons utiliser la méthode Alienor pour résoudre un problème de contrôle optimale régi par l'équation de Lotka-Volterra, ce que nous allons voir dans le chapitre 4.

## CHAPITRE 3

### APPLICATION DE LA MDA AU MODÈLE DE LOTKA-VOLTERRA

### 3.1 Introduction

Nous présentons dans ce chapitre l'utilisation de la méthode Adomian au modèle dynamique du Lotka-Volterra. Nous commençons par décrire le modèle, puis nous le résolvons par la MDA. Ensuite, nous comparons les solutions obtenues par la MDA avec ceux données par la méthode de Runge-Kutta.

### 3.2 Modèle Lotka-Volterra

La dynamique des populations s'intéresse au développement numérique de toutes les populations d'êtres vivants, et plus particulièrement de celles des animaux sexués.

On s'intéresse dans cette partie au modèle de **Lotka-Volterra**.

#### 3.2.1 Description du modèle Lotka-Volterra

En mathématiques, les équations de **Lotka-Volterra**, que l'on désigne aussi sous le terme de "**modèle proie-prédateur**", sont un couple d'équations différentielles non-linéaires du premier ordre, et sont couramment utilisées pour décrire la dynamique de systèmes biologiques dans lesquels un prédateur et sa proie interagissent, les proies et les prédateurs se rencontrent au hasard, à chaque rencontre un prédateur a la possibilité de manger sa proie. Les équations s'écrivent fréquemment [23] :

$$\begin{cases} \dot{x}(t) = \alpha x(t) - \beta x(t)y(t) \\ \dot{y}(t) = -\gamma y(t) + \delta x(t)y(t) \\ x(0) = x_0 \quad y(0) = y_0 \end{cases} \quad (3.1)$$

Elles ont été proposées indépendamment par **Alfred James Lotka** en 1925 et **Vito Volterra** en 1926. Ce système d'équations est classiquement utilisé comme modèle pour la dynamique du **Lynx** et du **Lièvre des neiges**.

Où :

- $t$  est le temps.
- $x(t)$  est le nombre des proies en fonction du temps (par exemple, *Lièvre des neiges*).
- $y(t)$  est le nombre des prédateurs en fonction du temps (par exemple, *Lynx*).
- les dérivées  $\dot{x}(t)$  et  $\dot{y}(t)$  représentent la variation des populations au cours du temps.

Les paramètres suivants caractérisent les interactions entre les deux espèces :

- $\alpha$  taux de reproduction des proies (constant, indépendant du nombre de prédateurs).
- $\beta$  taux de mortalité des proies dû aux prédateurs rencontrés.
- $\gamma$  taux de mortalité des prédateurs (constant, indépendant du nombre de proies).
- $\delta$  taux de reproduction des prédateurs en fonction des proies rencontrées et mangées.

$\alpha, \beta, \gamma, \delta$  sont des réels positifs.

### 3.2.2 Signification physique des équations

Une fois développées, les équations prennent une forme utile pour une interprétation physique.

**Proies :** L'équation de la proie devient :

$$\dot{x}(t) = \alpha x(t) - \beta x(t)y(t)$$

Les proies sont supposées avoir une source illimitée de nourriture et se reproduire exponentiellement si elles ne sont soumises à aucune prédation, cette croissance exponentielle est représentée dans l'équation ci-dessus par le terme  $\alpha x(t)$ . Le taux de prédation sur les proies est supposé proportionnel à la fréquence de rencontre entre les prédateurs et les proies, il est représenté ci-dessus par  $\beta x(t)y(t)$ . Si l'un des termes  $x(t)$  ou  $y(t)$  est nul, alors il ne peut y avoir aucune prédation. Avec ces deux termes, l'équation peut alors être interprétée comme : la variation du nombre de proies est donnée par sa propre croissance moins le taux de prédation qui leur est appliqué.

**Prédateurs :** L'équation du prédateur devient :

$$\dot{y}(t) = \delta x(t)y(t) - \gamma y(t)$$

Dans cette équation,  $\delta x(t)y(t)$  représente la croissance de la population prédatrice. (Notons la similarité avec le taux de prédation ; cependant, une constante différente est utilisée car la vitesse à laquelle la population des prédateurs augmente n'est pas nécessairement égale à celle à laquelle il consomme la proie). De plus,  $\gamma y(t)$  représente la mort naturelle des prédateurs, c'est une décroissance exponentielle.

L'équation représente donc la variation de la population de prédateurs en tant que croissance de cette population, diminuée du nombre de morts naturelles.

## 3.3 Résolution du modèle Lotka-Volterra par la MDA

La méthode décompositionnelle d'Adomian appliquée au système (3.1) consiste à :

- 1- Ecrire le système (3.1) sous forme canonique, en appliquant une intégration entre 0 et  $t$  au système différentiel.
- 2- Déterminer les polynômes d'Adomian  $A_j$  associés au terme non linéaire :

$$N = f(x, y) = xy = \sum_{j=0}^{\infty} A_j$$

- 3- Chercher les solutions  $x$  et  $y$  sous forme de séries :

$$x = \sum_{j=0}^{\infty} x_j$$

et

$$y = \sum_{j=0}^{\infty} y_j$$

Les conditions initiales données sont :

$$\begin{aligned} x_0 &= 18 \\ y_0 &= 7 \end{aligned} \quad (3.2)$$

Ecrivant le système (3.1) sous la forme canonique, on obtient :

$$\begin{cases} x = x_0 + \alpha L^{-1} \sum_{j=0}^{\infty} x_j - \beta L^{-1} \sum_{j=0}^{\infty} A_j \\ y = y_0 - \gamma L^{-1} \sum_{j=0}^{\infty} y_j + \delta L^{-1} \sum_{j=0}^{\infty} A_j \end{cases} \quad (3.3)$$

Utilisons la formule de calcul des polynômes d'Adomian  $A_j$  pour une fonction à deux variables (voir chapitre 1. Section 1.4.1), on aura :

$$\begin{cases} A_0 = f_{0.0}(x_0, y_0) = x_0 y_0 \\ A_1 = x_1 f_{1.0}(x_0, y_0) + y_1 f_{0.1}(x_0, y_0) = x_1 y_0 + x_0 y_1 \\ A_2 = x_2 f_{1.0}(x_0, y_0) + y_2 f_{0.1}(x_0, y_0) + \frac{1}{2} [x_1^2 f_{2.0}(x_0, y_0) + y_1^2 f_{0.2}(x_0, y_0)] + x_1 y_1 f_{1.1}(x_0, y_0) \\ \quad = x_2 y_0 + x_0 y_2 + x_1 y_1 \\ \cdot \\ \cdot \\ \cdot \end{cases}$$

Le système (3.3) peut s'écrire :

$$\begin{cases} x_0 = x(0), y_0 = y(0) \\ x_j = \alpha L^{-1} x_{j-1} - \beta L^{-1} A_{j-1} \\ y_j = -\gamma L^{-1} y_{j-1} + \delta L^{-1} A_{j-1}, j = 1, 2, \dots \end{cases} \quad (3.4)$$

Une solution approchée à l'ordre "s" pour les séries solutions  $x(t)$ ,  $y(t)$  :

$$\phi_{1s} = \sum_{j=0}^{s-1} x_j$$

et

$$\phi_{2s} = \sum_{j=0}^{s-1} y_j$$

où les termes  $x_j$ ,  $y_j$  sont obtenus à partir des formules (3.4).

Le polynôme d'Adomian  $A_0$  associé au terme non linéaire est :

$$A_0 = f(x_0, y_0) = x_0 y_0 = 126.$$

Les  $x_j$ ,  $y_j$  des séries tronquées  $\phi_{1s}$ ,  $\phi_{2s}$ , à l'ordre 3 sont déterminés à partir des formules (3.4) :

$$x_1 = \alpha \int_0^t x_0 du - \beta \int_0^t A_0 du = \alpha \int_0^t 18 du - \beta \int_0^t 126 du = (18\alpha - 126\beta)t$$

$$y_1 = -\gamma \int_0^t y_0 du + \delta \int_0^t A_0 du = -\gamma \int_0^t 7 du + \delta \int_0^t 126 du = (126\delta - 7\gamma)t$$

Pour le polynôme d'Adomian  $A_1$ , on obtient :

$$A_1 = x_1 y_0 + x_0 y_1 = 7(18\alpha - 126\beta)t + 18(126\delta - 7\gamma)t$$

Les  $x_2, y_2$  sont :

$$\begin{aligned} x_2 &= \alpha \int_0^t x_1 du - \beta \int_0^t A_1 du \\ &= \alpha \int_0^t (18\alpha - 126\beta)u du - \beta \int_0^t [7(18\alpha - 126\beta)u + 18(126\delta - 7\gamma)u] du \\ &= \frac{\alpha}{2}(18\alpha - 126\beta)t^2 - \frac{\beta}{2}[7(18\alpha - 126\beta)t^2 + 18(126\delta - 7\gamma)t^2] \end{aligned}$$

$$\begin{aligned} y_2 &= -\gamma \int_0^t y_1 du + \delta \int_0^t A_1 du \\ &= -\gamma \int_0^t (126\delta - 7\gamma)u du + \delta \int_0^t [7(18\alpha - 126\beta)u + 18(126\delta - 7\gamma)u] du \\ &= \frac{-\gamma}{2}(126\delta - 7\gamma)t^2 + \frac{\delta}{2}[7(18\alpha - 126\beta)t^2 + 18(126\delta - 7\gamma)t^2] \end{aligned}$$

L'approximation à l'ordre 3 des séries solutions est donnée :

pour les  $x_j$

$$\begin{aligned} \phi_{13} &= \sum_{j=0}^2 x_j \\ &= x_0 + x_1 + x_2 \\ &= 18 + (18\alpha - 126\beta)t + \frac{\alpha}{2}(18\alpha - 126\beta)t^2 - \frac{\beta}{2}[7(18\alpha - 126\beta)t^2 + 18(126\delta - 7\gamma)t^2] \end{aligned}$$

pour les  $y_j$

$$\begin{aligned} \phi_{23} &= \sum_{j=0}^2 y_j \\ &= y_0 + y_1 + y_2 \\ &= 7 + (126\delta - 7\gamma)t + \frac{-\gamma}{2}(126\delta - 7\gamma)t^2 + \frac{\delta}{2}[7(18\alpha - 126\beta)t^2 + 18(126\delta - 7\gamma)t^2] \end{aligned}$$

La MDA appliquée au modèle Lotka-Volterra donne les solutions sous forme de séries où les paramètres du modèle apparaissent explicitement dans les termes des séries  $x_j(t)$ ,  $y_j(t)$  qui sont des fonctions continues des paramètres  $(\alpha, \beta, \gamma, \delta)$  et la variable temps  $t$ .

### 3.3.1 Application numérique

On suppose que deux populations, des proies (papillons par exemple), et des prédateurs (araignées par exemple) sont modélisées par un modèle de **Lotka-Volterra** avec  $\alpha = 2$ ,  $\beta = 0.4$ ,  $\gamma = 1$ , et  $\delta = 0.1$ . [24]

Les conditions initiales sont données comme suit :

$$x_0 = 18 \quad , \quad y_0 = 7$$

En substituant ces données dans les solutions tronquées  $\phi_{13}$ ,  $\phi_{23}$ , on obtient :

$$\phi_{13} = \sum_{j=0}^2 x_j = x_0 + x_1 + x_2 = 18 - 14.4t - 14.4t^2$$

$$\phi_{23} = \sum_{j=0}^2 y_j = y_0 + y_1 + y_2 = 7 + 5.6t - 2.8t^2$$

Nous comparons ces solutions avec ceux données par la méthode de Runge-kutta sur l'intervalle de temps  $[0, 1]$  avec un pas  $h = 0.1$ .

Les figures (3.1) et (3.2) représentent la superposition des courbes des solutions obtenues par la MDA (blue) et ceux données par Runge-Kutta (noir) pour  $x$  et  $y$  :

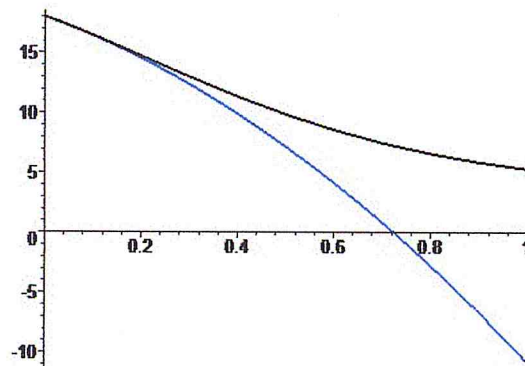


Figure 3.1. Superposition des courbes des solutions de  $x$

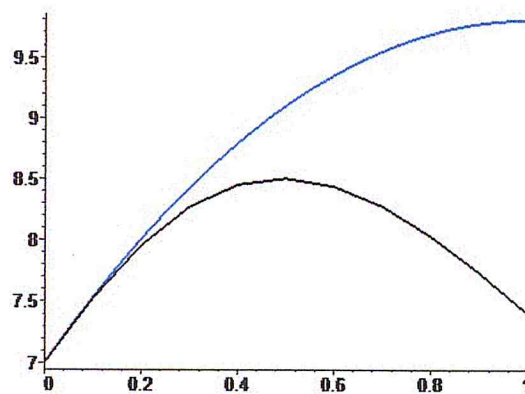


Figure 3.2. Superposition des courbes des solutions de  $y$

Nous avons constaté d'après ces courbes que nous avons une bonne superposition des solutions sur l'intervalle  $[0, 0.1]$ . Pour avoir une superposition sur de grands intervalles de temps, nous allons utiliser une technique dite le recollement des solutions de la MDA.



### 3.3.2 Le recollement des solutions par la MDA

Nous avons remarqué que la superposition des courbes n'est pas parfaite sur tous l'intervalle de temps, pour pallier à ce problème, on propose la technique "de recollment des solutions" ([12], [21]).

Elle permet pas à pas, en réitérant l'application de la méthode décompositionnelle de s'approcher la solution.

Le principe de la technique est le suivant [21] :

Les séries  $x_i = \sum_{j=0}^{\infty} x_{i,j}$ ,  $i = 1, \dots, p$  sont convergentes sur  $I_0 = [t_0, t_1]$ , où  $t_0 = 0$  et  $t_1$  est la borne supérieure pour laquelle la solution obtenue par la MDA coïncide avec la solution exacte.

On peut considérer que la restriction notée  $x_i$  à  $I_0$  est la solution du problème ( $P_0$ ) :

$$P_0 = \begin{cases} \frac{dx_i}{dt} = f_i(x_1, \dots, x_n, t) + g_i, & t \in [t_0, t_1], \quad t_0 = 0 \\ x_i(t_0) = x_{i,0}^*, & i = 1, \dots, p \end{cases}$$

On note  $x_{i,j}^{(0)}$  les restrictions à  $I_0$  des fonctions  $x_i$  obtenues par les relations récurrentes (1.29) (ou par (1.21) dans le cas d'équation différentielle) et par les solutions approchées du problème  $P_0$  :  $\phi_{is}^{(0)}(t) = \sum_{j=0}^{s-1} x_{i,j}^{(0)}$ ,  $i = 1, \dots, p$ .

On cherche les solutions  $x_i$ ,  $i = 1, \dots, p$  sur  $I_1 = ]t_1, t_2[$  telle que :

$$P_1 = \begin{cases} \frac{dx_i}{dt} = f_i(x_1, \dots, x_n, t) + g_i, & t \in [t_1, t_2] \\ x_i(t_1) = \phi_{is}^{(0)}(t_1), & i = 1, \dots, p \end{cases}$$

Ainsi, définissons le problème  $P_k$ .

Trouver  $x_i$ ,  $i = 1, \dots, p$  sur l'intervalle  $I_k = ]t_k, t_{k+1}[$  telles que :

$$P_k = \begin{cases} \frac{dx_i}{dt} = f_i(x_1, \dots, x_n, t) + g_i, & t \in [t_k, t_{k+1}] \\ x_i(t_k) = \phi_{is}^{(k-1)}(t_k), & i = 1, \dots, p \end{cases}$$

où  $\phi_{is}^{(k-1)} = \sum_{j=0}^{s-1} x_{i,j}^{(k-1)}$ ,  $i = 1, \dots, p$  sont les séries solutions tronquées du problème  $P_k$  obtenues par la MDA.

On définit le pas de recollement  $h$  par :

$$h = \min_{0 \leq j \leq r} |t_j - t_{j+1}|$$

où  $r$  est le nombre de recollements effectués pour balayer l'intervalle  $[0, T]$ .

Appliquons cette technique à l'exemple présenté dans la section (3.3), avec le pas de recollement égal à  $\frac{1}{10}$ , où  $x(t)$  et  $y(t)$  sont des solutions tronquées à l'ordre 3.

D'après les figures (3.3), (3.4) et (3.5) on remarque une superposition parfaite

Des courbes des solutions données par le recollement des solutions de la MDA (ligne) et ceux données par Runge-Kutta (points) :

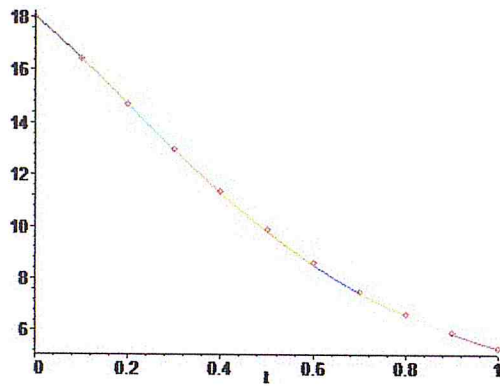


Figure 3.3. Courbe de la solution  $x$

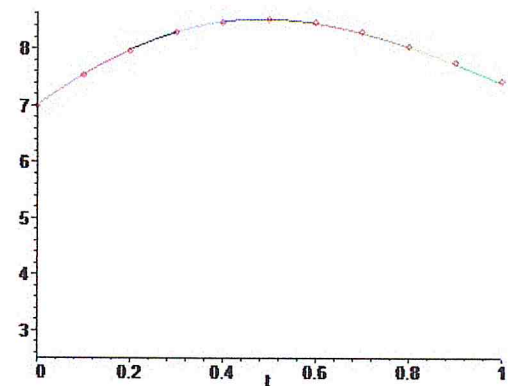


Figure 3.4. Courbe de la solution  $y$

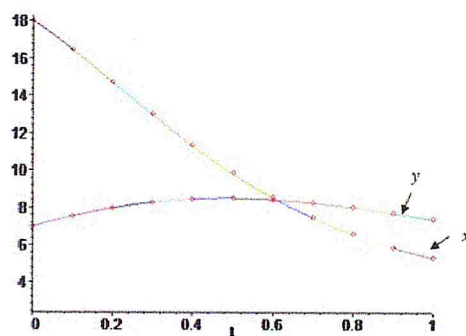


Figure 3.5. Courbe de la solution  $x$  et  $y$

### 3.4 Conclusion

On a considéré le système dynamique de Lotka-Volterra. Sa résolution par la méthode décompositionnelle d'Adomian permet de donner des solutions approchées sur des petits intervalles de temps. Afin d'avoir des solutions sur tous l'intervalle de temps on propose l'utilisation de l'approche *de recoulement de solution*.

## CHAPITRE 4

# CONTRÔLE OPTIMAL DU MODÈLE DE LOTKA-VOLTERRA

## 4.1 Introduction

La théorie de contrôle optimale permet de déterminer le contrôle d'un système qui minimise (ou maximise) un critère de performance.

Notre objectif consiste à chercher le contrôle optimal d'un modèle du Lotka-Volterra. La détermination du contrôle optimal est réalisée, en combinant la méthode décompositionnelle d'Adomian et la méthode Alienor qui permet de ramener le problème de minimisation d'une fonction dépendant explicitement des paramètres du contrôle à un problème de minimisation d'une fonction à une seule variable. Une application de la méthode combinée Adomian/Alienor au modèle de Lotka-Volterra à deux populations est considérée.

## 4.2 Formulation du problème de contrôle

La problématique du contrôle optimal est la suivante, considérons un système de contrôle sans contrainte (pour simplifier les notations, nous considérons le cas où l'instant initial et l'état initial sont fixés). Soit le système [14] :

$$\begin{cases} \dot{x}(t) = f(t, x(t), u(t)), \\ x(t_0) = x_0, \quad t \in \mathcal{I} \end{cases} \quad (4.1)$$

où  $f$  est une fonction continue de  $\mathcal{I} \times \mathbf{V} \times \mathbf{U}$  dans  $\mathbb{R}^n$ ,  $\mathcal{I}$  est un intervalle de  $\mathbb{R}$ ,  $\mathbf{V}$  un ouvert de  $\mathbb{R}^n$ ,  $\mathbf{U}$  un sous-ensemble de  $\mathbb{R}^m$ . La variable  $x$  est l'état et  $u$  est le contrôle.

Soit le critère de performance

$$J(t_f, u) = K(t_f, x_f) + \int_{t_0}^{t_f} \mathcal{L}(t, x(t), u(t)) dt$$

où  $\mathcal{L}$  vérifie les mêmes conditions que  $f$  et  $K$  est continûment différentiable sur  $\mathbf{V}$ .

Par commodité d'écriture on suppose dans toute la suite que  $t_0 = 0$ .

Soient  $M_0$  et  $M_1$  deux sous-ensembles de  $\mathbf{V}$ . Le problème de contrôle optimal est de déterminer  $x_u(\cdot)$  solutions de

$$\dot{x}_u(t) = f(t, x_u(t), u(t)),$$

telles que  $x_u(0) \in M_0$ ,  $x_u(t_f) \in M_1$ , et minimisant le coût  $J(t_f, u)$ . On dit que le problème de contrôle optimal est à temps final non fixé si le temps final  $t_f$  est libre, sinon on parle de problème à temps final fixé.

Ce problème peut être résolu par des méthodes classiques comme le principe du Maximum de Pontryaguine [[8], (14)] ou celles de la programmation dynamiques [(8), (27)].

Nous proposons l'utilisation des deux méthodes, à savoir la MDA et celle d'Alienor.

### 4.3 Principe du minimum de Pontryagin

Le principe du minimum [14] est basé sur la définition du Hamiltonien du système.

**Définition 4.1** *Le Hamiltonien du problème (4.1) est la fonction*

$$H : \mathbb{R} \times \mathbb{R}^n \times (\mathbb{R}^n \setminus \{0\}) \times \mathbb{R}^m \longrightarrow \mathbb{R}$$

$$(t, x(t), p(t), u(t)) \longrightarrow H(t, x(t), p(t), u(t)) = \langle p(t), f(t, x(t), u(t)) \rangle + \mathcal{L}(t, x(t), u(t))$$

où  $\langle, \rangle$  est le produit scalaire usuel de  $\mathbb{R}^n$ , et  $p(t)$  vecteur adjoint, et  $u$  un contrôle défini sur  $[0, t_f]$ , on note  $\mathcal{U}$  l'ensemble des contrôles admissibles sur  $[0, t_f]$

le vecteur adjoint définie comme solution du l'équation adjointe :

$$\frac{\partial H}{\partial x} = -\frac{\partial p(t)}{\partial t} = -\dot{p}(t)$$

D'où, le système Hamiltonien du problème :

$$\begin{cases} \frac{\partial H}{\partial p} = \dot{x}(t) \\ \frac{\partial H}{\partial x} = -\dot{p}(t) \end{cases} \quad (4.2)$$

Et pour  $p(t)$  et  $x(t)$  fixés, le Hamiltonien est fonction du seul paramètre  $u(t)$ .

Pour que le contrôle  $u^*(t)$  et la trajectoire correspondante  $x^*(t)$  soient optimales, il est nécessaire qu'il existe un vecteur adjoint  $p(t)$  tel que [26] :

1-  $p(t)$  vérifie l'équation adjointe.

2-  $\forall t \in [t_0, t_f]$ , le Hamiltonien  $H(t, x(t), p(t), u(t))$  atteint le minimum par rapport à la variable  $u$  au point  $u^*(t)$ .

### 4.4 Utilisation de la méthode combinée Adomian/Alienor

Le problème consiste à déterminer le contrôle  $u(t)$  minimisant le critère :

$$I = \int_0^T F(x, u) dt \quad (4.3)$$

sachant que  $x$  et  $u$  doivent satisfaire le système différentiel suivant :

$$\begin{cases} \dot{x} = \frac{dx}{dt} = f(x, u, t) \\ x_i(0) = \beta_i, \quad i = 1, \dots, p \end{cases} \quad (4.4)$$

On représente chaque composante  $u_j(t)$  de vecteur du contrôle  $u(t)$  par :

$$u_j(t) = \sum_{k=1}^r \mu_k^j \varphi_k(t), \quad j = 1, 2, \dots, l \quad (4.5)$$

où  $\mu_k^j$  sont des constantes à déterminer,  $\varphi_k(t)$  sont des fonctions connues correspondant à l'approximation choisie : (approximation polynomiale, par des fonctions

spline, des exponentielles,...). Le système différentiel (4.4) est résolu par la méthode décompositionnelle d'Adomian, après avoir exprimé le vecteur de contrôle  $u(t)$  par les approximations (4.5).

La méthode d'Adomian donnera les solutions approchées  $x_i(t)$ ,  $i = 1, \dots, p$  sous la forme de séries tronquées (à l'ordre  $s$ ) :

$$x_i(t) = \sum_{j=0}^{s-1} v_j^i(\mu_k^1, \dots, \mu_k^l, t), \quad i = 1, \dots, p \quad (4.6)$$

où chaque solution  $x_i(t)$  est fonction explicite de paramètres du vecteur de contrôle  $\mu_k^j$  et de la variable temps  $t$ .

En reportant les expressions (4.6) dans la fonction (4.3), on obtient le problème de minimisation :

$$I = \underset{\mu_k^1, \dots, \mu_k^l}{\text{Min}} \int_0^T F \left( \sum_{j=0}^{s-1} v_j^1(\mu_k^1, \dots, \mu_k^l, t), \dots, \sum_{j=0}^{s-1} v_j^p(\mu_k^1, \dots, \mu_k^l, t), \sum_{k=0}^r \mu_k^1 \varphi_k(t), \dots, \sum_{k=0}^r \mu_k^l \varphi_k(t) \right) dt \quad (4.7)$$

qui est un problème de minimisation par rapport aux paramètres  $\mu_k^j$ , soit  $r.l$  variables. Ces paramètres peuvent être réduits par Alienor à une seule variable  $\theta$ .

On définit la transformation réductrice pour les paramètres du vecteur de contrôle par :

$$\mu_k^i = \psi_k^i(\theta), \quad i = 1, \dots, l \text{ et } k = 1, \dots, r \quad (4.8)$$

La transformation  $\psi_k^i(\theta)$  est choisie de sorte qu'elle densifie l'espace  $\mathbb{R}^{r.l}$ . En substituant (4.8) dans la fonction (4.7), le problème de recherche de minimum global (4.7) est approché par un problème de minimisation d'une fonction à une seule variable :

$$\underset{\theta}{\text{Min}} I^*(\theta) \quad (4.9)$$

où :

$$I^*(\theta) = \int_0^T F \left( \sum_{j=0}^{s-1} v_j^1(\psi_k^1(\theta), \dots, \psi_k^l(\theta), t), \dots, \sum_{j=0}^{s-1} v_j^p(\psi_k^1(\theta), \dots, \psi_k^l(\theta), t), \sum_{k=0}^r \psi_k^1(\theta) \varphi_k(t), \dots, \sum_{k=0}^r \psi_k^l(\theta) \varphi_k(t) \right) dt$$

La fonction  $I^*(\theta)$  est une fonction continue sur un domaine fermé et borné, elle admet au moins un minimum dans  $\theta \in [0, \theta_{max}]$ . Y.Cherruault [7], a montré que toute solution du problème (4.9) est une approximation de la solution de (4.6), par conséquent le problème d'existence et de l'unicité du vecteur de contrôle et celui des paramètres est ramené à chercher le minimum global de  $I^*(\theta)$  par des méthodes de minimisation d'une fonction à une seule variable.

## 4.5 Application au modèle de Lotka-Volterra

Soit le problème de contrôle optimal de deux populations, des proies  $x$ , et des prédateurs  $y$ . On peut introduire un contrôle  $u(t)$  représentant la fonction de chasse qui peut être constant ou continu (exponentielle, polynômial ...). Le système de contrôle est donc représenté par :

$$\begin{cases} \frac{dx(t)}{dt} = x(t)(2 - 0.4y(t)) \\ \frac{dy(t)}{dt} = y(t)(-1 + 0.1x(t)) - c u(t)y(t) \\ x(0) = 18, \quad y(0) = 7 \\ 0 \leq u(t) \leq 1 \end{cases} \quad (4.10)$$

où  $c$  est une constante représente le niveau maximum de chasse à l'instant  $t$  des prédateurs (si  $c = 1$  désigne qu'il n'y a pas de limite pour la chasse).

le but c'est de chercher  $u(t) \in [0, 1]$  où  $u(t) = 0$  désigne qu'il n'y a pas de chasse, et  $u(t) = 1$  nous chassons le nombre maximum des individus (si  $c = 1$ , on suppose de qu'on a chassé toute la population à l'instant  $t$ ).

L'objectif sera donc de minimiser la fonctionnelle suivante :

$$I = \frac{1}{2} \int_0^T u^2(t) dt + (y(T) - d)^2$$

où  $T = 1$ . Ceci signifie de minimiser une fonction de contrôle durant  $[0, T]$  et maintenir le nombre de prédateurs à l'instant  $T = 1$  autour d'une constant  $d = 5.5$ . Nous avons normalisé la valeur  $d$  qui peut être 550 prédateurs en réalité.

On veut chercher une fonction de chasse optimale  $u^*(t)$  tel que :

$$u^*(t) = \min(1, u(t))$$

Nous recherchons le contrôle  $u(t)$  sous les formes suivantes :

$$u^e(t) = c_1 e^{-c_2 t}$$

$$u^1(t) = -c_3 t + c_4$$

$$u^2(t) = -c_5 t^2 + c_6 t + c_7$$

avec  $c_i$ ,  $i = 1..7$  sont des paramètres à déterminer en minimisant le critère  $I$ .

Le système muni de son contrôle est résolu par la méthode d'Adomian, la solution obtenue, fonction des paramètres de contrôle, est reportée dans le critère qui, à son tour, est minimisé par la méthode Alienor.

Les solutions tronquées à l'ordre 3 du système dépendent explicitement des paramètres du contrôle.

Après substitution de la solution  $y(t)$  et du contrôle  $u^j(t)$ ,  $j = 1, 2$  et  $e$  dans la fonctionnelle  $I$ , celle ci est minimisée par la méthode Alienor avec la transformation :

$$\begin{cases} \xi_1 = \theta \\ \xi_i = h_i(\theta), \quad i = 2, \dots, n \end{cases} \quad (4.11)$$

où  $n$  est le nombre de paramètres et les  $\xi_i$  correspondent aux paramètres de contrôle.

On pose les transformation suivantes :

$\mathbf{u^e(t)}$	$\mathbf{u^1(t)}$	$\mathbf{u^2(t)}$
$c_1 = \theta$	$c_3 = \theta$	$c_5 = \theta$
$c_2 = 0.5 * (1 - \sin(2\theta))$	$c_4 = 0.5 * (1 - \cos(3\theta))$	$c_6 = 1 - \cos(2\theta)$
		$c_7 = 1 - \cos(4\theta)$

Les résultats obtenus par la méthode combinée Adomian/Alienor pour différentes fonctions de contrôle sont résumés dans le tableau suivant (nous prenons le niveau maximum de chasse  $c = 0.65$ ) :

Tableau 4.1. Valeurs des paramètres obtenus par : Adomian/Alienor

Paramètres	$\mathbf{u^e(t)}$	$\mathbf{u^1(t)}$	$\mathbf{u^2(t)}$
$c_1$	0.59124	-	-
$c_2$	0.0372260692	-	-
$c_3$	-	0.93120	-
$c_4$	-	0.9700295672	-
$c_5$	-	-	0.28861
$c_6$	-	-	0.1620170778
$c_7$	-	-	0.5955692441
Valeur de I	0.1704461625	0.4603677090	0.1713258529

La figure (4.1) présente les courbes des différentes fonctions de contrôle :

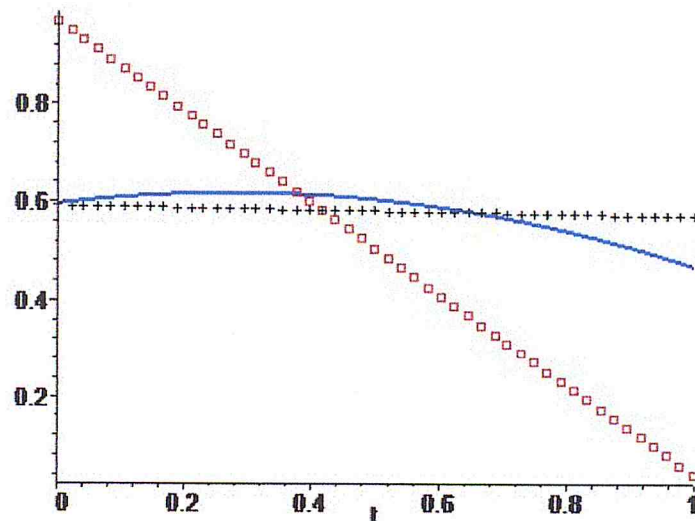


Figure 4.1. Courbes de contrôle  $u^e$  (+),  $u^1$  (points) et  $u^2$  (ligne)

La figure 4.2, représente la comparaison de la quantité des proies  $x$ , sans contrôle et avec contrôle optimal pour les différentes fonctions de contrôle,  $u^1$  (bleu),  $u^2$  (jaune) et  $u^e$  (rouge).



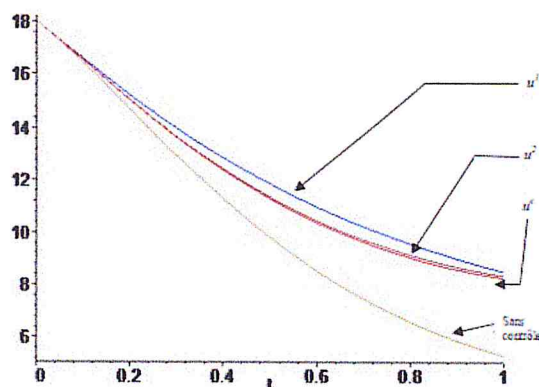


Figure 4.2. Superposition des courbes des solutions  $x$  sans contrôle et avec contrôle optimal

La figure 4.3, représente la comparaison de la quantité des prédateurs  $y$ , sans contrôle et avec contrôle optimal pour les différentes fonctions de contrôle,  $u^1$  (bleu),  $u^2$  (jaune) et  $u^e$  (rouge).

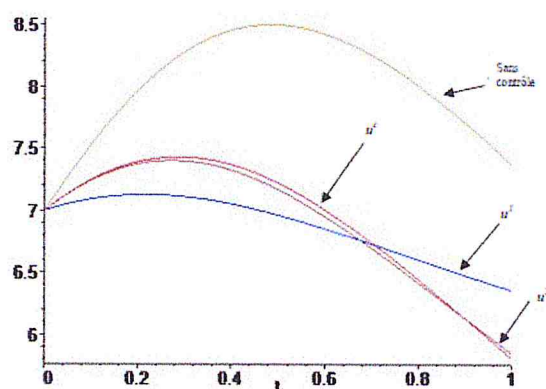


Figure 4.3. Superposition des courbes des solutions  $y$  sans contrôle et avec contrôle optimal

## 4.6 Discussion

On remarque que les trois fonctions de contrôle donnent des résultats satisfaisants dans le sens que le nombre de prédateurs à l'instant  $T = 1$  est maintenu autour d'une constante désiré  $d = 5.5$ . La valeur optimale du critère est atteinte pour une fonction de chasse  $u^e(t) = c_1 e^{-c_2 t}$  de coût  $I^* = 0.1704461625$  (voir Tableau 4.1.).

D'après la figure 4.2 on constate que le nombre des proies diminue sans contrôle, alors que il reste à un niveau acceptable pour  $u^1$ ,  $u^2$  et  $u^e$ .

On remarque que le nombre des prédateurs augmente sans contrôle, et pour les trois fonctions de contrôle il diminue et reste à un niveau acceptable (voir figure 4.3).

Concernant le choix de la fonction de chasse, le chasseur est libre dans son choix, en se basant sur :

- La valeur de la fonction objective.
- le niveau maximum de chasse.
- La courbe de contrôle (exponentielle, polynôme d'ordre 1, polynôme d'ordre 2).

## Conclusion générale

Il s'agit d'un travail consacré à l'étude de deux méthodes mathématiques, à savoir la méthode de décomposition d'Adomian et celle d'Alienor.

On a considéré la méthode décompositionnelle d'Adomian pour la résolution d'équations fonctionnelles non linéaires. Elle est basée sur la recherche des solutions sous forme de séries convergentes et sur la décomposition de l'opérateur non linéaire en série à l'aide de polynômes dits d'Adomian. La programmation de la méthode pour la résolution des équations différentielles a été réalisée à l'aide de logiciel de calcul formel, par exemple *MAPLE*. C'est ainsi que, même les systèmes dynamique non linéaires peuvent être résolus avec l'approximation souhaitée, grâce à cette méthode qui a comme avantage de ne discrétiser ni l'espace ni le temps et donne les solutions en fonction explicite des paramètres qui interviennent dans ces équations.

la méthode d'optimisation globale Alienor pour recherche le minimum d'une fonction à plusieurs variables. Cette méthode utilise essentiellement un concept développé par Y.Cherruault, elle consiste à exprimer les variables inconnues par des transformations réductrice densifiant l'espace des paramètres par une courbe  $\alpha$  - *dense* et permet de remplacer le problème de minimisation d'une fonction de " $n$ " variables par celui d'une fonction a une seule variable.

Au cours de ce travail nous avons proposé une méthodologie basée sur deux méthode, la MDA et d'Alienor, pour résoudre un problème de contrôle optimal régi par le système de Lotka-Volterra.

L'utilisation de la méthode combinée Adomian/Alienor pour la résolution de ce problème nous a permis de le réduire à un problème de minimisation d'une fonction à une seule variable.

Le travail qui nous a été proposé avait pour objectif de déterminer le contrôle optimal du modèle en minimisant une certaine fonction objective. Ce contrôle correspond à une fonction dépendant explicitement des paramètres de modèle et de la variable temps. Il est obtenu par la méthode combinée Adomian/Alienor.

Les perspectives que nous envisageons dans le prolongement de ce travail s'articulent autour des points suivants :

- \* Comparer notre méthodologie avec des méthodes d'optimisation locales ou globales.
- \* Résoudre des systèmes différentiels non linéaire d'ordre supérieure à un par la MDA.

- \* Contrôle optimal des équations aux dérivées partielles avec conditions aux limites.

Enfin, nous souhaitons que ce modeste travail contribuera à l'avancement de la recherche dans ce domaine et aidera les futurs chercheurs dans leurs travaux.

## BIBLIOGRAPHIE

- [1] Adomian. G, "Nonlinear Stochastic operators equations", Academic Press, (1986).
- [2] Adomian. G, "Nonlinear Stochastic Systems Theory and Applications to physic", Kluwer Academic, (1989).
- [3] Cherruault. Y, "Convergence of Adomian's Method", *Kybernetes*, V.18, n<sup>o</sup>2, 31 – 38, (1989).
- [4] Abbaoui. K, Cherruault. Y, "Convergence of Adomian's method applied to differentiel equations", *Computers Math applic*, V.28, n<sup>o</sup>5, 103 – 109, (1994).
- [5] Abbaoui. K, Cherruault. Y, "New ideas for proving convergence of decomposition methods", *Computers Math. Applic.*, V.29, n<sup>o</sup>7, 103 – 108, (1995).
- [6] Abbaoui. K, "Les fondements mathématiques de la méthode décompositionnelle d'Adomian et application à la résolution de problèmes issus de la biologie et de la médecine", Thèse de l'Université Paris VI. Laboratoire MEDIMAT, (1995).
- [7] Cherruault. Y, "Optimisation : méthodes locales et globales", Presses Universitaires de France (P.U.F), Paris, (1999).
- [8] Cherruault. Y, "Modèles et méthodes mathématiques pour les sciences du vivant", Presses Universitaires de France (P.U.F), Paris, (1998).
- [9] Abbaoui. K, Cherruault. Y, "New ideas for proving convergence of decomposition methods", *Computers Math. Applic*, V.29, n<sup>o</sup>7, 103 – 103, (1995).
- [10] Guellal. S, Grimalt. P, Cherruault. Y, "Numerical study of Lorentz's equation by the Adomian method", *Computers Math. Applic.*, V.33, n<sup>o</sup>3, 25 – 29, (1997).
- [11] Abbaoui. K, Cherruault. Y, "The decomposition method applied to the cauchy problem", *Kybernetes*, V.28n<sup>o</sup>1, 68 – 74, (1999).
- [12] Khelifa. S, "Equations aux dérivées partielles et méthode décompositionnelle d'Adomian", Thèse de Doctorat Es sciences, Université H.Boumedienne, Algérie, (2002).
- [13] Manseur. S, Attalah. K, Cherruault. Y, "Optimal control of chemotherapy of HIV model using the combined Adomian/Alienor methods", *Kybernetes*, V.34, n<sup>o</sup>7/8, 1200 – 1210, (2005).
- [14] Trélat. E, "Contrôle optimal : Théorie et applications Edition", Vuibert, (2005).

- [15] Abbaoui. K, Cherruault. Y, and N'Dour. M.N, "The decomposition method applied to differential systems", *Kybernetes*, V.24, n<sup>o</sup>8, 32 – 40, (1995).
- [16] Cherruault. Y, Guillez. A, "Une méthode pour la recherche du minimum global d'une fonctionnelle", *C.R.A.S, Paris, T.296, Série I*, 175 – 178, (1983).
- [17] Cherruault. Y, "New reducing transformation for global optimization (with Alienor method)", *Kybernetes*, V.34, n<sup>o</sup>7/8, 1084 – 1089, (2005).
- [18] Cherruault. Y, " $\alpha$  – dense curves and global optimization", *Kybernetes*, V.32, n<sup>o</sup>3, (2003).
- [19] Benneouala. T, Cherruault. Y, "Alienor Method for Global Optimization with a Large Number of Variables", *Kybernetes*, V.34, n<sup>o</sup>7/8, 1104 – 1111, (2005).
- [20] Cherruault. Y, Mora. G, "Optimisation globale : Théorie des courbes des courbes  $\alpha$  – denses", *Economica*, (2005).
- [21] Messaoudi. N.A, "Identification et contrôle des systèmes indéterminés non linéaire par la méthode combinée Adomian/Alienor", Thèse de magistère, Université de Blida 1, (2006).
- [22] Konfe. B.O, "Nouvelles méthodes mathématiques Alienor et Adomian, pour la Biomédecine. Thèse de Docteur". Université de Ouagadougou, (2005).
- [23] [https://fr.wikipedia.org/wiki/Équations\\_de\\_prédation\\_de\\_Lotka-Volterra](https://fr.wikipedia.org/wiki/Équations_de_prédation_de_Lotka-Volterra)
- [24] <https://perso.univ-rennes1.fr/guy.casale/enseignement/LotkaVolterra.pdf>
- [25] Ibañez. A, "Optimal control and turnpike properties of the Lotka Volterra model", Thèse de Master, Université País Vasco.
- [26] Manseur. S, " Cours : Théorie de contrôle optimale ", Université de Blida 1, (2014/2015).
- [27] Bellman. T, " Dynamic programming", Princeton university press, (1965).

## Annexe 1 : Programation de la MDA en Maple

```

> restart; Adomian := proc(ICS,Eqs,n)
local nEq, t0, m, v0, R, N, g, nOrder, u, k, nu, tempN, A, i;
nEq := nops(ICS)-1;
nOrder := [ seq(nops(ICS[m]), m=2..nEq+1) ];
t0 := ICS[1];
v0 := [seq(ICS[m],m=2..nEq+1)];
R := [seq(Eqs[m,1],m=1..nEq)];
N := [seq(Eqs[m,2],m=1..nEq)];
g := [seq(Eqs[m,3],m=1..nEq)];
u := array(1..nEq,0..n);
for m from 1 to nEq do
u[m,0] := g[m](t);
  for i from 1 to nOrder[m] do
    u[m,0] := v0[m,nOrder[m]-i+1]+int(subs(t=s,u[m,0]),s=t0..t);
  end do;
end do;
for k from 0 to n-1 do
nu := seq(sum('lambda^i*(u[m,i])', 'i'=0..k),m=1..nEq);
  for m from 1 to nEq do
    tempN := unapply((N[m])(nu),lambda);
    A[m,k] := (D@@k)(tempN)(0)/k!;
  end do;
  for m from 1 to nEq do
    u[m,k+1] := -R[m](seq(u[i,k],i=1..nEq))-A[m,k];
    for i from 1 to nOrder[m] do
      u[m,k+1] := int(subs(t=s,u[m,k+1]),s=t0..t);
    end do;
  end do;
end do;
[seq(sum('u[m,k]', 'k'=0..n),m=1..nEq)];
end:

```

## Annexe 2 : Méthodes de Runge-Kutta

Les méthodes de Runge-Kutta sont des méthodes d'analyse numérique d'approximation de solutions d'équations différentielles. Ces méthodes reposent sur le principe de l'itération, c'est-à-dire qu'une première estimation de la solution est utilisée pour calculer une seconde estimation, plus précise, et ainsi de suite<sup>1</sup>.

### Principe général :

On considère un problème de Cauchy<sup>2</sup>

$$\begin{aligned}y &= f(t, y), \quad t \in [t_0, T] \\ y(t_0) &= y_0\end{aligned}$$

et on cherche à discrétiser ce problème par rapport à une subdivision  $t_0 < t_1 < \dots < t_N = T$ . L'idée est de calculer par récurrence les points  $(t_n, y_n)$  en utilisant des points intermédiaires  $(t_{n,i}, y_{n,i})$  avec :

$$t_{n,i} = t_n + c_i h_n, \quad 1 \leq i \leq q, \quad c_i \in [0, 1]$$

où  $h_n = t_{n+1} - t_n$  est le pas de temps et  $c_i$  est dans l'intervalle  $[0, 1]$ . Pour chaque point intermédiaire, on note la pente correspondante :

$$p_{n,i} = f(t_{n,i}, y_{n,i}).$$

Soit  $y$  une solution exacte du problème. On a :

$$\begin{aligned}y(t_{n,i}) &= y(t_n) + \int_{t_n}^{t_{n,i}} f(t, y(t)) dt \\ &= y(t_n) + h_n \int_0^{c_i} f(t_n + u h_n, y(t_n + u h_n)) du, \quad \forall i = 1, \dots, q,\end{aligned}$$

grâce au changement de variable  $t = t_n + u h_n$ . De même :

$$y(t_{n+1}) = y(t_n) + h_n \int_0^1 f(t_n + u h_n, y(t_n + u h_n)) du$$

On calculera ces intégrales par une méthode d'intégration approchée, qu'on peut choisir différentes pour deux valeurs distinctes de  $i$  :

$$\int_0^{c_i} g(t) dt \approx \sum_{k=1}^{i-1} a_{ik} g(c_k) \quad \int_0^1 g(t) dt \approx \sum_{k=1}^q b_k g(c_k),$$

En appliquant ces méthodes d'intégration à  $g(u) = f(t_n + u h_n, y(t_n + u h_n))$ , il vient :

$$\begin{aligned}y(t_{n,i}) &= y(t_n) + h_n \sum_{k=1}^{i-1} a_{ik} f(t_{n,k}, y(t_{n,k})), \\ y(t_{n+1}) &= y(t_n) + h_n \sum_{k=1}^q b_k f(t_{n,k}, y(t_{n,k})).\end{aligned}$$

1. wikipedia : [https://fr.wikipedia.org/wiki/Méthodes\\_de\\_Runge-Kutta](https://fr.wikipedia.org/wiki/Méthodes_de_Runge-Kutta)  
2. Jean-Pierre DEMAILLY, ANALYSE NUMÉRIQUE ET ÉQUATIONS DIFFÉRENTIELLES, EDP Sciences, France, 2006

La méthode de Runge-Kutta d'ordre  $q$  sera donc donnée par l'algorithme :

$$\left\{ \begin{array}{l} \left[ \begin{array}{l} t_{n,i} = t_n + c_i h_n \\ y_{n,i} = y_n + h_n \sum_{k=1}^{i-1} a_{ik} p_{n,k} \\ p_{n,i} = f(t_{n,i}, y_{n,i}) \end{array} \right], \quad \forall i = 1, \dots, q \\ \\ \left[ \begin{array}{l} t_{n+1} = t_n + h_n \\ y_{n+1} = y_n + h_n \sum_{k=1}^q b_k p_{n,k} \end{array} \right] \end{array} \right.$$

On la représente conventionnellement par le tableau :

$c_1$					
$c_2$	$a_{21}$				
$c_3$	$a_{31}$	$a_{32}$			
.	.	.	.		
.	.	.	.	.	
.	.	.	.	.	
$c_q$	$a_{q1}$	$a_{q2}$	.....	$a_{q,q-1}$	
	$b_1$	$b_2$	.....	$b_{q-1}$	$b_q$

où les méthodes d'intégration approchées correspondent aux lignes. On pose par convention  $a_{ij} = 0$  pour  $j \geq i$ .

Programmation en Maple :

```
> Eq := diff(y(t), t) :
  dsolve({Eq, y(0)=y0}, y(t), type=numeric, output=listprocedure) ;
```

