



HAL
open science

Deep learning, pricing et calibration du modèle de Heston

Liu Qian

► **To cite this version:**

Liu Qian. Deep learning, pricing et calibration du modèle de Heston. Mathématiques [math]. 2022. dumas-03945488

HAL Id: dumas-03945488

<https://dumas.ccsd.cnrs.fr/dumas-03945488v1>

Submitted on 7 Jun 2023

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.



Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0 International License



Rapport de stage M1 (2A ENSIIE)

Deep learning, Pricing et Calibration du modèle de Heston

Qian LIU

Stage effectué du 23/05/2022 au 12/08/2022

Tuteurs de stage : M. Mohamed BEN ALAYA & M. Nicolas VERGNE

Enseignant référent : M. Abass SAGNA

Établissement d'enseignement : Université Paris-Saclay / ENSIIE - M1 Maths Appliquées

Entreprise d'accueil : LMRS - Av. de l'Université, BP.12, 76801 Saint-Étienne-du-Rouvray



Remerciements

Je tiens tout d'abord à me remercier d'avoir accompli ce stage sans abandon.

Je remercie M. Ionut DANAILA, Directeur du Laboratoire de Mathématiques Raphaël Salem (LMRS), de m'avoir accueillie comme stagiaire au sein de son laboratoire.

Merci également à M. Mohamed BEN ALAYA, mon tuteur de stage, Professeur d'Université et Responsable de l'équipe statistique du LMRS, pour ses conseils avisés et la confiance qu'il m'a accordée tout au long de mon stage.

Je tiens ensuite à remercier vivement et particulièrement M. Nicolas VERGNE, mon co-encadrant de stage, Maître de Conférence, pour l'attention et l'aide qu'il m'a apporté au quotidien pendant mon stage au sein du LMRS, surtout pour la correction du rapport.

Je tiens aussi à remercier l'Université Paris-Saclay et l'ENSIIE de m'avoir offert l'opportunité d'effectuer ce stage, et particulièrement M. Abass SAGNA d'être mon superviseur académique et pour m'avoir écoutée et accompagnée pédagogiquement.

Je n'oublie pas non plus mes proches qui m'ont sans cesse soutenue dans l'élaboration de mon projet professionnel et m'ont aidée à chaque étape de ma vie.

Enfin, un grand merci à l'ensemble du personnel du LMRS pour leur accueil et leur disponibilité pendant la durée de mon stage.

Toutes ces personnes ont contribué, par leur disponibilité et leur bonne humeur, à rendre mon stage enrichissant et motivant.

Résumé

En finance, la calibration d'un modèle financier joue un rôle important pour la gestion des risques et pour les produits dérivés, et nécessite d'être effectuée rapidement et avec précision. Un algorithme d'optimisation utilisé avec un réseau de neurones artificiels (RNA) permet de réaliser cet objectif. Pour ce faire, j'ai d'abord choisi une méthode classique de pricing pour générer des données d'apprentissage, puis entraîné un RNA pour remplacer la méthode de pricing. Le principe du RNA est aussi présenté dans ce rapport. La calibration s'est faite à l'aide de l'algorithme d'évolution différentielle, qui cherche le minimum global d'erreur entre la volatilité implicite (VI) prédite par le RNA et la VI observée sur le marché financier. Mon travail a permis à l'équipe de statistique du LMRS d'avoir un répertoire des méthodes de pricing et de calibration, de s'initier au machine learning (plus particulièrement au deep learning) et d'apprendre la construction des RNA. Les résultats motivent une étude plus approfondie des méthodes d'apprentissage du RNA afin d'améliorer les performances en terme de temps de calcul et de précision.

Mots clés : modèle financier, modèle de Black-Sholes, modèle de Heston, calibration, réseaux de neurones artificiels (RNA), machine learning, deep learning, optimisation.

Table des matières

1	Introduction	1
2	Environnement, contexte et objectifs du stage	2
2.1	Le secteur : secteur public de la recherche et de l'enseignement supérieur . . .	2
2.2	Le laboratoire LMRS	2
2.3	Contexte et problématique générale	3
2.4	Objectifs du stage	4
3	Travail réalisé	5
3.1	L'axe Finance	5
3.1.1	Notions de base	5
3.1.2	Méthodes de pricing	6
3.1.2.1	Modèle de Black-Sholes	6
3.1.2.2	Modèle de Heston	13
3.1.3	Méthodes de calibration	19
3.2	L'axe Machine Learning	21
3.2.1	Panorama du machine learning	22
3.2.2	Principe du RNA	23
3.2.3	Structure et conception des RNA pour le pricing	26
3.2.3.1	Étape forward	26
3.2.3.2	Étape backward : évolution différentielle (ED)	28
4	Résultats numériques et discussion	30
4.1	Méthodes de pricing	30
4.2	Étape forward	33
4.3	Étape backward	35
5	Conclusion et perspectives	38
	Glossaire	43
A	Annexe DD&RS	46
A.1	Développement durable	46
A.1.1	Les pressions environnementales	46
A.1.2	Politique, stratégie et communication sur les enjeux environnementaux	48
A.2	Responsabilité sociétale	50

1 Introduction

Du 23/05/2022 au 12/08/2022, j'ai effectué un stage de recherche au sein du Laboratoire de Mathématiques Raphaël Salem (LMRS). Le LMRS, situé à l'UFR des Sciences et Techniques, Université de Rouen Normandie, Avenue de l'Université, BP.12, 76801 Saint-Étienne-du-Rouvray, en France, est associé au Centre National de la Recherche Scientifique (CNRS) depuis 1979. Pendant mon stage, j'ai été co-encadrée par M. Mohamed BEN ALAYA (Professeur d'Université et responsable de l'équipe statistique du LMRS) et M. Nicolas VERGNE (Maître de Conférence et membre de l'équipe statistique du LMRS), avec qui j'ai pu me former dans d'excellentes conditions.

Dans un premier lieu, mon stage consiste en la lecture de livres et articles scientifiques amenant à la compréhension des modèles financiers de Black-Scholes et de Heston. Ensuite il porte sur l'étude des méthodes de [pricing](#) et de [calibration](#) de ces modèles et enfin sur la construction d'un réseau de neurones artificiels (RNA ou ANN en anglais pour Artificial Neural Network) pour calibrer ces modèles.

Mon stage a pour but, d'un côté, de m'aider à découvrir la vie de chercheuse, de comprendre des modèles financiers et d'apprendre la construction de RNA en python. D'un autre côté, il s'agit d'initier l'équipe de statistique du LMRS, à une nouvelle thématique de recherche autour de l'intelligence artificielle, du machine learning, du deep learning et des RNA, avec en particulier des applications dans le domaine de la finance. Plus précisément, en plus de la compréhension des méthodes d'apprentissage automatique, la maîtrise de la construction d'un RNA sera aussi acquise par le laboratoire.

Du point de vue théorique, des livres et des articles sont sélectionnés comme références, pour la compréhension du modèle de Black-Scholes, du modèle de Heston et des méthodes de deep learning. Du point de vue pratique, le langage de programmation `python`, l'interface interactive `Jupyter` et les bibliothèques disponibles en python, comme `torch` de *pyTorch* et `keras` dans *Tensorflow*, sont utilisés. Du matériel informatique, un MacBook Pro (Retina, 15-inch, Mid 2015) 2,8 GHz Intel Core i7 et un GPU virtuel sont mis à ma disposition.

Le reste du rapport est organisé comme suit. Une présentation de l'entreprise, du contexte et des objectifs du stage sont détaillés en section [2](#). Les méthodes de pricing, de calibration des modèles financiers et de RNA sont présentées en section [3](#). Dans cette section, mes contributions en rapport avec un RNA et les méthodes de pricing sont décrites. Tous les résultats numériques sont présentés en section [4](#). Finalement, une conclusion et des discussions sur les résultats obtenus sont présentées en section [5](#).

2 Environnement, contexte et objectifs du stage

2.1 Le secteur : secteur public de la recherche et de l'enseignement supérieur

Le secteur public de la recherche regroupe les entités qui exécutent ou financent des travaux de recherche et de développement expérimental (R&D) pour l'État, pour les établissements de l'enseignement supérieur ou pour les institutions sans but lucratif (associations et fondations). Le LMRS est sous la tutelle des deux organismes suivants :

Université de Rouen-Normandie L'Université de Rouen Normandie (auparavant Université de Rouen), est une université internationale française pluridisciplinaire, basée à Rouen en Normandie.

CNRS Le Centre National de la Recherche Scientifique (CNRS) est le plus grand Établissement Public français à caractère Scientifique et Technologique (EPST), placé sous la tutelle administrative du ministère de l'Enseignement supérieur, de la Recherche et de l'Innovation.

2.2 Le laboratoire LMRS

LMRS - UMR 6085 Le Laboratoire de Mathématiques Raphaël Salem (LMRS) est un laboratoire de mathématiques pures et appliquées, localisé à UFR des Sciences et Techniques, Université de Rouen Normandie Avenue de l'Université, BP.12, 76801 Saint-Étienne-du-Rouvray, France. Associé au CNRS depuis 1979, il est une Unité Mixte de Recherche (UMR 6085), sous la double tutelle de l'Université de Rouen Normandie et du CNRS. Il est rattaché à l'Institut National des Sciences Mathématiques et de leurs Interactions (INSMI) et fait partie de la Fédération Normandie-Mathématiques.

Services et ses équipes Le LMRS est dirigé par M. Ionut DANAILA et est structuré en trois équipes :

- **Équations aux dérivées partielles (EDP) et calcul scientifique :**
Analyse des équations aux dérivées partielles, applications en calcul scientifique, calcul haute-performance, parallélisation et optimisation de code ;
- **Probabilités et systèmes dynamiques :**
Mécanique statistique et modèles spatiaux aléatoires, systèmes dynamiques et théorèmes limites, analyse stochastique, éléments de topologie et d'algèbre ;
- **Statistique :**
Statistique des processus, processus de Lévy, méthodes de Monte-Carlo, statistique non-paramétrique, théorie de l'assurance et de la finance, contrôle optimal stochastique,

modèles markoviens et semi-markoviens, statistique du génome.

Durant mon stage, j'étais rattachée à l'équipe de Statistique, dans le cadre de la thèse CIFRE de Djibril SARR, salarié-doctorant à la société FBH ASSOCIES et inscrit à l'université Sorbonne Paris Nord, co-encadré par M. Ahmed KEBAIER de l'Université d'Évry et M. Mohamed BEN ALAYA. Le titre de la thèse est « Modélisation et calcul de risque de défaut par des techniques d'intelligence artificielle et de Monte Carlo imbriqués ». Ce stage a permis à l'équipe de statistique du LMRS de s'initier à une nouvelle thématique de recherche autour de l'intelligence artificielle, machine learning, deep learning et RNA. Au sein de l'équipe, la thématique de recherche « Théorie de l'assurance et de la finance », est en relation avec le sujet de mon stage. M. Mohamed BEN ALAYA assure la responsabilité et l'animation de cette équipe et il est aussi mon encadrant de stage.

2.3 Contexte et problématique générale

Ce stage s'inscrit dans le contexte de la modélisation mathématique de phénomènes financiers. La modélisation financière a une place importante dans le domaine de la finance d'entreprise et de marché. En particulier, on s'intéresse à la finance de marché, et surtout à la calibration de modèles financiers.

Dans le cadre du pricing et de la gestion des risques des contrats financiers, la calibration d'un modèle d'actifs consiste à retrouver les paramètres du modèle des équations différentielles stochastiques (EDS) de sous-jacent, à partir des données observées du marché. En d'autres termes, dans le cas des actions et des options financières, la calibration vise à déterminer les paramètres du modèle d'actions de sorte que les prix des différentes options puissent être retrouvés par le modèle mathématique. Cela nécessite souvent un algorithme d'optimisation minimisant la distance entre le prix du marché et le prix simulé par le modèle. Les modèles d'actifs calibrés sont ensuite utilisés soit pour déterminer un prix approprié pour les produits dérivés financiers exotiques, soit à des fins de couverture et de gestion des risques. Cette procédure est donc principalement destinée aux gestionnaires de risques, aux analystes quantitatifs (quant en anglais) et aux commerçants d'options (option traders).

On peut considérer qu'il y a deux étapes pour effectuer la calibration : simuler les prix d'options du marché, et retrouver les paramètres du modèle financier par des moyens adaptés (optimisations, inversion de la fonction objective, etc.). Pour l'étape de pricing, en général on privilégie la méthode de Monte-Carlo, le modèle trinomial ou bien la résolution d'équations différentielles partielles (EDP) par la méthode des éléments finis pour trouver le prix d'options américaines ou exotiques par le modèle de Black-Scholes. Cependant, après avoir bien entraîné un RNA, il peut accélérer le pricing voire remplacer les méthodes traditionnelles.

Un cadre de calibration générique et robuste peut être basé sur une technique d'optimisation globale en combinaison avec une méthode de pricing hautement efficace. Le cadre basé sur un RNA, proposé par [Liu+19], qui est capable de calibrer les paramètres des modèles de volatilité stochastique de grande dimension de manière efficace et précise, sera reconstruit dans ce stage.

La recherche des méthodes de calibration avec les RNA n'a pas commencé dès la naissance

de la méthode d'apprentissage par le perceptron [Ros58], mais avec le développement de logiciels et de la technologie. C'est en 2016 que l'utilisation des RNA pour la calibration est devenue à la mode. Cependant, malgré tous les articles publiés, sa popularité dans le monde des praticiens financiers commence seulement à être établie, principalement à cause des imprécisions des résultats. Ce thème de recherche est donc aujourd'hui un thème novateur et plein de promesses, auquel le LMRS souhaite se former. Calibrer avec des méthodes de machine learning, est en effet une nouvelle thématique pour le LMRS. Cette thématique permettra d'envisager une coopération entre les membres du département d'informatique et de celui de mathématiques.

2.4 Objectifs du stage

Le stage a pour l'objectif d'initier l'équipe de statistique du LMRS à une nouvelle thématique de recherche autour de l'intelligence artificielle, du machine learning et des réseaux de neurones (RNA), avec en particulier des applications dans le domaine de la finance. Il comporte principalement deux parties : une partie théorique contenant un répertoire des différentes méthodes de pricing pour les modèles financiers de Black-Scholes et de Heston et une partie pratique consistant à implémenter les méthodes de pricing en codes `python` afin de les comparer et à construire un RNA de calibration afin d'apprendre à l'équipe la construction de RNA.

Plus concrètement, il s'agit de :

- Comprendre les formules de pricing dans les modèles de Black-Scholes et de Heston.
- Comprendre les méthodes de calibration couramment utilisées.
- Introduire les techniques de deep-learning.
- Mettre en place des données d'apprentissage, des jeux de tests, construire et entraîner un RNA pour la calibration.
- Savoir déterminer les hyperparamètres appropriés de RNA.
- Comparer objectivement les résultats obtenus des différentes méthodes (en termes de précision, temps de calcul, stabilité etc.).

3 Travail réalisé

Mon travail porte principalement sur deux axes : l'axe finance et l'axe machine learning. L'axe finance contient la compréhension et le répertoriage des méthodes de pricing, pour le modèle de Black-Sholes sans dividendes et le modèle de Heston sans dividendes, puis l'implémentation en codes python pour comparer leur performance. L'axe machine learning consiste à présenter à l'équipe statistique du LMRS les méthodes d'apprentissage automatique, la construction de RNA et à proposer une structure de RNA pour effectuer la calibration.

Dans cette section, je présente les concepts théoriques et les outils de programmation à disposition. Ces concepts et notions théoriques aident à comprendre les modèles financiers, leurs méthodes de pricing et de calibration, le machine learning et le principe de RNA. Ensuite la conception du RNA de calibration est détaillée. Les résultats numériques sont présentés dans la section 4.

3.1 L'axe Finance

3.1.1 Notions de base

Options européennes

Une option financière est un produit dérivé, contrat entre deux parties, qui donne à l'acheteur le droit (le vendeur est en revanche tenu de se plier à la décision de l'acheteur) :

- d'acheter (option d'achat, appelée aussi **call**),
- ou de vendre (option de vente, appelée aussi **put**),

avec un prix d'exercice (**strike**) connu d'avance K et une date d'échéance T .

Le résultat d'une option à son échéance (appelé couramment **payoff**) - i.e. ce que va toucher son détenteur - ne dépend que du prix du sous-jacent. On note S_t le prix du sous-jacent à l'instant t . Supposons que le gain des options européennes à l'instant T soit $\Psi(S_T)$. Le payoff d'un call européen est donc

$$\Psi_{\text{call}}(S_T) = (S_T - K)^+ = \max(S_T - K, 0),$$

et celui d'un put européen est

$$\Psi_{\text{put}}(S_T) = (K - S_T)^+ = \max(K - S_T, 0),$$

avec $\Psi \in \{\Psi_{\text{call}}, \Psi_{\text{put}}\}$ et la valeur de l'option au temps $t \in [0, T]$:

$$v(t, S) = e^{-r(T-t)} E^{\mathbb{Q}}[\Psi(S_T) | S_t = S].$$

La probabilité martingale \mathbb{Q} , dite aussi probabilité risque neutre (voir section 3.1.2.1), est une probabilité sous laquelle $e^{-rt} S_t$ est une martingale.

Parité put-call

La parité put-call (Put-Call Parity) définit une relation entre le prix d'un call et celui d'un put, qui ont tous le même strike et la même échéance (**maturité** ou maturity en anglais). La

formule suppose que les options ne sont pas exercées avant échéance (dans le cas des options européennes, il ne peut en être autrement).

Elle s'énonce de la manière suivante :

$$C_t - P_t = S_t - KE^{\mathbb{Q}} \left(\frac{1}{S_T^0} \middle| \mathcal{F}_t \right) = S_t - Ke^{-r(T-t)}$$

où

- C_t : la valeur du call à l'instant t ;
- P_t : la valeur du put à l'instant t ;
- S_t : la valeur de l'actif sous-jacent à l'instant t ;
- S_T^0 : la valeur de l'actif sans risque à l'instant T ;
- \mathcal{F}_t : la filtration naturelle d'un mouvement brownien défini sur un espace probabilisé (voir probabilité risque neutre en section 3.1.2.1) ;
- K : le strike ;
- r : le taux d'intérêt sans risque (constant dans les modèles considérés).

En l'absence d'opportunités d'arbitrage, lorsque trois termes de l'équation sont connus parmi les quatre (call, put, obligation, valeur du sous-jacent), la formule permet de calculer la valeur implicite du quatrième.

Dans ce rapport, le call européen est appelé call. Dans la suite de ce rapport, sans précision, on ne s'intéresse qu'au prix du call européen. Sachant le prix d'un call, le prix du put du même strike et de la même maturité peut se calculer par la parité put-call.

3.1.2 Méthodes de pricing

Une méthode de pricing est une méthode pour évaluer le prix d'une option (call ou put), en utilisant des modèles mathématiques financiers. Dans cette partie, après une présentation des modèles financiers (modèle de Black-Scholes et Heston sans dividendes), les méthodes de pricing pour ces modèles sont présentées. Les paramètres de ces modèles, qui doivent être calibrés à l'aide de l'une ou l'autre des différentes méthodes de calibration (voir section 3.1.3), sont également listés.

3.1.2.1 Modèle de Black-Scholes

Le modèle de Black-Scholes ou modèle de Black-Scholes-Merton [BS73] est un modèle mathématique du marché avec une volatilité de l'actif constante, dans lequel le prix de l'action est un processus stochastique en temps continu, par opposition au « modèle binomial » aussi nommé « modèle Cox Ross-Rubinstein » [CRR79] qui suit un processus stochastique en temps discret. Une introduction au calcul stochastique, des explications et des démonstrations détaillées du modèle de Black-Scholes se trouvent dans [LL12].

Hypothèses du modèle Le modèle de Black-Scholes repose sur un certain nombre de conditions :

1. le prix de l'actif sous-jacent S_t suit un mouvement brownien géométrique avec une volatilité σ constante et un taux de dérive annualisé (rendement ou drift en anglais) μ constant ;
2. il n'y a pas d'occasions d'arbitrage ;

3. le temps est continu ;
4. les marchés sont parfaits :
 - le marché est fluide de liquide, c'est-à-dire que l'on peut décider d'acheter ou de vendre à tout instant (pas de risque de liquidité) ;
 - tous les actifs sont divisibles à l'infini (on peut par exemple acheter $1/100^e$ d'action) ;
 - il n'y a pas de coût de transactions ;
 - on peut emprunter et prêter au même taux ;
 - on peut vendre à découvert sans pénalité ;
 - pas de versement de dividendes.

Le marché parfait est une notion d'économiste. Les mathématiciens ont redéfini celle-ci en « marché complet », avec une définition plus simple : Un marché est dit complet si tous les actifs sont répliquables.

Le modèle, par ses hypothèses, ne correspond pas à la réalité des marchés financiers. Mais l'utilisation de ce modèle se généralise au concept « d'imitation rationnelle ». Il permet d'avoir un moyen de modéliser les changements du marché financier, et peut aider à la prise de décision.

Présentation du modèle Dans ce modèle, on suppose que S (prix de l'action) est un mouvement brownien géométrique. Plus précisément, sous la probabilité historique \mathbb{P} d'un espace de probabilité $(\Omega, \mathcal{F}, \mathbb{P})$, la dynamique de S est de la forme

$$dS_t = \mu S_t dt + \sigma S_t dB_t, \quad (1)$$

où

- t : temps, en années ;
- S_t : prix de l'actif sous-jacent S à l'instant t ;
- B_t : un mouvement brownien à l'instant t ;
- μ : drift de S ;
- σ : la volatilité (constante) du sous-jacent.

Notons que σ est le seul paramètre du modèle. D'après un simple calcul stochastique d'intégration, avec la formule d'Itô pour (1), on trouve qu'il existe une représentation explicite de S à chaque instant, notée

$$S_t = S_0 \exp \left(\sigma B_t + \left(\mu - \frac{1}{2} \sigma^2 \right) t \right).$$

On observe aussi que, puisque $B_t \sim \mathcal{N}(0, t)$, S_t a une distribution log-normale, telle que $\ln S_t$ est normale avec une moyenne $\left(\ln S_0 + \left(\mu - \frac{1}{2} \sigma^2 \right) t \right)$, et une variance $\sigma^2 t$.

Probabilité risque neutre Sur un espace de probabilité $(\Omega, \mathcal{F}, \mathbb{P})$, soit $(B_t)_{t \geq 0}$ un mouvement brownien de filtration naturelle $(F_t = \sigma((B_s)_{s \in [0, t]}))_{t \geq 0}$. On suppose que le prix S_t vérifie l'équation du modèle de Black-Scholes (1) et on considère

- r : le taux d'intérêt sans risque d'un emprunteur sûr dont la rentabilité est certaine sur une période donnée ;
- un actif sans risque de cours $S_t^0 = e^{rt}$;
- sous probabilité historique, un actif risqué de cours $S_t = S_0 e^{\sigma B_t + \left(\mu - \frac{\sigma^2}{2} \right) t}$;

- une option européenne de maturité T et de fonction de payoff $f : \mathbb{R}_+^* \rightarrow \mathbb{R}_+$ i.e. une option qui rapporte $f(S_T)$ à son acheteur en T . La fonction de payoff est supposée continue et à croissance polynomiale i.e. telle que $\sup_{y>0} \frac{f(y)}{1+y^n} < \infty$ pour un entier $n \in \mathbb{N}$.

D'après le théorème de Girsanov, pour $\alpha = \frac{\mu-r}{\sigma}$, il existe une probabilité, notée \mathbb{Q} et appelée probabilité risque neutre, de densité $\frac{d\mathbb{Q}}{d\mathbb{P}} = e^{-\alpha B_T - \frac{\alpha^2}{2}T}$ par rapport à la probabilité historique \mathbb{P} , sous laquelle $(W_t = B_t + \frac{\mu-r}{\sigma}t)_{t \in [0, T]}$ est un mouvement brownien. Pour $t \in [0, T]$, on a $F_t = \sigma((W_s)_{s \in [0, t]})$.

Notons que le prix actualisé de l'actif risqué \widetilde{S}_t , sous la probabilité \mathbb{Q} , vérifie

$$\begin{aligned} d\widetilde{S}_t &= \sigma \widetilde{S}_t dW_t = \sigma \widetilde{S}_t \left(\left(\frac{\mu-r}{\sigma} \right) dt + dB_t \right) \\ \widetilde{S}_t &= S_t e^{-rt} = S_0 e^{\sigma W_t - \frac{\sigma^2}{2}t} \end{aligned}$$

si bien que $(\widetilde{S}_t)_{t \in [0, T]}$ est une \mathbb{Q} -martingale. Donc sous la probabilité \mathbb{Q} , on a

$$dS_t = S_t(\sigma dW_t + r dt),$$

$$S_t = S_0 \exp \left(\sigma W_t + \left(r - \frac{1}{2} \sigma^2 \right) t \right). \quad (2)$$

Dans ce rapport, sauf mention contraire, on se place sous la probabilité \mathbb{Q} , avec la notation simplifiée de l'espérance : $E := E^{\mathbb{Q}}$.

Volatilité implicite (VI) de Black-Sholes La volatilité implicite d'un prix d'option donné est la volatilité du modèle Black-Scholes associée à ce prix, c'est-à-dire que l'on veut trouver la volatilité σ telle que

$$C_{\text{BS}}(t, S_t, K, T, \sigma) = C_{\text{obs}}(t, S_t, K, T),$$

où C_{BS} est le prix donné par le modèle Black-Scholes, C_{obs} est un prix observé sur le marché ou un prix simulé par un modèle donné. Dans la suite, ce prix pourrait être simulé par le modèle de Heston à l'aide de l'une des méthodes suivantes : Monte-Carlo, EDP, inversion de Fourier ou autre.

Puisque le prix observé ne change pas à un instant fixe dans le temps, c'est une constante, donc l'équation devient

$$C_{\text{BS}}(t, S_t, K, T, \sigma) - C = 0$$

où $C \in \mathbb{R}$ est une constante et la solution σ peut être trouvée en utilisant un algorithme itératif de recherche de racine. De plus, on peut remarquer que la formule de Black-Scholes **3** est bijective par rapport à σ .

1 - Méthode de Monte-Carlo

Le modèle de Black-Scholes, étant un simple modèle de pricing, peut être utilisé pour déterminer le juste prix ou la valeur théorique d'un call ou d'un put. Cela peut se faire facilement avec la méthode de Monte-Carlo. On note X une variable aléatoire d'intérêt à

valeurs dans \mathbb{R}^d et g une fonction de \mathbb{R}^d dans \mathbb{R} . Dans le cadre de l'estimation d'espérance, la méthode de Monte-Carlo s'utilise généralement lorsque la loi de X est connue mais $E(g(X))$ n'a pas de solution analytique ou bien lorsque la loi de X n'est pas explicite mais peut être simulée. On cherche alors à approcher $E(g(X))$ par Monte-Carlo pour un échantillon de variables indépendantes et identiquement distribuées (X_1, \dots, X_N) de X : on a alors $E(g(X))$ approché par $\frac{1}{N} \sum_{i=1}^N g(X_i)$.

Prenons l'exemple du calcul de prix d'un call, de payoff $\Psi_{\text{call}}(S_T) = (S_T - K)^+ = \max(S_T - K, 0)$, avec K le prix d'exercice. On se donne une valeur initiale S_0 et les paramètres nécessaires (σ, T, r, K) . Avec la formule (2), par la méthode de Monte-Carlo, on peut simuler des valeurs de S_T pour estimer l'espérance $E[\Psi_{\text{call}}(S_T)]$. On peut ensuite calculer le prix C_0 du call à l'instant $t = 0$, en fonction de S_0 :

$$C_0(S_0) = e^{-rT} E[\Psi_{\text{call}}(S_T)|S_0] = e^{-rT} E[\Psi_{\text{call}}(S_T)].$$

2 - Feynman-Kac EDP de Black-Scholes

Notons f le prix $f(t, S_t)$ d'une option dépendante de S et de t . On note aussi que $f(T, S_T) = \Psi(S_T)$ le payoff de l'option à la maturité T . D'après le lemme d'Itô et la constitution d'un portefeuille autofinancé, on arrive à l'équation aux dérivées partielles (EDP) de Black-Scholes donnée par [Kar13] :

$$\frac{\partial f}{\partial t} + rS_t \frac{\partial f}{\partial S_t} + \frac{1}{2} \sigma^2 S_t^2 \frac{\partial^2 f}{\partial S_t^2} = rf,$$

qui est aussi une EDP parabolique. Cette équation peut se résoudre par la méthode des éléments finis avec FreeFEM.

3 - Formule de Black-Scholes

Étant donné le prix actualisé de l'actif risqué \tilde{S}_t une martingale, on trouve la formule de Black-Scholes (3), qui détermine le prix des options de type européen, étant donnés le risque neutre r , le temps jusqu'à l'échéance (time to maturity) $\tau = T - t$ et son strike K .

En général, en prenant $t = 0$ comme l'instant initial où on évalue l'option, on a $\tau = T - t = T$. D'après [LL12], on obtient le prix du call :

$$C_0(S_0, K, r, T, \sigma) := f(t, S_t) = S_0 \mathcal{N}(d_1) - K e^{-rT} \mathcal{N}(d_2), \quad (3)$$

où

$$\mathcal{N}(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^z e^{-u^2/2} du$$

la fonction de répartition de la loi normale centrée réduite, et

$$d_1 = \frac{1}{\sigma\sqrt{T}} \left[\ln\left(\frac{S_0}{K}\right) + \left(r + \frac{\sigma^2}{2}\right) T \right],$$

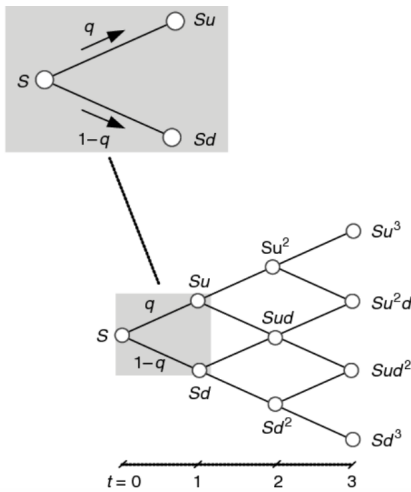
$$d_2 = \frac{1}{\sigma\sqrt{T}} \left[\ln\left(\frac{S_0}{K}\right) + \left(r - \frac{\sigma^2}{2}\right) T \right],$$

avec $d_2 = d_1 - \sigma\sqrt{T}$.

4 - Modèle binomial et trinomial

Comme expliqué dans [DD07], le modèle de Black-Scholes est la limite du modèle de Cox Ross-Rubinstein (CRR) [CRR79] connu sous le nom de modèle binomial, quand son nombre N de pas de discrétisation de l'intervalle $[0, T]$ tend vers plus l'infini.

Modèle binomial Le modèle binomial (Fig. 1) est utilisé pour modéliser l'évolution de l'actif sous-jacent, sous forme d'un arbre binomial, pour un nombre donné de « pas » qui correspond au temps entre la date initiale et la maturité de l'option : ainsi on discrétise l'espace de temps $[0, T]$. Chaque nœud de l'arbre (intersection entre deux branches de l'arbre) correspond à un prix possible du sous-jacent à un instant donné. On accepte qu'à chaque pas de la discrétisation, le prix du sous-jacent augmente (up) avec une probabilité q (on note le taux augmentant u), ou diminue (down) avec une probabilité $(1 - q)$ (on note le taux descendant d). Par définition, $u \geq 1$ et $0 < d \leq 1$.



Source : L. E. Brandão, J. Dyer, W. J. Hahn. *Decis. Anal.* 2005

FIG. 1 – Modèle binomial et arbre binomial correspondant, avec $N = 3$.

La création de l'arbre de prix s'effectue en partant de la date à laquelle on veut valoriser l'option et ce jusqu'à la date de maturité de l'option. Si S est le prix actuel de la date initiale, alors le prix du « pas » suivant sera $S_{up} = Su$ ou $S_{down} = Sd$. Les taux u et d utilisés pour évaluer l'augmentation ou la diminution du sous-jacent sont calculés en prenant en compte la volatilité σ du prix de sous-jacent S_t , et l'intervalle de temps Δt pour chaque étape, est mesuré en année. Si le temps à maturité est de 1 an et que l'on veut le discrétiser en N pas, on aura $\Delta t = \frac{T}{N} = \frac{1}{N}$, et $t_k = k \cdot \Delta t$ à la k^e étape.

En supposant que $V(\log S_t) = \sigma^2 t$ (voir P48-49, [Zay16]), on obtient :

$$u = e^{\sigma\sqrt{\Delta t}}, \quad d = e^{-\sigma\sqrt{\Delta t}} = \frac{1}{u}.$$

Dans ce modèle, on remarque que la valeur d'un nœud est donc :

$$S_n = S_0 \times u^{N_u - N_d},$$

Le processus d'évaluation est itératif : on part du nœud final de chaque branche puis on « remonte » jusqu'au premier nœud, qui correspond à la valeur de l'option de la date d'évaluation qu'on cherche. On remarque aussi qu'il s'agit d'un processus de Markov car le résultat à l'étape t ne dépend que de celui à l'étape $t - 1$.

Les étapes du processus d'évaluations sont les suivantes :

1. création de l'arbre ;
2. calcul de la valeur de l'option au nœud final de chaque branche ;
3. calcul progressif de la valeur de l'option à partir du nœud précédent, la valeur du premier nœud étant la valeur de l'option recherchée.

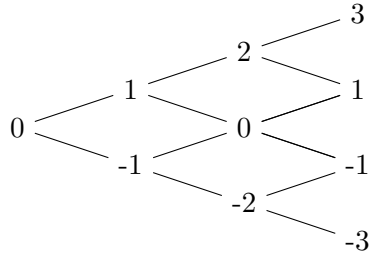


FIG. 2 – Numérotation i des noeuds de la Fig. 1

où N_u : nombre de pas correspondant à une augmentation jusqu'au pas n , N_d : nombre de pas correspondant à une baisse jusqu'au pas n et $n \in \{0, \dots, N\}$.

Les derniers nœuds des branches correspondant à la valeur intrinsèque de l'option (i.e. $(S_N - K)^+$ pour un call), l'hypothèse de neutralité du risque permet de remonter vers le premier nœud et de déterminer le call à l'instant initial.

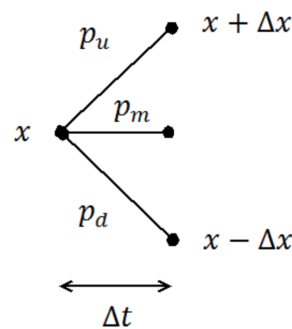
Notons $C_{t,i}$ le prix du call du nœud i à l'instant t . On note aussi $C_{t,i+1}$ le prix du call à l'instant t lors d'une augmentation et $C_{t,i-1}$ lors d'une baisse, en partant du nœud i (Fig. 2 pour la numérotation des nœuds). La formule suivante est appliquée pour calculer l'espérance à chaque nœud :

$$C_{t-\Delta t,i} = e^{-r\Delta t}(qC_{t,i+1} + (1-q)C_{t,i-1}),$$

où le paramètre q est le rendement du dividende du sous-jacent correspondant à la durée de vie de l'option. On choisit $q = \frac{e^{r\Delta t} - d}{u - d}$ en fonction de la loi binomiale qui simule le mouvement brownien géométrique de l'action avec les paramètres r et σ . Il faut $\Delta t < \frac{\sigma^2}{r^2}$ pour que $0 < q < 1$. On obtient ainsi de manière itérative le prix recherché du call $C_{0,0}$ à l'instant initial $t = 0$.

Modèle trinomial Comme vu dans [JA20], un modèle trinomial converge plus vite qu'un modèle binomial. Le modèle trinomial le plus connu est le modèle Kamrad-Ritchken ou modèle KR [KR91], qui est de l'ordre de convergence le plus bas et est le plus rapide pour le calcul du prix du call européen.

Considérons le cas sans dividende. Différent du modèle binomial, le modèle trinomial rajoute la possibilité que le prix du sous-jacent reste inchangé. Comme illustré dans



Source : Pier Giuseppe Giribone.

FIG. 3 – Modèle trinomial.

Fig. 3, dans un modèle trinomial, pour toutes les étapes, on a une probabilité d'augmentation p_u , une probabilité de stabilité p_m et une probabilité de diminution p_d . L'approche d'approximation trinominale de Kamrad et Ritchken (1991) est la suivante. Supposons qu'un

actif sous-jacent S suit un processus de diffusion avec un drift

$$v = r - \frac{\sigma^2}{2},$$

où r est le taux sans risque et σ est l'écart type instantané (la volatilité). Alors pour l'actif sur l'intervalle de temps $[t, t + \Delta t]$, nous avons :

$$\ln S(t + \Delta t) = \ln S(t) + A^*(t),$$

où la variable aléatoire normale $A^*(t)$ a une moyenne $v\Delta t$ et une variance $\sigma^2\Delta t$. Puisque nous devons approximer la distribution de $A^*(t)$ sur la période $[t, t + \Delta t]$, nous considérons donc une variable aléatoire discrète $A(t)$ telle que

$$A(t) = \begin{cases} \Delta x, & \text{avec probabilité } p_u, \\ 0, & \text{avec probabilité } p_m, \\ -\Delta x, & \text{avec probabilité } p_d, \end{cases}$$

où $\Delta x = \lambda\sigma\sqrt{\Delta t}$ et le paramètre $\lambda \geq 1$ est à déterminer. Afin de déterminer les probabilités p_u , p_m et p_d , nous assimilons la moyenne et la variance de $A(t)$ à la moyenne et à la variance de $A^*(t)$. Plus précisément :

$$\begin{cases} \Delta x(p_u - p_d) = v\Delta t \\ (\Delta x)^2(p_u + p_d) - (v\Delta t)^2 = \sigma^2\Delta t \\ p_u + p_m + p_d = 1. \end{cases}$$

On obtient donc les expressions suivantes :

$$\begin{aligned} p_u &= \frac{1}{2} \left[\frac{1}{\lambda^2} + \frac{v\sqrt{\Delta t}}{\sigma\lambda} + \frac{v^2\Delta t}{\sigma^2\lambda^2} \right], \\ p_d &= \frac{1}{2} \left[\frac{1}{\lambda^2} - \frac{v\sqrt{\Delta t}}{\sigma\lambda} + \frac{v^2\Delta t}{\sigma^2\lambda^2} \right], \\ p_m &= 1 - \left[\frac{1}{\lambda^2} + \frac{v^2\Delta t}{\sigma^2\lambda^2} \right]. \end{aligned}$$

Le modèle KR ne fournit pas de moyen de sélectionner λ , ce qui peut sérieusement limiter la mise en œuvre d'un tel modèle. Lorsque λ est choisi arbitrairement, les problèmes suivants peuvent survenir : (a) incapacité à garantir un ensemble réalisable de probabilités ; et (b) les valeurs de probabilité dépendent de λ , ce qui affectera les valeurs d'option. Or il existe un théorème qui nous dit que, si

$$\sqrt{1 + (v/\sigma)^2\Delta t} \leq \lambda \leq \frac{1 + (v/\sigma)^2\Delta t}{(v/\sigma)\sqrt{\Delta t}},$$

les probabilités du modèle seront toutes entre 0 et 1.

De manière similaire au cas binomial, on utilise la formule itérative

$$C_{t-\Delta t, i} = e^{-r\Delta t}(p_u C_{t, i+1} + p_m C_{t, i} + p_d C_{t, i-1}),$$

afin de remonter l'arbre et trouver le prix $C_{0,0}$ du call à l'instant initial $t = 0$.

3.1.2.2 Modèle de Heston

Le modèle de Heston [Hes93], du nom de Steven L. Heston, est aussi un modèle mathématique du marché. Il suppose que le cours de sous-jacent, S_t , suit un processus stochastique de type Black-Scholes, mais avec une variance (ou volatilité) stochastique v_t qui suit un processus de Cox, Ingersoll et Ross (1985) [CIR05].

La présentation du modèle vient principalement de [Rou13], où l'on trouve aussi des explications détaillées et des démonstrations pour certaines étapes.

Présentation du modèle Sous la probabilité historique \mathbb{P} , le modèle est représenté par le système bivarié d'équations différentielles stochastiques (EDS) :

$$\begin{aligned} dS_t &= \mu S_t dt + \sqrt{v_t} S_t dW_{1,t}, \\ dv_t &= \kappa (\bar{v} - v_t) dt + \gamma \sqrt{v_t} (\rho dW_{1,t} + \sqrt{1 - \rho^2} dW_{2,t}), \end{aligned} \quad (4)$$

avec

- $S_{t_0} = S_0, v_{t_0} = v_0, t_0 = 0$;
- $\frac{2\kappa\bar{v}}{\gamma^2} > 1$, condition Feller pour que $v > 0$;
- $W_{1,t}, W_{2,t}$: deux processus de Wiener standards et indépendants, $t \in [0, +\infty[$;

où

- S_t : prix de l'actif sous-jacent S à l'instant t ;
- v_t : variance stochastique ;
- μ : drift de S ;
- ρ : coefficient de corrélation des processus de Wiener ;
- $\kappa > 0$: taux de réversion auquel v_t revient vers \bar{v} ;
- $\gamma > 0$: volatilité de la variance (vol de vol) ;
- $\bar{v} > 0$: variance moyenne à long terme (variance longue) ;
- $v_0 > 0$: valeur initiale (instant zéro) de la variance.

$(\rho, \kappa, \bar{v}, \gamma, v_0)$ sont des paramètres du modèle.

Probabilité risque neutre Sous la probabilité risque neutre \mathbb{Q} , depuis (4), prenons le modèle :

$$\begin{aligned} dS_t &= r S_t dt + \sqrt{v_t} S_t dW_{1,t}^{\mathbb{Q}}, \\ dv_t &= \kappa (\bar{v} - v_t) dt + \gamma \sqrt{v_t} dW_{2,t}^{\mathbb{Q}}, \end{aligned} \quad (5)$$

avec $W_{1,t}^{\mathbb{Q}} := \left(W_{1,t} + \frac{\mu - r}{\sqrt{v_t}} t \right)$, $W_{2,t}^{\mathbb{Q}} := \rho dW_{1,t} + \sqrt{1 - \rho^2} dW_{2,t}$.

1 - Méthode de Monte-Carlo

Dans le modèle de Heston, la plus simple méthode d'estimation du prix d'option est celle de Monte-Carlo, avec un schéma de discrétisation d'Euler ou de Milstein [Rou].

En partant simplement de la dynamique (5), nous pouvons simuler les prix des options européennes du marché et évaluer d'autres types d'options avec des caractéristiques complexes à l'aide du modèle de Heston.

Posons T le temps à l'échéance de l'option et on discrétise le temps en $N + 1$ pas ($\Delta t = T/N$, $i \in \llbracket 0, N \rrbracket$). D'après le schéma de discrétisation d'Euler dans [Rou] et la dynamique (5), sous la probabilité risque neutre \mathbb{Q} , on a

$$\begin{aligned} S_{i+1} &= S_i \exp \left\{ \left(r - \frac{v_i}{2} \right) \Delta t + \sqrt{(v_i \Delta t)^+} V_{1,i+1}^{\mathbb{Q}} \right\}, \\ v_{i+1} &= v_i + \kappa(\bar{v} - v_i) \Delta t + \gamma \sqrt{(v_i \Delta t)^+} V_{2,i+1}^{\mathbb{Q}}, \end{aligned}$$

l'indice i correspondant à l'instant $i\Delta t$. Pour générer $V_{1,i+1}^{\mathbb{Q}}$ et $V_{2,i+1}^{\mathbb{Q}}$ avec corrélation ρ , on génère d'abord deux variables normales standard indépendantes, N_1 et N_2 , et on pose $V_{1,i+1}^{\mathbb{Q}} = N_1$ et $V_{2,i+1}^{\mathbb{Q}} = \rho N_1 + \sqrt{1 - \rho^2} N_2$.

Avec un nombre de simulation supérieur à 10^5 , on arrive à estimer $E[\Psi_{\text{call}}(S_T)] = E[(S_T - K)^+]$. Le prix du call à l'instant $t = 0$ est donc $C = e^{-rT} E[\Psi_{\text{call}}(S_T)]$.

2 - Feynman-Kac EDP

De manière similaire au cas Black-Scholes, pour dériver l'EDP de Heston, on construit un portefeuille autofinancé et on applique le lemme d'Itô sur celui-ci. Cependant, dans le modèle de Heston, on utilise deux produits dérivés au lieu d'un. Plus de détails de calcul se trouvent dans [Rou13].

Notons $U_t = U(S, v, t)$ le prix de l'option. L'EDP de Heston s'écrit

$$\frac{\partial U_t}{\partial t} + \frac{1}{2} v S_t^2 \frac{\partial^2 U_t}{\partial S_t^2} + \rho \gamma v S_t \frac{\partial^2 U_t}{\partial S_t \partial v} + \frac{1}{2} \gamma^2 v \frac{\partial^2 U_t}{\partial v^2} - r U_t + r S_t \frac{\partial U_t}{\partial S_t} + \kappa(\bar{v} - v) \frac{\partial U_t}{\partial v} = 0. \quad (6)$$

Les conditions aux limites suivantes sur l'EDP (6) sont valables pour une option du call européen de maturité T et d'exercice K . On a alors

$$U(S, v, T) = \max(S_T - K, 0).$$

Cette équation (6) peut se résoudre par les méthodes de résolution d'EDP ou la méthode des éléments finis avec FreeFEM. L'implémentation se trouve dans [Pir19].

3 - Formule semi-analytique de Heston

La formule semi-analytique de Heston permet de calculer directement le prix de l'option. On parle de « solution semi-analytique » puisque la solution analytique de ses intégrales n'est pas connue. Plusieurs approches existent, comme celle proposée par Heston dans [Hes93], celle dérivée par la méthode transformation de Fourier ou par cosinus expansion. Plus de détails de calcul et de diverses extensions d'expression de la formule se trouvent dans [Rou13].

Soit $P_1 = \mathbb{Q}^S(S_T > K)$, $P_2 = \mathbb{Q}(S_T > K)$ deux probabilités du prix du call expiré dans la monnaie ($S_T > K$) et

$$\frac{d\mathbb{Q}}{d\mathbb{Q}^S} = \frac{B_T/B_t}{S_T/S_t} = \frac{E^{\mathbb{Q}}[e^{xT}]}{e^{xT}},$$

avec $B_t = e^{rt}$ taux d'actualisation continu et $x_t = \ln S_t$. Alors on a

$$\begin{aligned} C(K) &= S_t \mathbb{Q}^S(S_T > K) - K e^{-rT} \mathbb{Q}(S_T > K) \\ &= S_t P_1 - K e^{-rT} P_2. \end{aligned} \quad (7)$$

D'après [Rou13], P_1 et P_2 suivent aussi l'EDP de Heston, ce qui donne, pour $j \in \{1, 2\}$:

$$\begin{aligned} \frac{\partial P_j}{\partial t} + \rho\gamma v \frac{\partial^2 P_j}{\partial v \partial x} + \frac{1}{2}v \frac{\partial^2 P_j}{\partial x^2} + \frac{1}{2}\gamma^2 v \frac{\partial^2 P_j}{\partial v^2} + (r + u_j v) \frac{\partial P_j}{\partial x} + (a - b_j v) \frac{\partial P_j}{\partial v} &= 0, \\ u_1 = \frac{1}{2}, u_2 = -\frac{1}{2}, a = \kappa\bar{v}, b_1 = \kappa + \lambda - \rho\gamma, b_2 = \kappa + \lambda, i^2 = -1. \end{aligned} \quad (8)$$

Selon le théorème d'inversion de Gil-Pelaez (1951) [Gil51], on peut retrouver la probabilité P_j par sa fonction caractéristique $f_j(\phi; x, v)$ qui est une application affine :

$$P_j = \frac{1}{2} + \frac{1}{\pi} \int_0^\infty \operatorname{Re} \left(\frac{e^{-i\phi \ln K} f_j(\phi; x, v)}{i\phi} \right) d\phi,$$

avec une approche proposée par Heston dans [Hes93] où

$$\begin{aligned} f_j(\phi; v_t, x_t) &= \exp[C_j(\tau, \phi) + D_j(\tau, \phi)v_t + i\phi x_t], \\ C_j(\tau, \phi) &= (r - q)i\phi\tau + \frac{a}{\gamma^2} \left((b_j - \rho\gamma i\phi + d_j)\tau - 2 \ln \frac{1 - g_j e^{d_j \tau}}{1 - g_j} \right), \\ D_j(\tau, \phi) &= \frac{b_j - \rho\gamma i\phi + d_j}{\gamma^2} \left(\frac{1 - e^{d_j \tau}}{1 - g_j e^{d_j \tau}} \right), \\ g_j &= \frac{b_j - \rho\gamma i\phi + d_j}{b_j - \rho\gamma i\phi - d_j}, \\ d_j &= \sqrt{(b_j - \rho\gamma i\phi)^2 - \gamma^2(2iu_j\phi - \phi^2)}. \end{aligned}$$

Cependant, dans l'application numérique, calculer une seule intégrale est plus rapide que en faire deux. On utilise donc l'expression (9), qui est équivalente à la formule de Heston (8).

Une astuce consistant à multiplier par $\exp(-d_j\tau)$ dans le numérateur et le dénominateur de D_j , appelée « little heston trap » dans [Rou13], permet de résoudre le problème de discontinuité de l'intégrand $\operatorname{Re} \left(\frac{e^{-i\phi \ln K} f_j(\phi; x, v)}{i\phi} \right)$. L'équation (7) peut alors être simplifiée en une intégrale comme suit :

$$\begin{aligned} C(K) &= S_t \left[\frac{1}{2} + \frac{1}{\pi} \int_0^\infty \operatorname{Re} \left(\frac{e^{-is \ln K} \cdot f_1}{is} \right) ds \right] - K e^{-r\tau} \left[\frac{1}{2} + \frac{1}{\pi} \int_0^\infty \operatorname{Re} \left(\frac{e^{-is \ln K} \cdot f_2}{is} \right) ds \right] \\ &= \frac{1}{2} (S_t - K e^{-r\tau}) + \left[\frac{S_t}{\pi} \int_0^\infty \operatorname{Re} \left[\frac{e^{-is \ln K} f_1}{is} \right] ds - \frac{K e^{-r\tau}}{\pi} \int_0^\infty \operatorname{Re} \left(\frac{e^{-is \ln K} f_2}{is} \right) ds \right] \\ &= \frac{1}{2} (S_t - K e^{-r\tau}) + \frac{1}{\pi} \left[S_t \int_0^\infty \operatorname{Re} \left(\frac{f_1}{is K^{is}} \right) ds - K e^{-r\tau} \int_0^\infty \operatorname{Re} \left(\frac{f_2}{is K^{is}} \right) ds \right] \\ &= \frac{1}{2} (S_t - K e^{-r\tau}) + \frac{1}{\pi} \int_0^\infty \operatorname{Re} \left[\frac{f_1 S_t}{is K^{is}} - K e^{-r\tau} \frac{f_2}{is K^{is}} \right] ds \\ &= \frac{1}{2} (S_t - K e^{-r\tau}) + \frac{1}{\pi} \int_0^\infty \operatorname{Re} \left[e^{r\tau} \frac{f(s-i)}{is K^{is}} - K \frac{f(s)}{is K^{is}} \right] ds, \end{aligned} \quad (9)$$

avec

$$f(s) = e^{r\tau} S_t^{is} \left(\frac{1 - g e^{-d\tau}}{1 - g} \right)^{-\frac{2\bar{v}\kappa}{\gamma^2}} \exp \left(\frac{\bar{v}\kappa\tau}{\gamma^2} (\kappa - \rho\gamma is - d) + \frac{v_0}{\gamma^2} (\kappa - \rho\gamma is + d) \frac{1 - e^{d\tau}}{1 - g e^{d\tau}} \right),$$

$$d = \sqrt{(\rho\gamma si - \kappa)^2 + \gamma^2(is + s^2)}, \quad g = \frac{\kappa - \rho\gamma is - d}{\kappa - \rho\gamma is + d}.$$

Avec $t = 0$ et une somme de Riemann approchant l'intégrale avec la méthode du point médian, on arrive à implémenter numériquement la variation de la formule de Heston (9), pour trouver le prix du call du sous-jacent à l'instant initial.

4 - Transformation de Fourier

Plusieurs approches de la formule semi-analytique de Heston par la transformation de Fourier se trouvent dans [Sch10]. Dans cette partie, on se concentre sur l'approche Carr-Madan et l'expansion en cosinus de Fourier.

Regardons d'abord la définition de transformation de Fourier. La transformation de Fourier \mathcal{F} est une opération qui transforme une fonction f intégrable sur \mathbb{R} en une autre fonction complexe \hat{f} , appelée transformée de Fourier, d'où $\mathcal{F}(f) = \hat{f}$.

Il existe plusieurs définitions de transformée de Fourier \hat{f} d'une fonction f . Ici on utilise

$$\hat{f}(u) = \int_{-\infty}^{\infty} e^{iux} f(x) dx,$$

où $i = \sqrt{-1}$ l'unité imaginaire. La fonction f peut être récupérée de \hat{f} par la transformée de Fourier inverse

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-iux} \hat{f}(u) du.$$

Lorsque cette transformation est utilisée en traitement du signal, on notera volontiers t à la place de x et ω ou $2\pi v$ à la place de u qui seront les variables respectives de temps et de pulsation ou de fréquence. On dira alors que f est dans le domaine temporel, et que \hat{f} est dans le domaine fréquentiel.

On remarque que la fonction caractéristique de X est une transformée de Fourier, telle que

$$\hat{f}(u) = E[e^{iuX}]$$

où f est la fonction densité de X .

Méthode de Carr-Madan En 1999, dans [CM99], Carr et Madan ont proposé une méthode pour calculer le prix du call avec la transformation de Fourier. Cette méthode a pour avantages de permettre d'appliquer la transformation de Fourier rapide (anglais : FFT ou fast Fourier transform) pour réduire le temps de calcul, et d'avoir moins d'oscillations sur l'intégration de la formule de Heston (9), avec un intégrant qui converge plus vite. Pourtant, comme le dit l'auteur dans [Kol15], cette méthode a aussi des problèmes :

1. L'estimation du facteur d'amortissement (damping factor) est un travail dur et non intuitif, qui dépend de l'expérience de l'expert.
2. La structure du modèle dépend du payoff, qui entraîne une modification de modèle chaque fois après le changement de payoff, et donc rend le modèle moins flexible.

On se place au temps $t = 0$. On note τ le temps jusqu'à la maturité ($\tau = T$). On note aussi q la fonction densité de S_T , $k = \ln K$ et $x = \ln S_T$. Commençons par noter le prix du call

$$\begin{aligned} C(k) &= e^{-rT} E[(S_T - K)^+] = e^{-rT} \int_k^{\infty} (e^x - e^k) q(x) dx \\ &= S_0 Q^S(S_T > e^k) - K e^{-rT} Q(S_T > e^k). \end{aligned}$$

Puisque $C(k)$ n'est pas L^1 -intégrable, il n'a pas de transformation de Fourier. Considérons donc l'expression L^1 -intégrable

$$c(k) = e^{\alpha k} C(k),$$

où $e^{\alpha k}$ est appelé damping factor.

Donc on peut trouver la transformée de Fourier de $c(k)$:

$$\begin{aligned} \varphi(u) &= \int_{-\infty}^{+\infty} e^{iuk} c(k) dk = \int_{-\infty}^{+\infty} e^{iuk} e^{\alpha k} C(k) dk \\ &= e^{-rT} \int_{-\infty}^{+\infty} e^{iuk} e^{\alpha k} \left(\int_k^{+\infty} (e^x - e^k) q(x) dx \right) dk \\ &= e^{-rT} \int_{-\infty}^{+\infty} q(x) \left[\int_{-\infty}^x e^{(\alpha+iu)k} (e^x - e^k) dk \right] dx \quad (\text{Fubini}) \\ &= e^{-rT} \int_{-\infty}^{+\infty} q(x) \left[\frac{e^{(\alpha+iu+1)x}}{(\alpha+iu)(\alpha+iu+1)} \right] dx \\ &= \left[\frac{e^{-rT}}{(\alpha+iu)(\alpha+iu+1)} \right] \varphi_{\ln S_T}(u - (\alpha+1)i). \end{aligned}$$

La fonction caractéristique $\varphi_{\ln S_T}$ est déduite dans [MCC98]. Avec la même expression que dans [CM99], après une organisation d'écriture, [Kol15] propose

$$\varphi_{\ln S_T}(u) = \frac{\exp\left(iu \ln S_0 + iurT + \frac{\kappa \bar{v} T (\kappa - i\rho\gamma u)}{\gamma^2}\right)}{\left[\cosh\left(\frac{\gamma(u)T}{2}\right) + \frac{\kappa i \rho u}{\gamma(u)} \sinh\left(\frac{\gamma(u)T}{2}\right)\right]^{\frac{2\kappa \bar{v}}{\gamma^2}}} \exp\left(\frac{-(u^2 + iu)v_0}{\gamma(u) \coth\left(\frac{\gamma(u)T}{2}\right) + \kappa - i\rho\gamma u}\right),$$

où

$$\gamma(u) = \sqrt{\gamma^2(u^2 + iu) + (\kappa - i\rho\gamma u)^2}.$$

Par la transformation de Fourier inverse, $C(k)$ peut être récupéré par la fonction caractéristique $\varphi(u)$, en enlevant le damping factor, d'où

$$C(k) = \frac{e^{-\alpha k}}{2\pi} \int_{-\infty}^{+\infty} e^{-iuk} \varphi(u) du = \frac{e^{-\alpha k}}{\pi} \int_0^{+\infty} \text{Re}[e^{-iuk} \varphi(u)] du$$

car φ est paire dans sa partie réelle.

Méthode COS La méthode COS, proposée dans la thèse PhD de F.Fang [Fan10], est une méthode d'évaluation des options pour les options européennes basée sur la série Fourier-cosinus. L'idée principale est dans la relation étroite de la fonction caractéristique avec les coefficients de la série de l'expansion Fourier-cosinus de la fonction de densité. L'avantage de cette méthode repose sur sa rapidité, sa capacité permettant d'appliquer un algorithme FFT et l'exactitude de son résultat. Des résumés concis de cette méthode existent dans [Kol15] et [FO09]. La méthode est présentée comme suit.

Idée générale : Commençons par écrire la formule de pricing d'options européennes, sous la probabilité risque neutre :

$$v(x, t_0) = e^{-r\Delta t} \mathbb{E}^{\mathbb{Q}}[v(y, T)|x] = e^{-r\Delta t} \int_{\mathbb{R}} v(y, T) f(y|x) dy, \quad (10)$$

où v désigne la valeur de l'option, Δt est la différence entre l'échéance T et la date initiale t_0 , et $\mathbb{E}^{\mathbb{Q}}[\cdot]$ est l'opérateur d'espérance sous mesure risque neutre \mathbb{Q} . x et y sont des variables aux temps t_0 et T respectivement, $f(y|x)$ est la densité de probabilité de y étant donné x , et r est le taux d'intérêt risque neutre.

Vu que la densité décroît rapidement vers zéro lorsque $y \rightarrow \pm\infty$ dans (10), sans perdre une précision significative, on peut tronquer l'intervalle d'intégration infini à $[a, b] \subset \mathbb{R}$, et nous obtenir l'approximation v_1 de v :

$$v_1(x, t_0) = e^{-r\Delta t} \int_a^b v(y, T) f(y|x) dy. \quad (11)$$

Puisque $f(y|x)$ est souvent inconnu alors que la fonction caractéristique v l'est, on remplace la densité par son développement en cosinus en y ,

$$f(y|x) = \sum'_{k=0}^{+\infty} A_k(x) \cos\left(k\pi \frac{y-a}{b-a}\right), \text{ avec } A_k(x) := \frac{2}{b-a} \int_a^b f(y|x) \cos\left(k\pi \frac{y-a}{b-a}\right) dy,$$

et \sum' indique que le premier terme de la sommation est pondéré par $\frac{1}{2}$. Alors

$$v_1(x, t_0) = e^{-r\Delta t} \int_a^b v(y, T) \sum'_{k=0}^{+\infty} A_k(x) \cos\left(k\pi \frac{y-a}{b-a}\right) dy.$$

On échange la sommation et l'intégration, et on donne la définition

$$V_k := \frac{2}{b-a} \int_a^b v(y, T) \cos\left(k\pi \frac{y-a}{b-a}\right) dy.$$

Ainsi

$$v_1(x, t_0) = \frac{b-a}{2} e^{-r\Delta t} \sum'_{k=0}^{+\infty} A_k(x) V_k. \quad (12)$$

Notons V_k les coefficients de la série cosinus de $v(y, T)$ en y . Ainsi ci-dessus, nous avons transformé le produit de deux fonctions réelles, $f(y|x)$ et $v(y, T)$ dans (11), à celui de leurs coefficients de série Fourier-cosinus (12).

Avec une troncature de somme ($v_2(x, t_0) = \frac{b-a}{2} e^{-r\Delta t} \sum'_{k=0}^{N-1} A_k(x) V_k$) et une prolongation du coefficient A_k , étant donné ϕ une fonction caractéristique, on obtient la formule COS pour les processus généraux des sous-jacents :

$$v(x, t_0) \approx v_3(x, t_0) = e^{-r\Delta t} \sum'_{k=0}^{N-1} Re \left\{ \phi\left(\frac{k\pi}{b-a}; x\right) e^{-ik\pi \frac{a}{b-a}} \right\} V_k. \quad (13)$$

Particulièrement, pour les options vanille (call ou put européen), on peut obtenir les valeurs de V_k analytiquement, puis utiliser (13) pour pricing.

Options européennes : Comme nous supposons ici que la fonction caractéristique du log-prix de l'actif est connu, nous représentons le payoff en fonction du log-prix d'actif. Notons les log-prix d'actifs par

$$x := \ln(S_0/K) \text{ et } y := \ln(S_T/K),$$

avec S_t le prix de sous-jacent au temps t et K le prix d'exercice. Le payoff des options européennes, en log-prix d'actif, est

$$v(y, T) := [\alpha \cdot K(e^y - 1)]^+ \text{ avec } \alpha = \begin{cases} 1 & \text{pour un call,} \\ -1 & \text{pour un put.} \end{cases}$$

Alors

$$V_k^{call} = \frac{2}{b-a} \int_0^b K(e^y - 1) \cos\left(k\pi \frac{y-a}{b-a}\right) dy = \frac{2}{b-a} K(\chi_k(0, b) - \psi_k(0, b)),$$

$$V_k^{put} = \frac{2}{b-a} K(-\chi_k(a, 0) + \psi_k(a, 0)),$$

avec χ_k et ψ_k des coefficients de séries cosinus sur $[c, d] \subset [a, b]$:

$$\chi_k(c, d) := \frac{1}{1 + \left(\frac{k\pi}{b-a}\right)^2} \left[\cos\left(k\pi \frac{d-a}{b-a}\right) e^d - \cos\left(k\pi \frac{c-a}{b-a}\right) e^c \right. \\ \left. + \frac{k\pi}{b-a} \sin\left(k\pi \frac{d-a}{b-a}\right) e^d - \frac{k\pi}{b-a} \sin\left(k\pi \frac{c-a}{b-a}\right) e^c \right]$$

et

$$\psi_k(c, d) := \begin{cases} \left[\sin\left(k\pi \frac{d-a}{b-a}\right) - \sin\left(k\pi \frac{c-a}{b-a}\right) \right] \frac{b-a}{k\pi} & k \neq 0, \\ (d-c) & k = 0. \end{cases}$$

Pour le modèle de Heston, avec v_0 la volatilité au temps initial, puisque

$$\phi(u; X, v_0) = \varphi_{hes}(u; v_0) \cdot e^{iuX} = \phi(u; 0, v_0) \cdot e^{iuX},$$

on obtient

$$v(X, t_0, v_0) \approx K e^{-r\Delta t} \cdot Re \left\{ \sum_{k=0}^{N-1} \varphi_{hes}\left(\frac{k\pi}{b-a}; v_0\right) U_k \cdot e^{ik\pi \frac{X-a}{b-a}} \right\},$$

avec

$$U_k = \begin{cases} \frac{2}{b-a} (\chi_k(0, b) - \psi_k(0, b)) & \text{pour un call,} \\ \frac{2}{b-a} (-\chi_k(a, 0) + \psi_k(a, 0)) & \text{pour un put.} \end{cases}$$

La fonction caractéristique de log-prix d'actif, $\varphi_{hes}(u; v_0)$, est

$$\varphi_{hes}(u; v_0) = \exp\left(iur\Delta t + \frac{v_0}{\gamma^2} \left(\frac{1 - e^{-D\Delta t}}{1 - Ge^{-D\Delta t}}\right) (\kappa - i\rho\gamma u - D)\right) \cdot \\ \exp\left(\frac{\kappa\bar{v}}{\gamma^2} \left(\Delta t(\kappa - i\rho\gamma u - D) - 2 \log\left(\frac{1 - Ge^{-D\Delta t}}{1 - G}\right)\right)\right),$$

avec

$$D = \sqrt{(i\rho\gamma u - \kappa)^2 + (iu + u^2)\gamma^2}, \quad G = \frac{\kappa - i\rho\gamma u - D}{\kappa - i\rho\gamma u + D}.$$

Dans (13), en remplaçant la fonction caractéristique ϕ par celle du modèle de Heston, ainsi on arrive à évaluer les options européennes par la méthode COS.

3.1.3 Méthodes de calibration

La calibration des modèles financiers peut être formulée comme un problème inverse : sur la base des prix observés des options, les paramètres du modèle financier sont déduits, de façon à ce que le modèle réplique les prix observés sur le marché. Cette procédure a une place importante dans la finance, et est principalement destinée aux gestionnaires de risques, aux analystes quantitatifs et aux commerçants d'options.

Une mesure d'erreur, qui est une distance entre les valeurs observées et celles données par le modèle où N est la quantité d'instruments de calibration disponibles, c'est-à-dire les valeurs observées (ou estimées) :

$$e_i := \|Q(\tau_i, m_i; \Theta) - Q^*(\tau_i, m_i)\|, \quad i = 1, \dots, N,$$

avec :

- m_i : Moneyness ($\frac{S_0}{K}$) de l'option i ;
- τ_i : Temps jusqu'à la maturité ($T - t$) de l'option i ;
- $Q^*(\tau, m)$: Prix d'une option (ou volatilité implicite) dans le marché ;
- $Q(\tau, m; \Theta)$: Prix de l'option (ou volatilité implicite) simulé par un modèle ;
- Θ : Vecteur en \mathbb{R}^p désignant les paramètres à calibrer, où p est le nombre de paramètres du modèle spécifique.

Dans la suite du rapport, Q et Q^* sont des volatilités implicites (VI). Remarquons qu'il existe une bijection entre le prix et la VI, et que les deux choix sont équivalents. On note $J(\Theta)$ une fonction cible représentant la différence totale entre l'observation et le modèle :

$$J(\Theta) := \sum_{i=1}^N \omega_i e_i + \bar{\lambda} \|\Theta\|, \quad (14)$$

où les ω_i sont des poids associés aux différences et $\bar{\lambda}$ un paramètre de régularisation.

Le jeu de paramètres à calibrer pour le modèle de Heston est $\Theta := (\rho, \kappa, \gamma, \bar{v}, v_0)$.

Notons que nous avons introduit une norme dans les définitions précédentes, mais nous n'avons pas précisé à quelle norme nous nous référons. Puisque, selon le choix de cette norme (et du paramètre de régularisation), nous pouvons définir différentes erreurs, nous laissons $\|\cdot\|$ comme concept général pour mesurer les distances. Par exemple, en définissant $\bar{\lambda} = 0$, $\omega_i = \frac{1}{N}$ et $e_i := \|Q(\tau_i, m_i; \Theta) - Q^*(\tau_i, m_i)\|_2^2$, nous obtenons l'erreur quadratique moyenne (MSE) bien connue. En un mot, le processus de calibration se réduit au problème de minimisation suivant :

$$\arg \min_{\Theta \in \mathbb{R}^n} J(\Theta),$$

c'est-à-dire que l'on souhaite trouver l'ensemble des paramètres Θ pour lequel la différence entre les prix (ou les volatilités implicites) du marché et ceux estimés par un modèle financier est la plus petite possible.

Afin de résoudre ce problème, un ensemble initial de paramètres est choisi (souvent de manière aléatoire avec certaines contraintes en fonction du modèle financier) et l'erreur correspondante avec les données du marché est calculée. Ensuite on détermine une direction de recherche pour trouver un nouvel ensemble de paramètres. Cela signifie que le problème est résolu par itérations et que le processus s'arrête lorsqu'un critère d'arrêt est atteint. Cela implique de répéter le processus plusieurs fois et cela se traduit souvent par un temps de calcul long. Cependant, il est nécessaire de le faire en un temps le plus court possible : puisque le marché change constamment, le processus de résolution doit donc être le plus efficace possible. En outre, et même si cela dépend des choix de la norme, des poids et de la régularisation, la fonction objective est généralement non linéaire et même non convexe. Cela rend difficile la recherche du minimum global puisqu'il est possible de rester coincé dans un minimum local

tout au long des itérations. L'algorithme d'évolution différentielle (ED) peut être une solution pour résoudre ce problème.

En ce qui concerne les choix de la fonction objective dans le cadre de la finance, les plus courants concernent les prix observés et les volatilités implicites (calculées numériquement via une itération de recherche de racine à partir du prix observé). La calibration avec la volatilité est plutôt naturelle car après un changement d'échelle, l'erreur entre les deux prix est faible. Cela explique aussi pourquoi les traders d'options ne citent souvent que la volatilité implicite et non le prix. Un autre avantage de choisir de calibrer par rapport à la volatilité implicite au lieu du prix est le fait qu'elle est indépendante du type d'option (call ou put). Dans notre cas, nous avons principalement utilisé la volatilité implicite, mais nous avons étudié les deux cas pour comparer les résultats.

Comme on peut voir dans [Pir19], il existe plusieurs méthodes de calibration :

1. Inversion d'un système non linéaire, avec un algorithme de recherche de racine (exemple : solver `Broyden1` dans python) ;
2. Optimisation minimisant la distance entre prix prédit et prix observé (exemple : optimisation stochastique ou non-linéaire par la méthode des moindres carrés) ;
3. RNA (RNA calibration ou RNA pricing + algorithme cherchant minimum global d'une fonction de perte).

Par la suite, nous ne nous concentrons que sur la calibration par des méthodes de machine learning, principalement les RNA.

3.2 L'axe Machine Learning

Aujourd'hui, le machine learning (ML, ou apprentissage automatique) est sans doute la technologie la plus performante et la plus largement utilisée pour résoudre des problèmes que les programmes simples ne peuvent pas résoudre. Par rapport aux algorithmes traditionnels qui suivent certaines règles prédéfinies, les algorithmes de machine learning reposent sur de vastes quantités de données observées dans la nature, générées par des humains ou par un autre algorithme. Une définition plus formelle est proposée par Mitchell : « Les programmes informatiques sont dit pouvant apprendre de l'expérience E , pour une classe de tâche T et une mesure de performance P , si sa performance sur les tâches en T , mesurées par P , s'améliorent avec l'expérience E ». Autrement dit, les algorithmes de ML permettent aux ordinateurs d'établir des règles à partir des données passées et en faire de la prédiction.

Dans cette section, on commence par une introduction des méthodes de ML couramment utilisées, dans un objectif de classification ou de régression et à l'aide d'un apprentissage supervisé ou non supervisé. Ensuite, des connaissances fondamentales de RNA sont présentées. Finalement on présente une structure de calibration, combinant une étape de pricing utilisant RNA et une étape de calibration utilisant l'algorithme d'ED pour chercher le minimum global de la fonction de perte.

3.2.1 Panorama du machine learning

Les méthodes de ML peuvent être distinguées par leur type d'apprentissage, supervisé ou non supervisé, et par leur objectif, de régression ou de classification. L'apprentissage supervisé (supervised learning en anglais) est une tâche d'apprentissage automatique consistant à apprendre une fonction de prédiction à partir d'exemples annotés, au contraire de l'apprentissage non supervisé, pour lequel les exemples ne sont pas annotés. La principale différence entre les algorithmes de régression et de classification est que les algorithmes de régression sont utilisés pour prédire les valeurs de variables continues telles que le prix, le salaire etc. et les algorithmes de classification sont utilisés pour prédire la modalité d'une variable qualitative ou discrète telles que le sexe, la nationalité, le nombre d'enfants etc. Les algorithmes des méthodes de ML ne se distinguent donc que par leur objectif, de classification ou de régression : le principe de l'algorithme reste identique.

Apprentissage supervisé

La classification à l'aide d'un apprentissage supervisé consiste à attribuer une classe à chacun des individus d'un jeu de données, connaissant la classe d'appartenance des individus des données d'apprentissage. La régression à l'aide d'un apprentissage supervisé consiste à prédire la valeur d'une variable associée à un individu, connaissant cette valeur pour les individus des données d'apprentissage.

Les principales méthodes de classification et de régression sont :

1. Régression linéaire et logistique ;
2. Arbre de décision (Decision Tree) et ses variantes (Random Forest, Gradient Boosted Decision Tree etc.) ;
3. Machines à Vecteurs de Support, connu sous le nom de SVM (Support Vector Machine) et SVR (Support Vector Regressor), sa version pour la régression ;
4. K plus proches voisins (k-NN, k-nearest neighbors) ;
5. Classification naïve bayésienne ;
6. RNA.

Apprentissage non supervisé

On parle de clustering lorsque les données proviennent d'un apprentissage non supervisé : on ne connaît pas la classe d'appartenance des individus ou la valeur de la variable d'intérêt pour ces individus. Il s'agit alors de former des groupes d'individus. Beaucoup d'algorithmes se basent sur la maximisation du rapport entre la variance inter-groupes et la variance intra-groupes. Les principales méthodes de clustering sont

1. Clustering hiérarchique ;
2. K-means et ses variantes k-medians etc. ;
3. Apprentissage par renforcement utilisant les RNA.

Le RNA est un sujet populaire dans le ML. Le développement de RNA commence depuis 1943, par la proposition d'un neurone de McCulloch-Pitts (aussi appelé modèle de MC)

[MP43]. En 1957, cela a ensuite évolué en neurone formel (aussi appelé perceptron) [Ros58]. En 1999, l'algorithme de perceptron [FS99], donnant automatiquement le poids et le biais du neurone (voir section 3.2.2), est proposé pour améliorer la performance du perceptron. Cet algorithme a aussi accéléré la naissance du perceptron multi-couche (MLP, multiple layer perceptron) et de l'algorithme de rétro-propagation par [RHW86] en 1986. Souvent la précision des prédictions du MLP est positivement corrélée avec sa complexité. Cependant, en raison de la limitation de la puissance de calcul des installations matérielles, à partir d'un moment on ne peut plus augmenter la performance en ajoutant plus de couches et plus de neurones. Pour résoudre ce problème, depuis 1980, en combinant les MLP et la logique mathématique, les réseaux de neurones convolutifs (CNN) se développent (voir [Fuk80] ou [FM82]). Cela a eu une grande réussite dans le traitement d'images et la reconnaissance vocale. En 2014, les réseaux adverses génératifs (GAN, generative adversarial networks), une classe d'algorithmes d'apprentissage non supervisé utilisant les RNA, ont été introduits par [Goo+14]. Ils permettent de générer des images avec un fort degré de réalisme. Très récemment, l'apprentissage par renforcement [KLM96] s'est aussi popularisé et s'est démarqué par son utilisation dans les jeux, la robotique et la conduite automatique. Jusqu'où nous mènera le ML ? Personne ne le sait.

3.2.2 Principe du RNA

Pour la calibration des modèles financiers, lorsqu'aucune méthode de ML n'a d'avantage évident et sachant qu'on peut générer autant de données qu'on veut selon les modèles financiers, on choisit dans un premier temps d'utiliser les RNA. Cependant, dans [Iva21], on voit aussi le potentiel des autres algorithmes de ML dans le pricing : cela mérite une étude approfondie.

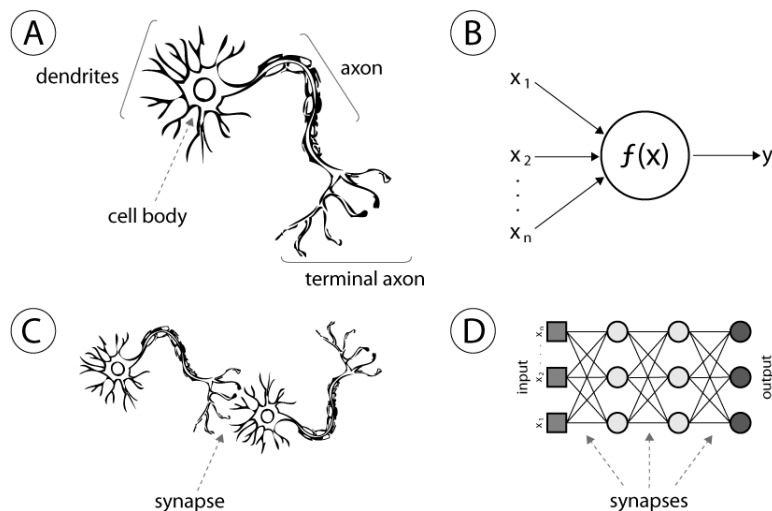


FIG. 4 – Parallèle entre le neurone biologique et le RNA, construit sur le principe du biomimétisme.

Les RNA sont inspirés des réseaux de neurones biologiques (A dans la Fig. 4). Une vraie cellule neuronale possède de nombreuses dendrites, qui peuvent recevoir une stimulation. Après traitement par le corps cellulaire, une réponse (Activation) est générée, qui est trans-

mise à l'axone puis envoyée à d'autres cellules. Le neurone artificiel (ⓑ) dans la Fig. 4) est très similaire et même plus simple. Il a de nombreuses entrées qui peuvent être écrites comme un vecteur $x = (x_1, x_2, \dots, x_n)$, tout comme les dendrites. En biologie, un neurone a différents degrés de sensibilité, donc dans le RNA une somme pondérée est effectuée, puis une fonction d'activation est utilisée pour obtenir une réponse non linéaire. Ainsi le corps cellulaire d'un neurone est une fonction f de x qui renvoie une valeur y :

$$y = f(x, \theta) = \varphi(w \cdot x + b) ,$$

où φ est appelée fonction d'activation, $\theta = (w, b) \in \mathbb{R}^n \times \mathbb{R}$. Différentes fonctions d'activation φ peuvent être utilisées, parmi lesquelles on peut citer :

- sigmoïde : $\varphi(x) = \frac{1}{1 + e^x}$;
- relu : $\varphi(x) = \max(0, x)$;
- tanh : $\varphi(x) = \tanh(x)$.

Il en existe plusieurs autres : le choix d'un type de fonction dépend de l'application. Ⓒ et Ⓓ dans la Fig. 4 indiquent que le réseau de neurones est constitué de neurones connectés.

En effet, quand on parle de RNA, souvent on parle de MLP, qui est un réseau en plusieurs couches, au sein desquelles une information circule de la couche d'entrée vers la couche de sortie uniquement (propagation directe ou feedforward). Chaque couche est constituée d'un nombre variable de neurones, les neurones de la dernière couche (dite « de sortie ») étant les sorties du système global. Un exemple illustratif de RNA est montré dans la Fig. 5. Ce MLP peut donc être écrit comme une fonction composite F renvoyant les sorties du système global :

$$F(x|\theta) = f^{(L)}(\dots f^{(2)}(f^{(1)}(x; \theta^{(1)}); \theta^{(2)}); \dots \theta^{(L)}) , \quad (15)$$

où $\theta^{(i)} = (w_i, b_i)$, w_i est une matrice de poids et b_i est un vecteur de biais. Pour plus de détails sur le perceptron, nous nous référons à [Ros58].

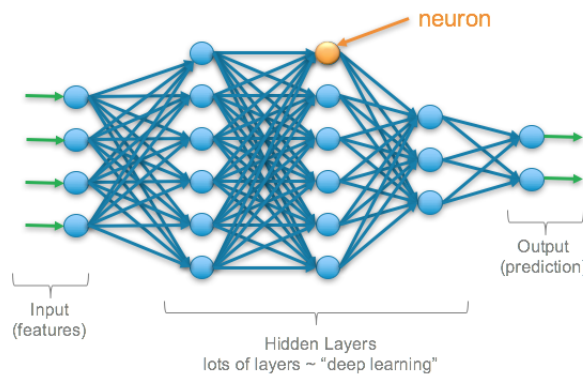


FIG. 5 – Illustration de RNA

La Fig. 6 détaille le principe d'un neurone et l'exemple 1 traite du cas d'un MLP à une couche cachée.

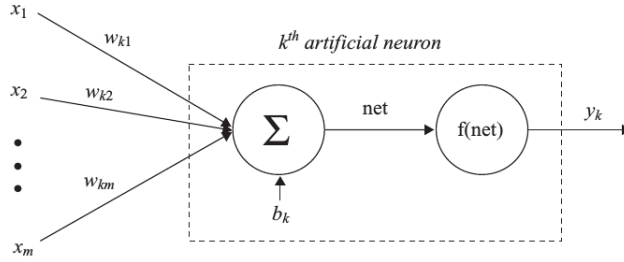


FIG. 6 – Un perceptron.

Exemple 1. *MLP à une couche cachée* Un MLP avec une couche cachée peut, par exemple, s'écrire comme suit,

$$\begin{cases} z_j^{(1)} = \varphi^{(1)}(\sum_i w_{ij}^{(1)} x_i + b_j^{(1)}), & \text{première couche,} \\ y(x) = \varphi^{(2)}(\sum_j w_j^{(2)} z_j^{(1)} + b^{(2)}), & \text{deuxième couche (sortie),} \end{cases}$$

avec w_j les poids inconnus, $\varphi(w_{1j}x_j + b_{1j})$ la fonction de base du neurone, $\varphi(\cdot)$ une fonction d'activation, i de 1 à n (le nombre de valeurs d'entrée), j de 1 à m (le nombre de neurones dans la couche cachée).

Par ailleurs, un modèle de réseau de neurones est une fonction qui apprend en minimisant une fonction de perte. Dans le cas de l'apprentissage supervisé, la fonction de perte L équivaut à une distance entre les prédictions du modèle $F(x|\theta)$ et les observations $f(x)$. Elle peut s'écrire sous la forme

$$L(\theta) := D(f(x), F(x|\theta)).$$

Dans les RNA, l'optimisation consiste à trouver la valeur minimale de la fonction de perte dans l'espace des paramètres par descente de gradient, avec l'algorithme de rétro-propagation (voir dans [RHW86]). Entraîner les RNA, c'est apprendre les poids et les biais optimaux dans l'équation (15) pour rendre la fonction de perte aussi petite que possible. Le processus d'entraînement des réseaux de neurones peut être formulé comme un problème d'optimisation :

$$\arg \min_{\theta} L(\theta|(X, Y)),$$

étant donné les paires entrée-sortie (X, Y) et une fonction de perte L définie par l'utilisateur. En supposant que l'ensemble des données d'apprentissage (X, Y) est suffisant pour définir la vraie fonction, les RNA avec suffisamment de neurones peuvent approximer cette fonction dans une certaine norme, par exemple la norme L^2 . Les RNA sont donc de puissants approximateurs de fonctions universelles et peuvent être utilisés sans supposer de relation prédéfinie entre l'entrée et le résultat.

Descente de gradient stochastique (DGS)

Grâce à la rétro-propagation [RHW86], la méthode de DGS est efficacement utilisée pour optimiser les (ω_i, b_i) des RNA. La DGS est une méthode numérique dérivée de la descente de gradient (DG) [Bot12]. Dans la DG, un gradient de la fonction de perte est calculé par rapport à chacun des paramètres pour mettre à jour les paramètres par étapes et trouver

le minimum global. La DGS est une simplification de la DG où un sous-ensemble aléatoire d'individus est choisi pour estimer le gradient à chaque étape pour limiter le calcul. Ainsi, au lieu de calculer le gradient sur les données complètes, il est estimé à chaque itération sur des sous-échantillons choisis au hasard dans l'échantillon total.

La minimisation de la fonction de perte L est utilisée pour mettre à jour les paramètres et les formules d'optimisation sont données par

$$\begin{cases} w^{(i+1)} \leftarrow w^{(i)} - \eta(i) \frac{\partial L}{\partial w} \\ b^{(i+1)} \leftarrow b^{(i)} - \eta(i) \frac{\partial L}{\partial b} \\ \text{pour } i = 0, 1, \dots, N_T, \end{cases}$$

où N_T est le nombre d'itérations, η le taux d'apprentissage (l'un des hyperparamètres du RNA) et $\theta = (w, b)$. Il faut choisir un point de départ $\theta^{(0)} = (\omega^{(0)}, b^{(0)})$ pour les itérations et il existe de nombreuses méthodes pour effectuer cette initialisation, qui peut avoir un impact important sur le résultat final.

3.2.3 Structure et conception des RNA pour le pricing

L'avantage des RNA est de pouvoir calculer les grecs très rapidement, en extrayant les informations de gradients depuis le RNA. Cependant, un RNA donne un bon résultat seulement s'il est bien entraîné. Or cela dépend des données et des hyperparamètres : avoir un réseau bien entraîné devient parfois un problème de l'art et de la chance.

Il y a en général deux façons d'utiliser les RNA pour la calibration :

1. On entraîne un réseau avec (S, K, r, T) comme paramètres d'entrée et $(\rho, \kappa, \gamma, \bar{v}, v_0)$ les paramètres du modèle de Heston comme sortie ;
2. On entraîne un réseau pour remplacer une des méthodes traditionnelles de pricing, puis on calibre avec un algorithme d'optimisation cherchant le minimum global de notre fonction objective : la distance entre le prix (ou volatilité) prédit(e) et le prix (volatilité) observé(e) du marché.

La première méthode ne marche généralement pas bien : cela n'a souvent pas de relation avec les structures des RNA, mais est dû au fait que la fonction objective n'est pas toujours convexe, mais stochastique. Cela signifie qu'il n'y a pas de bijection entre le prix et les jeux de paramètres, et qu'un prix peut correspondre à plusieurs combinaisons des paramètres possibles. Donc un RNA ne peut pas être bien entraîné dans ce cadre. Dans [Pir19] on peut voir un exemple montrant l'infaisabilité de cette méthode, que l'on utilise un CNN ou un MLP.

On choisit donc naturellement la deuxième méthode, comme dans [Liu+19], en construisant une étape de pricing avec un RNA (étape forward) et une étape de calibration cherchant un minimum global (étape backward).

3.2.3.1 Étape forward

La première partie d'un RNA de calibration, appelée l'étape forward, consiste à utiliser un RNA, sous la forme d'un MLP, pour apprendre la solution générée par différentes méthodes numériques (c'est-à-dire pour optimiser les paramètres des couches cachées) et mappe

ensuite l'entrée à la sortie d'intérêt (c'est-à-dire en négligeant les variables intermédiaires). Par exemple, afin d'approximer les volatilités implicites de Black-Scholes sur la base des paramètres d'entrée de Heston, deux méthodes numériques sont nécessaires, à savoir la méthode COS pour calculer les prix des options de Heston et l'algorithme de recherche de racine de Brent pour déterminer la VI correspondante. Comme présenté dans la Fig. 7, dans ce rapport, nous fusionnons ces deux étapes en utilisant un seul RNA. En d'autres termes, le Heston-IV-ANN (aussi appelé "RNA de pricing" dans ce rapport) est utilisé comme étape forward pour apprendre la correspondance entre les paramètres du modèle et la volatilité implicite. Après avoir formé le RNA, les paramètres trouvés sont utilisés pour faire des prédictions, dans ce cas, des volatilités implicites. Dans la phase de prédiction, le modèle peut être évalué par les erreurs entre les prédictions du modèle et les observations des données de test.

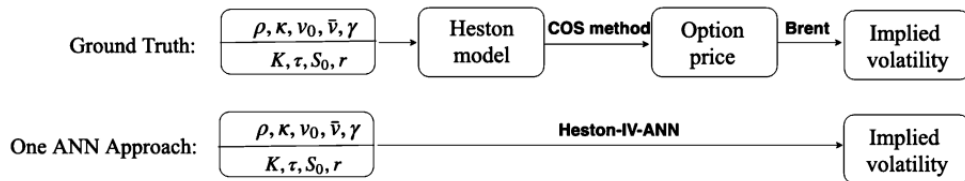


FIG. 7 – Principe du RNA de pricing.

Notons que l'augmentation du nombre de neurones ou l'utilisation d'une structure plus profonde peut conduire à de meilleures approximations, mais donne lieu à un réseau coûteux à calculer. Dans la section 4, on montre les résultats de plusieurs structures de RNA qu'on a testés. On nomme donc les RNA par l'ordre d'essai (RNA-(ordre d'essai)). Dans un premier lieu, on essaie de construire un RNA comme celui de [Liu+19]. Les hyperparamètres sélectionnés sont listés dans le Tab. 1. Ce RNA possède 4 couches cachées et 200 neurones pour chaque couche, en plus de la couche d'entrée (avec 8 neurones) et la couche de sortie linéaire (régression) (avec un neurone). Ainsi, nous obtenons 122 601 paramètres entraînaibles. L'étape forward est formée en minimisant une fonction de perte définie, qui est la MSE entre les valeurs prédites par le RNA et les valeurs réelles sur l'ensemble de données d'apprentissage. La structure globale est représentée sur la Fig. 8.

Paramètres	Options
Couches cachées	4
neurones (chaque couche)	200
Activation	ReLU
taux d'abandon (dropout rate)	0
Normalisation par lots (batch)	No
Initialisation	Glorot_uniform
Optimiseur	Adam
Taille du lot (batch size)	1024

TAB. 1 – Hyperparamètres du RNA-1. Le choix courant du modèle de RNA pricing et des hyperparamètres du RNA pour le pricing sont brièvement présentés dans [Pie20]. En général on fait une recherche aléatoire et une cross validation pour trouver les meilleurs hyperparamètres définissant le meilleur modèle RNA, comme fait dans [LOB19].

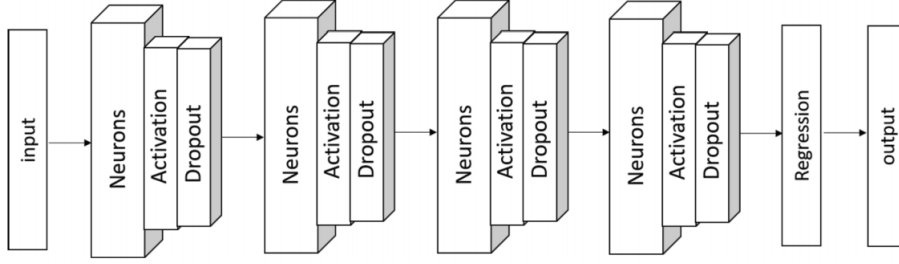


FIG. 8 – Structure de RNA.

Pour générer les données, nous avons simulé les paramètres du Tab. 3 par des lois unimodales dans les intervalles donnés, puis nous avons utilisé la méthode de pricing COS pour obtenir les prix de call européen, et enfin la méthode de Brent pour déterminer la volatilité implicite. Nous avons alors simulé environ 10^6 points de données.

Le RNA prend comme paramètres d'entrée $(m, r, T, \rho, \kappa, \bar{v}, \gamma, v_0)$ et renvoie en sortie une valeur approchée de la volatilité implicite σ de Black-Scholes. Comme mentionné dans le Tab. 1, l'optimiseur Adam est utilisé pour former le RNA sur l'ensemble de données générées. D'après [LOB19], le taux d'apprentissage initial est 10^{-3} et est divisé par 2 toutes les 500 époques (cycle d'apprentissage du RNA). L'apprentissage se compose de 3000 époques et l'algorithme a convergé. Une initialisation aléatoire adéquate est nécessaire à l'aide de `Glorot_uniform`, pour s'assurer que de bonnes valeurs de départ pour $\theta := (w, b)$ sont fournies. Le taux d'abandon est fixé à zéro.

3.2.3.2 Étape backward : évolution différentielle (ED)

Étant donné des options différentes (différents (S, K, r, T)), l'étape backward consiste à prédire leurs VI avec le RNA de pricing qu'on a entraîné à l'étape forward, et à trouver $(\rho, \kappa, \gamma, \bar{v}, v_0)$ les paramètres du modèle de Heston minimisant la distance de ces VI avec celles observées du marché, grâce à l'algorithme d'ED. C'est une méthode basée sur des mutations de la population servant à résoudre des problèmes d'optimisation qui ne nécessitent pas de différenciation.

La méthode comporte 4 phases : Initialisation, mutation, croisement et sélection. L'idée principale est de choisir une population initiale d'une certaine taille et d'apporter des modifications à chaque individu de la population pour construire de nouvelles populations. Les individus de ces nouvelles populations sont des candidats de la solution du problème d'optimisation et le but est de se rapprocher de la solution à chaque itération. Une fois l'algorithme arrêté, tous les individus de la population finale sont évalués à l'aide de la fonction objective et celui qui correspond le mieux au problème d'optimisation est sélectionné. Les populations sont des solutions possibles au problème d'optimisation, donc, dans notre cas, chaque individu de la population est un ensemble de paramètres d'un modèle financier. Nous utilisons le MSE (mean squared error) comme fonction cible pour minimiser la distance entre la volatilité implicite observée et la celle générée par le RNA dans l'étape forward. Nous décrivons maintenant chaque phase de l'algorithme.

1. Initialisation : une population de taille N_p est générée à partir de l'espace où les para-

mètres sont sélectionnées aléatoirement :

$$(\theta_1, \theta_2, \dots, \theta_{N_p}).$$

Chaque θ_i , $1 \leq i \leq N_p$, est appelé un individu et, dans notre cas, c'est un ensemble de paramètres d'un modèle financier.

2. Mutation : dans cette étape, une population de mutants est générée. Chaque individu est muté en ajoutant une différence échantillonnée au hasard appelée mutation différentielle.

$$\theta'_i = \theta_a + F \cdot (\theta_b - \theta_c),$$

où $1 \leq i \leq N_p$ et $\theta_a, \theta_b, \theta_c$ sont des individus sélectionnés dans la population avec la contrainte $a \neq i$. $F \in [0, \infty)$, appelé le poids différentiel, détermine la grandeur de la mutation. Ainsi, de grandes valeurs de F incrémentent le rayon de recherche, mais peuvent augmenter le temps de convergence de l'algorithme. Il existe plusieurs façons de choisir θ_a , les plus populaires étant `best1bin`, qui sélectionne le meilleur individu de l'itération précédente et `rand1bin`, qui sélectionne simplement un individu aléatoire de la population.

3. Crossover : dans cette étape, un critère est défini pour choisir entre l'individu initial ou son mutant correspondant et passer à l'étape suivante. Un seuil défini par l'utilisateur $0 \leq C_r \leq 1$, appelé possibilité de croisement (crossover), est défini. Ensuite, pour chaque indice $0 \leq i \leq N_p$, un nombre uniformément distribué $p_i \sim \mathcal{U}(0, 1)$ est sélectionné. Enfin, soit θ_i soit son mutant θ'_i est choisi comme suit :

$$\theta''_i = \begin{cases} \theta'_i, & \text{si } p_i \leq C_r \\ \theta_i, & \text{sinon.} \end{cases}$$

Notons que le paramètre C_r contrôle la quantité de mutants autorisée à l'étape suivante et qu'il s'agit finalement d'une probabilité. Plus C_r est élevé, plus la possibilité d'avoir des mutants à la génération suivante est grande. Dans cette phase, une population test $(\theta''_1, \theta''_2, \dots, \theta''_{N_p})$ est choisie.

4. Sélection : dans cette étape, la population test est comparée à la population au début de l'itération en utilisant la fonction objective pour choisir une population finale (pour l'itération) :

$$\theta_i \leftarrow \begin{cases} \theta''_i, & \text{si } L(\theta''_i) \leq L(\theta_i) \\ \theta_i, & \text{sinon.} \end{cases}$$

Puisque nous visons à trouver un minimum local, nous actualisons chaque θ_i individuellement (en choisissant entre le θ_i initial et son θ''_i correspondant dans la population d'essai) de sorte que nous obtenons le θ_i individuel qui donne la plus petite valeur après évaluation avec la fonction objective.

Les itérations sont effectuées en répétant les étapes 2 à 4 jusqu'à ce que l'algorithme converge ou qu'un critère d'arrêt prédéfini soit atteint. Une fois que cela se produit, l'individu de la population finale avec la plus petite valeur sous l'évaluation de la fonction objective est choisi comme solution Θ de notre problème d'optimisation :

$$\Theta = \arg \min_{\theta_i \in [1, N_p]} L(\theta_i).$$

4 Résultats numériques et discussion

Dans cette section nous allons décrire les résultats des différentes méthode de pricing puis des étapes forward et backward.

4.1 Méthodes de pricing

Black-Sholes

Pour modèle de Black-Sholes, on voit que la performance d'un arbre trinomial est aussi bien que la solution analytique (Fig. 9).

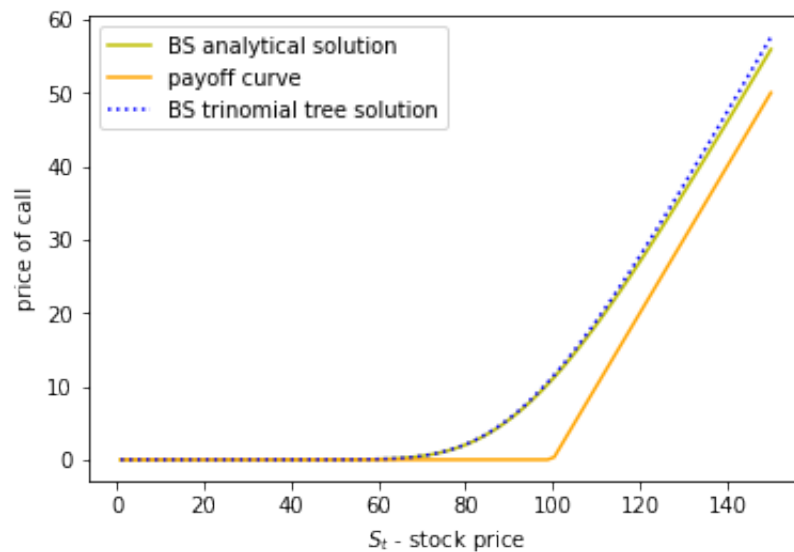


FIG. 9 – Arbre trinomial v.s. solution analytique du modèle de Black-Sholes. Avec $T = 1$, $r = 0.06$, $K = 100$, $\sigma = 0.2$, dividende = 0.0.

Heston

Avec T le temps jusqu'à l'échéance et

$$S_0 = 100, \quad T = 1, \quad r = 0.02, \quad K = 100,$$

$$\kappa = 3, \quad \bar{v} = 0.04, \quad v_0 = 0.0625, \quad \rho = 0.7, \quad \sigma = 0.6,$$

la performance de différentes méthodes contenant le temps de calcul et les prix de call simulés est présentée dans le Tab. 2. La méthode COS, avec $N = 150$ et l'intervalle de troncature $[a, b]$ (tel que $a = -L\sqrt{T}$, $b = L\sqrt{T}$, $L = 5$), est la plus rapide, et la méthode de Monte-Carlo est la plus lente. On observe aussi, qu'étant des méthodes basées sur la transformée de Fourier, la méthode de Carr-Madan et la méthode COS ont une performance comparable. Enfin, on voit dans la Fig. 10 que les 4 méthodes donnent des valeurs similaires.

Méthode	Temps de calcul (en s)	Prix de call
Monte-Carlo	4.577	9.055
Formule semi-analytique	0.348	9.133
Carr-Madan	0.033	9.050
Méthode COS	0.008	9.050

TAB. 2 – Performance des méthodes de pricing du modèle de Heston

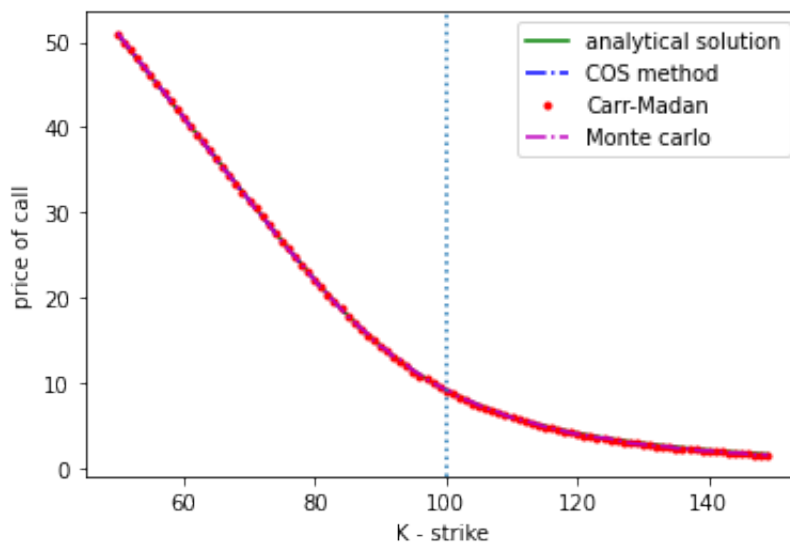


FIG. 10 – Comparaison des méthodes de pricing du modèle de Heston, montrant que les 4 méthodes sont comparables (les 4 courbes se superposent).

Choix de la méthode de pricing

L'étape forward consiste à entraîner un RNA pour remplacer les méthodes de pricing traditionnelles. Vu que le RNA fait partie de l'apprentissage supervisé, on va d'abord générer des données d'apprentissage. Comme dans le Tab. 2 la méthode de Carr-Madan et la méthode COS ont trouvé le même prix, on aimerait comparer les deux méthodes et choisir la plus rapide tout en gardant une performance robuste, pour générer les données d'apprentissage.

On génère des données en fixant $S_0 = 100$ et selon le Tab. 3.

RNA	Paramètres	Intervalle	Méthode
RNA entrée	Moneyness, $m = S_0/K$	[0.6, 1.4]	uniforme
	Temps jusqu'à la maturité (an), T	[0.05, 3]	uniforme
	Taux sans risque, r	[0, 0.05]	uniforme
	Corrélation, ρ	[-0.9, 0]	uniforme
	Taux de réversion, κ	(0, 3]	uniforme
	Volatilité de volatilité, γ	(0.01, 0.8]	uniforme
	Variance longue, \bar{v}	(0.01, 0.5]	uniforme
	Variance initiale, v_0	(0.05, 0.5]	uniforme
-	Prix de call européen	(0, 60)	COS
RNA sortie	Black-Sholes VI,	(0, 0.76)	Brent

TAB. 3 – Plage d'échantillonnage pour les paramètres de Heston. On utilise la distribution uniforme pour générer les données, la méthode COS pour le pricing et la méthode Brent, un algorithme de recherche de racine pour trouver les VI.

Parfois la méthode de Carr-Madan n'arrive pas à produire un résultat numérique (NaN pour Not a Number), à cause de 0 au dénominateur de certains calculs. La Fig. 11 montre la distribution des paramètres après avoir supprimé les individus produisant des NaN. On voit par exemple que cette méthode ne marche pas bien pour les call à échéance longue (T grand), et non plus pour ceux qui ont une grande variance (ν grand) : en effet le nombre de prix de call obtenus diminue lorsque ces paramètres augmentent.

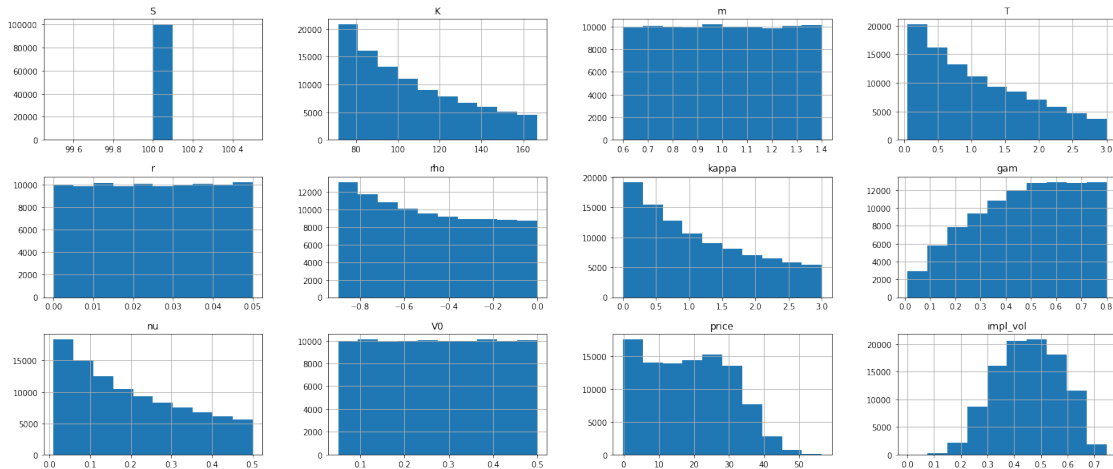


FIG. 11 – Histogrammes des paramètres dans le modèle de Heston à l'aide de la méthode de pricing Carr-Madan, avec 10^5 données

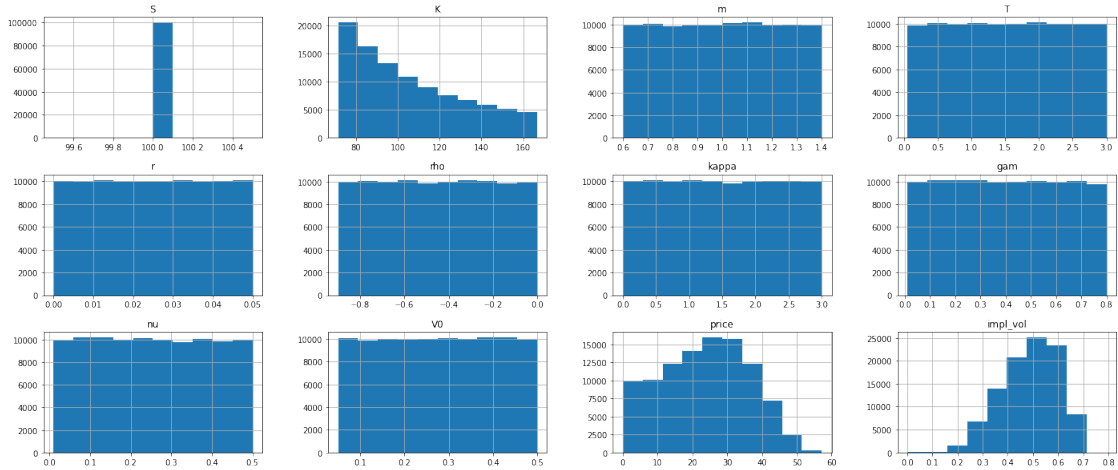


FIG. 12 – Histogrammes des paramètres dans le modèle de Heston à l’aide de la méthode de pricing COS, avec 10^5 données

La performance d’un RNA dépend de la qualité de ses données d’apprentissage. Contrairement à la méthode de Carr-Madan, dans la Fig. 12, on voit que la méthode COS est valable pour toutes les valeurs des paramètres dans les intervalles choisis. Elle est donc meilleure que la méthode de Carr-Madan et de plus elle est assez robuste. On utilise donc la méthode COS pour déterminer les prix de call.

4.2 Étape forward

Vu que *Tensorflow* est très bien documenté et qu’il est plus facile à utiliser que *pyTorch*, on l’utilise pour construire notre RNA de pricing. Remarquons que en général il faut utiliser une recherche aléatoire (random search) pour trouver les meilleurs hyperparamètres d’un RNA, comme fait dans [LOB19]. Par manque de temps et de ressources informatiques, on n’a pas pu réaliser cette étape, très chronophage. On a adopté deux types de RNA dont les structures sont présentées par la suite.

RNA-1

On essaie d’abord le modèle de RNA proposé dans [Liu+19]. Les valeurs des hyperparamètres du RNA-1 sont présentées dans le Tab. 1. Avec un total de 10^6 données (10% de données de validation, 90% de données d’apprentissage), on a entraîné le réseau avec 3000 époques, et une réduction du taux d’apprentissage (decay learning rate) de 1/2 chaque 500 époques. Le temps d’apprentissage est 6000 s (100 min).

Comme on peut voir dans la Fig. 13, avec une réduction du taux d’apprentissage, l’erreur sur les données d’apprentissage diminue, mais l’erreur sur les données de validation ne diminue pas en même temps que lui. On en conclut que notre modèle a sur-appris, ce qui signifie que le modèle est trop compliqué par rapport au problème qu’on veut résoudre. Cette conclusion est confirmée par le Tab. 4, en MAPE (mean absolute percentage error) : l’erreur sur les données de validation n’est pas dans le même ordre que celle des données d’apprentissage.

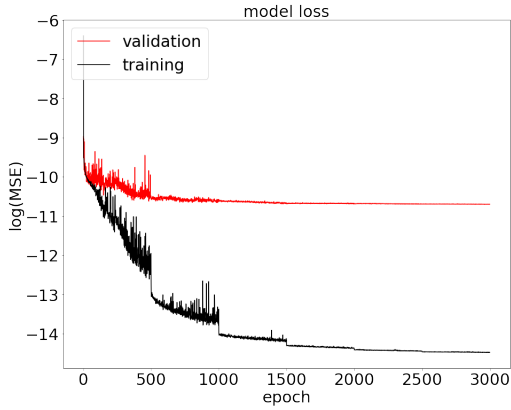


FIG. 13 – Variation de perte du RNA-1.

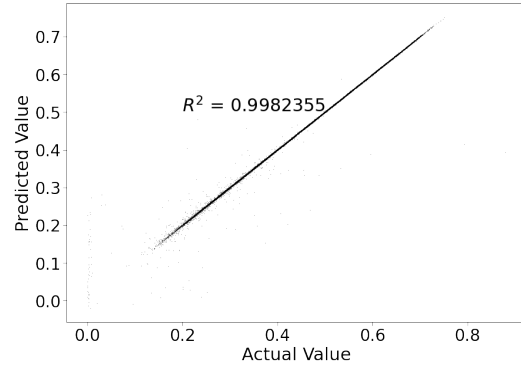


FIG. 14 – VI simulée v.s. VI prédite par le RNA-1.

ANN-1 volatility			
	Loss (MSE)	MAE	MAPE
train	5.15×10^{-7}	5.12×10^{-4}	0.15
test	2.3×10^{-5}	6.83×10^{-4}	2.66

TAB. 4 – Résultat d’apprentissage du RNA-1

En traçant le nuage de points des valeurs réelles et des valeurs prédites par le RNA-1 (Fig. 14), on voit que RNA-1 ne prédit pas bien quand la VI est petite (les valeurs prédites sont alors dispersées), malgré un R^2 qui semble assez grand.

RNA-2

Dans un deuxième temps, on essaie un modèle plus simple, comme proposé dans [Pie20]. Ce réseau, appelé RNA-2, possède trois couches cachées, qui ont respectivement 120, 120 et 95 neurones. Avec un total de 10^5 données (10% de données de validation, 90% de données d’apprentissage), il est entraîné avec 200 époques, un taux d’apprentissage constant valant 10^{-4} , et les hyperparamètres donnés dans le Tab. 5. Le temps d’apprentissage est 400 s.

Paramètres	Options
Couches cachées	3
neurones (chaque couche)	(120, 120, 95)
Activation	ReLU
taux d’abandon (dropout rate)	0
Normalisation par lots (batch)	No
Initialisation	Glorot_uniform
Optimiseur	Adam
Taille du lot (batch size)	64

TAB. 5 – Valeurs des hyperparamètres du RNA-2.

Dans la Fig. 15, on voit que ce réseau a bien appris, car à partir la 150-ième époque environ, l’erreur sur les données d’apprentissage diminue mais celle des données de validation reste inchangée. Au vu du Tab. 6 et de la Fig. 16, ce réseau a presque la même performance

que le RNA-1, avec des erreurs de prédiction similaires et des MAPE proches sur les données de validation.

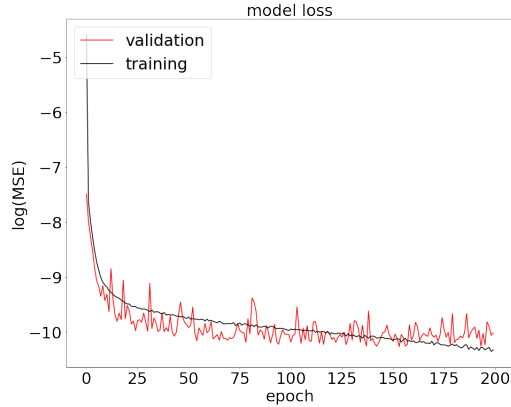


FIG. 15 – Variation de perte du RNA-2.

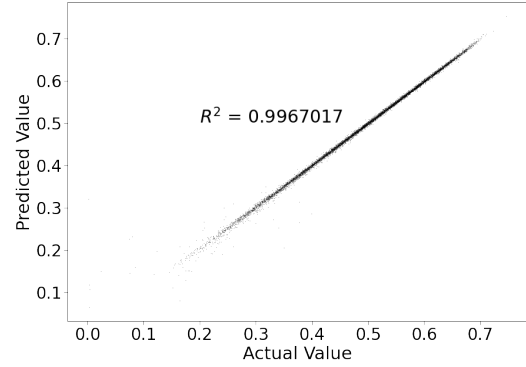


FIG. 16 – VI simulée v.s. VI prédite par le RNA-2.

ANN-2 volatility			
	Loss (MSE)	MAE	MAPE
train	3.3×10^{-5}	2.5×10^{-3}	3.99
test	4.5×10^{-5}	2.7×10^{-3}	2.02

TAB. 6 – Résultat d'apprentissage du RNA-2

4.3 Étape backward

On calibre avec une surface simulée de VI, associée à 2 prix de strikes et 3 valeurs de maturités, uniformément choisis dans leur intervalle (Tab. 7). Cela signifie que pour un jeu de paramètres $(\rho^0, \kappa^0, \gamma^0, \bar{v}^0, v_0^0)$, il existe 6 points de VI.

-	Paramètres	Intervalle	Échantillon
Données du marché	Moneyness, $m = S_0/K$	[0.85, 1.15]	2
	Temps jusqu'à la maturité (an), T	[0.5, 2.0]	3
	Taux sans risque, r	0.03	Fixé
	Prix de call européen	(0.0, 0.6)	-
Black-Sholes	Volatilité implicite	(0.2, 0.5)	6

TAB. 7 – Intervalle des cotations simulées du marché

Paramètres	Options
Taille de population	25
Stratégie de recherche	best1bin
Mutation	0.8
Probabilité de croisement	0.7
Tolérance de convergence	0.01

TAB. 8 – Paramètres de l'algorithme d'ED.

Paramètres	Intervalle de recherche
ρ	[-0.85, -0.05]
κ	[0.1, 2.0]
γ	[0.05, 0.75]
\bar{v}	[0.05, 0.45]
v_0	[0.05, 0.45]

TAB. 9 – Intervalle de recherche pour l'algorithme d'ED.

L'algorithme est construit avec les paramètres donnés dans le Tab. 8 et les intervalles de recherche du Tab. 9. On définit un critère de convergence à la manière de python : $\text{std}(\text{pop_loss}) \leq \text{atol} + \text{tol} * \text{abs}(\text{mean}(\text{pop_loss}))$, où pop_loss est une distance entre la VI prédite et la vraie VI (fonction L de la sect. 3.2.2), tol et atol des constantes fixées. Ce critère permet d'obtenir des individus bien regroupés.

Calibration avec la méthode COS pour le pricing

Pour tester l'algorithme d'ED, on l'utilise pour calibrer le modèle de Heston avec la méthode de pricing COS. On voit dans le Tab. 10 que, avec une surface de VI et les formules exactes de pricing, il est quand même difficile de retrouver exactement les mêmes paramètres du modèle. La plus grande erreur peut atteindre jusqu'à presque 17%. En effet, l'exactitude des résultats pourrait être améliorée par l'augmentation de la taille de population (plus de 35 ou 50), mais cela ralentit l'étape de calibration aussi. La variation de chaque paramètre est discutée dans [Liu+19], qui explique pourquoi l'erreur de κ est plus grande que les autres.

Résultats de la calibration avec méthode COS					
	ρ	κ	γ	\bar{v}	v_0
Paramètres initiaux	-0.364	0.510	0.427	0.300	0.250
Paramètres trouvés par l'ED	-0.341	0.425	0.446	0.318	0.250
MAE	$2.3 \cdot 10^{-2}$	$8.5 \cdot 10^{-2}$	$1.9 \cdot 10^{-2}$	$1.8 \cdot 10^{-2}$	$2.5 \cdot 10^{-4}$
MAPE	6.314%	16.743 %	4.438 %	6.073 %	0.100%

TAB. 10 – ED : Résultats de la calibration avec la méthode COS pour le pricing, faisant référence d'évaluation de l'algorithme d'ED. Temps de calcul : 1141.6 s.

Calibration avec un RNA pour le pricing

Comme proposé dans [Liu+19], pour calibrer avec un RNA de pricing, on ajoute une constante de régularisation $\bar{\lambda} = 10^{-7}$ pour la fonction de perte (voir l'équation (14)).

Résultats de la calibration avec RNA-2					
	ρ	κ	γ	\bar{v}	v_0
Paramètres initiaux	-0.364	0.510	0.427	0.300	0.250
Paramètres trouvés par l'ED	-0.150	0.137	0.050	0.245	0.239
MAE	0.214	0.373	0.377	$5.5 \cdot 10^{-2}$	$1.1 \cdot 10^{-2}$
MAPE	58.828 %	73.279%	88.279%	18.375%	4.251 %

TAB. 11 – ED : Résultats de la calibration avec le RNA-2 pour le pricing. Temps de calcul : 203 s.

La prédiction du RNA est rapide, vu l'écart de vitesse de calcul entre Tab. 10 et Tab. 11. Mais comme vu dans le Tab. 11, le résultat de la calibration utilisant RNA-2 est mauvais, car 3 paramètres calibrés sur 5 ont plus de 50% d'écart avec les vrais. Cela est probablement dû au fait que le réseau n'est pas parfaitement entraîné pour simuler les deux étapes dans

Fig. 7. Même si l’erreur d’entraînement est petite (MAE d’ordre 10^{-4} , MAPE autour de 2%), les petites erreurs pourront dévier le minimum global et donc donner un mauvais résultat.

D’ailleurs, contrairement à [Liu+19], on n’a pas pu faire la calibration en moins d’une seconde. Cela est dû au fait de ne pas maîtriser le parallélisme, qui consiste à mettre en œuvre des architectures numériques permettant de traiter des informations de manière simultanée.

Calibration avec un RNA pour la calibration

Une autre façon d’utiliser un RNA pour la calibration est de faire comme le 1. de la section 3.2.3 : On entraîne un réseau avec (m, r, T) comme paramètres d’entrée et $(\rho, \kappa, \gamma, \bar{v}, v_0)$ les paramètres du modèle de Heston comme sortie. Ce cas reste en discussion, car il est simple à implémenter et des résultats sont montrés dans [Pie20]. Dans ce cas là, la performance d’un CNN est meilleure que celle d’un MLP. Cependant, l’erreur peut quand même atteindre jusqu’à 13% pour un paramètre. Cette méthode ne donne pas non plus de résultats satisfaisants.

5 Conclusion et perspectives

Ce travail de stage est composé de 2 parties principales : une partie de compréhension et d'introduction aux méthodes de pricing sur les modèles de Black-Scholes et de Heston, et une partie de mise en oeuvre des RNA et de l'algorithme d'optimisation (ED) pour la calibration des paramètres de ces modèles financiers. Les objectifs du stage ont été remplis, avec plusieurs remarques et perspectives qui peuvent être mises en évidence.

Après avoir résumé les méthodes principales de pricing du modèle de Heston, j'ai introduit dans ce rapport la méthode de Carr-Madan et la méthode COS, plus récentes et plus performantes en exactitude et en temps de calcul. Pour la calibration des paramètres du modèle, plutôt que de me baser sur les prix de call ou de put, j'ai préféré, comme dans [Liu+19], une fonction de coût basée sur la distance entre la VI prédite par les RNA de pricing et la VI déterminée par la méthode de COS et de Brent. J'ai introduit et implémenté l'algorithme d'ED pour la recherche, par minimisation d'une fonction de coût, des paramètres du modèle financier.

Remarquons d'abord que pour un RNA, sa qualité dépend fortement de son résultat d'entraînement, et que sa performance dépend fortement des données d'apprentissage qu'on lui fournit. Pour un RNA bien entraîné d'une structure complexe, la phase d'apprentissage est souvent chronophage. Par ailleurs, plusieurs problèmes peuvent survenir et nécessiter de recommencer l'apprentissage, impliquant une consommation d'énergie supplémentaire : oublier de sauvegarder le modèle de RNA, effacer une variable par erreur, coupure d'électricité ou coupure d'internet. Pour les données d'apprentissage, chaque fois qu'on change le modèle de pricing (Heston sans dividendes, Heston avec saut ou autre), il faut aussi recommencer l'apprentissage à zéro. Deuxièmement, par manque de temps et de ressources informatiques (l'utilisation d'un serveur distant aurait pu s'avérer utile), nous n'avons pas pu appliquer la recherche aléatoire pour déterminer la meilleure structure du RNA de pricing : cela peut s'effectuer de manière similaire à [LOB19]. Par ailleurs, bien que l'algorithme d'ED ait l'avantage de chercher un minimum global, son défaut principal est le temps de calcul. Cela pourrait être résolu par le parallélisme et nécessite du développement informatique. Il existe probablement d'autres méthodes de calibration plus efficaces et robustes à découvrir.

Ma contribution à ce travail de recherche porte principalement sur le répertoriage et la mise en oeuvre des différentes méthodes de pricing et de calibration pour les modèles financiers de Black-Scholes et de Heston. Plus particulièrement, j'ai activement participé à l'introduction pour le laboratoire LMRS aux méthodes par RNA. Ce travail a donc permis au LMRS de s'initier aux RNA dans un objectif de calibration de modèles financiers. Les principales méthodes de pricing traditionnelles ont été décrites et de nouvelles ont été introduites : la méthode de Carr-Madan et la méthode COS. Ensuite j'ai aussi introduit pour le LMRS mes connaissances fondamentales en machine learning et surtout j'ai acquis et introduit les connaissances de base et les connaissances plus avancées sur les RNA : la construction d'un RNA et la recherche des meilleurs hyperparamètres de MLP. Par ailleurs, j'ai illustré avec plusieurs exemples les avantages et inconvénients de chacune des méthodes et chacun des modèles utilisés. Les codes ont été effectués en python et livrés sous forme d'une archive bien organisée et documentée. Cette archive contient donc

- toutes les méthodes de pricing listées ci-dessus ;
- des exemples de RNA en *pyTorch* et en *Tensorflow* ;
- la calibration de modèles de Heston en utilisant des RNA.

Mon travail s'est alors situé tout d'abord dans un objectif théorique de compréhension des modèles probabilistes et des difficultés computationnelles qu'ils engendrent. Ensuite la mise en oeuvre informatique des méthodes d'apprentissage a permis une comparaison de ces méthodes tant au point de vue théorique que computationnel.

Références

- [BS73] Fischer BLACK et Myron SCHOLES. « The Pricing of Options and Corporate Liabilities ». In : *Journal of Political Economy* 81.3 (1973), p. 637-654. ISSN : 00223808, 1537534X. URL : <http://www.jstor.org/stable/1831029> (visité le 16/06/2022).
- [Bot12] Leon BOTTOU. « Stochastic Gradient Descent Tricks ». In : *Neural Networks, Tricks of the Trade, Reloaded*. T. 7700. Lecture Notes in Computer Science (LNCS). Springer, jan. 2012, p. 430-445. URL : <https://www.microsoft.com/en-us/research/publication/stochastic-gradient-tricks/>.
- [CM99] Peter CARR et Dilip MADAN. « Option valuation using the fast Fourier transform ». In : *Journal of computational finance* 2.4 (1999), p. 61-73. URL : https://engineering.nyu.edu/sites/default/files/2018-08/CarrMadan2_0.pdf.
- [CIR05] John C. COX, Jonathan E. INGERSOLL JR et Stephen A. ROSS. « A theory of the term structure of interest rates ». In : *Theory of valuation*. World Scientific, 2005, p. 129-164.
- [CRR79] John C. COX, Stephen A. ROSS et Mark RUBINSTEIN. « Option pricing : A simplified approach ». In : *Journal of Financial Economics* 7.3 (sept. 1979), p. 229-263. URL : https://www.sciencedirect.com/science/article/pii/0304405X79900151/pdf?md5=ed89637331f4d9b8b2648cb9d3b735dc&pid=1-s2.0-0304405X79900151-main.pdf&_valck=1.
- [DD07] Francine DIENER et Marc DIENER. *Black-Scholes comme limite de CRR*. Rapp. tech. Université de Nice - Sophia Antipolis, 2007, chap.8. URL : <https://math.unice.fr/~diener/L3MASS08/chap8.pdf>.
- [FO09] F. FANG et C. W. OOSTERLEE. « A Novel Pricing Method for European Options Based on Fourier-Cosine Series Expansions ». In : *SIAM Journal on Scientific Computing* 31.2 (2009), p. 826-848. DOI : [10.1137/080718061](https://doi.org/10.1137/080718061). eprint : <https://doi.org/10.1137/080718061>. URL : <https://mpa.ub.uni-muenchen.de/8914/>.
- [Fan10] Fang FANG. « The COS Method : An Efficient Fourier Method for Pricing Financial Derivatives ». Thèse de doct. Friedrich-Alexander-Universit, Germany, 2010. URL : <https://core.ac.uk/download/pdf/301638164.pdf>.
- [FS99] Yoav FREUND et Robert E. SCHAPIRE. « Large margin classification using the perceptron algorithm ». In : *Machine learning* 37.3 (1999), p. 277-296.
- [Fuk80] Kunihiko FUKUSHIMA. « Neocognitron ». In : *Scholarpedia* 2.1 (1980), p. 1717.
- [FM82] Kunihiko FUKUSHIMA et Sei MIYAKE. « Neocognitron : A self-organizing neural network model for a mechanism of visual pattern recognition ». In : *Competition and cooperation in neural nets*. Springer, 1982, p. 267-285.

- [Gil51] J. GIL-PELAEZ. « Note on the inversion theorem ». In : *Biometrika* 38.3-4 (déc. 1951), p. 481-482. ISSN : 0006-3444. DOI : [10.1093/biomet/38.3-4.481](https://doi.org/10.1093/biomet/38.3-4.481). eprint : <https://academic.oup.com/biomet/article-pdf/38/3-4/481/718851/38-3-4-481.pdf>. URL : <https://doi.org/10.1093/biomet/38.3-4.481>.
- [Goo+14] Ian GOODFELLOW et al. « Generative adversarial nets ». In : *Advances in neural information processing systems* 27 (2014).
- [Hes93] Steven L. HESTON. « A Closed-Form Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options ». In : *The Review of Financial Studies* 6.2 (avr. 1993), p. 327-343. ISSN : 0893-9454. DOI : [10.1093/rfs/6.2.327](https://doi.org/10.1093/rfs/6.2.327). eprint : <https://academic.oup.com/rfs/article-pdf/6/2/327/24417457/060327.pdf>. URL : <https://doi.org/10.1093/rfs/6.2.327>.
- [Iva21] Codruț-Florin IVAȘCU. « Option pricing using Machine Learning ». In : *Expert Systems with Applications* 163 (2021), p. 113799. ISSN : 0957-4174. DOI : <https://doi.org/10.1016/j.eswa.2020.113799>. URL : <https://www.sciencedirect.com/science/article/pii/S0957417420306187>.
- [JA20] Dushko JOSHESKI et Mico APOSTOLOV. « A Review of the Binomial and Trinomial Models for Option Pricing and their Convergence to the Black-Scholes Model Determined Option Prices ». In : *Econometrics* 24.2 (2020), p. 53-85. DOI : [doi:10.15611/eada.2020.2.05](https://doi.org/10.15611/eada.2020.2.05). URL : <https://doi.org/10.15611/eada.2020.2.05>.
- [KLM96] Leslie Pack KAEHLING, Michael L. LITTMAN et Andrew W. MOORE. « Reinforcement learning : A survey ». In : *Journal of artificial intelligence research* 4 (1996), p. 237-285.
- [KR91] Bardia KAMRAD et Peter RITCHKEN. « Multinomial Approximating Models for Options with k State Variables ». In : *Management Science* 37.12 (1991), p. 1640-1652. URL : https://www.researchgate.net/publication/227446266_Multinomial_Approximating_Models_for_Options_with_k_State_Variables.
- [Kar13] Olle KARLSSON. *The Black-Scholes Equation and Formula*. 2013. URL : <https://www.diva-portal.org/smash/get/diva2:623701/FULLTEXT01.pdf>.
- [Kol15] Marek KOLMAN. « Option Pricing Using Fourier Transform with a Focus on the Cosine Expansion ». In : *Available at SSRN 2704074* (2015). URL : https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2704074.
- [LL12] Damien LAMBERTON et Bernard LAPEYRE. *Introduction au calcul stochastique appliqué à la finance*. Paris : ellipses, 2012.
- [LOB19] Shuaiqiang LIU, Cornelis W OOSTERLEE et Sander M BOHTE. « Pricing options and computing implied volatilities using neural networks ». In : *Risks* 7.1 (2019), p. 16.
- [Liu+19] Shuaiqiang LIU et al. « A neural network-based framework for financial model calibration ». In : *Journal of Mathematics in Industry* 9.1 (2019), p. 1-28. URL : <https://doi.org/10.1186/s13362-019-0066-7>.

- [MCC98] Dilip B. MADAN, Peter P. CARR et Eric C. CHANG. « The variance gamma process and option pricing ». In : *Review of Finance* 2.1 (1998), p. 79-105. URL : <https://engineering.nyu.edu/sites/default/files/2018-09/CarrEuropeanFinReview1998.pdf>.
- [MP43] Warren S. MCCULLOCH et Walter PITTS. « A logical calculus of the ideas immanent in nervous activity ». In : *Bulletin of mathematical biophysics* 5.4 (1943), p. 115-133.
- [Pie20] Pascal PIERROT. *Machine Learning methods for option pricing and model calibration, Study carried out by the Data Science Practice*. Rapp. tech. Coperneec, 2020. URL : <https://canopee-group.com/wp-content/uploads/2020/12/Machine-Learning-methods-Coperneec.pdf>.
- [Pir19] Olivier PIRONNEAU. « Calibration of Heston Model with Keras ». hal-02273889. Août 2019. URL : <https://hal.sorbonne-universite.fr/hal-02273889>.
- [Ros58] Frank ROSENBLATT. « The perceptron : a probabilistic model for information storage and organization in the brain. » In : *Psychological review* 65.6 (1958), p. 386.
- [Rou13] Fabrice ROUAH. *The Heston model and its extensions in Matlab and C# Fabrice Douglas Rouah ; [foreword by Steven L. Heston]*. Wiley finance series. Hoboken, N.J. : John Wiley & Sons, Inc., 2013.
- [Rou] Fabrice Douglas ROUAH. *Euler and Milstein Discretization*. URL : <https://frouah.com/finance%20notes/Euler%20and%20Milstein%20Discretization.pdf>.
- [RHW86] David E. RUMELHART, Geoffrey E. HINTON et Ronald J. WILLIAMS. « Learning representations by back-propagating errors ». In : *Nature* 323.6088 (1986), p. 533-536.
- [Sch10] Martin SCHMELZLE. « Option pricing formulae using Fourier transform : Theory and application ». In : *Preprint* (2010). URL : <https://pfadintegral.com/docs/Schmelzle2010%20Fourier%20Pricing.pdf>.
- [Zay16] Sami ZAYANI. *Liens entre le modèle binomial et les équations de Black-Scholes*. 2016. URL : <https://archipel.uqam.ca/8952/1/M14463.pdf>.

Glossaire

A | C | D | G | M | O | P | R | S

A

actif Un actif financier est un titre ou un contrat, généralement transmissible et négociable (par exemple sur un marché financier), qui est susceptible de produire à son détenteur des revenus ou un gain en capital, en contrepartie d'une certaine prise de risque. En général, on peut séparer les actifs en deux catégories : les actifs sans risque (par exemple les obligations, livrets d'épargne...) et les actifs risqués (par exemple les actions, les devises...). **3**

C

calibration La calibration (ou le calibrage) consiste à déterminer à partir des données de marché, les paramètres du modèle financier (ici modèle de Heston) qui permettent le mieux de retrouver les prix du marché. **1**

call Le call est une option d'achat sur un instrument financier. C'est un contrat qui permet à son souscripteur d'acquérir l'instrument concerné, appelé sous-jacent, à un prix fixé (strike) à l'avance et à une date déterminée (maturité). On parle de « call européen » si le souscripteur peut exercer son droit uniquement à la date de maturité. **5**

coupon zéro Les obligations à coupon zéro ne génèrent pas de coupons durant toute leur durée de vie. (voir dans « obligation »).

couverture Une couverture est un investissement qui protège les positions financières ouvertes d'une situation risquée. La couverture a pour but de minimiser ou de compenser les risques de perte de valeur des actifs considérés. Cette stratégie, aussi appelée « hedging », peut se révéler particulièrement intéressante en temps de crise sur les marchés. **3**

D

drift En théorie des probabilités, drift est aussi appelé la dérive stochastique, qui est le changement de la valeur moyenne d'un processus stochastique (aléatoire). **6**

G

grec Les lettres grecques ou grecs sont les instruments de base de la gestion financière des options. Elles découlent des principaux modèles d'évaluation d'option. Par exemple, le delta : $\delta = \frac{\partial P}{\partial S}$. **26**

M

maturité En finance, la maturité, ou maturité résiduelle, désigne le temps qui sépare la date à laquelle une obligation est émise, et la date à laquelle la valeur nominale de cette obligation est remboursée. En revanche, le terme « maturité » est un anglicisme, le terme adéquat est échéance. **5**

O

obligation Une obligation est un morceau de dette émis par une entreprise, une collectivité territoriale ou un État. En plus d'un capital qui sera restitué à l'acheteur de l'obligation à échéance fixe (en général, entre 5 à 30 ans), le débiteur s'engage à le rémunérer périodiquement (tous les ans, ou tous les trimestres notamment) selon un taux d'intérêt fixe ou variable (versement des coupons). Le taux d'intérêt qui vous rémunère est appelé un coupon. [6](#)

option Une option est un produit dérivé qui établit un contrat entre un acheteur et un vendeur. L'acheteur de l'option obtient le droit, et non pas l'obligation, d'acheter (call) ou de vendre (put) un actif sous-jacent à un prix fixé à l'avance (strike), pendant un temps donné ou à une date fixée (maturité), tandis que le vendeur a l'obligation de racheter cet actif au prix d'exercice si l'acheteur décide d'exercer l'option. [3](#)

P

payoff En finance, le mot anglais pay-off (ou payoff) se traduit généralement par bénéfice en français. Dans le cas d'une option, cette expression peut désigner aussi bien un profit qu'une perte. Le pay-off qualifie le rendement intrinsèque d'une option, après déduction de la prime d'acquisition. Ce pay-off varie en fonction des fluctuations des marchés à terme. Sa valeur réelle est calculée soit à sa date d'exercice, soit à sa date d'échéance. [5](#)

portefeuille Un portefeuille (en finance) désigne une collection d'actifs financiers détenus par un établissement ou un individu. La valeur d'un portefeuille est calculée par le produit de la valeur des actifs et leur quantité. Un portefeuille est autofinancé si le changement de sa valeur est uniquement lié au changement du prix des actifs (pas de changement de quantité). [9](#)

pricing Le pricing (ou la tarification en français), en mathématiques financières, est le processus par lequel on fixe, en utilisant des modèles financiers, le prix auquel des options se vendront. [1](#)

produits dérivés Un produit dérivé est un instrument financier : dont la valeur fluctue en fonction de l'évolution du taux ou du prix d'un autre produit appelé sous-jacent ; qui requiert peu ou pas de placement initial ; dont le règlement s'effectue à une date future. Ce produit consiste en un contrat entre deux parties, un acheteur et un vendeur, qui fixe des flux financiers futurs fondés sur ceux d'un actif sous-jacent, réel ou théorique, généralement financier. [3](#)

put Le put ou l'option de vente est une option contractuelle de vente par laquelle deux parties s'accordent pour échanger un actif (sous-jacent) à un prix fixé (strike) à une date prédéterminée (maturité). On parle de « put européen » si le souscripteur peut exercer son droit uniquement à la date de maturité. [5](#)

R

RNA Réseaux de Neurones Artificiels, une des méthodes d'apprentissage automatique et d'intelligence artificielle (IA) qui permet à la machine d'imiter la façon dont les humains acquièrent certains types de connaissances. [1](#)

S

sous-jacent Un actif sous-jacent (souvent appelé sous-jacent) est un actif sur lequel on peut porter une option ou plus largement un produit dérivé. Il peut être financier (actions, obligations, bons du Trésor, contrats à terme, devises, indices boursiers...) ou physique (matières premières agricoles ou minérales...). 3

strike Le strike désigne le prix d'exercice d'une option, qui correspond au prix fixé dans le contrat pour l'acquisition ou la cession du sous-jacent. 5

A Annexe DD&RS

Dans le cadre de la politique de développement durable de l'ENSIIE, une dimension de Développement Durable et Responsabilité Sociétale (DD&RS), présentant une description des politiques de développement durable et responsabilité sociétale de l'établissement de stage, doit être intégrée dans les rapports de stage et mémoires.

Pour ce faire, je me suis appuyée sur les documents et pages web disponibles sur le site de l'Université de Rouen Normandie et sur un entretien d'une heure avec Fabien THOUMIRE, responsable du Service Développement Durable et Responsabilité Sociétale de l'Université de Rouen Normandie.

A.1 Développement durable

A.1.1 Les pressions environnementales

Le secteur : Recherche et Enseignement supérieur

Sous la double tutelle de l'Université de Rouen Normandie et du CNRS, le LMRS se trouve dans le secteur d'activité de la recherche et de l'enseignement supérieur. À ces deux domaines, s'ajoute pour une université le domaine de la vie de campus universitaire. Les trois principaux aspects de ce secteur qui donnent des pressions sur l'environnement sont :

1. la mobilité : les déplacements du personnel et des étudiants de leur domicile à leur lieu de travail ou les déplacements professionnels (conférence, collaborations...);
2. la gestion des déchets : parc informatique, déchets chimiques, déchets papiers, autres déchets;
3. la consommation d'énergie : chauffage, électricité, centre de calcul, fabrication du parc informatique et autres outils de recherche.

Le secteur de la recherche et de l'enseignement supérieur est un secteur d'activités très large. Faisant partie du secteur tertiaire, son impact se situe plus principalement sur les enjeux abiotiques que biotiques. L'impact biotique existe de manière marginale : la modification de l'environnement (construction de parkings, tonte des pelouses, ...). Les enjeux abiotiques concernent les trois principaux aspects listés ci-dessus. En effet, la mobilité des chercheurs engendre la production de gaz à effets de serre et participe au réchauffement climatique; elle engendre aussi la production de particules fines toxiques polluant l'environnement. Par ailleurs, l'utilisation de produits chimiques, de matériel informatique et de tout autre produit (tel que le papier) participe à l'exploitation des ressources matérielles et au réchauffement climatique (déforestation). Elle engendre aussi des déchets qui ont un impact sur l'environnement (toxicité, ressources énergétiques supplémentaires pour le traitement des déchets). Enfin, la consommation d'énergie participe à l'exploitation des ressources énergétiques.

D'un point de vue courant, le secteur de la recherche et de l'enseignement supérieur n'est pas un secteur qui pollue l'environnement directement, comme l'industrie et les usines, mais indirectement, de par l'exploitation des ressources. Il existe tout de même des points à améliorer sur les aspects cités.

Il existe beaucoup de réglementations environnementales auxquelles le secteur est soumis. Ces réglementations sont imposées par la loi, en voici quatre exemples :

1. un établissement de recherche et d'enseignement supérieur doit produire un bilan carbone tous les 3 ans ;
2. le décret tertiaire¹ impose aux établissements à usage tertiaire de diminuer leur consommation d'énergie d'au moins 40% d'ici 2030.
3. l'élaboration d'un plan de mobilité² permettant de régir l'organisation des déplacements des personnes ou marchandises ;
4. l'interdiction de rejeter certains produits chimiques directement dans les égouts.

L'établissement : Université de Rouen Normandie

L'Université de Rouen Normandie s'étale sur 7 campus universitaires en Seine-Maritime et dans l'Eure. Cela engendre des déplacements entre les différents campus et lieux d'habitation. C'est la principale pression de l'établissement sur l'environnement.

La spécificité de l'Université de Rouen Normandie dans le secteur d'activité qui la concerne est la recherche. Les produits de laboratoires engendrent des déchets quotidiens, cliniques ou même radioactifs. Pour chaque type de déchet, le traitement est associé à un prestataire : l'Université doit suivre les filières adaptées. En particulier les déchets d'Activités de Soins à Risques Infectieux (DASRI) font l'objet d'une réglementation particulière. Le parc informatique de l'Université de Rouen Normandie dispose de plus de 10000 ordinateurs. Dans le domaine de recherche du Machine Learning, il y a nécessité de renouveler le matériel assez souvent (environ tous les 3 ans) mais pour le reste du parc informatique, l'amortissement peut se prolonger. L'équipement des laboratoires (spectrophotomètre, microscope électronique à balayage, sonde atomique) sont aussi des spécificités de l'établissement et ne sont pas achetés sur des critères environnementaux mais plutôt des critères de performance. Un autre exemple de spécificité de l'établissement concerne certaines recherches portant sur la période glaciaire et nécessitant des déplacements en Antarctique, ce qui engendre inévitablement d'autres pressions environnementales.

L'équipe : LMRS

J'ai effectué mon stage dans l'équipe Statistique du LMRS. Les pressions environnementales et la politique DDRS du laboratoire est commune aux trois équipes. Le LMRS n'a pas de particularités supplémentaires sur les pressions environnementales que celles du secteur. Il s'agit donc principalement des pressions liées aux trois aspects cités en début de section et plus particulièrement les déplacements et l'utilisation de l'informatique. De plus malgré la numérisation de la plupart des documents, l'utilisation du papier n'a pas beaucoup diminué. La plupart des enseignants-chercheurs préfèrent encore le format papier que ce soit pour les documents d'enseignements ou les documents de recherche.

1. <https://www.legifrance.gouv.fr/jorf/id/JORFTEXT000038812251>

2. https://www.legifrance.gouv.fr/codes/section_lc/LEGITEXT000023086525/LEGISCTA000023069061/

A.1.2 Politique, stratégie et communication sur les enjeux environnementaux

L'établissement : Université de Rouen Normandie

L'Université de Rouen Normandie a fait partie des premières universités françaises à obtenir le label DD&RS – Développement Durable et Responsabilité Sociétale³, en septembre 2020. Elle est engagée depuis plus d'une dizaine d'années dans une politique en faveur du développement durable à l'aide du service Développement durable et Responsabilité sociétale⁴, constitué de 9 personnes (le plus gros effectif pour une Université française), dont Fabien Thoumire est le responsable⁵.

Une nouvelle feuille de route validée par le Conseil d'Administration tout récemment fait de la gestion de l'environnement (qualité de l'air, eau, consommation d'énergie, mobilité, gestion des déchets) l'un des 6 principaux axes de l'Université. Plusieurs actions sont mises en place par l'établissement pour diminuer les pressions environnementales.

Actions de l'établissement : Tout d'abord en ce qui concerne la mobilité, un système de recensement des voyages en avion, train ou autre vient d'être instauré pour contrôler les émissions de gaz à effet de serre de 5 laboratoires ; l'objectif étant de l'étendre aux 39 laboratoires de l'Université. Le plan de mobilité de l'établissement a pour objectif d'inciter le personnel et les étudiants au covoiturage, à l'utilisation du vélo ou des transports en commun. Une action en ce sens a été organisée lors de mon stage : l'établissement a prêté des vélos durant un mois au personnel souhaitant tester ce mode de transport. Un autre exemple concerne les déplacements en conférences des chercheurs de l'Université : au lieu d'effectuer un déplacement d'une semaine par an à Seattle, les chercheurs d'un laboratoire ont décidé de faire un déplacement de 2 semaines une fois tous les deux ans et de faire en distanciel la rencontre l'année où ils ne se déplacent pas. Le télétravail est aussi encouragé dans certaines circonstances.

Ensuite, concernant la gestion des déchets, l'Université s'est engagée dans une démarche de quantification de la pollution chimique. Lors du remplacement d'équipement de laboratoire, l'Université initie la récupération des métaux et autres matières recyclables de ces équipements. Principalement, pour assurer une gestion du parc informatique dans un objectif de développement durable, l'Université allonge la durée d'amortissement des ordinateurs et engage une réflexion quant au recyclage des ordinateurs obsolètes. Enfin, concernant la consommation d'énergie, l'Université est engagée depuis 15 ans sur la rénovation des bâtiments parfois très anciens. Une directive du Ministère impose de réduire la consommation de chauffage en limitant à 19 degrés l'hiver. Malgré cette mesure, la facture de l'Université de Rouen Normandie va passer de 4 millions d'euros à 7,5 millions d'euros (augmentation du prix de l'énergie). Du côté des enseignements, l'établissement s'est engagée dans un service de ressources numériques nommé **Universitice**, avec en particulier l'accès en ligne aux documents d'enseignements des enseignants utilisant ce service.

3. <https://www.label-ddrs.org/>

4. <https://www.univ-rouen.fr/actualites/le-ddrs-au-coeur-de-nos-missions/>

5. <https://www.univ-rouen.fr/temoignages/fabien-thoumire-service-developpement-durable-et-responsabilite-societale/>

Actions du coeur de métier : Par ailleurs, l'Université intègre les enjeux environnementaux dans sa stratégie et son activité. Elle s'est engagée, en vue de la biodiversité, à la gestion différenciée des espaces verts, et ce depuis 2013 : l'arrêt de l'utilisation des produits chimiques pour le désherbage et la tonte moins fréquentes des espaces verts permet en effet à la biodiversité de se développer plus naturellement. Les laboratoires de l'Université sont depuis peu encouragés à effectuer leur bilan carbone. D'ailleurs le LMRS dans lequel j'ai effectué mon stage va proposer son premier bilan carbone d'ici la fin de l'année 2022. Dans sa nouvelle maquette d'enseignement, l'Université de Rouen Normandie a imposé des heures de formation DD&RS : ainsi chaque filière d'enseignement reçoit une sensibilisation à ce sujet. Enfin des formations sur la biodiversité, l'énergie, le réchauffement climatique ou bien sur le développement durable dans l'enseignement supérieur sont proposées au personnel de l'établissement. Enfin l'université donne un soutien aux initiatives pour une recherche responsable et pour les appels à projet dont les objectifs sont le développement durable.

Lieu de travail : LMRS

Concernant mon laboratoire de recherche, il existe une politique d'achat pour les appareils informatiques : le LMRS souscrit une garantie de 5 ans pour les nouveaux appareils et interdit de les éliminer avant la fin de garantie. D'autre part, le LMRS encourage le tri sélectif : chaque bureau de membre permanent est équipé de deux poubelles. Le LMRS préconise les déplacements en transports en commun et incite au transport ferroviaire plutôt qu'aérien pour les déplacements. Enfin comme déjà évoqué ci-dessus, le LMRS conçoit en ce moment-même son propre bilan carbone ce qui permet d'anticiper et de gérer au mieux les pressions environnementales.

Afin de faciliter la transition écologique, le LMRS a rédigé une page web (Transition écologique) de guide pour ses membres. On y trouve ainsi des documents comme : un guide d'action complet et une présentation expliquant les impacts environnementaux du numérique. La sensibilisation au sein du LMRS s'effectue principalement par l'envoi de mails d'informations, de sollicitation à la participation à des formations ou des activités, et des mails d'annonces des séminaires. Par contre, personnellement je n'en ai reçu aucune pendant mon stage.

Mon implication

J'ai limité au maximum mes impressions de documents. J'ai aussi limité, malgré mon sujet, l'utilisation de serveurs de calculs, en listant les tâches de calcul clairement à l'avance pour éviter des calculs non nécessaires. J'ai effectué au moins une journée de télétravail par semaine. Lors de mon stage, j'ai apporté à mes collaborateurs directs (encadrants et collègues) ce point de vue DD&RS, et j'ai pu sensibiliser l'équipe de travail sur ce sujet. Je me suis rendue compte que le développement durable n'est pas simplement des règles ou des gestes imposés, mais plutôt une mode de vie qu'il faut adopter.

J'ai d'ailleurs quelques idées concernant des améliorations possibles. La métropole de Rouen pourrait développer des transports en commun plus pratiques reliant plus rapidement les différents campus. L'université et le CROUS pourraient proposer davantage de logements près des campus. L'université pourrait donner les ordinateurs obsolètes aux étudiants qui

n'ont pas d'ordinateur, leur permettant d'effectuer des tâches non performantes (recherche internet, etc.).

A.2 Responsabilité sociétale

La responsabilité sociétale des entreprises (RSE) également appelée responsabilité sociale des entreprises, est définie par la commission européenne comme l'intégration volontaire par les entreprises de préoccupations sociales et environnementales à leurs activités commerciales et leurs relations avec les parties prenantes. En d'autres termes, la RSE c'est la contribution des entreprises aux enjeux du développement durable. Une entreprise qui pratique la RSE va donc chercher à avoir un impact positif sur la société tout en étant économiquement viable. La norme ISO 26000, standard international définit le périmètre de la RSE autour de sept thématiques centrales :

1. la gouvernance de l'organisation ;
2. les droits de l'homme ;
3. les relations et conditions de travail ;
4. l'environnement ;
5. la loyauté des pratiques ;
6. les questions relatives aux consommateurs ;
7. les communautés et le développement local.

Remarquons que le LMRS n'est pas une entreprise, mais un laboratoire qui est affilié principalement à l'Université de Rouen Normandie. On parle donc de la Responsabilité sociétale des entreprises de l'Université de Rouen Normandie. Dans le cadre de sa politique de responsabilité sociétale, l'Université de Rouen Normandie a effectué à l'aide de sondages, une grande étude sur la qualité de vie de ses employés et de ses étudiants. Les conclusions de ces études sont en cours de finalisation. Sur le point de l'égalité homme/femme, la loi impose un plan d'action et l'établissement possède un groupe de travail (constituée de trois femmes) sur ce sujet.