

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITÉ MOHAMED KHIDER, BISKRA

FACULTÉ des SCIENCES EXACTES et des SCIENCES de la NATURE et de la VIE

DÉPARTEMENT DE MATHÉMATIQUES



Mémoire présenté en vue de l'obtention du Diplôme :

MASTER en Mathématiques

Option : **ANALYSE**

Par **CHITOUR NOUR EL HOUDA**

Titre :

Utilisation des algorithmes méta-heuristiques dans L'optimisation numérique

Membres du Comité d'Examen :

Dr. OUAAR FATIMA	UMKB	Encadreur
Dr. BELLAGOUNE ABDELGHANI	UMKB	Examineur
Dr. KABOUL HANANE	UMKB	Examineur

Juin 2021

DÉDICACE

Je dédie cet humble travail à

Ma chère maman : **Fatiha.**

Mon cher papa : **Abd el kader.**

Ma chère grand-mère : **Fatima.**

Cher enseignant : **Assia Djemai**, Ce qui m'a encouragé dans mon cheminement vers l'excellence et le succès et a été une raison de me motiver.

Mes chères frères :

Fathi, Haitham, Yaakoub.

Mes chères oncles : **Morad, khaled.**

À toute la famille.

À tous mes **collègues**, mes **amies.**

À toutes les personnes qui ont contribués de près ou de loin pour la réalisation de ce travail.

CHITOUR NOUR EL HOUDA

REMERCIEMENTS

La réalisation de cette thèse est l'aboutissement d'un parcours académique long, souvent laborieux et au cours duquel, malgré les embûches, j'ai toujours été animée par la volonté de finir, ou plutôt d'en finir.

En premier lieu, je remercie "**Dieu**" le Tout-Puissant de m'avoir aidé à réussir ma quête et de m'avoir donné courage et patience pour réaliser ce travail.

L'accomplissement de ce travail n'aurait pu être réalisé sans le soutien et la collaboration de nombreuses personnes que je tiens à remercier sincèrement :

Je tiens tout d'abord à remercier madame **OUAAR Fatima** qui fut mon directeur de thèse et dont l'enthousiasme et la générosité me donnèrent le goût de la recherche. Elle a su m'encadrer en me laissant ma liberté puis ne pas perdre le contact, qui m'a amenée à découvrir l'avantage des Statistiques et qu'elle soit remercié de sa patience, son indéfectible disponibilité pour écouter les questions et m'aider à trouver les réponses. Pour toutes ces raisons, je la remercie chaleureusement.

Je tiens à remercier le membre du jury d'être patients et d'avoir toléré d'évaluer et de juger mon travail préparé durant plusieurs jours et nuits que j'espère d'être fructueux. Ils trouvent ici toute la reconnaissance commençons par Monsieur **BELLAGOUNE ABDELGHANI**, président de jury puis par Madame **KABOUL HANANE**, examinateur, je les exprime ma profonde gratitude.

Je remercie également, tous nos enseignant(es) du département de Mathématique à l'université de Mohamed khider, qui ont contribué à nos formations pendant les années de Licence et de Master.

J'éprouve aussi une grande reconnaissance à l'égard de mes parents, qui m'ont supporté tout au long de mes études universitaires. Ils savent combien ils comptent pour moi : **Maman, Papa** ; ils m'a redonné confiance au moment où j'en avais le plus besoin, ils m'a permis de continuer ce travail sans jamais abandonner. Pour tout ça et bien plus encore, je ne les remercierai jamais assez. Votre support m'a été très précieux. J'espère leurs avoir rendu un petit peu de ce qu'ils m'a apporté.

Je tiens aussi à remercier mes chères **Frères** et **Tantes**, avec qui j'ai pu m'aérer l'esprit par

quelques discussions et quelques journées aventures ; à qui je dédie toutes ces heures de travail consacrées pour ma thèse au détriment du temps que j'aurais du passer en leur compagnie. Ils demeurent ma plus grande source de motivation et ceux à qui je dois, en premier lieu, la réalisation de cette thèse.

Je voudrais dire aussi un mot à tous ceux avec qui j'ai ou j'ai eu le plaisir de collaborer : mes **Amies** et mes **Camarades**. Ce mot, quels qu'aient été les moments de tension, et pour tous les instants de jubilation, de joie ou toutes les digressions :

"MERCI"

J'ai sans aucun doute oublié d'autres personnes. Mais je suis persuadé qu'elles me pardonneront et je suis sûre qu'elles partageront avec moi ce moment d'euphorie tant attendu.

En fin je prie **Dieu** de nous retirer cette épidémie.

5255772.

Table des matières

D'édicace	i
Remerciements	i
Table des matières	iii
Liste des figures	v
Liste des tableaux	vi
Introduction Générale	1
1 Problèmes à Valeur Initiale	3
1.1 Equation Différentielle Ordinaire	3
1.1.1 Equation Différentielle d'Ordre un	4
1.2 Equation Différentielle Linéaire du Premier Ordre	5
1.2.1 Types d'une Equation Différentielle d'Ordre un	5
1.3 Problèmes à Valeure Initiale (PVI)	9
1.3.1 Résolution d'une (EDO) par la Méthode de Runge-Kutta	10
2 Optimisation Numérique par les Algorithmes Méta-heuristiques	15
2.1 Algorithmes Méta-heuristiques	15
2.1.1 Définition d'algorithmes Méta-heuristiques	15
2.1.2 Catégories des Algorithmes Méta-heuristiques	16
2.1.3 Avantages d'utiliser les algorithmes méta-heuristiques	17

2.1.4 Limitations des Méta-heuristiques	18
2.2 Bat Algorithme	18
2.2.1 Algorithme de chauve-souris (BA)	18
2.2.2 Écholocation des chauves-souris	20
3 Optimisation d'un (PVI) par (BA)	23
3.1 Formulation du problème	23
3.1.1 Fonction objectif	24
3.1.2 Consistence	24
3.2 Résoudre les EDO pour une RL circuit en série	25
3.3 Résultats numérique	26
3.3.1 Paramètres liés au BA	26
3.3.2 Paramètres liés au PVI	26
3.3.3 Application	27
3.4 Conclusion	28
Conclusion	30
Annexe A : Le logiciel Matlab	32
Annexe B : Abréviations et Notations	34
Annexe C : Code Matlab des Méthdes Etudiées	35
3.5 Code Matlab de la Méthode de (RK4)	36
3.6 Code Matlabe de BA	37

Table des figures

- 1.1 tableau de Butcher 11
- 1.2 RK 1 12
- 1.3 schéma prédicteur-correcteur explicite. 13
- 1.4 schéma de RK-4 14

- 2.1 Comportement d'échololocation des chauves-souris. 21

- 3.1 Une RL circuit en série. 25
- 3.2 Résultats exacts, résultats BA et RK4. 28
- 3.3 Erreur absolue. 29
- 3.4 Icône du logiciel Matlab. 33

Liste des tableaux

2.1 Quelques Algorithmes avec leurs inspiration	17
3.1 Paramètres adoptés par BA.	26
3.2 Résultats numérique de l'exemple pour d=10.	27
3.3 Erreur absolue.	29

Introduction Générale

L'optimisation numérique est une branche des mathématiques qui cherche à modéliser, analyser et résoudre des problèmes analytiquement ou numériquement. L'optimisation joue un rôle important dans toutes les disciplines quantitatives de l'informatique et de l'ingénierie à la recherche opérationnelle et l'économie, dans les mathématiques appliquées (fondamentales pour l'industrie et l'ingénierie), en analyse et en analyse numérique. Beaucoup de systèmes susceptibles d'être décrits par un modèle mathématique sont optimisés. La qualité des résultats et des prédictions dépend de la pertinence du modèle, de l'efficacité de l'algorithme et des moyens pour le traitement numérique. Une métaheuristique est un algorithme d'optimisation visant à résoudre des problèmes d'optimisation difficiles (souvent issus des domaines de la recherche opérationnelle, de l'ingénierie ou de l'intelligence artificielle) pour lesquels on ne connaît pas de méthode classique plus efficace. Les métaheuristices sont généralement des algorithmes stochastiques itératifs, qui progressent vers un optimum global, c'est-à-dire l'extremum global d'une fonction, par échantillonnage d'une fonction objectif. Elles se comportent comme des algorithmes de recherche, tentant d'apprendre les caractéristiques d'un problème afin d'en trouver une approximation de la meilleure solution (d'une manière proche des algorithmes d'approximation). Il existe un grand nombre de métaheuristices différentes, allant de la simple recherche locale à des algorithmes complexes de recherche globale. Ces méthodes utilisent cependant un haut niveau d'abstraction, leur permettant d'être adaptées à une large gamme de problèmes différents. Notre choix de la fonction objective est dans la domaine économique, précisément l'équation de l'offre et de la demande qui est généralement résolue à l'aide des méthodes numériques classiques, nous en appliquons la méthode de Runge-Kutta (*RK*) comme méthode numérique de l'offre et de la demande. Nous l'avons comparé à des méthodes modernes et efficaces, que nous étudierons dans ce travail. Face à de nombreux problèmes durs et complexes qui existaient dans le monde réel, les techniques d'optimisation traditionnelles démontrent leur insuffisance, C'est ce qui pousse

les chercheurs à exploiter la vaste inspiration de la nature qui résout et optimise leurs problèmes profonds en gardant l'équilibre entre leurs systèmes. Dans ce contexte, les algorithmes métaheuristiques (MA) exploitent différents processus d'opérateurs inspirés de la nature. Récemment, les techniques de MA sont devenues largement utilisées dans diverses branches de la science et de l'économie. Dans ce mémoire on va étudier comment optimiser un problème à valeur initiale dans une équation différentielle ordinaire linéaire du premier ordre on se basant sur l'algorithme de chauves-souris (BA).

Notre objectif est d'évaluer les performances du (BA) et d'étudier ses propriétés. Jusqu'à présent, le créateur de l'algorithme n'a fait que contribuer à ce domaine. Ils établissent la forme de base du (BA) et décrivent sa variante à plusieurs objectifs.

Ce mémoire est organisé de la manière suivante :

Chapitre 1 : Le premier chapitre est consacré à considérer les Problèmes à Valeur Initiale (PVI) dans les équations différentielles ordinaires (EDO) comme des problèmes d'optimisation, résolu en utilisant les méthodes de Runge-Kutta.

Chapitre 2 : Le deuxième chapitre consiste à étudier type d'algorithme méta heuristique : l'algorithme de chauve-souris (BA).

Chapitre 3 : Il décrit les résultats d'expériences numériques à travers une étude de simulation d'un (PVI) comme un exemple d'application dont les résultats obtenus par (BA) sont comparés avec celles de méthode de Runge-Kutta qui est considérée comme une méthode traditionnelle pour résoudre les (PVI). La performance de l'algorithme en relation avec la méthode de Runge-Kutta est également étudiée. Enfin, la dernière partie de ce chapitre contiennent des remarques finales sur le (BA) et des propositions pour son applications ultérieur.

Chapitre 1

Problèmes à Valeur Initiale

Les équations différentielles sont très importantes dans la modélisation mathématique de nombreux phénomènes. Le but de cet mémoire est considérer les Problèmes de Valeur Initiale (*PVI*) dans les Equations Différentielles Ordinaires (*EDO*) comme un problème d'optimisation, résolu en utilisant un algorithme méta-heuristique qui est considéré comme un moyen alternatif de trouver une approximation numérique de (*PVI*) puisque ils peuvent être résolus simplement par des outils mathématiques classiques précis.

1.1 Equation Différentielle Ordinaire

Une équation différentielle est une équation impliquant une ou plusieurs dérivées d'une fonction inconnue. Si toutes les dérivées sont prises par rapport à une seule variable ,on parle d'équation différentielle ordinaire (*EDO*).

Définition 1.1.1 Une équation différentielle ordinaire, également noté *EDO*, d'ordre n est une relation entre la variable réelle t , une fonction inconnue $t \mapsto y(t)$ et ses dérivées $y', y'', \dots, y^{(n)}$ au point t définit par :

$$F(t, y(t), y'(t), y''(t), \dots, y^{(n)}(t)) = 0. \quad (1.1)$$

on notera par :

$$F(t, y, y', y'', \dots, y^{(n)}) = 0.$$

dont les inconnues sont : une **fonction** $y : I \subset \mathbb{R} \rightarrow \mathbb{R}^n$ et ses dérivées ; l'intervalle de définition de la fonction y est I ; tel que $t \in I$.

Dans laquelle cohabitent à la fois y et ses dérivées $(\dot{y}, \ddot{y}, \dots, y^{(n)})$, (n est appelé l'ordre de l'équation), on pourra utiliser x de temps en temps au lieu de t i.e. $(y(t)$ ou $y(x))$.

Si F à valeur dans \mathbb{R} , alors cette équation [1.1](#) est scalaire si non elle dit vectorielle.

Une (EDO) est d'ordre k si elle contient les dérivées de y jusqu'à l'ordre k .

Exemple 1.1.1 les équations :

$$\dot{y}(t) - t = 0,$$

$$\dot{y}^2(t) - y(t) = 0,$$

$$\exp(\dot{y}^2(t)) - t^2 + y(t) = 0.$$

sont des équations différentielles ordinaires.

1.1.1 Equation Différentielle d'Ordre un

C'est une (EDO) tel que $n = 1$,

Définition 1.1.2 Une équation différentielle (ED) du premier ordre est une équation de la forme :

$$F(t, y, \dot{y}) = 0. \tag{1.2}$$

où :

y est la fonction inconnue de la variable t (ou x) à valeur dans \mathbb{R}^n , et \dot{y} sa dérivée.

F est une fonction d'une partie de $\mathbb{R} * \mathbb{R}^n * \mathbb{R}^n$ vers \mathbb{R}^n .

Remarque 1.1.1 Le qualificatif ordinaire pour l'équation [1.2](#) signifie inconnue y dépend d'une seul variable t . L'orsqu'il y a plusieurs dérivées $\frac{dy}{dt}$ on parle d'équations aux dérivées partielles (EDP). L'équation [1.2](#) est dite scalaire lorsque $n = 1$ si non elle est dite vectorielle.

Considérons pour simplifier que la fonction $y(t)$ est à valeurs dans \mathbb{R} (i.e $n = 1$) soit J un ouvert de $\mathbb{R} * \mathbb{R}$ et $f : J \rightarrow \mathbb{R}$, une fonction continue. On dit qu'une équation [1.2](#) est sous forme normée (résolue), si elle s'écrit :

$$\dot{y} = f(t, y); \quad f : J \rightarrow \mathbb{R}, J \subset \mathbb{R} * \mathbb{R}. \tag{1.3}$$

Exemple 1.1.2 $F(t, y, \dot{y}) = 2\dot{y} + t^2y - 3 = 0$ une équation différentielle d'ordre un, car l'équation $\dot{y} = -\frac{1}{2}t^2y + \frac{3}{2}$ est une équation sous forme résolue $\dot{y} = f(t, y) = -\frac{1}{2}t^2y + \frac{3}{2}$ qui est continue.

1.2 Equation Différentielle Linéaire du Premier Ordre

Définition 1.2.1 Une équation différentielle linéaire du premier ordre est une équation du type :

$$\dot{y} = \alpha(t)y + \beta(t). \quad (1.4)$$

où α et β sont des fonctions continues sur un intervalle I de \mathbb{R} . Cette équation (1.4) est appelée une équation linéaire du premier ordre avec le seconde membre ; $\beta(t)$ est le seconde membre.

Remarque 1.2.1 Toute équation de la forme (1.4) est dite homogène lorsque le seconde membre est nul $\beta(t) = 0$. Dans le cas où $\beta(t) \neq 0$ on parle d'une équation non homogène.

Exemple 1.2.1 $\dot{y} - ty = 2t^3 + 6$ est une équation différentielle non homogène.

$\dot{y} = yt^2 - (y - 2)$ est une équation différentielle non linéaire.

Equations différentielles à variables séparées :

Quand on peut isoler y et \dot{y} dans un membre et le reste dans l'autre membre de l'équation différentielle, on a une équation différentielle à variables séparées. En utilise la notation $\dot{y} = \frac{dy}{dt}$, alors on obtenu une équation qui peut se mettre sous la forme $\alpha(t) dt = \beta(y) dy$ où α et β sont deux fonctions.

Exemple 1.2.2

$$t - \dot{y} = 2 \Rightarrow t - \frac{dy}{dt} = 2 \Rightarrow \frac{dy}{dt} = 2 + t \Rightarrow dy = (2 + t)dt \text{ est séparée .}$$

Exemple 1.2.3

$$t + \dot{y} = 2 + y \Rightarrow t + \frac{dy}{dt} = 2 + y \Rightarrow \frac{dy}{dt} = 2 + y - t \text{ n'est pas séparée .}$$

1.2.1 Types d'une Equation Différentielle d'Ordre un

On donne quelques types des équations différentielles d'ordre 1 :

Equation de type $\dot{y}=by$

Théorème 1.2.1 Soit b un réel fini, soit l'équation différentielle :

$$\dot{y} = by. \tag{1.5}$$

L'équation (1.5) admette pour solution sur \mathbb{R} la famille des fonctions définies par :

$$g_\gamma(t) = \gamma \exp(bt), \quad \gamma \in \mathbb{R}.$$

et ce sont les seules.

Preuve. On réécrit l'équation différentielle sous la forme :

$$\frac{\dot{y}}{y} = b,$$

et on intègre à gauche et à droite pour trouver :

$$\ln |y(t)| = bt + c \quad ; c \in \mathbb{R},$$

On compose par l'exponentielle des deux côtés pour obtenir :

$$|y(t)| = e^{bt+c},$$

Autrement dit $y(t) = \pm e^c e^{bt}$ En posant $\gamma = \pm e^c$ on obtient les solutions (non nulles) cherchées ; $y(t) = \gamma e^{bt}$.

$$y(t) = \gamma e^{bt}.$$

- L'équation différentielle (1.5) admet donc une infinité de solutions (puisque l'on a une infinité de choix de la constante γ).
- La constante γ peut être nulle. Dans ce cas, on obtient la "solution nulle" : $y = 0$ sur \mathbb{R} , qui est une solution évidente de l'équation différentielle.
- Ce théorème peut aussi s'interpréter ainsi : si y_0 est une solution non identiquement nulle de l'équation différentielle (1.5) alors toutes les autres solutions y sont des multiples de y_0 . En

termes plus savants, l'ensemble des solutions forme un espace vectoriel de dimension 1.

■

Equation de type $\dot{y} = b(t)y$

C'est une équation différentielle homogène à coefficient variable.

Théorème 1.2.2 *Soit b une fonction continue sur un intervalle I de \mathbb{R} , soit l'équation différentielle :*

$$\dot{y} = b(t)y \tag{1.6}$$

les solutions de l'équation 1.6 sont des fonctions définies sur I par :

$$g_\gamma(t) = \gamma \exp(B(t)), \quad \gamma \in \mathbb{R}.$$

où B est une primitive de b sur I et $\gamma \in \mathbb{R}$.

Preuve. On a :

$$\dot{y} = by,$$

on obtient :

$$\frac{\dot{y}}{y} = b,$$

c-à-dire

$$\frac{dy}{dt} = by \implies \frac{dy}{y} = bdt,$$

et on intègre l'égalité pour trouver :

$$\int \frac{dy}{y} = \int b(t) dt \implies \ln |y| = B(t) + c,$$

On compose par l'exponentielle des deux côtés :

$$|y| = \exp(B(t) + c) = \exp(B(t)) \exp(c),$$

On obtient

$$y(t) = \pm \exp(c) \exp(B(t)),$$

posons : $\gamma = \pm \exp(c)$

alors

$$y(t) = \gamma \exp(B(t)).$$

■

Equation de type $y' = b(t)y + d(t)$

Une équation différentielle du premier ordre linéaire à coefficients variables est une équation de la forme :

$$y' = b(t)y + d(t) \tag{1.7}$$

où b et d sont deux fonctions continues sur I dans \mathbb{R} , Pour résoudre une équation différentielle linéaire d'ordre 1 [1.4](#) il n'y a pas de nouvelle formule à apprendre pour ce cas. Il suffit d'appliquer le principe de superposition : On cherche la solution générale de l'équation homogène y_h associée est :

$$y' = b(t)y$$

Ensuite on cherche d'une solution particulière y_p de [1.7](#). Finalement :

$$y = y_h + y_p$$

y est la solution générale de l'équation [1.4](#)

Recherche d'une solution particulière y_p de l'équation [1.4](#) : On a vu que la solution homogène est :

$$y_h(t) = \gamma \exp(B(t)),$$

où B est la primitive de la fonction b sur I et $\gamma \in \mathbb{R}$, pour trouver une solution particulière y_p de l'équation [1.4](#) on utilise la méthode de **variance de la constante (méthode de Lagrange)**

on fait comme γ est une fonction et on recherche la fonction définie par :

$$y_p(t) = \gamma(t) \exp(B(t)),$$

solution de l'équation différentielle [1.4](#)

$$\dot{y}_p(t) = b(t)\gamma(t)e^{B(t)} + \gamma(t)e^{B(t)} = b(t)y_p(t) + \gamma(t)e^{B(t)},$$

car $B' = b$

$$\dot{y}_p(t) - b(t)y_p(t) = \gamma(t) e^{B(t)},$$

Si on a :

$$\gamma(t) e^{B(t)} = j(t) \Leftrightarrow \gamma(t) = j(t)e^{-B(t)} \iff \gamma(t) = \int j(t)e^{-B(t)},$$

Alors y_p est une solution de [\(1.4\)](#). Donc on obtient une solution particulière de [\(1.4\)](#) sur I :

$$y_p(t) = \left(\int j(t)e^{-B(t)} \right) e^{B(t)}.$$

Ce qui ne donne la solution générale de [\(1.4\)](#) est donnée par :

$$y(t) = y_p(t) + \gamma e^{B(t)}; \quad \gamma \in \mathbb{R}.$$

1.3 Problèmes à Valeur Initiale (PVI)

En analyse, un **problème de Cauchy** est un problème constitué d'une équation différentielle dont on recherche une solution vérifiant une certaine condition initiale. Cette condition peut prendre plusieurs formes selon la nature de l'équation différentielle. Pour une condition initiale adaptée à la forme de l'équation différentielle, le théorème de **Cauchy-Lipschitz** assure l'existence et l'unicité d'une solution au **problème de Cauchy**¹. Dans le cas d'une équation différentielle de premier ordre, de la forme $\dot{y} = f(t, y)$ [1.3](#), la condition initiale adaptée sera la donnée d'une valeur initiale pour la fonction inconnue y , et prendra la forme d'une équation

¹Soit f une fonction de deux valeurs réelles à une valeur réelle $f : I_1 * I_2 \rightarrow \mathbb{R}$, ou I_1, I_2 sont deux intervalles de \mathbb{R} , $(t, y) \mapsto f(t, y)$.

$y(t_0) = y_0$. Les hypothèses du théorème de Cauchy-Lipschitz exigent une certaine régularité de la fonction f . Voici l'énoncé du théorème de Cauchy-Lipschitz dans le cas des équations différentielles linéaires du premier ordre :

Théorème 1.3.1 (Théorème de Cauchy-Lipschitz) *Soit une équation différentielle linéaire du premier ordre $y' = f(t)y + g(t)$, où $f, g : I \rightarrow \mathbb{R}$ sont des fonctions continues sur un intervalle ouvert I . Alors, pour tout $t_0 \in I$ et pour tout $y_0 \in \mathbb{R}$, il existe une et une seule solution y , telle que $y(t_0) = y_0$. D'après nos calculs précédents cette solution est :*

$$y(t) = \left(\int_{t_0}^t g(x) \exp(-F(x)) dx \right) \exp(F(x)) + y_0 \exp(F(t)) \implies y(t_0) = y_0,$$

où F est la primitive de f s'annulant en t_0 , et cette solution vérifie bien $y(t_0) = y_0$. On appelle problème de Cauchy la donnée d'une équation différentielle et d'une condition initiale :

$$\begin{cases} \frac{dy}{dt} = f(t, y) \\ y(t_0) = y_0 \end{cases} \quad (1.8)$$

C'est un problème de condition initiale, dont le théorème de Cauchy-Lipschitz garantit l'existence et l'unicité de la solution si f est lipschitzienne.

1.3.1 Résolution d'une (EDO) par la Méthode de Runge-Kutta

Les techniques de Runge-Kutta sont des schémas numériques à un pas qui permettent de résoudre les équations différentielles ordinaires. Elles font parties des méthodes les plus populaires de part leur facilité de mise en œuvre et leur précision. C'est **Carle Runge** et **Martin Kutta** qui, ont inventé ces méthodes en 1901. 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.

Le principe générale des méthodes de Runge-kutta : Considérons le problème suivant :

$$\begin{cases} \frac{dy}{dt} = f(t, y) \\ y(t_0) = y_0 \end{cases} .$$

que l'on va chercher à résoudre en un ensemble discret $t_0 < t_1 < \dots < t_n$. Plutôt que de chercher une méthode directe, les méthodes de Runge-Kutta proposent d'introduire les points

intermédiaires $\{(t_{n,i}, y_{n,i})\}_{1 \leq i \leq q}$ afin de calculer par récurrence les valeurs (t_n, y_n) avec

$$t_{n,i} = t_n + c_i h_n \quad ,$$

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

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

Ainsi, pour une solution exacte y du problème, on a :

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

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

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

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

calculées ici pour $g(u) = f(t_n + u h_n, y(t_n + u h_n))$. La méthode de Runge-Kutta d'ordre q sera donc donnée par :

$$y_{n+1} = y_n + h_n \sum_{k=1}^q b_k p_{n,k} \quad .$$

On résume la méthode souvent par le tableau des différents poids de quadrature, appelé tableau de Butcher :

$$\begin{array}{c|ccc} c_1 & & & \\ c_2 & a_{21} & & \\ c_3 & a_{31} & a_{32} & \\ \vdots & \vdots & & \ddots \\ c_q & a_{q1} & a_{q2} & \dots & a_{q,q-1} \\ \hline b_1 & b_2 & \dots & b_{q-1} & b_q \end{array}$$

FIG. 1.1 – tableau de Butcher

La méthode est consistante si $\forall i = 2, \dots, q, \sum_{k=1}^{i-1} a_{ik} = c_i$

Méthode de Runge-kutta d'ordre 1 (RK1) : Cette méthode est équivalente à la méthode d'Euler, une méthode simple de résolution d'équations différentielles du 1 ordre. Considérons le problème suivant :

$$\begin{cases} \dot{y} = f(t, y) \\ y(t_0) = y_0 \end{cases}$$

La méthode (RK1) utilise l'équation

$$y_{n+1} = y_n + hf(t_n, y_n)$$

où h est le pas de l'itération. Le problème s'écrit donc :

$$\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array}$$

FIG. 1.2 – RK 1

Méthode de Runge-kutta d'ordre 2 (RK2) : La méthode (RK2) du point milieu est une composition de la méthode d'Euler :

$$y_{n+1} = y_n + hf\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}f(t_n, y_n)\right)$$

où h est le pas de l'itération ; Elle consiste à estimer la dérivée au milieu du pas d'intégration :

$$\begin{aligned} y_{n+\frac{1}{2}} &= y_n + \frac{h}{2}f(t_n, y_n) \\ \dot{y}_{n+\frac{1}{2}} &= f\left(t_n + \frac{h}{2}, y_{n+\frac{1}{2}}\right) \end{aligned}$$

et à refaire le pas d'intégration complet à partir de cette estimation : $y_{n+1} = y_n + h\dot{y}_{n+\frac{1}{2}}$. Ce schéma est couramment appelé **schéma prédicteur-correcteur explicite**. C'est le cas particulier pour $\alpha = 1/2$ de la méthode plus générale :

$$\begin{array}{c|cc} 0 & 0 & 0 \\ \alpha & \alpha & 0 \\ \hline & 1 - \frac{1}{2\alpha} & \frac{1}{2\alpha} \end{array}$$

FIG. 1.3 – schéma prédicteur-correcteur explicite.

On reconnaît ainsi que la méthode de quadrature utilisée pour les temps intermédiaires est celle du point milieu. C'est une méthode d'ordre 2 car l'erreur est de l'ordre de h^3 . Un autre cas courant est la méthode de **Heun**, correspondant au cas $\alpha = 1$. La méthode de quadrature repose sur la méthode des **trapèzes**.

La méthode de Runge-Kutta classique d'ordre quatre (RK4) : C'est un cas particulier d'usage très fréquent, noté *RK4*. Considérons le problème suivant :

$$\begin{cases} \dot{y} = f(t, y) \\ y(t_0) = y_0 \end{cases}$$

La méthode *RK4* est donnée par l'équation :

$$y_{n+1} = y_n + \frac{h}{6} (k_1 + 2k_2 + 2k_3 + k_4)$$

où :

$$\begin{cases} k_1 = f(t_n, y_n) \\ k_2 = f(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_1) \\ k_3 = f(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_2) \\ k_4 = f(t_n + h, y_n + hk_3) \end{cases}$$

L'idée est que la valeur suivante (y_{n+1}) est approchée par la somme de la valeur actuelle (y_n) et du produit de la taille de l'intervalle (h) par la pente estimée. La pente est obtenue par une moyenne pondérée de pentes :

- k_1 est la pente au début de l'intervalle ;
- k_2 est la pente au milieu de l'intervalle, en utilisant la pente k_1 pour calculer la valeur de y au point $t_n + \frac{h}{2}$ par le biais de la méthode d'Euler ;
- k_3 est de nouveau la pente au milieu de l'intervalle, mais obtenue cette fois en utilisant la pente k_2 pour calculer y ;
- k_4 est la pente à la fin de l'intervalle, avec la valeur de y calculée en utilisant k_3 .

Dans la moyenne des quatre pentes, un poids plus grand est donné aux pentes au point milieu.

0	0	0	0	0
$\frac{1}{2}$	$\frac{1}{2}$	0	0	0
$\frac{1}{2}$	0	$\frac{1}{2}$	0	0
1	0	0	1	0
	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$

FIG. 1.4 – schéma de RK-4

La méthode *RK4* est une méthode d'ordre 4, ce qui signifie que l'erreur commise à chaque étape est de l'ordre de h^5 , alors que l'erreur totale accumulée est de l'ordre de h^4 . Ces formules sont aussi valables pour des fonctions à valeurs vectorielles.

Chapitre 2

Optimisation Numérique par les Algorithmes Méta-heuristiques

L'optimisation est essentiellement utilisée pour améliorer les performances et l'efficacité et pour réduire les coûts. Les ressources rares, telles que le temps et l'argent, doivent être déployées correctement et efficacement dans des applications réelles telles que la fabrication, la conception technique, la planification des activités et le développement de logiciels. Il est donc essentiel de trouver des solutions pour utiliser de manière optimale ces précieuses ressources sous diverses contraintes.

2.1 Algorithmes Méta-heuristiques

Mathématiquement, l'optimisation consiste à étudier des méthodes efficaces de planification et de résolution de problèmes de conception à l'aide d'outils mathématiques.

2.1.1 Définition d'algorithmes Méta-heuristiques

Une méta-heuristique est une méthode algorithmique capable de guider et d'orienter le processus de recherche dans un espace de solution, souvent très grand à des régions riches en solutions optimales. Le fait de rendre cette méthode abstraite et plus générique conduit à une vaste utilisation pour des champs d'application différentes.

A ces applications, les métaheuristiques permettent de trouver des solutions peut-être pas toujours optimales, en tout cas très proches de l'optimum et en un temps raisonnable.

En d'autres termes, une métaheuristique peut être considérée comme un cadre algorithmique général qui peut être appliqué à différents problèmes d'optimisation avec relativement peu de modifications pour les adapter à un problème spécifique. Il n'existe pas de définition qui fasse l'unanimité, mais tous s'accordent sur les points suivants :

1. Les métaheurstiques sont des stratégies permettant de guider la recherche d'une solution optimale.
2. Le but visé par les métaheurstiques est d'explorer l'espace de recherche efficacement afin de déterminer des solutions (presque) optimales.
3. Les métaheurstiques sont en général non-déterministes et ne donnent aucune garantie d'optimalité.
4. Les métaheurstiques peuvent contenir des mécanismes qui permettent d'éviter d'être bloqué dans des régions de l'espace de recherche.
5. Les concepts de base des métaheurstiques peuvent être décrit de manière abstraite, sans faire appel à un problème spécifique.

2.1.2 Catégories des Algorithmes Méta-heuristiques

Les algorithmes métaheurstiques sont classés en deux dominants classes : **techniques évolutives** et **intelligence de l'essaim**.

Les algorithmes évolutionnaires : imitent les concepts d'évolution dans la nature. L'algorithme le meilleur et le plus apprécié de cette classe est l' Algorithme Génétique (*GA*). Cet algorithme simule les concepts de la théorie darwinienne de l'évolution. En *GA*, l'optimisation est initiée avec un ensemble de solutions aléatoires pour un problème particulier.

Après avoir évalué les solutions par la fonction objective, il modifie les variables des solutions basées sur leur valeur de fitness, les meilleures personnes ont une plus grande probabilité d'impliquer dans l'amélioration d'autres solutions, les solutions initiales sont très probables être amélioré.

Les techniques d'intelligence en essaim : ces algorithmes sont fondés sur le comportement collectif d'un groupe de créatures. Par exemple, il est possible de garantir collectivement la survie d'une colonie sans avoir une unité de contrôle centralisée.

Algorithm	Inspiration
Algorithme génétique	Évolution darwinienne dans la nature
Recuit simulé	Processus de recuit des matériaux
Optimisation des colonies de fourmis	Comportement des fourmis en quête de nourriture
Algorithme d'abeille	Comportement des abeilles
Optimisation des essaims de particules	Comportement en essaim d'oiseaux et de poissons
Tabou recherche	Mémoire humaine
Recherche d'harmonie	Performance musicale
Algorithme Firefly	Clignotant caractéristique des lucioles
Recherche de coucou	Comportement parasite des espèces de coucous
Algorithme de chauve-souris	Echolocalisation caractéristique des microbes
Pollinisation des fleurs	Pollinisation des plantes à fleurs

TAB. 2.1 – Quelques Algorithmes avec leurs inspiration

Au cours des dernières décennies, différentes *MA* ont été proposées pour résoudre des problèmes d'optimisation complexes. Quelques exemples sont cités selon le tableau 2.1 avec leur système d'inspiration.

2.1.3 Avantages d'utiliser les algorithmes méta-heuristiques

Les techniques d'optimisation traditionnelles imposent de nombreuses limites lors de l'élaboration des modèles de recherche mathématique et opérationnelle. Les solutions modèles des algorithmes d'optimisation traditionnels dépendent principalement du type de fonction d'objectif et de contrainte.

L'efficacité des algorithmes traditionnels dépend de la taille de la solution, du nombre de variables et des contraintes utilisées pour résoudre les problèmes. De plus, les algorithmes traditionnels ne suggèrent pas de stratégies de solutions générales qui seraient utiles pour résoudre le problème avec différentes variables et contraintes.

Les chercheurs explorent de nouvelles méthodes de résolution des problèmes d'optimisation, dues à l'inefficacité des algorithmes d'optimisation traditionnels, en particulier pour résoudre des problèmes complexes et de grande taille. La popularité des algorithmes inspirés par la nature est principalement influencée par :

- La capacité des systèmes biologiques à s'adapter efficacement à un environnement fréquemment changeant.
- Les informations sur les essaims ont suscité un grand intérêt pour de nombreux chercheurs au

cours de la dernière décennie.

2.1.4 Limitations des Méta-heuristiques

La capacité des métaheuristiques se développe en réalité rapidement et est étendue. Ils offrent une option pour résoudre des problèmes complexes. Cela étant dit, les métaheuristiques ont encore quelques limitations. Ceux-ci sont :

- Comportement : Ils sont difficiles à prédire le comportement à partir des règles individuelles.
- Connaissances : Les fonctions de la colonie ne pouvaient pas être comprises avec les informations de travail d'un agent.
- Sensibilité : Même un petit changement dans les résultats des principes de base dans la conduite à différents niveaux de rassemblement.
- Action : le comportement de l'agent ressemble à une clameur car l'activité de décision est stochastique.

2.2 Bat Algorithme

Les algorithmes métaheuristiques tels que l'optimisation des essaims de particules, algorithme de luciole et la recherche d'harmonie devient maintenant des méthodes puissantes pour résoudre de nombreux problèmes d'optimisation. Dans ce chapitre, nous proposons une nouvelle méthode métaheuristique, l'algorithme de chauve-souris, basé sur le comportement d'écholocation des chauves-souris. Nous avons également l'intention de combiner les avantages des algorithmes existants dans le nouvel algorithme de chauve-souris. Après une formulation détaillée et explication de sa mise en œuvre, nous comparerons ensuite l'algorithme avec d'autres algorithmes existants, y compris des algorithmes génétiques et un essaim de particules optimisation. Les simulations montrent que l'algorithme proposé semble bien supérieur à d'autres algorithmes et d'autres études sont également discutés.

2.2.1 Algorithme de chauve-souris (BA)

L'algorithme de chauve-souris a été introduit par Yang en 2010 ([?] 2010). Il simule le comportement d'écholocation des microbats en tant que microbats peuvent générer une écholocation élevée. On peut résumer cet avantage algorithme en :

Mouvement des chauves-souris virtuelles

Dans les simulations, nous utilisons naturellement des chauves-souris virtuelles. Nous devons définir les règles de mise à jour de leurs positions X_i et vitesses V_i dans un espace de recherche dimensionnel. Les nouvelles solutions X_i^t et les vitesses V_i^t au pas de temps sont données. Le mouvement d'une chauve-souris est modélisé comme suit :

$$f_i = f_{\min} + (f_{\max} - f_{\min})\beta, \quad (2.1)$$

$$V_i^t = V_i^{t-1} + (X_i^{t-1} - X_*)f_i, \quad (2.2)$$

$$X_i^t = X_i^{t-1} + V_i^t, \quad (2.3)$$

où $\beta \in [0, 1]$ est un vecteur aléatoire tiré d'une distribution uniforme. Ici X_* est le meilleur emplacement global actuel (solution) qui se trouve après avoir comparé toutes les solutions parmi tous les n chauves-souris. Pour la partie recherche locale, une fois qu'une solution est sélectionnée parmi les meilleures solutions actuelles, une nouvelle solution pour chaque chauve-souris est générée localement en utilisant la marche aléatoire :

$$X_{new} = X_{old} + \epsilon A^t, \quad (2.4)$$

où $\epsilon \in [-1, 1]$ est un nombre aléatoire, tandis que $A^t = \langle A_i^t \rangle$ est le volume moyen de toutes les chauves-souris à ce pas de temps.

Bruit et émission d'impulsions

Afin de fournir un mécanisme efficace pour contrôler l'exploration et l'exploitation et passer en phase d'exploitation si nécessaire, nous devons faire varier la sonie A_i et le taux r_i d'émission des impulsions au cours des itérations. Étant donné que le volume sonore diminue généralement une fois qu'une chauve-souris a trouvé sa proie, tandis que le taux d'émission d'impulsions augmente, le volume sonore peut être choisi comme n'importe quelle valeur de commodité, entre A_{\min} et A_{\max} , en supposant que $A_{\min} = 0$ signifie qu'une chauve-souris vient de trouver la proie et a temporairement cessé d'émettre un son. Avec ces hypothèses, on a :

$$A_i^{t+1} = \alpha A_i^t, \quad (2.5)$$

et

$$r_i^{t+1} = r_i^0 [1 - \exp(-\gamma t)], \quad (2.6)$$

où α et γ sont des constantes. En essence, α est similaire au facteur de refroidissement d'un programme de refroidissement dans un recuit simulé ([7]). Pour toute $0 < \alpha < 1$ et $\gamma > 0$, on a :

$$A_i^t \rightarrow 0, \quad (2.7)$$

et

$$r_i^t \rightarrow r_i^0, \text{ comme } t \rightarrow \infty. \quad (2.8)$$

Dans le cas le plus simple, on peut utiliser $\alpha = \gamma$.

2.2.2 Écholocalisation des chauves-souris

Comportement des microbats

Il existe environ 1000 espèces différentes de chauves-souris ([3]). Leurs tailles peuvent varier considérablement, allant des minuscules chauves-souris bourdons (d'environ 1.5 à 2g) aux chauves-souris géantes d'une envergure d'environ 2m et pouvant peser jusqu'à environ 1kg. La plupart des chauves-souris utilisent l'écholocalisation dans une certaine mesure; parmi toutes les espèces, les microbats utilisent largement l'écholocation, alors que les mégabats ne le font pas. Les microbats utilisent généralement un type de sonar, appelé, écholocation, détecter une proie, éviter les obstacles, et localiser leurs crevasses perchées dans l'obscurité. Ils peuvent émettre une impulsion sonore très forte et écouter l'écho qui rebondit sur les objets environnants ([8]). Leurs légumineuses varient en propriétés et peuvent être corrélées à leurs stratégies de chasse, selon les espèces. La plupart des chauves-souris utilisent court, signaux modulés en fréquence pour parcourir environ une octave.

Acoustique de l'écholocalisation

Mais chaque impulsion dure quelques millièmes de seconde (jusqu'à environ 8 à 10ms) dans la gamme de fréquences de 25kHz à 150kHz. Typiquement, les microbats peuvent émettre environ

10 à 20 de ces salves sonores chaque seconde, et le taux d'émission peut être accéléré jusqu'à environ 200 impulsions par seconde lors du repérage sur leur proie. Puisque la vitesse du son dans l'air est d'environ $v = 340m/s$, la longueur d'onde λ des sursauts ultrasonores de fréquence constante f est donnée par :

$$\lambda = v/f, \quad (2.9)$$

qui est dans la gamme de $2mm$ à $14mm$ pour la gamme de fréquence typique de $25kHz$ à $150kHz$.



FIG. 2.1 – Comportement d'écholocation des chauves-souris.

De façon intéressante, ces longueurs d'onde sont dans le même ordre que la taille de leurs proies. Bien qu'en réalité des microbats puissent également utiliser le délai entre leurs oreilles et les variations de volume pour détecter un environnement tridimensionnel, nous nous intéressons principalement à certaines caractéristiques de l'écholocation afin de pouvoir les relier à la fonction objective d'un problème d'optimisation, ce qui permet de formuler un algorithme intelligent de chauve-souris (BA) (2.1).

Pseudo code de BA

Fonction objective $f(X), X = (x_1, \dots, x_d)^T$

Initialiser la population de chauves-souris $X_i (i = 1, 2, \dots, n)$ et V_i

Définir la fréquence d'impulsion f_i à X_i

Initialiser les fréquences d'impulsions r_i et le volume A_i

Tandis que ($t < \text{Nombre max d'itérations}$)

Générer de nouvelles solutions en ajustant la fréquence

et mise à jour des vitesses et des emplacements/solutions [équations(2) to (4)]

Si ($\text{rand} > r_i$)

Sélectionnez une solution parmi les meilleures solutions

Générer une solution locale autour de la meilleure solution sélectionnée

Fin Si

Générer une nouvelle solution en volant au hasard

Si ($\text{rand} < A_i \ \& \ f(X_i) < f(X_*)$)

Acceptez les nouvelles solutions

Augmenter r_i et réduire A_i

Fin Si

Classez les chauves-souris et trouvez les meilleures du moment X_*

Fin Pendant que

Résultats et visualisation du post-traitement

Chapitre 3

Optimisation d'un (PVI) par (BA)

Cet chapitre illustre une méthode efficace pour résoudre des Problèmes de Valeur Initiale (PVI) dans Equations Différentielles Ordinaires (EDO). Il est basé sur l'algorithme inspiré de chauve-souris ou Bat Algorithm (BA), qui imite le système de navigation à écholocations pour détecter et poursuivre leurs proies, créé par Xin-She Yang en 2010 ([?, ?]). Nous nous concentrons sur ses applications dans le domaine de l'électronique et de l'électrotechnique lorsque nous résolvons les PVI résultant d'un circuit RL constitué d'une résistance et d'un inducteur connectés en série dans le cas d'une tension constante. Les résultats expérimentaux via un exemple numérique dont la comparaison entre la solution exacte et les résultats de l'algorithme ont montré que l'algorithme BA donne une approximation précise et satisfaisante des solutions qui confirment l'utilité et l'efficacité de la méthode proposée.

3.1 Formulation du problème

Soit $f = f(x, y)$ une fonction à valeur réelle de deux variables réelles définies pour $a \leq x \leq b$ où a et b sont finis, et pour toutes les valeurs réelles de y . Les équations

$$\begin{cases} \dot{y} = f(x, y) \\ y(a) = y_0 \end{cases}, \quad (3.1)$$

sont appelés PVI du premier ordre; elles représentent le problème suivant : pour trouver une fonction $y(x)$, continue et dérivable pour $x \in [a, b]$ telque $\dot{y} = f(x, y)$ de $y(a) = y_0$ pour tout $x \in [a, b]$ ([5]). Ce problème a une solution unique si : f est continue sur $[a, b] \times \mathbb{R}$, et satisfait

la condition de lipchitz; il existe une constante réelle $k > 0$, comme $|f(x, \theta_1) - f(x, \theta_2)| \leq k|\theta_1 - \theta_2|$, pour tous les $x \in [a, b]$ et tous les couple $(\theta_1, \theta_2) \in \mathbb{R} \times \mathbb{R}$.

Trouver numériquement les solutions optimales d'un PVI est obtenu avec des approximations : $y(x_0 + h), \dots, y(x_0 + nh)$ où $a = x_0$ et $h = (b - a) / n + 1$ pour plus de précision de la solution, nous devons utiliser une très petite taille de pas h qui comprend un plus grand nombre de pas, donc plus de temps de calcul qui n'est pas disponible dans les méthodes numériques utiles comme les méthodes Euler et Runge-Kutta ([5]) qui peut approcher des solutions de PVI et peut-être fournir des informations utiles, souvent suffisantes en l'absence de solutions analytiques exactes.

3.1.1 Fonction objectif

L'idée principale dans la formulation de la fonction objectif est d'utiliser la formule des différences finies pour la dérivée et l'équation [3.1] que nous obtenons,

$$\frac{y(x_j) - y(x_{j-1})}{h} \approx f(x_{j-1}, y(x_{j-1})).$$

Ainsi,

$$\frac{y_j - y_{j-1}}{h} \approx f(x_{j-1}, y_{j-1}).$$

par conséquent, nous devons considérer la formule d'erreur :

$$\left[\frac{y_j - y_{j-1}}{h} - f(x_{j-1}, y_{j-1}) \right]^2$$

La fonction objectif, associée à $Y = (y_1, y_2, \dots, y_d)$ sera :

$$F(y) = \sum_{i=1}^d \left[\frac{y_j - y_{j-1}}{h} - f(x_{j-1}, y_{j-1}) \right]^2 \quad (3.2)$$

3.1.2 Consistence

Nous nous intéressons au calcul de $Y = (y_1, y_2, \dots, y_d)$ qui minimise la fonction objectif dans équation [3.2]. Nous avons la formule de Taylor d'ordre 1 :

$$y_j = y_{j-1} + h\dot{y}_{j-1} + O(h^2), \quad j = 1, \dots, d.$$

Alors,

$$\frac{y_j - y_{j-1}}{h} = \dot{y}_{j-1} + O(h)$$

Si nous soustrayons $f(x_{j-1}, y_{j-1})$ des deux côtés de la dernière équation, on obtient :

$$\frac{y_j - y_{j-1}}{h} - f(x_{j-1}, y_{j-1}) = \dot{y}_{j-1} - f(x_{j-1}, y_{j-1}) + O(h), \quad j = 1, \dots, d.$$

La dernière relation montre que la valeur finale $Y = (y_1, y_2, \dots, y_d)$ est une solution approchée de PVI, pour une petite valeur de h .

3.2 Résoudre les EDO pour une RL circuit en série

Le circuit RL représenté sur la figure 3.1 a une résistance et un inducteur connectés en série. Le voltage acoustique v est appliquée lorsque l'interrupteur est fermé ([1]).

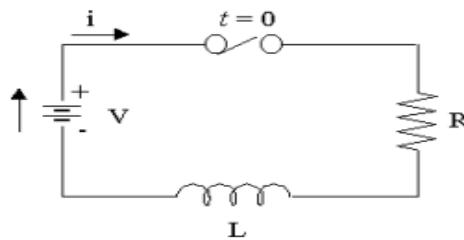


FIG. 3.1 – Une RL circuit en série.

Le voltage à travers la résistance est donnée par $V_R = Ri$. Le voltage à travers l'inducteur est donnée par $V_L = L\frac{di}{dt}$. La loi sur le voltage de kirchhoff indique que la somme dirigée des e voltages autour d'un circuit doit être nulle. Ce résultat dans l'équation différentielle suivante ([6]) :

$$Ri + L\frac{di}{dt} = V.$$

Une fois l'interrupteur fermé, le courant dans le circuit n'est pas constant. Au lieu de cela, il passera de zéro à un état stable. La solution de cette équation différentielle est :

Paramètre	Quantité
Dimension des variables de recherche (d)	10
Nombre de générations (N)	1000
Taille de la population (n)	20
Niveau sonore (constant ou décroissant) (A)	0.5
Rythme cardiaque (constant ou décroissant) (r)	0.5

TAB. 3.1 – Paramètres adoptés par BA.

$$i = \frac{V}{R} \left(1 - e^{-\left(\frac{R}{L}\right)t} \right). \quad (3.3)$$

3.3 Résultats numérique

Lors de notre étude expérimentale, le BA et ses opérateurs sont codés dans MaTLAB (R 2013a) pour la solution de PVI. Les résultats numériques sont présentés sous forme graphique et tabulaire. Le problème de traitement exigeant deux types de paramètres, le premier sont liés à BA et le second sont liés au PVI. Ces paramètres sont décrits comme suit :

3.3.1 Paramètres liés au BA

Dans ce travail, les paramètres adoptés par BA dans l'exemple traité sont résumés dans le tableau (3.1) :

3.3.2 Paramètres liés au PVI

BA est un instrument d'optimisation. Ensuite, l'équation différentielle essentielle est la conversion en forme de discrétisation. La formule de différence en arrière est utilisée pour convertir l'équation différentielle en forme de discrétisation lorsque le terme dérivé est remplacé sous la forme discrétisée par un quotient de différence pour les approximations.

L'intervalle de PVI est également divisé en $(n + 1)$ sous-intervalles équidistants avec la longueur $h = (b - a)/n + 1$. Où $n = 9$ est un nombre de nœuds intérieurs. Les paramètres liés au PVI sont les suivants :

1. Le nombre de nœuds intérieurs ($n = 9$).

j	x_j	Résultats exact	Résultats BA	Résultats RK4
0	0.00	0.0000	0.0000	0.00100
1	0.14	1.0068	1.0062	1.0169
2	0.28	1.5068	1.5060	1.5180
3	0.42	1.7551	1.7542	1.7670
4	0.56	1.8784	1.8774	1.8905
5	0.70	1.9396	1.9384	1.9528
6	0.84	1.9700	1.9684	1.9846
7	0.98	1.9851	1.9837	2.0005
8	1.12	1.9926	1.9909	2.0083
9	1.26	1.9963	1.9945	2.0131
10	1.40	1.9982	1.9960	2.0154

TAB. 3.2 – Résultats numérique de l'exemple pour $d=10$.

2. La taille de pas $h = 0.2$.
3. La condition initiale est $i = 0$ pour $t = 0$ et l'intervalle entre lequel l'équation différentielle est résolue est $t > 0$.
4. La fonction objectif

$$\begin{aligned}
 F(y_1, y_2, \dots, y_{10}) &= \sum_{j=1}^{10} \left[\frac{y_j - y_{j-1}}{h} - f(x_{j-1}, y_{j-1}) \right]^2 \\
 &= \sum_{j=1}^{10} \left[\frac{y_j - y_{j-1}}{h} - y_{j-1} \right]^2 .
 \end{aligned}$$

3.3.3 Application

Une RL circuiten série avec $R = 50\Omega$ et $L = 10H$ a un voltage constante $V = 100V$ appliquée à $t = 0$ par la fermeture d'un interrupteur. Nous voulons trouver l'expression du courant électrique dans ce cas. Nous pouvons utiliser la formule (3.3) ici uniquement parce que le voltage est constant. Cette formule ne fonctionnera pas avec une source de voltage variable. Nous avons la formule générale suivante :

$$i = 2(1 - e^{-5t}) \tag{3.4}$$

Lors du tracé de l'équation (3.4), le graphique montre la période de transition pendant laquelle le courant ajuste de sa valeur initiale de zéro à la valeur finale $\frac{V}{R}$, qui est l'état d'équilibre (Horowitz et Hill 1989). La constante de temps (TC), connu sous le nom de τ , de la fonction est le temps à $\frac{R}{L}$ est l'unité ($= 1$). Ainsi pour le transitoire RL, la constante de temps est $\tau = \frac{L}{R}$ secondes dans cet exemple, la constante de temps TC est de 0.2 secondes.

La figure (3.2) montre la représentation graphique des résultats du tableau (3.2). La comparaison entre les représentations de BA et la méthode de RK4 aux résultats exacts confirme que BA est meilleur que RK4, car il a une courbe très étroite à la courbe exacte contrairement à la méthode RK4.

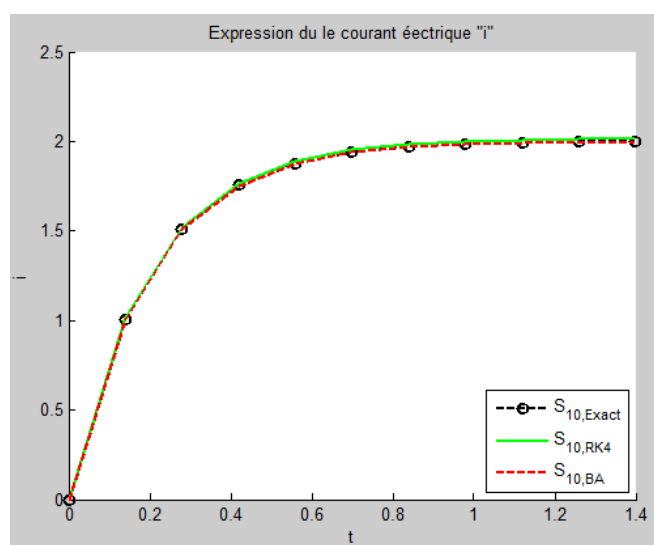


FIG. 3.2 – Résultats exacts, résultats BA et RK4.

Le tableau (3.3) présente l'erreur absolue entre les résultats exacts et les résultats des méthodes étudiées. La figure (4) montre leurs représentations graphiques. Dans les deux représentations de l'erreur absolue, la méthode BA offre une erreur absolue très négligeable par rapport à la méthode RK4.

3.4 Conclusion

Nos résultats de simulation indiquent que BA est simple, flexible et facile à manipuler, il réduit le temps en fournissant une convergence très rapide à un stade très initial en passant de l'exploration à l'exploitation. Il donne des solutions optimales prometteuses pour résoudre PVI.

j	x_j	BA	RK4
0	0.00	0.000	0.0010
1	0.14	0.0006	0.0101
2	0.28	0.0008	0.0112
3	0.42	0.0009	0.0119
4	0.56	0.0010	0.0121
5	0.70	0.0012	0.0132
6	0.84	0.0016	0.0146
7	0.98	0.0014	0.0154
8	1.12	0.0017	0.0157
9	1.26	0.0018	0.0168
10	1.40	0.0022	0.0172

TAB. 3.3 – Erreur absolue.

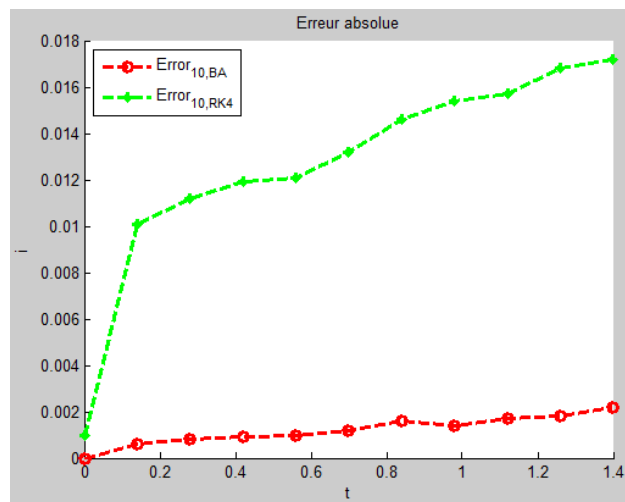


FIG. 3.3 – Erreur absolue.

Conclusion Générale

Les métaheuristiques sont souvent employées pour leur facilité de programmation et de manipulation. Elles sont en effet facilement adaptables à tout type de problème d'optimisation. Toutefois, elles sont le plus judicieusement employées sur des problèmes d'optimisation difficile, où des méthodes d'optimisation plus classiques (méthodes déterministes, notamment) montrent leurs limites.

La notion d'efficacité se rapporte généralement à deux objectifs contradictoires : la vitesse et la précision. La vitesse est souvent mesurée en nombre d'évaluations de la fonction objectif, qui est la plupart du temps la partie la plus gourmande en temps de calcul. La précision se rapporte à la distance entre l'optimum trouvé par la métaheuristique et l'optimum réel, soit du point de vue de la solution, soit de celui de la valeur. Bien souvent, un algorithme rapide est peu précis, et inversement.

Les techniques numériques pour résoudre les équations différentielles telles que les méthodes de Runge-Kutta sont entourées par la substitution de la dérivée qui se produit dans l'équation différentielle par des extensions approximatives de la fonction près des points d'intérêt. Evidemment, cette approximation n'est valable que si l'intervalle de temps dans lequel la fonction est suffisamment linéaire, si non la solution devient imprécise.

L'inexactitude de la résolution des problèmes complexes et des situations non linéaires nécessite une augmentation significative des tentatives de simulation des fonctions informatiques basées sur des méthodes plus efficaces.

Lorsque nous choisissons BA pour optimiser l' PVI dans le domaine économique, exactement l'équation de l'offre et de la demande qui est généralement résolue à l'aide des méthodes numériques classiques, dans ces notes, nous appliquons Runge-kutta ($RK4$) comme méthode numérique pour l'équation de l'offre et de la demande.

Grâce à l'exemple choisi et après avoir comparé les solutions exactes, les résultats de l'algorithme

(*BA*), et les résultats de Runge-Kutta d'ordre (*RK4*), nous avons constaté que le *BA* était plus en mesure de fournir des solutions précises avec une erreur minimale. Il est important de noter que les résultats actuels concernent principalement l'algorithme *BA* standard. Il serait utile que la recherche supplémentaire se concentre sur l'élargissement de la méthodologie proposée pour améliorer d'autres problèmes économiques par le biais d'autres variables de l'*BA*.

Bibliographie

- [1] Abdon, A., Juan Jose, N., (2015), *Numerical solution for the model of RLC circuit via the fractional derivative without singular kernel*, Advances in Mechanical Engineering, Vol. 7(10) 1–7.
- [2] Amos, G., (2007), *Matlab, an introduction with applications*, John Willey and Son.
- [3] Colin, T., (2000), *The Varienty of Life*, Oxford University Press.
- [4] Gilat, A., (2004), *Matlab : An Introduction with application*, John Wiley and Sons.
- [5] Henrici, P., (1964), *Elements of Numerical Analysis*, Mc Graw-Hill. New York.
- [6] James, W N., A Riedel ,S., (2015), *Electric Circuits*. Prentice Hall Publisher. ISBN : 13 :978-0-13-376003-3.
- [7] Kirkpatrick, S., Gelatt, C. D., and Vecchi, M.P., (1983), *Optimization by simulated annealing*. Science, 220, 671-680 .
- [8] Richardson, P., (2008). *Bats. Natural History Museum*, London.
- [9] Yang, XS, (2010) .*New Metaheuristic Bat-Inspired Algorithm*. Berlin, Heidelberg : Springer Berlin Heidelberg. ISBN 978-3-642-12538-6, p. 65{74. doi :10.1007/978-3-642-12538-6_6.

Annexe A : Le logiciel Matlab

MATLAB est un langage haute performance pour le calcul technique. Il intègre le calcul, la visualisation et la programmation dans un environnement facile à utiliser où les problèmes et les solutions sont exprimés dans une notation mathématique familière. Les utilisations typiques comprennent ([2, 4]) :

- Maths et calcul
- Développement d’algorithmes
- Modélisation, simulation et prototypage
- Analyse, exploration et visualisation des données
- Graphiques scientifiques et techniques
- développement d’applications, y compris la création d’interface utilisateur graphique

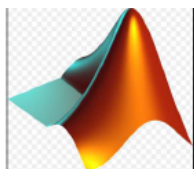


FIG. 3.4 – Icône du logiciel Matlab.

MATLAB est un système interactif dont l’élément de données de base est un tableau qui ne nécessite pas de dimensionnement. Cela vous permet de résoudre de nombreux problèmes informatiques techniques, en particulier ceux avec des formulations matricielles et vectorielles, en une fraction du temps que cela prendrait pour écrire un programme dans un langage non interactif scalaire tel que C ou Fortran. Le Nom **MATLAB** signifie laboratoire matriciel. **MATLAB** a été écrit à l’origine pour fournir un accès facile aux logiciels matriciels développés par **LINPACK** et **EISPACK**, qui représentent ensemble l’état de l’art des logiciels de calcul matriciel. **MATLAB** a évolué au fil des années avec la contribution de nombreux utilisateurs. Dans les environnements universitaires, il s’agit de l’outil pédagogique standard pour l’introduction et l’annonce

cours avancés en mathématiques, ingénierie, et la science. Dans l'industrie, **MATLAB** est l'outil de choix pour la recherche, le développement et l'analyse à haute productivité. Fonctionnalités **MATLAB** une famille de solutions spécifiques aux applications appelées boîtes à outils. Très important pour la plupart des utilisateurs de **MATLAB**, les boîtes à outils vous permettent d'apprendre et d'appliquer une technologie spécialisée. Boîtes à outils sont des collections complètes de fonctions **MATLAB** (M les) qui étendent **MATLAB** environnement pour résoudre des classes particulières de problèmes. Zones dans lesquelles des boîtes à outils sont disponibles capable d'inclure le traitement du signal, les systèmes de contrôle, les réseaux de neurones, la logique floue, les ondelettes, simulation, et bien d'autres.

Annexe B : Abréviations et Notations

Les différentes abréviations et notations utilisées durant cette thèse est représentées selon ce tableau :

Abréviations	Notations
<i>PVI</i>	Problèmes à valeur initiale.
<i>AG</i>	Algorithme génétique.
<i>BA</i>	Bat algorithme ou algorithme de chauve-souris.
<i>EDO</i>	Equation Différentielle Ordinaire.
<i>RK – 4</i>	Range Kutta d'ordre 4.
<i>AM</i>	Algorithmes Métaheuristiques.
\mathbb{R}	Ensemble des Nombres Réelles

Annexe C : Code Matlab des Méthodes Étudiées

3.5 Code Matlab de la Méthode de (RK4)

```
dydt=inline('10-5*y','t','y');
a=0;
b=1;
n=10;
h=(b-a)/n;
t=a:h: b;
epsilon=0.0001;
u(1)=1+epsilon;
for i=1:n-1
u1(i)=u(i);
u2(i)=u(i)+h/2*dydt(t(i),u1(i));
u3(i)=u(i)+h/2*dydt(t(i)+h/2,u2(i));
u4(i)=u(i)+h*dydt(t(i)+h/2,u3(i));
u(i+1)=u(i)+h/6*(dydt(t(i),u1(i)+2*dydt(t(i)+h/2,u2(i)) +2*dydt(t(i) +h/2,u3(i))
+dydt(t(i+1),u4(i))))
end
plot(t(1: end-1),u, ' o-g' )
```

3.6 Code Matlabe de BA

```
% ===== %
% Files of the Matlab programs included in the book: %
% Xin-She Yang, Nature-Inspired Metaheuristic Algorithms, %
% Second Edition, Luniver Press, (2010). www.luniver.com %
% ===== %
% ----- %
% Bat-inspired algorithm for continuous optimization (demo)%
% Programmed by Xin-She Yang @Cambridge University 2010 %
% ----- %
% Usage: bat_algorithm([20 1000 0.5 0.5]); %
% -----
% This is a simple demo version only implemented the basic %
% idea of the bat algorithm without fine-tuning the parameters, %
% Then, though this demo works very well, it is expected that %
% this demo is much less efficient than the work reported in %
% the following papers: %
% (Citation details): %
% 1) Yang X.-S., A new metaheuristic bat-inspired algorithm, %
% in: Nature Inspired Cooperative Strategies for Optimization %
% (NISCO 2010) (Eds. J. R. Gonzalez et al.), Studies in %
% Computational Intelligence, Springer, vol. 284, 65-74 (2010). %
% 2) Yang X.-S., Nature-Inspired Metaheuristic Algorithms, %
% Second Edition, Luniver Presss, Frome, UK. (2010). %
% 3) Yang X.-S. and Gandomi A. H., Bat algorithm: A novel %
% approach for global engineering optimization, %
% Engineering Computations, Vol. 29, No. 5, pp. 464-483 (2012). %
% -----
% Main programs starts here
function [best,fmin,N_iter]=bat_algorithm(para)
% Display help
```

```
help bat_algorithm.m
% Default parameters
if nargin<1, para=[20 1000 0.5 0.5]; end
n=para(1); % Population size, typically 10 to 40
N_gen=para(2); % Number of generations
A=para(3); % Loudness (constant or decreasing)
r=para(4); % Pulse rate (constant or decreasing)
% This frequency range determines the scalings
% You should change these values if necessary
Qmin=0; % Frequency minimum
Qmax=2; % Frequency maximum
% Iteration parameters
N_iter=0; % Total number of function evaluations
% Dimension of the search variables
d=10; % Number of dimensions
%Lb=-2*ones(1,d);% Lower limit/bounds/ a vector
%Ub=2*ones(1,d);% Upper limit/bounds/ a vector
Lb=0*ones(1,d);
Ub=8*ones(1,d);
% Initializing arrays
Q=zeros(n,1); % Frequency
v=zeros(n,d); % Velocities
% Initialize the population/solutions
for i=1:n,
Sol(i,:)=Lb+(Ub-Lb).*rand(1,d);
Fitness(i)=Fun(Sol(i,:));
end
% Find the initial best solution
[fmin,I]=min(Fitness);
best=Sol(I,:);
% ===== %
% Note: As this is a demo, here we did not implement the %
```

```
% reduction of loudness and increase of emission rates. %
% Interested readers can do some parametric studies %
% and also implementation various changes of A and r etc %
% ===== %
% Start the iterations -- Bat Algorithm (essential part) %
for t=1:N_gen,
% Loop over all bats/solutions
for i=1:n,
Q(i)=Qmin+(Qmin-Qmax)*rand;
v(i,:)=v(i,:)+(Sol(i,:)-best)*Q(i);
S(i,:)=Sol(i,:)+v(i,:);
% Apply simple bounds/limits
Sol(i,:)=simplebounds(Sol(i,:),Lb,Ub);
% Pulse rate
if rand>r
% The factor 0.001 limits the step sizes of random walks
S(i,:)=best+0.001*randn(1,d);
end
% Evaluate new solutions
Fnew=Fun(S(i,:));
% Update if the solution improves, or not too loud
if (Fnew<=Fitness(i)) & (rand<A) ,
Sol(i,:)=S(i,:);Fitness(i)=Fnew;
end
% Update the current best solution
if Fnew<=fmin,
best=S(i,:);
fmin=Fnew;
end
end
N_iter=N_iter+n;
end
```

```
% Output/display
disp(['Number of evaluations: ',num2str(N_iter)]);
disp(['Best =',num2str(best),' fmin=',num2str(fmin)]);

% Application of simple limits/bounds
function s=simplebounds(s,Lb,Ub)

% Apply the lower bound vector
ns_tmp=s;
I=ns_tmp<Lb;
ns_tmp(I)=Lb(I);

% Apply the upper bound vector
J=ns_tmp>Ub;
ns_tmp(J)=Ub(J);

% Update this new move
s=ns_tmp;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Objective function: your own objective function can be written here
% Note: When you use your own function, please remember to
% change limits/bounds Lb and Ub (see lines 52 to 55)
% and the number of dimension d (see line 51).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function z=Fun(u)

% objective function related to IVP

% z = ;
```

Résumer

Ce mémoire est vu comme une introduction a l'optimisation par des algorithmes Méta-heuristiques, il s'agit d'un algorithme qui s'appelle « Bat Algorithm » noté (BA) qui imite le système de navigation d'écholocations de chauve-souris pour détecter et poursuivre leurs proies appliqué pour optimiser d'une manière numérique un Problème à Valeur Initiale (PVI). Nous avons débuté par les concepts de bases avec plus de concentration sur BA. Nous nous concentrons dans la partie pratique et à l'aide du logiciel Matlab a appliquer cet algorithme dans le domaine de l'électronique et de l'électrotechnique lorsque nous résolvons les PVI résultant d'un circuit RL constitué d'une résistance et d'un inducteur connectés en série dans le cas d'une tension constante pour trouver l'expression du courant électrique. Les résultats expérimentaux via un exemple numérique dont la comparaison entre la solution exacte et les résultats de la méthode de Range-Kutta d'ordre 4 (RK4) indiquent que BA donne une approximation précise et satisfaisante des solutions.

Phrases et Mots Clés: Problème à Valeur Initiale (PVI), Optimisation numérique, Bat Algorithm (BA), Méthode de Range Kutta d'ordre 4 (RK4)

Abstract

This thesis is seen as an introduction to optimization by Meta-heuristic algorithms, it is about an algorithm called "Bat Algorithm" noted (BA) which mimics the navigation system of bats echolocations to detect and track their prey applied to numerically optimize an Initial Value Problem (PVI). We started with the basics with more focus on BA. We concentrate in the practical chapter and by using Matlab software in applying this algorithm in the field of electronics and electrical engineering when we solve IVPs resulting from an RL circuit consisting of a resistor and an inductor connected in series in the case of a constant voltage to find the expression of the electric current. The experimental results via a numerical example whose comparison between the exact solution and the results of the Range-Kutta method of order 4 (RK4) indicate that BA gives a precise and satisfactory approximation of the solutions.

Phrases & Keywords: Initial Value Problem (PVI), Numerical optimization, Bat Algorithm (BA), Range Kutta method of order 4 (RK4).

ملخص

يُنظر إلى هذه الأطروحة على أنها مقدمة للتحسين بواسطة خوارزميات *méta-heuristique* ، فهي خوارزمية تسمى "خوارزمية الخفافيش" المشار إليها (BA) والتي تحاكي نظام الملاحة الخاص بتكاثر الخفافيش لاكتشاف وتتبع فريستها المطبقة لتحسين مشكلة القيمة الأولية (PVI) بدأنا بالأساسيات مع مزيد من التركيز على خوارزمية الخفافيش. نركز في الفصل العملي وباستخدام برنامج Matlab في تطبيق هذه الخوارزمية في مجال الإلكترونيات والهندسة الكهربائية عندما نحل مشكلة القيمة الأولية الناتجة عن دائرة RL تتكون من مقاوم ومحث متصل على التسلسل في حالة وجود جهد ثابت من أجل إيجاد عبارة التيار الكهربائي. النتائج التجريبية و من خلال مثال رقمي تشير مقارنته بين الحل الدقيق ونتائج طريقة Range-Kutta من الرتبة 4 (RK4) إلى أن BA يعطي تقديراً تقريبياً دقيقاً ومرضياً للحلول

العبارات والكلمات الرئيسية: مشكلة القيمة الأولية (PVI) ، التحسين العددي ، خوارزمية الخفافيش (BA) ، طريقة Range Kutta من الترتيب 4 (RK4) .
