

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 HALIMA SAADIA

Titre :

Introduction a L'optimisation par des Algorithmes Inspirés de la Nature

Membres du Comité d'Examen :

Mm.	OUAAR Fatima	UMKB	Encadreur
Dr.		UMKB	Examineur
Dr.		UMKB	Examineur

Juin 2019

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**" de m'avoir venu en aide pour que je peux aboutir à la réussite.

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 *****, président de jury puis par Monsieur *****, examinateur, je les exprime ma profonde gratitude.

Je remercie vivement Monsieur **HAFAIED Mokhtar**, Professeur à l'Université de Biskra, chef du département de Mathématiques, pour ses encouragements, ses suggestions et son dynamisme.

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 chers **Frères** et **Soeurs**, 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.

شكرا.

Table des matières

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 Fonction exponentielle réelle	3
1.1.1 Définitions diverses de fonction exponentielle réelle	4
1.1.2 Propriétés algébrique de la fonction expontielle	7
1.1.3 Signe de la fonction expontielle	10
1.2 Equation différentielle linéaire du premier ordre	10
1.2.1 Equation différentielle de type $y'=ay$	11
1.2.2 Equation différentielle de type $y'=a(x)y$	12
1.2.3 Equation différentielle de type $y'=a(x)y+b(x)$	12
1.3 Problèmes à valeur initiale (PVI)	14
2 Optimisation Numérique par des Algorithmes Inspirés de la Nature	17
2.1 Algorithmes Inspirés de la Nature	17

2.1.1	Définition d’algorithmes inspirés de la nature	17
2.1.2	Avantages d’utiliser les algorithmes inspirés de la nature	18
2.2	Algorithme de Pollinisation des Fleurs (APF)	19
2.2.1	Description de processus de pollinisation chez les fleurs	20
2.2.2	Formulation d’(APF)	22
2.2.3	Application d’(APF)	22
3	Etude Experimentale	25
3.1	Formulation du problème	25
3.1.1	Fonction objectif	26
3.1.2	La consistance	27
3.2	Application : Le modèle exponentielle	27
3.2.1	Paramètres relatives au (PVI)	28
3.2.2	Paramètres relatives au (APF)	28
3.3	Résultats Numériques	29
	Conclusion	32
	Abréviations et Notations	36
	Annexe A : Logiciel <i>Matlab</i>	37
	Annexe B : Code Matlab des Algorithmes	39
3.4	Code Matlab de méthode d’Euler	39
3.5	Code Matlab d’Algorithme de Pollination de Fleurs (APF)	40

Table des figures

1.1	Fonction exponentielle.	5
1.2	Le pas h	15
2.1	Auto-pollinisation.	20
2.2	Pollinisation croisée.	21
3.1	Plot des résultats exactes et résultats numériques obtenus par (APF) pour d=10.	30
3.2	Plot d'erreur absolue	31
3.3	Icône du logiciel Matlab.	37

Liste des tableaux

2.1	Quelques Algorithmes avec leurs inspiration	19
2.2	Pseudo-code d'APF	23
3.1	Paramètres adoptés (APF).	28
3.2	Résultats exactes et résultats numériques obtenus par (APF) pour $d=10$	29
3.3	Erreur absolue entre les résultats exactes et les méthodes étudiées	30
3.4	Les fonctions les plus utilisées dans ce travail	38

Introduction Générale

L'optimisation est une branche des mathématiques cherchant à modéliser, à analyser et à résoudre analytiquement ou numériquement les problèmes qui consistent à minimiser ou maximiser une fonction sur un ensemble. L'optimisation joue un rôle important en recherche opérationnelle, dans les mathématiques appliquées, en analyse et en analyse numérique, en statistique pour l'estimation du maximum de vraisemblance d'une distribution, pour la recherche de stratégies dans le cadre de la théorie des jeux, ou encore en théorie du contrôle et de la commande. 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 difficile pour lesquels on ne connaît pas de méthode classique plus efficace. Les métaheuristicques 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 métaheuristicques 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.

Récemment un nouvel algorithme métaheuristique inspiré par la nature, nommé l'algorithme de pollinisation de fleur (*APF*), est développé par des scientifiques dont l'inspiration de la nature est une caractéristique de nombreuses métaheuristiques traditionnelles ou nouvellement développées. Parmi les exemples de mécanismes imités par de récents algorithmes heuristiques non conventionnels, on peut nommer, par exemple, communication bioluminescente des incendies, systèmes de coopération de l'araignée sociale.

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 polination des fleurs (*APF*)

Notre objectif est d'évaluer les performances d'(*APF*) 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 d'(*APF*) et décrivent sa variante à plusieurs objectifs

Ce mémoire est organisé de la manière suivante :

Chapitre 1 : Nous décrivons brièvement la fonction exponentielle dont leurs définitions et leurs propriétés principales en plus les types d'équations différentielles linéaires d'ordre 1 puis nous avons traité la méthode d'Euler.

Chapitre 2 : Fournit une base sur laquelle l'algorithme de pollinisation des fleurs a été construit avec sa description formelle.

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 (*APF*) sont comparés avec celles de méthode d'Euler qui est considérée comme une méthode traditionnelle pour résoudre les (*PVI*). La performance de l'algorithme en relation avec la méthode d'Euler est également étudiée. Enfin, la dernière partie de ce chapitre contient des remarques finales sur le (*APF*) et des propositions pour ses applications ultérieures.

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. En sélectionnant un exemple spécifié est choisi qui est considré comme un problème exponentiel qui joue un rôle impératif pour décrire de nombreux problèmes réels.

1.1 Fonction exponentielle réelle

En mathématiques, la fonction exponentielle est la fonction notée \exp qui est sa propre dérivée et qui prend la valeur 1 en 0. Elle est utilisée pour modéliser des phénomènes dans lesquels une différence constante sur la variable conduit à un rapport constant sur les images. Ces phénomènes sont en croissance dite "*exponentielle*".

1.1.1 Définitions diverses de fonction exponentielle réelle

Il existe plusieurs points d'entrée possibles pour la définition de la fonction exponentielle : par la propriété de sa dérivée (la dérivée est égale à la fonction), par ses propriétés algébriques (elle transforme une somme en produit), ou par son développement en série.

Définition 1.1.1 *Soit k un nombre réel strictement positif. L'unique solution de l'équation $e^x = k$ est appelée logarithme de k et sera notée $\ln(k)$ ou $\ln e^x = k \iff x = \ln(k)$.*

A l'oral "ln" se lit comme "Hélène". Ainsi

$$e^{\ln(k)} = k > 0.$$

Ce complément permettra d'étudier ici quelques équations de la forme $e^x = k$.

- La fonction exponentielle transforme les sommes en produits, c'est-à-dire que pour tous réels x et y , $e^{x+y} = e^x e^y$. On en déduit que pour tout réel x et tout rationnel b , $(e^x)^b = e^{bx}$. Pour b irrationnel, cette équation peut tenir lieu de définition, c'est-à-dire que l'une des façons de définir l'exponentielle de base a est de poser, pour tous réels $a > 0$ et b :

$$a^b = e^{b \ln(a)}.$$

Définition 1.1.2 *On note \exp la valeur de cette fonction en 1. Ce nombre e qui vaut approximativement 2,71828 s'appelle la base de la fonction exponentielle et permet une autre notation de la fonction exponentielle :*

$$\forall x \in \mathbb{R}, \quad \exp(x) = e^x.$$

La fonction exponentielle est la seule fonction continue sur \mathbb{R} qui transforme une somme en produit et qui prend la valeur e en 1. C'est un cas particulier des fonctions de ce type appelées exponentielles de base a . On peut la déterminer comme limite de suite ou à l'aide d'une série entière. C'est la bijection réciproque de la fonction logarithme népérien. Ces diverses définitions permettent d'étendre la définition de la fonction exponentielle à des

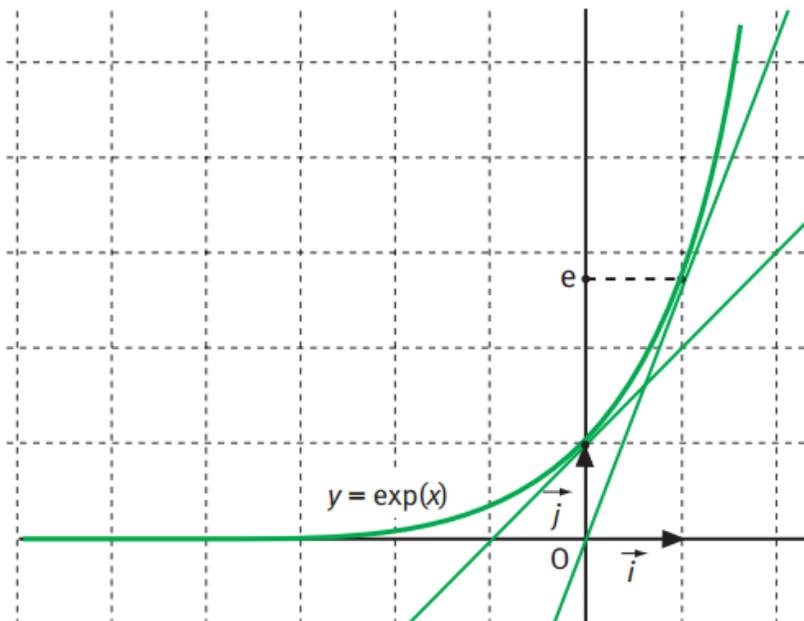


FIG. 1.1 – Fonction exponentielle.

fonctions de \mathbb{C} vers \mathbb{C}^* ou même à des espaces plus compliqués

Définition 1.1.3 *On appelle aussi parfois fonction exponentielle toute fonction dont l'expression est de la forme $f(x) = Ae^{\lambda x}$.*

Théorème 1.1.1 *Il existe une et une seule fonction f définie et dérivable sur \mathbb{R} telle que :*

$$f' = f, \quad f(0) = 1.$$

Preuve. L'existence d'une telle fonction est admise. Démontrons qu'il existe une seule fonction vérifiant ces conditions pour cela, on montre d'abord qu'une telle fonction ne peut pas s'annuler.

Soit donc f une fonction telle que $f' = f$ et $f(0) = 1$. On considère la fonction h définie sur \mathbb{R} par $h(x) = f(x) \times f(-x)$. La fonction h est le produit de deux fonctions dérivables

sur \mathbb{R} donc h est aussi dérivable sur \mathbb{R} et, pour tout réel x on a :

$$\begin{aligned} \acute{h}(x) &= \acute{f}(x) \times f(-x) + f(x) \times \left(-\acute{f}(-x)\right) \\ &= \acute{f}(x) \times f(-x) - f(x) \times \acute{f}(-x). \end{aligned}$$

Comme $\acute{f} = f$ on obtient :

$$\acute{h}(x) = f(x) \times f(-x) - f(x) \times f(-x) = 0.$$

La dérivée de la fonction h est nulle pour tout réel x donc la fonction h est une fonction constante sur \mathbb{R} pour déterminer sa valeur, on utilise la condition $f(0) = 1$. Ce qui donne :

Pour tout réel x

$$h(x) = h(0) = f(0) = f(-0) = 1.$$

Donc, pour tout réel x , le produit $f(x) \times f(-x)$ est égal à 1, il ne s'annule pas donc $f(x)$ n'est jamais nul.

Pour montrer l'unicité de la fonction f nous allons considérer deux fonctions, f et g , vérifiant les conditions $\acute{f} = f$ et $f(0) = 1$, et nous allons montrer que ces deux fonctions sont nécessairement égales. On définit sur \mathbb{R} la fonction k en posant $k(x) = \frac{g(x)}{f(x)}$ (ce qui est possible puisque la fonction f ne s'annule pas). La fonction k est dérivable sur \mathbb{R} alors :

$$\acute{k} = \frac{\acute{g}f - g\acute{f}}{f^2} = \frac{gf - gf}{f^2} = 0,$$

car $\acute{f} = f$ et $\acute{g} = g$. La fonction k est donc constante sur \mathbb{R} et, pour tout réel x , on a :

$$k(x) = k(0) = \frac{g(0)}{f(0)} = \frac{1}{1} = 1.$$

Donc, pour tout réel x , $\frac{g(x)}{f(x)} = 1$, soit $f(x) = g(x)$. Les deux fonction f et g sont donc égales ce qui prouve l'unicité de la fonction du théorème. ■

Définition 1.1.4 *L'unique fonction f définie et dérivable sur \mathbb{R} et telle que : $f' = f$ et f est appelée fonction exponentielle. On la note \exp .*

1.1.2 Propriétés algébrique de la fonction exponentielle

1. Pour tous réels a et b ,

$$\exp(a + b) = \exp(a) \exp(b).$$

Preuve. Le réel b étant fixé, on définit sur \mathbb{R} la fonction f en posant :

$$f(x) = \frac{\exp(x + b)}{\exp(b)}.$$

La fonction f est dérivable sur \mathbb{R} et on obtient :

$$f'(x) = \frac{1 \times (\exp)'(x + b)}{\exp(b)} = \frac{\exp(x + b)}{\exp(b)} = f(x) \text{ (} b \text{ est fixé et } \exp' = \exp \text{)}.$$

Comme $f(0) = \frac{\exp(0 + b)}{\exp(b)} = 1$, la fonction f vérifie les deux conditions de théorème

1, il s'agit donc de la fonction exponentielle. On en déduit que pour tout réel x :

$$\exp(x) = \frac{\exp(x + b)}{\exp(b)} \quad \text{d'où} \quad \exp(x + b) = \exp(x) \exp(b),$$

et en posant $x = a$:

$$\exp(a + b) = \exp(a) \exp(b).$$

Pour tout réel a , $\exp(a + a) = \exp(a) \exp(a)$ d'où $\exp(2a) = (\exp(a))^2$. ■

2. Pour tout réel x , $\exp(-x) = \frac{1}{\exp(x)}$.

3. $(\exp)' = \exp$ et $\exp(0) = 1$, pour tout réel x , $\exp(x) \neq 0$.

4. Pour tout réels a et b , $\exp(a - b) = \frac{\exp(a)}{\exp(b)}$

Preuve. On écrit $a - b = a + (-b)$ et on a :

$$\exp(a + (-b)) = \exp(a) \exp(-b) = \exp(a) \frac{1}{\exp(b)} = \frac{\exp(a)}{\exp(b)}.$$

■

5. Pour tout réel a et pour tout entier n de \mathbb{Z} : $\exp(na) = (\exp(a))^n$.

Preuve. Soit a un nombre réel et n un entier naturel, on veut montrer que :

$$\exp(na) = (\exp(a))^n.$$

On sait que ceci est vrai pour $n = 0$ ($\exp(0 \times a) = 1 = \exp(a)^0$), pour $n = 1$ ($\exp(a) = (\exp(a))^1$). Montrons par récurrence que l'égalité " $\exp(na) = (\exp(a))^n$ " est vraie pour tout entier n , $n \geq 1$.

Initialisation : L'égalité est vraie pour $n = 1$.

Hérédité : Soit k un entier naturel $k \geq 1$, pour lequel on suppose que $\exp(ka) = (\exp(a))^k$. On a alors :

$$\exp((k+1)a) = \exp(ka + a) = \exp(ka) \times \exp(a),$$

en appliquant la propriété 1 avec $b = ka$ puis, en utilisant l'hypothèse de récurrence, on obtient : $\exp((k+1)a) = (\exp(a))^k \times \exp(a) = (\exp(a))^{k+1}$. La proposition est donc héréditaire.

Conclusion : Pour tout $n \geq 1$ $\exp(na) = (\exp(a))^n$. Finalement, pour tout entier n de \mathbb{N} :

$$\exp(na) = (\exp(a))^n.$$

Si n est un entier naturel, on a aussi :

$$\exp(-na) = \frac{1}{\exp(na)} = \frac{1}{(\exp(a))^n} = (\exp(a))^{-n}.$$

La propriété est donc prouvée pour tout n de \mathbb{Z} ■

6. Pour tout réel x , $\exp(x)$ est strictement positif : $\exp(x) > 0$.
7. La fonction exponentielle est strictement croissante sur \mathbb{R} . On peut montrer que la fonction exponentielle dépasse toute fonction polynôme (à condition de choisir x suffisamment grand). C'est pourquoi, dans la vie courante, on utilise l'expression « croissance exponentielle » pour évoquer un phénomène dont la croissance est très forte.
8. Variations et limites : On a vu que la fonction exponentielle est à valeurs strictement positives et une des caractéristiques de la fonction exponentielle est d'être égale à sa fonction dérivée, on en déduit donc :

$$\lim_{x \rightarrow -\infty} \exp(x) = 0, \quad \lim_{x \rightarrow +\infty} \exp(x) = +\infty.$$

Autres limites :

$$\lim_{x \rightarrow +\infty} \frac{\exp(x)}{x} = +\infty; \quad \lim_{x \rightarrow +\infty} \frac{\exp(x) - 1}{x} = 1; \quad \lim_{x \rightarrow -\infty} x \exp(x) = 0.$$

On peut retenir les deux limites :

$$\lim_{x \rightarrow +\infty} \frac{e^x}{x} = +\infty; \quad \lim_{x \rightarrow -\infty} x e^x = 0.$$

en remarquant que, pour ces deux formes indéterminées, c'est "l'exponentielle qui l'a emporté sur x ".

9. Équations, inéquations : Pour tous réels a et b , on a :

$$e^a = e^b \iff a = b, \quad e^a < e^b \iff a < b.$$

1.1.3 Signe de la fonction exponentielle

Pour tout réel x , si on applique l'égalité $e^{a+b} = e^a \times e^b$ en posant $a = b = \frac{x}{2}$ on trouve

$$e^x = e^{\frac{x}{2} + \frac{x}{2}} = e^{\frac{x}{2}} \times e^{\frac{x}{2}} = \left(e^{\frac{x}{2}}\right)^2.$$

Ainsi, tout nombre e^x est un carré. On rappelle que l'exponentielle ne s'annule pas, d'où la propriété suivante.

1.2 Equation différentielle linéaire du premier ordre

Une équation différentielle est une équation :

- Dont l'inconnue est une fonction (généralement notée $y(x)$ ou simplement y) ;
- Dans laquelle apparaissent certaines des dérivées de la fonction (dérivée première \dot{y} , ou dérivées d'ordres supérieurs \ddot{y} , y^3 , ...).

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

$$\dot{y} = a(x)y + b(x). \tag{1.1}$$

où a et b sont des fonctions définies sur un intervalle ouvert I de \mathbb{R} .

Dans la suite on supposera que a et b sont des fonctions continues sur I . On peut envisager la forme :

$$\alpha(x)\dot{y} + \beta(x)y = \gamma(x).$$

On demandera alors que $\alpha(x) \neq 0$ pour tout $x \in I$. La division par α permet de retrouver la forme (1.1).

On va commencer par résoudre le cas où a est une constante et $b = 0$. Puis a sera une fonction (et toujours $b = 0$). On terminera par le cas général où a et b sont deux fonctions.

1.2.1 Equation différentielle de type $y'=ay$

Théorème 1.2.1 Soit a un réel. Soit l'équation différentielle :

$$\dot{y} = ay. \tag{1.2}$$

Les solutions de (1.2), sur \mathbb{R} , sont les fonctions y définies par :

$$y(x) = ke^{ax}.$$

où $k \in \mathbb{R}$ est une constante quelconque.

Ce résultat est fondamental. Il est tout aussi fondamental de comprendre d'où vient cette formule, via une preuve rapide (mais pas tout à fait rigoureuse).

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

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

que l'on intègre à gauche et à droite pour trouver :

$$\ln |y(x)| = ax + b.$$

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

$$|y(x)| = e^{ax+b}.$$

Autrement dit $y(x) = \pm e^b e^{ax}$. En posant $k = \pm e^b$ on obtient les solutions (non nulles) cherchées. ■

– L'équation différentielle (1.2) admet donc une infinité de solutions (puisque l'on a une infinité de choix de la constante k).

- La constante k 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.
- Le théorème (1.2.1) peut aussi s'interpréter ainsi : si y_0 est une solution non identiquement nulle de l'équation différentielle (1.2) 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.

1.2.2 Equation différentielle de type $y' = a(x)y$

Le théorème suivant affirme que, lorsque a est une fonction, résoudre l'équation différentielle $\dot{y} = a(x)y$ revient à déterminer une primitive A de a (ce qui n'est pas toujours possible explicitement).

Théorème 1.2.2 *Soit $a : I \rightarrow \mathbb{R}$ une fonction continue. Soit $A : I \rightarrow \mathbb{R}$ une primitive de a . Soit l'équation différentielle :*

$$\dot{y} = a(x)y, \tag{1.3}$$

est solutions sur I de (1.3) sont les fonctions y définies par :

$$y(x) = ke^{A(x)},$$

où $k \in \mathbb{R}$ est une constante quelconque, Si $a(x) = a$ est une fonction constante, alors une primitive est par exemple $A(x) = ax$ et on retrouve les solutions du théorème (1.2.1).

1.2.3 Equation différentielle de type $y' = a(x)y + b(x)$

Il nous reste le cas général de l'équation différentielle linéaire d'ordre 1 avec second membre :

$$\dot{y} = a(x)y + b(x), \tag{1.4}$$

où $a : I \rightarrow \mathbb{R}$ et $b : I \rightarrow \mathbb{R}$ sont des fonctions continues. L'équation homogène associée est :

$$\dot{y} = a(x)y. \quad (1.5)$$

Il n'y a pas de nouvelle formule à apprendre pour ce cas. Il suffit d'appliquer le principe de superposition : les solutions de (1.4) s'obtiennent en ajoutant à une solution particulière de les (1.4) solutions de (1.5). Ce qui donne :

Si y_0 est une solution de (1.4), alors les solutions de (1.4) sont les fonctions $y : I \rightarrow \mathbb{R}$ définies par :

$$y(x) = y_0(x) + ke^{A(x)} \quad \text{avec } k \in \mathbb{R},$$

où $x \mapsto A(x)$ est une primitive de $x \mapsto a(x)$.

La recherche de la solution générale de (1.4) se réduit donc à la recherche d'une solution particulière. Parfois ceci se fait en remarquant une solution évidente.

Exemple 1.2.1 *L'équation différentielle $\dot{y} = 2xy + 4x$ a pour solution particulière :*

$$y_0(x) = -2,$$

donc l'ensemble des solutions de cette équation sont les $y(x) = -2 + ke^{x^2}$, où $k \in \mathbb{R}$.

Recherche d'une solution particulière : méthode de variation de la constante :

Le nom de cette méthode est paradoxal mais justifié ! C'est une méthode générale pour trouver une solution particulière en se ramenant à un calcul de primitive. La solution générale de (1.5) $\dot{y} = a(x)y$ est donnée par $y(x) = ke^{A(x)}$, avec $k \in \mathbb{R}$ une constante. La méthode de la variation de la constante consiste à chercher une solution particulière sous la forme $y_0(x) = k(x)e^{A(x)}$, où k est maintenant une fonction à déterminer pour que \dot{y} soit une solution de (1.4) $\dot{y} = a(x)y + b(x)$. Puisque $\dot{A} = a$, on a :

$$\dot{y}_0(x) = a(x)k(x)e^{A(x)} + \dot{k}(x)e^{A(x)} = a(x)y_0(x) + \dot{k}(x)e^{A(x)}.$$

Ainsi :

$$\dot{y}_0(x) - a(x)y_0(x) = \dot{k}(x)e^{A(x)}.$$

Donc y_0 est une solution de (1.4) si et seulement si :

$$\dot{k}(x)e^{A(x)} = b(x) \Leftrightarrow \dot{k}(x) = b(x)e^{-A(x)} \iff k(x) = \int b(x)e^{-A(x)}.$$

Ce qui donne une solution particulière $y_0(x) = \left(\int b(x)e^{-A(x)} \right) e^{A(x)}$ de (1.4) sur I . La solution générale de (1.4) est donnée par :

$$y(x) = y_0(x) + ke^{A(x)}, \quad k \in \mathbb{R}.$$

1.3 Problèmes à valeur initiale (PVI)

On appelle problème de Cauchy la donnée d'une équation différentielle et d'une condition initiale :

$$\begin{cases} \frac{dx}{dt} = f(t, x) \\ x(t_0) = x_0 \end{cases} \quad (1.6)$$

où $x_0 \in \mathbb{R}$ est la condition initiale de la solution $x(t)$ au temps t_0 .

Méthode d'Euler comme méthodes numériques pour résoudre (PVI) : Soit un intervalle $[a, b]$ sur lequel on cherche la solution d'un problème de Cauchy

$$\begin{cases} \frac{dx}{dt} = f(t, x) \\ x(t_0) = x_0 \end{cases}$$

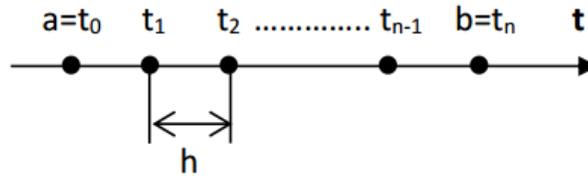


FIG. 1.2 – Le pas h .

La fonction $f(t, y(t))$ vérifie la condition de *Lipschitz* en y sur le rectangle \mathbb{R} . La première étape consiste à diviser l'intervalle donné en n points équidistants ce qui donne un pas d'intégration $h = \frac{b-a}{n}$ le point d'abscisse t_i est donné par $t_i = a + ib$ pour $i = 1, \dots, n$.

On va donc résoudre le problème sur l'intervalle $[a = t_0, b = t_n]$ avec $x(t_0 = a) = x_0$. Si les fonctions $y(t), \dot{y}(t)$, sont continues, on peut écrire le développement en série de Taylor pour $y(t)$ au voisinage de t_0 .

$$y(t) = y(t_0) + \dot{y}(t_0) \frac{(t-t_0)}{1!} + y''(t_0) \frac{(t-t_0)^2}{2!} + O(h^2) \dots$$

sachant que

$$\dot{y}(t_0) = f(t_0, y(t_0)) \quad \text{et} \quad h = t_1 - t_0.$$

Ecrivons

$$y(t_1) = y(t_0) + f(t_0, y(t_0)) h + y''(t_0) \frac{h^2}{2!}.$$

Si h est suffisamment petit alors on peut négliger h^2 et donc les termes d'ordre deux et plus, on obtient donc :

$$y(t_1) = y(t_0) + f(t_0, y(t_0)) h \longrightarrow y_1 = y_0 + hf(t_0, y_0).$$

C'est l'approximation d'Euler d'ordre 1, en répétant le procédé on génère une séquence de points $y_1, y_2, \dots, y_{n-1}, y_n$ approximant $y = y(t)$, en général :

$$\begin{cases} t_i = a + ih & i = 0, 1, 2, \dots, (n - 1) \\ y_{i+1} = y_i + hf(t_i, y_i), & y_0 = y(t_0) \end{cases}$$

Chapitre 2

Optimisation Numérique par des Algorithmes Inspirés de la Nature

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 Inspirés de la Nature

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 inspirés de la nature

Algorithmes ont inspiré par des chercheurs qui explorent de nouveaux moyens pour résoudre des problèmes d'optimisation dus à l'inefficacité des solutions traditionnelles. la

popularité d’algorithmes inspirés de la nature qui sont principalement influencés par la capacité des systèmes biologiques. Les renseignements sur les essaims ont suscité un grand intérêt [1] l’intelligence de l’essaim a été définie comme étant des algorithmes de conception ou des dispositifs distribués de résolution de problèmes basés sur la comportement collectif des sociétés animales et des colonies sociales.

Le but essentiel de cette étude est de développer des méta-heuristiques en adoptant les capacités de résolution de problèmes des insectes. De nombreuses études a met l’accent sur les insectes sociaux comme les termites, les abeilles, les guêpes et espèces de fourmis. Il s’agit d’un bon exemple d’essaim est que les abeilles pullulent autour de leurs ruches ; Cependant, la logique peut facilement être étendu à d’autres systèmes ayant un comportement similaire. Une colonie de fourmis et une volée d’oiseaux peuvent s’appeler un essaim.

Un système immunitaire est également un essaim de cellules et de molécules. Particulièrement, l’algorithme d’optimisation de l’essaim (*PSO*) a été développé sur la base du comportement social des oiseaux ou des poissons de la scolarisation.

Un recent algorithme inspiré de la nature est l’Algorithme de Pollinisation des Fleurs (*APF*), proposé par *Xin-She Yang* en 2012 [16]. Il est basé sur la pollinisation des fleurs.

Exemples d’algorithmes inspirés de la nature

Dans le tableau (2.1) on cite quelques algorithmes avec leur système d’inspiration.

2.1.2 Avantages d’utiliser les algorithmes inspirés de la nature

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

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	Echolocation caractéristique des microbes
Pollinisation des fleurs	Pollinisation des plantes à fleurs

TAB. 2.1 – Quelques Algorithmes avec leurs inspiration

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.

2.2 Algorithme de Pollinisation des Fleurs (APF)

L'(APF) est un algorithme récemment proposé par *Xin-She Yang* [15] qui a mis au point un algorithme robuste appliqué aux problèmes d'optimisation.

2.2.1 Description de processus de pollinisation chez les fleurs

La pollinisation est très importante. Cela mène à la création de nouvelles graines qui poussent dans de nouvelles plantes. Cela commence dans la fleur.

Les plantes à fleurs ont plusieurs parties différentes qui sont important dans la pollinisation [8]. Les fleurs ont des parties mâles appelées étamines produisant une poudre collante appelée pollen. Fleur sont également une partie féminine appelée le pistil. Le sommet du pistil s'appelle la stigmatisation et est souvent collante. Les graines sont faites au base du pistil, dans l'ovule. Pour être pollinisé, le pollen doit être déplacé d'une étamine à la stigmatisation [17]. Il y en a deux types de pollinisation :

Auto-pollinisation : Lorsque le pollen d'une fleur pollinise la même fleur ou les mêmes fleurs de la même plante, le processus s'appelle auto-pollinisation. Cela se produit lorsqu'une fleur contient à la fois les gamètes mâle et femelle voir (2.1).



FIG. 2.1 – Auto-pollinisation.

Pollinisation croisée : La pollinisation croisée se produit lorsque les grains de pollen sont déplacés vers une fleur d'une autre plante. Le processus de pollinisation croisée se produit à l'aide d'agents abiotiques ou biotiques tels que des insectes, des oiseaux, des

escargots, des chauves-souris et d'autres animaux en tant que pollinisateurs.

La pollinisation abiotique est un processus dans lequel la pollinisation a lieu sans implication d'agents externes. Seulement environ 10% des plantes entrent dans cette catégorie. Le processus de pollinisation qui nécessite des pollinisateurs externes est connu sous le nom de pollinisation biotique [3] pour déplacer le pollen de l'anthere au stigmate.

La pollinisation des insectes a lieu chez les plantes à pétales colorés et à forte odeur qui attirent les abeilles, les papillons de nuit, les coléoptères, les guêpes, les fourmis et les papillons [14]. Les insectes sont attirés par les fleurs en raison de la disponibilité du nectar et du pollen comestible. Lorsqu'un insecte est assis sur la fleur, les grains de pollen collent au corps. Lorsque l'insecte visite une autre fleur, le pollen est transféré dans la stigmatisation, ce qui facilite la pollinisation. La pollinisation est également facilitée par les vertébrés comme les oiseaux et les chauves-souris. Les fleurs pollinisées par les chauves-souris ont généralement des pétales de couleur blanche et une forte odeur. Les oiseaux pollinisent généralement les fleurs à pétales rouges et sans odeur voir (2.2).

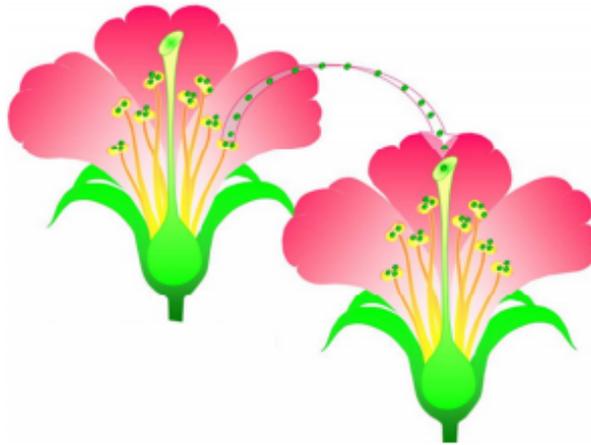


FIG. 2.2 – Pollinisation croisée.

2.2.2 Formulation d'(APF)

Les quatre règles ci-dessous sont utilisées pour résumer les ci-dessus caractéristiques du processus de pollinisation, constance florale et comportement des pollinisateurs [8]

1. La pollinisation biologique et croisée est considérée comme processus globale de pollinisation et pollinisateurs porteurs de pollen déplacer d'une manière qui confirme les vols Lévy.
2. Pour la pollinisation locale, la pollinisation abiotique et l'auto-pollinisation sont utilisés.
3. La constance florale peut être considérée comme une reproduction la probabilité est proportionnelle à la similitude de deux fleurs impliquées
4. La pollinisation locale et la pollinisation globale sont contrôlées par une probabilité de commutation $p \in [0.1]$.

Les principales étapes de (*APF*) ou simplement l'algorithme de la fleur [18] sont illustrés via le tableau (2.2), tandis que l'organigramme (*APF*).

2.2.3 Application d'(APF)

L'*APF* est un algorithme récemment proposé qui s'applique uniquement à certains problèmes. *Xin-She Yang* [5] a mis au point un algorithme de pollinisation des fleurs et l'a appliqué aux problèmes d'optimisation. Il a été utilisé pour :

1. *Oussama Abdel-Raouf, et. al* [10] ont proposé une nouvelle méthode hybride d'optimisation, appelée (algorithme de pollinisation des fleurs avec recherche d'harmonie chaotique), afin d'améliorer la précision de la recherche. Cet algorithme a été utilisé pour résoudre des énigmes de sudoku. Les résultats ont montré que le nouveau algorithme est précis et efficace par rapport à l'algorithme standard Harmony Search (*HS*). Les auteurs ont indiqué que le meilleur résultat est dû à la capacité de recherche supérieure du (*APF*).

Pseudo-code d'(APF)

Objectif min ou max $f(x)$, $x = (x_1, x_2, \dots)$ Initialiser la population de n fleurs/pollen gamètes avec des solutionsTrouver la meilleure solution g_* dans la population initialeDéfinir une probabilité de commutation $p \in [0, 1]$ **Tandis que** ($t < MaxGeneration$)**Pour** $i = 1 : n$ (toutes les n fleurs de la population)**Si** $rand < p$,Draw a (d-dimensional) step vector L which obeys a Levy distributionPollinisation globale via $x_i^{t+1} = x_i^t + l(g_* - x_i^t)$ **Autre**Tirer d'une distribution uniforme dans $[0, 1]$ Choisissez au hasard j et k parmi toutes les solutionsPollinisation locale via $x_i^{t+1} = x_i^t + \epsilon(x_i^t - x_k^t)$ **Fin Si**

Evaluer de nouvelles solutions

Si de nouvelles solutions sont meilleures, mettez-les à jour dans la population

Fin PourTrouvez la meilleure solution actuelle g_* **Fin Que**

TAB. 2.2 – Pseudo-code d'APF

2. *Oussama Abdel-Raouf et al* [9] ont présenté un algorithme hybride de pollinisation de fleurs avec optimisation de particules, afin de résoudre des problèmes d'optimisation sous contrainte. Le nouvel algorithme a été validé à l'aide de plusieurs problèmes de conception mathématiques et d'ingénierie de référence. Les résultats obtenus par l'algorithme proposé ont été comparés à d'autres algorithmes et ils sont supérieurs, plus précis, fiables et efficaces pour trouver une solution globale optimale par rapport aux autres algorithmes.
3. *Gaganpreet Kaur et al* [4] ont présenté une nouvelle technique pour améliorer la compression d'images fractales par l'optimisation basée sur la pollinisation. La qualité visuelle de l'image est bien meilleure avec l'optimisation basée sur la pollinisation par rapport à la compression d'images fractales, les résultats montrent que l'utilisation du (APF) réduit le temps d'encodage et améliore la qualité visuelle de l'image.
4. *Gaganpreet Kaur, et al* [5] utilisent l'(APF) pour segmenter les images en couleurs.

L'algorithme a été mis en œuvre pour extraire les grappes optimales des images colorées. L'algorithme proposé fonctionnait mieux sur les images colorées et le temps nécessaire à la segmentation des images était considérablement réduit par rapport à la plupart des algorithmes d'optimisation.

5. *E. Emry et coll* [2] ont proposé une nouvelle segmentation des vaisseaux rétiniens effectuée à l'aide de l'algorithme de recherche par pollinisation des fleurs (*FPSA*) afin de localiser la carte des vaisseaux rétiniens utilisant la fonction de condition physique potentielle des coefficients flous. Le résultat obtenu prouve la convergence rapide du (*FPSA*) et la robustesse des résultats obtenus même avec une image présentant des anomalies.
6. *Rui Wang et al* [12] ont proposé une amélioration dimension par dimension à l'aide de (*APF*). Dans cet algorithme modifié, une approche de recherche locale a été utilisée pour améliorer la capacité de recherche locale. Les enquêtes simulées ont montré que l'approche proposée peut améliorer la vitesse de convergence et la qualité des solutions.
7. *Prathiba R et al* [11] ont proposé un (*APF*) pour optimiser la répartition de charge économique dans le fonctionnement du système électrique. L'objectif était de minimiser les coûts de carburant.

Chapitre 3

Etude Experimentale

Dans ce chapitre on utilise le logiciel matlab pour tester l'efficacité de l'(APF) à travers une étude de simulation telles que la comparaison entre les résultats exacts est celles obtenus par la méthode Euler qui est considérée comme un outil classique de résolution numérique d'un (PVI). Les résultats affichés montrent une performance d'(APF) meilleur que la méthode d'Euler en termes de la qualité de la solution. L'(APF) donne une approximation précise et satisfaisante pour la solution avec un minimum des erreurs.

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

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

sont appelées problème à valeur initiale (PVI); ils symbolisent le problème suivant :

Trouver une fonction $y(x)$ continue et différentiable pour $x \in [a, b]$ telle que $\dot{y} = f(x, y)$ à partir de $y(a) = y_0$ pour tout $x \in [a, b]$. Ce problème possède une solution unique lorsque :

f est continu sur $[a, b] \times \mathbb{R}$, et satisfait la condition de *Lipschitz* ; il existe une constante $k \geq 0$ comme pour tout $x \in [a, b]$ et tout couple $(\theta_1, \theta_2) \in \mathbb{R} \times \mathbb{R}$. La recherche numérique des solutions optimales d'un problème de valeur initiale (*PVI*) s'obtient avec des approximations $y(x_0 + h), \dots, y(x_0 + nh)$ où $a = x_0$ et $h = \frac{(b - a)}{(n + 1)}$. Pour plus de précision sur la solution, nous devons utiliser une très petite taille de pas h , qui comprend un plus grand nombre d'étapes, donc un temps de calcul plus long qui n'est pas disponible dans les méthodes numériques utiles comme les méthodes d'Euler et de Runge-Kutta [6], ce qui est insuffisant et, par conséquent, des méthodes d'optimisation globales sont utilisées.

3.1.1 Fonction objectif

L'idée principale pour obtenir notre fonction objective est d'utiliser la formule de différence finie f est d'utiliser la formule de la différence finie pour le dérivé et l'équation (3.1) on obtient

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

Ainsi,

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

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

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

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

$$F(y) = \sum_{i=0}^d \left[\frac{y_j - y_{j-1}}{h} - f(x_j, y_{j-1}) \right]^2. \quad (3.2)$$

3.1.2 La consistance

Nous nous intéressons au calcul de $y = (y_1, y_2, \dots, y_d)$ qui minimise la fonction objective dans (3.2) Nous avons de formule de Taylor l'ordre 1

$$\frac{y_j - y_{j-1}}{h} = y'_{j-1} + 0(h^2) \quad j = 1, \dots, d.$$

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

$$\frac{y_j - y_{j-1}}{h} - f(x_{j-1}, y_{j-1}) = y'_{j-1} - f(x_{j-1}, y_{j-1}) + 0(h) \quad j = 1, \dots, d.$$

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

3.2 Application : Le modèle exponentielle

Pour illustrer la méthode traitée et démontrer sa efficacité dans le calcul, le modèle exponentiel est considéré en prenant une taille de pas uniforme h . La principale motivation dans la sélection d'exemple d'application vient de la grande importance de modèle exponentiel dans la modélisation des problèmes de la vie réelle. Les fonctions exponentielles sont fréquemment utilisées pour modéliser la croissance ou dépréciation des investissements financiers, croissance démographique, décroissance radioactive, et les phénomènes où une quantité est autorisés à subir une croissance sans contrainte.

Dans le tableau (3.2), les résultats de la (APF) et les résultats exacts pour le problème étudié est présenté. Le tableau (3.3) présente l'Erreur absolu. Pour plus de commodité, les paramètres à générés par (APF) sont présentés dans le tableau (3.1) Tous les calculs ont été effectué sur un système d'exploitation professionnel MSWindow 2007 dans le compilateur R2013a de la version de l'environnement Matlab sur PC Intel Duo Core 2.20 Ghz.

Paramètres	La valeur
Dimension des variables de recherche (d)	10
Nombre total d'itérations (N)	2000
Taille de la population (n)	20
Probabilité de commutateur (p)	0.8

TAB. 3.1 – Paramètres adoptés (APF).

Soit le (PVI) suivant :

$$\begin{cases} \frac{dy}{dx} = y \\ y(0) = 1 \end{cases} \quad 0 \leq x \leq 1.$$

Pour $d = 10, h = \frac{1-0}{10} = 0.1, x_0 = 0, y_0 = 1$ La solution exacte est

$$y(x) = e^x.$$

3.2.1 Paramètres relatives au (PVI)

Dans ce travail, les paramètres adoptés par (APF) pour traiter notre problème est cités dans le tableau 3.1 :

3.2.2 Paramètres relatives au (APF)

(APF) est un instrument d'optimisation. Ensuite, l'équation différentielle essentielle est la conversion en une forme de discrétisation. La formule de différence en arrière est utilisée pour convertir une équation différentielle en une forme de discrétisation lorsque le terme dérivé est remplacé dans la forme discrétisée par un quotient de différence pour des approximations. Les paramètres liés à (PVI) sont les suivants :

1. L'intervalle de (PVI) est également divisé en $(n + 1)$ sous-intervalles équidistants de longueur $h = (b - a) / (n + 1)$. Comme l'intervalle d'étude est $x \in [0, 1]$ alors $n = 9$ est le nombre de nœuds intérieurs et la taille de pas $h = 0.1$

i	x_i	Résultats exactes	Résultats d'(APF)	Résultats d'Euler
1	0.1	1.1052	1.1053	1.1000
2	0.2	1.2214	1.2215	1.2100
3	0.3	1.3499	1.3492	1.3310
4	0.4	1.4918	1.4907	1.4641
5	0.5	1.6487	1.6467	1.6105
6	0.6	1.8221	1.8191	1.7716
7	0.7	2.0138	2.0093	1.9487
8	0.8	2.2255	2.2195	2.1436
9	0.9	2.4596	2.4524	2.3579
10	1.0	2.7183	2.7094	2.5937

TAB. 3.2 – Résultats exactes et résultats numériques obtenus par (APF) pour $d=10$.

2. La condition initiale et $y(0) = 1$

3. La fonction objectif

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

Nous passons maintenant à l'étude expérimentale lorsque nous introduisons un (PVI) linéaire.

3.3 Résultats Numériques

Les résultats obtenus et la comparaison entre la performance du (APF) et la méthode d'Euler sont donnés par le tableau suivant :

Les résultats numériques du tableau (3.2) sont illustrés à la figure (3.1). La comparaison entre les performances de (APF) et d'Euler face aux résultats exacts confirme que (APF) est supérieur à Euler car il présente une courbe très proche de la courbe exacte contrairement à la méthode d'Euler.

Le tableau (3.3) présente l'erreur absolue entre les résultats exacts et les méthodes étu-

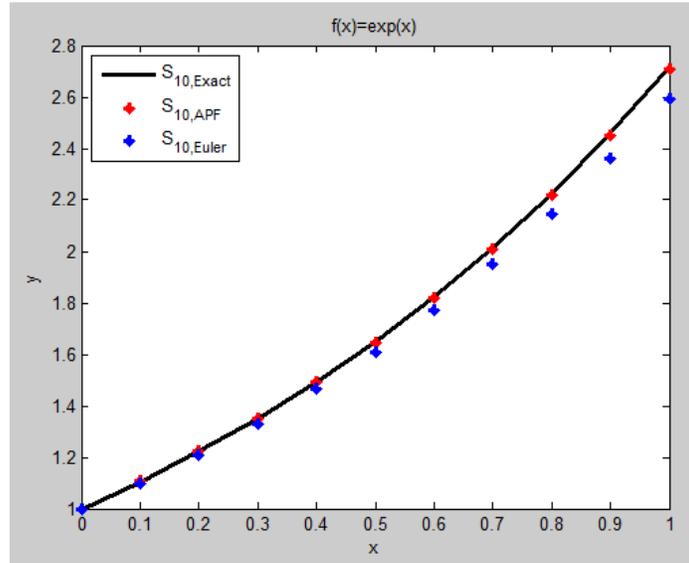


FIG. 3.1 – Plot des résultats exactes et résultats numériques obtenus par (APF) pour $d=10$.

diées

La figure (3.2) montre la représentation graphique de l'erreur absolue. On remarque que l'(APF) offre un erreur absolue très négligeable comparé au celui de la méthode d'Eleur dans ces deux représentations tabulaire et graphique.

Discussions : Les (APF) ont développé avec succès pour éliminer les caractéristiques de la pollinisation des fleurs. Les résultats de notre simulation indiquent que le (APF) est simple, qu'il réduit le temps, qu'il est flexible et qu'il est exponentiellement meilleur pour

i	x_i	Err. Abs. d'(APF)	Err. Abs. d'Euler
1	0.1	0.0001	0.0052
2	0.2	0.0001	0.0114
3	0.3	0.0007	0.00189
4	0.4	0.0011	0.0227
5	0.5	0.0020	0.0382
6	0.6	0.0030	0.0505
7	0.7	0.0045	0.0651
8	0.8	0.0060	0.0819
9	0.9	0.0072	0.1017
10	1.0	0.0089	0.1246

TAB. 3.3 – Erreur absolue entre les résultats exactes et les méthodes étudiées .

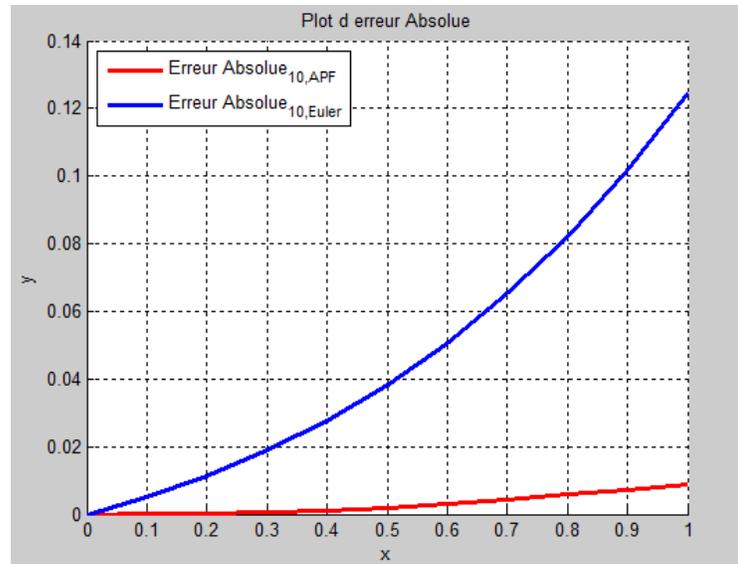


FIG. 3.2 – Plot d'erreur absolue

résoudre l'optimisation (*PVI*).

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.

L'utilisation de métaheuristiques peut paraître relativement simple, en première approche, mais il est souvent nécessaire d'adapter l'algorithme au problème optimisé.

Comme ce mémoire traite la technique d'optimisation par un algorithme inspiré de la nature qui est (*APF*), qui sont des alternatives pour éviter les limitations des techniques d'optimisation traditionnelles. Notre étude indique que l'(*APF*) est simple, flexible et exponentiellement meilleur pour résoudre le (*PVI*).

l'(*APF*) peut être utilisé pour résoudre des problèmes d'optimisation à objectif unique ou multi-objectif. Des études indiquent que l'(*APF*) réduit le temps, améliore les résultats et améliore les performances par rapport à d'autres techniques d'optimisation. l'(*APF*)

semble très prometteur et est encore au stade naissant et peut être appliqué à l'analyse d'images médicales. Il serait utile que les recherches ultérieures puissent se concentrer sur l'extension de la méthodologie proposée pour optimiser le (PVI) par d'autres variantes de l'(APF). En fin de compte, on peut s'attendre à ce que le problème proposé soit optimisé par d'autres algorithmes métaheuristiques.

Bibliographie

- [1] Baykasoglu Adil, Ozabakir Lale, Tapkan Pinar, “Artificial Bee Colony and its Application to Generalised Assignment Problem”, I-Tech Education and Publication, 2007.
- [2] E. Emary, Hossam M. Zawbaa , Aboul Ella Hassanien , Mohamed F. Tolba, “Retinal vessel segmentation based on flower pollination search algorithm”, IBICA- 2014, Advances in Intelligent Systems and Computing, Springer, 2104.
- [3] Flower Pollination by biology.tutorvista.com/animal-kingdom, Pearson, 2005
- [4] Gaganpreet Kaur Dheerendra Singh, Manjinder Kaur, “Robust and Efficient ‘RGB’ based Fractal Image Compression : Flower Pollination based Optimization”, International Journal of Computer Applications (0975 – 8887), Volume 78 – No.10, 2013
- [5] Gaganpreet Kaur, Dr. Dheerendra Singh, “Pollination Based Optimization For Color Image Segmentation”, International Journal of Computer Engineering and Technology Volume 3, Issue 2, 2012
- [6] Henrici P. Elements of Numerical Analysis. Mc Graw-Hill. New York 1964.
- [7] Laurent beroul, Isabelle tenoul, Sébastien cario "Sébastien cario", Cned 2013
- [8] N. M. Waser., (1986), “Flower constancy : definition, cause and measurement”, The American Naturalist, 127 (5), 596-603.
- [9] O. Abdel-Raouf, M. Abdel-Baset, I. El-henawy, “A New Hybrid Flower Pollination Algorithm for Solving Constrained Global Optimization Problems”, International Journal of Applied Operational Research Vol. 4, No. 2, pp. 1-13, Spring 2014.

- [10] Osama Abdel_Raouf, Ibrahim El-Henawy and Mohamed Abdel-Baset, “A Novel Hybrid Flower Pollination Algorithm with Chaotic Harmony Search for Solving Sudoku Puzzles”, Journal of Modern Education and Computer Science, 2014, pp 38-44.
- [11] R. Prathiba, M. Balasingh Moses, S. Sakthivel, Flower Pollination Algorithm Applied for Different Economic Load Dispatch Problems, International Journal of Engineering and Technology (IJET), Vol 6 No 2 Apr-May 2014.
- [12] Rui Wang and Yongquan Zhou, Flower Pollination Algorithm with Dimension by Dimension Improvement, Hindawi Publishing Corporation Mathematical Problems in Engineering Volume 2014, Article ID 481791.\
- [13] Théorie des équation différentielle ordinaire,A.Munnier,Institut élie cartan 2006-2007
- [14] Willmer P, “Pollination and Floral Ecology”, Princeton University Press.2011
- [15] Xin-She Yang, Flower Pollination Algorithm for Global Optimization, arXiv :1312.5673v1 [math.OC] 19 Dec2013.
- [16] Xin-She Yang, Mehmet Karamanoglu, Xingshi He, “Multi-objective Flower Algorithm for Optimization”, ICCS 2013, Elsevier.
- [17] X. S. Yang., (2104), “Book Nature Inspired Optimization Algorithm”, Elsevier.
- [18] X. S. Yang., (2013), “Flower Pollination Algorithm for Global Optimization”, arXiv : 1312. 5673v1 [math. OC] 19 Dec.

Abréviations et Notations

Abréviations	Explication
PVI	Problèmes à valeur initiale.
APF	Algorithme de Pollinisation des Fleurs.
PSO	Particle Swarm Optimization.
AG	Algorithme génétique.
HS	Harmony Search.
EDO	Equation Différentielle Ordinaire.
ASPF	Algorithme de recherche par pollinisation des fleurs.
\mathbb{R}	L'ensemble des nombres réels.
\mathbb{Z}	L'ensemble des nombres entiers.
\mathbb{N}	L'ensemble des nombres naturels.

Annexe A : Le logiciel Matlab

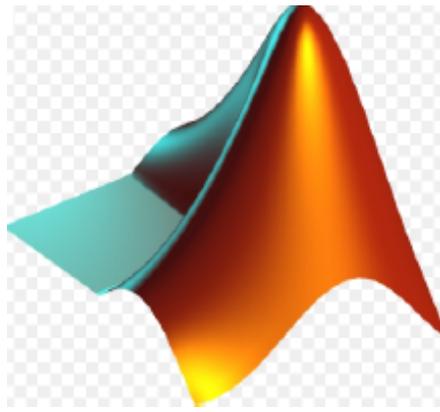


FIG. 3.3 – Icône du logiciel Matlab.

Matlab (3.3) est un logiciel interprété (donc sans phase préliminaire de compilation) qui exécute les opérations demandées séquentiellement, avec possibilité de boucle, test et saut.

Il ne manipule que des données numériques et ne sait effectuer aucun calcul formel à priori.

Variables Matlab : Les variables s'utilisent sans déclaration préalable de type, Matlab adoptant en interne l'objet matrice pour toutes les variables (scalaires ou non, réelle ou imaginaire, chaîne de caractères...). Le type (dimensionnement de la matrice) est donc géré automatiquement d'après l'usage (affectation) de la variable. Des calculs ou application de fonctions entre types non cohérents seront signalés par un message d'erreur et blocage de l'exécution du programme à cette ligne.

MATLAB est une application scientifique interactive orientée au calcul vectoriel et matriciel avec une puissante librairie de visualisation. Vous pouvez résoudre des problèmes de calcul très complexes d'une façon simple et rapide comparée aux langages de programma-

Fonction	Le role
Plot	permet de tracer une ou plusieurs courbes sur le même graphe
Legend	est un moyen utile pour étiqueter les séries de données tracées sur un graphique
hold on	permet de tracer plusieurs courbes sur la figure courante
grid on	met la grille sur le graphe tracé par plot

TAB. 3.4 – Les fonctions les plus utilisées dans ce travail

tion traditionnels du type *C* ou *FORTRAN*. Le nom MATLAB est dérivé de l'anglais MATrix LABoratory. Ce résumé vous présente d'une façon simple, les principales idées pour pouvoir utiliser cet outil de simulation sur les stations de travail disponibles au Centre de Calcul. Il ne remplace pas le guide de l'utilisateur fourni avec l'application (User's Guide and Reference Guide for MATLAB). Ce résumé, plus l'aide en ligne, sont largement pour développer des applications relatives au cours de CAN. Vous êtes invités à vérifier tous les exemples ici présentés directement sur un PC ou station UNIX. Une fois établie une session MATLAB, la commande `help nomdefonction`, fournit tous les renseignements relatifs à la fonction `nomdefonction`. Par exemple, la commande `help plot` donne toutes les options de la fonction de représentation graphique (plotting) en 2D. La commande `help` montre la liste des domaines qui ont une aide en ligne. Ainsi `help domaine` affiche toutes les fonctions en relation au domaine. Essayez, par exemple, `help comm`. Ces fonctions sont regroupées à la fin de cet annexe (extrait du guide de l'utilisateur). La commande `demo` présente un tour d'horizon de MATLAB sous la forme d'exemples.

Annexe B : Code Matlab des Algorithmes

3.4 Code Matlab de méthode d'Euler

```
function E=euler(f,a,b,ya,M)
    f=@(x) exp(x);
    a=0;
    b=1;
    ya=0;
    M=10;
    h=(b-a)/M;
    Y=zeros(1,M+1);
    T=a:h:b;
    Y(1)=ya;
    for j=1:M
        Y(j+1)=Y(j)+h*f(T(j));
    end
    E=[T' Y'];
```

3.5 Code Matlab d'Algorithme de Pollination de Fleurs (APF)

```
% -----%
% Flower pollination algorithm (FPA), or flower algorithm      %
% Programmed by Xin-She Yang @ May 2012                       %
% -----%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Notes: This demo program contains the very basic components of %
% the flower pollination algorithm (FPA), or flower algorithm (FA), %
% for single objective optimization. It usually works well for %
% unconstrained functions only. For functions/problems with %
% limits/bounds and constraints, constraint-handling techniques %
% should be implemented to deal with constrained problems properly. %
%                                                                %
% Citation details:                                           %
%1)Xin-She Yang, Flower pollination algorithm for global optimization,%
% Unconventional Computation and Natural Computation, %
% Lecture Notes in Computer Science, Vol. 7445, pp. 240-249 (2012). %
%2)X. S. Yang, M. Karamanoglu, X. S. He, Multi-objective flower %
% algorithm for optimization, Procedia in Computer Science, %
% vol. 18, pp. 861-868 (2013). %

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [best,fmin,N_iter]=fpa_demo(para)

% Default parameters
```

```
if nargin<1,
    para=[20 0.8];
end

n=para(1);           % Population size, typically 10 to 25
p=para(2);           % probability switch

% Iteration parameters
N_iter=2000;         % Total number of iterations

% Dimension of the search variables
d=10;
%Lb=-2*ones(1,d);
%Ub=2*ones(1,d);
Lb=0*ones(1,d);
Ub=4*ones(1,d);

% Initialize the population/solutions
for i=1:n,
    Sol(i,:)=Lb+(Ub-Lb).*rand(1,d);
    Fitness(i)=Fun(Sol(i,:));
end

% Find the current best
[fmin,I]=min(Fitness);
best=Sol(I,:);
S=Sol;
```

```
% Start the iterations -- Flower Algorithm
for t=1:N_iter,
    % Loop over all bats/solutions
    for i=1:n,
        % Pollens are carried by insects and thus can move in
        % large scale, large distance.
        % This L should replace by Levy flights
        % Formula:  $x_i^{t+1} = x_i^t + L (x_i^t - gbest)$ 
        if rand > p,
            %% L=rand;
            L=Levy(d);
            dS=L.*(Sol(i,:)-best);
            S(i,:)=Sol(i,:)+dS;

            % Check if the simple limits/bounds are OK
            S(i,:)=simplebounds(S(i,:),Lb,Ub);

            % If not, then local pollination of neighbor flowers
            else
                epsilon=rand;
                % Find random flowers in the neighbourhood
                JK=randperm(n);
                % As they are random, the first two entries also random
                % If the flower are the same or similar species, then
                % they can be pollinated, otherwise, no action.
                % Formula:  $x_i^{t+1} + \epsilon * (x_j^t - x_k^t)$ 
                S(i,:)=S(i,:)+epsilon*(Sol(JK(1),:)-Sol(JK(2),:));
            end
        end
    end
end
```

```
        % Check if the simple limits/bounds are OK
        S(i,:)=simplebounds(S(i,:),Lb,Ub);
    end

    % Evaluate new solutions
    Fnew=Fun(S(i,:));
    % If fitness improves (better solutions found), update then
    if (Fnew<=Fitness(i)),
        Sol(i,:)=S(i,:);
        Fitness(i)=Fnew;
    end

    % Update the current global best
    if Fnew<=fmin,
        best=S(i,:) ;
        fmin=Fnew ;
    end

end

% Display results every 100 iterations
if round(t/100)==t/100,
    best
    fmin
end

end

% Output/display
```

```
disp(['Total number of evaluations: ',num2str(N_iter*n)]);
disp(['Best solution=',num2str(best),' fmin=',num2str(fmin)]);

% Application of simple constraints
function s=simplebounds(s,Lb,Ub)
    % Apply the lower bound
    ns_tmp=s;
    I=ns_tmp<Lb;
    ns_tmp(I)=Lb(I);

    % Apply the upper bounds
    J=ns_tmp>Ub;
    ns_tmp(J)=Ub(J);
    % Update this new move
    s=ns_tmp;

% Draw n Levy flight sample
function L=Levy(d)
% Levy exponent and coefficient
% For details, see Chapter 11 of the following book:
% Xin-She Yang, Nature-Inspired Optimization Algorithms, Elsevier, (2014).
beta=3/2;
sigma=(gamma(1+beta)*sin(pi*beta/2)/(gamma((1+beta)/2)*beta
    *2^((beta-1)/2)))^(1/beta);
u=randn(1,d)*sigma;
```

```
v=randn(1,d);
step=u./abs(v).^(1/beta);
L=0.01*step;

% Objective function
function z=Fun(u)
z = (10.*(u(1)-1)-u(1)-0.0)^2+(10.*(u(2)-u(1))-u(2)-0.2)^2
+(10.*(u(3)-u(2))-u(3)-0.3)^2+(10.*(u(4)-u(3))-u(4)-0.4)^2
+(10.*(u(5)-u(4))-u(5)-0.5)^2+(10.*(u(6)-u(5))-u(6)-0.6)^2
+(10.*(u(7)-u(6))-u(7)-0.7)^2+(10.*(u(8)-u(7))-u(8)-0.8)^2
+(10.*(u(9)-u(8))-u(9)-0.9)^2+(10.*(u(10)-u(9))-u(10)-1)^2
```