

Mémoire présenté devant l'ISFA
pour l'obtention du diplôme de la filière Actuariat
et l'admission à l'Institut des Actuaire
le 01/09/2023

Par : **Adrien Cortes**

Titre : **Accélération des techniques d'analyse de facteurs par
méthodes d'apprentissage statistique et application
à un GSE**

Confidentialité : NON OUI (Durée : 1 an 2 ans)

Les signataires s'engagent à respecter la confidentialité indiquée ci-dessus

Membres présents du jury de la filière

Entreprise : Milliman

Nom : Stéphane LOISEL

Signature :

*Membres présents du jury de l'Institut
des Actuaire*

Directeur du mémoire en entreprise :

*Nom : BENNOUNA Mohammed,
TONDOLO Léo*

Signature :

**Autorisation de publication et de
mise en ligne sur un site de
diffusion de documents actuariels
(après expiration de l'éventuel délai de
confidentialité)**

Secrétariat :

Signature du responsable entreprise

Bibliothèque :

Signature du candidat

Table des matières

Résumé	4
Abstract	5
Remerciements	6
Introduction	7
I Motivations et cadre de l'étude	8
1 Générateurs de scénarios économiques et calcul ALM dans le cadre de solvabilité 2	9
1.1 Rappels réglementaires	9
1.2 Monde risque neutre, monde réel	14
1.3 Générateur de scénarios économiques	15
1.4 Asset and Liability Management (ALM)	18
1.5 Motivation du mémoire	20
1.6 Définitions élémentaires en finance	20
2 Un exemple introductif, les taux nominaux	23
2.1 Éléments théoriques du modèle	23
2.2 Calibrage du modèle	24
2.3 Statistiques descriptives du Shift	25
II Analyse de facteurs : accélération de la procédure par méthodes d'apprentissages	29
3 Etat de l'art en analyse de sensibilité	30
3.1 Définitions et concepts de base	30
3.2 Propriétés souhaitables d'un indice de sensibilité	31
3.3 Indices de Sobol et de Shapley	32
3.4 Théorie de l'information	35
4 Mutual information copula estimator : un nouveau cadre pour l'analyse de sensibilité	43
4.1 Lien entre copules et information mutuelle	43
4.2 MICE : le modèle	44
4.3 Applications de MICE	47

III	Prise en compte de la dynamique des données financières	52
5	Pré-requis en machine learning	53
5.1	Les réseaux de neurones récurrents (RNN)	53
5.2	Modèles génératifs, un focus sur les autoencodeurs variationnels	56
6	Classification des données de marché avec un VAE	61
6.1	Volatilité des options Call	61
6.2	Volatilités de Swaption	64
7	Etude de l'impact des données de marché	68
7.1	Impact des données d'options de Call	68
7.2	Impact des données d'options sur Swaptions	73
IV	Application aux paramètres embarqués d'un GSE	75
8	Le Shift	77
8.1	Output de CHESSE	77
8.2	Résultats ALM	79
9	GLMM	83
9.1	Inflation : Gaussian LIBOR Market Model (GLMM) et modèle de Vasieck à deux facteurs (2VF)	83
9.2	Étude historique des paramètres distributionnels et détermination de cibles	84
9.3	Résultats de l'impact des cibles historiques sur la VIF, en fonction du contexte de taux	85
10	Matrice de corrélations	91
10.1	Création de la matrice de corrélations	91
10.2	Range pour stresser les paramètres	93
10.3	Résultats numériques	96
11	Hiérarchie des paramètres embarqués d'un GSE	98
11.1	Résultats	98
V	Extension de notre étude et autres domaines d'applications possibles	101
12	Capacités et limites de nos outils pour anticiper le comportement des modèles dans des situations inédites	102
12.1	Générations de données de volatilité d'options sur Call	102
12.2	Génération de volatilité de Swaptions	104
12.3	Prédictions des sorties GSE dans un contexte nouveau	105
	Conclusion	107
	Bibliographie	107
	Table des figures	111
	Liste des tableaux	114

Annexes	117
A Perturber plusieurs paires de la matrice de corrélation pour une analyse de sensibilité	117
A.1 Stress simultané de plusieurs corrélations	117
B Analyse de sensibilité de la graine des Générateurs de nombre Pseudo Aléatoires	119
B.1 Propriétés désirables d'un générateur de nombres quasi-aléatoires	119
B.2 Présentation des générateurs de nombre quasi-aléatoires sélectionnés	121
B.3 Problématiques autour des générateurs de nombres quasi-aléatoires	123
B.4 Applications pratiques et résultats	124

Résumé

Les générateurs de scénarios économiques sont des outils indispensables pour répondre aux exigences réglementaires en assurances. Ils permettent de projeter des grandeurs économiques sur des horizons de plusieurs décennies. Leur objectif est de fournir aux modèles d'actif-passif de quoi valoriser les engagements futurs des assureurs, ainsi que leurs actifs futurs. Les générateurs de scénarios économiques modélisent les taux, les actions, les obligations risquées, ainsi que les corrélations entre ces différentes grandeurs. Ils font donc naturellement appel à des modèles plus ou moins complexes. Certains paramètres, appelés paramètres embarqués, ne sont parfois calibrés qu'une fois par an, comme la matrice de corrélations. Il est alors légitime de se demander s'il ne faudrait pas plutôt recalculer ces paramètres embarqués lorsque les conditions de marché changent, et quantifier ces changements de conditions de marché.

Une analyse de sensibilité peut être réalisée avec des méthodes classiques (indices de Sobol, de Shapley,..). Mais dans ce mémoire, nous proposons d'aller un peu plus loin, et de proposer des approches nouvelles pour calculer des indices de sensibilité. Le but est de pouvoir hiérarchiser l'importance des paramètres embarqués. Nous proposons également une méthodologie pour classifier et générer des données de marché, afin d'avoir une idée du comportement de notre générateur de scénarios économiques en fonction du contexte économique, et en fonction des inputs que nous lui donnons.

Mots clefs : *Machine Learning*, Analyse de sensibilité, Générateur de scénarios économiques

Abstract

Economic scenario generators are essential tools for meeting regulatory requirements in insurance. They allow projecting economic variables over horizons of several decades. Their objective is to provide asset-liability models with the means to value insurers' future liabilities and assets. Economic scenario generators model interest rates, equities, risky bonds, as well as correlations between these different variables. They naturally rely on more or less complex models. Some parameters, known as embedded parameters, are sometimes calibrated only once a year, such as the correlation matrix. It is then legitimate to wonder whether these embedded parameters should be recalculated when market conditions change, and to quantify these changes in market conditions.

Sensitivity analysis can be performed using classical methods (Sobol indices, Shapley indices, etc.). But in this thesis, we propose to go a step further and suggest new approaches for calculating sensitivity indices. The aim is to be able to prioritize the importance of embedded parameters. We also propose a methodology for classifying and generating market data, in order to understand the behavior of our economic scenario generator depending on the economic context, and depending on the inputs we provide it with.

Keywords : *Machine Learning*, Sensitivity Analysis, Economic Scenario Generator

Remerciements

Je souhaite exprimer ma gratitude envers l'équipe de recherche de Milliman ainsi qu'à sa direction pour avoir mis à ma disposition tout le matériel nécessaire à la réalisation de ce mémoire.

Je tiens tout particulièrement à remercier Stéphane Loisel, dont les conseils ont été déterminants non seulement pour la réalisation de ce mémoire, mais également pour orienter mon choix de carrière.

Mes remerciements chaleureux vont à Léo Tondolo et Mohammed Bennouna, mes tuteurs, pour leurs remarques pertinentes et leur disponibilité tout au long de ce travail. Je n'oublie pas de remercier Laetitia Orta, qui m'a formé à l'utilisation des générateurs de scénarios économiques malgré son changement d'entreprise en cours de route.

Un grand merci à Alexandre Boumezoued pour son soutien et ses précieux conseils tout au long de cette année. Je suis également reconnaissant envers Timothée Fabre, qui m'a d'abord parlé de Milliman, ainsi que Julien Vedani, qui m'a présenté l'entreprise.

Je tiens à exprimer toute ma gratitude à Didier Rullière, avec qui j'ai eu de nombreuses discussions enrichissantes sur des sujets de recherche. Didier, vous m'avez permis de mieux comprendre comment avancer dans un sujet de recherche, et je vous suis infiniment reconnaissant pour cela, ainsi que pour vos conseils éclairés.

Je remercie également Franck Gabriel, qui m'a apporté de nombreux éclairages sur la partie modèle en machine learning présentée dans ce mémoire.

Enfin, mes plus sincères remerciements vont à mes parents Vincent et Véronique pour leur soutien indéfectible, sans lequel tout cela n'aurait pas été possible.

Introduction

Un générateur de scénarios économiques (GSE) constitue un ensemble de modèles (tels que les taux, les actions, les spreads de crédit, l'inflation, etc.) élaborés dans le but de procéder à des simulations aléatoires de diverses trajectoires envisageables pour l'évolution des marchés financiers. Ces trajectoires exercent une influence sur la valeur et les performances des actifs détenus par l'entité d'assurance. Au coeur de cette approche, les paramètres embarqués du GSE désignent les paramètres inhérents aux modèles en question, qui demeurent constants au fil du temps sans être recalibrés. Comprendre l'impact de ces paramètres sur les résultats des calculs actuariels revêt alors une importance particulière.

L'analyse de sensibilité, telle que décrite dans la littérature, offre un moyen de déterminer l'importance relative des paramètres embarqués. Cependant, dans un souci d'approfondissement, nous avons innové en développant un tout nouveau cadre pour réaliser cette analyse. Notre approche repose sur l'utilisation de modèles génératifs de machine learning, qui nous permettent de calculer des indices de sensibilité de manière plus robuste et précise. Grâce à cette approche nous sommes en mesure de mieux appréhender l'impact de chaque paramètre sur les résultats des calculs actuariels et d'obtenir des informations plus riches pour une prise de décision.

Nous avons déployé des efforts considérables pour quantifier les divers régimes de marchés, car il est essentiel de tenir compte des données de marché dans toute analyse de sensibilité. Cette approche nous permet de prévenir toute surestimation de l'importance d'un paramètre embarqué.

Ce mémoire s'articule en plusieurs parties :

- La **première partie** présente le cadre réglementaire et vise aussi à introduire au lecteur les principaux problèmes que nous allons rencontrer dans ce mémoire. C'est-à-dire, traiter les distributions multimodales en analyse de sensibilité, et classifier les données de marché.
- La **deuxième partie** présente l'état de l'art de l'analyse de sensibilité et un modèle novateur que nous proposons pour estimer des indices de sensibilité.
- La **troisième partie** se concentre sur la classification des données de marché. L'utilisation d'autoencodeur variationnels Bayésiens permet de caractériser les différents régimes de marchés, et de générer de nouvelles données.
- La **quatrième partie** applique les résultats aux paramètres embarqués d'un GSE, et permet de répondre à la problématique de notre étude.
- La **cinquième partie**, plus courte, montre comment notre étude peut s'étendre à d'autres problèmes connexes.

Première partie

Motivations et cadre de l'étude

Chapitre 1

Générateurs de scénarios économiques et calcul ALM dans le cadre de solvabilité 2

La Directive 2009/138/CE, connue sous le nom de Solvabilité 2, est entrée en vigueur le 1^{er} janvier 2016. Elle visait à réformer le secteur de l'assurance en Europe. Inspirée des réformes bancaires telles que Bâle 2, elle détermine les exigences en fonds propres pour les compagnies d'assurance et de réassurance en fonction des risques encourus. Cette directive a eu un impact direct sur la gestion des investissements des assureurs. En effet, les exigences en capital plus strictes les ont amenés à privilégier des produits moins risqués, avec potentiellement des rendements plus faibles. Dans ce chapitre, nous examinons le contexte et les conséquences de Solvabilité 2 dans le secteur de l'assurance.

1.1 Rappels réglementaires

L'industrie de l'assurance est soumise à différents risques, tels qu'un changement de contexte de taux, l'antisélection¹ ou les aléas de la nature. Pour pouvoir s'assurer de la résistance des assureurs face à de tels risques, le Conseil de l'Union européenne [49], qui est l'une des sept institutions de l'Union Européenne, a mis en place entre 1973 et 1979 des règles prudentielles via deux directives appelées "Solvabilité 1".

1.1.1 Solvabilité 1

La **solvabilité** est la capacité d'une entreprise à faire face à ses engagements. Pour cela, elle dispose de différentes **ressources** incluant notamment les fonds propres, les obligations ou les actions. Afin de protéger au mieux les assurés, l'union Européenne a adopté en 1973 une directive sur les assurances non vie (73/239/CEE) puis en 1979 une directive sur les assurances vie (79/267/CEE). Des mises à jour successives de ces directives en 1988, 1990 et 1992 ont permis de libéraliser en Europe l'assurance et aux assureurs de vendre leurs différents produits à tous les états membres.

Avec les années, le nombre de sinistres a augmenté, de même que l'inflation. Cela aurait du engendrer une augmentation des fonds propres requis pour assurer un niveau de solvabilité suffisant. Cependant, cette situation n'était pas prévue par Solvabilité 1. Afin de mieux comprendre les enjeux, nous détaillons les 3 piliers de Solvabilité 1.

Le pilier 1 formalise l'exigence d'un niveau minimum de provisions techniques. Ces dernières doivent être égales au montant actualisé des engagements futurs de l'assureur. Elles

1. L'antisélection est un phénomène dans lequel les individus présentant un risque plus élevé ont une plus grande probabilité de souscrire une assurance par rapport à ceux présentant un risque plus faible. Cela peut entraîner un déséquilibre financier pour les assureurs.

doivent être évaluées avec des hypothèses suffisamment prudentes. **Le pilier 2** impose à l'assureur d'avoir des actifs de suffisamment bonne qualité pour être en mesure de rembourser ses assurés, son personnel et l'état. Dans ce cadre précis, la réglementation donne une liste des actifs que l'assureur peut acquérir ainsi que la part maximale de son portefeuille qu'il peut investir dans chacun de ces actifs. **Le pilier 3** exige la mise en place d'une marge de solvabilité supplémentaire pour permettre à l'assureur de faire face à des chocs imprévus (tels qu'une crise économique). Cette marge de solvabilité et son calcul sont définis dans les textes de la réglementation et nous renvoyons le lecteur désireux d'en apprendre plus vers ces derniers. La figure 1.1 présente les trois principaux piliers sur lesquels repose solvabilité 1.

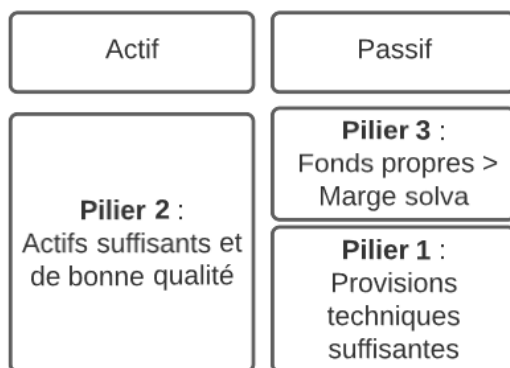


FIGURE 1.1 – Les trois piliers de solvabilité 1

La directive Solvabilité 1 a fait l'objet de critiques quantitatives et qualitatives. Premièrement, il a été reproché à la directive de ne pas prendre en compte les risques liés aux placements. En effet, la marge de solvabilité se concentre uniquement sur la souscription des contrats d'assurance. De plus, la réassurance n'est pas prise en compte, et les différentes branches d'activités ne sont pas clairement distinguées. Étant donné que les méthodes de calcul des provisions varient d'un pays à l'autre au sein de l'Union Européenne, il devient impossible de comparer les résultats des compagnies entre elles. Enfin, la directive ne tient pas compte de la diversification des investissements, qui pourrait pourtant réduire les risques encourus par les assureurs.

1.1.2 Solvabilité 2

Solvabilité 2 fut mise en place pour répondre aux critiques faites à Solvabilité 1. La motivation principale était de mieux rendre compte de la diversité des risques, principe oublié dans Solvabilité 1. Pendant plus de 10 ans, la Commission Européenne s'est appuyée les rapports de *l'European Insurance and Occupational Pensions Authority* (EIOPA) et les avis d'experts afin de déterminer un ensemble de **principes** qui constitueront Solvabilité 2. De nombreux auteurs voient Solvabilité 2 comme la traduction de Bâle 2 des banques vers les assurances. Cet ensemble de principes ne constitue pas des règles et ils sont donc soumis à l'interprétation du régulateur et des assureurs de chaque pays. Les objectifs de Solvabilité 2 étant pluriels, nous pouvons énumérer les suivants :

1. Réglementer et harmoniser le marché de l'assurance et de la réassurance dans l'union européenne,
2. Renforcer la protection des preneurs d'assurance et quantifier leur niveau de protection,
3. Prendre en compte la situation financière des entreprises d'assurances dans le processus de gestion des risques,

4. Mise en place d'un socle pour pouvoir comparer les entreprises d'assurance entre elles au niveau de l'Europe.

La directive ne s'appliquera pas à toutes les entreprises d'assurance. Par exemple, les petites mutuelles et petits assureurs (ne dépassant pas une certaine taille) ne sont pas soumis à Solvabilité 2, en accord avec les articles 4 à 12 de la directive 2009/138/CE. Solvabilité 2 s'appuie sur 3 piliers que nous allons expliciter.

Premier pilier de Solvabilité 2 Le premier pilier porte sur les exigences qualitatives. Une des nouveautés notables de Solvabilité 2 est la valorisation du bilan prudentiel en valeur de marché. Sous Solvabilité 1, la valorisation était en valeur comptable, ce qui, nous le verrons par la suite, ne demande pas de mettre en oeuvre les mêmes moyens. Les articles 76 à 86 de la directive 2009/138/CE redéfinissent le calcul de la provision technique en distinguant les flux de trésorerie couvrables (flux de trésorerie pour lesquels une valeur de marché est disponible) des non-couvrables.

Dans le cas où la trésorerie est *couvrable*, nous pouvons répliquer à l'aide des instruments financiers tous les paiements possibles liés aux flux de trésorerie et ainsi éliminer tous les risques liés au passif. Dans ce cas de figure, la méthode de calcul des provisions techniques est :

$$\text{Provisions techniques} = \text{Best Estimate} + \text{Marge de risque},$$

avec

1. Le Best Estimate est la moyenne pondérée en fonction de leur probabilité des futurs flux de trésorerie compte tenu de la valeur temporelle de l'argent, laquelle est estimée sur la base de la courbe des taux sans risque pertinente (article 77),
2. La marge de risque est quasiment nulle, car la valeur actuelle des flux de trésorerie est connue grâce aux marchés, et nous avons une couverture parfaite.

Lorsque les flux de trésorerie ne sont pas couvrables, nous calculons les provisions toujours de la même façon, comme :

$$\text{Provisions techniques} = \text{Best Estimate} + \text{Marge de risque},$$

Solvabilité 2 introduit à partir de l'article 129 la notion du *Minimum Capital Requirement* (MCR). Il s'agit du plus petit capital que doit avoir un assureur. Le *Solvency Capital Requirement* (SCR) est lui aussi défini par solvabilité 2. Le SCR est le capital requis pour faire face à d'éventuels imprévus et pertes d'une entreprise à horizon de 1 an. Ce capital doit suffire dans 99.5% des cas, autrement dit, il doit limiter le risque de faillite à une fois tous les 200 ans.

En pratique, le calcul du SCR peut se faire soit par modèle interne, soit par la formule d'agrégation donnée par la directive et que nous allons détailler.

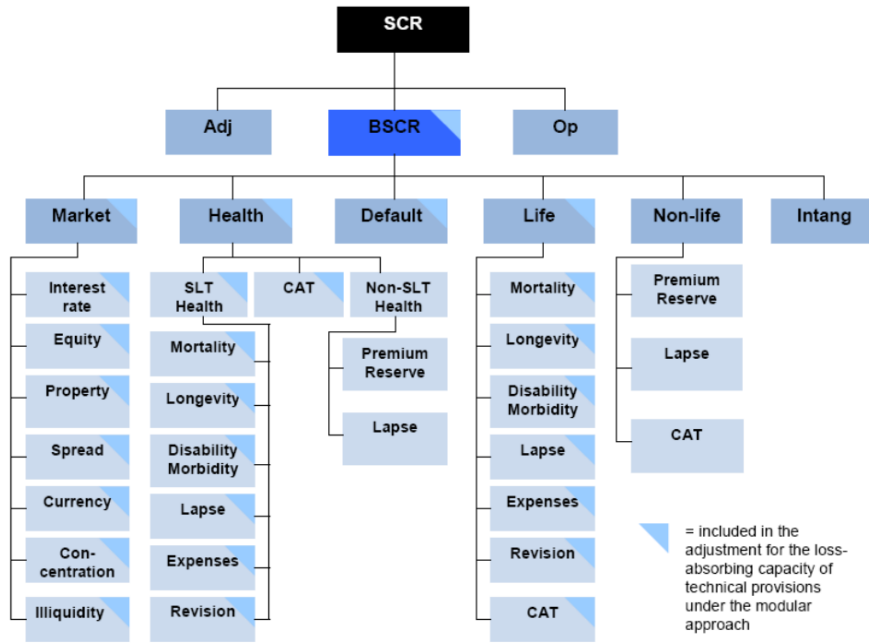


FIGURE 1.2 – La pieuvre de solvabilité 2

La figure 1.2 présente les différents modules de risque m pris en compte dans le calcul du SCR. Soit :

- R_m : l'ensemble des risques du module m ,
- C_i : le capital au titre du risque i ,
- $\rho_{i,j}^{R_m}$: le coefficient de corrélation permettant d'agrèger les capitaux des risques, i et j , du module m ,
- SCR_m : le capital économique du module m ,
- M : l'ensemble des modules,
- $\rho_{i,j}^M$: le coefficient de corrélation permettant d'agrèger les capitaux des modules i et j .

L'agrégation intra-modulaire s'obtient comme :

$$SCR_m = \sqrt{\sum_{(i,j) \in R_m^2} \rho_{i,j}^M \cdot C_i \cdot C_j},$$

nous trouvons ensuite le SCR de base en agréant les SCR des différents modules de risque :

$$SCR_{\text{de base}} = \sqrt{\sum_{(i,j) \in M^2} \rho_{i,j}^M \cdot SCR_i \cdot SCR_j},$$

Deux quantités vont être ajoutées au SCR de base.

1. Le $SCR_{\text{opérationnel}}$ qui quantifie le risque opérationnel. Ce risque découle de pertes potentielles dues à, des processus internes défailtants, des erreurs de membres du personnel, des événements extérieurs.
2. Un terme d'ajustement qui représente la compensation potentielle de pertes non anticipées par une diminution des provisions techniques ou des impôts différés. Cet ajustement est inhérent au caractère long terme de l'assurance. En effet, un assureur peut démontrer qu'il réalisera du bénéfice à l'avenir ou qu'il pourra réduire ses prestations futures pour couvrir des pertes non anticipées lors de leur survenance. L'ajustement va alors venir compenser ces pertes.

D'où :

$$SCR = \text{Ajustement} + SCR_{\text{de base}} + SCR_{\text{Opérationnel}}$$

Le deuxième pilier de Solvabilité 2 Le pilier 2 de Solvabilité 2 complète le pilier 1 en fournissant une évaluation approfondie de la solvabilité de l'entreprise à moyen et long terme. Il vise à s'assurer que les calculs effectués dans le pilier 1 sont réalisés de manière prudente, cohérente et en accord avec les principes et objectifs énoncés dans la directive. Tout d'abord, chaque entreprise d'assurance doit mettre en place un système de **contrôle interne** intégrant diverses fonctions :

- la fonction actuarielle (calcul des provisions techniques),
- la fonction de gestion des risques,
- la fonction d'audit interne (vérification de l'efficacité du contrôle interne),
- la fonction de vérification de la conformité (respect des orientations législatives, réglementaires, et administratives, engagées).

De même, un système de **contrôle externe** est mis en place, incarné par le régulateur, qui a le pouvoir d'exiger une marge de capitaux supplémentaire si celle-ci est jugée trop faible. Pour que toute cette mécanique puisse s'accorder correctement, l'article 45 de la directive demande le suivi d'un processus de contrôle des risques appelé *Own Risk and Solvency Assessment* (ORSA). Il doit notamment se pencher sur les points suivants :

- Le besoin global de solvabilité, compte tenu du profil de risque et des limites de tolérance au risque et de la stratégie commerciale de l'entreprise,
- Le respect permanent des exigences de capital ainsi que les exigences concernant les provisions techniques,
- L'analyse de l'adéquation au profil de risque avec le cadre induit par le pilier 1 de solvabilité II.

Pilier 3 : communication et transparence Il s'agit d'harmoniser la communication des assureurs à l'échelle européenne. La communication est à la fois à destination des assurés et des régulateurs. Par conséquent, des fichiers standardisés sur l'état de santé des assureurs, sur leur solvabilité et sur leur bilan vont devoir être remplis. Nous trouvons notamment :

- Les QRT : Quantitative Reporting Templates i.e., rapport sur les états quantitatifs, documents dont les principaux enjeux sont l'état des actifs et des provisions techniques à remettre à l'autorité de contrôle.
- Les RSR : Regular Supervisory Report, rapports réguliers au contrôleur (RSR),
- Les SFCR : Solvency and Financial Conditions Report, rapport réguliers sur la solvabilité et la situation financière. Les SFCR sont publics. Dans le rapport de préparation à Solvabilité 2 de 2015, *l'Autorité de contrôle prudentiel et de résolution* (ACPR) énonce que *"les rapports SFCR et RSR décrivent l'activité de l'organisme, son système de gouvernance, son profil de risque et complètent la remise des états quantitatifs annuels, en donnant notamment des informations sur les méthodes de valorisation utilisées ainsi que des précisions sur la gestion du capital"*.

Les 3 piliers de Solvabilité 2 sont décrits en figure 1.3.



FIGURE 1.3 – Les trois piliers de solvabilité 2

1.2 Monde risque neutre, monde réel

La directive Solvabilité 2 nécessite donc de pouvoir calculer *in fine* la probabilité de défaut de la compagnie à horizon de 1 an. Il faut donc être capable de projeter à 1 an les éléments dans le bilan d'une société, à savoir l'actif (actions, obligations, immobilier, OPCVM, ...) et le passif (provisions techniques, dettes financières, ...). L'obtention de ces grandeurs passe par la projection d'autres variables tels que les taux d'intérêts où l'inflation des prix.

Il est possible, selon l'objectif que nous souhaitons atteindre, de générer des scénarios sous les probabilités historiques ou risque-neutre.

La vision monde réel Elle permet de projeter des trajectoires compatibles avec les observations historiques. Les modèles sont calibrés sur des données historiques afin de représenter au mieux la réalité et les différents faits stylisés. Chaque actif offre un surplus de rendement appelé prime de risque. Dans la réalité, les actifs ont tous une prime de risque différente, ce qui rend les calculs plus complexes lorsqu'on souhaite évaluer un modèle. Le GSE monde réel est principalement utilisé pour calculer l'allocation optimale d'actifs financiers, en projetant le bilan sur un an dans le cadre de modèles internes ou pour gérer les risques financiers.

La vision risque neutre Elle est utilisée pour la valorisation. Dans cet univers, les agents économiques sont tous considérés comme neutres au risque. Cela se traduit par des primes de risque toutes nulles. L'existence d'une probabilité risque neutre, qui fournit un cadre pratique pour la valorisation de produits financiers, repose principalement sur deux hypothèses fondamentales :

- **L'absence d'opportunité d'arbitrage (AOA)** : impossibilité de réaliser un gain strictement positif avec une probabilité strictement positive pour un investissement nul,
- **La complétude des marchés financiers** : Il est possible de répliquer tous les flux financiers en utilisant un portefeuille auto-finançant qui est composé à la fois d'actifs sans risque et d'actifs risqués.

Sous l'hypothèse d'absence d'opportunités d'arbitrage, cela signifie qu'il n'est pas possible de réaliser des bénéfices sans risque en exploitant les déséquilibres de prix sur les marchés financiers. En d'autres termes, il n'y a pas de combinaison d'achats et de ventes d'actifs qui permettrait

à un investisseur de gagner de l'argent sans prendre de risque. L'hypothèse de complétude des marchés suppose que tous les actifs financiers possibles sont disponibles à la négociation et que les investisseurs peuvent acheter ou vendre n'importe quel actif. Cela signifie qu'il est possible de construire un portefeuille qui reproduit les flux de trésorerie futurs de n'importe quel actif. Sous ces deux hypothèses, il est démontré qu'il existe une unique mesure de probabilité risque neutre. Nous voulons obtenir une valorisation en adéquation avec les prix observés sur le marché, que nous appelons *Market-Consistent*. Cela se traduit par l'utilisation de prix ou de volatilités observés sur les marchés afin de réaliser le calibrage des modèles. Un GSE risque-neutre cherche donc à être cohérent avec le marché. Sous la probabilité risque neutre, les actifs des marchés financiers étant martingales, il est bien plus simple d'évaluer le prix de tels produits dans ce cadre. En effet, nous pouvons estimer la valeur d'un actif en pondérant ses rendements futurs possibles par leur probabilité respective, puis en actualisant ces rendements au taux d'intérêt sans risque. Cette approche simplifie le processus de valorisation car elle permet de se concentrer sur les rendements attendus plutôt que sur la variabilité des rendements.

Dans le présent mémoire, nous utiliserons un GSE risque neutre. En effet, celui-ci est largement utilisé ensuite pour valoriser le bilan économique des assureurs.

1.3 Générateur de scénarios économiques

Il existe une littérature très vaste sur les générateurs de scénarios économiques. [11] est utile pour avoir une intuition plus précise sur les choix possible de GSE et les critères utiles pour les valider. Lorsque nous construisons un générateur de scénarios économiques, il faut choisir sa structure, c'est à dire la manière dont vont interagir les facteurs de risque. Il existe principalement deux grandes catégories d'architecture pour les générateurs de scénarios économiques présentées dans la suite.

1.3.1 Utilité du GSE

Les GSE sont des outils indispensables pour répondre aux exigences réglementaires. En effet, la valorisation des options dans les contrats d'assurances nécessitent l'utilisation de scénarios économiques stochastiques. Historiquement, avec le développement de l'outil informatique, nous avons pu passer de la projection de scénarios *déterministes* à de très nombreux scénarios *stochastiques*, utilisant une approche probabiliste. Nous notons donc que la valorisation des options et garanties est effectuée par une approche de type Monte-Carlo, afin de capter les effets d'asymétrie.

De manière plus générale, les GSE servent à prendre en compte et à modéliser les investissements à long terme des sociétés d'assurance via une modélisation de l'évolution de l'économie. En général, nous motivons cela par la volonté de pouvoir garantir un surplus permanent².

1.3.2 Le Modèle intégré

Les modèles intégrés partent de la modélisation d'une ou de plusieurs variables de références pour pouvoir obtenir toutes les autres classes d'actifs. Souvent, nous utilisons l'inflation comme variable de référence dans les compagnies d'assurance, pour l'importance de cette dernière. L'avantage de ce genre de modèles est qu'ils sont faciles à implémenter et ont par construction une cohérence globale. Malheureusement certains facteurs de risques sont mal modélisés avec une telle méthode (tels que les taux nominaux).

2. écart entre valeur des actifs à une date donnée et la valeur des engagements actualisés à un taux d'intérêt de référence

1.3.3 Modèles composites

Dans un modèle composite, nous modélisons séparément chaque classe d'actifs. Elles ont leurs propres dynamiques, et sont traitées indépendamment les unes des autres. Nous utilisons ensuite une matrice de corrélation pour agréger les classes d'actifs. La modélisation et le calibrage sont ensuite effectués dans le respect de cette structure. Le défaut est la forte dépendance du modèle aux données sous-jacentes. Si les variables impliquées dans la modélisation ont un comportement futur différent de celui observé sur la période d'estimation historique, le GSE peut conduire à des projections non pertinentes. L'avis d'expert intervient alors pour corriger et fixer les niveaux de dépendance si nécessaire.

1.3.4 Construction d'un GSE sous solvabilité 2

Le mise en place d'un GSE suit un processus précis :

- **Identification des différents facteurs de risques** : taux d'intérêt, inflation, action, immobilier, crédit, devises,
- **Choix des modèles de diffusion** : Nous cherchons des modèles capables de répliquer les faits stylisés observés. Pour chaque facteur de risque nous avons un modèle décrivant son comportement. (nous présenterons plus en détail au chapitre 2 les modèles qui attirent notre attention),
- **Choix de la structure de dépendance** : Détermination de la matrice de corrélation, de la copule, entre les différents facteurs de risque,
- **Calibration** : Estimation des paramètres des modèles que nous avons proposé, en minimisant une distance entre les prix du modèles et les prix observés sur les marchés,
- **Validation des scénarios générés** : Nous faisons passer des tests de marginalités (dans le monde risque-neutre). Nous voulons nous assurer que les prix projetés actualisés au taux sans risque sont bien des martingales. Nous parlons de *tests de market consistency*.

Une synthèse des facteurs de risques et de leur données associées est présenté ci-dessous :

Facteur de risque	Données
Taux nominaux	La courbe des taux zéro-coupon (ZC) nominaux réglementaire Les volatilités implicites de swaptions ATM et AFM
Taux réels	La courbe de taux d'inflation break-even (Break-Even Inflation Rates ou BEIR)
Action	Les volatilités implicites de calls ATM Hypothèse de taux de dividende
Immobilier	Hypothèse de volatilité et de taux de loyer
Spread	Structures par terme des spreads logarithmiques pour les différents groupes de ratings souhaités (govies, corporates)

TABLE 1.1 – Résumé des données par facteur de risque

Les différents modèles utilisés pour modéliser les différentes grandeurs économiques sont aussi données ci-dessous :

Taux nominaux LIBOR Market Model (LMM) LIBOR Market Model à coefficient de déplacement (DDLMM) LIBOR Market Model à coefficient de déplacement et volatilité stochastique (DDSVLMM) Modèle Gaussien à deux facteurs (G2++)
Taux réels Modèle Vasicek à deux facteurs Gaussian LIBOR Market Model
Actions, immobilier, taux de change & autres indices Black-Scholes à volatilité constante Black-Scholes à volatilité dépendant du temps (paramétrique ou non-paramétrique) Modèle de Heston (volatilité stochastique) Modèle Stochastic Volatility Jump Diffusion (SVJD)
Dividendes Modèle de fonction réponse aux taux Modèle Cox-Ingersoll-Ross
Spread Modèle à probabilités de transition de rating de type Jarrow-Lando-Turnbull stochastique Modèle à intensité de défaut de type Longstaff-Mithal-Neis

TABLE 1.2 – Résumé des modèles

Une discussion générale des avantages/inconvénients est présentée ci-dessous.

<p>Taux nominaux</p> <p>Le modèle G2++ est un modèle de taux court relativement simple qui bénéficie d'un bon niveau d'interprétabilité ; à ce titre il peut permettre un meilleur contrôle de la distribution des taux. Le modèle DDLMM présente l'intérêt de prendre en compte les taux négatifs et de fournir des résultats de calibrage relativement stables et satisfaisants. Le modèle DDSVLMM dispose de davantage de paramètres et permet ainsi d'améliorer la réplication de la surface de volatilités de swaptions ; toutefois, la stabilité du calibrage est à contrôler compte tenu du nombre de paramètres en jeu.</p>
<p>Taux réels</p> <p>Le modèle Vasicek à deux facteurs permet de modéliser des structures de corrélations réalistes entre taux réels de différentes maturités, mais ne permet pas de reproduire la courbe initiale. A ce titre, le modèle GLMM constitue une extension intéressante car il repose sur la même structure mais permet une réplication parfaite de la courbe. Pour ce dernier modèle, il est clé de pouvoir estimer certains paramètres dits "distributionnels" a priori afin de bénéficier d'une stabilité de calibrage accrue.</p>
<p>Actions, immobilier, taux de change & autres indices</p> <p>Le modèle de type Back-Scholes à volatilité constante bénéficie d'une simplicité et d'une stabilité, cependant l'existence d'une structure par termes pour les surfaces de volatilités implicites conduit à l'utilisation de modèle intégrant une volatilité déterministe dépendant du temps. Le recours alternatif à un modèle de type Heston ou SVJD peut être envisagé en cas d'objectif de réplication du skew / smile.</p>
<p>Dividendes</p> <p>La modélisation des dividendes peut reposer sur une fonction lien permettant d'ajuster le niveau des dividendes au niveau des taux. En revanche, la corrélation avec les autres facteurs de risque reste subie dans ce modèle ; lorsque nous souhaitons modéliser plus finement les trajectoires stochastiques de taux de dividendes, un recours au modèle de Cox-Ingersoll-Ross peut être envisagé.</p>
<p>Spread</p> <p>Le modèle d'intensité de défaut (LMN) dispose d'un nombre de paramètres important (4 par classe de rating), permettant d'adresser la market consistency de manière satisfaisante. Cependant, ce type de modèle ne permet pas de modéliser les probabilités de transition de rating. A ce titre, un recours à des modèles de transition (JLT) peut être envisagé.</p>

TABLE 1.3 – Résumé des modèles utilisés dans notre GSE

1.4 Asset and Liability Management (ALM)

1.4.1 Généralités sur les modèles ALM

Le modèle ALM vise à modéliser les flux de l'actif et du passif en prenant en compte leurs interactions, à la lumière de la stratégie d'investissement, de revalorisation et du comportement des assurés. Cette problématique est particulièrement challengante pour les assureurs en raison de la complexité liée à l'analyse des interactions entre l'actif et le passif, à la nécessité de prendre en compte la stratégie d'investissement et ses conséquences sur les flux financiers, ainsi qu'à la difficulté de revaloriser les engagements futurs envers les assurés en raison des incertitudes inhérentes aux hypothèses utilisées.

Dans le cadre des calculs ALM, nous calculons le *Best Estimate* (BE) :

$$\text{Best Estimate} = \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^T \text{Flux}_{i,j} \times \delta_{i,j}$$

avec :

- N le nombre de scénarios économiques,
- T l'horizon de projection,
- Flux $x_{i,j}$ les flux de trésorerie de l'année j pour le scénario i ,
- $\delta_{i,j}$ le déflateur relatif à l'année j et au scénario i .

Les calculs ALM évaluent d'abord les flux de trésorerie pour un scénario et pour une année de projection donnée. Cette évaluation s'effectue en plusieurs étapes :

- **Étape 1** : Vieillessement du portefeuille financier,
- **Étape 2** : Gestion du portefeuille épargne,
- **Étape 3** : Calcul de l'assiette de trésorerie et ré-allocation des actifs,
- **Étape 4** : Revalorisation des contrats,
- **Étape 5** : Construction du bilan comptable et du compte de résultat en normes statutaires.

Un bilan comptable dans un moteur ALM a la forme suivante :

Actif	Passif
Actions	Fonds Propres, Réserve de Capitalisation
Obligations	Provision pour Risque d'exigibilité, Provision pour Participation aux Excédents
Immobilier	Provision Mathématique
Monétaire	PM

1.4.2 Quelques éléments sur le modèle ALM que nous avons utilisé

Le modèle que nous utilisons, "France Vie" est une entité fictive regroupant les principales caractéristiques des compagnies d'assurance vie exerçant sur le marché français. De ce fait, "France Vie" ne représente pas la moyenne du marché :

- Le modèle retenu est exclusivement le modèle standard (pas de prise en compte d'effets modèle interne, pas de prise en compte de mesures transitoires,...),
- La plateforme de valorisation actuarielle utilisée est "standard", elle ne prend pas en compte par nature l'ensemble des spécificités des diverses entreprises d'assurance du marché,
- Les données de "France Vie" sont pour la plupart issues de données de marché mais également basées sur la connaissance d'expert.

Ce modèle cherche à donner une idée de marché français de l'assurance. Il ne reflète pas un assureur en particulier, et les résultats que nous présenterons pourraient varier si nous passions d'un acteur à un autre.

Lorsque nous ne le mentionnerons pas, nous utiliserons les données France Vie au 31/12/2021, comme notre scénario central.

1.5 Motivation du mémoire

Suivre le processus de calcul du BE nécessite de faire un certain nombre d'hypothèses au fur et à mesure. Il est légitime d'identifier ces hypothèses et de tester leur validité.

Les paramètres embarqués, qui ont des valeurs fixées et qui ne sont pas changés d'un calibrage à l'autre, comme par exemple le choix de la graine d'un générateur de nombre pseudo aléatoires ou le choix de la valeur du shift dans le modèle de taux nominaux, peuvent être considérés comme des hypothèses.

Le présent mémoire vise à apporter d'avantage d'information sur ces hypothèses et leur domaine de validité, notamment par une analyse de sensibilité avancée.

1.6 Définitions élémentaires en finance

Afin d'introduire les différents modèles que l'on utilise par la suite, nous rappelons quelques définitions fondamentales en finance qui resteront valides dans toute la suite de ce mémoire. On se place dans un espace probabilisé $(\Omega, \mathcal{F}, (\mathcal{F}_t)_{t \geq 0}, \mathbb{P})$. Nous faisons le choix, comme dans le mémoire de [48], d'adopter le formalisme de [6].

Definition 1.6.1. (*Obligation zéro-coupon*) Une obligation zéro-coupon de maturité T est un titre financier qui verse un unique flux d'une unité monétaire à la date T . On note $P(t, T)$ sa valeur à la date t , avec $0 \leq t \leq T$.

Definition 1.6.2. (*Taux d'intérêt composé continûment*) Le taux d'intérêt composé continûment prévalant à la date t pour une maturité T , noté $R(t, T)$, est le taux auquel un investissement de $P(t, T)$ unité monétaire se capitalise continûment jusqu'à rapporter une unité de monnaie à l'échéance T , ce qui se traduit mathématiquement par les équivalences, pour $0 \leq t < T$:

$$e^{R(t, T)\tau(t, T)} P(t, T) = 1 \Leftrightarrow P(t, T) = e^{-R(t, T)\tau(t, T)} \Leftrightarrow R(t, T) = -\frac{\ln P(t, T)}{\tau(t, T)}$$

Definition 1.6.3. (*Taux d'intérêt composé simplement*) Le taux d'intérêt composé simplement prévalant à la date t , pour une maturité T , noté $L(t, T)$, est le taux auquel un investissement de $P(t, T)$ unité monétaire croît proportionnellement au temps d'investissement pour rapporter une unité de monnaie à l'échéance T , ce qui se traduit mathématiquement par les équivalences, pour $0 \leq t < T$:

$$(1 + L(t, T)\tau(t, T))P(t, T) = 1 \Leftrightarrow L(t, T) = \frac{1 - P(t, T)}{\tau(t, T)P(t, T)}$$

Definition 1.6.4. (*Taux d'intérêt composé annuellement*) Le taux d'intérêt composé annuellement prévalant à la date t , pour une maturité T , noté $Y(t, T)$, est le taux auquel un investissement de $P(t, T)$ unité monétaire croît en étant réinvesti une fois par an pour rapporter une unité de monnaie à l'échéance T , ce qui se traduit mathématiquement par les équivalences, pour $0 \leq t < T$:

$$(1 + Y(t, T))^{\tau(t, T)} P(t, T) = 1 \Leftrightarrow Y(t, T) = \frac{1}{P(t, T)^{1/\tau(t, T)}} - 1$$

Definition 1.6.5. (*Contrat FRA*) Un contrat FRA, évalué à l'instant t , de nominal N , d'expiration T et de maturité S , noté $FRA(t, T, S, \tau(T, S), N, K)$ est un contrat payant à son détenteur un taux d'intérêt fixe K à la date S contre le paiement à cette même date d'un taux d'intérêt

instantané variable $L(T, S)$ gelé en T pour la période $[T, S]$. Le pay-off versé au détenteur à la date S vaut :

$$N\tau(T, S)(K - L(T, S))$$

En injectant l'expression de $L(T, S)$ et en remarquant qu'un montant de $1/P(T, S)$ unité en S est équivalent à un montant de $P(t, T)$ unité en t , la valeur d'un contrat forward évalué en t est alors :

$$FRA(t, T, S, \tau(T, S), N, K) = N[P(t, S)\tau(T, S)K - P(t, T) + P(t, S)]$$

Definition 1.6.6. (Taux forward simplement composé) Le taux forward simplement composé, prévalant à la date t , expirant en $T > t$ et de maturité $S > T$, noté $F(t, T, S)$, est défini par :

$$F(t, T, S) = \frac{1}{\tau(T, S)} \left(\frac{P(t, T)}{P(t, S)} - 1 \right)$$

Definition 1.6.7. (Taux forward instantané) Le taux forward instantané, noté $f(t, T)$, est défini comme la limite du taux forward quand la maturité S tend vers son expiration T . Formellement :

$$f(t, T) = \lim_{S \rightarrow T^+} F(t, T, S) = -\frac{\partial \ln P(t, T)}{\partial T}$$

Definition 1.6.8. (Contrat swap forward) Soit un échéancier constitué des dates $T_i, i \in \{0, \dots, M\}$, avec $T_0 = t \leq T_1 \leq \dots \leq T_M$. Soient α et β tels que $0 \leq \alpha < \beta \leq M$. On note $\mathcal{T} = \{T_\alpha, \dots, T_\beta\}$ et $\tau = \{\tau_{\alpha+1}, \dots, \tau_\beta\}$ avec $\tau_i = T_i - T_{i-1}, i \in \alpha + 1, \beta$. Le contrat swap forward receveur, de nominal N , de maturité T_β et de tenor $T_\beta - T_\alpha$, évalué en t , noté $RFS(t, \mathcal{T}, \tau, N, K)$, est un contrat payant à son détenteur un flux fixe $\tau_i K$, contre un flux variable $\tau_i L(T_{i-1}, T_i)$, à chaque date $T_{\alpha+1}, \dots, T_\beta$. L'ensemble des flux fixes est appelé la jambe fixe du swap. Les flux variables en constituent la jambe variable. La valeur actualisée du contrat en t est :

$$\begin{aligned} RFS(t, \mathcal{T}, \tau, N, K) &= N \sum_{i=\alpha+1}^{\beta} P(t, T_i) \tau_i (K - F(t, T_{i-1}, T_i)) \\ &= N (P(t, T_\beta) - P(t, T_\alpha)) + N \sum_{i=\alpha+1}^{\beta} P(t, T_i) \tau_i K \end{aligned}$$

De manière analogue, un contrat swap forward payeur paie la jambe fixe et reçoit la jambe variable. Sa valeur actualisée est alors donnée par :

$$PFS(t, \mathcal{T}, \tau, N, K) = N \sum_{i=\alpha+1}^{\beta} P(t, T_i) \tau_i (F(t, T_{i-1}, T_i) - K)$$

Definition 1.6.9. (Taux swap forward) Le taux swap forward $S_{\alpha, \beta}(t)$ à la date t , d'échéancier \mathcal{T} , de fréquence τ est le taux fixe qui annule la valeur du contrat swap, i.e. $RFS(t, \mathcal{T}, \tau, N, K) = 0$. On obtient :

$$S_{\alpha, \beta}(t) = \frac{P(t, T_\alpha) - P(t, T_\beta)}{\sum_{i=\alpha+1}^{\beta} \tau_i P(t, T_i)}$$

Definition 1.6.10. (Taux swap) Le taux swap prévalant à la date t , noté $S_\beta(t)$, est le taux swap forward défini pour $T_\alpha = t$. Formellement :

$$S_\beta(t) = \frac{1 - P(t, T_\beta)}{\sum_{i=\alpha+1}^{\beta} \tau_i P(t, T_i)}$$

Definition 1.6.11. (*Swaption*) On se dote d'un ensemble de dates $T_0 \leq T_1 \dots \leq T_M$, avec $T_0 = 0$. Considérons deux entiers α et β tels que $0 \leq \alpha < \beta \leq M$, ainsi que l'échéancier de dates $\mathcal{T} = \{T_\alpha, \dots, T_\beta\}$. Une swaption ou option sur swap est un contrat donnant le droit à son détenteur d'entrer dans un swap de prix d'exercice, dit strike K , à la date de maturité T_α pour une durée $T_\beta - T_\alpha$ appelée tenor. Dans le cas où le détenteur de la swaption paie (resp. reçoit) le taux fixe K , on parle de swaption payeuse (resp. receveuse). Le pay-off de la swaption payeuse à la date T_α s'écrit :

$$\left(\sum_{i=\alpha+1}^{\beta} P(T_\alpha, T_i) \tau_i (F(T_\alpha, T_{i-1}, T_i) - K) \right)^+$$

Une swaption à la monnaie (ATM) a pour strike le taux swap forward :

$$K_{ATM} = S_{\alpha, \beta}(0) = \frac{P(0, T_\alpha) - P(0, T_\beta)}{\sum_{i=\alpha+1}^{\beta} \tau_i P(0, T_i)}$$

Par ailleurs, $K < K_{ATM}$ caractérise une swaption en dehors de la monnaie (OTM). Au contraire $K > K_{ATM}$ désigne une swaption dans la monnaie (ITM). Plus généralement, une swaption OTM ou ITM est dite away-from-the-money (AFM).

Sur les marchés des taux d'intérêt, les swaptions sont des produits dérivés très liquides. Les praticiens s'en servent pour le calibrage des modèles de taux.

Conclusion Dans ce mémoire, notre objectif est d'étudier les paramètres embarqués dans les modèles financiers utilisés dans un GSE. Certains de ces paramètres ne sont jamais recalibrés, ce qui soulève la légitime question de savoir quand il est nécessaire de mettre à jour leur valeur. Au lieu d'un calcul annuel ou tous les 2 ou 3 mois, ce recalcul doit être effectué dès que les conditions du marché changent. Notre étude se concentre donc sur l'analyse des différents paramètres dans des conditions de marché variables, ce qui conduit à des distributions multimodales. En effet, la valeur du BE et de la VIF variera considérablement en fonction du contexte des taux d'intérêt et de la volatilité des actions. Notre objectif est de développer des approches pour gérer ces situations complexes. Enfin, nous tenterons de proposer une hiérarchie des paramètres embarqués les plus importants, en relation avec le modèle ALM dont nous disposons.

Chapitre 2

Un exemple introductif, les taux nominaux

Le Displaced Diffusion Libor Market Model (DDLMM) est un modèle de marché de taux d'intérêt nominaux. Il modélise les taux forward observables sur le marché, contrairement aux taux forward instantanés utilisés dans le modèle de Heath Jarrow Morton, ou aux taux courts utilisés dans le modèle de Hull et White. De plus, sa volatilité est déterministe et sa distribution est déplacée par un *shift* afin d'obtenir des taux potentiellement négatifs. Le *shift* est un paramètre embarqué dont la valeur est fixée en amont de tout calibrage. Il devient alors légitime de chercher à quantifier l'impact d'une variation du shift sur les résultats ALM tels que la VIF ou le BE. Par la suite, le shift sera un bon exemple pour faire sentir au lecteur les différents enjeux liés à l'analyse de sensibilité des paramètres embarqués.

2.1 Éléments théoriques du modèle

Nous considérons, pour tout $k \in \mathbb{N}^*$ et tout $t \in [0, k]$, $\tilde{F}_{k(t)}$ le taux forward déplacé (pour tout facteur de shift $\delta > 0$). Cette variable est construite à partir de la définition de $F_{k(t)}$, le taux forward, de date d'expiration T_k et de maturité T_{k+1} :

$$\tilde{F}_k(t) = F_k(t) + \delta,$$

où $F_k(t) := \frac{1}{\Delta} \left(\frac{P(t, T_k)}{P(t, T_{k+1})} - 1 \right)$ avec Δ la période (supposée constante) entre la date d'expiration T_k et la maturité T_{k+1} .

Sous la mesure $k+1$ -forward neutre ($k \in \mathbb{N}^*$) associée au numéraire $P(t, T_{k+1})$, la dynamique du taux forward déplacé $\left(\tilde{F}_{k(t)} \right)_{t \in [0, k]}$ est la suivante :

$$d\tilde{F}_k(t) = \tilde{F}_k(t) \times \sum_{q=1}^{N_f} \xi_k^q(t) dZ_{k+1}^q(t),$$

où $(Z_{k+1}^q)_{q \in 1, N_f}$ est un mouvement Brownien multidimensionnel sous la mesure $k+1$ -forward neutre. $N_f \geq 1$ est un entier égal au nombre de facteurs dans le modèle. ξ_k^q (fonction déterministe définie sur $[0, k]$) est la volatilité du taux forward déplacé \tilde{F}_k associée au $q^{\text{ème}}$ mouvement Brownien Z_{k+1}^q .

Une première étape lors de la définition de la structure de volatilité $\xi_k^q(t)$ consiste à fournir la forme de la structure de corrélation inter-forward. En effet, dans la réalité, les taux forward sont corrélés les uns avec les autres. La corrélation entre les forwards $i^{\text{ème}}$ et $j^{\text{ème}}$ (avec $i \leq j \in \mathbb{N}^*$) est notée $\rho_{i,j}$ et est supposée être égale, pour tout $t \in [0; T_i]$, à

$$\rho_{i,j}(t) = \sum_{q=1}^{N_f} \beta_i^q(t) \beta_j^q(t),$$

où $(\beta_i^q(t))_{i \in \mathbb{N}^*, q \in [1, N_f]}$ sont des fonctions dépendant du temps appelées facteurs d'exposition, supposées vérifier la condition $\sum_{q=1}^{N_f} (\beta_i^q(t))^2 = 1$ (de sorte que $\rho_{i,i} = 1$). Afin de réduire les degrés de liberté, ces fonctions sont supposées être constantes par morceaux, en fonction du temps restant à maturité. Pour tout $k \leq i \in \mathbb{N}^*$, pour tout $q \in 1, N_f$ et pour tout $t \in]T_{k-1}, T_k]$,

$$\beta_i^q(t) = b_{i-(k-1)}^q,$$

la structure de volatilité $\sigma_k(t)$ est décomposée en deux fonctions paramétriques. Pour tout $k \in \mathbb{N}^*$ et pour tout $t \in]T_{k-1}, T_k]$,

$$\sigma_k(t) = \Phi(t) \times g(T_k - t),$$

avec Φ un facteur d'échelle dépendant du temps, pour l'ensemble de la structure de volatilité, défini par, pour tout $t \in \mathbb{R}^+$,

$$\Phi(t) = \theta + (1 - \theta) \times e^{-\kappa \times t},$$

où $(\theta, \kappa) \in (\mathbb{R}^+)^2$, θ est le niveau d'échelle asymptotique et κ est la vitesse de convergence vers θ . g dépend du temps restant à maturité, de sorte que, pour tout $u \in \mathbb{R}^+$,

$$g(u) = (bu + a) \times e^{-cu} + d,$$

avec $(a, b, c, d) \in (\mathbb{R}^+)^4$.

2.2 Calibrage du modèle

À chaque date $t \in [0, T_n]$, le taux Swap (associé aux swaptions, et noté $S_{n,N}(t)$) est défini par

$$S_{n,N}(t) := \frac{P(t, T_n) - P(t, T_N)}{A_{n,N}(t)},$$

où l'annuité $A_{n,N}(t)$ s'écrit

$$A_{n,N}(t) = \sum_{i=n}^{N-1} P(t, T_{i+1}).$$

Le modèle DDLMM donne la dynamique des taux forward. Afin d'exprimer la dynamique du taux du swap, il est nécessaire de l'écrire en fonction des taux forward. Pour tout $t \in [0, T_n]$,

$$S_{n,N}(t) = \frac{\sum_{j=n}^{N-1} P(t, T_{j+1}) F_j(t)}{A_{n,N}(t)}.$$

Nous pouvons ensuite, après avoir explicité la dynamique du taux swap¹, calculer le prix théorique $PS_{n,N}^{Th}$ d'une swaption de maturité T_n , tenor T_{N-n} et strike K .

Il est maintenant possible d'estimer les paramètres du modèle $\hat{\Theta} = (\hat{a}, \hat{b}, \hat{c}, \hat{d}, \hat{\theta}, \hat{\kappa}) \in (\mathbb{R}^+)^6$ qui résolvent le problème de minimisation suivant :

$$\operatorname{argmin}_{\Theta \in (\mathbb{R}^+)^6} \left(\sum_{n=1}^{N_1} \sum_{N=n+1}^{n+N_2} \omega_{n,N} \left(\frac{PS_{n,N}^{Mkt}(K_{n,N}, S_{n,N}, \sigma_{n,N}^{Mkt}) - PS_{n,N}^{Th}(K_{n,N}, \Theta)}{PS_{n,N}^{Mkt}(K_{n,N}, S_{n,N}, \sigma_{n,N}^{Mkt})} \right)^2 \right),$$

avec N_1 l'indice de maturité maximale des swaptions considérées, N_2 le tenor maximal, et $(\omega_{n,N})$ le poids de chaque swaption dans le processus de calibration. Le paramètre de shift, κ est un paramètre *embarqué*, il est fixé lors des calibrages.

1. nous ne détaillons pas ici la dynamique du taux swap.

2.3 Statistiques descriptives du Shift

Afin de fixer les idées, nous allons faire varier les valeurs du shift en gardant le paramétrage de notre modèle ALM au 31/12/2021. Pour réaliser les statistiques descriptives, nous nous concentrons sur le zéro coupon de maturité 1Y, à différentes dates de projections (1 an, 10 ans, 30 ans et 50 ans). Nous avons 3000 simulations et donc plus de 3000 trajectoires stochastiques. Cela nous permet de calculer différentes métriques telles que le taux minimum atteint, le pourcentage de taux négatifs, la moyenne des taux et différents quantiles. La figure 2.1 regroupe ainsi plusieurs statistiques descriptives.

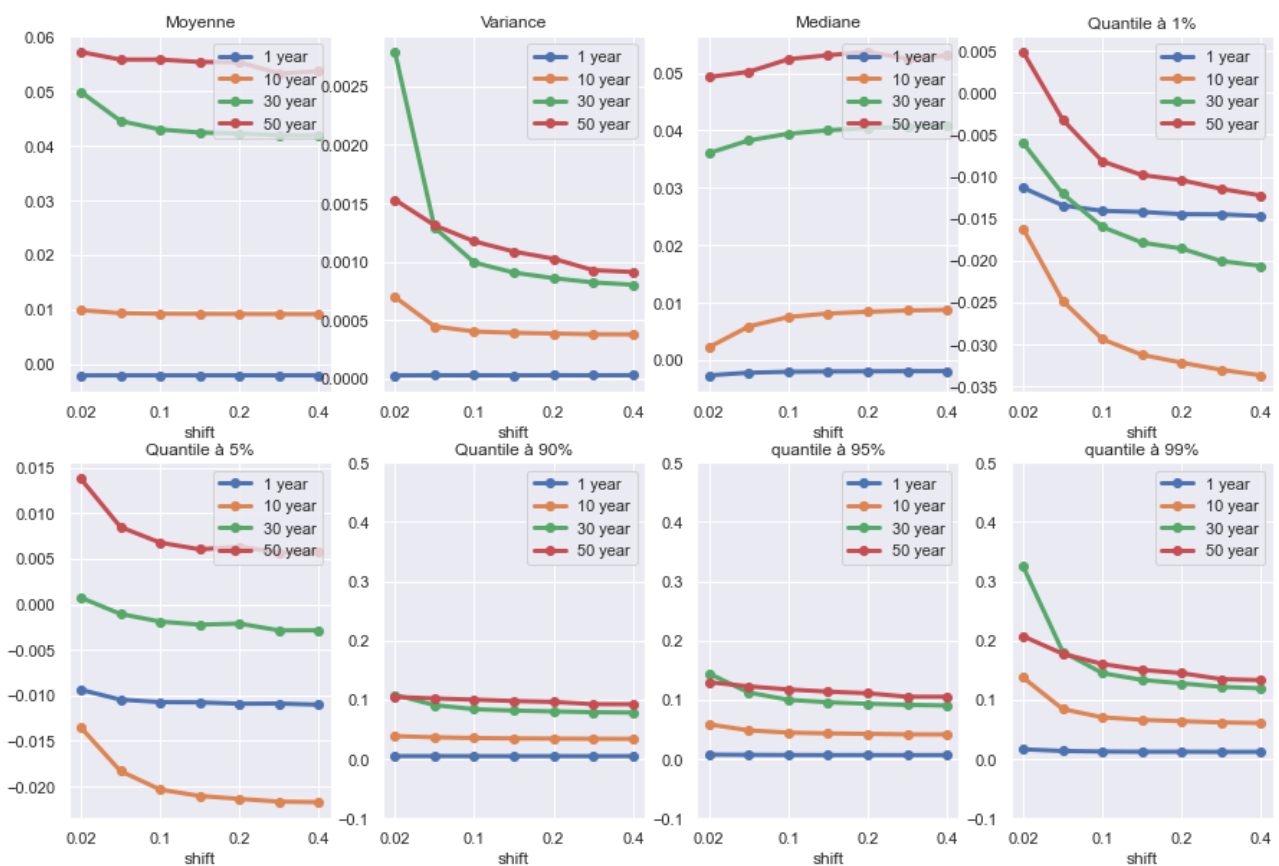


FIGURE 2.1 – Statistiques descriptives, rôle du shift au 31/12/2021.

Nous observons que les shifts faibles ont un impact plus important que les shifts élevés. L'utilisation d'un shift faible conduit à des taux explosifs. Nous remarquons que l'impact du shift s'affaiblit pour devenir rapidement nul lorsque l'on augmente beaucoup la valeur de ce dernier. La figure 2.2 vient appuyer ce constat en montrant l'évolution du pourcentage de taux négatifs en fonction de la valeur du shift.

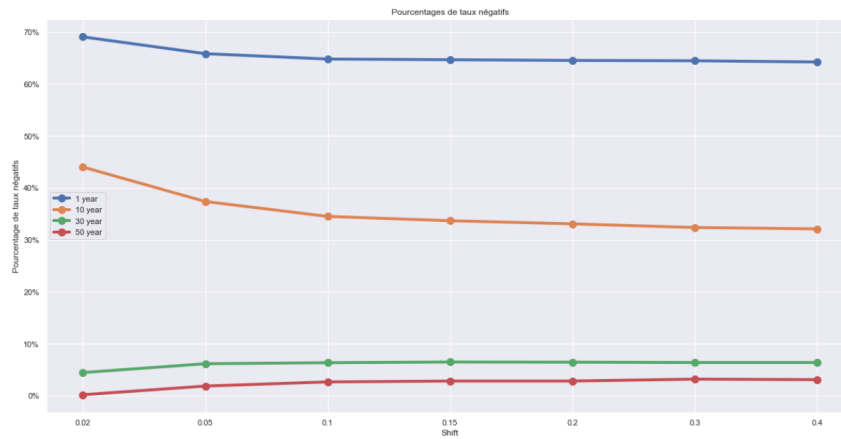


FIGURE 2.2 – Pourcentage de taux négatifs au 31/12/2021 en fonction de la valeur du shift.

Nous observons sur la figure 2.2 que les taux sont d’autant plus négatifs en moyenne que le shift est élevé. Nous constatons des valeurs plus explosives pour des shifts faibles. La figure 2.3 qui montre l’évolution des taux minimums atteints en fonction du shift vient elle aussi renforcer le constat que nous avons dressé.

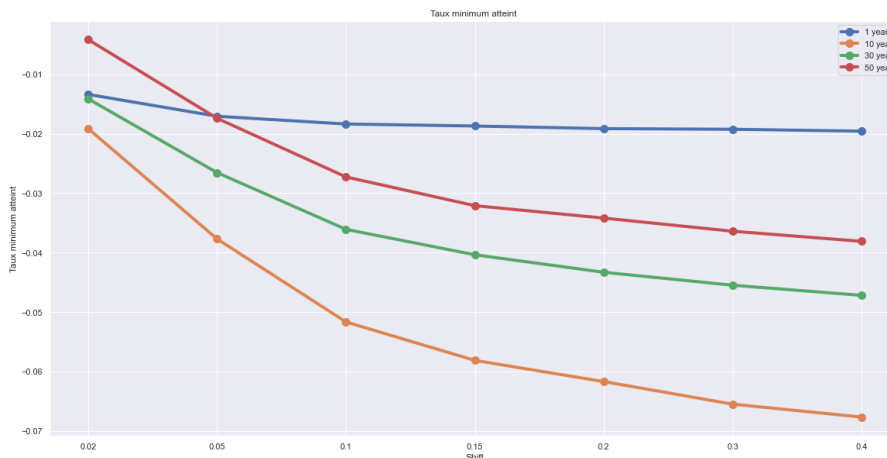


FIGURE 2.3 – Taux minimums atteints en fonction du shift.

Ces différentes statistiques descriptives nous montrent que l’impact du shift est maximum lorsque ce dernier est faible. Lorsque le shift devient plus important, son impact semble s’estomper.

Nous avons mené l’expérimentation avec des données de marché fixées, au 31/12/2021. Nous pourrions légitimement se demander si les conclusions resteraient les mêmes dans un contexte de marché différent ? En effet, là date à laquelle nous menons les analyses correspond à une date où les taux étaient bas. Bien sur, nous pourrions imaginer que le shift se comporte de manière totalement différente lorsque les taux seraient élevés.

La figure 2.4 représente les valeurs atteintes par la VIF pour différentes dates et différentes valeurs de shift.

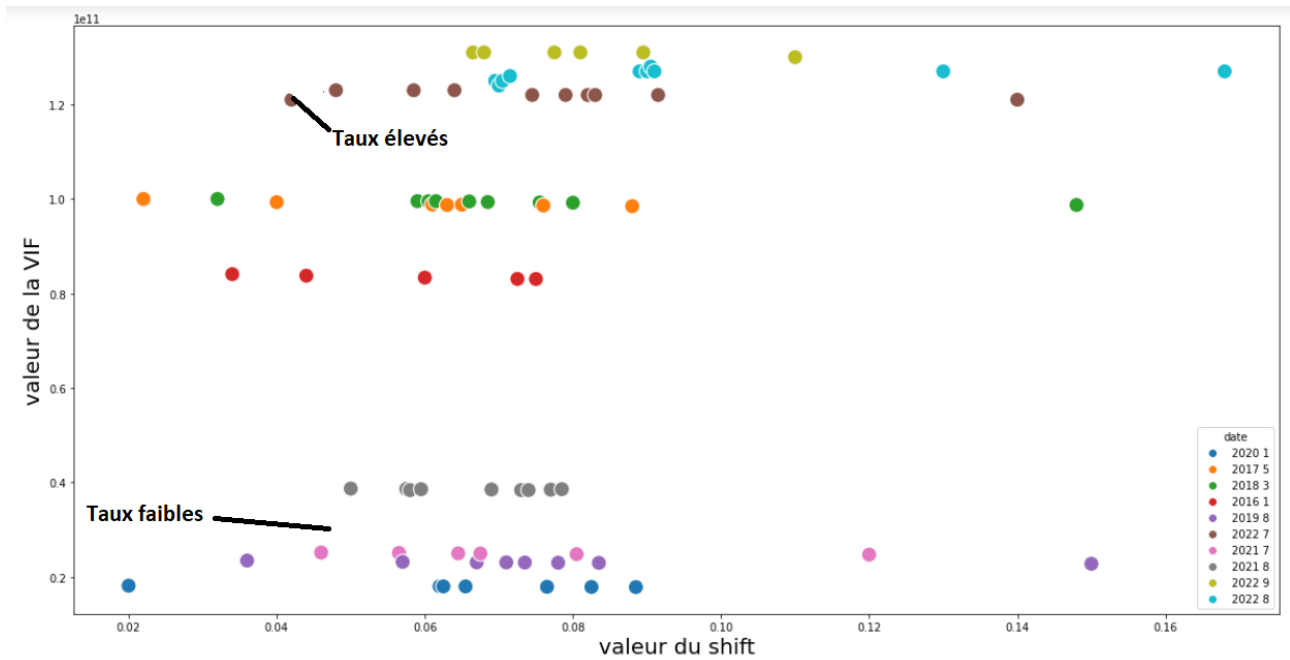


FIGURE 2.4 – Valeur de la VIF lorsque l’on change le shift et les données de marché. Les autres paramètres sont fixée et égaux à ceux du 31/12/2021.

Nous observons que le comportement n’est pas tout à fait le même en fonction du contexte de taux. En effet, lorsque les taux sont élevés, il semblerait que le shift ait un impact important. Nous observons aussi que la distribution est multimodale.

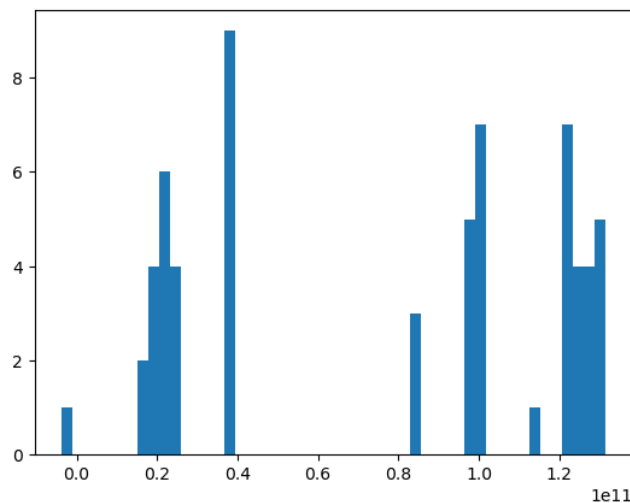


FIGURE 2.5 – Histogramme de la VIF. Nous nous rendons compte que la distribution est multimodale. Lorsque nous modifions les données de marché, nous modifions la moyenne de la distribution.

D’une part nous avons donc un comportement qui semble changer en fonction du contexte de taux et d’autre part nous semblons avoir une distribution multimodale.

Afin de fixer les idées dans la suite du mémoire, nous allons considérer une distribution multimodale simple. Celle ci a été choisie de manière arbitraire, et le point qui nous intéresse est son caractère multimodale. Considérons $X \sim N(0, 1)$, $Y \sim N(10, 1)$ et $Z \sim B(0.5)$. Les trois variables aléatoires étant indépendantes. Nous pouvons définir la fonction :

$$f : (x, y, z) \longrightarrow (2x^2 + 3y)(zx + (1 - z)y), \quad (2.1)$$

lorsque nous observons quelques réalisations de $f(X, Y, Z) \in L^2(\Omega)$, nous nous rendons très rapidement compte que la distribution est multimodale, voir figure ??.

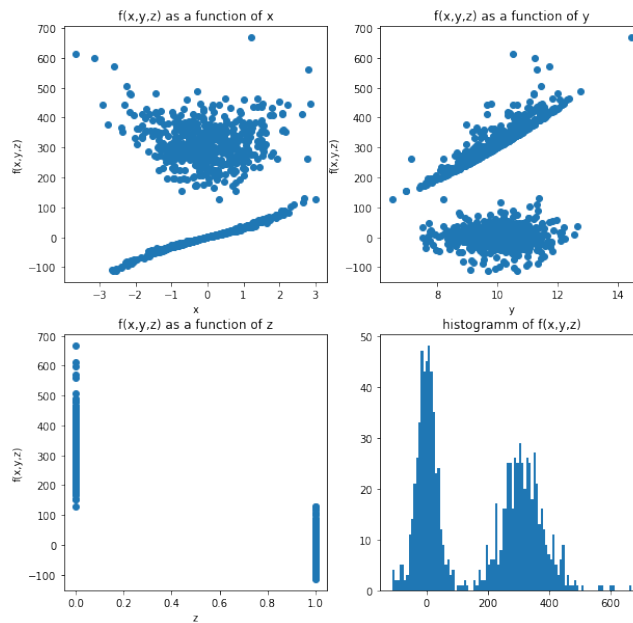


FIGURE 2.6 – Exemple fil rouge.

Nous allons utiliser l'exemple 2.6 pour éclairer le lecteur sur les propriétés des différents indices de sensibilité, et motiver nos choix.

Conclusion Nous en tirons les premières observations cruciales pour le mémoire : *comment quantifier l'impact des différentes données de marché, et comment gérer les distributions multimodales ?*

De nombreuses techniques très populaires pour faire de l'analyse de sensibilité ne sont pas efficace lorsque nous faisons face à des distributions multimodales. De plus, il va nous falloir mettre en place un cadre permettant de prendre en considération les différences de comportement selon le contexte de taux.

Nous avons besoin, afin d'analyser et de quantifier les impacts du shift et plus largement des différents paramètres embarqués, d'introduire différents outils dits d'analyse de sensibilité. La partie suivante du mémoire est dédiée à l'introduction de ces outils et sera pour nous l'occasion de présenter un modèle que nous avons développé pour accélérer l'analyse de sensibilité et la rendre un peu plus interprétable.

Deuxième partie

Analyse de facteurs : accélération de la
procédure par méthodes d'apprentissages

Chapitre 3

Etat de l'art en analyse de sensibilité

L'analyse de sensibilité est une technique utile pour comprendre la contribution des incertitudes des entrées sur l'incertitude des sorties d'un système [44]. Il existe deux types d'analyse de sensibilité : l'analyse locale et globale. L'analyse de sensibilité locale consiste à examiner l'impact des variations des entrées à proximité d'un point spécifique, tandis que l'analyse de sensibilité globale prend en compte les variations dans l'ensemble de l'espace des entrées réalisables [28, 43]. Deux grandes catégories d'indices de sensibilité ont été créées : les indices basés sur la variance, tels que les indices de Sobol [19], et ceux basés sur la théorie de l'information [16], qui sont utilisés en apprentissage automatique pour identifier les caractéristiques les plus significatives [4, 40]. L'information mutuelle est un exemple de cette dernière approche [1].

3.1 Définitions et concepts de base

Une histoire de facteurs Historiquement, l'analyse de sensibilité se confond avec l'étude des facteurs [55]. Dans le domaine de l'analyse de sensibilité, nous pouvons voir un facteur comme la donnée de son niveau $l \in L_F$ et des sous-ensembles $(F^{-1}(l))_{l \in L_F}$ qui forment une partition de Ω . Le facteur $COULEUR = \{ROUGE, NOIR\}$, peut s'écrire plutôt $F_{COULEUR} : \Omega \rightarrow COULEUR = \{ ROUGE, NOIR \}$. Remarquons que les facteurs qui ne diffèrent que par le labeling des classes ne devraient pas être distingués et que les niveaux que nous n'observons jamais expérimentalement peuvent être ignorés.

Les facteurs peuvent être ordonnés d'une manière évidente. Par exemple, si nous réalisons une expérience sur des patients, le facteur PATIENT (c'est à dire le numéro du patient) est **plus fin** que le facteur SEX (le sexe du patient).

Definition 3.1.1. Soit Ω un ensemble fini. Un facteur F est une application

$$F : \Omega \longrightarrow L_F$$

où L_F est un ensemble fini (l'ensemble de labels ou des niveaux de facteurs) de cardinal n_F représentant les niveaux du facteur.

Note. Nous pouvons voir un facteur comme une variable, ou un paramètre. L'exemple le plus simple de facteur est :

$$\begin{aligned} F_i : \Omega &\longrightarrow \Omega_i \\ \omega = (\omega_1, \dots, \omega_d) &\longmapsto \omega_i \end{aligned}$$

Dans la suite nous utiliserons le mot **variable** plutôt que le terme **facteur**. Cela est un abus de langage dont il faut être conscient. Cette distinction s'est montrée utile dans les éléments théoriques, et nous les laissons en annexe pour le lecteur intéressé plus précisément.

3.2 Propriétés souhaitables d'un indice de sensibilité

Il peut être intéressant de dire un mot sur l'analyse de sensibilité et ses objectifs avant de commencer toute présentation plus avancée. L'analyse de sensibilité étudie la contribution relative de l'incertitude (variabilité) des entrées x sur l'incertitude (variabilité) de la sortie y . Alors que l'analyse de sensibilité locale considère l'incertitude provenant des variations des entrées autour d'un point spécifique, l'analyse de sensibilité globale considère les variations des entrées dans tout leur espace de faisabilité. Souvent, trois objectifs spécifiques (paramètres) pour l'analyse de sensibilité globale sont définis [44] :

1. La priorisation des facteurs (FP) vise à classer les entrées x_i en fonction de leur contribution relative à l'incertitude de la sortie,
2. Factor Fixing (FF), ou screening, vise à déterminer les inputs, s'il y en a, qui ne contribuent pas à l'incertitude de la production,
3. La cartographie des facteurs (FM) vise à déterminer les régions dans l'espace des entrées qui produisent des valeurs de sortie spécifiques, par exemple au-dessus d'un seuil prescrit.

L'analyse de Sensibilité globale visant les FP et FF emploie souvent des indices de sensibilité, ou mesures d'importance. Il s'agit d'indices synthétiques qui quantifient la contribution relative de chaque entrée à l'incertitude des résultats. Un indice de sensibilité de zéro signifie que l'entrée associée est non influente (ce qui est utile pour FF), tandis que plus l'indice est élevé, plus l'entrée est influente (FP).

Nous allons étudier les différentes métriques d'analyse de sensibilité. Quelles seront les bonnes propriétés d'un indice de sensibilité? [28], s'appuyant sur [43] et [17], discutent des propriétés clés qu'un "bon" indice de sensibilité global devrait satisfaire. Nous pouvons résumer ces propriétés par :

1. **Être global**, c'est-à-dire prendre en compte les variations des entrées dans tout leur ensemble de définition,
2. **Être quantitatif**, c'est-à-dire calculable par une procédure numérique et reproductible,
3. **Être indépendant du modèle**, c'est-à-dire applicable indépendamment de la forme de la relation entrée-sortie du modèle, par exemple linéaire ou non linéaire, additive ou non additive,
4. **Être inconditionnel à toute valeur d'entrée supposée**. Un exemple d'approche qui ne satisfait pas à la propriété ci-dessus est l'approche basée sur l'entropie décrite plus en détail plus tard, qui calcule la sensibilité à la i -ième entrée en comparant les distributions de sortie obtenues lorsque toutes les entrées varient et lorsqu'elles varient toutes sauf x_i . Cette approche est globale, car nous laissons les entrées varier dans tout leur espace de faisabilité, mais elle n'est pas inconditionnelle, car les résultats dépendent de la valeur de conditionnement de x_i ,
5. **Être facile à interpréter**. Par exemple, les indices de sensibilité basés sur la variance sont "faciles à interpréter" dans la mesure où ils représentent la contribution à la variance de la sortie due aux variations d'une entrée. Ils prennent donc des valeurs comprises entre zéro et un, quelle que soit la plage de variation de la sortie y . Cela est très utile pour comparer les résultats entre différentes études de cas ou différentes définitions de la sortie,
6. **Être facile à calculer**. Lorsqu'on dit qu'un indice de sensibilité est "facile à calculer", cela signifie que la procédure de calcul pour son approximation est facile à mettre en œuvre, même si elle peut prendre du temps à exécuter. Par exemple, l'estimation des indices de sensibilité basés sur la variance peut prendre du temps, car elle nécessite de nombreuses évaluations, tout en restant facile à mettre en œuvre car, une fois que l'échantillon d'évaluations de sortie a été généré, les indices de sensibilité peuvent être approximés

en appliquant simplement une équation algébrique à forme fermée sur les échantillons de sortie,

7. **Être stable**, c'est-à-dire fournir des résultats cohérents d'un échantillon à l'autre ou d'une simulation à l'autre. Comme cette propriété peut être difficile à démontrer, une version faible pourrait ressembler à ceci : il devrait être possible d'évaluer facilement la robustesse de l'indice de sensibilité estimé pour différents échantillons ou différentes tailles d'échantillons. Dans cette version faible, la propriété est fortement liée à la propriété de facilité de calcul. En effet, si un indice de sensibilité est facile à calculer, il est également très facile d'évaluer sa robustesse et sa convergence en répétant les calculs sur différents rééchantillons bootstrapping et pour différentes tailles d'échantillons,
8. **Être indépendant du moment**, c'est-à-dire ne pas supposer un moment spécifique de la distribution de la sortie pour caractériser pleinement l'incertitude de la sortie.

Les indices de sensibilité basés sur la variance (Sobol) satisfont toutes les propriétés ci-dessus, sauf la dernière. En effet, ils reposent sur l'hypothèse que le moment de second ordre, c'est-à-dire la variance de la sortie, est suffisant pour caractériser complètement l'incertitude de la sortie, ce qui peut ne pas être le cas si la distribution de la sortie est multimodale ou si elle est fortement asymétrique.

3.3 Indices de Sobol et de Shapley

Cette partie n'étant pas cruciale dans la compréhension du mémoire, nous proposons ici de rappeler uniquement les résultats les plus importants sur Sobol.

3.3.1 Indices de Sobol

Les indices de Sobol sont les premiers indices basés sur la variance que nous allons présenter. Soit $\mathbf{x} = (x_1, \dots, x_p)$ un vecteur de l'hypercube unité

$$\Omega = \{\mathbf{x} \in \mathbb{R}^p, 0 \leq x_i \leq 1, \forall i \in [1 : p]\},$$

la décomposition fonctionnelle ANOVA de $\eta(\mathbf{x})$, une fonction quelconque définie sur Ω , consiste à exprimer η comme une somme de fonctions de dimension croissante :

$$\begin{aligned} \eta(\mathbf{x}) &= \eta_\emptyset + \sum_{i=1}^p \eta_i(x_i) + \sum_{1 \leq i < j \leq p} \eta_{i,j}(x_i, x_j) + \dots + \eta_{1, \dots, p}(\mathbf{x}) \\ &= \sum_{u \in S} \eta_u(\mathbf{x}_u), \end{aligned}$$

la fonction η se décompose ainsi en une somme de 2^p fonctions, où η_\emptyset est une constante, $\eta_i, i \in [1 : p]$ sont les effets principaux, les fonctions $\eta_{i,j}, i < j \in [1 : p]$ représentent les effets d'interaction d'ordre 2, etc.

Cette décomposition s'appelle aussi la décomposition HDMR. Elle n'est pas unique, sauf si nous imposons une condition d'orthogonalité sur les composantes

$$\int_{[0,1]} \eta_u(\mathbf{x}_u) dx_i = 0 \quad \forall i \in u, \forall u \in S,$$

une des conséquences de cette condition est que les termes de la décomposition sont deux à deux orthogonaux :

$$\int_{[0,1]^p} \eta_u(\mathbf{x}_u) \eta_v(\mathbf{x}_v) d\mathbf{x} = 0 \quad \forall u \neq v \in S,$$

revenons maintenant à un cadre stochastique. Considérons que ν est la mesure de Lebesgue sur $(\mathbb{R}^p, \mathcal{B}(\mathbb{R}^p))$ et \mathbf{X} est un vecteur de variables aléatoires indépendantes uniformes, c'est-à-dire $\mathbf{X} \sim \mathcal{U}([0, 1]^p)$ et $p_{\mathbf{X}} = \mathbb{I}_{[0, 1]^p}$, où $\mathbb{I}_{(\cdot)}$ dénote la fonction indicatrice. La décomposition ANNOVA, sous les contraintes données d'orthogonalité reste vraie sur le modèle $\eta(\mathbf{X})$. Celle-ci assure la décomposition de la variance globale en une somme de variances partielles comme donnée ci-dessous :

$$\begin{aligned} V(Y) &= \sum_{i=1}^p V[\eta_i(X_i)] + \sum_{1 \leq i < j \leq p} V[\eta_{i,j}(X_i, X_j)] + \cdots + V[\eta_{1, \dots, p}(\mathbf{X})] \\ &= \sum_{u \in S} V[\eta_u(\mathbf{X}_u)]. \end{aligned}$$

Il en résulte la définition des indices de Sobol, telle qu'on la trouve dans les travaux de l'auteur en question :

Definition 3.3.1. *L'indice de Sobol S_u d'ordre $|u|$ mesurant la contribution du groupe de variables \mathbf{X}_u dans le modèle $Y = \eta(\mathbf{X})$ est donné par*

$$S_u = \frac{V(\eta(\mathbf{X}_u))}{V(Y)} = \frac{V[\mathbb{E}(Y | \mathbf{X}_u)] + \sum_{v \subset u} (-1)^{|u|-|v|} V[\mathbb{E}(Y | \mathbf{X}_v)]}{V(Y)},$$

en particulier, l'indice d'ordre 1 mesurant la contribution de X_i s'exprime comme

$$S_i = \frac{V[\mathbb{E}(Y | X_i)]}{V(Y)}.$$

Si la condition d'orthogonalité est bien vérifiée, nous trouvons que

$$0 \leq S_u \leq 1, \quad \forall u \in S \setminus \{\emptyset\},$$

et

$$\sum_{u \in S \setminus \{\emptyset\}} S_u = 1,$$

plus une variable explique la *variabilité* de la sortie, plus son indice de Sobol sera proche de 1.

Afin de fixer les idées, nous reprenons la distribution multimodale que nous avons introduit dans le chapitre du shift, avec $X \sim N(0, 1)$, $Y \sim N(10, 1)$ et $Z \sim B(0.5)$. Avec :

$$f : (x, y, z) \longrightarrow (2x^2 + 3y)(zx + (1 - z)y), \quad (3.1)$$

nous rappelons que nous obtenons une distribution multimodale :

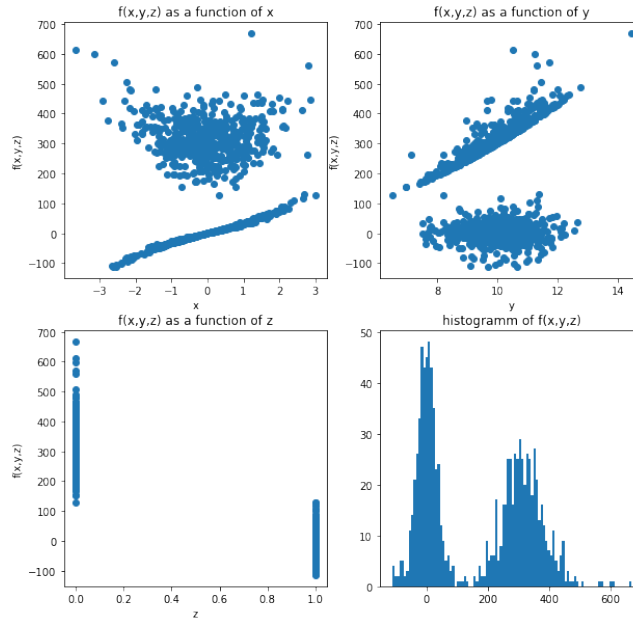


FIGURE 3.1 – Exemple fil rouge.

Nous pouvons, dans ce cas précis, calculer assez simplement les indices de Sobol, car nous disposons d’une quantité illimitée de données.

	X	Y	Nombre d’observations
Indice de sensibilité Sobol	0.06	0.04	12000

Ce tableau de résultat peut s’interpréter comme *la variable X ne contribue qu’à 6% de la variance de la sortie $f(X, Y, Z)$ tandis que Y ne contribue qu’à 4%. Nous ne mettons pas Z car cette variable ne contribue même pas*. Bien que nous ne l’ayons pas relevé dans le tableau, nous précisons que les intervalles de confiance obtenus pour les indices de Sobol présentés sont très resserrés autour des indices calculés¹.

Nous nous rendons pourtant compte que la connaissance de X et de Y permet d’expliquer en grande partie les valeurs prises par la variable aléatoire $f(X, Y, Z)$. Nous voyons ici que les indices de Sobol ne sont pas efficaces dans le cadre d’une distribution multimodale, comme celle que nous pourrions rencontrer pour le shift.

La figure 3.2 donne une intuition sur la raison pour laquelle les indices de Sobol ne fonctionnent pas. Nous nous rendons compte que l’espérance conditionnelle, $\mathbb{E}(Y | X_i)$, tracée en rouge, est très peu variable (due au caractère multimodale de la distribution). Par la suite, $V[\mathbb{E}(Y | X_i)]$ sera très faible, alors que $V(Y)$ est lui très grand. L’indice de Sobol $S_i = \frac{V[\mathbb{E}(Y | X_i)]}{V(Y)}$ n’a donc effectivement que peu de sens ici.

1. Voir la librairie SALib

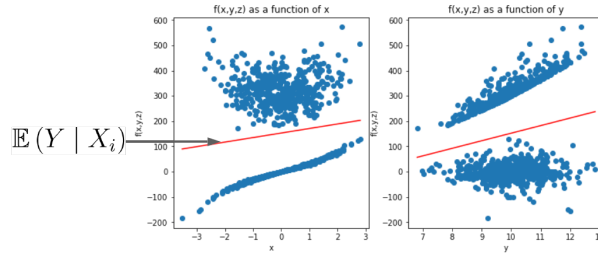


FIGURE 3.2 – Intuition sur les indices de Sobol.

3.3.2 Indices de Shapley

Les valeurs de Shapley, bien connues en théorie des jeux et en économie, ont été récemment proposées par [39] en analyse de sensibilité de modèles, via l'expression :

$$Sh_i = \sum_{u \subseteq -\{i\}} \frac{(d - |u| - 1)! |u|!}{d!} [c(u \cup \{i\}) - c(u)],$$

où $c(\cdot)$ est une fonction coût et $-\{i\}$ est l'ensemble des indices $\{1 \dots d\}$ privé de i . Dans le contexte de l'analyse de sensibilité (sans l'hypothèse d'entrées indépendantes) et en utilisant $c(u) = S_u^{\text{clo}} = \text{Var}(\mathbb{E}[Y | X_u]) / \text{Var}(Y)$ (S_u^{clo} contient les effets des variables et interactions qui sont dans u), [39] définit ainsi de nouveaux indices de sensibilité nommés "Shapley effects" (Indices de Shapley). Chacun de ces indices, dont la somme vaut un, comprend une part de la variance de Y . Par ailleurs, [50] montrent qu'il est équivalent d'utiliser $c(u) = \mathbb{E}[\text{Var}(Y | X_{-u})] / \text{Var}(Y)$.

Les indices de Shapley, eux aussi basés sur la variance, ne nous serons pas non plus d'une grande utilité par rapport aux indices de Sobol. Il convient alors de définir une nouvelle classe d'indices de sensibilité, qui pourra être adaptée à notre problématique.

3.4 Théorie de l'information

Les indices basés sur la décomposition de la variance montrent des limitations importantes sur de nombreux exemples concrets. Le papier de Huoh [16] présente une étude complète de sensibilité en utilisant la théorie de l'information. Cela conduira à définir une quantité, l'information mutuelle qu'il nous faudra estimer. Il est très intéressant de noter que la recherche récente en machine learning s'est penchée sur le sujet [4]. L'analyse de sensibilité a été en premier utilisée en Machine learning pour déterminer les features avec le plus d'importance [40]. [36] proposent une analyse de sensibilité de processus stochastiques, que nous ne traiterons pas ici.

Dans cette section aussi, lorsque nous allons évoquer des objets aléatoires, nous allons faire référence à l'espace probabilisé (Ω, \mathcal{E}, P) ainsi que $L^2(\Omega, \mathcal{E}, P; R)$ l'espace vectoriel des variables aléatoires réelles de carré intégrable défini sur (Ω, \mathcal{E}, P) . Nous munissons cet espace vectoriel du produit scalaire usuel $\langle Y_1, Y_2 \rangle_P = \int Y_1 Y_2 dP, Y_1, Y_2 \in L^2(\mathcal{E})$.

3.4.1 Définitions de l'information mutuelle

Il existe un très grand nombre de mesures d'entropies différentes. Une énumération peut être trouvée dans [36]. Pour commencer, nous proposons de reprendre la plus ancienne, l'entropie de Shannon [47].

Definition 3.4.1. Soit X une variable aléatoire discrète définie sur (Ω, \mathcal{E}, P) . L'entropie de X est définie avec des notations évidentes comme

$$H(X) = - \sum_{x \in \mathbf{X}} \log\{p(x)\}p(x),$$

si nous utilisons le log en base e , nous dirons que l'entropie est en "nats". Il est important de noter que celle-ci est toujours positive.

En conservant nos notations, nous définissons l'entropie jointe de deux variables aléatoires X et Y comme :

$$H(X, Y) = - \sum_{x \in \mathbf{X}} \sum_{y \in \mathbf{Y}} \log\{p(x, y)\}p(x, y),$$

puis l'entropie conditionnelle :

$$\begin{aligned} H(Y | X) &= - \sum_{x \in \mathbf{X}} \sum_{y \in \mathbf{Y}} \log\{p(y | x)\}p(x, y) \\ &= \sum_{x \in \mathbf{X}} p_x(x) \sum_{y \in \mathbf{Y}} \log\{p(y | x)\}p(y | x), \end{aligned}$$

L'intérêt de l'entropie et ses interprétations dans la suite de ce mémoire seront multiples. Cependant, il est important que c'est la règle de la chaîne qui motive son utilisation.

Theorem 3.4.1. (Règle de la chaîne) Soit X et Y des variables aléatoires définies sur (Ω, \mathcal{E}, P) . Alors, si les quantités suivantes sont bien définies, nous avons la relation suivante :

$$\begin{aligned} H(X, Y) &= H(X) + H(Y | X) \\ &= H(Y) + H(X | Y). \end{aligned}$$

Jusqu'ici nous n'avons fait aucune hypothèse sur la dépendance des variables aléatoires. Si celles-ci sont indépendantes, nous avons $H(X, Y) = H(X) + H(Y)$. Nous généralisons avec des notations évidentes l'entropie au cas continue comme $H(X) = - \int_{\mathbf{X}} \log\{f(x)\}f(x)dx$.

Definition 3.4.2. Soit X et Y des variables aléatoires définies sur (Ω, \mathcal{E}, P) de densité $f(x, y)$. L'information mutuelle entre X et Y est définie comme :

$$MI(X, Y) = \int_{\mathbf{X}} \int_{\mathbf{Y}} \log \left\{ \frac{f(x, y)}{f_X(x)f_Y(y)} \right\} f(x, y)dydx.$$

L'information mutuelle possède un bon nombre de propriétés, notamment la symétrie, $MI(X, Y) = MI(Y, X)$ et la non-négativité $MI(X, Y) \geq 0$ (avec égalité si et seulement si X et Y sont indépendantes). Dans le cas de variables aléatoires continues, l'information mutuelle peut prendre des valeurs infinies, par exemple si $X = Y$. Cependant, les caractéristiques suivantes font de l'information mutuelle un bon candidat pour l'analyse de sensibilité :

1. Son interprétation ne repose pas sur la structure de dépendance des variables d'inputs,
2. Son indépendance vis-à-vis du moment fait qu'elle convient à un large éventail de variables d'entrées et de variables de sorties (en particulier elle permet de lever le problème des distributions multimodales),
3. Sa nature continue est adaptée à l'analyse des systèmes continus.

La manipulation suivante :

$$\begin{aligned}
 MI(X, Y) &= \int_{\mathbf{X}} \int_{\mathbf{Y}} \log \left\{ \frac{f(x, y)}{f_X(x) f_Y(y)} \right\} f(x, y) dx dy \\
 &= - \int_{\mathbf{X}} \int_{\mathbf{Y}} \log \{f_X(x) f_Y(y)\} f(x, y) dx dy + \int_{\mathbf{X}} \int_{\mathbf{Y}} \log \{f(x, y)\} f(x, y) dx dy \\
 &= H(X) + H(Y) - H(X, Y),
 \end{aligned}$$

nous permet d'arriver à un résultat particulièrement intéressant :

$$\boxed{MI(X, Y) = H(Y) - H(Y | X)}$$

Nous avons que **l'information mutuelle est la diminution de l'entropie d'une variable due à la connaissance d'une autre variable.**

Enfin, afin de limiter les variations que nous observons pour l'information mutuelle (potentiellement infinie), nous utilisons une transformation de l'information mutuelle comme nouvel indice de sensibilité.

Récemment, [1] a proposé un nouvel indice de sensibilité appelé le coefficient d'information mutuelle, que nous utiliserons dans notre étude. Ce coefficient est défini comme

$$\rho_{MI}(X_1, X_2) = \sqrt{1 - \exp[-2MI(X_1, X_2)]},$$

au lieu de normaliser l'information mutuelle par l'entropie de Y comme dans [2]. ρ_{MI} est une *mesure appropriée de la dépendance*. Nous avons $\rho_{MI} = 0$ pour des variables indépendantes, et $\rho_{MI} = 1$ en cas de dépendance parfaite. De plus, ρ_{MI} et le coefficient de corrélation linéaire coïncident dans le cas d'une distribution conjointe gaussienne.

L'information mutuelle peut être considéré comme un indice de sensibilité adapté pour les distributions multimodales. Cependant, et c'est bien connu dans la littérature, son estimation n'est pas une chose aisée. Nous allons nous pencher dans la suite sur plusieurs méthodes d'estimation et nous allons proposer un nouveau cadre pour son estimation.

3.4.2 Estimation de l'information mutuelle

Nous allons présenter plusieurs méthodes d'estimation de l'information mutuelle, mais certaines d'entre elles demande d'avoir en tête la définition d'un réseau de neurones *multi layer perceptron*.

Le tout premier réseau de neurones construit est le multi layer perceptron (MLP). Nous prenons ici les notations de [35] afin d'introduire au lecteur cet objet.

Definition 3.4.3. Une fonction continue montone $\sigma : \mathbb{R} \rightarrow \mathbb{R}$ est appelée fonction d'activation.

Note. Les fonctions d'activations les plus courantes sont la sigmoïde, la tangente hyperbolique, le fonction ReLU. Nous aurons l'occasion par la suite de revenir sur ces points.

Definition 3.4.4. (Multi-Layer Perceptron ou perceptron multi-couches) Soit $X \in \mathbb{R}^n$, et $\sigma_1, \dots, \sigma_n$ des fonctions d'activation. Soit $\Sigma(X)$ donné par :

$$\Sigma(X) = \Sigma \left([x_1, \dots, x_n]^T \right) = [\sigma_1(x_1), \dots, \sigma_n(x_n)]^T,$$

soit $n_1, n_2, \dots, n_k \in \mathbb{N}$, alors la fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}^{n_k}$ définie par :

$$f(X) = f(X; W) = \Sigma_k \left(W_k \cdot \Sigma_{k-1} \left(W_{k-1} \cdot \Sigma_{k-2} \left(\dots \Sigma_2 \left(W_2 \cdot \Sigma_1 \left(W_1 \cdot X \right) \right) \right) \right) \right),$$

est appelée un Multi Layer Perceptron avec k couches. Avec $W_1 \in M_{n_1 \times n}(\mathbb{R})$, \dots , $W_i \in M_{n_i \times n_{i-1}}(\mathbb{R})$ les matrices des poids du réseau de neurones.

Nous pouvons noter par $\omega = (W_1, W_2, \dots, W_k)$ les poids du réseau de neurones. Nous choisissons de noter par \mathcal{W} l'ensemble des poids du réseau de neurones. Dès lors que l'on initialise les poids pour commencer à entraîner le réseau de neurones, nous notons :

$$\exists \omega \in \mathcal{W} / f(X) = f_\omega(X),$$

nous définissons ensuite une *loss function*, qui permet de quantifier pour un input donné à quel point le réseau de neurones produit un output qui est loin de la sortie voulue. L'apprentissage consiste à mettre à jour les poids $\omega \in \mathcal{W}$ du réseau de neurones afin de minimiser son erreur relativement à une fonction de perte.

3.4.2.1 Avec le réseau de neurones MINE

[4] introduit MINE, *mutual information copulas estimator* qui est un estimateur neuronal de l'information mutuelle. Les auteurs argumentent "*Nous soutenons que l'estimation de l'information mutuelle entre des variables aléatoires continues de haute dimension peut être réalisée par descente de gradient sur des réseaux neuronaux.*" Et présentent MINE un réseau de neurones particulièrement intéressant pour réaliser cette tâche précise. **Nous avons dans cette section un estimateur aussi précis que l'on veut de l'information mutuelle. La complexité de l'échantillon devra être en $\tilde{O}\left(\frac{d \log d}{\epsilon^2}\right)$, où d est la dimension de l'espace.**

Theorem 3.4.2. *L'information mutuelle est d'après [4] équivalente à la divergence de Kullback-Leibler (KL) entre P_{XY} et $P_X \otimes P_Y$:*

$$MI(X, Z) = D_{KL}(P_{XY} \| P_X \otimes P_Y),$$

où D_{KL} est définie par :

$$D_{KL}(P \| Q) := E_P \left[\log \frac{dP}{dQ} \right],$$

quand P est absolument continu par rapport à Q . L'interprétation est simple : plus la divergence entre la distributions jointes et produits marginales est grande plus la dépendance entre les variables X et Y est forte.

On peut donner une démonstration immédiate de ce résultat :

$$\begin{aligned} MI(X, Z) &= H(X) - H(X | Z) \\ &= - \int_x p(x) \log p(x) dx + \int_{x,z} p(x, z) \log p(x | z) dx dz \\ &= \int_{x,z} (-p(x, z) \log p(x) + p(x, z) \log p(x | z)) dx dz \\ &= \int_{x,z} \left(p(x, z) \log \frac{p(x, z)}{p(x)p(z)} \right) dx dz \\ &= D_{KL}(P(X, Z) \| P(X) \otimes P(Z)), \end{aligned}$$

le théorème de Donsker-Varadhan va nous permettre de nous ramener à un problème d'optimisation.

Theorem 3.4.3. (*Donsker-Varadhan*) *On a la représentation suivante de Donsker-Varadhan :*

$$D_{KL}(P \| Q) = \sup_{T: \Omega \rightarrow \mathbb{R}} E_P[T] - \log (E_Q [e^T]),$$

où le supremum est pris sur tout les fonctions T pour lesquelles les deux espérances sont finies.

Ce théorème étant fondamental, nous proposons de donner quelques éléments de preuve pour ce dernier. Nous commençons par définir la distribution de Gibbs :

$$G(x) = \frac{1}{Z} e^{T(x)} Q(x),$$

où $Z = \mathbf{E}_Q [e^T]$. Puis nous avons :

$$\begin{aligned} \mathbf{E}_P[T] - \log(\mathbf{E}_Q [e^T]) &= \mathbf{E}_{x \sim P(x)} [T(x) - \log \mathbf{E}_Q [e^T]] \\ &= \mathbf{E}_{x \sim P(x)} \left[\log \frac{e^{T(x)}}{\mathbf{E}_Q [e^T]} \right] \\ &= \mathbf{E}_{x \sim P(x)} \left[\log \frac{e^{T(x)} Q(x)}{\mathbf{E}_Q [e^T] Q(x)} \right] \\ &= \mathbf{E}_{x \sim P(x)} \left[\log \frac{G(x)}{Q(x)} \right] \\ &= \mathbf{E}_P \left[\log \left(\frac{G}{Q} \right) \right], \end{aligned}$$

et

$$\Delta = D_{KL}(P||Q) - (\mathbf{E}_P[T] - \log(\mathbf{E}_Q [e^T])) = \mathbf{E}_P \left[\log \left(\frac{P}{Q} \right) \right] - \mathbf{E}_P \left[\log \left(\frac{G}{Q} \right) \right],$$

enfin :

$$\Delta = \mathbf{E}_P \left[\log \left(\frac{P}{G} \right) \right] = D_{KL}(P||G),$$

on conclut par positivité de la KL divergence, avec $\Delta > 0$.

Dans la partie d'apprentissage statistique, nous allons approfondir la théorie de l'information pour trouver des estimateurs avancés de cette information mutuelle. Nous proposerons aussi une introduction aux modèles génératifs qui nous seront utiles pour classifier les données de marché.

On choisit une famille \mathcal{F} de fonctions T_θ approximées par un deep neural network de paramètres $\theta \in \Theta$. Nous appelons ce réseau de neurone *statistique network*. Nous exploitons la frontière :

$$MI(X; Y) \geq MI_\Theta(X, Y),$$

conséquence immédiate du théorème de Donsker-Varadhan. Avec $MI_\Theta(X, Y)$ est la *neural information measure*, définie par :

$$MI_\Theta(X, Y) = \sup_{\theta \in \Theta} \mathbb{E}_{\mathbb{P}_{XY}} [T_\theta] - \log(\mathbb{E}_{\mathbb{P}_X \otimes \mathbb{P}_Y} [e^{T_\theta}]),$$

cet objectif peut être atteint par une descente classique de gradient (apprentissage profond). Dans ce qui suit, étant donné une distribution \mathbb{P} , nous notons par $\hat{\mathbb{P}}(n)$ la distribution empirique associée à n i.i.d. samples.

Soit $\mathcal{F} = \{T_\theta\}_{\theta \in \Theta}$ un ensemble de fonctions paramétrées par un réseau de neurones. Nous apprenons par descente de gradient les paramètres du réseau de neurones. Des théorèmes classiques de machine learning assurent la convergence de cet algorithme.

Nous faisons les hypothèses suivantes : les fonctions T_θ sont M -bornées (c'est-à-dire $|T_\theta| \leq M$) et L -Lipschitziennes par rapport aux paramètres θ . Le domaine $\Theta \subset \mathbb{R}^d$ est borné, de sorte que $|\theta| \leq K$ pour une certaine constante K . Le théorème ci-dessous montre une complexité d'échantillonnage de $\tilde{O}\left(\frac{d \log d}{\epsilon^2}\right)$, où d est la dimension de l'espace des paramètres.

Theorem 3.4.4. *Étant donné des valeurs quelconques ϵ, δ des paramètres de précision et de confiance souhaités, nous avons,*

$$\Pr(|\widehat{MI}(X;Y)_n - MI_{\Theta}(X,Y)| \leq \epsilon) \geq 1 - \delta,$$

dès que le nombre n d'échantillons vérifie :

$$n \geq \frac{2M^2(d \log(16KL\sqrt{d}/\epsilon) + 2dM + \log(2/\delta))}{\epsilon^2}.$$

La démonstration peut être trouvée dans le papier original [4]. Lorsque l'on reprend nos calculs, en utilisant MINE, nous obtenons pour notre fonction fil rouge :

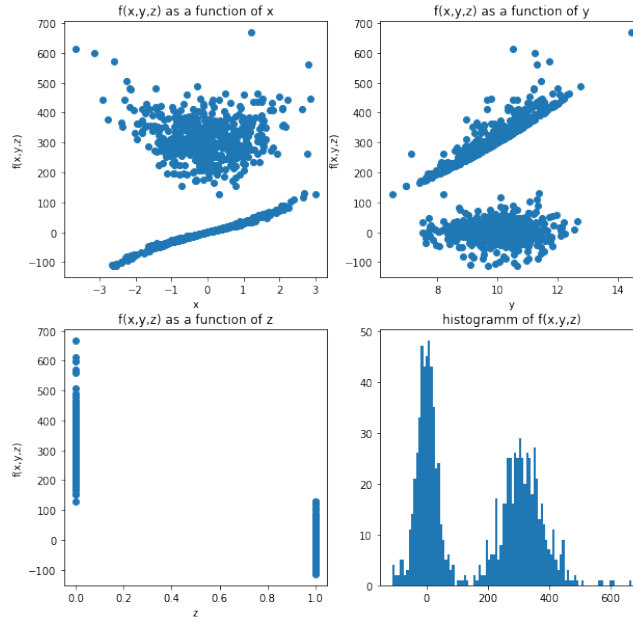


FIGURE 3.3 – Exemple fil rouge.

		X	Y	Nombre d'observations
Indices de variance	Sobol	0.06	0.04	12000
ρ_{MI}	MINE	0.94	0.53	1000

L'indice basée sur l'information mutuelle se trouve être bien plus performant que l'indice de Sobol. Il détecte que nous expliquons bien la sortie quand nous connaissons X ou Y , nous expliquons très bien les valeurs prises par la distribution. Pour le calcul du coefficient basé sur l'information mutuelle, l'intervalle de confiance, calculé avec la méthode précédente, est lui aussi très serré autour des indices annoncés. Il existe aussi une méthode basée sur les plus proches voisins pour estimer cette information mutuelle.

3.4.2.2 Avec les plus proches voisins (k-nn)

Cette méthode vise à estimer directement l'entropie. L'estimation de l'entropie du k ième plus proche voisin est la suivante :

$$\hat{H}_n(X) = -\psi(k) + \psi(n) + \frac{p}{n} \sum_{i=1}^n \log \{\epsilon_{i,k}\},$$

où ψ est la fonction digamma,

$$\psi(x) = \frac{d}{dx} \log \Gamma(x) = \frac{\Gamma'(x)}{\Gamma(x)},$$

p est la dimension de l'espace d'entrée \mathbf{X} , et $\epsilon_{i,k}$ est la distance de x_i à son k -ième plus proche voisin. Ici, le terme distance fait référence à la norme du supremum, à savoir

$$d(\mathbf{x}, \mathbf{y}) = \max_j \{x_1 - y_1, \dots, x_p - y_p\}.$$

Par analogie, nous allons pouvoir estimer ensuite l'information mutuelle :

$$\begin{aligned} \hat{M}I_n(X, Y) &= -\psi(k) + \psi(n) + \frac{p_x}{n} \sum_{i=1}^n \log \{\epsilon_{i,k}^x\} - \psi(k) + \psi(n) + \frac{p_y}{n} \sum_{i=1}^n \log \{\epsilon_{i,k}^y\} \\ &\quad + \psi(k) - \psi(n) - \frac{p_{(x,y)}}{n} \sum_{i=1}^n \log \{\epsilon_{i,k}^{(x,y)}\}, \end{aligned}$$

où $\epsilon_{i,k}^x$ est la distance entre x_i et son k -ième plus proche voisin, $\epsilon_{i,k}^y$ est la distance entre y_i et son k -ième plus proche voisin, et $\epsilon_{i,k}^{(x,y)}$ est la distance entre (x_i, y_i) et son k -ième plus proche voisin.

Note. Notez qu'étant donné qu'ils se trouvent dans des espaces différents, les voisins ne correspondent pas nécessairement pour les différents ϵ .

Après quelques calculs de simplification, nous obtenons que le premier estimateur d'information mutuelle est

$$\hat{M}I_n^{(1)}(X, Y) = \psi(k) + \psi(n) - \frac{1}{n} \sum_{i=1}^n [\psi(n_x(i) + 1) + \psi(n_y(i) + 1)].$$

Comme précédemment, ψ est la fonction digamma. Ici, $n_x(i)$ est le nombre de points d'échantillon dont les valeurs de x sont plus proches que la distance entre (x_i, y_i) et son k -ième plus proche voisin. Formellement :

$$n_x(i) = \# \left\{ x_j : d(x_i, x_j) \leq \epsilon_{i,k}^{(x,y)} \right\},$$

où, comme précédemment, $\epsilon_{i,k}^{(x,y)}$ est la distance de (x_i, y_i) de son k -ième plus proche voisin. Pour $\epsilon_{i,k}^{(x,y)}$ et $d(x_i, x_j)$, la distance utilisée est la norme du supremum. $n_y(i)$ est défini de manière similaire.

Alors, nous obtenons :

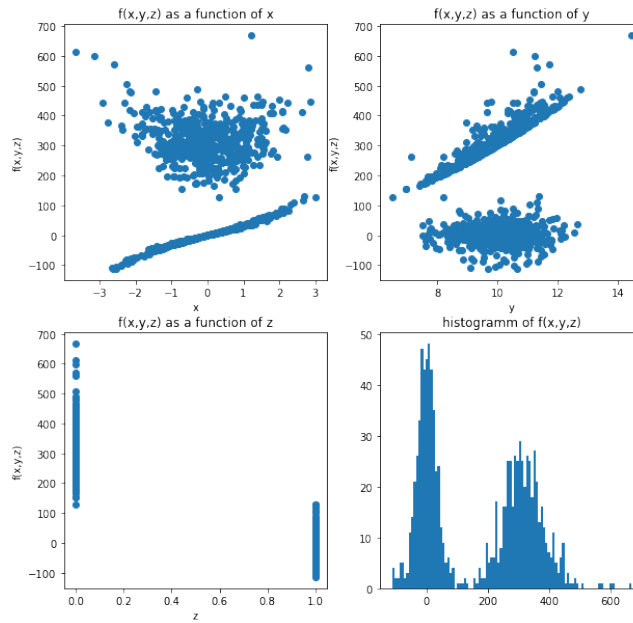


FIGURE 3.4 – Exemple fil rouge.

		X	Y	Nombre d'observations
Indices de variance	Sobol	0.06	0.04	12000
	MINE	0.94	0.53	1000
	K-NN	0.97	0.87	1000

3.4.3 Ces méthodes sont elles satisfaisantes ?

Nous pouvons nous poser des questions quant à l'interprétabilité des méthodes d'analyse de sensibilité dans un cadre général. Par exemple, dans le cas de MINE, comme dans le cas de la méthodes des plus proches voisins, aucune interprétation *physique* n'est donnée à la construction des indices de sensibilité.

De plus, si nous changeons le domaine de définition des variables que nous souhaitons analyser, c'est à dire si l'on souhaite faire de l'analyse de sensibilité locale, il n'y a pas d'autres choix que de relancer les calculs, et de comparer les indices, sans a priori avoir d'interprétation plus physique.

Dans le chapitre suivant, nous proposons un cadre nouveau pour l'estimation de l'information mutuelle. Celui ci sera plus explicite, et permettra des interprétations à différents niveaux.

Il découle d'une analyse approfondie réalisée lors du mémoire, et nous étayons nos propos avec des données chiffrées à l'appui.

Chapitre 4

Mutual information copula estimator : un nouveau cadre pour l'analyse de sensibilité

Récemment, les copules ont regagné de l'intérêt en apprentissage automatique en raison de leur capacité à modéliser les dépendances dans des espaces de grande dimension [21]. Certains travaux ont proposé des autoencodeurs de copules pour une utilisation en tant que modèles génératifs [53, 51].

Nous avons été particulièrement intéressés par deux articles récents, [20] et [27]. [20] a proposé le premier modèle universel et non paramétrique pour estimer les distributions de copules de grande dimension avec une garantie d'uniformité des distributions marginales. De plus, [27] a proposé une méthode pour apprendre la densité des copules à partir d'observations et a utilisé l'algorithme de Monte-Carlo par chaînes de Markov pour simuler des données. Bien que les modèles génératifs tels que les autoencodeurs [22, 10] puissent apprendre à simuler des données, l'apprentissage explicite de la densité des copules n'est pas simple.

Par conséquent, dans cette partie du mémoire, nous proposons de combiner le modèle de copules génératives implicites (IGC) [20] et l'estimateur neuronal de densité de copules (CODINE) [27] pour créer l'estimateur de copules d'information mutuelle (MICE). Notre modèle proposé est capable d'apprendre la simulation de copules et l'estimation de densité et définit un indice d'information mutuelle locale qui relie les analyses de sensibilité globale et locale dans un seul cadre. Nous comparerons également notre modèle proposé à l'état de l'art de l'estimateur neuronal d'information mutuelle (MINE) [4] et démontrerons en quoi notre modèle est plus performant.

1. Nous introduisons un cadre qui permet d'apprendre simultanément la simulation de copules et l'estimation de densité de manière non paramétrique, ce qui est une première du genre.
2. Nous utilisons les copules pour effectuer à la fois une analyse de sensibilité globale et locale avec une seule calibration du modèle, et nous définissons un indice local d'information mutuelle.
3. Nous comparons notre méthode aux normes en matière d'analyse de sensibilité, telles que les indices de Sobol.

4.1 Lien entre copules et information mutuelle

Theorem 4.1.1. *Les copules proviennent du mot latin copula, qui relie les distributions marginales en une distribution conjointe. Le théorème de Sklar [7] garantit que si le vecteur aléatoire $\mathbf{X} = (X_1, \dots, X_d)$ a une distribution conjointe F et des distributions marginales F_1, \dots, F_d , alors il existe une unique copule C qui est la distribution conjointe de $\mathbf{U} = (U_1, \dots, U_d) =$*

$(F_1(X_1), \dots, F_d(X_d))$. On dit que \mathbf{U} suit la soi-disant distribution de copule P^C , une distribution dans l'espace unité avec des marginales uniformes.

La vision des copules comme d'une certaine distribution de probabilité est en effet cruciale et très importante dans ce qui va suivre.

Comme nous l'avons présenté plus haut, le concept d'entropie d'une variable aléatoire X a été introduit pour la première fois par [47], et est défini comme $H(X) = -\int_{\mathbf{X}} \log f(x) f(x) dx$. L'information mutuelle entre deux variables aléatoires X et Y est définie comme la diminution de l'entropie de Y lorsqu'on dispose d'informations sur X :

$$MI(X, Y) = \int_{\mathbf{X}} \int_{\mathbf{Y}} \log \frac{f(x, y)}{f_X(x) f_Y(y)} f(x, y) dy dx,$$

l'information mutuelle est positive, symétrique et égale à zéro si les deux variables sont indépendantes. Une relation remarquable est $MI(X, Y) = H(Y) - H(Y | X)$.

De plus, [4] a proposé l'estimateur neuronal d'information mutuelle (MINE) basé sur la relation surprenante $MI(X, Z) = D_{KL}(\mathbb{P}_{XY} \| \mathbb{P}_X \otimes \mathbb{P}_Y)$ et $D_{KL}(\mathbb{P} \| \mathbb{Q}) = \sup_{T: \Omega \rightarrow \mathbb{R}} \mathbb{E}_{\mathbb{P}}[T] - \log(\mathbb{E}_{\mathbb{Q}}[e^T])$. MINE approxime la fonction T qui maximise cette quantité.

Lien entre les copules et l'information mutuelle Aussi surprenant que cela puisse paraître, il existe un lien fort entre les copules et l'information mutuelle [30]. Certains travaux, contrairement aux nôtres, utilisent des copules paramétriques pour estimer l'information mutuelle [18, 59].

Le lien entre les copules et l'information mutuelle peut s'exprimer comme suit :

$$MI(X, Y) = E_{(U, V) \sim c(u, v)}[\log c(U, V)], \quad (4.1)$$

où $X = F_1^{-1}(U)$ et $Y = F_2^{-1}(V)$. Nous proposons d'apprendre à simuler le couple (U, V) et d'apprendre la densité $c(u, v)$ afin d'apprendre directement l'information mutuelle.

Ce lien est intéressant à plusieurs niveaux. Premièrement, il nous apparaît directement une interprétation **physique** de l'information mutuelle. En effet, comme il est possible de représenter graphiquement la densité d'une copule, il nous est possible de donner une interprétation visuelle à la construction de l'indice d'information mutuelle. En plus de cela, nous avons la chance d'avoir un lien entre l'information mutuelle locale et globale.

On peut remarquer que cet indice mesure l'information mutuelle entre deux variables sur une plage de valeurs donnée, définie par les paramètres α et β . Plus précisément, nous proposons d'introduire l'indice d'information mutuelle locale :

$$MIL_{[\alpha, \beta]}(X, Y) = \int_{\alpha}^{\beta} \int_0^1 c(u, v) \log(c(u, v)) du dv, \quad (4.2)$$

où α et β sont les quantiles de X .

Note. Il est important de noter que lorsque α et β approchent respectivement 0 et 1, l'indice d'information mutuelle locale converge vers l'information mutuelle entre les deux variables.

Note. L'avantage de cet estimateur de sensibilité est qu'il peut être facilement calculé pour chaque α et β une fois que le modèle est entraîné, sans nécessiter de ressources informatiques supplémentaires.

4.2 MICE : le modèle

Notre proposition consiste à établir un lien entre IGC et CODINE. Nous allons dans un premier temps détailler le fonctionnement de ces deux algorithmes avant de poursuivre.

4.2.1 IGC : Implicit Generative Copulas

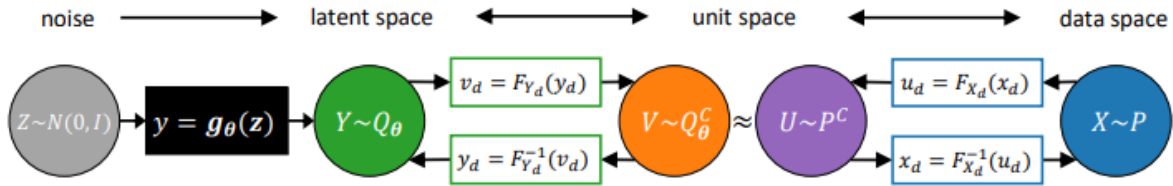


FIGURE 4.1 – Les étapes de transformations que suit IGC, schéma directement repris du papier original.

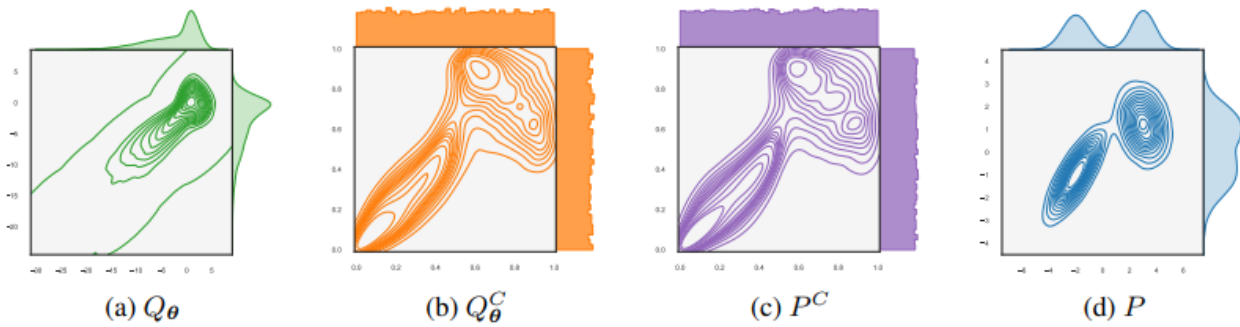


FIGURE 4.2 – Nous pouvons visualiser sur cette figures les étapes de transformations que propose IGC.

L'objectif de IGC est d'apprendre la distribution des copules P , comme indiquée dans le schéma précédent. En pratique, IGC donne une distribution Q_θ^C , et nous espérons après l'entraînement à avoir :

$$P^C = Q_\theta^C,$$

Pour ce faire, le modèle IGC suit une procédure qui peut se décomposer en plusieurs étapes :

1. Nous tirons des variables gaussiennes centrées réduites $\mathbf{Z} \in \mathbb{R}^K$,
2. Nous transformons la variable gaussienne en une distribution dite *latente* $\mathbf{Y} \in \mathbb{R}^D$ en utilisant la fonction $\mathbf{g}_\theta : \mathbb{R}^K \rightarrow \mathbb{R}^D$ approximée par un réseau de neurones :

$$\mathbf{y} = \mathbf{g}_\theta(\mathbf{z}),$$

3. Nous appliquons les fonctions de répartitions marginales $F_{Y_d}(y_d)$ aux composantes des variables latentes \mathbf{Y} . Cette étape de transformation est analogue au passage de \mathbf{X} à \mathbf{U} . Elle garantit à la fois que les valeurs d'échantillon v_d du modèle se situent dans l'intervalle unitaire $[0, 1]$ et qu'elles sont uniformément réparties, indépendamment de la distribution Q_θ .

Puisque nous voulons que $P^C = Q_\theta^C$, nous considérons la fonction de perte :

$$ED^2(P^C, Q_\theta^C) = 2\mathbb{E}\|\mathbf{U} - \mathbf{V}\| - \mathbb{E}\|\mathbf{U} - \mathbf{U}'\| - \mathbb{E}\|\mathbf{V} - \mathbf{V}'\|.$$

Ici, \mathbf{U} , \mathbf{U}' et \mathbf{V} , \mathbf{V}' sont des copies indépendantes d'un vecteur aléatoire ayant la distribution P^C et Q_θ^C , respectivement. Nous avons $ED^2(P^C, Q_\theta^C) = 0$ si et seulement si $P^C = Q_\theta^C$.

En pratique la différenciation n'est pas immédiate et le calcul du gradient n'est pas simple. Nous devons remplacer les fonction de répartitions par des approximations différenciables. Nous invitons le lecteur intéressé par le détail à se plonger dans le papier d'IGC.

4.2.2 CODINE : le modèle

[27] démontrent que si $\pi_U(\mathbf{u})$ représente une distribution uniforme multivariée sur l'hypercube unité $[0, 1]^d$, et

$$\mathcal{J}_f(T) = \mathbb{E}_{\mathbf{u} \sim c_U(\mathbf{u})} [T(\mathbf{u})] - \mathbb{E}_{\mathbf{u} \sim \pi_U(\mathbf{u})} [\exp((T(\mathbf{u})) - 1))], \quad (4.3)$$

alors

$$c_U(\mathbf{u}) = \exp((\hat{T}(\mathbf{u})) - 1),$$

où

$$\hat{T}(\mathbf{u}) = \arg \max_T \mathcal{J}_f(T).$$

Pour maximiser la fonction objectif dans l'équation 4.3, nous choisissons d'approximer la fonction T à l'aide d'un réseau neuronal¹.

4.2.3 MICE : notre modèle

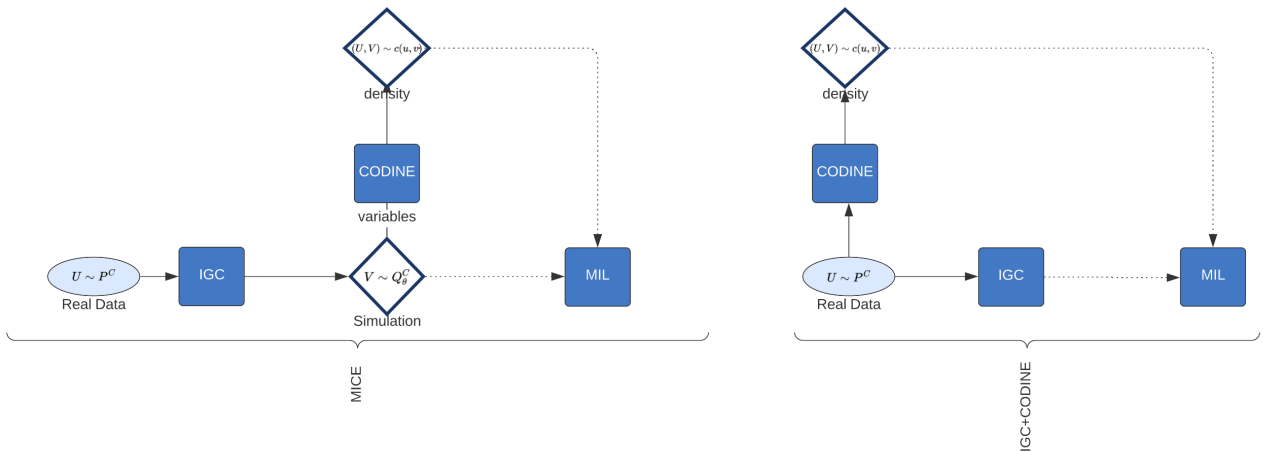


FIGURE 4.3 – MICE vs IGC et CODINE.

Contrairement à l'approche adoptée dans l'article original [27], qui implique un entraînement direct sur des observations de données, nous cherchons à entraîner notre modèle en utilisant des extraits d'échantillons issus de IGC.

Notre objectif est de capturer implicitement la distribution sous-jacente en utilisant un ensemble de données infini, ce qui peut être particulièrement avantageux lorsque l'on travaille avec des données limitées. Par conséquent, nous apprenons la densité implicite apprise par IGC si nous disposons d'un réseau neuronal suffisamment complexe pour CODINE. L'interprétation est alors directe, car nous pouvons analyser le comportement de IGC pour expliquer la variation du calcul du coefficient. De plus, cette approche est avantageuse car IGC peut simuler des données, rendant l'analyse encore plus simple. Comme le montre l'équation 4.1, il est essentiel d'apprendre comment simuler des variables et leur distribution afin de calculer des indices d'information mutuelle.

1. nous avons pris un MLP dans le mémoire

Validation et interprétation Notre modèle, comme illustré à la Figure 4.3, est facile à valider. Pour déterminer si le modèle a capturé toutes les dépendances, nous comparons simplement les copules simulées avec les copules empiriques. De plus, les réalisations et les densités de copules sont utiles pour évaluer le niveau de convergence atteint par le modèle. Notre modèle, basé sur la structure des copules, présente un avantage significatif par rapport à l’approche des indices de Sobol en termes de capacité à capturer les dépendances globales. Les indices de Sobol fournissent des mesures de sensibilité, indiquant la contribution de chaque variable d’entrée à la variabilité de la sortie. Cependant, cette approche ne fournit pas d’informations sur la structure de dépendance globale, qui est essentielle pour capturer avec précision le comportement global du système. En revanche, la structure des copules offre une vision plus globale des dépendances entre les variables d’entrée, ce qui est une force de notre modèle. En exploitant cette information, notre modèle peut mieux capturer les relations sous-jacentes entre les variables d’entrée et la sortie.

4.3 Applications de MICE

Dans ce paragraphe, nous considérons $(\Omega, \mathcal{F}, \mathbb{P})$ comme un espace probabilisé. Nous considérons une fonction :

$$f : \mathbb{R}^n \rightarrow \mathbb{R} \quad (4.4)$$

Ici, f est une fonction qui prend n variables d’entrée dans \mathbb{R}^n et renvoie une sortie réelle. f peut représenter divers objets tels qu’un système physique, une boîte noire ou toute autre fonction imaginable. Nous ne faisons aucune hypothèse sur f . Dans notre problème d’analyse de sensibilité, nous observons les entrées (X_1, \dots, X_n) et la sortie correspondante $f(X_1, \dots, X_n)$. En général, nous ne pouvons pas choisir les entrées que nous observons, et nous les modélisons comme des variables aléatoires. Il existe divers problèmes du monde réel qui impliquent une analyse de sensibilité, tels que l’analyse de la qualité de l’eau [52], les réacteurs nucléaires [42], ou les modèles chimiques [45].

Un cas classique d’application en analyse de sensibilité : Apprentissage des copules bivariées Dans cet exemple, nous allons réaliser une analyse de sensibilité de X_1 sur $f(X_1, \dots, X_n)$. Pour simplifier, supposons que nous observons le couple $(X_1, f(X_1, \dots, X_n))$. De plus, supposons que la distribution conjointe de ce couple est caractérisée par une copule connue. Plus précisément, nous supposons que $(X_1, f(X_1, \dots, X_n))$ est successivement décrit par plusieurs copules, et notre objectif est d’apprendre ces copules [24] en utilisant MICE. Ce faisant, nous pouvons mieux comprendre les limites de MICE dans la modélisation des copules complexes et utiliser ces informations pour affiner les méthodes traditionnelles d’analyse de sensibilité. Nous considérons :

1. **Clayton** $C_\theta(u, v) = (u^{-\theta} + v^{-\theta} - 1)^{-1/\theta}$ où $\theta \in [1, \infty)$,
2. **Gumbel** $C(u, v) = e^{-((-\ln u)^\theta + (-\ln v)^\theta)^{1/\theta}}$ où $\theta \in [1, \infty)$,
3. **Indépendance**, $C(u, v) = uv$,

Dans l’illustration 4.4, nous avons tracé la densité apprise par MICE sur 1000 observations du couple $(X_1, f(X_1, \dots, X_n))$ pour les copules de Gumbel et de Clayton avec $\theta = 2$. De plus, nous avons inclus une représentation de l’erreur entre la densité apprise et la densité théorique. Nous observons que MICE est capable d’approximer avec précision la densité dans les régions où elle n’est pas trop élevée. Cependant, en raison de l’architecture de notre réseau de neurones, nous rencontrons des difficultés pour estimer les régions de haute densité. Il s’agit d’un

point important à prendre en compte, car cela peut avoir un impact direct sur l'information mutuelle et conduire à une sous-estimation. Pour des copules telles que les copules indépendantes ou gaussiennes, qui ont moins de régions de haute densité, MICE fournit une très bonne approximation.

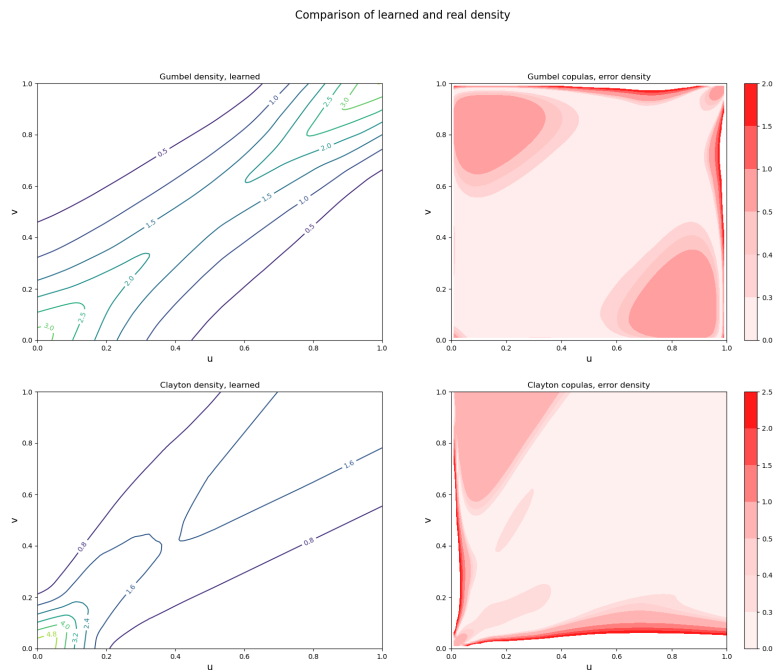


FIGURE 4.4 – Erreurs absolues.

Nous avons également tracé dans 4.5 l'erreur moyenne d'approximation pour différentes valeurs de quantiles de X_1 et $Y = f(X_1, \dots, X_n)$. Comme attendu, nous observons que l'erreur augmente significativement lorsque la densité est élevée, comme pour des valeurs de quantiles très petites ou très grandes. Cela est dû aux difficultés à estimer avec précision les régions de haute densité avec MICE, comme mentionné précédemment. Par conséquent, lors de l'analyse de sensibilité avec MICE, il est important de prendre en compte la sous-estimation potentielle de l'information mutuelle dans les régions de haute densité.

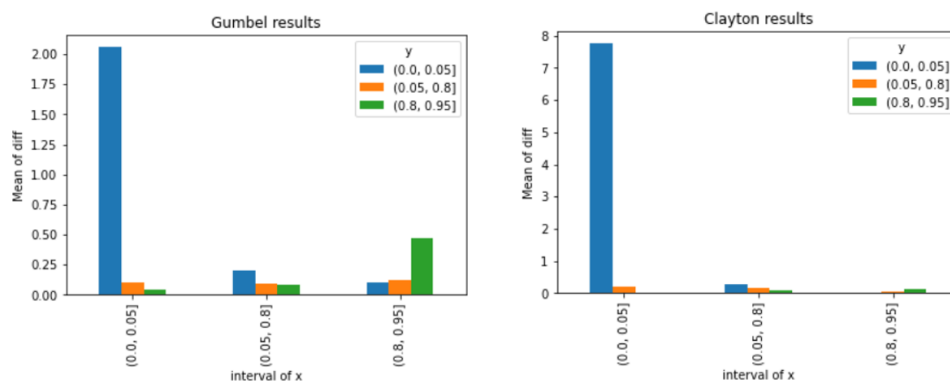


FIGURE 4.5 – Erreur moyenne d'approximation pour différentes valeurs de quantiles.

Les réseaux de neurones peuvent avoir du mal à approximer la région où la densité des variables aléatoires bivariées est très élevée.

Nous tenons également à souligner les avantages de former CODINE sur des échantillons extraits d'IGC, plutôt que directement sur les données. Tout d'abord, lorsque nous travaillons avec peu de données, apprendre la distribution implicitement apprise par IGC réduit le problème en se concentrant uniquement sur IGC. De plus, comme le montre le tableau ci-dessous, former CODINE sur les sorties d'IGC améliore sa précision. Nous calculons l'erreur absolue moyenne entre la densité prédite \hat{c} par MINE et la densité réelle de Gumbel c de paramètre $\theta = 2$ pour deux cas différents. Dans le cas A, nous calculons l'erreur sur toutes les valeurs de densité prédites $E_{U,V \sim u(0,1)}[|c(U,V) - \hat{c}(U,V)|]$. Dans le cas B, nous calculons l'erreur $E_{U,V \sim u(0.2,0.8)}[|c(U,V) - \hat{c}(U,V)|]$:

Nombre d'observations	MICE A	CODINE A	MICE B	CODINE B
50.0	0.308342	0.381140	0.152919	0.372463
100.0	0.340987	0.295519	0.243545	0.196574
150.0	0.241253	0.360648	0.172606	0.273210
200.0	0.208672	0.244702	0.135392	0.200315
400.0	0.245526	0.332591	0.137816	0.276372
2000.0	0.204087	0.236324	0.124744	0.164436
4000.0	0.204138	0.241717	0.127119	0.148564

Nous avons observé que pour un nombre fixe d'observations $n = 150$, l'erreur absolue moyenne d'estimation a une variance de **0,001220** pour MICE, tandis qu'elle est de **0,001956** pour CODINE. Cela indique une réduction de la variance d'estimation lors de l'utilisation de MICE.

Sur la base des résultats présentés dans les sections précédentes, plusieurs conclusions peuvent être tirées. Tout d'abord, MICE semble être plus précis que CODINE dans l'estimation de la densité d'une copule dans un contexte de faible volume de données. Cela est appuyé par le fait que l'erreur absolue moyenne d'estimation est plus faible pour MICE par rapport à CODINE. De plus, la formation de CODINE sur des échantillons extraits d'IGC, plutôt que de le former directement sur les données, contribue à accroître sa précision. Cela est démontré par le fait que l'erreur absolue moyenne d'estimation est plus faible lorsque CODINE est formé sur les sorties d'IGC par rapport à lorsqu'il est formé directement sur les données.

Cependant, il est important de noter que MICE a des difficultés à estimer les densités dans les régions où elles sont très élevées. Cela est particulièrement pertinent lorsqu'il existe une forte corrélation entre deux variables aléatoires dans la queue. Il est donc important de garder à l'esprit cette limitation lors de l'interprétation des résultats des calculs du coefficient d'information mutuelle.

Dans l'ensemble, MICE semble être un outil précieux pour interpréter les calculs du coefficient d'information mutuelle. Sa capacité à estimer avec précision la densité d'une copule dans un contexte de faible volume de données, ainsi que sa capacité potentielle à réduire la variance d'estimation, en font un outil prometteur pour les futures recherches dans le domaine de l'analyse de sensibilité.

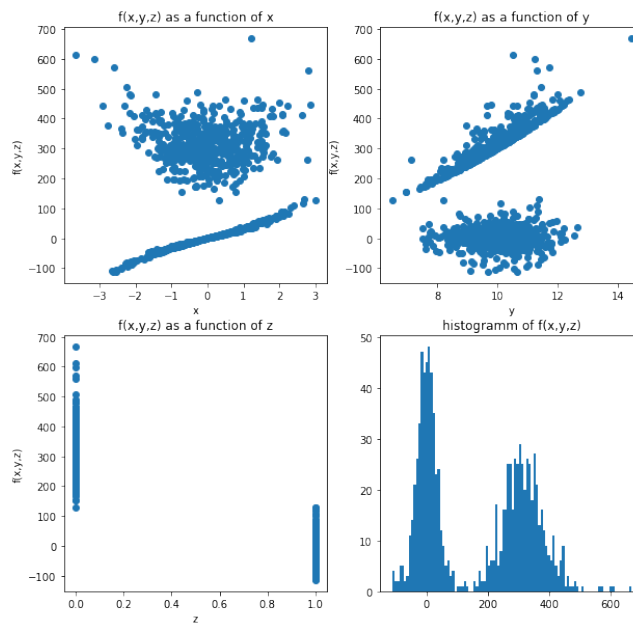


FIGURE 4.6 – Exemple fil rouge.

Exemple fil rouge

		X	Y	Nombre d'observations
ρ_{MI}	Sobol	0.06	0.04	12000
	MICE	0.86	0.73	1000
	MINE	0.94	0.53	1000
	K-NN	0.97	0.87	1000

Nous observons que l'indice se calcul correctement, comme prévu. Mais en plus, nous avons la possibilité de montrer physiquement la construction de l'indice de sensibilité et de l'interpréter. Selon l'analyse réalisée avec MICE, il existe une forte relation entre les valeurs de X et f , ainsi qu'entre les valeurs de Y et f . Cependant, les indices de Sobol ne parviennent pas à identifier cette relation et déclarent que X et Y n'ont aucun impact sur f . Nous avons également observé que MICE sous-estime le coefficient d'information mutuelle. Cela peut s'expliquer par le fait que le réseau neuronal a du mal à séparer différentes régions de l'espace, comme le montre la Figure 4.7.

Sensitive analysis of f

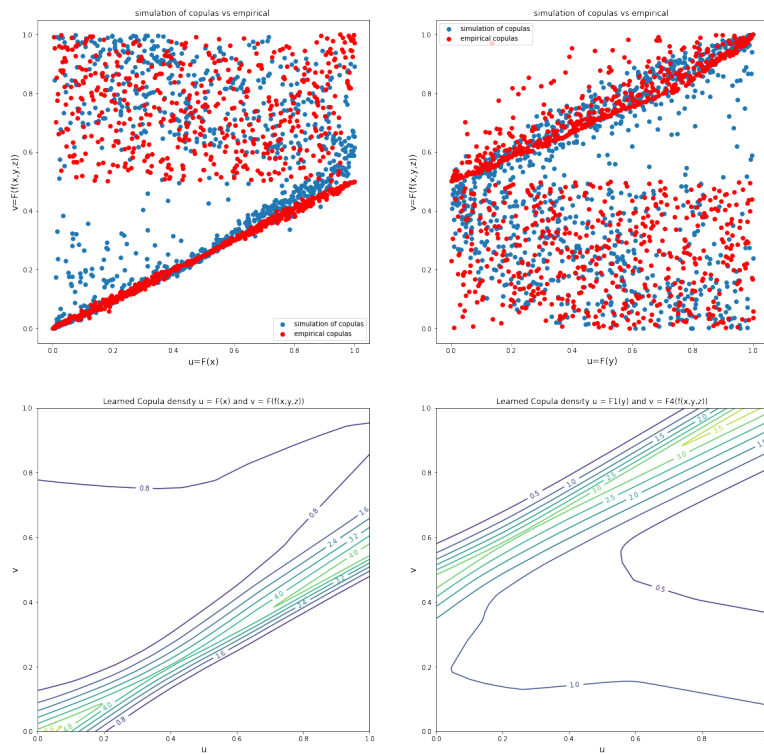


FIGURE 4.7 – Notre modèle a réussi à apprendre les copules et les niveaux de densité.

Conclusion Désormais nous sommes capables de mener une analyse de sensibilité lorsque les variables ont une distribution multimodale, et nous avons des modèles capables d'estimer l'information mutuelle, qui soient interprétables physiquement. La première question, à savoir *comment prendre en considération différentes données de marché et quantifier l'impact des paramètres selon les données de marché* reste en suspens. Dans la partie qui suit, nous proposons des méthodes un peu plus avancées afin de directement classifier les données de marché.

Troisième partie

Prise en compte de la dynamique des données financières

Chapitre 5

Pré-requis en machine learning

“The key to artificial intelligence has always been the representation.” — Jeff Hawkins

La classification et la génération de données de marché peuvent soulever différents problèmes. En effet, les données sont de grandes dimensions et peuvent parfois contenir des relations non linéaires, ce qui rend l'utilisation de techniques classiques difficile et peu pertinente. Récemment, de nombreuses méthodes ont vu le jour, s'appuyant sur la capacité des réseaux neuronaux à apprendre une représentation des données d'entrée afin de réaliser la classification et la synthèse de données. Pour introduire certaines de ces techniques, il est indispensable de présenter quelques éléments supplémentaires en apprentissage automatique, notamment les réseaux de neurones récurrents et leurs extensions.

5.1 Les réseaux de neurones récurrents (RNN)

5.1.1 Définitions et backpropagation through time

Les méthodes de l'état de l'art en **modélisation du langage**, en **génération de texte** ou encore en **reconnaissance audio** utilisent largement les réseaux de neurones récurrents (RNN). Les RNN sont usuellement utilisés pour détecter des patterns dans des séquences de données. Nous nous en servons notamment pour **extraire des faits stylisés** des nappes de volatilité de swaption. La différence entre les RNN et les MLP est la manière dont l'information passe à travers le système. Quand pour le MLP l'information passe sans aucun cycle, dans le cas du RNN, il y a des cycles et le réseau utilise de l'information ancienne en plus de l'information présentée. Pour formaliser mathématiquement le processus de passage de l'information à travers un tel réseau de neurones, nous reprenons le formalisme de [60]. Les réseaux de neurones récurrents utilisent une mémoire pour retenir de l'information et s'en servent pour prédire ses réponses. La définition suivante fixe ces différentes notions.

Definition 5.1.1. (*Réseau de neurones récurrent*) Nous notons par $\mathbf{H}_t \in \mathbb{R}^{n \times h}$ l'état caché du RNN à l'instant t , $\mathbf{X}_t \in \mathbb{R}^{n \times d}$ l'input donné au réseau de neurones récurrent, où n est le nombre de samples, d est le nombre d'input dans chaque sample et h est le nombre d'unités cachés. Nous désignons par $\mathbf{W}_{xh} \in \mathbb{R}^{d \times h}$ la matrice de poids, et $\mathbf{W}_{hh} \in \mathbb{R}^{h \times h}$ la matrice de transition de l'état caché vers l'état caché. Le biais est noté $\mathbf{b}_h \in \mathbb{R}^{1 \times h}$. ϕ désigne la fonction d'activation. Dans un RNN, l'état caché évolue selon l'équation :

$$\mathbf{H}_t = \phi_h (\mathbf{X}_t \mathbf{W}_{xh} + \mathbf{H}_{t-1} \mathbf{W}_{hh} + \mathbf{b}_h),$$

la sortie évolue comme :

$$\mathbf{O}_t = \phi_o (\mathbf{H}_t \mathbf{W}_{ho} + \mathbf{b}_o),$$

le réseau de neurones récurrent utilise donc un état caché faisant office de mémoire pour prédire la sortie suivante.

Pour entraîner un réseau de neurones récurrent, il faut utiliser l'algorithme de *backpropagation through time* (BPTT) qui est une adaptation de l'algorithme d'apprentissage des MLP classiques.

Pour cela, nous pouvons définir une fonction de perte $\mathcal{L}(\mathbf{O}, \mathbf{Y})$:

$$\mathcal{L}(\mathbf{O}, \mathbf{Y}) = \sum_{t=1}^T \ell_t(\mathbf{O}_t, \mathbf{Y}_t),$$

où \mathbf{O}_t est la sortie du réseau de neurone récurrent à l'instant t et \mathbf{Y}_t est la valeur cible. Etant donné que nous avons trois matrices de poids \mathbf{W}_{xh} , \mathbf{W}_{hh} et \mathbf{W}_{ho} , il est possible de dériver la fonction de perte par rapport à ces différents poids, en utilisant le calcul matriciel et la règle de la chaîne :

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}_{ho}} = \sum_{t=1}^T \frac{\partial \ell_t}{\partial \mathbf{O}_t} \cdot \frac{\partial \mathbf{O}_t}{\partial \phi_o} \cdot \frac{\partial \phi_o}{\mathbf{W}_{ho}} = \sum_{t=1}^T \frac{\partial \ell_t}{\partial \mathbf{O}_t} \cdot \frac{\partial \mathbf{O}_t}{\partial \phi_o} \cdot \mathbf{H}_t,$$

et :

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}_{hh}} = \sum_{t=1}^T \frac{\partial \ell_t}{\partial \mathbf{O}_t} \cdot \frac{\partial \mathbf{O}_t}{\partial \phi_o} \cdot \frac{\partial \phi_o}{\partial \mathbf{H}_t} \cdot \frac{\partial \mathbf{H}_t}{\partial \phi_h} \cdot \frac{\partial \phi_h}{\partial \mathbf{W}_{hh}} = \sum_{t=1}^T \frac{\partial \ell_t}{\partial \mathbf{O}_t} \cdot \frac{\partial \mathbf{O}_t}{\partial \phi_o} \cdot \mathbf{W}_{ho} \cdot \frac{\partial \mathbf{H}_t}{\partial \phi_h} \cdot \frac{\partial \phi_h}{\partial \mathbf{W}_{hh}},$$

enfin :

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}_{xh}} = \sum_{t=1}^T \frac{\partial \ell_t}{\partial \mathbf{O}_t} \cdot \frac{\partial \mathbf{O}_t}{\partial \phi_o} \cdot \frac{\partial \phi_o}{\partial \mathbf{H}_t} \cdot \frac{\partial \mathbf{H}_t}{\partial \phi_h} \cdot \frac{\partial \phi_h}{\partial \mathbf{W}_{xh}} = \sum_{t=1}^T \frac{\partial \ell_t}{\partial \mathbf{O}_t} \cdot \frac{\partial \mathbf{O}_t}{\partial \phi_o} \cdot \mathbf{W}_{ho} \cdot \frac{\partial \mathbf{H}_t}{\partial \phi_h} \cdot \frac{\partial \phi_h}{\partial \mathbf{W}_{xh}}.$$

Il s'agit de remarquer que \mathbf{H}_t dépend des étapes précédentes puisqu'il est un état caché. Nous pouvons alors réécrire les équations comme :

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \mathbf{W}_{hh}} &= \sum_{t=1}^T \frac{\partial \ell_t}{\partial \mathbf{O}_t} \cdot \frac{\partial \mathbf{O}_t}{\partial \phi_o} \cdot \mathbf{W}_{ho} \sum_{k=1}^t \frac{\partial \mathbf{H}_t}{\partial \mathbf{H}_k} \cdot \frac{\partial \mathbf{H}_k}{\partial \mathbf{W}_{hh}}, \\ \frac{\partial \mathcal{L}}{\partial \mathbf{W}_{xh}} &= \sum_{t=1}^T \frac{\partial \ell_t}{\partial \mathbf{O}_t} \cdot \frac{\partial \mathbf{O}_t}{\partial \phi_o} \cdot \mathbf{W}_{ho} \sum_{k=1}^t \frac{\partial \mathbf{H}_t}{\partial \mathbf{H}_k} \cdot \frac{\partial \mathbf{H}_k}{\partial \mathbf{W}_{xh}}, \end{aligned}$$

enfin, et c'est la dernière étape, nous pouvons réécrire :

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \mathbf{W}_{hh}} &= \sum_{t=1}^T \frac{\partial \ell_t}{\partial \mathbf{O}_t} \cdot \frac{\partial \mathbf{O}_t}{\partial \phi_o} \cdot \mathbf{W}_{ho} \sum_{k=1}^t (\mathbf{W}_{hh}^\top)^{t-k} \cdot \mathbf{H}_k, \\ \frac{\partial \mathcal{L}}{\partial \mathbf{W}_{xh}} &= \sum_{t=1}^T \frac{\partial \ell_t}{\partial \mathbf{O}_t} \cdot \frac{\partial \mathbf{O}_t}{\partial \phi_o} \cdot \mathbf{W}_{ho} \sum_{k=1}^t (\mathbf{W}_{hh}^\top)^{t-k} \cdot \mathbf{X}_k. \end{aligned}$$

Il est alors possible de mettre à jour les poids d'un réseau de neurones récurrents, en dérivant la fonction d'erreur par rapport aux poids du RNN. En pratique, comme nous sommes sur tous les instants t , l'erreur peut devenir très grande pour de très larges séquences. Une solution est d'utiliser l'algorithme BPTT tronqué [58].

5.1.2 Le problème des RNN : l'explosion ou la diminution du gradient

Nous voyons dans les équations précédentes que la multiplication par le terme $\frac{\partial \mathbf{H}_t}{\partial \mathbf{H}_k}$ sur des séquences potentiellement très longues peut poser des gros problèmes, notamment si le terme est plus petit que 1 (le gradient finit par être nul) ou si le terme est plus grand que 1 (le gradient finit par exploser) [8]. Ce problème rendait inutilisable les réseaux de neurones récurrents en pratique. Cela a motivé l'introduction des réseaux de type *Long Short-Term Memory Units* (LSTM), que nous avons utilisé dans le mémoire.

5.1.3 Long Short-Term Memory Units (LSTM)

Introduits en 1997, [15], les LSTM ont été proposés pour surmonter le problème de l'explosion du gradient. Les LSTM peuvent, grâce à leur architecture, stocker beaucoup plus d'informations et traiter des séquences beaucoup plus grandes que celles que nous avons avec un RNN basique.

Definition 5.1.2. (*LSTM*) Nous notons \mathbf{O}_t une porte de sortie du LSTM, \mathbf{I}_t la porte d'entrée pour lire le data et \mathbf{F}_t la porte d'oublie qui sert à effacer du contenu de la mémoire. Le LSTM contient aussi un état caché, \mathbf{H}_t . Les portes d'entrées, de sorties et d'oublie suivent les dynamiques :

$$\begin{aligned}\mathbf{O}_t &= \sigma(\mathbf{X}_t \mathbf{W}_{xo} + \mathbf{H}_{t-1} \mathbf{W}_{ho} + \mathbf{b}_o), \\ \mathbf{I}_t &= \sigma(\mathbf{X}_t \mathbf{W}_{xi} + \mathbf{H}_{t-1} \mathbf{W}_{hi} + \mathbf{b}_i), \\ \mathbf{F}_t &= \sigma(\mathbf{X}_t \mathbf{W}_{xf} + \mathbf{H}_{t-1} \mathbf{W}_{hf} + \mathbf{b}_f),\end{aligned}$$

avec $\mathbf{W}_{xi}, \mathbf{W}_{xf}, \mathbf{W}_{xo} \in \mathbb{R}^{d \times h}$ et $\mathbf{W}_{hi}, \mathbf{W}_{hf}, \mathbf{W}_{ho} \in \mathbb{R}^{h \times h}$ les différents poids qu'il faut mettre à jour, et $\mathbf{b}_i, \mathbf{b}_f, \mathbf{b}_o \in \mathbb{R}^{1 \times h}$ les biais respectifs. σ désigne la fonction d'activation sigmoïde. Puisque ces réseaux de neurones ont une mémoire, nous la notons $\tilde{\mathbf{C}}_t \in \mathbb{R}^{n \times h}$. En notant ses poids et biais $\mathbf{W}_{xc} \in \mathbb{R}^{d \times h}$, $\mathbf{W}_{hc} \in \mathbb{R}^{h \times h}$, $\mathbf{b}_c \in \mathbb{R}^{1 \times h}$, sa dynamique est donnée par :

$$\tilde{\mathbf{C}}_t = \tanh(\mathbf{X}_t \mathbf{W}_{xc} + \mathbf{H}_{t-1} \mathbf{W}_{hc} + \mathbf{b}_c),$$

pour que tout fonctionne bien, nous notons par $\mathbf{C}_{t-1} \in \mathbb{R}^{n \times h}$ l'ancienne mémoire du réseau de neurones. \odot désignant un produit élément-par-élément, celle-ci évolue comme :

$$\mathbf{C}_t = \mathbf{F}_t \odot \mathbf{C}_{t-1} + \mathbf{I}_t \odot \tilde{\mathbf{C}}_t,$$

enfin, l'état caché $\mathbf{H}_t \in \mathbb{R}^{n \times h}$ est mis à jour comme :

$$\mathbf{H}_t = \mathbf{O}_t \odot \tanh(\mathbf{C}_t),$$

ce qui nous donne l'évolution de chaque porte.

Il s'agit simplement d'un RNN auquel nous avons ajouté une mémoire en plus de l'état caché, utilisable par le réseau, et dont il peut décider de modifier des éléments. En réalité, nous nous rapprochons un peu plus du fonctionnement du cerveau humain sur cette catégorie de modèles. Il est possible de montrer comment mettre à jour les poids, les calculs étant similaires à ceux que l'on a présenté la section précédente. La figure 5.1, que nous avons prise directement dans [46], permet de mieux comprendre les connexions entre les différentes portes définies plus haut.

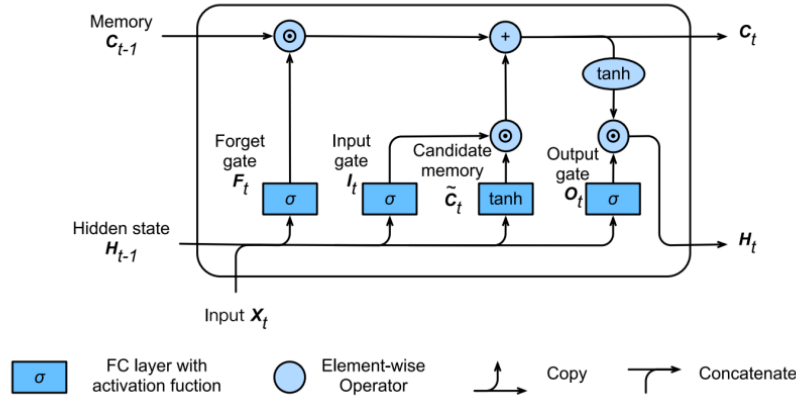


FIGURE 5.1 – Illustration d’une cellule LSTM, illustration prise dans [46]. Nous retrouvons les différentes *portes*, d’entrée, d’oublie et de sortie. Ce réseau de neurone a une mémoire qu’il utilise pour produire des sorties.

[46] montre que empiriquement les performances des LSTM sont bien meilleur que celles des premiers types de RNN. Dans la suite, nous utiliserons les LSTM comme une brique du modèle que nous allons utiliser pour classifier les données de marché.

5.2 Modèles génératifs, un focus sur les autoencodeurs variationnels

Une des grandes divisions en machine learning est la séparation entre les modèles discriminatifs et les modèles génératifs. D’un côté, les modèles discriminatifs essayent d’apprendre des **prédicteurs** étant donné un ensemble d’informations, quand les modèles génératifs essayent de résoudre des problématiques plus générales, en apprenant la distribution jointe d’un ensemble de variables. Un modèle **génératif** est capable de simuler comment des données sont générées dans le monde réel. Pour de nombreuses raisons, nous avons décidé d’utiliser une classe de modèle génératifs pour générer des données de marché. Nous avons été notamment séduit par l’interprétabilité que peut offrir une telle variété de modèles.

Plusieurs questions se posent :

- Pouvons-nous, si nous avons deux nappes de volatilités de swaptions les identifier ?
- Pouvons nous anticiper la valeur du BE et de la VIF en ayant une nappe de volatilité donnée ?

Répondre à ces questions revient à trouver un espace de dimension plus faible, par exemple 2, où nous pouvons représenter une nappe de volatilité précise par un point de cet espace.

5.2.1 Limitation des méthodes traditionnelles

L’analyse par composante principale (ACP), qui sert habituellement de méthode pour réduire la dimension d’un espace ne peut pas fonctionner correctement pour ce problème. En effet, cette dernière n’est capable de capter que des dépendances linéaires, alors que nos données contiennent de nombreuses dépendances non linéaires. De plus, l’ACP ne permet pas de générer de nouvelles données.

5.2.2 L'autoencodeur variationnel bayésien

Dans cette partie, nous nous proposons de montrer comment nous avons construit un Autoencodeur Variationnel Bayésien (VAE) afin de généraliser l'ACP. Nous considérons un ensemble $\mathcal{X} = \{\mathbf{x}^1, \dots, \mathbf{x}^N\}$ de N échantillons i.i.d. d'un vecteur aléatoire continu (par exemple une nappe de volatilité de swaptions). Nous supposons que ces données sont générées par un processus continu aléatoire \mathbf{z} . Nous faisons l'hypothèse que la génération des données suit donc un cheminement bien précis :

1. Une valeur $\mathbf{z}^{(i)}$ est générée depuis la distribution prior centrée isotrope gaussienne multivariée $p_{\theta^*}(\mathbf{z}) = N(\mathbf{z}; 0, \mathbf{I})$,
2. Une valeur de $\mathbf{x}^{(i)}$ est générée depuis la distribution conditionnelle gaussienne multivariée $p_{\theta^*}(\mathbf{x}|\mathbf{z})$, qui nous est inconnue.

Nous ne faisons pas d'hypothèse sur les probabilités marginales ou postérieures. Nous ne pouvons pas estimer $p_{\theta}(\mathbf{x})$ avec des algorithmes classiques comme EM (intégrale incalculable).

Nous introduisons un modèle de reconnaissance $q_{\phi}(\mathbf{z} | \mathbf{x})$ qui devra approximer la densité $p_{\theta^*}(\mathbf{z}|\mathbf{x})$. Nous appelons \mathbf{z} les variables "latentes", ou les "codes". Finalement, nous allons appeler $q_{\phi}(\mathbf{z} | \mathbf{x})$, un "encodeur", et $p_{\theta}(\mathbf{x}|\mathbf{z})$ un "decodeur". Nous remarquons que $p_{\theta^*}(\mathbf{z})$ et $p_{\theta^*}(\mathbf{z}|\mathbf{x})$ sont dérivables presque partout en θ et en \mathbf{z} .

Il est important de noter que le modèle $q_{\phi}(\mathbf{z} | \mathbf{x})$ qui devra approximer la densité $p_{\theta^*}(\mathbf{z}|\mathbf{x})$ peut théoriquement appartenir à une grande classe de densité. Nous faisons donc l'hypothèse qu'elle est approximativement gaussienne avec une matrice de covariance diagonale¹. Nous avons donc :

$$\log q_{\phi}(\mathbf{z} | \mathbf{x}^{(i)}) = \log \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}^{(i)}, \boldsymbol{\sigma}^{2(i)}\mathbf{I}),$$

avec $\boldsymbol{\mu}^{(i)}$ et $\boldsymbol{\sigma}^{2(i)}$ qui sont en pratique approximés par des réseaux de neurones.

En effet, les réseaux de neurones sont utilisés comme encodeurs et décodeurs probabilistes. Dans notre étude, l'encodeur est un RNN et le décodeur un MLP (multi-layer perceptron). Notre décodeur est donc donné explicitement par :

$$\begin{aligned} \log p(\mathbf{x} | \mathbf{z}) &= \log \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2\mathbf{I}), \\ \text{où } \boldsymbol{\mu} &= \mathbf{W}_4\mathbf{h} + \mathbf{b}_4, \\ \log \boldsymbol{\sigma}^2 &= \mathbf{W}_5\mathbf{h} + \mathbf{b}_5, \\ \mathbf{h} &= \tanh(\mathbf{W}_3\mathbf{z} + \mathbf{b}_3), \end{aligned}$$

avec $\{\mathbf{W}_3, \mathbf{W}_4, \mathbf{W}_5, \mathbf{b}_3, \mathbf{b}_4, \mathbf{b}_5\}$ les paramètres du réseau de neurones, que l'on représente par θ .

Pour notre encodeur, nous avons fait le choix de combiner une couche LSTM et une couche linéaire. Le choix de la couche LSTM repose sur le fait qu'elle généralise les processus auto-regressifs qui semblent apparaître de manière naturelle lorsque nous considérons, comme c'est notre cas ici, des nappes de volatilité.

1. Cette hypothèse est classique dans la littérature et s'est montrée relativement robuste sur des tests empiriques

$$\begin{aligned}
 \log q_\phi(\mathbf{z} | \mathbf{x}) &= \log \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2 \mathbf{I}), \\
 \text{où } \boldsymbol{\mu} &= \mathbf{W}_1 \mathbf{h} + \mathbf{b}_1, \\
 \log \boldsymbol{\sigma}^2 &= \mathbf{W}_2 \mathbf{h} + \mathbf{b}_2, \\
 \mathbf{h} &= \{h_t\}_{t \in [0,90]}, \\
 i_t &= \sigma(W_{ii}x_t + b_{ii} + W_{hi}h_{t-1} + b_{hi}), \\
 f_t &= \sigma(W_{if}x_t + b_{if} + W_{hf}h_{t-1} + b_{hf}), \\
 g_t &= \tanh(W_{ig}x_t + b_{ig} + W_{hg}h_{t-1} + b_{hg}), \\
 o_t &= \sigma(W_{io}x_t + b_{io} + W_{ho}h_{t-1} + b_{ho}), \\
 c_t &= f_t \odot c_{t-1} + i_t \odot g_t, \\
 h_t &= o_t \odot \tanh(c_t),
 \end{aligned}$$

avec ϕ désignant de manière analogue les différents poids du réseau de neurones.

Nous avons donc défini nos fonctions non linéaires, reste à trouver un moyen d'apprendre les poids ϕ et θ de manière conjointe pour pouvoir approximer nos fonctions de densité d'intérêt.

Il vient :

$$\log p_\theta(\mathbf{x}^{(i)}) = D_{KL}(q_\phi(\mathbf{z} | \mathbf{x}^{(i)}) \| p_\theta(\mathbf{z} | \mathbf{x}^{(i)})) + \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}),$$

avec :

$$\begin{aligned}
 \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}) &= -D_{KL}(q_\phi(\mathbf{z} | \mathbf{x}^{(i)}) \| p_\theta(\mathbf{z})) + \\
 &\quad \mathbb{E}_{q_\phi(\mathbf{z} | \mathbf{x}^{(i)})} [\log p_\theta(\mathbf{x}^{(i)} | \mathbf{z})].
 \end{aligned}$$

Nous pouvons ensuite remarquer que nous pouvons obtenir θ avec une analogie de l'estimation par maximum de vraisemblance². L'idée est ici de mimiser l'opposé de cette fonction de perte. Nous allons vouloir différencier $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)})$ par rapport à ϕ et par rapport à θ pour pouvoir réaliser cette descente de gradient. Cependant, le terme $\mathbb{E}_{q_\phi(\mathbf{z} | \mathbf{x}^{(i)})} [\log p_\theta(\mathbf{x}^{(i)} | \mathbf{z})]$ pose un problème et ne va pas être simple à différencier. Nous pouvons lever ce problème³ en écrivant : $\mathbf{z}^{(i,l)} = g_\phi(\mathbf{x}^{(i)}, \boldsymbol{\epsilon}^{(l)}) = \boldsymbol{\mu}^{(i)} + \boldsymbol{\sigma}^{(i)} \odot \boldsymbol{\epsilon}^{(l)}$ où $\boldsymbol{\epsilon}^{(l)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

Dans le cadre de nos hypothèses nous avons aussi :

$$\begin{aligned}
 -D_{KL}(q_\phi(\mathbf{z}) \| p_\theta(\mathbf{z})) &= \int q_\theta(\mathbf{z}) (\log p_\theta(\mathbf{z}) - \log q_\theta(\mathbf{z})) d\mathbf{z} \\
 &= \frac{1}{2} \sum_{j=1}^J (1 + \log((\sigma_j)^2) - (\mu_j)^2 - (\sigma_j)^2),
 \end{aligned}$$

nous avons donc comme fonction de perte :

$$\begin{aligned}
 \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}) &\simeq \frac{1}{2} \sum_{j=1}^J \left(1 + \log \left((\sigma_j^{(i)})^2 \right. \right. \\
 &\quad \left. \left. - (\mu_j^{(i)})^2 - (\sigma_j^{(i)})^2 + \frac{1}{L} \sum_{l=1}^L \log p_\theta(\mathbf{x}^{(i)} | \mathbf{z}^{(i,l)}) \right),
 \end{aligned}$$

où $\mathbf{z}^{(i,l)} = \boldsymbol{\mu}^{(i)} + \boldsymbol{\sigma}^{(i)} \odot \boldsymbol{\epsilon}^{(l)}$ et $\boldsymbol{\epsilon}^{(l)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

2. chaque terme de ces égalités à une signification bien précis que nous pouvons expliciter en se mettant dans le domaine de la théorie de l'information

3. Nous parlons de "trick reparametrization", c'est une astuce qui permet de réécrire les termes de manière à pouvoir effectuer une descente de gradient

Dans ce cas, grâce à l'automatisation de la différenciation qui est rendue possible par PyTorch, nous sommes capables d'apprendre conjointement les paramètres θ et ϕ . La figure 5.2 illustre le passage de l'information à travers un VAE et les différentes fonctions de perte que nous calculons. En rouge, nous pouvons comparer sur la partie gauche du graphique, le cas où nous n'utilisons pas de *trick parametrization* et à droite, le cas où nous travaillons avec cette reparamétrisation. Dans le cas de gauche, lors de la descente de gradient, l'information n'arrive pas à remonter dans le réseau de neurones, et l'apprentissage ne fonctionne pas.

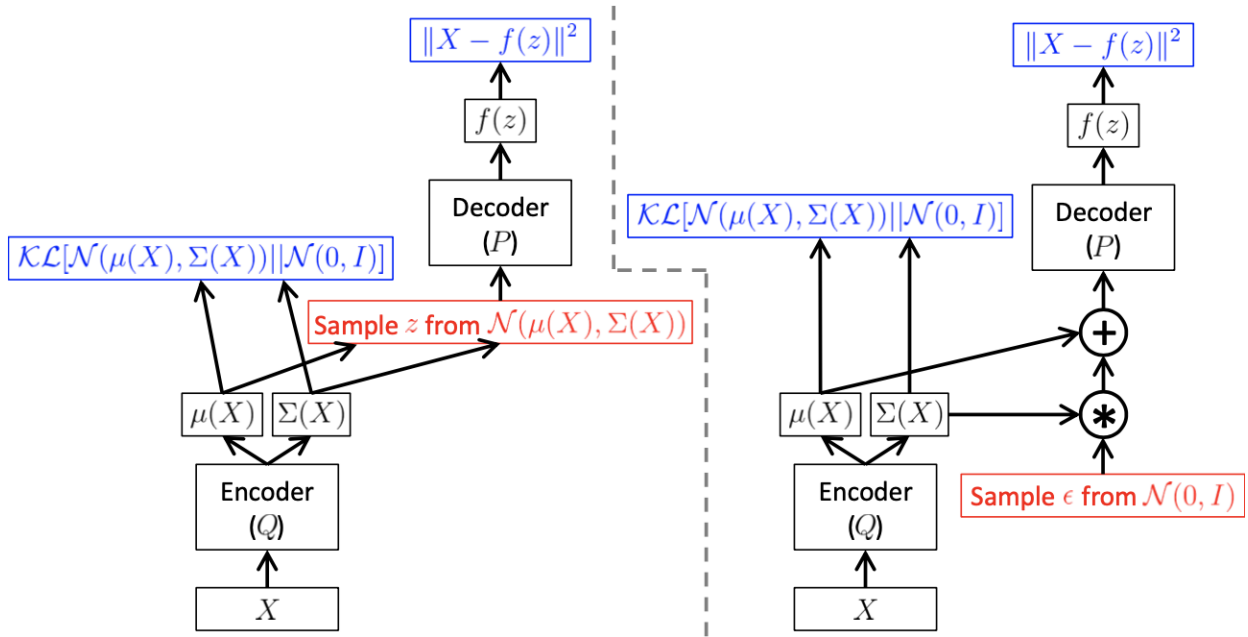


FIGURE 5.2 – Résumé de l'autoencodeur utilisé. Nous visualisons clairement l'importance de la "trick reparametrization" pour pouvoir effectuer la descente de gradient. Graphique pris dans l'article de blog [41].

Remarque Le VAE s'inscrit dans la continuité de ce que nous avons présenté dans le cadre de la théorie de l'information. Si nous reprenons la fonction de perte que nous venons de présenter, nous pouvons écrire :

$$\begin{aligned} & \log p_{\theta}(\mathbf{x}^{(i)}) - D_{KL}(q_{\phi}(\mathbf{z} | \mathbf{x}^{(i)}) || p_{\theta}(\mathbf{z} | \mathbf{x}^{(i)})) \\ &= -D_{KL}(q_{\phi}(\mathbf{z} | \mathbf{x}^{(i)}) || p_{\theta}(\mathbf{z})) + \mathbb{E}_{q_{\phi}(\mathbf{z} | \mathbf{x}^{(i)})} [\log p_{\theta}(\mathbf{x}^{(i)} | \mathbf{z})], \end{aligned}$$

il est possible de donner une interprétation *physique* de ces différents termes, permettant une meilleure compréhension du modèle.

1. $\log p_{\theta}(\mathbf{x}^{(i)})$ est le nombre de bits requis pour reconstruire $\mathbf{x}^{(i)}$ en utilisant un encodeur idéal,
2. $D_{KL}(q_{\phi}(\mathbf{z} | \mathbf{x}^{(i)}) || p_{\theta}(\mathbf{z} | \mathbf{x}^{(i)}))$ est une pénalité sur $q_{\phi}(\mathbf{z} | \mathbf{x}^{(i)})$ qui est un encodage sous-optimal,
3. Le terme de droite peut s'interpréter comme deux étapes successives qui sont d'abord l'utilisation de quelques bits pour reconstruire \mathbf{z} puis $\mathbb{E}_{q_{\phi}(\mathbf{z} | \mathbf{x}^{(i)})} [\log p_{\theta}(\mathbf{x}^{(i)} | \mathbf{z})]$ mesure le nombre de bits utilisés pour reconstruire $\mathbf{x}^{(i)}$ à partir de \mathbf{z} , si nous utilisons un encodeur idéal.

Remarque Nous montrons que si nous utilisons des MLP comme encodeurs et décodeurs, et que nous utilisons des fonctions d'activations non-linéaires (autoencodeur dit dégénéré), nous obtenons des résultats identiques à l'ACP. Le tableau suivant résume les points clés des méthodes dont nous venons de discuter.

ACP	Autoencodeur dégénéré	Autoencodeur
Relation linéaires	Relations linéaires	Relations non linéaires
Axes de projection orthogonaux	Axes de projection corrélés	Axes de projection corrélés
Interprétation simple	Interprétation par exploration	Interprétation par exploration
Aucune interpolation	Interpolation	Grande capacité d'interpolation

Nous laissons aussi quelques remarques supplémentaires permettant au lecteur de comprendre les extensions que nous pouvons apporter au modèle.

1. Nous pouvons rajouter dans la fonction de perte un terme forçant les axes de projections à être orthogonaux,
2. L'espace latent contient les informations «condensées», et peut servir features pour d'autres buts,
3. Historiquement les autoencodeurs sont des modèles « génératifs », capables de produire des données de synthèse.

Chapitre 6

Classification des données de marché avec un VAE

Les données de marché qui susciteront notre intérêt sont les données liées aux taux et aux actions. Les volatilités des swaptions et les courbes des taux sont utilisées pour l'étalonnage des modèles de taux, tandis que les volatilités des options CALL sont utilisées pour l'étalonnage des modèles d'actions. Dans ce chapitre, notre objectif est de classifier les données de marché et d'estimer la valeur probable de la VIF et du BE pour des scénarios donnés. Nous souhaitons également utiliser notre cadre de travail pour déterminer quelles données de marché sont les plus importantes et ont le plus d'impact sur les résultats.

Nous présenterons successivement la classification des données de volatilité d'options sur Call à l'aide d'un VAE, la classification des données de volatilité de swaptions, puis l'étude des effets combinés de ces deux types de données.

6.1 Volatilité des options Call

Allure des données utilisées Une grande classe de données de marché à laquelle nous sommes confrontés est celle des volatilités d'options sur Call. Nous en avons extrait une centaine de différentes à l'aide de *Refinitiv* (les données sont mensuelles, l'historique va de 2016 à 2023). Pour chaque date, nous disposons de maturités allant de 1 an à 20 ans d'une option Call à la monnaie. Concrètement, nous pouvons afficher un exemple de données que nous avons pu extraire de Refinitiv en figure 6.1. Sur cette figure, nous avons :

- en abscisse les maturités de notre Call à la monnaie (allant de 1 à 20 ans),
- en ordonnée la volatilité de notre Call (en pourcentage),
- une couleur par date considérée.

Entraînement du modèle Nous avons construit une base de données plus conséquent avec une méthode simple pour l'entraînement. Nous avons initialement un ensemble $\mathcal{X} = \{\mathbf{x}^1, \dots, \mathbf{x}^N\}$ où \mathbf{x}^i désigne la donnée des 20 volatilités pour les 20 maturités des 20 Call de la date i . Afin d'obtenir un peu plus de données nous avons choisi de créer des nouvelles observations \mathbf{x}^{new} telle que :

$$\mathbf{x}^{new} = \alpha \mathbf{x}^i + (1 - \alpha) \mathbf{x}^j,$$

α suivant une loi uniforme sur $[0, 1]$, i et j étant tirés aléatoirement dans l'ensemble $\{1, \dots, N\}$. L'avantage est d'avoir un nombre bien plus important de données pour le VAE, lui permettant d'atteindre de meilleures performances et de s'entraîner sur un grand ensemble.

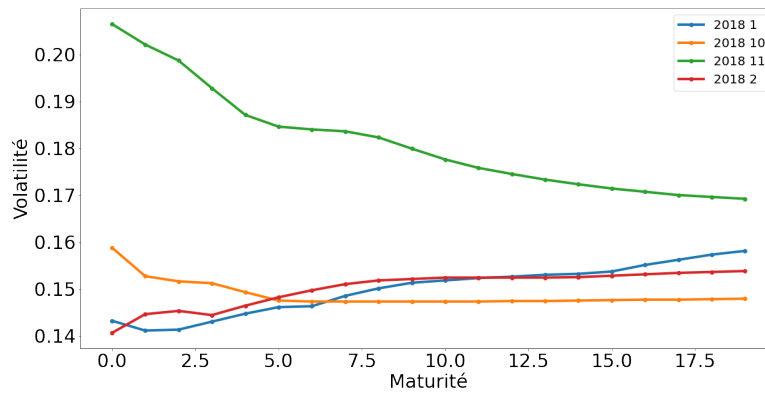


FIGURE 6.1 – Nous pouvons observer les différentes maturités allant de 1 an à 20 ans en abscisse. Nous avons les niveaux de volatilité en ordonnée et tout un ensemble de courbes qui correspondent aux différentes années. Les mois sont indexés de 0 pour janvier à 11 pour décembre.

Construction d'un espace latent En utilisant la méthodologie du chapitre précédent, nous obtenons après entraînement l'espace latent 6.2.

Definition 6.1.1. (VAE Equity) Dans toute la suite, lorsque que nous ferons référence à l'espace latent du VAE Equity pour des données de volatilités de Call, nous ferons référence à la projection de ces données dans l'espace décrit en figure 6.2. Lorsque que nous parlerons du premier axe latent, nous ferons référence à la valeur de la projection sur l'axe des abscisse et lorsque nous parlerons du second axe latent, il s'agira de la valeur de la projection sur l'axe des ordonnées.

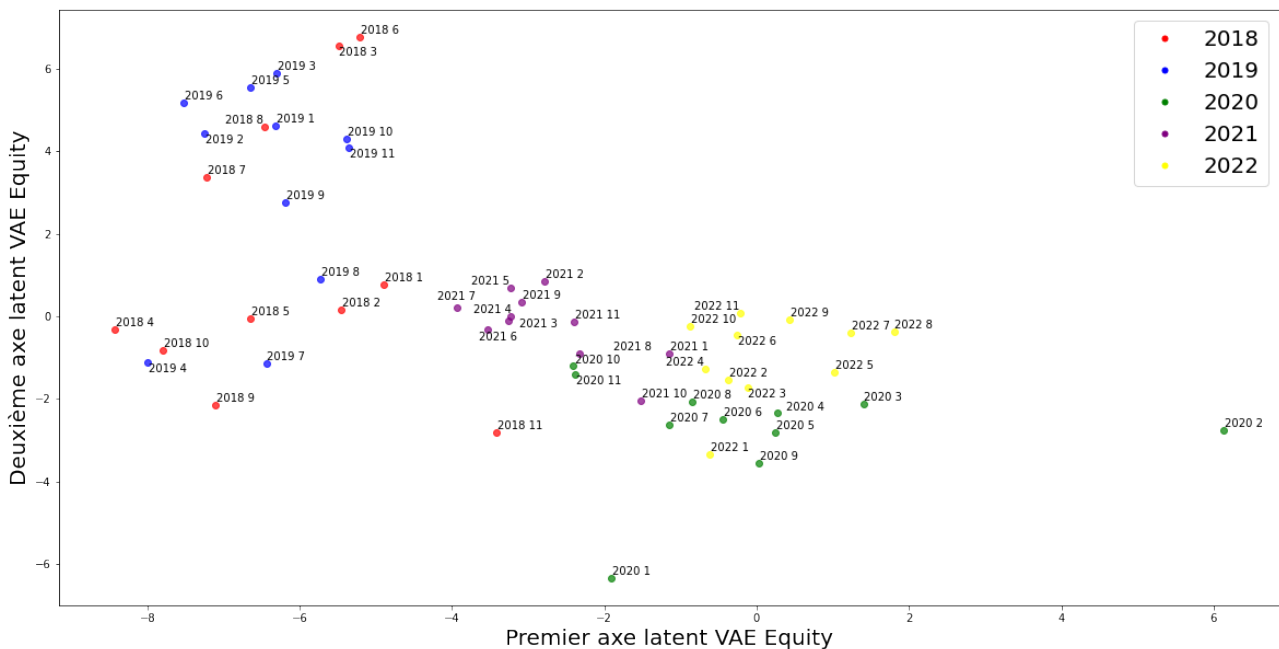


FIGURE 6.2 – Nous projetons avec le VAE les nappes de volatilité de Call dans un espace latent.

Il peut être particulièrement intéressant de comparer les performances d'une ACP et du VAE, puisque nous avons dit qu'utiliser un autoencodeur dégénéré revenait à faire une analyse par composante principale. La figure 6.3 permet de visualiser l'erreur moyenne de reconstruction

d'une courbe de volatilité pour d'une part un VAE qui projette en dimension 2 et d'autre part une ACP qui projette en dimension 2. Bien que l'erreur moyenne soit très proche, elle est meilleure pour le VAE, et nous réalisons que le VAE est capable de reconstruire des allures non linéaires, ce qui n'est pas le cas de l'ACP. Pour le calcul de l'erreur, nous calculons la RMSRE entre les données originelles et les données reconstruites. Il est intéressant de voir la performance du VAE Equity pour condenser et représenter des données.

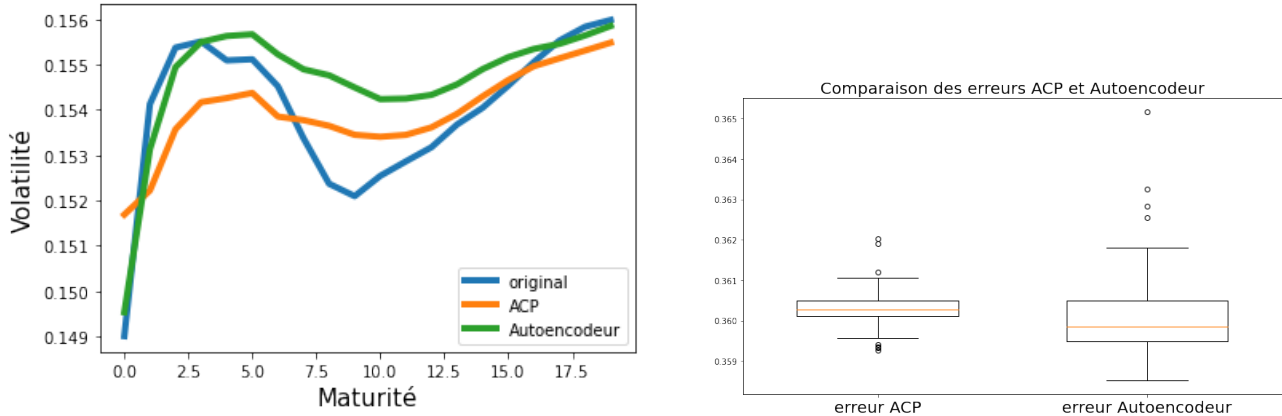


FIGURE 6.3 – ACP vs VAE.

Exploration de l'espace latent Afin de bien représenter les différents processus en jeu et de bien comprendre le fonctionnement du système utilisé, nous proposons au lecteur de se pencher sur la figure 6.4. Les courbes de volatilités, \mathbf{x}^i , y sont représentées et les processus latent associés, \mathbf{z}^i aussi. Nous remarquons plusieurs choses :

1. Si l'on prend une nappe de prix \mathbf{x}^i et que nous l'encodons, plusieurs points peuvent lui correspondre dans l'espace latent. Cela est parfaitement logique puisque nous utilisons un encodeur, et que comme nous l'avons dit la loi du processus latent \mathbf{z} conditionnellement à l'observation de \mathbf{x}^i suit une loi normale dont nous avons appris la moyenne et la variance. En d'autres termes : $\log q_\phi(\mathbf{z} | \mathbf{x}^{(i)}) = \log \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}^{(i)}, \boldsymbol{\sigma}^{2(i)}\mathbf{I})$. Pour construire la figure, nous avons donc décodé un point au hasard, par exemple $\mathbf{z}^i = (-8, 0)$, nous avons représenté une réalisation de $\log p(\mathbf{x} | \mathbf{z}) = \log \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2\mathbf{I})$ puis nous avons ré-encodé cette réalisation pour montrer les différentes valeurs de \mathbf{z} possibles.
2. En terme d'interprétabilité, c'est très élégant. En effet, tous les processus considérés suivent des lois normales dont nous avons appris la moyenne et la variance. Ainsi, si nous extrayons une nouvelle donnée du VAE en tirant un nouveau point au hasard, nous aurons une réalisation de $\log p(\mathbf{x} | \mathbf{z}) = \log \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2\mathbf{I})$!
3. L'espace latent a des variations continues. Si deux points sont proches dans cet espace, alors ils ont globalement les mêmes caractéristiques.
4. Nous observons que les volatilités sont d'autant plus fortes que nous nous plaçons à droite sur le premier axe latent du VAE Equity.

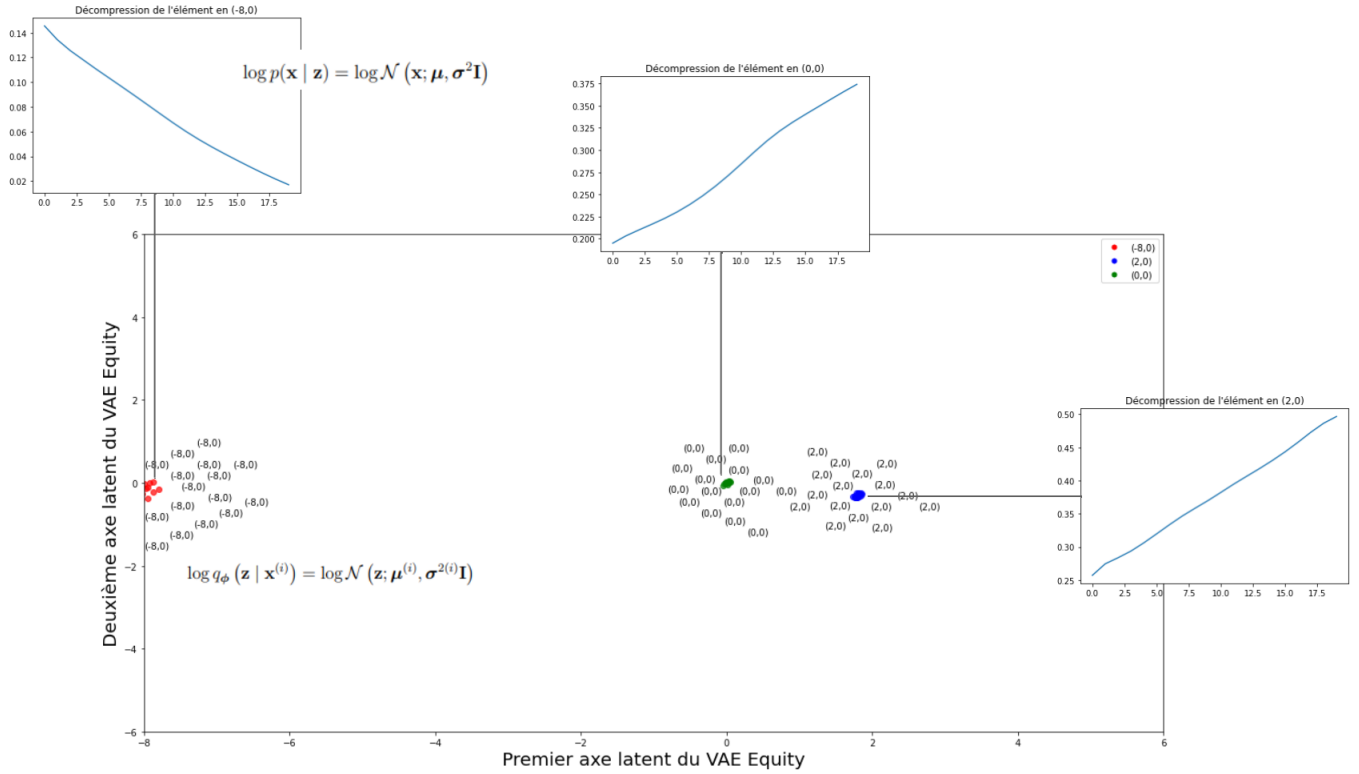


FIGURE 6.4 – Fonctionnement du VAE Equity. Nous observons le processus latent z , et les nappes de volatilités des options, les processus x .

Enfin, le dernier point qu’il nous semble important de montrer est la continuité des variations dans l’espace latent. La figure 6.5 montre la continuité des variations dans cet espace. En annexe nous proposons au lecteur intéressé d’explorer en détail les variations dans cet espace afin de pouvoir se convaincre des différentes régions.

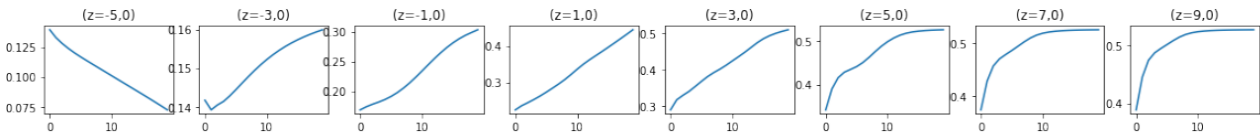


FIGURE 6.5 – Continuité des variations dans l’espace latent

6.2 Volatilités de Swaption

Allure des données utilisées Les nappes de volatilités de swaptions avec lesquelles nous travaillons sont constituées de 30 Tenors et de 30 Maturités. Ce sont des objets nettement plus complexes que ceux de la section précédente. Nous avons extrait les données avec *Refinitiv* et notre historique est constitué de données mensuelles allant de 2016 à 2023. Pour pouvoir comparer à l’ACP, il faut aplatir les données et de considérer un grand tableau Tenor*Maturités, cela nous ramène au cas précédent. Autrement, la comparaison avec l’ACP est impossible, car cette dernière serait incapable de capter autre chose qu’un plan.

Entraînement du modèle Les données étant relativement complexes et non disponibles en grand nombre, nous avons du nous entraîner sur une base de données déjà disponibles au préalable chez Milliman. La base de données que nous utilisons a été créée à partir de modèles

mathématiques et ne reflètent malheureusement ni des données **exactes** ni un contexte de marché **varié**. La figure 6.6 montre une nappe de volatilité de swaptions que nous avons dans l'ensemble d'entraînement, sa volatilité (bps) et la nappe que nous pouvons reconstruire avec le VAE après compression/décompression de celle-ci. Nous avons aussi calculé l'erreur comme la différence absolue entre la nappe initiale et la nappe reconstruite. Nous réalisons que sur l'ensemble d'entraînement, les résultats sont satisfaisants, l'erreur moyenne étant de **0.0003 bps** sur la volatilité.

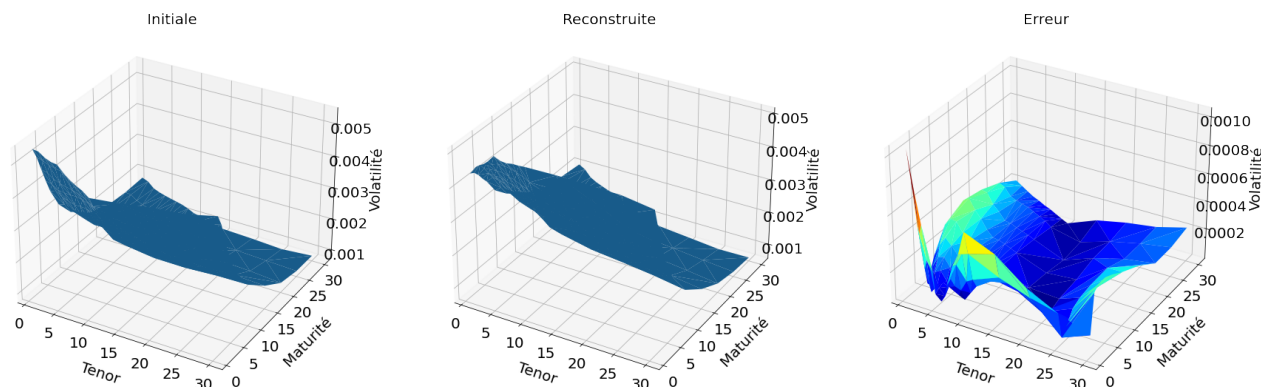


FIGURE 6.6 – Données de synthèses, entraînement

Les données sur lesquelles nous nous sommes entraînés permettent au VAE de généraliser ses performances sur des données réelles. L'erreur sur les données réelles de reconstruction est de **0.00026 bps**, ce qui est très proche de l'erreur sur les données de synthèses. Nous conservons donc des performances inchangées jusqu'ici. La figure 6.7 montre les résultats de la méthode sur des données réelles.

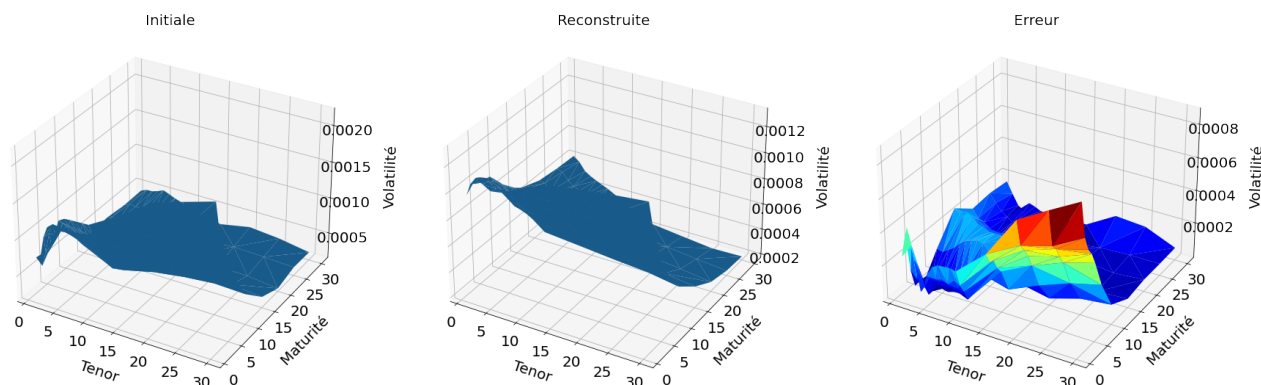


FIGURE 6.7 – Données réelles

Plusieurs remarques peuvent être faites suite à l'entraînement :

- Nous capturons non pas seulement un plan mais la **forme** des données. Cependant, nous n'avons pas un VAE parfait et il y a encore une perte d'information lors de la compression puis la décompression des données.
- Nous sommes capables de générer des données de swaptions de synthèses, en décompressant un point au hasard dans l'espace latent.
- Nous avons une erreur de reconstruction très faible devant la volatilité des nappes. Autrement dit, nous sommes très précis sur les ordres de grandeurs.

Nous pouvons, dans le cas des données d'entraînement, comparer les erreurs d'une ACP et du VAE. Il faut 3 composantes pour l'analyse par composantes principales pour réussir à

obtenir les mêmes résultats que le VAE avec 2 composantes. Nous avons calculé la RMSRE. 6.11.

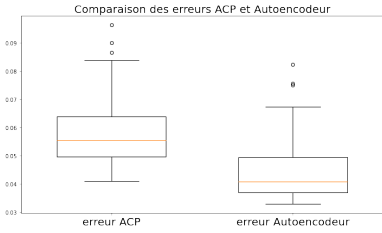


FIGURE 6.8 – 2 composantes pour l'ACP

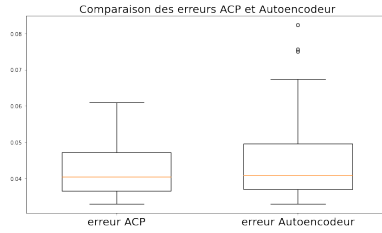


FIGURE 6.9 – 3 composantes pour l'ACP

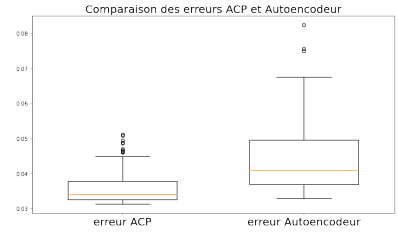


FIGURE 6.10 – 4 composantes pour l'ACP

FIGURE 6.11 – ACP vs VAE.

Definition 6.2.1. (VAE Swaptions) Dans toute la suite, lorsque que nous ferons référence à l'espace latent du VAE Swaptions pour des données de volatilités de swaptions, nous ferons référence à la projection de ces données dans l'espace décrit en figure 6.12. Lorsque que nous parlerons du premier axe latent, nous ferons référence à la valeur de la projection sur l'axe des abscisse et lorsque nous parlerons du second axe latent, il s'agira de la valeur de la projection sur l'axe des ordonnées.

Espace latent Nous pouvons observer en figure 6.12 l'espace latent du VAE Swaptions. Nous remarquons assez rapidement que la région à droite de cet espace est lié aux volatilités de swaptions élevées, et la région à gauche de l'espace aux volatilités de swaptions faibles. Nous ne tenons pas compte ici de la valeur de la courbe des taux, qui dans nos modèles varie en même temps que les volatilités de swaptions.

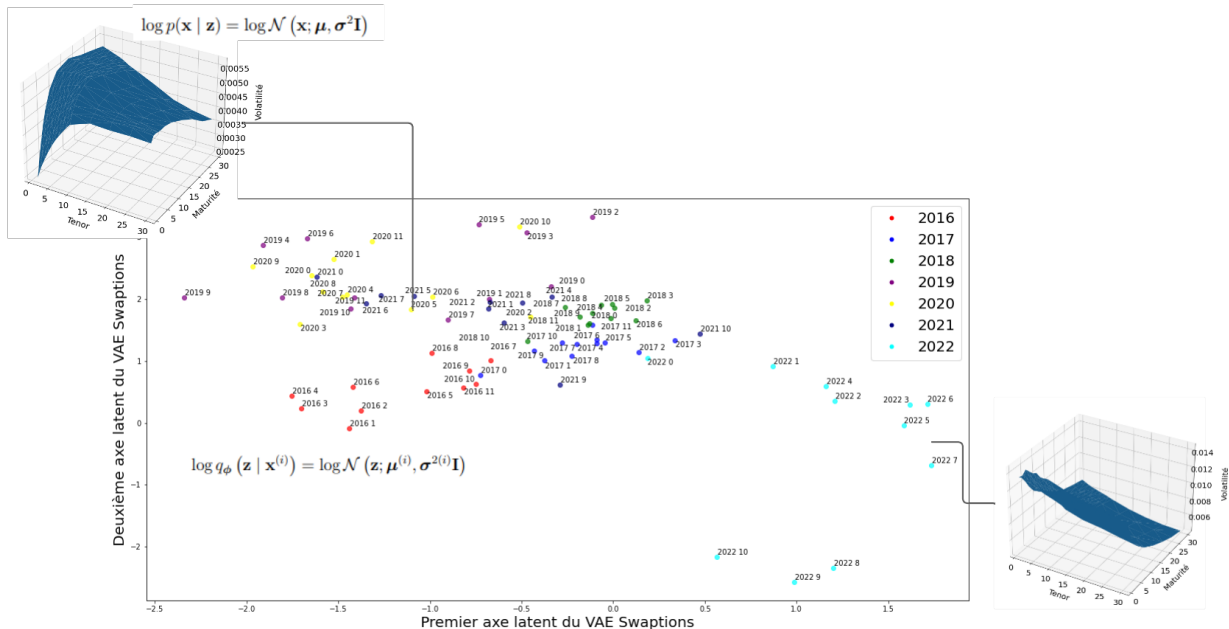


FIGURE 6.12 – Espace latent VAE Swaptions

Nous remarquons aussi que l'espace latent du VAE Swaptions a un bon comportement, et que les variations sont continues en son intérieur. La figure 6.13 permet de se faire une idée. Nous invitons le lecteur curieux à regarder en annexe pour plus de détails sur cet espace.

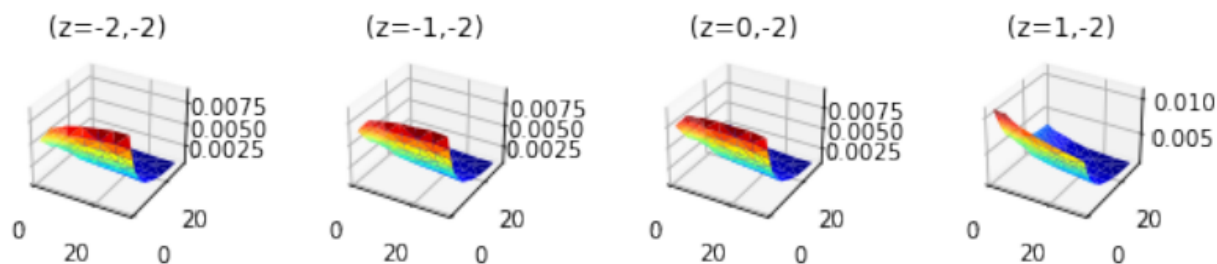


FIGURE 6.13 – Variations continues dans l'espace latent du VAE Swaptions.

Nous avons maintenant des outils pour caractériser précisément des données de marché. Nous sommes capables de raisonnablement les décrire, en généraliser et compléter des données manquantes à l'aide de ces méthodes. Naturellement nous aimerions maintenant évaluer l'impact des données de marché sur les valeurs de la VIF. En effet, bien que les données de marché soient imposées dans les calculs avec GSE, il est important de les étudier en détail afin de pouvoir mener une analyse de sensibilité sur les autres paramètres embarqués. Le schéma 6.14 illustre le fait que dans la vie réelle, les données de marché changent systématiquement d'un calibrage à l'autre. Ne pas en tenir compte lors d'une analyse de sensibilité pourrait conduire à d'importantes erreurs.

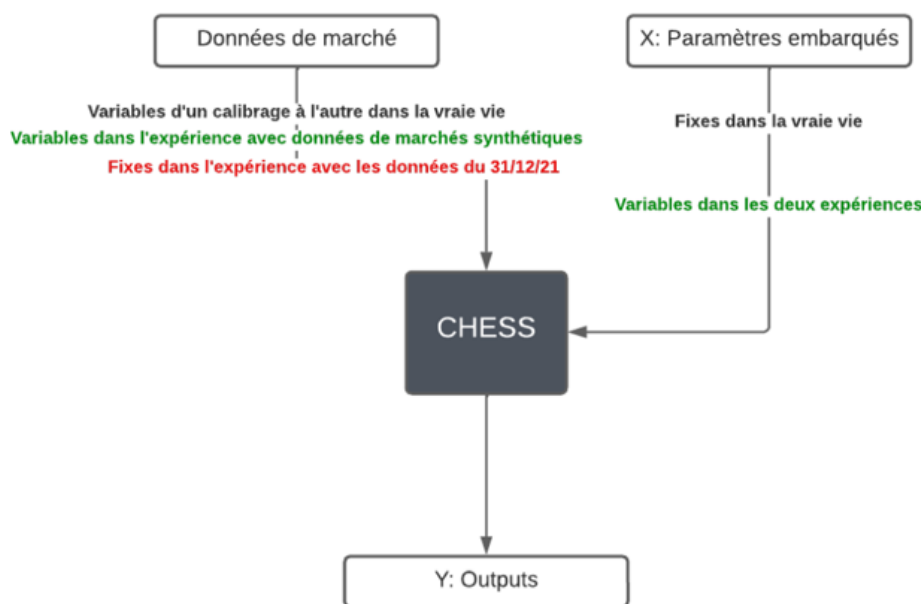


FIGURE 6.14 – Importance des données de marché

Nous proposons une étude réelle de la sensibilité aux données de marché dans le chapitre suivant.

Chapitre 7

Etude de l'impact des données de marché

Nous présentons dans ce chapitre les résultats de notre étude d'analyse de sensibilité aux données de marché.

7.1 Impact des données d'options de Call

Volatilités de swaptions fixées au 31/12/2021 Pour avoir quelques intuitions sur l'impact que peuvent avoir les données de volatilités de Call, nous proposons de garder les paramètres du modèle identique à ceux du 31/12/2021, qui, nous le rappelons, est notre date de référence. La figure 7.1 montre l'évolution de la VIF quand nous changeons les données de volatilités d'options, celles-ci étant décrites par leur position sur le premier axe latent du VAE Equity.

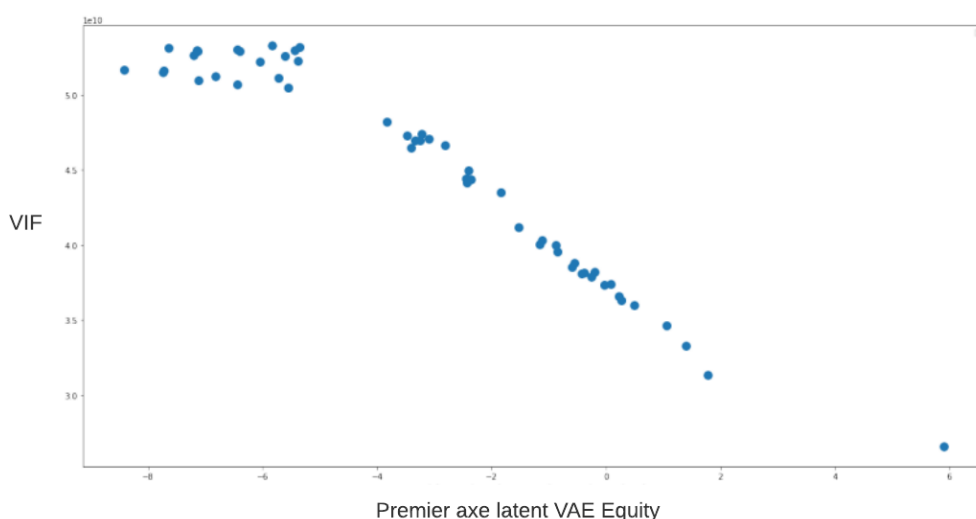


FIGURE 7.1 – VIF en fonction de la valeur x de l'abscisse dans l'espace latent des données de marché projetées.

Lorsque la volatilité des options call augmente, cela signifie généralement que le niveau de risque associé à ces options a également augmenté. Dans le contexte des assurances, cela peut se traduire par une augmentation de l'incertitude quant à la probabilité de déclenchement des options et des coûts associés.

Si la VIF fait référence à la valeur actuelle des polices d'assurance en cours (c'est-à-dire la valeur future actualisée des primes attendues moins les coûts et les sinistres anticipés), une augmentation de la volatilité des options call peut entraîner une diminution de la valeur de la value in force. Cela peut être dû à plusieurs facteurs :

1. **Risque accru** : L'augmentation de la volatilité des options call indique une plus grande variation potentielle des résultats futurs, ce qui peut augmenter le niveau de risque associé aux polices d'assurance. Les assureurs doivent souvent tenir compte de ce risque accru en ajustant leurs réserves et en prenant en compte une marge de sécurité supplémentaire, ce qui peut réduire la valeur de la VIF.
2. **Coûts de couverture** : Les assureurs peuvent utiliser des options call pour couvrir certains risques associés à leurs polices d'assurance. Si la volatilité des options call augmente, les coûts de couverture associés à ces options peuvent également augmenter, ce qui peut réduire la valeur nette de la VIF.
3. **Impact sur les tarifs** : L'augmentation de la volatilité des options call peut également avoir un impact sur les tarifs des polices d'assurance. Les assureurs peuvent ajuster leurs tarifs en fonction du niveau de risque perçu, ce qui peut influencer la valeur de la VIF.

Volatilités de swaptions changeante Afin d'avoir une idée plus précise de l'impact des données de volatilités d'options, et ce quel que soit le contexte de taux, nous proposons d'étudier leur impact dans un cadre où les volatilités de swaptions (et donc les taux) évoluent. Quelques dates différentes pour les données de swaptions sont étudiées dans la figure 7.2. Toutes les autres données sont fixes pour réaliser ces tests.

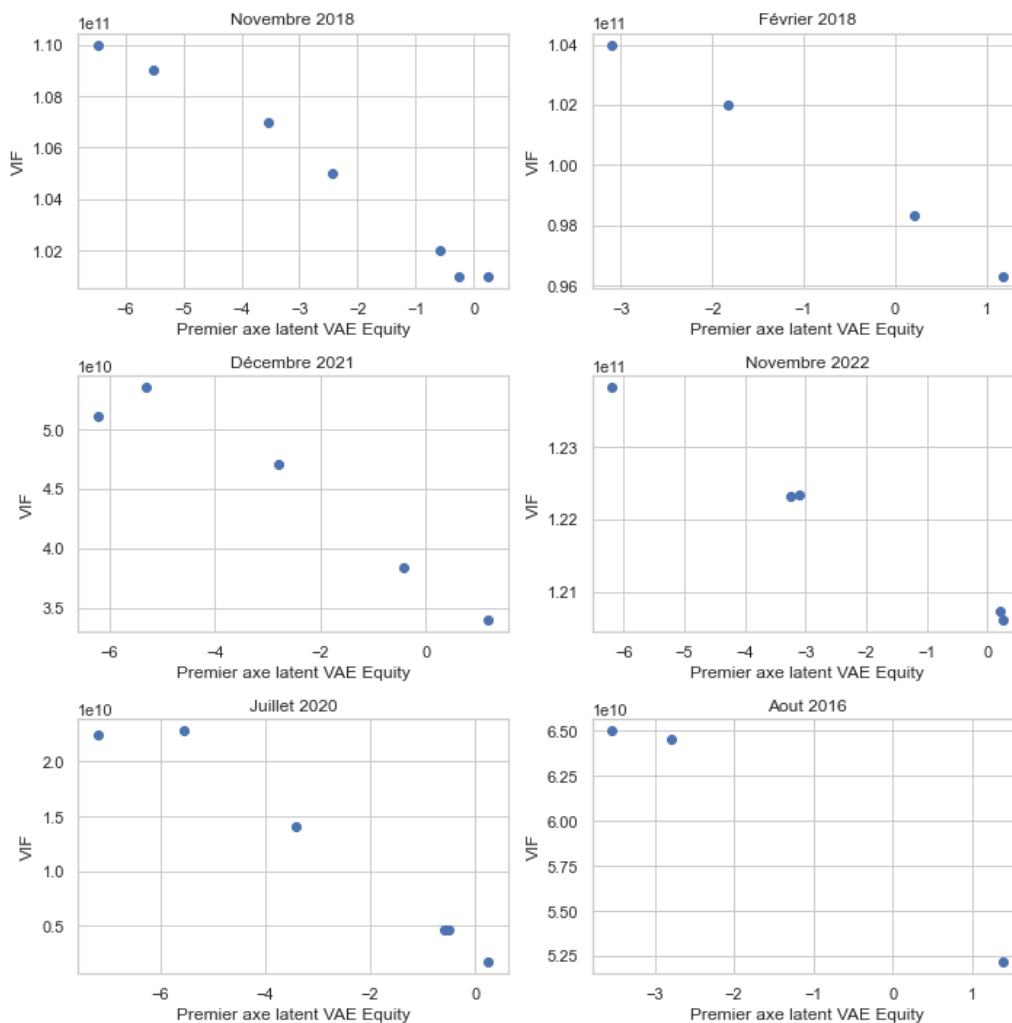


FIGURE 7.2 – Sensibilité de la VIF à la volatilité

Nous observons que quelque soit la date, l'impact semble rester le même. Nous proposons dans un premier temps de superposer tous ces résultats sur un même graphique afin de voir

plus clairement ce qu'il se passe. La figure 7.3 montre l'évolution de la VIF en fonction à la fois de la position dans l'axe latent premier du VAE Equity et de la date à laquelle nous considérons les données de swaptions.

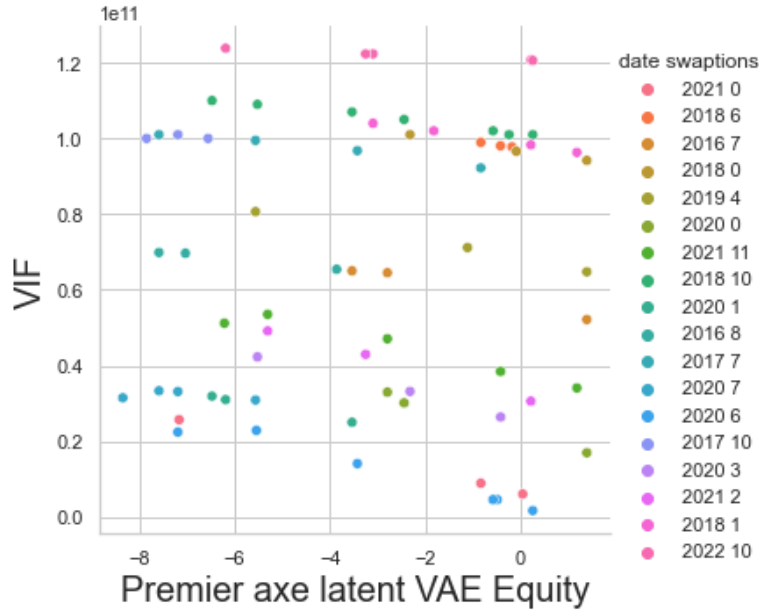


FIGURE 7.3 – Sensibilité de la VIF à la volatilité des CALL.

Les données font varier de manière non négligeables la VIF, et la distribution est devenue multimodale, puisque nous y avons fait varier les données de volatilité de swaptions. Afin de pouvoir comparer les impacts lors de différentes dates, il peut valoir le coup de réaliser une translation, afin de ramener toutes les données à la même échelle. Nous proposons de choisir un point de référence x_{ref} sur le premier axe latent du VAE equity. Nous allons ensuite calculer VIF_{ref}^i pour chaque condition i de marché différente et faire subir aux points VIF^i la transformation :

$$VIF^i \longleftarrow VIF^i - VIF_{ref}^i$$

En pratique, nous ne connaissons pas la valeur de VIF_{ref}^i . Nous proposons de calculer cette valeur avec une simple interpolation linéaire, avec ses plus proches voisins. La figure 7.2 motive ce choix et nous fait nous dire qu'il est raisonnable. Cette transformation est très utile dans le sens où elle nous permet de translater toutes les données de manière à avoir un même point de référence pour toute. Nous pourrions par la suite calculer le pourcentage de variations de manière aisée. La figure 7.4 illustre l'application de cette méthode très simple.

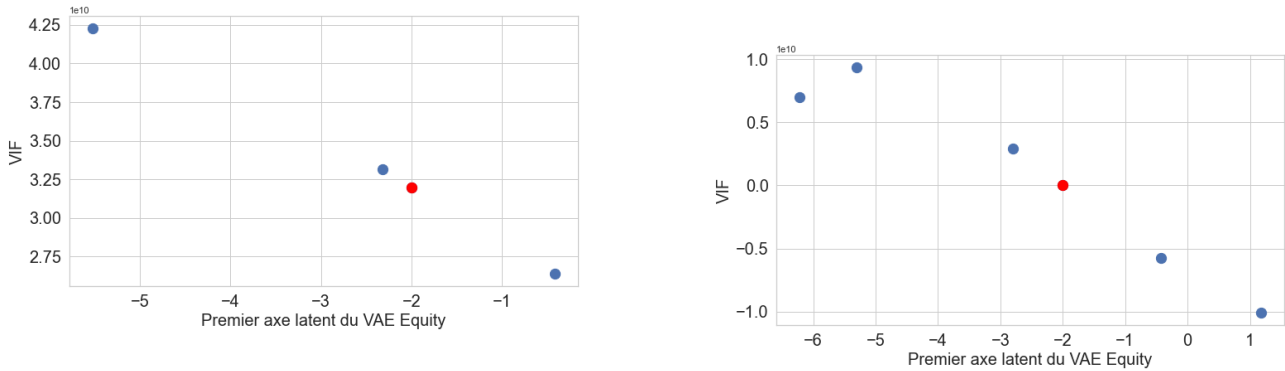


FIGURE 7.4 – Interpolation pour obtenir VIF_{ref}^i en $x_{ref} = -2$. Le point interpolé est en rouge. Deux dates différentes ont été considérées pour les volatilités de swaptions.

Nous pouvons maintenant mettre les différents points à la même échelle, nous prenons arbitrairement $x_{ref} = -2$. La figure 7.5a montre les variations brutes de la VIF en fonction des différents paramètres que nous avons évoqués jusqu'ici. Nous voyons que quel que soit le contexte de taux, les variations sont similaires. Dans notre modèle, il ne semble pas y avoir de relation excessive entre l'Equity et les Taux. Nous pouvons aller plus loin en regardant l'évolution en pourcentage de la VIF 7.5b. Nous voyons qu'en fonction des dates, l'amplitude n'est pas la même.

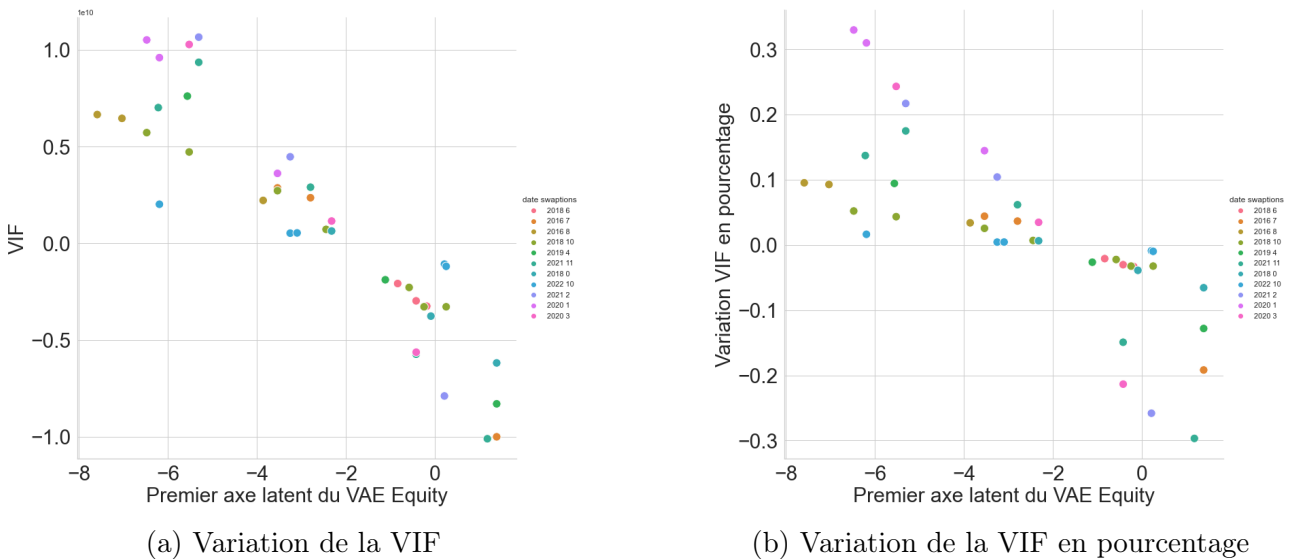


FIGURE 7.5 – Variation de la VIF pour les points recentrés, par rapport à $x_{ref} = -2$

Nous avons un VAE Swaptions qui permet de caractériser les données de volatilité de swaptions. Nous nous proposons de colorier les points en fonction de leur position sur le premier axe latent VAE Swaptions. Cela nous permettra de mieux comprendre le graphique 7.5b. L'observation de la figure 7.6, nous montre que **l'impact des volatilités d'options sur CALL est d'autant moins important que les volatilités de swaptions sont élevées (foncées)**.

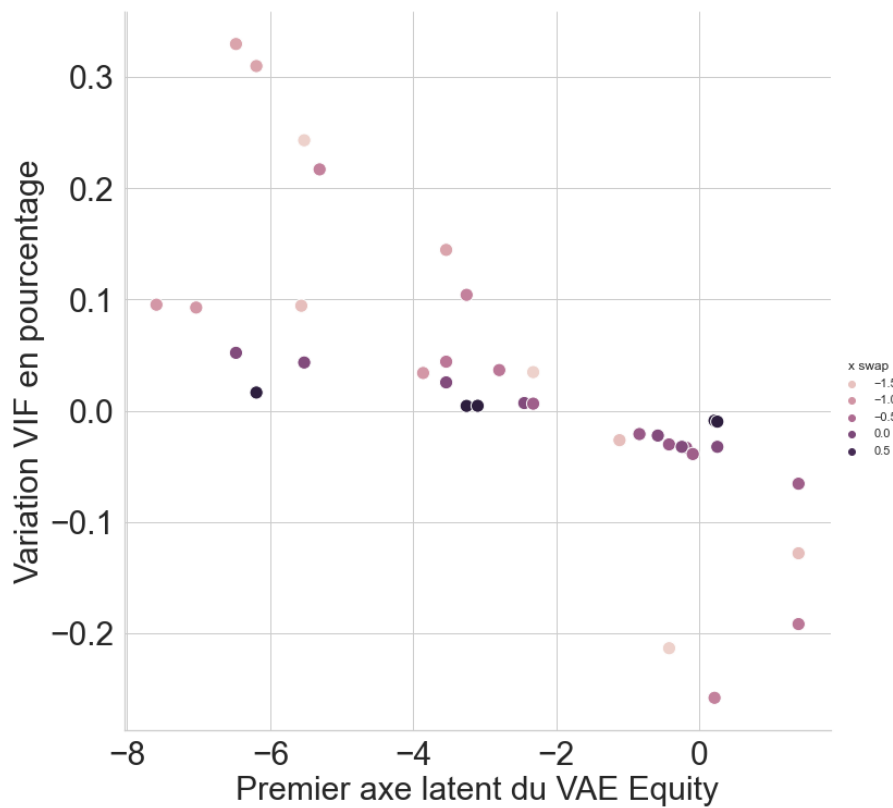
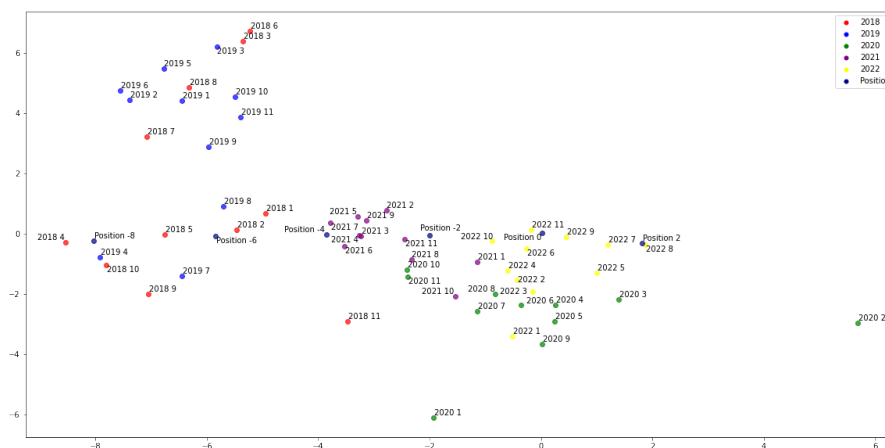


FIGURE 7.6 – Variation de la VIF en fonction du choc sur les vol equity et en fonction des vol de taux

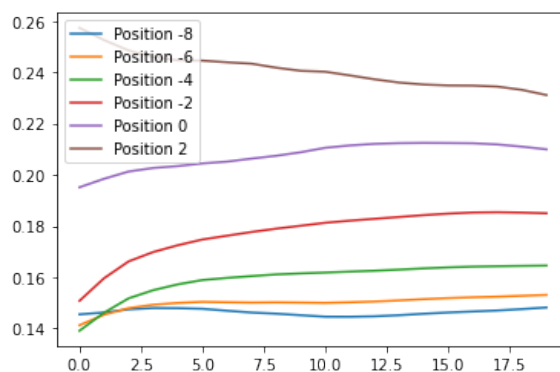
Quel impact précis sur la VIF d'un changement de volatilité sur les CALL ? Ces différentes analyses nous permettent de dresser un tableau des impacts moyens lorsque nous changeons les données de volatilités de CALL. Nous les avons obtenus par régression en choisissant des x_{ref} différents. Dans le tableau ci-dessous, nous regardons la position de départ sur le premier axe de l'espace latent du VAE Equity, et la position d'arrivée dans cette espace, lorsque nous modifions une donnée de marché.

Position de départ/d'arrivée	-8	-6	-4	-2	0	2
-8	0	-7%	-14%	-21%	-25%	-36%
-6	7%	0	-7%	-14%	-22%	-29%
-4	14%	6.8%	0	-8%	-15%	-23%
-2	22%	14%	7%	0	-8%	-15%
0	30%	22%	14%	7%	0	-8%
2	38%	30%	22%	14%	7%	0

En fonction de la position de départ dans le premier axe principal du VAE equity et de la position d'arrivée dans celui-ci, nous sommes en mesure de donner l'impact moyen que cela a sur la VIF (obtenu avec régression linéaire). Attention, il faut en revanche bien prendre en compte que pour les taux élevé l'impact sera diminué et que pour les taux plus faibles l'impact sera un peu plus important.



(a) Espace latent VAE Equity



(b) Volatilités de call associées

FIGURE 7.7 – Aide pour lire le tableau des positions

Ainsi, si nous changeons la nappe de volatilité de call qui est en position -8 par une nappe de volatilité différente, c'est à dire proche de la position 2 par exemple, alors nous faisons diminuer notre VIF de 36% en moyenne. Nous savons que cela va varier un peu en fonction du contexte de taux, mais nous avons tout de même une indication sur la violence du choc. En suivant cette méthodologie, nous sommes capables de dire, si toute chose reste égale par ailleurs, la variation de la VIF si nous changeons le contexte de taux.

Le modèle est sensible aussi aux volatilités long terme !

7.2 Impact des données d'options sur Swaptions

Les données de volatilité de swaptions permettent de calibrer les modèles de taux. Nous nous attendons à ce que leur impact soit relativement important. Nous proposons d'observer les variations de la VIF en fonction de la position sur le premier axe latent du VAE Swaptions, en figure 7.8. Nous voyons déjà que les données sont relativement bien ordonnées dans cet espace. Il y a cependant un bruit important du à la fois au changement de la courbe des taux et un autre du au changement des données de marché Equity.

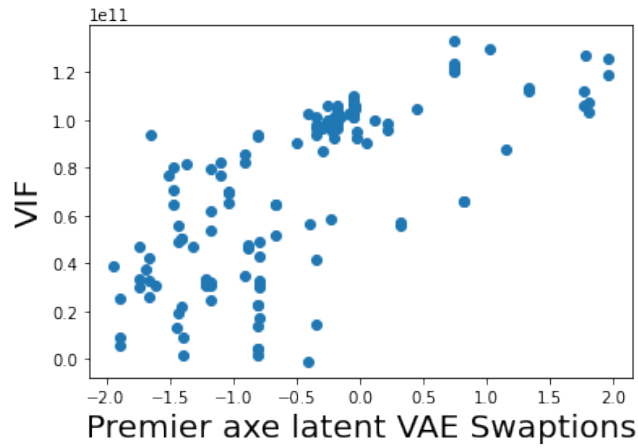


FIGURE 7.8 – VIF en fonction des volatilités de swaption

Bien que nous ne puissions pas capter les variations dues à la courbe des taux, nous pouvons de nouveau afficher les variations dues aux changements de données Equity, que nous avons présentées en détails dans la section précédente.

Nous sommes en mesure de faire une analyse de sensibilité de la VIF au premier axe latent du VAE Swaptions. Nous remarquons ici qu'il n'y a pas de distribution multimodale, nous pourrions a priori utiliser n'importe quel indicateur de sensibilité.

Conclusion L'étude des données de marché et de leur impact est une étape importante pour relativiser l'importance des paramètres embarqués d'un GSE et pour quantifier leur impact. Cette étude nous a permis de présenter la méthodologie que nous allons dérouler pour différents paramètres embarqués. Nous retiendrons qu'il peut être très utile d'utiliser des VAE pour clusteriser des volatilités, et que les données de volatilités de swaptions ainsi que la courbe des taux ont un impact très important sur les valeurs finales obtenues.

Quatrième partie

Application aux paramètres embarqués d'un GSE

Les résultats des chapitres précédents résultent des difficultés que nous avons pu rencontrer lors de l'analyse des paramètres embarqués d'un générateur de scénarios économiques. La prise en compte du contexte de marché, ainsi que de la multi-modalité des distributions est un élément important et peut être étendu à différents domaines.

Dans ce chapitre plus applicatif, nous proposons de mettre en évidence les paramètres embarqués ayant le plus d'impact sur la réponse du système. Ce sont ces paramètres que le praticien se doit de calibrer le plus régulièrement. Nous espérons aussi donner au praticien une indication sur l'importance de regarder le contexte du marché afin de savoir quand recalculer ces différentes valeurs.

Chapitre 8

Le Shift

Nous proposons dans cette section de terminer l'analyse de notre exemple introductif, à savoir le shift. Nous présentons l'impact d'une variation de ce paramètre embarqué sur les taux minimum atteints, ou encore les sorties ALM telles que la VIF.

8.1 Output de CHESS

Le shift est un paramètre embarqué présent dans les modèles de taux nominaux, il va impacter directement, avec les volatilités de swaptions, les taux projetés par le générateur de scénarios économiques. Pour fixer les idées, nous décidons de considérer un zéro coupon bond (ZCB) de maturité 1 an. Nous allons regarder la projection à 1 an de ce ZCB. Afin de permettre au lecteur de se faire une idée plus claire du shift, nous allons tout simplement regarder comment les données de swaptions et les valeurs du shift peuvent impacter les taux. Les figures ci-dessous montrent l'évolution de la valeur des taux minimum atteints pour les zéros coupons considérés, en fonction de la valeur du shift et de la position des swaptions utilisées dans l'espace latent du VAE Swaptions.

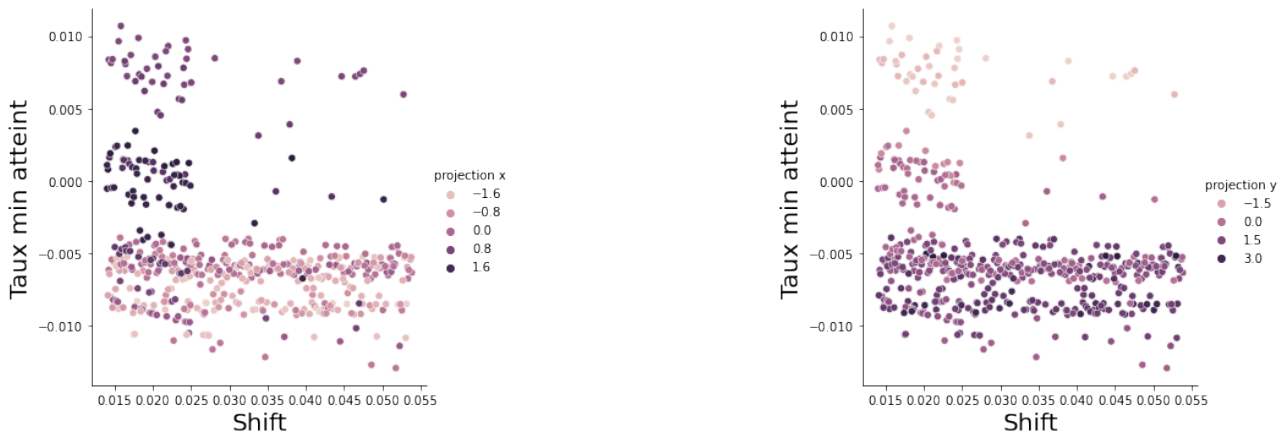


FIGURE 8.1 – Impact du shift sur les taux minimums atteints, prise en compte du contexte de taux.

Nous remarquons que lorsque les swaptions ont de grandes volatilité, (contexte de taux élevé), le shift a un impact d'autant plus important qu'il est faible. En revanche, il ne semble pas y avoir de dépendance particulière lorsque le contexte de taux est plus faible. Nous montrons aussi la dépendance aux données de marché ci-dessous.

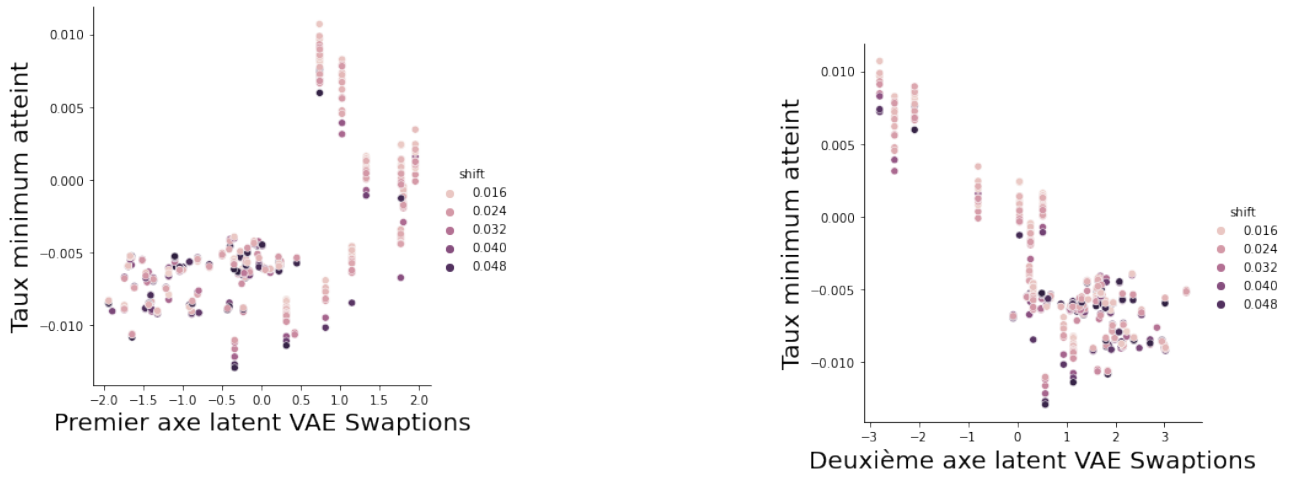


FIGURE 8.2 – Impact du shift sur les taux minimums atteints.

Nous sommes clairement dans un cas de distribution multimodale lorsque nous voulons analyser l’impact du shift sur les taux minimum atteints. Nous disposons aussi de suffisamment de données (plus de 500 résultats) pour mener une analyse approfondie des impacts du shift, et ce de manière globale et non dans un seul contexte de taux.

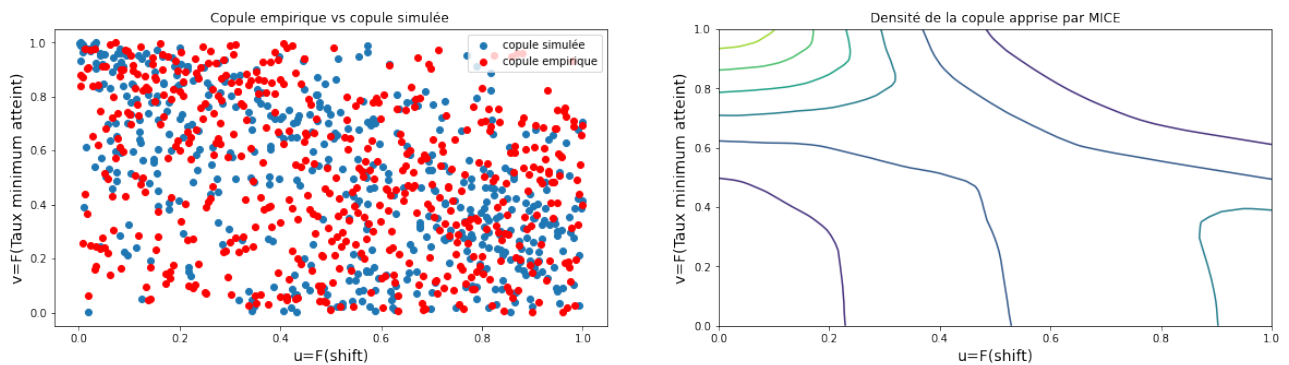


FIGURE 8.3 – Notre modèle a réussi à apprendre les copules et les niveaux de densité.

La figure 8.3 montre la copule que nous apprenons pour le couple shift-taux minimum atteint. Nous nous rendons compte que la dépendance est bien réelle, en haut à gauche, et que nous sommes sur une copule d’indépendance en bas à droite. Cela reflète que l’impact du shift est différent en fonction du contexte de taux. En plus de cela, nous remarquons que la densité est apprise, nous nous attendons donc à un résultat cohérent pour l’indice de sensibilité.

La figure suivante montre la copule que nous apprenons pour la position dans le deuxième axe de l’espace latent VAE Swaptions.

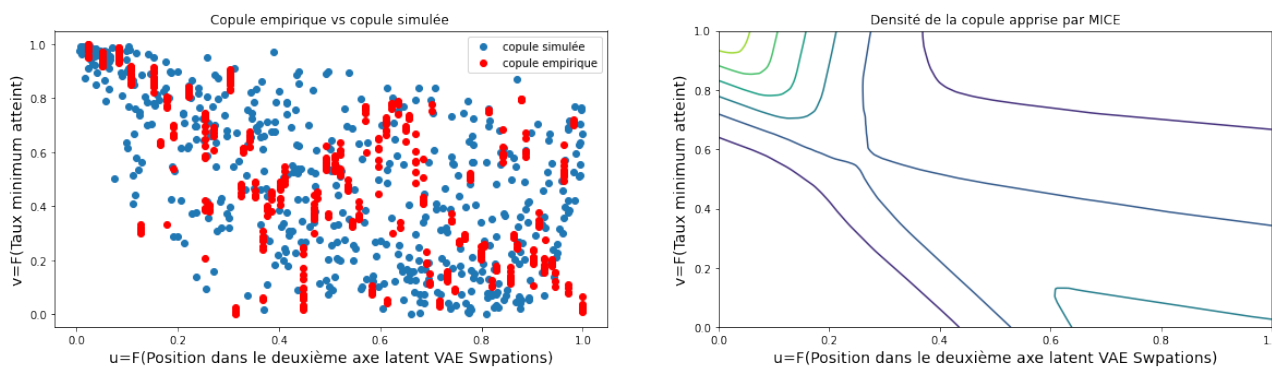


FIGURE 8.4 – Deuxième axe latent du VAE swaption, apprentissage de la densité.

Cela conduit à calculer les indices de sensibilité :

	Shift	Swaptions
ρ_{MI}	0.50	0.66

Nous obtenons que les volatilités de swaptions ont un impact plus important que le shift. Cependant, l'impact du shift n'est pas négligeable et il est réel, ce qui est d'autant plus vrai que les volatilités de swaptions sont élevées.

8.2 Résultats ALM

La figure 8.5 présente plusieurs statistiques décrivant l'impact du shift à différentes dates (les dates correspondent aux couples volatilité de swaption-courbe des taux).

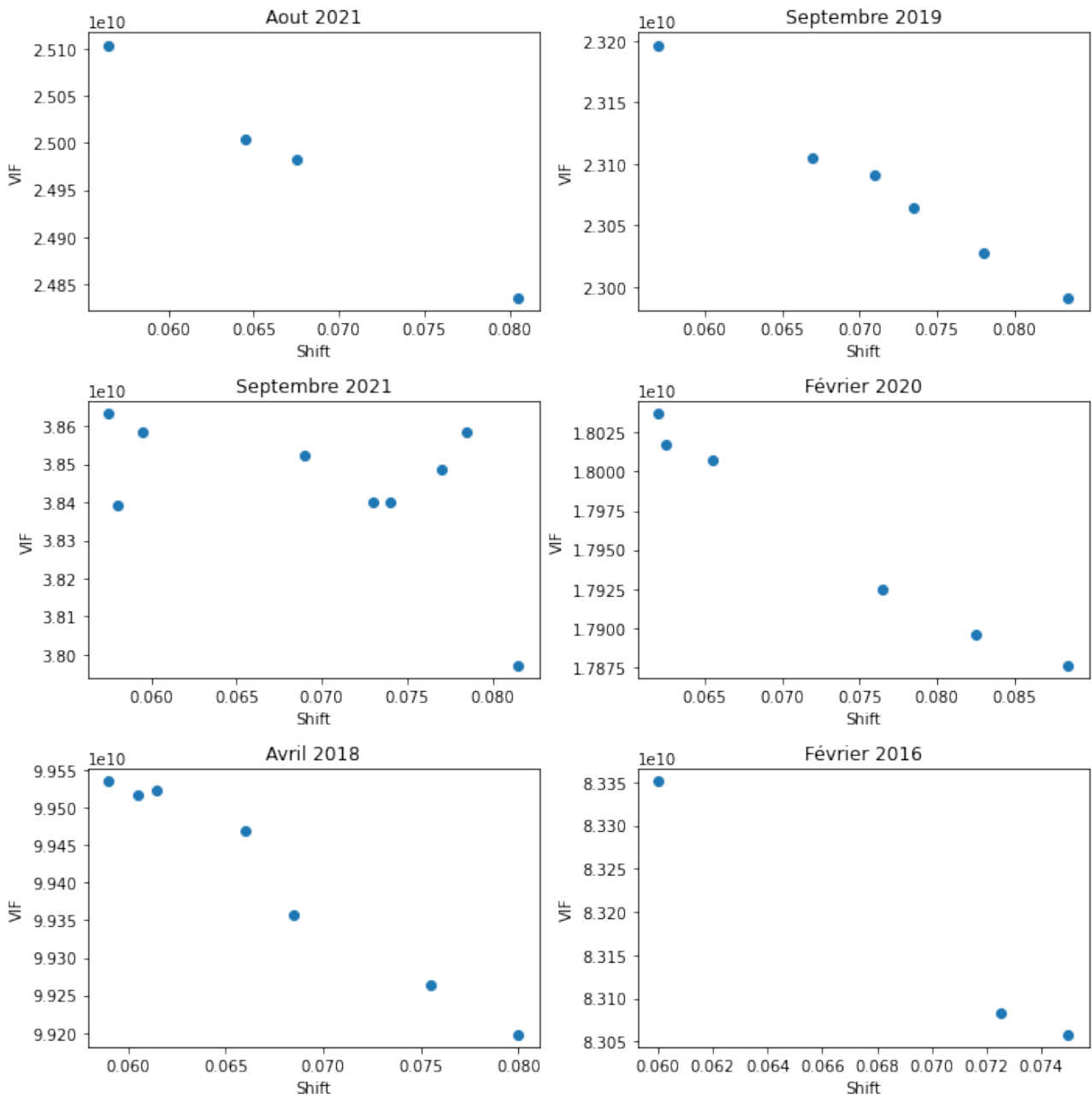


FIGURE 8.5 – Statistiques descriptives du shift

Nous voyons que l'impact du shift est globalement linéaire. Nous observons que pour des périodes de volatilités de swaptions plus élevées, le comportement est légèrement différent. Pour pouvoir comparer les effets du shift à différentes dates, nous utilisons la technique de translation présentée en 7.1.

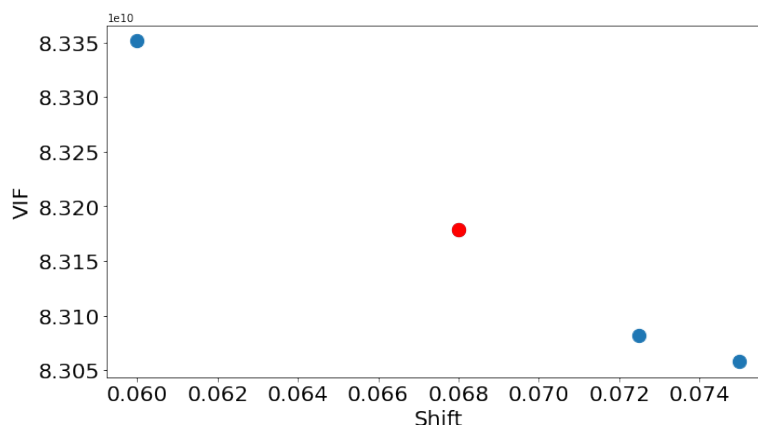


FIGURE 8.6 – Statistiques descriptives du shift

Nous avons choisi comme point de référence **shift** = **0.068**. Nous allons calculer ce point pour chaque date puis tout translater pour comparer les effets du shift en fonction des conditions de marché. (seules les volatilités de swaptions changent ici).

Nous pouvons dès lors afficher les résultats centrés par rapport au point de référence. Nous nous proposons, dans tous les cas, de calculer le pourcentage de variation du VIF. Nous représentons ci-dessous les résultats.

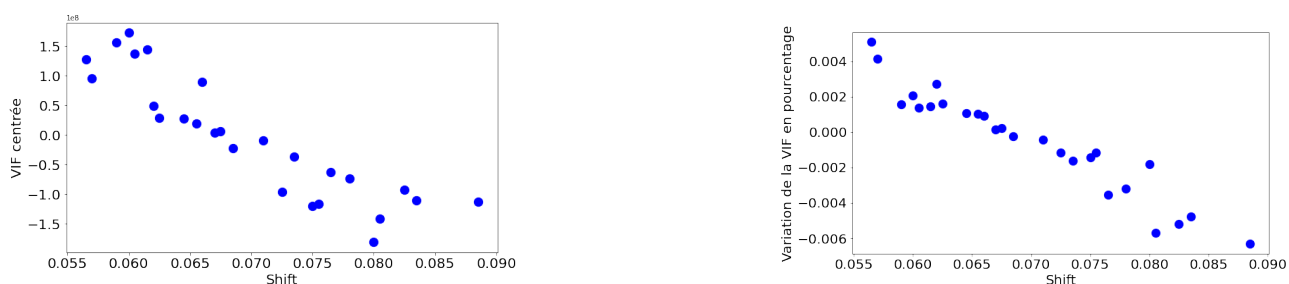


FIGURE 8.7 – Impact du Shift, résultats centrés

Nous remarquons que quand une modification des données de volatilité des options call engendre une variation allant jusqu'à 30% de la VIF, une modification du shift ne modifie que d'un ordre de grandeur de 0.4% la VIF, et ce quel que soit le contexte de taux! Afin d'avoir une analyse un peu plus fine de la situation, et en adéquation avec ce que nous avons pu observer précédemment, nous pouvons distinguer les différentes valeurs en fonction du niveau de volatilité des swaptions.

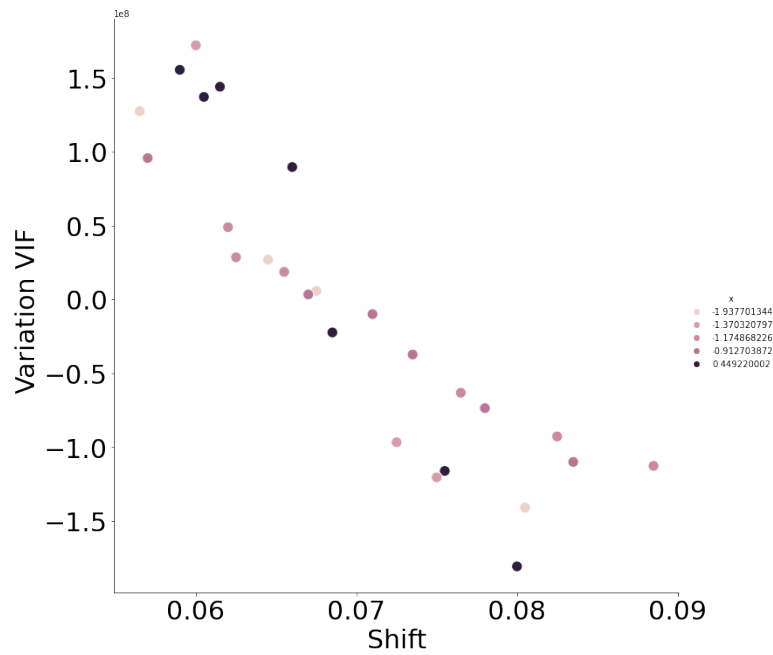


FIGURE 8.8 – Plus les volatilités de swaptions sont élevées, plus l’impact sur le shift est élevé

Nous observons que les points les plus foncés, correspondant aux volatilités de swaptions les plus élevés, sont plus sensible à un impact du shift. Cela est en adéquation avec les remarques que nous avons faites précédemment.

Conclusion Nous avons donné quelques intuitions sur l’impact du shift (son impact est d’autant plus fort que son niveau est bas, l’impact est plus élevé lorsque les taux sont hauts). Il a été possible de montrer l’impact directement en sortie du générateur de scénarios économiques et de l’analyser en détail. Il n’est pas pertinent, à ce stade, de calculer des indices de sensibilité sur les sorties ALM, étant donné la faible quantité de données disponibles.

Chapitre 9

GLMM

Cette partie est dédiée à l'étude des paramètres embarqués du modèle GLMM, utilisé pour modéliser l'inflation. Nous nous attarderons à bien définir ces paramètres et à étudier leur domaine de définition afin de procéder à leur analyse de sensibilité.

9.1 Inflation : Gaussian LIBOR Market Model (GLMM) et modèle de Vasieck à deux facteurs (2VF)

Le modèle Vasieck à deux facteurs (2FV) fournit la dynamique du taux d'intérêt réel à court terme r_t , ainsi que de sa moyenne à long terme instantanée m_t . Sous la mesure neutre au risque \mathbb{P}^* , leur dynamique est donnée par :

$$\begin{cases} dr_t = \alpha_1 (m_t - r_t) dt + \sigma_1 dW_t^{(1)} \\ dm_t = \alpha_2 (\mu - m_t) dt + \sigma_2 dW_t^{(2)} \end{cases},$$

avec $\Theta = \{r_0, m_0, \mu, \alpha_1, \alpha_2, \sigma_1, \sigma_2\}$ les sept paramètres du modèle et $W^{(1)}, W^{(2)}$ deux mouvements browniens indépendants. Les paramètres peuvent être interprétés comme suit :

- r_0, m_0 sont les valeurs initiales, respectivement, du taux court et de sa moyenne à long terme instantanée,
- α_1 est la vitesse de réversion moyenne du taux court,
- α_2 est la vitesse de réversion moyenne du processus de moyenne à long terme instantanée,
- μ est la moyenne à long terme du processus de moyenne à long terme instantanée,
- σ_1 est la volatilité du taux court,
- σ_2 est la volatilité du processus de moyenne à long terme instantanée.

Étant donné la courbe des taux forward initiale au temps $t = 0$, le modèle GLMM ne dépend que des paramètres de distribution $\Theta_{\mathbf{d}} = \{\alpha_1, \alpha_2, \sigma_1, \sigma_2\}$. Les paramètres du modèle 2FV peuvent être divisés en deux groupes :

- les paramètres de bord $\Theta_{\mathbf{b}} = \{r_0, m_0, \mu\}$ évaluant les valeurs initiales et les valeurs à long terme du taux court et de son processus de moyenne à long terme instantanée,
- les paramètres de distribution $\Theta_{\mathbf{d}} = \{\alpha_1, \alpha_2, \sigma_1, \sigma_2\}$ qui influencent la variance et la covariance du taux court et de son processus de moyenne à long terme instantanée.

D'un point de vue économique, nous pouvons soutenir que la courbe initiale des taux zéro-coupons réels ne fournit aucune information satisfaisante sur la variance des taux réels. Par conséquent, pour la cohérence et la stabilité des distributions des taux zéro-coupons réels, nous nous appuyons sur l'approche en deux étapes suivante :

- Pré-estimer un ensemble $\widehat{\Theta}_d$ de paramètres de distribution grâce à des études historiques et économiques dédiées. Ces paramètres de distribution resteront identiques entre les différentes calibrations.
- Calibrer les paramètres de frontière sur la courbe de taux réel initiale en tant que solution du problème de minimisation suivant :

$$\widehat{\Theta}_b = \underset{\Theta_b \in \mathbb{R}^3}{\operatorname{argmin}} \left\{ \sum_{k=1}^M w_k \left(P_r \left(0, t_k, \{ \Theta_b, \widehat{\Theta}_d \} \right) - P_r^{Mkt} \left(0, t_k \right) \right)^2 \right\},$$

finalement, l'ensemble final des paramètres estimés est $\widehat{\Theta} = \{ \widehat{\Theta}_b, \widehat{\Theta}_d \}$.

Estimation des paramètres distributionnels $\widehat{\Theta}_d$ La méthode repose sur trois étapes principales :

- Extraction de la série de données historiques de la structure à terme des taux d'intérêt nominaux et de la structure à terme des taux d'inflation anticipée (BEIR),
- Calcul des estimateurs historiques des objectifs de distribution,
- Calibration des paramètres de distribution en minimisant les erreurs quadratiques entre les objectifs historiques et les valeurs correspondantes du modèle dans le cadre du modèle 2FV.

Nous avons choisi d'utiliser trois cibles de distribution différentes :

- **Vol** la volatilité des incréments du taux de rendement réel ZC,
- **Correl** la corrélation entre les incréments des taux de rendement réel de différentes échéances,
- **Sd** l'écart-type du taux de rendement réel.

Les justifications pour choisir ces trois cibles de distribution sont les suivantes :

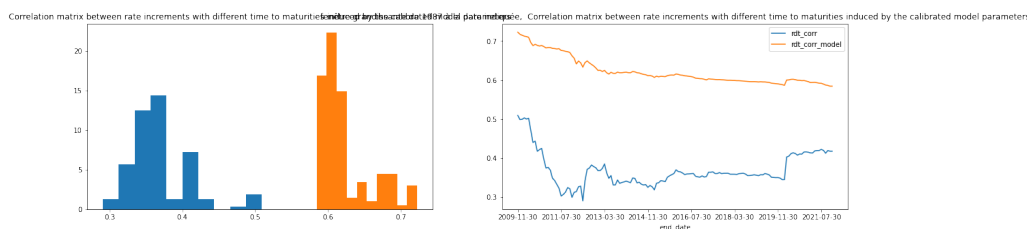
- La volatilité des incréments du taux de rendement ZC est une quantité clé principalement pilotée par les paramètres de volatilité,
- L'utilisation de cibles sur les corrélations permet de reproduire une structure de corrélation et d'éviter les taux à 100
- L'intégration de cibles sur les écarts-types permet d'estimer correctement les paramètres de vitesse de réversion à la moyenne α_1 et α_2 . Ces paramètres gouvernent le rapport entre la volatilité des incréments du taux de rendement ZC (écart-types des incréments de taux) et les écarts-types des taux (également appelés dispersion).

Quel est l'impact des paramètres distributionnels des modèles 2VF et GLMM ?

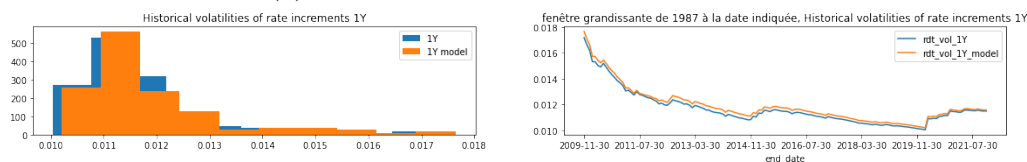
9.2 Étude historique des paramètres distributionnels et détermination de cibles

La figure 9.1 illustre l'évolution historique des cibles distributionnelles. Il est plus simple d'appliquer un choc aux cibles définies plus haut, et de regarder leurs paramètres embarqués associés, plutôt que d'appliquer un choc directement sur les paramètres embarqués. En effet, on exprime en fonction des paramètres embarqués. Par la suite, les paramètres embarqués sont déduits de la valeur des cibles.

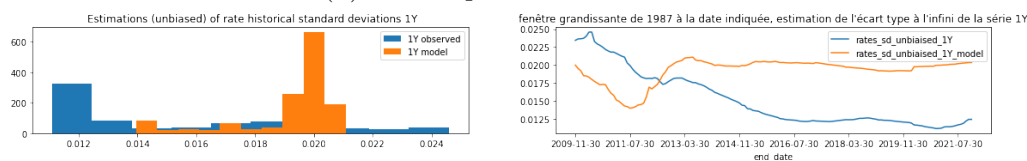
9.3. RÉSULTATS DE L'IMPACT DES CIBLES HISTORIQUES SUR LA VIF, EN FONCTION DU CONTEXTE DE TAUX



(a) Historique des cibles de corrélations



(b) Historiques de cibles de vol



(c) Historique des cibles sd

FIGURE 9.1 – Évolution historique des cibles distributionnelles.

Cette étude historique permet de savoir quels paramètres $\hat{\Theta}_d$ associer à un choc de $x\%$ sur une cible donnée. L'amplitude de chocs que nous appliquons aux cibles distributionnelles est donc :

- Correl : [-30%,+20%]
- Sd : [-20%,+80%]
- vol : [-10%, +40%]

9.3 Résultats de l'impact des cibles historiques sur la VIF, en fonction du contexte de taux

Les figures 9.3, 9.4 et 9.2 présentent respectivement des statistiques descriptives sur l'impact que peut avoir un choc sur les cibles de corrélation, de standard deviation et de corrélation sur la VIF. Nous remarquons que cet impact va varier d'une date à l'autre et qu'il n'est pas a priori une fonction linéaire du choc.

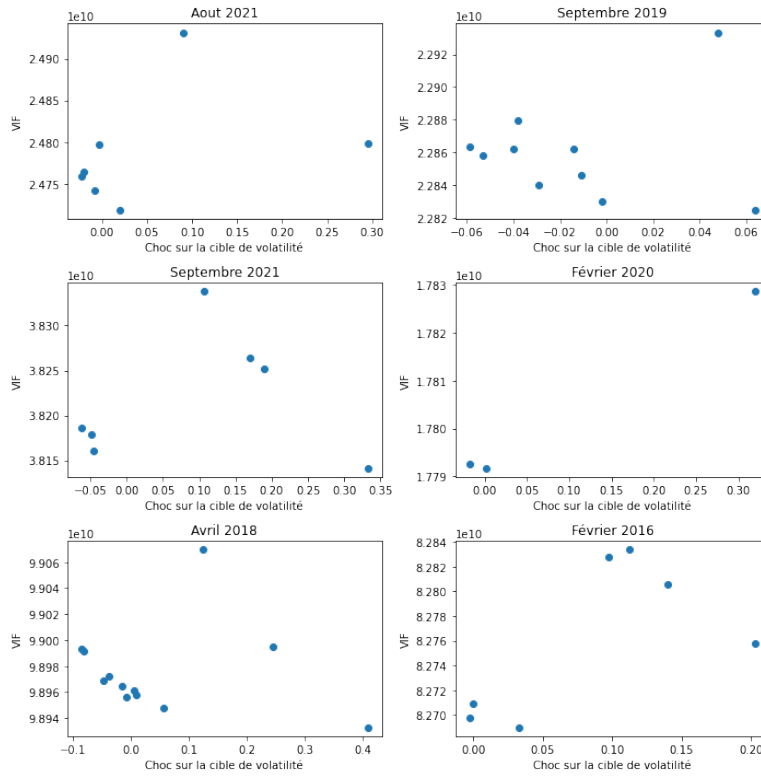


FIGURE 9.2 – Choc sur la cible de volatilité

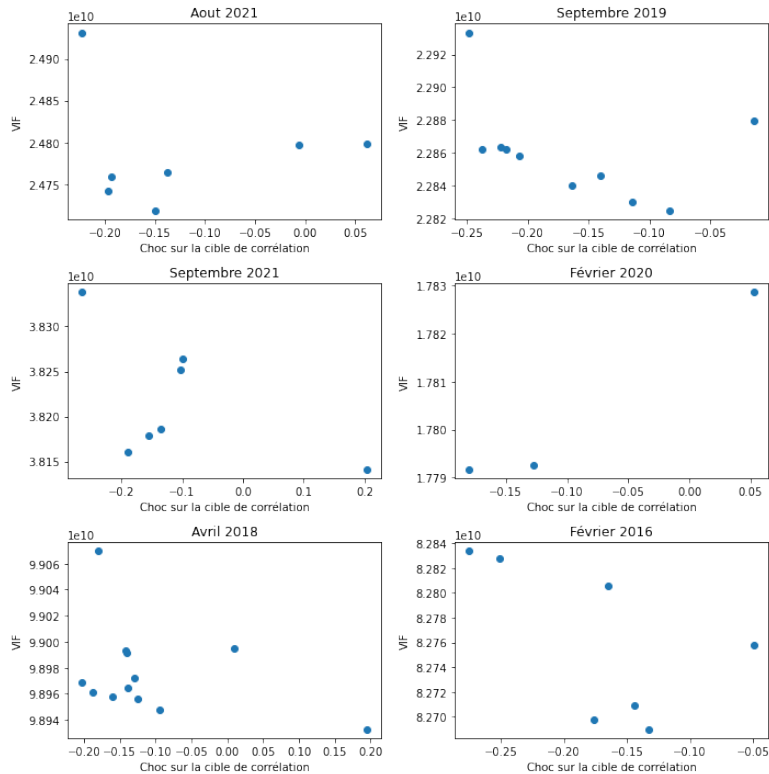


FIGURE 9.3 – Choc sur la cible de corrélation

9.3. RÉSULTATS DE L'IMPACT DES CIBLES HISTORIQUES SUR LA VIF, EN FONCTION DU CONTEXTE DE TAUX

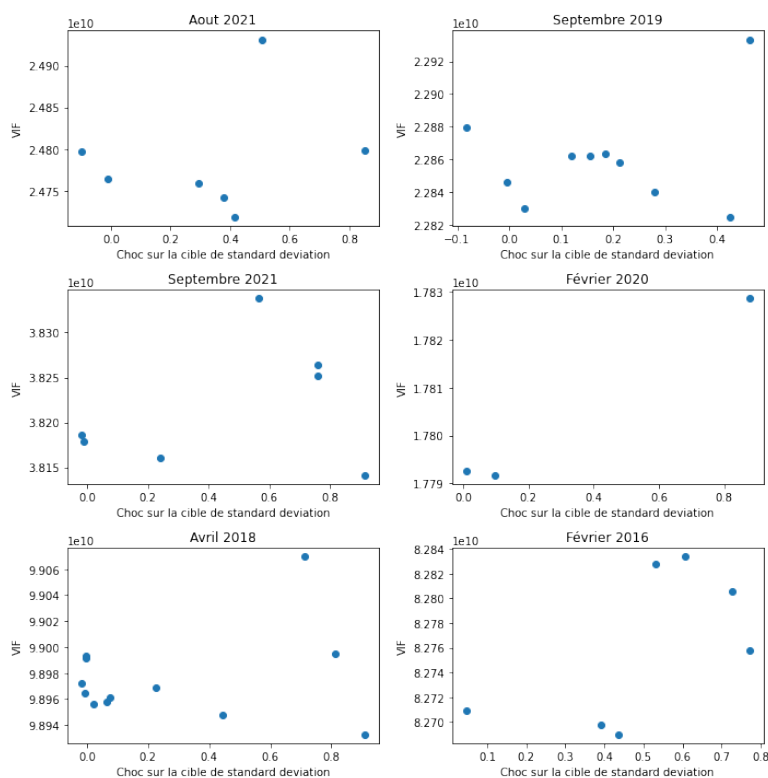


FIGURE 9.4 – Statistiques descriptives, choc sur la cible de standard deviation

L'utilisation de la méthode habituelle permet de centrer plus facilement les données afin de les comparer. Naturellement le choix du point de référence porte sur un choc de **0%** dans tous les cas. La figure 9.5 permet de visualiser la valeur de la VIF interpolée pour un choc de 0% sur la cible de standard deviation (point en rouge).

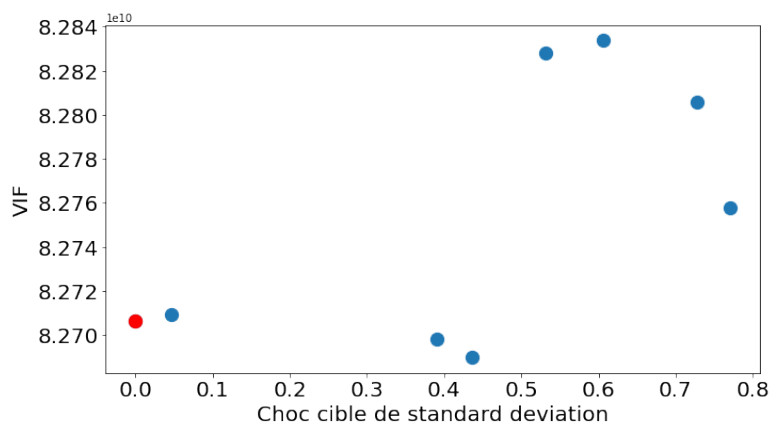


FIGURE 9.5 – Interpolation, exemple

Dés lors, il est possible d'afficher sur un même graphique les différentes dates, en centrant chaque date par rapport au point de référence. Cela permet d'une part de visualiser les variations de la VIF dues au choc, et d'autre part cela permet de comparer l'impact des différentes données de marché. 9.6 permet de visualiser les variations de la VIF en fonction de la valeur du choc sur la cible de corrélation.

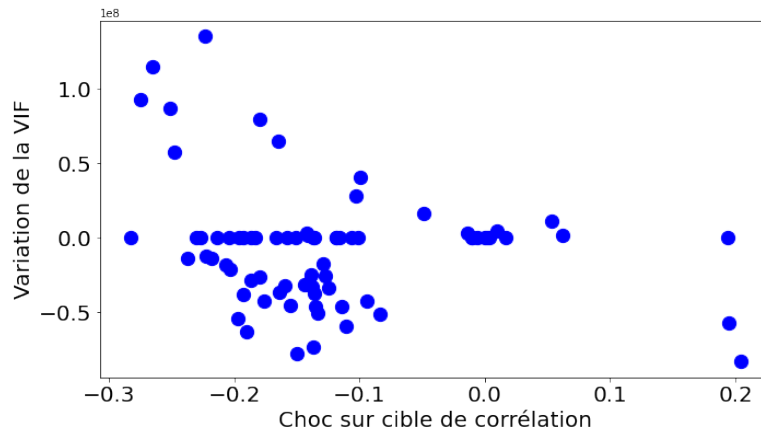


FIGURE 9.6 – Choc sur la cible de corrélation, VIF centrée

Afin de mieux comprendre l'impact des volatilité de swaptions, nous distinguons 9.7 les différentes positions sur le premier axe principal du VAE Swaptions.

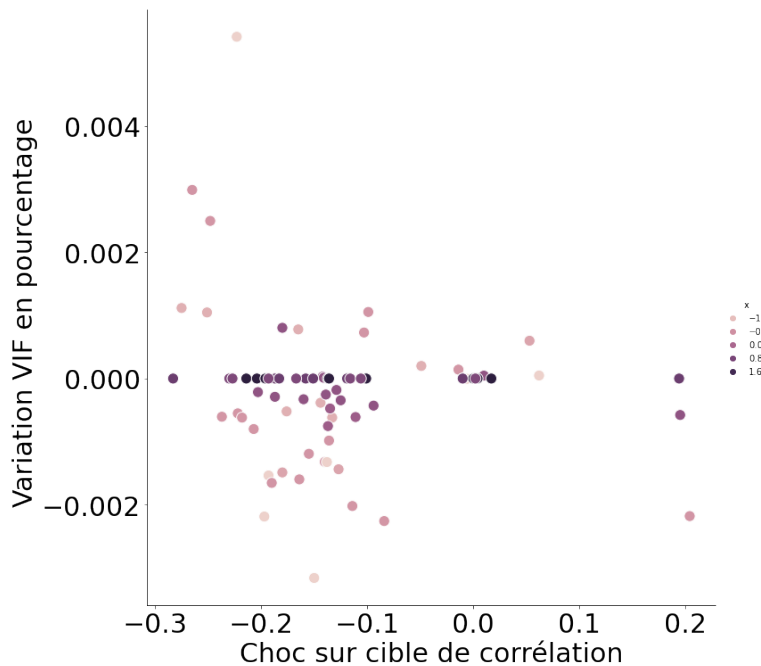


FIGURE 9.7 – Choc sur la cible de corrélation

Il apparaît clairement que dans un contexte de volatilité de swaptions élevées, l'impact du choc est moindre. Les paramètres embarqués du GLMM jouent un rôle légèrement plus important dans un contexte de taux bas. La même étude peut être menée pour la cible de volatilité. La figure 9.8 présente la valeur de la VIF centrée par rapport à un choc sur la cible de volatilité de 0%.

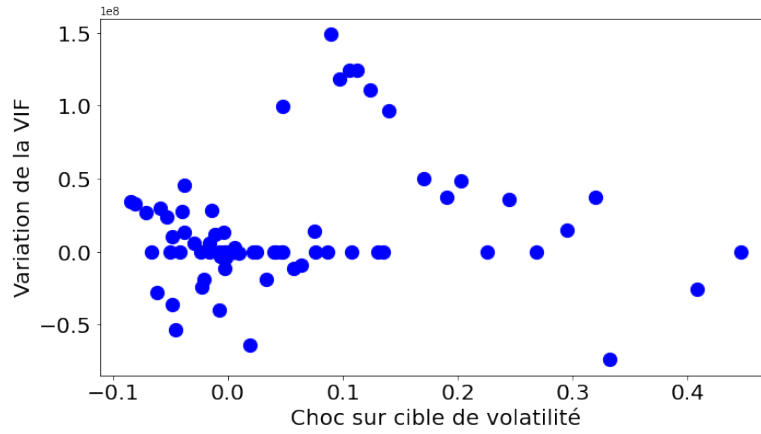


FIGURE 9.8 – Choc sur la cible de volatilité, VIF centrée

De même, il est possible de discriminer les valeurs à la lumière des niveaux de volatilité de swaptions, voir la figure 9.7.

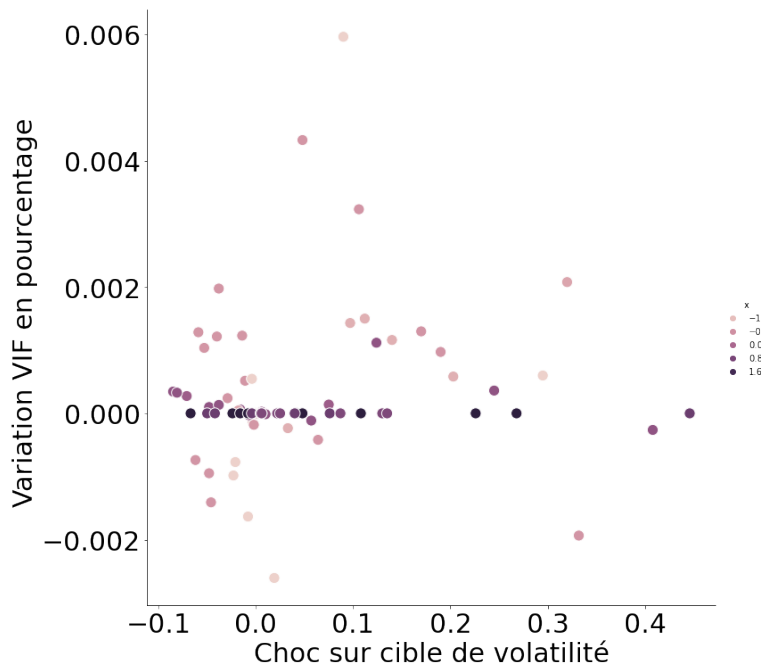


FIGURE 9.9 – Choc sur la cible de volatilité

Les conclusions sont identiques à celles que l'on a pu faire sur les corrélations. Le comportement est remarquablement différent en fonction des volatilités de swaptions, et il n'y a aucun impact de ces paramètres dès lors que le marché évolue dans un contexte de volatilité de swaptions élevé.

Le dernier point qu'il peut être intéressant de regarder est l'évolution de la cible de standard déviation. La figure 9.10 présente directement le résultat, en gardant les mêmes conventions que celles utilisées précédemment.

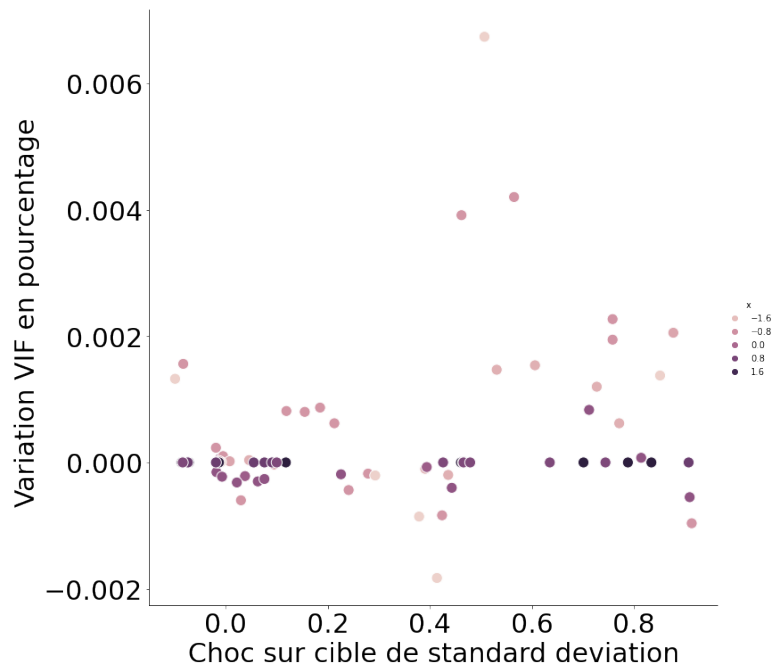


FIGURE 9.10 – Choc sur la cible de standard deviation

Conclusion L'impact des paramètres embarqués du GLMM est négligeable sur les valeurs de la VIF. En les modifiant, la VIF peut varier de plus ou moins 0.1%, ce qui est un ordre de grandeur extrêmement faible. La méthodologie que nous avons déroulé permet de mettre en lumière que le comportement des paramètres embarqués varie en fonction du contexte de volatilité de swaptions. Ainsi, nous avons identifié que lorsque les volatilité de swaptions sont élevées, ces paramètres embarqués n'ont absolument aucun impact sur la VIF. Dans le chapitre suivant, nous nous intéressons de plus prêt à la matrice de corrélations.

Chapitre 10

Matrice de corrélations

Dans ce chapitre, l'accent est mis sur l'étude du stress de la matrice de corrélations de Milliman ESG, qui reflète les relations entre divers facteurs de risque. Lorsque nous souhaitons lui appliquer un choc, il est crucial de garantir que la matrice reste semi-définie positive. Si ce n'est pas le cas, une procédure de retraitement de Higham est mise en oeuvre par défaut. Ce retraitement risque de perturber l'ensemble des corrélations, ce que nous souhaitons éviter lors d'une analyse de sensibilité, afin de contrôler précisément les corrélations que nous souhaitons affecter. Dans cette étude, nous proposons une méthodologie permettant de stresser la matrice de corrélations tout en maintenant sa semi-définie positivité. Nous avons opté pour une approche théorique pour résoudre ce problème, tout en gardant à l'esprit que nous aurions également pu examiner les corrélations historiques entre les différents facteurs de risque. Cependant, cela n'a pas été retenu dans le cadre de cette étude de sensibilité.

10.1 Création de la matrice de corrélations

Dans Milliman CHESS, la première étape de la simulation est de générer des mouvements Browniens. Pour modéliser les dépendances entre les différents facteurs de risques, une copule Gaussienne basée sur une matrice de corrélation estimée sur l'historique d'une série temporelle a été choisie. L'estimation de la matrice de corrélation ESG sans risque, qui est effectuée en trois étapes principales :

- Estimation des corrélations cibles entre les facteurs de risque observables basées sur des séries temporelles historiques,
- Dérivation de l'expression analytique de la corrélation entre les facteurs de risque observables en fonction de la corrélation entre les mouvements browniens et des paramètres des modèles,
- Inversion des formules établies à l'étape 2 afin d'estimer les paramètres de corrélation. Dans le cas où la matrice résultante n'est pas semi-définie positive, un ajustement est effectué lors de cette dernière étape.

Les drivers économiques simulées dans Milliman CHESS sont résumés dans la table ci-

dessous :

Economic driver	Model	Number of factors
Nominal rates	DDSVLMM	2
Real rates	2FV	2
Equity index	BSDV	1
Equity index	SVJD	1
Property index	BSDV	1
Credit	JLT	1
FX rate	GKDV	1

Le tableau ci-dessous présente les facteurs de risques observables utilisés dans la calibration de la matrice de corrélation.

Facteur simulé	Variable associée observée	Facteur de risque observé
Taux Forward $F_j(t)$	Taux nominal continûment composé $(R^n(t, m) = -\frac{1}{m} \ln(P^n(t, m)))_{t, m}$	Variation du taux nominal continûment composé $(R^n(t, m) - R^n(t - \Delta, m))_{t, m}$
Taux court réel r_t	Taux réel composé continûment $(R^r(t, m) = -\frac{1}{m} \ln(P^r(t, m)))_{t, m}$	Variations du taux réel continûment composé $(R^r(t, m) - R^r(t - \Delta, m))$
Index Actions $S(t)$	Index Actions $(S(t))_t$	Log-rendements de l'index actualisé $(\ln(\frac{S(t)P^n(t-\Delta, \Delta)}{S(t-\Delta)}))$
Immobilier $RE(t)$	Indice immobilier $(RE(t))_t$	Log-rendements de l'indice actualisé $(\ln(\frac{RE(t)P^n(t-\Delta, \Delta)}{RE(t-\Delta)}))$
Prime de risque $\pi(t)$	log-spread de Crédit $(Spd(t, m))_{t, m}$ $= -\frac{1}{m} \ln(\frac{P^C(t, m)}{P^n(t, m)})$	Variations des log-spreads $(Spd(t, m) - Spd(t - \Delta, m))_{t, m}$
Taux FX $X(t)$	Taux d'échange $(X(t))_t$	Log-rendements des taux FX actualisés $(\ln(\frac{X(t)P_d^n(t-\Delta, \Delta)}{X(t-\Delta)P_f^n(t-\Delta, \Delta)}))$

À partir de ce tableau et des dynamiques des différents modèles, nous sommes capable de suivre le processus que nous avons donné en amont et estimer les cibles de corrélations. Pour étudier par la suite l'importance des différentes corrélations, un certain nombre de problèmes relativement compliqués se posent.

Comment garantir que la matrice reste semi-définie positive lorsqu'elle est soumise à un choc ? Comment éviter le retraitement de Higham afin de conserver un contrôle précis sur les corrélations impactées ?

Il peut être difficile de visualiser et d'analyser une matrice de corrélations de cette taille. [29] souligne qu'il est possible de considérer une matrice de corrélation comme la matrice d'adjacence d'un graphe non orienté¹, où les nœuds représentent les facteurs de risque et les arêtes représentent les corrélations dépassant un certain seuil. La figure 10.1 ne connecte que les nœuds dont les facteurs de risque associés sont corrélés à au moins +/-20%. Les corrélations positives sont représentées par des flèches vertes, tandis que les corrélations négatives sont représentées par des flèches rouges.

1. le graphe est non orienté car la matrice de corrélations est symétrique

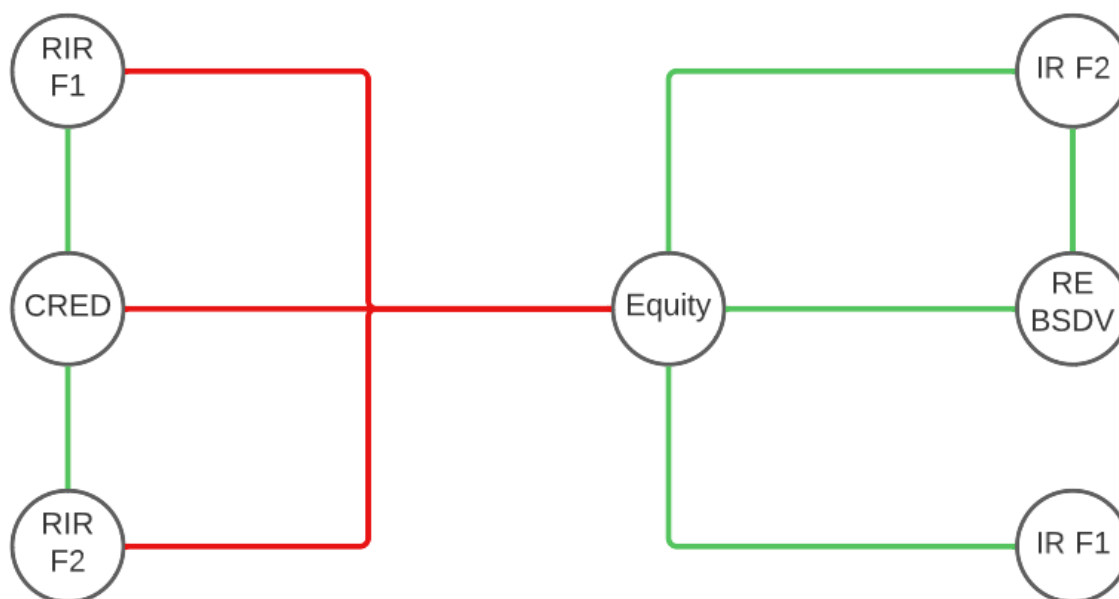


FIGURE 10.1 – Les sommets dont les corrélations sont de plus ou moins 20% entre eux sont reliés par des arrêtes. Nous désignons les termes *facteur 1*, *facteur 2* par F1 et F2. Pour les autres notations, nous pouvons nous référer directement au tableau présenté ci-dessus

Le schéma fournit plusieurs observations intéressantes :

1. Pratiquement tous les facteurs de risque présentent une forte connexion avec le facteur de risque **Equity**,
2. Deux grands groupes de facteurs de risque semblent se démarquer : ceux qui ont une corrélation positive avec **Equity** et ceux qui ont une corrélation négative avec **Equity**. Plus précisément, nous observons que les connexions négatives correspondent aux RIR, tandis que les corrélations positives sont davantage liées aux IR et RE.

Par avis d'expert, et suite à cette analyse visuelle, nous souhaitons appliquer un choc et étudier les paires Equity-IR F2 et Equity- IR F1. Toutefois, il reste à déterminer l'amplitude maximale du choc que nous pouvons appliquer et à comprendre les effets lorsque plusieurs facteurs de risque sont simultanément choqués. Dans la suite, nous présenterons une technique spécifique, tout en gardant à l'esprit que ce domaine est ouvert et qu'il existe plusieurs méthodes disponibles. Il convient notamment de noter l'article de [12], qui présente de nombreuses approches.

10.2 Range pour stresser les paramètres

10.2.1 Définitions

Dans un premier temps il faut voir dans quel range nous pouvons stresser les paramètres de manière à conserver une matrice de corrélations, et donc d'éviter le retraitement et la modification de toutes les corrélations. Une bonne intuition sur les coefficients de corrélations peut être retrouvée dans [26]. Cela est particulièrement utile lorsque nous travaillons sur une preuve.

Definition 10.2.1. Soit M une matrice symétrique réelle d'ordre n . Elle est dite définie positive si elle est positive et inversible, autrement dit si elle vérifie l'une des propriétés équivalentes suivantes :

1. Pour toute matrice colonne non nulle \mathbf{x} à n éléments réels :

$$\mathbf{x}^\top M \mathbf{x} > 0,$$

autrement dit, la forme quadratique définie par M est strictement positive pour $\mathbf{x} \neq 0$,

2. Toutes les valeurs propres de M (qui sont nécessairement réelles) sont strictement positives.

Définition 10.2.2. La matrice de corrélation d'un vecteur de p variables aléatoires $\vec{X} = \begin{pmatrix} X_1 \\ \vdots \\ X_p \end{pmatrix}$ dont chacune possède une variance (finie), est la matrice carrée définie par :

$$\rho_{i,j} = \text{Cor}(X_i, X_j),$$

Les termes diagonaux de cette matrice sont égaux à 1, elle est symétrique, semi-définie positive et ses valeurs propres sont positives ou nulles.

Dans le cas de la gestion de portefeuille, nous avons par exemple que si $\Omega \in S^n$ est la matrice de corrélation des différents rendements, alors elle est semi-définie positive. Autrement dit, la variance de n'importe quel portefeuille est non-négative :

$$\forall W, V, \quad \sigma_P^2 = Z^\top \Omega Z \geq 0$$

$Z = W \odot V = [\omega_1 v_1, \dots, \omega_n v_n]$, ω_i le poids et v_i le rendement du i ème actif.

Pour toute la suite, désignons la matrice de corrélation par $\Omega = (\Delta \rho_{ij}) \in \mathbb{R}^{n \times n}$, qui est semi-définie positive.

Theorem 10.2.1 (Cholesky). Soit A une matrice symétrique définie positive. Alors il existe une matrice triangulaire inférieure L telle que $A = L^t L$. De plus, si nous imposons aux coefficients diagonaux de L d'être positifs, cette factorisation est unique.

Cette décomposition est cruciale pour la suite de nos travaux. Nous voulons apporter une précision au théorème que nous venons d'énoncer.

Soit L la matrice de décomposition de Cholesky triangulaire inférieure de Ω telle que $\Omega = LL^T$ et soit l_i les colonnes de L telles que $L = (l_1, \dots, l_n)$. Comme L est une matrice triangulaire inférieure, les premières entrées $i - 1$ de l_i sont des zéros. La décomposition de Cholesky peut donc être exprimée d'une deuxième manière qui est :

$$\Omega = \sum_{i=1}^n l_i l_i^T$$

Puisque les $i - 1$ premières entrées de l_i sont des zéros, les $i - 1$ premières lignes et colonnes de $l_i l_i^T$ sont des zéros.

10.2.2 Bornes possibles pour choc sur matrice de corrélation

Nous nous plaçons dans le cas particulier où nous voulons appliquer un choc sur la corrélation entre deux facteurs de risque i et j . Nous nous plaçons dans le cadre donné par [56]. L'idée élégante donnée par cet article, repose sur le fait que changer l'ordre des facteurs de risque qui spécifient la matrice de corrélation n'affecte pas le caractère défini positif. Nous allons réordonner la matrice pour appliquer un choc sur la décomposition de Cholesky. Initialement, Ω est la matrice de corrélation des facteurs (F_1, \dots, F_n) . Nous réordonnons nos facteurs en

$(F_1, \dots, F_{i-1}, F_{i+1}, \dots, F_{j-1}, F_{j+1}, \dots, F_n, F_i, F_j)$ et nous notons $\tilde{\Omega} = (\Delta\tilde{\rho}_{ij})$ la matrice réordonnée. Nous avons donc placé tout à la fin de la matrice, les corrélations entre les facteurs i et j .

Soit $L = (l_1, \dots, l_n)$ la matrice de Cholesky triangulaire inférieure correspondante. Si un choc est appliquée aux éléments non nuls de l_{n-1} et l_n , elle affectera la sous-matrice en bas à droite de taille 2×2 de $\tilde{\Omega}$. En raison de notre réordonnement², cela correspond exactement à un choc sur les corrélations des facteurs i et j !

Il est ensuite possible de reconstruire la matrice et de réorganiser les facteurs pour les ramener à leur emplacement initial!

Pour être plus précis, désignons les vecteurs colonnes l_{n-1} et l_n comme :

$$l_{n-1} = (0, 0, \dots, 0, a, b)^T, \quad l_n = (0, 0, \dots, 0, 0, c)^T$$

Soit δ_a, δ_b , et δ_c des perturbations sur les termes a, b , et c respectivement. Après ces perturbations, nous obtenons

$$\hat{l}_{n-1} = (0, 0, \dots, 0, a + \delta_a, b + \delta_b)^T, \quad \hat{l}_n = (0, 0, \dots, 0, 0, c + \delta_c)^T$$

Alors :

$$L = (l_1, \dots, \hat{l}_{n-1}, \hat{l}_n)$$

et

$$\begin{aligned} \hat{\Omega} &= \hat{L}\hat{L}^T = \hat{l}_{n-1}\hat{l}_{n-1}^T + \hat{l}_n\hat{l}_n^T + \sum_{i=1}^{n-2} \hat{l}_i\hat{l}_i^T \\ &= \begin{pmatrix} 0 & \cdots & 0 & & \\ \vdots & & \vdots & & \\ \vdots & \ddots & 0 & & \\ 0 & 0 & (a + \delta_a)^2 & (a + \delta_a)(b + \delta_b) & \\ 0 & 0 & (a + \delta_a)(b + \delta_b) & (b + \delta_b)^2 (c + \delta_c)^2 & \end{pmatrix} + \sum_{i=1}^{n-2} l_i l_i^T \end{aligned}$$

Maintenant, nous voulons que des 1 sur la diagonale, pour conserver une matrice de corrélation. En plus nous voulons bien sûr avoir un choc le plus proche possible de celui cherché. Cela conduit à un problème de minimisation :

$$\begin{cases} \min_{\delta_a, \delta_b, \delta_c} (a\delta_b + b\delta_a + \delta_a\delta_b - \Delta p_{ij})^2 \\ \delta_a^2 + 2a\delta_a = 0, \delta_b^2 + 2b\delta_b + \delta_c^2 + 2c\delta_c = 0 \end{cases}$$

avec bien sûr Δp_{ij} le stress souhaité.

L'article [56] démontre le théorème suivant :

Theorem 10.2.2. *Soit $\Omega = (\rho_{ij}) \in \mathbb{R}^{n \times n}$ une matrice de corrélation. Supposons qu'un choc unique $\Delta\rho_{ij}$ doive être appliquée au terme de corrélation ρ_{ij} (et évidemment à ρ_{ji}). Soit $\tilde{\Omega}$ la matrice de corrélation réordonnée telle que les corrélations de S_i et S_j sont placées dans la sous-matrice inférieure droite 2×2 de $\tilde{\Omega}$. Soit $L = (l_1, \dots, l_{n-1}, l_n)$ une matrice triangulaire inférieure $n \times n$ telle que $\tilde{\Omega} = L^T$. Soit (a, b, c) défini comme précédemment. Si la contrainte $\Delta\rho_{ij}$ est dans l'intervalle ci-dessous*

$$\Gamma = \left[\min \left(-ab \left(1 \pm \sqrt{1 + \frac{c^2}{b^2}} \right) \right), \max \left(-ab \left(1 \pm \sqrt{1 + \frac{c^2}{b^2}} \right) \right) \right],$$

alors appliquer la contrainte directement sur Ω ne perturbe pas la définitude positive.

2. idée : une perturbation appliquée aux entiers non nuls de l_i affecte seulement la matrice de taille $(n - i + 1) \times (n - i + 1)$ de $\tilde{\Omega}$

Nous proposons donc de déterminer pour chaque terme de notre matrice le choc que l'on pourrait théoriquement lui appliquer en laissant inchangée la structure définie positive.

10.3 Résultats numériques

Nous trouvons comme intervalle possible pour les chocs

	IR F1	IR F2	RE BSDV
Equity	-96.1%/52.9%	-93.7%/46.1%	-67.8%/61.4%

Nous observons que nous pouvons potentiellement appliquer des chocs très importants tout en conservant le caractère semi-défini positif. En annexe, nous montrerons comment appliquer un choc à plusieurs paires de corrélation simultanément, mais dans le cadre de ce mémoire, nous avons choisi de ne pas aller plus loin au niveau de la matrice de corrélations. La figure 10.2 montre l'évolution de la valeur du BE en fonction du choc appliqué aux paires de corrélations Equity- IR F1 et Equity-IR F2. Les données de marché (swaptions) sont celles du 31/12/2021.

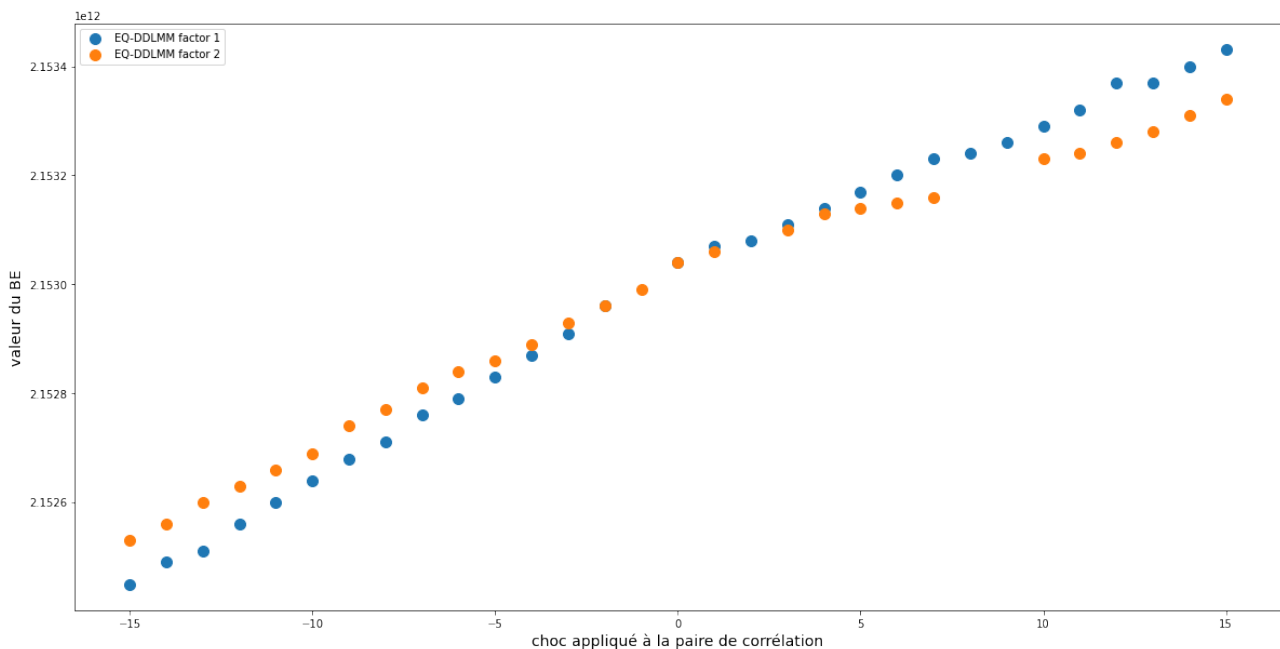


FIGURE 10.2 – Evolution du BE en fonction du choc matrice corrélation.

Il apparaît que la dépendance est linéaire, et qu'elle est relativement proche pour les deux paires de corrélations.

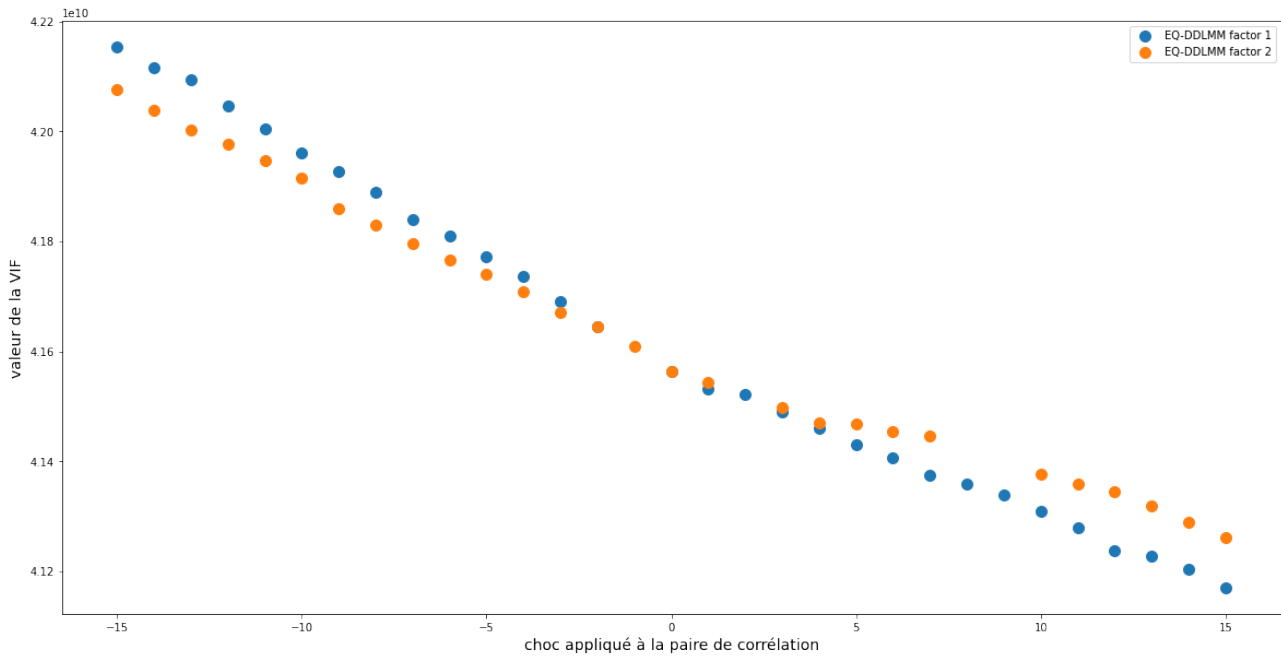


FIGURE 10.3 – Evolution de la VIF en fonction du choc matrice corrélation.

Nous pouvons ici faire le même commentaire.

Conclusion Faire l'analyse de sensibilité d'une matrice de corrélation demande une réflexion sur son retraitement, dans le cas où la matrice perdrait son caractère semi-défini positif. Notre étude s'est portée uniquement sur les paires de corrélations les plus importantes, choisies par avis d'expert, et nous renvoyons en annexe A pour l'étude simultanée de plusieurs paires de corrélation. Dans le chapitre suivant, nous proposerons d'étudier simultanément tous les paramètres embarqués que nous venons d'évoquer.

Chapitre 11

Hiérarchie des paramètres embarqués d'un GSE

Nous avons dans les chapitres qui précèdent passé en revue les principaux paramètres embarqués de notre générateur de scénarios économique. Il est clairement apparu que les données de volatilités de swaptions et la courbe des taux étaient celles qui avaient le plus d'impact sur le bilan de notre assureur, suivi des données de volatilité de call.

Nous aimerions alors savoir entre le shift, les cibles distributionnelles du GLMM et les corrélations de la matrice de corrélation, qui a le plus d'impact. Nous nous plaçons uniquement au 31/12/2021 pour cette étude, avec des données de marché fixées, et nous faisons l'hypothèse que l'impact reste globalement le même dans un contexte de données de marché différentes, avec les remarques que nous avons pu précédemment faire (variance plus importante de l'impact du GLMM si nous avons des taux faibles, shift plus impactant si nous avons des taux élevés). Nous faisons l'hypothèse que l'ordre d'importance reste le même, au vu des ordres de grandeurs que nous avons pu observer sur les chapitres précédents.

11.1 Résultats

Les premiers tests effectués sur les données ont consistés à calculer la valeur du VIF pour différentes valeurs de corrélation du facteur 2 DDLMM (proche de l'impact du facteur 1 DDLMM comme nous l'avons vu), du shift, et des chocs sur les cibles de standard déviation, de corrélation et de volatilité des paramètres embarqués du GLMM. La figure 11.1 permet de se faire une idée de l'impact des différents paramètres sur la VIF.

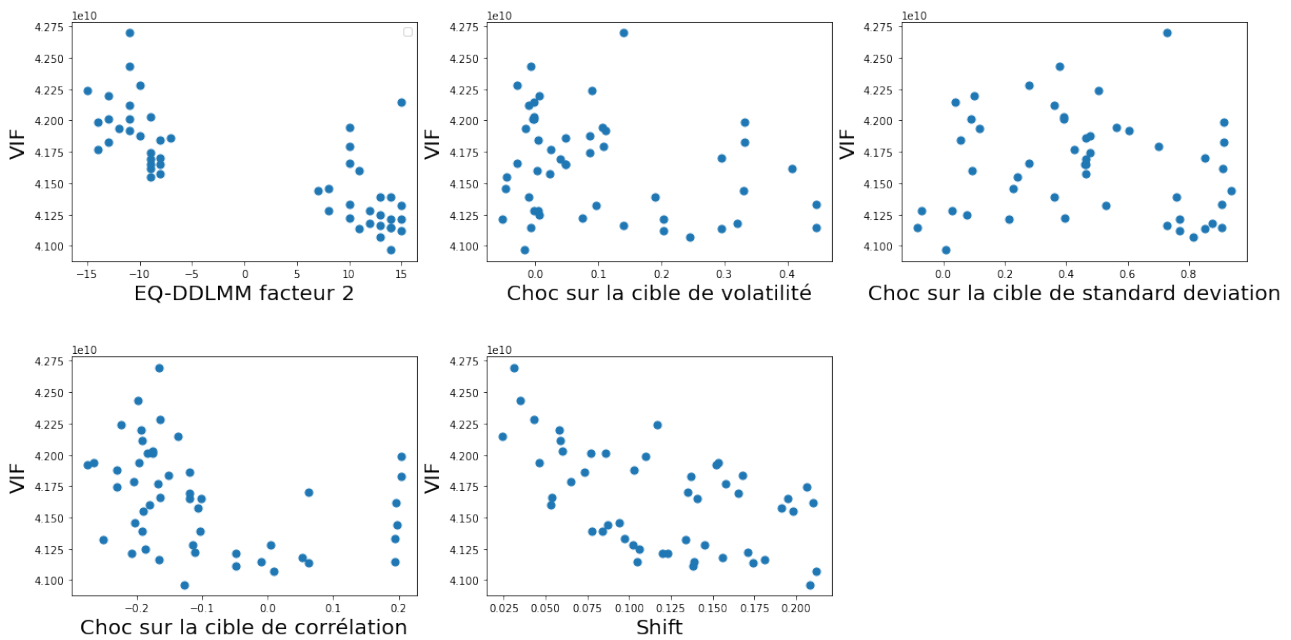


FIGURE 11.1 – Impacts sur la VIF

Lorsque nous regardons la figure 11.1 de plus près, nous nous rendons compte que les chocs sur les cibles de volatilité, de standard déviation ou de corrélation n'ont pas d'impact particulier sur la valeur de la VIF. Nous en déduisons que les paramètres embarqués n'ont aucun impact sur la valeur de la VIF. Nous pouvons nous pencher plus précisément sur le rôle du shift et sur le rôle des corrélations en figure 11.2.

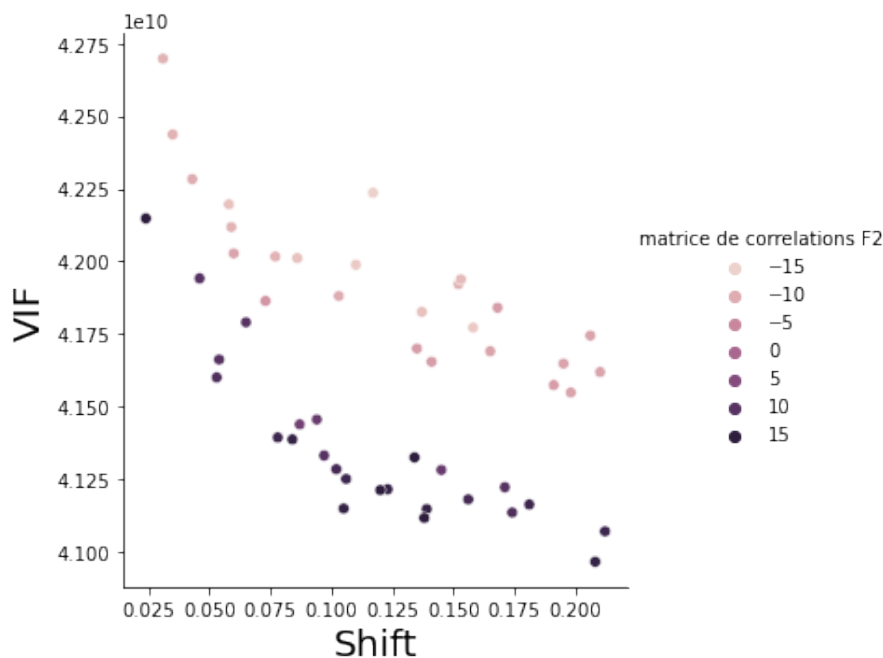


FIGURE 11.2 – Impacts sur la VIF, corrélations et shift

Nous réalisons ici que la VIF suit une distribution multimodale. Cela est créé directement par la modification de la valeur des corrélations. Nous réalisons ici que les corrélations ont elles aussi un impact très importante sur la VIF. Nous retrouvons notamment le fait qu'une corrélation négative entre Equity et DDLMM F2 conduit à une augmentation de la valeur de la VIF, et une corrélation positive à une diminution de celle-ci. D'un point de vue actuariel, cette

variation peut être expliquée de la manière suivante : lorsque la corrélation entre l'équity et le DDLMM F2 est négative, cela réduit la volatilité globale de la VIF. Les mouvements inverses des actions et des taux se compensent en partie, ce qui peut réduire les pertes potentielles pour la compagnie d'assurance et améliorer sa solvabilité perçue.

De plus, une corrélation négative entre les actions et les taux peut également contribuer à la diversification des actifs de la compagnie d'assurance. En répartissant ses investissements entre des actifs qui réagissent différemment aux conditions économiques, la compagnie peut réduire son risque global.

En revanche, une corrélation positive entre les actions et les taux signifierait que ces actifs évoluent généralement dans la même direction. Dans un tel scénario, une augmentation de la valeur de la VIF peut être associée à une corrélation négative entre les actions et les taux.

Conclusion Nous voyons visuellement que le shift est le paramètre embarqué le plus important, suivi directement des corrélations puis des paramètres embarqués du GLMM. L'impact de ces paramètres embarqués est réellement minime à comparer de l'impact des données de marché (volatilités et taux).

Cinquième partie

Extension de notre étude et autres
domaines d'applications possibles

Chapitre 12

Capacités et limites de nos outils pour anticiper le comportement des modèles dans des situations inédites

12.1 Générations de données de volatilité d'options sur Call

Notre travail sur les modèles génératifs (VAE) nous permet de générer de nouvelles données de marché. Tout se passe comme si nous étions capable de nous placer dans des contextes de marchés différents (d'un point de vue des volatilités de Swaptions et des volatilités d'options sur Call). Cela peut permettre de lancer de nouvelles sensibilité pour savoir, si par exemple, certains paramètres embarqués ont un comportement qui évolue dans un contexte différent.

Nous pouvons reprendre l'espace latent que nous avons obtenu pour les volatilités d'options (VAE Equity) en figure 12.1.

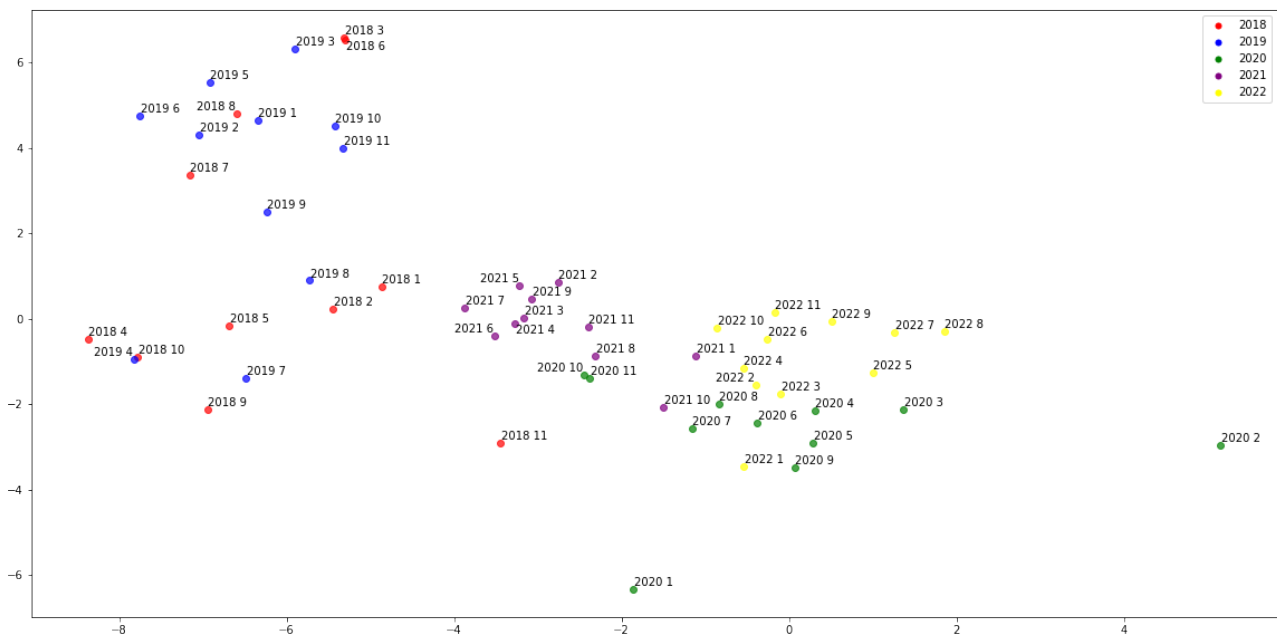


FIGURE 12.1 – Position des données de volatilité dans l'espace latent

Nous proposons d'utiliser les capacités génératives de notre modèle pour extraire des

contextes encore jamais rencontrés¹ :

1. **Un contexte de volatilité élevé** : nous souhaitons une volatilité relativement élevée, cependant moins que pendant le COVID, qui était une exception,
2. **Un contexte de volatilité faible** : l'objectif serait de générer des données de volatilité d'options Call faibles.

Cela nous conduit à échantillonner dans l'espace latent. La figure 12.2 indique les positions des points que nous avons choisi de décompresser².

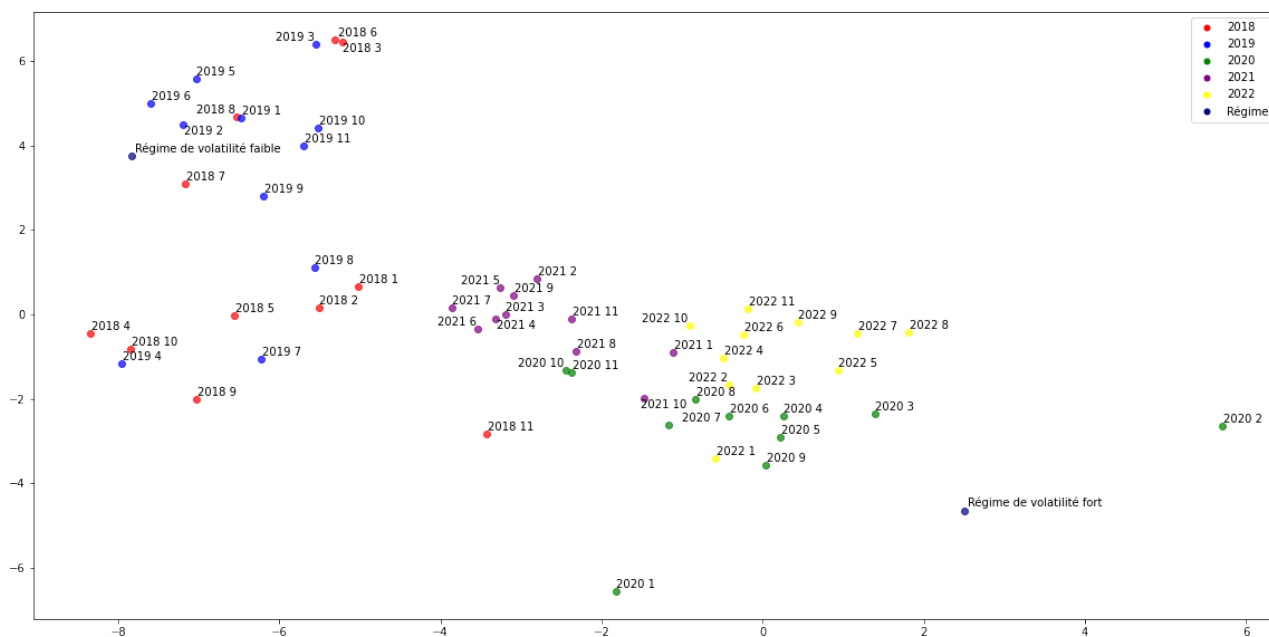


FIGURE 12.2 – Deux nouveaux points que nous allons décompresser

Suite à la décompression de ces nouveaux points, nous pouvons vérifier que les données générées sont vraisemblables. Pour cela il suffit de tracer les volatilités que l'on décomprime et de les comparer avec quelques dates de référence. La figure 12.3 montre des données décompressées ainsi que quelques données historiques observées. Nous avons en abscisse les maturités et en ordonnée les niveaux de volatilité sur ce graphique.

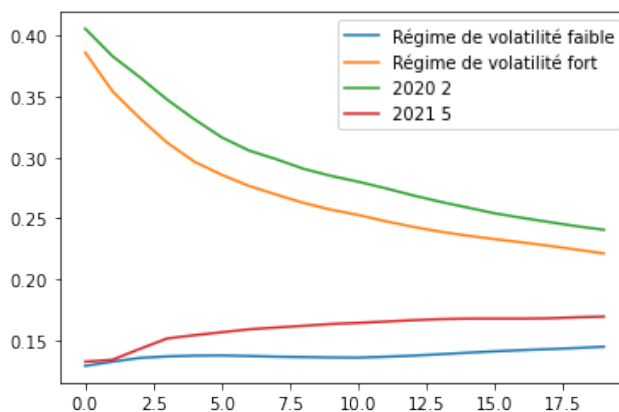


FIGURE 12.3 – Les données décompressées sont vraisemblables.

1. Il s'agit de décompresser des données de l'espace latent du VAE Equity
 2. Pour une analyse plus détaillée de l'espace latent, se référer au chapitre 6.

Nous avons caractérisé dans le chapitre 6 l'effet d'une modification des données de volatilité d'option Call. Au vu des résultats obtenus, et puisque nous ne changeons pas notre modèle ALM d'un calcul à l'autre, nous ne recalculerons pas les impacts VIF. La génération de données peut prendre tout son intérêt si nous décidons de modifier le modèle ALM. Dans ce cas, il peut être pertinent de prendre des données de marché très variables et potentiellement très grandes pour analyser la sensibilité à celles-ci.

12.2 Génération de volatilité de Swaptions

Nous cherchons ici aussi à générer de nouvelles données de volatilité de swaptions associées à un contexte encore jamais rencontré. La figure 12.4 montre les points que nous avons choisi de décompresser de l'espace latent VAE Swaption. Nous avons choisi ici aussi deux contexte de marché, taux élevés et taux faibles.

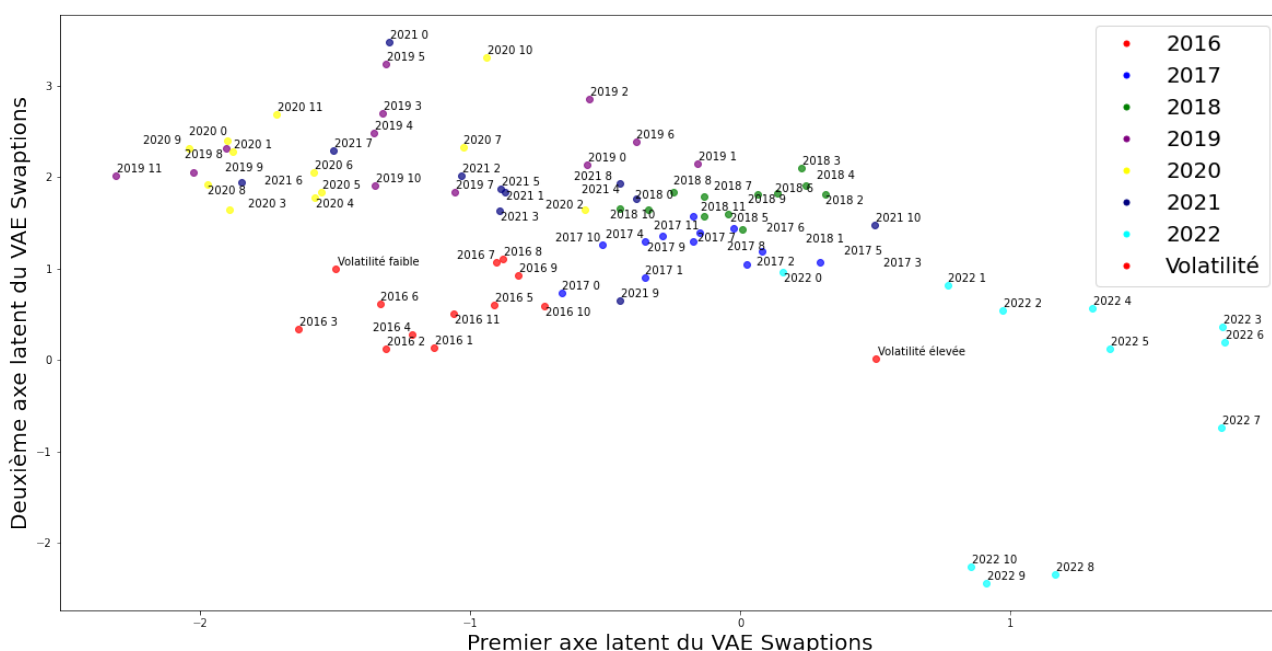


FIGURE 12.4 – Décompression de points VAE Swaptions.

Afin de fixer un peu plus les idées, nous proposons de visualiser une nappe de volatilité réelle (août 2022) et deux nappes de volatilités générées, correspondant aux points de la figure 12.4. La figure 12.5 nous permet de comparer les données générées avec des données réelles.

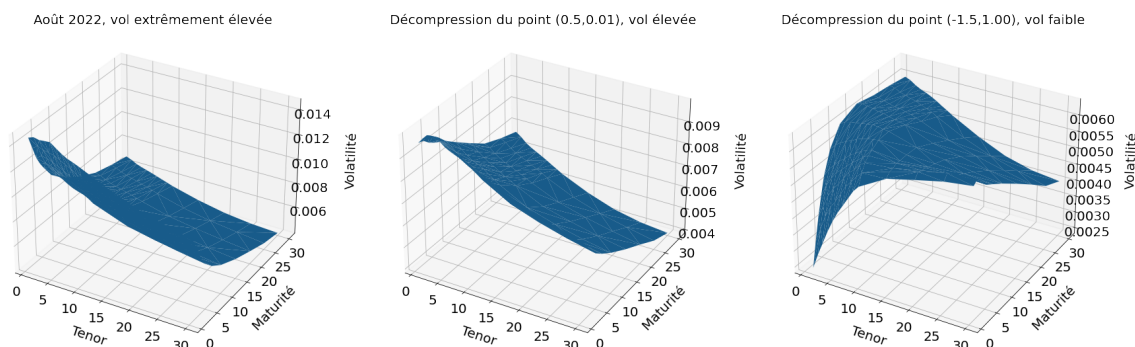


FIGURE 12.5 – Données générées par le VAE Swaptions.

Encore une fois, il peut être intéressant de générer des nouvelles données de marché pour voir comment le modèle peut se comporter dans un contexte de volatilité de swaption extrême. Ceci est particulièrement pertinent, notamment suite à une modification du modèle ALM.

Le modèle ALM va évoluer en fonction de la politique de l'assureur et des modifications effectuées par les équipes. Il paraît donc très compliqué de chercher à prédire la valeur de la VIF ou du BE simplement avec les données de marché. En revanche, nous pouvons plus simplement prédire les niveaux de taux en sortie du GSE à partir des données de marché. Ceux-ci ayant directement un impact sur la modélisation ALM, il est possible ensuite d'émettre un avis d'expert sur l'impact possible de données de marché sur la VIF.

12.3 Prédiction des sorties GSE dans un contexte nouveau

Afin d'appuyer nos propos précédents, nous proposons de chercher à prédire les taux minimums atteints par un zéro coupon de maturité 10 ans à 1 an de projection. Notre objectif ici est de montrer que la capacité à caractériser les données de marché permet de dire avec un modèle très simple quel sera l'impact des paramètres embarqués sur les sorties du GSE.

Nous générons de nouvelles données de marché de volatilité de swaption, nous faisons varier les valeurs prises par le shift, et nous construisons un modèle simple pour prédire la valeur des taux minimums atteints.

Les variables explicatives sont :

1. **Le shift,**
2. **Premier axe latent VAE Swaptions,**
3. **Deuxième axe latent VAE Swaptions.**

La variable à expliquer est le **taux minimum atteint à 1 an**. Nous construisons un simple modèle de régression linéaire.

Nous représentons dans le tableau ci-dessous le coefficient R2 de la régression obtenu en fonction des variables utilisées.

		<i>Shift</i>	<i>Axe1</i>	<i>Axe2</i>	Toutes les variables
R2	Régression	0.06%	52%	23%	57%

Nous réalisons que la prise en compte des données de marché permet de prédire largement mieux que le shift seul. Cela reste en cohérence avec ce que nous avons présenté dans ce mémoire plus haut. La figure 12.6 montre les points prédits en utilisant les différentes variables. Nous réalisons que cela est relativement proche des observations.

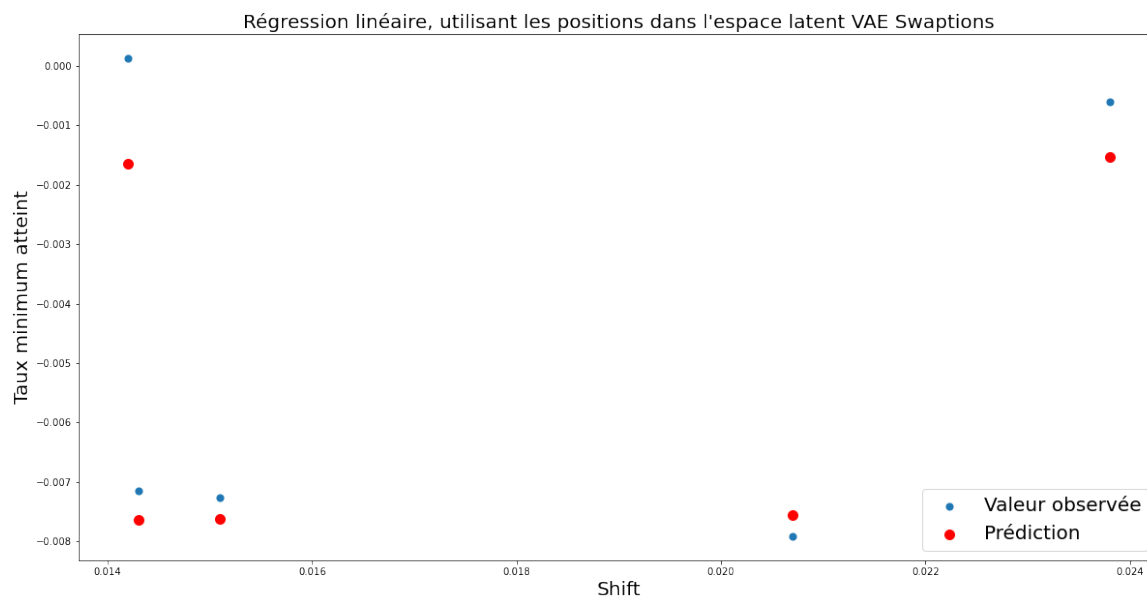


FIGURE 12.6 – Régression linéaire.

La figure 12.7 montre le résultat de la régression si nous ne tenons pas compte des données de marché.

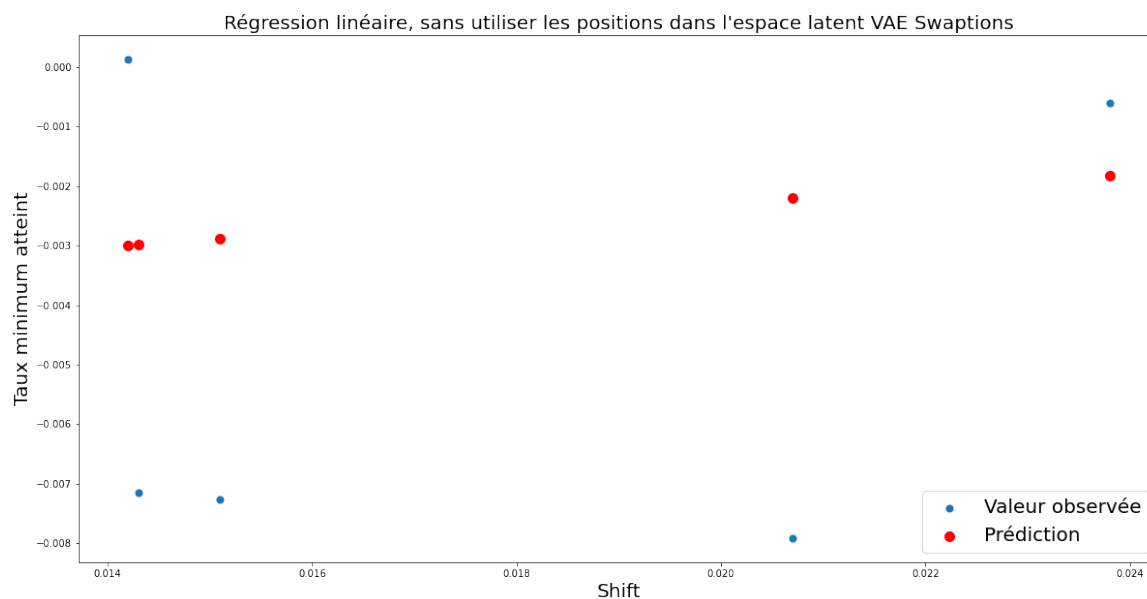


FIGURE 12.7 – Régression linéaire sans tenir compte des données de marché.

Les résultats sont nettement moins bons.

Conclusion Nous pourrions prédire d'autres grandeurs en sortie d'un générateur de scénario économique avec ce type de méthodologie. Cela peut conduire les experts à affiner leurs anticipations sur les résultats ALM, car, nous l'avons montré dans le mémoire, les résultats ALM dépendent fortement des sorties GSE.

Conclusion

Les générateurs de scénarios économiques comprennent des modèles avec de nombreux paramètres intégrés. L'objectif de cette étude était d'évaluer l'importance relative de ces paramètres et de les classer les uns par rapport aux autres.

Nous avons constaté que les autoencodeurs variationnels bayésiens peuvent être utilisés pour caractériser le contexte du marché, ce qui nous permet de comprendre comment l'importance des paramètres embarqués peut varier en fonction des données du marché.

Nos résultats indiquent que les paramètres embarqués ont un impact très limité sur les différentes métriques que nous avons définies. Cette observation s'explique en partie par les fortes fluctuations des données de marché, qui entraînent des variations brusques.

Cependant, il peut être utile pour les praticiens de prendre en compte les conclusions de cette étude, car elles fournissent des indications sur l'impact des différents paramètres embarqués et sur la manière dont les données de marché affectent les résultats d'un générateur de scénarios économiques.

Bibliographie

- [1] Umberto ALIBRANDI, Lars V ANDERSEN et Enrico ZIO. “Mutual information for global sensitivity analysis and adaptive-learning surrogate modelling”. In : *Proceedings of the REC2021, International Workshop on Reliable Engineering Computing, Taormina, Italy*. 2021, p. 17-20.
- [2] Néstor BARRAZA et al. “Mutual information and sensitivity analysis for feature selection in customer targeting : A comparative study”. In : *Journal of Information Science* 45.1 (2019), p. 53-67.
- [3] Nelson HF BEEBE. “A Bibliography of Publications of George Marsaglia”. In : (2004).
- [4] Mohamed Ishmael BELGHAZI et al. “Mutual information neural estimation”. In : *International conference on machine learning*. PMLR. 2018, p. 531-540.
- [5] David BLACKMAN et Sebastiano VIGNA. “Scrambled linear pseudorandom number generators”. In : *ACM Transactions on Mathematical Software (TOMS)* 47.4 (2021), p. 1-32.
- [6] Damiano BRIGO, Fabio MERCURIO et al. *Interest rate models : theory and practice*. T. 2. Springer, 2001.
- [7] H CARLEY et MD TAYLOR. “A new proof of Sklar’s theorem”. In : *Distributions with given marginals and statistical modelling* (2002), p. 29-34.
- [8] Gang CHEN. “A gentle tutorial of recurrent neural network with error backpropagation”. In : *arXiv preprint arXiv :1610.02583* (2016).
- [9] Guillermo COTRINA, Alberto PEINADO et Andrés ORTIZ. “Gaussian Pseudorandom Number Generator Using Linear Feedback Shift Registers in Extended Fields”. In : *Mathematics* 9.5 (2021), p. 556.
- [10] Antonia CRESWELL et al. “Generative adversarial networks : An overview”. In : *IEEE signal processing magazine* 35.1 (2018), p. 53-65.
- [11] Alaeddine FALEH, Frédéric PLANCHET et Didier RULLIÈRE. “Les générateurs de scénarios économiques : de la conception à la mesure de la qualité”. In : *Assurances et gestion des risques* 78.1 (2010), p. 1-30.
- [12] Roza GALEEVA et al. “Measuring correlation risk”. In : *publicly available manuscript* (2007).
- [13] Hiroshi HARAMOTO et al. “Efficient Jump Ahead for 2-Linear Random Number Generators”. In : *INFORMS Journal on Computing* 20.3 (2008), p. 385-390.
- [14] Nicholas J HIGHAM. “Computing a nearest symmetric positive semidefinite matrix”. In : *Linear algebra and its applications* 103 (1988), p. 103-118.
- [15] Sepp HOCHREITER et Jürgen SCHMIDHUBER. “Long short-term memory”. In : *Neural computation* 9.8 (1997), p. 1735-1780.
- [16] Yu-Jay HUOH. *Sensitivity analysis of stochastic simulators with information theory*. University of California, Berkeley, 2013.

-
- [17] Ronald L IMAN et Stephen C HORA. “A robust measure of uncertainty importance for use in fault tree system analysis”. In : *Risk analysis* 10.3 (1990), p. 401-406.
- [18] Robin AA INCE et al. “A statistical framework for neuroimaging data analysis based on mutual information estimated via a gaussian copula”. In : *Human brain mapping* 38.3 (2017), p. 1541-1573.
- [19] Bertrand IOOSS et Andrea SALTELLI. “Introduction to sensitivity analysis”. In : *Handbook of uncertainty quantification*. Springer, 2017, p. 1103-1122.
- [20] Tim JANKE, Mohamed GHANMI et Florian STEINKE. “Implicit generative copulas”. In : *Advances in Neural Information Processing Systems* 34 (2021), p. 26028-26039.
- [21] Harry JOE. *Dependence modeling with copulas*. CRC press, 2014.
- [22] Diederik P KINGMA et Max WELING. “Auto-encoding variational bayes”. In : *arXiv preprint arXiv :1312.6114* (2013).
- [23] Donald E KNUTH. *Art of Computer Programming, Volume 4, Fascicle 4, The : Generating All Trees—History of Combinatorial Generation*. Addison-Wesley Professional, 2013.
- [24] Erik KOLE, Kees KOEDIJK et Marno VERBEEK. “Selecting copulas for risk management”. In : *Journal of Banking & Finance* 31.8 (2007), p. 2405-2423.
- [25] Pierre L’ECUYER et François PANNETON. “F2-linear random number generators”. In : *Advancing the Frontiers of Simulation : A Festschrift in Honor of George Samuel Fishman*. Springer, 2009, p. 169-193.
- [26] Joseph LEE RODGERS et W Alan NICEWANDER. “Thirteen ways to look at the correlation coefficient”. In : *The American Statistician* 42.1 (1988), p. 59-66.
- [27] Nunzio A LETIZIA et Andrea M TONELLO. “Copula Density Neural Estimation”. In : *arXiv preprint arXiv :2211.15353* (2022).
- [28] Qiao LIU et Toshimitsu HOMMA. “A new computational method of a moment-independent uncertainty importance measure”. In : *Reliability Engineering & System Safety* 94.7 (2009), p. 1205-1211.
- [29] Marcos LOPEZ DE PRADO. “Building Diversified Portfolios That Outperform Out-of-Sample (Presentation Slides)”. In : *Available at SSRN 2713516* (2016).
- [30] Jian MA et Zengqi SUN. “Mutual information is copula entropy”. In : *Tsinghua Science & Technology* 16.1 (2011), p. 51-54.
- [31] Michael MASCAGNI et al. “Parallel pseudorandom number generation using additive lagged-Fibonacci recursions”. In : *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing : Proceedings of a conference at the University of Nevada, Las Vegas, Nevada, USA, June 23–25, 1994*. Springer. 1995, p. 263-277.
- [32] Makoto MATSUMOTO et Takuji NISHIMURA. “A nonempirical test on the weight of pseudorandom number generators”. In : *Monte Carlo and Quasi-Monte Carlo Methods 2000 : Proceedings of a Conference held at Hong Kong Baptist University, Hong Kong SAR, China, November 27–December 1, 2000*. Springer. 2002, p. 381-395.
- [33] Makoto MATSUMOTO et Takuji NISHIMURA. “Mersenne twister : a 623-dimensionally equidistributed uniform pseudo-random number generator”. In : *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 8.1 (1998), p. 3-30.
- [34] BD MCCULLOUGH. “Microsoft Excel’s ‘not the Wichmann–Hill’ random number generators”. In : *Computational Statistics & Data Analysis* 52.10 (2008), p. 4587-4593.
- [35] NM MISHACHEV. “Backpropagation in matrix notation”. In : *arXiv preprint arXiv :1707.02746* (2017).

- [36] Alireza NAMDARI et Zhaojun LI. “A review of entropy measures for uncertainty quantification of stochastic processes”. In : *Advances in Mechanical Engineering* 11.6 (2019), p. 1687814019857350.
- [37] Harald NIEDERREITER. “Nonlinear methods for pseudorandom number and vector generation”. In : *Simulation and Optimization : Proceedings of the International Workshop on Computationally Intensive Methods in Simulation and Optimization held at the International Institute for Applied Systems Analysis (IIASA), Laxenburg, Austria, August 23–25, 1990*. Springer. 1992, p. 145-153.
- [38] Melissa E O’NEILL. “PCG : A family of simple fast space-efficient statistically good algorithms for random number generation”. In : *ACM Transactions on Mathematical Software* (2014).
- [39] Art B OWEN. “Sobol’indices and Shapley value”. In : *SIAM/ASA Journal on Uncertainty Quantification* 2.1 (2014), p. 245-251.
- [40] Hanchuan PENG, Fuhui LONG et Chris DING. “Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy”. In : *IEEE Transactions on pattern analysis and machine intelligence* 27.8 (2005), p. 1226-1238.
- [41] QURE.AI. *Using Variational Autoencoders*. <https://blog.oure.ai/notes/using-variational-autoencoders>. Date de consultation.
- [42] Melisa REYES-FUENTES et al. “AZTUSIA : A new application software for Uncertainty and Sensitivity analysis for nuclear reactors”. In : *Reliability Engineering & System Safety* 209 (2021), p. 107441.
- [43] Andrea SALTELLI. “Sensitivity analysis for importance assessment”. In : *Risk analysis* 22.3 (2002), p. 579-590.
- [44] Andrea SALTELLI et al. *Global sensitivity analysis : the primer*. John Wiley & Sons, 2008.
- [45] Andrea SALTELLI et al. “Sensitivity analysis for chemical models”. In : *Chemical reviews* 105.7 (2005), p. 2811-2828.
- [46] Robin M SCHMIDT. “Recurrent neural networks (rnns) : A gentle introduction and overview”. In : *arXiv preprint arXiv :1912.05911* (2019).
- [47] Claude Elwood SHANNON et Warren WEAVER. *The mathematical theory of communication, by CE Shannon (and recent contributions to the mathematical theory of communication), W. Weaver*. University of illinois Press, 1949.
- [48] Caroline Hillairet SIGNATURE. “Par : Laetitia Orta Titre : Intégration du risque de change au sein d’un Générateur de Scénarios Économiques et impact sur la valorisation de portefeuilles d’assurance vie”. In : ().
- [49] Anne SIMON. *La nouvelle organisation du Conseil de l’Union Européenne : un recul ou une opportunité pour la coopération au développement de l’Union Européenne ?* Centre européen de gestion des politiques de développement (ECDPM), 2003.
- [50] Eunhye SONG, Barry L NELSON et Jeremy STAUM. “Shapley effects for global sensitivity analysis : Theory and computation”. In : *SIAM/ASA Journal on Uncertainty Quantification* 4.1 (2016), p. 1060-1083.
- [51] Suwon SUH et Seungjin CHOI. “Gaussian copula variational autoencoders for mixed data”. In : *arXiv preprint arXiv :1604.04960* (2016).
- [52] XY SUN et al. “Three complementary methods for sensitivity analysis of a water quality model”. In : *Environmental Modelling & Software* 37 (2012), p. 19-29.

-
- [53] Natasa TAGASOVSKA, Damien ACKERER et Thibault VATTER. “Copulas as high-dimensional generative models : Vine copula autoencoders”. In : *Advances in neural information processing systems* 32 (2019).
- [54] Shu TEZUKA et Pierre L’ECUYER. “Efficient and portable combined Tausworthe random number generators”. In : *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 1.2 (1991), p. 99-112.
- [55] Tue TJUR, Rolf SUNDBERG et Inge HELLAND. “Analysis of Variance and Design of Experiments [with Discussion and Rejoinder]”. In : *Scandinavian journal of statistics* (1991), p. 273-322.
- [56] Saygun TURKAY, Eduardo EPPERLEIN et Nicos CHRISTOFIDES. “Correlation stress testing for value-at-risk”. In : *Journal of Risk* 5 (2003), p. 75-90.
- [57] Sebastiano VIGNA. “It is high time we let go of the Mersenne Twister”. In : *arXiv preprint arXiv :1910.06437* (2019).
- [58] Ronald J WILLIAMS et Jing PENG. “An efficient gradient-based algorithm for on-line training of recurrent network trajectories”. In : *Neural computation* 2.4 (1990), p. 490-501.
- [59] X ZENG et TS DURRANI. “Estimation of mutual information using copula density function”. In : *Electronics letters* 47.8 (2011), p. 493-494.
- [60] Aston ZHANG et al. “Dive into deep learning”. In : *arXiv preprint arXiv :2106.11342* (2021).

Table des figures

1.1	Les trois piliers de solvabilité 1	10
1.2	La pieuvre de solvabilité 2	12
1.3	Les trois piliers de solvabilité 2	14
2.1	Statistiques descriptives, rôle du shift au 31/12/2021.	25
2.2	Pourcentage de taux négatifs au 31/12/2021 en fonction de la valeur du shift.	26
2.3	Taux minimums atteints en fonction du shift.	26
2.4	Valeur de la VIF lorsque l'on change le shift et les données de marché. Les autres paramètres sont fixés et égaux à ceux du 31/12/2021.	27
2.5	Histogramme de la VIF. Nous nous rendons compte que la distribution est multimodale. Lorsque nous modifions les données de marché, nous modifions la moyenne de la distribution.	27
2.6	Exemple fil rouge.	28
3.1	Exemple fil rouge.	34
3.2	Intuition sur les indices de Sobol.	35
3.3	Exemple fil rouge.	40
3.4	Exemple fil rouge.	42
4.1	Les étapes de transformations que suit IGC, schéma directement repris du papier original.	45
4.2	Nous pouvons visualiser sur cette figure les étapes de transformations que propose IGC.	45
4.3	MICE vs IGC et CODINE.	46
4.4	Erreurs absolues.	48
4.5	Erreur moyenne d'approximation pour différentes valeurs de quantiles.	48
4.6	Exemple fil rouge.	50
4.7	Notre modèle a réussi à apprendre les copules et les niveaux de densité.	51
5.1	Illustration d'une cellule LSTM, illustration prise dans [46]. Nous retrouvons les différentes <i>portes</i> , d'entrée, d'oubli et de sortie. Ce réseau de neurone a une mémoire qu'il utilise pour produire des sorties.	56
5.2	Résumé de l'autoencodeur utilisé. Nous visualisons clairement l'importance de la "trick reparamétrisation" pour pouvoir effectuer la descente de gradient. Graphique pris dans l'article de blog [41].	59
6.1	Nous pouvons observer les différentes maturités allant de 1 an à 20 ans en abscisse. Nous avons les niveaux de volatilité en ordonnée et tout un ensemble de courbes qui correspondent aux différentes années. Les mois sont indexés de 0 pour janvier à 11 pour décembre.	62
6.2	Nous projetons avec le VAE les nappes de volatilité de Call dans un espace latent.	62
6.3	ACP vs VAE.	63

6.4	Fonctionnement du VAE Equity. Nous observons le processus latent z , et les nappes de volatilités des options, les processus x .	64
6.5	Continuité des variations dans l'espace latent	64
6.6	Données de synthèses, entraînement	65
6.7	Données réelles	65
6.8	2 composantes pour l'ACP	66
6.9	3 composantes pour l'ACP	66
6.10	4 composantes pour l'ACP	66
6.11	ACP vs VAE.	66
6.12	Espace latent VAE Swaptions	66
6.13	Variations continues dans l'espace latent du VAE Swaptions.	67
6.14	Importance des données de marché	67
7.1	VIF en fonction de la valeur x de l'abscisse dans l'espace latent des données de marché projetées.	68
7.2	Sensibilité de la VIF à la volatilité	69
7.3	Sensibilité de la VIF à la volatilité des CALL.	70
7.4	Interpolation pour obtenir VIF_{ref}^i en $x_{ref} = -2$. Le point interpolé est en rouge. Deux dates différentes ont été considérées pour les volatilités de swaptions.	71
7.5	Variation de la VIF pour les points recentrés, par rapport à $x_{ref} = -2$	71
7.6	Variation de la VIF en fonction du choc sur les vol equity et en fonction des vol de taux	72
7.7	Aide pour lire le tableau des positions	73
7.8	VIF en fonction des volatilités de swaption	74
8.1	Impact du shift sur les taux minimums atteints, prise en compte du contexte de taux.	77
8.2	Impact du shift sur les taux minimums atteints.	78
8.3	Notre modèle a réussi à apprendre les copules et les niveaux de densité.	78
8.4	Deuxième axe latent du VAE swaption, apprentissage de la densité.	79
8.5	Statistiques descriptives du shift	80
8.6	Statistiques descriptives du shift	81
8.7	Impact du Shift, résultats centrés	81
8.8	Plus les volatilités de swaptions sont élevées, plus l'impact sur le shift est élevé	82
9.1	Évolution historique des cibles distributionnelles.	85
9.2	Choc sur la cible de volatilité	86
9.3	Choc sur la cible de corrélation	86
9.4	Statistiques descriptives, choc sur la cible de standard deviation	87
9.5	Interpolation, exemple	87
9.6	Choc sur la cible de corrélation, VIF centrée	88
9.7	Choc sur la cible de corrélation	88
9.8	Choc sur la cible de volatilité, VIF centrée	89
9.9	Choc sur la cible de volatilité	89
9.10	Choc sur la cible de standard deviation	90
10.1	Les sommets dont les corrélations sont de plus ou moins 20% entre eux sont reliés par des arrêtes. Nous désignons les termes <i>facteur 1</i> , <i>facteur 2</i> par F1 et F2. Pour les autres notations, nous pouvons nous référer directement au tableau présenté ci-dessus	93
10.2	Evolution du BE en fonction du choc matrice corrélation.	96

10.3	Evolution de la VIF en fonction du choc matrice corrélation.	97
11.1	Impacts sur la VIF	99
11.2	Impacts sur la VIF, corrélations et shift	99
12.1	Position des données de volatilité dans l'espace latent	102
12.2	Deux nouveaux points que nous allons décompresser	103
12.3	Les données décompressées sont vraisemblables.	103
12.4	Décompression de points VAE Swaptions.	104
12.5	Données générées par le VAE Swaptions.	104
12.6	Régression linéaire.	106
12.7	Régression linéaire sans tenir compte des données de marché.	106
B.1	125
B.2	125

Liste des tableaux

- 1.1 Résumé des données par facteur de risque 16
- 1.2 Résumé des modèles 17
- 1.3 Résumé des modèles utilisés dans notre GSE 18

Annexes

Annexe A

Perturber plusieurs paires de la matrice de corrélation pour une analyse de sensibilité

A.1 Stress simultané de plusieurs corrélations

Théorie

Comme nous l'avons explicité plus haut, nous souhaitons stresser des groupes de corrélations. Nous avons déterminé les range dans lesquels nous souhaitons réaliser le stress, mais il s'agit maintenant de s'assurer que ces différentes perturbations n'interagissent pas entre elles pour faire perdre les bonnes propriétés de notre matrice.

Pour cette partie, nous allons devoir présenter rapidement les travaux de Higham [14].

Definition A.1.1 (Norme de Frobenius). *La norme de Frobenius d'une matrice D est donnée par :*

$$\|D\|_F = \sqrt{\sum_{i=1}^n \sum_{j=1}^n |d_{ij}|^2}$$

Theorem A.1.1 (Higham 1988). *La matrice définie non positive A peut être décomposée en $A = Q\Lambda Q^T$, où $Q \in \mathbb{R}^{n \times n}$ est une matrice orthogonale, c'est-à-dire, $QQ^T = Q^TQ = I$, et Λ est une matrice diagonale des valeurs propres $\Lambda_{ii}, i = 1, \dots, n$ de A . Nous pouvons définir $\tilde{\Lambda}$ comme étant une matrice diagonale égale à Λ sauf que les termes diagonaux négatifs sont remplacés par des zéros. Si nous reconstruisons la matrice \tilde{A} en utilisant $\tilde{\Lambda}$ et le même ensemble de vecteurs propres qui sont représentés par Q de sorte que $\tilde{A} = Q\tilde{\Lambda}Q^T$, alors \tilde{A} est la matrice semi-définie positive la plus proche de A par rapport à la norme de Frobenius.*

Ce théorème est très intéressant pour nous dans le sens où le risque que nous aillons en appliquant un choc à nos différentes corrélations est d'obtenir une matrice bien symétrique définie mais non positive!

Nous avons suivi la méthode proposée par [56]. Nous réordonnons donc nos facteurs de risque de manière à placer dans la matrice en bas à droite de taille 4×4 nos différentes corrélations qui vont bouger. Nous appelons encore une fois $\tilde{\Omega} = (\tilde{\rho}_{ij})$ notre matrice réordonnée. Et grâce à la décomposition de Cholesky, nous écrivons $\tilde{\Omega} = (\tilde{\rho}_{ij}) = LL^T$.

On écrit ensuite $L = L_u + L_s$, de manière à ce que le choc soit uniquement concentré sur L_s . Par exemple, si nous avons une matrice de taille 5×5 et que nous voulons uniquement perturber

3 variables :

$$L_u = \begin{pmatrix} \times & 0 & 0 & 0 & 0 \\ \times & \times & 0 & 0 & 0 \\ \times & \times & 0 & 0 & 0 \\ \times & \times & 0 & 0 & 0 \\ \times & \times & 0 & 0 & 0 \end{pmatrix}, \quad L_s = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \times & 0 & 0 \\ 0 & 0 & \times & \times & 0 \\ 0 & 0 & \times & \times & \times \end{pmatrix}$$

On écrit assez naturellement que $\Omega_u = L_u L_u^T$. Nous supposons maintenant que nous allons stresser $\hat{\Omega}$. Nous obtenons la matrice stressée $\hat{\Omega}$.¹ On obtient avec des notations évidentes :

$$\hat{\Omega} - \Omega_u = \begin{pmatrix} 0_{(n_u \times n_u)} & 0_{(n_u \times n_s)} \\ 0_{(n_s \times n_u)} & \Delta \hat{\Omega}_{(n_s \times n_s)} \end{pmatrix}_{n \times n}$$

Or $\Delta \hat{\Omega}_{(n_s \times n_s)}$ n'est pas positive semi définie. C'est donc à cette matrice que nous souhaitons appliquer le théorème de Higham. En notant $\Delta \hat{\Omega}'_{(n_s \times n_s)}$ sont approximant le plus proche avec le théorème mentionné plus haut, nous avons que :

$$\Omega_u + \begin{pmatrix} 0_{(n_u \times n_u)} & 0_{(n_u \times n_s)} \\ 0_{(n_s \times n_u)} & \Delta \hat{\Omega}'_{(n_s \times n_s)} \end{pmatrix}_{n \times n}$$

est une matrice semi-définie positive. Et en plus de cela c'est la plus proche de $\hat{\Omega}$. L'avantage de cette méthode est que nous n'avons pas touché aux corrélations qui ne sont pas modifiées ! Nous ne modifions que les corrélations impliquées.

Enfin, il faut s'assurer que les coefficients sur la diagonale valent bien 1, ce qui demande un dernier petit retraitement :

Désignons la sous-matrice inférieure droite $n_s \times n_s$ de Ω_u par $\Omega_u^s = (\rho_u^s(ij))$. De toute évidence, $\rho_u^s(ii) < 1$. Appelons la sous-matrice redimensionnée $\Delta \hat{\Omega}'' = (\hat{\rho}''(ij))$. Alors $\hat{\rho}''(ii) = 1 - \rho_u^s(ii)$ doit être satisfait. Par conséquent, la mise à l'échelle suivante suffit :

$$\Delta \hat{\Omega}'' = Q \left(\Delta \hat{\Omega}' \right) Q^T$$

où la matrice d'échelle $Q = (q(ij))$ est donnée par :

$$q(ij) = \begin{cases} 0 & \text{if } i \neq j \\ \frac{\sqrt{1 - \rho_u^s(ii)}}{\sqrt{\hat{\rho}''(ii)}} & \text{if } i = j \end{cases}$$

Avec la mise à l'échelle indiquée ci-dessus, la matrice suivante

$$\Omega_u + \begin{pmatrix} 0_{(n_u \times n_u)} & 0_{(n_u \times n_s)} \\ 0_{(n_s \times n_u)} & \Delta \hat{\Omega}''_{(n_s \times n_s)} \end{pmatrix}_{n \times n}$$

est une approximation semi-définie positive de $\hat{\Omega}$ qui est la même dans les premières n_u lignes et colonnes.

1. Ici nous ne stressons pas la décomposition de cholsky mais uniquement la matrice

Annexe B

Analyse de sensibilité de la graine des Générateurs de nombre Pseudo Aléatoires

Le générateur de scénarios économique doit projeter différents facteurs de risque sur une période donnée. Cela passe donc par la génération d'aléas, et donc par l'utilisation d'une source de nombres aléatoires. Le choix de la source d'aléas est très importante et consiste en une hypothèse du générateur de scénarios économiques. En effet, il est simple de se convaincre que selon la manière dont nous allons répartir les aléas dans leur espace, nous allons arriver à converger plus ou moins rapidement.

B.1 Propriétés désirables d'un générateur de nombres quasi-aléatoires

Un générateur de nombres quasi-aléatoires doit vérifier certaines propriétés importantes afin de pouvoir être utilisé dans une grande variété d'applications. Le papier [38] détail les propriétés souhaitables d'un générateur de nombres quasi-aléatoires. Nous retrouvons notamment :

- **Périodicité et Uniformité** : tout générateur de nombre aléatoires a une période fixée après laquelle il va se répéter. L'uniformité que les outputs aient le même nombre d'occurrence sur une période.
- **Validation** : Nous ne pouvons pas valider analytiquement un générateur de nombre quasi-aléatoires. Des tests empiriques sont requis.
- **Non prévisible, non reproductibilité** : Les paramètres qui initialisent le générateur de nombre quasi-aléatoires sont usuellement appelés la *seed*. Si le générateur est invertible, dès lors que nous allons connaître son état initial, nous sommes capable de retrouver les nombres aléatoires qu'il a généré dans le passé, menant potentiellement à de gros problèmes de sécurité.
- **Vitesse** : Nous désirons que les générateurs de nombres quasi-aléatoires soient rapides.
- **Période** : La taille de la mémoire compte. Car l'usage de mémoire va avoir un impact sur la parallélisation. Si la période d'un générateur est trop courte, il peut se répéter lorsque nous l'utilisons, ce qui n'est pas désirable. Nous souhaitons aussi que le range d'output soit le plus grand possible.
- **Seekabilité** : Quelques fois il est utile de pouvoir "sauter" d'un nombre arbitraire k de termes dans la séquence pseudo-aléatoire. Nous voulons être capable d'avancer k étapes dans le générateur sans prendre en temps en $O(k)$.
- **Streams multiples** : Nous voulons pouvoir lancer plusieurs instances du générateur de nombres quasi-aléatoires.

— **k-équidistribution** : Au cours d'une période entière du générateur, tous les k-tuples possibles doivent apparaître, et ce le même nombre de fois.

De manière plus formelle, un PRNG peut être défini comme un tuple (S, f, g, U, x^0) où S est l'espace d'état du générateur, U est l'espace aléatoire de sortie, $f : S \rightarrow S$ est la fonction de transition et $g : U \rightarrow S$ est la fonction qui extrait un output d'un état donné. x^0 est la seed. La séquence de sortie $y^{(t+1)} \in U$ est donné par des opérations élémentaires que nous allons brièvement présenter.

Forme générale des générateurs classiques Dans la suite, nous utilisons la notation $\mathbb{F}_2 = \{0, 1\}$. Puisque nous travaillons avec des ordinateurs, il fait sens d'utiliser l'arithmétique modulo 2. Lorsque nous implémentons des générateurs de nombres quasi-aléatoires avec des opérations sur les bits tels que le *shift*, les *rotations*, le *ou exclusif*,... nous faisons référence à des générateurs \mathbb{F}_2 -linéaires.

Definition B.1.1. (*Générateurs de nombres quasi-aléatoires*)

Soit

$$\begin{aligned} \mathbf{x}_n &= \mathbf{A}\mathbf{x}_{n-1}, \\ \mathbf{y}_n &= \mathbf{B}\mathbf{x}_n, \\ u_n &= \sum_{\ell=1}^w y_{n,\ell-1} 2^{-\ell} = y_{n,0}y_{n,1}y_{n,2} \cdots, \end{aligned}$$

- $\mathbf{x}_n = (x_{n,0}, \dots, x_{n,k-1})^t \in \mathbb{F}_2^k$ est le vecteur d'état de k-bit à l'étape n
- $\mathbf{y}_n = (y_{n,0}, \dots, y_{n,w-1})^t \in \mathbb{F}_2^w$ est le vecteur de sortie de w-bit à l'étape n
- k et w sont des entiers positifs
- \mathbf{A} est une matrice de transition avec des éléments dans \mathbb{F}_2
- \mathbf{B} est une matrice de transition avec des éléments dans \mathbb{F}_2 .

Périodicité La sortie $u_n \in [0, 1)$ à l'étape n est un nombre réel. La période du type de RNG défini par cette relation de récurrence ne peut pas excéder $2^k - 1$, puisqu'il y a seulement $2^k - 1$ valeur non nulles possibles pour \mathbf{x}_n . Une fois que ce maximum est atteint, le RNG a une période maximale. Pour obtenir une périodicité maximale, comme expliqué dans [25] il faut introduire quelques définitions supplémentaires.

Premièrement, il existe un entier $\rho > 0$ tel que $(x_\rho, \dots, x_{\rho+k-1}) = (x_0, \dots, x_{k-1})$. Puisque $\mathbf{x}_n = \mathbf{A}\mathbf{x}_{n-1}$, nous pouvons montrer que

$$\mathbf{x}_n = (\alpha_1 \mathbf{x}_{n-1} + \cdots + \alpha_k \mathbf{x}_{n-k}) \quad (\text{in } \mathbb{F}_2)$$

en notant le polynôme caractéristique de la matrice A par $\mathbf{A} P(z)$ [37] montre que la récurrence a une période maximale si et seulement si $P(z)$ est une primitive du polynôme sur \mathbb{F}_2 . La primitive a une propriété plus forte que l'irréductibilité : $P(z)$ est une primitive si et seulement si elle est irréductible pour tous les premiers diviseurs p_i de $r = 2^k - 1$, $z^{r/p_i} \not\equiv 1 \pmod{P(z)}$.

Il existe d'autres résultats notables sur les primitives des polynômes. Parmi les 2^k polynômes de degré k dans $\mathbb{F}_2[z]$, la proportion de primitives polynômiales est exactement

$$\frac{1}{k} \prod_{i=1}^b \frac{p_i - 1}{p_i}$$

où p_1, \dots, p_b are the distinct prime factors of $r = 2^k - 1$. Pour trouver les primitives polynômiales, il faut tester des conditions sur les candidats sélectionnés par cette méthode.

Seekability On veut sauter de \mathbf{x}_n à $\mathbf{x}_{n+\nu}$ pour un très grand entier ν . [13] propose une manière efficace de performer cette opération.

Combiner des générateurs \mathbb{F}_2 Il est possible de combiner C générateurs $\mathbb{F}_2\mathbb{F}_2$ -linéaires distincts. Si la c -ième récurrence a les paramètres $(k, w, \mathbf{A}, \mathbf{B}) = (k_c, w, \mathbf{A}_c, \mathbf{B}_c)$, la sortie des générateurs combinés à l'étape n est définie par :

$$\mathbf{y}_n = \mathbf{B}_1\mathbf{x}_{1,n} \oplus \cdots \oplus \mathbf{B}_C\mathbf{x}_{C,n},$$

$$u_n = \sum_{\ell=1}^w y_{n,\ell-1} 2^{-\ell},$$

où \oplus fais référence à l'opération xor. [54] indique une méthode efficace pour calculer la période et les différentes caractéristiques de ces générateurs.

Equidistribution [32] montre que nous pouvons obtenir des résultats sur l'équidistribution.

Critique des générateurs \mathbb{F}_2 [57] propose une critique très intéressante des limites des générateurs \mathbb{F}_2 . Comme nous allons le souligner dans les enjeux de l'utilisation des nombres quasi-aléatoires, les générateurs \mathbb{F}_2 ne passent pas bien les tests de linéarité.

B.2 Présentation des générateurs de nombre quasi-aléatoires sélectionnés

Dans le cadre de notre étude, nous allons tester plusieurs générateurs de nombres quasi-aléatoires et nous allons essayer de déterminer lesquels permettent une vitesse de convergence plus rapide.

B.2.0.0.1 LGC Historiquement, la première famille de méthode est les générateurs linéaires congruenciel (LCG). Nous génèrons les nombres de la séquence par une relation de récurrence. Nous utilisons la formule suivante :

$$X_{n+1} = (a \cdot X_n + c) \pmod{m} \quad (\text{B.1})$$

où X_{n+1} est le nombre suivant de la séquence, a est un coefficient multiplicatif, X_n est le nombre actuel de la séquence, c l'incrément et m le module. m est directement lié à la période maximale que la séquence peut atteindre, plus m devient grand, moins le calcul de la séquence est rapide. Le théorème de la longueur maximale [23] donne un résultat pour choisir ces paramètres.

B.2.0.0.2 Lagged-Fib Le lagged-Fibonacci generators (LFib) [31] est une amélioration de LGC reposant sur les nombres précédements générés pour produire le nombre suivant de la séquence (on peut combiner les paramètres dans n'importe quelle opération, telle que XOR, l'addition, la soustraction...)

$$X_n = (X_{n-k} + X_{n-l}) \pmod{m} \quad (\text{B.2})$$

avec $l > k$. Cette famille peut obtenir une plus grande période que LGC. Cependant, ces générateurs ont été connus pour avoir des déficiences statistiques, parce qu'ils sont basés sur les valeurs précédentes.

B.2.0.0.3 Feedback shift register Un linear feedback shift register (LFSR) [9] utilise une relation de récurrence (linéaire) de la forme :

$$X_N = (a_1X_{N-1} + a_2X_{N-2} + \dots + a_KX_{N-K}) \pmod{m} \quad (\text{B.3})$$

Cela fit avec le framework général que nous avons introduit pour \mathbb{F}_2 si nous prenons :

$$\mathbf{A}_0 = \begin{pmatrix} 1 & & & \\ & \ddots & & \\ & & & 1 \\ a_k & a_{k-1} & \dots & a_1 \end{pmatrix}$$

Encore une fois, il est possible d'utiliser une multitude d'opérations, telles que le XOR, la multiplication,... Nous pouvons aussi appliquer une fonction, appelée le *scrambler* [5] qui va améliorer les performances de LFSR en introduisant une transformation non linéaire telle que la rotation ou la somme.

B.2.0.0.4 Le Mersenne Twister est un LFSR. Il a une longue période unique de $2^{199337} - 1$. Le document [33] détaille la construction de ce générateur. Il génère des séquences de vecteurs de mots, considérés comme des entiers pseudo-aléatoires uniformes entre 0 et $2^w - 1$. Le mot suivant de la séquence est exprimé par une relation récurrente :

$$x_{k+n} = x_{k+m} \oplus (x_k^u | x_{k+1}^l) A \quad k = 0, 1, \dots$$

où $|$ désigne l'opération "bitwise". La forme de la matrice A est choisie pour rendre la multiplication par A très rapide :

$$A = R = \begin{pmatrix} 0 & I_{w-1} \\ a_{w-1} & (a_{w-2}, \dots, a_0) \end{pmatrix}$$

Pour améliorer l'algorithme, chaque mot généré est multiplié par une matrice inversible T . Nous choisissons les transformations suivantes (équivalentes à la multiplication matricielle) :

$$\begin{aligned} y &= x \oplus (x \ggg u) \\ y &= x \oplus (y \lll s)b \\ y &= x \oplus (y < t)c \\ y &= x \oplus (x \ggg) \end{aligned}$$

Le lecteur intéressé trouvera plus de détails dans le document original que nous avons mentionné.

B.2.0.0.5 PCG, générateurs \mathbb{F}_2 avec brouilleurs La famille de générateurs de nombres pseudo-aléatoires PCG est un peu différente. Si $x \in 0, 1^n$ est un mot de n bits, nous notons $x[i : j]$ la sous-chaîne de bits $x_i x_{i+1} \dots x_{j-2} x_{j-1}$. Si U est une séquence et U_i est le i -ème élément de la séquence, alors $U \pmod{M}$ désigne la séquence $(U_0 \pmod{M}, U_1 \pmod{M}, \dots)$. Classiquement, nous notons par \oplus l'opérateur XOR, le $+$ modulaire et les rotations à gauche et à droite par \lll et \ggg respectivement. Le PCG64 a un état interne de 128 bits et est évalué par une opération de :

$$S_{i+1} = aS_i + c \pmod{2^{128}}$$

Le premier état du générateur est la graine. Ensuite, pour extraire un nombre de cet état interne, PCG64 utilise une opération non linéaire, un brouilleur, comme nous l'avons mentionné.

Conformément à nos notations, les sorties successives de 64 bits du générateur sont X_0, X_1, \dots où :

$$X_i = \underbrace{(S_i[0 : 64] \oplus S_i[64 : 128])}_{Y_i} \gg \underbrace{S_i[122 : 128]}_{r_i}$$

Par convention, le XOR des deux moitiés de l'état est noté Y_i et r_i désigne le nombre de décalages de la i -ème rotation. Plus de détails sur cette famille peuvent être trouvés dans l'article récent.

B.2.0.0.6 Xoroshiro128+, \mathbb{F}_2 générateur avec brouilleurs Ce site web donne toutes les informations sur la mise en œuvre de Xoroshiro ainsi que son propre code. [5], [57] expliquent le fonctionnement de Xoroshiro. Le lecteur intéressé peut lire la littérature, mais il y a deux idées principales derrière Xoroshiro, un moteur linéaire et un brouilleur :

- Le moteur linéaire est une transformation linéaire sur $\mathbf{Z}/2\mathbf{Z}$, représentable par une matrice, et il est utilisé pour faire avancer l'état interne du générateur. (ayant de bonnes propriétés statistiques et facile à programmer)
- Le brouilleur est une fonction arbitraire sur l'état interne qui calcule la sortie réelle du générateur. Ses fonctions non linéaires appliquées à la matrice d'état réduisent ou éliminent les artefacts linéaires.

B.2.0.0.7 Marsaglia Ce générateur proposé par Marsaglia [3] concatène deux générateurs à multiplication avec retenue de 16 bits basés sur les récurrences

$$\begin{aligned} y &= x \oplus (x \gg u) \\ y &= x \oplus (y \ll s)b \\ y &= x \oplus (y \ll t)c \\ y &= x \oplus (x \gg l) \end{aligned}$$

Les 16 bits supérieurs de x_n et y_n sont les retenues de l'équation respective. Le générateur renvoie $Z_n / (2^{32} - 1)$. Il a été inclus en tant que générateur par défaut dans le paquet GNU *R* sous le nom de Marsaglia-MultiCarry.

B.2.0.0.8 Wichmann-Hill Utilisé dans R.... et également utilisé sur Excel! (après 2010, excel passe à MT..)

Il s'agit du générateur implémenté par la fonction RAND dans Microsoft Office Excel 2003 Microsoft. Il utilise le générateur Wichmann-Hill [34].

$$\begin{aligned} x_i &= 170x_{i-1} \bmod 30323 \\ y_i &= 172y_{i-1} \bmod 30307 \\ z_i &= 171z_{i-1} \bmod 30269 \\ u_i &= \left(\frac{x_i}{30323} + \frac{y_i}{30307} + \frac{z_i}{30269} \right) \bmod 1. \end{aligned}$$

Les graines initiales sont x_0, y_0 et z_0 . Restrictions : $0 < x_0 < 30323, 0 < y_0 < 30307$ et $0 < z_0 < 30269$.

B.3 Problématiques autour des générateurs de nombres quasi-aléatoires

Nous voyons qu'il existe un grand nombre de générateurs de nombres quasi-aléatoires, et nous en avons présentés quelques uns de très populaires. Dans ce mémoire nous aimerions lever quelques points d'obscurités :

1. Quels sont les générateurs qui vérifient toutes les propriétés souhaitables ?
2. Quel est l'impact du choix du générateur de nombre quasi-aléatoire sur la valeur de la sortie ?
3. Pouvons-nous arriver à faire converger plus rapidement les calculs ALM en choisissant le bon générateur de nombres quasi-aléatoires ?

B.4 Applications pratiques et résultats

TestU01 est un ensemble de tests statistiques qui permettent d'évaluer la qualité d'un générateur de nombres pseudo aléatoires. Il propose trois batteries de tests, *SmallCrush*, *Crush* et *BigCrush*.

1. **SmallCrush** est la batterie de tests la plus simple et la moins rigoureuse de TestU01. Elle vise à détecter rapidement les défauts évidents d'un générateur de nombres aléatoires (RNG), tels qu'un biais important ou une corrélation entre les nombres générés. SmallCrush comprend 10 tests et est conçue pour être réalisée rapidement, ce qui la rend adaptée à un premier dépistage d'un RNG.
2. **Crush** est une batterie de tests plus rigoureuse que SmallCrush. Elle comprend 96 tests conçus pour évaluer un plus large éventail de propriétés d'un RNG, telles que la capacité à générer de longues séquences de nombres aléatoires et l'uniformité des nombres générés. Crush prend significativement plus de temps à s'exécuter que SmallCrush, mais reste relativement rapide par rapport à la batterie de tests la plus rigoureuse, BigCrush.
3. **BigCrush** est la batterie de tests la plus rigoureuse de TestU01. Elle comprend 106 tests et vise à détecter des motifs ou des corrélations subtiles dans un RNG qui pourraient ne pas être détectables par les batteries de tests moins rigoureuses. BigCrush vise à fournir une évaluation complète et exhaustive d'un RNG et peut prendre plusieurs heures pour être réalisée.

De même, les tests Diehard sont un ensemble de 15 tests individuels visant à évaluer la qualité des générateurs pseudo-aléatoires. Nous n'utiliserons ni PractRand ni DieHard, mais il est important de garder à l'esprit que ces bibliothèques constituent une alternative précieuse à TestU01. Nous présentons dans le tableau ci-dessous les différents résultats de TestU01.

Nom du générateur	Nom du test	Résultat
Marsaglia	Crush	45 failed
Wichmann-Hill	Crush	15 failed
Mersenne Twister	Crush	3 failed (linearity tests)
Xoshiro128+	BigCrush	3 failed (linearity tests)
PCG64	BigCrush	0 failed

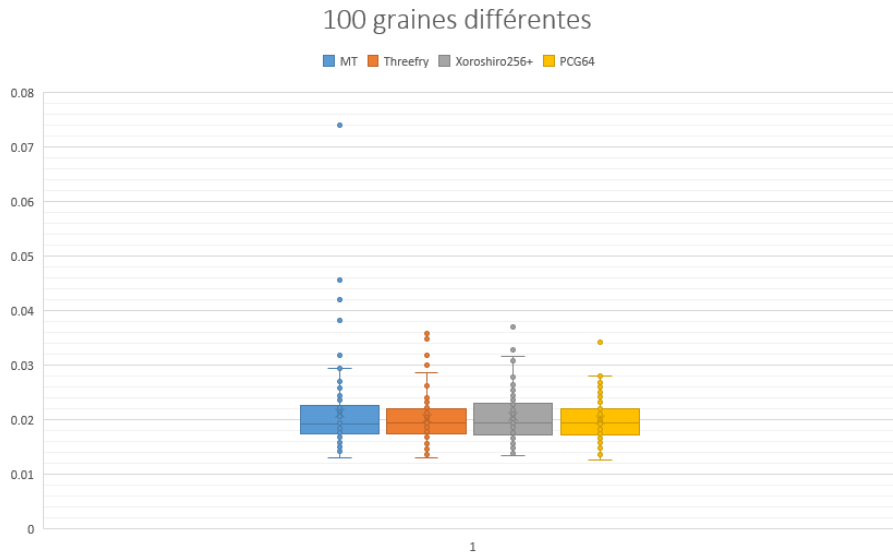


FIGURE B.1 – .

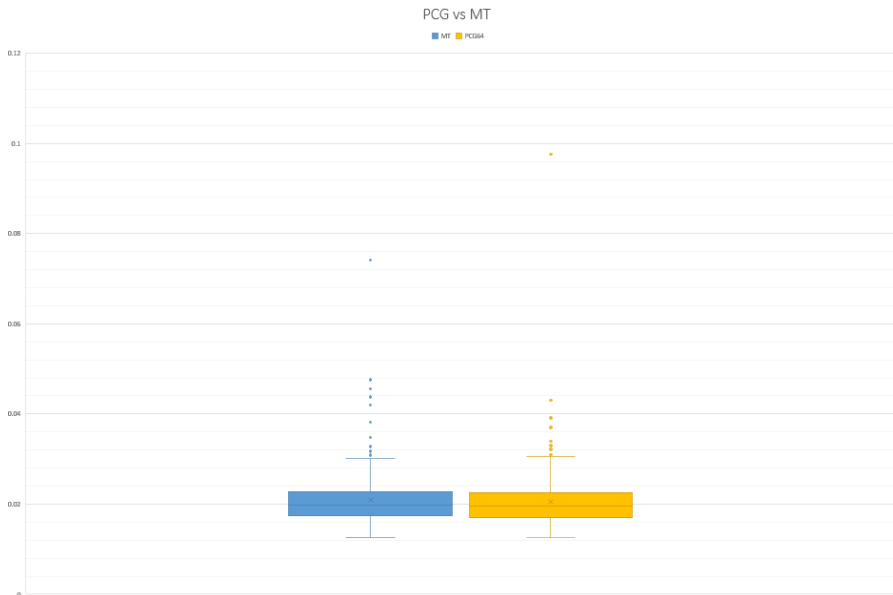


FIGURE B.2 – .

Nous réalisons que les générateurs de nombres pseudo-aléatoires sont indistinguables. À priori nous pouvons utiliser n'importe lequel de ces générateurs, il ne devrait pas y avoir d'impact.