



HAL
open science

Algorithmes stochastiques pour la gestion du risque et l'indexation de bases de données de média

Victor Reutenauer

► **To cite this version:**

Victor Reutenauer. Algorithmes stochastiques pour la gestion du risque et l'indexation de bases de données de média. Mathématiques générales [math.GM]. COMUE Université Côte d'Azur (2015 - 2019), 2017. Français. NNT : 2017AZUR4018 . tel-01544854

HAL Id: tel-01544854

<https://theses.hal.science/tel-01544854>

Submitted on 22 Jun 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



UNIVERSITÉ CÔTE D'AZUR
ÉCOLE DOCTORALE EDSFA N° 364
SCIENCES FONDAMENTALES ET APPLIQUÉES

THÈSE

pour obtenir le titre de

Docteur en Sciences

de l'Université Côte d'azur

Mention : MATHÉMATIQUES

Présentée et soutenue par

Victor REUTENAUER

Algorithmes stochastiques pour la gestion du risque et l'indexation de bases de données de media

Thèse dirigée par Denis TALAY

Thèse co-dirigée par Gilles PAGÈS

préparée à INRIA Sophia Antipolis, Projet TOSCA

Soutenance le 22 Mars 2017

Jury :

<i>Rapporteurs :</i>	Jean-François CHASSAGNEUX	-	UPD (LPMA)
	Benjamin JOURDAIN	-	ENPC (CERMICS)
<i>Directeurs :</i>	Gilles PAGÈS	-	UPMC (LPMA)
	Denis TALAY	-	INRIA (TOSCA)
<i>Examineurs :</i>	Nicole EL KAROUI	-	UPMC (LPMA)
	Emmanuel GOBET	-	Ecole Polytechnique (CMAP)

Remerciements et contexte du travail académique effectué

Remerciements

Je tiens tout d'abord à remercier Denis Talay et Gilles Pagès qui ont encadré ce travail. Je tiens aussi à remercier Etienne Tanré qui a été mon premier maître de stage avant que nous poursuivions notre collaboration au cours des années qui se sont succédé.

Je remercie mes deux rapporteurs Benjamin Jourdain et Jean-François Chasagneux pour leurs relectures de ce manuscrit ainsi que Nicole El Karoui et Emmanuel Gobet d'avoir accepté de compléter le jury. Je remercie un autre de mes co-auteurs, Antoine Lejay ainsi que l'équipe qui l'entoure à Nancy avec qui j'ai effectué de nombreux séjours d'immersion en 2007 et 2008.

En transition des remerciements académiques, je remercie Christophe Michel, qui en plus d'être le co-auteur du chapitre 3 a supervisé mon travail au sein du Crédit Agricole entre 2006 et 2011. Lui ainsi que Benjamin Mellul m'ont permis de collaborer avec Etienne, Denis et Antoine et leurs équipes et débiter ma carrière à la lisière du monde académique et industriel.

Je souhaite aussi remercier Mikaël Rousson et Matthieu Aubry ainsi que Lokmane Abbas-Turki avec qui j'ai eu de fructueux échanges qui ont nourri ce travail.

Je souhaite aussi remercier mes collègues devenus amis Ernesto Palidda et Maxence Jeunesse qui ont relu certaines parties de mon manuscrit. Je remercie aussi mes amis devenus associés Nilouphar Vossough et Raphaël Cesari pour leur soutien au cours des dernières années. Martin Gaspillard en étant à mes côtés au quotidien dans mon travail professionnel m'a aussi permis de m'investir dans ce travail académique.

J'oublie de nombreuses personnes, stagiaires, collègues, professeurs, employés, amis ou une combinaison de tout cela que j'ai à un moment croisés et qui m'ont parfois guidé ou appuyé durant les années où j'ai effectué et rédigé ce travail.

Je remercie ma mère Geneviève Houglet et ma famille plus généralement pour leur soutien au cours de ces années et j'ai bien sûr une pensée affectueuse pour Dalal.

Contexte

Ce travail a été réalisé en parallèle à deux professions principales. J'ai toujours eu à cœur de relier au maximum les problématiques industrielles et commerciales aux problématiques académiques.

Si cette démarche a pu surprendre dans les deux milieux professionnels dans lesquels j'ai exercé, elle est cependant utile par ses retombées économiques et humaines.

La salle des marchés

Durant sept ans, j'ai été *Quant* entre Londres et Paris. L'équipe de recherche et développement taux d'intérêt et hybrides du Crédit Agricole CIB, m'a permis d'effectuer de longs séjours au sein du laboratoire TOSCA de l'INRIA dans le cadre d'un contrat de partenariat qui a duré de 2007 à 2010. Ceci m'a permis de travailler sur des problématiques fines de simulations de trajectoires de processus, de méthodes de réduction de variance par quantification et d'application du gradient stochastique à l'optimisation de portefeuille en marché incomplet.

La vie en start-up

Depuis un peu plus de trois ans, je me suis lancé dans différents projets entrepreneuriaux sur les sujets de traitement et d'indexation de grandes bases de données de photos. Nous fournissons actuellement des services à l'industrie touristique pour *Crowd-sourcer* des medias illustrant l'attractivité des régions françaises par exemple. J'ai ainsi été amené à découvrir le lien entre les méthodes de gradient stochastique et de quantification avec la problématique de l'indexation de grandes bases de données de medias.

Table des matières

1	Introduction	1
1.1	Gradient stochastique	3
1.2	Quantification	17
1.3	Méthode d'évaluation sans biais	28
2	Réduction de variance par quantification	45
2.1	The algorithm	49
2.2	Numerical tests	53
2.3	Conclusion	56
2.4	Approx. SDE Brownian	57
3	Liquidité	69
3.1	Introduction	71
3.2	Our settings : swaps with liquidity cost	71
3.3	Hedging error minimization in Gaussian world	73
3.4	Numerical Validation	80
3.5	Empirical study liquidity costs	86
3.6	Conclusion	94
3.7	Appendix	96
4	Simulation exacte	99
4.1	Introduction	101
4.2	Unbiased Estimators	102
4.3	The detailed algorithm for the CIR Model	114
4.4	Numerical Results	116
4.5	Conclusion	124
5	Indexation de medias	129
5.1	Media in Euclidean Space	131
5.2	High Dimension Quantization	134
5.3	Hierarchical tree media indexing : non supervised learning	150
5.4	Enhancement of Lloyds algorithm : Split and Stick	151
5.5	Annex : Validation of Approximations	158
5.6	Conclusion	168

Chapitre 1

Introduction

Résumé

Ce premier chapitre présente différentes méthodes numériques s'appliquant à des problèmes d'optimisation couvrant des domaines variés. Nous nous intéressons dans la partie 1.1 aux problèmes de contrôle stochastique et mettons en évidence l'avantage des techniques de gradient stochastique en les comparant qualitativement à d'autres méthodes numériques comme les EDPs, les arbres et la méthode de Monte-Carlo pour la réduction de variance.

Nous introduisons aussi les problèmes d'indexation de medias pour des recherches visuelles qui utilisent aussi le gradient stochastique.

Un point fondamental de cette présentation est d'explicitier les différences et similarités dans la table 1.1 entre la calibration de réseaux neuronaux et l'optimisation de portefeuille solution d'un problème de contrôle stochastique.

La partie 1.2 traite de la méthode de quantification. Après une présentation théorique générale, nous évoquons quelques méthodes de construction. Puis nous précisons le lien existant avec la problématique d'indexation et de classification de médias.

Enfin une dernière partie 1.3 propose quelques exemples de techniques avancées pour le calcul de prix et la couverture de produits dérivés sur les marchés financiers. Différentes méthodes de réduction de variance sont évoquées. Nous présentons aussi l'approche de Malliavin pour Monte-Carlo pour le calcul des grecques et de la simulation trajectorielle exacte d'équations différentielles stochastiques unidimensionnelles. Nous abordons enfin les liens entre les imperfections de marchés et les méthodes numériques, d'une part via l'implication technique conséquente à la prise en compte de ces imperfections de marché et d'autre part en étudiant comment certaines approximations numériques déforment le marché.

Afin de ne pas alourdir le premier chapitre introductif, nous considérons d'une part comme familière pour le lecteur la méthodologie d'évaluation de Black-Scholes dans un marché financier complet et viable et certaines notions plus avancées en mathématiques financières telles que la couverture dynamique,

les options annulables et les marchés de taux. D'autre part une compréhension des réseaux neuronaux utilisés pour la classification d'images et la détection de formes, d'objets et de scènes est utile pour bien appréhender le chapitre final 5.1 de ce manuscrit. La partie 1.1.3.3 expose rapidement les réseaux neuronaux.

Abstract

This first chapter presents different numerical optimization methods applied to stochastic control problems. We compare them qualitatively to stochastic gradient descent in section 1.1. We draw a parallel between optimization of portfolio and calibration of neural network in table 1.1. Then section 1.2 presents quantization method. Finally section 1.3 presents advanced pricing methodologies for options in finance, among them variance reduction and exact simulation. Then we explain the links between numerical approximations and market imperfections.

1.1 Contrôle stochastique et problème d'optimisation : Les atouts du gradient stochastique

1.1.1 Contrôle stochastique : un état des lieux des méthodes et des applications

Nous présentons ici succinctement plusieurs types de problèmes d'optimisation, de contrôle stochastique ainsi que des méthodes numériques permettant de les résoudre de façon exacte ou approchée. Sans précision particulière, un paragraphe présente une méthode ou un problème dans le cadre d'un espace probabilisé $(\Omega, \mathcal{F}, \mathbb{P})$, $(B_t)_{t \geq 0}$ est un mouvement brownien standard sur cet espace, X est une variable aléatoire ou $(X_t)_{t \geq 0}$ un processus à valeur dans \mathbb{R}^N .

1.1.1.1 Une vision EDPiste des méthodes de contrôle stochastique : Hamilton Jacobi Bellman

Problème de contrôle optimal, programmation dynamique et solution de viscosité Nous présentons ici le problème d'optimisation de portefeuille. Il s'agit d'un exemple classique d'application du principe de programmation dynamique. Ce problème a motivé le développement de différentes méthodes numériques que nous étudions ensuite.

Définition 1. Partant en 0 d'une valeur initiale x , on considère la diffusion d'une richesse $X_t^{0,x,\pi}$ contrôlée par π :

$$dX_t^{0,x,\pi} = b\left(t, X_t^{0,x,\pi}, \pi\left(t, X_t^{0,x,\pi}\right)\right) dt + \sigma\left(t, X_t^{0,x,\pi}, \pi\left(t, X_t^{0,x,\pi}\right)\right) dB_t \quad (1.1)$$

où $(B_t)_{t \geq 0}$ est un mouvement brownien standard de dimension finie.

On se donne une maturité T et une fonction d'utilité U à valeur et variable réelle, croissante et concave. On note \mathcal{A} l'ensemble des stratégies de contrôle π , X_t -mesurables.

Partant en 0 d'une richesse initiale x , le problème d'optimisation de portefeuille consiste à résoudre :

$$\sup_{\pi \in \mathcal{A}} \mathbb{E} \left[U(X_T^{x,0,\pi}) | \mathcal{F}_0 \right]$$

Un tel problème est un problème de contrôle stochastique classique. On peut définir ce problème partant de t pour n'importe quelle richesse x , on notera $v(t, x)$ sa valeur.

On trouve dans [55] par Pham une introduction au principe de la programmation dynamique qui permet de résoudre concrètement ces problèmes, nous la présentons dans le paragraphe suivant.

Principe de la programmation dynamique Le principe de la programmation dynamique permet alors de montrer qu'optimiser ces fonctions valeurs consiste à résoudre le problème suivant par bout,

$$v(t, x) = \sup_{\pi \in \mathcal{A}} \mathbb{E}_t [U(X_T^{x,t,\pi})]$$

sur $[0, T] \times \mathbb{R}^n$ avec les notations de l'équation (1) et où $X_t^{x,s,\pi}$ représente la richesse à l'instant t partant de x à l'instant s sous le contrôle π .

En d'autre terme, la restriction sur un intervalle d'une stratégie optimale est toujours une stratégie optimale.

Proposition 2. Sous des hypothèses restrictives, la fonction valeur est solution de l'équation de Hamilton-Jacobi-Bellman suivante :

$$\begin{cases} \frac{\partial v}{\partial t} + \sup_{\pi \in \mathcal{A}} \mathcal{L}^\pi v(t, x) = 0 \\ v(T, x) = g(x) \end{cases}$$

où $\mathcal{L}^\pi w = b(x, a) \cdot \nabla_x w + \frac{1}{2} \text{tr}(\sigma(x, a) \sigma'(x, a) D_x^2 w)$, est appelé générateur de la diffusion. La seconde équation est appelé condition finale. De plus le contrôle optimal $\hat{\pi}$ est l'argument maximisant. On peut prouver son existence sous différentes hypothèses.

La preuve est basée sur un calcul d'Itô et le principe d'optimalité de Bellman.

On appelle cette approche un principe de programmation dynamique rétrograde ou *backward*.

Comme nous le présentons au paragraphe 1.1.1.1, cette spécification permet de trouver une approximation spatio-temporelle du contrôle optimal.

Solutions de viscosité La théorie des solutions de viscosité a été introduite pour étendre l'application du contrôle stochastique dans des cas où les solutions ne sont ni dérivables ni même continues. Elles en sont une extension, car si les solutions des problèmes de contrôle sont des fonctions continûment dérivables du temps et de la valeur des processus stochastiques, elles sont alors aussi des solutions de viscosité. Le travail de Pham dans [55] contient une présentation plus détaillée et formelle de ces objets.

Discrétisation d'HJB : schémas implicites et schémas explicites Dans le cas présenté dans le paragraphe 1.1.1.1 on peut discrétiser numériquement le problème en utilisant l'approche de résolution du calcul d'espérance en terme d'équation aux dérivées partielles (EDP).

La résolution numérique de ces équations repose sur la discrétisation du générateur infinitésimal. Nous présentons tout d'abord le théorème de Feynman-Kac liant le calcul d'espérance aux équations aux dérivées partielles.

Théorème 1. On considère la fonction bivariée : $f(t, x) = \mathbb{E} \left[g \left(X_T^{x,t,0} \right) \right]$ où le processus X_t suit une diffusion d'une EDS de type (4.1) et g est une fonction réelle au moins borélienne.

La fonction f est solution de l'équation :

$$\frac{\partial f}{\partial t} + b \frac{\partial f}{\partial x} + \frac{1}{2} \sigma^2 \frac{\partial^2 f}{\partial x^2} = 0 \text{ avec une condition terminale } f(x, T) = g(x).$$

On pourra se rapporter à [31], chapitre 5 pour une démonstration. Le principe de la résolution d'EDP par différences finies consiste à introduire une grille de discrétisation de l'espace. Des considérations sur la diffusion, avec un retour à la moyenne ou non, ou sur le problème et sa solution permettent d'introduire des conditions de bords en l'espace. Nous explicitons la condition au bord en temps. Considérant l'approximation des dérivées par différences finies, on peut alors écrire des équations linéaires reliant ces différentes valeurs discrétisées.

Nous nous plaçons ici sur une grille uniforme, δ_t représente le pas de temps et δ_x le pas d'espace considérés constants. La quantité $f_{i,j}$ est donc la discrétisation de la valeur de f à la date $i\delta_t$ et au point $m_X + j\delta_x$, m_X étant la plus petite valeur de X de la grille de discrétisation et 0 l'origine des temps du problème ainsi que de la grille de discrétisation.

Les différentes manières d'approcher les dérivées introduisent ainsi des conditions linéaires explicites ou implicites.

Voici les contraintes au centre de la grille dans le schéma explicite d'une résolution d'EDP rétrograde telle que présentée ici :

$$\frac{f_{i,j} - f_{i-1,j}}{\delta_t} + b_{i,j} \frac{f_{i,j+1} - f_{i,j-1}}{2\delta_x} + \frac{1}{2} \sigma_{i,j}^2 \frac{f_{i,j-1} - 2f_{i,j} + f_{i,j+1}}{\delta_x^2} = 0,$$

Ce schéma est dit explicite car l'on peut écrire simplement la valeur de $f_{i-1,j}$ en fonction de la valeur de f au temps suivant déjà calculé dans le cas d'une équation rétrograde. Cette équation est dite rétrograde car nous avons une condition limite en temps à la borne supérieure et cherchons la condition limite en temps à la borne inférieure.

Dans le schéma implicite pour une équation rétrograde, on remplacerait la discrétisation du premier terme qui est la dérivée en temps par $\frac{f_{i+1,j} - f_{i,j}}{\delta_t}$.

La résolution reste dans tous les cas assez efficace.

En effet les matrices de contraintes se révèlent assez creuses. Elles proviennent de l'arbre de transition de probabilité. Cet arbre peut être vu comme la discrétisation de la diffusion ainsi que l'ont présenté Pagès, Pham et Printems dans [48].

Même dans le cas implicite, on peut résoudre en une complexité linéaire, bien que la matrice de contrainte ait bien entendu une quantité de coefficients quadratique, mais nuls pour la plupart. Ces méthodes s'appellent réduction cyclique ou méthode de Thomas et sont par exemple présentées dans [13].

Cette approche peut s'adapter en dimension d'espace supérieur à 1. Ces méthodes sont très performantes tant que ces dimensions sont petites, par exemple inférieures à 3.

Elles sont utilisées en finance pour calculer des prix d'options. Nous présentons rapidement les principes de l'évaluation de prix en finance dans le paragraphe 1.3.1.

Elles sont des méthodes très efficaces de résolution de problèmes numériques dont on peut calculer explicitement des ordres de convergences, mais présentent

certains désavantages notamment de nécessiter une résolution séquentielle difficilement parallélisable.

De plus la combinatoire des données à manipuler est exponentielle en la dimension. Cela rend les méthodes de résolution des EDPs par différences finies inopérantes pour les problèmes en grandes dimensions.

On pourra se rapporter à [18] ou [57] pour des précisions sur les ordres de convergence de ces équations paraboliques, liées à un processus de diffusion.

1.1.1.2 Aperçu d'autres méthodologies numériques

Il existe d'autres méthodes numériques pour la résolution de problème de contrôle stochastique ou d'estimation de prix d'option annulable. Rogers a proposé dans [59] une solution duale au problème de recherche de temps d'arrêt optimal en montrant que le prix de l'option est le minimum d'une sur-martingale qui majore le *payoff*. Cependant cette méthode n'est pas très constructive et donne habituellement des majorations assez larges. Il existe aussi une technique utilisant les équations différentielles rétrogrades pour résoudre des problèmes de contrôle stochastique, on trouvera dans [5] une référence récente sur ce sujet. On peut aussi citer une méthode liée au calcul de Malliavin pour l'estimation d'espérance conditionnelle et ainsi l'estimation du prix d'option américaine ou l'optimisation de portefeuille.

1.1.1.3 Longstaff-Schwartz et les bases de fonctions pour l'estimation d'espérances conditionnelles

Définition du produit financier Les méthodes connues dans le milieu financier sous le nom de Monte-Carlo Américain permettent de calculer des prix d'options annulables en utilisant une méthode de type Monte-Carlo.

Nous présentons ici la version discrétisée de l'option américaine qui est appelé option bermudéenne.

Longstaff et Schwartz ont introduit cette méthode dans [38] pour l'évaluation de ces options.

Le principe de la programmation dynamique présenté dans le paragraphe 1.1.1.1 assure que le prix d'une option bermudéenne est l'espérance du maximum de sa valeur de continuation et de sa valeur d'exercice.

Nous expliquons ici comment la méthode de Monte-Carlo Américain est utilisée pour approcher la valeur de continuation et sa stratégie d'exercice.

On considère un actif ayant une diffusion de type $dS_t = S_t b(t, S_t) dt + S_t \sigma(t, S_t) dB_t$, une fonction g appelée le *payoff* (vanille) de cette option, et $t < T_0 < \dots < T_N$, N dates donnant le droit d'exercice de cette option. Le *payoff* étant vanille, il est donc markovien de la valeur de l'actif ou de son logarithme qu'on note : $X_t = \log(S_t)$. On définit τ comme étant la stratégie d'exercice de cette option, c'est-à-dire le temps d'arrêt où il est optimal de se faire payer le *payoff* plutôt que de continuer à garder l'option. \mathcal{A}_i est l'ensemble des stratégies admissibles à valeur dans $[T_i, \dots, T_N]$.

Principe de programmation dynamique La valeur de l'option en T_i peut donc s'écrire sans facteur d'actualisation (en absence de taux d'intérêt ou par utilisation de la probabilité dite de Monte-Carlo) :

$$v(T_0, X_{T_0}) = \operatorname{ess\,sup}_{\tau \in \mathcal{A}_0} \mathbb{E}[g(X_\tau)] \quad (1.2)$$

A chaque date, on définit la valeur de continuation :

$$\begin{aligned} c(T_i, X_{T_i}) &= \operatorname{ess\,sup}_{\tau \in \mathcal{A}_{i+1}} \mathbb{E}_{T_i}[g(X_\tau)] \\ &= \mathbb{E}_{T_i}[v(T_{i+1}, X_{T_{i+1}})] \end{aligned} \quad (1.3)$$

En étendant à une date future le prix introduit dans (1.2), on note $v(T_i, X_{T_i})$ le prix de l'option à une date T_i pour une valeur de l'actif X_{T_i} .

Le principe de programmation dynamique nous permet d'écrire le prix de l'option en T_i en tant que maximisation à cette date

$$v(T_i, X_{T_i}) = \max\{g(X_{T_i}), c(T_i, X_{T_i})\},$$

et donc en utilisant (1.3), on a

$$v(T_i, X_{T_i}) = \max\{g(X_{T_i}), \mathbb{E}_{T_i}[v(T_{i+1}, X_{T_{i+1}})]\}.$$

Par rétropropagation en partant de T_N , on va construire une approximation T_i -mesurable et convergente de la valeur de l'option en T_i .

Méthode de résolution numérique par regression Pour ce faire on va chercher à approcher la valeur de continuation, qui est une espérance conditionnelle, par des polynômes du processus du prix du sous-jacent.

On propose d'utiliser le fait que la meilleure approximation, au sens $\mathbb{L}^2(\Omega, \sigma(X_{T_i}))$ dans l'espace de probabilité, d'une espérance conditionnelle T_i mesurable, d'une quantité (appelons la *payoff*) $T_{i+1} > T_i$, par une fonction, de processus markovien qui permettent d'approcher l'espérance conditionnelle du *payoff*, consiste simplement à faire la regression du *payoff* en T_{i+1} par rapport aux différents moments des processus considérés.

Simplement, on supprime l'espérance conditionnelle du problème.

Dans la méthode numérique présentée dans le paragraphe 1.1.1.1, l'approximation de l'espace de contrôle est naturellement la valeur du contrôle sur la grille discrétisée. Ici il s'agit d'une fonction polynomiale aux dates où il est utile de prendre une décision. En sachant que l'ensemble des fonctions polynomiales, $\{x^k\}$ où k est un multi-indice, est une base hilbertienne de $L^2(\mathbf{R}^d, \mathbf{P}_{X_{T_i}})$, cela assure la convergence de l'algorithme lorsque l'on fera croître leur degré.

On note $\tilde{v}(T_i, X_{T_i})$ l'approximation ainsi obtenue du prix de l'option. En remarquant que $c(T_N, X_{T_N}) = 0$ car on ne peut plus attendre on initialise la procédure rétrograde de résolution numérique.

On montre qu'on peut simplement supprimer l'espérance conditionnelle du problème de regressions. En effet, on cherche à minimiser :

$$\mathbb{E}_t \left[\left(\sum_k \lambda_k^{(i)} X_{T_i}^k - \underbrace{\mathbb{E}_{T_i} [\tilde{v}(T_{i+1}, X_{T_{i+1}})]}_{T_i\text{-mesurable}} \right)^2 \right]$$

Il est immédiat, vu la règle de composition des espérances conditionnelles et la linéarité des conditions (dérivées en fonction des paramètres) d'optimum que ce problème a la même solution que

$$\mathbb{E}_t \left[\left(\sum_k \lambda_k^{(i)} X_{T_i}^k - \tilde{v}(T_{i+1}, X_{T_{i+1}}) \right)^2 \right]$$

Ainsi on a ramené les problèmes numériques en celui de la regression de variables aléatoires sur d'autres, à savoir le prix approché, que l'on optimise au fur et à mesure.

Cette première approche que nous venons de détailler, permet de construire une approximation du prix et une approximation de la stratégie d'exercice.

Dans une deuxième approche on va utiliser la stratégie ainsi définie pour estimer un minorant du prix de cette option.

Il est important de noter que l'on obtient des prix conservateurs en tant qu'acheteur de cette option d'exercice.

Remarque 1. En absence de dividendes, le Call américain, c'est-à-dire avec $g(x) = (x - K)_+$, est égal au Call européen grâce à la convexité du *payoff* et la formule de Jensen. Un praticien dirait que la valeur temps du call est toujours positive, ou encore son gamma.

Cette approche fonctionne en plus grande dimension disons jusqu'à 5 et un des avantages de Monte-Carlo est de pouvoir traiter des *payoffs* qui sont dit path-dépendant, c'est-à-dire dépendant eux-mêmes de la trajectoire.

Différentes améliorations proposées par des praticiens consistent par exemple à grouper les simulations, on va chercher alors une régression par morceaux ainsi que l'a présentée Villemin dans [62].

1.1.1.4 Prise en compte de la liquidité : C etin, Jarrow, Protter

Nous pr esentons ici l'approche propos ee par C etin et al. dans [15] de prise en compte du spread bid-ask pour l' evaluation d'instruments financiers. Elle se place dans un mod ele ou le prix *mid* -la moyenne entre le *bid* et l'*ask*- suit un mod ele de type Black-Scholes. L'approche num erique consiste  a discr etiser la diffusion de l'actif sur une grille de temps choisie, pour des valeurs provenant d'un arbre binaire recombina nt.

La construction des valeurs prises par l'arbre est tr es simple. Puis un calcul supposant l'absence d'opportunit e d'arbitrage permet de calculer la probabilit e pour le prix (*mid*) de monter ou de descendre.

La flexibilit e de cette m ethode num erique permet d'introduire de l'illiquidit e dans le mod ele de mani ere  a  etudier sa r epercussion sur les prix des actifs contingents, c'est- a-dire les options sur ces instruments.

Arbre binomial recombinant en dimension un Nous présentons ici rapidement le modèle de diffusion discret, en temps et en espace appelé arbre binomial recombinant. L'arbre est dit binomial car, à chaque date, l'actif monte ou descend d'une certaine quantité. Il est dit recombinant car si à deux étapes consécutives, l'actif monte puis descend, il revient au niveau initial. En d'autres termes on peut commuter ces étapes, par composition, la valeur de l'actif dépend juste du nombre de pas où il a monté et du nombre de pas où il a baissé.

Si l'on se place sous la probabilité risque-neutre, l'absence d'opportunité d'arbitrage nous permet de déterminer la probabilité de monter ou de descendre. Ces notions que nous réutilisons largement dans le paragraphe 1.3.1, sont présentée par exemple dans l'ouvrage de Lamberton et Lapeyre [31].

On va s'intéresser à des accroissements proportionnels aux prix des actifs, ce qui permet de garantir la positivité du processus des prix d'actifs.

On considère un pas de temps δ_t et d'espace δ_x . A chaque étape du temps, l'actif $X_{i\delta_t}$ va voir son prix multiplier par $1 + \delta_x$ où $\frac{1}{1 + \delta_x}$ de manière à être recombinant selon des probabilités respectives p et $1-p$. L'absence d'opportunité d'arbitrage (en considérant les taux d'intérêts nuls) permet de déterminer la valeur de p en fonction de δ_t .

On doit avoir

$$p(1 + \delta_x) + (1 - p) \frac{1}{1 + \delta_x} = 1$$

ce qui se résout simplement en

$$p = \frac{1}{2 + \delta_x}.$$

La diffusion du prix de transaction des actifs, sera celle du prix *mid*, c'est-à-dire où le carnet d'ordre d'achat et de vente se rejoint, ou encore pour l'achat d'une quantité d'actif nulle marginalement.

Le coût de liquidité On doit encore définir le coût de liquidité, i.e. le fait que les prix d'achat et de vente auxquels des transactions peuvent effectivement être réalisées sont différents.

Le prix d'échange d'une quantité m d'actif ($m > 0$ pour un achat), est donné par $X_t \int_0^m (1 + \Psi(x)) dx$.

Le coût de liquidité, ou spread bid-ask Ψ est positif et croissant, mais pas nécessairement continu.

Pour des *payoffs* markoviens, i.e. vanilles, on peut calculer de façon rétrograde, des couvertures optimales dans ce modèle et via cette approche numérique, en effectuant une rétropropagation des prix, la couverture dans tous les états de l'arbre.

On peut utiliser cette technique d'arbre recombinant en plusieurs dimensions, cependant une approche simpliste consiste à considérer à chaque étape le produit cartésien des espaces unidimensionnels. Cela amène la même complexité combinatoire que celle évoquée pour les EDPs dans le paragraphe 1.1.1.1.

Le paragraphe 1.1.2.2 présente la méthode pour effectuer l'évaluation de prix et le calcul de la couverture dans le cadre d'un arbre de transition de probabilité.

1.1.2 *The curse of dimensionality* : état des lieux des tentatives d'affranchissement

Les méthodes présentées dans le paragraphe 1.1.1 permettent de résoudre un certain nombre de problèmes non triviaux mais sont cependant peu adaptées à la résolution de problèmes en grande dimension (au delà de 5 ou 10 pour la plupart).

Différentes méthodes ont été proposées ces dernières années pour tirer parti d'avancées technologiques telles que la parallélisation présentée dans le paragraphe 1.1.2.1 ou théoriques et méthodologiques s'agissant de 1.1.2.2.

1.1.2.1 Méthodes avancées d'EDP

Elles sont plus efficaces notamment en grande dimension ou permettent une parallélisation des calculs.

David Pommier a travaillé dans sa thèse [56] sur le fait d'utiliser une discrétisation *sparse* des fonctions et de permettre d'utiliser ces méthodes en plus grandes dimensions. Une technique dite pararéelle a été développée par Yvon Maday dans [36] ou Pagès et al. dans [49]. L'idée est de discrétiser en temps de manière grossière et d'effectuer des résolutions en parallèle plus fine entre chaque intervalle défini par la première discrétisation.

1.1.2.2 Quantification vectorielle et arbre de transitions de probabilités : quantification de processus

Limitations de certaines méthodes par arbre Pour la dimension 10 un arbre binaire recombinaut sur chaque dimension aurait pour la date N , N^{10} éléments à considérer et serait donc inutilisable au delà d'une dizaine de dates.

Les EDPs, sont souvent présentées comme une méthode convergente de calcul d'espérance conditionnelle, c'est-à-dire que la fonction continue $f(t, X)$ est discrétisée en $\hat{f}(t_i, X_j)$. Le deuxième argument représentant l'espace peut-être de dimension supérieure à 1 et j est alors un multi-indice. Pourtant, on peut considérer la matrice de contraintes comme définissant des probabilités de transitions. Dans le cas explicite d'EDP rétrograde, le générateur infinitésimal est en fait simplement discrétisé et on peut en déduire facilement les probabilités de transition qui lui sont liées. Pour une grille grossière, les probabilités de transitions ne sont pas positives, mais les EDPs sont convergentes à la limite, ce cas n'est donc pas utile donc inintéressant. Dans le cas explicite, les transitions ne se font que d'un point aux points contigus de l'espace. Dans le cas implicite, la diffusion se fait d'un point à tous les points de la grille, car l'inverse d'une matrice triangulaire n'est pas creuse, ou *sparse*, c'est-à-dire qu'aucun de ses coefficients n'est nul. En outre, alors que la matrice de contrainte est creuse, son inverse est pleine.

Cependant dans ce cas, la combinatoire du nombre d'états est exponentielle en la dimension. Par exemple pour $N = 4$ valeurs différentes d'actifs ou de variables d'état (ce qui est très peu pour une méthode qui se veut une discrétisation d'un processus continu), en dimension $d = 10$, $4^{10} \approx 1,000,000$ points sont nécessaires pour pouvoir représenter l'espace d'état dans toutes ses dimensions.

Arbre de quantification d'un processus de Markov Dans le cas de ces arbres provenant de grilles de quantifications, quelques centaines voire milliers de points peuvent couvrir correctement l'espace.

Cette méthode donne des résultats corrects pour des dimensions de l'ordre de 5 à 10. Elle permet de s'affranchir de certaines complexités dues à la dimension que rencontrent d'autres méthodes numériques d'arbres.

On va considérer un processus multidimensionnel observé à un ensemble de dates définies comme un ensemble de variables vectorielles. On va considérer l'ensemble de ces discrétisations, pour chaque date choisie.

Le concept des arbres de transitions de probabilités entre ces grilles consiste à choisir des probabilités de transition d'un élément d'une grille à un élément de la grille de la date suivante.

On présente dans le paragraphe 1.2.1.2 la construction des arbres de quantification de processus.

La quantification vectorielle est une méthode convergente d'intégration numérique qui permet d'approcher l'espérance de variables vectorielles à densité continue. Elle se base en quelque sorte sur la discrétisation de densités de distribution. La partie 1.2 la présente plus en détail, notamment pour ce qui concerne certaines propriétés théoriques de convergence.

Citons les travaux de Bally et al. [47] qui proposent un algorithme d'évaluation d'options américaines dans le cadre d'un modèle de Black et Scholes et estiment le temps d'arrêt optimal qui définit la date d'exercice de l'option.

Comme pour le problème d'intégration numérique, l'utilisation de quantifieurs stationnaires permet dans cette application d'améliorer l'estimation numérique à travers le passage à un ordre supérieur de convergence (cf Pagès et Bally [6]).

De plus, Pagès et Pham [48] définissent une quantification markovienne de processus préservant la propriété de Markov vérifiée par le processus continu sous-jacent.

Ainsi, ils proposent une solution numérique à un problème de contrôle stochastique apparaissant dans le cadre financier de la gestion de portefeuille.

Nous expliquons à présent le principe de leur utilisation pour résoudre ce type de problème.

Utilisation des arbres pour le contrôle stochastique Nous avons déjà présenté certains aspects des problèmes d'évaluation d'option dans le cadre de l'algorithme de Longstaff-Schwartz dans le paragraphe 1.1.1.3.

Le principe de l'évaluation d'option par arbre ou de la résolution de problème de contrôle stochastique réside dans la possibilité de calculer des espérances conditionnelles en tout point de l'espace discrétisé. Ce sera aussi le cas de toutes les quantités markoviennes des variables d'état représentées dans l'espace. Les contrôles recherchés sont spécifiés comme étant markoviens, on peut d'ailleurs souvent montrer qu'ils sont optimaux. Certains problèmes de finance quantitative mettant en jeu des *payoffs* qui dépendent de la trajectoire (dit *path-dependent*) ne sont alors pas adaptés à ces approches.

Le principe de la programmation dynamique peut s'appliquer alors et l'on peut calculer par rétropropagation le prix des options et les stratégies optimales (dans l'espace discrétisé) de couvertures.

L'approche présentée dans le chapitre 3, et esquissée dans le paragraphe 1.3.5.2, est un exemple de problème de contrôle stochastique qui peut être traité par ce type de méthode, bien que nous ayons choisi de présenter ici une utilisation directe d'une technique de gradient stochastique. Il est intéressant de noter que le gradient stochastique est utile pour la construction des grilles de quantification et est naturellement présent si l'on traite le problème de contrôle par quantification.

1.1.2.3 Gradient stochastique

L'optimisation par descente du gradient stochastique permet de s'affranchir de beaucoup de contraintes liées à la dimension. Nous exposons certaines de ces applications dans les parties suivantes. D'une part, des techniques de construction de grilles de quantification sont présentées en 1.2.3. D'autre part des algorithmes d'optimisation de couverture en marché incomplet, sont développées en 1.3.5.2. En dernier lieu nous introduisons l'utilisation du gradient stochastique pour l'optimisation de réseaux neuronaux dans le paragraphe 1.1.3.3.

1.1.3 Le gradient stochastique : applications

1.1.3.1 Robbins Monro : gradient stochastique

Les travaux fondateurs de Robbins et Monro dans [58] introduisent la méthode de gradient stochastique permettant de résoudre des problèmes d'optimisation où la fonction cible est définie comme une espérance. Dans ce cas, à la différence de la méthode classique du gradient de Newton, à chaque étape d'optimisation des paramètres, au lieu de calculer le gradient, on utilise uniquement une variable aléatoire dont l'espérance est le gradient.

Le problème à résoudre est du type

$$\min_{\theta} \mathbb{E} [g(\theta, X)].$$

On se donne une suite $(\gamma_n)_{n \in \mathbb{N}}$ strictement positive, décroissante, non sommable et de carré sommable.

$$\sum \gamma_n > +\infty \text{ et } \sum \gamma_n^2 < +\infty.$$

On initialise θ arbitrairement ou en considérant certaines propriétés limites du problème.

On tire indépendamment des variables aléatoires $(X_{n \geq 0})$.

La procédure de mise à jour est alors :

$$\theta_{n+1} = \theta_n - \gamma_n \frac{\partial g}{\partial \theta}(\theta_n, X_n). \quad (1.4)$$

Sous des hypothèses assez générales, Lelong a précisé dans [34] des propriétés de convergence de cet algorithme. C'est une application du théorème central limite. Différents auteurs ont travaillé sur les propriétés théoriques mais aussi numériques de ces algorithmes, Dufflo dans [20] ainsi que Benveniste, et al. dans [10].

1.1.3.2 L'optimisation d'EDS contrôlées : des couvertures en marché financier incomplet

Comme le présente le chapitre 3 qui a été publié dans [43], on voit la couverture comme la solution à un problème de contrôle stochastique. Dans ce cas, on cherche à minimiser une mesure de risque pour optimiser des choix de couvertures.

Le paramètre θ est alors un vecteur qui permet de définir les contrôles.

Ce problème est numériquement bien plus complexe que la recherche d'une couverture dans un marché complet. Nous discutons dans le paragraphe 1.3.5.2 du lien entre les deux problèmes et présentons plus en détail cette utilisation du contrôle stochastique.

La première colonne de données du tableau 1.1 précise des notations propres à ce problème de contrôle stochastique.

1.1.3.3 Calibration de réseaux neuronaux pour la recherche d'image de milliers à des millions de paramètres à optimiser

Spécification du problème de classification d'images On veut savoir si l'image contient un certain objet ou bien si elle peut être rangée dans une certaine classe correspondant au type de scène. Le nombre de classes peut être de 2 à plusieurs milliers. On considère avoir à notre disposition plusieurs centaines de représentants variés de ces classes. On veut mettre au point un algorithme qui en apprenant sur une partie de ces classes sera capable de reproduire ce choix pour le reste.

Les meilleures solutions en l'état actuel sont les réseaux neuronaux multicouches, appelés aussi *Deep Learning*, accessibles par exemple dans la librairie Caffe [27].

Il s'agit essentiellement de la composition de fonctions affines et d'un type de fonction dite d'activation qui est souvent simplement la partie positive $x \mapsto x_+$.

Nous allons décrire rapidement ces objets dans le tableau 1.1.

Les données d'entrées sont simplement les pixels de l'image, on va dire qu'ils sont les données de la première couche.

Les données d'entrées de la couche n sont les sorties de la couche $n - 1$, chaque fonction (appelé neurone) de la couche n est alors la partie positive d'une combinaison linéaire des sorties de la couche $n - 1$.

On finalise le tout en mettant une dernière couche avec autant de neurones qu'il y a de classes à identifier et on utilise alors une fonction d'activation différente du type indicatrice afin d'obtenir 1 à cette sortie si l'image fait partie de la classe correspondante.

Le nombre de neurones et leur paramétrisation sont une partie importante de l'expertise indispensable à la mise en œuvre de ces outils.

Les paramètres du modèles, de l'ordre de plusieurs dizaines de milliers, sont calibrés par rétropropagation de leur dérivées depuis la dernière couche.

Pour des problèmes de capacité en mémoire vive des ordinateurs, il n'est pas possible d'utiliser toutes les images à chaque étape d'optimisation. Aussi, on utilise un algorithme de gradient stochastique en groupant les réalisations (méthode dite par batch) sur l'ensemble des images disponibles à chaque étape de la procédure d'optimisation des paramètres.

L'idée des réseaux convolutifs Dans ce cas, les paramètres des fonctions des premières couches sont les mêmes pour différentes zones de l'image, de manière à utiliser ces premières couches et chaque sous-zone d'une même image comme un outil d'apprentissage de filtre de bas niveau de compréhension et de codage des images.

Absence de convexité Comme de nombreux problèmes d'optimisation en grande dimension, il n'a pas été démontré que la calibration de modèle convolutif soit convexe. En grande dimension on peut s'étonner que la hessienne ait toutes ses valeurs propres négatives à un endroit où la dérivée s'annule. C'est-à-dire qu'il y a beaucoup de points selles. C'est-à-dire un endroit où le gradient s'annule et la hessienne a des valeurs propres négatives et d'autres positives comme illustré dans la figure 1.1. Notamment dans le cas où l'on rajoute une contrainte \mathbf{L}^2 pour éviter que les paramètres ne partent à l'infini. Cela aura pour effet de redresser des minima locaux en points selles, en agglomérant un minimum local sur au moins un point selle. Mais si ces contraintes sont suffisamment faibles pour garder certaines spécificité du problème de calibration il devrait rester des minima locaux. Notamment dans les zones qui nous intéressent, c'est-à-dire près du minimum global, il pourrait y avoir d'autres minima locaux dus à la convexité global du problème près du minimum global. Dans le paragraphe 5.2.3 nous étudions l'algorithme de Lloyds qui présente aussi des minima locaux.

Similarité entre le problème d'optimisation de portefeuille et le problème de calibration de réseau neuronal Nous proposons dans la table 1.1 un parallèle entre les problèmes d'optimisation de portefeuille traités dans le chapitre 3 et l'optimisation des réseaux neuronaux pour la détection et la classification d'images fruit de notre double expérience. On s'intéresse à la couverture d'une série de Cash Flow aléatoire noté CF_i et à la détection de propriété, notée



FIGURE 1.1 – Point selle savoureux en dimension 2 (par L. Bopp et T. Haettel)

labels l_k dans une image. Nous présentons plus en détail le calcul par indifférence dans le paragraphe 1.3.5.2 et l'utilisation des réseaux neuronaux dans le paragraphe 1.1.3.3. L'apprentissage dans les deux cas se fait par gradient stochastique.

Dans le cas des réseaux convolutifs, de nombreux paramètres définissant les contrôles sont liés entre eux, ou fixés à une valeur nuls.

1.1.3.4 Construction des grilles de quantification

La construction des grilles de quantification repose sur plusieurs méthodes. Parmi elles, les méthodes de type gradient stochastique offrent de nombreux avantages. Nous les présentons dans la partie 1.2.3.1 ainsi que les approches alternatives.

1.1.3.5 Amélioration des méthodes de gradient stochastique en utilisant des batches

Nous présentons des améliorations à la méthode du gradient stochastique présentée dans le paragraphe 1.1.3.1. Par rapport à une descente de gradient déterministe, la méthode de gradient stochastique cherche à économiser les efforts pour le calcul de ce gradient en utilisant qu'une seule réalisation. C'est

	$\mathbb{E}[U]$	Réseau neuronal
T_i	Date de trading	Couche du réseau
$x_k^{(i)}$	Quantités d'actifs k détenue en i	Signal k à la couche i
$\theta_{k,l}^{(i)}$	Contribution de la valeur de l'actif k sur la valeur placée additivement en l'actif l à la date i	A la couche i , contribution du signal d'entrée k au signal de sortie l
$\Psi_l^{(i)}$	Fonction concave, croissante, de la quantité d'actif l obtenue à la date i pour un investissement	Fonction d'activation du signal de sortie (en général indépendante de i et l), convexe $(\cdot)_+$
Aléa	Valeurs des actifs k à la date i : $S_k^{(i)}$ sans coût de transaction	Valeurs des signaux d'entrées : $x_k^{(0)}$ (+ Dropout éventuel) et label l_k
Itération	$x_l^{(i+1)} = \Psi_l^{(i+1)} \left(\frac{\sum_k \theta_{k,l}^{(i)} x_k^{(i)}}{S_l^{(i)}} \right) + x_l^{(i)}$	$x_l^{(i+1)} = \Psi_l^{(i+1)} \left(\sum_k \theta_{k,l}^{(i)} x_k^{(i)} \right)$
Contraintes	Autofinancement : $x_0^{(i+1)} = x_0^{(i)} - \sum_{k,l} \theta_{k,l}^{(i)} x_k^{(i)} + CF_i$	Dans le cas convolutif, de nombreux paramètres sont contraints à 0 et d'autres égaux. Il n'y a pas de cash $x_0^{(i)} = 0$.
Cible à optimiser	Utilité finale $U \left(x_0^{(N)} + \sum_k \Psi_k^{(N)} \left(-\frac{x_k^{(N)}}{S_k} \right) \right)$ avec U concave et croissante. Ou cash initial $x_0^{(0)}$ dans le cas d'un <i>pricing</i> par indifférence	Erreur de prédiction $\sum_k \left(x_k^{(N)} - l_k \right)^2$ (le problème perd systématiquement sa convexité)
Paramètres à optimiser	Cash initial $x_0^{(0)}$ et $\theta_{k,l}^{(i)}$ pour $i \leq 0$ et $k > 0$	$\theta_{k,l}^{(i)}$ pour $i \leq 0$ et $k > 0$

TABLE 1.1 – Comparaison entre les problèmes d'optimisation de portefeuille et les problèmes de calibration de réseaux neuronaux

efficace lorsque les paramètres sont loin d'une zone de minimum local. Cependant dans le cas contraire, faire baisser la variance et éviter de trop grands aléas plutôt que de déplacer les paramètres est important. Une méthode simple est de faire des groupes (ou batchs) de simulations avant chaque modification des paramètres.

On modifie dans ce cas l'équation (1.4), on se donne une taille de batch K , et l'incrément à l'étape n est donné par :

$$\theta_{n+1} = \theta_n - \frac{\gamma_n}{K} \sum_{j=1}^K \frac{\partial g}{\partial \theta}(\theta_n, X_{nK+j}).$$

Une autre méthode plus élaborée consiste à estimer le gradient avec une mémoire.

D'autres approches utilisent des estimateurs des dérivées secondes (appelées matrices Hessiennes) pour accélérer ces méthodes.

On pourra se rapporter aux travaux de Bach dans [4] où Schmidt dans [60] pour des études complètes de ces méthodes.

Il est intéressant de noter que dégénéré à la limite (c'est-à-dire qu'on utilise toute la distribution), la méthode de batch du calcul des grilles de quantifications par gradient stochastique consiste en fait en l'algorithme de Lloyds. Ces différents algorithmes de calculs de grilles sont présentés dans la partie 1.2.3.

1.2 Quantification optimale, quantification fonctionnelle, méthode du k-mean et applications

Le début de cette partie s'inspire de la présentation faite par A. Sellami dans sa thèse en partie publiée dans [52] sur la présentation générale des méthodes de quantification.

Les principales motivations historiques pour étudier et utiliser la quantification proviennent habituellement du traitement du signal et de l'information. C'est une méthode de codage par un ensemble discret d'un signal continu. Une première utilisation a été la transmission par ondes hertziennes de localisations géographiques, puis son utilisation s'est considérablement étendue à des applications plus diverses. Cette partie présente certains résultats théoriques nécessaires à une bonne compréhension de cette méthode, puis expose brièvement différentes utilisations en probabilités numériques ainsi que pour l'indexation de photos.

1.2.1 Définitions et résultats préliminaires

1.2.1.1 Quantification de variables aléatoires

On se donne un espace probabilisé $(\Omega, \mathcal{F}, \mathbb{P})$, ainsi qu'une variable aléatoire X de loi \mathbb{P}_X , à valeurs dans \mathbb{R}^d . On fixe un entier $N > 0$. On appelle N -

quantificateur une application borélienne $\pi_\Gamma : \mathbb{R}^d \rightarrow \mathbb{R}^d$ dont l'image de \mathbb{R}^d est un ensemble fini $\Gamma = \{x^1, \dots, x^N\} \subset \mathbb{R}^d$.

Afin de définir une application π_Γ à partir de ces points, on spécifie une partition $(A^i)_{1 \leq i \leq N}$ dans l'espace \mathbb{R}^d ce qui permet de définir :

$$\pi_\Gamma(X) := \sum_{i=1}^N x^i \mathbb{1}_{A^i}(X).$$

Le N -quantifieur est donc spécifié par la donnée de :

- Une partition borélienne $(A^i)_{1 \leq i \leq N}$ de l'espace \mathbb{R}^d . A chaque ensemble A^i sera associé un centre $x^i \in A^i$.
- $\Gamma = \{x^1, \dots, x^N\}$ appelée grille de quantification de taille N , ensemble des points, centres de quantification ou centroïdes ou tout simplement N -quantifieur associé à π_Γ .

Pour $X \in L^p$, on définit un N -quantifieur L^p -optimal de X par une application π_Γ^* solution du problème d'optimisation paramétré par la taille de la grille de quantification N :

$$\inf \left\{ \mathbb{E} \|X - \pi_\Gamma(X)\|_p^p, \pi_\Gamma : \mathbb{R}^d \rightarrow \mathbb{R}^d, \text{ fonction borelienne t.q. } \#(\pi_\Gamma(\mathbb{R}^d)) \leq N \right\}. \quad (1.5)$$

On peut aussi se rapporter à Kieffer dans [30] ou Cuesta et Matrán dans [17] pour l'existence d'équilibres en dimension finie ou infinie.

D'après les résultats établis par Graf et Luschgy [25], ce problème admet toujours une solution qui implique d'une part le choix d'une grille de quantification optimale vérifiant :

$$\mathbb{E} [|X - \pi_\Gamma^*(X)|^p] = \mathbb{E} \left[\min_{x \in \pi_\Gamma^*(\mathbb{R}^d)} |X - x|^p \right],$$

ainsi qu'une partition optimale $(A^i)_{1 \leq i \leq N}$ qui vérifiera :

$$\mathbb{E} [|X - \pi_\Gamma^*(X)|^p] = \sum_{i=1}^N \int_{x \in A^i} \mathbb{E} [|X - x|^p | X \in A^i] \mathbb{P}(X \in A^i).$$

On définit les cellules de Voronoï :

$$\mathbf{C}_i(\Gamma^*) = \left\{ \xi \in \mathbb{R}^d \text{ t.q. } |\xi - x^i| = \min_{1 \leq k \leq N} |\xi - x^k| \right\}. \quad (1.6)$$

On montre aussi que toute partition associée à un quantifieur optimal est de type Voronoï (voir figure 1.3), i.e. $A^{i^*} = \mathbf{C}_i(\Gamma^*)$ pour $1 \leq i \leq N$.

L'erreur de quantification, que l'on appelle aussi distortion, inertie intra-classe ou inertie locale, s'écrit alors

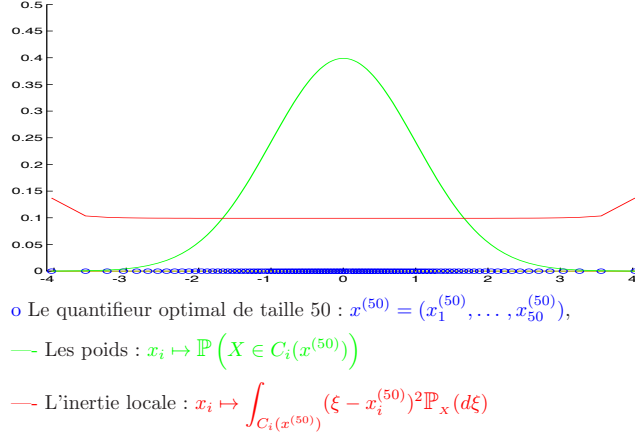
QUANTIFICATION QUADRATIQUE OPTIMALE DE TAILLE 50 DE $\mathcal{N}(0;1)$ 

FIGURE 1.2 – Inertie et poids de quantification optimale en dimension 1 (par J.-C. Fort et G. Pagès)

$$\underline{\mathcal{D}}_N^{X,p} := \mathbf{E} \|X - \pi_\Gamma^*(X)\|_p^p = \mathbf{E} \left\| \min_{1 \leq i \leq N} |X - x^i| \right\|_p^p \quad (1.7)$$

Par extension, pour une quantification non optimale, on parlera aussi d'inertie locale. Elle est utilisée dans les méthodes de construction qui sont détaillées dans le paragraphe 1.2.3.

Elle converge vers zéro lorsque la taille du quantifieur N tend vers $+\infty$, et ce taux de convergence est donné par le théorème de Zador que nous rappelons.

Il est intéressant de noter que pour les quantifieurs optimaux cherchent à égaliser cette inertie locale mais pas forcément leur probabilité. La figure 1.2 illustre ces différentes propriétés.

Théorème 2.a. On suppose que $\int_{\mathbb{R}^d} |\xi|^{p+\eta} \mathbb{P}_X(d\xi) < +\infty$ pour $\eta > 0$. Alors,

$$\lim_{N \rightarrow +\infty} \left(N^{\frac{p}{d}} \underline{\mathcal{D}}_N^{X,p} \right) = J_{p,d} \|\varphi\|_{\frac{d}{d+p}}$$

où $J_{p,d} > 0$ et $\mathbb{P}_X(d\xi) = \varphi(\xi) \lambda_d(d\xi) + \bar{\mu}(d\xi)$, $\bar{\mu} \perp \lambda_d$ (λ_d mesure de Lebesgue sur \mathbb{R}^d) et pour tout $q \in \mathbb{R}_+$, $\|g\|_q := \left(\int |g|^q(u) du \right)^{1/q}$

Théorème 2.b. Sous les mêmes hypothèses, il existe une constante universelle $C_{d,p,\eta}$ telle que :

$$\underline{\mathcal{D}}_N^{X,p} \leq C_{d,p,\eta} \sigma_{p+\eta}(X) N^{-\frac{p}{d}}$$

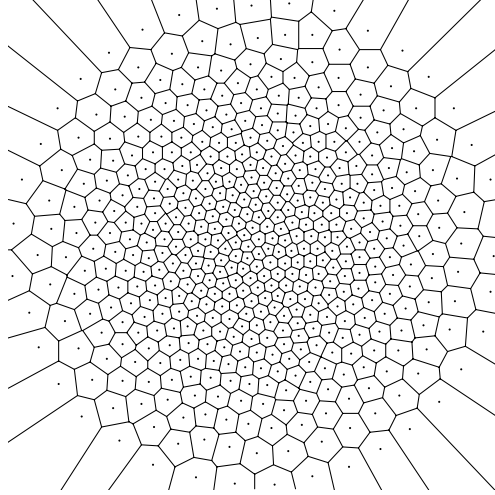


FIGURE 1.3 – Cellules de Voronoï en dimension 2 (par G. Pagès et J. Printems)

où

$$\sigma_{p+\eta}(X) = \inf_{a \in \mathbf{R}^d} \mathbf{E} \left[\|X - a\|_{p+\eta} \right]$$

Une preuve de la variante 2.a par Graf et Luschgy se trouve dans [25] tandis que la version générale non asymptotique se trouve dans [41].

Nous pouvons ainsi écrire que $\|X - \pi_{\Gamma}^*(X)\|_p = O(N^{-1/d})$ au voisinage de $+\infty$.

Par ailleurs, il sera utile de noter que les quantifieurs L^2 -optimaux vérifient une propriété dite de stationnarité, à savoir que :

$$\mathbf{E}[X | \pi_{\Gamma}^*(X)] = \pi_{\Gamma}^*(X). \quad (1.8)$$

Cette propriété permet d'utiliser des termes correcteurs de premier ordre dans les différentes applications de la quantification, comme nous allons le voir dans le paragraphe suivant pour l'intégration numérique puis plus loin dans les exemples d'évaluation d'options américains [6].

D'un point de vue pratique, définir la fonction $\pi_{\Gamma}^*(N)$ pour une taille de quantifieur fixée N s'avère être un problème d'optimisation assez délicat à résoudre.

Des méthodes de résolution numérique peuvent cependant être utilisées pour trouver des solutions approchées au problème, on citera par exemple la méthode

du point fixe ou celle de la descente de gradient, qui deviennent cependant difficiles à mettre en œuvre en dimensions supérieures à 1. Quelques autres méthodes numériques peuvent être adoptées dans des cas particuliers de lois à quantifier (exponentielles, $\gamma(a)$, $B(a, b)$) par exemple par Fort et Pagès dans [22]. Enfin, les algorithmes stochastiques d'apprentissage (*Competitive Learning Vector Quantization CLVQ algorithm*) offrent une bonne alternative aux problèmes posés par les méthodes précédentes. L'algorithme de Lloyds dont nous donnons une présentation plus complète dans le paragraphe 1.2.3.1 et des extensions présentées dans le paragraphe 5.4.2 s'avèrent particulièrement efficaces en grande dimension.

Nous introduisons à présent les processus quantifiés.

1.2.1.2 Arbre de quantification d'un processus de Markov

Nous présentons ici des méthodes de construction d'arbres dont l'utilisation pour l'estimation de prix et la couverture d'option a été présentée dans le paragraphe 1.1.2.2. Dans les différentes applications de la quantification, il est souvent requis de considérer la quantification d'un processus markovien à temps discret $(X_k)_{k \geq 0}$ dont on sait simuler la dynamique d'évolution. Une approche possible dans ce cas est de quantifier chaque variable X_k en tenant compte de sa loi marginale, on parle donc de *quantification marginale*. Pour cela, on doit se fixer une taille de grille N_k à chaque pas de temps et un N_k -quantifier L^p -optimal de $X_k \in L^P$ qu'on notera $\Gamma_k = \{x_k^1, \dots, x_k^{N_k}\}$. Par conséquent, on définit :

$$\hat{X}_k = \sum_{i=1}^{N_k} x_k^i \mathbb{1}_{C_i(\Gamma_k)}(X_k).$$

Le processus quantifié $(\hat{X}_k)_{k \geq 0}$ ne vérifie plus a priori la propriété de Markov. Des tests numériques ou des études théoriques montrent que ce n'est souvent pas le cas. On pourra se rapporter à Pagès et al. dans [48] pour comprendre les problèmes que cela implique et la façon de les éviter. Cependant, une approximation de la probabilité de transition entre différents états à deux dates successives reste possible à travers les paramètres compagnons p_k^{ij} , pour $1 \leq i \leq N_k$ et $1 \leq j \leq N_{k+1}$:

$$\begin{aligned} p_k^{ij} &= \mathbb{P}[X_{k+1} \in C_j(\Gamma_{k+1}) | X_k \in C_i(\Gamma_k)] \\ &= \mathbb{P}[\hat{X}_{k+1} = x_{k+1}^j | \hat{X}_k = x_k^i]. \end{aligned}$$

D'une manière générale, pour $0 \leq k \leq n$ et $1 \leq i \leq N_k$, on notera :

$$\hat{\mathbf{P}}_k f(x_k^i) = \mathbb{E} \left[f(\hat{X}_{k+1}) | \hat{X}_k = x_k^i \right] = \sum_{j=1}^{N_{k+1}} f(x_{k+1}^j) p_k^{ij}.$$

Cependant comme (\hat{X}_k) converge vers (X_k) quand $N_k \rightarrow +\infty$, (\hat{X}_k) a une dynamique de plus en plus markovienne. De plus si (X_k) est une chaîne de Markov stationnaire, les transitions (p_k^{ij}) ne dépendent plus de k .

Pour des horizons n pas trop élevés, il est donc possible de calculer et de stocker dans des tables facilement accessibles les grilles de quantifications et les paramètres compagnons. Ce pré-traitement des données, dit *off-line*, permet de minimiser les calculs d'éventuels estimateurs utilisant la quantification.

Construction des grilles de quantification par moment matching et calcul matriciel Pour des horizons pas trop faibles, il est possible de calculer ces paramètres compagnons, dans le cas du Brownien par exemple, sans effectuer de Monte-Carlo, simplement à l'aide du calcul matriciel ainsi que le présente Villemin dans [62].

Il s'agit d'utiliser les propriétés de composition du générateur pour une distribution stationnaire qu'on peut ramener à une diffusion brownienne multidimensionnelle de manière explicite et un moment matching pour estimer les probabilités de transition sur la grille.

On choisit un pas de temps assez petit. Les contraintes provenant du moment matching sur les probabilités de transition sont des contraintes linéaires sur les probabilités de transition. On obtient ensuite par composition ou calcul de la puissance de la matrice (grâce à l'exponentielle et au logarithme matriciel) les probabilités de transition pour tout pas de temps strictement supérieur.

Des tests numériques montrent que la convergence est bonne et la méthode efficace.

Dans ce cas, la difficulté principale réside dans la dépendance en temps de la matrice de variance-covariance des processus gaussiens.

Si elle est au moins constante par morceaux, on pourra donner une bonne approximation, certainement convergente, des transitions à partir de données précalculées indépendamment des paramètres du modèle.

Une contrainte numérique résidera comme nous l'avons évoqué au début du paragraphe dans le fait que le pas de temps ne soit pas trop faible. Les contraintes sont en effet difficiles à respecter voire impossibles quand le pas tend vers 0. Il est intéressant de noter que c'est un problème que l'on rencontre aussi pour l'estimation des probabilités de transition par Monte-Carlo. En effet, si l'horizon est trop faible, les points extrêmes de la grille seront peu atteints et l'estimation des transitions à partir de ces points très biaisée.

1.2.2 Application à l'intégration numérique : évaluation d'options et contrôle stochastique

La quantification est un outil essentiel de la théorie de l'information, le traitement du signal ainsi que de sa compression. Son intérêt pour les problèmes d'intégration numérique est à présent indéniable.

De fait, l'industrie de la finance l'utilise pour l'évaluation de produits dérivés.

On a déjà présenté dans le paragraphe 1.1.2.2 l'utilisation de la programmation dynamique dans le cas des arbres de quantification. Le pré-traitement qu'elle fait des paramètres compagnons notamment est très important dans ce cas-là.

1.2.2.1 Prix d'options européennes

Une application immédiate de la quantification est le calcul d'approximations numériques d'intégrales par rapport à une mesure donnée. On se pose le problème d'évaluation de l'intégrale $\mathbb{E}[f(X)]$, pour X de loi donnée absolument continue par rapport à la mesure de Lebesgue, de densité \mathbf{p} . Si \hat{X} désigne une N -quantification L^2 -optimale de X nous pouvons nous donner comme estimateur $\mathbb{E}[f(\hat{X})]$. Comme \hat{X} est une variable aléatoire discrète, le calcul de l'estimateur se résumera à une somme pondérée finie. En reprenant les notations des paragraphes précédents, on pose

$$\mathbb{E}[f(X)] \approx \mathbb{E}[f(\hat{X})] = \sum_{i=1}^N f(x^i) \int \mathbb{1}_{C_i(\Gamma)}(x) \mathbf{p}(x) dx = \sum_{i=1}^N f(x^i) \hat{\mathbf{p}}^i$$

Les pondérations $\hat{\mathbf{p}}^i$ sont aussi des paramètres compagnon qui peuvent être calculés en même temps que la grille de quantification Γ et stockés dans des tables accessibles pendant l'estimation.

L'erreur d'estimation est contrôlée par l'erreur de quantification $\Delta = X - \hat{X}$. En effet, quand $f \in \mathbf{C}_b^1$, il existe $\xi \in (X, \hat{X})$ tel que :

$$f(X) - f(\hat{X}) = \langle Df(\xi), \Delta \rangle.$$

Ce qui donne la majoration d'erreur qu'on appellera *d'ordre zéro* :

$$\left| \mathbb{E}[f(X)] - \mathbb{E}[f(\hat{X})] \right| \leq C \|\Delta\|_1 \leq C \|\Delta\|_2 \quad (1.9)$$

Quand $f \in \mathbf{C}_b^2$, on peut développer f à un ordre supérieur afin d'établir une majoration d'erreur *d'ordre 1*. En effet, il existe $\xi \in (X, \hat{X})$ tel que :

$$f(X) - f(\hat{X}) = \langle Df(\hat{X}), \Delta \rangle + \frac{1}{2} \Delta' D^2 f(\xi) \Delta.$$

Ainsi, comme \hat{X} vérifie la propriété de stationnarité (1.8), on montre que :

$$\begin{aligned} \left| \mathbb{E}[f(X)] - \mathbb{E}[f(\hat{X})] \right| &\leq \mathbb{E} \left[\left| \mathbb{E}[f(X) - f(\hat{X}) | \hat{X}] \right| \right] \\ &\leq C \mathbb{E}[\|\langle \Delta, \Delta \rangle\|] \leq C \|\Delta\|_2^2 \end{aligned} \quad (1.10)$$

En utilisant le théorème de Zador, (cf théorème 3.a), on obtient un taux de convergence en $O\left(N^{-\frac{1}{d}}\right)$ dans le cas de l'inégalité (1.9), et moyennant une

hypothèse plus restrictive sur la fonction f ($\|D^2 f\| < \infty$), on a un taux de convergence deux fois plus rapide $O\left(N^{-\frac{2}{d}}\right)$ à partir de (1.10). Rappelons que l'on note

$$\|D^2 f\| = \sup_{x \in \mathbb{R}^d} \sup_{\|u\|=1} |u^* D^2 f(x) u|.$$

1.2.2.2 Considérations pratiques

L'intégration numérique par quantification est une méthode qui s'approche dans son principe des méthodes de Monte Carlo : elle s'appuie sur une représentation de la loi de X par un ensemble discret fini pondéré. L'estimateur de Monte Carlo s'écrit en effet comme somme équipondérée d'un nombre fini M d'échantillons i.i.d. :

$$\mathbb{E}[f(X)] \approx \frac{1}{M} \sum_{i=1}^M f(X^i) \quad \text{où } (X^1, \dots, X^M) \text{ iid } \sim \mathbf{P}_X.$$

Mais si le principe reste le même, de grandes différences séparent les deux méthodes :

- Les grilles de quantification ainsi que les pondérations peuvent être calculées à froid (i.e. *off-line*), et stockées dans des tables accessibles par plusieurs applications à la fois. La complexité du calcul exclut donc la procédure d'optimisation des quantifieurs et compte seulement les opérations élémentaires de somme et de pondération. Au contraire, les méthodes de Monte Carlo utilisent une partie de la capacité de calcul dans la simulation en direct des échantillons X^i .
- L'estimateur Monte Carlo est un estimateur aléatoire, dont il faudra gérer la variance lors des applications par des procédures de contrôle et de minimisation de variance. A son opposé, l'estimateur par quantification est un estimateur déterministe.
- Le taux de convergence des estimateurs de Monte Carlo est en $O\left(N^{-\frac{1}{2}}\right)$. Il est indépendant de la dimension. Pour les méthodes de quantification, la convergence, bien que dépendant de la dimension, reste intéressante en dimension 2 si elle est de type *ordre 0* et en dimension 4 si elle est de type *ordre 1*.

Cependant nous proposons dans le chapitre 2, présenté rapidement dans le paragraphe 1.3.2.2, une approche qui garde le meilleur de ces deux approches dites par variable de contrôle quantifiée.

1.2.2.3 Quantification fonctionnelle

La quantification fonctionnelle a été introduite par Luschgy et al. dans [39], l'étude asymptotique des dimensions critiques de quantification a été effectuée dans [40]. Des travaux récents se trouvent aussi dans [42].

Nous traitons dans le chapitre 5 de l'optimisation de la quantification de descripteurs de photos qui sont décrits dans des espaces de dimension de l'ordre de quelques milliers. Nous présentons certaines des propriétés de la quantification optimale asymptotique du brownien et des solutions d'EDS.

On considère une diffusion de type (4.1) et de solution X , ainsi qu'un *payoff* Υ dépendant de la trajectoire jusqu'à une date T , le but de la quantification fonctionnelle est d'approcher rapidement $\mathbb{E}[\Upsilon(X^{app})]$ où X^{app} est une approximation du type schéma d'Euler de X . Nous détaillons dans le paragraphe 1.3.1.1, la construction et certaines propriétés de ces schémas.

L'idée est de remplacer la distribution des trajectoires $X \in \mathbf{L}^2([0, T])$ (c'est-à-dire de carré intégrable, parce que continue par exemple) par une distribution discrète de taille N .

Elle utilise la décomposition de Karhunen–Loève du Brownien sur une base Hilbertienne de $\mathbf{L}^2([0, T])$.

Dans [41], Luschgy et Pagès proposent une quantification des diffusions à partir de la décomposition de Karhunen–Loève.

D'après la décomposition de Karhunen–Loève, pour $0 \leq t \leq T$:

$$B_t = \sum_{n \in \mathbb{N}} \xi_n \sin\left(\pi \frac{t}{T} \left(n - \frac{1}{2}\right)\right) \sqrt{\frac{2}{T}} \frac{T}{\pi(n - 1/2)},$$

où ξ_n sont des Gaussiennes indépendantes centrées et réduites.

On choisit alors une troncature k de cette série ainsi qu'une méthode de quantification. On peut utiliser une quantification produit comme nous le faisons dans [33], c'est-à-dire en quantifiant chaque marginale. En grande dimension, cela introduit des problèmes de temps de calculs dus à la combinatoire de la taille des données à manipuler comme ceux évoqués dans le cas des équations aux dérivées partielles dans le paragraphe 1.1.1.1. Cependant la littérature et les tests numériques suggèrent d'utiliser une troncature assez faible de la décomposition de Karhunen–Loève. On peut aussi optimalement utiliser une quantification jointe, comme nous le présentons dans le paragraphe 1.2.1.1.

On obtient alors un ensemble fini de trajectoires continues $\{\hat{B}^{(1)}, \dots, \hat{B}^{(N)}\}$ qui sont une bonne approximation de la distribution du brownien B sur l'intervalle $[0, T]$.

Wong et Zakai dans [63] présentent la convergence des processus solution de l'EDS vers l'EDO de Stratonovich. Dans [41], les auteurs prouvent une vitesse de convergence sous différentes hypothèses en $O(\log(N))^{-\frac{1}{2}+\epsilon}$, $\forall \epsilon > 0$.

L'EDS peut être quantifiée "naturellement" sous réserve de la mettre sous sa forme Stratonovich ci-dessous, c'est-à-dire d'en modifier le drift dans le cas d'une écriture unidimensionnelle :

$$d\hat{X}_t = \tilde{b}(t, \hat{X}_t) dt + \sigma(t, \hat{X}_t) dB_t - \frac{1}{2} \sigma(t, \hat{X}_t) \frac{\partial \sigma}{\partial x}(t, \hat{X}_t) dt.$$

On se donne une méthode de discrétisation de cette equation différentielle ordinaire \hat{X}^{app} .

On définit l'espace $\mathcal{C}_b^r([0, T] \times \mathbb{R}^d)$, pour $r > 0$, l'ensemble des fonctions bornées $f : [0, T] \times \mathbb{R}^d \rightarrow \mathbb{R}^d$, différentiables $[r]$ fois dont les dérivées partielles jusqu'à l'ordre $[r]$ sont bornées et dont les dérivées partielles d'ordre $[r]$ sont Hölderienne de norme $r - [r]$.

On a alors un théorème de convergence :

Théorème 3. En utilisant les définitions ci-dessus, avec b et σ dans $\mathcal{C}_b^{2+\alpha}$ pour $\alpha > 0$

$$\mathbf{E} \left[\Upsilon \left(\hat{X}^{app} \right) \right] \xrightarrow[k, N \rightarrow +\infty]{} \mathbf{E} [\Upsilon (X^{app})]$$

Ce théorème est par exemple une conséquence du théorème 7 de Pagès dans [45] mais on peut aussi trouver des travaux précurseurs dans [41].

Par souci de complétude, on pourra aussi se rapporter à Printems et Pagès dans [50] ou [16] ou encore [41] pour une démonstration.

Nous développons des technique basées sur la quantification fonctionnelle dans le chapitre 2 pour servir de variable de contrôle, nous présentons rapidement ce résultat dans le paragraphe 1.3.2.2.

1.2.2.4 Approximation des espérances conditionnelles vers une méthode de contrôle stochastique

Pham et Pagès ont présenté dans [48] et [51] une méthode permettant de rendre Markovien les arbres de probabilités construits grâce aux paramètres compagnons. Ces arbres présentent de nombreux avantages comme nous avons présenté l'utilisation dans le paragraphe 1.1.2.2 et leur construction dans le paragraphe 1.2.1.2.

Pagès et al. dans [46] ont montré pour cette utilisation des formules de majoration de l'erreur

$$\left\| \mathbf{E} [f(X) | Y] - \mathbf{E} [f(\hat{X}) | \hat{Y}] \right\|_2^2 \leq [f]_{lip}^2 \|X - \hat{X}\|_2^2 + [Pf]_{lip}^2 \|Y - \hat{Y}\|_2^2$$

où $Pf(y) = \mathbf{E} [f(X) | Y = y]$

1.2.2.5 Variable de contrôle naturelle pour la réduction de variance

Nous montrons dans le chapitre 2 publié dans [33] comment utiliser les avantages combinés de la méthode de Monte Carlo et de la méthode d'approximation de prix obtenus par la quantification fonctionnelle. Nous la présentons plus en détail dans le paragraphe 1.3.2.2.

1.2.3 Méthode de construction : Lloyds et gradient stochastique

1.2.3.1 Méthode de construction des poids de quantification

Nous allons présenter plus en détail deux méthodes de construction des grilles de quantification.

Algorithme de Lloyds L'algorithme de Lloyds publié dans [37] est basé sur la propriété de stationnarité (1.8).

Il consiste en une initialisation puis une itération de deux étapes :

- On initialise des éléments au hasard où à partir de la grille optimale à $N - 1$ points à laquelle nous rajoutons un élément : $\Gamma_{(0)} = \{x_{(0)}^1, \dots, x_{(0)}^N\}$ que nous appelons centroïdes.
- On effectue M étapes, à l'étape k :
 - on recalcule les clusters, c'est-à-dire qu'on recherche l'ensemble des éléments les plus proches de chaque centroïde : $C_1(\Gamma_{(k)}), \dots, C_N(\Gamma_{(k)})$
 - On met à jour les centroïdes par la formule $x_{(k+1)}^i = \mathbb{E}[X|C_i(\Gamma_{(k)})]$ pour $1 \leq i \leq N$.
- Le résultat consiste en l'ensemble $\Gamma_{(M)} = \{x_{(M)}^1, \dots, x_{(M)}^N\}$.

La fonction que nous optimisons n'est pas convexe, mais cette descente du gradient assure que nous convergions vers une limite qui vérifie la propriété de stationnarité (1.8).

Dans le chapitre 5 nous présentons des améliorations de cet algorithme pour quantifier des distributions discrètes en très grande dimension (de plusieurs centaines à plusieurs milliers).

On pourra se rapporter à des travaux théoriques récents de Pagès et Yu [53] ou Du et al. [19] pour obtenir des éléments sur la convergence de cet algorithme.

L'idée importante de l'algorithme est qu'à chaque pas, l'inertie locale décroît. L'équation (5.1) la définit, on l'appelle aussi distortion.

Algorithme de Gradient stochastique Il existe un algorithme de gradient stochastique pour construire ces centroïdes. On utilise comme dans tout algorithme stochastique une suite γ_k ayant les propriétés définies dans le paragraphe 1.1.3.1. A chaque étape, on obtient une réalisation indépendante des précédentes X_k de la loi à quantifier, on cherche le centroïde $x_{(k)}^{i_k} := \pi_{\Gamma}^*(X_k)$ le plus près du tirage et on met à jour ce centroïde selon la formule : $x_{(k+1)}^{i_k} = (1 - \gamma_k)x_{(k)}^{i_k} + \gamma_k X_k$, les autres centroïdes restent fixes. Profitant aussi de la propriété de stationnarité cet algorithme converge vers un point fixe de (1.8) et ne nécessite pas de stocker de grandes quantités de données à chaque étape.

Pour une étude on pourra se reporter notamment à Pagès dans [44].

Une méthode hybride utilise un batch à chaque étape de l'algorithme, c'est-à-dire que l'on groupe plusieurs tirages avant de mettre à jour les centroïdes qui nécessitent de bouger. Ce type d'approche est présenté un peu plus longuement dans le paragraphe 1.1.3.5.

1.2.4 Création de cohorte de représentants homogènes : une méthode de *clustering* non paramétrique

Nous présentons dans le paragraphe 5.2 l'utilisation des algorithmes de quantification pour indexer une base de données de medias. Elle repose notamment sur le fait de créer des clusters, ou groupe de données homogènes.

Il est dit non-paramétrique, car il permet simplement d'extraire une structure simple d'une topologie donnée sous la forme d'une distance euclidienne. Par exemple, pour l'application d'indexation de medias, son utilisation permet de grouper des photos qui se ressemblent. Nous nous intéressons aussi à la construction des arbres de recherches hiérarchiques et leur auto-apprentissage dans le paragraphe 5.3.1.

1.3 Méthode d'estimation approchée et exacte pour les prix et couvertures d'actifs financiers

1.3.1 Processus stochastique et espérance pour l'estimation de prix et de couverture en finance

Depuis les travaux précurseurs menés par Black et Scholes dans [12] dans les années 1970, l'utilisation de méthodes probabilistes dans le monde financier s'est amplifiée.

Elles consistent à utiliser des processus stochastiques pour représenter des trajectoires futures de prix d'actifs et d'instruments financiers.

Les principes de marché complet, de viabilité et d'absence d'opportunité d'arbitrage permettent de proposer l'évaluation de prix d'options sur instruments financiers sous la forme d'espérance.

Sous ces hypothèses, on peut alors montrer l'existence et l'unicité de la probabilité sous laquelle on calcule ces espérances, elle est appelée probabilité risque-neutre ou implicite.

Sa distribution est estimée par des techniques de calibration sur les prix observés dans le marché, on parle alors de paramètres implicites pour définir cette probabilité.

L'adjectif se rattache souvent à la volatilité dite implicite.

On peut voir très simplement le *smile* de volatilité, c'est-à-dire la fonction de volatilité implicite comme une fonction qui définit la distribution. Les praticiens sur les marchés financiers sont habitués à regarder les distributions sous le prisme de cette mesure plutôt que de la fonction de repartition ou la densité.

1.3.1.1 Méthode de Monte-Carlo

La méthode de Monte-Carlo couplée aux méthodes de discrétisation d'équations différentielles stochastiques permettent d'obtenir des prix d'options sur actifs financiers. Ces méthodes introduisent deux types d'erreurs :

- Une erreur due à la discrétisation des processus : biais

— Une erreur due à la variance de l'estimateur : erreur statistique

Nous présentons dans le chapitre 1.3.2 des méthodes pour réduire l'erreur due à la variance simplement appelées réduction de variance. Nous présentons dans la chapitre 1.3.3 des méthodes permettant d'éviter l'erreur de discrétisation des processus.

Convergence des méthodes de discrétisation par le schéma d'Euler-Maruyama Les travaux de Bally et Talay [7] présentent des hypothèses sous lesquelles on peut expliciter des ordre de convergences faibles aux méthodes de discrétisations des EDS de type Euler.

Théorème 4. Considérant une EDS de type

$$dX_t = b(X_t) dt + \sigma(X_t) dB_t$$

Le schéma d'Euler de pas δ_t s'écrit :

$$\bar{X}_{(i+1)\delta_t} = \bar{X}_{i\delta_t} + b(X_{i\delta_t}) \delta_t + \sigma(X_{i\delta_t}) (B_{(i+1)\delta_t} - B_{i\delta_t})$$

Pour une fonction f mesurable et bornée, et si b et σ sont \mathbf{C}^∞ dont toutes les dérivées sont bornées. Une hypothèse supplémentaire provenant des conditions d'Hörmander sur b et σ doit être respectée. En dimension 1, il est par exemple suffisant que b et σ ne s'annulent pas au même point. Elle est liée au fait que la diffusion n'est pas dégénérée dans un sous-espace non trivial strictement contenu dans l'espace où elle est définie.

On a l'existence de fonction C_f et Q_{δ_t} et des constantes m, Q, q et une fonction $K(T)$ dépendant de b et σ et de leurs dérivées jusqu'à l'ordre m tel que :

$$\mathbf{E}[f(X_T^x)] - \mathbf{E}\left[f\left(\hat{X}_T^x\right)\right] = -C_f(T, x) \delta_t + Q_{\delta_t}(f, T, x) \delta_t^2$$

et

$$|C_f(T, x)| + \sup_{\delta_t} |Q_{\delta_t}(f, T, x)| \leq K(T) \|f\|_\infty \frac{1 + \|x\|^Q}{T^q}$$

La démonstration utilise les techniques du calcul de Malliavin. Ces propriétés permettent de mettre en place des schémas de Romberg pour supprimer les premiers ordres d'erreur.

On pourra trouver dans [61] une démonstration et une variante dans [8] sans hypothèse d'ellipticité si f est régulière.

1.3.2 Réduction de variance

Les méthodes de réduction de variance consistent à modifier la variance d'une variable aléatoire sans modifier son espérance de manière à pouvoir estimer avec plus de précision cette espérance. On peut souligner que ce n'est pas seulement la variance qu'il est important de pouvoir faire baisser, mais le facteur variance

fois temps de simulation de manière à pouvoir obtenir une précision plus grande dans le même temps de simulation. On appelle "effort" ce produit variance fois complexité. Il arrive que ce ne soit pas que le temps de simulation, mais le coût de calculs (dû essentiellement à la consommation électrique) qu'il est important d'optimiser, nous n'aborderons pas ce dernier aspect et ces implications dans la suite. On pourrait appeler "effort durable" ce produit là.

1.3.2.1 Stratification

Stratification de Vol sto La stratification est une méthode de réduction de variance consistant à stratifier les trajectoires, c'est-à-dire les grouper et à être capable de simuler plus de trajectoires conditionnellement à certains événements (ou strates) qui ont une plus grande variance et ainsi réduire la variance de l'ensemble plus efficacement. Dans un modèle du type Heston, on peut par exemple stratifier par réalisation de la volatilité et considérons notamment les ratios de temps de simulations pour optimiser notre stratégie de calcul de l'espérance.

Jourdain et al. ont présenté dans [29] les méthodes de stratification pour des variables aléatoires vectorielles et des moyens d'en optimiser l'efficacité.

1.3.2.2 Variables de contrôle

Les méthodes de variables de contrôle sont une autre méthode classique de réduction de variance. Il s'agit de rajouter un terme d'espérance nul dans et de s'arranger pour que sa covariance avec l'estimateur qui nous intéresse ne soit pas nulle. Nous présentons deux méthodes de construction de ces variables de contrôles.

Arouna : variable de contrôle construite par la technique de Robbins Monro grâce au gradient stochastique Un des premiers problèmes en finance quantitative qui a vu l'intérêt du gradient stochastique consistait à l'utiliser pour chercher des variables de contrôles dans le but de réduire la variance d'estimateur de Monte-Carlo.

En s'appuyant sur les travaux présentés dans le paragraphe 1.1.3.1, Arouna a proposé dans [3] une manière de choisir des paramètres d'une variable de contrôle.

Dans ce cas, les paramètres à optimiser influent sur la corrélation entre le *payoff* dont on cherche à estimer l'espérance et la variable de contrôle.

On peut s'intéresser aussi aux travaux précurseurs ou étendant certains résultats par Jourdain et Lelong dans [28], par Lemaire et Pagès dans [35] mais surtout Lapeyre et Lelong dans [32].

Variable de contrôle quantifiée D'autres travaux récents [16] ont été faits dans des directions similaires.

La partie 1.2 présente la méthode de quantification, certaines de ses applications et de ses méthodes de construction. Elle permet d'approcher des distributions de variables aléatoires vectorielles ou de distribution de processus par des

distributions discrètes. Dans les deux cas cela permet de construire des variables de contrôle comme le présente le chapitre 2.

L'idée est d'associer chaque réalisation provenant de la distribution continue dans un espace vectoriel ou un espace de fonction à un représentant de l'espace de probabilité discrétisé. On connaît précisément la probabilité d'associer chacune de ces trajectoires à un des représentants discrets, ces probabilités ont été calculées simplement pour chaque processus et ne dépendent pas du *payoff* dont on cherche à calculer l'espérance, on les appelle paramètres compagnons. La variable de contrôle consiste en la valeur du *payoff* sur l'espace discrétisé dont on soustrait le *payoff* sur chaque réalisation simulée au cours de la simulation de Monte-Carlo.

Le fait que les probabilités d'associer chaque trajectoire à un des représentants de la distribution discrétisée est intrinsèquement lié à la faculté de l'algorithme de réduire la variance de notre estimateur. Le calcul *off-line* de ces probabilités ou poids est fait sur des variables gaussiennes normalisées pour lesquelles on a pu les calculer efficacement et avec une grande précision. Les approches proposées dans la quantification fonctionnelle permettent de construire de manière explicite les représentants discrets des processus dont on cherche à évaluer une espérance.

Vu sous un autre angle, on peut aussi utiliser le fait que les variances au sein de chaque cellule de Voronoï, c'est-à-dire les ensembles de réalisation associé à un représentant de la distribution discrétisée, sont faibles par rapport aux variances inter-classes, autrement dit de la distribution complète et continue qui nous intéresse initialement. Dans ce cas plutôt que de parler de méthode de variable de contrôles pour la réduction de variance, on pourrait dire que la méthode consiste à enlever le biais dû à la discrétisation propre aux approximations d'espérance obtenus par les méthodes de quantification.

$$\frac{1}{NMC} \left(\underbrace{\sum_{j=0}^{NMC} \Upsilon(X_{(j)}^{app}) + \sum_i \underbrace{\hat{p}_i}_{\text{Paramètres compagnons}} \Upsilon(\hat{X}_i^{app}) - \sum_{j=0}^{NMC} \Upsilon(\pi_\Gamma(X_{(j)}^{app}))}_{\text{Variable de contrôle}} \right)$$

On peut voir aussi cette variable de contrôle comme le rajout d'un terme pour enlever le biais de la quantification :

$$\frac{1}{NMC} \left(\sum_i \underbrace{\hat{p}_i}_{\text{Paramètres compagnons}} \Upsilon(\hat{X}_i^{app}) + \sum_{j=0}^{NMC} (\Upsilon(X_{(j)}^{app}) - \Upsilon(\pi_\Gamma(X_{(j)}^{app}))) \right)$$

Le premier terme est déterministe, il s'agit de l'approximation quantifiée du prix.

Il est intéressant d'interpréter le deuxième terme comme l'estimation de l'erreur de quantification dont une majoration est donnée dans l'équation (1.10).

On peut s'intéresser à la variance de l'estimateur. En effet l'efficacité de cette méthode de réduction de variance est liée à la variance de ce deuxième terme. On peut majorer cette variance par le produit du carré du *payoff* Υ dès qu'il est borné ou au moins de carré intégrable (par Cauchy-Schwarz) et le terme que l'on appelle l'inertie, distortion ou erreur de quantification définie dans l'équation (5.1).

1.3.3 Méthode exacte : simulation sans biais

Nous présentons ici différentes méthodes de calculs exacts ou du moins sans biais. Ces dernières sont développées plus en détail dans le chapitre 4.

1.3.3.1 Quelques remarques sur les formules fermées

La plus connue des formules fermées d'évaluation de prix d'option est la formule de Black-Scholes. Il s'agit du calcul du prix des options dites *CALL* et *PUT* dans le cadre du modèle log-normal de distribution du prix des actifs. Le logarithme des actifs suit une loi gaussienne. Ces options sont aussi appelées options vanilles. Il existe dans d'autres cas des formules ou méthodes permettant par exemple dans le modèle d'Heston de calculer des prix d'options de manière semi-fermée. La chapitre 2 s'applique entre autre au modèle d'Heston qui a été présenté initialement dans [26]. Il s'agit alors d'utiliser le fait que les transformées de Fourier ou de Laplace des prix des actifs peuvent être calculées de manière explicite. Il reste à effectuer une intégration numérique pour calculer le prix des options CALL et PUT. La rapidité d'exécution pour une précision donnée de ces méthodes est importante pour permettre d'utiliser ces modèles et ces options dans le cadre de méthode de calibration où l'on cherche à trouver des paramètres du modèle reproduisant les prix des instruments qui sont facilement observables, les options vanilles.

Méthode de calcul de prix par transformée de Fourier On considère une diffusion de type (4.1) qui représente le logarithme des prix d'un actif. La méthode de transformée de Fourier est utile lorsque l'on sait calculer la quantité $\hat{C}(u) = \mathbf{E}[e^{iuX_T}]$.

On a alors :

Proposition 3. Si X_T admet une densité continue et des moments de tous ordres :

$$\mathbf{E}\left[(e^{X_T} - e^k)_+\right] = \frac{e^{\alpha k}}{2\pi} \int_{\mathbb{R}} e^{-iuv} \frac{\hat{C}(iu + \alpha + 1)}{(iu + \alpha)(iu + \alpha + 1)} du,$$

où α est un paramètre numérique strictement positif à choisir.

On peut intégrer numériquement cette intégrale unidimensionnelle par différentes méthodes, on pourra se ramener aux travaux de Carr [14] pour cette question. Une alternative à cette intégration numérique est d'utiliser les technique

de transformées de fourier discrète appelées communément FFT qui permettent de calculer en une transformation d'un vecteur de transformée de Fourier en un vecteur de prix d'options pour différents *strikes*, en d'autres termes de calculer tout le *smile* d'un seul coup.

Cette formule est utile par exemple pour une diffusion où b et $\sigma^*\sigma$ sont des fonctions affines du processus. En effet dans ce cas, la transformée de Fourier \tilde{C} est solution d'équation différentielle ordinaire du type Riccati. Elle peut se résoudre de manière explicite en une dimension, pour des modèles de type Wishart sur lesquels ont par exemple travaillé Ben Abid et al. dans [9] ou Palidda [54]. Les modèles dits quadratiques gaussiens permettent aussi d'être résolus explicitement comme dans le travail de El Karoui et al. dans [21].

D'autres modèles nécessitent une résolution numérique de ces équations qui reste néanmoins performante.

On peut alors utiliser ces méthodes pour calibrer les prix d'options vanilles sur ceux du modèle appelé alors exotique, données par la diffusion considérée.

1.3.3.2 Méthode de simulation trajectorielle exacte : Beskos-Roberts

La méthode proposée par Beskos et Roberts dans [11] permet de simuler des trajectoires de processus provenant d'équations différentielles stochastiques sans le biais de la discrétisation. Elle s'applique à des diffusions en une dimension dont la volatilité peut dépendre du processus mais uniquement du temps de manière constante sur des intervalles. On applique tout d'abord la transformation de Lamperti de manière à ramener le processus à un processus ayant une volatilité de 1. Pour une diffusion du type

$$dX_t = b(X_t) dt + \sigma(X_t) dB_t,$$

la transformée de Lamperti s'écrit (sous des hypothèses d'existence) $x \mapsto \int_0^x \frac{dy}{\sigma(y)}$.

Ensuite, une utilisation habile du théorème de Girsanov (changement de probabilité) permet alors de ramener le problème de simulation du processus à la simulation d'un brownien qu'on doit pondérer par l'exponentielle d'une intégrale en temps. On peut en effet supprimer la partie stochastique dans ce dernier terme grâce à une intégration par partie possible en dimension 1. La dernière étape de cette approche s'appuie sur le fait que, dans le cas où l'intégrand respecte certaines propriétés sur ces bornes aux bords ($\pm\infty$), on peut transformer ce terme en l'espérance d'événement relatif à la réalisation de variable de Poisson qu'on peut simuler sans biais.

Ceci rajoute donc un bruit dans le calcul des espérances en supprimant un biais.

Le travail présenté dans le chapitre 4 supprime certaines conditions sur les conditions de bornes aux bords. Il propose en plus une majoration de l'erreur dans beaucoup d'autres cas. Dans les cas où les termes de dérives dans l'équation différentielle stochastique sont polynomiaux, cette erreur est exponentielle en

l'opposé d'un paramètre dont le temps de calcul dépend linéairement. C'est-à-dire, que, de manière très peu coûteuse on peut réduire cette erreur dans des proportions très importantes.

Nous présenterons d'autres développements sur cette méthode dans le paragraphe 1.3.4.3.

1.3.3.3 Méthodes avancées de simulation

Alfonsi a développé depuis quelques années des techniques de simulation d'ordre élevés voire exacte par composition de générateurs. On peut citer [1]. Il existe aussi des méthodes multi-levels qui cherchent à supprimer les biais de discrétisation de schémas d'Euler en tenant parti du fait que les estimations des biais entre des schémas de pas de temps différents ont une variance assez faible. D'autre part on peut aussi citer des méthodes aussi simples que la tabulation de la fonction de répartition pour simuler efficacement avec une précision souhaitée.

1.3.4 Et la couverture ?

1.3.4.1 Marché complet AOA : la dérivée du prix est une couverture

La communauté de mathématiciens financiers et de praticiens appelle *hedge* ou couverture, les portefeuilles, stratégie ou quantité permettant de se couvrir dynamiquement contre le risque de marché et ainsi l'annuler. La théorie de Black Scholes montre que sous certaines hypothèses, la couverture d'une option est simplement la dérivée du prix de cette option par rapport à la valeur initiale (à la date d'aujourd'hui) des instruments auxquelles elle est sensible. Bien que les praticiens utilisent couramment des approximations par différences finies pour estimer ces couvertures, un enjeu important du monde académique consiste à la calculer efficacement, sans biais et de manière efficace.

1.3.4.2 Calcul de Malliavin enlevant le biais des dérivées : différenciez, différenciez : il restera toujours quelque chose des processus

Introduit par Fournié et al. pour l'estimation de *hedge* dans [23], la théorie du calcul de Malliavin permet de calculer des sensibilités par rapport à la valeur initiales des processus de prix d'actifs sans différences finies. Ils consistent à rajouter un terme dans l'espérance en facteur des *payoffs* et ne dépendent pas de ces derniers.

Le calcul de Malliavin utilise des notions telles que la dérivée du flot et des méthodes d'intégration par parties utilisant la dérivée du flot. La dérivée du flot peut être comprise comme étant la dérivée par rapport à la valeur du brownien à un instant donné. C'est un outil probabiliste qui a de nombreuses applications, dans le cas présent pour le calcul de sensibilité d'espérances dépendant d'un paramètre. Cela peut aussi servir au calcul d'espérance conditionnelle ou, comme nous l'évoquons dans le paragraphe 1.3.1.1, pour l'étude de la convergence faible du schéma de discrétisation d'Euler.

La formule principale indique pour une diffusion de type (4.1), pour σ constant par simplicité :

$$\frac{\partial \mathbb{E}[f(X(T))]}{\partial x_0} = \frac{1}{T} \mathbb{E} \left[f(X(T)) \int e^{\int_0^s \frac{\partial b}{\partial t}(u, X(u)) du} dB_s \right]$$

Cette formule fonctionne en dimension supérieure et s'adapte avec σ non constant.

1.3.4.3 Malliavin sans biais

Le chapitre 4 présente une utilisation originale du calcul de Malliavin dans le cadre de simulation trajectorielle exacte de diffusion 1D, de manière à pouvoir calculer diverses sensibilités sans biais. Elle est appliquée aux ordres un et deux pour calculer les quantités appelées par les praticiens de la finance quantitative delta et gamma.

Les possibilités du calcul de Malliavin d'estimer des sensibilités à l'aide d'un poids multiplicatif, sans hypothèse sur le *payoff*, ni sa dérivabilité, ni le fait qu'il dépende d'une seule valeur de l'actif ou de toute la trajectoire est utilisé dans tout son potentiel en parallèle à la méthode de simulation trajectorielle exacte évoquée dans le paragraphe 1.3.3.2.

1.3.4.4 La couverture est un contrôle optimal

Dans la partie 1.1 nous mettons en exergue différentes propriétés et perspectives dans l'utilisation du gradient stochastique pour le calcul de couverture optimale. Il convient ici de souligner que cette couverture est la solution d'un problème de contrôle stochastique.

1.3.5 Les approximations en finance et les imperfections de marché

Cette partie s'achève en discutant de différents liens entre les approximations numériques et leurs impacts avec les imperfections de marchés.

1.3.5.1 Les approximations numériques modifient le marché : les options CMS

Rappel et définition Les options sur instruments de taux les plus communes et qualifiées de vanille sont les swaptions. Les swaps sont des instruments échangeant durant une durée fixe des paiements de taux fixes contre des paiements de taux variables. La partie fixe présente la propriété de ne pas avoir de risque sur les intérêts à rembourser, mais l'on conserve un risque sur le capital car en cas de variation des taux, c'est la valeur de ce qu'il nous reste à rembourser qui va varier. Pour la partie variable, c'est le contraire. Pouvoir échanger l'un contre l'autre est donc utile et correspond à un SWAP comme nous l'avons dit.

Définition 4. Le taux de swap s'écrit

$$SWAP(t, T_0 < T_1 < \dots < T_N) = \frac{B(t, T_0) - B(t, T_N)}{\sum_{i=1}^N (T_i - T_{i-1}) B(t, T_i)}.$$

Une swaption donne le droit de faire cet échange à une date fixée. Selon qu'il s'agit d'un swaption payeur ou receveur (équivalent de CALL et PUT), il donnera le droit d'emprunter à taux fixe, ou de prêter à taux fixe. On peut aussi le définir comme une option sur taux de swap qui paie sous une mesure appelée LEVEL, dont le numéraire est défini comme la partie fixe du swap.

L'option CMS L'option CMS paie une formule en général linéaire du taux de swap à la date où il est fixé et non selon le LEVEL. Dégénéré à une date, l'on passe d'un caplet (ou floorlet) à une *Libor in Arrear*. Dans les deux cas il fait intervenir un changement de probabilité, de la probabilité T_2 forward à la probabilité T_1 forward pour un Caplet donnant le droit en T_1 d'emprunter jusqu'en T_2 . Dans le cas du swaption, il s'agit d'un changement de probabilité du LEVEL à la probabilité forward correspondant à la date de fixing du taux de swap.

On note $S(t)$ le taux de swap et $L(t)$ le level associé pour $t \leq T_F < T_0 < T_1 < \dots < T_N$ régulièrement espacé d'une année. On note $P(t, T)$ l'obligation zéro-coupon, $C(t)$ l'option CMS, \mathbb{Q} la probabilité risque neutre, \mathbb{Q}_T , la probabilité forward et \mathbb{Q}_L la probabilité level. On utilise la formule de changement de numéraire.

$$\begin{aligned} L(t) &= \sum_{1 \leq j \leq N} B(t, T_j) \\ S(t) &= \frac{B(t, T_0) - B(t, T_N)}{L(t)} \\ S(t) + 1 &= \sum_{i=1}^N \frac{B(t, T_i)}{\sum_{1 \leq j \leq N} B(t, T_j)} \frac{B(t, T_{i-1})}{B(t, T_i)} \\ C(t) &= \mathbb{E}^{\mathbb{Q}} \left[e^{-\int_t^{T_F} r(s) ds} S(T_F) \right] \\ &= B(t, T_F) \mathbb{E}^{\mathbb{Q}_{T_F}} [S(T_F)] \\ &= L(t) \mathbb{E}^{\mathbb{Q}_L} \left[\frac{S(T_F)}{L(T_F)} \right] \\ L(t) &\approx \frac{1}{a + bS(t)} \\ C(t) &\approx L(t) \mathbb{E}^{\mathbb{Q}_L} [aS(T_F) + bS^2(T_F)]. \end{aligned}$$

Le changement de probabilité correspond à l'inverse de l'instrument considéré. Si on peut l'utiliser facilement tel quel dans le cas du calcul du prix du

Libor In Arrear, on fait communément une approximation dans le cas de l'option CMS. Le but étant que le prix de l'option ne dépende que de la distribution unidimensionnelle du taux de swap qu'on peut observer de manière implicite sous la probabilité LEVEL risque-neutre, c'est-à-dire observer à partir des prix du marché. La formule exacte présente l'option CMS comme une option sur la distribution jointe de tous les taux forwards (Libor forward) sous-jacents. On peut considérer que cela correspond à une option sur panier.

Cette approximation numérique a un effet sur le marché car la formule approchée étant massivement utilisée pour estimer le prix des options CMS, c'est en fait la formule approchée qui est cotée par le marché et non la formule exacte bien plus compliquée.

1.3.5.2 Les imperfections vu comme une perturbation autour de la théorie du *pricing* de Black-Scholes

Le cadre de la théorie d'évaluation de Black-Scholes permet d'attribuer des prix à des produits dérivés. Cependant, plusieurs hypothèses classiques, ne sont pas toujours satisfaites sur les marchés. La complétude du marché, les intervenants sont de petits acteurs (ie leurs décisions n'ont pas d'influence sur les cours), la prise en compte des risques de crédit ne donne pas les mêmes opportunités à tous les acteurs.

Sous-entendant que la réalité est perfectible, on appelle cela des imperfections de marchés, en considérant que ce sont des différences avec une théorie idéale.

Le contrôle stochastique apporte une perturbation à la couverture de Black-Scholes Une première approche qui est notamment avancée dans la prise en compte des différentes imperfections de marchés est de les rajouter comme des différences indépendantes de la diffusion principale des prix de marché. Nous avons présenté et cité l'approche de Cètin et al. dans le paragraph 1.1.1.4.

Nous avons alors proposé une approche théoriquement correcte de prise en compte d'imperfections de marchés. Dans les cas où elles étaient faibles, nous cherchons les contrôles optimaux comme une déformation autour des contrôles optimaux dans la théorie de Black-Scholes.

Dans ce cas présent nous résolvons numériquement ce problème de contrôle stochastique en utilisant une technique de gradient stochastique.

Nous présentons dans le paragraphe suivant une extension pour une meilleure prise en compte des sensibilités au risque de chaque investisseur.

Le prix par indifférence, vers un prix sur mesure Dans un cadre de marché non complet où il n'est pas possible de transférer tous les risques au marché il est intéressant de considérer que chaque intervenant est sensible à une mesure de risque différente qu'il cherche à optimiser. Le prix par indifférence consiste à trouver la somme qu'un acteur est prêt à accepter en échange d'un risque. La somme minimale qui rend l'échange indifférent à sa mesure de risque

est le prix par indifférence. On comprend alors que le prix des différentes options n'est plus unique entre les différents acteurs.

On cherche la somme p optimisant :

$$\min_{(\max_{\pi, X_0(0)=p} \mathbb{E}[U(X^\pi(T))]) \geq U(0)} p$$

On a intérêt à transformer la contrainte sur l'utilité en une pénalisation pour la résolution numérique.

On pourra se rapporter à différentes références pour une étude plus complète de ce problème. L'auteur a travaillé sur des problème similaire dans [43], mais il a été aussi abordé par exemple par El Karoui et al dans [24].

Il est intéressant de noter que dans le cas de la prise en compte des taux d'intérêt, le changement de probabilité de la probabilité forward à la probabilité risque-neutre n'est pas trivial et offre plusieurs choix qui ne sont pas seulement des outils de calculs. c'est-à-dire que cela ne revient pas au même de considérer le facteur d'actualisation à l'intérieur ou à l'extérieur de l'utilité, nous en discutons dans [2].

Bibliographie

- [1] Aurélien Alfonsi. Affine Diffusions and Related Processes : Simulation, Theory and Applications, volume 6. Springer, 2015.
- [2] Aurélien Alfonsi, Benjamin Jourdain, Sophie Laruelle, Sebastian Niklitschek-Soto, and Victor Reutenauer. On two numerical problems in applied probability : discretization of stochastic differential equations and optimization of an expectation depending on a parameter. In ESAIM : Proceedings, volume 44, pages 260–275. EDP Sciences, 2014.
- [3] Bouhari Arouna. Adaptative monte carlo method, a variance reduction technique. Monte Carlo Methods and Applications mcma, 10(1) :1–24, 2004.
- [4] Francis Bach and Eric Moulines. Non-strongly-convex smooth stochastic approximation with convergence rate $o(1/n)$. In Advances in Neural Information Processing Systems, pages 773–781, 2013.
- [5] Achref Bachouch, Emmanuel Gobet, and Anis Matoussi. Empirical regression method for backward doubly stochastic differential equations. SIAM/ASA Journal on Uncertainty Quantification, 4(1) :358–379, 2016.
- [6] Vlad Bally, Jacques Printems, and Gilles Pagès. First-order schemes in the numerical quantization method. Mathematical finance, 13(1) :1–16, 2003.
- [7] Vlad Bally and Denis Talay. The law of the euler scheme for stochastic differential equations. Probability theory and related fields, 104(1) :43–60, 1996.
- [8] Vlad Bally and Denis Talay. The law of the Euler scheme for stochastic differential equations. II. Convergence rate of the density. Monte Carlo Methods Appl., 2(2) :93–128, 1996.
- [9] Anas Benabid, Harry Bensusan, and Nicole El Karoui. Wishart stochastic volatility : Asymptotic smile and numerical framework. 2008.
- [10] Albert Benveniste, Michel Métivier, and Pierre Priouret. Adaptive algorithms and stochastic approximations, volume 22 of Applications of Mathematics (New York). Springer-Verlag, Berlin, 1990. Translated from the French by Stephen S. Wilson.
- [11] A. Beskos, O. Papaspiliopoulos, and G. O. Roberts. Retrospective exact simulation of diffusion sample paths with applications. Bernoulli, 12(6) :1077–1098, 2006.

- [12] Fischer Black and Myron Scholes. The pricing of options and corporate liabilities. The journal of political economy, pages 637–654, 1973.
- [13] Billy L Buzbee. A fast poisson solver amenable to parallel computation. IEEE Transactions on Computers, 22(8) :793–796, 1973.
- [14] Peter Carr and Dilip Madan. Option valuation using the fast fourier transform. Journal of computational finance, 2(4) :61–73, 1999.
- [15] Cetin, U., Jarrow, R., Protter, P., Warachka, and M. Pricing options in an extended black scholes economy with illiquidity : Theory and empirical evidence. Review of Financial Studies, 19(2) :493–529, 2006.
- [16] Sylvain Corlay and Gilles Pagès. Functional quantization-based stratified sampling methods. Monte Carlo Methods Appl., 21(1) :1–32, 2015.
- [17] J. A. Cuesta and C. Matrán. The strong law of large numbers for k -means and best possible nets of Banach valued random variables. Probab. Theory Related Fields, 78(4) :523–534, 1988.
- [18] Arnaud Debussche and Jacques Printems. Weak order for the discretization of the stochastic heat equation. Mathematics of computation, 78(266) :845–863, 2009.
- [19] Qiang Du, Maria Emelianenko, and Lili Ju. Convergence of the lloyd algorithm for computing centroidal voronoi tessellations. SIAM journal on numerical analysis, 44(1) :102–119, 2006.
- [20] Marie Duflo. Algorithmes stochastiques, volume 23 of Mathématiques & Applications (Berlin) [Mathematics & Applications]. Springer-Verlag, Berlin, 1996.
- [21] Nicole El Karoui, Ravi Myneni, and Ravi Viswanathan. Arbitrage pricing and hedging of interest rate claims with state variables i : theory. preprint, University of Paris VI, 1992.
- [22] Jean-Claude Fort and Gilles Pagès. Asymptotics of optimal quantizers for some scalar distributions. J. Comput. Appl. Math., 146(2) :253–275, 2002.
- [23] E. Fournié, J.-M. Lasry, J. Lebuchoux, P.-L. Lions, and N. Touzi. Applications of Malliavin calculus to Monte Carlo methods in finance. Finance Stoch., 3(4) :391–412, 1999.
- [24] Helyette Geman, Nicole El Karoui, and Jean-Charles Rochet. Changes of numeraire, changes of probability measure and option pricing. Journal of Applied probability, pages 443–458, 1995.
- [25] Siegfried Graf and Harald Luschgy. Foundations of quantization for probability distributions. 2000.
- [26] S. Heston. A closed-form solution for options with stochastic volatility with applications to bond and currency options. Rev. Financ. Stud., 6(2) :327–343, 1993.
- [27] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe : Convolutional architecture for fast feature embedding. arXiv preprint arXiv :1408.5093, 2014.

- [28] Benjamin Jourdain and Jérôme Lelong. Robust adaptive importance sampling for normal random vectors. Ann. Appl. Probab., 19(5) :1687–1718, 2009.
- [29] Benjamin Jourdain, Jérôme Lelong, et al. Robust adaptive importance sampling for normal random vectors. The Annals of Applied Probability, 19(5) :1687–1718, 2009.
- [30] John C. Kieffer. Exponential rate of convergence for Lloyd’s method. I. IEEE Trans. Inform. Theory, 28(2) :205–210, 1982.
- [31] Damien Lamberton and Bernard Lapeyre. Introduction to stochastic calculus applied to finance. CRC press, 2007.
- [32] Bernard Lapeyre and Jérôme Lelong. A framework for adaptive Monte Carlo procedures. Monte Carlo Methods Appl., 17(1) :77–98, 2011.
- [33] Antoine Lejay and Victor Reutenauer. A variance reduction technique using a quantized brownian motion as a control variate. The Journal of Computational Finance, 16(2) :61–84, 2012.
- [34] J. Lelong. Asymptotic normality of randomly truncated stochastic algorithms. ESAIM Probab. Stat., 17 :105–119, 2013.
- [35] Vincent Lemaire and Gilles Pagès. Unconstrained recursive importance sampling. Ann. Appl. Probab., 20(3) :1029–1067, 2010.
- [36] Jacques-Louis Lions, Yvon Maday, and Gabriel Turinici. Résolution d’edp par un schéma en temps «pararéel». Comptes Rendus de l’Académie des Sciences-Series I-Mathematics, 332(7) :661–668, 2001.
- [37] Stuart Lloyd. Least squares quantization in pcm. IEEE transactions on information theory, 28(2) :129–137, 1982.
- [38] Francis A Longstaff and Eduardo S Schwartz. Valuing american options by simulation : a simple least-squares approach. Review of Financial studies, 14(1) :113–147, 2001.
- [39] Harald Luschgy and Gilles Pagès. Functional quantization of Gaussian processes. J. Funct. Anal., 196(2) :486–531, 2002.
- [40] Harald Luschgy and Gilles Pagès. Sharp asymptotics of the Kolmogorov entropy for Gaussian measures. J. Funct. Anal., 212(1) :89–120, 2004.
- [41] Harald Luschgy and Gilles Pagès. Functional quantization of a class of brownian diffusions : a constructive approach. Stochastic Processes and their Applications, 116(2) :310–336, 2006.
- [42] Harald Luschgy and Gilles Pagès. Constructive quadratic functional quantization and critical dimension. Electron. J. Probab., 19 :no. 50, 19, 2014.
- [43] Christophe Michel, Victor Reutenauer, Denis Talay, and Etienne Tanré. Liquidity costs : a new numerical methodology and an empirical study. Applied Mathematical Finance, 23(1) :57–79, 2016.
- [44] Gilles Pagès. A space quantization method for numerical integration. Journal of computational and applied mathematics, 89(1) :1–38, 1998.

- [45] Gilles Pagès. Quadratic optimal functional quantization of stochastic processes and numerical applications. In Monte Carlo and Quasi-Monte Carlo Methods 2006, pages 101–142. Springer, 2008.
- [46] Gilles Pagès. Introduction to vector quantization and its applications for numerics. In CEMRACS 2013—modelling and simulation of complex systems : stochastic and deterministic approaches, volume 48 of ESAIM Proc. Surveys, pages 29–79. EDP Sci., Les Ulis, 2015.
- [47] Gilles Pagès, Vlad Bally, and Jacques Printems. A quantization tree method for pricing and hedging multidimensional american options. Mathematical Finance, 15 :119–168, 2005.
- [48] Gilles Pagès, Huyèn Pham, and Jacques Printems. An optimal markovian quantization algorithm for multi-dimensional stochastic control problems. Stochastics and Dynamics, 4(04) :501–545, 2004.
- [49] Gilles Pagès, Olivier Pironneau, and Guillaume Sall. The parareal algorithm for American options. C. R. Math. Acad. Sci. Paris, 354(11) :1132–1138, 2016.
- [50] Gilles Pagès and Jacques Printems. Functional quantization for numerics with an application to option pricing. Monte Carlo Methods and Applications mcma, 11(4) :407–446, 2005. <http://quantization.math-phi.com/>.
- [51] Gilles Pagès and Jacques Printems. Optimal quantization for finance : from random vectors to stochastic processes. Handbook of Numerical Analysis, 15 :595–648, 2009.
- [52] Gilles Pagès and Afef Sellami. Convergence of multi-dimensional quantized sde’s. In Séminaire de probabilités XLIII, pages 269–307. Springer, 2011.
- [53] Gilles Pagès and Jun Yu. Pointwise convergence of the lloyd algorithm in higher dimension. 2013.
- [54] Ernesto Palidda. Modélisation du smile de volatilité pour les produits dérivés de taux d’intérêt. PhD thesis, Paris Est, 2015.
- [55] Huyèn Pham. Continuous-time stochastic control and optimization with financial applications, volume 61. Springer Science & Business Media, 2009.
- [56] David Pommier. Méthodes numériques sur des grilles sparse appliquées à l’évaluation d’options en finance. PhD thesis, Paris 6, 2008.
- [57] Jacques Printems. On the discretization in time of parabolic stochastic partial differential equations. ESAIM : Mathematical Modelling and Numerical Analysis, 35(6) :1055–1078, 2001.
- [58] Herbert Robbins and Sutton Monro. A stochastic approximation method. The Annals of Mathematical Statistics, 22(3) :400–407, September 1951.
- [59] LCG Rogers. Duality in constrained optimal investment and consumption problems : a synthesis. In Paris-Princeton Lectures on Mathematical Finance 2002, pages 95–131. Springer, 2003.

- [60] Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. arXiv preprint arXiv :1309.2388, 2013.
- [61] Denis Talay and Luciano Tubaro. Expansion of the global error for numerical schemes solving stochastic differential equations. Stochastic Anal. Appl., 8(4) :483–509 (1991), 1990.
- [62] François-Xavier Villemin. Pricing methods and quantization. 2010.
- [63] Eugene Wong and Moshe Zakai. On the convergence of ordinary integrals to stochastic integrals. The Annals of Mathematical Statistics, 36(5) :1560–1564, 1965.

Chapitre 2

Réduction de variance par quantification

Résumé

Ce travail présente une nouvelle méthode de réduction de variance pour des processus de diffusion où la variable de contrôle est construite en utilisant une quantification des coefficients de la décomposition de Karhunen-Loève du mouvement brownien sous-jacent. Cette méthode peut être utilisée pour des processus non gaussiens.

Ce travail a été publié dans [LR12] et effectué dans le cadre d'une collaboration entre l'équipe-projet TOSCA d'INRIA et l'équipe de recherche et développement dérivés de Taux et Hybrides du Credit Agricole CIB.

Abstract

This article exposes a new variance reduction technique for diffusion processes in which a control variate is constructed using a quantization of the coefficients of the Karhunen-Loève decomposition of the underlying Brownian motion. This method may indeed be used for other Gaussian processes.

This work was published in [LR12] and created from a scientific collaboration between TOSCA Team of INRIA and Hybrid and Interest Rates Quantitative Research and Development Team of Credit Agricole Corporate And Investment Bank.

Introduction

This article aims at exposing a new variance reduction technique for Monte Carlo simulations for computing expectations of quantities related to Brownian motion or solutions of stochastic differential equations (SDE). Monte Carlo methods are sometimes the only available tool to perform simulations. It is customary to use SDE to model the evolution of stocks prices and rates in mathematical finance, since analytical or semi-analytical formulae are not available except for simple options. Yet they are in general naturally and simply implemented. The prices of derivatives are obtained by numerically averaging over a large number of realizations of stock prices or rates. The precision of a Monte Carlo method depends on the variance of the random quantity whose expectation is sought. A large number of methods, called variance reduction techniques, have been proposed to obtain better estimators and to reduce the computational time. There is no need to underline the practical importance of such techniques (see for example the books [12] or [9]) which are still an active subject of research.

In order to simulate efficiently SDEs, we propose in this article a way to construct a control variate using quantization of Gaussian random variables. This scheme may be combined with standard techniques to simulate SDEs (Euler, Milstein, ...). We do not discuss here the problem of the simulation of SDE *per se* on which a large literature exists.

Quantization techniques consist in replacing a random variable X by a discrete one \widehat{X} which takes its values in a finite set of points $\{x_1, \dots, x_Q\}$ and is chosen to be optimal. For example the points $\{x_i\}_{i=1}^Q$ are such that $\mathbb{E}[|X - \widehat{X}|^2]$ is minimal. With $p_i = \mathbb{P}[\widehat{X} = x_i]$, $\mathbb{E}[\phi(\widehat{X})] = \sum_{i=1}^Q p_i \phi(x_i)$ is an approximation of $\mathbb{E}[\phi(X)]$. Taking advantage of the way \widehat{X} is chosen, Q could be much smaller than the number of realizations required to perform a Monte Carlo method to compute $\mathbb{E}[\phi(X)]$ with the same precision. Initially, quantization techniques have been developed in the field of signal theory. Recently, several applications to efficient simulation of diffusion processes and Monte Carlo methods have been proposed (see for example [1, 29, 21, 2, 7, 5] and the survey articles [28, 26]). A mathematical treatment of this theory can be found in [14].

Let X be the solution to some SDE living in $\mathbb{R}^{n'}$ and Υ be a functional on the space $\mathcal{C}([0, T]; \mathbb{R}^{n'})$ of continuous functions. We are interested in computing the quantity

$$J = \mathbb{E}[\Upsilon(X)]. \quad (2.1)$$

Under suitable hypotheses, provided that the SDE

$$X_t = x + \int_0^t \sigma(X_s) dB_s + \int_0^t b(X_s) ds \quad (2.2)$$

solved by X has a unique strong solution for a n -dimensional Brownian motion, the Yamada-Watanabe theorem [15] states that there exists a measurable map Ψ on $\mathcal{C}([0, T]; \mathbb{R}^n)$ such that $X = \Psi(B)$. Hence,

$$J = \mathbb{E}[\Upsilon \circ \Psi(B)].$$

Although no simple expression holds for this map Ψ in general, various schemes (Euler scheme, Milstein scheme, ...) allow one to construct an approximation X^{app} of X . With a scheme which depends on the underlying Brownian motion B , we write $X^{\text{app}} = \Psi^{\text{app}}(B)$. In general, Ψ^{app} is not a function of the whole path B but only of a part of it. For example, the Euler scheme (or the Milstein scheme when $n = 1$) depends only on the Gaussian vector $(B_{t_1^n} - B_0, \dots, B_{t_n^n} - B_{t_{n-1}^n})$ with $t_k^n = kT/n$.

The SDE (2.2) may be written under its Stratonovich form

$$X_t = x + \int_0^t \sigma(X_s) \circ dB_s + \int_0^t c(X_s) ds$$

with, for $\sigma = (\sigma_{i,j})_{i,j=1,\dots,n}$,

$$c_i(x) = b_i(x) - \sum_{\substack{j=1,\dots,n' \\ k=1,\dots,d}} \frac{1}{2} \frac{\partial \sigma_{i,k}(x)}{\partial x_j} \sigma_{j,k}(x).$$

For a piecewise smooth path U , one can also consider the ordinary differential equation (ODE)

$$Y_t = x + \int_0^t \sigma(Y_s) dU_s + \int_0^t c(Y_s) ds. \quad (2.3)$$

If σ is a function of class \mathcal{C}^2 and c a function of class \mathcal{C}^1 which have at most a linear growth, then the ODE has a unique solution Y and there exists a measurable map Θ on the set $\mathcal{C}_p^1([0, T]; \mathbb{R}^n)$ of continuous functions that are piecewise \mathcal{C}^1 such that $Y = \Theta(U)$. For a family $(U^n)_{n \in \mathbb{N}}$ of paths in $\mathcal{C}_p^1([0, T]; \mathbb{R}^n)$ that converges uniformly to a realization $B(\omega)$ of the Brownian motion, it is not always true that $\Theta(U^n)$ converges to the realization $X(\omega) = \Psi(B)(\omega)$. Wong-Zakai's theorem [15] provides some cases where the convergence holds. In dimension $n = 1$, this will be essentially true due to the approach of H. Doss [8] and H. Sussmann [30]. When $n > 1$, the recent theory of rough paths helps us to understand when the convergence holds, under stronger conditions on the regularity of the coefficients [22, 23, 17, 16, 10, 11]. The family of paths $(U^n)_{n \in \mathbb{N}}$ should then be carefully chosen.

Throughout this article, we assume that the coefficient σ is of class \mathcal{C}^2 with a bounded first order derivative and a second order derivative which is ϵ -Hölder continuous for some $\epsilon > 0$. The coefficient c is assumed to have a bounded first order derivative which is moreover ϵ -Hölder continuous. Note that these hypotheses may be too strong if the dimension of the underlying Brownian motion is equal to $n = 1$, where there is no need of the Hölder continuity of the derivatives of the coefficients. In addition, in the case of the Heston model, this condition is not true. However, this is not a real issue when we consider paths for the volatility that stay away from zero. We simply need these regularity conditions in order to apply the theory of rough paths that gives a pathwise interpretation of SDEs. Of course, one may take advantage of smoother coefficients in order to use sophisticated schemes (Milstein, Runge-Kutta, ...) to

solve numerically the SDE X and the ODE Y , while we only use the Euler scheme in our numerical examples.

The algorithm we propose relies on the following steps :

- Do N times
 - Simulate an approximation $X^{\text{app}} = \Psi^{\text{app}}(B)$ of X with a numerical scheme (Euler scheme, ...). Although we are free to choose the scheme, one shall record the realization of the vector of the marginals $\beta = (B_{t_1}, \dots, B_{t_d})$ of the underlying Brownian motion at a given set of times. This is our only constraint on the scheme.
 - With the help of a quantization technique, compute from β a smooth approximation \widehat{B} of the Brownian motion and compute an approximation $\Upsilon(\Theta^{\text{app}}(\widehat{B}))$ of $\Upsilon(\Theta(\widehat{B}))$ by solving numerically the ODE (2.3) with $U = \widehat{B}$.
 - Use $\Upsilon(\Theta^{\text{app}}(\widehat{B}))$ as a control variate for $\Upsilon(X^{\text{app}})$. This last quantity is itself an approximation of $\Upsilon(X) = \Upsilon(\Psi(B))$.
- Return the mean of the quantities $\Upsilon(X^{\text{app}}) - \Upsilon(\Theta^{\text{app}}(\widehat{B})) + \mathbb{E}[\Upsilon(\Theta^{\text{app}}(\widehat{B}))]$.

Written in some abstract way, the classical Monte Carlo estimator J^N for J in (2.1) is

$$J^N = \frac{1}{N} \sum_{i=1}^N \Upsilon \circ \Psi^{\text{app}}(B^{(i)}),$$

for N independent copies $B^{(1)}, \dots, B^{(N)}$ of the underlying Brownian motion. Note that this estimator may be biased due to the use of the approximation scheme Ψ^{app} , but we do not discuss this classical issue here. The variance of J^N is equal to $\text{Var}(\Upsilon \circ \Psi^{\text{app}}(B))/N$, and depends on the functions Ψ^{app} and Υ .

The estimator given by our algorithm is — with a minor improvement carried by the choice of good values of κ —,

$$\widehat{J}^N = \frac{1}{N} \sum_{i=1}^N (\Upsilon \circ \Psi^{\text{app}}(B^{(i)}) - \kappa(\Upsilon \circ \Theta^{\text{app}}(\widehat{B}^{(i)}) - \mathbb{E}[\Upsilon \circ \Theta^{\text{app}}(\widehat{B})])),$$

where $\widehat{B}^{(i)}$ is a copy of the path \widehat{B} associated to $B^{(i)}$ through its marginals at given times (t_1, \dots, t_d) . The constant κ may be taken equal to 1. But one can also choose the optimal constant

$$\kappa_{\text{opt}} = \text{Cov}(\Upsilon \circ \Psi^{\text{app}}(B), \Upsilon \circ \Theta^{\text{app}}(\widehat{B})) / \text{Var}(\Upsilon \circ \Theta^{\text{app}}(\widehat{B})), \quad (2.4)$$

which can be approximated using for example some schemes presented in [9, 31]... In order to simplify our presentation, we set $\kappa = 1$.

The feature of a good control variate associated to a Brownian motion B consists in finding \widehat{B} such that the realizations of \widehat{B} are close to the corresponding realizations of B and $\mathbb{E}[\Upsilon \circ \Theta(\widehat{B})]$ can easily be computed.

In our case, we are willing to find a finite number of trajectories which “summarize” the behavior of the Brownian motion. A possibility consists in using

a finite number of marginals $\beta = (B_{t_0}, B_{t_1}, \dots, B_{t_d})$ of the Brownian motion, to quantize the Gaussian vector of increments $(B_{t_1} - B_{t_0}, \dots, B_{t_d} - B_{t_{d-1}})$ and then to use linear interpolations to link the points. Another possibility consists in using for \widehat{B} a quantization of the truncated version of the Karhunen-Loève decomposition, that is

$$\widehat{B}_t = \sum_{i=1}^m \sqrt{\lambda_i} \widehat{\xi}_i f_i(t) \text{ for } t \in [0, T], \quad (2.5)$$

where the f_k 's are d elements of a basis of $L^2([0, T]; \mathbb{R})$, the λ_k 's are appropriate scalars, and $(\widehat{\xi}_1, \dots, \widehat{\xi}_m)$ is a quantization of a Gaussian vector (ξ_1, \dots, ξ_m) . This means that $(\widehat{\xi}_1, \dots, \widehat{\xi}_m)$ takes its values in a finite set of M points that have been optimized in some way. This quantization is called a functional quantization and the feature of this approach is that the truncation number m can be rather small ($m \leq 10$ for example) to properly catch the global behavior of a trajectory.

Hence, quantization provides us with a map $\beta \mapsto \widehat{B}$ such that $\Theta(\widehat{B})$ is "close" to $\Psi(B)$ (so $\Theta^{\text{app}}(\widehat{B})$ is close to $\Psi^{\text{app}}(B)$). And $\mathbb{E}[\Upsilon \circ \Theta^{\text{app}}(\widehat{B})]$ can be computed since \widehat{B} takes its values in a finite set of continuous paths.

The functional quantization has been proposed and studied by G. Pagès *et al.* as a way to compute quickly an approximation of $\mathbb{E}[\Upsilon \circ \Psi(B)]$ with the expression $\mathbb{E}[\Upsilon \circ \Theta^{\text{app}}(\widehat{B})]$ (see [21, 25]). He also proposed to use functional quantization as a control variate by simulating B with a truncated Karhunen-Loève decomposition. In [5], quantization is also proposed as a way to perform variance reduction through stratification.

Here, our approach is a bit different since we are not bound to replace the Brownian motion by its truncated Karhunen-Loève decomposition. Using a least square estimators, we construct \widehat{B} from $(B_{t_1}, \dots, B_{t_d})$, where (t_1, \dots, t_d) are fixed times.

This is important, especially when diffusion processes are simulated, since one can choose the most convenient way to do so and only record the vector of marginals of the underlying Brownian motion.

Note that in our algorithm, the Brownian motion may be replaced by any other Gaussian process, such as Ornstein-Uhlenbeck processes or Brownian bridge.

2.1 The algorithm

2.1.1 The idea

To simplify the presentation, we assume that the dimension of the underlying Brownian motion is $n = 1$. To deal with the multi-dimensional case, it is sufficient to apply the algorithm to each component of the Brownian motion.

The idea behind the algorithm is the following, which we summarize in Figure 2.1 : we fix some integers d and $m \leq d$, some times $0 \leq t_1 < \dots < t_d \leq T$, as well as an orthonormal basis $\{f_i\}$ of $L^2([0, T]; \mathbb{R})$. We

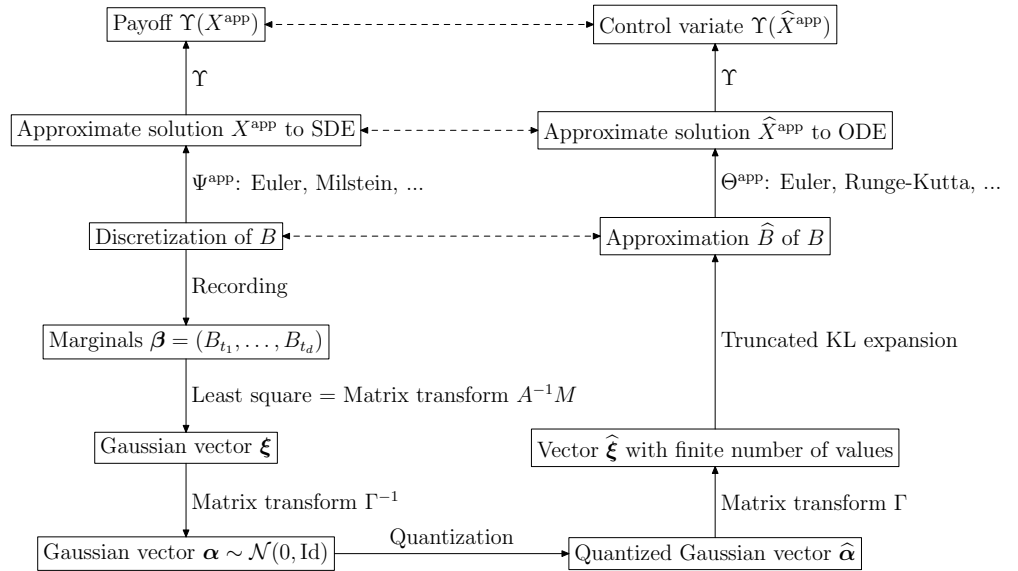


FIGURE 2.1 – The scheme

compute $\lambda_i = \text{Var} \left(\int_0^T f_i(s) B(s) ds \right)$. Given a realization $\boldsymbol{\beta} = (\beta_1, \dots, \beta_d)$ of $(B_{t_1}, \dots, B_{t_d})$, we define $\boldsymbol{\xi} = (\xi_1, \dots, \xi_m)$ as

$$\boldsymbol{\xi} = \underset{(\zeta_1, \dots, \zeta_m) \in \mathbb{R}^m}{\text{argmin}} \sum_{i=1}^d \left(\sum_{j=1}^m \zeta_j \sqrt{\lambda_j} f_j(t_i) - \beta_{t_i} \right)^2, \quad (2.6)$$

and choose a quantization $\widehat{\boldsymbol{\xi}} = (\xi_1, \dots, \xi_m)$ of this vector with at most Q points. Note that there are several ways to do so : One can compute the optimal quantization of the vector $\boldsymbol{\xi}$, or a product quantization (*i.e.* each component is quantized independently), or any mixing of these two approaches [27]. Besides, the quantization is chosen by optimizing the norm $\mathbb{E}[|\boldsymbol{\xi} - \widehat{\boldsymbol{\xi}}|^2]$ among all the possible random variables $\widehat{\boldsymbol{\xi}}$ with at most Q point. The optimal choice of $\widehat{\boldsymbol{\xi}}$ may indeed take Q' values with Q' smaller than Q .

We then use \widehat{B} defined by (2.5) as a control variate for B . This way, \widehat{B} takes its values in a set $\{\widehat{B}^{[i]}\}_{i=1, \dots, Q'}$ of Q' trajectories in $\mathcal{C}([0, T], \mathbb{R})$. Each trajectory $\widehat{B}^{[i]}$ is associated with a value of $\widehat{\boldsymbol{\xi}}$. Since all the possible values of $\boldsymbol{\xi}$ are known, $\Upsilon \circ \Theta^{\text{app}}(\widehat{B}^{[i]})$ can be computed prior to any simulation. For $p^{[i]} = \mathbb{P}[\widehat{\boldsymbol{\xi}} = (x_1^{[i]}, \dots, x_m^{[i]})]$ for the possible points $(x_1^{[i]}, \dots, x_m^{[i]})$ of the quantized vector, then

$$\mathbb{E}[\Upsilon \circ \Theta^{\text{app}}(\widehat{B})] = \sum_{i=1}^{Q'} p^{[i]} \Upsilon \circ \Theta^{\text{app}}(\widehat{B}^{[i]})$$

and this quantity may also be precomputed.

2.1.2 Preliminary computations

Let M be the $d \times m$ -matrix defined by $M_{i,j} = \sqrt{\lambda_j} f_j(t_i)$, $i = 1, \dots, d$, $j = 1, \dots, m$. We assume :

(H) The rank of M is m .

Solving the linear least squares is then equivalent to solving $M^T \boldsymbol{\xi} = \boldsymbol{\beta}$. As M is not necessarily a square matrix, one can indeed equivalently solve

$$A \boldsymbol{\xi} = M \boldsymbol{\beta} \text{ with } A = M M^T.$$

The matrix A is an invertible $m \times m$ -symmetric matrix and then $\boldsymbol{\xi} = A^{-1} M \boldsymbol{\beta}$.

Numerical remark 1. Numerically, one should be careful when inverting the matrix A to use a robust enough algorithm, such as the Cholesky decomposition or the QR method [13].

The vector of marginals $\boldsymbol{\beta} = (B_{t_1}, \dots, B_{t_n})$ is a Gaussian vector and it is easily seen that $\boldsymbol{\beta} = L \Delta \boldsymbol{\alpha}$ where $\boldsymbol{\alpha} \sim \mathcal{N}(0, \text{Id}_{d \times d})$, L is the lower triangular $d \times d$ -matrix defined by $L_{i,j} = 1$ if $i \leq j$ and $L_{i,j} = 0$ otherwise, and Δ is the diagonal matrix with $\sqrt{t_1}, \sqrt{t_2 - t_1}, \dots, \sqrt{t_d - t_{d-1}}$ as diagonal terms. Since, $\boldsymbol{\xi} = A^{-1} M \boldsymbol{\beta}$ and $\text{Cov}(\boldsymbol{\alpha}) = \text{Id}$,

$$\text{Cov}(\boldsymbol{\xi}) = (A^{-1} M L \Delta) (A^{-1} M L \Delta)^T.$$

In addition, there exists a unique, invertible (under Hypothesis (H)), lower triangular $d \times d$ -matrix Γ such that $\text{Cov}(\boldsymbol{\xi}) = \Gamma\Gamma^T$, and then $\boldsymbol{\alpha} = \Gamma^{-1}\boldsymbol{\xi} \sim \mathcal{N}(0, \text{Id}_{m \times m})$ is a m -vector of independent Gaussian random variables.

Numerical remark 2. From the numerical point of view, the matrix Γ will be computed with a Cholesky decomposition [13]*Theorem 5.2.3, p. 88.

The matrix Γ is important for practical purposes, since the quantization of $\boldsymbol{\xi}$ will be deduced from the quantization of $\boldsymbol{\alpha} \sim \mathcal{N}(0, \text{Id}_{m \times m})$, which depends only on our choice of m and Q .

2.1.3 The algorithm

The algorithm is then the following.

1. Choose a number d of marginals to use, a number $m \leq d$ of coefficients and the times $0 < t_1 < \dots < t_d \leq T$.
2. Choose an orthonormal basis $\{f_j\}_{j \geq 1}$ of $L^2([0, T]; \mathbb{R})$ and compute the corresponding weights $\{\lambda_j\}_{j \geq 1}$ in the Karhunen-Loève decomposition of the Brownian motion. Compute the matrices Γ , Γ^{-1} and $R = \Gamma^{-1}A^{-1}M$.
3. Choose a quantization of m -dimensional vectors $\boldsymbol{\alpha} \sim \mathcal{N}(0, \text{Id}_{m \times m})$, that is a discrete set $\{\widehat{\boldsymbol{\alpha}}^{[i]}\}_{i=1, \dots, Q'}$ called the code book and a “projection” π which associates to a realization of $\boldsymbol{\alpha}$ an element of the code book. Compute for $i = 1, \dots, Q'$, $p^{[i]} = \mathbb{P}[\pi(\boldsymbol{\alpha}) = \widehat{\boldsymbol{\alpha}}^{[i]}]$, $\phi^{[i]} = \Upsilon \circ \Theta^{\text{app}}(\widehat{B}^{[i]})$ where

$$\widehat{B}_t^{[i]} = \sum_{j=1}^m \widehat{\xi}_j^{[i]} \sqrt{\lambda_j} f_j(t) \text{ with } \widehat{\xi}^{[i]} = \Gamma \widehat{\boldsymbol{\alpha}}^{(i)},$$

$$\text{and } \widehat{J} = \sum_{i=1}^{Q'} \phi^{[i]} p^{[i]}.$$

4. Set $J \leftarrow 0$.
5. For i from 1 to N do
 - (a) Simulate a realization $B(\omega)$ of the Brownian motion B , record $\boldsymbol{\beta}(\omega) = (B_{t_1}(\omega), \dots, B_{t_d}(\omega))$ and compute $\phi = \Upsilon \circ \Psi^{\text{app}}(B(\omega))$.
 - (b) Compute $\boldsymbol{\alpha}(\omega) = R\boldsymbol{\beta}(\omega)$ which is a realization of $\boldsymbol{\xi} \sim \mathcal{N}(0, \text{Id}_{m \times m})$.
 - (c) Find the index i such that $\widehat{\boldsymbol{\alpha}}^{[i]} = \pi(\boldsymbol{\alpha})$.
 - (d) Set $J \leftarrow J + \phi - \phi^{[i]}$.
6. Return $N^{-1}J + \widehat{J}$, which is an estimator of $\mathbb{E}[\Upsilon \circ \Psi(B)]$.

Numerical remark 3. As explained in the Numerical remarks 1 and 2, we use in order to compute A^{-1} , Γ and Γ^{-1} a Cholesky decomposition. The feature of our algorithm is that it uses matrices with reasonably small size (of order 10×10).

Remark 1. For the Brownian motion, there is a natural choice of a basis $\{f_j\}_{j \geq 1}$, which is the set of eigenfunctions of the covariance operator $f \in \mathcal{C}([0, T]; \mathbb{R}) \mapsto$

$(t \mapsto \int_0^{+\infty} (s \wedge t) f(s) ds)$ of the Brownian motion. In this case,

$$\lambda_j = \left(\frac{T}{\pi(j - \frac{1}{2})} \right)^2 \quad \text{and} \quad f_j(t) = \sqrt{\frac{2}{T}} \sin \left(\pi \left(j - \frac{1}{2} \right) \frac{t}{T} \right) \quad (2.7)$$

for $j \geq 1$. Let us note that the algorithm may be applied to any Gaussian process. However, it is not always so easy to get the eigenfunctions of the corresponding covariance operator (see [29] for example).

One may also use another natural basis, the Schauder basis, although this one is not orthonormal. Numerical experiments on the Brownian motion give results that are not as good as the ones obtained with the basis above.

Remark 2. In this algorithm, we have chosen to use the quantization of Gaussian random vectors with covariance matrix equal to identity. With slight changes in the algorithm, we can either quantize the vector $\Delta^{-1}L^{-1}\beta \sim \mathcal{N}(0, \text{Id}_{d \times d})$ of size $d \geq m$ or the vector $\xi \sim \mathcal{N}(0, (A^{-1}ML\Delta)(A^{-1}ML\Delta)^T)$, which have covariances matrices different from identity. Our choice is justified by the numerical cost of the quantization procedure and the need for vectors to have the smallest size possible. Quantizations of random m -vectors with distribution $\mathcal{N}(0, \text{Id}_{m \times m})$ may be prepared prior to any simulation [27] and is then suitable for any SDE. In addition, tabulated values can be found in the WEB site [24], thanks to the works of G. Pagès and J. Printems.

2.2 Numerical tests

We have performed numerical tests with 3 classical models in finance : the Black & Scholes model

$$X_t = X_0 + \int_0^t \sigma X_s dB_s + \int_0^t \nu X_s ds = X_0 \exp \left(\sigma B_t + \left(\nu \frac{1}{2} \sigma^2 \right) t \right),$$

the Cox-Ingersoll-Ross (CIR) model

$$X_t = X_0 + \int_0^t k(a - X_s) ds + \sigma \int_0^t \sqrt{X_s} dB_s$$

and the Heston model

$$\begin{cases} V_t = V_0 + \int_0^t k(a - V_s) ds + \sigma \int_0^t \sqrt{V_s} dB_s^1, \\ X_t = X_0 + \int_0^t \nu X_s ds + \int_0^t \sqrt{V_s} X_s dB_s^2, \\ \langle B^1, B^2 \rangle_t = \rho t, \end{cases}$$

where B^1 and B^2 are two Brownian motions. The marginals $(B_{t_1}, \dots, B_{t_d})$ are taken at times $t_i = iT/10$, $i = 1, \dots, 10$.

To simulate the Black & Scholes model, we used the exponential representation, so that $X = \Psi(B)$ where $\Psi(B) = (x \exp(\sigma B_t + \mu t - \sigma^2 t/2))_{t \geq 0}$. We also use $\Theta = \Psi$.

To simulate the CIR and the Heston models, we used the modification of the Euler scheme proposed by Berkaoui, Bossy and Diop [4]. Any of the schemes studied in [20] could have been used, as long as one record the values of $(B_{t_1}, \dots, B_{t_d})$ while performing the simulations (of course, one may use much more marginals of B).

With these schemes, we produce N independent realizations $X^{(1)}, \dots, X^{(N)}$ (of approximations) of the path X . In all the cases, the Brownian motion is discretized using a time step $\delta t = 0.01$. From the realizations $(B_{t_1}^{(i)}, \dots, B_{t_d}^{(i)})$, $i = 1, \dots, N$, of the marginals, we obtain through our quantization procedure quantized paths $\widehat{B}^{(i)}$, $i = 1, \dots, N$ that are taken in the finite set $\{\widehat{B}^{[j]}\}_{j=1, \dots, Q'}$. We denote by $q(i)$ the integer such that $\widehat{B}^{(i)} = \widehat{B}^{[q(i)]}$. To compute $X^{[j]} = \Theta^{\text{app}}(\widehat{B}^{[j]})$ along the quantized paths, we use the Euler scheme for ODE after having transformed the SDEs into Stratonovich SDEs, and then replace the Brownian motion by $\widehat{B}^{[j]}$. For the sake of simplicity, we have also used an explicit Euler scheme with the same timestep as the Euler scheme for SDEs. Let us note that in our case, although a wide range of techniques are available (Runge-Kutta, implicit schemes, ...), we prefer keep a simple scheme. Indeed, the precision for the resolution of the ODE is not a real issue. A bias induced by the scheme has a very little impact on the Monte Carlo estimator : whatever the scheme, the control variate has a zero mean.

We compare three estimators v , v' and v'_κ of the empirical variance of $\Upsilon(X_T)$ associated to the three empirical means

$$\begin{aligned} \mu &= \frac{1}{N} \sum_{i=1}^N \Upsilon(X^{(i)}), \\ \mu' &= \frac{1}{N} \sum_{i=1}^N (\Upsilon(X^{(i)}) - (\Upsilon(\widehat{X}^{[q(i)]}) - \mathbb{E}[\Upsilon(\widehat{X})])), \\ \text{and } \mu'_\kappa &= \frac{1}{N} \sum_{i=1}^N (\Upsilon(X^{(i)}) - \kappa^{(i)}(\Upsilon(\widehat{X}^{[q(i)]}) - \mathbb{E}[\Upsilon(\widehat{X})])), \end{aligned}$$

where $\kappa^{(i)}$ is constructed from $X^{(1)}, \dots, X^{(i)}$ to approximate κ_{opt} in (2.4).

Remark 3. Unless one knows how to simulate exactly X from B , these estimators are biased. Yet the bias is only induced by the choice of the scheme used to compute X . The control variate does not induce any supplementary bias. We no longer discuss this issue which is out of the scope of this article.

The results are summarized in Tables 2.2, 2.3 and 2.4. The mean of the estimator is denoted by μ (in all the cases, the relative error between the different empirical estimator is very small), which is then the price of the option.

The efficiency of a variance reduction technique shall take the computation time into account. In Tables 2.2, 2.3 and 2.4, T_s is the time spent to compute the empirical mean μ without the control variate, while T'_s is the time spent to compute the empirical mean μ with the control variate (time θ_{simul} per simulation). With our control variate, the extra computational time is due to (a)

the initialization of the algorithm (time T_{init}), which depends on the number of points of quantization, and (b) the research of the nearest neighbor of the Gaussian coefficients among the quantized points (time θ_{search} per simulation). To research the vector's nearest neighbor, we have used a method called *kd-tree* [3] which appears to be efficient, while for a scalar, we used a search by dichotomy.

The method has been applied to several call options, for three values of K : $K = 0.5$ (In the money, ITM), $K = 1.0$ (At the money, ATM) and $K = 1.3$ (Out of the money, OTM). The options are

$$\text{European : } \Upsilon(X) = (X_T - K)_+,$$

$$\text{Asian : } \Upsilon(X) = \left(\frac{1}{T} \int_0^T X_s ds - K \right)_+,$$

$$\text{Asian binary : } \Upsilon(X) = \mathbf{1}_{\frac{1}{T} \int_0^T X_s ds > K},$$

$$\text{Ratchet : } F_0 = 0, F_i = (K + F_{i-1} + \rho_i)_+ \text{ and } \rho_i = \frac{X_{t_i} - X_{t_{i-1}}}{(t_i - t_{i-1})X_{t_i}},$$

$$\text{Lookback fixed : } \Upsilon(X) = X_T - \min_{t \in [0, T]} X_t,$$

$$\text{Lookback floating : } \Upsilon(X) = \left(\max_{t \in [0, T]} X_t - K \right)_+,$$

$$\text{Cliquet : } \max \left\{ F_0, \min \left\{ C_0, \sum_{i=1}^N \max\{F_i, \min\{C_i, R_i\}\} \right\} \right\}$$

where the floors $F_i = 0$ and the caps $C_i = K$, and $R_i = X_{t_i} - X_{t_{i-1}}$. For the cliquet and ratchet options, we use $t_i = iT/12$.

Thus, with N simulations and Q quantized path, the ratio of the times T'_s/T_s is given by

$$\frac{T'_s}{T_s} = \frac{T_{\text{init}}(Q) + N\theta_{\text{search}} + N\theta_{\text{simul}}}{N\theta_{\text{simul}}}.$$

Let us note that the initialization phase implies the resolution of Q ordinary differential equations, so that $T_{\text{init}} \approx Q\theta_{\text{simul}}$.

Thus, in Tables 2.2, 2.3 and 2.4, we give the efficiency ratio for 100 000 and 1 000 000 simulations. We call the efficiency ratio the quantities

$$e = \frac{T_s}{T'_s} \times \frac{v}{v'} \text{ and } e_\kappa = \frac{T_s}{T'_s} \times \frac{v}{v'_\kappa}.$$

When these quantities are close to 1, then the method provides no gain. It is then simpler to increase the size of the sample without any correction to obtain the same result. When these quantities are greater than 1, then for a given precision, it corresponds to the gain factor in execution time. If these quantities are smaller than 1, then the methods shall not be used.

In Table 2.1, we also give the prices obtained by computing the options prices by computing $\sum_{j=1}^{Q'} p^{[j]} \Upsilon(X^{[j]})$, that is the quantized price. The result of this table should be compared with the values given in the column μ (the values of

the Monte Carlo estimator) of Tables 2.2, 2.3 and 2.4. It is not surprising to see that the quantized prices can sometimes be far from the estimated prices for the Ratchet, Lookback and Cliquet options. Although the quantized price is computed quickly, this estimation is not always reliable.

Option	B&S	CIR	Heston
European OTM	0,015	0,024	0,063
European ATM	0,106	0,195	0,140
European ITM	0,549	0,697	0,462
Asian OTM	0	$1,0 \cdot 10^{-4}$	0,017
Asian ATM	0,053	0,099	0,081
Asian ITM	0,513	0,590	0,466
Asian Binary OTM	0	0,029	0,017
Asian Binary ATM	0,520	0,809	0,081
Asian Binary ITM	1	1	0,467
Ratchet OTM	72,7	81,16	63,8
Ratchet ATM	56,2	64,67	48,7
Ratchet ITM	29,3	37,35	26,7
Lookback fixed OTM	0,018	0,029	0,090
Lookback fixed ATM	0,143	0,228	0,222
Lookback fixed ITM	0,643	0,728	0,723
Lookback floating ITM	0,135	0,218	0,188
Cliquet OTM	0,165	0,236	0,265
Cliquet ATM	0,165	0,235	0,223
Cliquet ITM	0,084	0,097	0,008

TABLE 2.1 – Table of the prices computed using the same quantization of the Brownian motion as the one of Table 2.2, 2.3 and 2.4.

For most of the simulations, the variance reduction technique is efficient or very efficient : The efficiency ratios above 5 have been written in bold. For very small values of prices, the methods may be inefficient (the figures are then emphasized). In this case, using the estimator μ'_κ improves the quality of the estimator, but does not provide significant gains with respect to μ . Most of the time, using μ'_κ instead of μ' does not significantly improve the results. In addition, the method may also be inefficient when the payoff is discontinuous, as one may expect.

2.3 Conclusion

We have presented a new variance reduction technique which belongs to the class of hybrid quantization-Monte Carlo method related to the simulation of SDE.

This method relies on functional quantization and may be combined with other variance reduction techniques applied to the Brownian motion.

Numerical tests show the robustness and the efficiency of the method which has good gain factors — up to order 10 — even for complex payoffs. However, this method is not suitable for SDEs with a large number of underlying Brownian motions — as the number of quantized paths grows very quickly — and to estimate small quantities or prices.

As the end-user is free to fix a lot of parameters (number of quantization points, way to decompose the vectors, ...), the numerical results exposed here may be improved by tuning the parameters to get the best balance between a small ratio of variances and the extra computational cost. Note that however, this tuning should be model dependent.

2.4 Approximation of the solution of the SDE using approximation of the Brownian motion

2.4.1 Ordinary differential equation : the map Θ

For a coefficient σ of class \mathcal{C}^2 and a coefficient c of class \mathcal{C}^1 with at most a linear growth, we have introduced the map Θ which transforms a path $U \in \mathcal{C}_p^1([0, T]; \mathbb{R}^n)$ into the unique solution Y to (2.3).

Let \dot{U} be the derivative of U with respect to the time and set $\|U\|_{1\text{-var}} = \|\dot{U}\|_\infty$. From the Gronwall Lemma, it is standard that $\|\Theta(U)\|_\infty \leq K(1 + \|U\|_{1\text{-var}})$, where K depends on the Lipschitz constant of σ and b , as well as the starting point x and the horizon T .

Then for two paths U and V , again from the Gronwall lemma,

$$\|\Theta(U) - \Theta(V)\|_\infty \leq K'(1 + \|U\|_\infty)\|U - V\|_{1\text{-var}},$$

where K depends on the Lipschitz constant of σ and b , as well as the starting point x and the horizon T .

Note that if Θ^{aPP} is defined using the Euler scheme for ODE, then a similar computation with the discrete Gronwall inequality leads to

$$\|\Theta^{\text{aPP}}(U) - \Theta^{\text{aPP}}(V)\|_\infty \leq K''(1 + \|U\|_\infty)\|U - V\|_{1\text{-var}},$$

for some constant K'' that depends on the Lipschitz constant of σ and b , as well as the starting point x and the horizon T .

If $n = 1$, then we have a better estimate : it follows from the approach proposed by H. Doss [8] and H. Sussmann [30] (See also [15] for example) that Θ is locally Lipschitz continuous with respect to the uniform norm when σ has bounded first and second derivatives.

For an orthonormal basis $\{f_i\}_{i=1,2,\dots}$ of $L^2([0, T]; \mathbb{R})$, we consider the subset \mathfrak{K} of $\mathcal{C}_p^1([0, T]; \mathbb{R}^n)$ of paths of type

$$X_t^j = \sum_{i=1}^m \xi_i^j f_i(t), \text{ for } j = 1, \dots, n,$$

for some integer $m > 0$ and a family $\{\xi_i^j\}_{i=1,\dots,m,j=1,\dots,n}$ of scalars.

If the f_i belong to $\mathcal{C}_p^1([0, T]; \mathbb{R})$, for $U = \{\sum_{i=1}^m \xi_i^j f_i(t)\}_{j=1}^m$ and $V = \{\sum_{i=1}^m \zeta_i^j f_i(t)\}_{j=1}^m$ in \mathfrak{R} , then

$$\|\Theta(U) - \Theta(V)\|_\infty \leq C \|\xi - \zeta\|, \quad (2.8)$$

with C a constant which depends only on m, n, T and the coefficients σ and b , where $\xi = \{\xi_i^j\}_{i=1,\dots,m,j=1,\dots,n}$, $\zeta = \{\zeta_i^j\}_{i=1,\dots,m,j=1,\dots,n}$, and $\|\cdot\|$ is a matrix norm.

Again, if one uses the Euler scheme for ODE to construct Θ^{app} , then

$$\|\Theta^{\text{app}}(U) - \Theta^{\text{app}}(V)\|_\infty \leq C' \|\xi - \zeta\|,$$

for some constant C' which depends only on m, n, T and the coefficients σ and b .

2.4.2 Is our control variate a good approximation of the diffusion ?

As explained in the introduction, a good control variate for $\Upsilon(X)$ is given by a random variable $\Upsilon(Y)$ such that $\mathbb{E}[\Upsilon(Y)]$ may be computed and Y is strongly correlated and close to X .

Assuming that Υ is continuous, one may wonder if the process \widehat{X} constructed with our scheme is close to X , the solution to the SDE consider.

Let us denote by ξ^{KL} the first m coefficients of the truncated Karhunen-Loève decomposition B^m of the Brownian motion B :

$$B_t^m = \sum_{i=1}^m \xi_i^{KL} \sqrt{\lambda_i} f_i(t).$$

Let us also consider

$$\xi(d) = \operatorname{argmin}_{\zeta(d) \in \mathbb{R}^m} \sum_{i=1}^d \left(B_{t_i} - \sum_{j=1}^m \zeta_j(d) \sqrt{\lambda_j} f_j(t_i) \right)^2. \quad (2.9)$$

We then set

$$\widetilde{B}_t^m = \sum_{i=1}^m \xi_i(d) \sqrt{\lambda_i} f_i(t).$$

Using the above notation, the Gaussian vector $\xi(d)$ may be expressed as $\Gamma(d)\alpha(d)$ for some invertible matrix $\Gamma(d)$ such that $\Gamma(d)\Gamma(d)^T = \operatorname{Cov}(\xi(d))$, so that $\alpha(d) \sim \mathcal{N}(0, \operatorname{Id})$. The approximation \widehat{B}^m of the Brownian motion is

$$\widehat{B}_t^m = \sum_{i=1}^m \sqrt{\lambda_i} \Gamma^{-1}(d) \widehat{\alpha}_i f_i(t), \quad (2.10)$$

where $\widehat{\alpha}(d)$ is the quantization of $\alpha(d)$.

Obviously, the distance between \widehat{X} and X is given by $\|\Psi(X) - \Theta(\widehat{X})\|_\infty$ and may be controlled by the sum of three terms :

- The distance between $\Psi(B)$ and $\Theta(B^m)$, where B^m is the truncated Karhunen-Loève decomposition of the Brownian motion.
- The distance between $\Theta(B^m)$ and $\Theta(\tilde{B}^m)$.
- The distance between $\Theta(\tilde{B}^m)$ and $\Theta(\hat{B}^m)$.

Obviously with (2.8), the distances between $\Theta(B^m)$ and $\Theta(\tilde{B}^m)$, and between $\Theta(\tilde{B}^m)$ and $\Theta(\hat{B}^m)$ depend only on the distance between the coefficients in the Karhunen-Loève type decompositions.

2.4.3 Convergence of the solution to the least squares problem to the coefficients of the Karhunen-Loève decomposition

The Gaussian coefficients of the Karhunen-Loève decomposition are unique provided that they are independent.

Let us denote by $\Gamma(d)$ a $d \times d$ -matrix such that $\text{Cov}(\boldsymbol{\xi}(d)) = \Gamma(d)\Gamma(d)^T$, where $\boldsymbol{\xi}(d)$ is obtained by solving the least squares problem (2.9). Thus, this matrix depends on the number of coefficients m as well as the times (t_1, \dots, t_d) of the marginals of the Brownian motion.

We now consider that m is fixed and that d increases to $+\infty$.

Lemma 1. *If $t_i^d = iT/d$ for $i = 1, \dots, d$, then the $d \times d$ -matrix $\Gamma(d)$ with $\Gamma(d)\Gamma(d)^T = \text{Cov}(\boldsymbol{\xi}(d))$ converges to the identity as $d \rightarrow \infty$, $\boldsymbol{\xi}(d)$ converges almost surely to $\boldsymbol{\xi}^{KL}(d)$ and $\tilde{B}_t^m(d)$ converges almost surely uniformly to the Karhunen-Loève decomposition B^m of B truncated at order the m .*

Proof. Let $M(d)$ be the matrix defined by $M_{i,j}(d) = \sqrt{\lambda_j} f_j(t_i^d) B_{t_i^d}$ for $t_i^d = iT/d$, $i = 1, \dots, d$, $j = 1, \dots, m$, as in Section 2.1.2. Set $\boldsymbol{\beta}(d) = (B_{t_1^d}, \dots, B_{t_d^d})$. Thus as $d \rightarrow \infty$,

$$\frac{T}{d} M(d) \boldsymbol{\beta}(d) = \begin{bmatrix} \frac{T}{d} \sum_{i=1}^d \sqrt{\lambda_1} f_1(t_i^d) B_{t_i^d} \\ \vdots \\ \frac{T}{d} \sum_{i=1}^d \sqrt{\lambda_m} f_m(t_i^d) B_{t_i^d} \end{bmatrix} \xrightarrow{d \rightarrow \infty} \begin{bmatrix} \sqrt{\lambda_1} \int_0^T f_1(s) B(s) ds \\ \vdots \\ \sqrt{\lambda_m} \int_0^T f_m(s) B(s) ds \end{bmatrix}.$$

Let $\tilde{B}_t^m := \sum_{j=1}^m \xi_j \sqrt{\lambda_j} f_j(t)$, $t \in [0, T]$, be the truncated Karhunen-Loève decomposition of B . Since $\{f_i\}_{i \geq 0}$ is an orthonormal basis of $L^2([0, T])$, one gets easily that $\lambda_i \xi_i = \sqrt{\lambda_i} \int_0^T f_i(s) B(s) ds$ for $i = 1, \dots, m$, and then that $Td^{-1} M(d) \boldsymbol{\beta}(d)$ converges to the vector $D \boldsymbol{\xi}^{KL}$ with $\boldsymbol{\xi}^{KL} = (\xi_1^{KL}, \dots, \xi_m^{KL})$, where D is the diagonal matrix with $(\lambda_1, \dots, \lambda_m)$ as diagonal elements. On the other hand, one easily gets from the same argument that

$$\frac{T}{d} M(d) M(d)^T \xrightarrow{d \rightarrow \infty} \left(\sqrt{\lambda_i \lambda_j} \int_0^T f_i(s) f_j(s) ds \right)_{i,j=1, \dots, d} = D.$$

Since the λ_i are positive, $Q(d) = Td^{-1}M(d)M(d)^T$ is invertible if d is large enough and $Q(d)^{-1}$ converges to D^{-1} . It follows that

$$\boldsymbol{\xi}(d) = Q(d)^{-1}Td^{-1}M(d)\boldsymbol{\beta}(d) \xrightarrow{d \rightarrow \infty} D^{-1}D\boldsymbol{\xi}^{KL} = \boldsymbol{\xi}^{KL}.$$

The uniform convergence of \tilde{B}^m to B^m follows from the convergence of $\boldsymbol{\xi}(d)$ to the coefficients $\boldsymbol{\xi}^{KL}$ of the Karhunen-Loève decomposition of B , and the convergence of $\Gamma(d)$ to Id follows from the fact that $\text{Cov}(\boldsymbol{\xi}(d))$ converges to $\text{Cov}(\boldsymbol{\beta}) = \text{Id}$ and then $\Gamma(d)$ also converges to Id (see [13]*Theorem 5.2.3 for example, the convergence follows from the explicit construction of the Cholesky decomposition). \square

Remark 4. From a practical point of view, numerical experiments show that the covariance matrix $\text{Cov}(\boldsymbol{\xi}(d))$ is already close to Id even for small values of d when the basis given in Remark 1 is used. Thus, one can take for d values of order 10 when $m \leq 10$ to get from the least squares approach a good approximation of the truncated Karhunen-Loève decomposition of B .

2.4.4 Convergence of the quantized coefficients

With (2.8) and (2.10),

$$\|\Theta(\hat{B}^m) - \Theta(\tilde{B}^m)\|_\infty \leq C\|\boldsymbol{\alpha}(d) - \hat{\boldsymbol{\alpha}}(d)\|,$$

where C depends on the coefficients of Γ , m , d and the f_i .

The distance between $\Theta(\hat{B}^m)$ and $\Theta(\tilde{B}^m)$ therefore depends on the distance between $\hat{\boldsymbol{\alpha}}(d)$ and $\boldsymbol{\alpha}(d)$. There are various ways to choose an optimal quantized (block quantized, product quantizer, ...). For an optimal quantizer with at most Q points (which means that the whole vector $\boldsymbol{\alpha}(d)$ is quantized), the Zador theorem [14] implies that the $L^2(\mathbb{P})$ -distance between $\boldsymbol{\alpha}(d)$ and $\hat{\boldsymbol{\alpha}}(d)$ is equivalent for large Q to $C(d)/Q^{1/d}$.

Remark 5. In [21, 25, 26], G. Pagès *et al.* have studied the rate of convergence of the approximation of the Brownian motion by a truncation of the Karhunen-Loève decomposition. In their approach, they choose carefully the truncation order m as a function of Q to get the optimal rate. When constructing a control variate, we are not necessarily looking for the best approximation. In addition, it is good for efficiency to keep m fixed and rather small (in all our simulations, $m \leq 10$). Besides, we do not quantize the true coefficient of the Karhunen-Loève decomposition but only an approximation of them.

2.4.5 Convergence of the SDEs

A good control variate shall stay close to the computed quantities. Our algorithm provides for a realization of the Brownian motion B a path \hat{B}^m . If B^m is the truncated Karhunen-Loève decomposition of B , we have seen that when the number of marginals d we use and the number of points of quantization Q tend to infinity, then $\Theta(\hat{B}^m)$ converges to $\Theta(B^m)$.

Yet let us note that in practice, d and Q shall remain rather small in order to avoid a high computational cost which annihilate the whole interest of the method.

However, one may wonder how close $\Theta(B^m)$ is to $\Psi(B)$? Using the results of [6, 10, 25] based on the theory of rough paths [22, 23, 17, 16, 10, 11], $\Theta(B^m)$ converges to $\Psi(B)$ when m tends to infinity. However, the rate is not known. As said above in Remark 5, we do not aim at letting m tend to infinity. However, this result justifies the effectiveness of our approach.

Let us note however that this convergence holds due to the convergence of B^m to B in the γ -Hölder norm, $\gamma < 1/2$. The uniform convergence is not sufficient (See for example [17, 19] for explicit counter-examples). Besides, it shall be assumed that the second derivative of σ and the first derivative of c are ϵ -Hölder continuous for some $\epsilon > 0$. The distance between $\Theta(B^m)$ and $\Psi(B)$ depends on the γ -Hölder norm of $B - B^m$, but also on the γ -Hölder norms of B and B^m (see [17, 18]). For the CIR model, this result could only be invoked when the solution stay away from 0, because of the degeneracy of the diffusion coefficient at 0. The distance also depends on the minimum of the paths $\Theta(B^m)$ and $\Psi(B)$.

If the dimension of the space is equal to 1, then one may use the Doss-Sussman theory [8, 30] which implies that $\Theta(B^m)$ converges to $\Psi(B)$, since B^m converges uniformly to B . In addition, $\|\Psi(B) - \Theta(B^m)\|_\infty \leq K\|B - B^m\|_\infty$ for a constant K which depends on the maximal and minimal values B_T and B_T^m . The $L^2(\mathbb{P})$ -norm of the uniform distance between B and B^m is of order $\sqrt{\log m/m}$ (See [26]).

However, the gain in terms of variance reduction seems here pretty difficult to evaluate here. One does not know in general the variance of $\text{Var}(\Upsilon(\Psi(B)))$ and an evaluation of the distance between $\Psi(B)$ and $\Theta(B^m)$, which is out of the scope of this article.

In addition, several simulations on the same class of model, *e.g.*, Black & Scholes, have shown no clear correlation between the gain in variance and the values of the parameters. This means that the gain seems difficult to predict because the map Ψ is highly non-linear.

Acknowledgement. The authors are indebted to Aymen Bergaoui, Madalina Deaconu, Zied Ghazai, Samuel Herrmann, Christophe Michel, Gilles Pagès, Yi-qing Wang for interesting discussions about this method or problems related to quantization.

Option	v'/v	v'_k/v	$\frac{T'_{s,N}}{T_s}$	$\frac{vT'_{N,s}}{v'T_{N,s}}$	$\frac{vT'_s}{v'_k T'_{N,s}}$	$\frac{T'_{M,s}}{T_s}$	$\frac{vT'_s}{v'T_{M,s}}$	$\frac{vT'_s}{v'_k T'_{s,s}}$	μ	σ	σ'	σ'_k
European OTM	0,52	0,5	1,2	1,5	1,6	1,2	1,5	1,6	0,017	$4,2 \cdot 10^{-03}$	$2,2 \cdot 10^{-03}$	$2,1 \cdot 10^{-03}$
European ATM	0,29	0,28	1,2	2,8	2,9	1,2	2,8	2,9	0,11	0,024	$6,8 \cdot 10^{-03}$	$6,6 \cdot 10^{-03}$
European ITM	0,23	0,22	1,2	3,5	3,6	1,2	3,5	3,6	0,55	0,045	0,01	$9,9 \cdot 10^{-03}$
Asian OTM	1	*	1,2	0,8	*	1,2	0,8	*	$9,3 \cdot 10^{-04}$	$1,0 \cdot 10^{-04}$	$1,0 \cdot 10^{-04}$	*
Asian ATM	0,08	0,08	1,2	10	10	1,2	10	10	0,055	$6,4 \cdot 10^{-03}$	$5,1 \cdot 10^{-04}$	$5,1 \cdot 10^{-04}$
Asian ITM	0,056	0,056	1,2	14	14	1,2	14	14	0,52	0,014	$7,8 \cdot 10^{-04}$	$7,9 \cdot 10^{-04}$
Asian Binary OTM	1	*	1,2	0,8	*	1,2	0,8	*	0,016	0,016	0,016	*
Asian Binary ATM	0,31	0,28	1,2	2,6	2,8	1,2	2,6	2,8	0,52	0,25	0,077	0,071
Asian Binary ITM	*	*	1,2	*	*	1,2	*	*	*	*	*	*
Ratchet OTM	0,069	0,069	1,2	12	12	1,2	12	12	72	$1,8 \cdot 10^{+02}$	12	12
Ratchet ATM	0,07	0,07	1,2	11	11	1,2	11	11	56	$1,7 \cdot 10^{+02}$	12	12
Ratchet ITM	0,084	0,081	1,2	9,6	9,9	1,2	9,6	9,9	30	$1,4 \cdot 10^{+02}$	12	11
Lookback fixed OTM	0,34	0,33	1,2	2,4	2,4	1,2	2,4	2,4	0,029	$6,6 \cdot 10^{-03}$	$2,3 \cdot 10^{-03}$	$2,2 \cdot 10^{-03}$
Lookback fixed ATM	0,16	0,16	1,2	5	5	1,2	5	5	0,19	0,026	$4,2 \cdot 10^{-03}$	$4,2 \cdot 10^{-03}$
Lookback fixed ITM	0,16	0,16	1,2	5	5	1,2	5	5	0,69	0,026	$4,2 \cdot 10^{-03}$	$4,2 \cdot 10^{-03}$
Lookback floating ITM	0,42	0,4	1,2	1,9	2	1,2	1,9	2	0,17	0,023	$9,8 \cdot 10^{-03}$	$9,3 \cdot 10^{-03}$
Cliquet OTM	0,24	0,24	1,2	3,3	3,3	1,2	3,3	3,3	0,26	0,019	$4,6 \cdot 10^{-03}$	$4,6 \cdot 10^{-03}$
Cliquet ATM	0,27	0,26	1,2	3	3,1	1,2	3	3,1	0,25	0,015	$4,1 \cdot 10^{-03}$	$4,0 \cdot 10^{-03}$
Cliquet ITM	3,3	0,72	1,2	0,24	1,1	1,2	0,24	1,1	0,097	$1,6 \cdot 10^{-04}$	$5,2 \cdot 10^{-04}$	$1,1 \cdot 10^{-04}$

TABLE 2.2 – Results for the Black & Scholes model with $X_0 = 1$, $\sigma = 0.2$, $\nu = 0.05$, $T = 1$. The first coefficient of the KL-decomposition is decomposed with 7 points and the 5 next are decomposed with 20 points (140 quantized paths). For the simulations, $N = 100\,000$ and $M = 1\,000\,000$. The * indicates irrelevant results (the quantized price is equal to 0 and the algorithm for computing κ does not converge).

Option	v'/v	v'_k/v	$\frac{T'_{s,N}}{T_s}$	$\frac{vT'_s}{v'T_{N,s}}$	$\frac{vT'_s}{v'T_{N,s}}$	$\frac{vT'_s}{v'T_{N,s}}$	$\frac{T'_{M,s}}{T_s}$	$\frac{vT'_s}{v'T_{M,s}}$	$\frac{vT'_s}{v'T_{k,s}}$	μ	σ	σ'	σ'_k
European OTM	0,5	0,48	1,2	1,6	1,7	1,2	1,6	1,7	1,7	0,029	$4,7 \cdot 10^{-03}$	$2,4 \cdot 10^{-03}$	$2,3 \cdot 10^{-03}$
European ATM	0,33	0,32	1,2	2,4	2,6	1,2	2,4	2,6	2,6	0,21	0,024	$8,0 \cdot 10^{-03}$	$7,6 \cdot 10^{-03}$
European ITM	0,32	0,31	1,2	2,5	2,6	1,2	2,5	2,6	2,6	0,7	0,028	$9,1 \cdot 10^{-03}$	$8,7 \cdot 10^{-03}$
Asian OTM	0,94	0,88	1,2	0,87	0,93	1,2	0,87	0,93	0,93	$9,3 \cdot 10^{-04}$	$6,9 \cdot 10^{-05}$	$6,5 \cdot 10^{-05}$	$6,1 \cdot 10^{-05}$
Asian ATM	0,05	0,05	1,2	1,6	1,6	1,2	1,6	1,6	1,6	0,1	$7,4 \cdot 10^{-04}$	$3,7 \cdot 10^{-04}$	$3,7 \cdot 10^{-04}$
Asian ITM	0,05	0,05	1,2	1,6	1,6	1,2	1,6	1,6	1,6	0,6	$9,5 \cdot 10^{-03}$	$4,7 \cdot 10^{-04}$	$4,7 \cdot 10^{-04}$
Asian Binary OTM	1,1	0,74	1,2	0,75	1,1	1,2	0,75	1,1	1,1	0,022	0,022	0,024	0,016
Asian Binary ATM	0,21	0,19	1,2	3,8	4,3	1,2	3,8	4,3	4,3	0,84	0,14	0,029	0,026
Asian Binary ITM	*	*	1,2	*	*	1,2	*	*	*	1	0	*	*
Ratchet OTM	0,068	0,068	1,2	1,2	1,2	1,2	1,2	1,2	1,2	81	$1,1 \cdot 10^{+02}$	7,4	7,4
Ratchet ATM	0,068	0,068	1,2	1,2	1,2	1,2	1,2	1,2	1,2	65	$1,1 \cdot 10^{+02}$	7,3	7,3
Ratchet ITM	0,078	0,077	1,2	1,0	1,1	1,2	1,0	1,1	1,1	38	97	7,6	7,4
Lookback fixed OTM	0,31	0,31	1,2	2,6	2,7	1,2	2,6	2,7	2,7	0,049	$7,2 \cdot 10^{-03}$	$2,2 \cdot 10^{-03}$	$2,2 \cdot 10^{-03}$
Lookback fixed ATM	0,19	0,18	1,2	4,4	4,5	1,2	4,4	4,5	4,5	0,28	0,021	$3,8 \cdot 10^{-03}$	$3,8 \cdot 10^{-03}$
Lookback fixed ITM	0,19	0,18	1,2	4,4	4,5	1,2	4,4	4,5	4,5	0,78	0,021	$3,8 \cdot 10^{-03}$	$3,8 \cdot 10^{-03}$
Lookback floating ITM	0,47	0,44	1,2	1,7	1,9	1,2	1,7	1,9	1,9	0,26	0,021	$1,0 \cdot 10^{-02}$	$9,2 \cdot 10^{-03}$
Cliquet OTM	0,3	0,3	1,2	2,7	2,7	1,2	2,7	2,7	2,7	0,33	0,014	$4,2 \cdot 10^{-03}$	$4,2 \cdot 10^{-03}$
Cliquet ATM	0,35	0,32	1,2	2,4	2,5	1,2	2,4	2,5	2,5	0,32	0,011	$3,9 \cdot 10^{-03}$	$3,6 \cdot 10^{-03}$
Cliquet ITM	8,2	0,89	1,2	0,1	0,91	1,2	0,1	0,91	0,91	0,1	$1,0 \cdot 10^{-05}$	$8,5 \cdot 10^{-05}$	$9,3 \cdot 10^{-06}$

TABLE 2.3 – Results for the CIR model with $X_0 = 1$, $\sigma = 0,2$, $a = 1,5$, $k = 0,5$, $T = 1$. The first coefficient of the KL-decomposition is decomposed with 7 points and the 5 next are decomposed with 20 points (140 quantized paths). For the simulations, $N = 100\ 000$ and $M = 1\ 000\ 000$. The * indicates irrelevant results (the variance is already equal to 0).

Option	v'/v	v'_κ/v	$\frac{T'_N}{T}$	$\frac{v'T}{v'T'_N}$	$\frac{v'T'_N}{v'T}$	$\frac{T'_M}{T}$	$\frac{v'T}{v'T'_M}$	$\frac{v'T'_M}{v'T}$	μ	σ	σ'	σ'_κ
European OTM	0,3	0,28	1,5	2,2	2,3	1,3	2,6	2,9	0,096	0,086	0,026	0,024
European ATM	0,22	0,2	1,5	3	3,2	1,3	3,6	3,9	0,19	0,14	0,031	0,029
European ITM	0,15	0,15	1,5	4,2	4,4	1,3	5,2	5,4	0,53	0,24	0,036	0,034
Asian OTM	0,29	0,26	1,5	2,3	2,5	1,3	2,7	3	0,026	0,011	$3,1 \cdot 10^{-03}$	$2,8 \cdot 10^{-03}$
Asian ATM	0,13	0,11	1,5	5,1	5,6	1,3	6,2	6,8	0,1	0,035	$4,4 \cdot 10^{-03}$	$4,0 \cdot 10^{-03}$
Asian ITM	0,074	0,071	1,5	8,7	9,1	1,3	11	11	0,5	0,07	$5,2 \cdot 10^{-03}$	$5,0 \cdot 10^{-03}$
Asian Binary OTM	0,29	0,26	1,5	2,3	2,5	1,3	2,7	3	0,026	0,011	$3,1 \cdot 10^{-03}$	$2,8 \cdot 10^{-03}$
Asian Binary ATM	0,13	0,11	1,5	5,1	5,7	1,3	6,2	6,8	0,1	0,035	$4,4 \cdot 10^{-03}$	$4,0 \cdot 10^{-03}$
Asian Binary ITM	0,074	0,071	1,5	8,8	9,1	1,3	11	11	0,5	0,07	$5,2 \cdot 10^{-03}$	$5,0 \cdot 10^{-03}$
Ratchet OTM	0,086	0,083	1,5	7,5	7,8	1,3	9,1	9,4	65	$6,8 \cdot 10^{+02}$	59	57
Ratchet ATM	0,094	0,09	1,5	6,9	7,2	1,3	8,3	8,7	51	$5,9 \cdot 10^{+02}$	56	53
Ratchet ITM	0,12	0,11	1,5	5,4	5,7	1,3	6,5	6,9	30	$3,8 \cdot 10^{+02}$	46	43
Lookback fixed OTM	0,3	0,23	1,5	2,1	2,8	1,3	2,6	3,4	0,19	0,15	0,044	0,034
Lookback fixed ATM	0,21	0,17	1,5	3,1	3,8	1,3	3,7	4,6	0,39	0,2	0,042	0,034
Lookback fixed ITM	0,21	0,17	1,5	3,1	3,8	1,3	3,7	4,6	0,89	0,2	0,042	0,034
Lookback floating ITM	0,25	0,24	1,5	2,6	2,7	1,3	3,1	3,3	0,3	0,15	0,038	0,036
Cliquet OTM	0,26	0,26	1,5	2,5	2,5	1,3	3,1	3,1	0,5	0,1	0,026	0,026
Cliquet ATM	0,64	0,45	1,5	1	1,4	1,3	1,2	1,8	0,37	0,021	0,013	$9,5 \cdot 10^{-03}$
Cliquet ITM	7,7	0,86	1,5	0,084	0,75	1,3	0,1	0,91	0,098	$1,2 \cdot 10^{-04}$	$9,5 \cdot 10^{-04}$	$1,1 \cdot 10^{-04}$

TABLE 2.4 – Results for the Heston model with $V_0 = 0.2$, $X_0 = 1$, $\sigma = 0.2$, $k = 0.5$, $a = 0.2$, $\rho = 0.3$, $\nu = 0.02$, $T = 1$. Both for the underlying Brownian motion of the volatility and the price, the first coefficient of the KL-decomposition is quantized with 7 points, while the 4 next coefficients are quantized with 20 points (19 600 quantized paths). For the simulations, $N = 100\ 000$ and $M = 1\ 000\ 000$.

Bibliography

- [1] V. Bally, G. Pagès, and J. Printems. A quantization tree method for pricing and hedging multidimensional american options. Math. Finance, 15(1) :119–168, 2005.
- [2] O. Bardou, S. Bouthemy, and G. Pagès. Optimal quantization for the pricing of swing options. Appl. Math. Finance, 16(1–2) :183–217, 2009.
- [3] J.L. Bently. Multidimensional divide-and-conquer. Communications of the ACM, 23(4) :214–229, 1980.
- [4] A. Berkaoui, M. Bossy, and A. Diop. Euler scheme for SDEs with non-lipschitz diffusion coefficient : strong convergence. ESAIM Probab. Stat., 12 :1–11, 2008.
- [5] S. Corlay and G. Pagès. Functional quantization based stratified sampling methods. Monte Carlo Methods and Applications, 2015.
- [6] L. Coutin and N. Victoir. Enhanced gaussian processes and applications. ESAIM Probab. Stat., 13 :247–260, 2009.
- [7] F. Delarue and S. Menozzi. An interpolated stochastic algorithm for quasi-linear pdes. Math. Comp., 77(261) :125–158, 2008.
- [8] H. Doss. Liens entre équations différentielles stochastiques et ordinaires. Ann. Inst. H. Poincaré Sect. B (N.S.), 13(2) :99–125, 1977.
- [9] G. S. Fishman. Monte Carlo. Springer Series in Operations Research. Springer-Verlag, 1996. Concepts, algorithms, and applications.
- [10] P. Friz and N. Victoir. Differential equations driven by Gaussian signals II. Arxiv preprint, 2007. Preprint.
- [11] P. Friz and N. Victoir. Multidimensional Stochastic Processes as Rough Paths : Theory and Applications. Cambridge University Press, 2010.
- [12] P. Glasserman. Monte Carlo methods in financial engineering, volume 53 of Applications of Mathematics (New York). Springer-Verlag, 2004. Stochastic Modelling and Applied Probability.
- [13] G. H. Golub and C. F. Van Loan. Matrix computations. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, 3 edition, 1996.

- [14] S. Graf and H. Luschgy. Foundations of quantization for probability distributions, volume 1730 of Lecture Notes in Mathematics. Springer-Verlag, 2000.
- [15] I. Karatzas and S. E. Shreve. Brownian motion and stochastic calculus, volume 113 of Graduate Texts in Mathematics. Springer-Verlag, 2 edition, 1991.
- [16] A. Lejay. An introduction to rough paths. Séminaire de probabilités XXXVII, 1832 :1–59, 2003.
- [17] A. Lejay. Yet another introduction to rough paths. Séminaire de probabilités XLII, 1979 :1–101, 2009.
- [18] A. Lejay. Global solutions to rough differential equations with unbounded vector fields. Hal Preprint, 2010. Preprint.
- [19] A. Lejay and T. Lyons. On the importance of the lévy area for systems controlled by converging stochastic processes. application to homogenization. New Trends in Potential Theory, Conference Proceedings, Bucharest, September 2002 and 2003, pages 63–84, 2006.
- [20] R. Lord, R. Koekoek, and D. van Dijk. A comparison of biased simulation schemes for stochastic volatility models. Quant. Finance, 10(2) :177–194, 2010.
- [21] H. Luschgy and G. Pagès. Functional quantization of a class of brownian diffusions : a constructive approach. Stochastic Process. Appl., 116(2) :310–336, 2006.
- [22] T. Lyons and Z. Qian. System control and rough paths. Oxford Mathematical Monographs. Oxford University Press, 2002. Oxford Science Publications.
- [23] Terry J. Lyons. Differential equations driven by rough signals. Rev. Mat. Iberoamericana, 14(2) :215–310, 1998.
- [24] G. Pagès and J. Printems. Web site devoted to vector and functional optimal quantization : www.quantize.maths-fi.com, 2005.
- [25] G. Pagès and A. Sellami. Convergence of multi-dimensional quantized sde's. In Séminaire de Probabilités XLIII, volume 2006. Springer-Verlag, 2011.
- [26] Gilles Pages and Jacques Printems. Optimal quantization for finance : from random vectors to stochastic processes. Handbook of Numerical Analysis, 15 :595–648, 2008.
- [27] G. Pagès and J. Printems. Optimal quadratic quantization for numerics : the gaussian case. Monte Carlo Methods Appl., 9(2) :135–165, 2003.
- [28] Gilles Pagès, Huyên Pham, and Jacques Printems. Optimal quantization methods and applications to numerical problems in finance. In publisher=Birkhäuser Boston,, pages 253–297, 2004.
- [29] Gilles Pagès and Jacques Printems. Functional quantization for numerics with an application to option pricing. Monte Carlo Methods and Applications mcma, 11(4) :407–446, 2005. <http://quantization.math-phi.com/>.

- [30] H.J. Sussmann. On the gap between deterministic and stochastic ordinary differential equations. Ann. Probability, 6(1) :19–41, 1978.
- [31] R. Szechtman. Control variate techniques for monte carlo simulation. In Proceedings of the 2003 Winter Simulation Conference, 2003.

Bibliography

Chapitre 3

Prise en compte du spread bid-ask

Keywords : Interest rates derivatives, Optimization, Stochastic Algorithms.

Résumé

Nous considérons des taux de swaps qui échangent un taux fixe contre un taux variable en présence de coûts de transaction. Même pour des modèles simples de coûts de transaction, il n'y a pas de stratégie explicite qui optimise l'espérance de l'erreur de couverture. Nous proposons ici un algorithme efficace basé sur le gradient stochastique pour calculer une approximation de la stratégie sans résoudre un problème de contrôle stochastique. Nous validons notre algorithme par des tests numériques. Nous développons différentes variantes de l'algorithme et analysons leurs performances en terme de paramètre numérique et de coûts de transaction.

Ce travail a été publié dans [MRTT16] et effectué en grande partie dans le cadre de la collaboration entre l'équipe-projet TOSCA d'INRIA et l'équipe de recherche et développement dérivé de Taux et Hybrides du Credit Agricole CIB. Cette équipe est dirigée par Christophe Michel qui est un des co-auteurs de l'article publié.

Abstract

We consider rate swaps which pay a fixed rate against a floating rate in presence of bid-ask spread costs. Even for simple models of bid-ask spread costs, there is no explicit strategy optimizing an expected function of the hedging error. We here propose an efficient algorithm based on the stochastic gradient method to compute an approximate optimal strategy without solving a stochastic control problem. We validate our algorithm by numerical experiments. We also develop several variants of the algorithm and discuss their performances in terms of the numerical parameters and the liquidity cost.

This work was published in [MRTT16] and mainly created from a scientific collaboration with TOSCA Team of INRIA and Interest Rates and Hybrids Quantitative Research Team. This team is directed by Christophe Michel who is one of the authors of the published work.

3.1 Introduction

Classical models in financial mathematics usually assume that markets are perfectly liquid. In particular, each trader can buy or sell the amount of assets he/she needs at the same price (the “market price”), and the trader’s decisions do not affect the price of the asset. In practice, the assumption of perfect liquidity is never satisfied but the error due to illiquidity is generally negligible with respect to other sources of error such as model error or calibration error, etc.

However, the perfect liquidity assumption cannot hold true for interest rates derivatives market : on the one hand, the underlying interest rate is not directly exchangeable ; on the second hand, the liquidity costs to hedge interest rates derivatives are highly time varying (even though there exist maturities for which zero-coupon bonds are liquid, bonds at intermediate maturities may be extremely illiquid). Therefore, hedging of such derivatives absolutely needs to take liquidity risk into account. In this context, defining and computing efficient approximate perfect hedging strategies is a complex problem. The main purpose of this paper is to show that stochastic optimization methods are powerful tools to treat it without solving a necessarily high dimensional stochastic control problem, under the constraints that practitioners need to trade at prescribed dates and that relevant strategies depend on a finite number of parameters. More precisely, we construct and analyse an efficient original numerical method which provides strategies facing liquidity costs and minimizing hedging errors.

The outline of the paper is as follows. Section 3.2 introduces the model. In Section 3.3, we present our numerical method and analyse it from a theoretical point of view within the framework of a Gaussian yield curve model. Section 3.4 is devoted to numerical validation in the idealistic perfect liquidity context. In Section 3.5, we develop an empirical study of the efficiency of our algorithm in presence of liquidity cost.

3.2 Our settings : swaps with liquidity cost

3.2.1 A short reminder on swaps and swap options hedging without liquidity cost

One of the most common swaps on the interests rates market is as follows. The counterparts exchange two coupons : the first one is generated by a bond (with a constant fixed interest rate) and the second one is generated by a floating rate (e.g. a LIBOR).

Definition 5. In a perfectly liquid market, the price at time t of a zero-coupon bond paying 1 at time T is denoted by $B(t, T)$. The linear forward rate $L(T_F, T_B, T_E)$ is the price decided at time T_F , paid at time T_B to receive 1 at time T_E . These two prices satisfy the relation

$$L(T_F, T_B, T_E) = \frac{1}{T_E - T_B} \left(\frac{B(T_F, T_B)}{B(T_F, T_E)} - 1 \right). \quad (3.1)$$

A swap contract specifies :

- an agreement date t
- a time line ($t \leq$) $T_0 < \dots < T_N$
- a fixed interest rate r
- a floating interest rate
- the payoff at each time T_i ($1 \leq i \leq N$) is

$$P(i) := (T_i - T_{i-1}) (r - L(T_{i-1}, T_{i-1}, T_i)). \quad (3.2)$$

From (3.1), we deduce the equivalent expression

$$P(i) = r(T_i - T_{i-1}) - \frac{1}{B(T_{i-1}, T_i)} + 1. \quad (3.3)$$

In the sequel, we consider that the fixed rate r is chosen at the money (thus the swap at time t has zero value), and that the swap fixed coupons are received by the trader.

In the idealistic framework of a market without liquidity cost, the trader buys or sells quantities of zero-coupon bonds at the same price (i.e. the *market price*), and there exists a discrete dynamic perfect hedging strategy which is independent of any model of interest rates. In view of (3.3), the replication of the payoff $P(i)$ at time T_i can be split into three parts :

- the fixed part $r(T_i - T_{i-1})$ is replicated statically at time t by selling $r(T_i - T_{i-1})$ zero-coupon bonds with maturity T_i .
- the floating part $1/B(T_{i-1}, T_i)$ is replicated dynamically at time T_{i-1} by buying $1/B(T_{i-1}, T_i)$ zero-coupon bonds with maturity T_i . The price of this transaction is equal 1.
- the last (fixed) part 1 is used at time T_i to buy $1/B(T_i, T_{i+1})$ zero-coupon bonds with maturity T_{i+1} .

It is easy to see that this strategy is self-financing at time T_1, \dots, T_{N-1} . To make it self-financing at any time, at time t , 1 zero-coupon bond with maturity T_0 is bought and 1 zero-coupon bond with maturity T_N is sold.

To summarize, in the idealistic framework, we do not need hedging strategies within the set of all the $(\mathcal{F}_\theta^R, \theta \geq 0)$ adapted processes, where $(\mathcal{F}_\theta^R, \theta \geq 0)$ is the filtration generated by the interest rate $(R_\theta, \theta \geq 0)$, and we may restrict the admissible strategies to be adapted to the filtration generated by the rates at times $\{t, T_0, T_1, \dots, T_N\}$.

3.2.2 Hypotheses on markets with liquidity costs

We now consider markets with liquidity costs and need to precise our liquidity cost model. In all the sequel T_{-1} denotes t .

Hypothesis 6.

We assume that, for all $-1 \leq j < i \leq N$, the number $\pi(j, i)$ of zero-coupon bonds with maturity T_i bought or sold at time T_j ($j < i$) is measurable with respect to the filtration generated by $(R_t, R_{T_0}, \dots, R_{T_j})$. That means that the admissible strategies do not depend on the evolution of the rate R_θ between two dates T_m and T_{m+1} .

Denote by $\Psi(T, U, \pi)$ the buy or sell price for π zero coupon bonds. In perfectly liquid markets, $\Psi(T, U, \pi)$ is the linear function $B(T, U)\pi$, where $B(T, U)$ is defined in Definition 5. In the presence of liquidity costs, $\Psi(T, U, \pi)$ becomes a non-linear function of π .

Hypothesis 7.

For all T and U , the price $\Psi(T, U, \pi)$ is a $C^1(\mathbb{R})$, increasing, convex one-to-one map of π from \mathbb{R} to \mathbb{R} , and $\Psi(T, U, 0) = 0$.

Under the preceding hypothesis, the function Ψ is positive when $\pi > 0$ and negative when $\pi < 0$.

In the context of the swap options, we set

$$\Psi_{i,j}(\pi) := \Psi(T_i, T_j, \pi) \quad (3.4)$$

and we only consider self-financing strategies, that is, satisfying

$$\forall 0 \leq j \leq N-1, \quad \sum_{-1 \leq k < j} \pi(k, j) + P(j) = \sum_{j < i \leq N} \Psi_{j,i}(\pi(j, i)). \quad (3.5)$$

3.2.3 Optimization objective

In the presence of liquidity costs, the market is no more complete and the practitioners need to build a strategy which minimizes a given function S (e.g. a risk measure) of the hedging error. Such strategies are usually obtained by solving stochastic control problems. These problems require high complexity numerical algorithms which are too slow to be used in practice. We here propose an efficient and original numerical method to compute approximate optimal strategies. As the perfect hedging leads to a null portfolio at time T_N , we have to solve the optimization problem

$$\inf_{\pi \in \Pi} \mathbb{E}[S(W^\pi)], \quad (3.6)$$

where W^π is the terminal wealth (at time T_N) given the strategy π in the set Π of admissible strategies.

3.3 Hedging error minimization method in a Gaussian framework

The methodology we introduce in this section is based on the two following key observations :

- (1) We consider strategies and portfolios with finite second moment, and thus optimize within $L^2(\mu)$ for some probability measure μ . The Gram-Schmidt procedure provides countable orthogonal basis \mathcal{B} of a separable Hilbert space $L^2(\mu)$. The set Π of admissible strategies is defined by truncation of the basis, which reduces the a priori infinite dimensional optimization problem (3.6) to a finite dimensional parametric optimization problem of the type $\inf_{\theta \in \Theta} \mathbb{E}\Psi(\theta, X)$, where Θ is a subset of \mathbb{R}^p , X is a given random variable, Ψ is a convex function of θ .
- (2) Robbins Monro algorithm and Chen's extension are stochastic alternatives to Newton's method to numerically solve such optimization problems. These algorithms avoid to compute $\frac{d}{d\theta} \mathbb{E}\Psi(\theta, X)$. They evolve as

$$\theta_{\gamma+1} = \theta_{\gamma} - \rho_{\gamma+1} \frac{\partial}{\partial \theta} \Psi(\theta_{\gamma}, X_{\gamma+1}), \quad (3.7)$$

where $(\rho_{\gamma}, \gamma \geq 1)$ is a decreasing sequence and $(X_{\gamma}, \gamma \geq 1)$ is an i.i.d. sequence of random variables distributed as X .

We here consider the case of swap options in the context of a Gaussian yield curve. This assumption is restrictive but is satisfied by widely used models such as Vasicek model or Gaussian affine models. In [1] it is shown that using a three dimensional Gaussian model is sufficient to fit the term structure of interest rates products.

3.3.1 Step 1 : finite dimensional projections of the admissible controls space

Consider a Gaussian interest rates model $(R_{\theta}, \theta \geq 0)$. In view of Hypothesis 6, each control $\pi(j, i)$ belongs to the Gaussian space generated by $(R_t, R_{T_0}, \dots, R_{T_j})$ or, equivalently, to a space generated by $\ell(j) + 1$ standard independent Gaussian random variables $G^{(0)}, G^{(1)}, \dots, G^{(\ell(j))}$ (with $\ell(j) = j$ for one-factor models, $\ell(j) = 2j + 1$ for two-factor models, etc.). An explicit L^2 orthonormal basis of the space generated by $G^{(0)}, G^{(1)}, \dots, G^{(\ell(j))}$ is

$$\left(\prod_{m=0}^{\ell(j)} \frac{H_{n_m}(G^{(m)})}{\sqrt{n_m!}} \right)_{(n_0, \dots, n_{\ell(j)}) \in \mathbb{N}^{\ell(j)+1}}, \quad (3.8)$$

where $(H_n, n \geq 0)$ are the Hermite polynomials

$$H_n(x) = (-1)^n e^{x^2/2} \frac{d^n}{dx^n} e^{-x^2/2}$$

(see e.g. [9, p.236]).

Thus, the quantities of zero-coupon bonds bought by the trader can be written as

$$\sum_{\mathbf{n}_{(j)} \in \mathbb{N}^{\ell(j)+1}} \alpha^{\mathbf{n}_{(j)}}(j, i) \prod_{m=0}^{\ell(j)} \frac{H_{n_m}(G^{(m)})}{\sqrt{n_m!}}, \quad \mathbf{n}_{(j)} = (n_0, \dots, n_{\ell(j)}), \quad (3.9)$$

where the equality has an L^2 limit sense. A strategy can now be defined as a sequence of real numbers $\alpha^{\mathbf{n}^{(j)}}(j, i)$ for all $-1 \leq j < i \leq N - 1$ and $\mathbf{n}^{(j)} \in \mathbb{N}^{\ell(j)+1}$.

In order to be in a position to solve a finite dimensional optimization problem, we truncate the sequence $(\alpha^{\mathbf{n}^{(j)}}(j, i))$. Then a strategy is defined by a finite number of real parameters $\{\alpha^{\mathbf{n}^{(j)}}(j, i), -1 \leq j < i \leq N - 1\}$ where, for all j , $\mathbf{n}^{(j)}$ belongs to a finite subset $\Lambda^{(j)}$ of $\mathbb{N}^{\ell(j)+1}$. The truncated quantities of zero-coupon bonds bought by the trader write

$$\pi(j, i) = \sum_{\mathbf{n}^{(j)} \in \Lambda^{(j)}} \alpha^{\mathbf{n}^{(j)}}(j, i) \prod_{m=0}^{\ell(j)} \frac{H_{n_m}(G^{(m)})}{\sqrt{n_m!}}. \quad (3.10)$$

We discuss the efficiency of this truncation and its convergence in Sec. 3.3.4 and show numerical results in Sec. 3.4.1.

To simplify, we denote by $\boldsymbol{\alpha} = (\alpha^{\mathbf{n}^{(j)}}(j, i))_{i, j, \mathbf{n}^{(j)}}$ the parameters to optimize in \mathbb{R}^p (where the dimension p is known for each truncation $(\Lambda^{(j)}, j = -1, 0, \dots, N - 1)$), by $\boldsymbol{\pi}(\boldsymbol{\alpha})$ (or, when no confusion is possible, simply $\boldsymbol{\pi}$) the hedging strategy corresponding to a vector $\boldsymbol{\alpha} \in \mathbb{R}^p$, see (3.10). Given the strategy $\boldsymbol{\pi} = \boldsymbol{\pi}(\boldsymbol{\alpha})$, the terminal wealth $W^{(\boldsymbol{\alpha})}$ (at time T_N) satisfies

$$W^{(\boldsymbol{\alpha})} = \sum_{-1 \leq j < N} \boldsymbol{\pi}(\boldsymbol{\alpha})(j, N) + P(N) = \sum_{-1 \leq j < N} \pi(j, N) + P(N). \quad (3.11)$$

The problem (3.6) is now formulated as

$$\inf_{\boldsymbol{\alpha} \in \mathbb{R}^p} \mathbb{E} \left[S \left(W^{(\boldsymbol{\alpha})} \right) \right]. \quad (3.12)$$

3.3.2 Step 2 : stochastic optimization

Using the self-financing equation (3.5) one can express $W^{(\boldsymbol{\alpha})}$ as a function of $\boldsymbol{\alpha}$ and $(G^{(0)}, \dots, G^{(\ell(N))})$. Therefore one needs to minimize the expectation of a deterministic function of a parameter $\boldsymbol{\alpha}$ in \mathbb{R}^p and the random vector $(G^{(0)}, \dots, G^{(\ell(N))})$. Such problems can be solved numerically by classical stochastic optimization algorithms, such as those introduced in the pioneering work of Robbins and Monro [10] and its extensions (e.g. Chen and Zhu [3]). We refer the interested reader to the classical references [5, 2, 6].

In our context (3.12), the Robbins Monro algorithm (3.7) works as follows. Start with an arbitrary initial condition $\boldsymbol{\alpha}_0$ in \mathbb{R}^p . At step $\gamma + 1$, simulate independent Gaussian random variables $(G_{\gamma+1}^{(0)}, \dots, G_{\gamma+1}^{(\ell(N))})$ and compute the terminal wealth $W^{(\boldsymbol{\alpha}_\gamma)}$. Then, update the parameter $\boldsymbol{\alpha}$ by the induction formula

$$\boldsymbol{\alpha}_{\gamma+1} = \boldsymbol{\alpha}_\gamma - \rho_{\gamma+1} \nabla_{\boldsymbol{\alpha}} \left[S(W^{(\boldsymbol{\alpha}_\gamma)}) \right], \quad (3.13)$$

where (ρ_γ) is a deterministic decreasing sequence. In addition, one can use an improvement of this algorithm due to Chen and Zhu [3]. Let $(K_l, l \geq 0)$ be an

increasing sequence of compact sets such that

$$K_l \subset \text{Int}(K_{l+1}) \quad \text{and} \quad \lim_l K_l = \mathbb{R}^p, \quad (3.14)$$

where $\text{Int}(K_{l+1})$ denotes the interior of the set K_{l+1} . The initial condition α_0 is assumed to be in K_0 and we set $l(0) = 0$. At each step γ in (3.13), if $\alpha_{\gamma+1} \in K_{l(\gamma)}$, we set $l(\gamma+1) = l(\gamma)$ and go to step $\gamma+1$. Otherwise, that is if $\alpha_{\gamma+1} \notin K_{l(\gamma)}$, we set $\alpha_{\gamma+1} = \alpha_0$ and $l(\gamma+1) = l(\gamma)+1$. This modification avoids that the stochastic algorithm may blow up during the first steps and, from a theoretical point of view, allows to prove its convergence under weaker assumptions than for the standard Robbins Monro method.

3.3.3 Summary of the method

Our setting

- The interest rate model satisfies : for all $0 \leq j < N$, there exist an integer $\ell(j)$ and a function Φ_j such that

$$(R_t, R_{T_0}, \dots, R_{T_j}) \stackrel{\mathcal{L}}{=} \Phi_j(R_t, G^{(0)}, \dots, G^{(\ell(j))}).$$

- For all $0 \leq j < N$, a finite truncation $\Lambda^{(j)} \subset \mathbb{N}^{\ell(j)+1}$ is given and $\Lambda^{(-1)} = \{0\}$.
- A strategy $\pi = \pi(\alpha)$ is defined by $\alpha = (\alpha^{\mathbf{n}^{(j)}}(j, i), -1 \leq j < i \leq N - 1, \mathbf{n}^{(j)} \in \Lambda^{(j)})$. More precisely, the number of zero-coupon bonds with maturity T_i bought or sold at time T_j is

$$\pi(j, i) = \sum_{\mathbf{n}^{(j)} \in \Lambda^{(j)}} \alpha^{\mathbf{n}^{(j)}}(j, i) \prod_{m=0}^{\ell(j)} \frac{H_{n_m}(G^{(m)})}{\sqrt{n_m!}}, \quad \text{for } i \leq N - 1, \quad (3.15)$$

and $\pi(j, N)$ is deduced from the self-financing equation

$$\sum_{i < j} \pi(i, j) + P(j) = \sum_{i > j} \Psi_{j,i}(\pi(j, i)) \quad (3.16)$$

(one possibly needs to use a classical iterative procedure to solve this equation numerically).

- One is given an increasing sequence of compact sets $(K_l, l \geq 0)$ satisfying (3.14) and a sequence of parameters $(\rho_\gamma, \gamma \geq 1)$, decreasing to 0.

Our stochastic optimization algorithm

Assume that the parameter $\alpha_\gamma = (\alpha_\gamma^{\mathbf{n}^{(j)}}(j, i), -1 \leq j < i \leq N - 1, \mathbf{n}^{(j)} \in \Lambda^{(j)})$ and l_γ are given at step γ . At step $\gamma+1$:

1. Simulate a Gaussian vector $(G_{\gamma+1}^{(0)}, \dots, G_{\gamma+1}^{(\ell(N))})$.

2. Deduce the quantities of zero-coupon bonds from (3.15) and (3.16)

$$\pi_{\gamma+1}(j, i) = \sum_{\mathbf{n}^{(j)} \in \Lambda^{(j)}} \alpha_{\gamma}^{\mathbf{n}^{(j)}}(j, i) \prod_{m=0}^{\ell(j)} \frac{H_{n_m}(G_{\gamma+1}^{(m)})}{\sqrt{n_m!}}, \quad \text{for } i \leq N-1,$$

and get $\pi_{\gamma+1}(j, N)$ from the self-financing equation

$$\sum_{i < j} \pi_{\gamma+1}(j, i) + P(j) = \sum_{i > j} \Psi_{j,i}(\pi_{\gamma+1}(j, i)).$$

3. Compute the terminal wealth

$$W_{\gamma+1}^{(\alpha_{\gamma})} = \sum_{j < N} \pi_{\gamma+1}(j, N) + P(N).$$

4. Update the parameters

$$\alpha_{\gamma+1} = \alpha_{\gamma} - \rho_{\gamma+1} \nabla_{\alpha} \left[S(W_{\gamma+1}^{(\alpha_{\gamma})}) \right].$$

5. If $\alpha_{\gamma+1} \notin K_{l_{\gamma}}$, set $\alpha_{\gamma+1} = \alpha_0$ and $l(\gamma+1) = l(\gamma) + 1$.

6. Go to 1.

3.3.4 Error analysis

In this subsection we study the convergence (Theorem 8) and convergence rate (Theorem 10) of the stochastic algorithm used in Step 2, when the number of steps γ tends to infinity. We introduce some notation. Recall (3.13) and write

$$\alpha_{\gamma+1} = \alpha_{\gamma} - \rho_{\gamma+1} \mathbb{E} \left[\nabla_{\alpha} S(W_{\gamma+1}^{(\alpha_{\gamma})}) \right] - \rho_{\gamma+1} \delta M_{\gamma+1} + \rho_{\gamma+1} p_{\gamma+1}. \quad (3.17)$$

Here, $\delta M_{\gamma+1}$ is given by

$$\delta M_{\gamma+1} = \nabla_{\alpha} S(W_{\gamma+1}^{(\alpha_{\gamma})}) - \mathbb{E} \left[\nabla_{\alpha} S(W_{\gamma+1}^{(\alpha_{\gamma})}) \right]. \quad (3.18)$$

The last term in (3.17) represents the *reinitialization* of the algorithm if $\alpha_{\gamma+1} \notin K_{l(\gamma)}$.

Let us now recall the convergence theorem obtained by Lelong [7, Th. 1] in our setting.

Theorem 8. Assume

(A1) The function $\alpha \mapsto \mathbb{E}[S(W^{(\alpha)})]$ is strictly concave or convex,

(A2) $\sum_{\gamma} \rho_{\gamma} = \infty$, $\sum_{\gamma} \rho_{\gamma}^2 < \infty$,

(A3) The function $\alpha \mapsto \mathbb{E}[\|\nabla_{\alpha} S(W^{(\alpha)})\|^2]$ is bounded on compact sets.

Then the sequence $(\alpha_{\gamma}, \gamma \geq 1)$ converges a.s. to the unique optimal parameter α^* such that

$$\inf_{\alpha \in \mathbb{R}^p} \mathbb{E} \left[S \left(W^{(\alpha)} \right) \right] = \mathbb{E} \left[S \left(W^{(\alpha^*)} \right) \right].$$

Hypothesis 7 and (3.15) imply that (A3) is satisfied. Before giving examples of situations where (A1) is fulfilled, let us check that $W^{(\alpha)}$ is a concave function of α .

Proposition 9. The terminal wealth $W^{(\alpha)}$ is a concave function of the parameter α .

Proof. Recall that the terminal wealth is given by (3.11). The payoff of the swap $P(N)$ does not depend on α . We only have to deal with the quantities $\pi(i, N)$ of the zero-coupon bonds with maturity T_N bought at time T_i . They solve the self-financing equation (3.5) and thus

$$\pi(i, N) = \Psi_{i,N}^{(-1)} \left(\sum_{j < i} \pi(j, i) - \sum_{i < j < N} \Psi_{i,j}(\pi(i, j)) + P(i) \right), \quad (3.19)$$

where $\Psi_{i,N}^{(-1)}$ is the inverse of the price function $\Psi_{i,N}$ (see (3.4)). Moreover, quantities $\pi(i, j)$, $i < j \leq N - 1$ are linear in α (see (3.9)).

Recall that $\Psi_{i,j}$ is convex, thus $-\Psi_{i,j}$ is concave and the argument in (3.19) is a concave function of α . Finally $\Psi_{i,N}^{(-1)}$ is an increasing concave function, from which $\pi(i, N)$ is a concave function of α . \square

The preceding observation shows that (A1) is satisfied when S is a utility function (and thus increasing and concave) and satisfies $S(0) = 0$. Notice that the optimization problem (3.12) then penalizes the losses and promotes the gains. In Sections 3.4 and 3.5 we will see another situation where Theorem 8 applies.

Given suitable functions S , Theorem 8 guarantees the convergence of our algorithm towards the optimal parameters. The following theorem provides the rate of convergence (see [8]).

Theorem 10. Let

$$\rho_\gamma := \frac{v_1}{(v_2 + \gamma)^\beta}, \quad (3.20)$$

for some positive v_1, v_2 and $\beta \in (1/2, 1)$. Denote by Δ_γ the normalized centered error

$$\Delta_\gamma = \frac{\alpha_\gamma - \alpha^*}{\sqrt{\rho_\gamma}}.$$

Assume

(A1) The function $\alpha \mapsto \mathbb{E}[S(W^{(\alpha)})]$ is concave or convex.

(A4) For any $q > 0$, the series

$$\sum_{\gamma} \rho_{\gamma+1} \delta M_{\gamma+1} \mathbb{1}_{\{|\alpha_\gamma - \alpha^*| \leq q\}}$$

converges almost surely.

(A5) There exist two real numbers $A_1 > 0$ and $A_2 > 0$ such that

$$\sup_{\gamma} \mathbb{E} \left[|\delta M_{\gamma}|^{2+A_1} \mathbb{1}_{\{|\alpha_{\gamma} - \alpha^*| \leq A_2\}} \right] < \infty.$$

(A6) There exists a symmetric positive definite matrix Σ such that

$$\mathbb{E} \left[\delta M_{\gamma} \delta M_{\gamma}^t \middle| \mathcal{F}_{\gamma-1} \right] \mathbb{1}_{\{|\alpha_{\gamma-1} - \alpha^*| \leq \eta\}} \xrightarrow[\gamma \rightarrow \infty]{\mathbb{P}} \Sigma.$$

(A7) There exists $\mu > 0$ such that $\forall n \geq 0$, $d(\alpha^*, \partial K_n) \geq \mu$,

where ∂K_n denotes the boundary of K_n . Then, the sequence $(\Delta_{\gamma}, \gamma \geq 1)$ converges in distribution to a normal random variable with mean 0 and covariance

$$V = \int_0^{\infty} \exp(-At) \Sigma \exp(-A^t t) dt.$$

Remark 1. As explained in detail in [8, Sec. 2.4], the assumptions of Theorem 10 are satisfied as soon as

— There exists $A > 0$ and $C > 0$ such that

$$\mathbb{E} \left[\sup_{|\alpha| \leq C} \left| S(W(\alpha)) \right|^{2+A} \right] < \infty.$$

— The function $\alpha \mapsto \mathbb{E}[S(W(\alpha))]$ is strictly concave or convex.

3.3.5 Performance of the optimal truncated strategy without liquidity costs

The numerical error on the optimal wealth decreases when the $\Lambda^{(j)}$'s tend to $\mathbb{N}^{\ell(j)+1}$. In this subsection, we provide a theoretical estimate on the error resulting from the truncation in (3.9) in the idealistic context of no liquidity costs and general Gaussian affine models (see [4]).

In [4], general Gaussian affine models are introduced for which, for any times $s < t$, there exist standard independent Gaussian random variables $G^{(0)}, \dots, G^{(M)}$ and real numbers $\mu, \lambda_0, \dots, \lambda_M$ such that the prices of zero-coupon bonds have the form

$$B(s, t) = \exp \left(-\mu - \lambda_0 G^{(0)} - \dots - \lambda_M G^{(M)} \right).$$

A control of the error of truncation is given in the following proposition.

Proposition 11. In the above context, if the truncation set defined in (3.10) is $\Lambda^{(j)} := \{n_0 + \dots + n_{\ell(j)} \leq d\}$, then

$$\mathbb{E} \left(W(\alpha^*) \right)^2 \leq C_0 \frac{C_1^{d+1}}{(d+1)!}, \quad (3.21)$$

where C_0 and C_1 are some positive constants.

The proposition is a straightforward consequence of (3.11) and the next lemma applied to $X = \boldsymbol{\pi}(\boldsymbol{\alpha}^*)$. This lemma also allows one to precise the values of C_0 and C_1 .

Lemma 2. *Consider the random variable*

$$X := \exp\left(\mu + \sum_{m=0}^M \lambda_m G^{(m)}\right),$$

where $\mu, \lambda_0, \dots, \lambda_M$ are real numbers, and $G^{(0)}, \dots, G^{(M)}$ are independent standard Gaussian random variables. Consider the projection X^d of X on the subspace of $L^2(G^{(0)}, \dots, G^{(M)})$ generated by

$$\left(\prod_{m=0}^M \frac{H_{n_m}(G^{(m)})}{\sqrt{n_m!}}, \quad n_0 + \dots + n_M \leq d\right).$$

We have

$$\|X - X^d\|_2 \leq \exp(\mu + \lambda_0^2 + \dots + \lambda_M^2) \frac{(\lambda_0^2 + \dots + \lambda_M^2)^{\frac{d+1}{2}}}{\sqrt{(d+1)!}}. \quad (3.22)$$

We postpone the proof of this lemma to the Appendix.

3.4 Numerical validation of the optimization procedure : an example without liquidity costs

In this section we study the accuracy of our algorithm in the no liquidity costs case, in which a perfect replication strategy is known (see Section 3.2.1). The bond market model is the Vasicek model which is the simplest Gaussian model :

$$dR_\theta = A(r_\infty - R_\theta)d\theta + \sigma dB_\theta, \quad (3.23)$$

where A is the mean reverting rate, r_∞ is the mean of the equilibrium measure, σ is the volatility and $(B_\theta, \theta \geq 0)$ is a one-dimensional Brownian motion.

Notice that

$$\forall u < v, \quad R_v = r_\infty + (R_u - r_\infty)e^{-A(v-u)} + \sigma e^{-A(v-u)} \int_u^v e^{A(\theta-u)} dB_\theta. \quad (3.24)$$

Therefore, there exists an *i.i.d.* sequence $(G^{(0)}, G^{(1)}, \dots, G^{(N)})$ of $\mathcal{N}(0, 1)$ Gaussian random variables such that

$$\forall k = 0, \dots, N, \quad R_{T_k} = \Phi(T_k - T_{k-1}, R_{T_{k-1}}, G^{(k)}), \quad (3.25)$$

where

$$\Phi(\eta, r, G) = r_\infty + e^{-A\eta}(r - r_\infty) + G\sigma \sqrt{\frac{1 - e^{-2A\eta}}{2A}}.$$

In our numerical experiments, we have chosen the following typical values of the parameters $A = 10\%$, $r_\infty = 5\%$, $\sigma = 5\%$. With this choice of parameters, the mean yearly interest zero-coupon rates with maturity less than 10 years take values between 3% and 5%.

Our numerical study concerns the minimization of the quadratic mean hedging error which corresponds to the choice $S(x) = x^2$ in (3.12). This choice penalizes gains and losses in a symmetric way and aims to construct a strategy as close as possible to the exact replication strategy. In the no liquidity cost case, the terminal wealth $W^{(\alpha)}$ is a linear function of the parameter α and therefore assumption (A1) of Theorem 8 is obviously satisfied.

Our truncation $\Lambda^{(j)}$ is chosen as

$$\Lambda_d^{(j)} := \{\mathbf{n}_{(j)} = (n_0, \dots, n_j) \in \mathbb{N}^{j+1}, n_0 + \dots + n_j \leq d\}, \quad (3.26)$$

for a degree d of truncation. We have to optimize the real valued parameters $\alpha^{\mathbf{n}_{(j)}}(j, i)$ for $j < i$ and $\mathbf{n}_{(j)} \in \Lambda_d^{(j)}$ and the quantities of zero-coupon bonds to exchange are given by (3.10). The choice of the sequence $(\rho_\gamma, \gamma \geq 1)$ in (3.13) is crucial. Choose ρ_γ as in (3.20). We discuss the sensitivity of the method to the parameters v_1, v_2, β in Section 3.4.2.2. We also discuss the sensitivity of the results to the number Γ of steps.

In all the sequel, we use the following notation.

Notation For all vector $\alpha = (\alpha^{\mathbf{n}_{(j)}}(j, i), -1 \leq j < i \leq N - 1)$, we set

$$v(\alpha) := \mathbb{E} \left[W^{(\alpha)} \right]^2, \quad (3.27)$$

where the expectation is computed only with respect to the Gaussian distribution $(G^{(0)}, \dots, G^{(\ell(N))})$.

3.4.1 Empirical study of the truncation errors (Step 1)

In this subsection we develop an empirical validation of the projection step presented in Section 3.3.1

We observe that the quadratic mean hedging error decreases very fast to 0 when the degree of the truncation increases. For a notional of 1, the error is of the order of one basis point (a hundredth of percent) for a degree $d = 3$ and a small number of dates N and for $d = 4$ and for larger values of N .

Figure 3.1 below shows $v(\alpha^{*,d})$, with the optimal parameter $\alpha^{*,d}$ for truncation set given by (3.26). We have used the explicitly known finite dimensional projections of the optimal strategies without liquidity cost to obtain $\alpha^{*,d}$, and a Monte Carlo procedure to compute v . Table 3.1 shows some values used to plot Figure 3.1.

3.4.2 Empirical study of the optimization step (Step 2)

We know theoretically that the stochastic algorithm converges almost surely to the optimal coefficient α^* . In this part, we empirically study the convergence rate in terms of the number of steps Γ and the choice of the sequence (ρ_γ) .

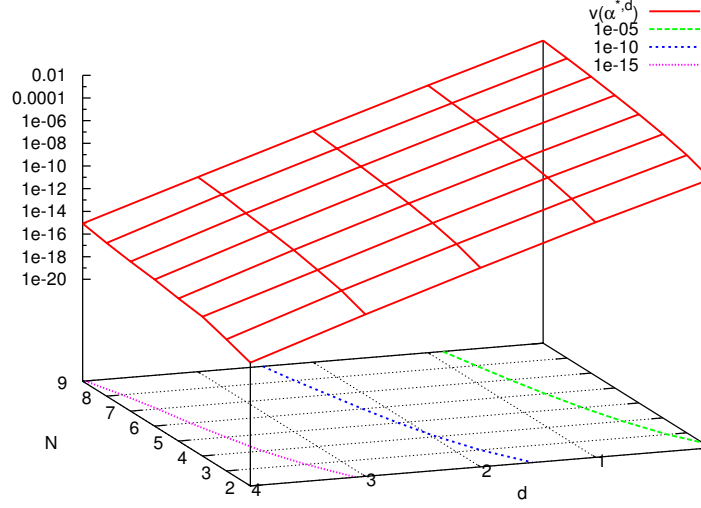


FIGURE 3.1 –

degree d	N=2	N=3
0	5.2 E-6	3.0 E-5
1	5.4 E-9	3.1 E-8
2	3.7 E-12	2.0 E-11
3	1.9 E-15	1.9 E-14
4	2.2 E-18	3.9 E-15

TABLE 3.1 – $v(\alpha^{*,d})$

3.4.2.1 A typical evolution of (α_γ)

In this subsection, we consider a swap with two payment dates ($N = 2$). We consider the truncation set $\Lambda^{(0)} = \{0, 1\}$. The objective is to approximate $\alpha^* = (\alpha^{0,*}(-1, 0), \alpha^{0,*}(-1, 1), \alpha^{0,*}(0, 1), \alpha^{1,*}(0, 1))$. In Figure 3.2, the four parameters $\alpha = (\alpha^0(-1, 0), \alpha^0(-1, 1), \alpha^0(0, 1), \alpha^1(0, 1))$ evolve according to (3.13) where the sequence $(\rho_\gamma, \gamma \geq 1)$ is defined by (3.20) with $v_1 = 10^7$, $v_2 = 1$ and $\beta = 1$.

As expected, the sequence (α_γ) converges to α^* . However, the evolution is quite *slow* although we have empirically chosen the parameters v_1 , v_2 and β in a favourable way.

In Figure 3.3, we plot (in purple) $v(\alpha^0(-1, 0), \alpha^0(-1, 1), \alpha^{0,*}(0, 1), \alpha^{1,*}(0, 1))$ as a function of $\alpha^0(-1, 0)$ and $\alpha^0(-1, 1)$. We also plot in green the path $(v(\alpha_\gamma), 0 \leq \gamma \leq \Gamma)$. The figure shows that after $\Gamma = 10000$ steps the hed-

ging error is small though the optimal parameters have not been approximated accurately (notice that the purple surface is flat).

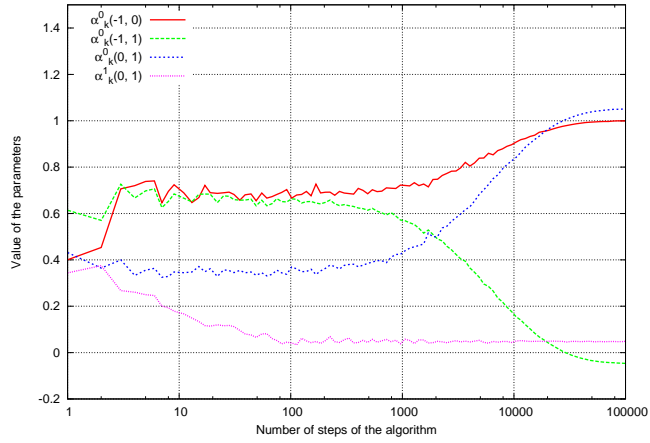


FIGURE 3.2 – Evolution of the parameters $\alpha_{\gamma}^0(-1, 0)$, $\alpha_{\gamma}^0(-1, 1)$, $\alpha_{\gamma}^0(0, 1)$, $\alpha_{\gamma}^1(0, 1)$ in terms of γ .

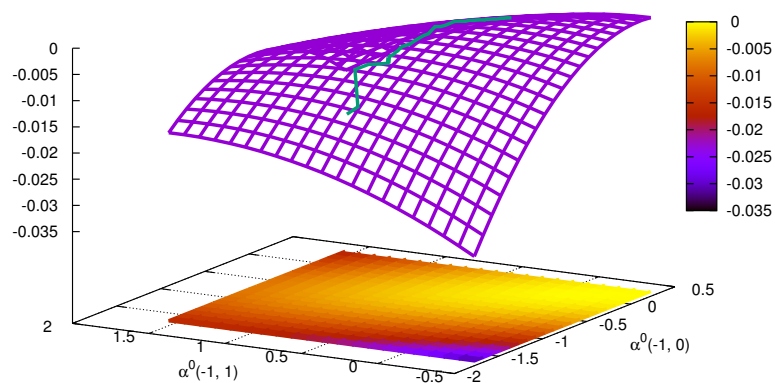


FIGURE 3.3 – Purple surface : $-v(\bullet, \bullet, \alpha^{0,*}(0, 1), \alpha^{1,*}(0, 1))$. Green curve : evolution of $(\alpha^0\gamma(-1, 0), \alpha_\gamma^0(-1, 1), -v(\alpha_\gamma^0(-1, 0), \alpha_\gamma^0(-1, 1), \alpha^{0,*}(0, 1), \alpha^{1,*}(0, 1)))$ in terms of $\gamma = 1, \dots, 10000$ ($v_1 = 10$, $v_2 = 1$ and $\beta = 0.6$).

3.4.2.2 Sensitivity to the choice of the sequence (ρ_γ)

Theorem 8 states the convergence of the optimization method for all sequence (ρ_γ) satisfying (A2). We here study the sensitivity of the results to the parameters v_1, v_2, β of sequences of type (3.20) and to the total number of steps Γ .

Tables 3.2 and 3.3 show the expected value function obtained after $\Gamma = 10E4, 10E5$ and $10E6$ steps. The expected value function is estimated by means of a classical Monte Carlo procedure. Table 3.4 shows the same results with a sequence $\rho_\gamma = v_1$ which does not satisfy condition (A2).

$\Gamma \backslash v_1$	1	10	100	1000	10 000	20 000	10E5	10E6
10E4	7.2 E-4	2.6 E-6	1.1 E-6	8.9 E-7	5.1 E-4	8.3 E-4	1.0 E-1	1.3 E-1
10E5	1.4 E-5	9.3 E-7	8.9 E-7	4.7 E-7	2.3 E-8	1.7 E-2	9.1 E+3	9.5 E+5
10E6	3.8 E-6	9.2 E-7	7.7 E-7	1.5 E-7	6.6 E-12	5.4 E-12	1.0 E-5	15.8

TABLE 3.2 – $v(\alpha_\Gamma)$ ($v_2 = 1000, \beta = 0.6$)

$\Gamma \backslash v_1$	1	10	100	1000	10 000	13 000	2E4
10E4	8.8 E-3	1.0 E-3	7.8 E-6	9.8 E-7	8.8 E-7	1.7 E-6	9.7 E-7
10E5	6.6 E-3	1.1 E-4	9.3 E-7	9.0 E-7	6.8 E-7	5.5 E-7	4.8 E-7
10E6	4.5 E-3	1.8 E-5	9.3 E-7	8.7 E-7	5.1 E-7	3.6 E-7	2.9 E-7

$\Gamma \backslash v_1$	1E5	5E5	1E6	2E6	3E6	4E6	5E6
10E4	1.3 E-3	3.3 E-2	7.3 E-1	4.5 E+1	1.1 E-2	6.5 E+2	6.9 E+1
10E5	6.6 E-7	1.2 E-5	4.1 E-4	2.7 E-1	8.1 E-3	1.7 E-2	1.4 E+1
10E6	6.8 E-8	1.1 E-10	5.3 E-12	7.4 E-12	7.0 E-12	2.6 E-5	1.5 E-6

TABLE 3.3 – $v(\alpha_\Gamma)$ ($v_2 = 1000, \beta = 0.9$)

$\Gamma \backslash v_1$	1	2	4	6	8	10	12	20
10E4	7.6 E-7	6.9 E-7	2.5 E-7	2.9 E-7	1.4 E-6	1.9 E-7	7.6 E-7	6.1 E-6
10E5	6.9 E-7	6.8 E-7	4.1 E-7	1.3 E-7	2.9 E-7	3.2 E-7	3.9 E-6	3.2 E-4
10E6	3.0 E-8	1.0 E-9	5.1 E-12	4.3 E-12	5.1 E-12	4.2 E-12	5.9 E-12	7.8 E-6

TABLE 3.4 – $v(\alpha_\Gamma)$ for a constant sequence $\rho_\gamma = v_1$

We observe that the efficiency of the algorithm depends on the choice of the parameters v_1, v_2, β and is really sensitive to it when the total number of steps Γ is small.

When Γ becomes large (e.g $\Gamma = 10E6$), then the algorithm may seem to diverge if β is chosen careless. In fact, the sequence $(\rho_\gamma, \gamma \geq 1)$ satisfies hypotheses (A2) of Theorem 8, thus the algorithm converges to the optimal parameters but

it is far from α^* after 10E6 steps. However, for each value β , there exists a choice for v_1 which reduces the mean square hedging error v to 5 E-12.

3.5 An empirical study of the bid-ask spread costs impact

We here present numerical results corresponding to two piecewise linear liquidity cost functions Ψ :

$$\Psi_\lambda^1(T, U, \pi) = (1 + \lambda \text{sign}(\pi))B(T, U)\pi \quad (3.28)$$

$$\Psi_{\lambda, C}^2(T, U, \pi) = \begin{cases} B(T, U)\pi & \text{for } |\pi| \leq C \\ B(T, U)(C + (1 + \lambda)(\pi - C)) & \text{for } \pi > C \\ B(T, U)(-C + (1 - \lambda)(\pi + C)) & \text{for } \pi < -C. \end{cases} \quad (3.29)$$

Despite the fact that we know there is no perfect hedging strategy in this context, we suppose the holder receives a null cash at time t (which is the price of the swap in a no liquidity cost market).

Notice that Theorem 8 does not apply to this context since Ψ_λ^1 and Ψ_λ^2 are piecewise linear and therefore are not continuously differentiable everywhere. Replace Ψ_λ^1 and Ψ_λ^2 by smooth approximations obtained by convolutions of the type $1/\sqrt{2\pi\varepsilon} \exp(-x^2/(2\varepsilon))$, ε small. Let α_ε^* be the unique optimal parameter corresponding to the new cost functions (existence and uniqueness of α_ε^* are provided by Theorem 8). In view of Rockafellar and Wets in [11, Th 7.33] (α_ε^* tends to α^* when ε tends to 0).

The preceding consideration is more theoretical than practical : in practice, the numerical results do not differ in the case small ε and $\varepsilon = 0$.

Given piecewise linear cost functions Ψ , it is easy to prove that the terminal wealth $W^{(\alpha)}$ is piecewise linear in α (see the proof of proposition 9) and that assumption (A1) of Theorem 8 is fulfilled.

3.5.0 Optimizing is really necessary

Consider two different strategies : (i) the strategy obtained with optimal parameters in the idealistic model without liquidity costs α_0 and (ii) the null strategy δ_0 defined as

$$\delta_0^{n(j)}(j, i) = 0, \quad \text{for all } -1 \leq j < i \leq N - 1 \text{ and } n_{(j)} \in \Lambda^{(j)}.$$

To satisfy the self-financing assumption (3.5), at time T_j the payoff $P(j)$ of the swap (3.2) is used to buy zero-coupon bonds with maturity T_N .

Figure 3.4 shows $-v(\alpha_0)$ and $-v(\delta_0)$ in terms of the parameter λ where the cost function Ψ is as (3.28). The mean square hedging error dramatically increases when the trader uses the optimal strategy in the no liquidity cost context, for λ larger than 4%.

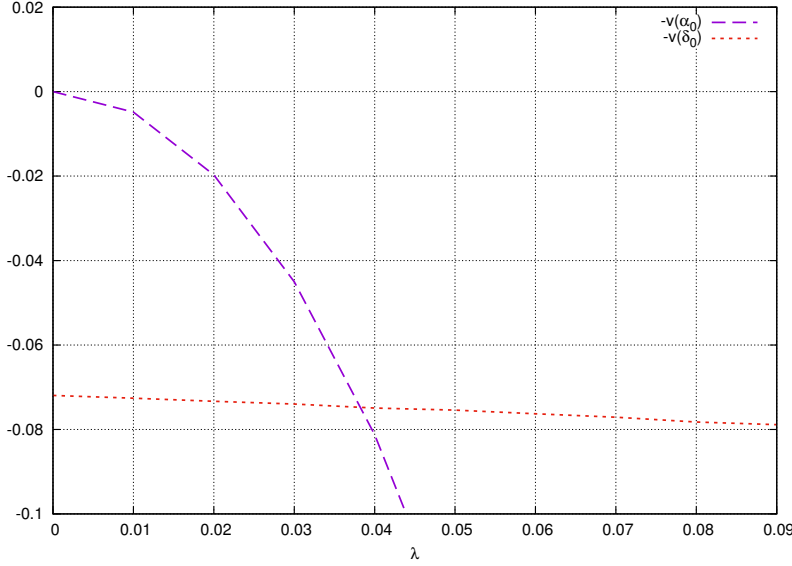


FIGURE 3.4 – (red) $-v(\delta_0)$ (purple) $-v(\alpha_0)$ in terms of the liquidity cost λ

3.5.1 Probability distribution of the hedging error in the case (3.28)

In this section, the liquidity cost function Ψ is chosen as in (3.28).

After Γ steps of the stochastic optimization procedure with a sample ω of the Gaussian vector $((G_\gamma^{(0)}, \dots, G_\gamma^{(N)}), \gamma = 1, \dots, \Gamma)$, one obtains a random approximation $\alpha_\Gamma(\omega)$ of the optimal parameter α^* .

Figure 3.5 shows the probability distribution of the random variable $v(\alpha_\Gamma(\omega))$ for $\Gamma = 10000$ and Table 3.5 shows its mean and standard deviation for different values of λ .

λ	0	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
Mean	2.5E-32	0.0031	0.012	0.024	0.038	0.049	0.058	0.065	0.11	0.12
Std dev.	6.7E-33	4.2E-5	2.8E-4	1.7E-3	0.016	0.017	0.046	0.081	1.1	1.3

TABLE 3.5 – Empirical mean and standard deviation of $v(\alpha_{10000})$.

3.5.2 Hedging error in the case (3.29)

In this section, the liquidity cost function Ψ is chosen as in (3.29).

Figure 3.6 shows $-v(\alpha_\Gamma)$ for $\Gamma = 0$ (green) and $\Gamma = 10E6$ (red). The initial parameter α_0 is the optimal one for a market without liquidity cost.

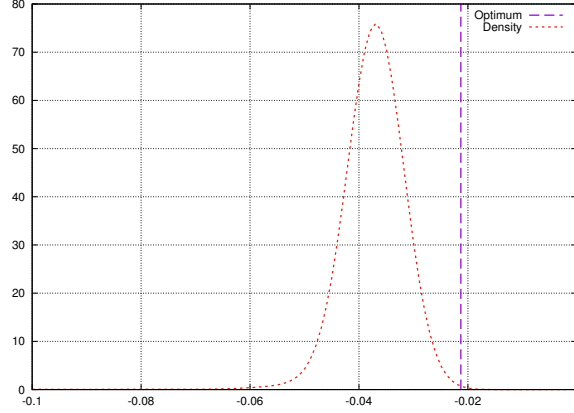


FIGURE 3.5 – Empirical density of $-v(\alpha_\Gamma(\omega))$ ($v_1 = v_2 = \beta = 1$, $\Gamma = 10000$, $\lambda = 0.04$). The purple vertical line corresponds to the $v(\alpha^*)$.

We observe that the main part of the loss is saved thanks to the optimization procedure. In Figure 3.7, we zoom on the surface resulting from the optimization procedure.

3.5.3 Influence of the initial values of the optimization procedure

We now show that, as expected, the optimized strategies are more efficient than the strategies which do not take liquidity costs into account. In Figure 3.8, we draw two functions of the liquidity cost λ : $-v(\alpha_\Gamma)$ and $-v(\delta_\Gamma)$, where (α_Γ) and (δ_Γ) are the parameters obtained after Γ steps of the optimization procedure but with different initial values α_0 and δ_0 as described in Subsec. 3.5.0. The performance of the strategies obtained after $\Gamma = 10E6$ steps are quite similar. It means that the sensitivity to the arbitrary initial parameter α_0 is no more observable after $\Gamma = 10E6$ steps.

3.5.4 Reducing the set of admissible strategies

Recall Hypothesis 6. So far, our admissible strategies at time T_j depend on all the past and present rates R_t, \dots, R_{T_j} . Thus the number of parameters $\alpha^{n(\omega)}(j, i)$ to optimize is at least of the order of magnitude of the binomial coefficient $\binom{N}{d}$, where N is the number of dates and the degree of truncation d is defined as in (3.26). This order of magnitude is a drastically increasing function of N . This crucial drawback leads us to try to simplify the complexity of the control problem (3.6) by reducing the size of the set of the admissible strategies Π . Observe that the optimal strategy under the perfect liquidity assumption has the property that $\pi^*(j, i)$ only depends on R_{T_j} . This observation suggests to face

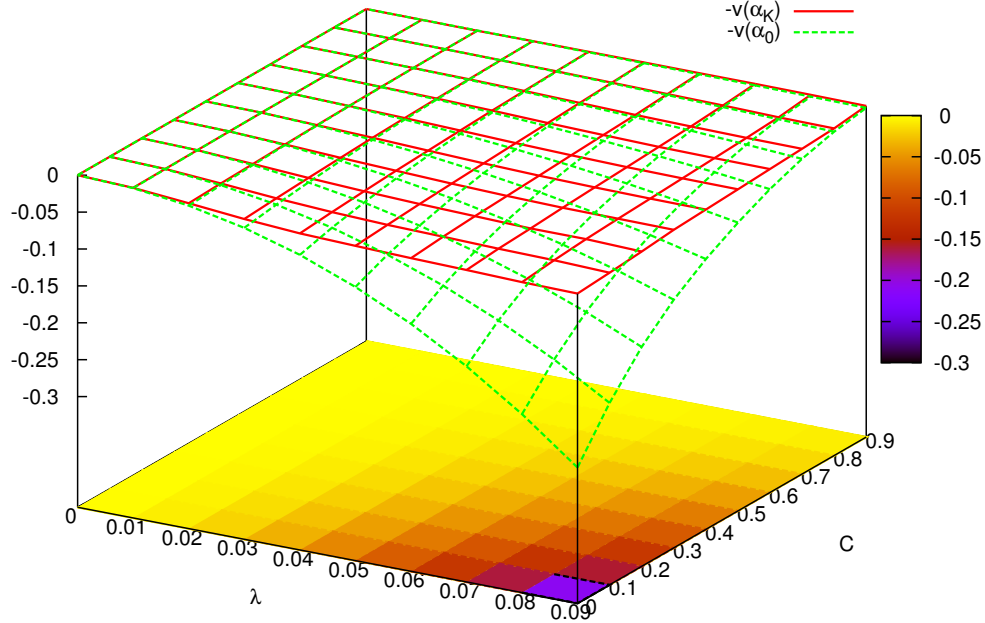


FIGURE 3.6 – Red surface : $-v(\alpha_k)$ ($\Gamma = 10E6$, $v_1 = v_2 = 100$, $\beta = 0.6$) for a liquidity cost function (3.29) in terms of the value λ of the liquidity cost and the size C of the compact. Green surface : $-v(\alpha_0)$.

large numbers of dates by reducing the set of controls to controls depending only on a small number of recent interest rates $R_{T_j}, R_{T_{j-1}}, \dots, R_{T_{j-q}}$.

Figures 3.9 and 3.10 illustrate that the optimal control problem (3.6) with admissible strategies defined as in hypothesis 6 may be used as benchmarks to possibly simpler to solve control problems. Indeed, we here consider swaps with $N = 5$ and $N = 10$ dates of payment. In each one of these two cases, we study the effect of choosing $q = 0$ (that is at time T_j , admissible strategies only depend on R_{T_j}), $q = 1$ (admissible strategies depend on R_{T_j} and $R_{T_{j-1}}$), $q = 2$.

Figure 3.9 shows the performance of the strategies obtained after $\Gamma = 10E6$ steps of the optimization algorithm for a SWAP with $N = 5$ dates of payments for $q = 0, 1$ and 2 .

Figure 3.10, shows similar quantities for a SWAP with $N = 10$ dates of payment. We observe that the numerical computation of the optimal strategy is quite unstable when q is too big, which reflects the difficulty to solve a high dimensional optimization problem. Therefore, one necessarily must choose q

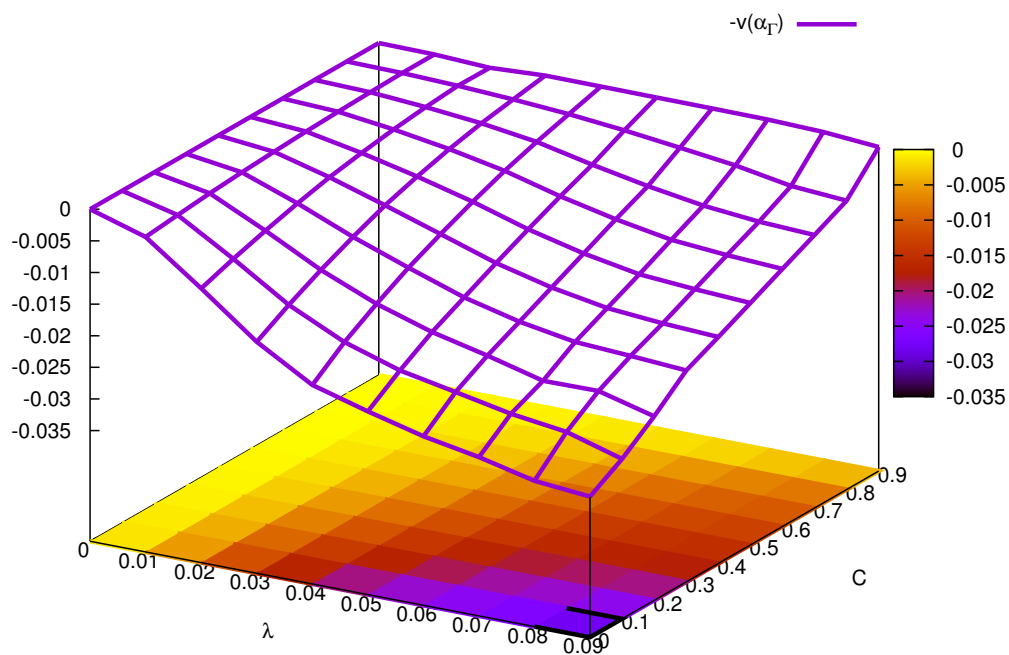


FIGURE 3.7 – A zoom on the red surface of Figure 3.6

small in order to get accurate approximations of optimal strategies belonging to reduced sets of admissible strategies.

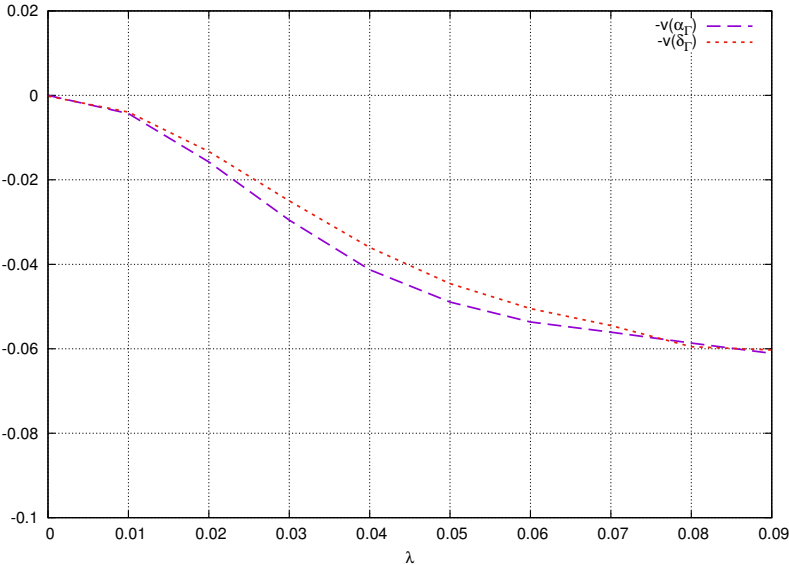


FIGURE 3.8 – (red) $-v(\delta_\Gamma)$ (purple) $-v(\alpha_\Gamma)$, in terms of the liquidity cost λ ($\Gamma = 10E6$, $v_1 = 0.1$, $v_2 = 100$, $\beta = 0.6$, $d = 3$).

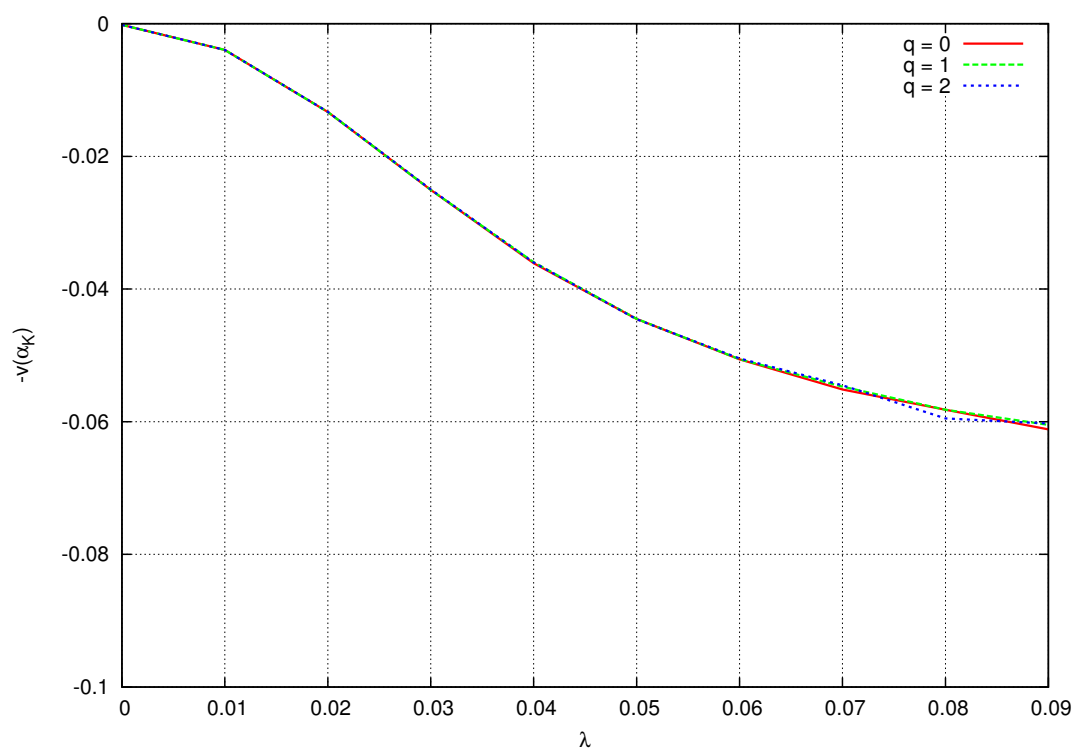


FIGURE 3.9 $-v(\alpha_\Gamma)$ in terms of λ for strategies depending only on recent rates $R_{T_j}, \dots, R_{T_{j-q}}$ ($q = 0, 1, 2$, $N = 5$, $\Gamma = 10E6$, $v_1 = 0.1$, $v_2 = 100$, $\beta = 0.6$, $d = 3$).

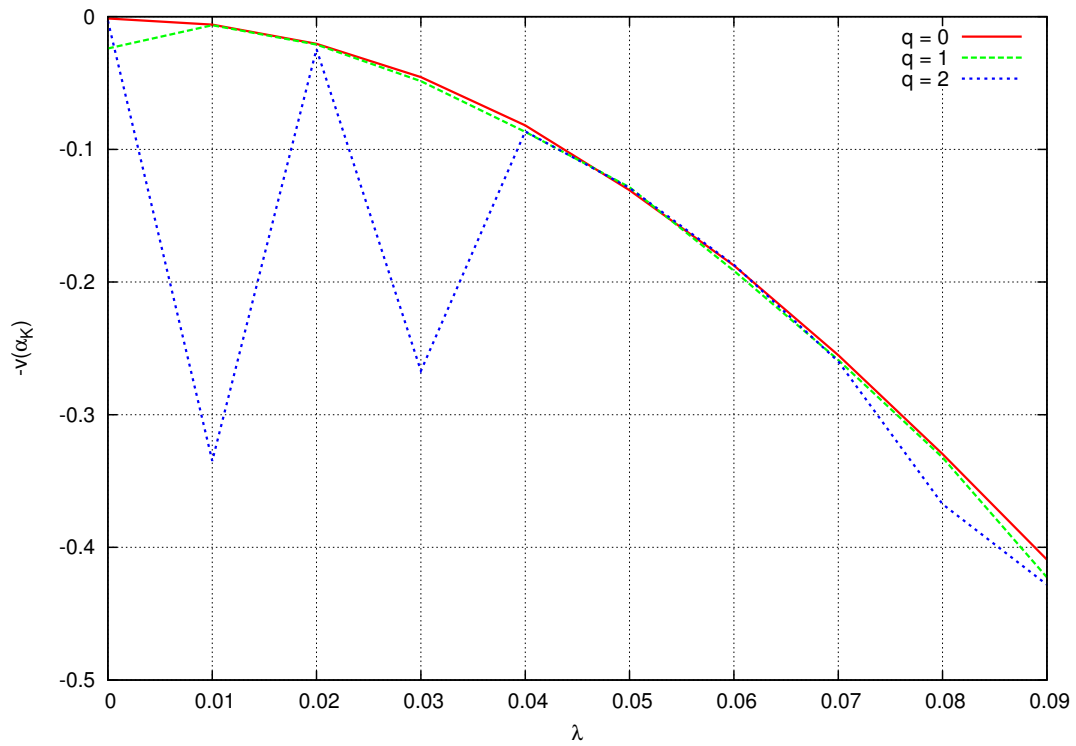


FIGURE 3.10 – $-v(\alpha_\Gamma)$ in terms of λ for strategies depending only on recent rates $R_{T_j}, \dots, R_{T_{j-q}}$ ($q = 0, 1, 2$, $N = 10$, $\Gamma = 10E6$, $v_1 = 0.1$, $v_2 = 100$, $\beta = 0.6$, $d = 3$).

3.6 Conclusion

Stochastic control problems generally have no explicit solutions and are difficult to solve numerically. In this paper, we have proposed an efficient algorithm to approximate optimal allocation strategies to hedge interest rates derivatives submitted to liquidity costs.

As discussed above, our methodology is constructive and efficient in a Gaussian paradigm. We optimize the projection of the allocation strategy to the space generated by the first Hermite polynomial and use a classical stochastic algorithm to optimize the coefficients of the projection.

We have illustrated this general approach by studying swaps in the presence of liquidity costs. We have discussed the performances of the numerical method in terms of all its algorithmic components.

We emphasize that our methodology can be applied to many control problems, e.g., the computation of indifference prices, when the model under consideration belongs to a Gaussian space.

Bibliography

- [1] S. H. Babbs and K. Ben Nowman. Kalman filtering of generalized Vasicek term structure models. Journal of Financial and Quantitative Analysis, 34(1) :115–130, 1999.
- [2] H.-F. Chen. Stochastic Approximation and its Applications, volume 64 of Nonconvex Optimization and its Applications. Kluwer Academic Publishers, Dordrecht, 2002.
- [3] H. F. Chen and Y.M. Zhu. Stochastic approximation procedures with randomly varying truncations. Sci. Sinica Ser. A, 29(9) :914–926, 1986.
- [4] Q. Dai and K. J. Singleton. Analysis of affine term structure models. The Journal of Finance, 55(5) :1943–1978, 2000.
- [5] M. Duflo. Random Iterative Models, volume 34 of Applications of Mathematics (New York). Springer-Verlag, Berlin, 1997.
- [6] H. J. Kushner and G. G. Yin. Stochastic Approximation and Recursive Algorithms and Applications, volume 35 of Applications of Mathematics (New York). Springer-Verlag, New York, second edition, 2003. Stochastic Modelling and Applied Probability.
- [7] J. Lelong. Almost sure convergence for randomly truncated stochastic algorithms under verifiable conditions. Statist. Probab. Lett., 78(16) :2632–2636, 2008.
- [8] J. Lelong. Asymptotic normality of randomly truncated stochastic algorithms. ESAIM Probab. Stat., 17 :105–119, 2013.
- [9] P. Malliavin. Integration and Probability, volume 157 of Graduate Texts in Mathematics. Springer-Verlag, New York, 1995.
- [10] H. Robbins and S. Monro. A stochastic approximation method. Ann. Math. Statistics, 22 :400–407, 1951.
- [11] R. T. Rockafellar and R. J.-B. Wets. Variational analysis, volume 317 of Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]. Springer-Verlag, Berlin, 1998.

3.7 Appendix

Proof of Lemma 2

Proof. Let us first prove the result for $M = 0$, that is $X = \exp(\mu + \lambda G)$.

$$X = \sum_{n=0}^{+\infty} \alpha(n) \frac{H_n(G)}{\sqrt{n!}}$$

where

$$\alpha(n) = \mathbb{E} \left[X \frac{H_n(G)}{\sqrt{n!}} \right] = \exp(\mu) \mathbb{E} \left[\exp(\lambda G) \frac{H_n(G)}{\sqrt{n!}} \right].$$

We then use the identity : $e^{\lambda x - \lambda^2/2} = \sum_{j \geq 0} \frac{\lambda^j}{j!} H_j(x)$ and obtain

$$\alpha(n) = \exp(\mu + \lambda^2/2) \mathbb{E} \left[\sum_j \frac{\lambda^j}{j!} H_j(G) \frac{H_n(G)}{\sqrt{n!}} \right].$$

As $(H_j(G)/\sqrt{j!}, j \geq 0)$ is an orthonormal basis of $L^2(G)$ we have

$$\alpha(n) = \exp(\mu + \lambda^2/2) \frac{\lambda^n}{\sqrt{n!}}.$$

Let X^d be the projection of X on the subspace generated by the H_0, \dots, H_d :

$$\begin{aligned} X^d &= \sum_{n=0}^d \alpha(n) \frac{H_n(G)}{\sqrt{n!}} = e^{\mu + \lambda^2/2} \sum_{n=0}^d \frac{\lambda^n}{\sqrt{n!}} \frac{H_n(G)}{\sqrt{n!}} \\ X - X^d &= e^{\mu + \lambda^2/2} \sum_{n=d+1}^{\infty} \frac{\lambda^n}{\sqrt{n!}} \frac{H_n(G)}{\sqrt{n!}}. \end{aligned}$$

The truncation error is

$$\begin{aligned} \|X - X^d\|_2^2 &= e^{2\mu + \lambda^2} \sum_{n=d+1}^{\infty} \frac{\lambda^{2n}}{n!} \\ &\leq e^{2\mu + \lambda^2} \frac{\lambda^{2(d+1)}}{(d+1)!} \sum_{n=0}^{\infty} \frac{\lambda^{2n}}{n!} \\ &\leq e^{2\mu + 2\lambda^2} \frac{\lambda^{2(d+1)}}{(d+1)!}. \end{aligned}$$

The desired result thus holds true for $M = 0$. Let M be a positive integer. We have

$$\|X - X^d\|_2^2 = e^{2\mu + \lambda_0^2 + \dots + \lambda_M^2} \sum_{n_0 + \dots + n_M > d} \prod_{m=0}^M \frac{\lambda_m^{2n_m}}{(n_m)!}$$

Recall the classical identity

$$\frac{(a_0 + \cdots + a_L)^N}{N!} = \sum_{n_0 + \cdots + n_L = N} \prod_{m=0}^L \frac{a_m^{n_m}}{(n_m)!}. \quad (3.30)$$

Thus,

$$\begin{aligned} \sum_{n_0 + \cdots + n_M > d} \prod_{m=0}^M \frac{\lambda_m^{2n_m}}{(n_m)!} &= \sum_{N=d+1}^{\infty} \sum_{n_0 + \cdots + n_M = N} \prod_{m=0}^M \frac{\lambda_m^{2n_m}}{(n_m)!} \\ &= \sum_{N=d+1}^{\infty} \frac{(\lambda_0^2 + \cdots + \lambda_M^2)^N}{N!} \\ &\leq \frac{(\lambda_0^2 + \cdots + \lambda_M^2)^{d+1}}{(d+1)!} \exp(\lambda_0^2 + \cdots + \lambda_M^2). \end{aligned}$$

This ends the proof for all positive M . □

Chapitre 4

Méthode de simulation trajectorielle exacte

Résumé

Dans ce travail réalisé en commun avec Etienne Tanré, nous présentons des extensions de l'algorithme de simulation trajectorielle exacte introduit par Beskos et al. dans [BPR06]. Tout d'abord, une modification de l'ordre dans lequel est faite la simulation accélère l'algorithme. De plus, nous proposons une méthode approchée de l'algorithme modifié. Nous obtenons un contrôle de l'erreur dans cette dernière version, exponentiellement petit en fonction du paramètre de troncature. Les temps de calculs sont linéaires en ce paramètre. Ensuite nous étendons l'algorithme pour des termes de dérive plus généraux. Notre résultat principal consiste en un algorithme sans biais pour approcher les deux premières dérivées en fonction de la condition initiale x des quantités de la forme $\mathbb{E}\Psi(X_T^x)$. We describe it in details in dimension 1 and also discuss its multi-dimensional extensions for the evaluation of On décrit en détail la dimension une et discutons du cas multi-dimensionnel pour l'évaluation de $\mathbb{E}\Psi(X_T^x)$. Enfin, nous appliquons l'algorithme au processus CIR et effectuons des tests numériques pour le comparer aux approximations classiques.

Ce travail a été débuté dans le cadre de la collaboration entre l'équipe-projet TOSCA d'INRIA et l'équipe de recherche et développement dérivé de Taux et Hybrides du Credit Agricole CIB. Un pre-print se trouve disponible ici [RT16].

Abstract

In this paper, we present extensions of the exact simulation algorithm introduced by Beskos et al. [BPR06]. First, a modification in the order in which the simulation is done accelerates the algorithm. In addition, we propose a truncated version of the modified algorithm. We obtain a control of the bias of this last version, exponentially small in function of the truncation parameter. Then, we extend it to more general drift functions. Our main result is an unbiased

algorithm to approximate the two first derivatives with respect to the initial condition x of quantities with the form $\mathbb{E}\Psi(X_T^x)$. We describe it in details in dimension 1 and also discuss its multi-dimensional extensions for the evaluation of $\mathbb{E}\Psi(X_T^x)$. Finally, we apply the algorithm to the CIR process and perform numerical tests to compare it with classical approximation procedures.

This work was mainly done during a scientific collaboration with TOSCA Team of INRIA and Interest Rates and Hybrids Quantitative Research Team. A preprint version is available here [RT16].

Keywords : Unbiased Monte Carlo methods; Monte Carlo Approximation of Derivatives; Exact Simulation of SDE. **AMS2010 class :** 65C05, 60J60.

4.1 Introduction

In this paper, we are interested in the approximation of the law of a one dimensional stochastic process $(X_t^x, t \geq 0)$, defined as the unique solution of a Stochastic Differential Equation (SDE)

$$X_T^x = x + \int_0^T \alpha(X_t^x) dt + \int_0^T \sigma(X_t^x) dW_t, \quad (4.1)$$

with smooth coefficients α and σ . Let Ψ be a measurable function. The quantities we aim to evaluate take form

$$P_\Psi(x) := \mathbb{E}\Psi(X_T^x). \quad (4.2)$$

We also evaluate their sensitivities to the parameters of the model. We are especially interested in the dependance on the initial condition x ,

$$\Delta_\Psi(x) := \frac{d}{dx} \mathbb{E}\Psi(X_T^x) \quad (4.3)$$

$$\Gamma_\Psi(x) := \frac{d^2}{dx^2} \mathbb{E}\Psi(X_T^x). \quad (4.4)$$

These two derivatives are known as Delta and Gamma in the context of financial mathematics.

The most simple method to approximate (4.2) consists in a time discretization (say with step δ) of (4.1) with an Euler scheme. For an approximation of (4.3) or (4.4), one should evaluate (4.2) with two or three values, say for $x - dx$, x and $x + dx$. Then, we use a finite difference approximation of the derivatives. This method is very simple to implement, but we have three sources of error :

- 1) two biases due to
 - a - the time discretization ;
 - b - the finite difference approximation ;
- 2) the statistical error.

In [4] and [3], the authors proposed an exact simulation algorithm for one dimensional SDE with constant diffusion coefficient (see Section 4.2.1). This method removes the bias of type *a* in the approximation of $P_\Psi(x)$. More recently, several authors have worked on algorithms without bias of type *a*. For instance, in Bally and Kohatsu-Higa [2], a theoretical stochastic representation of the parametrix method is developed and used successfully to reach this goal. Similar ideas are developed in [8] to evaluate P_Ψ for smooth functions Ψ and diffusion process in \mathbb{R}^d . Gobet and M'rad [7] have proposed a multilevel Monte Carlo method with random number of levels. They succeed to avoid bias for Lipschitz continuous function Ψ .

Otherwise, in [6] the Malliavin calculus theory is developed to obtain expression of the derivatives $\Delta_\Psi(x)$ and $\Gamma_\Psi(x)$ without bias of type b-

$$\frac{d}{dx} \mathbb{E}[\Psi(X_T^x)] = \mathbb{E}[\Psi(X_T^x) H_T],$$

where H_T is an explicit random weight.

In this paper, we extend Beskos et al. method of simulation : we simulate the Poisson process by ordering the points in increasing ordinate (see Sec. 4.2.4). With this modification, the rejection of Brownian bridge trajectories are decided faster and the efficiency of the algorithm is higher. Moreover, one should relax a little bit the assumption on the drift coefficient α . Our algorithm is efficient to compute P_Ψ even for full path dependent function Ψ . Thereby, even if it applies essentially for one dimensional diffusion processes, our work, in this setting, is more general than [8, 7].

Furthermore, we propose an unbiased algorithm to compute the derivatives (4.3) and (4.4). The idea combines Fournié et al. [6] formula and some generalization of Beskos et al. [3] rejection procedure.

The paper is organized as follows. We describe the algorithm in a general context in Section 4.2. Section 4.3 is devoted to a detailed presentation for the CIR model. We compare the efficiency of our algorithm with classical estimators in Section 4.4.

1

4.2 Unbiased Estimators

4.2.1 Beskos, Papaspiliopoulos and Roberts unbiased estimator

Here, we recall the main ideas developed in [4, 3] to exactly simulate the solution of one dimensional stochastic differential equations. Assume that the process X^x solves the equation

$$X_T^x = x + \int_0^T \alpha(X_t^x) dt + W_T \quad (4.5)$$

(i.e. $\sigma \equiv 1$ in (4.1)). The main idea is a smart use of Girsanov Theorem :

$$\mathbb{E} \left[\Psi(X_T^x) \right] = \mathbb{E} \left[\Psi(B_T^x) \exp \left(\int_0^T \alpha(B_t^x) dB_t^x - \int_0^T \frac{\alpha^2(B_t^x)}{2} dt \right) \right]$$

where $(B_t^x)_{t \geq 0}$ is a one dimensional Brownian motion with $B_0^x = x$. The dimension allows one to transform the stochastic integral :

$$\int_0^T \alpha(B_t^x) dB_t^x = A(B_T^x) - A(B_0^x) - \int_0^T \frac{\alpha'(B_t^x)}{2} dt, \quad (4.6)$$

where $A(x) = \int_0^x \alpha(y) dy$. Then, one obtains

$$\mathbb{E} \left[\Psi(X_T^x) \right] = \mathbb{E} \left[\Psi(B_T^x) \exp \left(A(B_T^x) - A(B_0^x) - \int_0^T \frac{(\alpha^2 + \alpha')}{2} (B_t^x) dt \right) \right]. \quad (4.7)$$

Next, we replace in (4.7) the Brownian motion $(B_t^x, 0 \leq t \leq T)$ by a Brownian bridge $(\tilde{B}_t^x, 0 \leq t \leq T)$, where the final value \tilde{B}_T^x has the distribution

$$\mathbb{P}(\tilde{B}_T^x \in d\theta) = C \exp\left(-\frac{(\theta - x)^2}{2T} + \int_0^\theta \alpha(y) dy\right) d\theta. \quad (4.8)$$

Then, denote

$$\varphi(y) = \frac{\alpha^2(y) + \alpha'(y)}{2}, \quad (4.9)$$

one has

$$\mathbb{E}[\Psi(X_T^x)] = \tilde{C} \mathbb{E}\left[\Psi(\tilde{B}_T^x) \exp\left(-\int_0^T \varphi(\tilde{B}_t^x) dt\right)\right]. \quad (4.10)$$

If we moreover assume that φ takes value in a compact set, say $0 \leq \varphi(y) \leq K$, one can exactly simulate the diffusion X^x with a rejection procedure. Namely, one simulates a path of the Brownian bridge \tilde{B}^x and accept it with probability $\exp\left(-\int_0^T \varphi(\tilde{B}_t^x) dt\right)$. To do it, one simulates a Poisson process (independent of \tilde{B}^x) of unit intensity on $[0, T] \times [0, K]$ and accepts the Brownian bridge path if and only if there is no point of the Poisson process in the hypograph $D(\omega)$ of $\varphi(\tilde{B}_t^x)$

$$D(\omega) = \left\{ (t, y) \in [0, T] \times [0, K], y \leq \varphi(\tilde{B}_t^x) \right\}. \quad (4.11)$$

It is easy to verify that the probability to accept the path is $\exp\left(-\int_0^T \varphi(\tilde{B}_t^x) dt\right)$. Furthermore, we only need to know the value of the Brownian bridge at a finite number of times $0 < t_1 < \dots < t_n \leq T$, the abscissas of the points of the Poisson process. So, we have

$$\mathbb{E}[\Psi(X_T^x)] = \mathbb{E}\left[\Psi(\tilde{B}_T^x) \Big| N \cap D(\omega) = \emptyset\right], \quad (4.12)$$

where N is a Poisson process with unit intensity on $[0, T] \times [0, K]$, independent of $(\tilde{B}_t^x, 0 \leq t \leq T)$.

Remark 2. We have written this short presentation under the assumption $0 \leq \varphi \leq K$. It should be easily generalized to the cases where :

1. φ is bounded, but not necessary nonnegative. In this case, we only have to replace in (4.10) the function φ by $\varphi - \inf_{\mathbb{R}} \varphi$ and the constant \tilde{C} by $\tilde{C} \exp(-T \inf_{\mathbb{R}} \varphi)$.
2. φ has no finite global upper bound, but has an upper bound in $+\infty$ or $-\infty$. For instance, $\limsup_{y \rightarrow -\infty} \varphi(y) = +\infty$ and $\limsup_{y \rightarrow +\infty} \varphi(y) < \infty$. Here, we only have to first simulate the infimum $m(\omega)$ of \tilde{B}^x on $[0, T]$ and the time $t_m(\omega)$ at which it is reached. Then, we simulate a Poisson process on $[0, T] \times [0, \tilde{K}(\omega)]$ with $\tilde{K}(\omega) = \sup_{y \geq m(\omega)} \varphi(y)$ (see [3]).

Williams decomposition of Brownian paths [12] gives the conditional law $(\tilde{B}_t^x, 0 \leq t \leq T | m(\omega), t_m(\omega))$: conditionally to m and t_m , the processes $(\tilde{B}_{t_m+t}^x - m, 0 \leq t \leq T - t_m)$ and $(\tilde{B}_{t_m-t}^x - m, 0 \leq t \leq t_m)$ are two independent Bessel bridges processes of dimension 3. Such a process is simple to exactly simulate at a finite number of times.

4.2.2 Unbiased estimator of the first derivative (Delta)

In this section, we present our main results. We generalise the unbiased algorithm introduced by Beskos et al. [3] to approximate the sensitivities $\frac{d}{dx}\mathbb{E}^x[\Psi(X_T)]$ with an unbiased estimator.

Proposition 12. Let $(X_t^x, 0 \leq t \leq T)$ be the solution of (4.5), starting from x , and Ψ a measurable function. Assume that $\forall y \in \mathbb{R}$, we have $-\hat{K} \leq \alpha'(y) \leq 0$ and $0 \leq \alpha^2(y) + \alpha'(y) \leq 2K$. Then, an unbiased Monte Carlo procedure to evaluate $\frac{d}{dx}\mathbb{E}\Psi(X_T^x)$ is available

$$\begin{aligned} \frac{d}{dx}\mathbb{E}\Psi(X_T^x) &= -\mathbb{E}\left[\frac{x\Psi(\tilde{B}_T^x)}{T}\Big|N \cap D = \emptyset\right] \\ &\quad + \mathbb{E}\left[\frac{\Psi(\tilde{B}_T^x)}{T}\left(\tilde{B}_T^x - T\alpha(\tilde{B}_{U_2T}^x)\right)\mathbb{1}_{\hat{N} \cap \hat{D} = \emptyset}\Big|N \cap D = \emptyset\right] \\ &\quad - \mathbb{E}\left[\Psi(\tilde{B}_T^x)\left(\tilde{B}_{U_1T}^x - U_1T\alpha(\tilde{B}_{U_1U_2T}^x)\right)\alpha'(\tilde{B}_{U_1T}^x)\mathbb{1}_{\hat{N} \cap \hat{D}^1 = \emptyset}\Big|N \cap D = \emptyset\right], \end{aligned}$$

where :

- $\varphi = (\alpha^2 + \alpha')/2$;
- $(\tilde{B}_t^x, 0 \leq t \leq T)$ is a Brownian bridge with \tilde{B}_T^x given by (4.8);
- D is the hypograph of $\varphi(\tilde{B}_t^x)$ (see (4.11));
- \hat{D} is the hypograph of $-\alpha'(\tilde{B}_t^x)$ and $\hat{D}_1 = \hat{D} \cap ([0, U_1T] \times \mathbb{R}_+)$;
- N and \hat{N} are two independent Poisson processes with unit intensity on $[0, T] \times [0, K]$ and $[0, T] \times [0, \hat{K}]$, (and independent of \tilde{B}^x);
- U_1 and U_2 are two independent random variables with uniform distribution on $[0, 1]$ (and independent of \tilde{B}^x , N and \hat{N}).

We first recall basic results on Malliavin calculus (see Fournié et al. [6]) useful to detail our algorithm. The process $(X_t^x, t \geq 0)$ is the unique solution of (4.5) with $X_0^x = x$. We denote by $(Y_t^x, t \geq 0)$ the associated first variation process

$$Y_t^x := \frac{d}{dx}X_t^x.$$

It solves the linear SDE

$$\begin{aligned} dY_t^x &= Y_t^x \alpha'(X_t^x) dt \\ Y_0^x &= 1. \end{aligned}$$

The solution is

$$Y_t^x = \exp\left(\int_0^t \alpha'(X_s^x) ds\right). \quad (4.13)$$

Furthermore, it is known that the Malliavin derivative $D_t X_T^x$ satisfies

$$\begin{aligned} dD_t X_s^x &= D_t X_s^x \alpha'(X_s^x) ds \\ D_t X_t^x &= 1. \end{aligned} \quad (4.14)$$

We deduce that Y^x and $D_t X^x$ are linked by the identity

$$D_t X_T^x = \frac{Y_T^x}{Y_t^x}. \quad (4.15)$$

We deduce

$$Y_T^x = Y_t^x D_t X_T^x = \int_0^T a(t) Y_t^x D_t X_T^x dt,$$

where a is any L^2 function such that $\int_0^T a(t) dt = 1$. For instance, we use in this paper $a(t) \equiv \frac{1}{T}$.

Following Fournié et al. [6], and using classical results on Malliavin calculus (integration by parts formula, see [11]), we obtain

$$\begin{aligned} \frac{d}{dx} \mathbb{E} \Psi(X_T^x) &= \mathbb{E} [\Psi'(X_T^x) Y_T^x] \\ &= \frac{1}{T} \mathbb{E} \left[\int_0^T \Psi'(X_T^x) D_t(X_T^x) Y_t^x dt \right] = \frac{1}{T} \mathbb{E} \left[\int_0^T D_t(\Psi(X_T^x)) Y_t^x dt \right] \\ &= \frac{1}{T} \mathbb{E} [\Psi(X_T^x) \delta(Y_t^x)] \\ &= \frac{1}{T} \mathbb{E} \left[\Psi(X_T^x) \int_0^T Y_t^x dW_t \right]. \end{aligned} \quad (4.16)$$

Remark 3. Even if we have used the notation Ψ' in the sketch of the proof of the result, the relation remains true if Ψ is not a smooth function.

After this short remind on Malliavin calculus theory, we now prove Proposition 12.

Proof of Proposition 12. We use the one dimension setting to remove the stochastic integral in (4.16)

$$\begin{aligned} \int_0^T Y_t^x dW_t &= W_T Y_T^x - W_0 Y_0^x - \int_0^T W_t dY_t^x \\ &= W_T Y_T^x - W_0 Y_0^x - \int_0^T W_t Y_t^x \alpha'(X_t^x) dt. \end{aligned} \quad (4.17)$$

The evaluation of the integral in the last term would introduce a bias. To avoid it, one uses a classical identity. Namely, consider a stochastic process $(\gamma_t, 0 \leq t \leq T)$, we have

$$\int_0^T \gamma_t dt = T \bar{\mathbb{E}}(\gamma_{UT}) \quad (4.18)$$

where U is random variable with uniform distribution on $[0, 1]$, independent of γ and $\bar{\mathbb{E}}$ denotes the expectation with respect to U . The drawback of the last

expression is the increase of the variance. See [10] for a discussion on this topic. Using this property and (4.13), we obtain

$$\frac{d}{dx} \mathbb{E} \Psi(X_T^x) = \mathbb{E} \left[\frac{\Psi(X_T^x)}{T} \left(W_T \exp \int_0^T \alpha'(X_s^x) ds - W_0 - TW_{U_1 T} \alpha'(X_{U_1 T}^x) \exp \int_0^{U_1 T} \alpha'(X_s^x) ds \right) \right],$$

where U_1 is a random variable independent of $(X_t^x, t \in [0, T])$ with uniform law on $[0, 1]$. As in Section 4.2.1, we finally apply Girsanov Theorem

$$\begin{aligned} \frac{d}{dx} \mathbb{E} \Psi(X_T^x) &= \tilde{C} \mathbb{E} \left[\frac{\Psi(\tilde{B}_T^x)}{T} \exp \left(-\frac{1}{2} \int_0^T \alpha^2(\tilde{B}_s^x) + \alpha'(\tilde{B}_s^x) ds \right) \right. \\ &\quad \times \left(\left(\tilde{B}_T^x - \int_0^T \alpha(\tilde{B}_t^x) dt \right) \exp \int_0^T \alpha'(\tilde{B}_s^x) ds - x \right. \\ &\quad \left. \left. - T \left(\tilde{B}_{U_1 T}^x - \int_0^{U_1 T} \alpha(\tilde{B}_s^x) ds \right) \alpha'(\tilde{B}_{U_1 T}^x) \exp \int_0^{U_1 T} \alpha'(\tilde{B}_s^x) ds \right) \right], \end{aligned}$$

where $(\tilde{B}_t^x, 0 \leq t \leq T)$ is a Brownian bridge with final distribution given by (4.8) and $U_1 \sim \mathcal{U}(0, 1)$ is independent of \tilde{B}^x . We use the same rejection procedure as in Section 4.2.1 to obtain

$$\begin{aligned} \frac{d}{dx} \mathbb{E} \Psi(X_T^x) &= \mathbb{E} \left[\frac{\Psi(\tilde{B}_T^x)}{T} \times \left(\left(\tilde{B}_T^x - \int_0^T \alpha(\tilde{B}_t^x) dt \right) \exp \int_0^T \alpha'(\tilde{B}_s^x) ds - x \right. \right. \\ &\quad \left. \left. - T \left(\tilde{B}_{U_1 T}^x - \int_0^{U_1 T} \alpha(\tilde{B}_s^x) ds \right) \alpha'(\tilde{B}_{U_1 T}^x) \exp \int_0^{U_1 T} \alpha'(\tilde{B}_s^x) ds \right) \middle| N \cap D(\omega) = \emptyset \right]. \end{aligned}$$

$$\begin{aligned} \frac{d}{dx} \mathbb{E} \Psi(X_T^x) &= -\mathbb{E} \left[\frac{x \Psi(\tilde{B}_T^x)}{T} \middle| N \cap D(\omega) = \emptyset \right] \\ &\quad + \mathbb{E} \left[\frac{\Psi(\tilde{B}_T^x)}{T} \left(\tilde{B}_T^x - T \alpha(\tilde{B}_{U_2 T}^x) \right) \exp \left(\int_0^T \alpha'(\tilde{B}_s^x) ds \right) \middle| N \cap D(\omega) = \emptyset \right] \\ &\quad - \mathbb{E} \left[\Psi(\tilde{B}_T^x) \left(\tilde{B}_{U_1 T}^x - U_1 T \alpha(\tilde{B}_{U_1 U_2 T}^x) \right) \alpha'(\tilde{B}_{U_1 T}^x) \right. \\ &\quad \left. \exp \left(\int_0^{U_1 T} \alpha'(\tilde{B}_s^x) ds \right) \middle| N \cap D(\omega) = \emptyset \right], \end{aligned}$$

where U_2 is a random variable with uniform distribution on $[0, 1]$, independent of \tilde{B}^x , N and U_1 .

It remains to remark that one again interprets the term $\exp\left(\int_0^T \alpha'(\tilde{B}_s^x) ds\right)$ as the probability for a Poisson process to have no point in a domain. More precisely, we consider a Poisson process \hat{N} with unit intensity on $[0, T] \times [0, \hat{K}]$ (where $\hat{K} = -\inf_{\mathbb{R}} \alpha'$), independent of \tilde{B}^x , N , U_1 and U_2 . We denote by \hat{D} the hypograph of $-\alpha'$

$$\hat{D}(\omega) = \left\{ (t, y) \in [0, T] \times \mathbb{R}^+, y \leq -\alpha'(\tilde{B}_t^x) \right\}$$

and by \hat{D}^1 its restriction to $[0, U_1 T] \times \mathbb{R}^+$. We finally have the unbiased estimator

$$\begin{aligned} \frac{d}{dx} \mathbb{E} \Psi(X_T^x) &= -\mathbb{E} \left[\frac{x \Psi(\tilde{B}_T^x)}{T} \middle| N \cap D(\omega) = \emptyset \right] \\ &\quad + \mathbb{E} \left[\frac{\Psi(\tilde{B}_T^x)}{T} \left(\tilde{B}_T^x - T \alpha(\tilde{B}_{U_2 T}^x) \right) \mathbb{1}_{\hat{N} \cap \hat{D} = \emptyset} \middle| N \cap D(\omega) = \emptyset \right] \\ &\quad - \mathbb{E} \left[\Psi(\tilde{B}_T^x) \left(\tilde{B}_{U_1 T}^x - U_1 T \alpha(\tilde{B}_{U_1 U_2 T}^x) \right) \alpha'(\tilde{B}_{U_1 T}^x) \mathbb{1}_{\hat{N} \cap \hat{D}^1 = \emptyset} \middle| N \cap D(\omega) = \emptyset \right]. \end{aligned} \tag{4.19}$$

□

Remark 4. Similarly to Remark 2, we can generalise the previous estimator to function φ with a global lower bound and an upper bound only in one side.

Furthermore, the same extension should be obtained if $-\alpha'$ has a global lower bound. In this case, we replace $-\alpha'$ by $-\alpha' + \sup_{\mathbb{R}}(\alpha')$ in the definition of \hat{K} , \hat{D} and \hat{D}^1 . We also replace $\mathbb{1}_{\hat{N} \cap \hat{D} = \emptyset}$ by $\exp(T \sup_{\mathbb{R}}(\alpha')) \mathbb{1}_{\hat{N} \cap \hat{D} = \emptyset}$ and $\mathbb{1}_{\hat{N} \cap \hat{D}^1 = \emptyset}$ by $\exp(U_1 T \sup_{\mathbb{R}}(\alpha')) \mathbb{1}_{\hat{N} \cap \hat{D}^1 = \emptyset}$.

Our unbiased estimator can be extended if $-\alpha'$ has only a local upper bound in the same side as φ (i.e. $\limsup_{y \rightarrow +\infty} \varphi(y)$ and $\limsup_{y \rightarrow +\infty} (-\alpha'(y))$ are both finite or $\limsup_{y \rightarrow -\infty} \varphi(y)$ and $\limsup_{y \rightarrow -\infty} (-\alpha'(y))$ are both finite).

4.2.3 Unbiased estimator of the second derivative (Gamma)

In this part, we detail an unbiased estimator of the second derivative $\frac{d^2}{dx^2} \mathbb{E} \Psi(X_T^x)$. We denote by Z_t^x the second variation process associated to X_t^x

$$Z_t^x = \frac{d^2}{dx^2} X_t^x.$$

It satisfies the linear stochastic differential equation

$$Z_T^x = \int_0^T \alpha''(X_s^x) (Y_s^x)^2 + \alpha'(X_s^x) Z_s^x ds.$$

The solution is

$$Z_T^x = Y_T^x \int_0^T \alpha''(X_s^x) Y_s^x ds. \tag{4.20}$$

We also need the Malliavin derivative of the first variation process Y^x . It satisfies

$$D_t Y_T^x = \int_0^T \alpha''(X_s^x) Y_s^x D_t X_s^x + \alpha'(X_s^x) D_t Y_s^x ds$$

The solution is

$$D_t Y_T^x = \frac{Y_T^x}{Y_t^x} \int_t^T \alpha''(X_s^x) Y_s^x ds. \quad (4.21)$$

As in the previous section, we present the computation under the assumption that Ψ is smooth. However, the final result remains true even if Ψ is only assumed to be measurable and bounded (see [6] for more details). Using (4.16), we formally derive with respect to x and obtain

$$\frac{d^2}{dx^2} \mathbb{E} \Psi(X_T^x) = \underbrace{\mathbb{E} \left[\frac{\Psi'(X_T^x)}{T} Y_T^x \int_0^T Y_t^x dW_t \right]}_{\Gamma_1(x)} + \underbrace{\mathbb{E} \left[\frac{\Psi(X_T^x)}{T} \int_0^T Z_t^x dW_t \right]}_{\Gamma_2(x)}.$$

The main steps to obtain a tractable unbiased expression of Γ_2 are identical to the ideas used in Section 4.2.2. We use the one dimensional setting to remove the stochastic integral and (4.21) to obtain

$$\begin{aligned} \Gamma_2(x) = \mathbb{E} \left[\frac{\Psi(X_T^x)}{T} \left(W_T Y_T^x \int_0^T \alpha''(X_t^x) Y_t^x dt - \int_0^T W_t (Y_t^x)^2 \alpha''(X_t^x) dt \right. \right. \\ \left. \left. - \int_0^T \int_0^t W_t \alpha'(X_t^x) \alpha''(X_u^x) Y_u^x Y_t^x du dt \right) \right] \quad (4.22) \end{aligned}$$

To simplify Γ_1 , we apply the Malliavin integration by part formula and (4.15)

$$\begin{aligned} \Gamma_1(x) &= \frac{1}{T^2} \mathbb{E} \left[\int_0^T D_t (\Psi(X_T^x)) Y_t^x \int_0^T Y_s^x dW_s dt \right] \\ &= \frac{1}{T^2} \mathbb{E} \left[\Psi(X_T^x) \delta \left(Y_t^x \int_0^T Y_s^x dW_s \right) \right]. \end{aligned}$$

Finally, we have to make explicit the divergence operator. We apply [11, Prop. 1.3.3] to obtain

$$\begin{aligned} \delta \left(Y_t^x \int_0^T Y_s^x dW_s \right) &= \int_0^T Y_s^x dW_s \delta(Y_t^x) - \int_0^T D_t \left(\int_0^T Y_s^x dW_s \right) Y_t^x dt \\ \text{and } D_t \left(\int_0^T Y_s^x dW_s \right) &= Y_t^x + \int_t^T D_t Y_s^x dW_s. \end{aligned}$$

We again simplify the stochastic integral

$$\begin{aligned} \int_t^T D_t Y_s^x dW_s &= (D_t Y_T^x) W_T - \int_t^T W_s \alpha''(X_s^x) Y_s^x \frac{Y_s^x}{Y_t^x} ds \\ &\quad - \int_t^T W_s \frac{\alpha'(X_s^x) Y_s^x}{Y_t^x} \int_t^s \alpha''(X_u^x) Y_u^x duds. \end{aligned}$$

Finally, denoting U_1 , U_2 and U_3 three uniform independent random variables, independent of W and using (4.18), we obtain

$$\begin{aligned} \frac{d^2}{dx^2} \mathbb{E} \Psi(X_T^x) &= \mathbb{E} \left[\Psi(X_T^x) \left(\frac{x^2}{T^2} - \frac{2x}{T^2} W_T Y_T^x + \frac{1}{T^2} (W_T)^2 (Y_T^x)^2 \right. \right. \\ &\quad \left. \left. + \frac{2x}{T} W_{U_1 T} \alpha'(X_{U_1 T}^x) Y_{U_1 T}^x \right. \right. \\ &\quad \left. \left. - \frac{1}{T} (Y_{U_1 T}^x)^2 + (U_1 - 1) W_{U_1 T} \alpha''(X_{U_1 T}^x) (Y_{U_1 T}^x)^2 \right. \right. \\ &\quad \left. \left. + W_{U_1 T} \alpha'(X_{U_1 T}^x) W_{U_2 T} \alpha'(X_{U_2 T}^x) Y_{U_1 T}^x Y_{U_2 T}^x \right. \right. \\ &\quad \left. \left. - \frac{2W_T}{T} W_{U_1 T} \alpha'(X_{U_1 T}^x) Y_T^x Y_{U_1 T}^x + W^x(T) (1 - U_1) \alpha''(X_{U_1 T}^x) Y_T^x Y_{U_1 T}^x \right. \right. \\ &\quad \left. \left. + U_1 T (U_1 U_2 - 1) W_{U_1 T} \alpha'(X_{U_1 T}^x) \alpha''(X_{U_1 U_2 T}^x) Y_{U_1 U_2 T}^x Y_{U_1 T}^x \right) \right]. \end{aligned}$$

Similarly to Sections 4.2.1 and 4.2.2, we apply Girsanov theorem and (4.13). We change in the previous expression

$$\begin{aligned} X_s^x &\rightarrow \tilde{B}_s^x \\ W_s &\rightarrow \tilde{B}_s^x - s\alpha(\tilde{B}_{U_s}^x). \end{aligned}$$

$$\begin{aligned}
\frac{d^2}{dx^2} \mathbb{E} \Psi(X_T^x) &= \mathbb{E} \left[\Psi(\tilde{B}_T^x) \left(\frac{x^2}{T^2} - \frac{2x}{T^2} (\tilde{B}_T^x - T\alpha(\tilde{B}_{U_3T}^x)) \right) \exp \left(\int_0^T \alpha'(\tilde{B}_\theta^x) d\theta \right) \right. \\
&+ \frac{1}{T^2} (\tilde{B}_T^x - T\alpha(\tilde{B}_{U_3T}^x)) (\tilde{B}_T^x - T\alpha(\tilde{B}_{U_4T}^x)) \exp \left(\int_0^T 2\alpha'(\tilde{B}_\theta^x) d\theta \right) \\
&+ \frac{2x}{T} (\tilde{B}_{U_1T}^x - U_1T\alpha(\tilde{B}_{U_1U_3T}^x)) \alpha'(\tilde{B}_{U_1T}^x) \exp \left(\int_0^{U_1T} \alpha'(\tilde{B}_\theta^x) d\theta \right) \\
&\quad - \frac{1}{T} \exp \left(\int_0^{U_1T} 2\alpha'(\tilde{B}_\theta^x) d\theta \right) \\
&+ (U_1 - 1) (\tilde{B}_{U_1T}^x - U_1T\alpha(\tilde{B}_{U_1U_3T}^x)) \alpha''(\tilde{B}_{U_1T}^x) \exp \left(\int_0^{U_1T} 2\alpha'(\tilde{B}_\theta^x) d\theta \right) \\
&+ (\tilde{B}_{U_1T}^x - U_1T\alpha(\tilde{B}_{U_1U_3T}^x)) \alpha'(\tilde{B}_{U_1T}^x) (\tilde{B}_{U_2T}^x - U_2T\alpha(\tilde{B}_{U_2U_4T}^x)) \alpha'(\tilde{B}_{U_2T}^x) \\
&\quad \times \exp \left(\int_0^{U_1T} \alpha'(\tilde{B}_\theta^x) d\theta \right) \exp \left(\int_0^{U_2T} \alpha'(\tilde{B}_\theta^x) d\theta \right) \\
&- \frac{2}{T} (\tilde{B}_T^x - T\alpha(\tilde{B}_{U_3T}^x)) (\tilde{B}_{U_1T}^x - U_1T\alpha(\tilde{B}_{U_1U_4T}^x)) \alpha'(\tilde{B}_{U_1T}^x) \\
&\quad \times \exp \left(\int_0^T \alpha'(\tilde{B}_\theta^x) d\theta \right) \exp \left(\int_0^{U_1T} \alpha'(\tilde{B}_\theta^x) d\theta \right) \\
&\quad + (\tilde{B}_T^x - T\alpha(\tilde{B}_{U_3T}^x)) (1 - U_1) \alpha''(\tilde{B}_{U_1T}^x) \\
&\quad \times \exp \left(\int_0^T \alpha'(\tilde{B}_\theta^x) d\theta \right) \exp \left(\int_0^{U_1T} \alpha'(\tilde{B}_\theta^x) d\theta \right) \\
&+ U_1T (U_1U_2 - 1) (\tilde{B}_{U_1T}^x - U_1T\alpha(\tilde{B}_{U_1U_3T}^x)) \alpha'(\tilde{B}_{U_1T}^x) \alpha''(\tilde{B}_{U_1U_2T}^x) \\
&\quad \left. \exp \left(\int_0^{U_1U_2T} \alpha'(\tilde{B}_\theta^x) d\theta \right) \exp \left(\int_0^{U_2T} \alpha'(\tilde{B}_\theta^x) d\theta \right) \right] \Big|_{N \cap D = \emptyset}.
\end{aligned}$$

To conclude, each term on the form $\exp(-\int_0^s \beta(\tilde{B}_\theta^x) d\theta)$ is replaced by $\mathbb{E} \mathbb{1}_{\{N^j \cap D^j = \emptyset\}}$ for appropriate Poisson processes N^j and hypograph D^j (similar terms are expressed in details p. 107).

4.2.4 Simulation of the Poisson Process

We have recalled in Section 4.2.1 the details of the algorithm developed in [3] to simulate exact paths of the solution of (4.5). The main point is the following. Consider a function φ with values in $[0, K]$, $\exp(-\int_0^T \varphi(\tilde{B}_\theta^x) d\theta)$ is the probability that $N \cap D = \emptyset$, where N is a Poisson process with unit intensity

on $[0, T] \times [0, K]$ independent of \tilde{B}^x . The hypograph D of $\varphi(\tilde{B}_\theta^x)$ is defined by (4.11).

For the rejection procedure, we simulate the Poisson process $(t_1, y_1), \dots, (t_{n(\omega)}, y_{n(\omega)})$ and the Brownian bridge at the times $t_1, \dots, t_{n(\omega)}$. If there exists $j \in [1, n(\omega)]$ such that $y_j < \varphi(\tilde{B}_{t_j}^x)$, the Brownian bridge path is rejected.

In [3], the Poisson process is generated on $[0, T] \times [0, K]$. The result is $((t_1, y_1), \dots, (t_{n(\omega)}, y_{n(\omega)}))$. Then, the authors simulate the Brownian bridge at time $t_1, t_2, \dots, t_{n(\omega)}$ and evaluate the cardinal number of $N \cap D$. In the present paper, we propose two variants of the algorithm. For both variants, immediately after the simulation of one point (t_j, y_j) , we simulate $\tilde{B}_{t_j}^x$. If $y_j < \varphi(\tilde{B}_{t_j}^x)$, we have to reject the Brownian bridge path. So, we do not need to simulate the full Poisson process N and stop immediately the algorithm. There is two simple variants for the simulation of the Poisson process : first, by increasing times ($t_1 < t_2 < \dots < t_{n(\omega)}$). Second, by increasing ordinates ($y_1 < y_2 < \dots < y_{n(\omega)}$). This last variant aims to reject as fast as possible the Brownian bridge trajectory. Roughly speaking, smaller is the ordinate, higher is the probability to be below $\varphi(\tilde{B}^x)$. We numerically compare the efficiency of the both variants in Section 4.4.1.2.

4.2.5 A truncated algorithm

The increasing ordinates variant should start, even if we do not know an explicit upper bound K to $t \mapsto \varphi(\tilde{B}_t^x)$. We propose to extend the Beskos et al. algorithm to SDE with drift α , such that $\limsup_{y \rightarrow -\infty} \varphi(y) = \limsup_{y \rightarrow +\infty} \varphi(y) = \infty$. According to [3], $\varphi = (\alpha^2 + \alpha')/2$.

For any $L > 0$, we denote N^L a Poisson process with unit intensity on $[0, T] \times [0, L]$. Our truncated algorithm is stopped and we accept a path of the Brownian bridge if $N^{\tilde{K}(\omega)} \cap D = \emptyset$, where $\tilde{K}(\omega) \leq \sup_{\theta \in [0, T]} \{\varphi(\tilde{B}_\theta^x)\}$. Larger is \tilde{K} , smaller is the probability to wrongly accept a path, but slower is the algorithm. A reasonable choice of $\tilde{K}(\omega)$ is

$$\tilde{K}(\omega) \geq \max\{K, \varphi(\tilde{B}_T^x), \varphi(\inf_{0 \leq s \leq T} \tilde{B}_s^x)\},$$

where K is an a priori threshold. Our algorithm is no more unbiased. However, Proposition 13 gives an upper bound of the error in the approximation of (4.2).

4.2.6 Theoretical control of the error

Proposition 13. Let X_T^x solution of (4.5) and $X_T^{x,K}$ its approximation obtained by the truncated rejection procedure presented in Section 4.2.5. Precisely, the Brownian bridge path is accepted if there is no point of an independent Poisson process on $[0, T] \times [0, K]$ in the hypograph D of φ (given by (4.11)). Then :

a.

$$\begin{aligned}
& \left| \mathbb{E}\Psi(X_T^x) - \mathbb{E}\Psi(X_T^{x,K}) \right| \\
& \leq \sqrt{\mathbb{E}\left[\Psi^2(\tilde{B}_T^x)\right]} \left(\frac{\mathbb{P}\left(\sup_{0 \leq \theta \leq T} \varphi(\tilde{B}_\theta^x) > K\right)}{p_K \sqrt{p_\infty}} + \frac{\sqrt{\mathbb{P}\left(\sup_{0 \leq \theta \leq T} \varphi(\tilde{B}_\theta^x) > K\right)}}{p_K} \right), \tag{4.23}
\end{aligned}$$

where p_K denotes the probability to accept a Brownian bridge path with the truncated algorithm at level K ,

$$p_K = \mathbb{E} \left[\exp \left(- \int_0^T K \wedge \varphi(\tilde{B}_\theta^x) d\theta \right) \right] \tag{4.24}$$

and p_∞ is given by

$$p_\infty = \mathbb{E} \left[\exp \left(- \int_0^T \varphi(\tilde{B}_\theta^x) d\theta \right) \right].$$

b. If moreover Ψ is bounded,

$$\left| \mathbb{E}\Psi(X_T^x) - \mathbb{E}\Psi(X_T^{x,K}) \right| \leq \frac{2 \|\Psi\|_\infty}{p_K} \mathbb{P} \left(\sup_{0 \leq \theta \leq T} \varphi(\tilde{B}_\theta^x) > K \right). \tag{4.25}$$

Remark 5. 1. If $\limsup_{y \rightarrow -\infty} \varphi(y) = \limsup_{y \rightarrow +\infty} \varphi(y) = +\infty$, for any Brownian bridge, the probability to wrongly accept the trajectory is positive. However, Proposition 13 gives a control of the error.

2. The result of Proposition 13 still holds true if we use the variant of the algorithm with the simulation of the minimum of the Brownian bridge (see point 2) of Remark 2 and [3]). Numerical results for this variant are given in Section 4.4.1.2.

3. If we have a control of the asymptotic behavior of φ (e.g. a polynomial growth at infinity), we deduce that the error of truncation decreases exponentially fast to 0 with K .

Proof. We denote by N^K a Poisson process on $[0, T] \times [0, K]$ and by N a Poisson

process on $[0, T] \times \mathbb{R}_+$. Thanks to (4.12), we have

$$\begin{aligned}\mathbb{E}\Psi(X_T^x) &= \mathbb{E}\left[\Psi(\tilde{B}_T^x) \mid N \cap D(\omega) = \emptyset\right] \\ &= \frac{\mathbb{E}\left[\Psi(\tilde{B}_T^x) \exp\left(-\int_0^T \varphi(\tilde{B}_\theta^x) d\theta\right)\right]}{\mathbb{E}\left[\exp\left(-\int_0^T \varphi(\tilde{B}_\theta^x) d\theta\right)\right]} \\ \mathbb{E}\Psi(X_T^{x,K}) &= \mathbb{E}\left[\Psi(\tilde{B}_T^x) \mid N^K \cap D(\omega) = \emptyset\right], \\ &= \frac{\mathbb{E}\left[\Psi(\tilde{B}_T^x) \exp\left(-\int_0^T K \wedge \varphi(\tilde{B}_\theta^x) d\theta\right)\right]}{\mathbb{E}\left[\exp\left(-\int_0^T K \wedge \varphi(\tilde{B}_\theta^x) d\theta\right)\right]}.\end{aligned}$$

We denote by p_K (resp. p_∞) the probability to accept a Brownian bridge path with the truncated algorithm at level K (resp. with the exact algorithm).

$$p_K = \mathbb{E}\left[\exp\left(-\int_0^T K \wedge \varphi(\tilde{B}_\theta^x) d\theta\right)\right], \quad (4.26)$$

$$p_\infty = \mathbb{E}\left[\exp\left(-\int_0^T \varphi(\tilde{B}_\theta^x) d\theta\right)\right]. \quad (4.27)$$

Thus, a control of the error is

$$\begin{aligned}\text{err}_K &= \left|\mathbb{E}\Psi(X_T^x) - \mathbb{E}\Psi(X_T^{x,K})\right| \\ &\leq \left|\frac{1}{p_\infty} - \frac{1}{p_K}\right| \mathbb{E}\left[\left|\Psi(\tilde{B}_T^x)\right| \exp\left(-\int_0^T \varphi(\tilde{B}_\theta^x) d\theta\right)\right] \\ &\quad + \frac{1}{p_K} \mathbb{E}\left[\left|\Psi(\tilde{B}_T^x)\right| \left(\exp\left(-\int_0^T K \wedge \varphi(\tilde{B}_\theta^x) d\theta\right) - \exp\left(-\int_0^T \varphi(\tilde{B}_\theta^x) d\theta\right)\right)\right].\end{aligned}$$

We apply Cauchy-Schwarz inequality and use that $x^2 \leq x$ for $0 \leq x \leq 1$

$$\begin{aligned}\text{err}_K &\leq \frac{p_K - p_\infty}{p_K p_\infty} \sqrt{\mathbb{E}\left[\Psi^2(\tilde{B}_T^x)\right]} \sqrt{p_\infty} \\ &\quad + \frac{1}{p_K} \sqrt{\mathbb{E}\left[\Psi^2(\tilde{B}_T^x)\right]} \sqrt{\mathbb{E}\left[\exp\left(-\int_0^T K \wedge \varphi(\tilde{B}_\theta^x) d\theta\right) - \exp\left(-\int_0^T \varphi(\tilde{B}_\theta^x) d\theta\right)\right]} \\ &\leq \sqrt{\mathbb{E}\left[\Psi^2(\tilde{B}_T^x)\right]} \left(\frac{p_K - p_\infty}{p_K \sqrt{p_\infty}} + \frac{\sqrt{p_K - p_\infty}}{p_K}\right).\end{aligned}$$

We finally observe that

$$p_K - p_\infty \leq \mathbb{P}\left(\sup_{0 \leq \theta \leq T} \varphi(\tilde{B}_\theta^x) > K\right).$$

The proof under the assumption that Ψ is bounded is very similar, except we do not need to apply Cauchy-Schwarz inequality. It is left to the reader. \square

4.2.7 Extension in finite dimension

One can easily extend the algorithm to a multi-dimensional setting under restrictive assumptions. For instance, assume that the drift derives from a potential, that is, there exists a function $\xi : \mathbb{R}^d \mapsto \mathbb{R}^d$ such that

$$X_T = X_0 + \int_0^T \nabla \xi(X_s) ds + W_T.$$

Girsanov Theorem gives the Radon-Nikodym derivative :

$$\exp \left(\int_0^T \sum_{j=1}^d \frac{\partial}{\partial x_j} \xi(W_s) dW_s^j - \int_0^T \sum_{j=1}^d \left(\frac{\partial}{\partial x_j} \xi(W_s) \right)^2 ds \right).$$

Similarly to the one-dimensional case, our assumption allows one to introduce the antiderivative of $\nabla \xi$ in order to remove the stochastic integral in the previous expression :

$$\xi(W_T) = \xi(W_0) + \int_0^T \sum_{j=1}^d \frac{\partial}{\partial x_j} \xi(W_s) dW_s^j + \frac{1}{2} \int_0^T \sum_{j=1}^d \frac{\partial^2}{\partial x_j^2} \xi(W_s) ds.$$

So, as in the one-dimensional case, we simulate a Brownian bridge with final distribution

$$\mathbb{P}(\tilde{B}_T^x \in d\theta) = C \exp \left(-\frac{\sum_j (\theta_j - x_j)^2}{2T} + \xi(\theta) \right) d\theta.$$

We only have to replace the function φ in (4.9) by $\frac{1}{2} \sum_{j=1}^d \left(\frac{\partial}{\partial x_j} \xi(y) \right)^2 + \frac{\partial^2}{\partial x_j^2} \xi(y)$.

The evaluation of the derivatives is more difficult to extend. Equation (4.16) has an equivalent in any finite dimension [6]. We can also write (4.17) but, for instance, it is not easy to write the transform of each component of the vector $Y_T^x W_T$ as the exponential of an integral.

4.3 The detailed algorithm for the CIR Model

This section is devoted to the extension of our algorithm to the simulation of the Cox Ingersoll Ross (CIR) process, a popular model in finance (for short rates or volatility for stochastic volatility model on asset, etc.) This process satisfies

$$V_T = V_0 + \int_0^T \kappa (V_\infty - V_t) dt + \varepsilon \int_0^T \sqrt{V_t} dW_t \quad (4.28)$$

where κ , V_∞ and ε are fixed constants. Usually, the parameter $d = \frac{4\kappa V_\infty}{\varepsilon^2}$ is called the degree of the CIR process. It is known that $\mathbb{P}(\inf_{\theta \in [0, T]} V_\theta > 0) = 1$

iff $d \geq 2$ (see e.g. [1]). We assume it is fulfilled. We apply the Lamperti transform to the process V , that is we set

$$X_t = \frac{2\sqrt{V_t}}{\varepsilon} =: \eta(V_t).$$

The process X satisfies the SDE

$$\begin{aligned} dX_t &= \eta'(V_t)dV_t + \frac{1}{2}\eta''(V_t)d\langle V \rangle_t \\ &= \frac{1}{\varepsilon\sqrt{V_t}} \left(\kappa(V_\infty - V_t)dt + \varepsilon\sqrt{V_t}dW_t \right) - \frac{\varepsilon^2 V_t}{4\varepsilon V_t^{3/2}} dt \\ &= \left(\frac{1}{X_t} \left(\frac{2\kappa V_\infty}{\varepsilon^2} - \frac{1}{2} \right) - \frac{\kappa X_t}{2} \right) dt + dW_t. \end{aligned} \quad (4.29)$$

It is an SDE of type (4.5) with

$$\alpha(y) = \frac{1}{y} \left(\frac{2\kappa V_\infty}{\varepsilon^2} - \frac{1}{2} \right) - \frac{\kappa y}{2} \text{ for } y > 0.$$

The associated function φ defined by (4.9) is

$$\varphi(y) = \left(\left(\frac{2\kappa V_\infty}{\varepsilon^2} - 1 \right)^2 - \frac{1}{4} \right) \frac{1}{2y^2} + \frac{\kappa^2}{8} y^2 - \frac{\kappa^2 V_\infty}{\varepsilon^2}.$$

The function φ is bounded below on $(0, +\infty)$ iff

$$\left(\frac{2\kappa V_\infty}{\varepsilon^2} - 1 \right)^2 \geq \frac{1}{4}$$

or equivalently that the degree d of the CIR satisfies $d \in (0, 1] \cup [3, \infty)$. In this paper, we assume $d \geq 3$.

Remark 6. In Section 4.2, the drift α is defined on \mathbb{R} . However, a classical Feller test proves that the process X^x , the solution of (4.29) starting from $x > 0$, never hits 0 almost surely. Formally, if we put $\alpha(y) = \varphi(y) = +\infty$ for all $y \leq 0$, the Brownian bridge paths \tilde{B} taking values in \mathbb{R}_- are almost surely rejected.

4.3.1 Final Value

In the first step, we generate the final value \tilde{B}_T^x according to (4.8). Its density is

$$h(y) = Ry^c \exp\left(-\frac{(y - \hat{x})^2}{2\sigma^2}\right) \mathbb{1}_{y \geq 0}$$

with :

$$c = \frac{2\kappa V_\infty}{\varepsilon^2} - \frac{1}{2}, \quad \hat{x} = 2\sigma^2 \frac{x}{2T}, \quad x = \frac{2}{\varepsilon} \sqrt{V_0}, \quad \sigma^2 = \frac{1}{\frac{\kappa}{2} + \frac{1}{T}}$$

and R is a normalization. Setting $\bar{x} = \frac{\hat{x} + \sqrt{\hat{x}^2 + 4c\sigma^2}}{2}$, there exists $C > 0$ such that

$$\forall y, \quad h(y) \leq C \exp\left(-\frac{(y - \bar{x})^2}{2\sigma^2}\right),$$

and we use the classical rejection procedure for random variables.

4.3.2 Simulation of the minimum

The second step consists in generating the random variables (m, t_m) , where

$$m = \inf_{0 \leq t \leq T} \left\{ \tilde{B}_t^x \mid \tilde{B}_0 = x, \tilde{B}_T^x = Y \right\} \quad \tilde{B}_{t_m}^x = m$$

This law is known (see for instance Karatzas-Shreve [9, p. 102])

$$\mathbb{P} \left[m \in d\alpha, t_m \in ds \mid \tilde{B}_T^x = Y \right] = \frac{\alpha(\alpha - Y)}{\sqrt{s^3(T-s)^3}} \exp\left(-\frac{\alpha^2}{2s} - \frac{(\alpha - Y)^2}{2(T-s)}\right) d\alpha ds.$$

In Beskos et al. [3, Prop. 2], the detailed random variables used to simulate (m, t_m) are given : the authors only need to simulate uniform, exponential and Inverse Gaussian distributions (see Devroye [5, p.149] for an efficient way to simulate Inverse Gaussian distributions).

4.3.3 Simulation of the Poisson process

We apply the method detailed in Section 4.2.4. We generate $z_1 \sim \mathcal{E}(T)$, $t_1 \sim \mathcal{U}(0, T)$, $\tilde{B}_{t_1}^x$ conditioned by $\tilde{B}_0^x, \tilde{B}_T^x, m, t_m$. If $\varphi(\tilde{B}_{t_1}^x) > z_1$, we reject the trajectory. Else, we generate $z_2 - z_1 \sim \mathcal{E}(T)$, $t_2 \sim \mathcal{U}(0, T)$, $\tilde{B}_{t_2}^x$ conditioned by $\tilde{B}_0^x, \tilde{B}_{t_1}^x, \tilde{B}_T^x, m, t_m$. If $\varphi(\tilde{B}_{t_2}^x) > z_2$, we reject the trajectory, etc.

4.3.4 Stopping condition

In this example, $\sup_{y \geq m(\omega)} \varphi(y) = +\infty$. So, we use the truncated algorithm presented in Section 4.2.5. We simulate the Poisson process on $[0, T] \times [0, \tilde{K}(\omega)]$ with

$$\tilde{K}(\omega) \geq \max\{K, \varphi(\tilde{B}_T^x), \varphi(m)\}, \quad (4.30)$$

where K is a fixed a priori threshold.

4.4 Numerical Results

In this Section, we present the numerical results. We first apply the algorithm to an academic example related to Orstein-Uhlenbeck process (Section 4.4.1). The drift α is constructed such that its associated function φ satisfies $\limsup_{y \rightarrow \infty} \varphi(y) < \infty$. In Section 4.4.2, the drift α is constructed in such a way

that the associated function φ satisfies $\limsup_{y \rightarrow -\infty} \varphi(y) = \limsup_{y \rightarrow \infty} \varphi(y) = \infty$. Finally, Section 4.4.3 is devoted to the CIR process (see Section 4.3), i.e. an example with a non Lipschitz continuous drift α .

We use the algorithms to approximate quantities (4.2), (4.3) and (4.4) for smooth and nonsmooth functions Ψ . We compare the efficiency of our algorithm to the use of a classical Euler scheme and finite difference approximation of the derivatives.

4.4.1 An academic example : a modified Ornstein Uhlenbeck

4.4.1.1 Definition

We introduce the process $(X_t^x, t \geq 0)$, solution of

$$dX_t^x = \left(-M \left(X_t^x + \frac{1}{2} \right) \mathbb{1}_{X_t^x \leq -1} + \frac{M}{2} (X_t^x)^2 \mathbb{1}_{-1 \leq X_t^x \leq 0} \right) dt + dW_t, \quad (4.31)$$

where $M \geq 1/2$ is a fixed parameter. The process X^x is solution of an SDE of type (4.5) with a drift $\alpha \in C^1(\mathbb{R})$. Its associated function φ is

$$\varphi(y) = \begin{cases} 0 & \text{if } y \geq 0 \\ \frac{M^2 y^4}{8} + \frac{M y}{2} & \text{if } -1 \leq y \leq 0 \\ \frac{M^2}{2} \left(y + \frac{1}{2} \right)^2 - \frac{M}{2} & \text{if } y \leq -1. \end{cases}$$

It satisfies

$$\lim_{y \rightarrow -\infty} \varphi(y) = +\infty \quad \text{and} \quad \lim_{y \rightarrow +\infty} \varphi(y) < \infty.$$

Then, SDE (4.31) satisfies the assumptions made in Section 4.2 and we are in position to apply our unbiased algorithm to approximate $\mathbb{E}(\Psi(X_T^x))$, $\frac{d}{dx} \mathbb{E}(\Psi(X_T^x))$ and $\frac{d^2}{dx^2} \mathbb{E}(\Psi(X_T^x))$ for general functions Ψ .

4.4.1.2 Algorithmic optimization of computation time

We have discussed in Section 4.2.4 two variants to simulate the Poisson process N used to reject (or accept) the Brownian bridge paths.

- **variant 1** by increasing times : a realization of N , say $\{(t_1, y_1), \dots, (t_{n(\omega)}, y_{n(\omega)})\}$, satisfies $t_1 < t_2 < \dots < t_{n(\omega)}$.
- **variant 2** by increasing ordinates : $\{(t_1, y_1), \dots, (t_{n(\omega)}, y_{n(\omega)})\}$ satisfies $y_1 < y_2 < \dots < y_{n(\omega)}$.

In this part, we compare the efficiency of the two variants. They only differ by the computation time used to accept a Brownian bridge path. Figure 4.1 represents the time of simulation as a function of the final time T . The size of the sample is $N_{MC} = 1e6$ and the parameters are $x = 0.04$, $M = 0.5$.

We observe that the times of simulation are very close for small values of T ; they both increase exponentially and, clearly, the rate is smaller for variant 2 than variant 1.

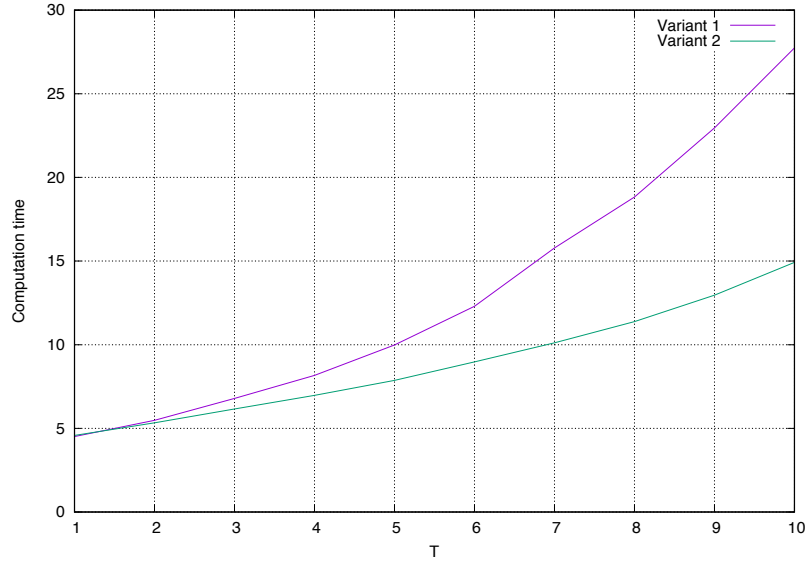


FIGURE 4.1 – Comparison of the times of simulation for two methods to generate the Poisson process : variant 1 (increasing times) and variant 2 (increasing ordinates). Times of simulation (in seconds) are given in function of the final time T . The process X^x solves (4.31). The parameters are $M = 0.5$, $x = 0.04$ and $N_{MC} = 1e6$.

We then fix the final time $T = 1$ and change the parameter M in the drift α (see (4.31)). The times of simulation of a sample of size $N_{MC} = 1e6$ are given in Table. 4.1. Again, the variant 2 is faster than variant 1.

4.4.1.3 A comparison of approximations of sensitivities

The unbiased evaluation of the sensitivities $\frac{d}{dx}\mathbb{E}^x\Psi(X_T)$ and $\frac{d^2}{dx^2}\mathbb{E}^x\Psi(X_T)$ are the main new results of the paper. They are themselves interesting theoretical results. However, we aim to compare their efficiency to classical numerical methods.

M	time (var. 1)	time (var. 2)	Ratio
1	7.79	7.74	1.01
10	31	14	2.21
100	254591	5148	49.4

TABLE 4.1 – Comparison of the times (in sec.) of simulation for variant 1 (increasing times) and variant 2 (increasing ordinates). We simulate $N_{\text{MC}} = 1e6$ values of X_T ($x = 0$, $T = 1$, $M = 1, 10, 100$).

Our unbiased estimator It is fully presented in Section 4.2. We apply a classical Monte Carlo procedure to evaluate the expressions (4.12), (4.19) and the expression p.110 for the second derivative. We denote the Monte Carlo estimators by

$$\tilde{P}_\Psi(N_{\text{MC}}), \quad \tilde{\Delta}_\Psi(N_{\text{MC}}), \quad \tilde{\Gamma}_\Psi(N_{\text{MC}}).$$

There is a unique source of error : the statistical error. It is only related to the variance of the expressions we evaluate. In Table 4.2, we present the results for three functions Ψ , two are smooth and the last one is discontinuous. We put in brackets the estimated statistical standard deviation with a sample of size $N_{\text{MC}} = 2e10$.

Standard estimator using Euler scheme and finite difference approximation We simulate $X_T^{x,\delta,1}, \dots, X_T^{x,\delta,N_{\text{MC}}}$, N_{MC} independent realisations of the explicit Euler scheme (with time step δ) to approximate the solution X_T^x of (4.5). The derivatives are approximated with a finite difference scheme. That is, we simulate $X_T^{x-dx,\delta,1}, \dots, X_T^{x-dx,\delta,N_{\text{MC}}}$ and $X_T^{x+dx,\delta,1}, \dots, X_T^{x+dx,\delta,N_{\text{MC}}}$ and use the estimators

$$\begin{aligned} \hat{P}_\Psi(N_{\text{MC}}, \delta) &:= \frac{1}{N_{\text{MC}}} \sum_{k=1}^{N_{\text{MC}}} \Psi(X_T^{x,\delta,k}) \\ &\approx \mathbb{E}(\Psi(X_T^x)), \\ \hat{\Delta}_\Psi(N_{\text{MC}}, \delta, dx) &:= \frac{1}{2dx N_{\text{MC}}} \left(\sum_{k=1}^{N_{\text{MC}}} \Psi(X_T^{x+dx,\delta,k}) - \sum_{k=1}^{N_{\text{MC}}} \Psi(X_T^{x-dx,\delta,k}) \right) \\ &\approx \frac{d}{dx} \mathbb{E}(\Psi(X_T^x)), \\ \hat{\Gamma}_\Psi(N_{\text{MC}}, \delta, dx) &:= \frac{1}{(dx)^2 N_{\text{MC}}} \left(\sum_{k=1}^{N_{\text{MC}}} \Psi(X_T^{x+dx,\delta,k}) - 2 \sum_{k=1}^{N_{\text{MC}}} \Psi(X_T^{x,\delta,k}) \right. \\ &\quad \left. + \sum_{k=1}^{N_{\text{MC}}} \Psi(X_T^{x-dx,\delta,k}) \right) \\ &\approx \frac{d^2}{dx^2} \mathbb{E}(\Psi(X_T^x)) \end{aligned}$$

These approximations are also very simple to simulate and evaluate. We now have two sources of error :

- a bias due to the parameters δ and dx .
- the statistical error, related to the variance of the quantities we estimate with a Monte Carlo procedure.

In practice, we have to carefully choose N , δ and dx . The best choice is obtained if the bias is close to the statistical error. It is not easy to reach such a balance because we do not know the bias.

We have chosen two set of parameters, $N_{MC} = 1e9$, $\delta = 0.1$ and $dx = 0.4$ in Table 4.4 and $N_{MC} = 5e7$, $\delta = 0.005$ and $dx = 0.1$ in Table 4.3.

$\Psi(y)$	$\tilde{P}_{\Psi}(N_{MC})$	$\tilde{\Delta}_{\Psi}(N_{MC})$	$\tilde{\Gamma}_{\Psi}(N_{MC})$
y^2	0.900933 (9.0e-6)	0.301072 (2.5e-5)	1.57485 (5.6e-5)
$\exp(-y)$	1.40071 (1.1e-5)	-1.16071 (2.8e-5)	0.703935 (7.2e-5)
$\mathbb{1}_{y>x}$	0.492925 (3.5e-6)	-0.3854 (4.7e-6)	-0.0219749 (8.3e-6)

TABLE 4.2 – Approximation for X^x solution of (4.31) obtained with our unbiased algorithms. $N_{MC} = 2e10$, $M = 0.5$, $x = 0.04$. The program runs 9e4 seconds.

$\Psi(y)$	$\hat{P}_{\Psi}(N_{MC}, \delta)$ $-P_{\Psi}$	$\hat{\Delta}_{\Psi}(N_{MC}, \delta, dx)$ $-\Delta_{\Psi}$	$\hat{\Gamma}_{\Psi}(N_{MC}, \delta, dx)$ $-\Gamma_{\Psi}$
y^2	8.8e-3 (4.1e-5)	5.0e-3 (1.1e-4)	1.1e-3 (1.1e-3)
$\exp(-y)$	1.5e-2 (4.9e-5)	-2e-2 (1.3e-4)	7.0e-3 (1.2e-3)
$\mathbb{1}_{y>x}$	-7.1e-5 (1.6e-5)	1.1e-2 (3.8e-5)	-2.6e-3 (3.9e-4)

TABLE 4.3 – Error with an Euler scheme with step $\delta = 0.1$ and finite difference approximation with step $dx = 0.4$. X^x is solution of (4.31), $N_{MC} = 1e9$, $M = 0.5$, $x = 0.04$. The program runs 5.6e3 seconds.

Conclusion To obtain an error of the same magnitude with our unbiased estimator, we have to use between $N_{MC} = 1e5$ and $N_{MC} = 1e6$ for the rough case (Table 4.3) and between $N_{MC} = 1e6$ and $N_{MC} = 1e7$ for the more precise case (Table 4.4). The size of the sample obviously depends on the function Ψ and the order of the derivative we approximate. Our algorithm is well adapted for the approximation of Δ_{Ψ} and Γ_{Ψ} .

In any cases, our algorithm is faster (10 to 100 times faster than the Euler scheme).

$\Psi(y)$	$\hat{P}_\Psi(N_{\text{MC}}, \delta)$ $-P_\Psi$	$\hat{\Delta}_\Psi(N_{\text{MC}}, \delta, dx)$ $-\Delta_\Psi$	$\hat{\Gamma}_\Psi(N_{\text{MC}}, \delta, dx)$ $-\Gamma_\Psi$
y^2	4.5e-4 (1.8e-4)	3.6e-3 (9.3e-4)	-2.2e-3 (1.8e-2)
$\exp(-y)$	8.1e-5 (2.1e-4)	-1.1e-3 (1.1e-3)	-1.2e-3 (2.1e-2)
$\mathbb{1}_{y>x}$	6.8e-5 (7.1e-5)	2.7e-3 (3.5e-4)	-5.0e-4 (7.0e-3)

TABLE 4.4 – Error with an Euler scheme with step $\delta = 0.005$ and finite difference approximation with step $dx = 0.2$. X^x is solution of (4.31), $N_{\text{MC}} = 5e7$, $M = 0.5$, $x = 0.04$. The program runs 2.9e3 seconds.

4.4.2 Symmetric modified Orstein-Uhlenbeck, convergence of the error of truncation

We test our unbiased algorithm to a second *toy* model. We only evaluate in this Section the error due to the truncation of the Poisson process. That is, we illustrate the results of Section 4.2.6. The comparison with an Euler scheme and finite difference approximation of the derivatives are very similar (in terms of complexity and of efficiency) to those obtained in the previous section. Thus, we do not include them for this example.

4.4.2.1 Introduction

We slightly modify the drift introduced in the previous example. In this part, we put

$$X_T^x = x + \int_0^T \alpha(X_t^x) dt + W_T$$

$$\alpha(x) = -M \left(x + \frac{1}{2} \right) \mathbb{1}_{x \leq -1} + \frac{M}{2} x^2 \mathbb{1}_{-1 \leq x \leq 1} + M \left(x - \frac{1}{2} \right) \mathbb{1}_{x \geq 1}. \quad (4.32)$$

Remark 7. For $y \leq 0$, the drift $\alpha(y)$ is identical to the drift in the previous example, but instead of putting $\alpha(y) = 0$ for $y \geq 0$, the drift is now symmetric. The associated function φ satisfies $\lim_{-\infty} \varphi = \lim_{+\infty} \varphi = +\infty$.

For any threshold K , we simulate the final value \tilde{B}_T^x , the minimum m of the Brownian bridge on $[0, T]$ and compute $\tilde{K}(\omega)$ according to (4.30). We then simulate a Poisson process N^K on $[0, T] \times [0, \tilde{K}(\omega)]$ and accept the path if $N^K \cap D(\omega) = \emptyset$, where $D(\omega)$ denotes the hypograph of $\varphi(\tilde{B}_t^x)$. We denote by $\tilde{X}_T^{x,K}$ the accepted values. We denote by p^K the probability to accept a Brownian bridge path (see (4.24)).

We use the notation $\tilde{P}_\Psi(N_{\text{MC}}, K)$ (resp. $\tilde{\Delta}_\Psi(N_{\text{MC}}, K)$, $\tilde{\Gamma}_\Psi(N_{\text{MC}}, K)$) for our Monte Carlo approximations of (4.2) (resp. (4.3), (4.4)), with a sample of size N_{MC} and a truncated Poisson process at level K .

4.4.2.2 Results

The result for $K = 100$ are given in Table 4.5 and are considered as benchmark.

$\Psi(y)$	$\tilde{P}_\Psi(N_{MC}, K)$	$\tilde{\Delta}_\Psi(N_{MC}, K)$	$\tilde{\Gamma}_\Psi(N_{MC}, 100)$
y^2	0.904526 (2.8e-5)	0.164247 (7.0e-5)	1.02012 (1.5e-4)
$\exp(-y)$	1.36243 (3.3e-5)	-1.08837 (8.7e-5)	0.564459 (2.2e-4)
$\mathbb{1}_{y>x}$	0.47637 (1.1e-5)	-0.357681 (1.4e-5)	-0.0531064 (2.6e-5)

TABLE 4.5 – Results for the approximation $X_T^{x,K}$ of the solution of (4.32). $K = 100$, $M = 0.5$, $x = 0.04$, $N_{MC} = 2e9$. The program runs 1.1e5 seconds.

In Tables 4.6, 4.7, 4.8, we can see the approximated biases for $K = 0, 1, 2$. We observe that according to Proposition 13, the bias decrease fast with K and the bias seems to be neglected for $K = 2$, even for the approximation of the derivatives.

Table 4.9 gives the empirical probability p^K to accept a Brownian bridge with the truncated algorithm at level K . It is obviously a monotonic function of K . We observe that $p^2 \approx p^{100}$ with a very large accuracy.

$\Psi(y)$	$\tilde{P}_\Psi(N_{MC}, K)$ $-\tilde{P}_\Psi(N_{MC}, 100)$	$\tilde{\Delta}_\Psi(N_{MC}, K)$ $-\tilde{\Delta}_\Psi(N_{MC}, 100)$	$\tilde{\Gamma}_\Psi(N_{MC}, K)$ $-\tilde{\Gamma}_\Psi(N_{MC}, 100)$
y^2	1.66e-2 (2.8e-5)	7.6e-4 (6.9e-5)	9.7e-3 (1.5e-4)
$\exp(-y)$	1.2e-3 (3.2e-5)	-4.9e-2 (8.6e-5)	1.9e-2 (2.2e-4)
$\mathbb{1}_{y>x}$	-8.0e-3 (1.1e-5)	-1.6e-2 (1.4e-5)	6.3e-3 (2.6e-5)

TABLE 4.6 – Errors with the truncated approximation $X_T^{x,K}$ of the solution of (4.32). $K = 0$, $M = 0.5$, $x = 0.04$, $N_{MC} = 2e9$. The program runs 1.1e4 seconds.

$\Psi(y)$	$\tilde{P}_\Psi(N_{MC}, K)$ $-\tilde{P}_\Psi(N_{MC}, 100)$	$\tilde{\Delta}_\Psi(N_{MC}, K)$ $-\tilde{\Delta}_\Psi(N_{MC}, 100)$	$\tilde{\Gamma}_\Psi(N_{MC}, K)$ $-\tilde{\Gamma}_\Psi(N_{MC}, 100)$
y^2	1.0e-4 (2.8e-5)	4.1e-4 (7.0e-5)	-9.5e-3 (1.5e-4)
$\exp(-y)$	1.0e-5 (3.3e-5)	1.4e-4 (8.7e-5)	-6.3e-3 (2.2e-4)
$\mathbb{1}_{y>x}$	1.6e-5 (1.1e-5)	5.0e-6 (1.4e-5)	6.0e-4 (2.6e-5)

TABLE 4.7 – Results for the approximation $X_T^{x,K}$ of the solution of (4.32). $K = 1$, $M = 0.5$, $x = 0.04$, $N_{MC} = 2e9$. The program runs 1.2e4 seconds.

$\Psi(y)$	$\tilde{P}_\Psi(N_{\text{MC}}, K)$ $-\tilde{P}_\Psi(N_{\text{MC}}, 100)$	$\tilde{\Delta}_\Psi(N_{\text{MC}}, K)$ $-\tilde{\Delta}_\Psi(N_{\text{MC}}, 100)$	$\tilde{\Gamma}_\Psi(N_{\text{MC}}, K)$ $-\tilde{\Gamma}_\Psi(N_{\text{MC}}, 100)$
y^2	6e-6 (2.8e-5)	-2.7e-5 (7.0e-5)	2.2e-4 (1.5e-4)
$\exp(-y)$	1.0e-5 (3.3e-5)	-1.0e-5 (8.7e-5)	4.2e-5 (2.2e-4)
$\mathbb{1}_{y>x}$	1.9e-5 (1.1e-5)	-9e-6 (1.4e-5)	-1.5e-5 (2.6e-5)

TABLE 4.8 – Results for the approximation $X_T^{x,K}$ of the solution of (4.32). $K = 2, M = 0.5, x = 0.04, N_{\text{MC}} = 2e9$. The program runs 1.3e4 seconds.

K	0	1	2	100
p^K	0.877731	0.832898	0.832884	0.832877

TABLE 4.9 – Probability p^K defined in (4.24) to accept a Brownian bridge path.

4.4.3 CIR

In this Section, we present the numerical results obtained for the simulation of the CIR process $(V_t, 0 \leq t \leq T)$, solution of (4.28) (see Section 4.3). There is a large literature on the evaluation of $\mathbb{E}(\Psi(V_T))$ (see e.g. [1]). Our aim is not to construct a specific algorithm for this particular case. However, we think that it is relevant to illustrate the efficiency of our algorithm to this non trivial case.

The numerical experiments are computed with parameters $\kappa = 0.5, V_\infty = 0.04, \varepsilon = 0.1, T = 1$ and the initial condition $v = 0.04$). The algorithm differs from the two previous examples. We first apply the Lamperti transform and simulate $X_t^x = \eta(V_t^v)$ with our (almost) unbiased algorithm (with $x = \eta(v)$). Then, for any function Ψ , we use the approximation $\tilde{P}_\Psi^X(x, N_{\text{MC}})$, $\tilde{\Delta}_\Psi^X(x, N_{\text{MC}})$ and $\tilde{\Gamma}_\Psi^X(x, N_{\text{MC}})$ constructed for the process X^x . We deduce the corresponding approximation for the CIR

$$\tilde{P}_\Psi^V(v, N_{\text{MC}}) = \tilde{P}_\Psi^X(\eta(v), N_{\text{MC}}) \quad (4.33)$$

$$\tilde{\Delta}_\Psi^V(v, N_{\text{MC}}) = \eta'(v) \tilde{\Delta}_\Psi^X(\eta(v), N_{\text{MC}}) \quad (4.34)$$

$$\tilde{\Gamma}_\Psi^V(v, N_{\text{MC}}) = \eta'(v)^2 \tilde{\Gamma}_\Psi^X(\eta(v), N_{\text{MC}}) + \eta''(v) \tilde{\Delta}_\Psi^X(\eta(v), N_{\text{MC}}). \quad (4.35)$$

Description of the compared algorithms We first (see 0-) remind the quantities we aim to estimate. Then, we describe the four algorithms we numerically compare in this Section. We explain the different quantities we compare in this Section.

0- The exact values are denoted by $P_\Psi(v)$, $\Delta_\Psi(v)$ and $\Gamma_\Psi(v)$, that is

$$\begin{aligned} P_\Psi(v) &= \mathbb{E}\Psi(V_T^v) \\ \Delta_\Psi(v) &= \frac{d}{dv} \mathbb{E}\Psi(V_T^v) \\ \Gamma_\Psi(v) &= \frac{d^2}{dv^2} \mathbb{E}\Psi(V_T^v). \end{aligned}$$

- 1- Our approximations $\bar{P}_\Psi(N_{MC})$, $\bar{\Delta}_\Psi(N_{MC})$ and $\bar{\Gamma}_\Psi(N_{MC})$ are defined in (4.33), (4.34) and (4.35).
- 2- The approximations using an Euler scheme and finite difference approximation are denoted $\hat{P}_\Psi(N_{MC}, \delta)$, $\hat{\delta}_\Psi(N_{MC}, \delta, dv)$ and $\hat{\Gamma}_\Psi(N_{MC}, \delta, dv)$ (see Sec. 4.4.1.3).
- 3- We also approximate with an Euler scheme the expression of the derivatives obtained after the Malliavin integration by part (see Section 4.2.2 and 4.2.3) : $\check{\Delta}_\Psi(N_{MC}, \delta)$ and $\check{\Gamma}_\Psi(N_{MC}, \delta)$.
- 4- Finally, we approximate Δ_Ψ and Γ_Ψ thanks to the finite difference approximation applied to our unbiased estimators of $\bar{P}_\Psi^v(N_{MC})$, $\bar{P}_\Psi^{v-dv}(N_{MC})$ and $\bar{P}_\Psi^{v+dv}(N_{MC})$. We will denote these approximations as $\tilde{\Delta}_\Psi(N_{MC}, dv)$ and $\tilde{\Gamma}_\Psi(N_{MC}, dv)$.

The results and the corresponding standard deviations of these estimators (with the truncated algorithm at level $K = 20$) are given in Tables 4.10, 4.11 and 4.12. We put in bold symbols the exact theoretical results when they are available. For the function $\Psi = \mathbb{1}_{y>v}$, we have put in the reference column (P_Ψ , Δ_Ψ , Γ_Ψ) the approximation with our methods with a sample of size $N_{MC} = 1e12$.

Discussion on the results In any column, except the third one, we observe bias for the non smooth function $\Psi(y) = \mathbb{1}_{y>v}$. Moreover, the variance of our algorithm is comparable to the variances of the biased one. In a fixed time devoted for simulation, our unbiased algorithm is always the most precise one in these examples.

Control of the error Even if the rigorous proof presented in Section 4.2.6 can not be directly used for the CIR process, a similar control of the error for the truncated algorithm should be obtained. For $K = 20$ and the bounded function Ψ case ($\Psi(y) = \mathbb{1}_{\{y>v\}}$), we obtained an accuracy of order $1e - 100$.

4.5 Conclusion

In this work, we slightly improved the Beskos et al. [3] exact method to simulate the solution of one dimensional SDEs. We simulated the Poisson process useful to reject the Brownian bridge paths in a more efficient order (by increasing ordinates). It also allowed us to extend the methodology to more general drift

$\Psi(y)$	P_Ψ	$\bar{P}_\Psi(N_{MC}^1) - P_\Psi$	$\hat{P}_\Psi(N_{MC}^2, \delta^1) - P_\Psi$	$\check{P}_\Psi(N_{MC}^2, \delta^2) - P_\Psi$
y	0.04	-4.5e-9 (1.6e-8)	7e-7 (5.0e-7)	2e-7 (5.1e-7)
$\mathbb{1}_{y>v}$	0.545628	0 (5.0e-7)	2.5e-4 (1.8e-5)	3.1e-3 (1.6e-5)
$\exp(-y)$	0.960910476	1e-9 (1.5e-8)	1.2e-7 (4.8e-7)	6e-6 (4.9e-7)

TABLE 4.10 – Estimation of the error (reference in bold) on the expectation and the corresponding standard deviation for the CIR with different methods. $N_{MC}^1 = 1e12$, $N_{MC}^2 = 1e9$, $\delta^1 = 0.001$, $\delta^2 = 0.1$.

Ψ	Δ_Ψ	$\bar{\Delta}_\Psi(N_{MC}^1)$ $-\Delta_\Psi$	$\hat{\Delta}_\Psi(N_{MC}^2, \delta^1)$ $-\Delta_\Psi$	$\check{\Delta}_\Psi(N_{MC}^3, \delta^2, dv)$ $-\Delta_\Psi$	$\tilde{\Delta}_\Psi(N_{MC}^3, dv)$ $-\Delta_\Psi$
y	0.606531	1e-6 (6.5e-6)	3.0e-3 (6.3e-5)	7.8e-3 (5.1e-5)	2e-6 (4.2e-5)
$\mathbb{1}_{y>v}$	-15.3247626	0 (8.5e-5)	-4.3e-2 (7.4e-4)	-8.0e-1 (1.5e-3)	-0.32 (1.3e-3)
$\exp(-y)$	-0.58053743	-2.1e-7 (1.3e-4)	6.2e-2 (1.2e-3)	7.6e-3 (4.9e-5)	-2e-6 (4.0e-5)

TABLE 4.11 – Estimation of the first derivative and the corresponding standard deviation for the CIR with different methods. $N_{MC}^1 = 1e12$, $N_{MC}^2 = 1e9$, $N_{MC}^3 = 1e10$, $\delta^1 = 0.001$, $\delta^2 = 0.1$, $dv = 0.01$

Ψ	Γ_Ψ	$\bar{\Gamma}_\Psi(N_{MC}^1)$ $-\Gamma_\Psi$	$\hat{\Gamma}_\Psi(N_{MC}^2, \delta^1)$ $-\Gamma_\Psi$	$\check{\Gamma}_\Psi(N_{MC}^3, \delta^2, dv)$ $-\Gamma_\Psi$	$\tilde{\Gamma}_\Psi(N_{MC}^3, dv)$ $-\Gamma_\Psi$
y	0	-5.1e-4 (1.9e-3)	-4.0e-2 (4.0e-3)	3.7e-4 (2.0e-2)	-1.1e-2 (1.7e-2)
$\mathbb{1}_{y>v}$	91.0234	0 (2.3e-2)	3.0e-1 (5.5e-2)	12 (6.2e-1)	-7.1 (5.2e-1)
$\exp(-y)$	0.35073	-3.1e-3 (3.6e-2)	-4.6e-1 (6.9e-2)	-1.0e-2 (2.0e-2)	1.0e-2 (1.6e-2)

TABLE 4.12 – Estimation of the second derivative and the corresponding standard deviation for the CIR with different methods. $N_{MC}^1 = 1e12$, $N_{MC}^2 = 1e9$, $N_{MC}^3 = 1e10$, $\delta^1 = 0.001$, $\delta^2 = 0.1$, $dv = 0.01$

functions α . In this case, we obtained a control of the error due to the truncation of the algorithm of simulation of the Poisson process.

In addition, we proposed to generalise the unbiased Monte Carlo algorithm to the estimation of the derivatives (4.3) and (4.4).

In comparison with the previous classical numerical methods, our algorithm seems to be more efficient if we want to obtain a sufficiently good accuracy. For rough approximations, the bias introduced by the Euler scheme has the same order as the statistical error of our algorithm.

Bibliography

- [1] A. Alfonsi. Affine diffusions and related processes : simulation, theory and applications, volume 6 of Bocconi & Springer Series. Springer, Cham ; Bocconi University Press, Milan, 2015.
- [2] V. Bally and A. Kohatsu-Higa. A probabilistic interpretation of the parametrix method. Ann. Appl. Probab., 25(6) :3095–3138, 2015.
- [3] A. Beskos, O. Papaspiliopoulos, and G. O. Roberts. Retrospective exact simulation of diffusion sample paths with applications. Bernoulli, 12(6) :1077–1098, 2006.
- [4] A. Beskos and G. O. Roberts. Exact simulation of diffusions. Ann. Appl. Probab., 15(4) :2422–2444, 2005.
- [5] L. Devroye. Nonuniform random variate generation. Springer-Verlag, New York, 1986.
- [6] E. Fournié, J.-M. Lasry, J. Lebuchoux, P.-L. Lions, and N. Touzi. Applications of Malliavin calculus to Monte Carlo methods in finance. Finance Stoch., 3(4) :391–412, 1999.
- [7] E. Gobet and M. Mrad. Strong approximation of stochastic processes at random times and application to their exact simulation. Stochastics, 2017. To appear.
- [8] P. Henry-Labordère, X. Tan, and N. Touzi. Unbiased simulation of stochastic differential equations. 2016. Arxiv preprint [arXiv:1504.06107v2](https://arxiv.org/abs/1504.06107v2).
- [9] I. Karatzas and S. E. Shreve. Brownian motion and stochastic calculus, volume 113 of Graduate Texts in Mathematics. Springer-Verlag, New York, second edition, 1991.
- [10] S. Maire and E. Tanré. Some new simulations schemes for the evaluation of Feynman-Kac representations. Monte Carlo Methods Appl., 14(1) :29–51, 2008.
- [11] D. Nualart. The Malliavin calculus and related topics. Probability and its Applications (New York). Springer-Verlag, Berlin, second edition, 2006.
- [12] D. Williams. Path decomposition and continuity of local time for one-dimensional diffusions. I. Proc. London Math. Soc. (3), 28 :738–768, 1974.

Chapitre 5

Indexation de medias

Résumé

Nous présentons ici des outils numériques et une méthode de construction d'une base de données de recherches de photos utilisable à grande échelle. Nous expliquons comment encoder les images, les stocker de manière à ce que la distance euclidienne de l'espace de représentation soit consistante avec la notion de similarité visuelle. Nous présentons des méthodes de compression de manière à réduire la dimension de définition des images et à utiliser les facultés de clusterisation des algorithmes de quantification ou de k-mean aussi appelés algorithmes de Lloyds. Cela nous amène à étudier la distribution de l'espace des descripteurs. Nous la comparons à des simulations gaussiennes pour lesquels nous connaissons certaines propriétés d'après des travaux sur la quantification. Nous nous intéressons au choix de la dimension critique de quantification, c'est-à-dire, celle au delà de laquelle, pour un nombre de quantifieurs donné la prise en compte de dimension plus grande n'apporte aucune amélioration. En effet la théorie nous donne des asymptotes à cette dimension critique, et il est intéressant de la comprendre dans le but de trouver la méthode de construction la plus efficace des arbres de recherches de kmeans. Nous présentons donc la façon dont nous les construisons.

Notre résultat principal consiste à proposer une amélioration de la méthode du K-mean pour être utiliser en très grande dimension. étudier la dimension critique de quantification d'un point de vue théorique dans la partie 5.2.1.3, mais surtout de la chercher numériquement dans 5.2.3.1 et de la comparer avec des données gaussiennes simulées ayant le meme spectre (distribution des valeurs propres) dans 5.2.3.2. Nous étudions aussi la distribution du spectre conditionnel aux clusters que nous présentons dans 5.3.1. Nous proposons aussi des méthodes pour améliorer l'algorithme de Lloyds dans 5.4.2 qui doivent permettre d'éviter un certain nombre de minima locaux en particulier en grande dimension.

Abstract

The main results of this chapter is to propose a enhancement of Lloyds algorithm and study the distribution of deep learning feature in order to propose a way of indexing visually media.

Indexing visually media means to be able to query efficiently a set of media with a new one by retrieving the subset that looks like the most like the one used as a query.

This chapter is organized as follow. In section 5.1 we explain how a media (photo or video) is encoded as a vector, and using neural network in a sensible way, where small euclidean distance is linked to similarity of the represented subject of the images. We introduce in 5.1.2 two compression methods. Section 5.2 present functional quantization and theoretical optimal quantization for gaussian vector in high dimension, known as critical dimension. Then in ?? using the PCA compression method we deduce from the theoretical computation the optimal quantization of the data representing the media. Section 5.4.1 explain the Lloyds Algorithm, also known as K-mean, display several metrics to better understand the datas and the numerical algorithm and introduce an enhancement based on these empirical observations in 5.4.2. Finally in 5.2.3, we uses Lloyds algorithm to find the critical dimension. Using gaussian simulations we validate numerically the theoretical approach and observe difference with the data representing the media that comes from the dependance even after empirical decorrelation.

5.1 Media as vector of an Euclidean space

We describe here how a media (photo or video) is encoded as a vector, and using neural network in a sensible way, where small euclidean distance is linked to similarity of the represented subject of the images.

5.1.1 Images : From a pixel to a sensible compressed vector

5.1.1.1 Brute Euclidean space for the definition of images

A photo displayed on a screen is a matrix of pixels. Each pixel is defined as a three variable vector for each primary color : red, green and blue for the RGB standard. Since around the 2000', the intensity of each color is defined on an integer scale going from 0 to 255 ($256 = 2^8$ different values and can be coded on one byte (i.e. the concatenation of 8 bits)). This means 16 millions of different colors (256^3). Ultra High definition aims to increase slightly this amount to 2^{12} for one primary color of one pixel, which would only increase by 50% the amount of data needed to store one pixel. If we consider for the sake of simplicity a 1000×1000 images pixels, we need 3 millions bytes to store without any compression this image (up to 4.5 millions in Ultra High definition colors).

Despite the aestheticism and the quality of the image, this amount of pixel is enough for the problem we aim to solve, either recognizing a face, an object, a monument, or classifying a scene (landscape or sport or event among others). So is the case even if the interesting part of the images covers only one percent of its surface. Instead of classifying an image, this task would be called a detection.

Without any compression or sensible transformation, the objects we are trying to manipulate are, for this specific case in 3,000,000 dimension.

Moreover, in this case the Euclidean distance is not very sensible, if for local (in the Euclidean way) deformation of an image will still remain a representation of the same object, scene, face, with either some obfuscation of a part of the image, or change in luminosity it is not mainly the case.

Indeed a change in the "cropping" of the image - this means changing the position of the camera which would take a photo or cutting differently an image coming from a bigger one - could induce a huge Euclidean distance whereas it could remain visually the same object or scene for a human eye.

It is not the case if we consider a view of a beach with a clear blue sky (without sun directly visible) where the sand, the water and the sky would have light color. But if we take a photo of clouds, a forest or most of the human scene (an inner living room, office or street view) with a lot of contrasts, the distance will become huge and would not be sensible to describe the visual similarity of the media.

In others terms we would need to label a huge amount of images in order to map this space. And the set of images representing different objects, would have very complex topology in this space.

For example the set of images that represent either a car or a plane (which are two industrial objects, externally made of brilliant material transparent or not, in some specific configurations) would be very intricate and in very little amount regarding all the "Picasso" like transformation that could be performed between one and another and have no interest for common representation of everyday objects. The aim of building such map of this space would be to automatize a description of everyday scenes and objects for marketing purpose or automatization of car driving in the streets for example. In such a situation, we are not interested in "all the images we do not see". A short and restrictive list of these images would be the view of a biological cell, the inner constituent of stones or earth material and the view of interplanetary gaz and nebulae.

In order to manipulate easily large amount of media and store them in a sensible way, we would need to compress the data which start brutally around a million of dimension.

A first step in the compression of an image can be done by processing classical (around 2012-2013) neural networks of photo classification or detection. A commonly used reference comes from the participant of ImageNet challenge [16], with former versions available in the Caffe Library [8].

5.1.1.2 Our space of description, a database to index for quick search

We consider therefore having transformed our images through one of these neural networks which initially meant to classify or detect object. These network are pre-trained on learning set containing between thousands and millions of media. The sets are usually labelled manually.

Therefore we now consider having compressed an image around 10,000 dimensional vector.

We have a database of around one billion media (photos and videos) that are described as thousands of parameters. In this vector space, Euclidean distance is linked to visual similarity. We consider the problem of querying this database with a new media also defined in the same vector space. We would like to "index" our database and approximate distances in order to search more efficiently for similar media than computing all the distances.

In the following, we will consider successive stages of indexing and compressing this database. The first one consists in using neural network encoder known as siamese network to compress our space or using a principal component analysis of the distribution of the data. The second one consists in using k-mean encoder and more specifically, k-mean tree known as hierarchical k-mean to design an index structure and enable a quicker search in the database.

5.1.2 Compression of the vector of descriptors of our images

5.1.2.1 Principal component analysis

Principal component analysis methodology consists in finding a linear transform that will sort the explanatory variable by their importance. The idea is simply to use the orthonormal basis of eigenvectors of the covariance matrix ordered according to decreasing eigenvalues.

This is commonly used when the data live in an affine manifold. When they live in a curved manifold (sphere, tore, ...), this method is not considered as efficient. However in huge dimension it remains useful to determine the affine manifold with the lowest dimension which "almost" contains all the data (eg \mathbb{R}^3 if the data lives in a sphere) .

Two important property of this methodology is to orthogonalize the data after transformation by empirically remove their correlation and to sort the data by their decreasing variance. Some observation on the datas describing the media seems to show that the data are not independent, ie that they are not Gaussian as shown in figure 5.12. Therefore keeping an arbitrary amount of dimension will be the best choice of these dimension in order to keep the most information from the data. This property is used in Lemma 3.a.

It can also be seen as finding the Gaussian distribution that will best fit the observed distribution and project the data on the eigenvectors basis.

We have tested the impact of the dependance against the fact it is not Gaussian while searching the critical dimension.

The figure 5.10 shows two estimation of critical dimension that are very near. One was done while simulating Gaussian distribution with the same eigenvalues. The second was done while simulating each dimension independantly but with the same marginal distribution as the descriptors of our photos. To be precise, we have simulated data by taking each coordinate from one media descriptor but using independant media for each coordinate. It shows that the critical dimension is in this case very near the Gaussian case.

Hint to remove empirical dependance in the datas To complete the study it would be of some interest to find a decomposition that not only removes the empirical correlation but also removes dependance coming from joint moment of higher order. This study is not made in this work. We would need to design a transformation that could be non linear or at least have more constraint than removing decorrelation. Neural network could be interesting for such goal. We could also design a linear transformation that would remove the most important dependance of joint order up to 4 or 5. We would like to see difference in the critical dimension in this case. If the transformation is linear we know some property that the sequence of eigenvalues should keep, for example their sum.

5.1.2.2 Compression through neural network

Another way of compressing our data relies on artificial neural network training. It makes sense to use neural network problem specification to precisely learn for this task. The parametrization of our neural network with its different layer and the loss function is the key aspect of this approach. It can also add non linearity in our compression algorithm which should be always better.

We have used Keras library [3] in order to perform this task.

For confidentiality reason, we are not allowed to describe precisely what is done in this stages, we will give some auxiliary results and provide deeper analysis with the principal component analysis.

Such approach has also been presented in [5].

5.2 High Dimension Quantization using functional quantization methodology

We present in this section functional quantization and theoretical optimal quantization for gaussian vector in high dimension, known as critical dimension.

[1] is a good reference for vector quantization and data encoding. [14] introduces the uses of numerical application of quantization.

The section 1.2 presents some general property, constructions and applications of vector quantization.

5.2.1 Functional quantization in the Gaussian setting

5.2.1.1 Construction and application to numerical integration

The section 1.2.2.3 present this approach, we recall here the main ideas of functional quantization of the Brownian motion which is a Gaussian process.

Karhunen-Loève Decomposition The idea is to replace the distribution of trajectories $W \in \mathbf{L}^2([0, T])$ with a discrete distribution whose support has size N .

We use the Karhunen-Loève decomposition of Brownian motion on an Hilbert basis of $\mathbf{L}^2([0, T])$.

Luschgy and Pagès propose to quantize W in [13] using the K-L decomposition that we explicit here.

Definition 14. The Karhunen-Loève decomposition of a standard Brownian motion gives $\forall 0 \leq t \leq T$

$$B_t = \sum_{n \in \mathbb{N}} \xi_n \sin \left(\pi \frac{t}{T} \left(n - \frac{1}{2} \right) \right) \sqrt{\frac{2}{T}} \frac{T}{\pi (n - 1/2)},$$

where ξ_n are indepent Gaussian random variable of normalized variance and of mean 0.

Regarding the study in section 5.2.2.1 that discuss the distribution of eigenvalues of covariance matrix, it is important to observe that the sequence of variance are equivalent to $\frac{1}{n^2}$ which is of finite sum.

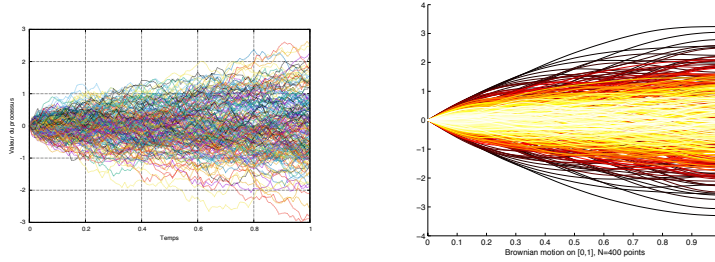
Functionnal quantization We then choose a truncation k of this sequence as well as a quantization method. We can use a product quantization like the one done in [9], it means by quantizing each one dimension distribution independently. In higher dimension it can causes combinatory exponential increase of the number of parameters.

For derivatives pricing, a truncation in small dimension is enough. We will see theoretically in 5.2.1.3 how it is linked to some asymptotic property of the eigenvalues of the covariance matrix of the random variable to quantize.

We can also make a joint quantization like we do in our numerical study of critical dimension in 5.2.3.

We then have a finite number of continuous trajectory $\{\hat{B}^{(1)}, \dots, \hat{B}^{(N)}\}$ that are a good approximation of the distribution of the Brownian motion B on interval $[0, T]$.

For a good understanding of these approximation, here are example of simulated brownian motion and of quantized brownian motion.



Reference [7] gives us hint how to choose the optimal dimension to quantize in asymptotical case, we want to study these calculations and observe how it can fits our problem.

5.2.1.2 Asymptotic property and distribution of the eigenvalues

We now introduces some concept needed to define the problem of the critical quantization dimension, some of its existence property and derive some of its asymptotic behavior.

Regular variation function : definitions

Definition 15. A fonction $\varphi : \mathbb{R}_+^* \rightarrow \mathbb{R}$ is said to be of regular variation if exists b such that $\lim_{x \rightarrow +\infty} \frac{\varphi(tx)}{\varphi(x)} = t^{-b}$, let b be its regularity module.

Definition 16. A function is said to be of slow variation if it is a regular varying function of module 0.

Proposition 17. If a function φ is of regular variation of module b , then $x \in \mathbb{R}_+^* \mapsto \varphi(x) x^b$ is of slow variation.

This can be applied to functions taking values in \mathbb{N}_* .

Critical quantization dimension

Definition 18. Let X be a random variable taking value in an Hilbertian space H . Let E be a subspace of H . We consider a quantizer of size N approximating the random variable X . The critical dimension d_N if exists, is the lowest integer d such that there exists a subspace $E \subset H$ and an optimal quantizer of size N of X that is included in E , itself of dimension d .

Proposition 19. The critical dimension exists.

Proof. Given the existence of the optimal quantizer, of cardinal N , it is included in a subspace of dimension N , therefore $d \leq N$, and it exists. \square

We can refer to [4] or [11] for the existence of optimal quantizer in infinite dimension or a more recent work in [7].

Theorem of critical dimension

Theorem 20. Let H be a Hilbert space and X a Gaussian random variable in this space. Let λ_i be the decreasing sequence of eigenvalues of its covariance matrix. If λ_i is a slow varying sequence of module b .

Then the critical dimension $d \underset{N \rightarrow +\infty}{\sim} \frac{2}{b} \log(N)$

This is linked to the Zador theorem 2 and calculation developped in 5.2.1.3.

We study numerically in section 5.2.3 the critical dimension. In this case we start from small dimension.

Functional quantization is a method of quantization of a process on a finite interval. However it implies theoretically to quantize an infinite number of dimension. However it is of finite variance, so the sum of the variance for each dimension is finite. From Karhunen-Loève decomposition, deeply studied for the Brownian motion, it is a sequence equivalent to $\frac{1}{n^2}$.

Our test shows that our space of media is very different The paragraph 5.2.2.1 estimate the module of finite variation of the sequence of eigenvalues.

We are interested in the critical dimension to quantize for a fixed number of quantizer. The theorem 20 gives asymptotic behavior.

For information, in these case, the remaining variance that is not quantized is of the order of the local inertia after optimization of the quantizer. In the case of the Karhunen-Loève decomposition, half of the total variance comes from the first dimension. To reach such a threshold in our case we need to quantize at least these few first hundred of dimension.

For this and in view of the building of hierarchical search tree that we discuss in 5.3.1, it is of high interest to consider the distribution of the variance inside a cluster. In fact we can understand that while the remaining variance of the

first dimension is higher than the variance of the others dimension, the optimal quantizer remains in this subspace. We have to remind that the variance of the whole distribution not restricted to one cluster of the dimension are independent after PCA projection.

We note $(\lambda_i)_{i \in \mathbb{N}}$ the infinite sequence of ordered eigenvalues.

Lemma 3.a. *There exists an optimal quantizer for a critical dimension d_N that has for projection an optimal quantizer on the subset of the first d_N eigenvectors attached to the first ordered eigenvalues.*

Lemma 3.b. *If $\lambda_{d_N} > \lambda_{d+1}$, then every optimal quantizer reaching d_N has this property.*

Definition 21. We use E_i to denote the subspace generated by the i first eigenvectors of the decreasing eigenvalues.

We remind the quantization error cited in 5.1. For a random variable X , and an integer N , Γ_N^* denote its optimal quantization grid and π_{Γ^*} its projection on it. Its local inertia is

$$\underline{\mathcal{D}}_N^{X,p} := \mathbf{E} \|X - \pi_{\Gamma_N^*}(X)\|_p^p = \mathbf{E} \left\| \min_{x^i \in \Gamma_N^*} |X - x^i| \right\|_p^p. \quad (5.1)$$

We will study numerically in the following section 5.2.3 the quadratic quantization error to find numerically the critical dimension.

We will note the restriction of the quadratic quantization error to the subset E_i as $\underline{\mathcal{D}}_N^{X,2,i}$. For example we have $\lambda_i = \underline{\mathcal{D}}_1^{X,2,i} - \underline{\mathcal{D}}_1^{X,2,i-1}$.

5.2.1.3 Theoretical study of the quantization error and of the critical dimension

Asymptotical optimal dimension The following calculation are inspired from [12] where the critical dimension for a quantizer of size N is derived in rigorous manner.

The idea here is to understand how the critical dimension behaves in different case, here we consider its asymptotic form in the case of Gaussian random variable. We will see later the numerical difference between the Gaussian case and our studied media database.

Having $X \sim (\otimes_{i=1}^d \mathcal{N}(0, \lambda_i))^2$. According to Zador's theorem that holds in finite dimension d

$$e_{N,p}(X)^p \sim J_{p,d} \times N^{\frac{p}{d}} \|\varphi_\lambda\|_{\frac{d}{d+p}},$$

where φ_λ is the density of X .

$$\varphi_\lambda = \frac{e^{-\frac{1}{2} \sum_{i=1}^d \frac{(\xi^i)^2}{\lambda_i}}}{\det(\text{Diag}(\lambda_1, \dots, \lambda_d))^{\frac{1}{2}} (2\pi)^{d/2}}$$

$$\begin{aligned}
\|\varphi_\lambda\|_{\frac{d}{d+p}} &= \int e^{-\frac{d}{2(d+p)} \sum_{i=1}^d \frac{(\xi^i)^2}{\lambda_i}} \frac{d\xi}{\sqrt{\left(\prod_{i=1}^d \lambda_i\right)^{\frac{d}{d+p}} (2\pi)^{\frac{d}{2} \frac{d}{d+p}}}} \\
&= \int e^{-\frac{|u|^2}{2}} \frac{\prod_{i=1}^d \left(\sqrt{\lambda_i} \left(1 + \frac{p}{d}\right)^{\frac{1}{2}}\right) du}{\left(\prod_{i=1}^d \lambda_i\right)^{\frac{d}{2(d+p)}} (2\pi)^{\frac{d}{2} \frac{d}{d+p}}} \\
&= \left(1 + \frac{p}{d}\right)^{\frac{d}{2}} \left(\prod_{i=1}^d \lambda_i\right)^{\frac{1}{2} - \frac{d}{2(p+d)}} (2\pi)^{\frac{d}{2} \left(1 - \frac{d}{d+p}\right)} \underbrace{\int e^{-\frac{|u|^2}{2}} \frac{du}{(2\pi)^{d/2}}}_{=1} \\
&= \left(1 + \frac{p}{d}\right)^{\frac{d}{2}} \left(\prod_{i=1}^d \lambda_i\right)^{\frac{p}{2(p+d)}} (2\pi)^{\frac{dp}{2(d+p)}} \times 1
\end{aligned}$$

Therefore

$$\|\varphi_\lambda\|_{\frac{d}{d+p}} = \left(1 + \frac{p}{d}\right)^{\frac{d}{2} \left(1 + \frac{p}{d}\right)} \left(\prod_{i=1}^d \lambda_i\right)^{\frac{p}{2d}} (2\pi)^{\frac{p}{2}}$$

$$\text{For } p = 2, \|\varphi_\lambda\|_{\frac{d}{d+p}} = \left(1 + \frac{2}{d}\right)^{\left(1 + \frac{d}{2}\right)} \left(\prod_{i=1}^d \lambda_i\right)^{\frac{1}{d}} 2\pi.$$

From finite dimension to infinite dimension The previous calculation holds for finite dimension. In order to use these results to the infinite dimension, we will consider an optimization of the chosen finite dimension $d \in \mathbb{N}$.

For a Gaussian process which Karhunen-Loève (PCA) decomposition has for spectrum $(\lambda_i)_{i \geq 1}$ with $\sum_{i \geq 1} \lambda_i < +\infty$.

If we decide to quantize $\otimes_{i \geq 0} \mathcal{N}(0, \lambda_i)$ as follows :

- Optimal quantization on the d -first component,
- Trivial quantization (1 quantizer) on the remaining dimensions.

Then, thanks Pythagoras theorem, using a slight abuse of notation to describe the quantifier in the infinite dimension space projected on a subspace, we have

$$e_{N,2} \left(\otimes_{i \geq 0} \mathcal{N}(0, \lambda_i), \Gamma_N^{(d)} \right)^2 = e_{N,2} \left(\otimes_{i=0}^d \mathcal{N}(0, \lambda_i) \right)^2 + \sum_{i \geq d+1} \lambda_i$$

We make the hypothesis that this is the optimal quantizer can be found by optimizing over the chosen d . Numerical tests are shown using this hypothesis to find the critical dimension in 5.2.3.

We now consider the asymptotic theorem of Zador as a non asymptotic result holding for every N and d . Then we have

$$e_{N,2} \left(\otimes_{i \geq 0} \mathcal{N}(0, \lambda_i), \Gamma_N^{(d)} \right)^2 \simeq J_{2,d} N^{-\frac{2}{d}} \left(1 + \frac{2}{d} \right)^{1+\frac{d}{2}} \left(\prod_{i=1}^d \lambda_i \right)^{\frac{1}{d}} 2\pi + \sum_{i \geq d+1} \lambda_i$$

We know furthermore that $J_{p,d} \sim \left(\frac{d}{2\pi e} \right)^{\frac{p}{2}}$ therefore $J_{2,d} \sim \frac{d}{2\pi e}$ when $d \rightarrow \infty$, we assume there is again equality when d is big. We use the fact that $\left(1 + \frac{2}{d} \right)^{1+\frac{d}{2}} \sim e$

Therefore

$$J_{2,d} N^{-\frac{2}{d}} \left(1 + \frac{2}{d} \right)^{1+\frac{d}{2}} \left(\prod_{i=1}^d \lambda_i \right)^{\frac{1}{d}} 2\pi \simeq N^{-\frac{2}{d}} d \left(\prod_{i=1}^d \lambda_i \right)^{\frac{1}{d}} .$$

Finally

$$e_{N,2} \left(\otimes_{i \leq 1} \mathcal{N}(0, \lambda_i), \Gamma_N^{(d)} \right)^2 \simeq d \left(\prod_{i=1}^d \lambda_i \right)^{\frac{1}{d}} N^{-\frac{2}{d}} + \sum_{i \geq d+1} \lambda_i \quad (5.2)$$

If $\lambda_i = \frac{c}{i^b}$, $b > 1$ (in fact $\lambda_i = \varphi(i)$, with φ a regular variation function of coefficient $-b$).

$$\left(\prod_{i=0}^d \lambda_i \right)^{\frac{1}{d}} = \frac{c}{d!^{\frac{b}{d}}}$$

and

$$\sum_{i \geq d+1} \lambda_i \simeq (\text{or } \leq) c \int_d^\infty \frac{du}{u^b} = \frac{c}{b-1} d^{1-b}$$

Inserting this result in equation 5.2 we get :

$$e_{N,2} \left(\otimes_{i \geq 1} \mathcal{N}(0, \lambda_i), \Gamma_N^{(d)} \right)^2 \simeq c \left[dd!^{-\frac{b}{d}} N^{-\frac{2}{d}} + \frac{d^{1-b}}{b-1} \right]$$

We derive from Stirling formula ($d! \sim \left(\frac{d}{e} \right)^d \sqrt{2\pi d}$), when $d \rightarrow +\infty$,

$$dd!^{-\frac{b}{d}} \sim d \left(\frac{e}{d} \right)^b (2\pi d)^{-\frac{b}{2d}} \sim d^{1-b} e^b$$

So that

$$\begin{aligned} e_{N,2} \left(\otimes_{i \geq 1} \mathcal{N}(0, \lambda_i), \Gamma_N^{(d)} \right)^2 &\simeq c \left[d^{1-b} e^{b-\frac{2}{d} \log(N)} + \frac{d^{1-b}}{b-1} \right] \\ &\simeq cd^{1-b} \left[e^{b-\frac{2}{d} \log(N)} + \frac{1}{b-1} \right]. \end{aligned}$$

We derive formally in d to determine the infimum :

$$cd^{-b}(1-b) \left[e^{b-\frac{2}{d}\log(N)} + \frac{1}{b-1} \right] + cd^{1-b} \left(\frac{2}{d^2} \log(N) \right) e^{b-\frac{2}{d}\log(N)} = 0$$

i.e.

$$e^{b-\frac{2}{d}\log(N)} \left[1-b + \frac{2}{d} \log(N) - e^{-(b-\frac{2}{d}\log(N))} \right] = 0$$

If $d \simeq \frac{2}{b} \log(N)$, the inner part is $1-b + b-1 \simeq 0$.

Therefore the resulting optimal quantization error is

$$c \left(\frac{2}{b} \log(N) \right)^{1-b} \left(1 + \frac{1}{b-1} \right) = c \frac{b}{b-1} \times \left(\frac{2}{b} \right)^{1-b} (\log(N))^{1-b}$$

So that

$$e_{N,2} \left(\otimes_{i \geq 1} \mathcal{N}(0, \lambda_i), \Gamma_N^{(d)} \right) \simeq \sqrt{c} \left(\frac{b}{b-1} \right)^{\frac{1}{2}} \times \left(\frac{2}{b} \right)^{\frac{1-b}{2}} \frac{1}{(\log(N))^{\frac{b-1}{2}}}$$

Non finite sum case When $b < 1$, with a truncated sequence in some high dimension D , we need to replace the tail of the variance with :

$$\sum_{i=d+1}^D \lambda_i \simeq c \int_d^D \frac{du}{u^b} = \frac{c}{1-b} (D^{1-b} - d^{1-b}),$$

And then from Stirling formula

$$\begin{aligned} e_{N,2} \left(\otimes_{i \geq 1} \mathcal{N}(0, \lambda_i), \Gamma_N^{(d)} \right)^2 &\simeq c \left[dd!^{-\frac{b}{d}} N^{-\frac{2}{d}} + \frac{D^{1-b} - d^{1-b}}{1-b} \right] \\ &\simeq c \left[d^{1-b} e^{b-\frac{2}{d}\log(N)} + \frac{D^{1-b} - d^{1-b}}{1-b} \right] \end{aligned} \quad (5.3)$$

Choosing $d = \rho D$, $\rho \in]0, 1[$, we have from 5.3

$$cD^{1-b} \left[\rho^{1-b} e^{b-\frac{2}{\rho D}\log(N)} + \frac{1-\rho^{1-b}}{1-b} \right]$$

And when derivating regarding ρ and noting A the formula above :

$$\begin{aligned} \frac{\partial A}{\partial \rho} &= (1-b) \rho^{-b} e^{b-\frac{2}{\rho D}\log(N)} + \frac{2}{\rho D} \log(N) e^{b-\frac{2}{\rho D}\log(N)} - 1 \\ &= \rho^{-b} \left[1-b + \frac{2}{\rho D} \log(N) - e^{-(b-\frac{2}{\rho D}\log(N))} \right] e^{b-\frac{2}{\rho D}\log(N)} \\ &= 0 \end{aligned}$$

If and only if $\frac{2}{\rho D} \log(N) = b$, ie $\rho = \frac{2}{bD} \log(N)$.

Therefore the optimal choice is $\rho = \rho_N = \frac{2}{bD} \log(N)$

with the condition that $\rho \in]0, 1[$ ie $\frac{2}{bD} \log(N) \ll 1$ ie $N \ll e^{\frac{bD}{2}}$

These computations are adapted from an asymptotic case, we are comparing it with the critical dimension studied numerically in 5.2.3.

We will now discuss more in detail about property that gives us idea how to quantize efficiently our media database presented in 5.1, optimally in some way.

5.2.2 Topology of deep learning features

Topology of deep learning features have been studied in several publication, we just cite [2].

PCA enable to compress the data but also to analyse it in term of sequence of eigenvalues, or independent decreasing variance.

In figure 5.1, we can observe the sequence for the main case we have studied.

We can observe that the slope b of the following graph which is an approximation of the module of finite variation, gives a sequence of variance of infinite sum. We remind that from the Riemann sequences, $\frac{1}{n^{-b}}$ is of finite sum if and only if $b > 1$.

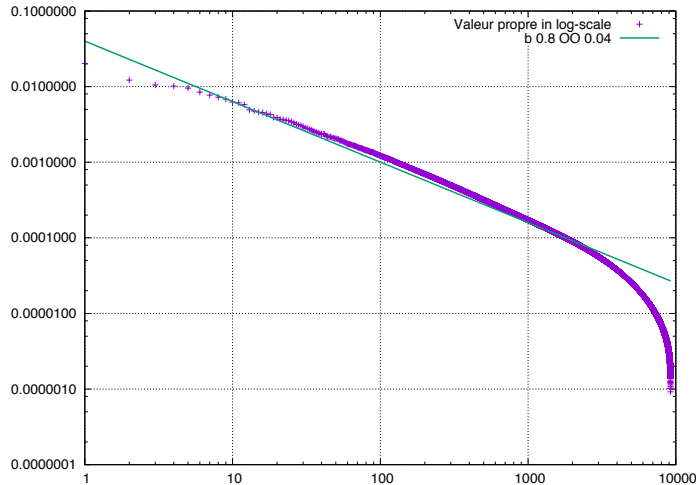


FIGURE 5.1 – Interpolation of eigenvalues

5.2.2.1 Spectrum : Impact of the intelligence of the network on the distribution of the variance $b < 1$

You will find in table 5.1 more information on the neural network and different processing chosen to describe our space where we have gather the different value of the module of finite variation of the sequence of eigenvalues.

We observe that for more general layer describing feature of photos, the parameter is lower because the information is better split around the different dimension. For this observation, we can simply compare feature 1 and 2 and 1 and 3. Trivial compressing increases this module whereas cropping the images decreases it. We can observe this between 15 and 2 and 15 and 12. Using wider learning set with places (scenes) increases it as 15 and 13 shows it. Cropping seems to have a much greater impact than adding scenes in the learning set.

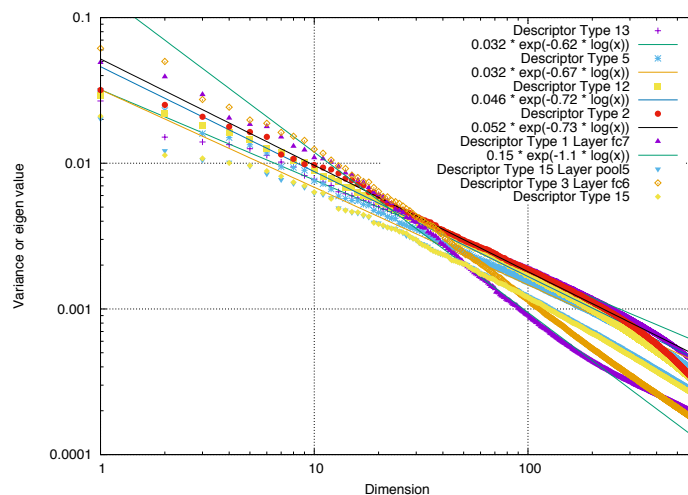


FIGURE 5.2 – Interpolation of eigenvalues for others descriptors

5.2.2.2 Consistency of the K-mean for backward similar photo search : order of correctness

Definition 22. We define the property of being a correct cluster of order n by the fact that the n nearest neighbours of the dataset belong to the cluster.

For cloud images of around a million in one thousand dimension and a hundred of quantifier, most of them are not correct of order 1. This is due to the distribution of the images in the space.

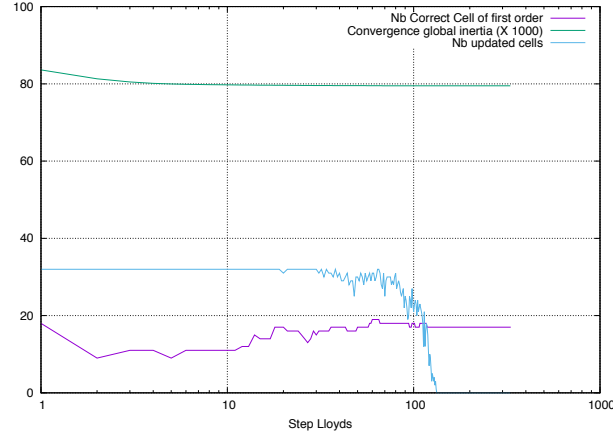


FIGURE 5.3 – Long term Convergence of inertia and number of correct cell of order one as defined in 22

5.2.2.3 Study of the topology of nearest and farthest item

In Figure 5.4, we observe a metric of the filling of the space in different dimension for different number of items in the space. The metric is simply the mean of the ratio of the smallest and highest distances. The mean is taken over some item that are used as query.

5.2.3 Searching for the critical dimension

5.2.3.1 Our main case

We display here the study performed on our main case, listed as PD_{15} in Table 5.1.

On figure 5.5 we simply display the total inertia after optimization thanks Lloyd's algorithm.

Figure 5.6 shows the amount of inertia that has been optimized. If we add the inertia of the dimension that have not been included in this optimization process, we would get the efficiency of the optimized quantizer.

For a fixed number of quantizers, when increasing the number of dimensions in figure 5.6 no longer increases the optimized inertia, we have reached the critical dimension. In figure 5.7 we plot the ratio of optimized inertia for each number of quantizer divided by the maximum optimized inertia that is reached for the highest number of dimension considered. The critical dimension can be approximated by the contour line.

The yellow line, that is the contour when 97% of the inertia is optimized is

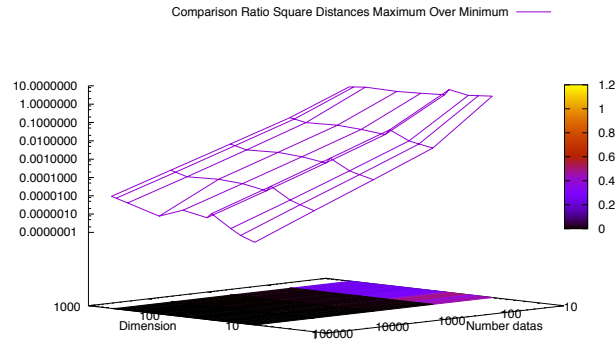


FIGURE 5.4 – Comparison of mean value of ratio of nearest neighbor and farthest one

a numerical approximation of the critical dimension. We have given asymptotic property in the Gaussian case (infinite dimension but finite variance due to convergence of the sequence of the eigenvalues) in section 5.2.1.2.

Figure 5.11 shows these different amount and validates the critical dimension for Gaussian data.

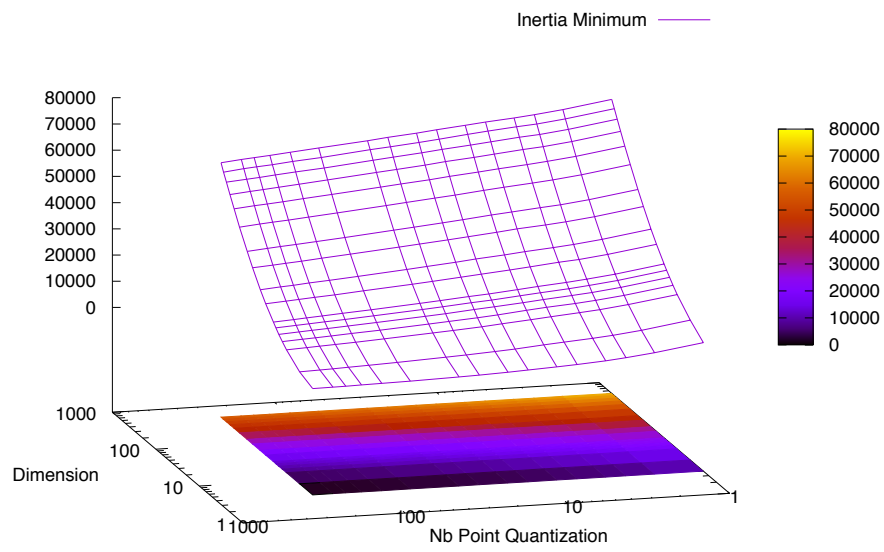


FIGURE 5.5 – $\underline{\mathcal{D}}_N^{X,2,d}$: Minimum Value of the local inertia for different dimension d and number of quantization points N (PCA of PD_{15})

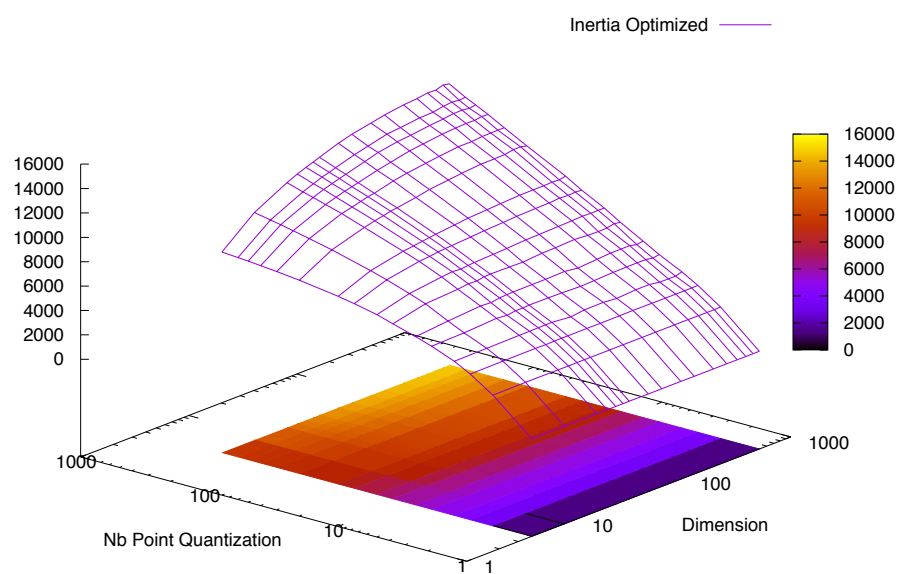


FIGURE 5.6 – $\underline{\mathcal{D}}_1^{X,2,d} - \underline{\mathcal{D}}_N^{X,2,d}$ Amount of inertia that can be optimized (ACP of PD_{15})

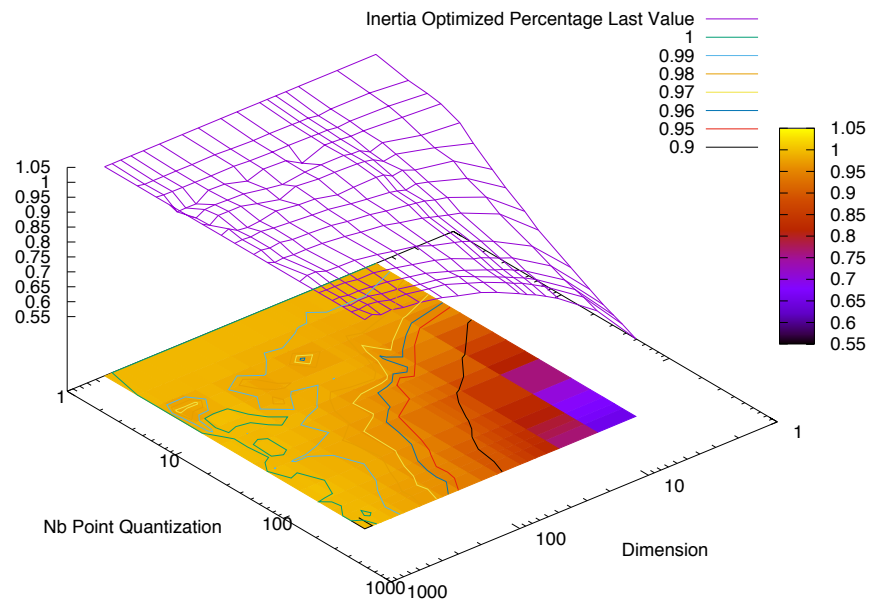


FIGURE 5.7 – $\frac{\underline{\mathcal{D}}_1^{X,2,d} - \underline{\mathcal{D}}_N^{X,2,d}}{\underline{\mathcal{D}}_1^{X,2,d_M} - \underline{\mathcal{D}}_N^{X,2,d_M}}$ with contour line approximating the critical dimension in term of percentage of inertia that can be minimized (PCA of PD_{15}), d_M is the highest dimension. $1e5$ data were used for this approximation.

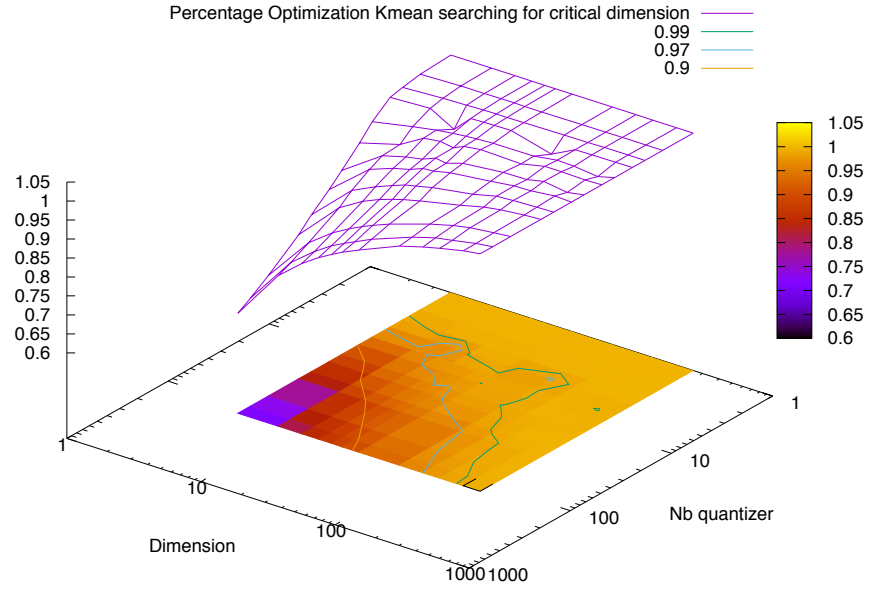


FIGURE 5.8 – $\frac{\underline{\mathcal{D}}_1^{X,2,d} - \underline{\mathcal{D}}_N^{X,2,d}}{\underline{\mathcal{D}}_1^{X,2,d_M} - \underline{\mathcal{D}}_N^{X,2,d_M}}$ with contour line approximating the critical dimension in term of percentage of inertia that can be minimized (PCA of PD_{15}), d_M is the highest dimension with $10E5$ media item.

5.2.3.2 Comparison with Gaussian simulation

We use the same spectrum but simulate independent Gaussian random variable.

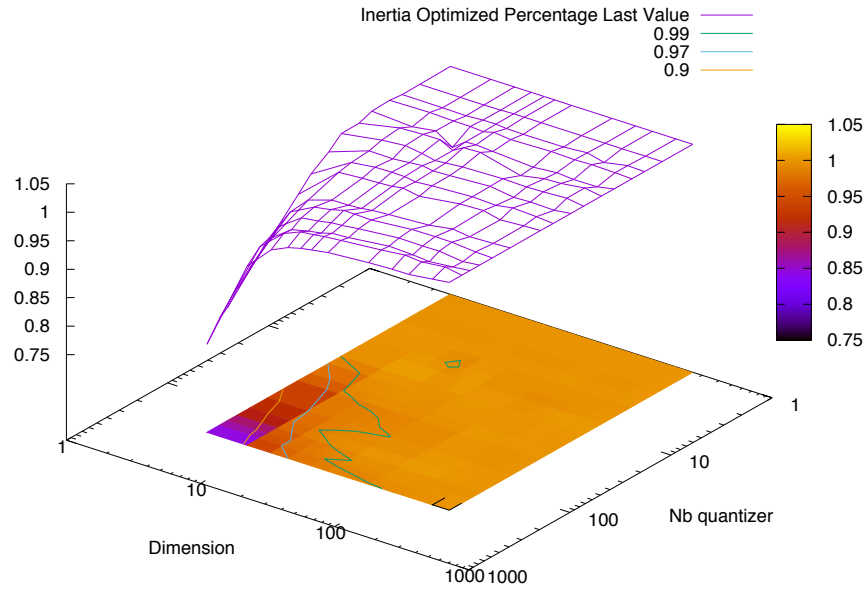


FIGURE 5.9 – $\frac{\underline{\mathcal{D}}_1^{X,2,d} - \underline{\mathcal{D}}_N^{X,2,d}}{\underline{\mathcal{D}}_1^{X,2,d_M} - \underline{\mathcal{D}}_N^{X,2,d_M}}$ with contour line approximating the critical dimension in term of percentage of inertia that can be minimized (PCA of PD_{15}), d_M is the highest dimension with $10E4$ media item.

The estimation of the critical dimension looks very different and seems to be smaller in the Gaussian case.

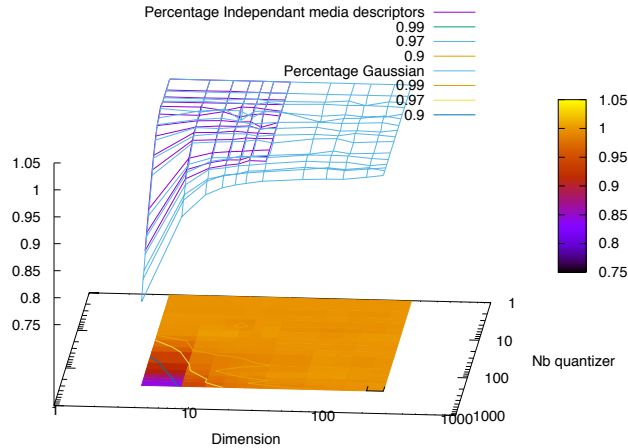


FIGURE 5.10 – Simulation of descriptors with same marginal distribution by dimension but independant

5.3 Hierarchical tree media indexing : non supervised learning

5.3.1 Kmean Tree for media indexing : Hierarchical Search Tree

We study in this section how hierarchical tree are used to index and search in huge dataset of media. We need for this purpose to make a first optimal quantization in a chosen dimension and then index the media through this first index made of a cluster.

Each layer of the tree corresponds to a quantization made locally from a cluster of an higher level in the search tree.

This is a generalization of product quantization in the sense that if we choose one dimension at each step, we will get a product quantization.

It has been already used for others features describing media for example in [5].

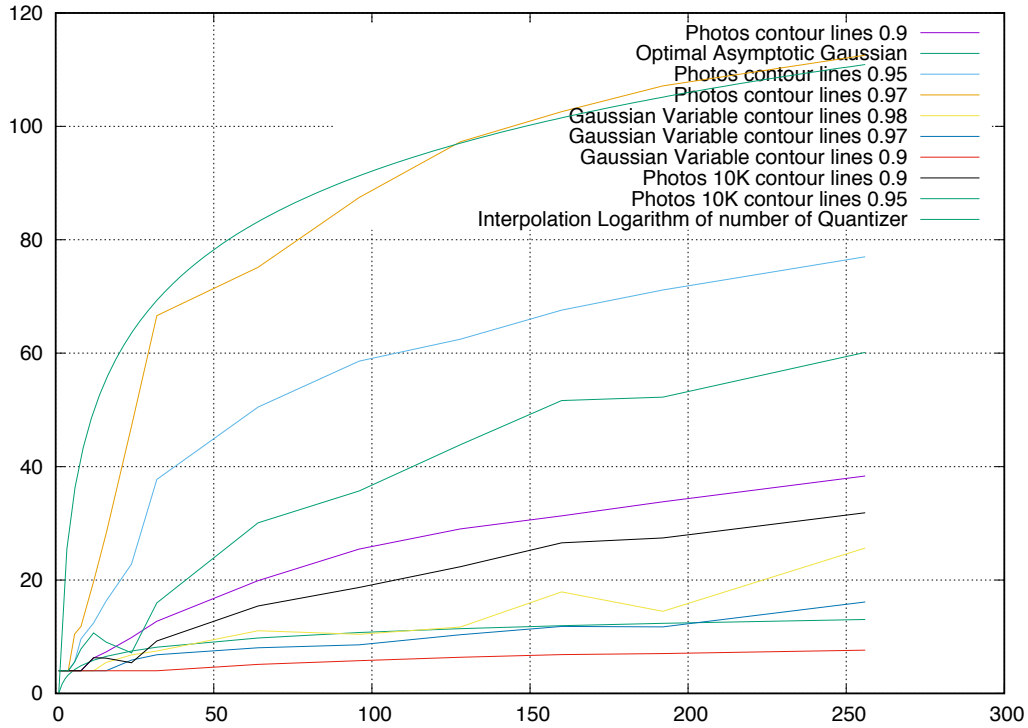


FIGURE 5.11 – Comparison contour lines and optimal asymptotic

5.4 Enhancement of Lloyds algorithm : Split and Stick

5.4.1 Lloyds algorithm

The Lloyds algorithm is a recursive algorithm designed to find an approximation of the optimal quantizer of a random variable.

It is based on the fixed point equation derived from the stationarity property (1.8) that we present in section 1.2.1.1.

It was first published in [10].

We consider realizations of the random variable to quantize. The quantization grid or centroid are initialized randomly among the realizations. At each step the clusters are computed by finding for each realization of the random variable its nearest centroid. The centroids are then updated by taking the mean

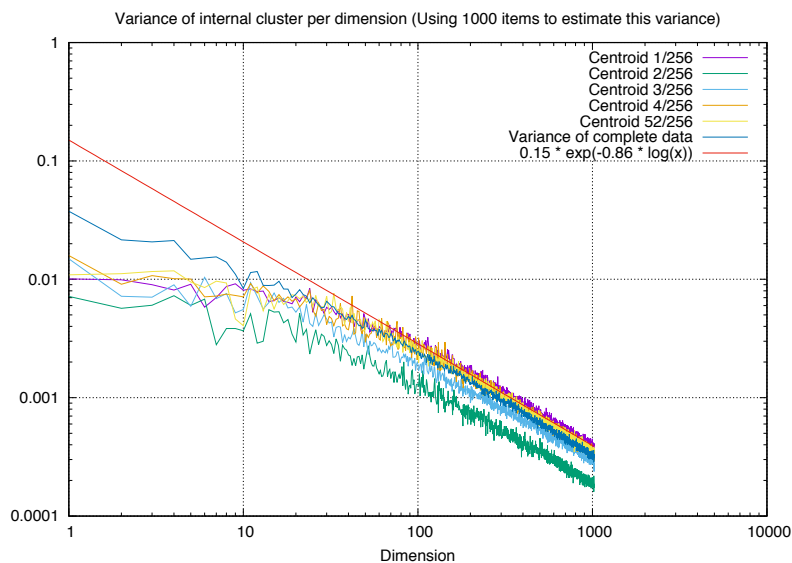


FIGURE 5.12 – Distribution of variance inside one cluster for a quantization with 256 quantizers

of each cluster as its new centroid. The new clusters are designed from these centroids.

We present here Lloyds algorithm in detail as in section 1.2.3.1.

It starts with an initialization followed by an iteration in two steps :

- Items are initialized randomly or from the optimal grid with $N - 1$ items plus one item added : $\Gamma_{(0)} = \{x_{(0)}^1, \dots, x_{(0)}^N\}$. They are called centroids.
- M steps are done, at step k :
 - clusters are recomputed, ie we look for the nearest items for each centroid, with equation (1.6) : $C_1(\Gamma_{(k)}), \dots, C_N(\Gamma_{(k)})$
 - We update centroids thanks the formula $x_{(k+1)}^i = \mathbb{E}[X|C_i(\Gamma_{(k)})]$ for $1 \leq i \leq N$.
- The results consists in the set $\Gamma_{(M)} = \{x_{(M)}^1, \dots, x_{(M)}^N\}$.

We optimize a non convex function, but the gradient descent algorithm ensure that the limit verify the stationnarity property defined in (1.8).

Recent theoretical works from Pagès and Yu [15] or Du and al. [6] give elements on the convergence of this algorithm.

The main idea is at each step the local inertia decreases. Equation (5.1) define this distortion.

5.4.2 Method to split and agglomerate Lloyds : Split and Stick

We propose a way to split and agglomerate clusters obtained through Lloyds algorithm.

Figure 5.13 shows the update of the local inertia for five different steps of the Lloyds algorithm.

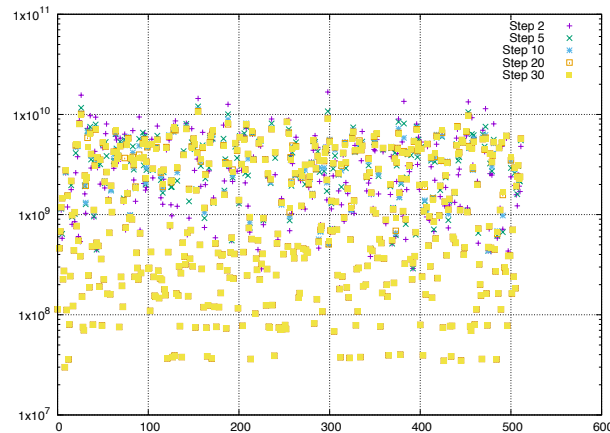


FIGURE 5.13 – Value of local inertia for each cluster and three different steps of a Lloyds algorithm

In abscissa we have the different clusters.

We observe three type of behavior in the sequence of inertia per cluster under the Lloyd algorithm :

- The highest inertia become smaller
- Some of the medium inertia become higher
- The smallest inertia does not change.

It seems that the global structure of the clusters are kept along the step of Lloyd algorithm.

Observing the fact that some clusters were not moving as expected, we proposed two ways to enhance the algorithm, at least during the first iterations.

The splitting procedure : in this case we split the biggest cluster in two clusters, each having approximately half of the points of the original one and

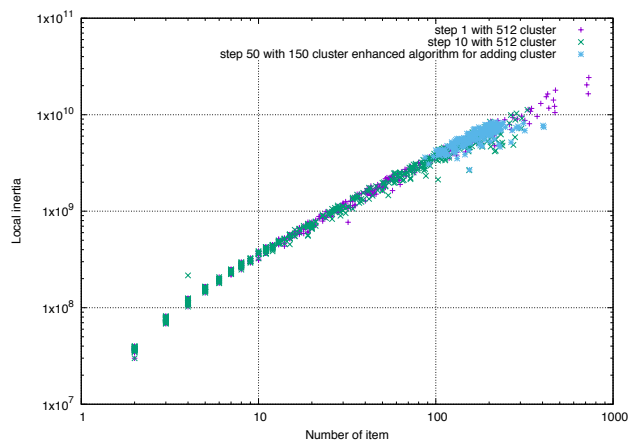


FIGURE 5.14 – Joint size of distribution of cluster (by its number of items) and its local inertia

initialize a new cluster with half of the points of this selected cluster at every steps of the Lloyds algorithm.

This enable us to have clusters with closer local internal inertia. Indeed the ratio between the cluster with highest inertia and the lowest one decreases by some order of magnitude. For more than 100, it becomes less than 10 with the enhanced adding (meaning splitting) procedure.

The second way of enhancing the algorithm is to concentrate on the clusters with the lowest inertia. We simply remove them. Figure 5.15 shows the impact of this enhancement. The items are naturally assigned to the nearest centroid.

When done at the same rate as the splitting procedure, it enables the algorithm to keep the same number of cluster at each step.

The figure 5.16 illustrates the different type of convergence of the global inertia. The global inertia is defined as the sum of the local inertia for each cluster.

We present on figures 5.17 and 5.18 the differences between using split and stick and using classical Lloyds algorithm. For high dimension, our enhancement gives higher convergence rate and in fact enables to leave local minima that are pretty bad. We can also consider that this problem comes from the lack of data. These graphs have been drawn with $1e4$ items. But this behavior comes also from the high dimension considered. We see very unoptimal local minima for small amount of quantizer. On figure 5.18, you can observe in fact that for the highest dimension, the first value in term of quantizer are not optimized at all, in fact Lloyds reach a local minimum where one centroid is very near the barycenter of our data and a few centroids are only singleton. For this highest

5.4. ENHANCEMENT OF LLOYD'S ALGORITHM : SPLIT AND STICK155

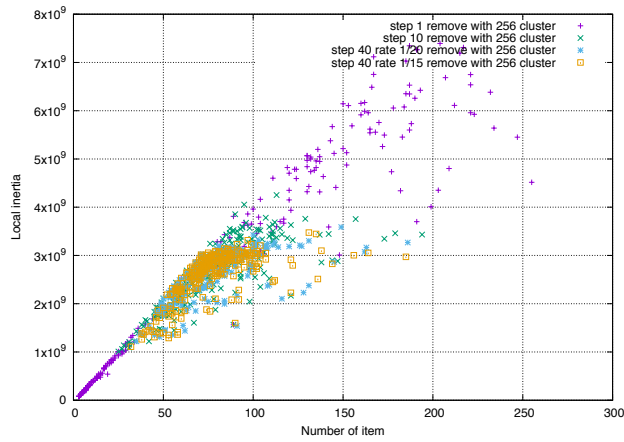


FIGURE 5.15 – Joint size distribution of clusters (by number of items) and its local inertia (split enhancement of Lloyd's algorithm.)

dimension, for high number of quantizer, the value toward it converges are not optimal, since the amount of variance reduced is lower than the one for lower dimension. These local minima are quite different.

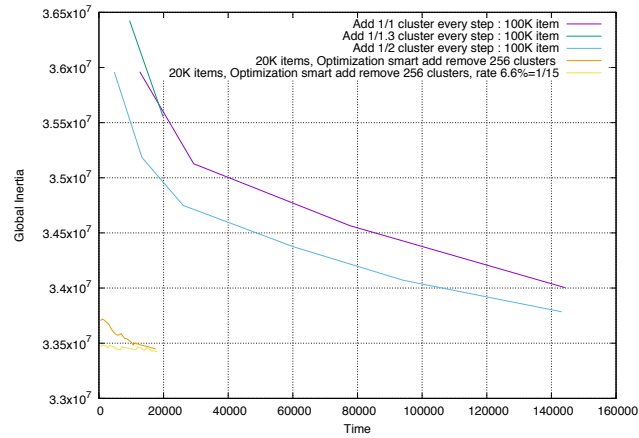


FIGURE 5.16 – Decrease of global inertia for different types of Lloyd procedure

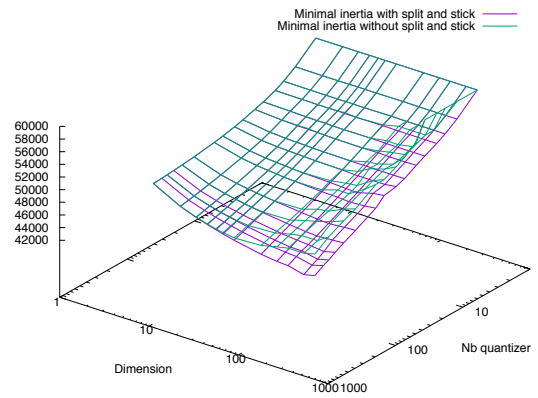


FIGURE 5.17 – Minimal inertia reached through classic Lloyds algorithm and Enhance split and stick method

5.4. ENHANCEMENT OF LLOYDS ALGORITHM : SPLIT AND STICK 157

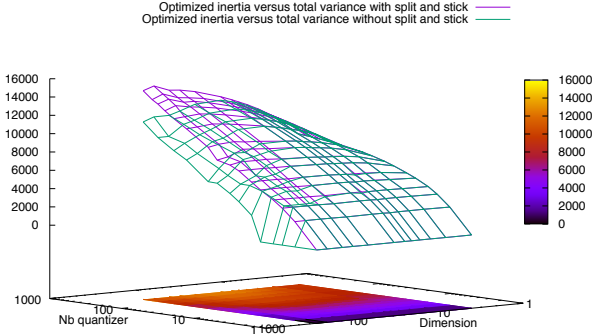


FIGURE 5.18 – Amount of optimized inertia

5.4.3 Others enhancement

5.4.3.1 Triangular Inequality

We also observe the distribution of distances and one of its lower bound obtained thanks to triangular inequality. It does not seem that this method can enhance sensibly our methodology. In fact in high dimension the distribution of distance is denser, and we can rarely eliminate a candidate to nearest neighbor thanks its distance to an already known item. Thus further tests should be done. It must be usable only up to dimension around 10 which is a low dimension in our study.

5.4.3.2 Partial distance search

Partial distance search is a method to avoid a full calculation of distance for the farrest element that use Pythagoras theorem to eliminate these item from a nearest neighbour search. We do not have fully tested this methodology since our parallel computation framework makes it more difficult to use.

5.4.3.3 Others enhancement

We present here different enhancement to nearest neighbor search.

The computation of the distances implies only to compute the scalar product of our data, since we can store the norm of each data and use remarkable identity like $|a - b|^2 = |a|^2 - 2 < a|b > + |b|^2$.

5.5 Numerical validation of approximation and implementation

5.5.1 Description of the space of study

The vector space defining our descriptors depends on the specification of the neural network as well as the dataset used to train it.

It depends also on the post process made on the data.

Here are different case that we have used for some preliminary tests or deeper study.

For the sake of consistency, we will uses the same identifier as in our implementation, they are listed in Table 5.1.

The range (Column 3 of Table 5.1) of value are only integer if not specified otherwise.

5.5.2 Rescaling and truncation of the data

After pca projection, we have also rescaled the data and truncated them on integer between 0 and 255 included. We see that different scale and shift parameters impact the distribution of the eigenvalues. We display in section 5.5.3 the distribution before this rescaling.

ID	Dim.	Proj. Range	DataSet	Specific information	Rescale, shift	b
15	9216	[0-255]	ImageNet Objects	Pool5 layer, Reference Caffe Cropped	10, 0	0.75
2	4096	[0-255]	ImageNet Objects	Pool5 layer, Reference Caffe with trivial compression by summation	10, 0	0.73
1	4096	[0-255]	ImageNet Objects	Fully Connected 7 layer, Reference Caffe	10, 0	1.1
3	4096	[0-255]	ImageNet Objects	Fully Connected 6 layer, Reference Caffe	10, 0	1.1
12	9216	[0-255]	ImageNet Objects and Places	Pool 5 layer, Reference Caffe	10, 0	0.72
13	9216	[0-255]	ImageNet Objects and Places	Pool 5 layer, Reference Caffe cropped	10, 0	0.62
5	4096	[0-255]	ImageNet Objects and Places	Pool 5 layer, Reference Caffe with trivial compression by summation and cropped	10, 0	0.68
8	4096	[0-255]	ImageNet Places	Pool 5 layer, Reference Caffe with trivial compression by summation and cropped	10, 0	
10	9216	[0-1]	ImageNet Objects and Places	Pool 5 layer, Reference Caffe	10, 0	
468	1024	[0-255]	ImageNet Objects	PCA after Type 15 : Main Case Studied	0.05, -1250	0.75

TABLE 5.1 – Parameters defining the different type of descriptors.

We show in figure 5.19 and 5.20, the distribution of cumulative variance, the increasing partial sum of eigenvalues divided by the total variance i.e. the total sum.

We show in figure 5.21 the numerical estimation of the variance of one dimension ie eigenvalues after this rescaling. We detail in section 5.5.1 the shift, rescale and projection on integer done on our data. The shift between the different curve comes simply from the logscale for different scaling parameters. The noise comes from the fact we have not used enough data to compute it, thus we observe the usual trend. We have observed this noise more in detail by using different number of data in figure 5.22. This validates the fact that a small number of data can estimate the general shape of this distribution.

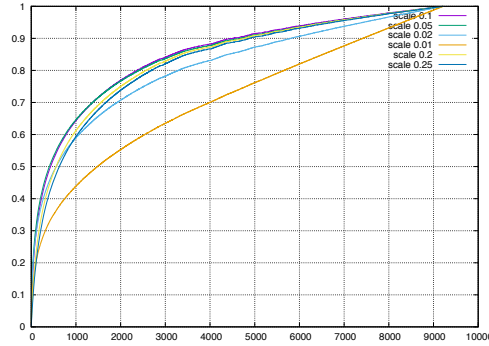


FIGURE 5.19 – $d \mapsto \frac{\mathcal{D}_N^{X,2,d}}{\mathcal{D}_N^{X,2,d_M}}$

Numerical convergence of the sequence of eigenvalues We compare here the distribution of the eigenvalues computed with respectively 50,000 and 200,000 media items, images in our main case studied which is the descriptor of type 15 as explained in Table 5.1.

Figures 5.23 and 5.24 shows the same data with dot or lines. We observe that both estimation of the distribution of the sequence of variance are near one each others. The estimation with 200,000 items was done with a different algorithm to diagonalize the matrix that did not need to store all the vectors representing the different items in the memory at the same time. Otherwise diagonalization would have failed.

We observe more accurately the instability in the figure 5.24 with lines instead of point.

We see that this smarter algorithm in terms of memory management shows instability for the small eigenvalues with two different mode of the regression. But the main mode that is present for the first (and huge majority of the

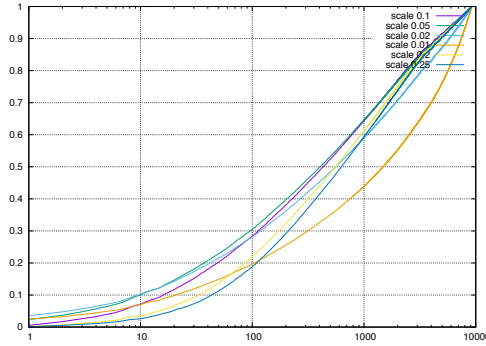


FIGURE 5.20 – $d \mapsto \frac{\mathcal{D}_N^{X,2,d}}{\mathcal{D}_N^{X,2,d_M}}$ in logscale in abscissa

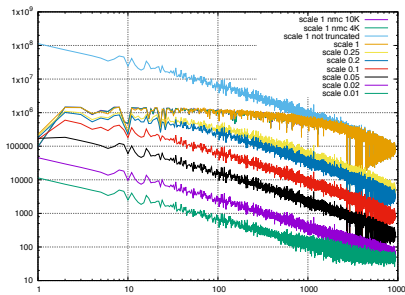


FIGURE 5.21 – Distribution eigen value numerical

eigenvalues) seems to stay up to the last when neglecting the second mode that can be interpreted only as a numerical noises.

Other estimation of this distribution with only a few hundred of items show the same distribution. It is very interesting that we can use a set of size smaller of some order of magnitude to estimate a large number of eigenvalues.

Thanks principal component analysis, we study the distribution of eigenvalues and the impact of the learning set and specification of the neural network to encode media item.

It is intriguing to consider that these sequences are sometime the beginning of a sequence of finite sum and sometime of infinite sum.

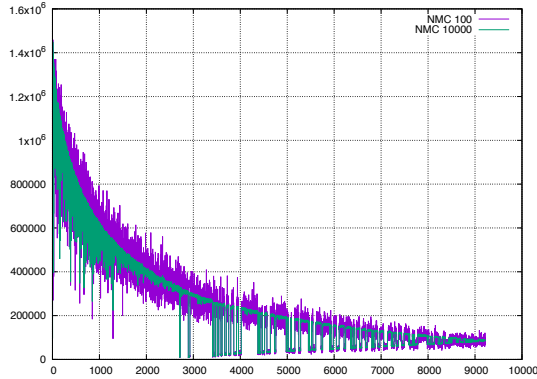


FIGURE 5.22 – Distribution eigen value numerical not logscale

5.5.3 Distribution function and density of the data

To have a better understanding of the data we are using, we display here the density and repartition function of the first dimension after pca projection. This is done before rescaling of the data as explained in section 5.5.2.

Kolmogorov Smirnov tests made on the first coordinate of the data exclude the fact it fits a Gaussian distribution, we display the optimal Gaussian distribution with the distribution of this first dimension in figure 5.26.

5.5.4 Integer versus Float and Projection on integer : Lloyds convergence

We are testing here the impact of using data stored on integers between 0 and 255. For this purpose, without changing our data stored in this space, we observe the impact of projecting the centroid on these clusters.

In this case the approximated Lloyds algorithm does not respect anymore its decrease of inertia at each step, so we need to test numerically.

Figure 5.27 shows the difference of total inertia between the two algorithms. We do not see the difference, we show them on figure 5.28 and they are very small which validate this approximation.

5.5.5 Metric to compare the efficiency of our methodology

We propose here some metrics to compare the efficiency of our methodology of indexing. We show some numerical estimation in a simpler manner in figure 5.29 than the one displayed here.

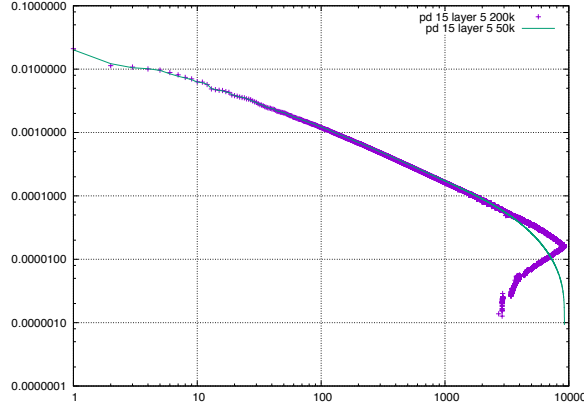


FIGURE 5.23 – Interpolation of eigenvalues for the main case

5.5.5.1 Based on siamese compression or pca compression (with or without kmean compression)

We are interested in approximating the problem of finding the n nearest neighbour to one "query" element in the initial vector space among N datas.

In order to have some information on the efficiency of our compression, we will compute two indicators, the computation time on the one hand and a accuracy measure that we define here.

Definition of the accuracy measure The initial dimension of our raw data, coming from a layer of a neural network will be noted N_1 and is about $10E4$.

We define $E_1 = \mathbb{R}^{N_1}$, the initial vector space, and $E_2 = \mathbb{R}^{N_2}$ after compression.

We note $d_{(1)}$ the Euclidean distance in E_1 .

We note $\Pi^{(1)} : X^{(1)} \in \mathbb{R}^{N_1} \mapsto X^{(2)} \in \mathbb{R}^{N_2}$ the compression function.

We note $\pi_{\Gamma}^{(2)} : X^{(2)} \in \mathbb{R}^{N_2} \mapsto \hat{X}^{(2)} \in \Gamma^{(K_2)} = \{x_1^{(2)}, \dots, x_{K_2}^{(2)}\}$ the function that for an item gives its nearest quantifier.

We note $d^{(2)}$ the Euclidean distance in E_2 and $\hat{d}_k^{(2)}$ the distance linked to k nearest quantifiers. To be precise, in our nearest neighbour problem, we will not consider the element outside the k nearest clusters corresponding to the k nearest quantifier (or centroids). We will start with $k = 1$. We will consider for example that the distance for two elements outside these k nearer centroids is equal to some constant that could be chosen as a fraction of the diameter of our space.

The precision measure will be therefore :

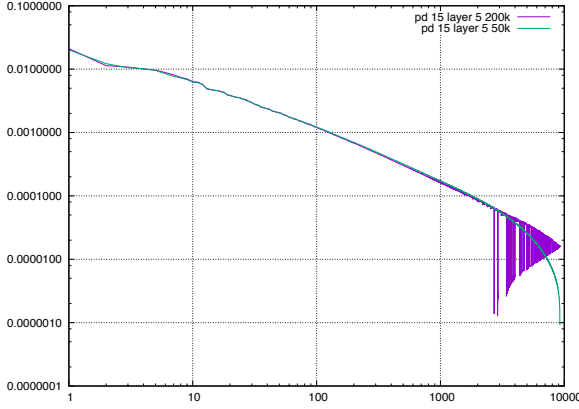


FIGURE 5.24 – Interpolation of eigenvalues for the main case as lines

$$\alpha = \mathbb{E} \left[\left((d^{(1)}(X) - d^{(1)}(Y))^2 - (\hat{d}_k^{(2)}(\Pi^{(1)}(X)) - \hat{d}_k^{(2)}(\Pi^{(1)}(Y)))^2 \right)^2 \mid X \in E_1 \text{ near } Y \in E_1 \right]$$

We still need to define what near means in the above formula. The reason is that we are interested in using Euclidean distance to find images that are the same, not to rate how far a castle is from an orchid.

Definition of the mean computation time We introduce also the computation time. In the initial space with N data, we are interested in finding the n nearest neighbors. Let us call $\theta_{n,N}^{(1)} = \mathbb{E} [\theta_{n,N}^{(1)}(X^{(1)})]$.

We define in the same way : $\hat{\theta}_{n,N,k}^{(2)} = \mathbb{E} [\hat{\theta}_{n,N,k}^{(2)}(X^{(2)})]$

This dimension will be :

$$\beta = \frac{\theta_{n,N}^{(1)}}{\hat{\theta}_{n,N,k}^{(2)}}$$

We will study in this space $[\alpha, \beta]$ the efficiency of our tools.

Remark 8. In the computer vision community, classification problem are usually rated through the probability of having the correct choice as the best element of some methodology to rate, or within the 5 first elements. It is not clear that our measure has the same property, but regarding the problem of having huge amount of photos, we would need at least to find the percentage of correct answers (the n nearer), among for example the $10 * n$ answers of the approximation methodology that we want to rate. At least our measure enables to compare k-mean compression to neural network siamese compression.

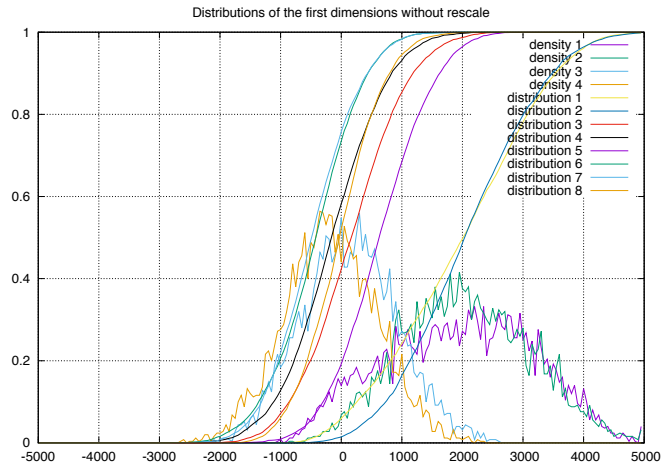


FIGURE 5.25 – distribution

Figure 5.29 shows the mean computation time and mean number of correct results in the 100 nearest. We consider the correct results as being the 100 nearest element computed in the most precise case when we search here in 100 clusters among 256. This test is done on a very small number of simulation, only 25 while searching in 256,000 items split over the 256 clusters. We have used a dimension 1000 as defined for Descriptor Type 468 defined in Table 5.1.

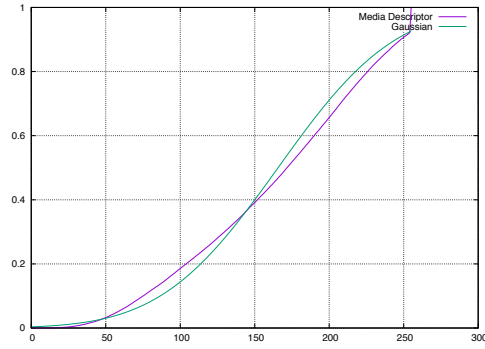


FIGURE 5.26 – distribution

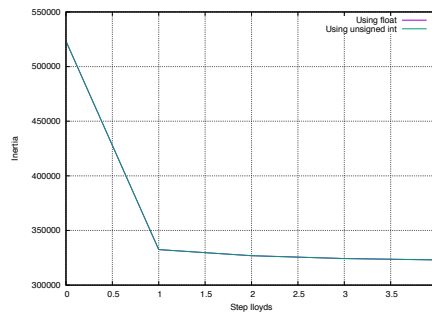


FIGURE 5.27 – Comparison float vs integer

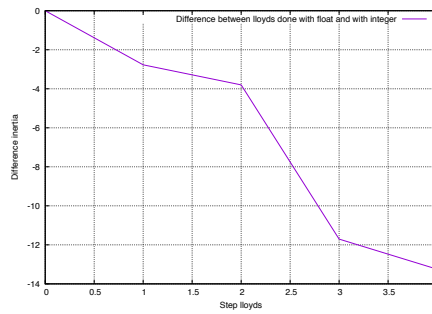


FIGURE 5.28 – Difference float vs integer

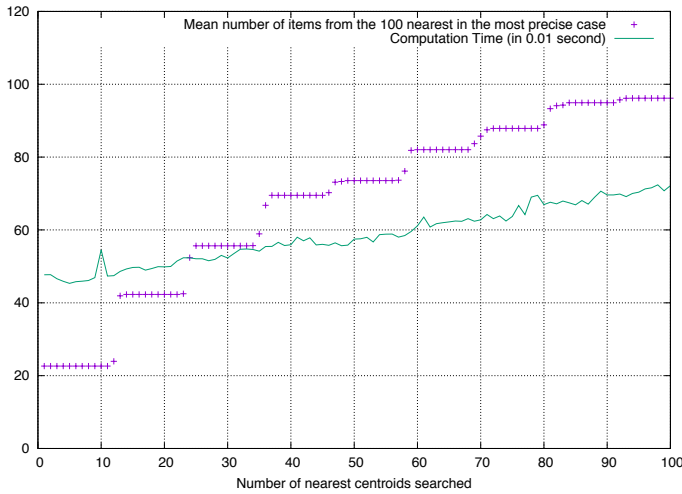


FIGURE 5.29 – Efficiency and computation speed for a search in different number of clusters

5.6 Conclusion

We have studied in details two different steps of indexing database of media. We have provided asymptotical theoretical optimum of our methodology in a toy example which is empirically validated. We have observed some of its bias with the studied datas. We understand the difference comes from non independent although uncorrelated of order 1 properties of our datas.

It is finally clear that the value of the module of finite variation which gives the distribution of variance per dimension does not change drastically the critical dimension. Of course it changes the dimension to reach half of the variance of the distribution or in case of infinite dimension.

We also observed that adding enough data will remove some of the unintuitive behavior observed in high dimension.

It would be of high interest to be able to follow this study while having binary value for our data, instead of quantizing them on 256 values, simply keep 0 or 1.

Anyway we were able to provide efficient tools to compress our data that can reduce drastically search time with nearest neighbor while keeping efficient results.

Our main conjecture is that with dependent data like the one we have, the optimal quantizer can "jump" over the dimension and therefore the way we have searched for the critical dimension is not optimal with these data. First we have observed from figure 5.12 that the variance above the dimension quantized depends on the dimension quantized. We have also observed that when using independent descriptors with the same marginal distribution we have the same behavior as Gaussian distribution like in figure 5.10. This is the difference with the data named Descriptor Type 468 which are the representation of a media.

Bibliography

- [1] Robert M. Gray Allen Gersho. Vector quantization and signal compression. Kluwer Academic Publishers, 1997 (6th) edition, 1992.
- [2] Mathieu Aubry and Bryan C Russell. Understanding deep features with computer-generated imagery. In Proceedings of the IEEE International Conference on Computer Vision, pages 2875–2883, 2015.
- [3] François Chollet. Keras. <https://github.com/fchollet/keras>, 2015.
- [4] J. A. Cuesta and C. Matrán. The strong law of large numbers for k -means and best possible nets of Banach valued random variables. Probab. Theory Related Fields, 78(4) :523–534, 1988.
- [5] Marius Muja David G. Lowe. Fast matching of binary features. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 36(11), November 2014.
- [6] Qiang Du, Maria Emelianenko, and Lili Ju. Convergence of the lloyd algorithm for computing centroidal voronoi tessellations. SIAM journal on numerical analysis, 44(1) :102–119, 2006.
- [7] Siegfried Graf, Harald Luschgy, Gilles Pages, et al. The local quantization behavior of absolutely continuous probabilities. The Annals of Probability, 40(4) :1795–1828, 2012.
- [8] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe : Convolutional architecture for fast feature embedding. arXiv preprint arXiv :1408.5093, 2014.
- [9] Antoine Lejay and Victor Reutenauer. A variance reduction technique using a quantized brownian motion as a control variate. The Journal of Computational Finance, 16(2) :61–84, 2012.
- [10] Stuart Lloyd. Least squares quantization in pcm. IEEE transactions on information theory, 28(2) :129–137, 1982.
- [11] Harald Luschgy and Gilles Pagès. Functional quantization of Gaussian processes. J. Funct. Anal., 196(2) :486–531, 2002.
- [12] Harald Luschgy and Gilles Pagès. Sharp asymptotics of the Kolmogorov entropy for Gaussian measures. J. Funct. Anal., 212(1) :89–120, 2004.

- [13] Harald Luschgy and Gilles Pagès. Functional quantization of a class of brownian diffusions : a constructive approach. Stochastic Processes and their Applications, 116(2) :310–336, 2006.
- [14] Gilles Pagès, Huyên Pham, and Jacques Printems. An optimal markovian quantization algorithm for multi-dimensional stochastic control problems. Stochastics and Dynamics, 4(04) :501–545, 2004.
- [15] Gilles Pagès and Jun Yu. Pointwise convergence of the lloyd algorithm in higher dimension. SIAM J. Control Optim., 54(5) :2354–2382, 2016.
- [16] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV), 115(3) :211–252, 2015.

Bibliographie des Résumés/Bibliography of abstracts

- [BPR06] A. Beskos, O. Papaspiliopoulos, and G. O. Roberts. Retrospective exact simulation of diffusion sample paths with applications. *Bernoulli*, 12(6) :1077–1098, 2006.
- [LR12] Antoine Lejay and Victor Reutenauer. A variance reduction technique using a quantized brownian motion as a control variate. *The Journal of Computational Finance*, 16(2) : 61–84, 2012.
- [LP04] Harald Luschgy and Gilles Pagès. Sharp asymptotics of the Kolmogorov entropy for Gaussian measures. *J. Funct. Anal.*, 212(1) :89–120, 2004.
- [MRTT16] Christophe Michel, Victor Reutenauer, Denis Talay and Etienne Tanré, Liquidity costs : a new numerical methodology and an empirical study. *Applied Mathematical Finance*, 23(1) : 57-79, 2016.
- [RT16] Victor Reutenauer and Etienne Tanré, An unbiased Monte Carlo estimator for derivatives. Application to CIR., [Preprint](#), 2016.

Résumé & Abstract

Résumé en français

Cette thèse s'intéresse à différents problèmes de contrôle et d'optimisation dont il n'existe à ce jour que des solutions approchées. D'une part nous nous intéressons à des techniques visant à réduire ou supprimer les approximations pour obtenir des solutions plus précises voire exactes. D'autre part nous développons de nouvelles méthodes d'approximation pour traiter plus rapidement des problèmes à plus grande échelle.

Nous étudions des méthodes numériques de simulation d'équation différentielle stochastique et d'amélioration de calculs d'espérance. Nous mettons en œuvre des techniques de type quantification pour la construction de variables de contrôle ainsi que la méthode de gradient stochastique pour la résolution de problèmes de contrôle stochastique. Nous nous intéressons aussi aux méthodes de clustering liées à la quantification, ainsi qu'à la compression d'information par réseaux neuronaux.

Les problèmes étudiés sont issus non seulement de motivations financières, comme le contrôle stochastique pour la couverture d'option en marché incomplet mais aussi du traitement des grandes bases de données de medias communément appelé Big Data dans le chapitre 5. Théoriquement, nous proposons différentes majorations de la convergence des méthodes numériques, d'une part pour la recherche d'une stratégie optimale de couverture en marché incomplet dans le chapitre 3, d'autre part pour l'extension à la technique de Beskos-Roberts de simulation d'équation différentielle dans le chapitre 4. Nous présentons une utilisation originale de la décomposition de Karhunen–Loève pour une réduction de variance de l'estimateur d'espérance dans le chapitre 2.

Mots-clefs

Problème d'optimisation, Contrôle stochastique, Quantification, Marché incomplet, Indexation d'images, Réduction de variance, Volatilité stochastique, Sensibilité par Malliavin, Simulation trajectorielle exacte, Apprentissage automatique

Abstract in english : Stochastic Algorithms for risk management and indexing of database of Media

This thesis proposes different problems of stochastic control and optimization that can be solved only thanks approximation. On one hand, we develop methodology aiming to reduce or suppress approximations to obtain more accurate solutions or sometime exact ones. On another hand we develop new approximation methodology in order to solve quicker larger scale problems.

We study numerical methodology to simulate stochastic differential equations and enhancement of computation of expectations. We develop quantization methodology to build control variate and gradient stochastic methods to solve stochastic control problems. We are also interested in clustering methods linked to quantization, and principal component analysis or compression of data thanks neural networks.

We study problems motivated by mathematical finance, like stochastic control for the hedging of derivatives in incomplete market but also to manage huge databases of media commonly known as Big Data in chapter 5.

Theoretically we propose some upper bound for convergence of the numerical method used. This is the case of optimal hedging in incomplete market in chapter 3 but also of an extension of Beskos Roberts methods of exact simulation of stochastic differential equations in chapter 4.

We present an original application of Karhunen-Loève decomposition for a control variate of computation of expectation in chapter 2.

Key Word

Optimization Problem, Control Stochastic, Quantization, Incomplete Market, Media Indexing, Variance reduction, Stochastic volatility, Malliavin Sensitivity, Exact Simulation, Machine Learning