

THÈSE DE DOCTORAT DE L'UNIVERSITÉ PARIS 6

Spécialité :
Mathématiques

Présentée par
Anne AUGER

pour obtenir le grade de
DOCTEUR de l'UNIVERSITÉ PARIS 6

CONTRIBUTIONS THÉORIQUES ET NUMÉRIQUES À L'OPTIMISATION CONTINUE PAR ALGORITHMES ÉVOLUTIONNAIRES

Soutenue le 20 Décembre 2004 devant le jury composé de

M. Raphaël CERF	Rapporteur
M. Petros KOUMOUTSAKOS	Examineur
M. Jean LACROIX	Examineur
M. Claude LE BRIS	Directeur de thèse
M. Pierre ROUCHON	Rapporteur
M. Marc SCHOENAUER	Directeur de thèse

Table des matières

Introduction générale	7
I Algorithmes évolutionnaires pour l'optimisation continue	17
1 Etat de l'art	19
1.1 Introduction	20
1.2 Le contexte: optimisation stochastiques et optimisation globale	21
1.3 Les Stratégies d'Evolution (ES)	24
1.3.1 Premières idées d'adaptation dans les ESs: la règle des 1/5	25
1.3.2 Les SA-ES: Muter les paramètres de mutation	26
1.3.3 Les ES dérandomisés	31
1.4 Autres algorithmes évolutionnaires continus: DE et GA	39
1.4.1 Les Algorithmes génétiques	39
1.4.2 L'algorithme Differential Evolution (DE)	42
1.5 Les Algorithmes à Evolution de Distribution (EDAs)	43
1.5.1 Population Based Incremental Learning (<i>PBIL_c</i>)	44
1.5.2 EMNA (Estimation of Multivariate Normal Algorithm)	44
1.6 Adaptation et auto-adaptation: illustrations	44
1.7 Mythes et réalités de l'optimisation globale	47
Conclusion	49
2 Méthode explicite d'adaptation de la matrice de covariance	51
2.1 Introduction	52
2.2 Matrice de covariance optimale	53
2.3 Quelles matrices de covariances pour les méthodes historiques?	55

2.4	Apprendre $(\frac{1}{2}H)^{-1}$: minimisation au sens des moindres carrés	56
2.5	L'algorithme LS-CMA-ES	58
2.5.1	L'algorithme $(1, \lambda)$ -LS-CMA-ES	59
2.5.2	Résultats expérimentaux	62
2.5.3	L'algorithme $(\mu/\mu, \lambda)$ -LS-CMA-ES	67
2.6	Robustesse au bruit	69
2.7	Discussion et perspectives	73

II Etudes théoriques des Algorithmes Evolutionnaires sur un espace continu

77

3	Introduction	79
3.1	Etat de l'art	81
3.1.1	La Théorie du <i>progress rate</i>	81
3.1.2	Convergence globale	82
3.1.3	Convergence locale	83
3.2	Contexte des résultats théoriques de cette thèse	85
4	Analyse de la convergence de deux algorithmes adaptatifs	87
4.1	Introduction	88
4.2	Notations, définitions et rappel	90
4.3	Convergence sur la fonction sphère	91
4.3.1	Convergence L^p	91
4.3.2	Lien avec la théorie du <i>progress rate</i>	92
4.3.3	Convergence presque sûre	93
4.4	Convergence du $(1, \lambda)$ -ES avec $H(x) = x $	94
4.4.1	Convergence	94
4.4.2	Vitesse de convergence	98
4.5	Convergence du $(1, \lambda)$ -ES avec $H(x) = f'(x) $	99
4.5.1	Convergence de $f'(X_n)$	99
4.5.2	Vitesse de convergence	101
4.5.3	Optimalité des constantes sur la fonction sphère	102
4.6	En dimension quelconque	103

4.7	Expérimentations numériques	104
4.7.1	Approximation des constantes	105
4.7.2	Optimalité des constantes	106
4.8	Discussion	108
5	Convergence d'Algorithmes Evolutionnaires auto-adaptatifs	111
5.1	Introduction	112
5.2	Algorithme $(1, \lambda)$ -SA-ES et chaînes de Markov	120
5.2.1	L'algorithme étudié	120
5.2.2	Définition de la chaîne de Markov sous-jacente au $(1, \lambda)$ -SA-ES	122
5.2.3	Résultats préliminaires: Conditions suffisantes sur $(Z_n)_{n \geq 0}$ pour la convergence du $(1, \lambda)$ -SA-ES	123
5.2.4	Résumé des résultats principaux: Conditions garantissant la stabilité de $(Z_n)_{n \geq 0}$	126
5.3	Irréductibilité, small sets, apériodicité	129
5.3.1	Noyau de transition pour $(Z_n)_{n \geq 0}$	129
5.3.2	φ -irréductibilité, apériodicité, small sets	130
5.4	Conditions de drift	130
5.4.1	Sur l'ensemble Λ_α	135
5.5	Convergence du $(1, \lambda)$ -SA-ES	136
5.5.1	Positivité, récurrence au sens de Harris et Ergodicité géométrique pour $(Z_n)_{n \geq 0}$	136
5.5.2	Propriétés de la mesure invariante de $(Z_n)_{n \geq 0}$	137
5.5.3	Convergence p.s. de $\frac{1}{n} \ln(\ X_n\)$ et $\frac{1}{n} \ln(\sigma_n)$	139
5.5.4	Théorème Central Limite pour $(Z_n)_{n \geq 0}$	141
5.5.5	Application: approximation du taux de convergence	142
5.6	Discussion et conclusion	144
5.7	Annexe	147
5.7.1	Preuves de la Section 5.2	147
5.7.2	Etude du $(1, \lambda)$ -ES à pas constant	149
5.7.3	Etude du cas $\lambda = 1$	150
5.7.4	Preuve du Théorème 5	154
5.7.5	Preuve du Lemme 4	159

III Une application au contrôle par laser	161
6 Algorithmes évolutionnaires pour le contrôle de l'orientation de molécules	163
6.1 Introduction	166
6.2 Statement of the control problem	170
6.2.1 The system under study and the control problem	170
6.2.2 Choice of the set of electric fields	172
6.2.3 Choice of the cost function	174
6.2.4 Identification and classification of the fields obtained	175
6.3 Methodology	176
6.3.1 Gradient like algorithms	177
6.3.2 Evolutionary Algorithms	188
6.4 Results for the orientation problem	191
6.4.1 Optimized fields for (6.2.9) and (6.2.10)	192
6.4.2 Results for the hybrid criterion	193
6.4.3 Results for the train of kicks	194
6.5 Conclusion and future directions	194
Conclusion générale	199
Bibliographie	216

Introduction générale

L'optimisation continue s'intéresse aux problèmes où la fonction à optimiser, à valeurs réelles, est définie sur un espace de variables réelles. De tels problèmes apparaissent dans de très nombreux domaines, aussi divers que la physique, la chimie, la biologie ou encore l'ingénierie. Les différents contextes inhérents à ces domaines posent plusieurs difficultés. Tout d'abord, la fonction à optimiser peut être *multimodale*, c'est-à-dire présenter de nombreux optima locaux. Elle peut aussi être non différentiable, ou tout simplement, l'information d'ordre un ou deux sur cette fonction peut être difficilement disponible (dans le cas par exemple où la fonction à optimiser est donnée par un programme informatique) et l'approximation par différences finies du gradient peut se révéler imprécise. Par ailleurs, la fonction à optimiser peut résulter d'une expérience en physique, chimie ou biologie, et être en conséquence bruitée, mouvante, changeant au gré des conditions expérimentales. Dans ce contexte, les algorithmes évolutionnaires sont une alternative aux méthodes d'optimisation traditionnelles. Inspiré de la théorie de l'évolution des espèces de Darwin, plus précisément des principes de *variations aveugles* et *survie des plus adaptés*, le principe des algorithmes évolutionnaires est de faire évoluer un ensemble de points de l'espace de recherche dans le but de se concentrer sur les optima de la fonction à optimiser. Avant de discuter les avantages et inconvénients des méthodes évolutionnaires, rappelons brièvement le principe de base de ces algorithmes, ainsi que l'historique du domaine.

Principe de base des Algorithmes Evolutionnaires Les algorithmes évolutionnaires ont un principe commun : faire évoluer une *population* d'*individus*, *i.e.* un ensemble de points de l'espace de recherche, dans le but de minimiser une fonction f , dite fonction *fitness*, définie sur un espace de recherche E , à valeurs réelles. L'évolution de cette population d'individus est schématisée sur la Figure 1. Notons qu'un problème de maximisation peut être résolu en minimisant l'opposé de la fonction selon le même principe.

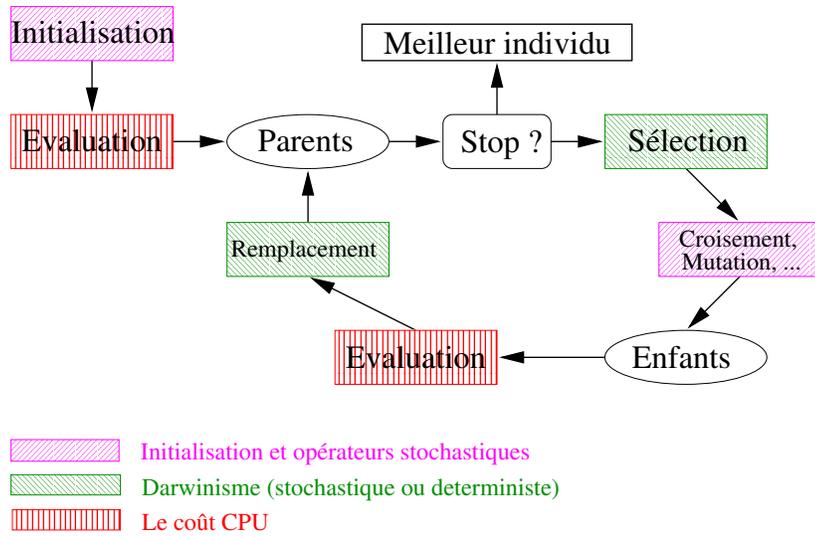


FIG. 1: Cycle d'un Algorithme Evolutionnaire

En pratique, on commence par choisir aléatoirement une population d'individus dans l'espace de recherche, qui forme la population initiale de *parents*. Cette population évolue ensuite jusqu'à ce qu'un critère d'arrêt soit atteint suivant les étapes (*générations*) suivantes.

1. **Sélection** d'individus (favorisant plutôt ceux de petite fitness) parmi la population (étape Darwinienne, où les meilleurs sont plus aptes à se reproduire).
2. **Variations** aléatoire des individus sélectionnés par
 - **croisement**: avec un autre point choisi aléatoirement (opérateur de $E \times E$ dans E).
 - **mutation**: (opérateur stochastique de E dans E).
3. **Remplacement**: On sélectionne parmi les enfants et (ou) parents ceux qui vont constituer les parents de la génération suivante (étape Darwinienne, où les meilleurs sont plus aptes à survivre).

En plus de ce vocabulaire emprunté à la biologie et à la théorie de l'évolution des espèces, on dit qu'un individu est *meilleur* qu'un second individu, si la fitness du premier est inférieure à la fitness du second.

Le cas continu: Dans le cas continu qui sera celui de cette thèse, un opérateur typique de mutation consistera à rajouter un vecteur gaussien au parent ; un croisement sera par exemple un barycentre entre deux parents tirés aléatoirement.

Historique des Algorithmes Evolutionnaires: Les algorithmes évolutionnaires ont été introduits dans les années 60. A l'époque, trois branches apparaissaient simultanément, donnant ce que sont aujourd'hui les algorithmes évolutionnaires : la programmation évolutionnaire, les algorithmes génétiques et les stratégies d'évolution. Plus tard, au début des années 90, apparaît une quatrième branche, la programmation génétique. Les différentes branches sont précisées ci-dessous.

L.J. Fogel [62], en Californie, fait évoluer des automates à états finis pour la prédiction de séries temporelles. Cette méthode, impliquant l'évolution de programmes, donne naissance à la programmation évolutionnaire (EP, pour *Evolutionary Programming*). Celle-ci a été ensuite développée et son domaine a été considérablement élargi par D.B. Fogel pour être appliqué, entre autre, à l'optimisation continue [61].

J. Holland [81], dans le Michigan, développe le concept d'algorithmes génétiques (GA pour *Genetic Algorithm*) dans le but de simuler et comprendre les mécanismes d'adaptation des espèces. Les algorithmes génétiques sont formalisés comme algorithmes d'optimisation sur un espace binaire par K.A. De Jong [43]. Ces algorithmes sont ensuite popularisés par Goldberg [68] et ont été utilisés dans de très nombreux domaines, y compris l'optimisation continue [69, 58, 121]. Ce sont encore à l'heure actuelle les algorithmes évolutionnaires les plus connus.

I. Rechenberg [122] et H.P. Schwefel [131], en Allemagne introduisent les stratégies d'évolution (ES pour *Evolution Strategy*) pour résoudre des problèmes d'ingénierie en aéronautique, pour lesquels les techniques d'optimisation classiques étaient prises en défaut. Beaucoup de concepts clés pour les algorithmes évolutionnaires continus ont été introduits dans cette branche. Parmi ces concepts, la notion d'*adaptation* et d'*auto-adaptation* des paramètres d'un algorithme, qui reviendront tout au long de cette thèse, sera formalisée plus loin.

La dernière branche, apparue plus récemment, est la programmation génétique (GP pour *Genetic Programming*), introduite par J. Koza [93]. Le principe ici est de faire évoluer des structures d'arbre. La programmation génétique, à l'origine, est apparue pour faire évoluer des sous-programmes du langage LISP.

Ainsi, l'introduction des algorithmes évolutionnaires n'est pas récente, cependant seules des puissances de calcul croissantes ont permis leur réelle émergence à la fin des années 80.

Désormais, ces quatre branches constituent les algorithmes évolutionnaires. Un ouvrage de référence a longtemps été le livre de Z. Michalewicz [101], jusqu'à la parution récente du livre de A.E. Eiben and J.E. Smith [55].

Avantages et inconvénients de l'optimisation par algorithmes évolutionnaires Les avantages des algorithmes évolutionnaires pour l'optimisation continue sont de différentes natures. Tout d'abord, ils ne nécessitent pas d'autre information sur la fonction à optimiser que les valeurs de la fonction elle-même : ce sont des algorithmes d'ordre zéro. De plus ils sont robustes aux minima locaux : ce sont en effet, d'une part des algorithmes stochastiques et d'autre part des algorithmes à base de population. Ceci est une différence majeure par rapport aux méthodes d'optimisation déterministes classiques (algorithme de type gradient), qui ne rentrent pas dans le cadre de cette thèse, même si les propriétés locales des algorithmes évolutionnaires seront étudiées. Par ailleurs, les algorithmes évolutionnaires sont aussi, en tant qu'algorithmes stochastiques, robustes au bruit. En outre, en tant qu'algorithme à base de population, leur parallélisation est aisée : il suffit de distribuer l'étape d'évaluation de la fitness.

Les algorithmes évolutionnaires possèdent une autre particularité. Un même algorithme peut être appliqué à de très larges classes de problèmes. Plus précisément, une fois définis les opérateurs de croisement et mutation adaptés au problème à résoudre, un même schéma d'évolution peut être appliqué pour la minimisation de problèmes très différents. Le temps de développement d'une méthode évolutionnaire pour un problème nouveau est donc facile par rapport, par exemple, à la mise en oeuvre d'une méthode de différentiation automatique, ou au fait de poser et résoudre le problème adjoint pour le calcul d'un gradient. La contrepartie est que, lorsque une méthode d'optimisation *ad hoc* à un problème donné existe, les performances de cet algorithme seront en général meilleures que celles d'un algorithme évolutionnaire. Notons tout de même que des techniques plus spécifiques pour améliorer les performances sur certaines classes de fonctions sont développées à l'intérieur même de la communauté évolutionnaire, illustrant le fait que le coût à payer pour améliorer les performances dans des cadres précis est un coût de développement. C'est ce que nous verrons dans le cadre de cette thèse, dans le cas de l'optimisation continue.

Ainsi, leur robustesse au bruit, aux minima locaux, la relative simplicité de leur mise en place, dû au fait qu'aucune hypothèse spécifique sur la fonction à optimiser n'est requise, et l'émergence de puissances de calcul parallèle font que les algorithmes évolutionnaires sont devenus de plus en plus populaires ces dernières années. Le nombre croissant d'applications à des problèmes issus de divers domaines témoignent de cette popularité.

Mais le principal inconvénient des algorithmes évolutionnaires est leur coût. Ils nécessitent en effet un grand nombre d'évaluations de la fonction fitness : c'est la contrepartie du fait qu'ils

ne requièrent aucune autre information sur la fonction, et du fait qu'ils s'appliquent sur de très larges classes de problèmes. Par ailleurs, l'utilisateur doit généralement régler de nombreux paramètres pour chaque nouveau problème. Le manque de connaissances théoriques sur la manière d'effectuer ces réglages, ou plus généralement le manque de compréhension de leur fonctionnement rend ces réglages nécessairement expérimentaux et fastidieux. Un objectif de cette thèse est de remédier à ces inconvénients dans le cadre de l'optimisation continue.

Avant d'énoncer plus précisément nos objectifs, ainsi que le plan de cette thèse, intéressons-nous au point clé de la recherche d'un optimum par tout algorithme stochastique, à savoir l'équilibre entre exploration et exploitation, ainsi qu'aux mécanismes spécifiques introduits pour adapter les divers paramètres d'un algorithme à ces deux phases et faciliter par la même occasion le réglage de certains de ces paramètres.

Le dilemme exploration / exploitation Un point essentiel dans la recherche d'un optimum global par tout algorithme d'optimisation globale, mais par les algorithmes évolutionnaires en particulier, est l'équilibre entre l'*exploration* de régions de l'espace de recherche non encore visitées et l'*exploitation* des individus performants rencontrés jusque-là, autour desquels se trouvent peut-être des individus encore meilleurs. Une façon de gérer cet équilibre est d'essayer de privilégier *a priori* l'exploration dans la phase initiale de la recherche, et l'exploitation lors de la phase finale. C'est ce qui est par exemple fait dans un algorithme de recuit simulé classique, où le schéma de décroissance de la température, fixé à l'avance, détermine un haut niveau d'exploration dans les premières itérations de l'algorithme, puis fait décroître ce niveau avec la température.

Adaptation et auto-adaptation Un mécanisme idéal pour réguler exploration et exploitation serait un mécanisme capable de détecter à quel moment il faut modifier les importances relatives entre exploration et exploitation, et comment modifier les paramètres de l'algorithme pour arriver au point d'équilibre souhaité.

Ces deux points ont orienté les recherches dans le domaine évolutionnaire ces dernières années et ont motivé l'introduction d'opérateurs et d'algorithmes spécifiques. Plus généralement, la question de l'équilibre entre *exploitation* et *exploration* a également été centrale dans d'autres domaines. Par exemple, des techniques spécifiques pour adapter les paramètres d'un algorithme de recuit simulé aux deux phases ont été introduites dans [82, 83]. De même, afin

d'améliorer l'exploration dans les méthodes de *Monte Carlo Markov Chains* (MCMC), des méthodes adaptatives ont été proposées dans [71].

Les premières méthodes proposées pour l'adaptation automatique des paramètres d'un algorithme évolutionnaire qui seront détaillées dans le chapitre 1, l'ont été dans la branche des stratégies d'évolutions, algorithmes pour lesquels, à l'origine, seul l'opérateur de mutation était présent. L'enjeu a été (et est toujours) de contrôler les différents paramètres de la mutation gaussienne (essentiellement sa matrice de covariance).

Un algorithme présentant des caractéristiques lui permettant de détecter un changement entre les phases d'exploration et d'exploitation et d'adapter ses paramètres à chacune de ces phases porte désormais le nom d'algorithme *adaptatif* ou *auto-adaptatif*. Même si à l'heure actuelle, la signification exacte de ces deux termes ne fait pas l'unanimité dans la communauté des algorithmes évolutionnaires, tout au long de cette thèse, nous appellerons algorithme *adaptatif* un algorithme capable de modifier de façon automatique certain de ses paramètres pour s'adapter aux phases d'exploitation et d'exploration. La technique spécifique introduite pour cette adaptation peut avoir diverses origines. Par exemple, l'utilisation explicite du nombre de mutations réussies au cours d'une génération comme indicateur pour décider d'augmenter ou de diminuer le pas de la mutation est un mécanisme adaptatif. Par opposition, un algorithme qui diminue le pas de la mutation de manière pré-définie par l'utilisateur (comme dans le recuit simulé) ne sera pas qualifié d'adaptatif.

Une sous classe importante des algorithmes adaptatifs est constituée des algorithmes *auto-adaptatifs*, pour lesquels le mécanisme d'adaptation est implicite, et n'est pas fondé sur un mécanisme déterministe.

Les algorithmes adaptatifs et auto-adaptatifs seront centraux dans cette thèse. D'une part des contributions numériques seront abordées dans la première et troisième partie de cette thèse et d'autre part des contributions théoriques pour une meilleure compréhension de ces algorithmes seront présentées dans la deuxième partie.

Résultats théoriques Malgré la popularité croissante des algorithmes évolutionnaires, il existe à l'heure actuelle peu de résultats mathématiques rigoureux à leur sujet. A cela deux raisons principales: d'une part, historiquement les algorithmes évolutionnaires ont été introduits par des ingénieurs, mettant plus l'accent sur des résultats pratiques. Encore à l'heure actuelle, le développement des algorithmes évolutionnaires se fait généralement en suivant une démarche intuitive, empirique et pragmatique, accompagnée d'approches heu-

ristiques pour justifier la démarche mais ne se fondant pas sur des résultats mathématiques préalables. D'autre part, même en se plaçant dans des cadres *a priori* très simples, l'analyse mathématique des algorithmes évolutionnaires se révèle être très difficile.

Cependant, dans le cadre des algorithmes génétiques, pour un espace de recherche binaire, un certain nombre de résultats rigoureux ont été obtenus dans divers contextes simplifiés.

Tout d'abord, les premiers résultats de convergence asymptotique, lorsque l'intensité de la sélection tend vers l'infini et le taux de mutation tend vers 0, ont été obtenus par R. Cerf [36, 35], dans le cadre de la théorie de Freidlin-Wentzell. Avec le même type d'approche, O. François [63] montre la convergence asymptotique d'un algorithme évolutionnaire appelé *algorithme d'exploration / sélection*. Ensuite des résultats asymptotiques lorsque la taille de la population tend vers l'infini ont été obtenus par P. Del Moral [103]. Les propriétés des algorithmes évolutionnaires en population infinie ont également été étudiées par M. Vose [149].

Enfin les premiers résultats de complexité des algorithmes génétiques ont été obtenus par J. Garnier, L. Kallel, et M. Schoenauer [65], dans le cadre simplifié de la fonction *onemax*¹. S. Droste, T. Jansen et I. Wegener [51] ont généralisé ce résultat. Plus généralement l'équipe de I. Wegener, à Dortmund, appartenant à la communauté informatique de la complexité, étudie intensivement la question de la complexité des algorithmes génétiques.

Dans le cadre nous intéressant ici, celui des algorithmes sur un espace continu, beaucoup moins de résultats théoriques existent. Un état de l'art détaillé est l'objet du chapitre 3 qui précède l'exposé des contributions théoriques de cette thèse.

Objectifs Notre objectif principal est de contribuer à la compréhension des algorithmes évolutionnaires pour l'optimisation continue, et plus particulièrement aux méthodes d'adaptations introduites pour contrôler les différents paramètres dans le but de s'adapter aux différentes phases de recherche. Pour cela, nous allons adopter plusieurs approches. Tout d'abord une approche algorithmique, guidée par notre expérience des problèmes réels (dont un est décrit dans la troisième partie de cette thèse), sera validée sur des fonctions tests classiques pour l'optimisation globale. Ensuite une approche théorique dans laquelle des techniques classiques pour l'analyse d'algorithmes stochastiques seront utilisées pour étudier la convergence d'algorithmes adaptatifs.

1. Comptant le nombre de 1 dans une chaîne de bit

Plan de la thèse Ce travail de thèse a entre autre débuté par l’application d’un algorithme évolutionnaire à un problème de contrôle dans le domaine de la chimie quantique. A l’époque les algorithmes que nous pensions être les meilleurs en terme d’optimisation continue étaient les algorithmes à mutation auto-adaptative SA-ES (qui seront précisés au cours de cette thèse). Ce sont donc ces algorithmes que nous avons utilisés pour cette application. Ceci a motivé le reste du travail présenté ici. Les problèmes rencontrés lors de cette application, à savoir, la convergence prématurée de certains essais, la non optimalité de certains des paramètres de l’algorithme, ainsi que le nombre important de points évalués puis jetés sans être utilisés, a motivé la première partie de ce manuscrit : tout d’abord l’étude d’un algorithme plus récent, moins connu, l’algorithme CMA et ensuite le développement d’une technique pour utiliser plus efficacement l’information des points traditionnellement jetés. Ensuite, les contributions théoriques portent sur les algorithmes adaptatifs et en particulier sur l’algorithme SA-ES utilisé pour le problème de contrôle. Au final, trois parties constituent cette thèse.

La première partie comprend deux chapitres. Le chapitre 1 est consacré à un état de l’art des principaux algorithmes évolutionnaires utilisés en pratique, dont les algorithmes SA-ES et CMA, ainsi qu’à l’introduction des différents concepts sous-jacents à ces algorithmes. Le chapitre 2 présente une méthode explicite pour l’adaptation de la matrice de covariance, permettant, sur certaines classes de fonctions, d’améliorer les performances expérimentales de l’algorithme. La robustesse au bruit de la méthode proposée est par ailleurs étudiée.

La deuxième partie (chapitre 3, 4 et 5) présente des résultats théoriques de convergence d’algorithmes adaptatifs et auto-adaptatifs. Le chapitre 3 présente un état de l’art des résultats théoriques existant concernant les algorithmes évolutionnaires continus, et résume les résultats qui seront démontrés dans les deux chapitres suivants. Pour ces études nous nous restreignons à des classes de fonctions bien particulières. Dans le chapitre 4, nous étudions, la convergence d’algorithmes adaptatifs pour des fonctions de régularité C^2 , sous des hypothèses s’apparentant à de la faible convexité. Ensuite au chapitre 5, nous étudions sur la fonction sphère ($f(x) = \|x\|^2$) la convergence de l’algorithme auto-adaptatif “historique” qui a en particulier été utilisé dans la partie appliquée de cette thèse (Partie 3). Bien sûr, se placer sur une telle classe de fonctions peut paraître très restrictif puisque les algorithmes évolutionnaires sont des algorithmes d’optimisation globale. Cependant, dans la perspective du dilemme *exploration / exploitation*, l’étude de la convergence sur des fonctions convexes

constitue l'étude de cette phase d'exploitation, et plus précisément de l'adaptation des paramètres pour la phase d'exploitation.

Ces deux analyses font appel à des techniques probabilistes classiques pour l'analyse d'algorithme stochastique [53, 52, 21], théorie des martingales pour le chapitre 4, et chaînes de Markov à temps discret et espace d'état continu pour le chapitre 5.

La troisième partie expose l'application d'un algorithme évolutionnaire à un problème de contrôle en physique quantique, pour lequel une approche traditionnelle (un algorithme de quasi Newton avec calcul du gradient par méthode de différentiation automatique) avait été testée dans des travaux précédents. La fonction à optimiser s'est révélée présenter trop de minima locaux pour être optimisée efficacement par la méthode déterministe. La mise en oeuvre d'un algorithme évolutionnaire a permis d'obtenir des résultats bien au delà du champ d'application de la méthode déterministe. Par ailleurs, l'application pratique d'un algorithme évolutionnaire, plus précisément d'un algorithme à mutation auto-adaptative (SA-ES) a, entre autre, permis de soulever certains problèmes intrinsèques de cet algorithme, guidant, comme nous l'avons évoqué plus haut les deux autres parties de cette thèse.

Première partie

Algorithmes évolutionnaires pour l'optimisation continue

Chapitre 1

Etat de l'art

1.1 Introduction

Ce chapitre présente un état de l'art des algorithmes évolutionnaires pour l'optimisation continue de problèmes du monde réel, dont la popularité croissante est perceptible par le nombre important d'applications de ces algorithmes dans divers domaines (voir par exemple [138, 11, 155, 148, 28, 106]). Le point de vue adopté pour cette présentation est le même que celui des concepteurs de ces algorithmes, suivant une démarche intuitive et empirique. Notons cependant que quelques algorithmes présentés ici feront l'objet d'une étude théorique dans la deuxième partie de cette thèse.

Un enjeu majeur des algorithmes évolutionnaires depuis leur introduction, est le contrôle des différents paramètres de l'algorithme dans le but de réguler les phases d'exploration et d'exploitation. Ce contrôle des paramètres est apparu sous différentes formes. Historiquement la première méthode de contrôle des paramètres est la règle des 1/5 introduite par Rechenberg [122] et détaillée à la Section 1.3.1, où le taux de mutations réussies est utilisé pour mettre à jour l'écart-type d'un opérateur de mutation gaussien. En même temps que l'introduction de ces techniques spécifiques est apparue une terminologie consacrée : les algorithmes faisant usage de méthodes explicites ou implicites pour l'adaptation aux différentes phases (exploration et exploitation) ont été appelés algorithmes *adaptatifs* et *auto-adaptatifs*. Pour préciser ce qui est généralement entendu par méthode explicite ou implicite, rappelons le contre exemple donné en introduction du recuit simulé où le schéma de température fixé au départ détermine quelle sera la part d'exploration et d'exploitation dans l'algorithme. Cette méthode n'entre pas dans ce que nous appellerons adaptation, les modifications étant déterminées extérieurement une bonne fois pour toute. Inversement, la règle des 1/5 est une méthode adaptative explicite. D'autres exemples de ce que sont ces méthodes explicites ou implicites seront donnés tout au long du chapitre, puisque les algorithmes évolutionnaires les plus performants en pratique et présentés pour cette raison dans ce chapitre s'avèrent utiliser ces méthodes.

A l'heure actuelle, la définition exacte des termes *algorithmes adaptatifs* et *algorithmes auto-adaptatifs* ne fait pas l'unanimité dans la communauté évolutionnaire et nous suivons les définitions données dans l'article [54]. Un algorithme *adaptatif* utilise une méthode explicite pour le contrôle des paramètres de l'algorithme (par exemple la règle des 1/5 évoquée plus haut) et un algorithme *auto-adaptatif* une méthode implicite pour le contrôle des paramètres. Deb et Beyer [42] mettent en évidence l'adaptation d'un algorithme à travers plu-

sieurs expériences numériques, comme des tests où le paysage de la fonction à optimiser change au cours de l'évolution, en particulier la localisation de l'optimum change. L'intérêt d'étudier une telle fonction où un algorithme idéal doit faire alterner une phase d'exploration et une phase d'exploitation, est justement de tester les capacités des algorithmes à détecter le changement de phase et s'adapter aux différentes phases. Nous reviendrons sur ce point en fin de chapitre pour illustrer les propriétés d'adaptation des algorithmes présentés.

Le problème Dans tout ce chapitre nous allons considérer le problème de *minimisation*¹ d'une fonction f , dite fonction *fitness*, définie d'un sous ensemble \mathcal{X} de \mathbb{R}^d ($d \in \mathbb{N}_*$) dans \mathbb{R} :

$$\min_{x \in \mathcal{X}} f(x). \quad (1.1.1)$$

Par souci de clarté tout l'exposé sera fait avec $\mathcal{X} = \mathbb{R}^d$.

Le plan de ce chapitre est le suivant. Le contexte des algorithmes évolutionnaires est celui des algorithmes stochastiques et celui des algorithmes d'optimisation globale (ces deux notions n'étant bien entendu pas disjointes), un rapide résumé de cet ensemble de méthodes sera donné à la Section 1.2. La section 1.3 présente ensuite des algorithmes appartenant à la famille des Stratégies d'Evolution. Plus précisément trois techniques pour le contrôle des paramètres de la mutation sont exposées. En section 1.4 sont présentés deux algorithmes implicitement auto-adaptatifs, l'algorithme Differential Evolution (DE) et croisement spécifique pour les algorithmes génétiques (croisement SBX). La section 1.5 est consacrée à une branche plus récente, branche des algorithmes à Evolution de Distribution (EDA). Nous montrerons à la Section 1.6 comment Deb et Beyer [42] mettent en évidence les propriétés d'adaptation de l'algorithme SBX. Enfin la section 1.7 s'intéresse aux comparaisons qui existent dans la littérature entre ces différents algorithmes.

1.2 Le contexte : optimisation stochastiques et optimisation globale.

Le contexte des algorithmes évolutionnaires est celui des algorithmes d'optimisation stochastiques et des algorithmes d'optimisation globale. Par algorithme d'optimisation globale, il est généralement entendu algorithmes relativement robustes aux minima locaux par opposition aux algorithmes d'optimisation locale, par exemple de type gradient (ou Newton ou quasi Newton), dont on sait qu'ils ne recherchent que le minimum local le plus proche du point initial. Pour ces méthodes qui sortent du cadre de cette thèse, on pourra consulter les

1. Sans perte de généralité puisque minimiser f c'est maximiser $-f$

ouvrages de référence [25, 39]. Dans cette section plusieurs algorithmes pour l'optimisation globales, stochastiques ou non sont présentés.

L'algorithme de Recuit Simulé Le principe de l'algorithme de Recuit Simulé [92, 2, 1] est de générer une suite de variables aléatoires $(X_n)_{n \geq 0}$ comme suit : si à l'étape n on a $X_n = x$, on génère aléatoirement un nouveau point y tiré selon une *loi de proposition* $p(\cdot, x)$. On calcule $f(y)$. Si $f(y) < f(x)$ (le point que l'on vient de trouver est meilleur) alors $X_{n+1} = y$ et si $f(y) > f(x)$, pour éviter de rester piégé dans les minima locaux on va accepter le nouveau point avec une probabilité égale à

$$\exp\left(\frac{f(y) - f(x)}{T_n}\right)$$

où T_n est la *température*, que l'on va faire décroître au cours des itérations (c'est à dire que l'on accepte de moins en moins de mauvais points). Ce principe de sélection est aussi appelé sélection de Boltzmann.

On retrouve posé pour les algorithmes de Recuit Simulé le problème de l'adaptation aux différentes phases de la recherche. Celle ci concerne le schéma de décroissance en température ainsi que la loi de proposition p qui sont par ailleurs étroitement liés. Dans [97] se trouve un résumé des différentes techniques introduites pour cette adaptation. Pour l'adaptation de la loi de proposition, les méthodes plus précisément introduites dans [147] sont assez semblables aux méthodes dites *dérandomisées* que nous allons présenter dans ce chapitre.

L'algorithme de recuit simulé le plus utilisé en pratique est l'algorithme ASA (*Adaptive Simulated Annealing*) [82, 83]. Dans ASA, les densités de probabilité pour la loi de proposition sont différentes dans chacune des directions, et dépendent de la température. On retrouve donc des solutions analogues à celles proposées dans le cadre évolutionnaire. Cependant, quoique reconnu comme l'un des algorithmes de recuit les plus performants, il est également considéré comme très délicat à régler – le prix à payer pour les nombreuses possibilités qu'il offre étant un très grand nombre de paramètres.

Notons enfin que l'algorithme du recuit simulé peut également être vu comme un algorithme évolutionnaire avec une population de taille 1, un unique opérateur de mutation, qui consiste à choisir un voisin du point courant, et une procédure de sélection de Boltzmann.

Le cas particulier d'un algorithme de recuit simulé à température constante correspond à l'algorithme dit de Métropolis [99], appartenant à la famille plus générale des Méthodes de Monte Carlo par chaîne de Markov (Monte Carlo Markov Chains, MCMC). Le but de ces

méthodes est de simuler un loi de probabilité (par exemple pour approcher une espérance). Pour cela une chaîne de Markov, admettant pour mesure invariante la loi que l'on cherche à approcher, est simulée. Dans ces méthodes aussi, la notion d'adaptation pour essayer d'échantillonner au mieux la mesure invariante existe. Les premières méthodes adaptatives ont été introduites par H. Haario, H. Saksman et J. Tamminen en 2001 pour l'algorithme de Métropolis [71].

Méthode de surface de réponse Le principe des méthodes dites de *surface de réponse* consiste à construire un modèle de fonction à optimiser et à optimiser le modèle approché. Introduits pour la première fois dans [29], les algorithmes les plus récents utilisant cette technique sont décrits dans [86, 87, 70].

Méthode de lissage local (en anglais *smoothing techniques*). Le principe de base est de lisser la fonction que l'on cherche à optimiser, et ensuite d'optimiser la fonction lissée avec un algorithme classique de recherche locale. Ce lissage peut être itératif auquel cas la fonction est d'autant plus lissée que les itérations augmentent et on recommence la recherche locale à partir du point où on s'est arrêté à l'itération précédente. Ces techniques ont en particulier été introduites pour l'étude de la conformation de molécules sous le nom de *diffusion equation method* dans [96, 118]. Dans ce cas, le lissage repose sur la résolution de l'équation de diffusion et ne peut être effectué que pour des petites tailles de problème, ou bien dans le cas où la fonction est séparable.

Plus récemment Addis et a.l. [3] présentent ce même genre d'idée, mais leur technique de lissage peut s'appliquer à des problèmes de plus grande dimension.

Couplage stochastique et recherche locale Afin d'accélérer la recherche d'optima par une méthode stochastique, on peut remplacer la simple étape d'évaluation de la fonction par une recherche locale. Cette recherche locale peut être effectuée, par exemple, par un algorithme déterministe de type gradient ou quasi-Newton si le gradient de la fonction est connu. Un tel couplage avec ces méthodes déterministes porte le nom de *X-restart* (par exemple BFGS-restart). Cette idée de couplage s'appelle aussi, dans le domaine de la chimie moléculaire, méthode de *basin hopping* [151], et a en particulier été introduite pour la minimisation du potentiel de Lennard-Jones dans [151].

Lorsque l'algorithme stochastique est un algorithme évolutionnaire, l'algorithme hybride

s'appelle aussi un *algorithme mémétique*, terme introduit par Moscato [104]. On trouvera un résumé récent sur les algorithmes mémétiques dans [55].

1.3 Les Stratégies d'Evolution (ES)

Pour la branche des stratégies d'évolutions, l'adaptation aux phases d'exploration et d'exploitation s'est essentiellement traduite par l'adaptation de l'opérateur de mutation², la plupart du temps gaussien (voir [153] pour des ESs avec opérateur de mutation non gaussien), c'est à dire, consistant à rajouter un vecteur gaussien de moyenne nulle au parent, afin de créer un nouvel individu. Plus précisément, étant donné X_n un parent à la génération n , un enfant est créé de la façon suivante:

$$X_n + \sigma \mathcal{N}(0, C)$$

où $\sigma \in \mathbb{R}_+^+$ que l'on appelle par abus de langage *écart-type de la mutation* ou encore *pas de la mutation*³, et où C est une matrice symétrique définie positive, qui est la *matrice de covariance de la mutation* (par abus de langage également). Notons que $\sigma \mathcal{N}(0, C)$ suit la même loi que $\mathcal{N}(0, \sigma^2 C)$, mais on découpe généralement *écart-type de la mutation* et *matrice de covariance de la mutation* parce que l'adaptation de ces deux paramètres se fait à des échelles de temps différentes (voir par exemple [77, Section 3.2]) (lié au fait que dans la matrice de covariance, il y a $\frac{d^2+d}{2}$ coefficients à ajuster et simplement un seul pour ajuster le *pas* de la mutation).

La notation employée pour un algorithme ES est soit (μ, λ) -ES, soit $(\mu + \lambda)$ -ES, où μ désigne le nombre de parents et λ le nombre d'enfants générés à partir de ces parents. La notation “,” désigne que l'étape de remplacement va garder parmi les λ enfants, les μ meilleurs pour constituer la génération suivante. Notons que dans ce cas là, on aura obligatoirement $\mu \leq \lambda$. La notation “+” désigne que parmi les λ enfants **plus** les μ parents, on va garder les μ meilleurs pour constituer la génération suivante. La particularité des ESs par rapport au schéma général donné en introduction est qu'il n'y a pas d'étape de sélection. Ainsi le schéma général d'un $(1 \dagger \lambda)$ -ES est le suivant

2. En fait, au début, seul l'opérateur de mutation était présent dans les ESs, expliquant le fait que les recherches se sont concentrées sur cet opérateur

3. Traduction de l'anglais *step-size*

Algorithme $(1 \dagger \lambda)$ -ES On initialise en tirant aléatoirement un point dans l'espace de recherche: X_0

1. On génère λ enfants par mutation:

$$X_n^i = X_n + \sigma \mathcal{N}^i(0, C) \text{ pour } 1 \leq i \leq \lambda$$

2. Stratégie “,” : On choisit le meilleur parmi les λ enfants

$$X_{n+1} = \arg \min \{f(X_n^i), 1 \leq i \leq \lambda\}$$

Stratégie “+” : On choisit le meilleur parmi les λ enfants et le parent

$$X_{n+1} = \arg \min \{f(X_n), f(X_n^i), 1 \leq i \leq \lambda\}$$

Le schéma d'un $(\mu \dagger \lambda)$ -ES se déduit du schéma précédent : pour générer λ enfants, on applique l'étape 1 à partir des μ parents choisis les uns après les autres autant de fois qu'il faut pour créer λ enfants.

Se voulant générale, la définition de l'algorithme $(1 \dagger \lambda)$ -ES, ne précise pas ce que sont σ et C qui vont être ajustés au cours de l'évolution, trois méthodes pour cet ajustement vont être présentées dans cette fin de section consacrée aux ESs : la règle des 1/5, les méthodes de mutation des paramètres de la mutation, dites *mutations auto-adaptatives* et enfin les approches dérandomisées.

1.3.1 Premières idées d'adaptation dans les ESs : la règle des 1/5

Cette méthode est la première méthode d'adaptation apparue et elle est donc ici citée plus pour sa teneur historique que parce qu'elle est effectivement utilisée en pratique. Elle a été introduite par Rechenberg en 1973 [122]. Pour cette méthode la mutation considérée est isotrope, c'est à dire de matrice de covariance identité ($C = I_d$) et le seul paramètre à adapter est donc l'écart-type de la mutation σ . Le principe de base de cette méthode consiste à augmenter l'écart-type de la mutation lorsque “trop” de mutations sont réussies, c'est à dire lorsque “trop” d'enfants ont une *fitness* plus basse que celle des parents et réciproquement de diminuer l'écart-type lorsque pas assez de mutations sont réussies.

L'intuition derrière cette méthode est de dire que si trop de mutations sont réussies c'est que l'on est en train de faire une recherche trop locale et donc il faut augmenter l'écart-type

de la mutation pour “aller voir plus loin”. A l'inverse si on a peu de mutations réussies c'est que, éventuellement on est dans une phase de recherche trop exploratoire et il faut donc diminuer l'écart-type.

Concrètement, cela donne la règle suivante, traditionnellement utilisée sur un $(1+1)$ -ES, et obtenue à partir d'approximations pour des grandes dimensions faites sur 2 fonctions particulières (fonctions sphère et corridor) utilisant la théorie du *progress rate* (cf Chapitre 3). On commence par se fixer un temps d'observation T (ce temps d'observation correspond à un nombre fixé d'itérations) et, toutes les T générations, on calcule le taux τ de mutations réussies durant les T dernières générations. Si ce τ est supérieur à 0.2 alors σ est augmenté d'un facteur 1.22 ($\sigma := 1.22\sigma$), sinon σ est diminué d'un facteur 0.83 ($\sigma := 0.83\sigma$).

Pour plus de détails sur cette règle, en particulier sur le choix des différents facteurs, nous renvoyons aux ouvrages [122, 131].

Dans la pratique cette règle est bien souvent prise en défaut. Tout d'abord les réglages des différents facteurs, prévus pour la fonction sphère ou corridor, se transposent mal à d'autres fonctions. Par ailleurs, une telle règle ne permet le réglage que d'un seul paramètre, l'écart-type de la mutation. Ainsi, contrairement aux méthodes SA-ES que nous verrons en section suivante, dans le cas d'une utilisation avec plusieurs parents et enfants cette règle ne permet pas le réglage de paramètres différenciés pour chaque individu.

1.3.2 Les SA-ES : Muter les paramètres de mutation

L'acronyme SA-ES, pour *Self-Adaptive ES*, a été introduit par Beyer [22]. Ce sont ces méthodes qui ont permis l'essor des ESs. Introduites par Schwefel [131] en 1981, l'idée sous-jacente à ces méthodes est de laisser l'évolution adapter les paramètres de la mutation. Pour cela, l'idée fondamentale est d'associer des paramètres de la mutation (on va voir plus précisément dans la suite quels sont ces paramètres) à chaque individu et de modifier ces paramètres par mutation également. L'étape de remplacement qui s'effectue à partir des valeurs de fitness des individus, va sélectionner les individus avec leurs paramètres de mutation. L'intuition derrière cette idée est que des individus ayant de “mauvais” paramètres de mutation ne vont pas survivre longtemps, puisque plusieurs mutations successives avec de “mauvais” paramètres, pas adaptés à la forme locale de la fonction à optimiser (par exemple petit σ pour fort gradient), seront inévitablement dépassés par des individus munis de “bons” paramètres. Notons que cet ajustement est gratuit, c'est une effet de bord de l'étape de remplacement.

Il existe plusieurs variantes de SA-ES, suivant le modèle choisi pour la matrice de co-

variance, modèle qui va imposer dans quelles directions la recherche va s'effectuer. Les trois modèles sont le modèle *isotrope*, un modèle anisotrope appelé modèle *non-isotrope* et le modèle *corrélé*. Ces modèles, ainsi que la description précise des mutations associées à ces modèles sont décrits ci-dessous.

SA-ES isotrope: Dans ce cas là, la matrice de covariance de la mutation est l'identité I_d . Pour ce modèle, un seul paramètre va évoluer avec chaque individu: l'écart-type σ . Soit X un parent et σ l'écart-type de la mutation associé à ce parent. Les deux étapes de la mutation sont les suivantes: on commence par muter l'écart-type du parent en le multipliant par une loi log-normale

$$\sigma := \sigma \exp(\tau \hat{\mathcal{N}}(0, 1)) \quad (1.3.2)$$

Ce nouvel écart type est ensuite utilisé pour muter le parent et créer un enfant :

$$X := X + \sigma \mathcal{N}(0, I_d). \quad (1.3.3)$$

Dans les deux équations précédentes, τ est une constante à régler par l'utilisateur, $\hat{\mathcal{N}}(0, 1)$ et $\mathcal{N}(0, I_d)$ sont respectivement des tirages indépendants de lois normales centrées réduites et de vecteur gaussien centré de matrice de covariance l'identité. Notons que comme la médiane d'une loi log-normale est 1, la multiplication de σ par une loi log-normale fait que la probabilité d'augmenter ou de diminuer σ à chaque étape est $\frac{1}{2}$. La valeur de τ préconisée par Schwefel [131] d'après une étude sur la fonction sphère que nous expliquerons brièvement dans le chapitre 3 est $\frac{1}{\sqrt{2n}}$.

SA-ES non-isotrope: Dans de nombreux cas cependant, les propriétés locales de la fonction nécessiteront des écart-types différents dans différentes directions. Par abus de langage nous emploierons également le terme de *topologie* de la fonction ou *paysage* de la fonction. C'est ce qui justifie l'introduction d'un modèle non-isotrope, dont la matrice de covariance est diagonale:

$$C = \begin{pmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \vdots & 0 & \sigma_d^2 \end{pmatrix}$$

Un vecteur $(\sigma_1, \dots, \sigma_d)$ d'écart-types dans chaque direction est donc attaché à chaque individu. Notons par ailleurs aucun écart-type global σ n'apparaît ici.

La mutation, en deux étapes a lieu de la manière suivante. Soit X un parent et $(\sigma_1, \dots, \sigma_d)$ son vecteur d'écart-type associé

$$\begin{cases} \sigma_i : = \sigma_i \exp(\tau' \mathcal{N}(0, 1)) \exp(\tau \mathcal{N}_i(0, 1)) \text{ pour } 1 \leq i \leq d \\ X_i : = X_i + \sigma_i \mathcal{N}'_i(0, 1) \text{ pour } 1 \leq i \leq d \end{cases} \quad (1.3.4)$$

où, $\mathcal{N}(0, 1)$, $\mathcal{N}_i(0, 1)$ et $\mathcal{N}'_i(0, 1)$ sont $2n + 1$ réalisations de variables aléatoires gaussiennes centrées réduites indépendantes. On notera que $\exp(\tau' \mathcal{N}(0, 1))$ est un facteur global pour la mutation des n écart-types σ_i , $1 \leq i \leq n$. Les valeurs recommandées par Schwefel pour les facteurs τ et τ' sont respectivement $\frac{1}{\sqrt{2n}}$ et $\frac{1}{\sqrt{2\sqrt{n}}}$.

SA-ES corrélé: Le dernier type d'algorithme SA-ES utilise une matrice de covariance pleine, pour prendre en compte le fait que la base canonique n'est pas toujours la *bonne* base. Par bonne base nous entendons par exemple pour une fonction f quadratique,

$$f(x) = \frac{1}{2} x^T H x$$

avec H symétrique définie positive, la base propre de H (qui en toute généralité n'est pas la base canonique). Dans la suite, le terme à *variables corrélées* désignera une fonction pour laquelle la base propre de la matrice hessienne n'est pas la base canonique. Le modèle prenant en compte une matrice de covariance complète est le *modèle corrélé*.

Formellement, la matrice de covariance pour le modèle corrélé s'écrit :

$$C_{(\vec{\sigma}, \vec{\alpha})} = \left(\prod_{i=1}^{d-1} \prod_{j=i+1}^d R(\alpha_{ij}) \right) \begin{pmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \vdots & 0 & \sigma_d^2 \end{pmatrix}$$

où les matrices $R(\alpha_{ij})$ sont définies par

$$R(\alpha_{ij}) = \begin{pmatrix} 1 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & \cdots & \cdots & 0 \\ \cdots & \cos \alpha_{ij} & \cdots & -\sin \alpha_{ij} & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \sin \alpha_{ij} & \cdots & \cos \alpha_{ij} & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & \cdots & 0 & 1 \end{pmatrix}$$

Notons que la matrice de covariance est représentée sous forme de produit d'une matrice diagonale (les termes diagonaux sont les variances suivant les directions propres) et de $d(d-1)/2$ matrices de rotations dans des plans définis par les couples de vecteurs de base parce qu'il serait très difficile de garantir la symétrie et la positivité de la matrice après mutation de ses termes. Par ailleurs, comme toute matrice symétrique définie positive peut se représenter de la sorte, il n'y a aucune perte de généralité dans le modèle considéré.

La mutation est alors similaire au cas non-isotrope, auquel il faut bien entendu ajouter la mutation (gaussienne) des angles α_{ij}

$$\begin{cases} \sigma_i : = \sigma_i \exp(\tau' \mathcal{N}(0, 1)) \exp(\tau \mathcal{N}_i(0, 1)) \text{ pour } 1 \leq i \leq n \\ \alpha_{ij} : = \alpha_{ij} + \beta \mathcal{N}_{ij}(0, 1) \text{ pour } 1 \leq i < j \leq n \\ X : = X + \mathcal{N}(0, C_{(\vec{\sigma}, \vec{\alpha})}) \end{cases}$$

où $\mathcal{N}(0, 1)$, $\mathcal{N}_i(0, 1)$ et $\mathcal{N}_{ij}(0, 1)$ sont des tirages de normales centrées réduites indépendantes et $\mathcal{N}(0, C_{(\vec{\sigma}, \vec{\alpha})})$ est un tirage de vecteur gaussien de matrice de covariance $C_{(\vec{\sigma}, \vec{\alpha})}$. Les valeurs α_{ij} sont prisent dans l'intervalle $[-\pi, \pi]$, les valeurs de τ et τ' sont celles de la mutation non-isotrope, et la valeur préconisée par Schwefel pour β est d'environ 5 degrés.

Même si ce modèle peut générer les matrices de covariance les plus générales, il a été montré dans [74] que cette technique de mutation n'est pas invariante par transformation linéaire. Plus précisément l'algorithme va présenter des performances très différentes sur une fonction

$$x \rightarrow f(x)$$

ou sur la même fonction où on l'on applique en plus une transformation orthogonale à savoir

$$x \rightarrow f(Px),$$

où P est une matrice orthogonale.

1.3.2.1 Ajout d'un opérateur de croisement

En pratique les performances des ESs que nous venons de voir sont améliorées dès que l'on rajoute un opérateur de croisement, en particulier sur des fonctions multimodales [131, 153]. Notons que le terme opérateur de croisement est employé ici même si on ne croise pas vraiment deux individus de la population (à l'origine un opérateur de croisement est un opérateur de $\mathcal{X} \times \mathcal{X} \rightarrow \mathcal{X} \times \mathcal{X}$) Cependant, plusieurs types d'opérateurs de recombinaison ont été proposés par Schwefel [131]. Notons X^n et Y^n deux vecteurs de \mathbb{R}^d représentant deux parents

à la génération n , choisis de manière uniforme dans la population. Les différents types de croisements proposés sont:

- **Croisement discret**, pour $1 \leq i \leq d$

$$X_i^n \rightarrow \begin{cases} X_i^n & \text{avec probabilité } \frac{1}{2} \\ Y_i^n & \text{avec probabilité } \frac{1}{2} \end{cases}$$

où $U_i[0, 1]$ sont d variables aléatoires uniformes tirées entre 0 et 1 indépendantes.

- **Croisement intermédiaire**, pour $1 \leq i \leq d$

$$X_i^n \rightarrow (\alpha X_i^n + (1 - \alpha)Y_i^n) \quad \text{où } \alpha \sim U[0, 1]$$

- **Croisement globale discret ou intermédiaire**. Les opérateurs sont les mêmes que les opérateurs discrets ou respectivement intermédiaires à la différence que pour chaque coordonnée, un vecteur nouveau $(Y^n)_{\text{rand}}$ est choisi.

Les mêmes types de d'opérateurs de croisement sont utilisés sur les paramètres de mutation.

1.3.2.2 Avantages et inconvénients des SA-ES

Outre le fait que les modèles de mutations ici sont plus riches que le modèle permis par la règle des $1/5$, le point fort de ces méthodes est que la *compétition*⁴ entre les individus permet d'inclure un peu plus d'information sur la topologie de la fonction optimisée. D'autant plus d'information sur la topologie de la fonction que l'on permet des corrélations entre les variables.

Cependant, il existe plusieurs limitations à la technique d'adaptation des paramètres que nous venons de présenter. Tout d'abord, comme souligné dans [131, 115], des tailles de population relativement grandes sont nécessaires pour l'ajustement des paramètres, parce que cet ajustement repose sur un tirage stochastique des nouveaux paramètres de mutation. Ceci est d'autant plus vrai que la complexité du modèle pour la matrice de covariance augmente. Ceci est en particulier vrai par exemple sur une fonction elliptique (unimodale), alors que le fait que l'on ait qu'un seul minimum global fait que l'on souhaiterait utiliser une petite taille de population.

4. Compétition au sens les meilleurs individus survivent.

Par ailleurs, dans cette méthode, les paramètres que l'on souhaite apprendre ne sont jamais explicitement mentionnés. Nous verrons par exemple au chapitre suivant que dans le cas d'une fonction elliptique les paramètres à apprendre, si on veut une convergence rapide, correspondent à l'inverse de la Hessienne de la fonction : rien ne guide l'algorithme SA-ES à apprendre ces paramètres et nous verrons d'ailleurs qu'il n'apprend pas vraiment ces valeurs.

Un autre inconvénient de cette technique de mutation, où deux tirages indépendants sont effectués pour muter un individu, est le fait que la sélection d'un individu venant d'être créé grâce à une grande ou petite variation ne conduit pas forcément à une variation dans le même sens du pas de la mutation. Par exemple, une variation de la distance entre l'enfant et le parent $\|X_{n+1} - X_n\|$ relativement grande peut avoir lieu malgré un petit tirage (proche de 0) de loi log-normale à cause d'un tirage de loi normale relativement grand. Ce problème est surtout significatif dans le cas non-isotrope et dans le cas corrélé, comme remarqué par Schwefel [133]. En effet dans le cas isotrope, la norme d'un vecteur gaussien varie très peu lorsque la dimension d augmente, et ainsi c'est le tirage de la loi log-normale qui détermine entièrement la longueur de la mutation.

En pratique, pour pallier ce problème il est préconisé d'augmenter la taille de la population. Pour la mutation non-isotrope, Ostermeier *et al.* [116] recommandent un taille de population de $10 * d$. Cependant, parmi les trois algorithmes présentés, Bäck *et al.* [15] recommandent l'utilisation de la mutation non-isotrope.

1.3.3 Les ES dérandomisés

Les points faibles des méthodes auto-adaptatives évoqués à la section ci-dessus ont motivé l'introduction de méthodes d'adaptation dérandomisées⁵ par opposition aux méthodes de la section précédente où l'ajustement des paramètres est stochastique. Plus récentes, ces méthodes sont à l'heure actuelle moins populaires, car moins connues, au sein même de la communauté évolutionnaire. Par exemple le livre de référence [55] ne les mentionne pas. Pourtant, comme nous le verrons à la Section 1.7, ces techniques sont parmi les plus performantes à l'heure actuelle.

5. traduction de l'anglais *derandomized*

1.3.3.1 Les approches dérandomisées : concept

Le premier article proposant une approche dérandomisée date pourtant de 1994 [115]. Dans ce papier, Ostermeier *et al.*, se placent dans le cadre d'un ES non-isotrope (avec un écart-type pour la mutation par variable) et proposent de remplacer le tirage d'une variable log-normale par l'utilisation de la variable aléatoire qui a été tirée pour muter le paramètre. C'est en ce sens que l'adaptation du paramètre de la mutation est *dérandomisée* : on enlève une composante stochastique pour se rapprocher d'une mise à jour déterministe.

Plus précisément, le couple d'équations (1.3.4) est remplacé par le processus suivant: Soit $X = (X_1, \dots, X_d)$ un parent avec son vecteur d'écart-types pour la mutation $(\sigma_1, \dots, \sigma_d)$. Un enfant est créé de la manière suivante

$$X_i : = X_i + \mathcal{B}(\alpha, 1/2)\sigma_i\mathcal{N}_i(0, 1) \text{ pour } 1 \leq i \leq d \quad (1.3.5)$$

où $\mathcal{N}(0, 1) = (\mathcal{N}_1(0, 1), \dots, \mathcal{N}_d(0, 1))$ est un vecteur gaussien de matrice de covariance l'identité et où $\mathcal{B}(\alpha, 1/2)$ désigne une variable aléatoire de loi $\frac{1}{2}\delta_\alpha + \frac{1}{2}\delta_{1/\alpha}$ avec $\alpha > 0$ (*i.e.* qui vaut α ou $1/\alpha$ avec probabilité $\frac{1}{2}$).

A l'étape de remplacement, les meilleurs individus sont sélectionnés. Soit X^* un tel individu et $\mathcal{N}^*(0, 1)$ la réalisation du vecteur gaussien qui a permis de le créer. Alors, les différents écart-types vont être adapté de la façon suivante

$$\sigma_i : = \sigma_i\mathcal{B}(\alpha, 1/2)^\beta \exp(|\mathcal{N}_i^*| - \sqrt{2/\pi})^{\beta_{\text{scal}}} \text{ pour } 1 \leq i \leq d \quad (1.3.6)$$

Détaillons un peu les différents termes de l'équation (1.3.6):

- Le terme $\exp(|\mathcal{N}_i^*| - \sqrt{2/\pi})$ traduit le fait que si la longueur de \mathcal{N}_i^* est plus grande que la valeur moyenne de la norme d'une variable aléatoire gaussienne centrée réduite ($\sqrt{2/\pi}$), alors le pas dans la i ème direction est augmenté. L'idée est de comparer la longueur du pas qui a été fait avec la longueur du pas qui aurait été fait si il n'y avait pas eu de sélection, et d'augmenter ou diminuer l'écart-type dans le même sens. On voit que cette formule, contrairement à la méthode de l'équation (1.3.4), permet une variation dans le même sens entre le pas qui a été fait par l'individu dans une direction et l'écart type de la mutation dans cette direction.
- La deuxième nouveauté de la mutation présentée ici est la présence des facteurs β et β_{scal} dans l'équation (1.3.6) afin d'atténuer la valeur sélectionnée et de ne pas lui

donner un poids trop important. Le souhait des auteurs étant que l'adaptation des paramètres puisse avoir lieu avec des tailles de populations relativement faibles, mais qu'en contrepartie l'adaptation se fasse en plusieurs générations. Ainsi, il est intuitif de donner un poids plus faible aux paramètres sélectionnés (la sélection ayant lieu sur un faible nombre de tirages). Dans [115] les résultats de cet algorithme ont été comparés à l'algorithme SA-ES non isotrope montrant une amélioration claire sur une large classe de fonctions elliptiques.

Les concepts introduits ci-dessus ont été réutilisés et améliorés par les mêmes auteurs dans l'algorithme CSA présenté dans la section suivante puis plus tard avec l'algorithme CMA présenté en Section 1.3.3.3. Par ailleurs Ostermeier, Hansen et Gawelczyk, dans les algorithmes qu'ils ont introduits, ont toujours souhaité travailler avec des petites tailles de population, se concentrant ainsi sur la phase d'exploitation, quitte à paralléliser ces algorithmes par la suite pour la phase d'exploration [73]. Pour cela, mis à part l'algorithme que nous venons de voir, ils ont toujours considéré des algorithmes dans lesquels il n'y a qu'un *seul* jeu de paramètres pour l'opérateur de mutation (*i.e.* un seul couple (σ, C)) qui est donc commun à toute la population et dont toute la population participe à la mise à jour.

Il a par ailleurs été montré par Scheel [129] que lorsque l'on travaille avec des petites tailles de population, il est préférable de tirer tous les enfants à partir d'un même point avec un même opérateur (on retrouve le jeu unique de paramètres pour la mutation pour toute la population) et qu'un choix astucieux pour ce point est un barycentre des parents:

$$\langle X \rangle = \sum_{i=1}^{\mu} w_i X_i \quad (1.3.7)$$

où (X_1, \dots, X_{μ}) sont les μ parents et $w_i \in [0, 1]$ sont des poids (qui généralement favorisent les meilleurs) tels que $\sum_1^{\mu} w_i = 1$. On utilisera aussi la notation

$$\langle X \rangle = \sum_{i=1}^{\mu} w_i X_{i:\lambda} \quad (1.3.8)$$

où la notation $X_{i:\lambda}$ désigne le i ème meilleur parmi les λ enfants (c'est-à-dire au sens de f), à ne pas confondre avec la notation usuelle pour les statistiques d'ordre.

Le nouveau cadre algorithmique avec un seul jeu de paramètres pour l'opérateur de mutation et utilisant un barycentre des parents est le suivant :

Algorithme $(\mu/\mu, \lambda)$ -ES On initialise en tirant aléatoirement μ points dans l'espace de recherche.

- On génère λ enfants par mutation:

$$X_n^i = \langle X \rangle^n + \sigma \mathcal{N}^i(0, C) \text{ pour } 1 \leq i \leq \lambda$$

- On calcule $\langle X \rangle^{n+1} = \sum_{i=1}^{\mu} w_i X_{i:\lambda}^n$

1.3.3.2 Cumulative Step-Size Adaptation (CSA)

L'idée ici est de regarder le chemin parcouru pour adapter les écart-types de la mutation (soit un écart-type global, soit un écart-type par direction). Plaçons nous dans le cas du $(\mu/\mu, \lambda)$ -ES. Soit c_n le chemin suivi par l'algorithme entre la génération $n - 1$ et n , c'est à dire

$$c_n = \langle X \rangle^n - \langle X \rangle^{n-1} .$$

Une façon simple d'utiliser deux chemins consécutifs c_n et c_{n+1} pour mettre à jour un écart-type global pour la génération suivante serait de regarder le signe du produit scalaire $c_n^T c_{n+1}$ et d'augmenter l'écart-type si celui ci est positif: durant deux itérations successives, l'algorithme a suivi la même direction et on aurait donc pu aller plus vite si l'écart-type global avait été plus grand. A l'inverse, si le signe du produit scalaire $c_n^T c_{n+1}$ est négatif, c'est que des directions opposées ont été suivies et que donc les écart-types sont trop élevés. Cette idée a par exemple été utilisée en 1990 dans un autre domaine, celui de la rétropropagation dans les réseaux de neurones par Silva et Almedia dans [136] où le signe du produit scalaire est utilisé pour régler le facteur d'apprentissage.

Pour les algorithmes évolutionnaires cette idée a été introduite en 1994 par Ostermeier *et al.* [116] sous le nom de CSA (*Cumulative Step Size Adaptation*) dont le principe est une généralisation de l'idée précédente à plus de deux chemins. Nous allons formaliser cela dans le cas d'un ES isotrope (avec un seul écart-type global) par souci de clarté mais notons que dans l'article [116], ce concept a été introduit pour adapter à la fois un écart-type global et des écart-types par variables. Par ailleurs dans l'algorithme CMA (*Covariance Matrix Adaptation*) décrit dans la section 1.3.3.3, l'écart-type global est réglé de cette manière. Le principe du CSA est le suivant [116, 73, 115]: Supposons qu'à la génération n , λ enfants sont générés de la façon suivante, pour $k = 1, \dots, \lambda$

$$X_k^n = \langle X \rangle^n + \sigma^n \mathcal{N}(0, I_d). \tag{1.3.9}$$

L'étape de sélection choisit les μ meilleurs avec lesquels on forme le nouveau barycentre $\langle X \rangle^{n+1}$. On calcule alors le vecteur

$$p_\sigma^{n+1} = (1 - c_\sigma)p_\sigma^n + \sqrt{c_\sigma(2 - c_\sigma)}\sqrt{\mu} \frac{\langle X \rangle^{n+1} - \langle X \rangle^n}{\sigma^n} \quad (1.3.10)$$

où $c_\sigma \in]0, 1]$ et $p_\sigma^0 = 0$. La mise à jour de l'écart-type proposée dans [116, 73, 115] est la suivante,

$$\sigma^{n+1} = \sigma^n \exp \left(\frac{1}{d_\sigma} \left(\frac{\|p_\sigma^{n+1}\|}{E(\|\mathcal{N}(0, I_d)\|)} - 1 \right) \right). \quad (1.3.11)$$

où d est un facteur d'atténuation comme β dans l'équation (1.3.6). En pratique, $E[\|\mathcal{N}(0, I_d)\|] = \sqrt{2}\Gamma(\frac{n+1}{2})/\Gamma(\frac{n}{2})$ est approché par $\sqrt{d}(1 - \frac{1}{4d} + \frac{1}{21d^2})$.

Pour comprendre d'où viennent ces équations, remarquons tout d'abord que les poids $(1 - c_\sigma)$ et $\sqrt{c_\sigma(2 - c_\sigma)}$ des différentes composantes de la somme dans l'équation (1.3.10) sont tels que si

$$p_\sigma^n \sim \mathcal{N}(0, I_d)$$

et

$$\sqrt{\mu} \frac{\langle X \rangle^{n+1} - \langle X \rangle^n}{\sigma^n} \sim \mathcal{N}(0, I_d)$$

et que de plus les vecteurs p_σ^n et $\sqrt{\mu} \frac{\langle X \rangle^{n+1} - \langle X \rangle^n}{\sigma^n}$ sont indépendants alors $p_\sigma^{n+1} \sim \mathcal{N}(0, I_d)$.

Ceci aura une importance dans la suite.

L'idée maintenant est de dire que si $\|p_\sigma^n\|$ est grand, c'est que les différents pas suivis par l'algorithme pointent dans la même direction, et que le même chemin aurait pu être suivi plus rapidement si l'écart type avait été plus élevé. De même si $\|p_\sigma^n\|$ est petit, c'est que ces pas se compensent et l'écart-type doit être réduit.

Pour spécifier ce qu'est "grand" et "petit" pour $\|p_\sigma^{n+1}\|$, il est raisonnable de faire l'hypothèse que si la sélection consistait non pas à prendre les μ meilleurs individus mais à choisir aléatoirement μ individus parmi λ ou bien simplement à prendre les λ enfants (pas de sélection du tout), *i.e.*

$$\langle X \rangle^{n+1} = \frac{1}{\lambda} \sum_{i=1}^{\lambda} x_i^n, \quad (1.3.12)$$

alors l'écart-type de la mutation devrait rester constant. Dans la terminologie introduite par la branche des Algorithmes Genetiques on dirait que l'on essaye de minimiser la dérive génétique [55]. Or d'après la théorème de la limite centrale, on voit que

$$\sqrt{\lambda} \frac{\frac{1}{\lambda} \sum_{i=1}^{\lambda} X_i^n - \langle X \rangle^n}{\sigma^n} \xrightarrow{\lambda \rightarrow \infty} \mathcal{L} \mathcal{N}(0, I_d). \quad (1.3.13)$$

On va ainsi supposer que la loi de $\sqrt{\lambda} \frac{1}{\lambda} \sum_{i=1}^{\lambda} X_i^n - \langle X \rangle^n$ est bien approchée par une loi normale centrée de matrice de covariance l'identité. Puisqu'il n'y a pas de sélection, p_σ^n et $\sqrt{\lambda} \frac{\langle X \rangle^{n+1} - \langle X \rangle^n}{\sigma^n}$ sont indépendants. En supposant de plus que $p_\sigma^n \sim \mathcal{N}(0, I_d)$, d'après la remarque faite sous l'équation (1.3.10), on voit alors que

$$p_\sigma^{n+1} \sim \mathcal{N}(0, I_d). \quad (1.3.14)$$

Ainsi, sans sélection, la loi de $\|p_\sigma^{n+1}\|^2$ peut être approché par une loi du Chi-deux à d degrés de liberté. Nous noterons dans la suite $\|p_\sigma^{n+1}\| \sim \|\mathcal{N}(0, I_d)\|$. Comme précisé précédemment, dans [116, 73, 115], Hansen *et al.* proposent une mise à jour de l'écart type global de façon à ne pas avoir de biais si il n'y a pas de sélection, plus précisément en prenant le logarithme de l'équation (1.3.11) on voit que

$$\ln(\sigma^{n+1}) = \ln(\sigma^n) + \frac{1}{d_\sigma} \left(\frac{\|p_\sigma^{n+1}\|}{E(\|\mathcal{N}(0, I_d)\|)} - 1 \right) \quad (1.3.15)$$

En conséquence, $\ln(\sigma^n)$ est sans biais dès que $\|p_\sigma^n\|$ est mis à jour alors qu'il n'y a pas de sélection, ce qui explique la mise à jour proposée dans l'équation 1.3.11. Les paramètres par défaut [116, 91] sont $c_\sigma = 10/(d+20)$ et $d_\sigma = \max(1, \frac{3\mu}{d+10}) + \frac{1}{c_\sigma}$, la valeur initiale pour p_σ est 0.

1.3.3.3 L'algorithme CMA

L'algorithme CMA est à l'heure actuelle la forme la plus aboutie d'algorithmes dérandomisés. La version présentée dans cette section est basé sur l'algorithme introduit en 2001 [77] qui a été significativement améliorée en 2003 [76], puis légèrement amélioré en 2004 [75]. Un nombre croissant d'application des stratégies d'évolutions pour l'optimisation continue utilise cet algorithme, plusieurs de ces applications sont référencées sur la page web de Nikolaus Hansen, auteur du CMA [72].

Comme le nom de l'algorithme le suggère, dans le CMA, le modèle de matrice de covariance qui évolue est une matrice de covariance pleine. Le schéma de base de l'algorithme est celui introduit à la section précédente pour l'algorithme CSA à savoir un $(\mu/\mu, \lambda)$ -ES. A chaque génération λ enfants suivant la loi

$$X_k^{n+1} \sim \mathcal{N}(\langle X \rangle_\mu^n, (\sigma^n)^2 C^n) \quad (1.3.16)$$

sont créés, où les notations suivantes sont utilisées

Notations

$X_k^{n+1} \in \mathbb{R}^d$, k-ième enfant à la génération $n + 1$.

$\langle X \rangle_\mu^n := \sum_{i=1}^\mu w_i X_{i:\lambda}^n$, $w_i \in \mathbb{R}^+$, moyenne pondérée des μ meilleurs individus de la génération n . L'index $i : \lambda$ dénote le i^{eme} meilleur individu.

$\sigma^n \in \mathbb{R}^+$, pas de la mutation à la génération n .

C^n , matrice symétrique définie positive de taille $d \times d$.

B^n , matrice de passage, orthogonale, de la matrice C^n .

D^n , racine de la matrice des valeurs propres de C^n (bien définie puisque C^n est définie positive).

Notons que la relation liant les 3 dernières matrices introduites est $C^n = B^n D^n (B^n D^n)^T$, où l'indice T désigne la transposée d'une matrice.

Notons que comme précédemment mentionné, l'adaptation de la matrice C^n et l'adaptation du pas global σ^n sont découplés parce que ces deux paramètres sont adaptés à des échelles de temps différentes (cf. [77]).

Adaptation du pas global σ^n L'adaptation de l'écart-type global de la mutation se fait en utilisant la technique CSA expliquée en section 1.3.3.2. Les équations diffèrent légèrement des équations (1.3.10) et (1.3.11) pour prendre en compte le fait qu'ici la matrice de covariance est adaptée et n'est plus l'identité. Cette adaptation a donc lieu en calculant le vecteur p_σ^{n+1} suivant

$$p_\sigma^{n+1} = (1 - c_\sigma) p_\sigma^n + \sqrt{\mu} \sqrt{c_\sigma (2 - c_\sigma)} (C^n)^{-\frac{1}{2}} \frac{\langle X \rangle_\mu^{n+1} - \langle X \rangle_\mu^n}{\sigma^n} \quad (1.3.17)$$

où $p_\sigma^0 = 0$ et où $(C^n)^{-\frac{1}{2}}$ dénote l'inverse de la racine de C^n . La multiplication par $(C^n)^{-1/2} = (B^n)^T (D^n)^{-1/2} B^n$ remet à l'échelle le pas $\langle X \rangle_\mu^{n+1} - \langle X \rangle_\mu^n$ dans la base propre de C^n donnée par les vecteurs colonnes de B^n et $c_\sigma \in]0, 1[$ détermine la mémoire que l'on souhaite avoir.

Ensuite, comme dans l'équation (1.3.11), la norme de $\|p_\sigma^{n+1}\|$ est comparée à la moyenne de la norme d'un vecteur gaussien de dimension d

$$\sigma^{n+1} = \sigma^n \exp \left(\frac{1}{d_\sigma} \left(\frac{\|p_\sigma^{n+1}\|}{E[\|\mathcal{N}(0, I_d)\|]} - 1 \right) \right) \quad (1.3.18)$$

où, $d_\sigma \geq 1$ est un facteur d'atténuation.

$E[\|\mathcal{N}(0, I_d)\|] = \sqrt{2} \Gamma(\frac{n+1}{2}) / \Gamma(\frac{n}{2})$ est approché en pratique par $\sqrt{d}(1 - \frac{1}{4d} + \frac{1}{21d^2})$.

Adaptation de la matrice C^n La matrice C^n est mise à jour à chaque génération en calculant le vecteur de \mathbb{R}^d p_c^{n+1} suivant

$$p_c^{n+1} = (1 - c_c)p_c^n + \sqrt{\mu}\sqrt{c_c(2 - c_c)} \frac{\langle X \rangle_\mu^{n+1} - \langle X \rangle_\mu^n}{\sigma^n}. \quad (1.3.19)$$

avec, $c_c \in]0, 1]$. Dans la version du CMA de [77], ce vecteur est utilisé pour rajouter la matrice de rang 1, $p_c^{n+1}p_c^{n+1T}$ à C^n . Plus précisément

$$C^{n+1} = (1 - c_{\text{cov}})C^n + c_{\text{cov}}p_c^{n+1}p_c^{n+1T} \quad (1.3.20)$$

avec, $c_{\text{cov}} \in]0, 1]$. Intuitivement, p_c^{n+1} est une direction de descente ($\frac{\langle X \rangle_\mu^{n+1} - \langle X \rangle_\mu^n}{\sigma^n}$ pouvant être vu comme une grossière approximation du gradient de la fonction), et la mise à jour se fait donc en rajoutant la matrice $p_c^{n+1}p_c^{n+1T}$, de rang un, dont la direction propre est donnée par p_c^{n+1} .

Dans la version du CMA présentée dans [76], la mise à jour s'effectue en rajoutant à la matrice C^n en plus de la matrice de rang 1 $p_c^{n+1}(p_c^{n+1})^T$, la matrice de rang μ^6 suivante

$$U^{n+1} = \sum_{i=1}^{\mu} \frac{(X_{i:\lambda} - \langle X \rangle_\mu^n)^T (X_{i:\lambda} - \langle X \rangle_\mu^n)}{(\sigma^n)^2}. \quad (1.3.21)$$

L'équation 1.3.20 devient

$$C^{n+1} = (1 - c_{\text{cov}})C^n + c_{\text{cov}}(\alpha_{\text{cov}}p_c^{n+1}p_c^{n+1T} + (1 - \alpha_{\text{cov}})U^{n+1}) \quad (1.3.22)$$

Réglages des paramètres Le réglage par défaut des paramètres du CMA fait l'objet d'une partie du papier [76], et les paramètres optimaux pour maximiser un taux de succès empirique sur des fonctions elliptiques sont bien approchés par les valeurs suivantes pour c_c , c_σ , d_σ et c_{cov} et $w_{i=1\dots\mu}$ sont

$$c_c = \frac{4}{d+4}, \quad c_\sigma = \frac{10}{d+20}, \quad d_\sigma = \max(1, \frac{3\mu}{d+10}) + \frac{1}{c_\sigma} \quad (1.3.23)$$

$$c_{\text{cov}} = \frac{1}{\mu} \frac{2}{(d+\sqrt{2})^2} + (1 - \frac{1}{\mu}) \min(1, \frac{2\mu-1}{(d+2)^2 + \mu}) \quad (1.3.24)$$

$$w_{i=1\dots\mu} = \frac{\ln(\mu+1) - \ln(i)}{\sum_{j=1}^{\mu} \ln(\mu+1) - \ln(j)}$$

Les valeurs initiales sont $p_\sigma^0 = 0$, $p_c^0 = 0$ et $C^0 = I_d$.

Le réglage du nombre de parents et d'enfants dépend quant à lui du problème, cependant les valeurs par défaut préconisées pour les problèmes elliptiques sont $\mu = 4 + \lfloor 3 \ln(d) \rfloor$ et $\mu = \lfloor \frac{\lambda}{2} \rfloor$.

6. En fait cette matrice est de rang μ presque sûrement

Le CMA : Illustration Durant cette thèse, l'algorithme CMA a été largement étudié dans le cadre notamment de l'algorithme qui sera présenté au chapitre 2. Etudions brièvement en pratique comment marche l'algorithme CMA. Tout d'abord la première question à se poser (cf. chapitre 2) est de savoir quelle est la matrice de covariance idéale. Dans le cas d'une fonction elliptique

$$f(x) = x^T H x \quad (1.3.25)$$

pour laquelle par souci de clarté nous prenons le minimum en zéro, nous verrons au chapitre 2 qu'une matrice pertinente à apprendre est $(\frac{1}{2}H)^{-1}$. Regardons maintenant sur la fonction elliptique la plus simple, à savoir la fonction sphère ($(\frac{1}{2}H)^{-1} = I_d$) quelle est la matrice apprise par le CMA. La figure 1.1 illustre le comportement de l'algorithme CMA sur la fonction sphère en dimension 2, avec un point initial loin de l'optimum, à savoir $(0, 10^9)$ et un écart-type global initial de 1. Dans la colonne de droite on peut voir l'écart-type global ainsi que les valeurs propres de la matrice de covariance qui a été initialisée à l'identité. On voit clairement qu'à partir de la génération 800 le CMA apprend une matrice proportionnelle à la matrice identité (les 2 valeurs propres sont égales, graphe en bas à droite). Par contre, de la génération 1 à la génération 800 ce n'est pas la matrice identité qui est apprise par l'algorithme CMA, même si la matrice de covariance initiale est l'identité. On peut noter que la phase où l'algorithme commence à apprendre la matrice identité correspond à une phase où l'algorithme tourne autour de l'optimum, comme on peut le voir sur le graphique en bas à gauche de la figure 1.1, où l'on observe que le signe de la deuxième coordonnée du parent change. Par ailleurs, dans la première phase, l'algorithme s'oriente grâce à la direction donnée par p_c^n qui est une direction de descente plus ou moins proche de la direction du gradient de la fonction.

1.4 Autres algorithmes évolutionnaires continus : DE et GA_____

1.4.1 Les Algorithmes génétiques

Cette branche, la plus connue des algorithmes évolutionnaires se caractérisait à l'origine [81] par une représentation binaire de l'espace de recherche et par une prépondérance de l'utilisation de l'opérateur de croisement sur l'opérateur de mutation. Pour traiter les problèmes d'optimisation continue, les premières approches avec des algorithmes génétiques consistaient donc à coder les nombres réels sous forme binaire, et à appliquer ensuite les opérateurs clas-

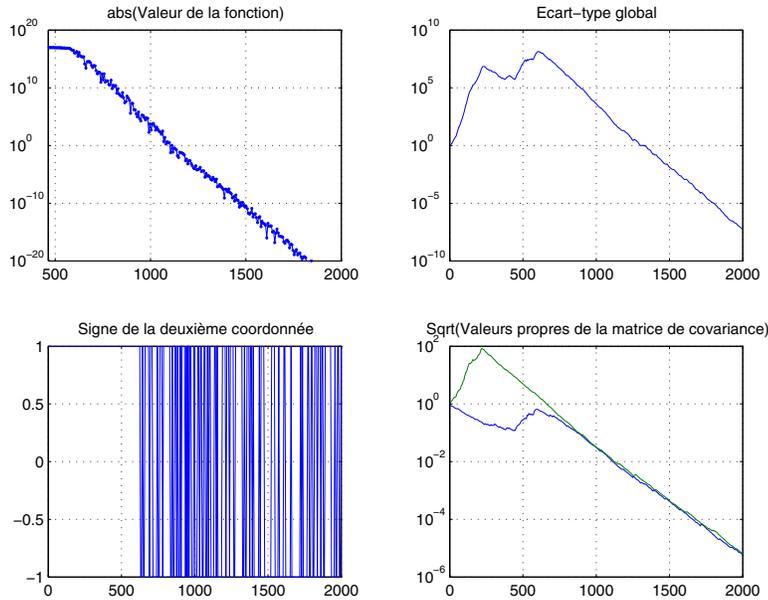


FIG. 1.1: *Algorithme CMA sur la fonction sphère, initialisé en $(0, 10^9)$.*

siques sur les chaînes de bits. Il est assez facile d'imaginer que cette représentation engendre des problèmes (voir [121]). Par exemple la topologie de \mathbb{R} n'est pas respectée par un codage binaire, plus précisément deux points voisins dans \mathbb{R} au sens de la topologie usuelle ne sont pas en toute généralité envoyés sur des points voisins de l'espace de chaînes de bits (au sens distance de Hamming par exemple). Or les opérateurs des algorithmes génétiques classiques étant conçus pour une représentation binaire, ils n'exploitent pas au mieux la topologie locale de la fonction à optimiser. En considérant des codages binaires de type Gray, ces problèmes peuvent être atténués, mais mis à part par quelques fervents défenseurs de la représentation binaire, l'approche binaire laisse place de nos jours à une représentation des individus de l'algorithme évolutionnaire sous forme de vecteurs de réels [69, 58].

Algorithme génétique avec croisement SBX Le croisement SBX développé par Deb dans en 1995 [41] spécialement pour l'optimisation continue constitue le premier croisement présentant des performances comparables aux SA-ES de Schwefel (cf Section 1.3.2) sur certaines fonctions tests. Par ailleurs dans l'article [42], Deb et Beyer montrent que ce croisement exhibe des propriétés similaires aux SA-ES, propriétés qualifiées *d'auto-adaptatives* dans ce papier. L'idée d'un tel croisement est de générer des enfants plutôt proches des parents si les parents sont proches et des enfants pouvant être plus éloignés si les parents sont éloignés.

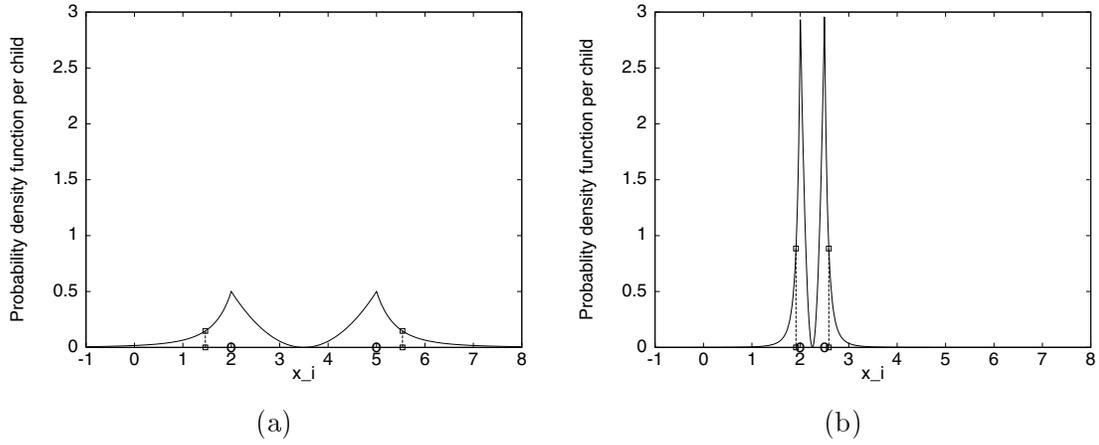


FIG. 1.2: Densité de probabilité de la loi des enfants dans l'algorithme SBX pour $\eta = 1$, la position des parents est indiquée avec un cercle. Ces figures sont extraites de [42] avec l'aimable permission des auteurs

Notons $(X_1^n)_i$ la i ème coordonnée d'un premier parent à la génération n et soit $(X_2^n)_i$ celle d'un deuxième parent. On génère les i èmes coordonnées des enfants $(Y_1^n)_i$ et $(Y_2^n)_i$ de la façon suivante:

On définit β_i la variable aléatoire suivante

$$\beta_i = \begin{cases} (2U_i[0, 1])^{1/\eta+1} & \text{si } U_i[0, 1] \leq \frac{1}{2}; \\ (1/(2(1 - U_i[0, 1])))^{1/\eta+1} & \text{sinon.} \end{cases} \quad (1.4.26)$$

où $U_i[0, 1]$ est une variable aléatoire de loi uniforme entre 0 et 1 et η un paramètre positif. Les i èmes coordonnées des enfants sont alors générées grâce à

$$(X_1^{n+1})_i = 0.5 \left((1 + \beta_i)(X_1^n)_i + (1 - \beta_i)(X_2^n)_i \right) \quad (1.4.27)$$

$$(X_2^{n+1})_i = 0.5 \left((1 - \beta_i)(X_1^n)_i + (1 + \beta_i)(X_2^n)_i \right) \quad (1.4.28)$$

La figure 1.2 illustre la densité de probabilité d'apparition des enfants sachant la position des parents (en 2 et 5 sur la figure 1.2 (a) et 2 et 2.5 sur la figure 1.2 (b)) pour un paramètre η ayant une valeur de 1.

Adaptation dans SBX Le mécanisme implicite introduit pour adapter la loi de probabilité des enfants en fonction de la distance entre les parents confère au croisement SBX. Nous illustrerons ces propriétés d'adaptation suivant Deb et Beyer [42] à la section 1.6.

1.4.2 L'algorithme Differential Evolution (DE)

Bien que très simple dans sa description l'algorithme DE [140, 141], introduit en 1996, est très efficace, comme nous le verrons dans la section 1.7. Le principe est le suivant: un enfant est créé en appliquant séquentiellement un opérateur de mutation puis un opérateur de croisement. L'enfant créé remplace le parent dans la population courante s'il est meilleur que le parent. Nous allons noter (X_1^n, \dots, X_μ^n) les μ parents de la génération n .

Mutation dans DE Pour chaque parent \mathbf{X}_i^n , pour $i = 1, \dots, \mu$, le vecteur *mutant* suivant est créé

$$V_i^{n+1} = X_{r_1}^n + F(X_{r_2}^n - X_{r_3}^n) \quad (1.4.29)$$

où, r_1, r_2, r_3 sont trois entiers distincts tirés uniformément dans $\{1, \dots, \mu\}$ et F est le facteur dit *d'amplification* qui est un réel dans $[0, 2]$.

Croisement dans DE Un nouveau vecteur U_i^{n+1} croisement entre le parent \mathbf{X}_i^n et le vecteur V_i^{n+1} qui vient d'être formé par mutation est créé. On commence par tirer un entier j_0 uniformément dans $\{1 \dots, d\}$, ensuite

$$(U_i^n)_j = \begin{cases} (V_i^n)_j & \text{si } (U_j[0, 1] \leq CR) \text{ ou } j = j_0 \\ (\mathbf{X}_i^n)_j & \text{si } (U_j[0, 1] > CR) \text{ et } j \neq j_0 \end{cases}$$

où $CR \in [0, 1]$ est un paramètre de l'algorithme, $U_j[0, 1]$ est la $j^{\text{ième}}$ évaluation d'une variable aléatoire uniforme dans $[0, 1]$. L'index j_0 que l'on a tiré au début assure qu'au moins une coordonnée de \mathbf{X}_i^n est changée.

Remarque sur le croisement Le croisement précédent dépend fortement de la base dans laquelle on se place. Comme nous le verrons dans la section 1.7, ceci induit des mauvaises performances dès que la base adaptée à la fonction à optimiser n'est pas la base canonique⁷.

Adaptation dans DE C'est l'équation (1.4.29) qui permet l'adaptation dans DE, en utilisant la taille caractéristique de la population (représentée par la distance entre deux individus dans la population) pour la mutation. Cette adaptation, implicite, fait que l'algorithme DE rentre dans la catégorie des algorithmes *auto-adaptatifs* introduite en début de chapitre, cela sera illustré à la Section 1.6.

7. Où nous appelons base adaptée à la fonction, dans le cas d'une fonction elliptique par exemple la base propre de la matrice hessienne.

1.5 Les Algorithmes à Evolution de Distribution (EDAs)

Les EDAs constituent une branche récente des algorithmes évolutionnaires : les premiers algorithmes EDAs pour l'optimisation continue datent de 1998 [134] et le premier livre traitant des EDAs (sur un espace d'état discret et continu) date de 2001 [94]. Conceptuellement les EDAs sont différents des EAs classiques puisque l'idée n'est plus de faire évoluer un ensemble de points mais une distribution⁸ toute entière, censée contenir plus d'information. Ce domaine a connu pas mal d'évolutions ces cinq dernières années ([27, 117, 112, 94]), l'enjeu a été d'apprendre avec des techniques plus ou moins compliquées et des modèles d'autant plus complexes la distribution des points sélectionnés, modèle de distribution allant de la simple loi normale avec matrice de covariance diagonale (algorithme *PBIL_c* ci dessous) à des réseaux Baysiens ([117, 112, 111]), c'est pourquoi nous avons décidé de le mentionner dans ce chapitre. Cependant les performances de ces algorithmes étant encore moins bonnes que les performances des algorithmes évolutionnaires classiques, comme nous le verrons à la section 1.7, nous avons simplement choisi ici de présenter deux de ces EDAs, tout d'abord pour illustrer le concept et ensuite parce que ces EDAs présentent des similitudes avec les algorithmes présentés jusqu'à présent.

La première étape dans un EDA est de se fixer un modèle pour l'ensemble des distributions $(\mathcal{D}^n)_{n \geq 1}$ qui vont évoluer. Nous allons noter de manière générique dans la suite \mathcal{D}_Φ une distribution paramétrée par des paramètres Φ . Pour fixer les idées cette distribution peut être une distribution gaussienne paramétrée par sa moyenne et sa matrice de covariance. Le schéma classique d'un EDA est le suivant

Schéma d'un EDA On choisit \mathcal{D}_Φ^0 . On répète les étapes qui suivent jusqu'à ce qu'un critère d'arrêt soit atteint

A l'étape (ou génération) n

- On échantillonne λ fois la distribution \mathcal{D}^n , on obtient ainsi λ points $\{x_1^n, \dots, x_\lambda^n\}$.
- On sélectionne μ points parmi les λ (biais vers les meilleurs).
- On apprend la distribution \mathcal{D}^{n+1} à partir de ces μ points et éventuellement \mathcal{D}^n .

8. Traduction littérale de l'anglais distribution (pour loi de probabilité).

Remarques: L'étape de sélection consiste souvent à garder la moitié des individus ayant la fitness la plus basse et ensuite à apprendre la distribution de ces individus. Ceci peut aboutir dans certain cas à des convergences prématurées (cf Section 1.5.2).

1.5.1 Population Based Incremental Learning ($PBIL_c$)

C'est le premier EDA sur un espace d'état continu qui a été introduit [134]. Le modèle de distribution considéré est une distribution gaussienne de matrice de covariance diagonale. Ainsi à chaque étape il y a $2d$ paramètres à apprendre, la moyenne de la nouvelle distribution et les écart-types $(\sigma_1, \dots, \sigma_d)$. Notons M^n la moyenne de la distribution à la génération n , la mise à jour de la moyenne se fait de la façon suivante

$$M^{n+1} = (1 - \alpha)M^n + \alpha(X_{1:\lambda}^n + X_{2:\lambda}^n - X_{\lambda:\lambda}^n) \quad (1.5.30)$$

où $\alpha \in]0, 1]$ et $X_{i:\lambda}^n$ désigne le i ème meilleur individu à la génération n . Nous notons σ_i^n l'écart-type dans la direction i à la génération n . Pour cette mise à jour les K meilleurs individus sont considérés (typiquement $K = \frac{\lambda}{2}$). La mise à jour proposée pour les écart-types est la suivante

$$\sigma_i^{n+1} = (1 - \alpha)\sigma_i^n + \alpha \sqrt{\frac{\sum_{j=1}^K ((X_{j:\lambda}^n)_i - \bar{X}_{K,i}^n)^2}{K}} \quad (1.5.31)$$

où $(X_{j:\lambda}^n)_i$ désigne la i ème coordonnée du j ème individu à la génération n et $\bar{X}_{K,i}^n$ la i ème coordonnée de la moyenne empirique des K meilleurs individus à la génération n .

Remarquons que l'algorithme $PBIL =_c$ peut être vu comme un $(1, \lambda)$ -ES avec une mise à jour pour le parent légèrement différent des méthodes des ES traditionnelles, et une règle particulière de mise à jour des écart-types.

1.5.2 EMNA (Estimation of Multivariate Normal Algorithm)

L'algorithme EMNA [94] considère une distribution gaussienne avec une matrice de covariance pleine. La mise à jour de la matrice se fait en apprenant la distribution au sens du maximum de vraisemblance des $\lambda/2$ meilleurs individus sélectionnés.

1.6 Adaptation et auto-adaptation : illustrations

Nous illustrons simplement dans cette section, la notion d'adaptation et d'auto-adaptation à travers les tests numériques classiquement réalisés pour justifier qu'un algorithme possède des propriétés d'adaptation.

Tout d'abord, les notions d'adaptation et d'auto-adaptation sont reliées à la convergence log-linéaire sur la fonction sphère (et sur des fonctions elliptiques), traduisant que l'on est capable d'améliorer infiniment (dans les limites de la précision numérique) les performances d'un algorithme.

Par ailleurs, Deb et Beyer [42] ont été les premiers à associer la notion d'adaptation au fait que l'on soit capable de suivre l'optimum d'une fonction. En effet, comme nous l'avons évoqué dans l'introduction de ce chapitre, sur une telle fonction, un algorithme idéal devrait alterner exploration et exploitation et donc être capable de détecter le changement de phase et d'adapter les paramètres de l'algorithme en conséquence. Ceci est ici illustré pour les algorithmes, SA-ES isotrope, SBX, DE et CMA sur les figures 1.3⁹ et 1.4. Ces dernières montrent la valeur de la *fitness* en fonction du nombre de générations sur la fonction elliptique suivante

$$f_{\text{rand}} = \sum_{i=1}^d r_i (x_i - x_i^*)^2$$

où toutes les K générations (précisé plus loin), la localisation de l'optimum $(x_i^*)_{1 \leq i \leq d}$ est modifiée, tirée aléatoirement dans $[-1, 1]$. De même, les coefficients de la matrice hessienne de la fonction $(r_i)_{1 \leq i \leq d}$ sont des entiers compris entre 1 et d qui changent toutes les K générations. Sur la figure 1.3 les moments où l'optimum change sont clairement visibles.

Les paramètres utilisés pour l'algorithme SBX sont décrits dans [42]. Pour la même fonction, la même courbe est tracée à la figure 1.4 pour les algorithmes DE et CMA. Ces algorithmes sont utilisés avec leurs valeurs par défaut (codes Matlab des auteurs).

Les moments où la fitness change est différent d'un algorithme à l'autre pour rendre plus visible l'effet de l'adaptation.

Par ailleurs, on voit également, sur la figure 1.3, l'écart-type empirique (sur l'ensemble des individus de la population) pour la 15^{ième} coordonnée. Sur la figure 1.4, en bas à gauche, on voit l'écart-type empirique (sur l'ensemble des individus de la population) coordonnée par coordonnée est montré. Enfin, sur la figure 1.4, en bas à droite on voit l'écart type global dans l'algorithme CMA. Ceci illustre l'adaptation de l'ensemble de la population.

9. Figures extraites de l'article [42]

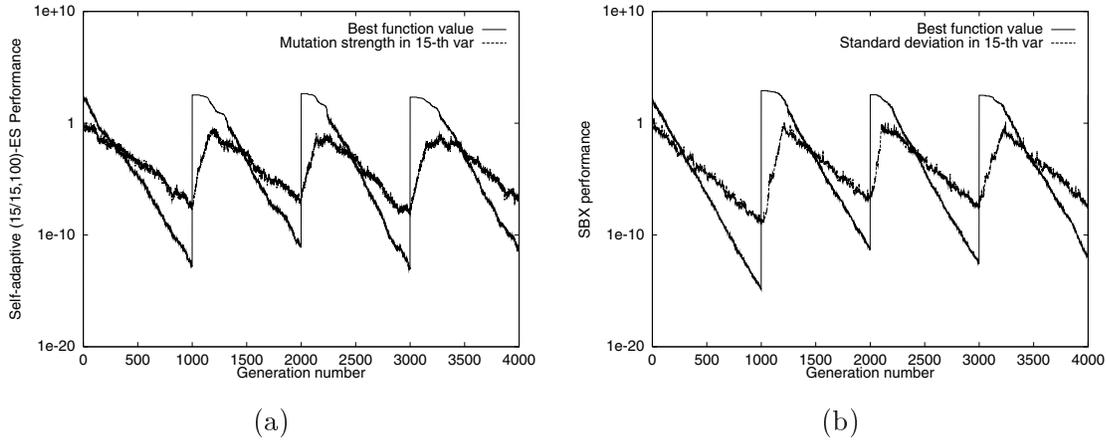


FIG. 1.3: *Algorithme SA-ES non-isotrope (a) et SBX (b), sur f_{rand} pour $d = 20$, figures tirées de [42] avec l'aimable permission des auteurs. Pour chaque figure est tracé, la fitness et également l'écart-type empirique calculé à partir des individus dans la population pour la quinzième variable.*

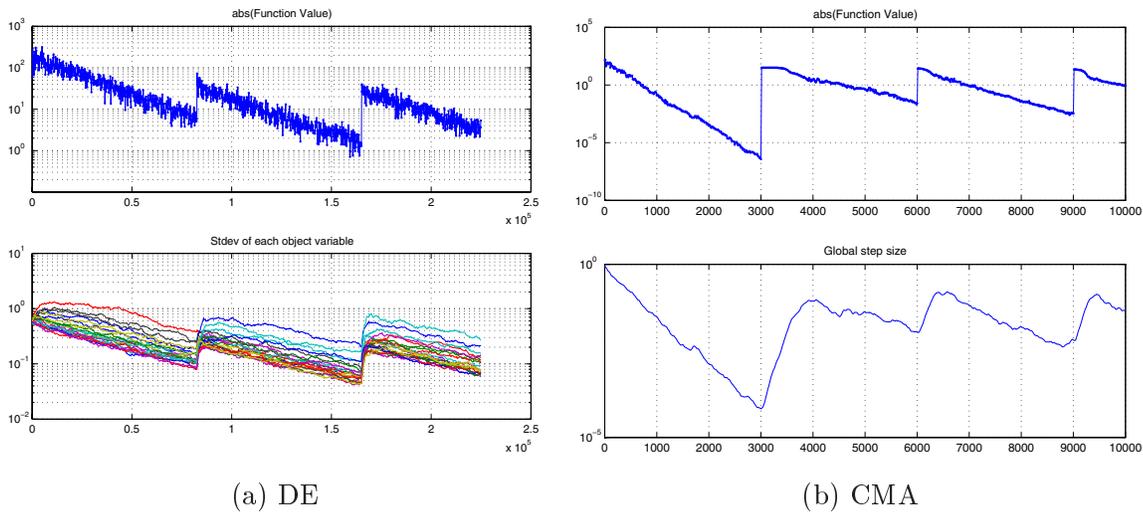


FIG. 1.4: *Algorithme DE (a) et CMA (b), sur f_{rand} pour $d = 20$. En haut fitness en fonction du nombre de génération. En bas à gauche, écart-type empirique, calculé à partir des individus de la population, coordonnée par coordonnée. En bas à droite, pas de la mutation de l'algorithme CMA.*

1.7 Mythes et réalités de l'optimisation globale

Les algorithmes SA-ES, CMA, DE, et SBX présentés dans les sections précédentes sont largement utilisés à l'heure actuelle dans divers domaines allant de l'ingénierie automobile [138, 113, 114], à la mécanique quantique [11, 155, 148] et à la robotique [28]. Plusieurs pages web recensent les applications de ces divers algorithmes à des problèmes réels témoignant de leur efficacité. Mentionnons ici la page web [139] pour l'algorithme DE et la page web [72] pour l'algorithme CMA.

Pour autant, répondre à la question de quel algorithme utiliser parmi ceux que nous venons précisément de décrire n'est pas chose facile. D'une part, cela peut dépendre du problème. D'autre part, il n'existe pas d'étude exhaustive comparant tous ces algorithmes sur les mêmes fonctions tests, étude qui permettrait peut-être de déterminer quel algorithme utiliser sur telle ou telle classe de fonction. Cependant, plusieurs articles [131, 141, 75, 91] que nous allons commenter brièvement ici, comparent certains de ces algorithmes entre eux, et plus généralement comparent les algorithmes évolutionnaires à d'autres méthodes d'optimisation globale. Dans la suite, les comparaisons entre algorithmes seront faites en terme de nombre d'évaluation de fonctions pour atteindre le minimum avec une précision fixée.

Algorithmes Evolutionnaires versus autres algorithmes d'optimisation globales

Plusieurs articles et livres comparent les performances des algorithmes évolutionnaires à d'autres méthodes d'optimisation globale. Nous ne présentons ici que les plus récents, tout d'abord un article tentant de comparer l'algorithme DE [141] et ensuite un article comparant CMA-ES à DE et à d'autres méthodes d'optimisation globale [75].

L'article [141] de 1997 compare l'algorithme DE (Section 1.4.2) à d'autres méthodes d'optimisation globale (reconnues pour être parmi les meilleures en terme d'optimisation globales) sur des fonctions tests classiques pour l'optimisation globale. L'algorithme DE est tout d'abord comparé aux méthodes de recuit simulé les plus performantes, à savoir le recuit simulé adaptatif [84] et la stratégie de recuit de Nelder et Mead [120]. Sur les fonctions elliptiques testées (sauf sur la fonction sphère) les performances de DE sont meilleures. Dans le cas de fonctions multimodales, la différence est encore plus importante : les algorithmes de recuit simulé contrairement à DE ne convergent pas tout le temps vers le minimum global, et lorsqu'ils convergent, ils convergent moins vite. Ensuite DE est comparé à une méthode

de gradient stochastique [4] sur des fonctions multimodales et DE est bien meilleur que la méthode de gradient stochastique pour toutes les fonctions multimodales testées.

L'article [75] de 2004 compare quant à lui l'algorithme CMA à l'algorithme DE qui est devenu entre temps une référence en matière d'optimisation globale ainsi qu'à un algorithme de BFGS où un lissage préalable de la fonction est effectué. Moyennant un réglage de la taille de la population du CMA et de la façon dont on initialise la population, les performances de l'algorithme CMA sur des fonctions multimodales sont toujours meilleures que l'algorithme de BFGS avec lissage et sont légèrement inférieure aux performances de DE lorsque la fonction multimodale admet pour base propre la base canonique. Mais lorsque l'on applique une rotation préalable de la base canonique faisant que la base adaptée de la fonction n'est plus la base canonique, CMA reprend l'avantage.

Algorithmes Evolutionnaires versus Algorithmes Evolutionnaires

A l'heure actuelle trois algorithmes évolutionnaires sont généralement recommandés pour l'optimisation continue. Tout d'abord les SA-ES, certainement les algorithmes évolutionnaires les plus connus à l'heure actuelle, ensuite l'algorithme DE et enfin, le plus récent l'algorithme CMA. Dans, l'article [77], l'algorithme CMA est comparé aux algorithmes SA-ES. Les auteurs montrent que l'algorithme CMA présente des performances meilleures que les algorithmes SA-ES pour des fonctions elliptiques mal conditionnées, ou pour des fonctions pour lesquelles la base propre n'est pas la base canonique. Dans l'article de 2004 [75], CMA est comparé à SA-ES et DE sur des fonctions multimodales. Il est plus précisément montré que l'algorithme CMA peut présenter des performances aussi bonnes que les SA-ES et l'algorithme DE sur des fonctions multi-modales dès qu'un réglage propre de l'écart-type initial ainsi que de la taille de la population est effectué. Par ailleurs, dans [75], les auteurs montrent que les performances de l'algorithme DE se dégradent considérablement même sur une fonction elliptique si la base propre n'est plus la base canonique.

Dans [42], il a été montré que les performances de l'algorithme SBX sont légèrement inférieures à celle des algorithmes SA-ES. Notons enfin que les algorithmes EDA mentionnés dans ce chapitre sont en terme de performances largement surclassés (pour plus de détail nous renvoyons à [91]).

Conclusion

Nous venons de présenter un état de l'art sur l'optimisation continue par algorithmes évolutionnaires, en insistant plus particulièrement sur les stratégies d'évolutions, algorithmes que nous retrouverons tout au long de cette thèse. Parmi les algorithmes présentés ici, les algorithmes GA-SBX, SA-ES, DE et CMA sont largement utilisés pour des problèmes d'optimisation du monde réel issus de divers domaines. Nous avons tenté de dégager des tendances générales de plusieurs articles comparatifs, d'où il ressort que l'algorithme CMA-ES est sans doute le meilleur choix aujourd'hui, modulo le réglage de la taille de la population et de l'écart-type initial. Parmi les algorithmes présentés il est notamment le seul dont les performances sont invariantes par rotation – sachant que pour les problèmes réels nous ne disposons pas en général d'information sur la base propre de la matrice Hessienne de la fonction objectif.

Nous avons par ailleurs introduits les concepts clés d'adaptation et d'auto-adaptation développés ces dernières années pour l'optimisation continue par algorithmes évolutionnaires. Ces techniques présentent deux avantages: d'une part, elles permettent à l'algorithme de modifier son comportement en ligne, durant l'évolution, et en fonction de l'évolution, soit de manière explicite (e.g. la mise à jour adaptative de la matrice de covariance dans la méthode CMA), soit en laissant faire le hasard (mutation des paramètres de mutation dans les méthodes auto-adaptatives SA-ES, variance de la mutation réglée de fait par l'ensemble de la population dans l'algorithme DE, lui conférant des propriétés d'auto-adaptativité). D'autre part, à défaut de diminuer le nombre de paramètres à régler, elle en diminue la sensibilité: les paramètres τ de la mutation log-normale de l'algorithme SA-ES par exemple ont une large plage de validité une fois normalisés, alors que le réglage direct du pas de la mutation est difficilement envisageable.

Ces avantages ont d'ailleurs motivé l'utilisation du concept d'adaptation dans d'autres domaines que le réglage de la mutation gaussienne pour l'optimisation continue. Ainsi, lors de la prise en compte de contraintes par méthode de pénalisation, le réglage de l'intensité des coefficients de pénalité peut s'avérer délicat – de plus, il est en général recommandé de les choisir petits au début de la recherche, pour autoriser l'exploration de zones non admissibles, et de les augmenter en fin de recherche pour favoriser l'apparition d'individus admissibles. Par exemple, dans le domaine continu, l'algorithme ASCHEA (*Adaptive Segregated Constraint Handling Evolutionary Algorithm*) [19, 20] augmente ou diminue ce facteur de pénalisation en

fonction de la proportion d'individus admissibles dans la population ; dans le domaine discret des Satisfactions de Contraintes (problèmes SAT), l'algorithme SAW (*Self-Adaptive Weight*) [56] augmente le poids dans la fonction objectif des contraintes qui sont systématiquement non satisfaites par les meilleurs individus successifs de la population.

Par ailleurs, l'auto-adaptativité (adaptativité stochastique) est a priori encore plus attirante dans le contexte des algorithmes évolutionnaires. Ainsi, son illustration dans ce chapitre par l'algorithme SA-ES montre bien la simplicité de sa mise en oeuvre (il suffit de rajouter à l'individu ses propres paramètres de mutation). Par contre, son extension à d'autres champs que celui de l'optimisation continue n'a pour l'instant pas donné de résultats très probants – citons par exemple dans le cadre des chaînes de bits l'auto-adaptation de la probabilité de mutation par bit proposée par Obalek [110] mais dont les performances n'ont jamais été à la hauteur de simples décroissance géométrique bien paramétrée [14].

Dans ce contexte, et même si du strict point de vue du paradigme Darwinien, l'algorithme CMA-ES peut être vu comme un recul par rapport aux SA-ES puisque laissant moins faire le hasard [42], il est clair qu'une adaptation déterministe bien menée peut permettre d'utiliser de manière plus efficace les informations contenues dans tous les individus évalués. Une telle méthode sera l'objet du chapitre suivant.

Chapitre 2

Méthode explicite d'adaptation de la matrice de covariance pour la mutation gaussienne

Une partie des résultats de ce chapitre fait l'objet de l'article [10].

- [10] A. Auger, N. Vanhaecke, and M. Schoenauer. *LS-CMA-ES: A second-order algorithm for covariance matrix adaptation*. In Xin Yao et al., editor, *Proceedings of the 8th Conference on Parallel Problems Solving from Nature*, pages 182–191. Springer-Verlag, LNCS 3242, 2004 (*Best Student Paper Award*).

2.1 Introduction

Ce chapitre correspond chronologiquement au dernier travail effectué durant cette thèse et fait en particulier suite à l'application d'un algorithme SA-ES pour la résolution du problème de contrôle par laser faisant l'objet de la Partie 3. Lors de cette application, nous avons pu constater que de nombreux individus étaient perdus lors de l'étape de remplacement sans qu'il soit fait usage de l'information sur la fonction qu'ils contenaient. Ceci était d'autant plus vrai dans la phase d'exploitation de l'algorithme. Ces individus étaient souvent générés avec des paramètres non optimaux: par exemple, dans la fin de la recherche, au voisinage d'un minimum local, le pas de la mutation était souvent trop petit par rapport à la courbure de la fonction (que l'on a évaluée a posteriori par différences finies). Pour ce problème d'optimisation, le coût d'une évaluation de fonction était de l'ordre de la minute, motivant d'autant plus la recherche de techniques faisant un meilleur usage des points évalués puis jetés.

Ainsi, le point de départ de ce travail est la constatation que même les algorithmes évolutionnaires correspondant à l'état de l'art du domaine (Notons qu'à l'époque, l'algorithme CMA n'était pas très connu), ne font pas usage de toute l'information qu'ils ont à leur disposition: la procédure de remplacement jette une partie des individus sans utiliser l'information qu'ils contiennent.

Le cadre algorithmique considéré est celui des stratégies d'évolutions présentées au chapitre précédent. Plus précisément c'est l'opérateur de mutation gaussien consistant à rajouter au parent un vecteur aléatoire gaussien qui va nous intéresser. Rappelons que les opérateurs de mutations gaussiens découpent généralement la matrice de covariance du vecteur gaussien en deux parties, d'une part le *pas* de la mutation que nous noterons tout au long de ce chapitre σ^n , n étant l'index de génération, et d'autre part une matrice que nous appellerons par abus de langage la *matrice de covariance* que nous noterons C^n . Ainsi, la forme générique de l'opérateur de mutation appliqué à l'étape n à une variables aléatoire X à valeur dans \mathbb{R}^d que nous allons considérer est la suivante

$$X := X + \sigma^n \mathcal{N}(0, C^n)$$

Nous plaçant ensuite dans une perspective d'optimisation locale, nous nous allons poser la question de savoir quel est la "meilleure" matrice de covariance. Dans le contexte d'une

fonction fitness elliptique

$$f_E(x) = \frac{1}{2}x^T H x$$

où, H est une matrice symétrique définie positive, le choix de $(\frac{1}{2}H)^{-1}$ comme matrice de covariance semble raisonnable, comme nous le verrons à la section 2.2. Dans le cas d'une fonction plus générale suffisamment régulière, l'approximation à l'ordre 2 de la fonction fait que l'on peut considérer localement la fonction comme elliptique. Ainsi, le but de ce chapitre va être d'essayer d'apprendre la courbure locale de la fonction et de l'utiliser dans la mise à jour de la matrice de covariance.

La question est ensuite comment trouver une approximation de $(\frac{1}{2}H)^{-1}$ en utilisant l'information à notre disposition, c'est à dire un ensemble de points avec leur valeur de fitness correspondante. Nous allons voir à la section 2.4 comment une minimisation au sens des moindres carrés permet de répondre à cette question.

La démarche que nous allons adopter tout au long de ce chapitre pour valider nos choix est purement numérique: d'une part certains arguments que nous avançons reposent sur des articles eux mêmes basés sur des validations numériques; d'autre part nous allons nous aussi valider les méthodes numériques proposées grâce à des tests numériques. En particulier, le nouvel opérateur de mutation sera comparé sur des fonctions tests classiques à l'algorithme CMA, considéré aujourd'hui au niveau pratique comme un des algorithmes évolutionnaires les plus performants.

2.2 Matrice de covariance optimale

La matrice de covariance de la mutation détermine les directions dans lesquelles la recherche va s'effectuer. Dans le contexte d'une fonction elliptique ($f_E(x) = \frac{1}{2}x^T H x$ où H est symétrique définie positive) intuitivement, un choix raisonnable pour C semble être $(\frac{1}{2}H)^{-1}$, ce qui correspond à une "forme" de matrice de covariance proportionnelle aux lignes de niveau de la fonction (Figure 2.1). En effet, un tel choix correspond à faire des pas plus larges dans les directions où la fonction varie peu que dans les directions où la fonction varie beaucoup. De manière plus formelle, si $C = (\frac{1}{2}H)^{-1}$, C et H ont mêmes vecteurs propres, et des valeurs propres correspondantes inversement proportionnelles.

Un autre argument vient du cas particulier de la fonction sphère: un changement de variable montre en effet que minimiser f_E avec l'opérateur de mutation ayant pour matrice de covariance $(\frac{1}{2}H)^{-1}$ revient à minimiser la fonction sphère ($f_S(x) = x^T x$) avec un opérateur de

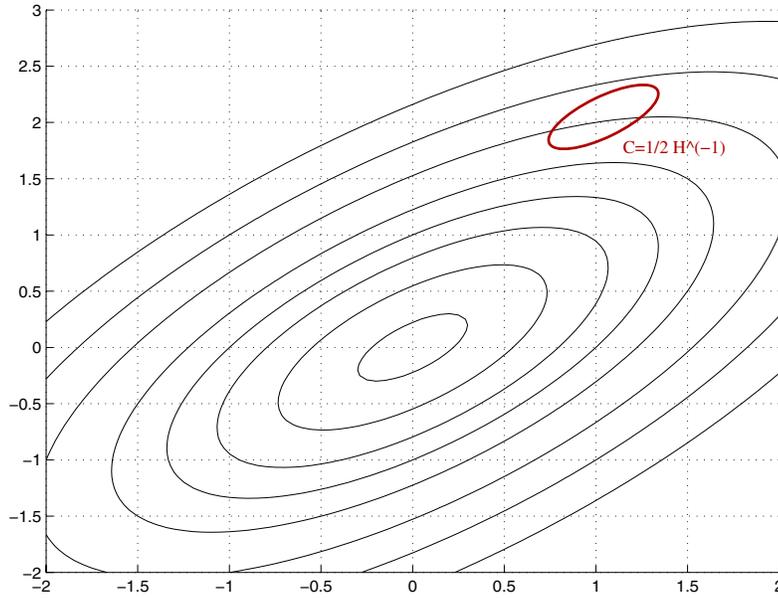


FIG. 2.1: Lignes de niveaux d'une fonction elliptique et support d'un vecteur gaussien adapté à ces lignes de niveaux

mutation ayant pour matrice de covariance l'identité. Or sur la fonction sphère, des tests numériques effectués avec des opérateurs de mutation ayant plusieurs formes de matrice de covariance montrent que la matrice de covariance identité [13, 15] est la meilleure parmi l'ensemble des tests effectués. Ainsi, *a priori* le choix de matrice de covariance $(\frac{1}{2}H)^{-1}$ pour une fonction elliptique est raisonnable.

D'autre part, d'après les formules de Taylor, toute fonction suffisamment régulière s'approche localement à l'ordre 2 par une fonction elliptique:

$$f(X) = f(X_0) + (X - X_0)^T \nabla f(X_0) + \frac{1}{2}(X - X_0)^T H(X_0)(X - X_0) + o(\|X - X_0\|^2) \quad (2.2.1)$$

où $\nabla f(X_0)$ désigne le gradient de f au point X_0 et $H(X_0)$ la matrice Hessienne de f en X_0 . Ainsi, cela semble raisonnable d'essayer d'utiliser au sein de l'opérateur de mutation une approximation de l'inverse de la matrice Hessienne de la fonction *fitness*.

2.3 Quelles matrices de covariances pour les méthodes historiques?

Historiquement, il ne semble pas que cela soit toujours la préoccupation de “quelle matrice de covariance apprendre?” qui ait guidé la mise au point des différentes techniques d’adaptation des algorithmes évolutionnaires continus. En particulier les techniques d’auto-adaptation des paramètres de mutation sont justifiées par Schwefel dans [131] par “l’évolution va ajuster gratuitement les bons paramètres”, dans la mesure où le modèle considéré le permet. Regardons tout de même numériquement quels sont les valeurs apprises par les différents algorithmes existants.

Tout d’abord considérons le cas du ES non isotrope (voir 1.3.2) où le modèle de matrice de covariance considéré dans ce cas là est diagonal. Plaçons nous sur une fonction elliptique mal conditionnée, plus particulièrement ici

$$f_{elli} = \frac{1}{2} \sum_{i=1}^d (10^6)^{\frac{i-1}{n-1}} x_i^2$$

dont la base propre est la base canonique pour ne pas désavantager le modèle isotrope. Dans ce cas là $(\frac{1}{2}H)^{-1}$ vaut

$$\left(\frac{1}{2}H\right)^{-1} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \vdots & 0 & 10^{-6} \end{pmatrix}$$

La figure 2.2 (a) présente un essai typique de l’algorithme (1, 100)-SA-ES isotrope: à gauche, la valeur de la fitness à chaque génération en fonction du nombre d’évaluations de la fonction: à droite, les valeurs des écarts types de la matrice de covariance¹ dans les différentes directions. On remarque que ces valeurs propres ne sont pas proportionnelles à celles de $(\frac{1}{2}H)^{-1}$. On pourra s’étonner aussi des paliers que l’on peut voir au niveau de la fitness et de la fluctuation des valeurs propres de la matrice de covariance: ces phénomènes déjà observés [16], ont entre autre motivé l’introduction du croisement que nous avons décrit au chapitre 1, qui permet d’atténuer ces phénomènes.

La figure 2.2 (b) montre exactement les mêmes courbes que précédemment pour l’algorithme CMA avec le réglage par défaut des paramètres donnés au chapitre précédent. On remarque que à partir de 6000 évaluations les valeurs propres de la matrice de covariance

1. Nous appelons écarts types de la matrice de covariance les racines des valeurs propres de la matrice

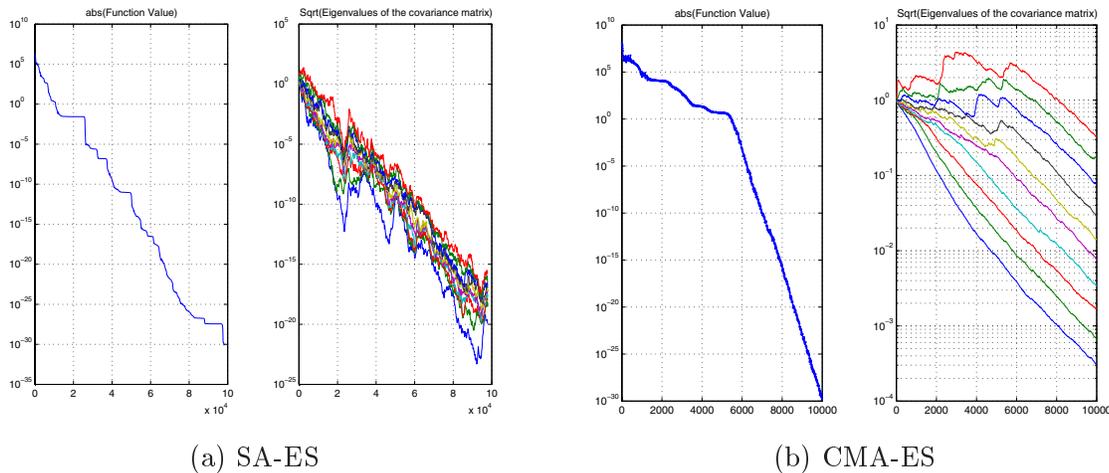


FIG. 2.2: *Fitness (ici f_{elli} en dimension 10) en fonction du nombre d'évaluations (en échelle log), et racines des valeurs propres de la matrice de covariance de l'opérateur de mutation pour respectivement un (1, 100)-SA-ES non isotrope (a) et le CMA-ES avec ses valeurs par défaut (b).*

sont proportionnelles à celles de $(\frac{1}{2}H)^{-1}$. Ainsi, du moins en fin de recherche, l'algorithme CMA-ES apprend une matrice proportionnelle à $(\frac{1}{2}H)^{-1}$. Signalons que nous avons illustré ici notre propos sur la fonction elliptique f_{elli} dont la matrice hessienne est diagonale dans la base canonique, mais l'algorithme CMA-ES apprend aussi une matrice de covariance proportionnelle à $(\frac{1}{2}H)^{-1}$ dans le cas général.

Notons que la même question se pose aussi pour la *loi de proposition* dans le contexte des algorithmes de recuit simulé. Dans [147], il est également proposé de prendre une loi de proposition (cf Chapitre 1, section 1.2 pour la définition de la loi de proposition) gaussienne de matrice de covariance H^{-1} .

2.4 Apprendre $(\frac{1}{2}H)^{-1}$: minimisation au sens des moindres carrés

La question suivante est maintenant de trouver une méthode utilisant l'information disponible dans un algorithme évolutionnaire afin d'apprendre au mieux $\frac{1}{2}H^{-1}$. L'information de base disponible dans un algorithme évolutionnaire est un ensemble de points avec les valeurs de fonctions correspondantes $(X_k, f(X_k))_{1 \leq k \leq N}$. Une méthode immédiate pour apprendre une approximation de la matrice hessienne consiste alors à considérer le problème suivant de

minimisation au sens des moindres carrés:

$$\min_{g \in \mathbb{R}^d, H \in \mathcal{S}(\mathbb{R}^d)} \frac{1}{N} \sum_{k=1}^N \left(f(X_k) - f(X_0) - (X_k - X_0)^T g - \frac{1}{2} (X_k - X_0)^T H (X_k - X_0) \right)^2 \quad (2.4.2)$$

Les inconnues de ce problème sont $g = (g_1, \dots, g_d)$ (d inconnues), et $H = (H_{ij})_{1 \leq i \leq j \leq d}$ ($d(d+1)/2$ inconnues). Le système linéaire correspondant s'écrit de la façon suivante : Notons Y le vecteur

$$Y = (g_1, \dots, g_d, H_{11}, \dots, H_{1d}, H_{22}, \dots, H_{dd})^T$$

de $\mathbb{R}^{d(d+3)/2}$ et A la matrice à N lignes et $d(d+3)/2$ colonnes, dont la ligne k que nous notons A_k s'écrit

$$A_k = ((X_k - X_0)_1, \dots, (X_k - X_0)_d, (X_k - X_0)_1(X_k - X_0)_1, \dots, \\ \dots, (X_k - X_0)_1(X_k - X_0)_d, (X_k - X_0)_2(X_k - X_0)_2, \dots, (X_k - X_0)_d(X_k - X_0)_d)^T$$

où la notation $(X_k - X_0)_i$ désigne la i ème ligne du vecteur $(X_k - X_0)$. Notons enfin F le vecteur de \mathbb{R}^d suivant

$$F = (f(X_1) - f(X_0), \dots, f(X_d) - f(X_0))$$

L'équation (2.4.2) se réécrit

$$\min_Y \|AY - F\|_2^2 \quad (2.4.3)$$

Une solution de l'équation (2.4.3) est un point critique de $\|AY - F\|_2$. La différentiation de cette équation donne $2A^T AY - 2A^T F$, et dès que l'on possède plus de $d(d+3)/2$ points, le système d'équations $Ay = F$ est surdéterminé. Si $(A^T A)$ est inversible, la solution au sens des moindres carrés de ce système est donnée par $(A^T A)^{-1} A^T F$. La matrice $(A^T A)^{-1} A^T$ est appelée pseudo-inverse, le coût de calcul d'une telle matrice par une méthode directe est de $(d(d+3)/2)^3$ soit de l'ordre de d^6 .

Critère de qualité pour l'approximation Il sera important dans la mise en oeuvre pratique d'être capable d'estimer la qualité de l'approximation ainsi calculée. Dans le cas d'une fonction elliptique, la solution de (2.4.2) est exactement donnée par $\nabla f(X_0)$ et $H(X_0)$, et la valeur du résidu (valeur de l'expression entre parenthèses) est zéro. Dans un cas général, si on note par \hat{g} et \hat{H} la solution du problème (2.4.2), la valeur en le minimum n'est pas zéro, et une mesure de la qualité de l'approximation du gradient par \hat{g} et de la Hessienne par \hat{H} est donnée par

$$\sum_{k=1}^N \left(f(X_k) - f(X_0) - (X_k - X_0)^T \hat{g} - \frac{1}{2} (X_k - X_0)^T \hat{H} (X_k - X_0) \right)^2.$$

Cependant, afin d'obtenir une quantité invariante par dilatation et par translation de la fonction f nous avons considéré le critère de qualité suivant

$$\mathcal{Q} = \frac{1}{N} \sum_{k=1}^N \left(\frac{f(X_k) - f(X_0) - (X_k - X_0)^T \hat{g} - \frac{1}{2} (X_k - X_0)^T \hat{H} (X_k - X_0)}{f(X_k) - f(X_0) - (X_k - X_0)^T \hat{g}} \right)^2, \quad (2.4.4)$$

qui s'est avéré après des tests numériques discriminer les cas où l'approximation était bonne ou les cas où elle était mauvaise.

2.5 L'algorithme LS-CMA-ES

Les idées précédentes sont mise en oeuvre dans l'algorithme présenté dans cette section.

Adaptation de la matrice de covariance

La méthode d'approximation décrite dans la section précédente requiert au moins $d(d+3)/2$ points avec les valeurs de fonction correspondantes. Ces points seront pris parmi les points déjà évalués dans l'algorithme évolutionnaire. Un compromis doit être fait entre la précision de l'approximation et le coût : tirer $d(d+3)/2$ points à chaque génération serait trop coûteux : seuls les points qui ont été évalués lors du déroulement de l'algorithme évolutionnaire seront utilisés, de préférence les plus récents.

De plus, l'approximation pourrait être calculée toutes les générations en utilisant une fenêtre glissante (c'est à dire en réutilisant certains points d'une fois à l'autre). Mais le calcul de la solution de (2.4.2) étant lui aussi coûteux, ce calcul ne sera pas fait toutes les générations et la même matrice de covariance sera utilisé entre deux calculs successifs de la solution de (2.4.2).

Un dernier problème à résoudre pour l'adaptation de la matrice de covariance est posé par la qualité de l'approximation : il semble en effet déraisonnable d'utiliser une mauvaise approximation de la hessienne comme matrice pour la mutation. Deux questions se posent alors : comment détecter les situations à risque, et comment y remédier. C'est précisément pour répondre à la première question que nous avons mis au point le critère de qualité présenté en fin de section précédente. Enfin, comme l'algorithme CMA est ce qui se fait de mieux aujourd'hui en terme de mise à jour de la matrice de covariance, il semble logique de se

rabattre sur cette méthode dès qu'on détecte une mauvaise approximation à l'aide du critère de qualité \mathcal{Q} .

Adaptation du pas de la mutation

Jusqu'à présent, nous avons simplement discuté la mise à jour de la matrice de covariance de la mutation gaussienne. L'autre point important est le réglage du pas de la mutation. Nous avons présenté au chapitre précédent plusieurs techniques pour l'adaptation du pas de la mutation. Pour les algorithmes présentés ici, nous avons retenu deux méthodes. Nous avons vu que apprendre $(\frac{1}{2}H)^{-1}$ comme matrice de covariance pour une fonction elliptique revient à transformer la minimisation d'une fonction elliptique en la minimisation de la fonction sphère. Donc il a semblé adapté ici d'utiliser la technique d'adaptation du pas par mutation du pas lui-même (technique SA-ES), qui présente parmi les meilleures performances sur la fonction sphère (cf [131] pour une comparaison entre cette technique et la règle des 1/5 et [91] pour une comparaison entre les approches dérandomisées et la règle des 1/5). C'est la première technique que nous avons testée pour l'algorithme $(1, \lambda)$ -LS-CMA-ES [10]. Celle-ci sera décrite précisément à la Section 2.5.1.

D'autre part, voulant mettre en oeuvre un algorithme avec plusieurs parents, pour voir en particulier comment se comportait la méthode dans un cas de fonctions multimodales, une technique dérandomisée pour le pas de la mutation (technique du CSA introduite au chapitre 1 Section 1.3.3.2) a été mise au point et testée ensuite. En effet, intuitivement, plus il y a des parents, plus une technique dérandomisée semble adaptée pour déduire de l'information sur la fonction et donc pour mettre à jour le pas de la mutation. Dans ce cas là, l'algorithme sera appelé le $(\mu/\mu, \lambda)$ -LS-CMA-ES.

2.5.1 L'algorithme $(1, \lambda)$ -LS-CMA-ES

Nous nous intéressons dans un premier temps à l'aspect convergence locale. Pour cela, nous commençons par mettre en oeuvre un algorithme de type $(1, \lambda)$ -ES, dont le pseudo-code est donné au tableau 2.1, où n est l'indice de génération, $x^{(n)}$ le parent à la génération n , $\sigma^{(n)}$ le pas à la génération n et $C^{(n)}$ la matrice de covariance.

La variable booléenne *Mode* dans le tableau 2.1 indique si l'on utilise la matrice Hessienne approchée pour la mutation ou si l'algorithme doit utiliser la règle de mise à jour standard de l'algorithme CMA-ES. Lorsque la qualité de l'approximation (donnée par le critère 2.4.4)

est “bonne”, concrètement lorsque Q est en dessous d'un certain seuil, on sera en mode **LS**, sinon en mode **CMA**.

Nous discutons maintenant des différents points de l'algorithme de manière plus détaillée.

Un $(1, \lambda)$ -SA-ES: L'algorithme de base pour le LS-CMA est un ES auto-adaptatif (Section 1.3.2). A la ligne 4 du tableau 2.1, λ enfants sont générés par mutation gaussienne de matrice de covariance C^n , avec un pas qui est le résultat d'une mutation par une loi log-normale du pas du parent. Les lignes 5 et 6 sont respectivement l'étape d'évaluation et de sélection.

Mise à jour de la matrice de covariance – méthode CMA: Les lignes 8 et 10 correspondent à l'étape de mise à jour de la matrice de covariance dans l'algorithme CMA-ES [76] (cf Chapitre 1, Section 1.3.3.3). Le calcul de vecteur p_c est effectuée même lorsque l'on est en mode LS de façon à représenter proprement le chemin qui a été parcouru pour le cas où l'on devrait passer en mode CMA. En mode LS, la matrice de covariance de la mutation est inchangée (ligne 9).

Approximation quadratique Toutes les n_{upd} générations (ligne 11), l'approximation de la matrice Hessienne est calculée en résolvant le problème 2.4.2 (ligne 12), à partir des points ayant été archivés au cours de l'évolution (ligne 7).

Si l'erreur de l'approximation est en dessous d'un certain seuil, alors la nouvelle approximation remplace l'ancienne, et le mode devient alors le mode LS (ligne 13). Dans le cas contraire, on passe en mode CMA (ligne 14), et la boucle principale recommence (ligne 16).

Les paramètres: Les valeurs de certain paramètres de l'algorithme LS-CMA-ES doivent être fixés manuellement. Les valeurs suivantes, tirées de [131] ou [76] lorsque cela a un sens, ou ajustées à la main durant des essais préliminaires, ont été utilisés dans tous les tests de la Section 2.5.2. Le nombre d'enfants λ a été réglé à 10 pour le LS-CMA-ES et à $4 + \lfloor 3 * \log(N) \rfloor$ pour l'algorithme CMA, c_c et c_{cov} , les paramètres de relaxation, à respectivement $\frac{4}{n+4}$ et $\frac{2}{(n+\sqrt{2})^2}$ [76]. Le paramètre pour le mise à jour de la mutation auto-adaptative log-normale, τ , a été réglé à $\frac{1}{\sqrt{n}}$ [131]. Le seuil Q_{th} pour l'erreur d'approximation pour la matrice hessienne et qui détermine le basculement entre les modes LS et CMA a été réglé à 10^{-3} , après des essais essentiellement sur des fonctions elliptiques et la fonction Rosenbrock. Enfin, n_{upd} a

```

Initialisation:  $x^{(0)} \leftarrow x_0; \sigma^{(0)} \leftarrow \sigma_0; C^{(0)} \leftarrow I_n; p_c^{(0)} \leftarrow 0; \text{archive} \leftarrow \emptyset; \text{Mode} \leftarrow \text{LS}$  1
while ( not critère d'arrêt ) do 2
     $g \leftarrow g + 1$  3
    Création de  $\lambda$  enfants:  $x_j^{(n+1)} \leftarrow x^{(n)} + \sigma^{(n)} \exp(\tau \hat{N}_j(0, 1)) N_j(0, C^{(n)})$ ,  $j \in [1, \lambda]$  4
    Évaluation de enfants: Calcul  $f(x_j^{(n+1)})$  pour tout  $1 \leq j \leq \lambda$  5
    Sélection de meilleur enfant  $x^{(b)}$ :  $x^{(n+1)} \leftarrow x_b^{(n+1)}$ ;  $\sigma^{(n+1)} \leftarrow \sigma^{(n)} \exp(\tau \hat{N}_b(0, 1))$  6
    Stoker les enfants dans l'archive 7
     $p_c^{(n+1)} \leftarrow (1 - c_c) p_c^{(n)} + \frac{\sqrt{c_c(2-c_c)}}{\sigma^{(n)}} (x^{(n+1)} - x^{(n)})$  8
    if mode LS  $C^{(n+1)} \leftarrow C^{(n)}$  9
    sinon ( mode CMA )  $C^{(n+1)} \leftarrow (1 - c_{cov}) C^{(n)} + c_{cov} p_c^{(n+1)} (p_c^{(n+1)})^T$  10
    if ( n mod  $n_{upd} = 0$  ) 11
        Calculer  $\hat{g}_{x^{(n)}}$ ,  $\hat{H}_{x^{(n)}}$  à partir des  $d^2$  plus récents individus archivés 12
        if  $\mathcal{Q}(\hat{g}_{x^{(n)}}, \hat{H}_{x^{(n)}}) < \mathcal{Q}_{th}$  Mode  $\leftarrow$  LS;  $C^{(n+1)} \leftarrow (\frac{1}{2} \hat{H}_{x^{(n)}})^{-1}$  13
        sinon Mode  $\leftarrow$  CMA 14
    end if 15
end while 16

```

TAB. 2.1: Pseudo code pour l'algorithme $(1, \lambda)$ -LS-CMA-ES

été réglé à 100 avec bien sûr au début un retard possible pour la première mise à jour afin d'avoir suffisamment de points disponibles (au moins d^2).

2.5.2 Résultats expérimentaux

Cette section présente les premiers résultats expérimentaux obtenus en utilisant l'algorithme $(1, \lambda)$ -LS-CMA-ES. Ces résultats sont comparés à ceux de l'algorithme CMA-ES, pour lequel on a utilisé le code Matlab de N. Hansen, disponible sur son site Web [72]. Deux variantes du CMA-ES ont été utilisées, le meilleur algorithme CMA-ES, qui utilise $\lfloor \frac{\lambda}{2} \rfloor$ parents comme valeur par défaut, et un $(1, \lambda)$ -CMA-ES, pour une comparaison plus juste puisque un seul parent est utilisé pour LS-CMA-ES ici. Les équations définissant les fonctions tests sont données dans le tableau 2.5.2. Tous les tests ont été effectués indépendamment 100 fois, à partir d'un même point initial $(5, 5, \dots, 5)$ et d'un pas initial de 1. A chaque génération, le minimum, la médiane et le maximum de ces 100 essais ont été calculés.

Premiers résultats : La fonction elliptique f_{elli} et la fonction Rosenbrock f_{Ros} ont fait l'objet de nombreux tests numériques. *A priori*, si les hypothèses de la Section 2.2 à propos de l'optimalité de $(\frac{1}{2}H)^{-1}$ en tant que matrice de covariance sont correctes, l'algorithme LS-CMA-ES devrait être meilleur sur la fonction elliptique. La fonction Rosenbrock, quand à elle, est assez différente, en particulier nous avons constaté que l'erreur d'approximation Q définie à l'équation 2.4.4 est plus importante.

Cas elliptique Les résultats pour f_{elli} (Figure 2.3) sont très clairs et sont en accord avec les prédictions : les trois groupes de trois courbes différents, de gauche à droite sont respectivement le min, la médiane et le max pour LS-CMA-ES, ensuite le min, la médiane et le max pour le CMA-ES avec les paramètres par défauts habituels et enfin, le min, la médiane et le max pour le $(1, \lambda)$ -CMA-ES. Les variances sont très petites pour les trois algorithmes. Pour les deux algorithmes CMA-ES, le plateau avant de plonger vers 0 est le temps nécessaire pour apprendre la matrice de covariance correcte – alors que LS-CMA-ES a cette bonne matrice dès le début, quelle que soit la dimension (seul le nombre de points minimal pour apprendre la première matrice est nécessaire). Notons, tout de même que de tels plateaux peuvent faire une grande différence dans les résultats en pratique, lorsque le critère d'arrêt est donné par un coût de calcul ou lorsque la valeur du minimum n'est pas connue.

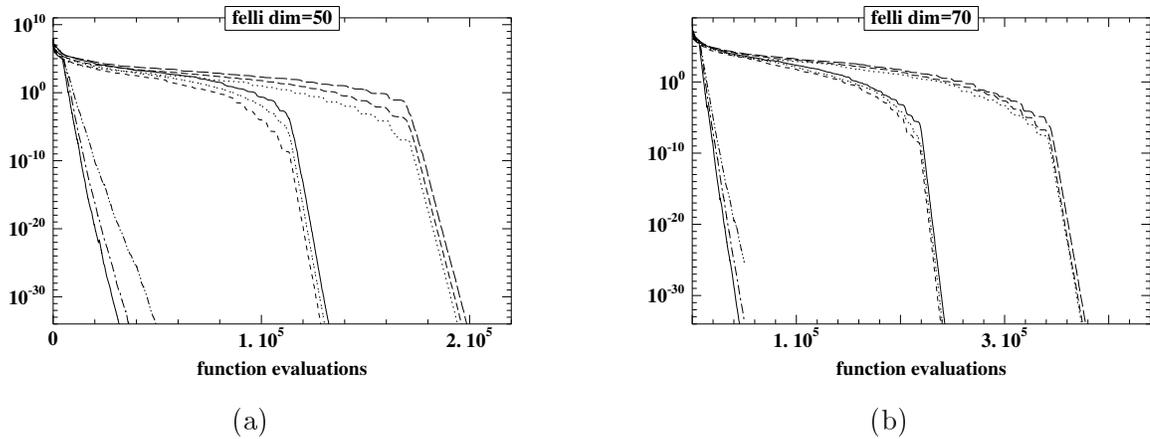


FIG. 2.3: Résultats comparatifs pour la fonction elliptique pour les dimensions 50 et 70 de gauche à droite min, median et max pour $(1, \lambda)$ -LS-CMA-ES, $(1, \lambda)$ -CMA-ES et (μ/μ) -CMA-ES. Le mode LS est toujours actif.

		LS-CMA-ES			CMA-ES			$(1, \lambda)$ -CMA		
Fonction	dimension	min	med	max	min	med	max	min	med	max
f_{Ros}	20	0.98	6.03	–	1.4	2.3	–	1.4	3.1	–
	50	1.1	30	–	9.5	22	–	9.5	12	–

TAB. 2.2: Résultats comparatifs pour la fonction Rosenbrock. Les chiffres représentent le nombre d'évaluation de fonctions (une unité = 10^4) avant que le min, med, et max n'atteigne 10^{-10} (ou – lorsqu'il ne l'atteint jamais en 10^5 évaluations.)

Rosenbrock : Pour la fonction Rosenbrock, les choses se passent différemment : cette fonction n'est pas unimodale, et de plus présente une vallée très plate entre le minimum local et le minimum global. Pour cette fonction LS n'est pas actif et LS-CMA-ES se comporte comme CMA-ES, la différence qui peut être observée dans le tableau 2.5.2 vient de la différence dans l'adaptation des pas de la mutation qui est auto-adaptatif pour le LS-CMA-ES et dérandomisée pour le CMA-ES. Notons que puisque la fonction est multi-modale, certains essais n'ont pas convergé vers l'optimum global (– pour le max).

Autres fonctions D'autres fonctions, aussi utilisées dans [77, 76] (à l'exception de f_{exp}), ont été testées pour valider ces premières conclusions. Certains de ces résultats sont montrés au tableau 2.5.2. Les tendances observées sur la fonction elliptique et sur la fonction Ro-

Nom	Fonction
Elliptique	$f_{elli} = \sum_{i=1}^d (10^6)^{\frac{i-1}{d-1}} x_i^2$
Rosenbrock	$f_{Ros} = \sum_{i=1}^{d-1} 100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2$
Cigar-Tablet	$f_{cigtab} = x_1^2 + \sum_2^{d-1} 10^4 x_i^2 + 10^8 x_d^2$
Tablet	$f_{tablet} = 10^6 x_1^2 + \sum_2^d x_i^2$
Cigar	$f_{cigar} = x_1^2 + \sum_2^d 10^6 x_i^2$
diff-pow	$f_{diff-pow} = \sum_1^d x_i ^{2+10\frac{i-1}{d-1}}$
Exp	$f_{exp} = \exp(\ x\ ^2) - 1$

TAB. 2.3: Fonctions tests unimodales et fonction Rosenbrock

Fonction	(1, λ)-LS-CMA-ES		CMA-ES		(1, λ)-CMA	
	moyenne	écart-type	moyenne	écart-type	moyenne	écart-type
f_{elli}	5405	395	21500	210	26300	190
f_{cigtab}	6030	400	13048	350	25034	412
f_{tablet}	4758	490	14070	856	26085	1104
f_{cigar}	5220	345	9020	203	10108	189
$f_{diff-pow}$	22241	4400	11360	612	22308	500
f_{exp}	2600	205	2633	211	4368	412

TAB. 2.4: Résultats comparatifs pour différentes fonctions tests en dimension 20. Les chiffres représentent le nombre d'évaluation moyen (sur 100 essais) de fonctions avant que l'algorithme ait atteint 10^{-10} .

senbrock sont confirmées : f_{cigar} , f_{tablet} et f_{cigtab} sont des fonctions elliptiques, sur lesquelles l'algorithme LS-CMA-ES est meilleur que l'algorithme CMA-ES. D'autre part, $f_{diff-pow}$ et f_{exp} ne sont pas elliptiques, f_{exp} est même "infiniment plate (toute les dérivées sont nulles au niveau du minimum 0), et les résultats sont très similaires à ceux de la fonction Rosenbrock. Notons toutefois que, alors que LS-CMA-ES a des performances quasiment identiques à celles de (1, λ)-CMA-ES pour la fonction $f_{diff-pow}$, il atteint celles du $(\mu/\mu, \lambda)$ -CMA-ES pour la fonction f_{exp} .

Discussion Ces résultats expérimentaux montrent des améliorations notables des performances de LS-CMA-ES par rapport à CMA-ES sur des fonctions elliptiques. Ceci n'est pas

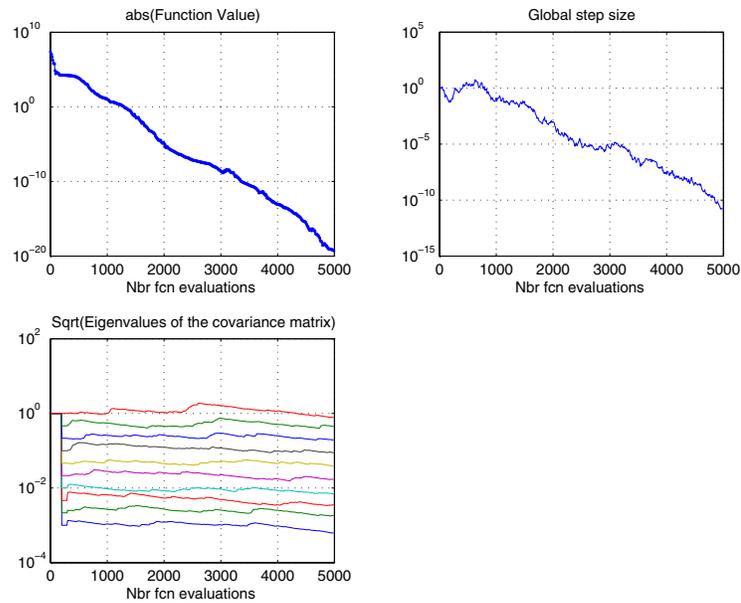


FIG. 2.4: $(1, \lambda)$ -LS-CMA-ES : Fitness, pas global de la mutation, et valeurs de la matrice de covariance sur f_{elli} pour $d = 10$, pour un essai typique.

surprenant si on se réfère aux méthodes d'optimisation classiques: CMA utilise en fait une sorte d'information d'ordre 1 pour mettre à jour sa matrice de covariance, en rajoutant une matrice de rang 1 (plus une autre matrice, cf Section 1.3.3.3 du chapitre précédent) dont l'axe propre est orienté selon p_c , qui représente une direction de descente: en ce sens il peut être vu comme un algorithme proche d'un algorithme de gradient. L'algorithme LS-CMA-ES, quant à lui, fait usage d'une information d'ordre 2 avec l'approximation de la courbure de la fonction donnée à travers la matrice Hessienne. Il peut en conséquence être vu comme une méthode de type quasi Newton.

Visualisation des matrices de covariances Les figures 2.4 et 2.5 illustrent sur la fonction elliptique f_{elli} en dimension 10 pour un essai de l'algorithme, la valeur de la fonction, les valeurs du pas global ainsi que les valeurs propres de la matrice de covariance en fonction du nombre de générations pour respectivement les algorithmes $(1, \lambda)$ -LS-CMA-ES et CMA-ES (avec les valeurs par défaut pour le nombre de parents et d'enfants). On voit dans le cas du $(1, \lambda)$ -LS-CMA-ES, que l'inverse de la matrice hessienne est correctement apprise dès la première minimisation au sens des moindres carrés effectuée (à savoir après, 100 générations). Dans le cas du CMA-ES, une matrice proportionnelle à l'inverse de la hessienne est apprise,

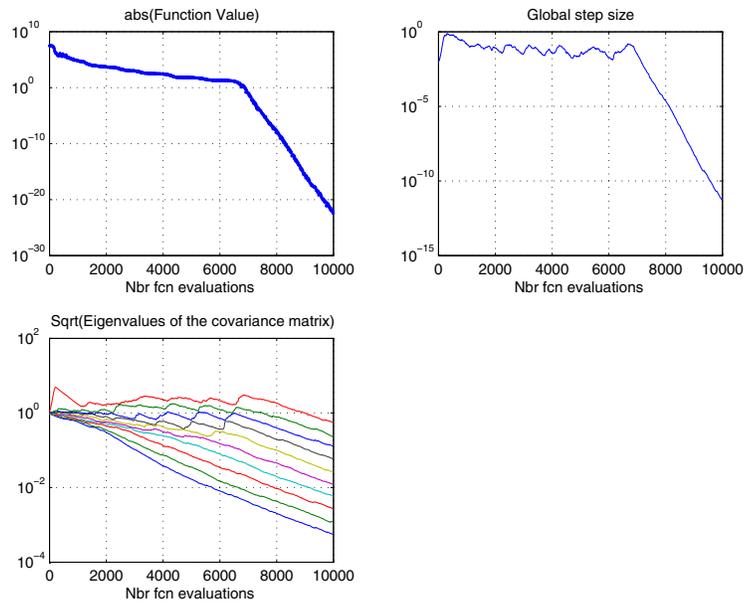


FIG. 2.5: $(\mu/\mu, \lambda)$ -CMA-ES : *Fitness, pas global de la mutation et valeurs de la matrice de covariance sur f_{elli} pour $d = 10$, pour un essai typique.*

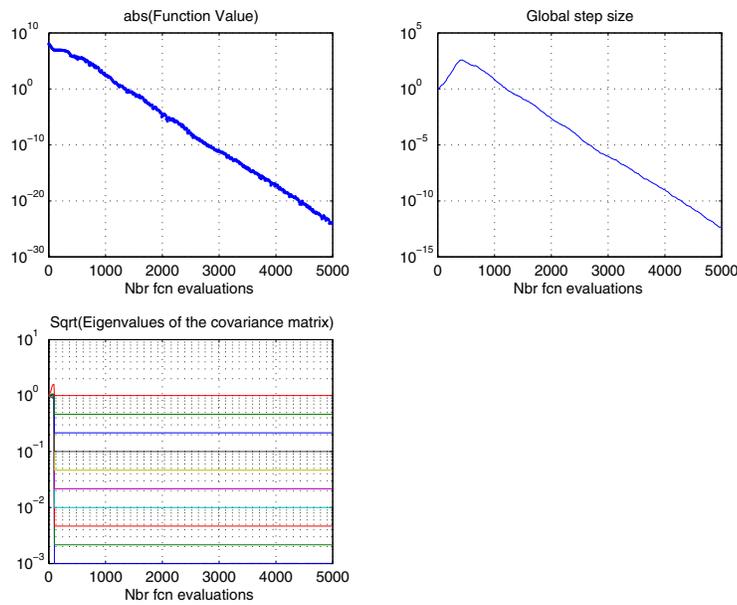


FIG. 2.6: $(\mu/\mu, \lambda)$ -LS-CMA-ES : *Fitness, pas global de la mutation et valeurs de la matrice de covariance sur f_{elli} pour $d = 10$, pour un essai typique.*

un peu avant la génération 7000. Le coefficient de proportionnalité dépend du temps et plus précisément décroît vers 0.

Pour autant, les améliorations ont simplement été démontrées dans un cas de fonctions unimodales ou quasiunimodales alors que les algorithmes évolutionnaires sont plutôt des méthodes d'optimisation globales dans un contexte multimodal. Pour rendre l'algorithme compétitif face à d'autres méthodes d'optimisation globales nous avons testé un algorithme à plusieurs parents, le $(\mu/\mu_I, \lambda)$ -LS-CMA-ES. Les tests sur cet algorithme sont présentés dans la section suivante.

2.5.3 L'algorithme $(\mu/\mu, \lambda)$ -LS-CMA-ES

Nous nous sommes jusqu'à présent concentrés sur l'aspect convergence locale de l'algorithme LS-CMA-ES, mettant en avant des améliorations dans le cas elliptique. Tester l'algorithme LS-CMA-ES sur des fonctions multimodales nécessite l'introduction d'un algorithme avec plusieurs parents pour la phase d'exploration. Le choix d'une matrice de covariance commune à l'ensemble de la population est l'extension naturelle de l'algorithme que nous venons de voir et c'est celle que nous envisageons ici. Notons qu'une méthode utilisant des *clusters* de l'ensemble des points, avec la mise à jour d'une matrice de covariance par *cluster*, serait aussi une technique possible que nous n'avons pas testée pour le moment.

Il semble ici plus naturel d'utiliser une technique dérandomisée, introduite à la Section 1.3.3.2 pour l'adaptation du pas global de la mutation. Les avantages d'une telle technique par rapport à une auto-adaptation du paramètre de la mutation ont été décrite en détail à la Section 1.3.3.1 du chapitre précédent.

Ainsi l'algorithme $(\mu/\mu, \lambda)$ -LS-CMA-ES que nous considérons ici utilise maintenant la mise à jour donnée par les équations 1.3.10 et 1.3.11 données au chapitre précédent pour la mise à jour du pas de la mutation (la ligne 4 du pseudo-code est remplacée par ces équations). De même lorsque l'on est en mode CMA, la mise à jour de la matrice utilise maintenant le fait que l'on a plusieurs parents, et l'équation de la ligne 10 du pseudo code est remplacé par l'équation (1.3.22) donnée au chapitre précédent pour l'algorithme CMA.

Validation par rapport au $(1, \lambda)$ -LS-CMA-ES Nous commençons dans un premier temps par comparer les algorithmes $(\mu/\mu, \lambda)$ -LS-CMA-ES et $(1, \lambda)$ -LS-CMA-ES. Tout d'abord, la Figure 2.6 montre un essai de l'algorithme $(\mu/\mu, \lambda)$ -LS-CMA-ES sur f_{elli} , cette figure est à comparer aux Figures 2.4 et 2.5. On remarque que l'adaptation dérandomisée, ainsi que

Fonction	(1, λ)-LS-CMA-ES		(μ/μ , λ)-LS-CMA-ES	
	moyenne	écart-type	moyenne	écart-type
$f_{elli} = \sum_{i=1}^d (10^6)^{\frac{i-1}{d-1}} x_i^2$	5405	395	5800	158
$f_{cigtab} = x_1^2 + \sum_2^{d-1} 10^4 x_i^2 + 10^8 x_d^2$	6030	400	6302	142
$f_{tablet} = 10^6 x_1^2 + \sum_2^d x_i^2$	4758	490	5064	202
$f_{cigar} = x_1^2 + \sum_2^d 10^6 x_i^2$	5220	345	5630	173
$f_{diff-pow} = \sum_1^d x_i ^{2+10 \frac{i-1}{d-1}}$	22241	4400	14708	935

FIG. 2.7: Résultats comparatifs pour différentes fonctions tests en dimension 20. Les chiffres représentent le nombre d'évaluation moyen (sur 100 essais) de fonctions avant que l'algorithme n'atteigne 10^{-10} , ainsi que l'écart-type.

l'introduction de plusieurs parents atténue comme on pouvait s'y attendre les fluctuations observées à la Figure 2.4 pour l'algorithme (1, λ)-LS-CMA-ES au niveau de l'adaptation du pas, mais aussi des valeurs propres de la matrice de covariance.

Ensuite, la figure 2.7 illustre sur les mêmes fonctions qui nous ont servi à valider précédemment l'algorithme (1, λ)-LS-CMA-ES, le nombre moyen d'évaluation avant d'atteindre le critère d'arrêt f_{stop} pour respectivement les algorithmes (1, λ)-LS-CMA-ES et (μ/μ , λ)-LS-CMA-ES. On remarque qu'en moyenne les performances sont légèrement meilleures pour le (1, λ)-LS-CMA-ES mais avec un écart-type plus élevé (dû au caractère "plus stochastique" du premier algorithme) sauf pour la fonction $f_{diff-pow}$ pour laquelle principalement le mode CMA est actif, ce qui justifie que le step-size dérandomisé fonctionne mieux.

Sur des fonctions multimodales On teste ici les performances du (μ/μ , λ)-LS-CMA-ES sur des fonctions multimodales. Ces tests sont simplement des premiers tests pour illustrer que l'algorithme LS-CMA-ES peut également présenter des avantages dans le cas de fonctions multimodales. Les deux fonctions du tableau 2.5.3 ont été testées. Elles correspondent à des fonctions pour lesquelles l'algorithme CMA est l'algorithme de référence [75], moyennant une augmentation de la taille de la population (par rapport aux valeurs par défaut) et un pas initial suffisamment élevé. Pour ces fonctions multimodales, on n'atteint pas toujours l'optimum global et les valeurs des minima locaux atteints diffèrent d'un essai à l'autre, c'est pourquoi nous ne montrons pas ici des moyennes et écart-types pour atteindre une valeur limite mais montrons 10 essais typiques des algorithmes. Aucun effort particulier n'a été fait

Nom	Fonction
Rastringin	$f_{Rast} = 10d + \sum_{i=1}^d (x_i^2 - 10 \cos(2\pi x_i))$
Scaled Rastringin	$f_{ScalRast} = 10d + \sum_{i=1}^d (((10^6)^{\frac{i-1}{d-1}} x_i)^2 - 10 \cos(2\pi (10^6)^{\frac{i-1}{d-1}} x_i))$

TAB. 2.5: *Fonctions tests multimodales : toutes les fonctions ont un minimum localisé en 0 qui a pour valeur 0*

pour régler les paramètres de l'algorithme LS-CMA-ES, nous prenons simplement une taille de population plus élevée que dans le cas unimodal, le pas initial est aussi augmenté par rapport au cas unimodal, la valeur de $Q_t h$ a été fixée à 10^{-2} . Le réglage pour n_{upd} est de 50 générations. Il a été montré dans [75] que, pour que l'algorithme CMA-ES présente des performances optimales, la taille de la population doit être augmentée, ainsi que le pas de la distribution initiale. C'est ce que nous ferons par la suite. La Figure 2.8 présente pour la fonction $f_{ScalRast}$ en dimension 10, 10 essais des algorithmes LS-CMA-ES et CMA-ES. La distribution initiale de la population est tirée avec une variable aléatoire gaussienne isotrope de moyenne nulle et d'écart-type 5. Les 10 courbes de gauches pour les figures (a) et (b) représentent 10 essais de l'algorithme LS-CMA pour une taille de population de 20. Pour (a), la taille de la population pour l'algorithme CMA est également de 20 et est de 100 pour la courbe de gauche. On voit clairement que les performances de l'algorithme LS-CMA-ES sont meilleures que celles de l'algorithme CMA. Pour la dimension 20, les résultats sont illustrés sur la figure 2.9. Les réglages sont identiques excepté la taille de la population (voir la légende de la figure). Une courbe typique pour voir le changement entre le mode LS et le mode CMA est illustré sur la figure 2.10, les conditions expérimentales sont celles de la figure 2.9. On peut voir dans ce cas là c'est principalement le mode *LS* qui est actif, sauf dans le cas où la valeur de la fonction stagne.

Pour la fonction f_{Rast} , en revanche le mode *LS* n'est jamais actif (valeur de Q au dessus de 10^3) et les performances sont identiques entre les deux algorithmes.

2.6 Robustesse au bruit

Le bruit apparaît naturellement dans de nombreux domaines. Nous présentons ici des premiers tests pour évaluer la robustesse au bruit des algorithmes présentés précédemment. Plus précisément, la robustesse de la convergence locale est testée sur les fonctions décrites

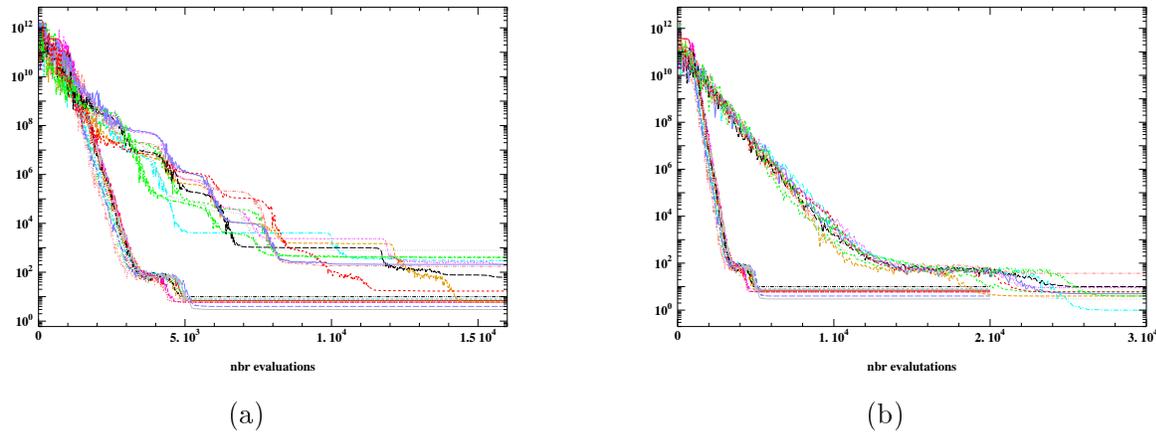


FIG. 2.8: *LS-CMA-ES vs CMA-ES, pour f_{Rast} avec $d = 10$. La distribution initiale de la population est tirée avec une variable aléatoire gaussienne isotrope de moyenne nulle et d'écart-type 5. Les 10 courbes de gauches pour les figures (a) et (b) représentent 10 essais de l'algorithme LS-CMA pour une taille de population de 20. Pour (a), la taille de la population pour l'algorithme CMA est également de 20 et est de 100 pour la courbe de gauche*

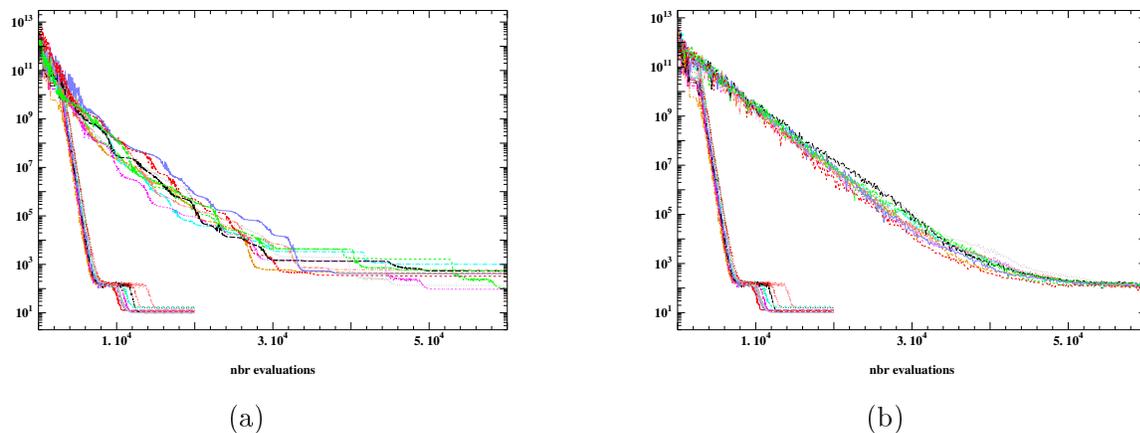


FIG. 2.9: *LS-CMA-ES vs CMA-ES, pour f_{Rast} avec $d = 20$. La distribution initiale de la population est tirée avec une variable aléatoire gaussienne isotrope de moyenne nulle et d'écart-type 5. Les 10 courbes de gauches pour les figures (a) et (b) représentent 10 essais de l'algorithme LS-CMA pour une taille de population de 30. Pour (a), la taille de la population pour l'algorithme CMA est également de 30 et est de 150 pour la courbe de gauche*

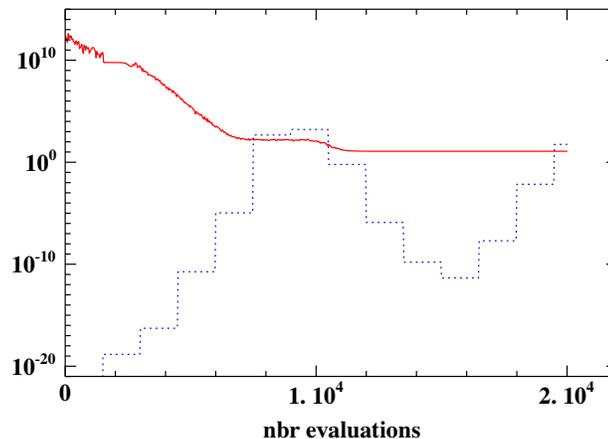


FIG. 2.10: *Essai typique, conditions expérimentales de la figure 2.9. La fitness (courbe du haut) et les valeurs de \mathcal{Q} (courbe du bas) sont tracés en fonction du nombre de générations.*

dans le tableau 2.11 correspondant tout d'abord à une fonction elliptique avec un bruit blanc gaussien additif et ensuite à cette même fonction elliptique avec 3 niveaux de bruit gaussien différents, réglés par l'écart-type des lois normales. Nous allons noter ce dernier, $niveau_{br}$, il vaut respectivement 10^{-6} , 10^{-4} et 10^{-2} . Notons que le bruit est normalisé de façon à s'appliquer dans toutes les directions.

La valeur que l'on va pouvoir atteindre est de l'ordre de $d\|N(0, 1)\|^2 \times niveau_{br}$. Or pour d grand, $E(\|N(0, 1)\|)$ est de l'ordre de \sqrt{d} , ce qui nous donne comme approximation de la valeur limite que l'on peut espérer atteindre : $d^2 \times niveau_{br}$. Dans notre cas $d = 20$, ce qui fixe les valeurs de f_{stop} (valeurs d'arr pour l'algorithme) à respectivement 10^{-10} , 10^{-6} et 10^{-2} .

Les résultats sont présentés dans le tableau 2.6. Le nombre d'évaluation moyen nécessaire pour atteindre f_{stop} est donné pour respectivement $(\mu/\mu, \lambda)$ -LS-CMA-ES et $(\mu/\mu, \lambda)$ -CMA-ES. Les moyennes ont été effectuées sur 100 essais. Dans le cas du LS-CMA-ES, un réglage supplémentaire doit être effectué, celui de la valeur \mathcal{Q}_{th} qui a été fixé pour les 3 fonctions à 10^{-2} (après plusieurs essais). Les trois premières lignes du tableau présentent les résultats pour les valeurs de f_{stop} données plus haut. Comme par ailleurs, ces valeurs là ne sont pas atteintes par l'algorithme CMA, les trois lignes suivantes du tableau donnent les mêmes résultats pour trois valeurs plus élevées de f_{stop} , afin de pouvoir comparer les algorithmes CMA et LS-CMA.

On peut remarquer d'une part (lignes 4 à 6 du tableau) que l'algorithme LS-CMA-ES

garde son avantage sur l'algorithme CMA-ES, c'est à dire que dans le cas bruité, il continue à présenter des meilleures performances dans le cas elliptique. Par ailleurs, on peut noter que contrairement à l'algorithme CMA, l'algorithme LS-CMA-ES atteint toujours la limite du bruit théorique. Notons que le fait que CMA-ES n'atteigne pas la limite du bruit n'est pas très surprenant au vu des courbes de la figure 2.5 ou de la figure 2.2 montrant que les valeurs propres de la matrice de covariance décroissent vers 0, contrairement à l'algorithme LS-CMA. Ainsi lorsque ces valeurs propres passent au dessous du seuil du bruit, l'algorithme CMA-ES reste bloqué.

Par ailleurs, comparant les 6 premières lignes du tableau aux 5 avant-dernières (correspondant au nombre moyen d'évaluations dans un cas non bruité), on peut noter que le nombre moyen d'évaluations de fonction pour atteindre f_{stop} dans le cas où la fonction est bruitée (fonctions $f_{elli}^{b,1}$, $f_{elli}^{b,2}$ et $f_{elli}^{b,3}$) est très légèrement supérieur au nombre moyen d'évaluations nécessaires pour atteindre f_{stop} pour la même fonction non bruitée, ceci pour les 2 algorithmes LS-CMA-ES et CMA-ES.

Le cas où le bruit est additif (dernière ligne du tableau) est identique.

Pour tenter de comprendre, pourquoi l'algorithme LS-CMA est plus robuste au bruit que CMA, écrivons ce que devient le problème de minimisation au sens des moindres carrés dans le cas d'un bruit additif. Le bruit étant additif, l'équation 2.4.3 devient maintenant

$$\min_Y \|AY - F + n_{br}N(0, 1)\|_2^2 \quad (2.6.5)$$

où $N(0, 1)$ est un vecteur gaussien centré dont les coordonnées sont les bruits associés à chaque individu, indépendants les uns des autres. Cette équation se réécrit

$$\min_Y (\|AY - F\|_2^2 + \|n_{br}N(0, 1)\|_2^2 + 2 \langle AY - F, n_{br}N(0, 1) \rangle_2) \quad (2.6.6)$$

Le terme du milieu ne change pas le problème de minimisation et nous avons vérifié numériquement sur 10 essais que le dernier terme était négligeable. Plus précisément pour la fonction $f_{elli}^{b,0}$ avec $n_{br} = 10^{-4}$ correspondant au niveau de bruit additif testé dans cette section (cf tableau 2.6) et pour la même dimension, nous avons obtenu une valeur de l'ordre de 10^{-5} pour $\|AY - F\|_2^2$ et de l'ordre de 10^{-9} pour $\langle AY - F, n_{br}N(0, 1) \rangle_2$ (sur les tests effectués, les valeurs de $\|AY - F\|_2^2$ et $\langle AY - F, n_{br}N(0, 1) \rangle_2$, calculées à chaque fois qu'une minimisation au sens des moindres carrés est effectuée, garde le même ordre de grandeur tout au long de l'algorithme). Ainsi, pour les tests effectués, le dernier terme de l'équation 2.6.6 est négligeable

Fonctions
$f_{elli}^{b,0} = \sum_{i=1}^d (10^4)^{\frac{i-1}{d-1}} (x_i)^2$
$f_{elli}^{b,a} = \sum_{i=1}^d (10^4)^{\frac{i-1}{d-1}} (x_i)^2 + \mathbf{10}^{-4} N(0, 1)$
$f_{elli}^{b,1} = \sum_{i=1}^d (10^4)^{\frac{i-1}{d-1}} \left(x_i + \frac{\mathbf{10}^{-6} N_i(0,1)}{(10^2)^{\frac{i-1}{d-1}}} \right)^2$
$f_{elli}^{b,2} = \sum_{i=1}^d (10^4)^{\frac{i-1}{d-1}} \left(x_i + \frac{\mathbf{10}^{-4} N_i(0,1)}{(10^2)^{\frac{i-1}{d-1}}} \right)^2$
$f_{elli}^{b,3} = \sum_{i=1}^d (10^4)^{\frac{i-1}{d-1}} \left(x_i + \frac{\mathbf{10}^{-2} N_i(0,1)}{(10^2)^{\frac{i-1}{d-1}}} \right)^2$

FIG. 2.11: *Fonctions avec du bruit*

par rapport au premier, expliquant que le bruit n'a pas d'influence sur la solution trouvée au problème de minimisation au sens des moindres carrés. Bien sûr des tests plus poussés devront être effectués pour valider ces premières explications.

2.7 Discussion et perspectives

Un inconvénient de l'algorithme est son coût : comme nous l'avons mentionné à la Section 2.4, la résolution directe de la minimisation au sens des moindres carrés a une complexité en $o(d^6)$. La mise en oeuvre de l'algorithme a pour l'instant été réalisée (avec Matlab) en utilisant une procédure directe pour la résolution du problème de minimisation au sens des moindres carrés, reposant sur une décomposition QR de la matrice $A^T A$, de taille $(\frac{3d+d^2}{2}, \frac{3d+d^2}{2})$. Le coût que nous avons mesuré pour un essai est le suivant : sur un Pentium 3.0GHz, il faut à peu près, pour un calcul de l'approximation de la Hessienne par méthode des moindres carrés et pour son inversion, 1, 280, et 1320 secondes respectivement en dimension 20, 50 et 70.

Une première remarque est qu'en grande dimension, ce coût supplémentaire peut probablement être réduit en utilisant une procédure itérative pour la résolution de l'équation 2.4.2.

En revanche, dans le cas d'une fonction chère à calculer (ce qui est le cas dans de nombreux problèmes industriels), l'algorithme que nous venons de présenter permet, sans avoir d'évaluations de fonctions supplémentaires à faire, d'utiliser plus d'information sur la fonction (à savoir sa courbure locale), si toutefois l'approximation au sens des moindres carrés a permis d'obtenir une hessienne suffisamment proche de la vraie hessienne. Le fait que le critère Q_{th} discrimine les cas où l'approximation est bonne de ceux où elle ne l'est pas permet par

		$(\mu/\mu, \lambda)$ -LS-CMA-ES		$(\mu/\mu, \lambda)$ -CMA-ES	
Fonction	f_{stop}	moyenne	écart-type	moyenne	écart-type
$f_{elli}^{b,1}$	10^{-10}	5226	124	–	–
$f_{elli}^{b,2}$	10^{-6}	4160	119	–	–
$f_{elli}^{b,3}$	10^{-2}	3157	146	★ 0.40	
$f_{elli}^{b,1}$	10^{-6}	4124	119	11743	365
$f_{elli}^{b,2}$	10^{-4}	3585	108	11003	414
$f_{elli}^{b,3}$	10^{-1}	2856	145	★ 0.23	
$f_{elli}^{b,0}$	10^{-10}	5173	128	12892	310
	10^{-6}	4105	117	11677	353
	10^{-4}	3568	109	10968	418
	10^{-2}	3026	97	9775	601
	10^{-1}	2762	98	8860	641
$f_{elli}^{b,a}$	10^{-2}	3130	115	12120	450

TAB. 2.6: Nombre moyen d'évaluations de fonction nécessaires pour atteindre f_{stop} , le symbole – indique qu'aucun des 100 essais n'a atteint cette valeur et le symbole ★ p_e que la proportion d'essais n'ayant pas atteint cette valeur est p_e . Tous les tests ont été effectués en dimension 20.

ailleurs de choisir une autre stratégie (la stratégie CMA) lorsque cette approximation n'est pas bonne. Même si les améliorations ont surtout été montrées dans des cas de fonctions elliptiques, les bons résultats que l'on peut obtenir sur les fonctions multimodales montrent que la méthode ne se restreint pas au cas elliptique. La robustesse au bruit de l'algorithme est aussi cruciale en vue d'applications à des problèmes réels.

Dans le cas de la fonction Rosenbrock, l'approximation de la hessienne donnée par la méthode des moindres carrés n'est pas suffisamment bonne pour être utilisée efficacement, cependant nous avons simplement considéré une approximation à l'ordre 2 de la fonction. Envisager de chercher, par la même méthode, une approximation de la fonction d'un ordre supérieur à l'ordre 2 (même si le coût de la minimisation au sens des moindres carrés sera d'autant plus élevé) permettrait sans doute de rendre exploitable l'approximation de la hessienne obtenue.

Cependant, au delà du simple algorithme que nous venons de présenter, se trouve aussi l'idée que l'apprentissage de l'inverse de la hessienne comme matrice de covariance est une idée essentielle pour la construction d'opérateurs de mutations efficaces. Cette méthode permet de le faire sans évaluations de fonctions supplémentaires et fonctionne même dans le cas de fonctions bruitées. L'algorithme CMA-ES, dans ces cas là, a besoin de plus d'évaluations de fonctions pour parvenir au même résultat.

Pour continuer dans cette voie là, notons que les techniques d'optimisation classique de type DFP et BFGS permettent, moyennant la connaissance du gradient de la fonction, de connaître l'inverse de la hessienne d'une fonction elliptique en d itérations. La recherche de directions conjuguées pour le produit scalaire induit par la forme quadratique sous-jacente à une fonction elliptique de matrice hessienne symétrique définie positive, est aussi un moyen de construire, en d itérations, par addition de matrices de rang 1 ayant pour axes propres ces directions conjuguées, l'inverse de la hessienne. Transposer certaines de ces techniques, en utilisant une approximation du gradient donnée par la simple connaissance des points et de la valeur de la fonction correspondante, est certainement envisageable.

Enfin, un parallèle peut être fait entre la méthode que nous venons de présenter et les méthodes où l'idée est de chercher une approximation de la fonction (coûteuse en général) que l'on cherche à optimiser pour minimiser cette approximation avant de calculer une nouvelle approximation et réitérer le processus. Un résumé des différentes techniques existantes se trouve dans [85]. En particulier, l'article récent de Poland [119] considère un modèle quadratique pour la fonction approchant la fonction à optimiser. Il utilise également une minimisation

au sens des moindres carrés pour trouver cette approximation quadratique. Une fois qu'il a un modèle quadratique approché, il utilise le centre de ce modèle comme nouveau parent de son algorithme évolutionnaire. Mais les améliorations de cet algorithme par rapport à l'algorithme CMA-ES n'ont pour l'instant été montrées que sur la fonction sphère et la fonction Rosenbrock.

Deuxième partie

**Etudes théoriques des Algorithmes
Evolutionnaires sur un espace
continu**

Chapitre 3

Introduction

Nous abordons dans cette partie certains aspects théoriques des algorithmes considérés dans la première partie de cette thèse.

Il existe à l'heure actuelle peu d'études théoriques de la convergence des algorithmes évolutionnaires continus. A cela deux raisons principales, d'une part, la conception des algorithmes évolutionnaires n'est pas liée à des études mathématiques préalables mais à une démarche plus intuitive et empirique, d'autre part les études théoriques existantes concernant les algorithmes évolutionnaires portent essentiellement sur des algorithmes sur un espace d'état discret. Un rapide état de l'art de ces résultats a été donné dans l'introduction de cette thèse. Pourtant, l'enjeu de telles études théoriques est important : comme pour toute justification mathématique, cela permet d'apporter de la crédibilité aux algorithmes évolutionnaires mais aussi, cela permet de fournir des bases de comparaisons pour les différents algorithmes existants, base de comparaison fondée à l'heure actuelle uniquement sur l'expérimentation des algorithmes sur des séries de fonctions tests.

D'un point de vue mathématique, les algorithmes évolutionnaires sont des algorithmes stochastiques, et il existe toute une littérature concernant l'étude de certaines classes d'algorithmes. Citons en particulier les livres de Duflo [53, 52] ou le livre de Benveniste, Métivier et Priouret [21]. Même si à notre connaissance les algorithmes évolutionnaires qui nous intéressent n'ont jamais été pris en compte dans les études théoriques de ces algorithmes stochastiques, les méthodes développées pour ces études sont pertinentes pour ce qui nous intéresse et nous allons utiliser dans cette partie deux techniques classiques pour l'étude des algorithmes stochastiques. Nous allons plus particulièrement faire usage dans les deux chapitres qui suivent, d'une part de la théorie des martingales, outil mathématique puissant pour l'analyse d'algorithmes stochastiques et d'autre part de la théorie des chaînes de Markov.

Les deux études théoriques que nous présentons au chapitre 4 et chapitre 5 concernent l'étude d'algorithmes évolutionnaires adaptatifs et auto-adaptatifs. Rappelons que pour ces algorithmes le terme *adaptatif* désigne l'adaptation aux phases d'exploration et d'exploitation (recherche locale) lors de la recherche d'un minimum. Pour les algorithmes que nous allons plus particulièrement étudier, l'adaptation se fait grâce à la modification au cours de l'évolution du pas de la mutation.

Expérimentalement, ces algorithmes montrent sur le modèle le plus simple de fonction qu'est la fonction sphère, une convergence log-linéaire¹ accompagnée d'une convergence log-

1. Soit f la fonction à minimiser, f_n la plus petite fitness à la génération n , asymptotiquement $\ln(f_n)$ se comporte comme n .

linéaire également du pas de la mutation. La première étude que nous présentons s'intéresse à la convergence de deux algorithmes adaptatifs dans le cas de fonctions de régularité C^2 . L'analyse théorique est faite à l'aide de la théorie des martingales.

La deuxième étude s'intéresse à la convergence de l'algorithme auto-adaptatif $(1, \lambda)$ -SA-ES sur la fonction sphère, présenté à la Section 1.3.2 du chapitre 1. Pour cette étude, nous allons exploiter la dynamique markovienne de l'algorithme et utiliser plus précisément le même type d'approche que pour l'analyse de convergence d'algorithmes d'approximation stochastique [52, 21] ou l'analyse des vitesses de convergence des méthodes de Monte Carlo [143, 137, 98]. Même si toute l'analyse est menée sur le cas particulier de l'algorithme $(1, \lambda)$ -SA-ES, cette étude est générale et peut s'appliquer notamment au cas des algorithmes dérandomisés, algorithme CSA en particulier, présenté au Chapitre 1.

Pour replacer les travaux effectués durant cette thèse dans leur contexte, nous donnons dans la suite un état de l'art théorique des algorithmes évolutionnaires pour l'optimisation continue.

3.1 Algorithmes évolutionnaires continus: Etat de l'art des résultats théoriques

3.1.1 La Théorie du *progress rate*

La majeure partie des travaux théoriques sur les algorithmes évolutionnaires continus utilisent ce qui s'appelle la théorie du *progress rate*. Introduit par les fondateurs de ces algorithmes Rechenberg et Schwefel ([131, 122]) elle a été développée par Beyer dans son livre "The theory of Evolution Strategies" [22]. Le principe du *progress rate* est à mettre en parallèle avec l'évolution historique des algorithmes évolutionnaires: Lorsque Rechenberg et Schwefel ont introduit le *progress rate*, l'enjeu sur un plan pratique était de trouver une façon efficace de régler le pas de la mutation utilisé pour créer de nouveaux enfants à chaque itération. Ceci a entre autre donné naissance à la règle des $1/5$, puis aux mutations auto-adaptatives log-normales. Pour comprendre comment régler le pas d'une itération à l'autre de l'algorithme, Rechenberg et Schwefel étudiaient le *progress rate* défini de la manière suivante:

$$\mathbb{E}(f(X_n) - f(X_{n+1})|X_n) \tag{3.1.1}$$

où f est la fonction que l'on cherche à minimiser et X_n est la variable aléatoire modélisant le parent à la génération n (on se place dans le cadre d'un seul parent). L'idée, à partir

de l'équation (3.1.1) était de trouver les différents paramètres de l'algorithme (pas de la mutation, nombre d'enfants) pour maximiser le *progress rate* (3.1.1)². Dans un cadre général (f quelconque), le *progress rate* (3.1.1) est dur à calculer, cependant, dans le cas de la fonction sphère

$$f_S(x) = \sum_{i=0}^d x_i^2 \quad (3.1.2)$$

les calculs deviennent plus simple. En particulier des estimations asymptotiques en la dimension d de l'espace de recherche ont été faites. Ainsi, les travaux théoriques utilisant le *progress rate* consistent, pour un algorithme donné, à trouver des estimations asymptotiques du *progress rate* (3.1.1) et étudier le comportement vis à vis des différents paramètres de l'algorithme de ces estimations.

Limitations:

Même sur des fonctions très simples (fonction sphère), le *progress rate* (3.1.1) dépend du point X_n et trouver des réglages optimaux indépendants de X_n repose bien souvent sur des approximations peu rigoureuses. Par ailleurs étudier le *progress rate* (3.1.1) ne suffit pas (sauf dans des cas simples que nous détaillerons au chapitre 3) pour prévoir la dynamique de l'algorithme, même si sous certaines approximations, Beyer [22] déduit la convergence de certains algorithmes de l'étude de *progress rate* (3.1.1).

3.1.2 Convergence globale

La terminologie convergence globale est utilisée ici pour désigner les études théoriques s'intéressant à la convergence sur des fonctions sur lesquelles peu d'hypothèses sont faites. En particulier ces fonctions pour les résultats que nous présentons ici sont éventuellement multimodales.

- Dans le cadre d'un $(1 + 1)$ -ES à pas constant, *i.e.*

$$X_{n+1} = \operatorname{argmin}\{f(X_n), f(X_n + \sigma N_n(0, 1))\}$$

où $X_0 \in \mathbb{R}^d$, $\sigma \in \mathbb{R}^+$, $X_0 \in \mathbb{R}$ et $(N_n(0, 1), n \in \mathbb{N})$ sont des variables aléatoires indépendantes suivant une loi normale centrée réduite, Rudolph [125] puis Chonghui et a.l. [38] démontrent que si f est supposée continue définie sur Ω borné³ de \mathbb{R}^d , en notant $f^* = \min\{f(x), x \in \Omega\}$ il y a convergence presque sûre de $f(X_n)$ vers f^* . Dans ces deux travaux, la preuve est faite

2. pour le nombre d'enfants il faut considérer une normalisation de (3.1.1) par le nombre d'enfants

3. Dans ce cas là, on effectue un repliement lorsque $X_n + \sigma N_n(0, 1)$ sort de Ω .

en utilisant le lemme de Borel Cantelli (même si Rudolph ne mentionne pas explicitement le Lemme de Borel Cantelli, il redémontre la partie qui l'intéresse pour sa preuve).

- Dans le même cadre que précédemment, He and Kang [78] montrent que le noyau de transition de la chaîne de Markov $((X_n), n \in \mathbb{N})$ définie par

$$P(x, A) = P(X_1 \in A | X_0 = x)$$

satisfait une condition de Doeblin ($P(x, A) \geq \delta \Phi(A)$, où Φ est une mesure de probabilité sur Ω et $0 < \delta \leq 1$) et ils en déduisent l'existence d'une unique mesure de probabilité invariante μ vers laquelle la loi de X_n , $\mathcal{L}(X_n)$ converge en variation totale quelle que soit la loi initiale de X_0 . Plus précisément le résultat est:

$$\|\mathcal{L}(X_n) - \mu\|_{var} \leq \delta^n$$

quelle que soit la loi initiale $\mathcal{L}(X_0)$.

Limitations: Les résultats précédents reposent d'une part sur le fait que d'une génération à l'autre on ne peut dégrader la solution et d'autre part sur le fait que le support de $N_n(0, 1)$ recouvre tout Ω . Il est facile de construire un contre-exemple à ces résultats avec une loi à support borné pour la mutation. Par ailleurs, la condition très forte de Doeblin est vérifiée ici parce que la mutation est à pas constant mais ne s'applique plus dans un cadre où le pas de mutation n'est pas borné inférieurement.

- Dans le cas où le pas est adapté (et n'est donc plus borné inférieurement), Rudolph prouve qu'il n'y a plus convergence globale. Plus précisément, Rudolph [127] construit un contre-exemple de fonction continue avec un minimum global unique et un (autre) minimum local tel que si l'on part du minimum local, et que l'on considère l'algorithme $(1 + 1)$ -ES avec la règle des $1/5$ pour adapter le pas de la mutation, alors il existe un voisinage du minimum global de mesure (de Lebesgue) strictement positive tel que la probabilité de ne jamais voir ce voisinage étant parti du minimum local est strictement positive.

3.1.3 Convergence locale

Par convergence locale nous désignons les travaux qui montrent la convergence vers un minimum local de la fonction à optimiser ou qui s'intéressent à la convergence sur des fonctions ne présentant qu'un seul minimum. Dans le cadre des algorithmes *adaptatifs* ou *auto-adaptatifs* mis au point pour adapter en particulier le pas de la mutation aux phases d'exploitation et d'exploration, l'étude de la convergence locale relève de la compréhension du

phénomène d'adaptation. Sur la plus simple des fonctions unimodales, la fonction sphère, il est connu des utilisateurs d'algorithmes adaptatifs et auto-adaptatifs que la convergence est log-linéaire et que le pas de la mutation converge également de manière log-linéaire [15, 131].

3.1.3.1 Convergence des algorithmes adaptatifs

La convergence d'algorithmes adaptatifs a été analysée dans [126] d'une part et dans [23] d'autre part. Ces deux algorithmes peuvent s'écrire de la façon suivante

$$X_{n+1} = \arg \min \{f(X_n + \sigma H(X_n)\xi^i), i = 1 \dots \lambda\} \quad (3.1.3)$$

où $\sigma \in \mathbb{R}^+$ et $H : \mathbb{R} \rightarrow \mathbb{R}^+$ est précisée ci-dessous.

- Rudolph [126] analyse la convergence de l'algorithme 3.1.3 avec $H(x) = \|\nabla x\|$ et ξ suit une loi uniforme. Dans le cadre de fonctions (K, Q) -strictement convexes continues différentiables Rudolph démontre la convergence L^1 géométrique de $f(X_n)$ pour un σ suffisamment petit.
- Bienvenüe et François [23] montrent que sur la fonction sphère, pour l'algorithme 3.1.3 avec $H(x) = \|x\|$, asymptotiquement la convergence est log-linéaire, plus précisément ils montrent que

$$\frac{1}{n} \ln(\|X_n\|) \xrightarrow[n \rightarrow \infty]{} \kappa(\sigma, \lambda)$$

où $\kappa(\sigma, \lambda) \in \mathbb{R}$.

3.1.3.2 Convergence des algorithmes auto-adaptatifs

Les premiers travaux concernant la convergence d'algorithmes auto-adaptatifs datent de 2002 [44, 135] et utilisent la théorie des martingales.

- Hart *et al.* [44] considèrent la convergence d'un $(1, \lambda)$ -SA-ES isotrope (cf chapitre 1), où la mutation est discrétisée, c'est à dire qu'à partir du parent X_n à la génération n seulement 6 nouveaux points peuvent être créés (l'espace d'état dans lequel vit $(X_n, n \in \mathbb{N})$ est alors dénombrable, dense dans \mathbb{R}) sur la fonction sphère en dimension 1. Le résultat démontré est qu'il existe λ_0 tel que pour $\lambda \geq \lambda_0$, X_n converge presque sûrement vers 0.

Limitations: Il y a essentiellement deux limitations à ce résultat, d'une part il n'est pas montré que $\lambda_0 \leq 6$ et donc il n'est pas clair que l'algorithme étudié soit plus performant qu'une recherche exhaustive du minimum parmi les 6 points possibles à chaque génération. Par ailleurs aucune vitesse de convergence n'est donnée.

- Semenov et Terkel [135] étudient eux aussi le $(1, \lambda)$ -SA-ES isotrope sur la fonction sphère et considèrent la mutation où les lois normales sont remplacées par des lois uniformes entre -1 et

1. Semenov et Terkel [135] montrent à l'aide de simulations numériques que certaines fonctions sont candidates à vérifier des conditions sous lesquelles ils savent prouver la convergence vers 0.

- Dans [23] Bienvenüe et François montrent que l'on peut déduire la convergence log-linéaire du $(1, \lambda)$ -SA-ES isotrope sur la fonction sphère de l'étude de propriété de la stabilité d'une chaîne de Markov sous-jacente à l'algorithme.

3.2 Contexte des résultats théoriques de cette thèse_____

Les résultats théoriques de cette thèse concernent l'aspect convergence locale des algorithmes adaptatifs et auto-adaptatifs. Dans les algorithmes considérés, seul le pas de la mutation est adapté. Dans le chapitre 4 nous analysons la convergence des algorithmes adaptatifs définis par l'équation (3.1.3) avec respectivement $H(x) = |x|$ et $H(x) = |f'(x)|$ (où f' est la dérivée, en dimension 1). Le premier algorithme, analysé dans [23] sur la fonction sphère est ici analysé pour un cadre plus général de fonctions de régularité C^2 , où, de plus, des hypothèses s'apparentant à de la faible convexité sont faites (en particulier la fonction a un seul optimum). Pour le deuxième algorithme, seule une hypothèse de régularité C^2 est nécessaire dans notre analyse pour conclure à la convergence vers un point critique. Par contre pour obtenir une vitesse de convergence nous avons besoin également d'hypothèses s'apparentant à de la faible convexité.

Le chapitre 5 analyse dans le cadre de la fonction sphère, la convergence de l'algorithme $(1, \lambda)$ -SA-ES isotrope. L'analyse que nous faisons englobe en particulier l'algorithme utilisé en pratique où les lois pour les mutations (loi normale et log-normale) sont absolument continues. L'analyse qui est faite exploite la dynamique markovienne de l'algorithme, elle est inspirée de l'article [23]. Nous étudions précisément les propriétés de stabilité énoncées dans cet article pour conclure à la convergence et montrons une condition, exprimée de manière simple en fonction du moment de la variable aléatoire de la mutation et du nombre d'enfants λ sous laquelle ces propriétés de stabilité sont vérifiées.

L'étude menée sur l'algorithme $(1, \lambda)$ -SA-ES en particulier peut s'étendre également à l'analyse d'algorithmes dérandomisés, comme nous le verrons dans la discussion de ce chapitre.

Chapitre 4

Analyse de la convergence de deux algorithmes adaptatifs

Ce chapitre reprend pour sa majeure partie les résultats publiés dans [9], ainsi que dans le rapport de recherche [8].

[9] A. Auger, C. Le Bris, and M. Schoenauer. *Dimension-independent Convergence Rate for Non-isotropic $(1, \lambda)$ -ES*, In E. Cantu-Paz et al., editor, *Proceedings of the Genetic and Evolutionary Conference 2003*, pages 512–524. LNCS 2723 and 2724, Springer Verlag, 2003.

[8] A. Auger, C. Le Bris, and M. Schoenauer, *Rigorous analysis of some simple adaptive ES*, Rapport de recherche INRIA, RR-4914.

4.1 Introduction

Nous analysons dans ce chapitre la convergence de deux algorithmes évolutionnaires adaptatifs appartenant à la famille des stratégies d'évolutions. Etant donné une fonction f que l'on cherche à minimiser, définie de \mathbb{R} à valeurs dans \mathbb{R} dans un premier temps, et étant donné un entier strictement positif, λ , la forme générale des algorithmes qui nous intéressent est donnée par

$$\begin{cases} X_0 \in \mathbb{R}, \\ X_{n+1} = \arg \min \{f(X_n + \sigma H(X_n)\xi_i^n), \text{ pour } 1 \leq i \leq \lambda\}. \end{cases} \quad (4.1.1)$$

Dans l'équation précédente, $\sigma \in \mathbb{R}_*^+$ et ξ_i^n pour $1 \leq i \leq \lambda$ sont λ variables aléatoires symétriques indépendantes de loi que nous noterons ν_ξ . La fonction H définie sur \mathbb{R} à valeurs dans \mathbb{R} dans un premier temps va être ici $H(x) = |x - x^*|$ où x^* est le minimum global¹ de la fonction à optimiser, puis, dans un deuxième temps, $H(x) = |f'(x)|$, où f' est la dérivée de la fonction f . Notons que sur la fonction sphère ces deux algorithmes sont les mêmes à un facteur 2 près.

Bien que présentant peu d'intérêt d'un point de vue pratique, le premier algorithme a une place particulière dans l'histoire des stratégies d'évolutions. C'est en effet, l'algorithme implicitement considéré dans la plupart des travaux utilisant le *progress rate* [122, 131, 22] sur la fonction sphère. C'est sur cet algorithme que le pas de la mutation σ maximisant le *progress rate* est cherché.

Le deuxième algorithme, quant à lui, présuppose la connaissance de la dérivée de la fonction que l'on optimise, ce qui n'est pas le cadre habituel des algorithmes évolutionnaires. Historiquement, on retrouve cet algorithme à plusieurs reprises. Tout d'abord, cette technique adaptative pour le contrôle du pas de la mutation a été envisagée par Rechenberg [122] et Schwefel [131], avant l'introduction de techniques d'ordre zéro comme la règle des 1/5 ou la mutation du pas de la mutation lui-même (algorithmes SA-ES cf chapitre 1). Ensuite Scheel [129], puis Rudolph [125, 126] montrent que dans le cadre de fonctions (K,Q)-strictement convexes, cet algorithme est optimal au sens du progress rate, c'est à dire que pour cette classe de fonctions, cette technique de contrôle du pas de la mutation va donner la plus grande progression de $E(f(X_{n+1})|X_n) - f(X_n)$.

La convergence de ces algorithmes a été analysée à deux reprises. Tout d'abord, le deuxième algorithme est étudié par Rudolph [126] dans le cadre de fonctions (K,Q)-strictement

1. que l'on suppose unique

convexes continues différentiables et pour ξ suivant une loi uniforme sur la sphère. Il montre une convergence L^1 de $f(X_n)$ pour un σ suffisamment petit. Ensuite Bienvenüe et François [23] démontrent dans le cadre de la fonction sphère la convergence du premier algorithme et donnent une vitesse de convergence.

Le but de cette étude est de généraliser les résultats de [23] à des fonctions de régularité C^2 . Les variables aléatoire pour la mutation ξ que nous allons considérer ne se restreindront pas aux lois uniformes: nous aurons simplement besoin de l'existence d'un moment d'ordre 2. En ce sens l'étude que nous allons faire est plus générale que celle menée dans [126].

Pour l'analyse de ces deux algorithmes, nous allons faire appel à la théorie des martingales. Le schéma de l'analyse de la convergence sera la même dans les deux cas: le point-clé consistera à exhiber une constante que nous appelons Σ pour cette introduction, telle que pour tout σ en dessous de cette constante Σ , la processus $(f(X_n), n \in \mathbb{N})$ soit une sur-martingale. La convergence s'en suivra ainsi que des vitesses de convergence.

Le plan de ce chapitre est le suivant. Après avoir rappelé des définitions et résultats utiles en Section 4.2, nous allons dans un premier temps, à la Section 4.3, nous placer dans le cadre de la fonction sphère et énonçons les résultats de convergence que nous voulons généraliser, nous ferons aussi le lien avec les travaux concernant le *progress rate*, en particulier nous interpréterons la constante exhibée par la maximisation du *progress rate* (approche traditionnelle) en terme de dynamique de l'algorithme. Cette partie s'appuie sur des résultats de Bienvenüe et François [23]. Ensuite, aux Section 4.4 et 4.5 nous analysons respectivement la convergence du premier et du deuxième algorithme. Dans les deux cas nous donnons aussi des vitesses de convergence. A la Section 4.6 nous considérons une généralisation du deuxième algorithme à une dimension quelconque, le résultat remarquable qui sera prouvé est que pour cet algorithme la constante Σ en dessous de laquelle nous savons conclure à la convergence est indépendante de la dimension. Enfin, les constantes Σ que nous allons exhiber et en dessous desquelles on a convergence sont des conditions suffisantes pour la convergence, afin de tester la validité de ces conditions, nous présenterons à la Section 4.7 des tests numériques validant notre approche et les valeurs des constantes Σ .

4.2 Notations, définitions et rappel

Nous nous plaçons sur un espace de probabilité $(\Omega, \mathcal{F}, \mathbb{P})$ et considérons les variables aléatoires $(X_n, n \in \mathbb{N})$ définies de Ω à valeurs réelles par l'algorithme 4.1.1. Nous noterons $(\mathcal{F}_n, n \in \mathbb{N})$ la filtration naturelle adaptée au processus $(f(X_n), n \in \mathbb{N})$, où f est supposée mesurable, et nous noterons $(\mathcal{H}_n, n \in \mathbb{N})$ la filtration naturelle adaptée au processus $(X_n, n \in \mathbb{N})$. Comme f est mesurable, l'inclusion $\mathcal{F}_n \subset \mathcal{H}_n$ est vérifiée, ce qui sera utile par la suite. Certains arguments des preuves que nous allons utiliser consisteront à montrer que $(f(X_n), n \in \mathbb{N})$ est une sur-martingale: Rappelons que $f(X_n)$ est une \mathcal{F}_n -sur martingale si $f(X_n) \in L^1$ et satisfait $\mathbb{E}(f(X_{n+1})|\mathcal{F}_n) \leq f(X_n)$ presque sûrement.

Nous allons à deux reprises faire usage du Lemme suivant qui est un résultat classique mais dont nous rappelons la démonstration.

Lemme 1. *Soit \mathcal{F}_n une filtration croissante adaptée au processus X_n . Supposons que $X_n \xrightarrow[n \rightarrow \infty]{p.s.} 0$ et que X_n est borné.*

Alors

$$\mathbb{E}(X_n | \mathcal{F}_n) \xrightarrow[n \rightarrow \infty]{p.s.} 0$$

PREUVE: Soit n_0 un entier positif. On a

$$|\mathbb{E}(X_n | \mathcal{F}_n)| \leq \mathbb{E}(\sup_{p \geq n_0} |X_p| | \mathcal{F}_n) \text{ pour tout } n \geq n_0$$

Ainsi

$$\limsup |\mathbb{E}(X_n | \mathcal{F}_n)| \leq \mathbb{E}(\sup_{p \geq n_0} |X_p| | \mathcal{F}_\infty)$$

En utilisant le Théorème de convergence dominée nous obtenons

$$\mathbb{E}(\sup_{p \geq n_0} |X_p| | \mathcal{F}_\infty) \xrightarrow[n_0 \rightarrow \infty]{} 0,$$

et ainsi

$$\mathbb{E}(X_n | \mathcal{F}_n) \xrightarrow[n \rightarrow \infty]{p.s.} 0.$$

□

4.3 Convergence sur la fonction sphère

Sur la fonction sphère les deux algorithmes que nous considérons sont les mêmes à un facteur deux près. L'étude de la dynamique de l'algorithme dans ce cas là s'avère être relativement simple, comme remarqué par Bienvenue et François [23]. Cette convergence repose sur le Lemme suivant

Lemme 2. *Pour la fonction sphère, la variable aléatoire X_n définie par (4.1.1) avec $H(x) = |x|$ est telle que*

$$X_{n+1} = X_n(1 + \sigma\xi^*(\lambda)) \quad (4.3.2)$$

où l'égalité précédente à lieu en loi et où la variable aléatoire $\xi^*(\lambda)$ est définie par

$$1 + \sigma\xi^*(\lambda) = \arg \min\{(1 + \sigma\xi_1^n)^2, \dots, (1 + \sigma\xi_\lambda^n)^2\} \quad (4.3.3)$$

où ξ_i^n sont λ tirages indépendants de variables aléatoires de loi ν_ξ .

La preuve détaillée ce Lemme se trouve dans [23].

Remarque 1. *Dans (4.3.2), les variables aléatoires X_n et $(1 + \sigma\xi^*(\lambda))$ sont indépendantes.*

4.3.1 Convergence L^p

Le Théorème qui suit est une conséquence immédiate du Lemme précédent.

Théorème 1. (Convergence L^p) *Soit $p \geq 1$, un entier. Si $\mathbb{E}(|1 + \sigma\xi^*(\lambda)|^p) < +\infty$, alors*

$$\mathbb{E}(|X_n|^p) = \mathbb{E}(|X_0|^p)\mathbb{E}(|1 + \sigma\xi^*(\lambda)|^p)^n. \quad (4.3.4)$$

et ainsi l'algorithme converge ou diverge géométriquement en norme L^p . De plus, il existe une valeur $\sigma_c(\lambda, p)$ telle que X_n converge en norme L^p si et seulement si $\sigma \in]0, \sigma_c(\lambda, p)[$. Cette valeur est définie par

$$\sigma_c(\lambda, p) = \inf\{\sigma \text{ tel que } \mathbb{E}(|1 + \sigma\xi^*(\lambda)|^p) \geq 1\}. \quad (4.3.5)$$

PREUVE: Soit p un entier. De l'égalité (4.3.2) et de l'indépendance entre X_n et $1 + \sigma\xi^*(\lambda)$, on déduit que pour tout n

$$\mathbb{E}(|X_{n+1}|^p) = \mathbb{E}(|X_n|^p)\mathbb{E}(|1 + \sigma\xi^*(\lambda)|^p)^n$$

qui nous donne l'équation 4.3.4. Considérons la fonction $\sigma \rightarrow \mathbb{E}(|1 + \sigma \xi^*(\lambda)|^p)$: elle est continue (conséquence du Théorème de convergence dominée), plus petite que 1 dans un voisinage de zéro et tend vers l'infini lorsque σ tend vers l'infini. Ainsi, $\sigma_c(\lambda, p)$ défini par l'équation 4.3.5 existe et est strictement positif.

□

Remarque 2. *La fonction $\sigma \rightarrow \mathbb{E}(|1 + \sigma \xi^*(\lambda)|^p)$ étant continue, elle est bornée et atteint ses bornes, et vaut 1 pour $\sigma = 0$ et $\sigma = \sigma_c(\lambda, p)$. Le théorème des valeurs intermédiaires garantit l'existence de $\sigma_s(\lambda, p) \in]0, \sigma_c(\lambda, p)[$ tel que $\mathbb{E}(|1 + \sigma \xi^*(\lambda)|^p)$ atteigne son minimum. Ce minimum correspond à la vitesse de convergence la plus élevée, et il est défini plus précisément par*

$$\sigma_s(\lambda, p) = \operatorname{argmin}\{\mathbb{E}(|1 + \sigma \xi^*(\lambda)|^p), \sigma \in]0, \sigma_c(\lambda, p)[\}. \quad (4.3.6)$$

4.3.2 Lien avec la théorie du *progress rate*

Il est intéressant de faire le lien entre ce que nous venons de voir et les études précédentes [122, 131, 22] qui s'intéressent au *progress rate* φ_p :

$$\varphi_p(X_n, \sigma, \lambda) = \mathbb{E} \left(\frac{|X_n|^p - |X_{n+1}|^p}{|X_n|^p} \middle| X_n \right). \quad (4.3.7)$$

Notons que la définition que nous donnons ici pour φ_p est normalisée, alors que nous avons donné dans le chapitre précédent l'expression non normalisée (dans [22], Beyer appelle plus précisément la quantité φ_p , le *normalized progress rate*). Les travaux sur le *progress rate* consistent à déterminer pour un λ fixé, le pas de la mutation σ optimal qui maximise la quantité (4.3.7). Pour cela des estimations asymptotiques (lorsque la dimension de l'espace de recherche tend vers l'infini) de φ_p sont déterminées. Mais en général, ces estimations dépendent du point courant X_n et ne sont pas d'un grand intérêt d'un point de vue de la dynamique de l'algorithme.

En fait, sur la fonction sphère pour l'algorithme (4.1.1), une conséquence directe du Lemme 2 est que le *progress rate* est simplement une fonction des deux paramètres de l'algorithme à savoir λ et σ mais ne dépend plus du point courant, plus précisément

$$\forall n > 0, \quad \varphi_p(X_n, \sigma, \lambda) = \mathbb{E}(1 - |1 + \sigma \xi^*(\lambda)|^p). \quad (4.3.8)$$

Ainsi, maximiser le *progress rate* comme cela est fait dans [131, 22] revient alors à trouver σ tel que $\mathbb{E}(1 - |1 + \sigma \xi^*(\lambda)|^p)$ soit maximal, ou encore tel que $\mathbb{E}(|1 + \sigma \xi^*(\lambda)|^p)$ soit minimal –

ce qui est exactement la valeur donnée par l'équation (4.3.6), donnant la valeur de σ telle que la convergence de l'algorithme soit la plus rapide. Ceci donne un sens du point de vue de la dynamique de l'algorithme aux valeurs prédites par Schwefel et Beyer dans [131, 22] en maximisant le *progress rate*. Notons que les valeurs prédites par Schwefel et Beyer sont pour $p = 2$, correspondant donc à la valeur maximisant la convergence L^2 de l'algorithme. En particulier cette valeur garantit la convergence L^2 et presque sûre de l'algorithme, comme remarqué dans Bienvenüe et François [23].

4.3.3 Convergence presque sûre

Citons tout d'abord le Théorème suivant de Bienvenüe et François [23] qui va donner la convergence en probabilité de $\frac{1}{n} \ln(|X_n|)$.

Théorème 2. (Convergence en probabilité, [23]) *Supposons que $\mathbb{E}(\ln(|1 + \sigma\xi^*(\lambda)|)) < +\infty$. Alors, pour la fonction sphère, la variable aléatoire X_n définie par (4.1.1) avec $H(x) = |x|$ est telle que*

$$\frac{1}{n} \ln(|X_n|) \xrightarrow[n \rightarrow \infty]{} \mathbb{E}(\ln(|1 + \sigma\xi^*(\lambda)|)) \text{ en probabilité.}$$

Remarque 3. *En fait le type de convergence n'est pas précisé dans l'article [23], mais l'analyse de la démonstration nous permet de déduire que la méthode utilisée pour la preuve permet d'obtenir une convergence en probabilité.*

Pour obtenir la convergence presque sûre de $\frac{1}{n} \ln(|X_n|)$, on peut procéder de la même façon que dans le chapitre suivant reposant sur l'étude de la chaîne de Markov $(X_n, n \in \mathbb{N})$. On démontre alors de manière analogue le Théorème suivant :

Théorème 3. (Convergence presque sûre) *Supposons que $\mathbb{E}(\ln(|1 + \sigma\xi^*(\lambda)|)) < +\infty$. Alors, pour la fonction sphère, la variable aléatoire X_n définie par (4.1.1) avec $H(x) = |x|$ est telle que*

$$\frac{1}{n} \ln(|X_n|) \xrightarrow[n \rightarrow \infty]{} \mathbb{E}(\ln(|1 + \sigma\xi^*(\lambda)|)) \text{ p.s.}$$

□

La valeur critique $\sigma_c(\lambda, \text{p.s.})$ en dessous de laquelle la convergence asymptotique p.s. vers 0 a lieu pour $|X_n|$ et au dessus de laquelle il y a divergence presque sûre vers l'infini est

définie par $\sup\{\sigma \setminus \mathbb{E}(\ln(|1 + \sigma \xi^*(\lambda)|)) < 0\}$. Remarquons que d'après l'inégalité de Jensen, on a $\sigma_c(\lambda, 1) \leq \sigma_c(\lambda, \text{p.s.})$.

Nous allons maintenant voir dans les deux sections suivantes comment étendre ces résultats pour des fonctions plus générales.

4.4 Convergence du $(1, \lambda)$ -ES avec $H(x) = |x|$ _____

Considérons l'algorithme 4.1.1 avec $H(x) = |x - x^*|$ pour x^* un minimiseur de la fonction f .

$$\begin{cases} X^0 \in \mathbb{R}, \\ X_{n+1} = \arg \min\{f(X_n + \sigma |X_n - x^*| \xi_i^n), i \in [1, \lambda]\}, \end{cases} \quad (4.4.9)$$

avec $\sigma \in \mathbb{R}_*^+$, $X_0 \in \mathbb{R}$ et ξ_n^i , sont λ variables aléatoires indépendantes de loi ν_ξ .

4.4.1 Convergence

La première étape de l'analyse consiste à trouver une valeur $\sigma_c(\lambda, \beta, M)$ telle que pour $\sigma \in]0, \sigma_c(\lambda, \beta, M)[$, $f(X_n)$ est une surmartingale. C'est de cette propriété que nous allons déduire la convergence des processus $(f(X_n), n \in \mathbb{N})$ et $(X_n, n \in \mathbb{N})$.

Énonçons maintenant précisément les hypothèses que nous allons faire sur les fonctions f que nous considérons.

Hypothèses (H1)

(i) La fonction f a un unique minimum global x^* . Sans perte de généralité, on suppose que $x^* = 0$ et $f(0) = 0$, et en conséquence que $\forall x \in \mathbb{R}, f(x) > 0$.

(ii) La fonction f est de classe C^2 .

(iii) Il existe M fini tel que, pour tout $x \in \mathbb{R}$, $|f''(x)| \leq M$.

(iv) Il existe $\beta > 0$ tel que, pour tout $x \neq 0$, $\frac{|f'(x)|}{|x|} \geq \beta > 0$.

Les hypothèses clés pour nos preuves sont (i) et (iv). L'hypothèse (iv) revient à dire que f est strictement coercive dans un voisinage du minimum.

Remarque 4. Notons que toutes les preuves marchent si le processus $(X_n, n \in \mathbb{N})$ est remplacé par $\inf(\sup(X^n, -A), A)$ dans l'équation (4.1.1) pour A suffisamment grand. Une telle modification permet en particulier de rendre l'hypothèse (H1)(iii) facile à satisfaire.

Remarque 5. L'hypothèse (H1) ci-dessus implique que f décroît sur \mathbb{R}^- et croît sur \mathbb{R}^+ . En considérant l'extension de la remarque ci-dessus, cette hypothèse revient à imposer la décroissance sur $[-A, 0]$ et la croissance sur $[0, A]$.

Énonçons maintenant les hypothèses sur la variable aléatoire ξ de loi ν_ξ pour la mutation

Hypothèse (H2)

(i) ξ est symétrique.

(ii) $\mathbb{E}(|\xi|^2) < +\infty$.

Considérons ξ vérifiant les hypothèses H2 et définissons la fonction suivante

$$g(\sigma, \lambda, \beta, M) = \mathbb{E} \left(\min_{1 \leq i \leq \lambda} \left(\beta \xi^i + \sigma \frac{M}{2} |\xi^i|^2 \right) \right), \quad (4.4.10)$$

et définissons $\sigma_c(\lambda, \beta, M)$ comme la solution de

$$g(\sigma_c(\lambda, \beta, M), \lambda, \beta, M) = 0 \quad (4.4.11)$$

L'existence et l'unicité de $\sigma_c(\lambda, \beta, M)$ est démontrée au Lemme 3 ci-dessous. La convergence de $f(X_n)$ pour $\sigma < \sigma_c(\lambda, \beta, M)$ est énoncée dans le Théorème suivant

Théorème 4. Soit $\lambda \geq 2$, si f satisfait les hypothèses (H1), ξ satisfait les hypothèses (H2) et $\sigma \in]0, \sigma_c(\lambda, \beta, M)[$ avec $\sigma_c(\lambda, \beta, M)$ défini par l'équation (4.4.11), alors quand n tend vers $+\infty$, $f(X_n)$ converge vers 0, à la fois presque sûrement et dans L^1 , et X_n converge vers 0 presque sûrement et dans L^2 .

La preuve de ce Théorème repose sur les Lemmes 3 et 4. Nous allons dans un premier temps énoncer ces deux Lemmes et donner leur preuve avant de passer à la démonstration du Théorème 4.

Lemme 3. Soit β comme dans l'hypothèse (H1)(iv). Soit $\lambda \geq 2$ un entier. Soit g définie par (4.4.10). Alors $g(\sigma, \beta, \lambda, M)$ est une fonction strictement croissante et continue en σ . Elle satisfait $g(0, \beta, \lambda, M) < 0$ et $\lim_{\sigma \rightarrow +\infty} g(\sigma, \beta, \lambda, M) = +\infty$. En conséquence il existe un unique $\sigma_c(\lambda, \beta, M)$ tel que $g(\sigma_c(\lambda, \beta, M), \beta, \lambda, M) = 0$ et donc $g(\sigma, \beta, \lambda, M) < 0$ lorsque $\sigma \in [0, \sigma_c(\lambda, \beta, M)[$.

PREUVE: Soit σ_1 et σ_2 tels que $\sigma_1 < \sigma_2$, alors

$$\min_{1 \leq i \leq \lambda} \beta \xi^i + \sigma_1 \frac{M}{2} |\xi^i|^2 < \min_{1 \leq i \leq \lambda} \beta \xi^i + \sigma_2 \frac{M}{2} |\xi^i|^2 \text{ p.s.}$$

et ainsi,

$$g(\sigma_1, \beta, \lambda, M) = \mathbb{E}(\min_{1 \leq i \leq \lambda} \beta \xi^i + \sigma_1 \frac{M}{2} |\xi^i|^2) < \mathbb{E}(\min_{1 \leq i \leq \lambda} \beta \xi^i + \sigma_2 \frac{M}{2} |\xi^i|^2) = g(\sigma_2, \beta, \lambda, M).$$

La continuité de $g(\sigma, \lambda, \beta, M)$ peut être prouvée par extraction d'une suite croissante σ_n telle que $\sigma_n \rightarrow \sigma$ et en utilisant le Théorème de convergence monotone. Le fait que $g(0, \lambda, \beta, M) = \mathbb{E}(\min_{1 \leq i \leq \lambda} \xi^i) < 0$ pour $\lambda \geq 2$ est un résultat qui vient du fait que $\mathbb{E}(\xi) = 0$. Enfin, $\lim_{\sigma \rightarrow \infty} g(\sigma, \lambda, \beta, M) = +\infty$ est une conséquence du Théorème de convergence monotone. □

La propriété de sur-martingale du processus $(f(X_n), n \in \mathbb{N})$ est énoncée dans le Lemme 4.

Lemme 4. *Si f satisfait les hypothèses (H1) et ξ les hypothèses satisfait (H2), alors*

$$\mathbb{E}(f(X_{n+1}) | \mathcal{F}_n) \leq f(X_n) + \sigma |X_n|^2 g(\sigma, \lambda, \beta, M). \quad (4.4.12)$$

En conséquence $f(X_n)$ est une \mathcal{F}_n -sur-martingale pour $0 \leq \sigma \leq \sigma_c(\lambda, \beta, M)$.

PREUVE: Des Hypothèses (H1), comme $f(0) = 0$ et $f'(0) = 0$, on déduit que $\forall x, f(x) \leq \frac{M}{2}|x|^2$. Ainsi, afin de montrer que $f(X_n) \in L^1$, il suffit de montrer que $E(|X_n|^2) < \infty$. Montrons cela par récurrence. D'après la monotonie de f (cf. Remarque 5) on a l'inégalité suivante

$$|X_{n+1}| \leq |X_n| + \sigma |X_n| \max_{1 \leq i \leq \lambda} (|\xi^i|).$$

De plus $|X_n|$ et $(1 + \sigma \max_{1 \leq i \leq \lambda} (|\xi^i|))$ sont indépendants et de $E(|\xi|^2) < \infty$ on déduit que $E((\max_{1 \leq i \leq \lambda} (1 + \sigma \max_{1 \leq i \leq \lambda} (|\xi^i|)))^2) < \infty$. Ainsi, $E(|X_n|^2) < \infty$ et donc $f(X_n) \in L^1$.

Par ailleurs par construction de $(X_n, n \in \mathbb{N})$ on sait que pour tout $1 \leq i \leq \lambda$

$$f(X_{n+1}) \leq f(X_n + \sigma |X_n| \xi_i^n)$$

En utilisant l'inégalité de Taylor Lagrange ainsi que l'hypothèse (H1)-(iii), nous obtenons

$$f(X_{n+1}) \leq f(X_n) + \sigma |X_n| \xi_i^n f'(X_n) + \frac{\sigma^2}{2} |X_n|^2 M |\xi_i^n|^2 \quad \forall 1 \leq i \leq \lambda$$

ainsi en utilisant l'hypothèse (H1)(iv),

$$f(X_{n+1}) \leq f(X_n) + \sigma |X_n| |f'(X_n)| (\text{sign}(f'(X_n)) \xi_i^n + \frac{M\sigma}{2\beta} |\xi_i^n|^2) \quad \forall 1 \leq i \leq \lambda,$$

d'où

$$f(X_{n+1}) \leq f(X_n) + \sigma |X_n| |f'(X_n)| \min_{1 \leq i \leq \lambda} (\text{sign}(f'(X_n)) \xi_i^n + \frac{M\sigma}{2\beta} |\xi_i^n|^2).$$

Considérons maintenant \mathcal{H}_n et \mathcal{F}_n les filtrations naturelles adaptées au processus $(X_n, n \in \mathbb{N})$ et $(f(X_n), n \in \mathbb{N})$, on a $\mathcal{F}_n \subset \mathcal{H}_n$. On prouve tout d'abord que $f(X_n)$ est une \mathcal{H}_n -sur-martingale et on va en déduire ensuite que $f(X_n)$ est une \mathcal{F}_n -sur-martingale. Pour cela prenons l'espérance conditionnelle de l'inégalité précédente

$$\mathbb{E}(f(X_{n+1})|\mathcal{H}_n) \leq f(X_n) + \sigma |X_n| |f'(X_n)| \mathbb{E} \left(\min_{1 \leq i \leq \lambda} (\text{sign}(f'(X_n)) \xi_i^n + \frac{M\sigma}{2\beta} |\xi_i^n|^2) | \mathcal{H}_n \right)$$

Grâce à la symétrie de ξ_i^n on a

$$\mathbb{E} \left(\min_{1 \leq i \leq \lambda} (-\xi_i^n + \frac{M\sigma}{2\beta} (\xi_i^n)^2) \right) = \mathbb{E} \left(\min_{1 \leq i \leq \lambda} (+\xi_i^n + \frac{M\sigma}{2\beta} |\xi_i^n|^2) \right)$$

Ainsi,

$$\mathbb{E}(f(X_{n+1})|\mathcal{H}_n) \leq f(X_n) + \sigma |X_n| \frac{|f'(X_n)|}{\beta} g(\sigma, \lambda, \beta, M).$$

Comme pour $0 < \sigma \leq \sigma_c(\lambda, \beta, M)$, $g(\sigma, \lambda, \beta, M) \leq 0$ nous obtenons

$$\mathbb{E}(f(X_{n+1})|\mathcal{H}_n) \leq f(X_n) + \sigma |X_n|^2 g(\sigma, \lambda, \beta, M),$$

i.e. (4.4.12). La propriété $\mathbb{E}(\mathbb{E}(\cdot|\mathcal{H}_n)|\mathcal{F}_n) = \mathbb{E}(\cdot|\mathcal{F}_n)$ (voir [152]), conduit au fait que pour $0 < \sigma \leq \sigma_c(\lambda, \beta, M)$, $f(X_n)$ est une \mathcal{F}_n -sur-martingale.

□

On peut maintenant passer à la preuve du Théorème 4.

PREUVE: [du Théorème 4] On part de l'inégalité (4.4.12) du Lemme 4, et on prend son espérance, ce que nous pouvons faire puisque l'on a vu dans la preuve du Lemme 4 que $f(X_n)$ et $|X_n|^2$ sont tous les deux dans L^1 . Nous obtenons

$$\mathbb{E}(f(X_{n+1})) \leq \mathbb{E}(f(X_n)) + \sigma \mathbb{E}(|X_n|^2) g(\sigma, \lambda, \beta, M) \quad (4.4.13)$$

On tire du Lemme 3 que pour $\sigma \in]0, \sigma_c(\lambda, \beta, M)[$, $g(\sigma, \lambda, \beta, M) < 0$, ainsi pour $\sigma \in]0, \sigma_c(\lambda, \beta, M)[$, $\mathbb{E}(f(X_n))$ est une suite décroissante positive qui converge donc. Réécrivons (4.4.13) comme

$$\mathbb{E}(|X_n|^2)(-\sigma g(\sigma, \lambda, \beta, M)) \leq \mathbb{E}(f(X_n)) - \mathbb{E}(f(X_{n+1})). \quad (4.4.14)$$

Pour tout $\sigma \in]0, \sigma_c(\lambda, \beta, M)[$, le membre de droite de (4.4.14) converge vers zéro, et le membre de gauche est positif. Ceci implique que $\mathbb{E}(|X_n|^2)$ converge vers 0. On peut en conséquence extraire une suite $|X^{\gamma(n)}|^2$ de $|X_n|^2$ qui converge presque sûrement vers zéro. Comme f est continue $f(X^{\gamma(n)}) \xrightarrow[n \rightarrow \infty]{\text{p.s.}} f(0) = 0$. D'après le Lemme 4, on sait que pour $\sigma \in]0, \sigma_c(\lambda, \beta, M)[$, le processus $(f(X_n), n \in \mathbb{N})$ est une sur-martingale positive. Ceci implique que $f(X_n)$ converge presque sûrement (voir [152]). De l'unicité de la limite pour $(f(X^n), n \in \mathbb{N})$, on en déduit que $(f(X_n), n \in \mathbb{N})$ converge presque sûrement vers 0. Pour la convergence L^1 de $f(X_n)$, on déduit de l'hypothèse (H1)(iii) que $\forall x, f(x) \leq \frac{M}{2}|x|^2$ et on utilise ensuite la convergence de $\mathbb{E}(|X_n|^2)$. Il s'en suit que $\mathbb{E}(f(X_n))$ converge vers zéro. De plus d'après l'inégalité 4.4.12 on a

$$|X_n|^2 \leq \frac{\mathbb{E}(f(X_n) - f(X_{n+1})|\mathcal{F}_n)}{-\sigma g(\sigma, \lambda, \beta, M)}$$

On déduit de cette inégalité et du Lemme 1 que X_n converge presque sûrement vers 0.

□

4.4.2 Vitesse de convergence

Théorème 5. *Supposons que f satisfait les hypothèses (H1) et que $\sigma \in]0, \sigma_c(\lambda, \beta, M)[$, avec $\sigma_c(\lambda, \beta, M)$ défini par (4.4.11). Alors $f(X_n)$ converge géométriquement vers 0 au sens suivant:*

(i) **(Convergence p.s.) :** $\frac{f(X_n)}{(1 + \sigma Cg(\sigma, \lambda, \beta, M))^n}$ converge vers une variable aléatoire Y ,

(ii) **(Convergence L^1) :** $\mathbb{E}(f(X_n)) \leq (1 + \sigma Cg(\sigma, \lambda, \beta, M))^n \mathbb{E}(f(X^0))$,

où $C = \frac{2}{M}$ et M sont définis par (H1)(iii). De plus, le meilleur taux de convergence est atteint pour $\sigma = \sigma_s(\lambda, \beta, M)$ où $\sigma_s(\lambda, \beta, M)$ est l'unique valeur de σ qui minimise $1 + \sigma Cg(\sigma, \lambda, \beta, M)$.

PREUVE: Soit $\sigma < \sigma_c(\lambda, \beta, M)$, en utilisant l'inégalité (4.4.12)

$$\mathbb{E}(f(X_{n+1})|\mathcal{H}_n) \leq f(X_n) + \sigma |X_n|^2 g(\sigma, \lambda, \beta, M)$$

Comme cela est mentionné plus haut, l'hypothèse (H1) implique en particulier que $f(x) \leq \frac{M}{2}|x|^2$ pour tout x , et en conséquence on obtient

$$\mathbb{E}(f(X_{n+1})|\mathcal{H}_n) \leq f(X_n)(1 + \sigma \frac{2}{M}g(\sigma, \lambda, \beta, M))$$

Pour σ tel que $0 < 1 + \sigma \frac{2}{M}g(\sigma, \lambda, \beta, M) < 1$, $f(X_n)/(1 + \sigma \frac{2}{M}g(\sigma, \lambda, \beta, M))^n$ est une sur-martingale positive qui converge presque sûrement. En prenant l'espérance on en déduit la convergence géométrique de $\mathbb{E}(f(X_n))$ vers zéro au taux annoncé dans le Théorème.

□

4.5 Convergence du $(1, \lambda)$ -ES avec $H(x) = |f'(x)|$

Nous nous intéressons maintenant à l'algorithme

$$\begin{cases} X^0 \in \mathbb{R}, \\ X_{n+1} = \arg \min \{f(X_n + \sigma |f'(X_n)| \xi_i^n), i \in [1, \lambda]\}. \end{cases} \quad (4.5.15)$$

Le schéma général de la démonstration dans ce cas est le même que celui de la section précédente : commencer par trouver une valeur Σ telle que $f(X_n)$ soit une sur-martingale pour $\sigma \in]0, \Sigma[$; en déduire ensuite la convergence et le taux de convergence de $f(X_n)$.

Moins d'hypothèses sur les fonctions f considérées dans cette section vont être requises, plus précisément nous allons considérer les fonctions f vérifiant

Hypothèses (H3)

(i) La fonction f est bornée inférieurement (disons par zéro),

(ii) f est de classe C^2 ,

(iii) Il existe M fini tel que, pour tout $x \in \mathbb{R}$, $|f''(x)| \leq M$.

Remarque 6. Pour la convergence de l'algorithme nous n'avons pas besoin d'imposer l'existence d'un unique minimum, en contrepartie, on va montrer la convergence vers un minimum local. En revanche, pour obtenir des vitesses de convergence, nous supposons l'existence d'un unique minimum.

Remarque 7. Encore une fois, utiliser l'astuce mentionnée à la Remarque 4 permet d'affaiblir ces hypothèses qui seront alors satisfaites pour toute fonction C^2 .

4.5.1 Convergence de $f'(X_n)$

Afin d'alléger les notations nous allons noter

$$h(\sigma, \lambda) = g(1, \sigma, \lambda) = \mathbb{E} \left(\min_{1 \leq i \leq \lambda} \left(\xi^i + \sigma \frac{M}{2} |\xi^i|^2 \right) \right) \quad (4.5.16)$$

et définir $\sigma'_c(\lambda)$ comme la solution de

$$g(1, \sigma'_c(\lambda), \lambda) = 0. \quad (4.5.17)$$

La preuve de l'existence et l'unicité de cette constante est exactement la même que la preuve du Lemme 3 de la section précédente. La convergence du processus $(f'(X_n), n \in \mathbb{N})$ vers 0 est énoncée dans le Théorème suivant.

Théorème 6. *Supposons que f satisfait les Hypothèses (H3) et ξ satisfait (H2). Supposons que $\lambda \geq 2$ et $\sigma \in]0, \sigma'_c(\lambda)[$. Alors $\mathbb{E}(|f'(X_n)|^2)$ converge vers 0. Si de plus on suppose que $f(X_n)$ est bornée alors $f'(X_n)$ converge presque sûrement vers 0.*

La preuve de ce Théorème repose sur les deux Lemmes suivant. Dans la suite \mathcal{F}_n désigne comme à la section précédente une filtration adaptée au processus $(f(X_n), n \in \mathbb{N})$.

Lemme 5. *Soit $\sigma'_c(\lambda)$ défini par (4.5.17). Alors $(f(X_n), n \in \mathbb{N})$ est une \mathcal{F}_n -sur-martingale pour $0 \leq \sigma \leq \sigma'_c(\lambda)$.*

PREUVE: La preuve suit les mêmes étapes que la preuve du Lemme 4. On commence par remarquer que

$$f(X_{n+1}) \leq f(X_n + \sigma|f'(X_n)|\xi_i^n) \quad \forall 1 \leq i \leq \lambda$$

En utilisant successivement l'inégalité de Taylor Lagrange et la propriété (H3)(iii), et en prenant le minimum en i du terme de droite nous obtenons

$$f(X_{n+1}) \leq f(X_n) + \sigma|f'(X_n)|^2 \min_{1 \leq i \leq \lambda} (\text{sign}(f'(X_n))\xi_i^n + \frac{M\sigma}{2}|\xi_i^n|^2) \quad (4.5.18)$$

Soit \mathcal{H}_n la filtration adaptée au processus X_n . En prenant l'espérance conditionnelle de l'inégalité précédente, nous obtenons

$$\mathbb{E}(f(X_{n+1})|\mathcal{H}_n) \leq f(X_n) + \sigma|f'(X_n)|^2 \mathbb{E} \left(\min_{1 \leq i \leq \lambda} (\text{sign}(f'(X_n))\xi_i^n + \frac{M\sigma}{2}|\xi_i^n|^2) | \mathcal{H}_n \right)$$

et donc

$$\mathbb{E}(f(X_{n+1})|\mathcal{H}_n) \leq f(X_n) + \sigma|f'(X_n)|^2 h(\sigma, \lambda) \quad (4.5.19)$$

Pour $0 < \sigma \leq \sigma'_c(\beta, \lambda)$ ceci implique (en prenant l'espérance des deux cotés) que par récurrence $f(X_n) \in L^1$, et que de plus $f(X_n)$ est une \mathcal{F}_n -sur-martingale.

□

PREUVE: [du Théorème 6] Pour le premier point du Théorème on prend l'espérance conditionnelle de l'équation (4.5.19),

$$\mathbb{E}(f(X_{n+1})) \leq \mathbb{E}(f(X_n)) + \sigma \mathbb{E}(|f'(X_n)|^2)h(\sigma, \lambda).$$

Ainsi pour $0 \leq \sigma \leq \sigma'_c(\beta, \lambda)$, $\mathbb{E}(f(X_n))$ est une suite positive décroissante qui en conséquence converge. De plus la majoration

$$-\sigma \mathbb{E}(|f'(X_n)|^2)h(\sigma, \lambda) \leq \mathbb{E}(f(X_n)) - \mathbb{E}(f(X_{n+1}))$$

implique que pour $0 < \sigma < \sigma'_c(\beta, \lambda)$,

$$\mathbb{E}(|f'(X_n)|^2) \rightarrow 0. \quad (4.5.20)$$

Afin d'avoir la convergence presque sûre on revient à l'inégalité (4.5.19) et au Lemme 1. D'après (4.5.19) nous savons que

$$|f'(X_n)|^2 \leq \frac{\mathbb{E}(f(X_n) - f(X_{n+1})|\mathcal{F}_n)}{-\sigma h(\sigma, \lambda)}$$

Grâce au Lemme 5, $f(X_n)$ est une sur-martingale positive. Elle converge ainsi presque sûrement. Ainsi $f(X_n) - f(X_{n+1})$ converge presque sûrement vers 0. Comme $f(X_n)$ est bornée par hypothèse, on déduit du Lemme 1 que $\mathbb{E}(f(X_n) - f(X_{n+1})|\mathcal{F}_n)$ converge presque sûrement vers 0. Ainsi $|f'(X_n)|^2$ converge presque sûrement vers 0.

□

4.5.2 Vitesse de convergence

Une hypothèse supplémentaire est faite pour estimer la vitesse de convergence, hypothèse reliée à la convexité. Avant d'énoncer l'hypothèse supplémentaire, nous allons tout d'abord supposer sans perte de généralité que

$$\inf_{\mathbb{R}} f = 0,$$

sinon $f(x)$ doit être remplacée par $f(x) - \inf_{\mathbb{R}} f$ dans l'hypothèse ci-dessus.

Hypothèse (H4)

- Il existe $C > 0$ tel que

$$\inf_{\mathbb{R}} \frac{|f'(x)|^2}{f(x)} \geq C.$$

Remarque 8. Notons que l'hypothèse (H4) implique que l'ensemble des minima globaux x^* de f est un intervalle, et que tout point critique est un minimiseur.

Théorème 7. Supposons que f satisfait les Hypothèses (H3)(H4), ξ satisfait les hypothèses (H2) et que $\sigma \in]0, \sigma'_c(\lambda)[$. Alors $f(X_n)$ converge géométriquement vers 0 à la vitesse $(1 + \sigma Ch(\sigma, \lambda))$ presque sûrement et dans L^1 (au sens du Théorème 5, et avec la constante C définie dans (H3)). Le meilleur taux de convergence est atteint pour $\sigma = \sigma'_s(\lambda)$ où $\sigma'_s(\lambda)$ minimise $1 + \sigma Ch(\sigma, \lambda)$.

PREUVE: Soit $\sigma \in]0, \sigma'_c(\lambda)[$, en utilisant (4.5.19) et l'hypothèse (H4),

$$\mathbb{E}(f(X_{n+1})|\mathcal{F}_n) \leq f(X_n) (1 + \sigma Ch(\sigma, \lambda)) \quad (4.5.21)$$

Pour σ tel que $0 < 1 + \sigma Ch(\sigma, \lambda) < 1$, $f(X_n)/(1 + \sigma Ch(\sigma, \lambda))^n$ est une sur-martingale positive qui converge donc presque sûrement. En prenant l'espérance de l'équation (4.5.21), nous obtenons que $\mathbb{E}(f(X_n))$ converge géométriquement vers zéro au taux $(1 + \sigma Ch(\sigma, \lambda))^n$.

4.5.3 Optimalité des constantes sur la fonction sphère

Comparons, dans le cas de la fonction sphère, d'une part les résultats obtenus dans les sections précédentes (avec $H(x) = |f'(x)|$ et $H(x) = |x|$) et d'autre part aux résultats de la Section 4.3 consacrée entièrement à la fonction sphère.

Rappelons tout d'abord que les deux algorithmes diffèrent seulement d'un facteur 2. Sur la fonction sphère les constantes β , M et C des Hypothèses (H1) et (H4) sont respectivement égales à 2, 2 et 4. Le Théorème 7 nous donne alors comme taux de convergence $1 + 4\sigma g(1, \sigma, \lambda)$. Par ailleurs le Théorème 5 nous donne $1 + \sigma \frac{2}{M} g(\beta, \sigma, \lambda)$ comme taux de convergence. Comme par définition de g , on a $1 + 2\sigma g(2, 2\sigma, \lambda) = 1 + 4\sigma g(1, \sigma, \lambda)$, on en déduit que les valeurs critiques (en dessous de laquelle il y a convergence L^1 de $f(X_n)$ et pour laquelle le taux de convergence est maximal) exhibées respectivement lors de l'analyse des algorithmes avec $H(x) = |x|$ et $H(x) = |f'(x)|$ dans le cadre de la fonction sphère correspondent.

Par ailleurs la valeur critique en dessous de laquelle on a prouvé qu'il y a convergence L^1 de $f(X_n)$ (qui est sur la sphère, équivalente à la convergence L^2 de X_n) dans les sections 4.4 et 4.5 est définie implicitement à l'équation 4.4.11 (avec $\beta = 2$ et $M = 2$) comme zéro de la fonction

$$\sigma \rightarrow \mathbb{E}(\min_{1 \leq i \leq \lambda} (2\xi^i + \sigma |\xi^i|^2)).$$

Or, dans le Théorème 1, la convergence L^2 a lieu pour tout σ tels que

$$\mathbb{E}(\min_{1 \leq i \leq \lambda} (1 + \sigma \xi^i)^2) < 1$$

c'est à dire pour tout σ tels que

$$1 + \sigma \mathbb{E}(\min_{1 \leq i \leq \lambda} (2\xi^i + \sigma |\xi^i|^2)) < 1$$

ou encore pour tout σ tel que $\mathbb{E}(\min_{1 \leq i \leq \lambda} (2\xi^i + \sigma |\xi^i|^2)) < 0$, ce qui correspond exactement à l'ensemble des σ pour lesquels on a prouvé la convergence vers 0 dans les Théorèmes 5 et 7. Comme dans le Théorème 1, la valeur critique nous donne une condition nécessaire et suffisante pour la convergence L^2 de X_n , on en déduit que nos estimations de valeurs critiques dans le cadre de la sphère sont optimales.

4.6 En dimension quelconque

L'algorithme défini par (4.1.1) est légèrement modifié pour la dimension $d > 1$. La forme générale de l'algorithme considéré ici est:

$$\begin{cases} X^0 \in \mathbb{R}^d, \\ X_{n+1} = \arg \min \{f(X_n + \sigma(H_k(X_n)\xi_k^{n,i}))_{k \in [1,d]}, i \in [1, \lambda]\}, \end{cases} \quad (4.6.22)$$

où X_n est la variables aléatoire modélisant le parent à la génération n , et $(\xi_k^{n,i})$, $k \in [1, d]$, $i \in [1, \lambda]$ sont des variables aléatoires indépendantes symétriques, de même loi que ξ et $H_k(x)$, $k \in [1, d]$ sont d fonctions réelles. Des pas différents dans toutes les directions sont appliqués ici, ce qui est classiquement fait par exemple dans le cas des algorithmes auto-adaptatifs [131].

Nous allons simplement traiter le cas où $H_k(x) = \frac{\partial f(x)}{\partial x_k}$. La situation est alors identique à celle étudiée dans la section 4.5, un résultat similaire au Lemme 5 peut être obtenu. Donnons rapidement la généralisation à faire. L'hypothèse (H3)(iii) devient alors

Il existe M fini tel que $D^2 f$, la matrice Hessienne de f , satisfasse

$$D^2 f \leq M \mathbf{Id}, \quad (4.6.23)$$

au sens des matrices symétriques.

et l'hypothèse (H4) est remplacée par

Il existe $C > 0$ tel que

$$\|\nabla f\|^2 \geq C f.$$

On écrit ensuite que pour tout $1 \leq i \leq \lambda$,

$$\begin{aligned}
f(X_{n+1}) &\leq f(X^n + \sigma(\frac{\partial f(X_n)}{\partial x_k} \xi_k^{n,i})) \\
&= f(X^n) + \sigma \sum_{k=1}^d (\frac{\partial f(X_n)}{\partial x_k})^2 \xi_k^{n,i} \\
&\quad + \frac{1}{2} D^2 f(X^n) \cdot (\sigma(\frac{\partial f(X_n)}{\partial x_k} \mathcal{N}_k^{n,i})_k, \sigma(\frac{\partial f(X_n)}{\partial x_k} \xi_k^{n,i})_k) \\
&\leq f(X^n) + \sigma \sum_{k=1}^d (\frac{\partial f(X_n)}{\partial x_k})^2 [\xi_k^{n,i} + \frac{M}{2} \sigma(\xi_k^{n,i})^2]
\end{aligned}$$

Il est alors immédiat de voir que tous les arguments de transposent au cas de la dimension d et conduisent à la même conclusion et au même critère de convergence. Notons que cette propriété n'est pas très étonnante au vu la majoration de la matrice hessienne de l'équation 4.6.23 considérée. En effet, la matrice majorant la matrice hessienne de f induit une fonction séparable.

En particulier, la valeur critique σ_c en dessous de laquelle la convergence a lieu est encore définie par les équations (4.5.16) et (4.5.17). Le fait remarquable ici est que la valeur critique (et en conséquence le taux de convergence qui va avec) ne dépend pas de la dimension. Ce résultat n'est pas étonnant et provient de la séparabilité implicite qui est considéré sur la fonction à travers la majoration de la matrice hessienne de la fonction par une homothétie (équation 4.6.23).

4.7 Expérimentations numériques

Considérons les constantes implicitement définies à travers les équations (4.4.11) et (4.5.17), en dessous desquelles nous avons démontré la convergence presque sûre et dans L^1 de $(f(X_n), n \in \mathbb{N})$ des deux algorithmes que nous avons analysés (respectivement aux Théorèmes 4 et 7). Nous avons vu à la Section 4.5.3 que ces constantes sont optimales sur la fonction sphère dans le sens où elles donnent dans ce cas la une condition nécessaire et suffisante de convergence L^1 de $(f(X_n), n \in \mathbb{N})$.

Le but de cette section est de tester numériquement sur d'autres fonctions que la fonction sphère cette optimalité. Ce que nous appelons optimalité ici est le fait que les constantes que nous avons exhibées et nous donnant des conditions suffisantes pour la convergence se rapprochent d'une valeur pour σ en dessous de laquelle il y aurait convergence et au dessus

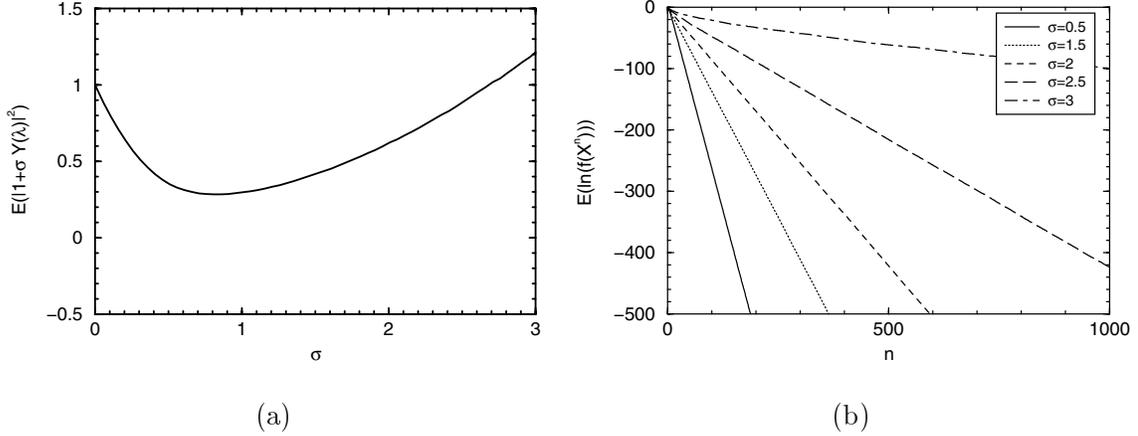


FIG. 4.1: (a) $\mathbb{E}(|1 + \sigma Y(4)|^2)$ en fonction de σ – voir équation (4.3.3). (b) $\mathbb{E}(\ln(f(X_n)))$ en fonction du nombre de génération n pour différentes valeurs de σ .

de laquelle il y aurait divergence. Notons que nous ne savons pas si une telle valeur existe, mais les tests numériques effectués nous conduisant à émettre cette conjecture.

Tous les tests numériques présentés dans la suite reposent sur la méthode de Monte Carlo pour approcher l'espérance d'une variable aléatoire. Plus précisément étant donné une variable aléatoire de moyenne finie, $\mathbb{E}(Z)$ s'approche par

$$\mathbb{E}(Z) \approx \frac{1}{K} \sum_{k=1}^K Z_k,$$

où $(Z_k)_{1 \leq k \leq K}$ sont K tirages indépendants de variables aléatoire de même loi que Z . Si de plus Z a un moment d'ordre 2, une conséquence du Théorème central limite est que pour des grandes valeurs de K ($K = 1500$ dans toutes les expériences numériques présentées ci-dessous), avec une probabilité de 0.95

$$\mathbb{E}(Z) \in \left[\frac{1}{K} \sum_{k=1}^K Z_k - \sqrt{\text{Var} Z} \frac{1.96}{\sqrt{K}}, \frac{1}{K} \sum_{k=1}^K Z_k + \sqrt{\text{Var} Z} \frac{1.96}{\sqrt{K}} \right] \quad (4.7.24)$$

4.7.1 Approximation des constantes

La méthode de Monte-Carlo décrite ci-dessus a été utilisée pour calculer des approximations des constantes $\sigma_c(\lambda, \beta, M)$ et $\sigma_s(\beta, \lambda)$.

Un premier exemple est donné par le tracé de $\mathbb{E}(|1 + \sigma \xi^*(\lambda)|^2)$ en fonction de σ pour la fonction sphère sur la figure 4.1 (a), pour $\lambda = 4$. Une valeur approchée de la valeur limite de σ pour laquelle $\mathbb{E}(|1 + \sigma \xi^*(\lambda)|^2) \leq 1$ est $\sigma_c(\lambda, 2) = 2.7$, et la valeur correspondante pour

le min de $\mathbb{E}(|1 + \sigma\xi^*(\lambda)|^2)$ est $\sigma_s(\beta, \lambda, 2) \approx 0.8$. Notons que cette méthode permet aussi de tracer le *progress rate* dont la formule est rappelée à l'équation (4.3.8) pour n'importe quelle dimension d . C'est une différence essentielle par rapport à ce qui est fait classiquement dans ce domaine [131, 22], où seules des estimations asymptotiques en la dimension sont données.

4.7.2 Optimalité des constantes

L'idée ici est de comparer les constantes $\sigma_c(\lambda, \beta, M)$, $\sigma_s(\beta, \lambda)$, $\sigma'_c(\lambda)$ et $\sigma'_s(\lambda)$ définies aux Théorèmes 5 et 7 pour des fonctions non quadratique, afin de tester l'optimalité des constantes.

Retour sur l'analyse théorique Commençons par soulever le problème suivant. Lorsque l'on évalue $\mathbb{E}(f(X_n))$ par méthode de Monte Carlo, l'erreur relative donnée par le Théorème Central Limite ($\sqrt{\text{Var}(f(X_n))} \frac{1.96}{\sqrt{K\mathbb{E}(f(X_n))}}$) croît géométriquement avec le nombre de générations n (le calcul exact peut être fait plus facilement sur la fonction sphère pour se rendre compte du problème). D'autre part, si on considère l'erreur relative que l'on fait si on évalue $\mathbb{E}(\ln(f(X_n)))$, alors cette erreur décroît en $\frac{1}{\sqrt{n}}$. C'est pourquoi tous les tests numériques ont été faits en fait sur le processus $(\ln(f(X_n)), n \in \mathbb{N})$. Nous devons en conséquence revenir sur l'analyse théorique. Il s'avère que les arguments utilisés pour la minimisation de f sont encore vrais pour la minimisation de $\ln(f)$. Bien sûr, comme la convergence presque sûre de $(f(X_n), n \in \mathbb{N})$ implique celle de $\ln(f(X_n))$, nous avons des conditions suffisantes pour une telle convergence. Mais, nous avons plus que cela: $\ln(f(X_n))$ converge de la même façon et sous les mêmes conditions que $f(X_n)$ avec un taux de convergence arithmétique qui remplace le taux de convergence géométrique des Théorèmes 5 et 7. Indiquons rapidement les modifications des arguments nécessaires pour établir cela. Nous expliquons cela dans le cas $H(x) = |f'(x)|$. Nous commençons par (4.5.18), à savoir

$$f(X_{n+1}) \leq f(X_n) + \sigma |f'(X_n)|^2 \min_{1 \leq i \leq \lambda} (\text{sign}(f'(X_n)) \xi_i^n + \frac{M\sigma}{2} |\xi_i^n|^2),$$

on prend ensuite le logarithme des deux membres de cette inégalité et on utilise le fait que $\ln(1+u) \leq u$ pour obtenir

$$\ln(f(X_{n+1})) \leq \ln(f(X_n)) + \sigma \frac{|f'(X_n)|^2}{f(X_n)} \min_{1 \leq i \leq \lambda} (\text{sign}(f'(X_n)) \xi_i^n + \frac{M\sigma}{2} |\xi_i^n|^2).$$

Ensuite on prend l'espérance conditionnelle et on utilise l'hypothèse (H4) pour écrire lorsque $h(\sigma, \lambda) < 0$

$$\mathbb{E}(\ln(f(X_{n+1})) | \mathcal{H}_n) \leq \ln(f(X_n)) + \sigma Ch(\sigma, \lambda).$$

Cela montre que $\ln(f(X_n))$ (et en fait $-\sigma Ch(\sigma, \lambda) + \ln(f(X_n))$) est une sur-martingale. Une conséquence est aussi que la borne inférieure

$$\mathbb{E}(\ln(f(X^n))) \leq \mathbb{E}(\ln(f(X^0))) + n\sigma Ch(\sigma, \lambda) \quad (4.7.25)$$

qui conduit à la convergence de $\mathbb{E}(\ln(f(X^n)))$ vers $-\infty$. Tout les résultats pour la convergence de $\ln(f)$ s'en suivent, avec les mêmes valeurs critiques pour les taux de convergence. Ces taux sont bien sûr arithmétiques au lieu de géométriques. Quoi qu'il en soit, cette équivalence nous permet de tester l'optimalité de nos constantes de notre analyse en étudiant $\ln(f(X_n))$.

Résultats numériques Seuls les résultats concernant le cas $H(x) = |f'(x)|$ sont montrés ici. Les résultats dans le cas $H(x) = |x|$ sont très semblables. Les fonctions f_M , définies par les équations (4.7.26) ci-dessous sont des exemples parmi la classe de fonction non symétriques satisfaisant les hypothèses (H3) et (H4) qui seront utilisées dans les tests numériques (où $M > 0$ désigne la constante des hypothèses (H3)-(ii)).

$$f_M(x) = \frac{M}{2} \begin{cases} x^2 & \text{si } x < 0 \\ x \arctan(x) & \text{si } x > 0 \end{cases} \quad (4.7.26)$$

La figure 4.1 (b) montre le tracé de $\frac{1}{K} \sum_{k=1}^K \ln(f_2(X_n^k))$ en fonction du nombre de générations n pour différentes valeurs de σ . L'erreur relative

$$\frac{\mathbb{E}(\ln(f_2(X_n))) - \frac{1}{K} \sum_{k=1}^K \ln(f_2(X_n^k))}{\frac{1}{K} \sum_{k=1}^K \ln(f_2(X_n^k))}$$

donnée par l'équation (4.7.24), est ici bornée par 0.01. Cette figure confirme le taux de convergence linéaire prédit par l'étude théorique.

La figure 4.2 (a) montre d'une part pour différentes valeurs de σ (en abscisse) la pente correspondante de la courbe obtenue par un tracé du type de celui de la figure 4.1 (b) (cette pente est obtenue en faisant une régression linéaire sur les données du type de celles de la Figure 4.1 (b)), ou de manière équivalente, la figure 4.2 (a) montre la vitesse de convergence de l'algorithme observée numériquement pour un sigma donné en fonction de sigma. Sur la même figure est également tracée la valeur $\sigma Ch(\sigma, \lambda)$, majoration théorique pour la vitesse de convergence (cf équation 4.7.25). Les tracés sont réalisés pour $\lambda = 4$ et pour les fonctions f_2 et f_8 . On remarque d'une part que les courbes correspondant aux valeurs observées

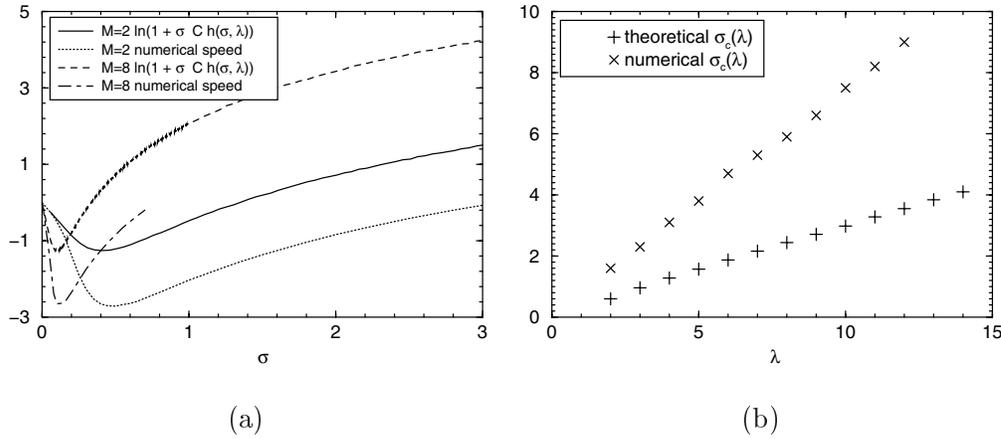


FIG. 4.2: (a) Taux de convergence théoriques et numériques pour les fonctions f_2 et f_8 . (b) Valeurs numériques pour $\sigma'_{cnum}(\lambda)$ et théoriques $\sigma'_c(\lambda)$.

numériquement et les courbes correspondant aux majorations théoriques ont les mêmes allures et que les bornes théoriques sous estiment les valeurs observées numériquement comme on pouvait s'y attendre.

On peut aussi remarquer sur la figure 4.2 (a) que la courbe théorique et la courbe numérique présentent la même transformation d'échelle lorsque l'on augmente M – même si les bornes théoriques sont pessimistes.

Intéressons nous plus précisément à la fonction f_2 , l'intersection entre la courbe théorique et la courbe numérique sur l'axe des x donne une approximation numérique $\sigma'_c(4) \approx 1.4$ de la valeur théorique $\sigma_c(4)$ – et dans la suite, $\sigma'_{cnum}(4)$ désignera l'intersection entre la courbe expérimentale et l'axe des x . A partir de la figure 4.2 (a), il vient que $\sigma'_{cnum}(4) \approx 3.1$. En définissant de la même façon $\sigma'_{snum}(4)$ comme la valeur critique de la courbe numérique, on peut aussi remarquer sur la même figure que $\sigma'_s(4) \leq \sigma'_{snum}(4)$. Ces deux valeurs pour $\lambda = 4$ sont tracés pour différents λ à la figure 4.2 (b) (toujours pour la fonction f_2). On note que les deux fonctions sont croissantes en λ .

4.8 Discussion

L'analyse de la convergence des deux algorithmes adaptatifs à une sous classe de fonction C^2 que nous venons de présenter généralise les études traditionnellement menées sur la fonction sphère. En particulier les estimations théoriques des bornes pour le pas de la mutation

en dessous desquelles on a convergence des algorithmes adaptatifs s'avèrent être optimales sur la fonction sphère, c'est-à-dire, dans ce cas là, les valeurs trouvées donnent des conditions nécessaires et suffisantes pour la convergence de l'algorithme.

De plus, nous avons montré que les constantes que l'on cherche à approcher dans l'approche traditionnelle du *progress rate* ont un sens pour la dynamique de l'algorithme et correspondent précisément à la valeur du paramètre σ maximisant la vitesse de convergence L^2 de l'algorithme. En particulier, comme remarqué dans [23], cette constante garantit la convergence de l'algorithme.

Par ailleurs, la généralisation de notre analyse à un algorithme non isotrope en dimension quelconque, révèle que le taux de convergence est indépendant de la dimension. Ce résultat, conséquence immédiate de l'analyse menée en dimension 1, et qui pourrait paraître contre intuitif au premier abord, provient simplement de la séparabilité implicite que nous considérons par la majoration donnée à l'équation 4.6.23.

Chapitre 5

Convergence d'Algorithmes Evolutionnaires auto-adaptatifs

Une partie des résultats de ce chapitre fait l'objet de l'article à paraître dans *Theoretical Computer Science* [12].

[12] Anne Auger, *Convergence results for $(1,\lambda)$ -SA-ES using the theory of φ -irreducible markov chains*, *Theoretical Computer Science*, à paraître.

5.1 Introduction

Depuis l'invention des Stratégies d'Evolution, dans le milieu des années 60 [122, 131], les études théoriques se sont concentrées sur l'étude du *progress rate* (cf chapitre 3 ou [22, 131]), mesurant le progrès d'une itération à l'autre de l'algorithme. Ces études ont en particulier apporté beaucoup d'intuitions sur la façon de régler les écart-types de la mutation (voir chapitre 1) mais elles ne prennent pas en compte la dynamique globale de l'algorithme et en particulier ne s'intéressent pas à sa convergence.

Comme nous l'avons vu dans le détail au chapitre 1, tout l'enjeu des stratégies d'évolution au cours des années a été de créer des algorithmes s'adaptant aux différentes phases de recherche, à savoir la phase d'exploration et la phase d'exploitation. A l'heure actuelle, les algorithmes recommandés pour l'optimisation numérique, les algorithmes SA-ES, CSA, DE (cf chapitre 1) se trouvent être des algorithmes particulièrement performant pour la phase d'exploitation, c'est cette phase qui va nous intéresser ici. Il est en particulier connu des utilisateurs d'algorithmes adaptatifs en général que sur la plus simple des fonctions unimodales, la fonction sphère, la convergence est log-linéaire et que le pas de la mutation converge également de manière log-linéaire. Pour étudier ces propriétés d'adaptation dans le cas de la fonction sphère et plus particulièrement la convergence log-linéaire, nous allons traiter dans le détail le cas d'un algorithme auto-adaptatif, celui introduit par Rechenberg et Schwefel [122, 131] dont le principe de base que nous avons détaillé au chapitre 1 repose sur la mutation du paramètre de mutation lui même. Nous noterons cet algorithme le $(1, \lambda)$ -SA-ES où SA est l'acronyme de *self-adaptation*, en accord avec les notations habituelles (cf. Beyer [22]). Nous verrons à la fin de ce chapitre comment l'analyse faite sur le $(1, \lambda)$ -SA-ES se généralise à une classe plus large d'algorithmes adaptatifs dont l'algorithme CSA.

Bien que la fonction sphère soit une des fonctions les plus simples que l'on peut considérer, l'étude théorique de la convergence d'algorithmes auto-adaptatifs sur cette fonction s'avère être complexe. Comme nous l'avons vu au chapitre 3, les premières études de convergence d'algorithmes auto-adaptatifs sur cette fonction (qui est à l'heure actuelle la seule qui a été étudiée théoriquement) datent de 2002 [44, 135]. Ces deux études reposent sur la théorie des martingales. Les limitations de ces études mentionnées au chapitre 3 sont essentiellement que les simplifications faites sur l'algorithme dans [44] ne garantissent pas que l'algorithme considéré soit plus pertinent qu'une recherche exhaustive et par ailleurs aucune vitesse de

convergence n'est étudié dans cet article. Dans [135], une partie de la démonstration repose sur des simulations numériques.

Dans l'analyse que nous allons faire ici, nous allons exploiter la dynamique markovienne de l'algorithme. L'étude que nous présentons est inspirée de l'article [23] dans lequel Bienvenue et François énoncent des propriétés de stabilité¹ sur une chaîne de Markov sous-jacente au $(1, \lambda)$ -SA-ES qui permettent de déduire la convergence de l'algorithme ainsi qu'une vitesse de convergence asymptotique. Ces propriétés de stabilité sont la récurrence de la chaîne de Markov ainsi que l'existence d'une mesure de probabilité invariante. La majeure partie de ce chapitre est consacrée à trouver des conditions sur les paramètres de l'algorithme pour garantir ces propriétés de stabilité. Nous allons donner une condition, exprimée de manière simple en fonction du moment de la variable aléatoire de la mutation et du nombre d'enfants λ sous laquelle ces propriétés de stabilité seront vérifiées. Nous allons pour cela utiliser des techniques classiques dans l'analyse d'algorithmes stochastiques [53, 52, 21] ou dans l'analyse de vitesse de convergence des méthodes de Monte Carlo (MCMC), [100, 143, 137, 98] reposant sur l'utilisation de conditions pratiques, qui seront pour nous des conditions de drift, pour l'établissement des conditions de stabilité.

Le plan que nous allons adopter dans ce chapitre est le suivant. Tout d'abord nous allons rapidement donner quelques rappels sur les outils que nous allons utiliser dans la suite, à savoir les chaînes de Markov sur un espace d'état continu. La section 5.2.1 va ensuite être consacrée à la description détaillée de l'algorithme $(1, \lambda)$ -SA-ES, puis la section 5.2.2 va définir une chaîne de Markov $(Z_n)_{n \geq 0}$ sous-jacente au $(1, \lambda)$ -SA-ES dont l'étude des conditions de stabilité va conduire à la convergence de l'algorithme. Les explications de ce lien entre convergence de l'algorithme et stabilité de $(Z_n)_{n \geq 0}$ sont données à la section 5.2.3 sous forme plus précisément de conditions suffisantes pour la convergence. Cette section 5.2.3 se conclura par une explication sur la démarche adoptée pour le reste du chapitre. Un résumé des résultats est donné en Section 5.2.4, en particulier le Théorème 4, qui regroupe tous les résultats concernant la convergence. Le reste du chapitre est ensuite consacré à l'étude de ces conditions de stabilité et plus précisément à la Section 5.3 à des propriétés de base pour $(Z_n)_{n \geq 0}$, à la Section 5.4 à l'établissement de condition de drift et enfin à la Section 5.5 à la fin des démonstrations des Théorèmes principaux.

1. Dans le contexte des chaînes de Markov, ce terme désigne un ensemble de propriétés relatives à la "stabilité" de la chaîne, en particulier l'irréductibilité, la récurrence d'une chaîne de Markov sont des propriétés de stabilité.

Chaînes de Markov : prérequis

Nous donnons ici un résumé des définitions et notions concernant les chaînes de Markov sur un espace d'état général (plus général que fini ou dénombrable) dont nous aurons besoin tout au long de ce chapitre. Pour plus de détails nous référons aux ouvrages de Meyn et Tweedie [100] de Revuz [123], les Théorèmes énoncés dont nous ferons usage dans ce chapitre portent la mention **M** et se trouvent dans [100], les références exactes étant données au fur et à mesure.

Pour l'exposé de ces notions, nous nous plaçons dans le cadre d'un espace d'état E que l'on suppose métrisable, muni de la tribu $\mathcal{B}(E)$. Simplement pour fixer les idées : pour nous, cet espace sera \mathbb{R}^d muni de la tribu borélienne $\mathcal{B}(\mathbb{R}^d)$.

Une chaîne de Markov est une suite de variables aléatoires que l'on peut se représenter comme évoluant au cours du temps avec une probabilité de transition dépendant simplement de l'état particulier dans lequel est la chaîne. Il est donc naturel de définir la chaîne avec son *noyau de transition*, fonction qui détermine ces transitions.

Noyau de Transition Un noyau de transition est une fonction P définie sur $(E \times \mathcal{B}(E))$ telle que

1. pour tout $A \in \mathcal{B}(E)$, $P(\cdot, A)$ est une fonction positive mesurable sur E .
2. pour tout $x \in E$, $P(x, \cdot)$ est une mesure de probabilité sur $\mathcal{B}(E)$.

La donnée d'un noyau de transition induit l'opération intégrale suivante : Etant donné une fonction f mesurable bornée, on définit

$$Pf(x) = \int_{\mathbb{E}} P(x, dy) f(y)$$

Chaîne de Markov Etant donné une mesure de probabilité μ sur $\mathcal{B}(E)$ et un noyau de transition P , on peut définir une mesure de probabilité \mathbb{P}_μ sur $\mathcal{B}(E)^{\otimes \mathbb{N}}$ et une suite de variables aléatoires $(X_n)_{n \geq 0}$ telle que pour tout $n \geq 0$ et tout $A_0 \dots A_n \in \mathcal{B}(E)$

$$\mathbb{P}_\mu(X_0 \in A_0, \dots, X_n \in A_n) = \int_{A_0 \times \dots \times A_n} \mu(dx_0) P(x_0, dx_1) \dots P(x_{n-1}, dx_n) \quad (5.1.1)$$

Alors une suite de variables aléatoire $(X_n)_{n \geq 0}$ est une chaîne de Markov de noyau de transition P et de loi initiale μ si pour tout n , (X_0, \dots, X_n) satisfait (5.1.1). Une autre définition équivalente est donnée ci-dessous.

Définition équivalente: Soit $(X_n)_{n \geq 0}$ une suite de variables aléatoires définies d'un espace filtré $(\Omega, \mathcal{F}_n, \mathbb{P})$ à valeur dans $(E, \mathcal{B}(E))$, P un noyau de transition et μ une mesure de probabilité sur $(E, \mathcal{B}(E))$. Alors $(X_n, n \geq 0)$ est une chaîne de Markov de noyau de transition P et de loi initiale μ si, X_n est \mathcal{F}_n mesurable, X_0 a pour loi μ et pour toute fonction f mesurable bornée on a

$$\mathbb{E}(f(X_{n+1})|\mathcal{F}_n) = Pf(X_n).$$

Des exemples classiques² de chaînes de Markov sont donnés par

$$X_{n+1} = f(X_n, Y_{n+1})$$

avec X_0 de loi μ , $(Y_n)_{n \geq 0}$ une suite de variables aléatoires indépendantes identiquement distribuées et f une fonction mesurable.

Dans la suite de cette section $(X_n)_{n \geq 0}$ désignera une chaîne de Markov de loi initiale μ et de noyau de transition P .

Noyau de transition au n-ième pas En notant le noyau de transition P par $P^1(., .)$, le noyau de transition au n-ième pas est défini par récurrence par

$$P^n(x, A) = \int_E P(x, dy)P^{n-1}(y, A), x \in E, A \in \mathcal{B}(E) \quad (5.1.2)$$

ou encore $P^n(x, A) = \mathbb{P}_x(X_n \in A)$ où \mathbb{P}_x désigne \mathbb{P}_{δ_x} .

Irréductibilité - Apériodicité Sur un espace d'état général on a besoin d'une mesure pour définir la notion d'irréductibilité. On dit que $(X_n)_{n \geq 0}$ est ϕ -irréductible si il existe une mesure non triviale ϕ sur $\mathcal{B}(E)$ (appelée mesure d'irréductibilité) telle que pour tout ensemble $A \in \mathcal{B}(E)$ de ϕ mesure non nulle, pour tout $x \in E$, il existe $n \geq 1$ et $P^n(x, A) > 0$. Une mesure d'irréductibilité φ est dite maximale si toute autre mesure d'irréductibilité φ' est dominée par φ ³. On montre qu'à partir d'une mesure d'irréductibilité, on peut toujours construire une mesure d'irréductibilité maximale et que les mesures maximales sont équivalentes⁴. Dorénavant, φ désignera donc une mesure d'irréductibilité maximale.

On dit que $(X_n)_{n \geq 0}$ est apériodique si il n'existe pas d ensembles mesurables $E_i, i = 1, \dots, d, d \geq 2$, dont l'union est de φ mesure 1 et tels que $P(x, E_{i+1}) = 1$ pour tout $x \in E_i, i = 1, \dots, d$ (avec $E_{d+1} = E_d$).

2. Les chaînes que nous considérons dans ce chapitre rentrent dans ce cadre

3. φ domine φ' si pour tout $A \in \mathcal{B}(E)$ tel que $\varphi'(A) > 0$ on a $\varphi(A) > 0$

4. Deux mesures φ et φ' sont équivalentes si φ domine φ' et φ' domine φ

Ensemble "small"⁵ Il est connu [100, Chapter 5] que pour une chaîne φ -irréductible, tout ensemble A tel que $\varphi(A) > 0$ contient un small set, *i.e.* un ensemble C tel que il existe $\delta > 0$, $n > 0$, et une mesure non triviale ν tels que

$$P^n(x, \cdot) \geq \delta \nu(\cdot), \quad x \in C. \quad (5.1.3)$$

On montre que la chaîne est **apériodique** si pour un (et ensuite pour tous) small set avec $\varphi(C) > 0$, 1 est le plus grand commun diviseur de toutes les valeurs n pour lesquelles l'équation (5.1.3) a lieu.

Récurrence et transience Soit $A \in \mathcal{B}(E)$, considérons la variable aléatoire comptant le nombre de visite de (X_n) à A , c'est-à-dire $\eta_A = \sum_{n=1}^{\infty} \mathbb{1}\{X_n \in A\}$.

Alors, l'ensemble A est uniformément transient si il existe $M < +\infty$ tel que $\mathbb{E}_x[\eta_A] \leq M$ pour tout $x \in A$. L'ensemble A est récurrent si $\mathbb{E}_x[\eta_A] = \infty$ pour tout $x \in A$.

Pour une chaîne φ -irréductible, il existe la dichotomie suivante: Soit tout ensemble $A \in \mathcal{B}(E)$ de φ mesure strictement positive est récurrent, soit il existe un recouvrement de E par des ensemble uniformément transients.

Récurrence au sens de Harris Sur un espace d'état continu, contrairement au cas fini ou dénombrable, il n'y a pas équivalence entre $\mathbb{E}_x[\eta_A] = \infty$ et $P_x(\eta_A = \infty) = 1$ pour $x \in A$. Seule l'implication de droite à gauche est vraie. Ceci emmène à la définition d'une forme plus forte de récurrence: la récurrence au sens de Harris.

Une chaîne φ -irréductible est récurrente au sens de Harris si, pour $A \in \mathcal{B}(E)$

$$\varphi(A) > 0 \Rightarrow \mathbb{P}_x(\eta_A = \infty) = 1, \quad x \in E \quad (5.1.4)$$

Mesure invariante - Positivité Une mesure μ σ -finie sur $\mathcal{B}(E)$ est invariante si pour tout $A \in \mathcal{B}(E)$

$$\mu(A) = \int_E \mu(dx) P(x, A)$$

On montre que toute chaîne récurrente admet une unique (à une constante multiplicative près) mesure invariante. Lorsque une mesure invariante est une mesure de probabilité, la chaîne est dite positive. On montre que toute chaîne positive est récurrente.

5. Nous ne traduisons pas ce terme par petit ensemble, car le concept de *petite set* existe aussi et désigne un objet différent

Loi Forte des Grands Nombres Une chaîne positive et Harris récurrente satisfait la loi forte des grands nombres (LGN). C'est ce qui est formulé dans le Théorème suivant [100, Théorème 17.0.1] :

Théorème M.1 (LGN pour des chaînes positives, récurrente au sens de Harris). *Supposons que X_n soit positive récurrente au sens de Harris et admette comme mesure invariante de probabilité μ , alors la LGN est vérifiée pour tout g satisfaisant $\mu(|g|) = \int |g|(x)\mu(dx) < +\infty$:*

$$\frac{1}{n} \sum_{k=0}^n g(X_k) \xrightarrow[n \rightarrow \infty]{p.s.} \mu(g)$$

Ergodicité Pour une chaîne φ -irréductible aperiodique et récurrente au sens de Harris et admettant μ comme mesure invariante, le noyau de transition va converger pour φ presque toute condition initiale $x \in E$ vers la mesure invariante μ :

$$\lim_n \|P^n(x, \cdot) - \mu(\cdot)\|_{VT} = 0$$

où la norme en *variation totale* d'une mesure signée π est donnée par

$$\|\pi\|_{VT} := \sup_{A \in \mathcal{B}(E)} \pi(A) - \inf_{A \in \mathcal{B}(E)} \pi(A)$$

La convergence en variation totale implique que pour toute fonction $g : E \rightarrow \mathbb{R}$ bornée, pour φ - presque tout x ,

$$\lim_n |P^n g(x) - \mu(g)| = 0$$

avec $P^n g(x) := \int P^{n-1} g(y) P(x, dy)$, $P^0(x, dy) := \delta_x(dy)$ et $\mu(g) := \int g(y)\mu(dy)$.

Nous allons considérer dans ce chapitre une notion plus forte d'ergodicité, en particulier à une notion garantissant la convergence géométrique vers la mesure stationnaire. Cette notion est l'ergodicité géométrique et sera définie plus bas dans le Théorème M.4.

Conditions de drift de Foster Lyapunov Il existe des conditions dites *conditions de drift* permettant d'établir des propriétés de stabilité d'une chaîne. Nous allons plus précisément énoncer des conditions de drift permettant d'établir qu'une chaîne est récurrente, positive, géométriquement ergodique parce que ce sont les conditions dont nous allons avoir besoin dans la suite.

On appelle condition de drift toute inégalité de la forme

$$\Delta V \leq -F + b\mathbb{1}_C$$

où F et V sont des fonctions mesurables positives, b est une constante finie et $C \in \mathcal{B}(E)$ et Δ est l'opérateur dit de drift défini par $\Delta V(x) := \int P(x, dy)V(y) - V(x)$

Les conditions de drift que nous énonçons sont plus particulièrement des conditions de Foster Lyapunov, elles font intervenir des fonctions V dites *non bornée en dehors des small sets*, qui sont telles que tous les sous ensembles suivants

$$C_V(n) = \{y : V(y) \leq n\}$$

sont des small sets.

Nous énonçons ci-dessous trois conditions de drift donnant des conditions suffisantes pour respectivement la récurrence au sens de Harris, la positivité et l'ergodicité géométrique.

Théorème M.2 (Conditions de drift pour la récurrence au sens de Harris [100, Théorème 9.1.8]). *Supposons que $(X_n)_{n \geq 0}$ est φ -irréductible. S'il existe un small set C et une fonction positive V non bornée en dehors des small sets telle que*

$$\forall x \in C^c, \Delta V(x) \leq 0, \quad (5.1.5)$$

alors $(X_n)_{n \geq 0}$ est récurrente au sens de Harris.

Théorème M.3 (Conditions de drift pour la positivité [100, Théorème 13.0.1]). *Supposons que $(X_n)_{n \geq 0}$ soit φ -irréductible apériodique et Harris récurrente. Alors $(X_n)_{n \geq 0}$ est positive si et seulement si il existe un small set C , une fonction V , non bornée en dehors des small sets, et un réel b tels que*

$$\Delta V(z) \leq -1 + b\mathbb{1}_C. \quad (5.1.6)$$

Théorème M.4 (Conditions pour l'ergodicité géométrique [100, Chapitre 15]). *Supposons que $(X_n)_{n \geq 0}$ est φ -irréductible et apériodique. Alors les propositions suivantes sont équivalentes pour tout $V \geq 1$,*

1. *La chaîne est géométriquement ergodique au sens où il existe une fonction $V \geq 1$, finie au moins pour un z , et des constantes $\rho < 1$ et $R < \infty$ tels que pour tout $n \in \mathbb{X}^+$ et tout z tel que la fonction V de l'équation (5.1.8) soit finie*

$$\|P^n(z, \cdot) - \mu\|_V \leq RV(z)\rho^n \quad (5.1.7)$$

où pour toute mesure signée μ la V norme est définie par $\|\nu\|_V = \sup_{|g| \leq V} |\int \nu(dy)g(y)|$.

2. Il existe une fonction $V \geq 1$, finie au moins pour un z , un small set C , $\lambda_C > 0$ et $b_C < \infty$ tels que la condition suivante dite de drift géométrique soit satisfaite

$$\Delta V \leq -\lambda_C V + b_C \mathbb{1}_C \quad (5.1.8)$$

Théorème Central Limite Un résultat clé qui découle de la preuve de l'ergodicité géométrique est le Théorème central limite (TCL) [100, Chapitre 17].

Pour toute fonction g telle que $\mu(|g|) < +\infty$, définissons

– $S_n(g)$ par

$$S_n(g) := \sum_{k=1}^n g(X_k).$$

– $\bar{g}(X_k) = g(X_k) - \mu(g)$.

On dit que le TCL est satisfait pour tout g dès qu'il existe une constante $0 \leq \gamma_g \leq \infty$ telle que pour toute condition initiale z

$$\frac{1}{\sqrt{n\gamma_g^2}} S_n(\bar{g}) \xrightarrow[n \rightarrow \infty]{loi} N(0, 1) \quad (5.1.9)$$

Comme énoncé dans [100, Théorème 17.0.1]:

Théorème M.5 (TCL pour des chaînes géométriquement ergodiques). *Supposons que (X_n) soit une chaîne de Markov admettant une mesure de probabilité invariante μ et qu'il existe V telle que l'équation (5.1.8) soit satisfaite. Soit g une fonction satisfaisant $g^2 \leq V$. Alors la valeur*

$$\gamma_g = \mathbb{E}_\mu[\bar{g}^2(X_0)] + 2 \sum_{k=1}^{\infty} \mathbb{E}_\mu[\bar{g}^2(X_0)\bar{g}^2(X_k)] \quad (5.1.10)$$

est bien définie, positive et finie, et

$$\lim_{n \rightarrow \infty} \frac{1}{n} \mathbb{E}_\mu \left[\left(\sum_{k=1}^n \bar{g}(X_k) \right)^2 \right] = \gamma_g^2$$

De plus, si $\gamma_g^2 > 0$, le TCL est satisfait pour g .

5.2 Algorithme $(1, \lambda)$ –SA-ES et chaînes de Markov

Nous commençons cette section par l'introduction de l'algorithme étudié ainsi que les notations (Section 5.2.1). Ensuite, en Section 5.2.2, nous introduisons une chaîne de Markov particulière dont l'étude va conduire à la convergence de l'algorithme étudié. Nous établissons ensuite en Section 5.2.3 des résultats préliminaires qui sont des conditions suffisantes pour la convergence, conditions suffisantes qui sont énoncées sous forme de propriétés de stabilité de la chaîne de Markov introduite en Section 5.2.2. C'est à l'étude précise de ces conditions de stabilité que le reste de ce chapitre est consacré et un résumé de ces principaux résultats est donné en Section 5.2.4. Pour plus de lisibilité nous omettons les preuves dans cette section et renvoyons à l'annexe 5.7.1 pour les démonstrations détaillées.

5.2.1 L'algorithme étudié

L'algorithme qui nous intéresse ici est une généralisation de l'algorithme $(1, \lambda)$ -SA-ES qui a été décrit dans le chapitre 1. Après avoir rappelé le cadre classique nous allons expliquer sa généralisation.

Le cadre classique: Soit $f : S \rightarrow \mathbb{R}$ une fonction que l'on souhaite minimiser sur l'espace de recherche $S \subset \mathbb{R}^d$. Le principe de l'algorithme $(1, \lambda)$ -SA-ES est de faire évoluer un vecteur de l'espace de recherche, *un parent*, que nous noterons X_n , ainsi qu'un paramètre réel positif σ_n qui joue le rôle de l'écart-type de la mutation (cf. Chapitre 1). Cette évolution est régie par les étapes suivantes:

Tout d'abord, l'étape de *mutation* consiste à créer λ enfants. Pour cela la mutation a lieu en deux temps. Premièrement on mute l'écart-type de la mutation σ_n en le multipliant par une variable aléatoire de loi log-normale. Deuxièmement on mute le parent X_n , en lui rajoutant une variable aléatoire gaussienne d'écart-type la valeur obtenue lors de la mutation de σ_n (cf (5.2.12)). Plus précisément, en se plaçant dans le cas où $S = \mathbb{R}$, cela donne pour chaque $i = 1 \dots \lambda$

$$\sigma_{n+1}^i = \sigma_n \exp(\tau \mathcal{N}_n^i(0, 1)) \quad (5.2.11)$$

$$X_{n+1}^i = X_n + \sigma_{n+1}^i \hat{\mathcal{N}}_n^i(0, 1) \quad (5.2.12)$$

où $\mathcal{N}_n^i(0, 1)$ et $\hat{\mathcal{N}}_n^i(0, 1)$ sont 2λ variables aléatoires indépendantes de loi normale, de moyenne 0, d'écart-type 1, $\tau \in \mathbb{R}^+$ un paramètre choisi par l'utilisateur, $\sigma_0 \in \mathbb{R}_*^+$ et $X_0 \in \mathbb{R}$ choisis par l'utilisateur également.

La deuxième étape est la *sélection* parmi les λ enfants du parent X_{n+1} (et de l'écart-type σ_{n+1} associé) pour la prochaine *génération*. Le nouveau parent X_{n+1} est celui parmi les λ enfants qui minimise la fonction f

$$X_{n+1} = \arg \min \{f(X_{n+1}^i), i = 1 \dots \lambda\} \stackrel{\text{def.}}{=} X_{n+1}^* \quad (5.2.13)$$

et le nouvel écart-type σ_{n+1} est celui qui a créé l'enfant sélectionné

$$\sigma_{n+1} = \sigma_{n+1}^* \quad (5.2.14)$$

Comme expliqué au chapitre 1, σ_n est ajusté par l'évolution elle-même, ce qui explique la terminologie d'algorithme *auto-adaptatif*.

Le cadre plus général: Le cadre que nous allons considérer est plus général en ce sens que la mutation de l'écart type (5.2.11) va maintenant consister à multiplier σ_n par une variable aléatoire positive η_n^i pour $1 \leq i \leq \lambda$. Nous supposons que cette variable aléatoire est absolument continue par rapport à la mesure de Lebesgue, admettant une densité par rapport à la mesure de Lebesgue que nous noterons p_η et de loi que nous noterons ν_η . De même les variables aléatoires $\hat{\mathcal{N}}^i(0, 1)$ pour $1 \leq i \leq \lambda$ de l'équation (5.2.12) sont remplacées par des variables aléatoires symétriques plus générales notées ξ_n^i pour $1 \leq i \leq \lambda$, que l'on va aussi supposer absolument continues par rapport à la mesure de Lebesgue, de densité p_ξ et de distribution ν_ξ . Par ailleurs, nous aurons besoin pour démontrer les résultats de ce chapitre de faire d'autres hypothèses sur ces densités que nous énoncerons au fur et à mesure.

La fonction qui va nous intéresser est la fonction sphère

$$f(x) = \|x\|$$

et par souci de simplicité nous allons exposer les résultats en dimension 1, c'est à dire que l'espace de recherche est \mathbb{R} muni de la tribu Borélienne $\mathcal{B}(\mathbb{R})$. Ainsi sauf mention explicite du contraire, nous considérons dans toute la suite que la fonction f est la fonction sphère et que $d = 1$. L'algorithme que nous allons étudier s'écrit alors

$$X_{n+1} = \arg \min \{ \|X_n + \sigma_n \eta_n^1 \xi_n^1\|, \dots, \|X_n + \sigma_n \eta_n^\lambda \xi_n^\lambda\| \} \quad (5.2.15)$$

$$\sigma_{n+1} = \sigma_n \eta_n^*, \star \text{ défini par } X_{n+1} = X_n + \sigma_n \eta_n^* \xi_n^* \quad (5.2.16)$$

avec $X_0 \in \mathbb{R}$ et $\sigma_0 \in \mathbb{R}^+$. Notons que seul l'ordre induit par f compte pour la définition de $(X_n)_{n \geq 0}$, ainsi pour l'algorithme considéré, il revient au même (au sens où les processus sont

identiques) de minimiser $f(x) = \|x\|$ ou n'importe quelle fonction symétrique (par exemple $f(x) = \ln(1 + |x|)$).

Remarque 1. La notation \star que nous utilisons pour l'indice correspondant à l'enfant sélectionné ne doit pas faire oublier le fait que, a priori, η_n^\star ainsi que ξ_n^\star dépendent de X_n , σ_n , η_n^i et ξ_n^i pour $1 \leq i \leq \lambda$ et que η_n^\star et ξ_n^\star ne sont pas indépendantes.

5.2.2 Définition de la chaîne de Markov sous-jacente au $(1, \lambda)$ -SA-ES

Parce que nous nous plaçons sur la fonction sphère, la dépendance de l'indice \star mentionnée à la Remarque 1 se simplifie. Considérons en effet le processus $Z_n \stackrel{p.s.}{=} \frac{X_n}{\sigma_n}$, alors $(Z_n)_{n \geq 0}$ est une chaîne de Markov. Plus précisément on a le Lemme suivant

Lemme 1. Soit (X_n, σ_n) défini par les équations (5.2.15) et (5.2.16), et $Z_n \stackrel{p.s.}{=} \frac{X_n}{\sigma_n}$. Alors $(Z_n)_{n \in \mathbb{N}}$ est une chaîne de Markov homogène telle que

$$Z_{n+1} = \frac{Z_n + \eta_n^\star(Z_n)\xi_n^\star(Z_n)}{\eta_n^\star(Z_n)} \quad (5.2.17)$$

avec $Z_0 = \frac{X_0}{\sigma_0} \in \mathbb{R}$ et où $\eta_n^\star(Z_n)$ et $\xi_n^\star(Z_n)$ sont définies par

$$\|Z_n + \eta_n^\star(Z_n)\xi_n^\star(Z_n)\| = \min_{1 \leq i \leq \lambda} \|Z_n + \eta_n^i \xi_n^i\|. \quad (5.2.18)$$

où η_n^i et ξ_n^i sont les λ tirages indépendants qui ont engendré (X_n, σ_n) (équations (5.2.15) et (5.2.16)).

Le point clé de la preuve qui est donnée en Annexe 5.7.1 vient de l'égalité

$$\min_{1 \leq i \leq \lambda} \|X_n + \sigma_n \eta_n^i \xi_n^i\| = \sigma_n \min_{1 \leq i \leq \lambda} \left\| \frac{X_n}{\sigma_n} + \eta_n^i \xi_n^i \right\|$$

de laquelle on déduit que le couple que nous notons $(\eta_n^\star(X_n, \sigma_n), \xi_n^\star(X_n, \sigma_n))$ réalisant le minimum à gauche c'est-à-dire tel que

$$\|X_n + \sigma_n \eta_n^\star(X_n, \sigma_n)\xi_n^\star(X_n, \sigma_n)\| = \min_{1 \leq i \leq \lambda} \|X_n + \sigma_n \eta_n^i \xi_n^i\|$$

est le même que le couple, $(\eta_n^\star(X_n/\sigma_n, 1), \xi_n^\star(X_n/\sigma_n, 1))$ réalisant le minimum à droite. Avec des notations évidentes nous résumons cela en:

$$(\eta_n^\star(X_n, \sigma_n), \xi_n^\star(X_n, \sigma_n)) \stackrel{p.s.}{=} (\eta_n^\star(Z_n), \xi_n^\star(Z_n))$$

Remarque 2. Ce précédent Lemme diffère du Lemme 4.1 de l'article [23] où Bienvenüe et François considèrent la chaîne $Z_n = \frac{\|X_n\|}{\sigma_n}$ pour laquelle l'égalité suivante

$$Z_{n+1} = \frac{\|X_{n+1}\|}{\sigma_{n+1}} = \frac{\|Z_n + \eta_n^*(Z_n)\xi_n^*(Z_n)\|}{\eta_n^*(Z_n)} \quad (5.2.19)$$

n'est pas vraie presque sûrement mais en loi. Avoir une égalité presque sûre entre $(Z_n)_{n \geq 0}$ et $(X_n, \sigma_n)_{n \geq 0}$ est ce qui va permettre d'établir la convergence **presque sûre** de $\frac{1}{n} \ln(\|X_n\|)$.

Notations : Les variables aléatoires $(\eta_1^*(z), \xi_1^*(z))$ définies à l'équation (5.2.18) seront notées par la suite $(\eta^*(z), \xi^*(z))$ pour alléger les notations. Notons que ces variables aléatoires à l'étape n dépendent de la valeur de $(Z_n)_{n \geq 0}$ mais aussi de η_n^i et ξ_n^i pour $1 \leq i \leq \lambda$, une notation plus explicite mais plus lourde que nous n'utiliserons donc pas serait :

$$(\eta^*(z, \eta_1, \dots, \xi_\lambda), \xi_1^*(z, \eta_1, \dots, \xi_\lambda))$$

. Remarquons aussi que nous avons omis dans la notation de la chaîne (X_n, σ_n) ou de la chaîne $(Z_n)_{n \geq 0}$ la dépendance par rapport à λ .

5.2.3 Résultats préliminaires: Conditions suffisantes sur $(Z_n)_{n \geq 0}$ pour la convergence du $(1, \lambda)$ –SA-ES

Dans [23, Théorème 4.1], les auteurs prouvent que s'intéresser à des propriétés de stabilité de la chaîne définie par l'équation (5.2.19) permet d'établir la convergence de $\frac{1}{n} \ln(\|X_n\|)$. Ceci est encore vrai pour la chaîne $(Z_n)_{n \geq 0}$ définie dans le Lemme 1. Le lien entre stabilité de $(Z_n)_{n \geq 0}$ et convergence de $\frac{1}{n} \ln(\|X_n\|)$ repose sur l'égalité suivante

$$\frac{1}{n} \ln(\|X_n\|) = \frac{1}{n} \ln(\|X_0\|) + \frac{1}{n} \sum_{k=0}^{n-1} \ln\left(\left\|\frac{Z_k + \eta_k^*(Z_k)\xi_k^*(Z_k)}{Z_k}\right\|\right) \quad (5.2.20)$$

où l'on voit que si la chaîne $(Z_n)_{n \geq 0}$ possède les propriétés garantissant qu'elle vérifie une loi forte des grands nombres, alors le terme de droite converge, d'où on en déduit la convergence de terme de gauche. Comme rappelé dans l'introduction, au Théorème M.1, une condition suffisante pour qu'une chaîne de Markov vérifie la Loi Forte des Grands nombres est qu'elle soit positive, récurrente au sens de Harris. Ceci nous fournit donc des conditions suffisantes pour la convergence. Le Théorème suivant démontré en Annexe 5.7.1 formalise ces idées.

Théorème 1. *Supposons que la chaîne de Markov $(Z_n)_{n \geq 0}$ est positive récurrente au sens de Harris de mesure de probabilité invariante μ et supposons que $\ln((\min_{1 \leq i \leq \lambda} \|z + \eta^i \xi^i\|)/\|z\|)$*

est intégrable contre $\mu \otimes \nu_\eta \otimes \dots \otimes \nu_\eta \otimes \nu_\xi \dots \otimes \nu_\xi$. Alors

$$\frac{1}{n} \ln(\|X_n\|) \xrightarrow[n \rightarrow \infty]{p.s.} \int_{\mathbb{R}} E[\ln \left\| \frac{z + \eta^*(z)\xi^*(z)}{z} \right\|] d\mu(z). \quad (5.2.21)$$

Remarque 3. La notation $\int_{\mathbb{R}} E[\ln \left\| \frac{z + \eta^*(z)\xi^*(z)}{z} \right\|] d\mu(z)$ désigne :

$$\int \dots \int (\ln(\frac{\min_{1 \leq i \leq \lambda} \|z + x^i y^i\|}{\|z\|})) \nu_\eta(dx^1) \dots \nu_\eta(dx^\lambda) \nu_\xi(dy^1) \dots \nu_\xi(dy^\lambda) \mu(dz).$$

Une équation similaire à l'équation (5.2.20) peut être formulée pour σ_n , en effet de $\sigma_{n+1} = \sigma_n \eta^*(Z_n)$ (on a en effet vu que $\eta^*(X_n/\sigma_n) = \eta^*(X_n, \sigma_n)$ au sens presque sûr), on déduit que

$$\frac{1}{n} \ln(\sigma_n) = \frac{1}{n} \ln(\sigma_0) + \frac{1}{n} \sum_{k=0}^{n-1} \ln(\eta^*(Z_k)) \quad (5.2.22)$$

Sous l'hypothèse que $(Z_n)_{n \geq 0}$ est positive Harris récurrente, le terme de droite converge et on déduit le Théorème suivant dont la preuve immédiate est donnée dessous.

Théorème 2. Supposons que la chaîne $(Z_n)_{n \geq 0}$ est positive récurrente au sens de Harris, de mesure de probabilité invariante μ et supposons de plus que $|\ln(\eta)| < +\infty$. Alors

$$\frac{1}{n} \ln(\sigma_n) \xrightarrow[n \rightarrow \infty]{p.s.} \int_{\mathbb{R}} E[\ln(\eta^*(z))] d\mu(z) \quad (5.2.23)$$

où μ est la mesure invariante de $(Z_n)_{n \geq 0}$.

PREUVE: La chaîne $(Z_n)_{n \geq 0}$ étant positive, récurrente au sens de Harris, on applique la loi forte des grands nombres au terme de droite de (5.5.58). L'intégrabilité de $\ln(\eta^*(z))$ est ici simple à prouver et se déduit de l'intégrabilité de $\ln(\eta)$. En effet, on a l'inégalité suivante

$$|\ln(\eta^*(z))| \leq \max(|\ln(\eta^{1:\lambda})|, |\ln(\eta^{\lambda:\lambda})|)$$

où $\eta^{1:\lambda}$ et $\eta^{\lambda:\lambda}$ représentent respectivement la première et la dernière statistique d'ordre parmi les λ variables aléatoires à partir desquelles ont été choisi $\eta^*(z)$. De plus de la définition de $\eta^{1:\lambda}$ et $\eta^{\lambda:\lambda}$, on déduit que $E(|\ln(\eta^{1:\lambda})|) \leq \lambda E(|\ln(\eta)|)$ et $E(|\ln(\eta^{\lambda:\lambda})|) \leq \lambda E(|\ln(\eta)|)$. Ainsi $\ln(\eta^*(z))$ est intégrable.

□

Remarquons de plus que nous allons montrer à la Proposition 5 que sous certaines hypothèses

$$\int_{\mathbb{R}} E[\ln(\eta^*(z))] d\mu(z) = \int_{\mathbb{R}} E[\ln(\frac{z + \eta^*(z)\xi^*(z)}{z})] d\mu(z)$$

Ceci implique en particulier que sous les hypothèses de la Proposition 5, $\frac{1}{n} \ln(\|X_n\|)$ et $\frac{1}{n} \ln(\sigma_n)$ convergent vers la même quantité.

D'autre part nous allons nous intéresser à des propriétés d'ergodicité géométrique de la chaîne $(Z_n)_{n \geq 0}$, *i.e.* propriétés, garantissant que le noyau de transition de $(Z_n)_{n \geq 0}$ au nième pas $P^n(\cdot, \cdot)$ converge vers la mesure de probabilité invariante de la chaîne $(Z_n)_{n \geq 0}$ à une vitesse géométrique. (Voir Théorème [M.4] pour la définition formelle). Lorsque une chaîne de Markov est géométriquement ergodique, alors elle vérifie le Théorème central limite (TCL) qui va dans notre cas impliquer le Théorème suivant, que nous démontrerons à la Section 5.5.4:

Théorème 3. *Soit $(Z_n)_{n \geq 0}$ définie comme au Lemme 1. Si on suppose que $(Z_n)_{n \geq 0}$ est géométriquement ergodique, alors*

$$\sqrt{n} \left(\frac{1}{n} \sum_{k=0}^{n-1} \ln(\eta^*(Z_k)) - \int_{\mathbb{R}} E[\ln(\eta^*(z))] d\mu(z) \right) \xrightarrow[n \rightarrow \infty]{\mathcal{L}} \mathcal{N}(0, \gamma^2)$$

où γ est donné par

$$\gamma^2 = \lim_{n \rightarrow \infty} \frac{1}{n} \mathbb{E}_{\mu} \left[\left(\frac{1}{n} \sum_{k=0}^{n-1} \ln(\eta^*(Z_k)) - \int_{\mathbb{R}} E[\ln(\eta^*(z))] d\mu(z) \right)^2 \right]$$

La conséquence essentielle de ce Théorème est de fournir des intervalles de confiance pour l'approximation du taux de convergence $\int_{\mathbb{R}} E[\ln(\eta^*(z))] d\mu(z)$ (cf Section 5.5.5).

Cheminement du reste du chapitre Nous sommes maintenant en mesure de formuler un plan précis de ce qui va être notre démarche dans le reste de ce chapitre. Le but est d'étudier la chaîne $(Z_n)_{n \geq 0}$ pour voir dans quel cas les hypothèses des Théorèmes 1, 2 et 3 sont satisfaites. Plus précisément on va s'intéresser à ses propriétés de récurrence au sens de Harris, de positivité et d'ergodicité géométrique. Dans un premier temps nous allons démontrer des propriétés de base sur la chaîne $(Z_n)_{n \geq 0}$, à savoir sa φ -irréductibilité, son apériodicité et identifier que les compacts non vides sont des small sets. Cela fera l'objet de la Section 5.3. Ensuite nous allons effectivement nous attaquer aux propriétés de récurrence au sens de Harris, de positivité et d'ergodicité géométrique. Nous allons chercher pour cela des conditions de drift et plus particulièrement des conditions de drift de Foster-Lyapunov telles qu'énoncées aux Théorèmes M.2, M.3 et M.4, impliquant respectivement la récurrence au sens de Harris, la positivité et l'ergodicité géométrique. Etablir ces conditions sera l'objet de la Section 5.4. Notons que lorsque nous parlons d'établir ces conditions, nous entendons

chercher des conditions sur les données de l'algorithme (à savoir sur les variables aléatoires η , ξ et la nombre d'enfants λ) pour que les propriétés de stabilité de $(Z_n)_{n \geq 0}$ soient vérifiées. Enfin, à la Section 5.5 nous déduisons de ces conditions de drift que les hypothèses des Théorèmes 1, 2 et 3 sont satisfaites, en particulier l'intégrabilité du membre de droite de l'équation (5.2.21) sera déduite de ces conditions de drift ainsi que d'autres propriétés sur la mesure stationnaire que nous établirons.

5.2.4 Résumé des résultats principaux: Conditions garantissant la stabilité de $(Z_n)_{n \geq 0}$

Un résumé des principaux résultats de la suite de ce chapitre est donné ici. Commençons par formaliser les hypothèses que nous avons faites lors de la description de l'algorithme (Section 5.2.1):

Hypothèses H.1. (i) ξ est symétrique

(ii) η et ξ sont absolument continues par rapport à la mesure de Lebesgue de densités respectives p_η et p_ξ .

Rappelons que ces hypothèses incluent le cas classique de l'algorithme $(1, \lambda)$ -SA-ES pour lequel η est suit loi log-normale et ξ une loi normale. C'est par ailleurs ce cadre classique qui nous a guidé dans ce choix d'hypothèses, même si considérer par exemple que η est singulière simplifierait certaines preuves et permettrait même de préciser certains point important (cf Section 5.6 pour une discussion plus détaillée sur ce sujet).

Dans un premier temps, à la Section 5.3 nous allons nous intéresser à des propriétés de base sur $(Z_n)_{n \geq 0}$, à savoir, décrire son noyau de transition, montrer son irréductibilité, son apériodicité et identifier que les compacts non vides de \mathbb{R} sont des small sets. Les hypothèses faites pour montrer cela sont les suivantes:

Hypothèses H.2. (i) p_ξ est continue presque partout, $p_\xi \in L^\infty(\mathbb{R})$ et $0 \in \mathring{\text{supp}} p_\xi$ (i.e. $\exists B_\xi > 0$ t.q. $[-B_\xi, B_\xi] \subset \text{supp } p_\xi$),

(ii) $1 \in \mathring{\text{supp}} p_\eta$.

et le Théorème qui sera montré est:

Théorème Soit $(Z_n)_{n \geq 0}$ la chaîne de Markov définie par les équations (5.2.15) et (5.2.16).

On suppose que p_ξ et p_η satisfont les hypothèses **H.1** et **H.2**. Alors $(Z_n)_{n \geq 0}$ est μ^{Leb} -irréductible⁶, aperiodique et les compacts non vides de \mathbb{R} sont des small sets.

Ensuite dans la section 5.4, nous allons chercher des conditions de *drift* garantissant la récurrence au sens de Harris, la positivité, et l'ergodicité géométrique de $(Z_n)_{n \geq 0}$. Ces conditions de drift vont être établies sous la condition que l'espérance de l'inverse de la variable aléatoire $\hat{\eta}(\lambda)$ définie ci-dessous est plus petite que 1. De manière plus précise, définissons le couple de variables aléatoires $(\hat{\eta}(\lambda), \hat{\xi}(\lambda))$ comme étant le couple réalisant le minimum parmi λ tirages indépendants de couples de variables aléatoires de lois ν_η et ν_ξ . C'est à dire

$$\hat{\eta}(\lambda)\hat{\xi}(\lambda) = \min\{\eta^1\xi^1, \dots, \eta^\lambda\xi^\lambda\} \quad (5.2.24)$$

où η^i pour $1 \leq i \leq \lambda$ sont λ variables aléatoires indépendantes de loi ν_η et ξ^i pour $1 \leq i \leq \lambda$ sont λ variables aléatoires indépendantes de loi ν_ξ .

Définissons Γ_0 de la façon suivante

$$\Gamma_0 = \{\gamma > 0 \text{ tel que } \mathbb{E}(|\frac{1}{\eta}|^\gamma) < +\infty \text{ et } \mathbb{E}(|\xi|^\gamma) < +\infty\}.$$

Définissons Λ par

$$\Lambda = \{\lambda \text{ tel que } \exists \alpha \in \Gamma_0 \text{ tel que } \mathbb{E}(\frac{1}{\hat{\eta}(\lambda)^\alpha}) < 1\}, \quad (5.2.25)$$

Une autre définition équivalente dont nous aurons besoin pour Λ est $\Lambda = \bigcup_{\alpha \in \Gamma_0} \Lambda_\alpha$ avec

$$\Lambda_\alpha = \left\{ \lambda \in \mathbb{N} \text{ tels que } E\left(\left(\frac{1}{\hat{\eta}(\lambda)}\right)^\alpha\right) < 1 \right\}$$

Alors nous allons montrer au Théorème 6 que si η et ξ satisfont les Hypothèses **H.1** et **H.2** pour tout $\lambda \in \Lambda$, $V(z) = |z|^\alpha + 1$ est une fonction de drift où $\alpha \in \Gamma_0$ est tel que $\mathbb{E}(\frac{1}{\hat{\eta}(\lambda)^\alpha}) < 1$.

De cette condition de drift et de l'hypothèse suivante

Hypothèses H.3. $\mathbb{E}(|\ln(\eta)|) < +\infty$

nous déduisons l'intégrabilité des taux de convergence dans les équations (5.2.21) et (5.2.23). La conséquence en sera le Théorème suivant, qui résume les résultats des Théorème 8, Théorème 9 et Théorème 10 qui font l'objet de la fin de ce chapitre:

6. μ^{Leb} désigne la mesure de Lebesgue sur \mathbb{R}

Théorème 4. Soient η et ξ satisfaisant les Hypothèses **H.1**, **H.2** et **H.3**, alors pour tout $\lambda \in \Lambda$

$$\begin{aligned} \frac{1}{n} \ln(\|X_n\|) &\xrightarrow[n \rightarrow \infty]{p.s.} \int_{\mathbb{R}} \mathbb{E}[\ln \left\| \frac{z + \eta^*(z)\xi^*(z)}{z} \right\|] d\mu(z) \\ \frac{1}{n} \ln(\sigma_n) &\xrightarrow[n \rightarrow \infty]{p.s.} \int_{\mathbb{R}} \mathbb{E}[\ln(\eta^*(z))] d\mu(z) \end{aligned}$$

et

$$\int_{\mathbb{R}} \mathbb{E}[\ln \left\| \frac{z + \eta^*(z)\xi^*(z)}{z} \right\|] d\mu(z) = \int_{\mathbb{R}} \mathbb{E}[\ln(\eta^*(z))] d\mu(z)$$

De plus on a

$$\sqrt{n} \left(\frac{1}{n} \sum_{k=0}^{n-1} \ln(\eta^*(Z_k)) - \int_{\mathbb{R}} \mathbb{E}[\ln(\eta^*(z))] d\mu(z) \right) \xrightarrow[n \rightarrow \infty]{\mathcal{L}} \mathcal{N}(0, \gamma^2)$$

où γ est donné par

$$\gamma^2 = \lim_{n \rightarrow \infty} \frac{1}{n} \mathbb{E}_{\mu} \left[\left(\sum_{k=1}^n (\ln(\eta^*(Z_k)) - \int_{\mathbb{R}} \mathbb{E}[\ln(\eta^*(z))] d\mu(z)) \right)^2 \right]$$

Remarque 4. La convergence de $\frac{1}{n} \ln(\|X_n\|)$ ou de $\frac{1}{n} \ln(\sigma_n)$ ne permet pas de conclure directement à la convergence de $\|X_n\|$ ou σ_n puisqu'il faudrait pour cela connaître le signe de la limite. Dans le cas de l'algorithme $(1, \lambda)$ -SA-ES nous ne sommes pas capables de donner ce signe en général.

Remarque 5. Ce Théorème illustre la propriété d'auto-adaptation, bien connue des utilisateurs d'algorithmes évolutionnaires, que le pas de la mutation σ_n converge⁷ vers 0 tout comme $\|X_n\|$. Par ailleurs le résultat est ici encore plus fort puisque $\|X_n\|$ et σ_n convergent asymptotiquement à la même vitesse.

Pour insister sur cette notion d'auto-adaptation, mettons en parallèle ce résultat avec le résultat que l'on obtient sans auto-adaptation dans le cadre de l'algorithme $(1, \lambda)$ -ES à pas constant que nous traitons dans le détail en Annexe de ce chapitre, à la Section 5.7. Dans ce cas on a :

$$\frac{1}{n} \ln \|X_n\| \rightarrow 0$$

De même comme le pas σ_n est constant on a aussi

$$\frac{1}{n} \ln \|\sigma_n\| \rightarrow 0.$$

7. dans le cas où le signe du taux de convergence est négatif

Il n'est pas évident de trouver théoriquement dans un cas général à quoi est égal l'ensemble Λ_α , il est cependant très facile de simuler $E(\frac{1}{\eta(\lambda)})^\alpha$ et de trouver numériquement l'ensemble Λ_α , nous illustrerons cela dans la section 5.4.1.2. Nous prouverons à la Proposition 1 que si les densités p_η et p_ξ sont à support borné (plus précisément si il existe γ avec $0 < \gamma < 1$ et $\beta > 0$ tel que $\text{supp } p_\eta = [1 - \gamma, 1 + \beta]$ et il existe $A > 0$ tel que $\text{supp } p_\xi = [-A, A]$), alors il existe un $\lambda_0 \in \mathbb{N}^*$ tel que

$$\{\lambda_0, \dots, +\infty\} \subset \Lambda_\alpha.$$

Notons aussi que le cas de l'algorithme pour $\lambda = 1$, qui s'apparente alors à une marche aléatoire et ne présente pas d'intérêts d'un point de vue optimisation, sera traité en détail dans la partie Annexe en Section 5.7.3.

5.3 Irréductibilité, small sets et apériodicité pour Z_n

5.3.1 Noyau de transition pour $(Z_n)_{n \geq 0}$

Lemme 2. Soit $(Z_n)_{n \geq 0}$ la chaîne de Markov définie par (5.2.17) et (5.2.18). Supposons que η et ξ vérifient l'hypothèse **H.1** (ii). Alors le noyau de transition de $(Z_n)_{n \geq 0}$ est absolument continu par rapport à la mesure de Lebesgue, plus précisément, pour tout $z \in \mathbb{R}$, $A \in \mathcal{B}(\mathbb{R})$ le noyau de transition est donné par

$$P(z, A) = \int_{\mathbb{R}} \mathbb{1}_A(v) p(z, v) dv \quad (5.3.26)$$

où pour $z \in \mathbb{R}$ et $v \in \mathbb{R}$

$$p(z, v) = \lambda \int_{\mathbb{R}^+} p_\eta(u) p_\xi(v - \frac{z}{u}) \mathbb{P}(|uv| < |z + \eta\xi|)^{\lambda-1} du. \quad (5.3.27)$$

PREUVE: Soit $A \in \mathcal{B}(\mathbb{R})$. Par définition $P(z, A) = \mathbb{P}(\frac{z + \eta^*(z)\xi^*(z)}{\eta^*(z)} \in A)$. Comme les λ couples de variables aléatoires (η_1^i, ξ_1^i) pour $1 \leq i \leq \lambda$ jouent le même rôle

$$P(z, A) = \lambda \mathbb{P}(\{\frac{z + \eta_1^1 \xi_1^1}{\eta_1^1} \in A\} \cap \bigcap_{i=2}^{\lambda} \{|z + \eta_1^i \xi_1^i| < |z + \eta_1^1 \xi_1^1|\})$$

ou encore $P(z, A) = \lambda \mathbb{E}(\mathbb{1}_{\{\frac{z + \eta_1^1 \xi_1^1}{\eta_1^1} \in A\}} \prod_{i=2}^{\lambda} \mathbb{1}_{\{|z + \eta_1^i \xi_1^i| < |z + \eta_1^1 \xi_1^1|\}})$. En conditionnant par rapport à (η_1^1, ξ_1^1) nous obtenons

$$P(z, A) = \lambda \mathbb{E} \left(\mathbb{1}_{\{\frac{z + \eta_1^1 \xi_1^1}{\eta_1^1} \in A\}} \mathbb{E} \left(\prod_{i=2}^{\lambda} \mathbb{1}_{\{|z + \eta_1^i \xi_1^i| < |z + \eta_1^1 \xi_1^1|\}} \mid \eta_1^1, \xi_1^1 \right) \right)$$

Les couples de variables aléatoires $(\eta_1^i, \xi_1^i)_{2 \leq i \leq \lambda}$ étant indépendants et de même loi on a

$$P(z, A) = \lambda \mathbb{E} \left(\mathbb{1}_{\left\{ \frac{z + \eta_1^1 \xi_1^1}{\eta_1^1} \in A \right\}} \mathbb{E} \left(\mathbb{1}_{\{|z + \eta_1^1 \xi_1^1| < |z + \eta_1^2 \xi_1^2|\}} | \eta_1^1, \xi_1^1 \right)^{\lambda-1} \right)$$

En introduisant les densités p_η et p_ξ

$$P(z, A) = \lambda \iint \mathbb{1}_A \left(\frac{z + xy}{x} \right) p_\eta(x) p_\xi(y) \mathbb{P}(|z + xy| < |z + \eta\xi|)^{\lambda-1} dx dy \quad (5.3.28)$$

où η et ξ désignent deux variables aléatoires indépendantes de lois respectives ν_η et ν_ξ . Un changement de variable dans l'équation (5.3.28) donne

$$P(z, A) = \lambda \iint \mathbb{1}_A(v) p_\eta(u) p_\xi \left(v - \frac{z}{u} \right) \mathbb{P}(|uv| < |z + \eta\xi|)^{\lambda-1} du dv \quad (5.3.29)$$

En introduisant $p(z, v)$ défini par l'équation (5.3.27), nous obtenons

$$P(z, A) = \int_{\mathbb{R}} \mathbb{1}_A(v) p(z, v) dv.$$

□

Remarque 6. *Sous l'hypothèse que $p_\xi \in L^\infty(\mathbb{R})$, on remarque, d'après l'expression de la densité p donnée par (5.3.27), que $p \in L^\infty(\mathbb{R} \times \mathbb{R})$.*

5.3.2 φ -irréductibilité, apériodicité, small sets

Théorème 5. *Soit $(Z_n)_{n \geq 0}$ la chaîne de Markov définie par les équations (5.2.15) et (5.2.16).*

*On suppose que p_ξ et p_η satisfont les hypothèses **H.1** et **H.2**. Alors $(Z_n)_{n \geq 0}$ est μ^{Leb} -irréductible, apériodique et les compacts non vides de \mathbb{R} sont des small sets.*

PREUVE: Bien que l'idée de base de ce Théorème est très simple (prendre le min de l'équation (5.3.26)), la preuve est assez technique nécessitant l'établissement de deux Lemmes intermédiaires, c'est pourquoi nous avons décidé de donner sa preuve en Annexe, section 5.7.4.

□

5.4 Conditions de drift pour les propriétés de récurrence au sens de Harris, de positivité et d'ergodicité géométrique _____

Le but de cette section est de trouver des conditions liant λ , η et ξ pour que $(Z_n)_{n \geq 0}$ satisfasse des conditions de drift qui vont garantir qu'elle est positive, récurrente au sens de Harris

et géométriquement ergodique. Nous allons dans un premier temps montrer que l'équation (5.1.8) est satisfaite. Une conséquence immédiate sera que les équations (5.1.5) et (5.1.6) seront également satisfaites.

Rappelons la définition des variables aléatoires $\hat{\eta}(\lambda)$ et $\hat{\xi}(\lambda)$

$$\hat{\eta}(\lambda)\hat{\xi}(\lambda) = \min\{\eta^1\xi^1, \dots, \eta^\lambda\xi^\lambda\} \quad (5.4.30)$$

où η^i pour $1 \leq i \leq \lambda$ sont λ variables aléatoires indépendantes de loi ν_η et ξ^i pour $1 \leq i \leq \lambda$ sont λ variables aléatoires indépendantes de loi ν_ξ . Pour simplifier les notations introduisons χ la variable aléatoire $\eta\xi$ où η et ξ sont indépendantes. La densité de $\hat{\eta}(\lambda)$ est alors donnée par

$$p_{\hat{\eta}(\lambda)}(t) = \lambda p_\eta(t) \int_{\mathbb{R}} p_\xi(u) (1 - F_\chi(tu))^{\lambda-1} du. \quad (5.4.31)$$

où $F_\chi(t) = P(\chi \leq t)$ désigne la fonction de répartition de χ . A λ fixé, $p_{\eta^*(z)}(t)$ converge lorsque $|z|$ tend vers $+\infty$ vers $p_{\hat{\eta}}(t)$, c'est ce qui est établi dans le Lemme suivant :

Lemme 3. *La densité de $\eta^*(z)$ est donnée par*

$$p_{\eta^*(z)}(t) = \lambda p_\eta(t) \int p_\xi(u) P(|z + tu| \leq |z + \chi|)^{\lambda-1} du \quad (5.4.32)$$

Pour presque tout $t \in \mathbb{R}^+$

$$\lim_{|z| \rightarrow +\infty} p_{\eta^*(z)}(t) = p_{\hat{\eta}(\lambda)}(t). \quad (5.4.33)$$

Considérons $\alpha > 0$ tel que $\mathbb{E}((\frac{1}{\eta})^\alpha) < +\infty$, alors

$$\lim_{|z| \rightarrow +\infty} \mathbb{E}((\frac{1}{\eta^*(z)})^\alpha) = \mathbb{E}((\frac{1}{\hat{\eta}(\lambda)})^\alpha) \quad (5.4.34)$$

PREUVE: L'équation (5.4.32) vient du fait que $\eta^*(z)$ est égal à t lorsque une des $\eta^1, \dots, \eta^\lambda$ est égal à t et lorsque $|z + t\xi_z^*| \leq |z + \eta^i\xi^i|$ pour tout indice $i \neq \star$ où \star correspond à l'indice tel que $\eta^*(z) = t$ et s'obtient de la même façon qu'au Lemme 2.

Pour montrer (5.4.33), considérons dans un premier temps que z est positif. Soit $t \in \mathbb{R}^+$, $u \in \mathbb{R}$ et $\epsilon > 0$. Alors

$$\begin{aligned} \mathbb{P}(|z + tu| \leq |z + \chi|) &= \mathbb{P}((|z + tu| \leq |z + \chi|) \cap (z + \chi \geq 0)) \\ &\quad + \mathbb{P}((|z + tu| \leq |z + \chi|) \cap (z + \chi \leq 0)) \end{aligned} \quad (5.4.35)$$

Le second terme du membre de droite de l'équation (5.4.35) est borné par $P(z + \chi \leq 0)$ qui est tel que $\lim_{z \rightarrow +\infty} P(z + \chi \leq 0) = 0$. On en déduit que

$$\lim_{z \rightarrow +\infty} \mathbb{P}((|z + tu| \leq |z + \chi|) \cap (z + \chi \leq 0)) = 0 \quad (5.4.36)$$

De plus le premier terme du membre de droite de l'équations (5.4.35) peut être réécrit de la façon suivante

$$\mathbb{P}((-z - \chi \leq z + tu \leq z + \chi) \cap (z + \chi \geq 0)),$$

qui est la probabilité de l'intersection des trois évènements:

$$\{-z - \chi \leq z + tu\} \cap \{z + tu \leq z + \eta' \xi'\} \cap \{z + \chi \geq 0\}$$

Cette intersection est incluse dans le deuxième évènement qui s'écrit aussi $\{tu \leq \eta' \xi'\}$. De plus quand $z \rightarrow +\infty$, cette intersection de trois évènements tend vers $\{tu \leq \eta' \xi'\}$, on en déduit que

$$\lim_{z \rightarrow \infty} \mathbb{P}((-z - \chi \leq z + tu \leq z + \chi) \cap (z + \chi \geq 0)) = \mathbb{P}(tu \leq \chi). \quad (5.4.37)$$

En utilisant (5.4.36) et (5.4.37) dans (5.4.35) on obtient

$$\lim_{z \rightarrow \infty} \mathbb{P}(|z + tu| \leq |z + \chi|) = \mathbb{P}(tu \leq \chi). \quad (5.4.38)$$

En utilisant le Théorème de convergence dominée pour passer à la limite lorsque $z \rightarrow +\infty$ dans l'intégrale de (5.4.32), nous en déduisons

$$\lim_{z \rightarrow +\infty} p_{\eta^*(z)}(t) = p_{\hat{\eta}(\lambda)}(t)$$

Le cas $z < 0$ étant symétrique, on en déduit (5.4.33).

Pour démontrer (5.4.34), on utilise la majoration suivante vraie pour tout $z \in \mathbb{R}$

$$p_{\eta^*(z)}(t) \leq \lambda p_{\eta}(t)$$

qui est encore vraie pour $p_{\hat{\eta}(\lambda)}(t)$. Comme $\mathbb{E}(\frac{1}{\eta^\alpha}) < +\infty$, $\lambda p_{\eta}(t)/t^\alpha$ est intégrable. De plus, on a la convergence presque partout de $p_{\eta^*(z)}(t)$ vers $p_{\hat{\eta}(\lambda)}(t)$ lorsque $|z|$ tend vers $+\infty$. Le Théorème de convergence dominée nous donne alors

$$\lim_{|z| \rightarrow +\infty} \mathbb{E}\left(\left(\frac{1}{\eta^*(z)}\right)^\alpha\right) = \mathbb{E}\left(\left(\frac{1}{\hat{\eta}(\lambda)}\right)^\alpha\right).$$

□

Rappelons la définition de Γ_0

$$\Gamma_0 = \{\gamma > 0 \text{ tel que } \mathbb{E}\left(\left|\frac{1}{\eta}\right|^\gamma\right) < +\infty \text{ et } \mathbb{E}(|\xi|^\gamma) < +\infty\}. \quad (5.4.39)$$

et de Λ

$$\Lambda = \{\lambda \text{ tel que } \exists \alpha \in \Gamma_0 \text{ tel que } \mathbb{E}\left(\frac{1}{\hat{\eta}(\lambda)^\alpha}\right) < 1\}, \quad (5.4.40)$$

Une définition équivalente pour Λ est $\Lambda = \bigcup_{\alpha \in \Gamma_0} \Lambda_\alpha$ avec

$$\Lambda_\alpha = \left\{ \lambda \in \mathbb{N} \text{ tels que } \mathbb{E}\left(\left(\frac{1}{\hat{\eta}(\lambda)}\right)^\alpha\right) < 1 \right\}$$

Théorème 6. *Soient η et ξ satisfaisant les hypothèses **H.1** et **H.2**. Alors pour tout entier $\lambda \in \Lambda$, la fonction $V(z) = |z|^\alpha + 1$ avec $\alpha \in \Gamma_0$ tel que $\mathbb{E}\left(\frac{1}{\hat{\eta}(\lambda)^\alpha}\right) < 1$ est une fonction de drift au sens de l'équation (5.1.8) pour $(Z_n)_{n \geq 0}$. Plus précisément on a*

$$\limsup_{z \rightarrow \pm\infty} \frac{\Delta V(z)}{V(z)} < 0.$$

PREUVE: Soit $\lambda \in \Lambda$, il existe $\alpha \in \Gamma_0$ tel que $\mathbb{E}\left(\frac{1}{\eta(\lambda)^\alpha}\right) < 1$. Considérons la fonction $V(z) = |z|^\alpha + 1$. Comme $\alpha \in \Gamma_0$, $\mathbb{E}(|\xi|^\alpha) < \infty$ et $\mathbb{E}\left(\left|\frac{1}{\eta}\right|^\alpha\right) < \infty$. On sait alors d'après l'inégalité de Lyapunov ([24]) que pour tout $\hat{\alpha} \leq \alpha$, $\mathbb{E}(|\xi|^{\hat{\alpha}}) < \infty$ et $\mathbb{E}\left(\left|\frac{1}{\eta}\right|^{\hat{\alpha}}\right) < \infty$. On a aussi $\mathbb{E}((|\xi|^{\lambda:\lambda})^\alpha) < \infty$ et $\mathbb{E}\left(\left(\frac{1}{\eta}\right)^{\lambda:\lambda}\right)^\alpha < \infty$, où $Y^{\lambda:\lambda}$ désigne la λ^{ieme} λ -statistique d'ordre de la variable aléatoire Y . Ainsi nous obtenons

$$\forall \hat{\alpha} \leq \alpha, \mathbb{E}(|\xi^*(z)|^{\hat{\alpha}}) \leq \mathbb{E}((|\xi|^{\lambda:\lambda})^{\hat{\alpha}}) < \infty \quad (5.4.41)$$

et

$$\forall \hat{\alpha} \leq \alpha, \mathbb{E}\left(\left|\frac{1}{\eta^*(z)}\right|^{\hat{\alpha}}\right) \leq \mathbb{E}\left(\left(\frac{1}{\eta}\right)^{\lambda:\lambda}\right)^{\hat{\alpha}} < \infty \quad (5.4.42)$$

L'opérateur de drift peut alors être réécrit de la façon suivante

$$\Delta V(z) = \mathbb{E}\left(\left|\frac{z}{\eta^*(z)} + \xi^*(z)\right|^\alpha\right) - |z|^\alpha.$$

Si $0 < \alpha \leq 1$, nous obtenons que $\Delta V(z) \leq \mathbb{E}\left(\left|\frac{z}{\eta^*(z)}\right|^\alpha\right) + |\xi^*(z)|^\alpha - |z|^\alpha$ ou

$$\Delta V(z) \leq |z|^\alpha \left(\mathbb{E}\left(\frac{1}{\eta^*(z)^\alpha}\right) - 1\right) + \mathbb{E}(|\xi^*(z)|^\alpha) \quad (5.4.43)$$

Si $\alpha \geq 1$, on définit $\beta > 0$ tel que, $\alpha = [\alpha] + \beta$, où $[\alpha]$ désigne la partie entière de α . D'après l'inégalité triangulaire on a

$$\left(\left|\frac{z}{\eta^*(z)} + \xi^*(z)\right|\right)^\alpha \leq \left(\left|\frac{z}{\eta^*(z)}\right| + |\xi^*(z)|\right)^\alpha$$

En appliquant la formule de Taylor Lagrange au membre de droite de l'égalité précédente, nous obtenons

$$\begin{aligned} \left|\frac{z}{\eta^*(z)} + \xi^*(z)\right|^\alpha &\leq \left|\frac{z}{\eta^*(z)}\right|^\alpha + \alpha |\xi^*(z)| \left|\frac{z}{\eta^*(z)}\right|^{\alpha-1} + \dots \\ &\dots + \frac{\alpha(\alpha-1)\dots(\alpha-[\alpha]+1)}{[\alpha]!} (|\xi^*(z)|)^{[\alpha]} |Y|^\beta, \end{aligned}$$

où Y est une variable aléatoire telle que $Y \in [|\frac{z}{\eta^*(z)}|, |\frac{z}{\eta^*(z)}| + |\xi^*(z)|]$. Comme on a supposé que $\mathbb{E}(|\xi|^\alpha) < \infty$, on a $\mathbb{E}(|\xi|^*(z)) < +\infty$ (cf. (5.4.41)) ce qui justifie que l'on peut passer à l'espérance dans la formule de Taylor précédente. En majorant Y par $|\frac{z}{\eta^*(z)}| + |\xi^*(z)|$ nous obtenons

$$\begin{aligned} \Delta V(z) \leq & |z|^\alpha \left(\mathbb{E}\left(\frac{1}{\eta^*(z)^\alpha}\right) - 1 \right) + |z|^{\alpha-1} \left(\alpha \mathbb{E}\left(\frac{|\xi^*(z)|}{(\eta^*(z))^{\alpha-1}}\right) \right) + \dots \\ & \dots + \frac{\alpha \dots (\alpha - [\alpha] + 1)}{[\alpha]!} \mathbb{E} \left(|\xi^*(z)|^{[\alpha]} \left(\left| \frac{z}{\eta^*(z)} \right|^\beta + |\xi^*(z)|^\beta \right) \right) \end{aligned} \quad (5.4.44)$$

Intéressons nous au membre de droite de l'inégalité précédente. Tous les coefficients (dépendant de z) en facteur devant les $|z|^{\alpha-p}$ pour $1 \leq p \leq [\alpha]$ sont bornés par des constantes ne dépendant pas de z . En effet prenons par exemple le terme en facteur devant $|z|^{\alpha-1}$, c'est-à-dire le terme $(\alpha \frac{|\xi^*(z)|}{(\eta^*(z))^{\alpha-1}})$, alors

$$\frac{|\xi^*(z)|}{(\eta^*(z))^{\alpha-1}} \leq |\xi|^{\lambda:\lambda} \left(\left(\frac{1}{\eta} \right)^{1:\lambda} \right)^{\alpha-1} \quad (5.4.45)$$

où $|\xi|^{\lambda:\lambda}$ désigne la λ^{ieme} statistique d'ordre de $|\xi|$ et $(\frac{1}{\eta})^{1:\lambda}$ la première statistique d'ordre de $\frac{1}{\eta}$. Comme $|\xi|^{\lambda:\lambda}$ et $(\frac{1}{\eta})^{1:\lambda}$ sont indépendantes, en prenant l'espérance dans l'équation (5.4.45), on obtient que $\mathbb{E}\left(\frac{|\xi^*(z)|}{(\eta^*(z))^{\alpha-1}}\right)$ est borné par la constante suivante, indépendante de z :

$$\mathbb{E}\left(\frac{|\xi^*(z)|}{(\eta^*(z))^{\alpha-1}}\right) \leq \mathbb{E}(|\xi|^{\lambda:\lambda}) \mathbb{E}\left(\left(\frac{1}{\eta}\right)^{1:\lambda}\right)^{\alpha-1} \quad (5.4.46)$$

Par ailleurs pour le terme dominant du membre de droite de l'équation (5.4.44), on sait d'après le Lemme 3, que

$$\lim_{|z| \rightarrow +\infty} \mathbb{E}\left(\frac{1}{\eta^*(z)^\alpha}\right) - 1 = \mathbb{E}\left(\left(\frac{1}{\hat{\eta}(\lambda)}\right)^\alpha\right) - 1$$

En divisant l'inégalité (5.4.44) par V et en prenant la limsup on en déduit que

$$\lim_{|z| \rightarrow \infty} \sup \frac{\Delta V(z)}{V(z)} = \mathbb{E}\left(\left(\frac{1}{\hat{\eta}(\lambda)}\right)^\alpha\right) - 1$$

En choisissant maintenant $\lambda \in \Lambda_\alpha$ nous obtenons que

$$\lim_{|z| \rightarrow \infty} \sup \frac{\Delta V(z)}{V(z)} < 0.$$

Comme les compacts de \mathbb{R} sont les small set pour $(Z_n)_{n \geq 0}$ on en déduit que V est bien une condition de drift au sens de l'équation (5.1.8).

□

5.4.1 Sur l'ensemble Λ_α

Il n'est pas évident dans un cas général de caractériser l'ensemble Λ , bien qu'il soit facile de simuler $\mathbb{E}(\frac{1}{\hat{\eta}(\lambda)^\alpha})$ et d'avoir une idée de l'ensemble Λ_α , comme nous l'illustrerons à la fin de ce paragraphe. Cependant dans le cas où les variables aléatoires η et ξ sont à support borné, il est facile de voir qu'il existe un λ_0 tel que

$$\{\lambda_0, \dots, \infty\} \subset \Lambda_\alpha.$$

C'est ce que nous allons démontrer à la Proposition 1.

5.4.1.1 Cas borné

Proposition 1. *Supposons que $\text{supp } p_\eta = [1 - \gamma, 1 + \beta]$ avec $0 < \gamma < 1$ et $\beta > 0$, supposons que $\text{supp } p_\xi = [-A, A]$ avec $A > 0$ alors*

$$\lim_{\lambda \rightarrow \infty} \mathbb{E}((\frac{1}{\hat{\eta}(\lambda)})^\alpha) = (\frac{1}{1 + \beta})^\alpha \quad (5.4.47)$$

PREUVE: Nous allons tout d'abord montrer que $\frac{1}{\hat{\eta}(\lambda)}$ converge en probabilité vers $\frac{1}{\beta+1}$ et nous déduirons ensuite la convergence L^1 de l'uniforme intégrabilité de la suite $(\frac{1}{\hat{\eta}(\lambda)}, \lambda \geq 1)$. Pour montrer la convergence en probabilité de $\frac{1}{\hat{\eta}(\lambda)}$, montrons tout d'abord la convergence en probabilité de $\chi^{1:\lambda}$ vers $-A(1 + \beta)$. D'après les hypothèses faites sur η et ξ le support de χ est égal à $[-A(1 + \beta), A(1 + \beta)]$. Pour $x \in \mathbb{R}$, on a

$$P(\chi^{1:\lambda} > x) = P(\chi > x)^\lambda,$$

et donc pour tout $\epsilon > 0$, on a $P(\chi^{1:\lambda} > -A(1 + \beta) + \epsilon) = P(\chi > -A(1 + \beta) + \epsilon)^\lambda$ qui tend vers 0 car $P(\chi > -A(1 + \beta) + \epsilon) < 1$. On en déduit que la suite $(\chi^{1:\lambda}, \lambda \geq 1)$ converge en probabilité vers $-A(1 + \beta)$.

Soit $\epsilon > 0$, il existe $\epsilon_1 > 0$ tel que l'évènement $\{\frac{1}{\hat{\eta}(\lambda)} > \frac{1}{1+\beta} + \epsilon\}$ est inclus dans l'évènement $\{\hat{\eta}(\lambda) < 1 + \beta - \epsilon_1\}$. De même il existe $\epsilon_2 > 0$ tel que ce dernier évènement est inclus dans l'évènement $\{\chi^{1:\lambda} > -A(1 + \beta) + \epsilon_2\}$. Comme nous venons de voir que $(\chi^{1:\lambda})$ converge en probabilité vers $-A(1 + \beta)$, on en déduit que $(\frac{1}{\hat{\eta}(\lambda)}, \lambda \geq 1)$ converge en probabilité vers $\frac{1}{1+\beta}$. On en déduit que $((\frac{1}{\hat{\eta}(\lambda)})^\alpha, \lambda \geq 1)$ converge en probabilité vers $(\frac{1}{1+\beta})^\alpha$. De plus comme pour tout λ , $1/(1 + \beta)^\alpha \leq (\frac{1}{\hat{\eta}(\lambda)})^\alpha \leq 1/(1 - \gamma)^\alpha$

$$\mathbb{E}((\frac{1}{\hat{\eta}(\lambda)})^\alpha) \leq (\frac{1}{1 - \gamma})^\alpha$$

la suite $((\frac{1}{\hat{\eta}(\lambda)})^\alpha, \lambda \geq 1)$ est uniformément intégrable. On en déduit donc la convergence L^1 , c'est à dire l'équation (5.4.47).

□

Un corollaire immédiat est que

Corollaire 1. *Sous les Hypothèses de la Proposition 1, il existe $\lambda_0 \geq 0$ tel que*

$$\{\lambda_0, \dots, \infty\} \subset \Lambda_\alpha.$$

5.4.1.2 Cas utilisé en pratique : simulations numériques

Dans le cas où η suit une loi log-normale et ξ suit une loi normale, nous simulons pour α et λ donnés, l'espérance $\mathbb{E}((\frac{1}{\hat{\eta}(\lambda)})^\alpha)$ par une méthode de Monte Carlo (cf chapitre précédent Section 4.7 pour plus de détails sur la méthode). Pour les simulations effectuées nous avons utilisé 10 000 tirages et calculé l'intervalle de confiance à 95%. Dès que l'on trouve α et λ tels que la borne supérieure de l'intervalle de confiance est au dessous de 1 nous en déduisons que $\lambda \in \Lambda_\alpha$. Pour trois valeurs du paramètre τ de la loi log normale, respectivement 0.5, 1 et 2 nous avons testés pour plusieurs valeurs de α (entre 1 et 10^{-4}), 20 valeurs de λ entre 2 et 100. Nous n'avons trouvé aucune valeur de α telle que la borne supérieure de l'intervalle de confiance soit au dessous de 1 pour $\lambda = 2$. En revanche pour $\alpha = 0.1$, on a trouvé que dès que λ est supérieur à 3, $\mathbb{E}((\frac{1}{\hat{\eta}(\lambda)})^\alpha) < 1$. Nous avons par ailleurs observé la monotonie en λ de l'espérance. Tout ceci laisse conjecturer que pour les trois valeurs de τ testées

$$\Lambda = \{3, \dots, \infty\}$$

5.5 Convergence du $(1, \lambda)$ —SA-ES

5.5.1 Positivité, récurrence au sens de Harris et Ergodicité géométrique pour

$$(Z_n)_{n \geq 0}$$

Du Théorème 6 on déduit immédiatement le Théorème suivant :

Théorème 7. *Sous les hypothèses du Théorème 6, on a que pour tout $\lambda \in \Lambda_\alpha$, $(Z_n)_{n \geq 0}$ est positive, récurrente au sens de Harris et géométriquement ergodique.*

PREUVE: Du Théorème 6, on déduit que $V(z) = |z|^\alpha + 1$ est une condition de drift au sens de l'équation (5.1.8) et donc aussi au sens des équations (5.1.6) et (5.1.5), on en déduit alors que $(Z_n)_{n \geq 0}$ est récurrente au sens de Harris, positive et géométriquement ergodique.

□

5.5.2 Propriétés de la mesure invariante de $(Z_n)_{n \geq 0}$

5.5.2.1 Equivalence entre μ et μ^{Leb}

Une conséquence immédiate de la μ^{Leb} -irréductibilité et de la récurrence de $(Z_n)_{n \geq 0}$ est que la mesure de probabilité invariante de $(Z_n)_{n \geq 0}$, μ , est équivalente à la mesure de Lebesgue (cf. [100, Théorème 10.4.9]). Ainsi, sous les hypothèses de Théorème 6, μ est équivalente à μ^{Leb} . Comme μ est absolument continue par rapport à la mesure de Lebesgue, on sait d'après le Théorème de Radon-Nykodym qu'elle admet une densité par rapport à la mesure de Lebesgue. De plus, le support de cette densité est \mathbb{R} puisque la mesure de Lebesgue est absolument continue par rapport à μ . En notant f_μ cette densité, la mesure μ s'écrit maintenant, pour tout $A \in \mathcal{B}(\mathbb{R})$

$$\mu(A) = \int_{\mathbb{R}} \mathbb{1}_A(z) f_\mu(z) dz.$$

Par ailleurs, la proposition suivante va nous donner une représentation de la densité f_μ en fonction de μ et de p , la densité du noyau de transition donné par l'équation (5.3.27).

Proposition 2.

$$f_\mu(y) = \int_{\mathbb{R}} p(z, y) \mu(dz) \quad \mu^{Leb} \text{ p.p.}$$

PREUVE: La preuve se trouve dans le livre de M. Duflo [53]. Nous reproduisons cependant ici la preuve qui s'avère être relativement simple.

Soit $B \in \mathcal{B}(\mathbb{R})$,

$$\begin{aligned} \int_B f_\mu(y) dy &= \mu(B) \\ &= \int_{\mathbb{R}} \mu(dz) P(z, B) \\ &= \int_{\mathbb{R}} f_\mu(z) \int_B p(z, y) dy dz \\ &= \int_B \int_{\mathbb{R}} f_\mu(z) p(z, y) dz dy \end{aligned}$$

Comme ceci est vrai quel que soit $B \in \mathcal{B}(\mathbb{R})$, on en déduit que

$$f_\mu(y) = \int_{\mathbb{R}} p(z, y) \mu(dz) \quad \mu^{Leb} \text{ p.p.}$$

□

On a vu à la Remarque 6 que si $p_\xi \in L^\infty(\mathbb{R})$ alors $p \in L^\infty(\mathbb{R} \times \mathbb{R})$, on en déduit en utilisant la représentation du Lemme précédent le corollaire suivant

Corollaire 2. *Si $p_\xi \in L^\infty(\mathbb{R})$, $f_\mu \in L^\infty(\mathbb{R})$.*

5.5.2.2 Moments de la mesure invariante

Proposition 3. *Supposons que $V(z) = |z|^\alpha + 1$ soit une fonction de drift pour $(Z_n)_{n \geq 0}$ au sens de l'équation (5.1.8). Alors*

$$\int_{\mathbb{R}} |z|^\alpha \mu(dz) < +\infty.$$

PREUVE: Soit $V(z) = |z|^\alpha + 1$, une fonction de drift au sens de l'équation (5.1.8). D'après l'équation (5.1.7) du Théorème M.4, on sait qu'il existe $\rho < 1$ et $R < +\infty$ tels que pour tout $z \in \mathbb{R}$, pour tout $n \in \mathbb{N}$

$$\|P^n(z, \cdot) - \mu\|_V \leq RV(z)\rho^n. \quad (5.5.48)$$

D'après l'inégalité triangulaire, pour tout $n \in \mathbb{N}$, pour tout $z \in \mathbb{R}$ on a

$$\|\mu\|_V \leq \|P^n(z, \cdot) - \mu\|_V + \|P^n(z, \cdot)\|_V \quad (5.5.49)$$

Or

$$\|P^n(z, \cdot)\|_V = \sup_{|g| \leq V} \left| \int P^n(z, dy) g(y) \right| \quad (5.5.50)$$

$$\leq \sup_{|g| \leq V} \int |P^n(z, dy) g(y)| \quad (5.5.51)$$

$$\leq \int P^n(z, dy) V(y) \quad (5.5.52)$$

Or pour $n = 0$ on a $\int P^n(z, dy) V(y) = V(z)$, ainsi le terme de droite du membre de droite de l'équation (5.5.49) pour $n = 0$ est fini pour tout $z \in \mathbb{R}$. Par ailleurs le terme de gauche de ce second membre pour $n = 0$ est aussi fini grâce à l'égalité (5.5.48). Ainsi $\|\mu\|_V < +\infty$.

Comme de plus

$$\int_{\mathbb{R}} |z|^\alpha \mu(dz) = \int_{\mathbb{R}} V(z) \mu(dz) - 1 = \|\mu\|_V + 1,$$

on en déduit que $\int_{\mathbb{R}} |z|^\alpha \mu(dz) < +\infty$.

□

5.5.2.3 Intégrabilité de $\ln(|z|)$ contre la mesure invariante

Proposition 4. *Si $p_\xi \in L^\infty(\mathbb{R})$ et si il existe $\alpha > 0$ tel que $|z|^\alpha$ soit intégrable contre μ alors*

$$\int_{\mathbb{R}} |\ln(|z|)| \mu(dz) < +\infty.$$

PREUVE: Si $p_\xi \in L^\infty(\mathbb{R})$, on a vu au Corollaire 2 que $f_\mu \in L^\infty(\mathbb{R})$. Ainsi $\ln(|z|)$ est intégrable sur tout borné de la forme $[-A, A]$ puisque $\ln(|z|)$ est intégrable sur $[-A, A]$ contre la mesure de Lebesgue. Par ailleurs en l'infini, $|\ln(|z|)|$ est dominé par $|z|^\alpha$ qui est intégrable contre μ donc $\ln(|z|)$ est intégrable sur \mathbb{R} . □

5.5.3 Convergence p.s. de $\frac{1}{n} \ln(\|X_n\|)$ et $\frac{1}{n} \ln(\sigma_n)$

Rappelons que nous avons défini Γ_0 par

$$\Gamma_0 = \{\gamma > 0 \text{ tel que } \mathbb{E}(|\frac{1}{\eta}|^\gamma) < +\infty \text{ et } \mathbb{E}(|\xi|^\gamma) < +\infty\}.$$

et pour $\alpha \in \Gamma_0$, nous avons défini les sous ensembles de \mathbb{N} suivant:

$$\Lambda_\alpha = \left\{ \lambda \in \mathbb{N} \text{ tels que } \mathbb{E}\left(\left(\frac{1}{\hat{\eta}(\lambda)}\right)^\alpha\right) < 1 \right\}$$

Théorème 8. *Soient η et ξ satisfaisant les hypothèses **H.1** et **H.2** et telles que Γ_0 soit non vide. Supposons de plus que $\mathbb{E}(|\ln(\eta)|) < +\infty$. Soit Λ le sous ensemble de \mathbb{N} suivant*

$$\Lambda = \bigcup_{\alpha \in \Gamma_0} \Lambda_\alpha \tag{5.5.53}$$

Alors pour tout $\lambda \in \Lambda$ la chaîne $(Z_n)_{n \geq 0}$ vérifie les hypothèses du Théorème 1 et l'on a donc

$$\frac{1}{n} \ln(\|X_n\|) \xrightarrow[n \rightarrow \infty]{p.s.} \int_{\mathbb{R}} \mathbb{E}[\ln \left\| \frac{z + \eta^*(z)\xi^*(z)}{z} \right\|] d\mu(z). \tag{5.5.54}$$

PREUVE: Soit $\lambda \in \Lambda$, alors il existe $\alpha > 0$ tel que $\mathbb{E}(1/\hat{\eta}(\lambda)^\alpha) < 1$. D'après le Théorème 6, la fonction $V(z) = |z|^\alpha + 1$ est une fonction de drift pour $(Z_n)_{n \geq 0}$ qui est en conséquence positive et récurrente au sens de Harris d'après le Théorème 7. Pour voir que le membre de droite de (5.5.54) est fini, on écrit

$$\begin{aligned} & \int_{\mathbb{R}} \mathbb{E}(|\ln \|1 + \frac{\eta^*(z)\xi^*(z)}{z}\||) d\mu(z) \\ &= \int_{\mathbb{R}} \mathbb{E}(|\ln \left\| \frac{z + \eta^*(z)\xi^*(z)}{\eta^*(z)} \right\| + \ln(\eta^*(z)) - \ln(\|z\|)|) d\mu(z) \\ &\leq \int_{\mathbb{R}} \mathbb{E}(|\ln \left\| \frac{z + \eta^*(z)\xi^*(z)}{\eta^*(z)} \right\|) d\mu(z) + \int_{\mathbb{R}} \mathbb{E}(|\ln \eta^*(z)|) d\mu(z) + \int_{\mathbb{R}} \mathbb{E}(|\ln \|z\||) d\mu(z) \end{aligned}$$

Le dernier terme $\int_{\mathbb{R}} \mathbb{E}(|\ln \|z\||) d\mu(z)$ est égal à $\int_{\mathbb{R}} |\ln \|z\|| d\mu(z)$ qui est intégrable d'après la Proposition 4.

Le premier terme s'écrit

$$\int_{\mathbb{R}} \mathbb{E}(|\ln \|\frac{z + \eta^*(z)\xi^*(z)}{\eta^*(z)}\||) d\mu(z) = \int_{\mathbb{R}} \int_{\mathbb{R}} |\ln \|y\|| P(z, dy) d\mu(z)$$

et comme $\int_{\mathbb{R}} |\ln \|z\|| d\mu(z)$ est intégrable, en appliquant le Théorème de Fubini et en utilisant le fait que μ est invariante

$$\int_{\mathbb{R}} \int_{\mathbb{R}} |\ln \|z\|| P(z, dy) d\mu(z) = \int_{\mathbb{R}} |\ln \|y\|| d\mu(y) < +\infty.$$

De plus, on déduit de l'hypothèse $\mathbb{E}(\ln(\eta)) < +\infty$, que $\ln(\eta^*(z))$ est intégrable comme on peut le voir par exemple avec la majoration suivante,

$$|\ln(\eta^*(z))| \leq \max(|\ln(\eta^{1:\lambda})|, |\ln(\eta^{\lambda:\lambda})|) \quad (5.5.55)$$

où $\eta^{1:\lambda}$ et $\eta^{\lambda:\lambda}$ représentent respectivement la première et la dernière statistique d'ordre parmi les λ variables aléatoires à partir desquelles ont été choisi $\eta^*(z)$. Ainsi le membre de droite de (5.5.54) est fini. Toutes les hypothèses du Théorème 1 sont donc vérifiées.

□

5.5.3.1 Autre expression du taux de convergence

Nous allons déterminer ici une autre expression du taux de convergence donné par

$$\int_{\mathbb{R}} E[\ln \|\frac{z + \eta^*(z)\xi^*(z)}{z}\||] d\mu(z)$$

Proposition 5. *Sous les hypothèses du Théorème 8*

$$\int_{\mathbb{R}} E[\ln \|\frac{z + \eta^*(z)\xi^*(z)}{z}\||] d\mu(z) = \int_{\mathbb{R}} E[\ln(\eta^*(z))] d\mu(z) \quad (5.5.56)$$

PREUVE: Réécrivons le terme de gauche de (5.5.56)

$$\begin{aligned} E[\ln(\|1 + \frac{\eta^*(z)\xi^*(z)}{z}\|)] d\mu(z) \\ = \int_{\mathbb{R}} E[\ln(\|\frac{z + \eta^*(z)\xi^*(z)}{\eta^*(z)}\|) + \ln(\eta^*(z)) - \ln(\|z\|)] d\mu(z) \end{aligned}$$

En appliquant comme dans la démonstration du Théorème 8 le Théorème de Fubini on a

$$\int_{\mathbb{R}} E[\ln(\|1 + \frac{\eta^*(z)\xi^*(z)}{z}\|)] d\mu(z) = \int_{\mathbb{R}} \ln(\|z\|) d\mu(z).$$

On en déduit l'équation (5.5.56).

□

5.5.3.2 Convergence de $\frac{1}{n} \ln(\sigma_n)$

Théorème 9. *Sous les hypothèses du Théorème 8*

$$\frac{1}{n} \ln(\sigma_n) \xrightarrow[n \rightarrow \infty]{p.s.} \int_{\mathbb{R}} \mathbb{E}[\ln(\eta^*(z))] d\mu(z) \quad (5.5.57)$$

PREUVE: On part de $\sigma_{n+1} = \sigma_n \eta^*(Z_n)$ (on a en effet vu dans la démonstration du Lemme 1 que $\eta^*(X_n/\sigma_n) = \eta^*(X_n, \sigma_n)$ au sens presque sûr), on déduit que

$$\frac{1}{n} \ln(\sigma_n) = \frac{1}{n} \ln(\sigma_0) + \frac{1}{n} \sum_{k=0}^{n-1} \ln(\eta^*(Z_k)) \quad (5.5.58)$$

Comme $(Z_n)_{n \geq 0}$ est positive et récurrente au sens de Harris on en déduit de manière analogue que dans le Théorème 1 que

$$\frac{1}{n} \ln(\sigma_n) \xrightarrow[n \rightarrow \infty]{p.s.} \int_{\mathbb{R}} E[\ln(\eta^*(z))] d\mu(z).$$

□

5.5.4 Théorème Central Limite pour $(Z_n)_{n \geq 0}$

Théorème 10. *Sous les hypothèses du Théorème 8*

$$\sqrt{n} \left(\frac{1}{n} \sum_{k=0}^{n-1} \ln(\eta^*(Z_k)) - \int_{\mathbb{R}} \mathbb{E}[\ln(\eta^*(z))] d\mu(z) \right) \xrightarrow[n \rightarrow \infty]{\mathcal{L}} \mathcal{N}(0, \gamma^2) \quad (5.5.59)$$

où γ est donné par

$$\gamma^2 = \lim_{n \rightarrow \infty} \frac{1}{n} \mathbb{E}_{\mu} \left[\left(\sum_{k=1}^n (\ln(|\eta^*(Z_k)|)) - \int_0^{\infty} \mathbb{E}[\ln(\eta^*(z))] d\mu(z) \right)^2 \right] \quad (5.5.60)$$

PREUVE: Pour la preuve de ce Théorème nous considérons la chaîne de Markov $\Phi_n = (Z_n, \eta_n^1, \dots, \eta_n^\lambda, \xi_n^1, \dots, \xi_n^\lambda)$. Le Lemme 4 ci-dessous nous dit que Φ_n est géométriquement ergodique dès que $(Z_n)_{n \geq 0}$ l'est et plus précisément nous dit que Φ_n satisfait la condition de Drift donnée par l'équation (5.1.8) pour tout V donné par (5.5.61). Or,

$$(\ln(\eta^*(z)))^2 \leq (\ln(\eta^1 \wedge \dots \wedge \eta^\lambda))^2 + (\ln(\eta^1 \vee \dots \vee \eta^\lambda))^2 + 1$$

En prenant $V_2 = (\ln(\eta^1 \wedge \dots \wedge \eta^\lambda))^2 + (\ln(\eta^1 \vee \dots \vee \eta^\lambda))^2 + 1$, nous obtenons

$$\left(\ln|\eta^*(z)| \right)^2 \leq V + V_2$$

pour tout V tel que $(Z_n)_{n \geq 0}$ est V -géométriquement ergodique. En utilisant alors le Théorème M.5 avec $g = \ln|\eta^*(z)|$ on obtient (5.5.59) et (5.5.60).

□

Pour compléter la preuve du Théorème 10, il ne reste plus qu'à démontrer le Lemme suivant dont la preuve est donnée en Annexe:

Lemme 4. *Soit Φ_n la chaîne de Markov $\Phi_n = (Z_n, \eta_n^1, \dots, \eta_n^\lambda, \xi_n^1, \dots, \xi_n^\lambda)$ définie sur l'espace d'état $(\mathbb{R}^+)^{\lambda+1} \times (\mathbb{R})^\lambda$. Supposons que $(Z_n)_{n \geq 0}$ est géométriquement ergodique et plus précisément satisfait pour une fonction V_1 la condition de drift donnée par l'équation 5.1.8. Alors Φ_n est $\mu^{Leb} \otimes_\lambda \nu_\eta \otimes_\lambda \nu_\xi$ -irréductible, apériodique et satisfait la condition de drift donnée par l'équation 5.1.8 pour*

$$V(z, \eta^1, \dots, \eta^\lambda, \xi^1, \dots, \xi^\lambda) = V_1(z) + V_2(\eta^1, \dots, \eta^\lambda, \xi^1, \dots, \xi^\lambda) \quad (5.5.61)$$

où V_2 est une fonction supérieure ou égale à 1.

5.5.5 Application : approximation du taux de convergence

Le reste de cette section illustre le précédent résultat en utilisant l'algorithme $(1, \lambda)$ -SA-ES standard, i.e. avec $\xi \sim N(0, 1)$ et $\eta \sim \exp(\tau N(0, 1))$ ($\tau > 0$). Nous avons vu à la section 5.4.1.2 que des estimations numériques de Λ laissent conjecturer que dans ce cas là $\Lambda = \{3, \dots, \infty\}$. En conséquence, l'expression du taux de convergence

$$\int_{\mathbb{R}} \mathbb{E}[\ln(\eta^*(z))] d\mu(z)$$

est valable pour tout λ supérieur ou égal à 3.

Cela ouvre la voie au genre d'analyses menées dans [131], où le but est de déterminer pour les Stratégies d'évolutions auto adaptatives quels sont les meilleurs paramètres à choisir (parmi le nombre d'enfants et les paramètres de mutation, ...). Ces techniques sont fondées sur la détermination des meilleurs paramètres maximisant le progress rate

$$E(f(X_{n+1})|X_n) - f(X_n)$$

pour $f(x) = \|x\|^2$. L'inconvénient majeur de cette technique est qu'elle ne prend pas en compte la dynamique de l'algorithme. Ainsi l'estimation du taux de convergence que nous proposons ici pourrait être une alternative à cette technique. Le but ici est d'illustrer cette alternative mais pas de chercher les meilleurs paramètres pour le $(1, \lambda)$ -SA-ES.

D'après le Théorème 10, une estimation naturelle pour le taux de convergence

$$\int_{\mathbb{R}} \mathbb{E}[\ln(\eta^*(z))] d\mu(z)$$

est

$$\frac{1}{n} \sum_{k=0}^{n-1} \ln(|\eta^*(Z_k)|) \quad (5.5.62)$$

et de plus, le Théorème 10 nous donne qu'un intervalle de confiance à 95% pour cette approximation est:

$$\frac{1}{n} \sum_{k=0}^{n-1} \ln(|\eta^*(Z_k)|) \pm 1.96 \frac{\gamma}{\sqrt{n}}$$

où la variance asymptotique γ^2 est donnée par l'équation (5.5.60). L'estimation de la variance est faite ici à l'aide d'une méthode de *batch-means*. Comme le but ici est simplement d'illustrer comment calculer le taux de convergence, nous n'avons pas passé de temps particulier pour tester différentes approches pour estimer la variance asymptotique mais nous référons à [67] pour une discussion sur diverses méthodes pour estimer précisément une telle variance asymptotique, tout comme nous référons à [124] pour des techniques plus récentes.

La méthode *batch-means* consiste à diviser un ensemble de n échantillons en m groupes, avec $N = \frac{n}{m}$ observations dans chaque groupe. Pour chaque groupe i , on calcule la moyenne suivante

$$\widehat{\mu}_N^i = \frac{1}{N} \sum_{k=(i-1)N+1}^{iN} \ln(|\eta^*(Z_k)|)$$

L'estimateur final est donné par l'équation 5.5.62 pour le taux de convergence

$$\widehat{\mu}_m = \frac{1}{m} \sum_{i=1}^m \widehat{\mu}_N^i.$$

Finalement la variance asymptotique est estimée à l'aide de

$$\gamma_m = \frac{1}{m-1} \sum_{i=1}^m (\widehat{\mu}_N^i - \widehat{\mu}_m)^2$$

et en conséquence l'approximation de l'intervalle à 95% est donné par

$$\widehat{\mu}_m \pm 1.96 \frac{\gamma_m}{\sqrt{m}}.$$

D'après [130] où il est dit que le nombre de groupe ne doit pas excéder, nous avons choisi ici $m = 30$. Chaque groupe contient 1000 points. La Figure 5.1 montre, pour $\tau = 1$ et $\tau = 10$, l'estimation de la moyenne $\widehat{\mu}_m$ tout comme l'intervalle de confiance approximé à 95% pour plusieurs valeurs de λ . Pour $\tau = 1$ on peut voir que avec un intervalle de confiance à 95% pour tout $\lambda \geq 3$, le taux de convergence est négatif, i.e. the $(1, \lambda)$ -SA-ES converge vers zero. La même assertion pour $\tau = 10$ peut seulement être faite pour $\lambda \geq 5$.

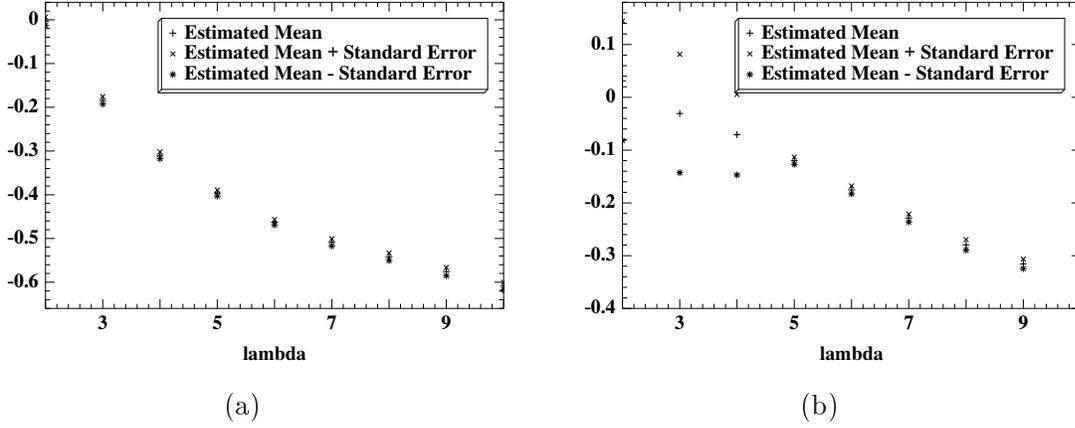


FIG. 5.1: $\widehat{\mu}_m$ avec l'intervalle de confiance à 95% pour $\tau = 1$ (a) et $\tau = 10$ (b). En abscisse le nombre d'enfants λ .

5.6 Discussion et conclusion

L'analyse que nous venons de présenter constitue une avancée pour l'analyse du phénomène d'auto-adaptation dans les ESs. Par ailleurs, cette étude que nous avons menée dans le cas particulier de l'algorithme $(1, \lambda)$ -SA-ES en dimension 1 peut être généralisée.

Nous mentionnons dans ce qui suit des généralisations plus ou moins immédiates de cette analyse. Ces généralisations concernent d'une part le même algorithme, sur la même fonction, mais pour une dimension plus grande que 1, et d'autre part d'autres algorithmes adaptatifs. En particulier, l'étude des méthodes dérandomisées présentées au Chapitre 1 sur la fonction sphère peut se faire de la même façon.

Généralisation à une dimension supérieure Tout d'abord si on considère la minimisation de la fonction sphère en dimension $d > 1$ par le $(1, \lambda)$ -SA-ES. L'algorithme à étudier est donc le suivant,

$$X_{n+1} = \arg \min \{ \|X_n + \sigma_n \eta_n^1 \xi_n^1\|, \dots, \|X_n + \sigma_n \eta_n^\lambda \xi_n^\lambda\| \} \quad (5.6.63)$$

$$\sigma_{n+1} = \sigma_n \eta_n^*, \star \text{ défini par } X_{n+1} = X_n + \sigma_n \eta_n^* \xi_n^* \quad (5.6.64)$$

où $\sigma_0 \in \mathbb{R}$, $X_0 \in \mathbb{R}^d$, ξ_n^i , pour $1 \leq i \leq \lambda$ sont λ vecteurs aléatoires indépendants de dimension d et η_n^i pour $1 \leq i \leq \lambda$ sont des variables aléatoires indépendantes réelles positives.

De manière analogue à la dimension 1, on définit la chaîne de Markov $Z_n = \frac{X_n}{\sigma_n}$, et comme

dans le cas de la dimension 1 nous avons l'égalité

$$\min_{1 \leq i \leq \lambda} \|X_n + \sigma_n \eta_n^i \xi_n^i\| = \sigma_n \min_{1 \leq i \leq \lambda} \left\| \frac{X_n}{\sigma_n} + \eta_n^i \xi_n^i \right\|. \quad (5.6.65)$$

qui nous permet de définir $(Z_n)_{n \geq 0}$ comme dans le Lemme 1. L'étude de la stabilité de cette chaîne et plus précisément de sa récurrence au sens de Harris et de sa positivité nous conduit comme dans le Théorème 1 à la convergence de $\frac{1}{n} \ln(\|X_n\|)$ vers $\int_{\mathbb{R}^d} E[\ln \left\| \frac{z + \eta^*(z) \xi^*(z)}{z} \right\|] d\mu(z)$. Notons que dans ce cas là les conditions de drift sont plus techniques à obtenir du fait de la dimension d .

Généralisation à d'autres algorithmes Nous pouvons de même généraliser la technique à une version élitiste de l'algorithme, à savoir le $(1 + \lambda)$ -SA-ES qui diffère de l'algorithme que nous venons d'étudier par le fait qu'à la génération n , le futur parent va être choisi parmi les λ enfants qui viennent d'être créés et le parent de la génération courante. Ainsi d'une génération à l'autre la fitness de la solution ne peut pas se dégrader. Bien que cet algorithme soit moins utilisé en pratique que la version non élitiste, parce qu'il reste plus facilement bloqué dans les minima locaux, son étude peut être réalisée de la même manière. Nous allons préciser ici les changements que cela induit. Considérons la chaîne $((X_n, \sigma_n), n \geq 1)$ de l'algorithme $(1 + \lambda)$ -SA-ES. Elle est définie par

$$X_{n+1} = \arg \min \{ \|X_n + \sigma_n \eta_n^1 \xi_n^1\|, \dots, \|X_n + \sigma_n \eta_n^\lambda \xi_n^\lambda\|, \|X_n\| \} \quad (5.6.66)$$

Nous allons alors définir les variables aléatoires η_n^* et ξ_n^* telles que X_{n+1} soit égal à $X_n + \sigma_n \eta_n^* \xi_n^*$ avec $\eta_n^* = 1$ et $\xi_n^* = 0$ si $X_{n+1} = X_n$. Notons alors que sous les hypothèses faites pour le cas non élitiste sur η et ν_ξ (en particulier si elles absolument continues par rapport à la mesure de Lebesgue) les variables aléatoire η_n^* et ξ_n^* possèdent une partie singulière. Considérons alors $Z_n = \frac{X_n}{\sigma_n}$, de manière analogue au Lemme 1, $(Z_n)_{n \geq 0}$ est une chaîne de Markov que l'on peut définir par (5.2.17) et (5.2.18). La différence principale pour l'étude des propriétés de stabilité de la chaîne $(Z_n)_{n \geq 0}$ est le fait que le noyau de transition possède une partie singulière.

Le CSA-ES Enfin les algorithmes dérandonisés présentés au chapitre 1 peuvent s'analyser de la même manière. Considérons l'exemple de l'algorithme CSA:

$$\begin{aligned} X_{n+1} &= \arg \min \{ |X_n + \sigma_n \xi_n^1|, \dots, |X_n + \sigma_n \xi_n^\lambda| \} \\ \sigma_{n+1} &= \sigma_n \exp(|\xi_n^*| - \mathbb{E}(|\xi|)), \star \text{ défini par } X_{n+1} = X_n + \sigma_n \xi_n^* \end{aligned}$$

L'étude de la convergence du CSA passe par l'étude de $Z_n = \frac{X_n}{\sigma_n}$ qui va être très similaire à l'étude du SA-ES. Notons que dans ce cas là, la chaîne de Markov sera récurrente au sens de Harris pour tout $\lambda \in \Lambda_{CSA}$ définit par:

$$\Lambda_{CSA} = \cup \left\{ \lambda \text{ tels que } \mathbb{E} \left(\exp(\mathbb{E}(|\xi|) - |\xi^{1:\lambda}|)^\alpha \right) < 1 \right\}$$

Ensemble qui peut s'avérer plus simple à étudier que pour l'algorithme SA-ES, en particulier il est facile de voir que si $\lambda_0 \in \Lambda_{CSA}$, alors $\{\lambda_0, \dots, \infty\} \subset \Lambda_{CSA}$.

Les généralisations possibles que nous venons d'évoquer montrent que la technique de preuve utilisée ici permet une base de comparaison de différents algorithmes adaptatifs et auto-adaptatifs sur la fonction sphère. Cependant, il est vrai que pour établir des comparaisons, il serait nécessaire d'étudier plus en détail la mesure stationnaire du processus Z_n .

Les choix que nous avons faits pour le cadre de l'analyse que nous venons de présenter ont été guidés par l'algorithme évolutionnaire utilisé en pratique. En particulier le cadre que nous avons choisi pour les variables aléatoires de la mutation englobe le cas de la mutation gaussienne pour ξ et le cas de la mutation log normale pour η . Cependant, considérer par exemple une loi singulière pour η permettrait par exemple une caractérisation plus simple de l'ensemble Λ .

5.7 Annexe

5.7.1 Preuves de la Section 5.2

Rappelons le Lemme 1 que nous allons démontrer:

Lemme Soit (X_n, σ_n) défini par les équations (5.2.15) et (5.2.16) $Z_n \stackrel{p.s.}{=} \frac{X_n}{\sigma_n}$. Alors $(Z_n)_{n \in \mathbb{N}}$ est une chaîne de Markov homogène telle que

$$Z_{n+1} = \frac{Z_n + \eta_n^*(Z_n) \xi_n^*(Z_n)}{\eta_n^*(Z_n)} \quad (5.7.67)$$

avec $Z_0 = \frac{X_0}{\sigma_0} \in \mathbb{R}$ et où $\eta_n^*(Z_n)$ et $\xi_n^*(Z_n)$ sont définies par

$$\|Z_n + \eta_n^*(Z_n) \xi_n^*(Z_n)\| = \min_{1 \leq i \leq \lambda} \|Z_n + \eta_n^i \xi_n^i\|. \quad (5.7.68)$$

où η_n^i et ξ_n^i sont les λ tirages indépendants qui ont engendré (X_n, σ_n) (cf équation (5.2.15) et (5.2.16)).

PREUVE: Considérons la chaîne de Markov (X_n, σ_n) définie par les équations (5.2.15) et (5.2.16). Commençons par l'égalité suivante

$$\min_{1 \leq i \leq \lambda} \|X_n + \sigma_n \eta_n^i \xi_n^i\| = \sigma_n \min_{1 \leq i \leq \lambda} \left\| \frac{X_n}{\sigma_n} + \eta_n^i \xi_n^i \right\|. \quad (5.7.69)$$

On en déduit que le couple que nous notons $(\eta_n^*(X_n, \sigma_n), \xi_n^*(X_n, \sigma_n))$ réalisant le minimum à gauche de (5.7.69) c'est-à-dire,

$$\|X_n + \sigma_n \eta_n^*(X_n, \sigma_n) \xi_n^*(X_n, \sigma_n)\| = \min_{1 \leq i \leq \lambda} \|X_n + \sigma_n \eta_n^i \xi_n^i\|$$

est le même que le couple, $(\eta_n^*(X_n/\sigma_n, 1), \xi_n^*(X_n/\sigma_n, 1))$ réalisant le minimum à droite de (5.7.69).

$$\|X_n + \eta_n^*(X_n/\sigma_n, 1) \xi_n^*(X_n/\sigma_n, 1)\| = \min_{1 \leq i \leq \lambda} \|X_n/\sigma_n + \eta_n^i \xi_n^i\|$$

Définissons $Z_n = \frac{X_n}{\sigma_n}$, alors

$$Z_{n+1} = \frac{X_{n+1}}{\sigma_{n+1}} = \frac{X_n + \sigma_n \eta_n^*(X_n, \sigma_n) \xi_n^*(X_n, \sigma_n)}{\sigma_n \eta_n^*(X_n, \sigma_n)}$$

ou encore

$$Z_{n+1} = \frac{Z_n + \eta_n^*(X_n, \sigma_n) \xi_n^*(X_n, \sigma_n)}{\eta_n^*(X_n, \sigma_n)} \quad (5.7.70)$$

où toutes les égalités précédentes sont au sens presque sûr. D'après ce que nous venons de voir, $(\eta_n^*(X_n/\sigma_n, 1), \xi_n^*(X_n/\sigma_n, 1)) = (\eta_n^*(Z_n), \xi_n^*(Z_n))$ presque sûrement, où ce dernier couple

est défini par l'équation (5.2.18). D'après (5.7.70), on voit que $(Z_n)_{n \geq 0}$ est une chaîne de Markov. Partant de $Z_0 = z$, on voit que

$$Z_1 = \frac{z + \eta_1^*(z)\xi_1^*(z)}{\eta_1^*(z)} \quad (5.7.71)$$

avec $\|z + \eta_1^*(z)\xi_1^*(z)\| = \min_{1 \leq i \leq \lambda} \|z + \eta_1^i \xi_1^i\|$

□

Théorème *Supposons que la chaîne de Markov $(Z_n)_{n \geq 0}$ est positive récurrente au sens de Harris de mesure de probabilité invariante μ et supposons que $\ln((\min_{1 \leq i \leq \lambda} \|z + \eta^i \xi^i\|)/\|z\|)$ est intégrable contre $\mu \otimes \nu_\eta \otimes \dots \otimes \nu_\xi$. Alors*

$$\frac{1}{n} \ln(\|X_n\|) \xrightarrow[n \rightarrow \infty]{p.s.} \int_{\mathbb{R}} E[\ln \|\frac{z + \eta^*(z)\xi^*(z)}{z}\|] d\mu(z). \quad (5.7.72)$$

PREUVE: A partir de l'équation (5.7.70) on a l'égalité suivante au sens presque sûr

$$\frac{1}{n} \ln(\|X_n\|) = \frac{1}{n} \ln(\|X_0\|) + \frac{1}{n} \sum_{k=0}^{n-1} \ln \left(\left\| \frac{Z_k + \eta_k^*(Z_k)\xi_k^*(Z_k)}{Z_k} \right\| \right) \quad (5.7.73)$$

Puisque $(Z_n)_{n \geq 0}$ est positive récurrente au sens de Harris, on sait que $(Z_n)_{n \geq 0}$ admet une unique mesure de probabilité invariante μ . Remarquons alors que si l'on considère la chaîne de Markov $\Phi_n = (Z_n, \eta_n^1, \dots, \eta_n^\lambda, \xi_n^1, \dots, \xi_n^\lambda)$ les propriétés de récurrence et de positivité de $(Z_n)_{n \geq 0}$ se transmettent à la chaîne Φ_n . En effet, le noyau de transition \mathbf{P} pour cette chaîne étant donné par

$$\mathbf{P} \left((z, \eta^1, \dots, \xi^\lambda), A \times A^1 \times \dots \times A^\lambda \times B^1 \times \dots \times B^\lambda \right) = P(z, A) \nu_\eta(A^1) \dots \nu_\eta(A^\lambda) \nu_\xi(B^1) \dots \nu_\xi(B^\lambda) \quad (5.7.74)$$

où P est le noyau de transition de la chaîne $(Z_n)_{n \geq 0}$, la récurrence au sens de Harris de $(Z_n)_{n \geq 0}$ implique celle de Φ_n , de même pour la positivité. On a de plus que la mesure de probabilité invariante est $\mu \otimes \nu_\eta \otimes \dots \otimes \nu_\xi$. D'après la loi forte des grands nombres satisfaite par une chaîne de Markov récurrente au sens de Harris et positive (cf Théorème M.1), on a que pour tout g tel que $(\mu \otimes \nu_\eta \otimes \dots \otimes \nu_\xi)(|g|) < \infty$

$$\frac{1}{n} \sum_{k=1}^n g(\Phi_k) \xrightarrow[n \rightarrow \infty]{p.s.} \mu \otimes \nu_\eta \otimes \dots \otimes \nu_\xi(g).$$

Comme on a supposé que $\ln((\min_{1 \leq i \leq \lambda} \|z + \eta^i \xi^i\|) / \|z\|)$ est intégrable contre $\mu \otimes \nu_\eta \otimes \dots \otimes \nu_\xi$, on en déduit en appliquant la loi forte des grands nombre à l'équation (5.7.73) que

$$\frac{1}{n} \ln(\|X_n\|) \xrightarrow[n \rightarrow \infty]{p.s.} \mu \otimes \nu_\eta \otimes \dots \otimes \nu_\xi \left(\ln \left(\frac{\min_{1 \leq i \leq \lambda} \|z + \eta^i \xi^i\|}{\|z\|} \right) \right) \quad (5.7.75)$$

ce qui achève la démonstration du Théorème.

□

5.7.2 Etude du $(1, \lambda)$ -ES à pas constant

Nous allons voir ici l'étude du $(1, \lambda)$ -ES à pas constant comme un cas particulier de ce que nous venons de faire. Considérons $(1, \lambda)$ -ES à pas constant sur la fonction sphère en dimension 1:

$$X_{n+1} = \arg \min \{ (X_n + \sigma \xi_n^1)^2, \dots, (X_n + \sigma \xi_n^\lambda)^2 \} \stackrel{def}{=} X_n + \sigma \xi_n^*(X_n, \sigma)$$

où $\sigma \in \mathbb{R}^+$, $X_0 \in \mathbb{R}$ et où pour tout n , ξ_n^i , $1 \leq i \leq \lambda$ sont λ variables aléatoires indépendantes identiquement distribuées de loi symétriques que nous noterons ν_ξ . Comme dans le cas auto-adaptatif que nous venons de présenter, l'égalité suivante

$$\min_{1 \leq i \leq \lambda} (X_n + \sigma \xi_n^i)^2 = \sigma \min_{1 \leq i \leq \lambda} \left(\frac{X_n}{\sigma} + \xi_n^i \right)^2$$

implique que $\xi_n^*(X_n, \sigma) = \xi_n^*(X_n/\sigma, 1)$. En définissant $Z_n = \frac{X_n}{\sigma}$, on a comme dans le Lemme 1 que $(Z_n)_{n \geq 0}$ est un chaîne de Markov que l'on peut définir par:

$$Z_{n+1} = Z_n + \xi_n^*(Z_n)$$

avec $\xi_n^*(Z_n)$ défini par $(Z_n + \xi_n^*(Z_n))^2 = \min_{1 \leq i \leq \lambda} (Z_n + \xi_n^i)^2$. Le noyau de transition de $(Z_n)_{n \geq 0}$ est donné pour tout $z \in \mathbb{R}$, $A \in \mathcal{B}(\mathbb{R})$:

$$P(z, A) = \int \mathbb{1}_A(v) p_\xi(v - z) \lambda \mathbb{P}(|v| < |z + \xi|)^{\lambda-1} dv$$

Comme dans le cas auto-adaptatif, si $p_\xi \in L^\infty(\mathbb{R})$ et est continue presque partout on peut montrer que $(Z_n)_{n \geq 0}$ est μ^{Leb} -irréductible, apériodique et que les compacts non vides de \mathbb{R} sont des small sets. Pour montrer que la chaîne est positive, récurrente au sens de Harris et géométriquement ergodique, on utilise la fonction $V(z) = z^2 + 1$. Alors

$$\mathbb{E}_z(V(Z_1)) = z^2 + 2z\mathbb{E}(\xi^*(z)) + \mathbb{E}(\xi^*(z)^2) + 1.$$

En supposant que $\mathbb{E}(|\xi|^2) < \infty$, on a $\mathbb{E}(\xi^*(z)^2) < \lambda \mathbb{E}(|\xi|^2) < \infty$. En procédant de manière à la preuve du Lemme 3, on montre que

$$\lim_{z \rightarrow +\infty} \mathbb{E}(\xi^*(z)) = \mathbb{E}(\xi^{1:\lambda}) < 0 \text{ pour } \lambda \geq 2$$

et

$$\lim_{z \rightarrow -\infty} \mathbb{E}(\xi^*(z)) = \mathbb{E}(\xi^{\lambda:\lambda}) > 0 \text{ pour } \lambda \geq 2$$

On en déduit que

$$\limsup_{z \rightarrow \pm\infty} \frac{\Delta V(z)}{V(z)} < 0.$$

et donc $V(z)$ est une fonction de drift au sens des équations (5.1.8), (5.1.6) et (5.1.5). On en déduit que $(Z_n)_{n \geq 0}$ est positive, Harris récurrente et géométriquement ergodique pour $\lambda \geq 2$. Un corollaire va être comme à la Proposition 4, l'intégrabilité de $\ln(|z|)$ contre l'unique mesure invariante μ de $(Z_n)_{n \geq 0}$.

En appliquant ensuite la loi forte des grands nombres on va en déduire que

$$\frac{1}{n} \ln(|X_n|) \xrightarrow[n \rightarrow \infty]{p.s.} \int_{\mathbb{R}} \mathbb{E}(\ln |1 + \xi^*(z)/z|) d\mu(z)$$

De plus

$$\int_{\mathbb{R}} \mathbb{E}(\ln |1 + \xi^*(z)/z|) d\mu(z) = \int_{\mathbb{R}} \mathbb{E}(\ln |z + \xi^*(z)|) d\mu(z) - \int_{\mathbb{R}} \ln |z| d\mu(z).$$

En appliquant le Théorème de Fubini, on a $\int_{\mathbb{R}} \mathbb{E}(\ln |z + \xi^*(z)|) d\mu(z) = \int_{\mathbb{R}} \ln |z| d\mu(z)$ (preuve analogue à la preuve dans le Théorème 8). On obtient

$$\frac{1}{n} \ln(|X_n|) \xrightarrow[n \rightarrow \infty]{p.s.} 0. \quad (5.7.76)$$

5.7.3 Etude du cas $\lambda = 1$

Jusqu'à présent nous avons cherché des conditions pour que la chaîne initiale (X_n, σ_n) "converge" ou "diverge", où par convergence ou divergence nous entendons le cas où le terme (5.5.56) est respectivement strictement négatif ou strictement positif bien que nous ne soyons pas capable de d'en donner le signe⁸. Ceci se traduit pas une propriété de transience de la chaîne. A présent nous allons étudier le cas limite où $\lambda = 1$ et allons chercher à

8. Notons que lorsque le terme (5.5.56) est nul, on ne peut pas conclure à la convergence de $\|X_n\|$ vers 0 ou à sa divergence vers $+\infty$

montrer que dans ce cas là la chaîne est récurrente. C'est ce que nous allons montrer sous la condition que $E((\eta\xi)^2) < \infty$. Dans le cas $\lambda = 1$ l'algorithme se réduit à:

$$X_{n+1} = X_n + \sigma_n \eta_n \xi_n \quad (5.7.77)$$

$$\sigma_{n+1} = \sigma_n \eta_n \quad (5.7.78)$$

où η_n et ξ sont deux variables aléatoires indépendantes de lois respectives ν_η et ν_ξ . Notons que dans cet algorithme, la fonction à minimiser n'intervient pas. La chaîne de Markov (X_n, σ_n) est une marche aléatoire dans $\mathbb{R} \times \mathbb{R}^+$ tel que *l'incrément* pour X_n est de moyenne nulle. Le noyau de transition de cette chaîne est donné pour tout $(x, \sigma) \in \mathbb{R} \times \mathbb{R}^+$, $A \in \mathcal{B}(\mathbb{R})$ et $B \in \mathcal{B}(\mathbb{R}^+)$ par

$$P((x, \sigma), A \times B) = \iint \mathbb{1}_A(u) \mathbb{1}_B(v) p_\eta\left(\frac{v}{\sigma}\right) p_\xi\left(\frac{u-x}{v}\right) \frac{dudv}{\sigma v} \quad (5.7.79)$$

De la même manière qu'au Théorème 5, on peut montrer sous les hypothèses **H.2** que la chaîne (X_n, σ_n) est irréductible pour la mesure $\mu^{Leb} \times \mu^{Leb+}$ et que les small sets sont des compacts. Sous la condition $E((\eta\xi)^2) < \infty$ la chaîne est récurrente.

Théorème 11. *Si $E((\eta\xi)^2) < \infty$, (X_n, σ_n) est récurrente au sens de Harris.*

PREUVE: La preuve qui suit est similaire à la preuve du Meyn et Tweedie [100] concernant la récurrence de la marche aléatoire sur \mathbb{R}^+ avec incrément de moyenne nulle et de moment d'ordre 2 fini.

Nous allons utiliser la fonction de drift suivante

$$V(x, \sigma) = \begin{cases} \log(1+x) & \text{si } x > R \\ \log(1-x) & \text{si } x < -R \\ 0 & \text{sinon} \end{cases} \quad (5.7.80)$$

où R va être choisi plus loin.

Soit $x > R$, calculons les deux termes

$$V_1(x, \sigma) = E(\log(1+x + \sigma\eta\xi) \mathbb{1}\{x + \sigma\eta\xi > R\}) \quad (5.7.81)$$

$$V_2(x, \sigma) = E(\log(1-x - \sigma\eta\xi) \mathbb{1}\{x + \sigma\eta\xi < -R\}) \quad (5.7.82)$$

tels que $E(V(x, \sigma)) = V_1(x, \sigma) + V_2(x, \sigma)$. Calculons le premier terme,

$$V_1(x, \sigma) = \log(1+x)P(x + \sigma\eta\xi > R) + E\left(\log\left(1 + \frac{\sigma\eta\xi}{1+x}\right) \mathbb{1}\{x + \sigma\eta\xi > R\}\right)$$

Or pour $x > -1$, $\log(1+x) \leq x - \frac{x^2}{2} \mathbb{1}\{x < 0\}$, on en déduit que

$$E\left(\log\left(1 + \frac{\sigma\eta\xi}{1+x}\right)\right) \leq \frac{\sigma}{1+x} E(\eta\xi \mathbb{1}\{x + \sigma\eta\xi > R\}) - \frac{\sigma^2}{2(1+x)^2} E((\eta\xi)^2 \mathbb{1}\{R < x + \sigma\eta\xi < 0\})$$

En notant

$$V_3(x, \sigma) = \frac{\sigma}{1+x} E(\eta\xi \mathbb{1}\{x + \sigma\eta\xi > R\}) \quad (5.7.83)$$

$$V_4(x, \sigma) = \frac{\sigma^2}{2(1+x)^2} E((\eta\xi)^2 \mathbb{1}\{R < x + \sigma\eta\xi < 0\}) \quad (5.7.84)$$

nous avons

$$E\left(\log\left(1 + \frac{\sigma\eta\xi}{1+x}\right)\right) \leq V_3(x, \sigma) + V_4(x, \sigma) \quad (5.7.85)$$

Or comme $E((\eta\xi)^2) < \infty$

$$V_4(x, \sigma) = \frac{\sigma^2}{2(1+x)^2} E((\eta\xi)^2 \mathbb{1}\{\sigma\eta\xi < 0\}) - o\left(\frac{1}{x^2}\right)$$

Et d'après le Lemme 5, $V_3(x, \sigma) = o\left(\frac{1}{x^2}\right)$. Ainsi,

$$V_1(x, \sigma) \leq \log(1+x)P(x + \sigma\eta\xi > R) - \frac{\sigma^2}{2(1+x)^2} E((\eta\xi)^2 \mathbb{1}\{\sigma\eta\xi < 0\}) + o\left(\frac{1}{x^2}\right) \quad (5.7.86)$$

Par ailleurs, comme $x > R$, $1-x < 0$ et le calcul du terme $V_2(x, \sigma)$ nous donne

$$V_2(x, \sigma) = E\left(\log\left((x-1)\left(-1 + \frac{\sigma\eta\xi}{x-1}\right)\right)\right) \mathbb{1}\{x + \sigma\eta\xi < -R\} \quad (5.7.87)$$

$$= \log(x-1)P(x + \sigma\eta\xi < -R) + E\left(\log\left(-1 + \frac{\sigma\eta\xi}{x-1}\right)\right) \mathbb{1}\{x + \sigma\eta\xi < -R\} \quad (5.7.88)$$

Pour $x > 1$, $\log(-1+x) \leq x-2$, on en déduit que

$$E\left(\log\left(-1 + \frac{\sigma\eta\xi}{x-1}\right)\right) \mathbb{1}\{x + \sigma\eta\xi < -R\} \leq \frac{\sigma}{x-1} E(\eta\xi \mathbb{1}\{x + \sigma\eta\xi < -R\}) - 2P(x + \sigma\eta\xi < -R) \quad (5.7.89)$$

Or d'après le Lemme 5, $\frac{\sigma}{1-x} E(\eta\xi \mathbb{1}\{x + \sigma\eta\xi < -R\}) = o\left(\frac{1}{x^2}\right)$, on en déduit avec (5.7.88) et (5.7.89) que

$$V_2(x, \sigma) \leq (\log(x-1) - 2)P(x + \sigma\eta\xi < -R) + o\left(\frac{1}{x^2}\right) \quad (5.7.90)$$

Avec (5.7.86) et (5.7.90), nous obtenons

$$E(V(x, \sigma)) \leq \log(1+x)P(x + \sigma\eta\xi > R) - \frac{\sigma^2}{2(1+x)^2} E((\eta\xi)^2 \mathbb{1}\{\sigma\eta\xi < 0\}) + (\log(x-1) - 2)P(x + \sigma\eta\xi < -R) + o\left(\frac{1}{x^2}\right) \quad (5.7.91)$$

En réécrivant le terme de droite, nous obtenons

$$E(V(x, \sigma)) \leq \log(1+x) - \frac{\sigma^2}{2(1+x)^2} E((\eta\xi)^2 \mathbb{1}\{\sigma\eta\xi < 0\}) \\ - \log(1+x)P(x + \sigma\eta\xi < R) + (\log(x-1) - 2)P(x + \sigma\eta\xi < -R) + o\left(\frac{1}{x^2}\right) \quad (5.7.92)$$

De plus d'après le Lemme 5,

$$-\log(1+x)P(x + \sigma\eta\xi < R) + (\log(x-1) - 2)P(x + \sigma\eta\xi < -R) \leq 0$$

en injectant cela dans (5.7.92), nous obtenons que

$$E(V(x, \sigma)) \leq \log(1+x) - \frac{\sigma^2}{2(1+x)^2} E((\eta\xi)^2 \mathbb{1}\{\sigma\eta\xi < 0\}) + o\left(\frac{1}{x^2}\right)$$

En choisissant R suffisamment grand nous obtenons que pour $x > R$

$$E(V(x, \sigma)) \leq V(x, \sigma)$$

La situation où $x < -R$ est exactement symétrique, on en déduit donc que V est une fonction de drift au sens de l'équation (5.1.5).

□

Lemme 5. *Sous les hypothèses du Théorème précédent*

$$\lim_{x \rightarrow +\infty} xE(\eta\xi \mathbb{1}\{x + \sigma\eta\xi > R\}) = 0. \quad (5.7.93)$$

$$\lim_{x \rightarrow +\infty} -xE(\eta\xi \mathbb{1}\{x + \sigma\eta\xi < -R\}) = 0. \quad (5.7.94)$$

$$-\log(1+x)P(x + \sigma\eta\xi < R) + (\log(x-1) - 2)P(x + \sigma\eta\xi < -R) \leq 0 \quad (5.7.95)$$

PREUVE: Comme $E(\eta\xi) = 0$ on a $E(\eta\xi \mathbb{1}\{\eta\xi > (R-x)/\sigma\}) = -E(\eta\xi \mathbb{1}\{\eta\xi < (R-x)/\sigma\})$ On en déduit que la limite à calculer pour l'équation (5.7.93) est égale à

$$\lim_{x \rightarrow -\infty} xE(\eta\xi \mathbb{1}\{\eta\xi < (R+x)/\sigma\}) \quad (5.7.96)$$

De plus on a

$$0 \leq (R+x)/\sigma \int_{-\infty}^{(R+x)/\sigma} uv p_\eta(u) p_\xi(v) dudv \leq \int_{-\infty}^{(R+x)/\sigma} (uv)^2 p_\eta(u) p_\xi(v) dudv$$

Or à la limite le terme de droite de l'équation précédente est nul. De plus

$$\lim_{x \rightarrow -\infty} (R)/\sigma \int_{-\infty}^{(R+x)/\sigma} uv p_\eta(u) p_\xi(v) dudv = 0$$

On en déduit l'équation (5.7.96). Avec un changement de variable l'équation (5.7.94) revient à l'équation (5.7.96) qui vient d'être montrée. Réécrivons le terme (5.7.95) de la façon suivante

$$\begin{aligned} & -\log(1+x)P(x+\sigma\eta\xi < R) + (\log(x-1) - 2)P(x+\sigma\eta\xi < -R) = \\ & \quad - P(x+\sigma\eta\xi < R) \log\left(\frac{1+x}{x-1}\right) \\ & \quad + \log(x-1)(P(x+\sigma\eta\xi < -R) - P(x+\sigma\eta\xi < R)) - 2P(x+\sigma\eta\xi < -R) \end{aligned}$$

Comme $P(x+\sigma\eta\xi < -R) - P(x+\sigma\eta\xi < R) < 0$, tous les termes du membre de droite sont négatifs.

□

5.7.4 Preuve du Théorème 5

Nous allons distinguer deux cas, dépendant du support de la densité p_ξ . Le premier cas présenté dans la section 5.7.4.1, est lorsque $\text{supp } p_\xi = \mathbb{R}$ (c'est le cas intéressant en pratique). Pour le second cas, développé dans la section 5.7.4.2, nous supposons simplement qu'il existe un intervalle non trivial $[-B_\xi, B_\xi]$ inclut dans $\text{supp } p_\xi$. Bien que le premier cas est inclus dans le deuxième, nous présentons pour ce premier cas une preuve plus simple.

La preuve de ce Théorème requiert deux Lemmes techniques, prouvés en Section 5.7.4.3 : le Lemme 6 et le Lemme 7, établissant des propriétés de la densité du noyau de transition défini par l'équation (5.3.27).

5.7.4.1 Cas où $\text{supp } p_\xi = \mathbb{R}$

Soit C un compact de \mathbb{R} . Nous commençons par l'expression suivante du noyau $P(.,.)$ de $(Z_n)_{n \geq 0}$ donné par l'équation (5.3.26)

$$P(z, A) = \int \mathbb{1}_A(v) p(z, v) dv.$$

D'après le Lemme 6, nous avons que p est strictement positive ce qui assure la μ^{Leb} -irréductibilité de $(Z_n)_{n \geq 0}$. Par ailleurs pour $z \in C$ on a la borne la borne inf suivante

$$P(z, A) \geq \int \mathbb{1}_A(v) \min_{z \in C} p(z, v) dv. \quad (5.7.97)$$

Une conséquence directe du Lemme 6 est que pour tout $v \in \mathbb{R}$, $z \rightarrow p(z, v)$ est continue et strictement positive. Comme C est un compact, on en déduit que pour tout $v \in \mathbb{R}$,

$\min_{z \in C} p(z, v) > 0$. Ceci implique que

$$\nu(A) = \int \mathbb{1}_A(v) \min_{z \in C} p(z, v) dv, \quad (5.7.98)$$

est une mesure non triviale. En introduisant ν dans (5.7.97), on en déduit que

$$\forall z \in C, P(z, A) \geq \nu(A). \quad (5.7.99)$$

Ainsi C est un small set, et $(Z_n)_{n \geq 0}$ est apériodique⁹.

5.7.4.2 Cas où $\exists B_\xi; [-B_\xi, B_\xi] \subset \text{supp } p_\xi$

Soit $z_0 \in \mathbb{R}$ et $A \in \mathcal{B}(\mathbb{R})$. Soit $C_{z_0} = B(z_0, \frac{B_\xi}{4})$ la boule fermée de centre z_0 et de rayon B_ξ . De l'équation (5.3.26) on a, $P(z, A) = \int \mathbb{1}_A(v) p(z, v) dv$. Ainsi

$$P(z, A) \geq \int \mathbb{1}_A(v) \min_{z \in C_{z_0}} p(z, v) dv \quad (5.7.100)$$

Du Lemme 7, on sait que p est continue, et que $p(z, v) > 0$ pour tout $(v, z) \in C_{z_0} \times C_{z_0}$. Comme C_{z_0} est un compact, on a que $\min_{z \in C_{z_0}} p(z, v) > 0$ pour tout $v \in C_{z_0}$. Définissons ν par

$$\nu(A) = \int \mathbb{1}_A(v) \min_{z \in C_{z_0}} p(z, v) dv,$$

ν est une mesure non triviale et

$$\forall A \subset C_{z_0}, \nu(A) > 0. \quad (5.7.101)$$

L'équation (5.7.100) devient alors

$$\forall z \in C_{z_0}, \forall A \in \mathcal{B}(\mathbb{R}) P(z, A) \geq \nu(A) \quad (5.7.102)$$

L'équation (5.7.102) va nous donner la μ^{Leb} -irréductibilité de $(Z_n)_{n \geq 0}$: Soit $z_0 \in \mathbb{R}$ et $A \in \mathcal{B}(\mathbb{R})$ tel que $\mu^{Leb}(A) > 0$. Alors, prouver la μ^{Leb} -irréductibilité revient à prouver que

$$\exists n_{z_0, A} \text{ tel que } P^{n_{z_0, A}}(z_0, A) > 0. \quad (5.7.103)$$

Afin de faire cela, considérons le recouvrement de \mathbb{R} suivant $\bigcup_{k \in \mathbb{Z}} B(k \frac{B_\xi}{8}, \frac{B_\xi}{8})$. Alors, il existe k_0 tel que $\mu^{Leb} \left(A \cap B(k_0 \frac{B_\xi}{8}, \frac{B_\xi}{8}) \right) > 0$. De plus $P(z_0, A) \geq P(z_0, A \cap B(k_0 \frac{B_\xi}{8}, \frac{B_\xi}{8}))$.

9. Fortement apériodique.

En prenant cette fois ci $C_{z_0} = B(k_0 \frac{B_\xi}{8}, \frac{B_\xi}{8})$ et en faisant le même raisonnement qui nous a permis d'arriver aux équations (5.7.102) et (5.7.101) on en déduit que

$$\forall z' \in B(k_0 \frac{B_\xi}{8}, \frac{B_\xi}{8}), P(z', A) > \nu(A) \quad (5.7.104)$$

Comme par ailleurs $A \subset B(k_0 \frac{B_\xi}{8}, \frac{B_\xi}{8})$, l'équation équivalente à l'équation (5.7.101) que l'on va trouver nous donne que $\nu(A) > 0$, d'où on en déduit que

$$\forall z' \in B(k_0 \frac{B_\xi}{8}, \frac{B_\xi}{8}), P(z', A) > 0 \quad (5.7.105)$$

Par ailleurs il existe $n'_{z,A}$ tel que

$$P^{n'_{z,A}}(z, B(k_0 \frac{B_\xi}{8}, \frac{B_\xi}{8})) > 0. \quad (5.7.106)$$

En effet, prenons $C_{z_0} = B((k+1)\frac{B_\xi}{8}, \frac{B_\xi}{4})$ et $A = B((k+1)\frac{B_\xi}{8}, \frac{B_\xi}{8})$ dans l'équation (5.7.102), nous obtenons

$$\forall z \in B(k\frac{B_\xi}{8}, \frac{B_\xi}{8}) \subset C_{z_0}, P\left(z, B\left((k+1)\frac{B_\xi}{8}, \frac{B_\xi}{8}\right)\right) > 0. \quad (5.7.107)$$

Prenons de plus $z_0 \in B((k_1)\frac{B_\xi}{8}, \frac{B_\xi}{8})$: de l'équation (5.7.107), et de l'équation de Chapman Kolmogorov [100, Théorème 3.4.2], on déduit

$$P^{|k_1-k_0|}(z_0, B(k_0 \frac{B_\xi}{8}, \frac{B_\xi}{8})) > 0. \quad (5.7.108)$$

A partir des équations (5.7.105) et (5.7.108), on obtient finalement

$$P^{|k_1-k_0|+1}(z_0, A) > 0 \quad (5.7.109)$$

qui n'est autre que l'équation (5.7.103), avec $n_{z_0,A} = |k_1 - k_0| + 1$.

Venons en maintenant à l'apériodicité et aux small set pour la chaîne $(Z_n)_{n \geq 0}$, comme $\nu(C_{z_0}) > 0$, on déduit de l'équation (5.7.102) que la boule C_{z_0} est un small set et que la chaîne est apériodique. De [100, Théorème 5.5.7, Prop 5.5.5] on déduit que toute union de boules C_{z_i} pour $i \in I$ avec $\text{card } I \in \mathbb{N}$ est un small set et que donc tout ensemble compact est un small set puisqu'inclu dans un small set formé d'unions de C_{z_i} .

5.7.4.3 Lemme pour le Théorème 5

Lemme 6. Soit $p : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}^+$ la densité du noyau de transition de $(Z_n)_{n \geq 0}$ donnée par

$$p(z, v) = \lambda \int_{\mathbb{R}^+} p_\eta(u) p_\xi(v - \frac{z}{u}) \mathbb{P}(|uv| < |z + \eta\xi|)^{\lambda-1} du$$

Si p_ξ est continue presque partout, $\text{supp } p_\xi = \mathbb{R}$ et $p_\xi \in L^\infty(\mathbb{R})$.

Alors pour tout $(z_0, v_0) \in \mathbb{R} \times \mathbb{R}$, p est continue en (z_0, v_0) .

De plus pour tout $(z, v) \in \mathbb{R} \times \mathbb{R}$, $p(z, v) > 0$.

PREUVE: Soit $g(z, u, v) = \mathbb{P}(|uv| < |z + \eta\xi|)^{\lambda-1}$, alors une expression de g est

$$g(z, u, v) = \iint \mathbb{1}_{\{|uv| < |z+xy|\}}(x, y) p_\eta(x) p_\xi(y) dx dy$$

Soit $(z_0, v_0) \in \mathbb{R} \times \mathbb{R}$. Soit $u_0 \in \mathbb{R}^+$. Alors pour presque tout (x, y) ,

$$(z, u, v) \rightarrow \mathbb{1}_{\{|uv| < |z+xy|\}}(x, y)$$

est continue en (z_0, u_0, v_0) . De plus pour tout $(z, u, v) \in \mathbb{R} \times \mathbb{R} \times \mathbb{R}$

$$\mathbb{1}_{\{|uv| < |z+xy|\}}(x, y) p_\eta(x) p_\xi(y) \leq p_\eta(x) p_\xi(y). \quad (5.7.110)$$

D'après le Théorème de convergence dominée on déduit que $(z, u, v) \rightarrow g(z, u, v)$ est continue en (z_0, u_0, v_0) . De plus en intégrant l'équation (5.7.110), on obtient que $g(z, u, v) \leq 1$.

Comme p_ξ est supposée continue presque partout, il est clair que pour presque tout u , $(z, v) \rightarrow p_\xi(v - \frac{z}{u})$ est continue en (z_0, v_0) . Ainsi pour presque tout u

$$(z, v) \rightarrow p_\eta(u) p_\xi(v - \frac{z}{u}) g(z, u, v)^{\lambda-1}$$

est continue en (z_0, v_0) . Par ailleurs pour tout $(z, v) \in \mathbb{R} \times \mathbb{R}$ et pour presque tout u ,

$$p_\eta(u) p_\xi(v - \frac{z}{u}) g(z, u, v)^{\lambda-1} \leq \|p_\xi\|_{L^\infty(\mathbb{R})} p_\eta(u).$$

D'après le Théorème de convergence dominée on en déduit que $(z, v) \rightarrow p(z, v)$ est continue en (z_0, v_0) .

Soit $(z, u, v) \in \mathbb{R} \times \mathbb{R}^+ \times \mathbb{R}$, comme $\text{supp } p_\xi = \mathbb{R}$, on déduit que $g(z, u, v) > 0$. Il s'en suit que pour tout $(z, v) \in \mathbb{R} \times \mathbb{R}$, $p(z, v) > 0$.

□

Lemme 7. Soit $p : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}^+$ la densité du noyau de transition de $(Z_n)_{n \geq 0}$ donnée par

$$p(z, v) = \lambda \int_{\mathbb{R}^+} p_\eta(u) p_\xi(v - \frac{z}{u}) \mathbb{P}(|uv| < |z + \eta\xi|)^{\lambda-1} du$$

Si p_ξ est continue presque partout, et telle qu'il existe $B_\xi > 0$ avec $[-B_\xi, B_\xi] \subset \text{supp } p_\xi$ ¹⁰, $p_\xi \in L^\infty(\mathbb{R})$ et $1 \in \text{supp } p_\eta$.

10. On suppose que zéro est un point intérieur de $\text{supp } p_\xi$.

Alors pour tout $(v_0, z_0) \in \mathbb{R} \times \mathbb{R}$, F est continue en (v_0, z_0) . De plus, pour tout $(v, z) \in C_{z_0} \times C_{z_0}$, $p(z, v) > 0$ avec

$$C_{z_0} = B\left(z_0, \frac{B_\xi}{4}\right)$$

PREUVE: Soit $g(z, u, v) = \mathbb{P}(|uv| < |z + \eta\xi|)^{\lambda-1}$, alors une expression de g est

$$g(z, u, v) = \iint \mathbb{1}_{\{|uv| < |z+xy|\}}(x, y) p_\eta(x) p_\xi(y) dx dy.$$

Pour la continuité la preuve est la même que dans le Lemme 6. Il ne reste plus qu'à prouver maintenant la positivité de p sur $C_{z_0} \times C_{z_0}$ où $z_0 \in \mathbb{R}$. On va supposer que $z_0 \in \mathbb{R}^+$, la preuve dans le cas où $z_0 \in \mathbb{R}^-$ est analogue. Tout d'abord prouvons que

$$\forall (v, z) \in C_{z_0} \times C_{z_0}, \exists \epsilon_1 \text{ tel que } \forall u \in [1 - \epsilon_1, 1 + \epsilon_1], \quad g(z, u, v) > 0. \quad (5.7.111)$$

Soit $0 < \epsilon < 1$ suffisamment petit pour que $[1 - \epsilon, 1 + \epsilon] \subset \text{supp } p_\eta$ et soit $u \in [1 - \epsilon, 1 + \epsilon]$ et $(z, v) \in C_{z_0} \times C_{z_0}$. On peut donc écrire

$$(1 - \epsilon)\left(z_0 - \frac{B_\xi}{4}\right) \leq |uv| \leq (1 + \epsilon)\left(z_0 + \frac{B_\xi}{4}\right)$$

Pour $(x, y) \in [1 - \epsilon, 1 + \epsilon] \times [-B_\xi, B_\xi]$ on obtient

$$z + xy \leq z_0 + \frac{B_\xi}{4} + (1 + \epsilon)B_\xi$$

et

$$-z_0 - \frac{B_\xi}{4} - (1 - \epsilon)B_\xi \leq -z - xy$$

Si ϵ est tel que

$$\begin{cases} (1 + \epsilon)\left(z_0 + \frac{B_\xi}{4}\right) < z_0 + \frac{B_\xi}{4} + (1 + \epsilon)B_\xi \\ -z_0 - \frac{B_\xi}{4} - (1 - \epsilon)B_\xi < (1 - \epsilon)\left(z_0 - \frac{B_\xi}{4}\right) \end{cases} \quad (5.7.112)$$

alors

$$\mu^{Leb} \otimes \mu^{Leb} (\{(x, y) \in \text{supp } p_\eta \times \text{supp } p_\xi \text{ tel que } |uv| < |z + xy|\}) > 0$$

et ainsi

$$\mu^{Leb} \otimes \mu^{Leb} (\text{supp } \mathbb{1}_{\{|uv| < |z+xy|\}}) > 0.$$

Choisissons $0 < \epsilon_1 < 1$ tel que l'équation (5.7.112) soit satisfaite (il est possible de choisir un tel ϵ_1 comme on peut le voir en passant par exemple à la limite dans l'équation (5.7.112)) et tel que $[1 - \epsilon_1, 1 + \epsilon_1] \subset \text{supp } p_\eta$, l'équation (5.7.111) s'en suit.

Montrons maintenant que l'on peut choisir ϵ_3 tel que $[1 - \epsilon_3, 1 + \epsilon_3] \subset \text{supp } p_\eta$ et tel que pour $u \in [1 - \epsilon_3, 1 + \epsilon_3]$ et $(v, z) \in C_{z_0} \times C_{z_0}$

$$p_\eta(u)p_\xi(v - \frac{z}{u})g(z, u, v)^{\lambda-1} > 0. \quad (5.7.113)$$

Soit $0 < \epsilon < 1$, suffisamment petit pour que $[1 - \epsilon, 1 + \epsilon] \subset \text{supp } p_\eta$, et soit $u \in [1 - \epsilon, 1 + \epsilon]$ et $(v, z) \in C_{z_0} \times C_{z_0}$. On peut écrire

$$(z_0 - \frac{B_\xi}{4}) + \frac{-z_0 - \frac{B_\xi}{4}}{1 + \epsilon} \leq v - \frac{z}{u} \leq z_0 + \frac{B_\xi}{4} + \frac{(\frac{B_\xi}{4} - z_0)}{1 - \epsilon}$$

En choisissant $0 < \epsilon_2 < 1$ tel que $[1 - \epsilon_2, 1 + \epsilon_2] \subset \text{supp } p_\eta$ et les inégalités suivantes ont lieu

$$\begin{cases} z_0 + \frac{B_\xi}{4} + \frac{(\frac{B_\xi}{4} - z_0)}{1 - \epsilon} \leq B_\xi \\ -B_\xi \leq (z_0 - \frac{B_\xi}{4}) + \frac{-z_0 - \frac{B_\xi}{4}}{1 + \epsilon} \end{cases} \quad (5.7.114)$$

il vient que pour tout $u \in [1 - \epsilon_2, 1 + \epsilon_2]$, $|v - \frac{z}{u}| \leq B_\xi$ lorsque $v \in C_{z_0}$ et $z \in C_{z_0}$. Soit $\epsilon_3 = \epsilon_1 \wedge \epsilon_2$, pour $u \in [1 - \epsilon_3, 1 + \epsilon_3]$ et $(v, z) \in C_{z_0} \times C_{z_0}$

$$p_\eta(u)p_\xi(v - \frac{z}{u})g(z, u, v)^{\lambda-1} > 0.$$

Ce qui achève de démontrer (5.7.113) et implique que

$$p(z, v) \geq \int_{[1 - \epsilon_3, 1 + \epsilon_3]} p_\eta(u)p_\xi(v - \frac{z}{u})g(z, u, v)^{\lambda-1} du > 0.$$

□

5.7.5 Preuve du Lemme 4

PREUVE: Pour la preuve de ce Lemme, on considère la chaîne de Markov

$$\Phi_n = (Z_n, \eta_n^1, \dots, \eta_n^\lambda, \xi_n^1, \dots, \xi_n^\lambda)$$

, que l'espace d'état $(\mathbb{R}^+)^{\lambda+1} \times (\mathbb{R})^\lambda$ muni de la tribu borélienne $\mathcal{B}((\mathbb{R}^+)^{\lambda+1} \times (\mathbb{R})^\lambda)$. Le noyau de transition \mathbf{P} pour cette chaîne est donné par

$$\begin{aligned} \mathbf{P} \left((z, \eta^1, \dots, \xi^\lambda), A \times A^1 \times \dots \times A^\lambda \times B^1 \times \dots \times B^\lambda \right) = \\ P(z, A)\nu_\eta(A^1) \dots \nu_\eta(A^\lambda)\nu_\xi(B^1) \dots \nu_\xi(B^\lambda) \end{aligned} \quad (5.7.115)$$

où P est le noyau de transition de la chaîne $(Z_n)_{n \geq 0}$ et ν_η (resp. ν_ξ) est la loi de η (resp. ξ). Nous allons montrer que la chaîne Φ_n satisfait la condition de drift donnée par l'équation (5.1.8) pour $V = V_1(z) + V_2(\eta^1, \dots, \eta^\lambda, \xi^1, \dots, \xi^\lambda)$, où V_1 est tel que la chaîne $(Z_n)_{n \geq 0}$ satisfait (5.1.8) et V_2 est une fonction ≥ 1 . On peut tout d'abord remarquer que l'équation (5.1.3) implique que $(\mathbb{R}^+)^{\lambda} \times (\mathbb{R})^{\lambda}$ est un small set pour la chaîne $(\eta_n^1, \dots, \eta_n^\lambda, \xi_n^1, \dots, \xi_n^\lambda)$. Alors, pour $V_2 \geq 1$, l'opérateur de drift Δ appliqué à $(\eta^1, \dots, \eta^\lambda, \xi^1, \dots, \xi^\lambda) \in (\mathbb{R}^+)^{\lambda} \times (\mathbb{R})^{\lambda}$ est égal à

$$\begin{aligned} \Delta V_2(\eta^1, \dots, \eta^\lambda, \xi^1, \dots, \xi^\lambda) = \\ E(V_2(H^1, \dots, H^\lambda, \Xi^1, \dots, \Xi^\lambda)) - V_2(\eta^1, \dots, \eta^\lambda, \xi^1, \dots, \xi^\lambda) \end{aligned} \quad (5.7.116)$$

où pour $1 \leq i \leq \lambda$, H^i (resp. Ξ^i) est une variable aléatoire suivant la distribution ν_η (resp. ν_ξ). Alors l'équation (5.7.116) implique que la condition de drift donnée par l'équation (5.1.8) est satisfaite avec $C_2 = (\mathbb{R}^+)^{\lambda} \times (\mathbb{R})^{\lambda}$ comme small set, $b_C^2 = E(V_2(H^1, \dots, H^\lambda, \Xi^1, \dots, \Xi^\lambda))$ et $\lambda_C^2 = 1$. On prouve maintenant que la chaîne Φ_n satisfait la condition de drift donnée par l'équation 5.1.8 pour

$$V(z, \eta^1, \dots, \eta^\lambda, \xi^1, \dots, \xi^\lambda) = V_1(z) + V_2(\eta^1, \dots, \eta^\lambda, \xi^1, \dots, \xi^\lambda)$$

où V_1 est tel que la chaîne $(Z_n)_{n \geq 0}$ satisfait l'équation (5.1.8) *i.e.*

$$\Delta V_1(z) \leq -\lambda_C^1 V_1(z) + b_C^1 \mathbb{1}_{C_1}(z) \quad (5.7.117)$$

et V_2 est une fonction ≥ 1 . En effet, grâce à l'expression 5.7.115 du noyau de transition de Φ_n , nous obtenons

$$\Delta V(z, \eta^1, \dots, \eta^\lambda, \xi^1, \dots, \xi^\lambda) = \Delta V_1(z) + \Delta V_2(\eta^1, \dots, \eta^\lambda, \xi^1, \dots, \xi^\lambda) \quad (5.7.118)$$

En utilisant une condition de drift satisfaite par à la fois V_1 et V_2 , nous obtenons

$$\begin{aligned} \Delta V(z, \eta^1, \dots, \eta^\lambda, \xi^1, \dots, \xi^\lambda) \\ \leq (-\lambda_C^1 V_1 + b_C^1 \mathbb{1}_{C_1})(z) + (-\lambda_C^2 V_2 + b_C^2 \mathbb{1}_{C_2})(\eta^1, \dots, \eta^\lambda, \xi^1, \dots, \xi^\lambda) \end{aligned}$$

Cela nous donne

$$\Delta V \leq -\lambda_C^1 \wedge \lambda_C^2 V + b_C^1 \vee b_C^2 \mathbb{1}_{C_1 \times C_2} \quad (5.7.119)$$

qui est l'équation de Drift 5.1.8.

□

Troisième partie

Une application au contrôle par laser

Chapitre 6

Algorithmes évolutionnaires pour le contrôle de l'orientation de molécules

Ce chapitre reprend l'intégralité de l'article [11]:

[11] A. Auger, A. Ben Haj Yedder, E. Cancès, C. Le Bris, C. M. Dion, A. Keller, and O. Atabek. Optimal laser control of molecular systems: methodology and results. *Mathematical Models and Methods in Applied Sciences*, 12(9):1281-1315, 2002.

Résumé

Les travaux présentés sont les résultats d'une collaboration avec C. M. Dion, A. Keller et O. Atabek, physiciens du *Laboratoire de Photophysique Moléculaire* d'Orsay. Le problème physique posé est celui du contrôle de l'orientation d'un système moléculaire lorsque celui-ci est soumis à un champ laser. Deux approches physiques existent pour ce problème. D'une part une approche expérimentale [6] et d'autre part une approche numérique [48, 47]. C'est cette dernière approche qui a été utilisée. Elle consiste à modéliser un système moléculaire simple et à le résoudre numériquement. La molécule précisément étudiée ici est la molécule *HCN*. L'évolution de cette molécule, lorsqu'elle est soumise à un champ laser, est décrite par l'équation de Schrödinger (cf équation 6.2.7). En toute généralité, la résolution numérique de ce système moléculaire est un problème en soi, parce qu'il s'apparente à la résolution

de systèmes d'équations en grande dimension. Dans le cas particulier de ce chapitre, la modélisation faite [45] permet de réduire la dimension à 2, ce qui permet de traiter numériquement le problème. La résolution, réalisée par C. Dion [45] consiste, étant donné un champ laser (voir la paramétrisation utilisée pour ce champs laser à la Section 6.2.2) à calculer l'évolution de la fonction d'onde décrivant l'état du système sur une grille de pas de temps. Notons que cette modélisation physique présuppose, entre autre, que le système est à température nulle, ce qui est une limitation du modèle considéré pour une application des résultats sur une expérience réelle.

Le problème de contrôle par laser se pose en ces termes: On cherche un champ électrique (ou champ laser) qui va amener le système moléculaire dans une orientation donnée. Ce problème de contrôle est traduit en problème d'optimisation. Étant donné un champ électrique, l'orientation du système soumis à ce champ est mesurée grâce à différents critères calculés à partir de l'état du système (résolu numériquement dans [45]). Une discussion sur les différents critères choisis pour mesurer cette orientation est donnée à la Section 6.2.2. Le critère ainsi choisi est la fonction à valeur réelles à optimiser: les paramètres du champ laser déterminent l'espace de recherche.

Les caractéristiques connues *a priori* sur ce problème d'optimisation sont les suivantes. La fonction est chère à calculer puisque une évaluation de fonction prend environ 1 minute (sur un Pentium II, 500 MHz) et cette fonction n'étant pas connue analytiquement (résultat de la résolution numérique de l'équation de Schrödinger), son gradient n'est accessible que par différences finies ou par une méthode de différentiation automatique. L'espace de recherche étant de dimension 70 (en fait la dimension de cet espace a été réduite dans un second temps), la méthode des différences finies n'était pas envisageable. En conséquence, dans sa thèse, A. Ben Hay Yedder [154] a mis en oeuvre une méthode de différentiation automatique pour le calcul du gradient d'un des critères, et testé la méthode BFGS pour l'optimisation de ce critère. Il s'est avéré que l'algorithme convergeait rapidement vers un minimum local proche du point initial. Un algorithme évolutionnaire, connu pour être robuste aux minima locaux, a donc été mis en place pour l'optimisation. Ceci est décrit dans ce chapitre, tout comme les résultats obtenus. Notons que la souplesse de la mise en oeuvre des algorithmes évolutionnaires a permis d'optimiser différents critères qui intéressaient les physiciens. Il a suffit de remplacer une fonction fitness par une autre, ceci est incomparablement plus rapide que de mettre en place avec une méthode nécessitant le gradient de la fonction (l'implémentation d'une

méthode de différentiation automatique pour le calcul du gradient ou simplement poser le problème adjoint ne sont pas des choses immédiates).

Deux algorithmes évolutionnaires ont été utilisés pour résoudre le problème, ils sont décrits en détail à la Section 6.3.2. Le deuxième algorithme est précisément un algorithme SA-ES non isotrope (cf chapitre 1).

Les résultats obtenus par l'approche évolutionnaire sont décrits à la section 6.4, ils ont par ailleurs fait l'objet de la publication dans la revue de physique [155]. D'autres publications dans des journaux de physique sur le même type de problèmes et utilisant une approche par algorithmes évolutionnaires pour l'optimisation, témoignent du succès de cette approche pour ce domaine de la physique [50, 7].

Enfin, comme mentionné dans l'introduction de cette thèse, c'est entre autre la mise en oeuvre de l'algorithme SA-ES sur ce problème qui a motivé le chapitre 2. En fait, beaucoup d'essais lancés, convergeaient "prématurément" vers des minima locaux non intéressants¹. Ceci était encore plus vrai pour certains jeux de paramètres que pour d'autres, ce qui nécessitait pas mal de réglages des paramètres comme la taille de la population, les écart-types initiaux et les paramètres τ et τ' pour la mutation log normale (cf chapitre 1). Pour ces essais qui convergeaient vers des minima locaux non intéressants, les valeurs des paramètres de la mutation n'étaient clairement pas optimaux, généralement trop petits par rapport au paysage de la fitness (testé à la main en calculant la hessienne locale de la fonction là où avait lieu la convergence prématurée), et beaucoup d'individus étaient évalués plus jetés. Ceci a motivé l'introduction d'une méthode faisant un meilleur usage des individus et de leur valeur de *fitness*.

1. En fait, on parle de minima locaux non intéressants parce que, les champs lasers trouvés sont aussi des minima locaux au sens où la valeur minimale théorique du critère est -1, et les valeurs du critère correspondantes aux champs intéressants pour les physiciens étaient au alentour de -0.25

Optimal laser control of molecular systems: methodology and results

A. AUGER, A. BEN HAJ YEDDER, E. CANCES, and C. LE BRIS

*CERMICS, École Nationale des Ponts et Chaussées
6 & 8, avenue Blaise Pascal, Cité Descartes,
Champs sur Marne, 77455 Marne-La-Vallée Cedex 2, FRANCE*

C. M. DION, A. KELLER and O. ATABEK

*Laboratoire de Photophysique Moléculaire
Bâtiment 213, Campus d'Orsay, 91405 Orsay, FRANCE*

We report on some mathematical and numerical work related to the control of the evolution of molecular systems using laser fields. More precisely, the control of the orientation of molecules is our goal. We treat this as an optimal control problem and optimize the laser field to be used experimentally by using both deterministic and stochastic algorithms. Comparisons between the different strategies are drawn. In particular, when gradients of the cost functional are used, the different ways for their computation are compared and analyzed.

6.1 Introduction

We wish to report on theoretical and numerical work devoted to the modeling of the control of chemical reactions by laser fields. The laser control of chemical reactions is indeed a very active field of laser physics, at the crossroads between quantum chemistry, quantum mechanics, and theoretical and experimental femtophysics. Manipulation of molecular systems using laser fields is today an experimental reality [6], provided one restricts his aims to reasonable goals, as will be seen below. This leads to a mostly unexplored field for mathematical analysis and numerical simulation. Numerical simulations can indeed efficiently complement the experimental strategy, both by explaining the deep nature of the phenomena involved and by optimizing the parameters to be used experimentally.

We present here the contributions of our team, which is composed both of mathematicians and physicists. The emphasis is here on the mathematical aspects and the numerical

techniques. A companion article [155] focusing on the physical aspects appears elsewhere. The most striking result of our work is given in [49].

Before we discuss the technicalities, let us briefly state in a rather formal way the problem we shall deal with. All details will be given in Section 6.2, and for pedagogical purposes we prefer to only give a vague setting in this explanatory survey.

The evolution of a molecular system subjected to a laser field $\vec{\mathcal{E}}$ is modeled by the time-dependent Schrödinger equation

$$i\hbar \frac{\partial \psi}{\partial t} = H_0 \psi + \vec{\mathcal{E}}(t) \cdot \vec{D}(\vec{\mathcal{E}}(t)) \psi, \quad (6.1.1)$$

complemented with the initial condition $\psi(t=0) = \psi_0$. In this equation, the wave function ψ is assumed to depend only on the coordinates of the various nuclei the molecular system is composed of. The presence of the electrons is accounted for through an effective potential acting on the nuclei, and contained in the Hamiltonian H_0 of the free system (when the laser is turned off). We denote by $\vec{D}(\vec{\mathcal{E}}(t))$ the dipole moment of the molecule in presence of an external electric field $\vec{\mathcal{E}}(t)$; at the first order perturbation theory, one can use the form $\vec{D}(\vec{\mathcal{E}}(t)) = \vec{\mu}_0 + \vec{\alpha} \vec{\mathcal{E}}$. More sophisticated models would feature higher order expansion of $\vec{D}(\vec{\mathcal{E}}(t))$ interactions, still in the perturbation setting, or even a true dependence of the wave function ψ and the Hamiltonian H with respect to the coordinates of *all nuclei and electrons* of the molecular system. To the present day, the latter model is out of reach of numerical treatment.

In order to state an optimal control problem, we need, in addition to the direct equation (6.1.1) modeling the evolution of the system, to define a cost function. Minimizing this cost function will give a formal sense to the physical target we want to reach. In our work, we consider a linear molecular system and intend to orient it in the direction of the linearly polarized laser field. The cost function we adopt will therefore reflect this wish. Just to fix the ideas, let us mention an example of cost function in the simplest case when the state of the system ψ (solution to equation 6.1.1) is a function of time t and of the angle θ between the axis of the system and the direction of the laser field:

$$J(\mathcal{E}) = \frac{1}{T} \int_{t=0}^{t=T} \int_{\theta=0}^{\theta=\pi} |\psi(t, \theta)|^2 \cos \theta \sin \theta d\theta dt. \quad (6.1.2)$$

The reason why we choose an orientation problem as our control problem, and consequently such an objective function will be made clear below. Other forms of the cost function will also be given later in this article.

The simple setting we have just indicated above suffices to now underline the peculiarities of the optimal control problem we have to tackle, with respect to other optimal control problems that the reader may have in mind and that come from more usual domains of the engineering world (aeronautics, ...). Let us now emphasize these peculiarities.

From the standpoint of the mathematical theory, this problem is *bilinear* (the control $\vec{\mathcal{E}}$ multiplies the state ψ) which at once puts the problem on a very high level of mathematical difficulty. Indeed, the mathematical theoretical results on bilinear control are very rare. In infinite dimension, i.e., for the PDE (6.1.1), since the celebrated work by Ball and Slemrod [17], no real progress has been made, to the best of our knowledge. For the finite dimensional approximation of (6.1.1), there exist some results that can also be extended to the infinite dimensional case but that are not very easy to exploit (so far). We refer to the work of G. Turinici et al. [144, 145, 146] for some recent progress on the theory of exact controllability for systems such as those we deal with here. For the optimal control problem, *some* minor things can be done. We refer in particular to [34] where some of us have proven the existence of an optimal field in a very academic and simplified setting. We shall not elaborate any longer on these theoretical aspects and now concentrate on more practical ones.

A noticeable peculiarity is the fact that, in most cases, the control $\vec{\mathcal{E}}$ is *distributed in time*, and not in space. It is not a crucial fact for the sequel (cases when $\vec{\mathcal{E}}$ depends both on time and space could be treated in the same fashion, however with slightly more tedious computations) but it is rather convenient and constitutes a very reasonable approximation in the case of small molecular systems such as atoms and small molecules. At the scale of such a system, the laser light is indeed seen as homogeneous in space. Such a distributed in time control is not that usual for a partial differential equation such as (6.1.1).

In addition, special attention must be paid to the fact that although our goal is to drive the system from one initial state to some other specific state through a controlled time-dependent evolution, the cost function we choose to formulate our mathematical problem is not a distance to a target state, but the mean value of an observable (a measure of the orientation of the molecular system with the field). We wish to comment a little bit further on this point. The ultimate goal of the manipulations we want to model is the control of chemical reactions. This means for instance making a system ABC split into $AB + C$ rather than into $A + BC$ (see [33] for an introduction to this problem). Succeeding in making a chemical reaction possible does not necessarily mean driving the initial state to the final one, but sometimes (and in fact most of the times) only succeeding in *preparing* the initial system in a good way so that afterwards the desired reaction spontaneously happens. In that respect, orienting a molecule in space is both a modest and sufficient goal. Once it is conveniently “geometrically” prepared, the goal is almost reached. Nature will do the rest of the job. In addition, there are today experimental evidences showing that aligning a molecule (orientation is one step forward alignment) with a laser *is* feasible, and constitutes a significant step that can be used to efficiently control reactions (see the groundbreaking work by H. Stapelfeldt [128, 95]). The problem of orientation is therefore a good problem to look at.

It is also enlightening to consider this problem from the practical standpoint. Let us first indicate some orders of magnitude. Typically, the space scale is that of a molecule, namely a few angströms (10^{-10} m), and the time scale is that of the vibration of a molecular bond,

namely ten femtoseconds (10^{-14} s). The total time of simulation for equation (6.1.1) is thus typically the picosecond (10^{-12} s). There are two main consequences of this time scale. First, the control needs to be an *open-loop* control, since it is clear that one cannot update the field in real time with electronic devices. In other words, the only system that can react as fast as the molecular system is precisely the system itself. The second consequence is that we must think of this problem in a completely different way from the way we think of usual control problems: we are here in a framework where we can do thousands of experiments within a minute (while we cannot launch a rocket thousands times). This ability to make many experiments has in turn two consequences. First, one can imagine, and it is indeed done, to couple the numerical search for the optimal field not with the numerical simulation of equation (6.1.1), but with the experiment itself [6, 89]. The experimental solution of (6.1.1) is indeed much faster than its resolution on a computer. There is here some matter of reflection for experts in scientific computing. Second, one of the major problems of this field is the tremendous amount of data that are at our disposal. A challenge is to find a way to exploit them in the optimization cycle. We shall not give in this article any definite answer to the questions and concerns raised above, but it is sound to keep in mind these points.

One must also know about the practical parameters for a laser field. One of us has presented in [31, 30] a rapid account of this point, and we refer to it, or to the comprehensive report [142] for more details. Let us only say that a trade-off has to be made between the power of the laser, its time resolution, its repetition frequency, and also its price and its size. The laser fields we shall make use of have intensities in the range $[10^{12}, 10^{13}]$ W/cm², are able to have a risetime of the order of 10^{-14} s, and the light they create has frequency around 10^{14} Hz. A very peculiar feature appears here again. One can ask the question whether it is better to optimize upon *only* the fields that are today experimentally feasible or to consider all fields without taking into account any contemporary technological constraint. Both approaches may be useful. In particular, the second one may help in designing the lasers physicists do need for the next generation. In the present article, we mostly choose the first approach, taking explicitly into account the technical requirements. We shall however also explore the second one (see more on this point below when we optimize with ten laser fields).

The stage is now set. Let us say a few words on the methodology we choose for the search of the optimal laser field.

First and foremost, we must emphasize that the present study is far from being the first attempt to find numerically the optimal laser field. There exist many theoretical studies based upon the construction of small systems of ODEs approximating (6.1.1) so that the optimal (or exact) control problem can be treated explicitly “by hand”. The leading experts of this approach, fundamentally based upon a deep knowledge of (or intuition of) the main mechanisms are P. Brumer, P. Schapiro and coworkers [33, 32]. Other outstanding contributions, in particular on intense laser fields are due to A. Bandrauk [37]. On the other hand, the optimal control methodology in the sense applied mathematicians speak about it has already been

thoroughly explored by physicists, in the first row of which stands H. Rabitz [156, 88]. See also works by Fujimura [80], Sakai [90]. However, in all these contributions, the algorithms used for the numerical search for the optimized field are seen as black boxes, and not as topics for research. Our own approach aims at complementing the work of these leading researchers in physics by exploring the capabilities of the most recent optimization tools, by comparing them to one another on the present problem, by drawing conclusions on the best tools to be used, and also, when possible, by improving the physical conclusions.

On the present problem, we shall investigate mainly the following issues, which are of general interest, but whose response may differ from one problem to another:

- use on this specific case of deterministic algorithms (gradient-like algorithms), of stochastic algorithms (genetic algorithms and evolutionary strategies) and of algorithms mixing the two approaches, such as genetic algorithms accelerated by mutation by gradient
- comparison of the different ways to compute the gradient when needed: discretization of the adjoint equation, computation of the adjoint of the discrete equation, automatic differentiation
- impact of the choice of the cost function on the result, multicriteria approaches,...

The sequel of this article is organized as follows. In the next section, we give a detailed presentation of the problem under study, making more precise the quantities (Hamiltonian H , state ψ , electric field $\vec{\mathcal{E}}$, dipole moment \vec{D} , cost function J) we have described above in a somewhat vague way. Section 6.3 describes the different optimization methods we shall make use of. For some of them, we shall need to compute the gradient of the cost function. In Section 6.3.1, we therefore make a numerical analysis to determine which strategy is the best one to compute this gradient. In Section 6.3.2 we give a short description of stochastic algorithms we employed. Section 6.4 then gives the results obtained for our problem with deterministic algorithms and with stochastic ones. Finally, in Section 6.5, we shall summarize our main results and indicate the directions of our current and future research.

6.2 Statement of the control problem ---

6.2.1 The system under study and the control problem

The molecular system we study is the linear HCN molecule (hydrogen cyanide). This molecule has been chosen because it is linear in its ground state and should stay so if the laser frequency is out of resonance with respect to the bending modes. Therefore it constitutes a perfect toy object for testing our methodology. We use the so-called Jacobi coordinates ($\mathbf{R} = (R, r), \theta, \varphi$)

to parameterize the state of the molecule (see Figure 6.1). The free Hamiltonian H_0 can be written as $H_0 = H_{\text{vib}}(\mathbf{R}) + H_{\text{rot}}(\mathbf{R}, \theta, \varphi) + V(\mathbf{R})$ and the dipole moment is written as $D(\mathcal{E}(t)) = -\mu_0(R, r) \cos \theta - \frac{\mathcal{E}(t)}{2} [\alpha_{\parallel}(R, r) \cos^2 \theta + \alpha_{\perp}(R, r) \sin^2 \theta]$. Then the general form for the Hamiltonian, given in [48], is

$$H(\mathbf{R}, \theta, \varphi, t) = H_{\text{vib}}(\mathbf{R}) + H_{\text{rot}}(\mathbf{R}, \theta, \varphi) + V(\mathbf{R}) + H_{\text{laser}}(\mathbf{R}, \theta, \varphi, t), \quad (6.2.3)$$

where $T_{\text{rot}} + H_{\text{rot}}$ denotes the kinetic energy operator with

$$H_{\text{vib}}(\mathbf{R}) = -\frac{\hbar^2}{2\mu_{HCN}} \frac{1}{R^2} \frac{\partial}{\partial R} \left(R^2 \frac{\partial}{\partial R} \right) - \frac{\hbar^2}{2\mu_{CN}} \frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial}{\partial r} \right),$$

$$H_{\text{rot}}(\mathbf{R}, \theta, \varphi) = -\frac{\hbar^2}{2(\mu_{HCN} R^2 + \mu_{CN} r^2)} \left[\frac{1}{\sin \theta} \frac{\partial}{\partial \theta} \left(\sin \theta \frac{\partial}{\partial \theta} \right) + \frac{1}{\sin^2 \theta} \frac{\partial^2}{\partial \varphi^2} \right],$$

where $V(\mathbf{R})$ denotes the effective potential resulting from the electrostatic interaction between nuclei and electrons (in their ground state), while

$$\begin{aligned} H_{\text{laser}}(\mathbf{R}, \theta, \varphi, t) &= \mathcal{E}(t) \cdot D(\mathcal{E}(t)) \\ &= -\mu_0(R, r) \mathcal{E}(t) \cos \theta - \frac{\mathcal{E}^2(t)}{2} [\alpha_{\parallel}(R, r) \cos^2 \theta + \alpha_{\perp}(R, r) \sin^2 \theta] \end{aligned}$$

denotes the interaction between the molecule and the laser field. In the former formulas, μ_{CN}

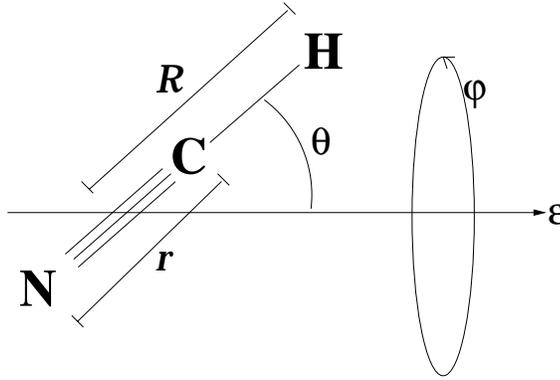


FIG. 6.1: Model for the HCN molecule.

and μ_{HCN} represent the reduced masses:

$$\mu_{CN} = \frac{m_C m_N}{m_C + m_N}, \quad \mu_{HCN} = \frac{m_H (m_C + m_N)}{m_H + m_C + m_N}$$

and μ_0 is the permanent dipole moment. The coefficients α_{\parallel} and α_{\perp} are respectively the parallel and the perpendicular components of the diagonal polarizability tensor $\overline{\alpha}$ given by

$\alpha_{\parallel} = \alpha_{zz}$ and $\alpha_{\perp} = \alpha_{xx} = \alpha_{yy}$ when (Oz) is the molecular axis.

As a first step toward the treatment of the sophisticated model (6.2.3), we consider in all the remainder of this article the case of a rigid rotor: the problem depends only on the angular variables θ, ϕ . Furthermore, symmetry conservation around the laser polarization axis allows us to separate the motion in ϕ from the motion in θ , and consider only the latter in our calculations. The Hamiltonian (6.2.3) therefore reduces to

$$H = H(\theta, t) = H_{rot}(\theta) + H_{laser}(\theta, t), \quad (6.2.4)$$

with

$$H_{rot}(\theta) = -\frac{\hbar^2}{2(\mu_{HCN}R^2 + \mu_{CN}r^2)} \frac{1}{\sin \theta} \frac{\partial}{\partial \theta} \left(\sin \theta \frac{\partial}{\partial \theta} \right)$$

and

$$H_{laser}(\theta, t) = -\mu_0(R, r)\mathcal{E}(t) \cos \theta - \frac{\mathcal{E}^2(t)}{2} [\alpha_{\parallel}(R, r) \cos^2 \theta + \alpha_{\perp}(R, r) \sin^2 \theta],$$

where R and r are fixed at their equilibrium value. The objective function $J(\mathcal{E})$ we are optimizing will be detailed in Section 6.2.3 but let us now introduce the instantaneous criterion $j(t)$ used to compute $J(\mathcal{E})$ and which is the measure of the orientation at time t (see [64] for more details),

$$j(t) = \langle \cos \theta \rangle = \int_0^{\pi} \cos \theta \mathcal{P}(\theta, t) \sin \theta d\theta, \quad (6.2.5)$$

where $\mathcal{P}(\theta, t)$ is the angular distribution of the molecule. In the case of rigid rotor angular distribution is reduced to $\mathcal{P}(\theta, t) = \|\psi\|_{\mathbb{C}}^2$ where $\|\psi\|_{\mathbb{C}}^2$ denotes the squared norm of the complex ψ . The instantaneous criterion therefore becomes

$$j(t) = \int_0^{\pi} \cos \theta \|\psi\|_{\mathbb{C}}^2 \sin \theta d\theta. \quad (6.2.6)$$

The instantaneous criterion $j(t)$ takes its values in the range $[-1, 1]$, the values -1 and 1 corresponding respectively to a molecule pointing in the direction of the laser field polarization axis and in the opposite direction.

The Schrödinger equation

$$\begin{cases} i\hbar \frac{\partial \psi}{\partial t} = H \psi, \\ \psi(t=0) = \psi_0. \end{cases} \quad (6.2.7)$$

depending only on the variable θ is numerically solved with an operator splitting method [60] coupled with a FFT for the kinetic part as shown in [40, 109]. Table 6.1 summarizes the parameters of the HCN molecule for R and r fixed at their equilibrium value.

6.2.2 Choice of the set of electric fields

We now describe the set of laser fields we minimize upon. As said in the introduction, both strategies of restricting oneself to the experimental state of the art or of considering the most

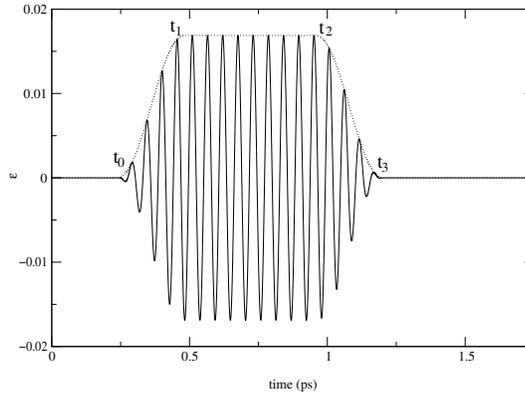
TAB. 6.1: *Parameters of the HCN molecule.*

$B = \frac{\hbar^2}{2(\mu_{HCN}R^2 + \mu_{CN}r^2)}$ (a.u.)	μ_0 (a.u.)	α_{\parallel} (a.u.)	α_{\perp} (a.u.)
6.638×10^{-6}	1.141	20.05	8.638

general laser fields are of some interest. We begin with the second one, by considering that the electric field $\mathcal{E}(t)$ we have at our disposal is the sum of N (≤ 10) individual linearly-polarized pulses: $\mathcal{E}(t) = \sum_{n=1}^N \mathcal{E}_n(t) \sin(\omega_n t + \phi_n)$. The envelope functions $\mathcal{E}_n(t)$ are of given sine-square form,

$$\mathcal{E}_n(t) = \begin{cases} 0 & \text{if } t \leq t_{0n} \\ \mathcal{E}_{0n} \sin^2 \left[\frac{\pi}{2} \left(\frac{t-t_{0n}}{t_{1n}-t_{0n}} \right) \right] & \text{if } t_{0n} \leq t \leq t_{1n} \\ \mathcal{E}_{0n} & \text{if } t_{1n} \leq t \leq t_{2n} \\ \mathcal{E}_{0n} \sin^2 \left[\frac{\pi}{2} \left(\frac{t_{3n}-t}{t_{3n}-t_{2n}} \right) \right] & \text{if } t_{2n} \leq t \leq t_{3n} \\ 0 & \text{if } t \geq t_{3n} \end{cases} \quad (6.2.8)$$

each pulse being characterized by a set of 7 adjustable parameters, namely its frequency ω_n , relative phase ϕ_n , maximum field amplitude \mathcal{E}_{0n} , together with 4 times determining its shape (origin t_{0n} , rise time $t_{1n} - t_{0n}$, plateau $t_{2n} - t_{1n}$, and extinction time $t_{3n} - t_{2n}$). All beams

FIG. 6.2: *A typical laser field $\mathcal{E}_i(t)$.*

are polarized along the same axis. This makes a total of $7 \times 10 = 70$ parameters. It should be once more emphasized that by considering such a superposition we do not have in mind to model a situation that is experimentally feasible, but only to generate a “generic” form of signal $\mathcal{E}(t)$.

As it will be seen below, using such a generic field has one main disadvantage (in addition

to that obvious huge difficulty to minimize over \mathbb{R}^{70} !): the optimized laser field that is obtained through minimization is likely to be too difficult to analyze! Indeed, as we have very pragmatic purposes, we aim at providing the experimenter with a well identified field to generate. Obviously, a typical field obtained by such a minimization and shown on Figure 6.10 cannot be easily analyzed. Therefore, the main part of our work will be along the first strategy: restrict ourselves to a superposition of two, or at most three, different lasers of the shape of Figure 6.2 even if the price to pay for this is to lose a little on the optimality.

Apart from sticking to experimental reality (for instance a system of two lasers with the same pulsation but with two different phases is nothing else than the same laser with different optical paths), it greatly simplifies the post-treatment of results. In this view, one of our first results has been that when we use 3 lasers, i.e., when we allow for 3 different lasers in the minimization procedure, the algorithm ends up with an optimized field where the third laser has a very small amplitude (see Table 6.5 in Section 6.4). In other words, considering two lasers is enough. We shall therefore concentrate on this latter case.

6.2.3 Choice of the cost function

The cost function is the mathematical formulation of our physical goal. Its choice is so difficult in our context that it has not been done *a priori*, but has been the result of an “iterative process”. We have tested different ones and compared (on mathematical and physical bases) the results they produce. In this process, we have kept in mind the crucial following points: if a function produces (after minimization) a field which is too difficult to understand, it can be replaced however by another (possibly less) efficient that produces more understandable results. Most of the time we shall therefore handle many different cost functions, and not only one.

Basically, our physical goal is twofold:

- we want to have the molecule oriented with the field in a very good way at (at least) one time during the interval of time considered. The criterion for this purpose is:

$$J = \min_{t \in [0, T]} j(t), \quad (6.2.9)$$

- and/or we want this orientation to be kept as long as possible, even if it is not so perfect. Then the criterion to be used is:

$$J = \frac{1}{T} \int_0^T j(t) dt. \quad (6.2.10)$$

The latter criterion J is what we have written $J(\mathcal{E})$ in the equation (6.1.2). Unless otherwise mentioned, we shall deal henceforth with a criterion J that denotes either of the two criteria (6.2.9) or (6.2.10). In both formulas, let us recall that $j(t)$ is the quantity introduced in

formula (6.2.6) of Section 6.2.1 and which the orientation at time t .

In the second setting, it should be made precise that “as long as possible” typically means relatively long compared to the rotation period of the molecule, namely 11 ps for HCN, which indeed is quite a long time in our context.

In the following, we shall call “narrow” (see Figure 6.3 (a)) a function $j(t)$ produced mainly by the optimization in the first setting and “wide” (see Figure 6.3 (b)) a function $j(t)$ produced mainly in the second one. Let us also mention that a multicriteria approach is possible and

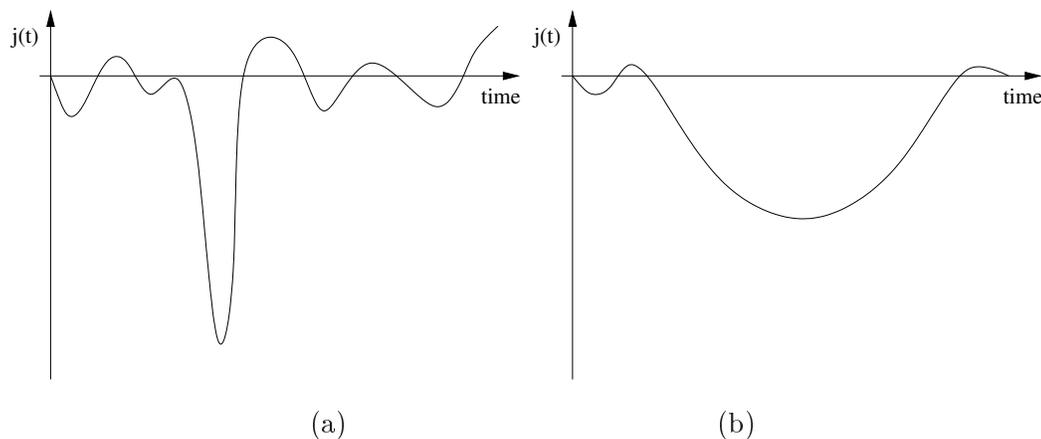


FIG. 6.3: Typical shape of an optimized $j(t)$ obtained with criterion J_1 (a) and criterion J_2 (b). They are respectively called a “narrow” and a “wide” in the text.

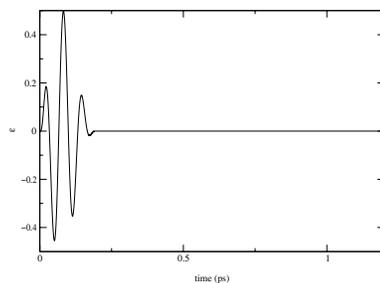
that it may possibly result in obtaining many different minima and/or the best one in some sense to be defined (see Section 6.4).

6.2.4 Identification and classification of the fields obtained

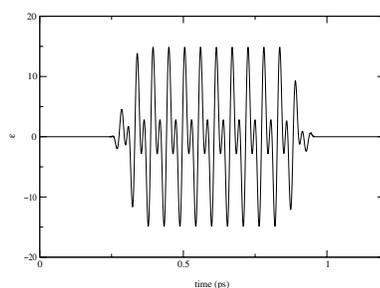
Of primary interest is the need to understand the fields produced by the optimization algorithm. It will allow one to identify the underlying main mechanisms, to imagine scenarii, and to further simplify the electric field to suggest the most simple field to be experimentally generated.

The huge number of optimization processes we have run, with different sets of parameters, with different ranges of values of these parameters, and with different criteria, has resulted in an enormous data set of optimized fields $\mathcal{E}(t)$. We believe that a good way to classify them is:

- fields of the form of a *kick* (see Figure 6.4), which is an initial sudden (of approximately 0.25 ps, *i.e.*, much shorter than the rotational period of 11 ps) and asymmetric (with respect to its sign) pulse.

FIG. 6.4: *Example of a “kick” field.*

- fields of the form $(\omega, 2\omega)$ (see Figure 6.5), which are a superposition of two laser fields with the pulsation of one being twice the pulsation of the other one.

FIG. 6.5: *Example of an $(\omega, 2\omega)$ field.*

- succession in time of two fields with a short overlay time (see Figure 6.6)
- other types of fields, apparently too complicated to be easily described.

6.3 Methodology

The way we have tackled the optimization of the orientation problem is based on two different classes of algorithms: first the gradient like algorithms, and second the evolutionary algorithms (EAs). The former ones are purely deterministic and are known to be from far the more rapidly convergent ones but present the drawback from running the risk of remaining trapped in a local minima. The latter ones are stochastic algorithms based on Artificial Darwinism. They are less sensitive to the number of local minima but as they are zero-order methods, the convergence is slower. A way to exploit the forces of both deterministic and stochastic algorithms is to use hybrid methods. We have explored one of these methods with an evolutionary algorithm using a gradient mutation operator.

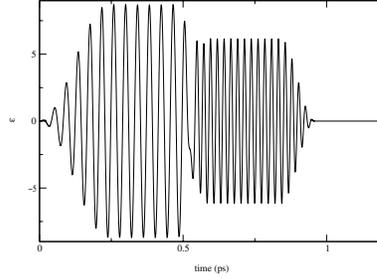


FIG. 6.6: *Example of a succession of two laser fields.*

This section presents in a first part different ways of computing the gradient of the criterion to optimize and compares the different methods. In a second part, this section briefly explains the basic steps of EAs, and next mention which purely stochastic and hybrid EAs have been used for the results presented in Section 6.4.

6.3.1 Gradient like algorithms

In this part, we present different ways to compute the gradient of the differentiable cost function $J(\mathcal{E})$, defined by (6.2.10) (the criterion (6.2.9) is not differentiable) needed for the gradient-like algorithms. We use two gradient like algorithms: the Polak-Ribière non linear conjugated gradient algorithm with Wolfe or Goldstein-Price line-search (hereafter abbreviate as PRLS) and the BFGS algorithm. For a complete presentation of these algorithms see [26, Part 1]. The most natural and the most easiest way to compute the gradient is the finite differences method, which is unfortunately very time consuming. So the need is to find another, less time consuming, way to compute the gradient. The well known adjoint method may be implemented in (at least) two ways: one can either discretize the continuous adjoint equation or one can do the adjoint calculus on the discretized form of the direct equation. It is not clear at all (at least to us) whether there is a general recipe claiming which of the two approaches is the best one. Therefore we shall test both approaches on our specific situation. In fact, the second approach (adjoint calculus on the discretized form) can be itself subdivided into two approaches: the semi-discrete approach, and the fully discrete one (see below). In addition, we shall also compare these methods with that of automatic differentiation (which in principle amounts to doing calculus on the fully discretized form of the equation, but which, in fact differs from this strategy because of implementation details). The tool we use in this latter approach is *Odyssée* [59].

We begin in Section 6.3.1.1 by presenting the continuous approach which consists in discretizing the continuous adjoint equation. Next Section 6.3.1.2 details an intermediate approach where one does the adjoint calculus on the semi-discretized equations (which means equations only discretized in time) and next discretizes in space (θ) the so-obtained adjoint problem. In

Section 6.3.1.3 we then compare this approach to the continuous one on a simplified example. In Section 6.3.1.4, we present the approach (called the discrete approach) consisting in doing adjoint calculus on the fully discretized equations (both in time and space). Finally, in Section 6.3.1.5, we present the automatic differentiation approach. The numerical results are presented in Section 6.3.1.6.

6.3.1.1 Discretization of the adjoint of the continuous problem

To find the equations satisfied by the adjoint state p , let us see the control problem as a minimization problem under the constraint $i\hbar \frac{\partial \psi}{\partial t} = H \psi$ and $\psi(t=0) = \psi_0$. We emphasize that this is only a formal method to determine the adjoint problem and to compute the gradient. We shall skip the rigorous verification that the adjoint problem we find is indeed the correct one and that it yields the correct gradient. Using definitions given by Equation (6.2.4), we will write in this section the Hamiltonian H in the form: $H = H_{rot} + H_{laser}$. We recall that only H_{laser} depends on \mathcal{E} .

Let us first introduce some definitions and notations that we will use throughout this section. For $\mathcal{E}_1, \mathcal{E}_2 \in V_t = L^2([0, T], \mathbb{R})$ we define the scalar product

$$\langle \mathcal{E}_1 | \mathcal{E}_2 \rangle_{t,C} = \int_0^T \mathcal{E}_1(t) \mathcal{E}_2(t) dt,$$

and for $\phi_1, \phi_2 \in V_\theta = L^2([0, 2\pi], \mathbb{C})$ the scalar product

$$\langle \phi_1 | \phi_2 \rangle_{\theta,C} = \int_0^\pi \Re \left(\phi_1(\theta) \overline{\phi_2(\theta)} \right) \sin \theta d\theta.$$

We also define for $\psi_1, \psi_2 \in V = L^2([0, T] \times [0, 2\pi], \mathbb{C})$ the scalar product

$$\langle \psi_1 | \psi_2 \rangle_{t,\theta,C} = \int_0^T \int_0^\pi \Re \left(\psi_1(\theta, t) \overline{\psi_2(\theta, t)} \right) \sin \theta d\theta dt.$$

The subscript C aims at recalling the ‘‘continuous’’ nature of the scalar product, in comparison with the semi-discrete or the discrete ones which will be used later on. We emphasize that when differentiating functions with complex variables we consider these functions as two-variable functions and more precisely the complex variable is taken as an element of \mathbb{R}^2 . For a given laser field \mathcal{E} , we denote by $\psi_{\mathcal{E}}$ the solution of Equation (6.1.1). Therefore, we define \tilde{J} using the criterion J as: $J(\mathcal{E}) = \tilde{J}(\psi_{\mathcal{E}})$. Thus for $\mathcal{E} \in V_t$ and $(\psi, p) \in V^2$ we write the Lagrangian L^C of the continuous problem as follows:

$$\begin{aligned} L^C(\mathcal{E}, \psi, p) &= \tilde{J}(\psi) + \left\langle \left(i\hbar \frac{\partial}{\partial t} - H_{rot} - H_{laser} \right) \psi \middle| p \right\rangle_{t,\theta,C} \\ &\quad + \langle \psi(\cdot, t=0) - \psi^0 | p(\cdot, t=0) \rangle_{\theta,C}. \end{aligned} \quad (6.3.11)$$

With standard, but tedious, calculations mainly based upon the linearity of the scalar product and that of the operators H_{rot} and H_{laser} , with an integration by part and with

$$\tilde{J}'(\psi) \cdot \delta\psi = \frac{1}{T} \int_0^T \int_0^\pi \Re [2\bar{\psi} \cos \theta \delta\psi] \sin \theta d\theta dt, \quad (6.3.12)$$

we obtain $\frac{\partial \mathbf{L}^C}{\partial \psi}(\mathcal{E}, \psi_\mathcal{E}, p) \cdot \delta\psi$, which when set to zero gives the adjoint problem

$$\begin{cases} i\hbar \frac{\partial p}{\partial t} = H_{rot}p + H_{laser}p - \frac{2}{T}\psi_\mathcal{E} \cos \theta, \\ p(T) = 0. \end{cases} \quad (6.3.13)$$

We next formally compute the gradient $\nabla^C J$ using the Lagrangian \mathbf{L}^C . When using $\psi = \psi_\mathcal{E}$ the expression of the Lagrangian is $\mathbf{L}^C(\mathcal{E}, \psi_\mathcal{E}, p) = \tilde{J}(\psi_\mathcal{E}) = J(\mathcal{E})$, thus we get

$$J'(\mathcal{E}) \cdot \delta\mathcal{E} = \frac{\partial \mathbf{L}^C}{\partial \psi}(\mathcal{E}, \psi_\mathcal{E}, p) \cdot \frac{\partial \psi_\mathcal{E}}{\partial \mathcal{E}} \cdot \delta\mathcal{E} + \frac{\partial \mathbf{L}^C}{\partial \mathcal{E}}(\mathcal{E}, \psi_\mathcal{E}, p) \cdot \delta\mathcal{E},$$

which is simplified into $J'(\mathcal{E}) \cdot \delta\mathcal{E} = \frac{\partial \mathbf{L}^C}{\partial \mathcal{E}}(\mathcal{E}, \psi_\mathcal{E}, p) \cdot \delta\mathcal{E}$ when p is the adjoint state $p_\mathcal{E}$. Therefore, the gradient $\nabla^C J = \frac{dJ}{d\mathcal{E}}$ is obtained by

$$\begin{aligned} \langle \nabla^C J | \delta\mathcal{E} \rangle_{t,C} &= \frac{\partial \mathbf{L}^C}{\partial \mathcal{E}}(\mathcal{E}, \psi_\mathcal{E}, p_\mathcal{E}) \cdot \delta\mathcal{E} \\ &= \left\langle - \left(\frac{\partial H_{laser}}{\partial \mathcal{E}}(\mathcal{E}) \cdot \delta\mathcal{E} \right) \psi_\mathcal{E} \middle| p_\mathcal{E} \right\rangle_{t,\theta,C} \\ &= \int_0^T \int_0^\pi \Re [(\mu_0 \cos \theta + \mathcal{E} [\alpha_{\parallel} \cos^2 \theta + \alpha_{\perp} \sin^2 \theta]) \psi_\mathcal{E} \bar{p}_\mathcal{E} \delta\mathcal{E}] \sin \theta d\theta dt. \end{aligned}$$

The discretization of Equation (6.2.7) is done with an operator splitting method,

$$\begin{cases} \psi^0, \\ \psi^{n+1} = e^{-\frac{i}{\hbar} \frac{\Delta t}{2} H_{laser}^n} e^{-\frac{i}{\hbar} \Delta t H_{rot}} e^{-\frac{i}{\hbar} \frac{\Delta t}{2} H_{laser}^n} \psi^n, \end{cases} \quad (6.3.14)$$

where H_{laser}^n is the time-dependent operator taken at time step t^n . Using this scheme to discretize the linear part of Equation (6.3.13) we obtain

$$\begin{cases} p^N = 0, \\ p^{n-1} = e^{\frac{i}{\hbar} \frac{\Delta t}{2} H_{laser}^n} e^{\frac{i}{\hbar} \Delta t H_{rot}} e^{\frac{i}{\hbar} \frac{\Delta t}{2} H_{laser}^n} p^n + \frac{2}{T} \psi_\mathcal{E}^n \cos \theta \frac{\Delta t}{i\hbar}. \end{cases} \quad (6.3.15)$$

In addition, we use the same schemes for the time and space discretizations as the ones used for computing $J(\mathcal{E})$. More precisely, we use for time discretization a simple Riemann rule integration scheme. For the integration in θ , the method used is the Simpson rule

$$\int_0^\pi g(\theta) d\theta = \frac{\pi}{2N} \sum_{k=0}^{2N} \alpha_k g(\theta_k)$$

$$= \frac{\Delta\theta}{3} \left[g(\theta_0) + 4 \sum_{k=0}^{N-1} g(\theta_{2k+1}) + 2 \sum_{k=0}^{N-2} g(\theta_{2k+2}) + g(\theta_{2N}) \right], \quad (6.3.16)$$

where $\Delta\theta = \frac{\pi}{2N}$ and where $(\theta_k)_{k=0,2N}$ are the equally-spaced integration points. Therefore the discretization of the gradient (6.3.14) reads, with an approximation in $(\Delta t)^2$ and in $(\Delta\theta)^4$, $\langle \nabla^C J | \delta \mathcal{E} \rangle_{t,C} =$

$$\sum_{n=0}^{N-1} \sum_{k=0}^{2M} \Re \left[\mu_0 \cos \theta_k + \mathcal{E}^n [\alpha_{\parallel} \cos^2 \theta_k + \alpha_{\perp} \sin^2 \theta_k] \psi_k^n \bar{p}_k^n \right] \delta \mathcal{E}^n \Delta t \alpha_k \sin \theta_k \Delta \theta. \quad (6.3.17)$$

6.3.1.2 Adjoint calculus on the semi-discretized equations

The discretization of the time-dependent Schrödinger Equation (6.2.7) is given by (6.3.14) while the discretization of the criterion (again by the Riemann rule integration scheme) yields the semi-discrete Lagrangian

$$\begin{aligned} \mathbb{L}^{SD}(\mathcal{E}, \Psi, P) &= \frac{1}{T} \sum_{n=0}^{N-1} \int_0^{\pi} \|\psi^n\|_{\mathbb{C}}^2 \cos \theta \sin \theta d\theta \Delta t + \langle \psi^0 - \psi_0 | p^0 \rangle_{\theta,C} \\ &\quad + \left\langle \Psi^S - e^{-\frac{i}{\hbar} \frac{\Delta t}{2} H_{laser}} e^{-\frac{i}{\hbar} \Delta t H_{rot}} e^{-\frac{i}{\hbar} \frac{\Delta t}{2} H_{laser}} \Psi \middle| P \right\rangle_{t,\theta,SD}, \end{aligned} \quad (6.3.18)$$

where $\Psi^S = (\psi^1, \dots, \psi^N)$, $\Psi = (\psi^0, \dots, \psi^{N-1})$ and $P = (p^0, \dots, p^{N-1})$ are elements of $(V_{\theta})^N$ and where $\mathcal{E} = (\mathcal{E}^0, \dots, \mathcal{E}^{N-1})$ is an element of \mathbb{R}^N . The scalar product $\langle \cdot | \cdot \rangle_{\theta,C}$ is the one given in the previous section and the scalar product $\langle \cdot | \cdot \rangle_{t,\theta,SD}$ is given by $\langle \Psi_1 | \Psi_2 \rangle_{t,\theta,SD} = \sum_{n=1}^N \langle \psi_1^n | \psi_2^n \rangle_{\theta,C} \Delta t$ with $\Psi_1, \Psi_2 \in (V_{\theta})^N$. We also define for $\mathcal{E}_1, \mathcal{E}_2 \in \mathbb{R}^n$ the scalar pro-

duct $\langle \mathcal{E}_1 | \mathcal{E}_2 \rangle_{t,SD} = \sum_{n=1}^N \mathcal{E}_1^n \mathcal{E}_2^n$. For a given laser field \mathcal{E} we denote $\Psi_{\mathcal{E}}$ the solution of Equation (6.3.14). As in the previous section, by computing $\frac{\partial \mathbb{L}^{SD}}{\partial \Psi}(\mathcal{E}, \Psi_{\mathcal{E}}, P) \cdot \delta \Psi$ and then by setting it to zero we get the following discrete adjoint problem:

$$\begin{cases} p^{N-1} = 0, \\ p^{n-1} = e^{\frac{i}{\hbar} \frac{\Delta t}{2} H_{laser}^n} e^{\frac{i}{\hbar} \Delta t H_{rot}} e^{\frac{i}{\hbar} \frac{\Delta t}{2} H_{laser}^n} p^n - \frac{2}{T} \psi_{\mathcal{E}}^n \cos \theta. \end{cases} \quad (6.3.19)$$

And the gradient is obtained by:

$$\begin{aligned} &\frac{\partial \mathbb{L}^{SD}}{\partial \mathcal{E}}(\mathcal{E}, \Psi_{\mathcal{E}}, P_{\mathcal{E}}) \cdot \delta \mathcal{E} \\ &= \sum_{n=0}^{N-1} \left\langle -\frac{i \Delta t}{\hbar} \frac{\partial H_{laser}^n}{\partial \mathcal{E}_n} \delta \mathcal{E}^n e^{-\frac{i}{\hbar} \frac{\Delta t}{2} H_{laser}^n} e^{-\frac{i}{\hbar} \Delta t H_{rot}} e^{-\frac{i}{\hbar} \frac{\Delta t}{2} H_{laser}^n} \psi^n \middle| p^n \right\rangle_{\theta,C} \\ &= \sum_{n=0}^{N-1} \left\langle -\frac{i \Delta t}{\hbar} \frac{\partial H_{laser}^n}{\partial \mathcal{E}_n} \delta \mathcal{E}^n e^{-\frac{i}{\hbar} \frac{\Delta t}{2} H_{laser}^n} \psi^{n+1} \middle| p^n \right\rangle_{\theta,C}. \end{aligned} \quad (6.3.20)$$

Thus, with an approximation at the order $(\Delta\theta)^4$, we obtain

$$\begin{aligned} \langle \nabla^{SD} J | \delta \mathcal{E} \rangle_{t,SD} = \\ - \sum_{n=0}^{N-1} \sum_{k=0}^{2M} \Re \left[\frac{i}{\hbar} \psi_k^{n+1} \bar{p}_k^n (\mu_0 \cos \theta_k + \mathcal{E}^n [\alpha_{\parallel} \cos^2 \theta_k + \alpha_{\perp} \sin^2 \theta_k]) \delta \mathcal{E}^n \right] (\Delta t)^2 \sin \theta_k \alpha_k \Delta \theta. \end{aligned} \quad (6.3.21)$$

6.3.1.3 Comparison of the continuous and the semi-discretized approaches

In order to understand which of the formulae (6.3.17) or (6.3.21) is more accurate, we give below some illustrative example. Although very basic, this example allows one to understand the fundamental difference between formulae (6.3.17) and (6.3.21). Let us argue on the following Schrödinger equation:

$$\begin{cases} i\hbar \frac{\partial \psi}{\partial t} = \psi \mathcal{E}(t) \cos \theta, \\ \psi(0) = \psi^0, \end{cases} \quad (6.3.22)$$

(obtained by simply setting H_0 to zero in (6.2.7)) with the criterion written in the form

$$J(\mathcal{E}) = \tilde{J}(\psi) = \frac{1}{T} \int_0^T \int_0^{\pi} f(\psi) \sin \theta d\theta dt.$$

The first way to proceed is the one we have followed in Section 6.3.1.1, namely by discretizing the adjoint equation. For Equation (6.3.22), basic calculus shows that the adjoint equation is given by

$$\begin{cases} i\hbar \frac{\partial p}{\partial t} = \mathcal{E}(t) p \cos \theta - \bar{f}'(\psi \mathcal{E}), \\ p(T) = 0, \end{cases} \quad (6.3.23)$$

which once discretized with the same scheme as the one used for the direct equation, yields

$$\begin{cases} p^N = 0, \\ p^{n-1} = e^{\frac{i}{\hbar} \frac{\Delta t}{2} H_{laser}^n} e^{\frac{i}{\hbar} \Delta t H_{rot}} e^{\frac{i}{\hbar} \frac{\Delta t}{2} H_{laser}^n} p^n - \bar{f}'(\psi_{\mathcal{E}}^n) \frac{\Delta t}{i\hbar}. \end{cases} \quad (6.3.24)$$

We compute the gradient of the criterion,

$$\nabla J(\mathcal{E}) = \frac{1}{T} \int_0^T \int_0^{\pi} f'(\psi \mathcal{E}) \frac{\partial \psi}{\partial \mathcal{E}} \sin \theta d\theta dt,$$

where $\frac{\partial \psi}{\partial \mathcal{E}}$ solves

$$\begin{cases} i\hbar \frac{\partial}{\partial t} \left(\frac{\partial \psi}{\partial \mathcal{E}} \right) = \psi \cos \theta + \mathcal{E}(t) \cos \theta \frac{\partial \psi}{\partial \mathcal{E}}, \\ \left. \frac{\partial \psi}{\partial \mathcal{E}} \right|_{t=0} = 0. \end{cases} \quad (6.3.25)$$

Thus, by using Equation (6.3.23) and integration by part we obtain

$$\langle \nabla J | \delta \mathcal{E} \rangle_{t,C} = - \int_0^T \left(\int_0^\pi \psi \bar{p} \cos \theta \sin \theta d\theta \right) \delta \mathcal{E} dt. \quad (6.3.26)$$

We now discretize this integral by the Riemann scheme which yields

$$\int_0^T \left(\int_0^\pi \psi \bar{p} \cos \theta \sin \theta d\theta \right) \delta \mathcal{E} dt = \sum_{n=0}^{N-1} \left(\int_0^\pi \psi^n \bar{p}^n \cos \theta \sin \theta d\theta \right) \delta \mathcal{E}^n \Delta t,$$

and thus the following approximation of the gradient:

$$\langle \nabla J(\psi) | \delta \mathcal{E} \rangle_{t,C} = - \sum_{n=0}^{N-1} \left(\int_0^\pi \psi^n \bar{p}^n \cos \theta \sin \theta d\theta \right) \delta \mathcal{E}^n \Delta t, \quad (6.3.27)$$

is the exact analogous of formula (6.3.17). Using this Riemann discretization scheme, the numerical error is controlled by the following estimate:

$$\left| \int_0^T g(t) dt - \sum_{n=0}^{N-1} g^n \Delta t \right| \leq T \Delta t \|g'\|_{L^\infty}. \quad (6.3.28)$$

Applying this result to $g = \left(\int_0^\pi \psi \bar{p} \cos \theta \sin \theta d\theta \right) \delta \mathcal{E}$, we obtain the control of the numerical error of the approximation (6.3.27) of the gradient

$$\begin{aligned} |\varepsilon_{\Delta t}^C| &\leq T \Delta t \left\| \frac{\partial}{\partial t} \left[\left(\int_0^\pi \psi \bar{p} \cos \theta \sin \theta d\theta \right) \delta \mathcal{E} \right] \right\|_{L^\infty} \\ &\leq T \Delta t \left\| \left[\frac{\partial}{\partial t} \left(\int_0^\pi \psi \bar{p} \cos \theta \sin \theta d\theta \right) \right] \delta \mathcal{E} + \left(\int_0^\pi \psi p \cos \theta \sin \theta d\theta \right) \frac{\partial}{\partial t} (\delta \mathcal{E}) \right\|_{L^\infty} \\ &\leq CT \Delta t \left(\|\delta \mathcal{E}\|_{L^\infty} + \left\| \frac{\partial}{\partial t} (\delta \mathcal{E}) \right\|_{L^\infty} \right), \end{aligned} \quad (6.3.29)$$

where the constant C depends on norms of $\psi|_{t=0}$ and \mathcal{E} but not on $\delta \mathcal{E}$.

On the other hand, if we now discretize the equation and the criterion, we obtain as in (6.3.20)

$$\langle \nabla J | \delta \mathcal{E} \rangle_{t,SD} = \sum_{n=0}^{N-1} \left(\int_0^\pi \psi^{n+1} p^n \cos \theta \sin \theta d\theta \right) \delta \mathcal{E}^n \Delta t. \quad (6.3.30)$$

Applying the same numerical analysis, we see that the error in the approximation of the gradient is now obtained by setting $g = \int_0^\pi f'(\psi) \delta \psi \sin \theta d\theta$ in (6.3.28), which yields

$$|\varepsilon_{\Delta t}^{SD}| \leq T \Delta t \left\| \int_0^\pi \left(\frac{\partial}{\partial t} [f'(\psi) \delta \psi] \right) \sin \theta d\theta \right\|_{L^\infty}. \quad (6.3.31)$$

Now

$$\frac{\partial}{\partial t} (f'(\psi) \delta \psi) = f''(\psi) \frac{\partial \psi}{\partial t} \delta \psi + f'(\psi) \frac{\partial (\delta \psi)}{\partial t},$$

where $\frac{\partial \psi}{\partial t} = \frac{1}{i\hbar} (\mathcal{E}\psi \cos \theta)$ and $\frac{\partial (\delta \psi)}{\partial t} = \frac{1}{i\hbar} ((\delta \mathcal{E}) \psi \cos \theta + \mathcal{E} x (\delta \psi))$. It follows that

$$\left\| \int_0^\pi \left(\frac{\partial}{\partial t} [f'(\psi) \delta \psi] \right) \sin \theta d\theta \right\|_{L^\infty} \leq C \|\delta \mathcal{E}\|_{L^\infty},$$

where C only depends on norms on $\psi|_{t=0}$ and \mathcal{E} . Therefore

$$|\varepsilon_{\Delta t}^{SD}| \leq CT \Delta t \|\delta \mathcal{E}\|_{L^\infty}. \quad (6.3.32)$$

Comparing this estimate to (6.3.29), we see that the control in (6.3.32) is better, in particular for variations $\delta \mathcal{E}$ of \mathcal{E} that have large variations in time, which will precisely be the case for us (oscillatory laser fields). It is therefore expected that *in our case* the adjoint calculus on the discrete equation will yield a better accuracy for the computation of the gradient than the approach consisting in discretizing the continuous adjoint equation. Let us emphasize that the main difference between the two approaches is the following formal (non rigorous) integration by parts:

$$\int_0^T f'(\psi) \delta \psi \approx \int_0^T \left(\frac{\partial p}{\partial t} \right) \delta \psi \approx \int_0^T p \left(\frac{\partial \delta \psi}{\partial t} \right) \approx \int_0^T p \psi \delta \mathcal{E},$$

which is done before or after discretization and thus allows one to have the control of the error basically either by

$$\frac{\partial}{\partial t} (f'(\psi) \delta \psi) \approx f'(\psi) \delta \mathcal{E} \quad (6.3.33)$$

or by

$$\frac{\partial}{\partial t} (p \psi \delta \mathcal{E}) \approx p \psi \frac{\partial}{\partial t} (\delta \mathcal{E}). \quad (6.3.34)$$

In the case (6.3.33) the numerical error of integration is reported on ψ and $\delta \psi$ while in the case (6.3.34) the numerical error of integration is directly reported on $\delta \mathcal{E}$. Figure 6.7 summarizes the main ideas presented here.

6.3.1.4 Adjoint calculus on the fully discretized equations

In this section we begin by discretizing Equation (6.2.7) both in time and in θ -space and then do the adjoint calculus. The numerical propagation of the θ operator H_θ and the laser operator H_{laser} can be written in the matrix form

$$\Psi^{n+1} = A_\theta^n B A_\theta^n \Psi^n,$$

where Ψ^n is the vector (ψ_k^n) , where A_θ^n is the diagonal matrix of the laser operator propagation, and where B is the matrix corresponding to the θ operator propagation. Only the matrix A_θ^n depends on the laser field \mathcal{E} .

We write the discrete Lagrangian as follows:

$$\begin{aligned} \mathbb{L}^D(\mathcal{E}, \underline{\Psi}, \underline{P}) &= \sum_{n=0}^{N-1} \sum_{k=0}^{2M} \frac{1}{T} \|\psi_k^n\|_{\mathbb{C}}^2 \cos \theta \sin \theta \alpha_k \Delta \theta \Delta t \\ &+ \langle \underline{\Psi}^S - \underline{\Psi}^M | \underline{P} \rangle_{t,D} + \langle \Psi^0 - \Psi_0 | P^0 \rangle_{\theta,D}, \end{aligned} \quad (6.3.35)$$

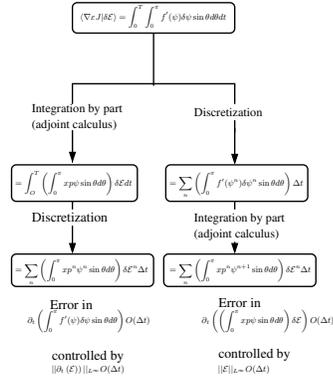


FIG. 6.7: Comparison of the two approaches of Section 6.3.1.1 and 6.3.1.2 to compute the gradient.

where

$$\underline{\underline{\Psi}}^S = (\Psi^1, \dots, \Psi^N),$$

$$\underline{\underline{\Psi}}^M = (A_\theta^0 B A_\theta^0 \Psi^0, \dots, A_\theta^{N-1} B A_\theta^{N-1} \Psi^{N-1}),$$

and

$$\underline{\underline{P}} = (P^0, \dots, P^{N-1})$$

are elements of $(\mathbb{R}^{2M})^N$ and where $\mathcal{E} = (\mathcal{E}^0, \dots, \mathcal{E}^{N-1})$ is an element of \mathbb{R}^N .

The scalar product $\langle \cdot | \cdot \rangle_{t,D}$ is given for $\underline{\underline{\Psi}}_1, \underline{\underline{\Psi}}_2 \in (\mathbb{R}^{2M})^N$ by

$$\langle \underline{\underline{\Psi}}_1 | \underline{\underline{\Psi}}_2 \rangle_{t,D} = \sum_{n=1}^N \langle \Psi_1^n | \Psi_2^n \rangle_{\theta,D} \Delta t$$

with $\langle \Psi_1 | \Psi_2 \rangle_{\theta,D} = \sum_{n=0}^{2M} \Re(\psi_{1k} \overline{\psi_{2k}}) \alpha_k \sin \theta \Delta \theta$ when $\Psi_1, \Psi_2 \in \mathbb{R}^{2M}$.

Therefore, equation $\frac{\partial \mathcal{L}^D}{\partial \underline{\underline{\Psi}}}(\mathcal{E}, \underline{\underline{\Psi}}_\mathcal{E}, \underline{\underline{P}}) \cdot \delta \underline{\underline{\Psi}} = 0$, we get

$$\begin{cases} P^{N-1} = 0 \\ p_k^{n-1} = \left[(\overline{A_\theta^n} \overline{B} A_\theta^n)^T P^n \right]_k - \alpha_k \frac{2}{T} \psi_k^n \cos \theta. \end{cases} \quad (6.3.36)$$

The gradient is obtained by

$$\frac{\partial \mathcal{L}^D}{\partial \mathcal{E}}(\mathcal{E}, \underline{\underline{\Psi}}_\mathcal{E}, \underline{\underline{P}}_\mathcal{E}) \cdot \delta \mathcal{E}$$

$$= - \left\langle \left(\frac{\partial \underline{\underline{\Psi}}^M}{\partial \mathcal{E}} \right) \cdot \delta \mathcal{E} | \underline{\underline{P}} \right\rangle_{t,D}$$

$$= - \sum_{n=0}^{N-1} \sum_{k=0}^{2M} \Re \left(\left[\left(\frac{\partial A_\theta^n}{\partial \mathcal{E}^n} B A_\theta^n + A_\theta^n B \frac{\partial A_\theta^n}{\partial \mathcal{E}^n} \right) \Psi^n \right]_k \bar{p}_k^n \delta \mathcal{E}^n \right) \alpha_k \sin \theta \Delta \theta \Delta t,$$

where $\frac{\partial A_\theta^n}{\partial \mathcal{E}^n}$ is the matrix obtained by differentiating the matrix A_θ^n . We obtain for the gradient formula

$$\langle \nabla^D J | \delta \mathcal{E} \rangle_{t,D} =$$

$$\sum_{n=0}^{N-1} \sum_{k=0}^{2M} \Re \left[(\mu_0 \cos \theta_k + \mathcal{E}^n [\alpha_{\parallel} \cos^2 \theta_k + \alpha_{\perp} \sin^2 \theta_k]) \widetilde{\psi}_k^n \bar{p}_k^n \delta \mathcal{E}^n \right] \alpha_k \sin \theta \Delta \theta \Delta t,$$

where $\widetilde{\Psi}^n = \left(\frac{\partial A_\theta^n}{\partial \mathcal{E}^n} B A_\theta^n + A_\theta^n F^{-1} A B F \frac{\partial A_\theta^n}{\partial \mathcal{E}^n} \right) \Psi^n$. This formula is to be compared with (6.3.17) and (6.3.21).

6.3.1.5 Computing the gradient using Automatic Differentiation tools

In this section we briefly present another method to compute the gradient, which uses the Automatic Differentiation tool *Odyssée* [66, 59]. Automatic Differentiation tools can be seen as black boxes taking as input a program computing a cost function $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ and giving as output another program computing the gradient $\frac{\partial f(x)}{\partial x}$.

Odyssée is able to use two modes: the tangent mode and the adjoint mode, which is similar to the adjoint method. We emphasize that the cost of the gradient computation is proportional to n with the tangent mode (as with finite differences) and it is proportional to m with the adjoint mode. Thus, the tangent mode has to be used when $n \ll m$ and the adjoint mode has to be used when $n \gg m$. For our problem we have $m = 1$ and $1 \ll n \leq 70$, so we use only the adjoint mode. As in the direct program we have 50 000 iterations, the adjoint program needs a lot of memory to run. In order to reduce the size of memory needed, we have modified the adjoint program by deleting the temporary variables in the linear parts of the program. Table 6.2 gives an idea of the size of the direct code and that of the adjoint code. The calculation times refer to a Pentium II, 466 Mhz Celeron with 128 Mb RAM running with Linux.

TAB. 6.2: *Technical requirements with Odyssée.*

		standard	post-processed
	direct code	adjoint code	adjoint code
Size (lines)	433	2075	1190
Memory needed	12 Ko	520 Mo	103 Mo
Time (CPU)	60 s	—	141 s

6.3.1.6 Numerical results

The purpose of this section is to compare numerically the different methods presented above for computing the gradient. For the numerical tests we use one laser field of the form $\mathcal{E}(t) = E \sin(\omega t + \phi)$. The gradient with respect to the parameters E , ω and ϕ is denoted as

$$\nabla J = \begin{pmatrix} \nabla_E J \\ \nabla_\omega J \\ \nabla_\phi J \end{pmatrix}.$$

We have computed the gradient using the methods presented in the previous sections, more precisely the continuous approach (C), the semi-discrete approach (SD), the discrete approach (D), and *Odyssée* (AD). We have also computed the gradient using the finite differences approach (FD), where for each variable x ($x = E, \omega, \phi$)

$$\nabla_x^{FD} J = \lim_{\delta x \rightarrow 0} \frac{J(x + \delta x) - J(x)}{\delta x}.$$

The gradient given by FD has been computed with different values of δx to make sure that we have reached the $\delta x \rightarrow 0$ limit. Next we compare the gradient obtained using the different approaches with the gradient obtained using the finite differences approach, which is therefore taken as a reference value. For each method we will compute the relative error

$$e_{E,\omega,\phi} = \left| \frac{\nabla_{E,\omega,\phi} J - \nabla_{E,\omega,\phi}^{FD} J}{\nabla_{E,\omega,\phi}^{FD} J} \right|.$$

The comparison is done for both low and high frequencies, using two different representative points

$(E, \omega, \phi) = (10^{11} \text{ W/cm}^2, 500 \text{ cm}^{-1}, 0)$ and $(E, \omega, \phi) = (10^{11} \text{ W/cm}^2, 4000 \text{ cm}^{-1}, 0)$, respectively.

Table 6.3 shows that all the methods we have presented in this section give good results compared to the finite differences method. We can also see on this table that the best results are obtained using *Odyssée*, where we have a better precision than with the other methods. In general the precision is increased by at least one order.

We can also see on this table that the results agree with the comparison we made in Section 6.3.1.3 between the continuous approach and the semi-discrete approach, except for the ∇_E component.

Let us now take the results given by automatic differentiation as a reference and make the same comparison with the other methods as we have done with the finite differences approach. On Table 6.4 we see that, compared to the AD approach, the best results are those given by the discrete approach (again except for the component ∇_E). We recall that with the discrete approach we make the adjoint calculus on the fully discretized equation and that the

automatic differentiation tools make also adjoint calculus on the fully discretized equation with some implementation differences. We also emphasize that for the component ∇_E we obtain results which are different from the results we obtain with the components ∇_ω and ∇_ϕ . We still unable to explain such a difference.

In practice, the size of the parameter vector for our problem can go up to 70, so we can only use an adjoint based method and not the finite differences one. Indeed the CPU time needed to compute the gradient depends on the parameter vector size for the finite differences method and is independent of this size for the other methods presented below. More precisely, the CPU time needed for these other methods is about 3 times the one needed to compute the criterion. For implementation, the continuous approach and the semi-discrete approach are easiest to implement than the discrete approach. Finally, for *Odyssée*, let us recall that even if it gives automatically the gradient, some post-processing of the adjoint code is needed before running it.

TAB. 6.3: *Relative error, with respect to the FD, of the gradient.*

(a) : gradient computed at $(E, \omega, \phi) = (10^{11}, 500, 0)$					
	FD	AD	C	SD	D
e_E	0.	$25. \times 10^{-8}$	$89. \times 10^{-6}$	$20. \times 10^{-3}$	$41. \times 10^{-3}$
e_ω	0.	$92. \times 10^{-6}$	$13. \times 10^{-4}$	$47. \times 10^{-5}$	$32. \times 10^{-5}$
e_ϕ	0.	$18. \times 10^{-6}$	$81. \times 10^{-5}$	$25. \times 10^{-5}$	$25. \times 10^{-5}$
(b) : gradient computed at $(E, \omega, \phi) = (10^{11}, 4000, 0)$					
	FD	AD	C	SD	D
e_E	0.	$75. \times 10^{-5}$	$68. \times 10^{-5}$	$20. \times 10^{-3}$	$40. \times 10^{-3}$
e_ω	0.	$11. \times 10^{-5}$	$76. \times 10^{-4}$	$51. \times 10^{-4}$	$25. \times 10^{-4}$
e_ϕ	0.	$22. \times 10^{-3}$	$41. \times 10^{-3}$	$33. \times 10^{-3}$	$26. \times 10^{-3}$

TAB. 6.4: *Relative error, with respect to the AD, of the gradient.*

(a) : gradient computed at $(E, \omega, \phi) = (10^{11}, 500, 0)$					
	FD	AD	C	SD	D
e_E	$25. \times 10^{-8}$	0.	$89. \times 10^{-6}$	$20. \times 10^{-3}$	$41. \times 10^{-3}$
e_ω	$92. \times 10^{-6}$	0.	$14. \times 10^{-4}$	$57. \times 10^{-5}$	$22. \times 10^{-5}$
e_ϕ	$18. \times 10^{-6}$	0.	$79. \times 10^{-5}$	$23. \times 10^{-5}$	$27. \times 10^{-5}$
(b) : gradient computed at $(E, \omega, \phi) = (10^{11}, 4000, 0)$					
	FD	AD	C	SD	D
e_E	$75. \times 10^{-5}$	0.	$73. \times 10^{-6}$	$20. \times 10^{-3}$	$41. \times 10^{-3}$
e_ω	$11. \times 10^{-5}$	0.	$75. \times 10^{-4}$	$50. \times 10^{-4}$	$24. \times 10^{-4}$
e_ϕ	$22. \times 10^{-3}$	0.	$18. \times 10^{-3}$	$10. \times 10^{-3}$	$36. \times 10^{-4}$

6.3.2 Evolutionary Algorithms

This section presents the stochastic algorithms that we have used for the orientation problem which belong to the family of Evolutionary Algorithms (EAs). Their common feature is to imitate the principle of natural evolution. This section is organized as follows: Section 6.3.2.1 briefly introduces EAs and their basic terminology and gives also a short state of the art in EAs while Section 6.3.2.2 presents more precisely the EAs that we have implemented for the orientation problem.

6.3.2.1 Introduction to Evolutionary Algorithms

This section briefly explains the basic steps of an EA. The problem is to optimize a given *objective function* f over a given search space. A population of individuals (i.e., a P-uple of points in the search space) undergoes some artificial Darwinian evolution based on the *fitness* F of each individual. The fitness of an individual is directly related to the value of the objective function of this individual (a typical example of a fitness function is the objective function itself, denoted by J in our work). The evolution operators applied to the individuals are defined upon the so-called *genotype space* noted E . It may be different from the definition space of the fitness called the *phenotype space*. The choice of this genotype space is the *representation*.

Figure 6.8 illustrates the framework of an EA: after an initialization of the population (generally a uniform random initialization) the fitness of each individual is computed. This is the *evaluation step*. Then, the loop of the algorithm called *a generation* is made up of the following steps:

- *Stopping criterion*: a basic stopping criterion is when the maximum number of generations fixed by the user is reached.
- *Selection*: the selection operator selects among the parents those who will generate offsprings, the genitors. There exists several selection operators, either of deterministic or of stochastic type. All of them are based on the fitness of the individuals and implement the first phase of Artificial Darwinism: the fittest allowed to reproduce.
- *Creation of new individuals*: there are basically two ways to create new individuals in the population from the genitors, namely the *crossover* and the *mutation*. These variation operators are stochastic operators: the crossover is a stochastic operator from E^k into E (typically $k = 2$), it is a recombination of k parents, and the mutation operator is a stochastic operator from E into E .
- *Evaluation*: for each offspring the fitness is computed.
- *Replacement*: this operator discriminate among the individuals of the current population those who will be the parents for the next generation. This operator, like the

selection operator, is based on the fitness of the individuals and implements the second step of Darwin's theory: survival of the fittest.

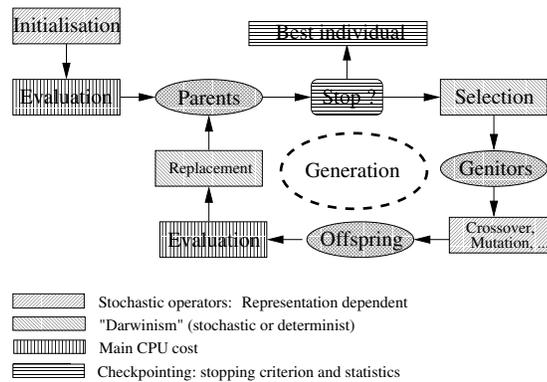


FIG. 6.8: *General EA scheme.*

Despite the common features of all EAs, several trends can be discriminated, mainly due to historical differences. We will only detail here the instances of EAs we have been using, referring to [18] and references therein for a complete description. The four main branches are (in alphabetical order):

- The evolutionary programming (EP), originally developed in California to evolve finite state machines.
- The evolution strategies (ESs) developed in Germany to solve numerical optimization problems for real search spaces. The genotype space is the phenotype space, namely a subset of \mathbb{R}^N . A precise description is given below.
- The genetic algorithms (GAs) developed in Michigan to study some adaptation mechanisms of populations for biology. These algorithms have later been used for optimization problems. More precision are given below.
- The genetic programming (GP), which has appeared more recently, consists in evolving tree structures.

The Canonical GA:

The genotype space is $\{0, 1\}^n$, the selection operator is the so-called roulette wheel, where the probability P_{X_p} to select the individual X_p is proportional to the fitness $F(X_p)$:

$$P_{X_p} = \frac{F(X_p)}{\sum_{i \in Population} F(X_i)}.$$

The crossover operator replaces some bits in the first parent string by the corresponding bits from the second parent, and the mutation operator randomly flips a bit of the parent. The replacement is *generational*: the offsprings at the generation n become the parents of the generation $n + 1$. Modern GAs are commonly used with any kind of representation as long as crossover and mutation can be defined.

The ES:

The ES [132] have been designed to optimize real functions, thus the natural search space is \mathbb{R}^N . The individuals undergo Gaussian mutations, namely addition of zero-mean Gaussian variables of standard deviation σ . The particularity of ES is that the parameter σ is a part of the genetic information. For a so-called isotropic ES, an individual is of the form $I = (x_1, \dots, x_N, \sigma)$ and, for a non isotropic ES, $I = (x_1, \dots, x_N, \sigma_1, \dots, \sigma_N)$ (there also exists a third type of ES not discussed here, the correlated ES). Consequently the mutation parameters are subjected to recombination and mutation as well. More precisely, the *adaptive mutation* takes place in two steps, first a mutation of the mutation parameters, second a mutation of object variables x_i . For an isotropic ES the two steps are

$$\begin{aligned}\sigma^{(t+1)} &= \sigma^{(t)} \exp(\tau_0 N(0, 1)), \\ x_i^{(t+1)} &= x_i^{(t)} + N_i(0, \sigma^{(t+1)}),\end{aligned}$$

and, for a non isotropic ES,

$$\begin{aligned}\sigma_i^{(t+1)} &= \sigma_i^{(t)} \exp(\tau_0 N(0, 1) + \tau N_i(0, 1)), \\ x_i^{(t+1)} &= x_i^{(t)} + N_i(0, \sigma_i^{(t+1)}),\end{aligned}$$

where $N(0, 1)$ stands for a Gaussian random variable. The crossover operator selects randomly two parents, $(x_1^1, \dots, x_N^1, \sigma_1^1, \dots, \sigma_N^1)$ and $(x_1^2, \dots, x_N^2, \sigma_1^2, \dots, \sigma_N^2)$, to produce an offspring $(x_1^{q_1}, \dots, x_N^{q_N}, \sigma_1^{q_1}, \dots, \sigma_N^{q_N})$ where $q_i = 1$ or $q_i = 2$ with equal probability. This crossover operator can also involve all individuals in the population, this is a *global crossover*. The replacement operator is strictly deterministic, based on the rank. For example, if μ (respectively λ) is the number of parents (respectively offsprings), $(\mu, \lambda) - ES$ selects the parents for next generation by taking the μ best offsprings and $(\mu + \lambda) - ES$ selects the parents for next generation by taking the μ best among the λ offsprings and μ parents.

It is now commonly accepted that the incorporation of specific knowledge, of the problem to optimize, by means of representation and specific operators, is the best way and the only way to enhance the performances of an EA. But, when using an EA without introducing some specificities of the problem, ES is generally the most efficient EA for parametric optimization. ES, like GA, are implemented on the EOlib class library available from [57].

6.3.2.2 The algorithms used

The orientation problem is a minimization problem on a real space of size $7N$, where N is the number of laser fields to superpose. We use two kinds of EAs: the first one is based on a classical GA with a real representation (roulette wheel selection and *barycentric* or *multi-point* crossover [102]) and the second one is the ES described above and taken from EOLib [57].

The first algorithm is an improved GA, adding some specific operators and some specific features, which are known to improve the performances of GAs. We will name this algorithm in the sequel EGA (for Enhanced GA). *Niching* and *Rescaling* are two specific features of this algorithm. Rescaling is a way to avoid some bias in the roulette wheel selection; niching is to avoid that all the population concentrates on a region of the search space (see [102] for more precisions). Then, the mutation strength on EGA decreases with the number of generations. A specific gradient mutation operator is also used (EGA-CG), replacing the parent by the result of a few iterations of a conjugated gradient algorithm using the parent as initial value. The purpose of such an operator is to accelerate the convergence by taking advantages of a gradient algorithm.

We have tested EGA, EGA-CG and ES on several test functions taken from the literature (Sphere, Rosenbrock and Shekel functions). We refer to [154] for the details. We present here shortly some conclusions of these tests. First, for all the functions tested, a comparison with a classical GA has shown that EGA, EGA-CG, and ES converge more often, and faster than GA. Moreover, they are able to improve continuously their precision whereas GA stops at some non-zero distance of the solution. Second, the tests have confirmed that the gradient mutation operator accelerates the convergence, except for too “chaotic” functions. Third, the test cases have helped us for a crucial point of EAs, namely the setting of the parameters, which is specific to each function. Several trends can be discriminated for the setting, mainly taken from the literature and confirmed with test cases. As we have built our own EGA, it is difficult to give succinctly the parameters to set. With respect to ES, three important steps are given: First, the probability to mutate an individual is greater than the probability to cross two individuals (typically $p_{mut} = 0.8$ and $p_{cross} = 0.2$). Second, the size of the population is typically $(7, 49)$ – *ES* and the number of parents should be increased if the number of local minima increases. Third, the initial mutation strength σ should also be increased when the number of local minima increases.

6.4 Results for the orientation problem

A preliminary study of the orientation problem with the purely deterministic PRLS and BFGS algorithms (see Section 6.3.1), for the differentiable criterion (6.2.10), showed the need to use stochastic methods. Indeed, these algorithms converge after a few iterations towards a

local minimum close to the initial guess: the cost function presents numerous local minima. We know from the literature and from test cases that for such functions, ES, EGA, and EGA-CG perform better. As far as EGA-CG is concerned, using it to minimize the criterion (6.2.10) does not improve the results in a significant way. More precisely, our best results have been obtained without the gradient mutation operator. However, using the gradient algorithm after EGA can improve the result of the optimization, as we will see in Section 6.4.1.

We present in this section the main results obtained with our algorithms on the orientation problem. The sequel of this section is organized as follows: in Section 6.4.1, we give the fields we have obtained by minimizing criteria (6.2.9) and (6.2.10). For both criteria we give the best results. Next, we explain how the addition of the CG at the end of the EGA improves the optimization of the criterion (6.2.10). In Section 6.4.2, we introduce a new hybrid criterion in order to approach both goals of Section 6.2.3: obtaining at some given time a good orientation and keeping it as long as possible. Then, in Section 6.4.3, we present results obtained by a different form of laser fields. This form of laser field, named a *train of kicks* is a succession of fields of *kick* form presented in Section 6.4.1. As we will see, these fields really improve the results of Section 6.4.1 on both criteria (6.2.9) and (6.2.10).

6.4.1 Optimized fields for (6.2.9) and (6.2.10)

All the results presented in the sequel have been obtained by optimizing upon a superposition of two or three lasers in order to better understand the physical meaning of the results. Indeed, our trials for optimizing (6.2.10) on a superposition of ten laser fields have given results shown on Figure 6.10, which are not sufficiently easy to understand and interpret. We have therefore left this strategy aside.

Figures 6.11 and 6.12 show the optimized fields and their instantaneous criterion $j(t)$ obtained respectively with criteria (6.2.9) and (6.2.10). They have been obtained with a non-isotropic ES and with EGA, respectively. However, let us emphasize that the two algorithms give similar results. Indeed, EGA has given fields and instantaneous criterion of the same form as the ones shown on Figure 6.11 and ES has also given results of the form shown on Figure 6.12.

As it may be noticed on Figure 6.11, the minimum value of $j(t)$, namely -0.46 , is less than that on Figure 6.12 but the orientation does not last as long, which is expected in view of the criterion chosen. The first instantaneous criterion (Figure 6.11) is what we call a *narrow* $j(t)$ (see Section 6.2.3) and the second one (Figure 6.12) is what we call a *wide* $j(t)$. As for the fields, the first field is what we call in Section 6.2.4 a $(\omega, 2\omega)$ field and the second one is what we call a *kick* field. Table 6.5 shows the parameters of this latter field. The fact that a field of the form of a *kick* is a very efficient field for optimizing the criterion 6.2.10 is one of our most striking result from a physical viewpoint. It is reported and commented on in [49]. In the latter reference, the $(\omega, 2\omega)$ field is also analyzed. As explained above, using the EGA-CG does not improve the results. However, CG is useful for a local search and we have

TAB. 6.5: Parameters of the optimized pulse with 3 laser fields.

n	\mathcal{E} (W/cm ²)	ω (cm ⁻¹)	ϕ (π rad)	t_0 (ps)	t_1 (ps)	t_2 (ps)	t_3 (ps)
1	1.01364×10^{08}	1389.541	1.98066	0.	0.312024	0.613023	1.193727
2	2.99976×10^{12}	500.051	1.82249	0.075077	0.270294	0.838110	1.562814
3	2.99989×10^{12}	500.000	0.82337	0.109518	0.235767	0.808280	1.080066

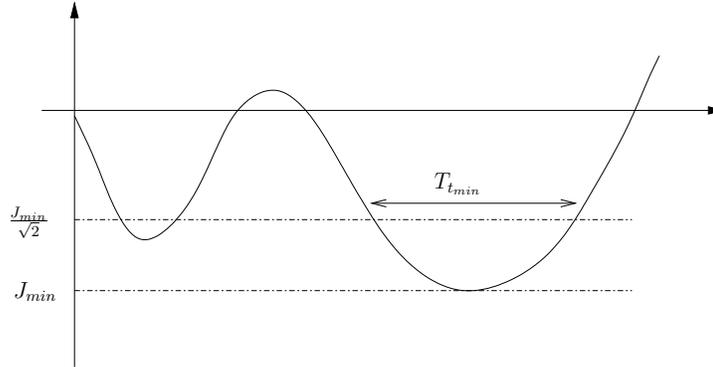


FIG. 6.9: Construction of the hybrid criterion.

tested how it could improve the result when used only at the end of a stochastic search. For this purpose we have first made an optimization on criterion (6.2.10) using EGA (the result is presented on Figure 6.13 with dotted lines) and then, we have applied the BFGS algorithm (the gradient has been computed with *Odyssée*) using the laser field so obtained as an initial guess for the conjugate gradient algorithm. After 100 CG iterations, the criterion is improved as may be seen on Figure 6.13 with solid lines. Such a result reconfirms that CG is useful for the local improvement search.

6.4.2 Results for the hybrid criterion

In view of the results of the previous section, it is a natural idea to introduce a new criterion aimed at approaching two goals together: obtaining at some given time a good orientation and keeping it as long as possible. Thus, we basically define a new criterion

$$J = J_{min} - J_{kept} + |J_{min} + J_{kept}|, \quad (6.4.37)$$

where $J_{min} = \min_{t \in [0, T]} j(t)$ and $J_{kept} = \frac{T_{t_{min}}}{T}$ where $T_{t_{min}}$ is the length the connex component of $\{t \in [0, T] \mid J_{min} \leq j(t) \leq \frac{J_{min}}{\sqrt{2}}\}$ including $t_{min} = \sup\{t \mid J(t) = J_{min}\}$ (see Figure 6.9). This criterion is a sum of three terms. The first one, J_{min} , measures the way the molecule is oriented. The second one, J_{kept} , measures how long the orientation is kept. The third part, $|J_{min} + J_{kept}|$, is a penalty term to ensure that J_{min} and $-J_{kept}$ are simultaneously

minimized.

On Figure 6.14, we show a field obtained with this criterion and which is a succession in time of two fields with a short overlay time (see Section 6.2.4). For the physical meaning of such a result, we refer to [155].

6.4.3 Results for the train of kicks

An other idea consists of starting with a field previously classified as a kick shape and using a succession of such fields in order to orient the molecule. The purpose of the optimization is thus to find the good delay between two successive *kicks*. Indeed we hope that by kicking several times the molecule we can lower the instantaneous criterion. The results are quite interesting: Figure 6.15 (a), is the result of an optimization of the criterion (6.2.10) with ES and it clearly illustrates the idea of kicking several times the molecule. This result is also interesting because the instantaneous criterion remains for a long time under the value -0.2 . Figure 6.15 (b) is the result of the optimization with the criterion (6.2.9). The criterion value (-0.82) is the best value we have ever had. However, the production of such fields remains an experimental challenge.

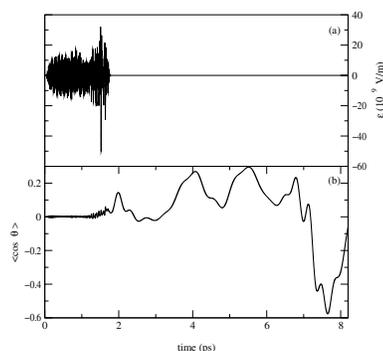


FIG. 6.10: Results obtained by optimizing upon 10 laser fields. In this figure and the following ones (6.11 to 6.14), the electric field is shown on top while $\langle \cos \theta \rangle$ which measures the instantaneous orientation of the molecule is shown on bottom. Time evolves from left to right at the same scale.

6.5 Conclusion and future directions

We have implemented and tested various strategies for the optimization of the laser field to be used for the orientation of the HCN molecule.

The best results have been obtained using evolutionary algorithms rather than purely

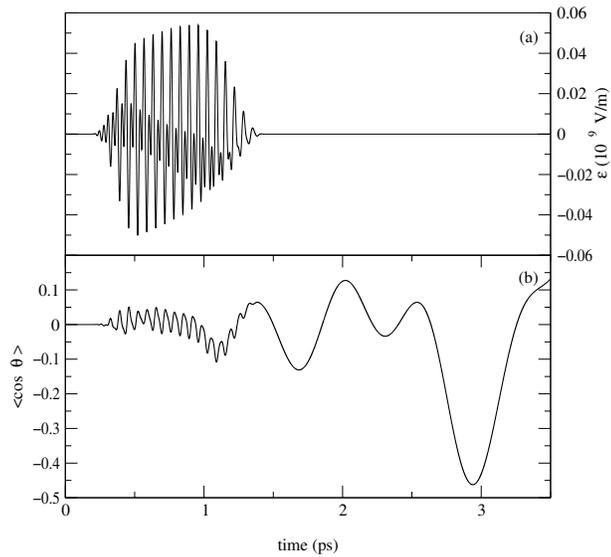


FIG. 6.11: *Best result for $J = \min_{t \in [0, T]} j(t)$.*

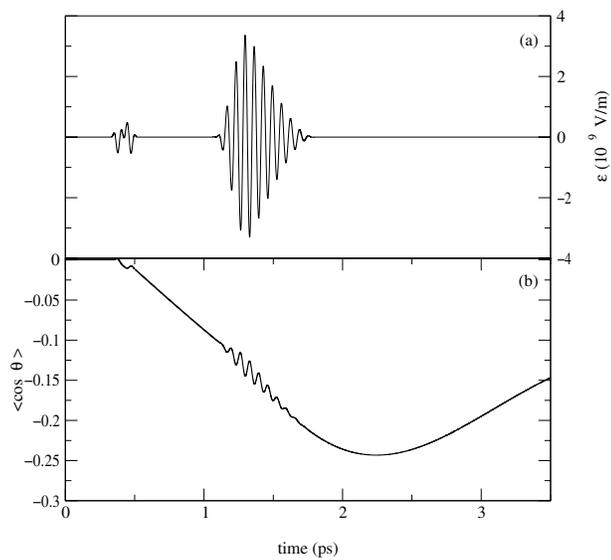


FIG. 6.12: *Best result for $J = \frac{1}{T} \int j(t) dt$.*

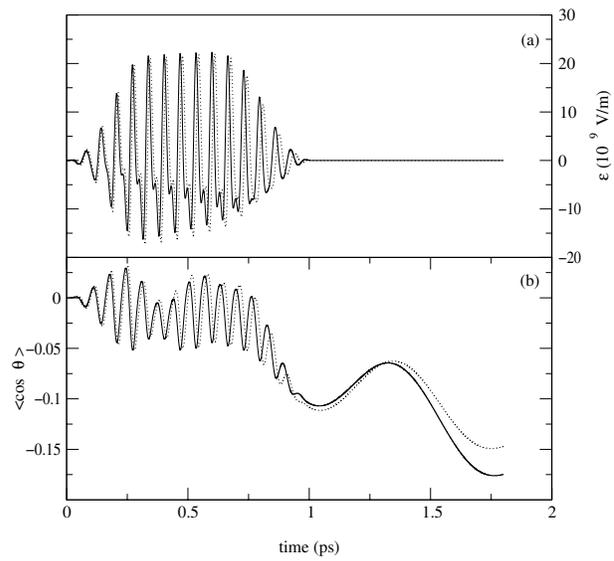


FIG. 6.13: Optimization by CG after optimization by GA.

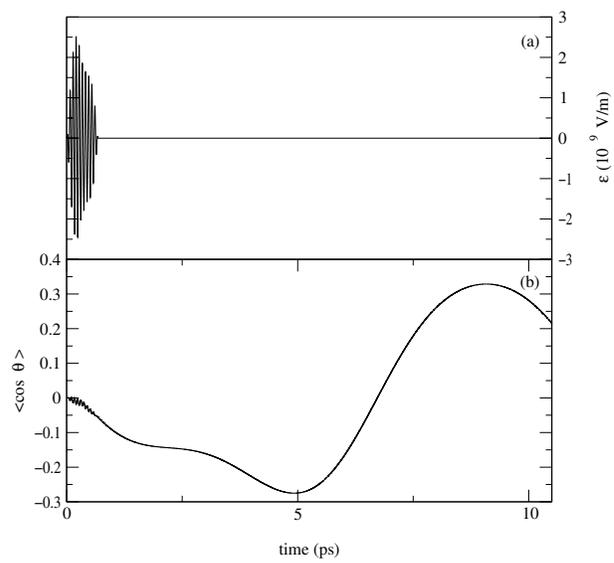


FIG. 6.14: Best result for the hybrid criterion given by equation 6.4.37.

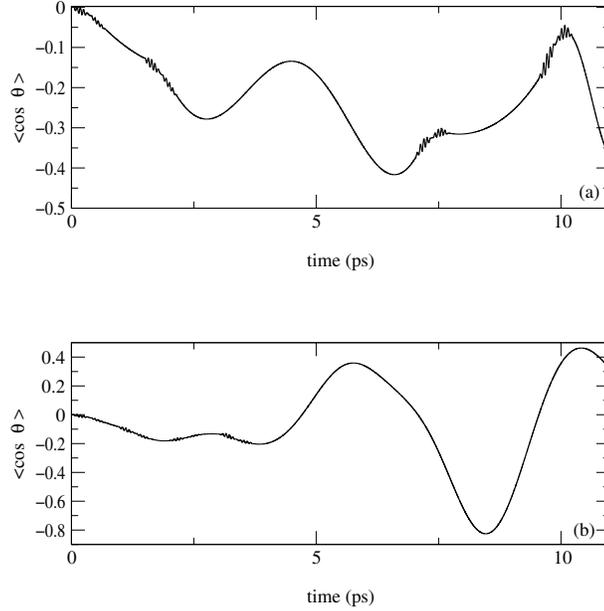


FIG. 6.15: (a): Best result for $J = \frac{1}{T} \int j(t) dt$ with the train of kicks. (b): Best result for $J = \frac{1}{T} \int j(t) dt$ with the train of kicks.

deterministic algorithms such as gradient-like algorithms. However, in the case where the criterion is differentiable, we have shown that gradient like algorithms can efficiently complement the EA, not necessary when being used throughout the generations as mutations operators (the genetic algorithms with mutation by gradient have not yielded a real benefit in our specific case), but when being used as a final step in the optimization, once the population has been optimized by EA.

In order to understand how to compute the gradient of the criterion when needed, we have performed many tests, together with a numerical analysis on a toy equation related to our case of interest. They both show that the most efficient strategy (amenable in any case) is to compute the gradient by adjoint calculus on the discretized form of the equation or, if one does not fear a tedious post-processing work, to compute the gradient with an automatic differentiation tool.

As far as the choice of the criterion is concerned, we have tested many criteria, depending upon our physical aims. A multicriteria approach has also been implemented.

From the physical standpoint, our results have allowed us to identify two specific forms of laser fields that are most promising for the future: the $(\omega, 2\omega)$ field [46, 90] and the *kick* field [47, 79]. Definite conclusions about the efficiency of these fields are yet to be obtained and will be the purpose of some of our work in the future. It is anyway to be emphasized that such physically relevant fields have been obtained through our optimization methodology

used as a blind tool, *i.e.*, without any specification of this form of fields. This is sufficient to give us some hope and confidence both in the physical and in the mathematical validity of our methodology.

Acknowledgements

The financial support of the *Action Concertée Incitative Jeunes Chercheurs* from the French Ministry of Research and Technology is gratefully acknowledged. CLB would like to thank Herschel Rabitz and André Bandrauk for enlightening discussions. The expertise of Marc Schöenauer on evolutionary strategies has been instrumental. Finally, Gabriel Turinici and Yvon Maday are to be thanked for constant stimulating interactions.

Conclusion générale

Parmi les techniques spécifiques introduites ces dernières années pour améliorer les performances des algorithmes évolutionnaires, de nombreux travaux ont été consacrés au développement de mécanismes pour l'adaptation aux phases d'exploration et d'exploitation, et ont donné naissance aux classes d'algorithmes qualifiés d'adaptatifs et d'auto-adaptatifs. Les travaux effectués au cours de cette thèse sont centrés autour de l'étude de ces algorithmes.

Dans une première partie, consacrée à des contributions algorithmiques, nous proposons une méthode pour l'adaptation de la matrice de covariance d'un opérateur de mutation gaussien. L'état de l'art précédant l'exposé de la méthode montre les différentes techniques d'adaptations de cet opérateur développées au cours des années: de la fameuse règle des 1/5 (historiquement la première méthode adaptative) à l'adaptation dérandomisée de l'algorithme CMA (*Covariance Matrix Adpatation*) en passant par les techniques auto-adaptatives des SA-ES (*Self-Adaptive Evolution Strategies*). Mais, au-delà des technique d'adaptation des opérateurs de mutation gaussienne, l'exposé présente également l'algorithme DE (*Differential Evolution*) ainsi que l'opérateur de croisement SBX (*Simulated Binary Crossover*), s'adaptant tous les deux de manière implicite à l'état de la population pour réguler exploitation et exploration. Le point commun entre ces différentes techniques est par ailleurs mis en évidence à travers des tests numériques sur des fonctions dont l'optimum change au cours du temps, clarifiant quelque peu la notion d'adaptation. Enfin, l'accent est plus particulièrement mis sur l'algorithme CMA, l'algorithme le plus récent parmi les méthodes présentées, mais de fait un des algorithmes les moins connus (il n'est encore mentionné dans aucun ouvrage de référence). Il ressort de cet état de l'art qu'à l'heure actuelle c'est sans doute le meilleur choix. En outre, le principe de base de cet algorithme étant l'adaptation de la matrice de covariance, c'est cet algorithme qui sert de référence pour la nouvelle méthode présentée pour l'adaptation de la matrice de covariance, ayant donné naissance à l'algorithme LS-CMA-ES (*Least-Square Covariance Matrix Adaptation Evolution Strategy*).

La motivation essentielle pour cette méthode est d'essayer de faire un meilleur usage de l'ensemble des points dont la valeur de *fitness* a été calculée qui sont disponibles dans tout algorithme évolutionnaire. Ces derniers ne font traditionnellement pas usage de toute cette information puisque la plupart des mauvais individus ne passent pas l'étape de remplacement,

et l'information qu'ils contiennent n'est pas utilisée. L'idée développée dans le chapitre 2 est d'apprendre la courbure de la fonction, *i.e* sa hessienne, à partir de ces exemples, et d'injecter l'inverse de cette hessienne approchée comme matrice de covariance pour l'opérateur de mutation. Cette approximation est calculée par une minimisation au sens des moindres carrés. Mais cette matrice ne doit être utilisée comme matrice de covariance de la mutation qu'à condition d'être une bonne approximation de la vraie hessienne. Pour ce faire, une mesure de qualité de l'approximation est proposée: en deçà d'un seuil de cette mesure de la qualité, seuil mis au point par essai erreur, l'approximation est utilisée comme matrice de covariance; sinon la technique développée dans l'algorithme CMA est utilisée pour la mise à jour itérative de la matrice de covariance. L'algorithme ainsi obtenu s'avère présenter des performances bien meilleures que l'algorithme CMA pour des fonctions elliptiques, quasi-elliptiques ou sur certaines fonctions multimodales. Pour toutes ces fonctions, la minimisation au sens des moindres carrés nous permet d'obtenir beaucoup plus rapidement que le CMA une bonne approximation de la courbure. Par ailleurs, les performances sur des fonctions multi-modales testées sont également meilleures que les performances de l'algorithme CMA, c'est-à-dire parmi les meilleures dans le paysage évolutionnaire aujourd'hui. Signalons enfin que la résistance au bruit de l'algorithme LS-CMA-ES est largement meilleure que celle de CMA sur les fonctions quasi-elliptiques.

Même si plusieurs améliorations sont envisageables, en particulier au niveau de la méthode d'approximation elle-même, pour élargir le champ des fonctions pour lesquelles on apprend une bonne approximation de la matrice hessienne, l'objectif de faire un meilleur usage des points est atteint.

Au delà de la méthode proposée, se trouve aussi l'idée que l'apprentissage la courbure pour la mise à jour de la matrice de covariance est une direction de recherche essentielle pour le développement d'opérateurs efficaces. Les techniques d'optimisation déterministes classique de type DFP et BFGS fournissent des méthodes pour cet apprentissage moyennant la connaissance du gradient de la fonction. La transposition de ces techniques, nécessitant certes de trouver une bonne approximation du gradient, est une voie de recherche possible et naturelle pour continuer le travail présenté.

La deuxième partie de cette thèse est consacrée à l'étude théorique d'algorithmes adaptatifs et auto-adaptatifs, et plus particulièrement aux méthodes concernant le réglage du pas de la mutation. Elle présente plus précisément des résultats de convergence d'algorithmes dans les cadres simplifiés de fonctions quasi-convexes. En termes du dilemme exploration / exploitation, ceci constitue l'étude de la phase d'exploitation. Les résultats théoriques prouvés sont des convergences log-linéaires, habituellement observées de manière expérimentale pour les algorithmes adaptatifs et auto-adaptatifs.

L'étude présentée plus précisément au chapitre 5 s'avère être une des premières études théoriques d'un algorithme utilisé en pratique. En ce sens cela constitue une avancée impor-

tante dans le domaine. Par ailleurs, c'est aussi le premier exemple de l'utilisation de chaînes de Markov sur un espace d'états continu dans le domaine évolutionnaire. L'utilisation de conditions pratiques de drift pour vérifier les propriétés sur une chaîne de Markov sous-jacente à l'algorithme s'est révélée être un outil puissant. Le champ d'application de ces techniques, même si nous nous sommes essentiellement restreint à un algorithme précis, ne s'y limite pas et permettra très certainement la compréhension de bien d'autres algorithmes.

Ces études permettent également de faire le lien avec les approches classiquement utilisées pour le réglage des paramètres d'algorithmes adaptatifs et auto-adaptatifs. D'une part, nous démontrons au chapitre 4 que le réglage du paramètre de mutation proposé dans l'approche traditionnelle dite du *progress rate* correspond précisément à la vitesse de convergence optimale pour l'algorithme analysé sur la fonction sphère. Dans cette approche traditionnelle, seul ce paramètre est explicité. Notre étude de convergence, qui est complète sur la fonction sphère, permet de donner en outre la signification des autres paramètres en terme de vitesse de convergence (ou divergence) de l'algorithme, ce qui est là aussi nouveau par rapport aux approches antérieures. D'autre part notre analyse des algorithmes auto-adaptatifs et dérandomisés montre que sous les conditions suffisantes que nous donnons pour la convergence log-linéaire presque sûre, un théorème central limite est vérifié. Cela nous donne une méthode numérique avec un intervalle de confiance pour calculer le taux de convergence de l'algorithme par une méthode de Monte Carlo et en déduire les paramètres optimaux pour l'algorithme – ce qui permet dans ce cas à de répondre au problème crucial du réglage des paramètres, en cherchant les paramètres maximisant ce taux de convergence.

La troisième partie est consacrée à l'application d'un algorithme évolutionnaire à un problème de contrôle de l'orientation de molécules par laser, résultat d'une collaboration avec des physiciens. Ce problème d'optimisation continue est un exemple de problème où l'utilisation d'un algorithme évolutionnaire est avantageuse de plusieurs points de vue. Tout d'abord, la fonction à optimiser s'étant révélée multimodale, des méthodes d'optimisation classiques sont prises en défaut. Ensuite la souplesse de la mise en oeuvre a permis de tester rapidement, avec le même algorithme, et sans nécessiter la mise en oeuvre (souvent fastidieuse) d'un calcul de gradient, l'optimisation de plusieurs fonctions intéressant les physiciens.

Le travail effectué au cours de cette thèse souligne le fait que l'amélioration des performances d'un algorithme évolutionnaire passe par l'introduction de techniques spécifiques, quitte à perdre, pragmatisme oblige, le parallèle entre les algorithmes évolutionnaires et la biologie.

Par ailleurs nous montrons que l'analyse théorique d'algorithmes introduits avec une vision intuitive et pragmatique est faisable. Ces études théoriques, permettent en outre de comprendre comment régler certains paramètres des algorithmes étudiés, fournissant ainsi un fondement théorique aux réglages habituellement empiriques et fastidieux.

Un enjeu des analyses théoriques est également de comparer les différents algorithmes entre eux. Par exemple pour l'optimisation combinatoire, l'analyse de la complexité de différents algorithmes génétiques sur le *Minimum spanning tree problem* [107], permet de comparer les approches évolutionnaires à des méthodes de recherches heuristiques, donnant ainsi plus de crédibilité aux algorithmes évolutionnaires. Pour les algorithmes adaptatifs, l'analyse développée dans le chapitre 5 est transposable à d'autres types d'algorithmes. L'analyse plus précise du taux de convergence des différents algorithmes peut être une façon de comparer les différents algorithmes entre eux. Sur un plan purement théorique, ceci passe par l'analyse de la mesure invariante de la chaîne $(Z_n)_{n \in \mathbb{N}}$, qui peut s'avérer assez difficile, en particulier parce que la chaîne n'est pas réversible. C'est une perspective du travail présenté ici. Cependant une méthode combinant théorie et simulation de Monte Carlo est plus immédiatement accessible comme cela a été montré en fin de chapitre 5.

Par ailleurs, seule l'adaptation du pas de la mutation a été analysée pour l'instant, or l'analyse des algorithmes CMA et LS-CMA passe par l'analyse de l'adaptation de la matrice de covariance également. Les méthodes développées par [5] pour l'analyse d'algorithmes MCMC adaptatifs devraient être transposable pour ces algorithmes.

Tout aussi naturelle qu'elle soit, la notion d'adaptation des paramètres des algorithmes évolutionnaires n'en est d'autant moins cruciale pour réduire le coût des algorithmes. En effet, même si la plupart des fonctions considérées ici sont des fonctions tests immédiates à calculer, pour des problèmes réels, la minimisation du coût en terme de nombre d'évaluations de la fonction objectif devient essentielle. Soulignons aussi le fait que dans le domaine des algorithmes évolutionnaires, le concept d'adaptation ne s'est pas réduit à l'adaptation des paramètres de mutation ou croisement mais a par exemple également été introduit pour de l'optimisation sous contrainte pour régler les coefficients de pénalité [19, 20].

Enfin, l'apparition récente de techniques adaptatives dans d'autres domaines que celui des algorithmes évolutionnaires (méthode de recuit simulé et méthodes MCMC) souligne que la notion d'adaptation des paramètres d'un algorithme est naturelle et ne se limite pas au domaine évolutionnaire mais surtout permet d'envisager un champ d'application plus large des techniques présentées et développées ici.

Publications

- [10] A. Auger, N. Vanhaecke, and M. Schoenauer. *LS-CMA-ES: A second-order algorithm for covariance matrix adaptation*. In Xin Yao et al., editor, *Proceedings of the 8th Conference on Parallel Problems Solving from Nature*, pages 182–191. Springer-Verlag, LNCS 3242, 2004 (*Best Student Paper Award*).
- [12] Anne Auger, *Convergence results for $(1, \lambda)$ -SA-ES using the theory of φ -irreducible markov chains*, *Theoretical Computer Science*, à paraître.
- [150] M. Wagner, A. Auger, M. Schoenauer. A New Robust Estimation of Distribution Algorithms. *Inria Research Report RR-5190*, 2004.
- [9] A. Auger, C. Le Bris, and M. Schoenauer. *Dimension-independent Convergence Rate for Non-isotropic $(1, \lambda)$ -ES*, In E. Cantu-Paz et al., editor, *Proceedings of the Genetic and Evolutionary Conference 2003*, pages 512–524. LNCS 2723 and 2724, Springer Verlag, 2003.
- [8] A. Auger, C. Le Bris, and M. Schoenauer, *Rigorous analysis of some simple adaptive ES*, Rapport de recherche INRIA, RR-4914.
- [108] M. Nicolau, A. Auger, C. Ryan. Functional dependency and degeneracy: detailed analysis of the GAuGE system. *Proceedings of the 6th International Conference on Evolutionary Computation EA'03*, 2003.
- [105] M.Schoenauer, A. Auger, A. Ben Haj Yedder. Overview and Software Guide of Evolutionary Algorithms; Case Study in Quantum Control. *CRM Proc. Lecture Notes*, 2003.
- [11] A. Auger, A. Ben Haj Yedder, E. Cancès, C. Le Bris, C. M. Dion, A. Keller, and O. Atabek. Optimal laser control of molecular systems: methodology and results. *Mathematical Models and Methods in Applied Sciences*, 12(9):1281-1315, 2002.
- [155] A. Ben Haj Yedder, A. Auger, C. M. Dion, E. Cancès, A. Keller, C. Le Bris, O. Atabek. Numerical optimization of laser fields to control molecular orientation. *Phys. Rev. A*, 66, 063401 , 2002.

Bibliographie

- [1] E.H.L. Aarts and T.J. Kors. *Simulated annealing and Boltzmann machines: stochastic approach to combinatorial optimization and neural computing*. John Wiley, New York, 1989.
- [2] E.H.L. Aarts and P.J.M. Van Laarhoven. *Simulated annealing: theory and applications*. D. Reidel, Dordrecht, 1987.
- [3] B. Addis, M. Locatelli, and F. Schoen. Local optima smoothing for global optimization, 2003.
- [4] F. Aluffi-pentini, V. Parisi, and F. Zirilli. Global Optimization and Stochastic Differential Equations. *Journal of Optimization Theory and Applications*, 1(47):1–16, 1985.
- [5] C. Andrieu and E. Moulines. On the ergodicity properties of some adaptive mcmc algorithms. *Annals of Applied Probability*, submitted.
- [6] A. Assion, T. Baumer, M. Bergt, T. Brixnerand, B. Kiefer, V. Seyfried, M. Strehle, and G. Gerber. Control of chemical reactions by feedback-optimized phase-shaped femtosecond laser pulses. *Science*, 282:919–922, 1998.
- [7] Osman Atabek, Claude M Dion, and Adel Ben Haj Yedder. Evolutionary algorithms for the optimal laser control of molecular orientation. *Journal of Physics B: Atomic, Molecular and Optical Physics*, 36(23):4667–4682, 2003.
- [8] A. Auger, C. Le Bris, and M. Schoenauer. Rigorous analysis of some simple adaptive ES. Rapport de recherche INRIA, RR-4914.
- [9] A. Auger, C. Le Bris, and M. Schoenauer. Dimension-independent Convergence Rate for Non-isotropic $(1, \lambda)$ -ES. In E. Cantu-Paz et al., editor, *Proceedings of the Genetic and Evolutionary Conference 2003*, pages 512–524. LNCS 2723 and 2724, Springer Verlag, 2003.
- [10] A. Auger, N. Vanhaecke, and M. Schoenauer. LS-CMA-ES: A second-order algorithm for covariance matrix adaptation. In Xin Yao et al., editor, *Proceedings of the 8th*

Conference on Parallel Problems Solving from Nature, pages 182–191. Springer-Verlag, LNCS 3242, 2004.

- [11] A. Auger, A. Ben Haj Yedder, E. Cancès, C. Le Bris, C. M. Dion, A. Keller, and O. Atabek. Optimal laser control of molecular systems: Methodology and results. *Mathematical Models and Methods in Applied Sciences*, 12(9):1281–1315, 2002.
- [12] Anne Auger. Convergence results for $(1, \lambda)$ -SA-ES using the theory of φ -irreducible markov chains. *Theoretical Computer Science*, à paraître.
- [13] T. Bäck, F. Hoffmeister, and H.-P. Schwefel. A survey of Evolution Strategies. In R. K. Belew and L. B. Booker, editors, *Proceedings of the 4th International Conference on Genetic Algorithms*, pages 2–9. Morgan Kaufmann, 1991.
- [14] T. Bäck and M. Schütz. Intelligent mutation rate control in canonical GAs. In Z. W. Ras and M. Michalewicz, editors, *Foundation of Intelligent Systems 9th International Symposium, ISMIS '96*, pages 158–167. Springer Verlag, 1996.
- [15] Th. Bäck and H.-P. Schwefel. An Overview of Evolutionary Algorithms for Parameter Optimization. *Evolutionary Computation*, 1(1):1–23, 1993.
- [16] Thomas Bäck. Personal communication.
- [17] J.M. Ball, J.E. Marsden, and M. Slemrod. Controllability for distributed bilinear systems. *SIAM J. Control Optim.*, 20:575–597, 1982.
- [18] Th. Bäck, D.B. Fogel, and Z. Michalewicz, editors. *Handbook of Evolutionary Computation*. Oxford University Press, 1997.
- [19] S. BenHamida and M. Schoenauer. An adaptive algorithm for constrained optimization problems. In M. Schoenauer et al., editor, *Proceedings of the 6th Conference on Parallel Problems Solving from Nature*, LNCS, pages 529–539. Springer Verlag, 2000.
- [20] S. BenHamida and M. Schoenauer. Aschea: New results using adaptive segregational constraint handling. pages 884–889. IEEE Press, 2002.
- [21] A. Benveniste, M. Métiver, and P. Priouret. Algorithmes adaptatifs et approximations stochastiques, 1987.
- [22] H.-G. Beyer. *The Theory of Evolution Strategies*. Springer, Heidelberg, 2001.
- [23] A. Bienvenüe and O. François. Global convergence for evolution strategies in spherical problems: Some simple proofs and difficulties. *Theoretical Computer Science A*, 306(1-3):269–289, 2003.

- [24] P. Billingsley. *Probability and Measure*. Wiley, New York, 1986.
- [25] F. Bonnans, J.C. Gilbert, C. Lemarechal, and C. Sagastizábal. *Optimisation Numérique, aspects théoriques et pratiques*, volume 23 of *Mathématiques & Applications*. Springer Verlag, 1997.
- [26] J. F. Bonnans, J. C. Gilbert, C. Lemaréchal, and C. Sagastizabal. *Optimisation Numérique : aspects théoriques et pratiques*. Springer, Berlin, 1997.
- [27] P. A. N. Bosman and D. Thierens. Advancing Continuous IDEAs with Mixture Distributions and Factorization Selection Metrics. In *Optimization by Building and Using Probabilistic Models*, pages 208–212. San Francisco, California, USA, 2001.
- [28] Amine Boumaza. *Introduction de techniques d'évolution artificielle en vision tridimensionnelle et en robotique mobile*. PhD thesis, Université René Descartes, Paris V, Paris, France, 2004.
- [29] G. E. P. Box and K. B. Wilson. On the experimental attainment of optimum conditions. *Journal of the Royal Statistical Society, Series B*, 13:1–45, 1951.
- [30] C. Le Bris. Problématiques numériques pour la simulation moléculaire. In *Actes du 32ème Congrès national d'analyse numérique*, volume 9, page 2000. ESAIM : Proceedings, September.
- [31] C. Le Bris. Control theory applied to quantum chemistry: Some tracks. In *International Conference on Control of Systems Governed by PDEs (Nancy, March 1999)*, volume 8, pages 77–94. ESAIM PROC, 2000.
- [32] P. Brumer and M. Shapiro. Control of unimolecular reactions using coherent light. *Chem. Phys. Lett.*, 126:541–546, 1986.
- [33] P. Brumer and M. Shapiro. Laser control of chemical reactions. *Scientific American*, pages 34–39, March 1995.
- [34] E. Cancès, C. Le Bris, and M. Pilot. Contrôle optimal bilinéaire sur une équation de Schrödinger. *Note aux Compte Rendu de l'Académie des Sciences*, pages 567–571, 2000.
- [35] R. Cerf. The dynamics of mutation-selection algorithms with large population sizes. *Ann. Inst. H. Poincaré Probab. Statist.*, 32(4):455–508, 1996.
- [36] R. Cerf. Asymptotic convergence of genetic algorithm. *Adv. in Appl. Probab.*, 30(2):521–550, 1998.
- [37] S. Chelkowski, M. Zamojski, and A. D. Bandrauk. Laser-phase directional control of photofragments in dissociative ionization of H_2^+ using two-color intense laser pulses. *Phys. Rev. A*, 63:023409, 2001.

- [38] G. Chonghui and Tang Huanwen. Global convergence properties of evolution strategies. *Chinese Journal of Numerical Mathematics and Applications*, 23(2):78–84, 2001.
- [39] J.C. Culioli. *Introduction à l'optimisation*. Ellipses, 1994.
- [40] C. E. Dateo and H. Metiu. Numerical solution of the time-dependent schrodinger equation in sperical coordinates by fourier-transformation methods. *J. Chem. Phys*, 95:7392–7400, 1991.
- [41] K. Deb and R. W. Agrawal. Simulated binary crossover for continuous search space. *Complex Systems*, 9:115–148, 1995.
- [42] K. Deb and H.-G. Beyer. Self-adaptive genetic algorithms with simulated binary crossover. *Evolutionary Computation*, 1999.
- [43] K.A. DeJong. *The Analysis of the Behavior of a Class of Genetic Adaptive Systems*. PhD thesis, University of Michigan, Ann Harbor, 1975. *Dissertation Abstract International*, 36(10), 5140B. (University Microfilms No 76-9381).
- [44] J.M DeLaurentis, L. A. Ferguson, and W.E. Hart. On the convergence properties of a simple self-adaptive evolutionary algorithm. In W.B. Langdon & al., editor, *Proceedings of the Genetic and Evolutionary Conference*, pages 229–237. Morgan Kaufmann, 2002.
- [45] C. M. Dion. *Dynamique de l'alignement et de l'orientation moléculaire induite par laser. Simulations numériques sur HCN en champ infrarouge*. PhD thesis, Université de Sherbrooke et Université de Paris-Sud, 1999.
- [46] C. M. Dion, A. D. Bandrauk, O. Atabek, A. Keller, H. Umeda, and Y. Fujimura. Two-frequency ir laser orientation of polar molecules. numerical simulations for hcn. *Chem. Phys. Lett.*, 302:215–223, 1999.
- [47] C. M. Dion, A. Keller, and O. Atabek. Orienting molecules using half-cycle pulses. *Eur. Phys. J. D*, 14:249–255, 2001.
- [48] C. M. Dion, A. Keller, O. Atabek, and A. D. Bandrauk. Laser-induced alignment dynamics of HCN: Roles of the permanent dipole moment and the polarizability. *Phys. Rev. A*, 59:1382–1391, 1999.
- [49] C. M. Dion, A. Ben Haj Yedder, E. Cancès, A. Keller, C. Le Bris, , and O. Atabek. Optimal laser control of orientation: The kicked molecule. submitted.
- [50] C. M. Dion, A. Ben Haj Yedder, E. Cancès, A. Keller, C. Le Bris, and O. Atabek. Optimal laser control of orientation: The kicked molecule. *Phys. Rev. A*, 65:063408, 2002.

- [51] Stefan Droste, Thomas Jansen, , and Ingo Wegener. On the analysis of the (1+1) evolutionary algorithm. *Theoretical Computer Science*, 276:51–81, 2002.
- [52] M. Dufflo. *Méthodes récursives aléatoires*. Masson, Paris, 1990.
- [53] M. Dufflo. *Algorithmes stochastiques*. Springer-Verlag, Paris, 1996.
- [54] A. E. Eiben, R. Hinterding, and Z. Michalewicz. Parameter control in Evolutionary Algorithms. *IEEE Transactions on Evolutionary Computation*, 3(2):124, 1999.
- [55] A.E. Eiben and J.E. Smith. *Introduction to Evolutionary Computing*. Springer, 2003.
- [56] A.E. Eiben and J. I. van Hemert. Saw-ing eas: adapting the fitness function for solving constrained problems. In D. Corne, M. Dorigo, and F. Glover, editors, *New ideas in optimization, chapter 26*, pages 389–402. McGraw-Hill, 1999.
- [57] EO. C++ class library, <http://eodev.sourceforge.net>.
- [58] L. Eshelman and J. D. Schaffer. Real-coded genetic algorithms and interval-schemata. In L. D. Whitley, editor, *Foundations of Genetic Algorithms 2*, pages 187–202, Los Altos, CA, 1993. Morgan Kaufmann.
- [59] C. Faure and Y. Papegay. *Odyssée*. Technical report.
- [60] M.D. Feit, J.A. Fleck, and A. Steiger. Solution of a Schrodinger equation by a spectral method. *J. Comput. Phys.*, 47:412–433, 1982.
- [61] D. B. Fogel. *Evolving artificial intelligence*. PhD thesis, University of California, San Diego, CA, 1992.
- [62] L. J. Fogel, A. J. Owens, and M. J. Walsh. *Artificial Intelligence through Simulated Evolution*. New York: John Wiley, 1966.
- [63] O. François. Global optimization with exploration/selection algorithms and simulated annealing. *Ann. Appl. Probab.*, 1(248-271), 2002.
- [64] B. Freidrich and D. R. Herschbach. On the possibility of orienting rotationally cooled polar molecules in an electric field. *Z. Phys. D*, 18:153–161, 1991.
- [65] J. Garnier, L. Kallel, and M. Schoenauer. Rigorous hitting times for binary mutations. *Evolutionary Computation*, 7(2):167–203, 1999.
- [66] Andreas Griewank Laurent Hascoët George Corliss, Christèle Faure and Uwe Naumann, editors. *Optimal laser control of chemical reactions using automatic differentiation.*, New York, 2001. Springer-Verlag.

- [67] C. J. Geyer. Practical markov chain monte carlo. *Statistical Science*, 7(4):473–511, 1993.
- [68] D. E. Goldberg. *Genetic algorithms in search, optimization and machine learning*. Addison Wesley, 1989.
- [69] D. E. Goldberg. Real-coded genetic algorithms, virtual alphabets, and blocking. *Complex Systems*, 5:139–168, 1991.
- [70] H.-M. Gutmann. A radial basis function method for global optimization. *Journal of Global Optimization*, 19(3):201–227, 2001.
- [71] H. Haario, H. Saksman, and J. Tamminen. An adaptive metropolis algorithm. *Bernoulli*, 7:223–242, 2001.
- [72] N. Hansen. CMA-ES for Noisy Optimization: Implementations in Matlab. http://www.bionik.tu-berlin.de/user/niko/cmaes_inmatlab.html.
- [73] N. Hansen. *Verallgemeinerte individuelle Schrittweitenregelung in der Evolutionsstrategie: eine Untersuchung zur entstochastisierten, koordinatensystemunabhängigen Adaptation der Mutationsverteilung*. Mensch und Buch Verlag, 1998.
- [74] N. Hansen. Invariance, Self-Adaptation and Correlated Mutations in Evolution Strategies. In M. Schoenauer et al., editor, *Proceedings of the 6th Conference on Parallel Problems Solving from Nature*, pages 355–364. Springer-Verlag, LNCS 1917, 2000.
- [75] N. Hansen and S. Kern. Evaluating the CMA Evolution Strategy on Multimodal Test Functions. In Xin Yao et al., editor, *Proceedings of the 8th Conference on Parallel Problems Solving from Nature*, pages 281–291. Springer-Verlag, LNCS 3242, 2004.
- [76] N. Hansen, S. Müller, and P. Koumoutsakos. Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES). *Evolutionary Computation*, 11(1):1–18, 2003.
- [77] N. Hansen and A. Ostermeier. Completely Derandomized Self-Adaptation in Evolution Strategies. *Evolutionary Computation*, 9(2):159–195, 2001.
- [78] Jun He and Lishan Kang. On the convergence rates for genetic algorithms. *Theoretical Computer Science*, (229):23–39, 1999.
- [79] N. E. Henriksen. Molecular alignment and orientation in short pulse laser fields. *Chem. Phys. Lett.*, 312:196–202, 1999.
- [80] K. Hoki and Y. Fujimura. Quantum control of alignment and orientation of molecules by optimized laser pulses. *Chem. Phys.*, 267:187–193, 2001.

- [81] J. H. Holland. *Adaptation in natural and artificial systems*. University of Michigan Press, Ann Arbor, 1975.
- [82] L. Ingber. Very fast simulated re-annealing. *Mathl. Comput. Modelling*, 12(8):967–973, 1989.
- [83] L. Ingber. Simulated annealing: Practice versus theory. *Math. Comput. Modeling*, 18:29–57, 1993.
- [84] L. Ingber and B. Rosen. Genetic Algorithms and Very Fast Simulated Reannealing: A Comparison. *Journal of Mathematical and Computer Modeling*, 11(16):87–100, 1992.
- [85] Yaochu Jin. A comprehensive survey of fitness approximation in evolutionary computation. *Soft Computing*, 2004.
- [86] D. Jones, M. Schonlau, and W. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13:455–492, 1998.
- [87] D. R. Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 21(4):345–383, 2001.
- [88] R. S. Judson, K. K. Lehmann, H. Rabitz, and W. S. Warren. Optimal design of external fields for controlling molecular motion: Application to rotation. 223:425–456, 1990.
- [89] R. S. Judson and H. Rabitz. Teaching lasers to control molecules. *Phys. Rev. Lett.*, 68:1500–1503, 1992.
- [90] T. Kanai and H. Sakai. Numerical simulation of molecular orientation using strong, nonresonant, two-color laser fields. *J. Chem. Phys.*, 115:5492–5497, 2001.
- [91] S. Kern, S.D. Müller, N. Hansen, D. Büche, J. Ocenasek, and P. Koumoutsakos. Learning Probability Distributions in Continuous Evolutionary Algorithms - A Comparative Review. *Natural Computing*, 3:77–112, 2004.
- [92] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.
- [93] J. R. Koza. *Genetic Programming: On the Programming of Computers by means of Natural Evolution*. MIT Press, Massachusetts, 1992.
- [94] P. Larranaga and J. A. Lozano. *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation*. Kluwer Academic Publishers, 2001.
- [95] J. J. Larsen, I. Wendt-Larsen, and H. Stapelfeldt. Controlling the branching ratio of photodissociation using aligned molecules. *Phys. Rev. Lett.*, 83:1123–1126, 1999.

- [96] Z. Li and H.A. Scheraga. Monte carlo approach to the multiple-minima problem in protein folding. *Proceedings. National Academic Science USA*, 84:15–29, 1987.
- [97] M. Locatelli. Simulated annealing algorithms for continuous global optimization: Convergence conditions, 2000.
- [98] K. Mengersen and R. Tweedie. Rates of convergence of the hastings and metropolis algorithms. 1996.
- [99] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. Equation of State Calculations by Fast Computing Machines. *J. Chem. Phys.*, 6(21):1087–1092, 1953.
- [100] S.P. Meyn and R.L. Tweedie. *Markov Chains and Stochastic Stability*. Springer-Verlag, New York, 1993.
- [101] Z. Michalewicz. *Genetic Algorithms+Data Structures=Evolution Programs*. Springer Verlag, New-York, 1996. 3rd edition.
- [102] Z. Michalewicz. *Genetic algorithms + data structure = evolution programs*. Springer, 1999.
- [103] P. Del Moral. Measure valued processes and interacting particle systems. application to nonlinear filtering problems. *Ann. Appl. Probab*, 8(2):438–495, 1998.
- [104] P. Moscato. On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms. Technical Report Caltech Concurrent Computation Program, Report. 826, California Institute of Technology, Pasadena, California, USA, 1989.
- [105] M.Schoenauer, A. Auger, and A. Ben Haj Yedder. Overview and software guide of evolutionary algorithms; case study in quantum control. In *CRM Proc. Lecture Notes*, 2003.
- [106] S.D. Müller. *Bio-Inspired Optimization Algorithms for Engineering Applications*. Diss. ETH No. 14719, 2002.
- [107] F. Neumann and I. Wegener. Randomized local search, evolutionary algorithms, and the minimum spanning tree problem. In *Proceedings of Gecco 2004*.
- [108] Miguel Nicolau, Anne Auger, and Conor Ryan. Functional dependency and degeneracy: detailed analysis of the GAuGE system. *Proceedings of the 6th International Conference on Evolutionary Computation EA'03*, 2003.

- [109] R. Numico, A. Keller, and O. Atabek. Laser-induced molecular alignment in dissociation dynamics. *Phys. Rev. A*, 52:1298–1309, 1995.
- [110] J. Obalek. *Rekombinationsoperatoren für Evolutionsstrategieren*. PhD thesis, Universität Dortmund, Fachbereich Informatik, 1994.
- [111] J. Ocenasek, S. Kern, N. Hansen, and P. Koumoutsakos. A mixed bayesian optimization algorithm with variance adaptation. In Xin Yao et al., editor, *Proceedings of the 8th Conference on Parallel Problems Solving from Nature*, pages 352–361. Springer-Verlag, LNCS 3242, 2004.
- [112] J. Ocenasek and J. Schwarz. Estimation of distribution algorithm for mixed continuous-discrete optimization problems. In *2nd Euro-International Symposium on Computational Intelligence*, pages 227–232. IOS Press, 2002.
- [113] M. Olhofer, T. Arima, T. Sonoda, B. Sendhoff, and H.A. Schreiber. Optimisation of a stator blade used in a transonic compressor cascade with evolution strategies. In I.C. Parmee editor, editor, *Adaptive Computing in Design and Manufacture (ACDM)*, pages 45–54. Springer Verlag, 2000.
- [114] M. Olhofer, Y. Jin, and B. Sendhoff. Adaptive encoding for aerodynamic shape optimization using evolution strategies. In *Congress on Evolutionary Computation (CEC)*, pages 576–583. IEEE Press, 2001.
- [115] A. Ostermeier, A. Gawelczyk, and N. Hansen. A Derandomized Approach to Self Adaptation of Evolution Strategies. *Evolutionary Computation*, 2(4):369–380, 1994.
- [116] A. Ostermeier, A. Gawelczyk, and N. Hansen. Step-size Adaptation Based on Non-local Use of Selection Information. In Y. Davidor, H.-P. Schwefel, and R. Manner, editors, *Proceedings of the 3rd Conference on Parallel Problems Solving from Nature*, pages 189–198. Springer-Verlag, LNCS 866, 1994.
- [117] M. Pelikan and E. Goldberg, D.E. and Cantu-Paz. Boa: the bayesian optimization algorithm. In D.E. Goldberg & al., editor, *Proceedings of the Genetic and Evolutionary Conference 99*, pages 525–532. Morgan Kaufmann, 1999.
- [118] L. Piela, J. Kostrowicki, and H.A. Scheraga. The multiple-minima problem in conformational analysis of molecules. deformation of the protein energy hypersurface by the diffusion equation method. *J. Phys. chem*, 93:3339–3346, 1989.
- [119] J. Poland. Explicit local models: Towards "optimal" optimization algorithms. In *Proceedings of the 15th European Conference on Machine Learning (ECML)*, pages 569–571, 2004.

- [120] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in C*. Cambridge University press, 1992.
- [121] N. J. Radcliffe and P. D. Surry. Real representations. In L. D. Whitley and R. K. Belew, editors, *Foundations of Genetic Algorithms 4*, pages 51–72. Morgan Kaufmann, 1997.
- [122] I. Rechenberg. *Evolutionstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution*. Fromman-Holzboog Verlag, Stuttgart, 1973.
- [123] D. Revuz. *Markov Chains*. North-Holland Publishing Co, Amsterdam, 1984. second edition.
- [124] C. P. Robert. Convergence control techniques for markov chain monte carlo algorithms. *Statistical Science*, 10(3):231–253, 1996.
- [125] G. Rudolph. *Convergence Properties of Evolutionary Algorithms*. Kovac, Hamburg, 1997.
- [126] G. Rudolph. Convergence rates of evolutionary algorithms for a class of convex objective functions. *Control and Cybernetics*, 26(3):375–390, 1997.
- [127] Günter Rudolph. Self-adaptive mutations may lead to premature convergence. Technical Report CI-73/99, Fachbereich Informatik, Universität Dortmund, 44221 Dortmund, 28 1999.
- [128] H. Sakai, C. P. Safvan, J. J. Larsen, K. M. Hilligsoe, K. Hald, , and H. Stapelfeldt. Controlling the alignment of neutral molecules by a strong laser field. *J. Chem. Phys.*, 110:10235–10238, 1999.
- [129] A. Scheel. *Beitrag zur Theorie der Evolutionsstrategie*. Mensch und Buch Verlag, 1985.
- [130] B. W. Schmeiser. Batch size effects in the analysis of simulation outputs. *Operational Research*, 30:556–568, 1982.
- [131] H.-P. Schwefel. *Numerical Optimization of Computer Models*. John Wiley & Sons, New-York, 1981. 1995 – 2nd edition.
- [132] H. P. Schwefel. *Numerical Optimization of Computer Models*. John Wiley & Sons, New-York, 1995 – 2nd edition edition, 1981.
- [133] Hans-Paul Schwefel. Collective phenomena in evolutionary systems. In P. Checkland and I. Kiss, editors, *Problems of Constancy and Change – The Complementarity of Systems Approaches to Complexity, Papers presented at the 31st Annual Meeting Int'l Soc. General System Research*, volume 2, pages 1025–1033, Budapest, June 1–5, 1987. Int'l Soc. for General System Research.

- [134] M. Sebag and A. Ducoulombier. Extending population-based incremental learning to continuous search spaces. In Th. Bäck, G. Eiben, M. Schoenauer, and H.-P. Schwefel, editors, *Proceedings of the 5th Conference on Parallel Problems Solving from Nature*, pages 418–427. Springer Verlag, 1998.
- [135] M.A. Semenov. Convergence velocity of an evolutionary algorithm with self-adaptation. In W.B. Langdon & al., editor, *Proceedings of the Genetic and Evolutionary Conference*, pages 210–213. Morgan Kaufmann, 2002.
- [136] F. M. Silva and L. B. Almeida. Speeding up backpropagation. In editor R. Eckmiller, editor, *Advanced Neural Computers*, pages 151–158, 1990.
- [137] A.F.M. Smith and G.O. Roberts. Bayesian computation via the gibbs sampler and related markov chain monte carlo methods (with discussion). *J. Roy. Stat. Soc. Ser. B*, 55:3–24, 1993.
- [138] T. Sonoda, Y. Yamaguchi, T. Arima, M. Olhofer, B. Sendhoff, and H.A. Schreiber. On the experimental attainment of optimum conditions. *Journal of Turbomachinery*, 2004.
- [139] R. Storn. References to DE Applications. <http://www.icsi.berkeley.edu/storn/code.html#appl>.
- [140] R. Storn and K. Price. Minimizing the real functions of the icec'96 contest by differential evolution. In *Proc. of ICEC96*, pages 842–844, 1996.
- [141] R. Storn and K. Price. Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization*, (11):341–359, 1997.
- [142] Rapports sur la science et la technologie de l'Académie des Sciences. Sciences aux temps ultracourts - de l'attoseconde aux petawatts. Technical report, Académie des Sciences, 2000.
- [143] L. Tierney. Markov chains for exploring posterior distributions. *Annal of Statistics*, 22:1701–28, 1994.
- [144] G. Turinici. Controlabilité exacte de la population des états propres dans les systèmes quantiques bilinéaires. *Note aux Compte Rendu de l'Académie des Sciences*, pages 327–332, 2000.
- [145] G. Turinici and H. Rabitz. Quantum wave function controllability. *J. Chem. Phys.*, 267:1–9, 2001.
- [146] G. Turinici and H. Rabitz. *Wavefunction controllability in quantum systems*. Preprint, 2001.

- [147] D. Vanderbilt and S.G. Louie. A monte carlo simulated annealing approach to optimization over continuous variables. *Journal of Computational Physics*, 56(259-271), 1984.
- [148] N. Vanhaecke, Ch. Lisdat, B. T'Jampens, D. Comparat, A. Crubellier, and P. Pillet. Accurate asymptotic ground state potential curves of cs_2 from two-colour photoassociation. *Eur. Phys. J. D*, 28:351–360, 2004.
- [149] M.D. Vose. *The Simple Genetic Algorithm: foundations and theory*. MIT Press, 1999.
- [150] M. Wagner and A. Auger M. Schoenauer. A new robust estimation of distribution algorithms. Rapport de recherche INRIA, RR-5190, 2004.
- [151] David Wales and Jonathan Doye. Global optimization by basin-hopping and the lowest energy structures of lennard-jones clusters containing up to 110 atoms. *J. Phys. Chem. A*, 101:5111–5116, 1997.
- [152] D. Williams. *Probability with Martingales*. Cambridge University Press, Cambridge, 2000.
- [153] X. Yao and Y. Liu. Fast evolution strategies. In B. Reynolds and M. Conrad, editors, *Proceedings of the 6th Annual Conference on Evolutionary Programming*, LNCS 1213, pages 151–162. Springer Verlag, 1997.
- [154] A. Ben Haj Yedder. *Optimisation numérique et Contrôle optimal: applications en chimie moléculaire*. PhD thesis, Ecole Nationale des Ponts et Chaussées, 2002.
- [155] A. Ben Haj Yedder, A. Auger, C. M. Dion, E. Cancès, A. Keller, C. Le Bris, and O. Atabek. Numerical optimization of laser fields to control molecular orientation. *Phys. Rev. A*, 66:063401, 2002.
- [156] W. Zhu and H. Rabitz. A rapid monotonically convergent iteration algorithm for quantum optimal control over the expectation value of a positive definite operator. *J. Chem. Phys.*, 109:385–391, 1998.