CEA | CENTRE D'ÉTUDES ACTUARIELLES

INSTITUT DES ACTUAIRES

Mémoire présenté devant le Centre d'Etudes Actuarielles pour la validation du cursus à la Formation d'Actuaire du Centre d'Etudes Actuarielles et l'admission à l'Institut des Actuaires le 20 mars 2017

Par : Clément Saillard

Titre : Utilisation d'algorithmes génétiques pour évaluer la robustesse du processus de PSDisation lors de l'agrégation des risques dans les modèles internes

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

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

Membre présent du jury de l'Institut des Actuaires :

_____

_____

_____

Membres présents du jury du Centre d'Etudes Actuarielles :

_____

_____

_____

_____

_____

_____

_____

_____

Secrétariat :

Bibliothèque :

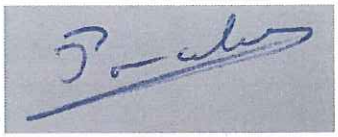
Entreprise : BNP Paribas Cardif

Nom : Henri-Brice Salle

Signature / Cachet :

Directeur de mémoire en entreprise :

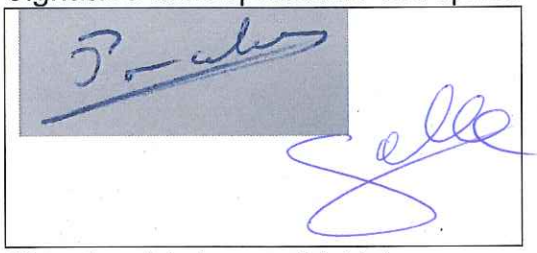
Nom : Victorien Poncelet (ancien employeur : ACPR)

Signature : 

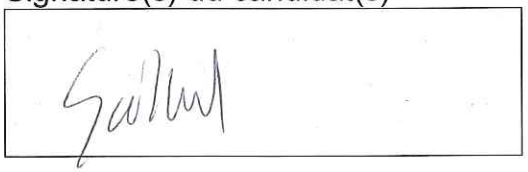
Autorisation de publication et de mise en ligne sur un site de diffusion de documents actuariels

(après expiration de l'éventuel délai de confidentialité)

Signature du responsable entreprise



Signature(s) du candidat(s)



CEA | CENTRE D'ÉTUDES ACTUARIELLES

BNP PARIBAS CARDIF

Mémoire d'actuariat

Utilisation d'algorithmes génétiques pour évaluer la robustesse du processus de PSDisation lors de l'agrégation des risques dans les modèles internes

Auteur :
Clément Saillard

Encadrants :
Victorien Poncelet
Xavier Milhaud

28 Janvier 2017

Table des matières

Remerciements **4**

Introduction **7**

1 Contexte et définitions **11**

1.1 Modélisation de la dépendance en probabilité 11

1.2 Matrice PSD 13

1.3 Normes matricielles utilisées dans ce mémoire 16

1.4 Algorithmes d'optimisation classiques 16

1.4.1 Algorithmes linéaires de type gradient à pas optimal (sans contraintes) 17

1.4.2 Algorithmes quadratiques (sans contraintes) 18

1.4.3 Algorithme d'optimisation avec contraintes inégalités . . 20

1.4.4 Algorithmes génétiques (avec ou sans contraintes) . . . 20

1.5 Problématique de ce mémoire 25

2 PSDisation d'une matrice de pseudo-corrélation **26**

2.1 Algorithme des projections alternées 26

2.2 Algorithme de Newton 27

2.3 Un algorithme utilisé par les banques et les assurances : Rebonato-Jäckel 29

2.4 Extension de l'algorithme de Rebonato-Jäckel aux cas des contraintes inégalités et difficultés rencontrées 34

2.5 Ajustements numériques finaux 36

2.6 Notations pour la suite du mémoire 37

3 Méthodes d'agrégation pour le calcul de l'exigence de capital **38**

3.1 Agrégation de capitaux pour le calcul du SCR 38

3.2 Agrégation des fonctions de pertes pour le calcul du SCR . . . 40

3.2.1 Distinction entre facteurs de pertes et facteurs de risques 40

3.2.2 Les fonctions d'approximation des pertes 40

3.2.3 L'agrégation de facteurs de risques 41

3.3 Copules 42

3.4 Générateurs de nombres aléatoires et stabilité des simulations . 48

4 Utilisation d'algorithmes génétiques pour étudier la sensibilité autour de l'étape de PSDisation 53

4.1 Optimisation du SCR à l'aide d'algorithmes génétiques 53

4.1.1 Optimisation génétique du SCR avec une permutation des facteurs de pertes 53

4.1.2 Optimisation génétique du SCR en fonction des coefficients de pondération 56

4.2 Autres sensibilités autour de la PSDisation 57

4.2.1 Ajout d'une dimension vide 57

4.3 Conclusion 58

5 Interprétation des résultats numériques 59

5.1 Présentation des exemples choisis 59

5.2 Comparaison des algorithmes sans contraintes sur plusieurs exemples 61

5.3 Analyse de la matrice S^H_{101} obtenue 62

5.4 Impact de la PSDisation sur le SCR 62

5.5 Difficultés rencontrées par l'algorithme avec contraintes inégalités 63

5.6 Choix du générateur de nombres aléatoires 64

5.7 Rappel sur les critères d'évaluation de l'impact d'une transformation 64

5.8 Impact sur le SCR d'une permutation génétique des marginales 64

5.9 Impact sur le SCR d'une modification génétique de la pondération de la PSDisation 66

5.10 Impact sur le SCR de l'ajout d'une dimension vide 66

5.11 Conclusion 67

Conclusion 69

Références 71

A Paramètres et tests effectués 72

A.1 Matrices utilisées dans ce mémoire 72

A.2 Marginales utilisées dans ce mémoire 74

A.3 Fonctions de pertes utilisées dans ce mémoire 75
A.4 Comparaison des 3 algorithmes sans contraintes 75
A.5 Résultats des PSDisations avec pondération 78
A.6 Problème des minima locaux dans le cas de l'algorithme avec contraintes 79
A.7 Test de la stabilité de la simulation du SCR avec un générateur de nombres aléatoires de Mersenne 80
A.7.1 Cas d'une copule gaussienne et de facteurs de pertes (Mersenne) 80
A.7.2 Cas d'une copule de Student et de facteurs de pertes (Mersenne) 82
A.7.3 Cas d'une copule gaussienne et de facteurs de risques (Mersenne) 83
A.7.4 Cas d'une copule de Student et de facteurs de risques (Mersenne) 85
A.8 Test de la stabilité de la simulation du SCR avec un générateur de nombres aléatoires de Sobol 87
A.8.1 Cas d'une copule gaussienne et de facteurs de pertes (Sobol) 87
A.8.2 Cas d'une copule de Student et de facteurs de pertes (Sobol) 89
A.8.3 Cas d'une copule gaussienne et de facteurs de risques (Sobol) 90
A.8.4 Cas d'une copule de Student et de facteurs de risques (Sobol) 92
A.9 Résultats d'optimisation obtenus et sensibilités 94
A.9.1 Hypothèses de travail pour la suite 94
A.9.2 Rappels sur les critères d'évaluation d'impact 94
A.9.3 Cas d'une copule gaussienne et de facteurs de pertes . . 94
A.9.4 Cas d'une copule de Student et de facteurs de pertes . . 98
A.9.5 Cas d'une copule gaussienne et de facteurs de risques . . 102
A.9.6 Cas d'une copule de Student et de facteurs de risques . 105

B Notice du code informatique (sous R) 109

Remerciements

Je tiens à remercier Victorien Poncelet, mon tuteur professionnel et Xavier Milhaud, mon tuteur académique qui m'ont guidé tout au long de la construction de ce mémoire. Je souhaite également remercier BNP Paribas Cardif (mon employeur actuel), l'Autorité de Contrôle Prudentiel et de Résolution (mon précédent employeur), ainsi que le Centre d'Etudes Actuarielles. Un grand merci également à Henri-Brice Salle pour sa relecture attentive et à Francis Maisonneuve et François Aubonnet pour leur soutien à diverses étapes de mes années d'études.

Evidemment, aucun des propos tenus dans ce mémoire n'engagent mon employeur. Toutes les méthodes et toutes les données utilisées dans ce mémoire sont publiques.

La démarche présentée ici, comme les résultats, peuvent contenir des erreurs, malgré les nombreuses relectures. Dans ce cas, j'en porte l'entière responsabilité. J'ai essayé d'être le plus transparent possible, dans le corps du texte avec une explication détaillée des algorithmes, et dans les annexes avec la présentation intégrale des paramètres et des résultats. Ceci devrait permettre au lecteur qui identifierait une erreur d'en détecter la source.

Résumé

Dans le régime prudentiel Solvabilité 2, les assureurs peuvent, sous réserve de l'approbation des superviseurs, utiliser un modèle interne afin de calculer leur exigence de fonds propres. Celui-ci doit correspondre au quantile à 99,5 de leur distribution de pertes potentielles à un an. La grande majorité des modèles internes requièrent trois grandes étapes pour établir la distribution de pertes potentielles : premièrement un calibrage des distributions marginales de chacun des facteurs de risques considérés indépendamment (inondation, marché etc.), ensuite la détermination de fonctions d'approximation afin de relier ces facteurs de risques aux pertes potentielles de l'assureur et enfin l'agrégation des marginales dans une distribution jointe représentant la possibilité pour les facteurs de risques de se réaliser simultanément.

Les copules les plus utilisées par les assureurs pour l'étape d'agrégation sont paramétrées par une matrice positive semi-définie représentant les corrélations entre les facteurs de risques. En pratique, cette matrice, dont les dimensions peuvent être importantes ($\approx 1000 \times 1000$), est remplie par des mesures statistiques (typiquement pour les facteurs de risque de marché), par des jugements d'experts (quand des hypothèses raisonnables peuvent être formulées) et par des formules de remplissage automatique (pour les corrélations totalement inconnues). Néanmoins, la matrice de pseudo-corrélation ainsi obtenue est rarement positive semi-définie et ne peut donc pas être utilisée telle quelle pour paramétrer la copule utilisée. Les assureurs doivent donc recourir à des algorithmes de « PSDisation » afin de rendre la matrice positive semi-définie.

L'objectif de ce mémoire est de répondre numériquement à des questions très pratiques que peuvent se poser les superviseurs et les assureurs quant à la robustesse d'un tel processus de PSDisation. En particulier, les impacts sur l'exigence de fonds propres d'une permutation des facteurs de risques, de l'ajout d'une « dimension vide » ou encore de sensibilités sur la pondération des coefficients dans l'algorithme de PSDisation sont analysés sur quelques exemples. Des algorithmes génétiques, qui sont des algorithmes d'optimisation inspirés par l'évolution biologique, sont utilisés afin de déterminer des bornes minimum et maximum dans certains cas listés ci-dessus.

Le travail présenté ici montre que des transformations théoriquement neutres de la matrice de pseudo-corrélation ou des sensibilités légères peuvent avoir un impact significatif sur le SCR de l'assureur considéré ($\approx 5\%$ du SCR dans certains exemples testés), ce qui peut représenter plusieurs centaines de millions d'euros pour certains assureurs systémiques. Ces résultats confirment la nécessité d'établir un environnement de contrôle interne robuste autour de l'étape de PSDisation lors de l'agrégation des risques d'un modèle interne.

Mots-clés : agrégation, matrice de corrélation, copule, SCR, positive semi-définie, Rebonato-Jäckel, algorithmes génétiques.

Abstract

Under Solvency II environment, insurers can calculate their own funds requirements, which represents the $99,5^{th}$ quantile of their potential loss distribution, with an internal model once the latter is approved by supervisors. Nearly all internal models require roughly three steps of modelling in order to determine the potential loss of distribution : first a calibration of marginal distributions of individual risk factors (flood, market etc.), then a fitting of proxy functions to link these risk factors to the insurer's potential loss and finally the aggregation of marginal distributions into a joint distribution which represents the ability of these individual risks to occur simultaneously. The $99,5^{th}$ quantile of the loss distribution can then be computed with simulation techniques.

The copulas which are most widely used by insurers for the aggregation step described above require as input a positive semi-definite matrix representing the correlations between individual risk factors. This matrix, which dimensions can be important ($\approx 1000 \times 1000$), is generally filled in partly with statistical measures (typically for market risk factors), partly with expert judgement (when assumptions can be reasonably made) and partly with automatic formulas (for the totally unknown correlations). Nevertheless, the pseudo-correlation matrix thus obtained is rarely positive semi-definite, and therefore cannot be used directly to aggregate risks with a copula. Insurers need to apply specific « PSDisation » algorithms in order to make the matrix positive semi-definite.

The aim of this work is to answer numerically some very practical questions supervisors and insurance companies can ask themselves about the robustness of such a PSDisation process. For example, the questions of how the own funds requirements calculation is impacted by a permutation of risk factors, by the addition of a « blank » dimension in the initial matrix, or by small sensitivities of the weights used for the algorithm are answered on several examples. Genetic algorithms, which are optimization algorithms inspired by biological evolution, are used to estimate the minimum and maximum of the own funds requirement according to the different configurations listed above.

This work shows that theoretically neutral transformations of the matrix or small sensitivities can lead to significant changes of SCR (up to $\approx$ 5% of the SCR based on some examples tested), which can represent several hundreds of millions of for systemic insurers. These results confirm the need for very strong internal control around the PSDisation step of the aggregation process in internal models.

Keywords : aggregation, correlation matrix, copula, SCR, positive semi-definite, Rebonato-Jäckel, genetic algorithms.

Introduction

Lorsqu'un contrat d'assurance est conclu entre un assureur et un assuré, la compagnie d'assurance accepte de prendre un risque, encadré par le contrat, au travers de son engagement futur envers son assuré, en contrepartie du paiement d'une prime. Il est possible de modéliser le risque pris par l'assureur par une variable aléatoire. La réglementation Solvabilité 2 dispose que l'assureur doit inscrire dans ses comptes une provision technique dite *Best Estimate* qui correspond à l'espérance de cette variable. Le *Best Estimate* représente la somme des flux futurs actualisés et probabilisés. Le calcul du *Best Estimate* peut être relativement compliqué, en particulier pour les contrats d'assurance-vie qui nécessitent la prise en compte des interactions actif-passif (modélisation des rachats structurels et conjoncturels, modélisation de la participation aux bénéfices en fonction du rendement des actifs en portefeuille etc.). En plus de cette provision *Best Estimate*, l'assureur doit disposer de capitaux propres, lui permettant de faire face à ses engagements dans 99,5 cas sur 100. Cette exigence se traduit par l'inégalité suivante :

$$\textit{Provisions} + \textit{capitaux propres} > \textit{Quantile}\ 99,5\ \textit{des engagements}. \quad (1)$$

Pour donner des ordres de grandeur, le tableau suivant récapitule les chiffres donnés dans les communications publiques de quelques grands assureurs européens, au 31/12/2015 [1]. Ce tableau montre qu'une erreur de 1 pour cent sur l'exigence de capital peut représenter plusieurs centaines de millions d'euros :

Chiffres en Md€	Allianz	Aviva	Axa	Generali
Exigence de capital	29,0	16,4	28,8	20,5

En pratique, la réglementation Solvabilité 2 prévoit une formule dite formule standard » pour le calcul du quantile à 99,5 des engagements, mais offre la possibilité aux assureurs de recourir à un modèle interne propre au profil de risque de la société. Ce choix est justifié pour les assureurs dont le profil de risque serait très différent de celui de la formule standard pour des raisons diverses (politique de souscription très stricte, clientèle de niche, diversification importante en raison d'une activité internationale etc.). Afin de pouvoir utiliser ce modèle interne pour afficher et piloter sa solvabilité, les assureurs doivent obtenir l'approbation du ou des superviseurs concernés. La décision des superviseurs est fondée sur la robustesse des méthodes statistiques et des processus internes à la société, sur la prudence des hypothèses retenues lorsque

1. D'après les éléments disponibles sur les sites internets respectifs des assureurs le 25/03/2016, et lorsqu'applicable, converti avec le taux de change EUR/STER en vigueur au 31/12/2015, soit 0,73680.

les données sont insuffisantes et bien sûr sur le respect des dispositions prévues par la réglementation.

Un modèle interne repose classiquement sur les trois étapes suivantes :

1. **Modélisation des facteurs de risques** auxquels l'assureur est exposé (risque de taux, risque de catastrophe naturelle etc.). Cette étape repose sur le choix de lois marginales (log-normale etc.) et sur le calibrage de leurs paramètres selon des méthodes statistiques classiques (maximum de vraisemblance par exemple) ;
2. **Calibrage de fonctions d'approximation** (quelle que soit leur forme : portefeuille réplicants avec des formules fermées etc.) qui représentent l'impact de la réalisation des facteurs de risques sur les fonds propres ;
3. Modélisation de la manière dont ces risques sont susceptibles de se réaliser conjointement. Cette étape est couramment appelée l'**agrégation des risques**. Le résultat final du calcul d'exigence de capital est très sensible à la manière dont les risques sont agrégés, ce qui rend cette étape très sensible au regard du superviseur.

L'étape 1 ci-dessus repose sur l'identification des facteurs de risques et le calibrage des lois de probabilité associées. Les facteurs de risques sont des variables aléatoires qui, si elles viennent à évoluer, peuvent engendrer des pertes pour l'assureur. Par exemple, le risque de taux est un facteur de risque, puisqu'en cas de hausse des taux, la valeur des obligations en portefeuille de l'assureur diminue.

L'étape 2 ci-dessus peut s'avérer complexe et ne fait pas partie à proprement parler du sujet de ce mémoire. En effet, pour calculer la perte induite par une variation d'un facteur de risque, une transformation est nécessaire, et celle-ci peut être très compliquée. Dans l'exemple du risque de taux, la hausse des taux a un effet sur le prix des obligations qui peut être estimé assez simplement, quitte à effectuer un calcul approché avec la duration et la convexité. Néanmoins, la hausse des taux a également un impact sur les provisions *Best Estimate* (modification du rendement des réinvestissements futurs, modification des rachats etc.) dont l'estimation nécessite le recours à des techniques de simulation de type Monte-Carlo. En pratique, pour des raisons de temps de calcul, l'impact global de la variation d'un facteur de risque doit donc être estimé au travers de fonctions d'approximation. En revanche, dans certains cas, l'assureur peut directement modéliser des facteurs de pertes qui calculent directement sa perte. Par exemple, le risque de réserves en assurance santé, qui correspond à la volatilité des provisions pour les sinsitres à payer, peut être modélisé directement par des méthodes classiques (*bootstrap* par exemple). Dans ce cas, la variation du facteur de perte correspond à la perte subie par l'assureur et ne nécessite pas de transformation complexe. Au final, la perte

de l'assureur est calculée par (f étant une fonction, et les autres termes étant des variables aléatoires) :

$$Pertes = f(facteurs\ de\ risques) + facteurs\ de\ pertes. \tag{2}$$

L'étape 3 ci-dessus correspond à la mesure de dépendance des facteurs de risques ou de pertes considérés. En effet, une sommation brutale de tous les quantiles des facteurs de risques ou de pertes ne serait pas justifiée car certains risques n'ont aucune raison de se matérialiser simultanément. Par exemple, le risque de longévité (couteux pour l'activité de retraite de l'un assureur) et le risque de mortalité (couteux pour l'activité de temporaire décès de l'assureur) ne peuvent pas se réaliser simultanément. De même, le risque de tempête aux Caraïbes et le risque de catastrophe industrielle en Europe semblent *a priori* peu corrélés. En revanche, le risque de baisse des marchés actions en France et le risque de baisse des marchés actions en Allemagne sont intuitivement plus corrélés. Dans la simulation des pertes de l'assureur en vue de calculer le quantile à 99,5, il faut donc tenir compte de ces effets de corrélations.

Le théorème de Sklar montre que toute loi conjointe de plusieurs variables peut s'exprimer en fonction des marginales de chacune de ces variables, à l'aide d'une fonction appelée « copule ». Un choix relativement classique de copule pour l'agrégation des risques dans un modèle interne est la copule gaussienne, qui permet contrairement à l'agrégation sous forme quadratique (approche dite « var-covar ») d'obtenir l'ensemble de la distribution de pertes. Les copules gaussiennes, comme d'autres classes de copules, nécessitent de connaître la matrice de variance-covariance des variables considérées. Cette matrice de variance-covariance doit être positive semi-définie, comme toute matrice de variance-covariance, afin de pouvoir être être utilisée dans une copule.

Néanmoins, en pratique, dans les modèles internes, les matrices de variance-covariance sont remplies en partie par des mesures statistiques (comme pour la corrélation entre 2 risques financiers pour lesquels les données sont légion), en partie sur base de jugements d'expert (pour des risques rarement observés mais pour lesquels il est possible d'avoir une bonne « intuition » du niveau de corrélation) et en partie automatiquement, sur la base de règles prédéfinies (pour les corrélations les plus difficiles à mesurer, comme entre la hausse de mortalité en Asie et le taux de rachat en Allemagne). La matrice ainsi obtenue est par construction symétrique donc diagonalisable mais rarement définie. Certaines valeurs propres étant négatives, il est nécessaire de modifier la matrice, afin de pouvoir recourir à une copule. Cette étape, dite de « PSDisation » (en référence au terme anglais de *Positive Semi-Definite*) est cruciale et son impact sur les chiffres d'exigence de capital est important. Certains algorithmes de PSDisation permettent de pondérer ou encadrer les coefficients de matrice lors de cette transformation. Cette possibilité est très

importante en pratique car les assureurs peuvent ainsi figer (dans une certaine mesure) les coefficients les plus importants (c'est-à-dire avec le plus fort impact sur l'exigence de capital) et de différencier les coefficients issus d'une mesure statistique de ceux définis par jugement d'experts.

La figure 1 résume de manière très simplifiée un processus classique de calcul par un assureur de son exigence de fonds propres (*Solvency Capital Requirement*) avec un modèle interne :

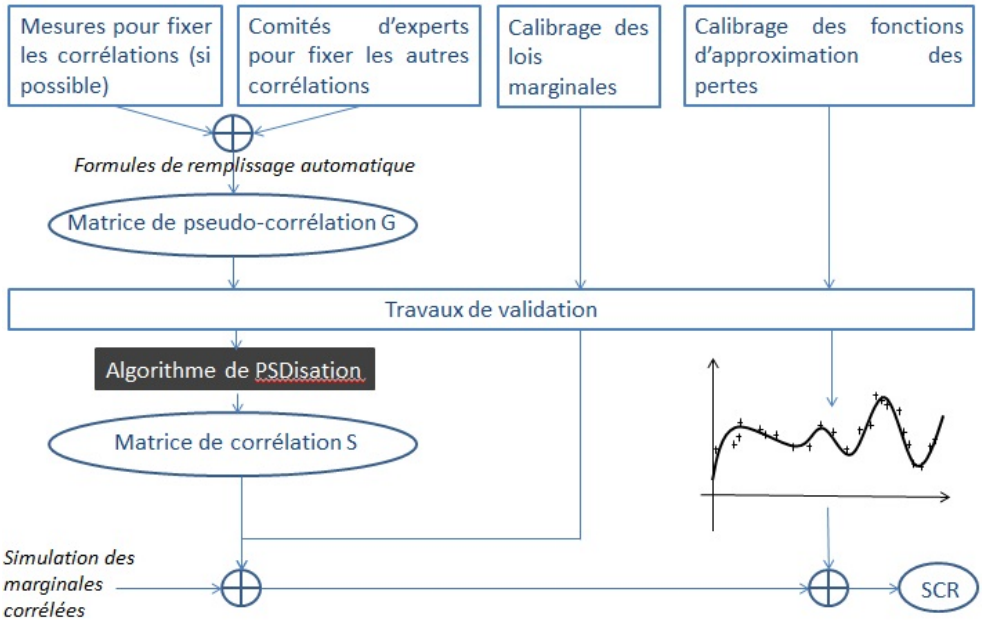


FIGURE 1 – Processus classique de calcul de l'exigence de capital (SCR) en modèle interne

L'objectif de ce mémoire est de mesurer la robustesse de cette étape de PSDisation, en utilisant des algorithmes d'optimisation génétique pour estimer l'intervalle auquel l'exigence de capital appartient compte-tenu de sensibilités sur le choix des bornes de pondération ou en réalisant des transformations théoriquement neutres (permutation des variables de la matrice initiale ou ajout d'une colonne vide corrélée aux variables existantes).

1 Contexte et définitions

L'objectif de cette section est de présenter succinctement les principales notions mathématiques utilisées dans ce mémoire. Les algorithmes sont présentés dans la section 2.

1.1 Modélisation de la dépendance en probabilité

Quelques notions de probabilité doivent être clairement définies pour poser plus précisément la problématique de ce mémoire. Le recours au formalisme de la théorie de la mesure ne sera pas nécessaire. L'espace Ω est appelé univers des évènements.

Définition 1.1. *Une variable aléatoire absolument continue de densité de probabilité f et de fonction de répartition F est une application $X : \Omega \to [0;1]$; telle que $P(X < x) = F(x) = \int_{-\infty}^{x} f(u)du$.*

En pratique, dans ce mémoire, toutes les variables aléatoires seront absolument continues. En conséquence, pour la suite, il ne sera plus fait référence explicitement à la notion d'absolue continuité. Les variables aléatoires considérées représenteront des facteurs de risque (risque de baisse des marchés actions etc.) ou de pertes (risque de sous-estimation des réserves, risque de sous-estimation des provisions de primes etc.), et les transformations de ces variables aléatoires au travers de fonctions d'approximation représenteront une nouvelle variable aléatoire représentant les pertes de l'assureur. Voici un exemple de variable aléatoire particulièrement adaptée à la modélisation de facteurs de pertes ou de risques qui sera utilisée tout au long de ce mémoire.

Exemple 1.2. *Une variable aléatoire lognormale de paramètres μ et σ est la variable aléatoire définie par la densité de probabilité f suivante : $f(u) = \frac{1}{u\sigma\sqrt{2\pi}} \exp\Big(-\frac{(\ln(x)-\mu)^2}{2\sigma^2}\Big)\mathbf{1}_{[0;+\infty]}$. Son espérance vaut $\exp\Big(\mu + \frac{\sigma^2}{2}\Big)$ et sa variance vaut $\exp(2\mu+\sigma^2)(\exp(\sigma^2)-1)$.*

Cette loi de probabilité est très utile en assurance pour représenter un facteur de perte. Comme le montre la figure 2 représentant sa fonction de répartition, on remarque (très schématiquement) que la perte peut être infinie (limite à droite infinie) mais le gain fini (limite à gauche finie).

En pratique, il est souvent nécessaire de modéliser plusieurs variables aléatoires conjointement, dans un vecteur nommé vecteur aléatoire. En effet, la connaissance des variables aléatoires une par une ne donne pas d'information sur leur capacité à se réaliser en même temps. La mesure de la dépendance des facteurs de risques et de pertes auxquels un assureur est exposé est pri-

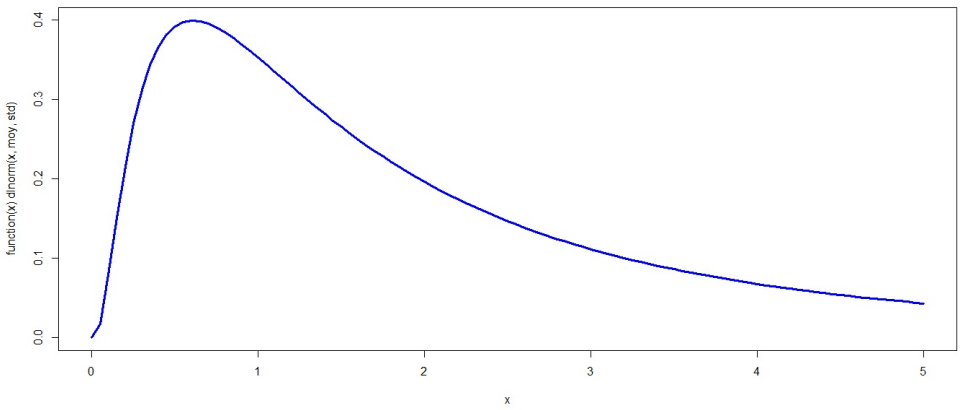


FIGURE 2 – Densité d'une loi lognormale ($\mu = 0{,}5$ et $\sigma = 1$)

mordiale pour déterminer son exigence de capital, correspondant au quantile à 99,5 de la fonction de perte de l'assureur.

Définition 1.3. *Un vecteur aléatoire X aléatoire de fonction de répartition F est un vecteur $\in \mathbb{R}^n$ dont chacune des n composantes X_i est une variable aléatoire et tel que :* $\forall (x_1, \ldots, x_n) \in \mathbb{R}^n, P\left(\bigcap_{i=1}^{n}(X_i < x_i)\right) = F(x_1, \ldots, x_n)$.

A l'aide de la fonction de répartition jointe F du vecteur aléatoire X, il est aisé de déterminer la fonction de répartition de chacune des variables aléatoires sous-jacentes, comme le montre la définition ci-dessous :

Définition 1.4. *La i-ème loi marginale du vecteur aléatoire X est la fonction de répartition donnée par :* $F_i(x_i) = \lim_{\forall j \neq i, x_j \to \infty} F(x_1, \ldots, x_n)$. *Cette fonction de répartition décrit le comportement de la variable aléatoire X_i.*

Par contre, connaissant les lois marginales de chacune des variables aléatoires X_i, il est plus difficile de reconstituer la loi jointe du vecteur aléatoire X. Il faut pour cela reconstituer le degré de corrélation entre toutes les variables aléatoire, c'est à dire leur capacité à se réaliser conjointement. Une mesure de corrélation assez facile à estimer à partir de données historiques, est donnée par la covariance entre deux variables aléatoires.

Définition 1.5. *La covariance de deux variables aléatoires X_1 et X_2 est la valeur réelle* $Cov(X_1, X_2) = E((X_1 - E(X_1))(X_2 - E(X_2))) = E(X_1 X_2) - E(X_1)E(X_2)$. *On appelle matrice de variance-covariance d'un vecteur aléatoire X la matrice W donnée par :* $\forall (i, j) \in [\![1; n]\!]^2, W_{ij} = Cov(X_i, X_j)$. *De manière*

à standardiser les lois marginales, on aura recours par la suite principalement à la notion de matrice de corrélation S, définie par $\forall (i,j) \in [\![1;n]\!]^2, S_{ij} = \frac{Cov(X_i,X_j)}{\sqrt{Var(X_i)Var(X_j)}}$.

La donnée de la matrice de variance-covariance d'un vecteur aléatoire fournit une information intéressante mais incomplète sur la manière dont les variables marginales sont corrélées. De manière générale, la théorie des copules permet de reconstruire à partir des lois marginales et d'une fonction appelée copule $C : [0,1]^n \to [0,1]$ toute fonction de répartition d'un vecteur aléatoire constitué de ces lois marginales, ce qu'on écrit comme $\forall (x_1,\dots,x_n) \in \mathbb{R}^n, F(x_1,\dots,x_n) = C(F_1(x_1),\dots,F_n(x_n))$. Cet aspect sera abordé plus en détail dans la section 3. Le seul point important à retenir pour la compréhension de la problématique de ce mémoire est de savoir que les familles de copules les plus classiques (et les plus utilisées par l'industrie) sont paramétrées par la matrice de corrélation du vecteur aléatoire dont les marginales sont les facteurs de risques et les facteurs de pertes de l'assureur.

Les deux termes ci-dessous seront utilisés de manière récurrente dans ce mémoire :

Définition 1.6. *On appelle facteur de pertes une variable aléatoire représentant directement la perte de l'assureur. Par exemple, la variable aléatoire représentant la variation des réserves de l'assureur en santé est un facteur de pertes. On appelle facteur de risques une variable aléatoire qui nécessite une transformation potentiellement complexe pour être reliée aux pertes potentielles de l'assureur. Par exemple, la variable aléatoire représentant le mouvement de la première composante principale de la courbe des taux EIOPA est un facteur de risques.*

1.2 Matrice PSD

Au delà des notions de probabilités de base exposées ci-dessus, il est nécessaire de rappeler quelques grandes propriétés des formes quadratiques et des matrices symétriques avant de poser formellement la problématique de ce mémoire.

Définition 1.7. *Une matrice* $M \in \mathbb{R}^{n\times n}$ *est dite symétrique ou auto-adjointe si* $\forall (i,j) \in [\![1;n]\!]^2, M_{ij} = M_{ji}$. *Une matrice de pseudo-corrélation G est une matrice symétrique à coefficients dans l'intervalle* $[-1;1]$ *telle que* $\forall i \in [\![1;n]\!], G_{ii} = 1$. *Une matrice de corrélation S est la matrice de variance-covariance d'un vecteur aléatoire.*

Il est important de noter que la définition ci-dessus est cohérente avec la définition donnée dans la section précédente. Une matrice de corrélation per-

met de synthétiser toute l'information (linéaire) sur la réalisation conjointe de variables corrélées linéairement, mais s'avère insuffisante dans des cas de corrélations plus complexes, avec par exemple des fortes dépendances de queues de distribution.

Définition 1.8. *La forme quadratique associée à une matrice* $M \in \mathbb{R}^{n\times n}$ *est la fonction* $q : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ *telle que* $\forall x \in \mathbb{R}^n, q(x) = x^T Mx$. *Une forme quadratique est dite définie si* $\forall x \in \mathbb{R}^n, q(x) = 0 \Rightarrow x = 0$. *Une forme quadratique est dite positive semi-définie, ou encore « PSD », si et seulement si* $\forall x \in \mathbb{R}^n, q(x) \geq 0$.

Définition 1.9. *Une matrice M est dite diagonalisable s'il existe une matrice inversible P et une matrice diagonale D telle que* $M = PDP^{-1}$. *Dans ce cas, les coefficients diagonaux de la matrice D sont appelés les valeurs propres de la matrice M, et les vecteurs colonnes de la matrice P sont appelés les vecteurs propres de la matrice M.*

Le théorème suivant est très classique en algèbre linéaire et fondamental pour ce mémoire :

Théorème 1.10. *Toute matrice symétrique* $M \in \mathbb{R}^{n\times n}$ *est diagonalisable dans une base orthonormée de* $\mathbb{R}$. *En d'autres termes,* $\exists P \in \mathbb{R}^{n\times n}$ *telle que* $PP^T = I_n$ *et* $M = PDiag(\lambda_1, \ldots, \lambda_n)P^T$.

Démonstration. Ce résultat se démontre par récurrence (forte) sur n. Le résultat est immédiat pour n=1. Pour $n \geq 2$, on suppose le résultat vrai jusqu'en n-1. L'endomorphisme u de $\mathbb{R}^n$ représenté par la matrice M a au moins une valeur propre complexe λ. L'endomorphisme u, considéré comme un endomorphisme de $\mathbb{C}^n$ étant également auto-adjoint ($\forall (z, z') \in \mathbb{C}^n \times \mathbb{C}^n, \langle z|u(z')\rangle = \langle x + iy|u(x' + iy')\rangle = \langle x|u(x')\rangle + i\langle y|u(x')\rangle - i\langle x|u(y')\rangle + \langle y|u(y')\rangle = \langle u(x)|x'\rangle + i\langle u(y)|x'\rangle - i\langle u(x)|y'\rangle + \langle u(y)|y'\rangle = \langle u(z)|z'\rangle$), on a pour $z \in \mathbb{C}^n$ le vecteur propre associé à la valeur propre λ : $\langle z, u(z)\rangle = \langle u(z), z\rangle$ soit $\overline{\lambda}\|z\|^2 = \lambda\|z\|^2$, donc $\lambda \in \mathbb{R}$. En tant qu'endomorphisme de $\mathbb{C}^n$, u a donc une valeur propre réelle. Les polynômes caractéristiques de u en tant qu'endomorphisme de $\mathbb{C}^n$ et de u en tant qu'endomorphisme de $\mathbb{R}^n$ sont les mêmes, donc l'endomorphisme u en tant qu'endomorphisme de $\mathbb{R}^n$ a également $\lambda \in \mathbb{R}$ comme valeur propre. Soit e_1 le vecteur propre réel associé à λ tel que $\|e_1\| = 1$. On peut alors considérer le sous-espace vectoriel F donné par $F = Vect(e_1)^{\perp}$. Comme u est autoadjointe, F est stable par u. On peut appliquer l'hypothèse de récurrence à la restriction de u à F. □

Le théorème suivant, également très classique, permet de faire le lien entre le caractère défini d'une matrice et ses valeurs propres :

Théorème 1.11. *Une forme quadratique q associée à une matrice M est positive semi-définie si et seulement si toutes les valeurs propres de M sont positives ou nulles. Cette forme quadratique est définie si et seulement si toutes les valeurs propres de M sont strictement positives.*

Enfin, le théorème suivant rappelle une propriété importante de toutes les matrices de corrélations.

Théorème 1.12. *Toute matrice de corrélation S est positive semi-définie.*

Démonstration. Soit S une matrice de corrélation, et X le vecteur aléatoire associé (réduit). Soit U un vecteur quelconque $\in \mathbb{R}^n$. Alors soit Y la variable aléatoire définie par $Y = U^T X = \sum_{i=1}^{n} u_i X_i$. La variance de Y vaut $V(Y) = Cov(Y,Y) = \sum_{(i,j)\in[\![1;n]\!]^2} u_i u_j Cov(X_i, X_j) = \sum_{(i,j)\in[\![1;n]\!]^2} u_i u_j S_{i,j} = U^T S U$. Or, $\forall U \in \mathbb{R}^n, V(U^T X) = V(Y) \geq 0$. Donc $\forall U \in \mathbb{R}^n, U^T S U \geq 0$. La matrice de corrélation S est donc positive semi-définie. □

En pratique, lors de l'agrégation des risques dans les modèles internes, les matrices de corrélation utilisées sont remplies en partie par des jugements d'experts et en partie par des formules de remplissage automatique. Les matrices ainsi obtenues sont donc des matrices de pseudo-corrélation n'ont donc aucune raison d'être positives semi-définies (« PSD »). Néanmoins, les copules utilisées par les assureurs afin d'agréger leurs risques doivent être paramétrées avec une matrice positive semi-définie pour pouvoir être simulées. Une étape dite de « PSDisation » de la matrice de pseudo-corrélation s'avère donc nécessaire avant de pouvoir calculer l'exigence de capital finale.

Enfin, le théorème suivant, attribué à Gerschgorin, permet de localiser les valeurs propres d'une matrice de pseudo-corrélation. Ce théorème indique donc que les valeurs propres d'une matrice de pseudo-corrélation ne peuvent pas être « trop négatives ». Le spectre d'une matrice A est noté Spec(A) :

Théorème 1.13. $\forall A \in \mathbb{C}^{n\times n}$, $\forall \lambda \in$ *Spec(A),* $\exists i \in [\![1;n]\!]$ *tel que* $\lambda \in \{z \in \mathbb{C}, |z - a_{ii}| \leq \sum_{j\neq i} |a_{ij}|\}$. *En particulier, si G est une matrice de pseudo-corrélation de dimension* $n \times n$, *toute valeur propre* λ *appartient à l'intervalle réel* $[2-n; n]$.

Ce thérorème signifie qu'en pratique, plus un assureur recherchera une granularité importante dans la modélisation des corrélations des risques auxquels il est exposé, plus il pourra potentiellement générer une matrice éloignée de la cible PSD. Ceci signifie que les algorithmes de PSDisation déformeront fortement la matrice, modifiant ainsi un grand nombre de corrélations fixées initialement par l'assureur.

1.3 Normes matricielles utilisées dans ce mémoire

Pour la suite du mémoire, les normes matricielles suivantes sur $\mathbb{R}^{n\times n}$ seront fréquemment utilisées :

Définition 1.14. *La « norme 2 » ou « norme de Frobenius » sur* $\mathbb{R}^{n\times n}$ *est notée* $\|.\|_2$ *et est définie par* $\forall A \in \mathbb{R}^{n\times n}, \|A\|_2^2 = \sum_{(i,j)\in[\![1;n]\!]^2} a_{ij}^2 = tr(AA^T)$.

Définition 1.15. *Pour toute matrice W définie positive de* $\mathbb{R}^{n\times n}$*, la « norme W » sur* $\mathbb{R}^{n\times n}$ *est notée* $\|.\|_W$ *et est définie par* $\forall A \in \mathbb{R}^{n\times n}, \|A\|_W = \|W^{\frac{1}{2}}AW^{\frac{1}{2}}\|_2$.

Définition 1.16. *Pour toute matrice* $H \in \mathbb{R}^{*+n\times n}$*, la « norme H » sur* $\mathbb{R}^{n\times n}$ *est notée* $\|.\|_H$ *et est définie par* $\forall A \in \mathbb{R}^{n\times n}, \|A\|_H^2 = \sum_{(i,j)\in[\![1;n]\!]^2} h_{ij}a_{ij}^2$.

On remarque que la « norme 2 » et la « norme H » coincident lorsque $H = [1]_{(i,j)\in[\![1;n]\!]^2}$. De plus, la « norme W » et la « norme H » coincident lorsque W est diagonale ($W = diag(w_i)_{i\in[\![1;n]\!]}$) et H est de rang 1 ($H = [(w_iw_j)^{\frac{1}{2}}]_{(i,j)\in[\![1;n]\!]^2}$).

1.4 Algorithmes d'optimisation classiques

Cette section décrit les grands principes des différentes familles d'algorithmes d'optimisation utilisés dans ce mémoire. Ces algorithmes vont s'avérer intéressants à double titre : d'une part pour trouver la matrice positive semi-définie la plus proche de la matrice de pseudo-corrélation initiale, et d'autre part les algorithmes d'optimisation génétiques vont permettre de réaliser des sensibilités « intelligentes » permettant de donner une bonne approximation des bornes [min ;max] auxquels le SCR appartient. Le problème que les algorithmes d'optimisation permettent de résoudre est le suivant :

Algorithmes d'optimisation Soit $f : V \subset \mathbb{R}^n \to \mathbb{R}$ une fonction sur V un ouvert de $\mathbb{R}^n$. On recherche $\min_{x\in V}(f(x))$ et ω tel que $f(\omega) = \min_{x\in V}(f(x))$. Lorsqu'un tel ω existe et est unique, on le note habituellement $\omega = argmin(f)$.

La recherche d'un maximum est équivalente conceptuellement, quitte à remplacer $f \leftarrow -f$. Les algorithmes du gradient et de Newton ici sont très classiques, et exposés par exemple dans [PET]. Les algorithmes génétiques, quant à eux, ont été introduits par l'auteur de [GOL].

1.4.1 Algorithmes linéaires de type gradient à pas optimal (sans contraintes)

Avant de décrire plus en détail cet algorithme, il est nécessaire de rappeler quelques définitions :

Définition 1.17. *Une fonction $f : V \subset \mathbb{R}^n \to \mathbb{R}$ est dite différentiable au point $a \in V$ s'il existe une application linéaire $df_a \in \mathcal{L}(\mathbb{R}^n)$ telle que $\forall h \in \mathbb{R}^n$, $f(a+h) = f(a) + df_a(h) + o(\|h\|)$. L'unique vecteur correspondant canoniquement à cette application linéaire df_a dans le bidual $\mathbb{R}^{n**}$ est appelé le gradient de f en a et noté $\nabla f(a) = (\frac{\partial f(a)}{\partial x_i})_{i \in [\![1;n]\!]}$.*

Voici un algorithme préliminaire appelé « méthode de la section dorée » qui sera utile dans l'algorithme du gradient à pas optimal. Cet algorithme est appelé ainsi car il nécessite l'utilisation du nombre d'or $\phi = \frac{\sqrt{5}+1}{2}$ (et l'opposé de son conjugué $\tau = -\overline{\phi} = \frac{\sqrt{5}-1}{2}$ dans le corps de rupture $\mathbb{Q}[\sqrt{5}]$) :

Algorithme 1.18. *(Algorithme de la section dorée). Soit $f : [a,b] \to \mathbb{R}$ une fonction réelle définie sur un compact. Implémenter les étapes suivantes :*

1. *Subdiviser l'intervalle $[a,b]$ en trois intervalles : $[a,(1-\tau)(b-a)]$, $[(1-\tau)(b-a),\tau(b-a)]$ et $[\tau(b-a),b]$. Si $f((1-\tau)(b-a)) > f(\tau(b-a))$ alors $[a,b] \leftarrow [(1-\tau)(b-a),b]$. Sinon, $[a,b] \leftarrow [a,\tau(b-a)]$.*
2. *Répéter l'étape 1 jusqu'à ce que la largeur des intervalles soit inférieure à une tolérance fixée préalablement ou jusqu'à un nombre d'itérations maximum.*

Ceci permet de mettre en place l'algorithme suivant faisant appel à la dérivée :

Algorithme 1.19. *(Algorithme du gradient à pas optimal). Soit $f : \mathbb{R}^n \to \mathbb{R}$ une fonction réelle définie sur $\mathbb{R}^n$. Implémenter les étapes suivantes :*

1. *Choisir un point initial $x \leftarrow x_0 \in \mathbb{R}^n$.*
2. *Remplacer $x \leftarrow x - p\nabla f(x)$ en choisissant $p \in \mathbb{R}$ comme minimum de la fonction $g : \mathbb{R} \to \mathbb{R}$ telle que $\forall p \in \mathbb{R}, g(p) = f(x - p\nabla f(x))$, à l'aide d'un algorithme de section dorée.*
3. *Répéter l'étape 2 jusqu'à ce que la norme du gradient $\|\nabla f(x)\|$ soit inférieure à une tolérance fixée préalablement ou jusqu'à un nombre d'itérations maximum.*

Cet algorithme peut être visualisé de la manière suivante dans le plan, pour le passage d'un point « 1 » à un point « 2 » (en particulier, l'optimisation du pas retenu permet d'éviter les phénomènes de « rebonds » qui peuvent avoir lieu avec un algorithme du gradient à pas constant) :

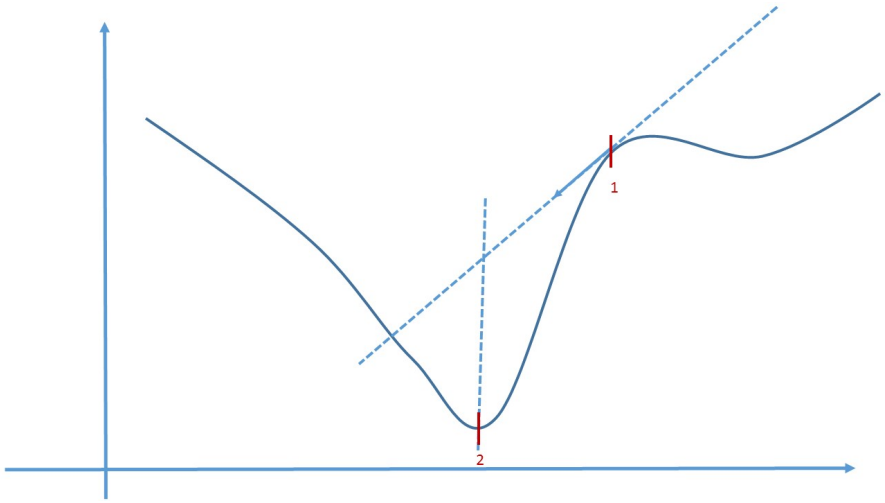


FIGURE 3 – Illustration d'une itération de l'algorithme du gradient à pas optimal pour une courbe plane (passage d'un point 1 à un point 2)

Cet algorithme du gradient à pas optimal (sans contraintes) présente les avantages et les inconvénients suivants :

Avantages Cet algorithme est très maniable et facile à implémenter lorsqu'il est possible de calculer le gradient de la fonction qu'on cherche à optimiser ;

Inconvénients Néanmoins cet algorithme requiert la différentiabilité de la fonction à optimiser. De plus, en cas de « mauvais » choix du point initial x_0, cet algorithme peut converger vers un minimum local, et non pas global. En pratique, il peut être utile de rechercher le point x_0 en cherchant à minimiser de manière empirique la valeur $f(x_0)$ (quitte à rechercher x_0 sous une forme particulière et choisir parmi $(x_0^i)_{i\in[\![1;n]\!]}$).

1.4.2 Algorithmes quadratiques (sans contraintes)

La définition suivante est utile en préambule de la présentation de l'algorithme de Newton :

Définition 1.20. *On appelle Hessien de la fonction f au point $a \in V$ la matrice symétrique de $\mathbb{R}^{n\times n}$ suivante : $\nabla^2 f(a) = (\frac{\partial^2 f(a)}{\partial x_i \partial x_j})_{(i,j)\in[\![1;n]\!]^2}$.*

L'algorithme suivant repose sur l'existence du Hessien (et son inversibilité) :

Algorithme 1.21. *(Algorithme de Newton). Soit $f : \mathbb{R}^n \to \mathbb{R}$ une fonction réelle définie sur $\mathbb{R}^n$. Implémenter les étapes suivantes :*

1. *Choisir un point initial* $x = x_0 \in \mathbb{R}^n$.
2. *Remplacer* $x \leftarrow x - (\nabla^2 f(x))^{-1} \nabla f(x)$.
3. *Répéter l'étape 2 jusqu'à ce que la norme du gradient* $\|\nabla f(x)\|$ *soit inférieure à une tolérance fixée préalablement ou jusqu'à un nombre d'itérations maximum.*

Cet algortihme de Newton présente les avantages et les inconvénients suivants :

Avantages Cet algorithme converge rapidement lorsque le Hessien existe en tout point et est inversible en tout point ;

Inconvénients Néanmoins les conditions pour implémenter cet algorithme sont relativement fortes (2-différentiabilité et inversibilité). De plus, les problèmes associés au choix du point initial x_0 sont maintenus.

Il existe une version « allégée » de l'algorithme de Newton. La présentation de celle-ci nécessite une première définition :

Définition 1.22. *On appelle conditions de Wolfe de paramètres* c_1 *et* c_2*, avec* $0 < c_1 < c_2 < 1$*, pour* $(x, l, p) \in \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^n$ *les conditions suivantes :*

$$\begin{cases} f(x + lp) \leq f(x) + c_1 l \nabla f(x)^T p \\ \nabla f(x + lp)^T p \geq c_2 \nabla f(x)^T p \end{cases}$$

Cette définition permet la description suivante de l'algorithme d'optimisation (le choix des paramètres des conditions de Wolfe c_1 et c_2 étant laissé libre, en respectant $0 < c_1 < c_2 < 1$) :

Algorithme 1.23. *(Algorithme de quasi-Newton). Soit* $f : \mathbb{R}^n \to \mathbb{R}$ *une fonction réelle définie sur* $\mathbb{R}^n$*. Implémenter les étapes suivantes :*

1. *Choisir un point initial* $x = x_0 \in \mathbb{R}^n$.
2. *Remplacer* $x \leftarrow x - B^{-1} \nabla f(x)$ *avec* B *une matrice symétrique positive mise à jour à chaque itération. Dans l'algorithme BFGS (Broyden, Fletcher, Goldfarb, Shanno), la matrice* B *est mise à jour sur la base des formules suivantes (avec un pas* l *qui respecte les conditions de Wolfe de paramètres* c_1 *et* c_2*) :*

$$\begin{cases} x' \leftarrow x - l B^{-1} \nabla f(x) \\ s \leftarrow x' - x \\ y \leftarrow \nabla f(x') - \nabla f(x) \\ \gamma \leftarrow \frac{1}{y^T s} \\ B^{-1} \leftarrow (I - \gamma s y^T) B^{-1} (I - \gamma s y^T) + \gamma s s^T \end{cases}$$

3. *Répéter l'étape 2 jusqu'à ce que la norme du gradient* $\|\nabla f(x)\|$ *soit inférieure à une tolérance fixée préalablement ou jusqu'à un nombre d'itérations maximum.*

Comme l'algorithme de Newton, cet algorithme présente de bonnes propriétés en termes de vitesse de convergence mais ne permet pas de répondre aux difficultés rencontrées par l'algorithme du gradient (convergence possible vers des minima locaux selon le point initial choisi).

1.4.3 Algorithme d'optimisation avec contraintes inégalités

Sans rentrer dans les détails de la théorie des algorithmes d'optimisation avec contraintes (et en particulier en écartant la théorie autour des conditions nécessaires pour la convergence des algorithmes), voici un algorithme dit « dual » utile dans les cas d'optimisation sous k contraintes inégalités que l'on note $c(x) < 0$ avec $c : \mathbb{R}^n \to \mathbb{R}^k$. La description de l'algorithme considéré demande une première définition :

Définition 1.24. *On appelle Lagrangien (respectivement multiplicateurs de Lagrange) la fonction $\mathcal{L} : V \times \mathbb{R}^k \to \mathbb{R}$ (respectivement les paramètres $\lambda \in \mathbb{R}^k$) définie par :*

$$\forall (x, \lambda) \in V \times \mathbb{R}^k,\ \mathcal{L}(x, \lambda) = f(x) + \lambda^T c(x).$$

Ceci permet la définition de l'algorithme ci-dessous :

Algorithme 1.25. *(Algorithme d'Uzawa). Implémenter les étapes suivantes :*

1. *Choisir $x \leftarrow x_0 \in \mathbb{R}^n$, $\lambda \leftarrow \lambda_0 \in \mathbb{R}^k$, et $\alpha \in \mathbb{R}^+$;*
2. *A l'aide d'un algorithme d'optimisation, résoudre le problème d'optimisation $\min_{x \in \mathbb{R}^n} \mathcal{L}(x, \lambda)$ et implémenter $x \leftarrow argmin(\mathcal{L}(x, \lambda))$ (avec λ fixé) ;*
3. *Implémenter $\lambda \leftarrow P(\lambda + \alpha c(x))$ en notant $P : \mathbb{R}^k \to (\mathbb{R}^+)^k$ la projection sur $(\mathbb{R}^+)^k$.*
4. *Répéter les étapes 2 et 3 jusqu'à ce que la distance entre 2 itérations $\|\lambda' - \lambda\|$ soit inférieure à une certaine tolérance ou jusqu'à un nombre d'itérations maximales.*

Néanmoins, l'interprétation des résultats de cet algorithme est plus complexe que pour les algorithmes sans contraintes, et les problèmes de convergence vers des minimas locaux n'en sont que plus forts (plusieurs appels à un algorithme d'optimisation).

1.4.4 Algorithmes génétiques (avec ou sans contraintes)

Les algorithmes génétiques reposent conceptuellement sur une imitation des phénomènes d'évolution naturelle, conformément à la théorie darwinienne de l'évolution. Les étapes suivantes doivent être modélisées dans cette famille d'algorithmes :

Algorithme 1.26. *(Algorithme génétique). Soit $f : \mathbb{R}^n \to \mathbb{R}$ une fonction réelle définie sur $\mathbb{R}^n$. Implémenter les étapes suivantes :*

1. *(Création population initiale) : Cette étape est basée sur un tirage aléatoire d'une population $(x_i)_{i\in[\![1;N]\!]} \in V^N$ avec N donné, correspondant à la taille de la population initiale.*

FIGURE 4 – Création de la population initiale

2. *(Classement de la population) : La fonction f que l'on cherche à optimiser est vue comme une fonction d'évaluation ou de survie, permettant de classer les individus selon leur aptitude à survivre. Comme on cherche ici à minimiser la fonction f, on considère que l'individu x_i avec les meilleures dispositions pour survivre est l'individu qui dispose de l'évaluation $f(x_i)$ la plus basse possible.*

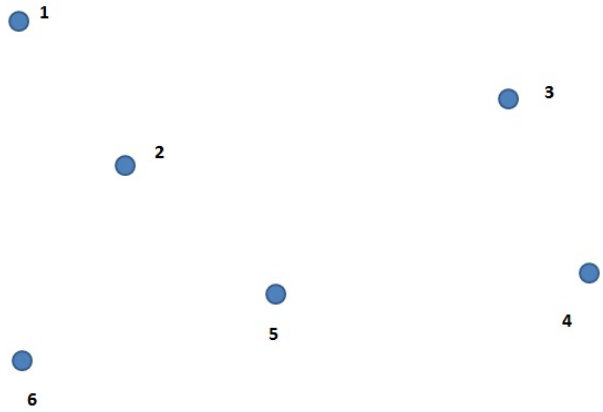


FIGURE 5 – Classement de la population

3. *(Reproduction de deux individus) : Cette étape consiste à faire se reproduire les individus deux à deux, en les sélectionnant selon leur aptitude à survivre (le meilleur et le second se reproduisent ensemble, puis le troisième et le quatrième et ainsi de suite), et permettant de donner deux descendants à chaque couple. Dans chaque cas, il faut donner un sens mathématique à cette reproduction. Par exemple, si les individus sont des matrices, il est possible de considérer qu'une reproduction de deux individus consiste à créer deux matrices filles, en intercroisant les deux matrices mères (après un découpage préalable de chacune des matrices mères en deux blocs, constitués par exemple des j premières colonnes et des n-j suivantes).*

FIGURE 6 – Reproduction de deux individus

4. *(Mutation) : Dans le processus de reproduction, il est important de prévoir une probabilité de mutation* $0.05 < p_m < 0.1$ *de manière à ce que l'algorithme ne converge pas vers des extremas locaux. Cette probabilité de mutation ne doit pas être trop élevée, sans quoi l'algorithme n'aura plus d'autre sens que celui d'une sensibilité de la fonction f sur un tirage totalement aléatoire. La mutation est un phénomène aléatoire permettant de modifier des composants d'un individu lors de la reproduction. Dans l'exemple où la population est une population de matrice, il est possible par exemple de tirer un coefficient au hasard dans la matrice et de le remplacer par un nombre aléatoire.*

FIGURE 7 – Mutation aléatoire d'un individu

5. *(Evolution de la population) : Plusieurs choix sont possibles pour faire évoluer la population : doublement de la taille de la population à chaque étape ou bien sélection uniquement d'un certain nombre d'individus, en fonction de leur aptitude à la survie. Dans ce mémoire, la population toute entière sera conservée à chaque étape, de manière à pouvoir déterminer le minimum et le maximum de la fonction dans le même algorithme (par symétrie).*

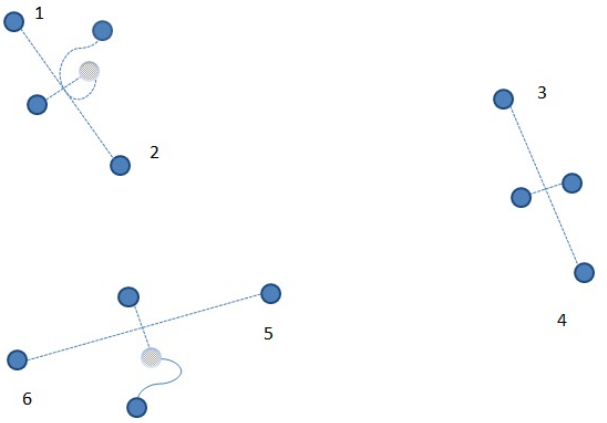


FIGURE 8 – Evolution de la population

6. *(Sélection de l'individu le plus apte à survivre) : Enfin, le minimum de la fonction est retenu comme étant l'individu le plus apte à l'issue d'un*

nombre K d'itérations (i.e. l'individu avec la fonction d'évaluation la plus basse sur la population). Le critère d'arrêt de la boucle peut être défini par un nombre maximum d'itérations en absolu ou encore un arrêt lorsque le même minimum est retenu un certain nombre de fois consécutives.

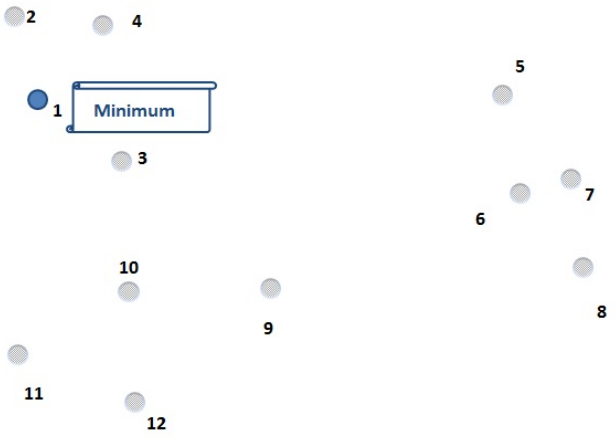


FIGURE 9 – Sélection du survivant

Les algorithmes génétiques présentent les avantages et les inconvénients suivants :

Avantages Ces algorithmes présentent des bonnes propriétés de convergence dans certains cas simples (fonctions convexes etc.). En pratique, les algorithmes génétiques sont fort utiles lorsque notre connaissance de la fonction f à minimiser est très mauvaise, et notamment lorsque le calcul du gradient et/ou du hessien ne sont pas possibles. Par rapport aux algorithmes linéaires et quadratiques, les algorithmes génétiques permettent un meilleur traitement des minima locaux (notamment grâce au phénomène de mutation). Par exemple, pour minimiser une fonction de deux variales dont l'image serait une surface de la forme d'une « boîte d'oeufs » un algorithme génétique fournira une bien meilleure solution qu'un algorithme du gradient qui sera rapidement piégé par un minimum local ;

Inconvénients Néanmoins, si ces algorithmes permettent de donner une bonne évaluation du minimum, ils ne permettent pas de garantir l'atteinte du minimun dans le cas général (en particulier pour les fonctions qui seront étudiées dans le mémoire). De plus, ces algorithmes ne permettent pas de garantir que l'individu sélectionné est bien le point en lequel le minimum est atteint. Il n'est donc pas possible de tirer des conclusions fiables sur l'argmin.

1.5 Problématique de ce mémoire

Les quelques définitions et théorèmes présentés dans les précédentes sections permettent de poser complètement la problématique de ce mémoire :

Objectif du mémoire Sur plusieurs exemples numériques choisis (choix de la matrice, choix de la pondération, choix des facteurs de risques ou de pertes, choix de la copule), estimer à l'aide d'algorithmes génétiques l'intervalle de type [min ; max] auquel le SCR appartiendrait en effectuant des modifications théoriquement neutres de la matrice initiale de corrélation (comme l'intervertion de l'ordre des facteurs de risques) ou très légères de la matrice de pondération (incertitude des coefficients de pondération dans l'algorithme de PSDisation).

Remarque : Il est important de noter que par simplicité, dans la suite du mémoire, l'intervalle noté [min ; max] signifiera (lorsqu'applicable) : « l'intervalle défini par la plus petite et la plus grande valeur obtenues par un calcul de l'algorithme génétique ». Cette notation ne doit pas induire en erreur : compte-tenu de la compléxité des fonctions sur lesquelles seront appliqués les algorithmes génétiques, il ne sera pas possible de garantir que les valeurs obtenues soit effectivement des extrema absolus (pas de garantie de convergence des algorithmes génétiques).

2 PSDisation d'une matrice de pseudo-corrélation

L'objectif de cette section est de présenter plus en détail les principaux algorithmes existants sur le marché pour rendre une matrice positive semi-définie, et d'expliquer le choix de l'algorithme finalement retenu. Il est à noter que dans le cadre d'un modèle interne, les assureurs peuvent développer d'autres algorithmes propres à leur société, en fonction de leurs besoins. En pratique, les algorithmes retenus sont souvent inspirés des algorithmes présentés ci-dessous.

2.1 Algorithme des projections alternées

Cet algorithme d'optimisation linéaire a été défini dans [HIG]. Il permet de trouver la matrice de corrélation S la plus proche d'une matrice de pseudo-corrélation initiale $G \in [-1;1]^{n\times n}$ donnée, en « norme W ». L'interprétation géométrique de cet algorithme nécessite les définitions suivantes :

Définition 2.1. *On note $\mathcal{U}$ l'ensemble des matrices symétriques à coefficients diagonaux unitaires $\mathcal{U} = \{Y \in \mathbb{R}^{n\times n} | Y = Y^T, \forall i \in [\![1;n]\!],\ y_{ii} = 1\}$ et on note $\mathcal{S}^+$ l'ensemble des matrices symétriques positives $\mathcal{S}^+ = \{Y \in \mathbb{R}^{n\times n}, Y = Y^T, Y \geq 0\}$.*

De plus, l'opérateur suivant sera nécessaire :

Définition 2.2. *Lorsque la confusion n'est pas possible, on appelle et note indistinctement (à une majuscule près) l'opérateur diagonal les deux opérateurs suivants :*

- *$diag : \mathbb{R}^{n\times n} \to \mathbb{R}^n$, défini par $\forall A \in \mathbb{R}^{n\times n}, diag(A) = [a_{ii}]_{i\in[\![1;n]\!]}$;*
- *$Diag : \mathbb{R}^n \to \mathbb{R}^{n\times n}$, défini par $\forall v \in \mathbb{R}^n, Diag(v) = [m_{ij}]_{(i,j)\in[\![1;n]\!]^2}$, tel que $\forall i \in [\![1;n]\!], m_{ii} = v_i$, et $\forall i \neq j, m_{ij} = 0$.*

Algorithme 2.3. *(Algorithme des projections alternées). Implémenter les étapes suivantes :*

1. *Initialiser l'algorithme avec $S \leftarrow G$ et $\Delta S \leftarrow 0$.*
2. *Appliquer la correction dite « de Dykstra » à la matrice $S \leftarrow S - \Delta S$.*
3. *En notant $P_{\mathcal{S}^+}$ l'opérateur de projection sur $\mathcal{S}^+$ donné par la relation suivante (en « norme W ») :*

$$P_{\mathcal{S}^+}(A) = W^{\frac{-1}{2}}(W^{\frac{1}{2}}AW^{\frac{1}{2}})_+W^{\frac{-1}{2}}$$

où $(.)_+$ représente l'opérateur suivant (λ désigne les valeurs propres de la matrice symétrique A, et Q la matrice symétrique de diagonalisation de A dans une base orthogonale) :

$$A_+ = QDiag(max(\lambda_i; 0))Q^T.$$

Implémenter :

$$\begin{cases} \Delta S \leftarrow S - P_{\mathcal{S}^+}(S) \\ S \leftarrow P_{\mathcal{S}^+}(S). \end{cases}$$

4. *En notant $P_{\mathcal{U}}$ l'opérateur de projection sur $\mathcal{U}$ donné par la relation suivante (en « norme W ») :*

$$P_{\mathcal{U}}(A) = A - W^{-1} Diag(\mu) W$$

où $\mu = (\mu_1, \ldots, \mu_n)^T$ est une solution du système linéaire suivant : $(W^{-1} \circ W^{-1})\mu = diag(A - I)$. I représente la matrice identité de $\mathbb{R}^{n \times n}$ et $\circ$ représente le produit de Hadamard de deux matrices : $(A \circ B)_{ij} = A_{ij} B_{ij}$. Implémenter :

$$S \leftarrow P_{\mathcal{U}}(S).$$

5. *Itérer les étapes 2 à 4 jusqu'à ce que $\frac{\|S_k - S_{k-1}\|}{\|S_k\|}$ soit inférieur à un certain seuil de tolérance ou que le nombre d'itérations k soit inférieur à un nombre maximal.*

Cet algorithme correspond à des projections alternées de la matrice de pseudo-corrélation G sur les ensembles $\mathcal{S}^+$ et $\mathcal{U}$ (avec une petite correction ΔS permettant une meilleure convergence (voir [HIG]). Conceptuellement, la matrice recherchée est à l'intersection de deux ensembles convexes fermés, dont l'intersection est donc également convexe et fermée. La projection de la matrice de pseudo-corrélation sur $\mathcal{S}^+ \cap \mathcal{U}$ est donc unique (voir par exemple le théorème II.2.9 de [PCO]). En résumé, l'algorithme des projections alternées présente les avantages et inconvénients suivants :

Avantages Cet algorithme est très simple conceptuellement et permet de démontrer l'existence et l'unicité de la solution au problème d'optimisation (dans un cas très simple sans pondération ni encadrement) ;

Inconvénients Néanmoins cet algorithme ne permet pas la pondération selon la norme H qui peut intéresser les assureurs. De plus, cet algorithme n'est pas facilement transposable dans le cas de contraintes inégalités (encadrement).

2.2 Algorithme de Newton

Cet algorithme a été introduit par [QIS]. Il nécessite dans un premier temps la présentation de l'algorithme dit du gradient conjugué, qui permet de calculer la solution d'une équation linéaire de type $Ax = b$ sans devoir connaitre l'expression de la matrice A, et sans devoir calculer son inverse (voir [PET]) :

Algorithme 2.4. *(Algorithme du gradient conjugué). Implémenter les étapes suivantes :*

1. *Choisir $x \leftarrow x_0 \in \mathbb{R}^n$ quelconque, et implémenter $r \leftarrow Ax - b$ et $p \leftarrow -r$.*
2. *Implémenter les calculs suivants (qui ne nécessitent pas la connaissance de la matrice A mais seulement la capacité à calculer Ax) :*

$$\begin{cases} l \leftarrow \frac{r^T r}{p^T A p} \\ x \leftarrow x + lp \\ r' \leftarrow r + lAp \\ \beta \leftarrow \frac{\|r'\|^2}{\|r\|^2} \\ r \leftarrow r' \\ p \leftarrow -r + \beta p \end{cases}$$

3. *Itérer l'étape 2 jusqu'au respect d'un critère de tolérance préalablement fixé (voir algorithme 2.5).*

Ceci permet la mise en place de l'algorithme suivant :

Algorithme 2.5. *(Algorithme de PSDisation de Newton). Implémenter les étapes suivantes :*

1. *Choisir des paramètres $y \leftarrow y_0 \in \mathbb{R}^n$, $\eta \in]0;1[$, $\rho \in]0, \frac{1}{2}[$ et $\sigma \in]0, \frac{1}{2}[$.*
2. *Appliquer l'algorithme du gradient conjugué (voir algorithme 2.4) pour trouver une solution d au problème 3 ci-dessous :*

$$\begin{cases} \text{Approximer la solution de } \nabla\theta(y) + Vd = 0 \\ \text{tel que } \|\nabla\theta(y) + Vd\| \leq \eta\|\nabla\theta(y)\| \text{ et } \nabla\theta(y)^T d \leq -\eta\|d\|^2 \end{cases} \quad (3)$$

Avec V une matrice telle que $\forall d \in \mathbb{R}^n$,

$$Vd = diag(P(M \circ (P^T Diag(d) P))P^T)$$

où M est une matrice de la forme suivante :

$$M = \begin{pmatrix} E_{\alpha\alpha} & E_{\alpha\beta} & (\tau_{ij})_{i\in\alpha, j\in\gamma} \\ E_{\beta\alpha} & (\omega_{ij})_{i\in\beta, j\in\beta} & 0 \\ (\tau_{ji})_{i\in\alpha, j\in\gamma} & 0 & 0 \end{pmatrix}$$

Dans l'expression de M ci-dessus, P est une matrice orthogonale qui diagonalise la matrice $C(y) = G + Diag(y)$ (sous la forme : $C(y) = PDiag(\lambda)P^T$), α représente le nombre de valeurs propres strictement positives de la matrice $C(y)$, β représente le nombre de valeurs propres nulles de la matrice $C(y)$, γ représente le nombre de valeurs propres strictement négatives de la matrice $C(y)$, $(\omega_{ij})_{i\in\beta, j\in\beta}$ représente une matrice symétrique appartenant à $[0;1]^{\beta\times\beta}$, et $\tau_{ij} = \frac{\lambda_i}{\lambda_i - \lambda_j}$ où λ_i représentent les valeurs propres strictement positives de la matrice $C(y)$ et

λ_j les valeurs propres strictement négatives de la matrice C(y). Enfin, $E_{\alpha\beta}$ représente la matrice de dimension $\alpha \times \beta$ ne contenant que des 1. La notation $\circ$ représente le produit de Hadamard de deux matrices. De plus, dans les équations 3 ci-dessus, le vecteur gradient de la fonction $\theta : y \to \frac{1}{2}\|G + Diag(y)\|_2^2 - \left\langle (1,\ldots,1)^T \middle| y \right\rangle$ s'exprime comme :

$$\nabla\theta(y) = diag((G + Diag(y))_+) - \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}.$$

S'il n'est pas possible de trouver une solution d au problème défini par les équations 3 posées ci-dessus, implémenter $d \leftarrow -B^{-1}\nabla\theta(y)$ où B est une matrice symétrique définie positive (qui peut changer à chaque itération de l'algorithme).

3. *Trouver le plus petit entier $m \in \mathbb{N}$ tel que :*

$$\theta(y + \rho^m d) - \theta(y) \leq \sigma\rho^m \nabla\theta(y)^T d$$

puis implémenter $y \leftarrow y + \rho^m d$.

4. *Itérer les étapes 2 et 3 ci-dessus jusqu'à ce que $\|\nabla\theta(y)\|$ soit inférieur à une tolérance donnée ou jusqu'à un nombre d'itérations maximal.*

En résumé, l'algorithme de « PSDisation » de Newton présente les avantages et inconvénients suivants :

Avantages Cet algorithme converge plus rapidement que l'algorithme 2.3 des projections alternées ;

Inconvénients Néanmoins, cet algorithme (dans la version présentée dans ce mémoire) n'utilise que la norme 2. L'extension de cet algorithme à d'autres normes nécessite des complications importantes qui augmentent considérablement le temps de calcul (voir [QIS2] : optimisation dans l'optimisation etc.). De plus, cet algorithme est globalement plus compliqué et plus difficile à interpréter que les algorithmes des projections alternées (voir algorithme 2.3) ou hypersphère (voir algorithme 2.9).

2.3 Un algorithme utilisé par les banques et les assurances : Rebonato-Jäckel

Le principe de cet algorithme a été défini dans [REB]. Cet algorithme est basé sur une mise en forme astucieuse pour les matrices de corrélation, qui repose sur l'interprétation géométrique suivante pour décrire les coordonnées d'un vecteur unitaire située sur une hypersphère (une sphère de rayon 1 en dimension n) :

Définition 2.6. *Soit v un vecteur de $\mathbb{R}^{n\times n}$ tel que $\|v\| = 1$. Alors il existe un vecteur $(\theta_1, \theta_2, \ldots, \theta_n) \in [0; 2\pi]$ appelé coordonnées hypersphériques tel que les coordonnées cartésiennes $(v_1, v_2, \ldots, v_n)$ du vecteur v vérifient :*

$$\begin{cases} v_1 = \cos(\theta_1) \\ v_2 = \cos(\theta_2)\sin(\theta_1) \\ \vdots \\ v_j = \cos(\theta_j) \cdot \prod_{k=1}^{j-1} \sin(\theta_k) \\ \vdots \\ v_{n-1} = \cos(\theta_{n-1}) \cdot \prod_{k=1}^{n-2} \sin(\theta_k) \\ v_n = \prod_{k=1}^{n-1} \sin(\theta_k). \end{cases}$$

Ce type de coordonnées est très utilisé en dimension 3, mais le plus souvent sous la forme de coordonnées sphériques (comme cela est illustré dans la figure suivante). Dans ce cas, l'ordre des variables est interchangé par rapport à notre définition ($z = \cos(\theta), x = \cos(\phi)\sin(\theta), y = \sin(\phi)\sin(\theta)$), et les variables ϕ et θ sont appelées respectivement nutation et précession. Le changement de l'ordre des variables ne pose pas de difficultés particulières dans notre cas car la notion d'orientation ne sera pas nécessaire.

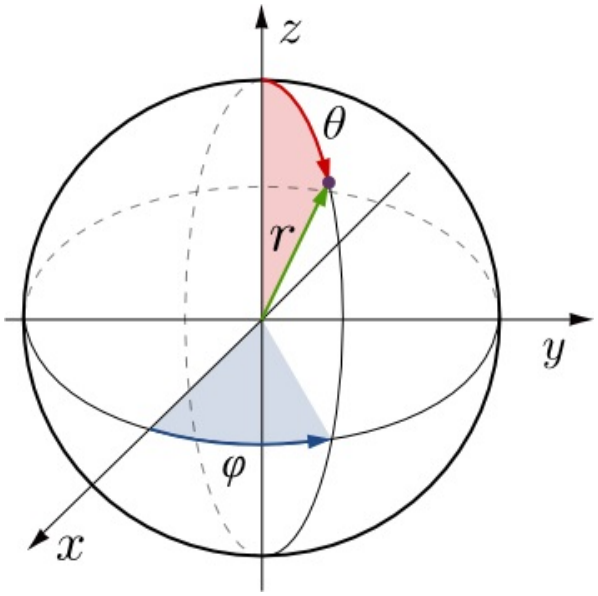


FIGURE 10 – Coordonnées sphériques en dimension 3 (avec un rayon quelconque)

L'écriture des vecteurs dans un système de coordonnées hypersphériques a inspiré la mise en forme suivante pour certaines matrices :

Définition 2.7. *Une matrice $M \in \mathbb{R}^n$ est dite sous forme d'hypersphère lorsqu'elle peut s'écrire sous la forme suivante :*

$$M = BB^T$$

avec la matrice $B \in \mathbb{R}^{n\times n}$ dont les coefficients peuvent s'écrire sous la forme suivante :

$$\begin{cases} \forall (i,j) \in [\![1;n]\!] \times [\![1;n-1]\!], B_{ij} = \cos(\theta_{ij}) \prod_{k=1}^{j-1} \sin(\theta_{ik}); \\ \forall i \in [\![1;n]\!], B_{in} = \prod_{k=1}^{n-1} \sin(\theta_{ik}). \end{cases}$$

On dit alors que la matrice B est une « décomposée hypersphère » de la matrice M, et on dit que $\theta \in [0, 2\pi]^{n\times(n-1)}$ est un vecteur angulaire associé à la décomposition hypersphère de M.

Le théorème suivant est très intéressant d'un point de vue théorique pour la suite de ce mémoire. En pratique, il est aisé numériquement de considérer que toute matrice positive semi-définie est en réalité définie, quitte à modifier très légèrement les valeurs propres nulles (autrement dit $GL_n(\mathbb{R})$ est dense dans l'espace des matrices $\mathbb{R}^{n\times n}$).

Théorème 2.8. *Toute matrice de corrélation définie S peut être mise sous forme d'hypersphère. De plus, le vecteur angulaire θ associé à la décomposition hypersphère de S est unique si les conditions suivantes sont respectées :*

$$\begin{cases} \forall (i,j) \in [\![1;n]\!] \times [\![1;n-1]\!], \theta_{ij} \in [0, \pi]; \\ \forall i \leq j, \theta_{ij} = 0. \end{cases}$$

Démonstration. La démonstration se fait en deux étapes (analyse - synthèse) :

Analyse (existence) : S étant une matrice symétrique définie positive (en tant que matrice de corrélation), elle représente un produit scalaire canonique dans la base $\mathcal{B} = (e_1, e_2 \dots e_n)$ de $\mathbb{R}^n$. Le procédé d'orthonormalisation de Schmidt (voir partie 18 de [PCO]) garantit l'existence et l'unicité d'une base orthonormée $\mathcal{B}' = (e'_1, e'_2 \dots e'_n)$ telle que : $\forall k \in [\![1;n]\!], Vect(e'_1, e'_2 \dots e'_n) = Vect(e_1, e_2 \dots e_n)$ et telle que $\forall k \in [\![1;n]\!]$, le produit scalaire $\langle e'_k | e_k \rangle$ est strictement positif. En notant P la matrice de passage de la base $\mathcal{B}$' à $\mathcal{B}$, on peut écrire : $S = P^T P$ avec P une matrice triangulaire supérieure. Pour la suite, on considérera la matrice $L = P^T$ qui est triangulaire inférieure. La matrice S s'écrit donc sous la forme : $S = LL^T$. En utilisant le fait que S est une matrice de corrélation, on a : $\forall i \in [\![1;n]\!], S_{ii} = 1$, ce qui revient, en notant l_i les lignes de la matrice L, à $\|l_i\| = 1$. Les lignes de L étant des vecteurs de norme 1 dans $\mathbb{R}^n$, il est possible de les écrire en coordonnées sphériques, qu'on notera $(\theta_{i1}, \theta_{i2}, \dots, \theta_{in})$ pour chaque ligne l_i. Ceci revient exactement à la mettre S sous forme d'hypersphère (en notant B := L).

Synthèse (unicité) : La démonstration de l'unicité se fait en deux temps :

1. Le procédé d'orthonormalisation de Schmidt garantit l'unicité de la matrice triangulaire inférieure à coefficients diagonaux strictement positif L telle que : $S = LL^T$;
2. On montre ensuite l'unicité du vecteur angulaire θ lorsque les conditions du thérorème sont respectées. En effet, la matrice L est triangulaire inférieure de coefficient diagonaux égaux à 1. On écrit la matrice L sous la forme :

$$\begin{cases} \forall (i,j) \in [\![1;n]\!] \times [\![1;n-1]\!], L_{ij} = \cos(\theta_{ij}) \prod_{k=1}^{j-1} \sin(\theta_{ik}); \\ \forall i \in [\![1;n]\!], L_{in} = \prod_{k=1}^{n-1} \sin(\theta_{ik}). \end{cases}$$

ce qui peut se représenter plus visuellement sous la forme (en tenant compte des contraintes sur le vecteur angulaire θ) :

$$\begin{pmatrix} L_{11} & 0 & & & 0 \\ L_{21} & L_{22} & 0 & \cdots & 0 \\ \vdots & & \ddots & & \vdots \\ L_{n1} & & \cdots & & L_{nn} \end{pmatrix} =$$

$$\begin{pmatrix} 1 & 0 & & \cdots & 0 \\ \cos(\theta_{21}) & \sin(\theta_{21}) & 0 & \cdots & 0 \\ \cos(\theta_{31}) & \cos(\theta_{32})\sin(\theta_{31}) & \sin(\theta_{32})\sin(\theta_{31}) & 0 \quad \cdots & 0 \\ \vdots & & & \ddots & \vdots \\ \cos(\theta_{n1}) & & \cdots & & \prod_{k=1}^{n-1} \sin(\theta_{nk}) \end{pmatrix}$$

Alors, en analysant la deuxième ligne de L, on voit que la contrainte $L_{22} > 0$ implique l'unicité de $\theta_{21} \in [0, \pi]$ tel que $\cos(\theta_{21}) = L_{21}$ et tel que $\sin(\theta_{21}) = L_{22} > 0$. De la même manière, en analysant la troisième ligne de L, on choisit l'unique $\theta_{31} \in [0, \pi]$ tel que $\cos(\theta_{31}) = L_{31}$ et $\sin(\theta_{31}) > 0$. Ensuite, on choisit l'unique $\theta_{32} \in [0, \pi]$ tel que $\cos(\theta_{32})\sin(\theta_{31}) = L_{32}$ et tel que $\sin(\theta_{32}) > 0$. On peut ensuite itérer ce raisonnemment ligne par ligne jusqu'à la dernière ligne de la matrice L, montrant ainsi l'unicité du vecteur θ respectant les contraintes du théorème.

□

Le théorème ci-dessus permet de justifier la pertinence de l'algorithme ci-dessous, dit « algorithme de Rebonato-Jäckel » ou « algorithme hypersphère ». De plus, la démonstration ci-dessus permet l'obtention d'un processus pratique pour calculer le vecteur angulaire θ correspondant à une matrice de corrélation donnée.

Algorithme 2.9. *(Algorithme Hypersphère). Implémenter les étapes suivantes :*

1. *Choisir une matrice initiale S_0 en ne retenant que les valeurs propres strictement positives de la matrice G :*

$$S_0 = P\, Diag(\lambda^+)\, P^T$$

où P est la matrice orthogonale qui diagonalise G, λ sont les valeurs propres de la matrice G, et $\forall \lambda \in \mathbb{R}$, $\lambda^+ = \max(\lambda, \epsilon)$, ϵ représentant un nombre réel faiblement positif. On chosira en pratique $\epsilon = 10^{-5}$.

2. *Remplacer $S_0 \leftarrow D^{-\frac{1}{2}} S_0 D^{-\frac{1}{2}}$ où $D = Diag(diag(S_0))$. Cette étape est nécessaire pour que la diagonale de la matrice S_0 soit unitaire.*
3. *A l'aide d'une méthode classique (processus d'orthonormalisation de Schmidt), décomposer la matrice S_0 sous la forme de Cholesky : $S_0 = L_0 L_0^T$, avec L_0 une matrice triangulaire inférieure.*
4. *Implémenter la méthode décrite dans la démonstration du théorème 2.8 afin de déterminer le vecteur angulaire θ_0 correspondant à la matrice L_0, soit : $S_0 = S(\theta_0)$. Retenir θ_0 comme point de départ initial de l'algorithme.*
5. *Implémenter un algorithme du gradient à pas optimal pour minimiser la fonction $g : [0, \pi]^{n \times (n-1)} \to \mathbb{R}$ telle que $\forall \theta \in [0, \pi]^{n \times (n-1)}$ respectant les contraintes du théorème 2.8, $g(\theta) = \|S(\theta) - G\|^2$, en notant $\|.\|$ la norme considérée.*

En pratique, l'implémentation de l'algorithme 2.9 ne présente pas de complexité particulière. Le seul point qui mérite d'être mentionné ici est le calcul du gradient de la fonction g en « norme H ». Voici les étapes de calcul permettant le calcul des dérivées partielles (facilement ordonnables dans un deuxième temps dans un vecteur gradient de dimension $(n^2 - n) \times (1)$) :

$$g(\theta) = \|S(\theta) - G\|_H^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} H_{ij} (S_{ij}(\theta) - G_{ij})^2 \tag{4}$$

Ce qui implique donc, $\forall (\alpha, \beta) \in [\![1; n]\!] \times [\![1; n-1]\!]$ tels que $\alpha > \beta$:

$$\frac{\partial g(\theta)}{\partial \theta_{\alpha\beta}} = \sum_{i=1}^{n} \sum_{j=1}^{n} 2 H_{ij} \frac{\partial S_{ij}}{\partial \theta_{\alpha\beta}} (S_{ij} - G_{ij}) \tag{5}$$

On va donc chercher à expliciter $\frac{\partial S_{ij}}{\partial \theta_{\alpha\beta}}$. Or, l'équation suivante est valable : $S = BB^T$. Donc $\forall (\alpha, \beta) \in [\![1; n]\!] \times [\![1; n-1]\!]$ tels que $\alpha > \beta$:

$$\frac{\partial S}{\partial \theta_{\alpha\beta}} = \frac{\partial B}{\partial \theta_{\alpha\beta}} B^T + B (\frac{\partial B}{\partial \theta_{\alpha\beta}})^T \tag{6}$$

Donc le calcul de $\frac{\partial S_{ij}}{\partial \theta_{\alpha\beta}}$ peut se ramener au calcul de $\frac{\partial B_{ij}}{\partial \theta_{\alpha\beta}}$ où B est la décomposée hypersphère de S. Or, les coefficients de B s'expriment en fonction des coefficients du vecteur θ (voir la définition 2.7). On peut donc obtenir les équations suivantes (avec la division par les cosinus et sinus lorsque ces termes sont non nuls) :

$$\begin{cases} si\ i \neq \alpha\ ou\ si\ j < \beta, \frac{\partial B_{ij}}{\partial \theta_{\alpha\beta}} = 0 \\ si\ i = \alpha\ et\ si\ j = \beta, \frac{\partial B_{ij}}{\partial \theta_{\alpha\beta}} = -\prod_{k=1}^{\beta-1} \sin(\theta_{ik}) = \frac{-\sin(\theta_{\alpha\beta})B_{\alpha\beta}}{\cos(\theta_{\alpha\beta})} \\ sinon\ (i = \alpha\ et\ j > \beta), \frac{\partial B_{ij}}{\partial \theta_{\alpha\beta}} = \cos(\theta_{\alpha j})\cos(\theta_{\alpha\beta}) \prod_{k \in [\![1;j-1]\!] | k \neq \beta} \sin(\theta_{\alpha k}) = \frac{\cos(\theta_{\alpha\beta})B_{\alpha\beta}}{\sin(\theta_{\alpha\beta})} \end{cases} \tag{7}$$

Au final, les équations 5, 6 et 7 permettent donc de calculer le gradient $\nabla g(\theta)$, utile pour l'implémentation de l'algorithme 2.9.

En résumé, l'algorithme hypersphère présente les avantages et inconvénients suivants :

Avantages Il est simple d'utilisation, robuste, aisé à interpréter et peut être appliqué avec plusieurs normes. En particulier, il permet de recourir à la « norme H » qui est peut-être la norme la plus intuitive (différentiation de l'importance attachée aux coefficients de la matrice en fonction de la façon dont ils ont été déterminés et en fonction de leur matérialité sur le SCR). De plus, cet algorithme présente l'avantage d'être utilisé depuis plusieurs années par le secteur assurantiel comme bancaire. En conséquence, certaines compagnies informatiques commercialisent des logiciels informatiques basés sur cet algorithme. Son utilisation dans ce mémoire permet donc d'être représentatif de la démarche suivie par de nombreux industriels ;

Inconvénients La convexité de la fonction $g(\theta)$ n'étant pas garantie, il est possible que l'utilisation d'un algorithme du gradient mène à un minimum local. En pratique, la première étape de l'algorithme 2.9 permet de réduire ce risque en partant d'une matrice initiale qui n'est déjà pas trop éloignée de la matrice de pseudo-corrélation. La solution obtenue sera une très bonne approximation de la meilleure solution théorique.

2.4 Extension de l'algorithme de Rebonato-Jäckel aux cas des contraintes inégalités et difficultés rencontrées

Jusqu'ici, tous les algorithmes présentés réalisent une optimisation avec différentes normes, qui permettent une pondération des coefficients proportionnelle au souhait de l'assureur de maintenir les coefficients en question

inchangés. Néanmoins, ces algorithmes ne permettent pas un encadrement des coefficients. L'algorithme suivant avec contraintes inégalités constitue une application de l'algorithme Uzawa (voir algorithme 1.25) au cas d'une matrice sous forme hypersphère. On note k le nombre de contraintes (si on souhaite encadrer tous les coefficients de la matrice de corrélation on obtiendra typiquement $k = \frac{n(n-1)}{2} \times 2 = n(n-1)$). Etant donné deux matrices symétriques C^{min} et C^{max}, on peut écrire les contraintes sur la solution S de la forme suivante :

$$\forall (i,j) \in [\![1;n]\!]^2, C_{ij}^{min} < S_{ij} < C_{ij}^{max} \tag{8}$$

Avec le formalisme décrit dans la partie 1, il est possible de mettre ces contraintes dans une fonction $c : [0;\pi]^{n\times(n-1)} \to \mathbb{R}^k$ définie par :

$$c(\theta) = \begin{pmatrix} S_{21}(\theta) - C_{21}^{max} \\ S_{31}(\theta) - C_{31}^{max} \\ S_{41}(\theta) - C_{41}^{max} \\ \vdots \\ S_{n,1}(\theta) - C_{n,1}^{max} \\ S_{32}(\theta) - C_{32}^{max} \\ S_{42}(\theta) - C_{32}^{max} \\ \vdots \\ S_{n,n-1}(\theta) - C_{n,n-1}^{max} \\ -S_{21}(\theta) + C_{21}^{min} \\ \vdots \\ -S_{n,n-1}(\theta) + C_{n,n-1}^{min} \end{pmatrix} \tag{9}$$

Algorithme 2.10. *(Algorithme de PSDisation sous contraintes). Implémenter les étapes suivantes :*

1. *Initialiser $\lambda \leftarrow \lambda_0 = (1 \dots 1)^T \in \mathbb{R}^k$. Implémenter l'algorithme hyperpshère (voir algorithme 2.9) pour trouver la matrice S_0 la plus proche de G en norme H, avec $H = [\frac{1}{C_{ij}^{max} - C_{ij}^{min}}]_{(i,j)\in[\![1;n]\!]^2}$. Cette étape de pondération, permet d'accorder une importance à chaque coefficient qui est inversement proportionnelle à la largeur de l'intervalle de contraintes fixées.*
2. *Implémenter les étapes 3 et 4 de l'algorithme 2.9 pour déterminer le vecteur initial θ_0 tel que $S_0 = S(\theta_0)$.*
3. *A lambda fixé, trouver le θ en lequel le minimum de la fonction $\theta \to \|S(\theta) - G\|_H^2 + \lambda^T c(\theta)$ est atteint(à l'aide d'un algorithme du gradient).*
4. *Implémenter $\lambda \leftarrow P(\lambda + c(\theta))$ avec P la projection sur $(\mathbb{R}^+)^k$.*

5. *Itérer les étapes 3 et 4 ci-dessus jusqu'à ce que $\|\lambda' - \lambda\|$ soir inférieur à une certaine tolérance ou jusqu'à un nombre d'itérations maximal.*

L'implémentation de l'algorithme ci-dessus ne présente pas de difficultés particulières, si on remarque que le gradient de la fonction $w : \theta \to \|S(\theta) - G\|_H^2 + \lambda^T c(\theta) = g(\theta) + \lambda^T c(\theta)$ est déjà partiellement calculé (avec g la fonction définie dans l'équation 4). En effet, $\forall(\alpha, \beta) \in [\![1; n]\!] \times [\![1; n-1]\!]$, :

$$\frac{\partial w(\theta)}{\partial \theta_{\alpha\beta}} = \frac{\partial g(\theta)}{\partial \theta_{\alpha\beta}} + \lambda^T \frac{\partial c(\theta)}{\partial \theta_{\alpha\beta}}.$$

Or, les dérivées partielles de la fonction g sont déjà calculées (voir les équations 5 à 7), et le calcul des dérivées partielles de la fonction c est immédiat à partir de l'équation 9 et l'expression des dérivées partielles de S (voir équations 6 et 7).

En résumé, cet algorithme de PSDisation avec contraintes inégalités présente les avantages et inconvénients suivants :

Avantages Il permet de garantir que chacun des coefficients ne varie pas en dehors d'un intervalle fixé par l'assureur ;

Inconvénients Néanmoins l'existence d'une solution n'est pas toujours garantie en fonction des contraintes fixées. Si celles-ci sont trop contraignantes, l'algorithme ne pourra pas converger. En pratique, il est donc souvent nécessaire de procéder par itération, en faisant évoluer les bornes et de procéder à des tests pour vérifier la convergence, ce qui accroit encore l'importance des jugements d'expert dans le modèle. Il est important de noter que les superviseurs souhaitent généralement que les processus purement mathématiques comme celui-ci soient parfaitement encadrés et ne laissent pas de place à l'arbitraire. Enfin, cet algorithme en particulier présente des difficultés dans son implémentation (atteinte de minima locaux etc.) qui sont décrites sur des exemples dans la section 5.

2.5 Ajustements numériques finaux

En pratique, comme certaines valeurs propres des matrices initiales de pseudo-corrélation sont négatives, les algorithmes présentés ci-dessus fournissent une matrice solution positive (notée S) mais dont certaines valeurs propres sont très proches de 0 (typiquement de l'ordre de 10^{-15}). En vue d'utiliser les matrices solutions pour simuler des copules gaussiennes ou de Student, il est nécessaire de mettre en place une dernière étape afin d'augmenter ces valeurs propres (par exemple pour les rendre de l'ordre de 10^{-4}) pour des raisons numériques. Il faut pour cela diagonaliser la matrice S dans

une base orthogonale : $S = PDiag((\lambda_i)_{i\in[\![1;n]\!]})P^T$ et remplacer les valeurs propres trop petites : si $\lambda_i < 10^{-5}, \lambda_i \leftarrow 10^{-5}$. Ceci permet d'obtenir une matrice solution, toujours symétrique définie positive, numériquement tout aussi proche de la matrice initiale G, et pour laquelle une décomposition de Cholesky pourra être appliquée (étape nécessaire pour la simulation des copules). Après cette transformation, la matrice S obtenue ne disposant pas de 1 sur la diagonale, une dernière étape est nécessaire : remplacer $S \leftarrow D^{-\frac{1}{2}}SD^{-\frac{1}{2}}$ où $D = Diag(diag(S))$.

2.6 Notations pour la suite du mémoire

Pour la suite du mémoire, on notera $PSD_H(G)$ ou S^H (respectivement $PSD(G)$) la matrice S issue de la PSDisation de G avec l'algorithme de Rebonato 2.9 pondéré par la matrice H (respectivement sans pondération $H = [1]_{(i,j)\in[\![1;n]\!]^2}$).

De même, on notera $PSD_C(G)$ ou S^C la matrice S issue de la PSDisation de G avec l'algorithme hypersphère 2.10 avec les contraintes suivantes sur la matrice G : $\forall(i,j) \in [\![1;n]\!]^2, C_{ij}^{min} < G_{ij} < C_{ij}^{max}$ où C^{min} et C^{max} sont deux matrices symétriques.

3 Méthodes d'agrégation pour le calcul de l'exigence de capital

L'objectif de cette partie est de comparer deux grandes méthodes classiques d'agrégation (agrégation de type var-covar », et agrégation par copules) et de présenter un peu plus en détail les bases techniques pour le calcul de l'exigence de capital d'un assureur par copules et pour l'étape de simulation associée.

3.1 Agrégation de capitaux pour le calcul du SCR

Comme expliqué dans la section 1, la perte d'un assureur, qui correspond à la variation de fonds propres prudentiels d'une année à l'autre, peut être modélisée comme une variable aléatoire. Dans le cas le plus simple où la perte P est vue comme une fonction uniquement de facteurs de pertes, il est possible de modéliser cette variable aléatoire dans les modèles internes de la manière suivante :

$$P = \sum_{i \in \mathcal{P}} X_i$$

où on note $\mathcal{P}$ l'ensemble des indices des facteurs de pertes modélisés, et $(X_i)_{i \in \mathcal{P}}$ les facteurs de pertes correspondants. En pratique, ces facteurs de pertes peuvent représenter par exemple la perte en cas de sous-estimation des réserves en auto, la perte en cas de tempête dans les Caraïbes, la perte en cas d'inondation en France ou encore d'autres facteurs de pertes. Ces facteurs de pertes doivent être calibrés par l'assureur (en cas de recours à un modèle interne) ou sont calculés sur la base de scénarios de chocs standards (en cas de recours à la formule standard). Les quantiles à 99,5 de tous ces facteurs de pertes ne se réalisant pas en même temps, il existe un effet de diversification permettant d'affirmer qu'en général $q_{99,5}(P) \neq \sum_{i \in \mathcal{P}} q_{99,5}(X_i)$.

Dans la formule standard, la perte de l'assureur dans le scénario i étant notée SCR_i, il existe des coefficients standards de corrélation définis par la réglementation, notés ici $\mid c_{ij} \mid \leq 1$, entre chaque scénario de perte (catastrophe, hausse des taux, défaut d'un réassureur etc.). La réglementation prévoit une agrégation des pertes sous l'approche dite « var-covar » suivante, en notant SCR_{final} l'exigence de capital finale pour la société (*Solvency Capital Requirement*) :

$$SCR_{final} = \sqrt{\sum_{(i,j) \in \mathcal{P}^2} c_{ij} SCR_i SCR_j}.$$

Remarque : Le règlement délégué (UE) 2015/35 dispose que les assureurs utilisant un modèle interne partiel nécessitant le recours à une méthode d'agrégation « var-covar » pour consolider les exigences de capital sur les

deux périmètres (périmètre soumis au modèle interne et périmètre soumis à la formule standard) doivent utiliser une matrice positive semi-définie. Cet exigence est importante car une matrice disposant d'une valeur propre négative (associée à un vecteur propre dont toutes les composantes sont positives) engendrerait la situation irréaliste suivante : il existerait une combinaison de risques (le vecteur propre associé à la valeur propre négative) pour laquelle un accroissement de la prise de risque par l'assureur conduirait à une diminution de l'exigence de marge. La présence de valeurs propres négatives doit donc être évitée, même dans le cas d'une agrégation de type « var-covar ».

Le calcul d'agrégation « var-covar » présente les avantages et les inconvénients suivants :

Avantages Le grand avantage de cette technique d'agrégation est sa simplicité. En particulier, la réalisation de sensibilités du calcul du SCR agrégé aux coefficients de la matrice de corrélation est très facile à implémenter ;

Inconvénients Néanmoins cette technique d'agrégation soulève un ensemble de difficultés. En premier lieu, cette méthode ne permet pas l'obtention de l'ensemble de la ditribution de pertes mais uniquement du quantile à 99,5. Pourtant, dans le cadre de l'utilisation de son modèle interne, un assureur pourrait être intéressé par la connaissance d'autres quantiles de sa distribution de pertes. Cette première difficulté soulève la question du calibrage de la matrice de corrélation, afin de garantir l'atteinte du quantile à 99,5 de la distribution jointe. En particulier, cette technique d'agrégation ne permet pas de modéliser des dépendances non-linéaires (dépendances de corps différentes des dépendances de queues etc.). Enfin, la technique « var-covar » ne permet pas une agrégation directe des facteurs de risques (par exemple, la modélisation directe des rachats au lieu de la modélisation des pertes de fonds propres en cas de rachat). Ceci rend difficile l'interprétation des scénarios menant aux pertes dans le quantile étudié. Il s'avère pourtant qu'en général la région du capital (autour du quantile à 99,5) peut contenir des scénarios très disparates (hausse des taux, baisse des actions, catastrophes naturelles etc.). Enfin, une discussion approfondie des limites de l'agrégation « var-covar » de la formule standard peut être trouvée dans les mémoires d'actuariat [DEC] et [DRI].

En pratique, les assureurs utilisant un modèle interne utilisent quasi-systématiquement des méthodes numériques leur permettant de simuler un grand nombre de réalisations conjointes des facteurs de risques (à l'aide de copules, voir ci-dessous). Ceci permet d'en déduire la distribution complète de la variable aléatoire P représentant les pertes et d'en extraire le quantile à 99,5, correspondant à l'exigence de capital.

3.2 Agrégation des fonctions de pertes pour le calcul du SCR

3.2.1 Distinction entre facteurs de pertes et facteurs de risques

De manière générale, il est difficile de connaitre directement la perte en cas de réalisation d'un risque, et il est souvent plus aisé de décrire le comportement des facteurs de risques (au travers du calibrage de lois marginales) que de décrire directement le comportement des facteurs de pertes (voir pour rappel la définition 1.6).

Par exemple, dans le cas du risque de hausse des taux, la perte est difficile à modéliser car un mouvement de ce facteur de risques a des effets à la fois sur l'actif (baisse du prix des obligations en portefeuille) et sur le passif (la hausse des taux impacte les réinvestissements futurs, qui viendra donc augmenter le taux servi mais peut déclencher des rachats etc.).

3.2.2 Les fonctions d'approximation des pertes

Le calcul du passif sous Solvabilité 2 (*Best Estimate*) peut être vu comme un calcul d'une option financière, avec des options de taux technique correspondant à un *floor* et de participation aux bénéfices discrétionnaires qui doit non seulement respecter le minimum réglementaire (voir article A331-4 du Code des assurances) mais également être cohérente avec la politique réelle de la société (selon le scénario considéré). La valorisation des options se calcule traditionellement à l'aide d'une probabilité risque neutre, sous laquelle le prix des actifs est égal à l'espérance de la valeur actualisée au taux sans risque des flux futurs. En pratique, le calcul d'un *Best Estimate* peut demander un grand nombre de simulations à lui tout seul (plus de 1000 scénarios risque neutre). Dans l'optique de calculer la distribution complète des pertes (comme expliqué dans la section 3), il faudra recalculer un *Best Estimate* dans chacun des scénarios de choc (dits « scénarios monde réel » qui peuvent atteindre plusieurs dizaines de milliers pour obtenir une certaine stabilité de la valeur du quantile à 99,5) afin de pouvoir simuler la variation des fonds propres. La figure 11 illustre ces difficultés numériques.

Pour remédier à cette difficulté numérique, les assureurs utilisent donc des fonctions d'approximation calibrées préalablement sur un jeu de scénarios réduit (principalement à l'aide de techniques des portefeuilles réplicants, de *curve-fitting* ou *least-square monte carlo*), afin de répliquer la variation de fonds propres en fonction des facteurs de risques considérés. Soit en notant P la perte de l'assureur, $\mathcal{P}$ l'ensemble des indices des facteurs de pertes, $\mathcal{R}$

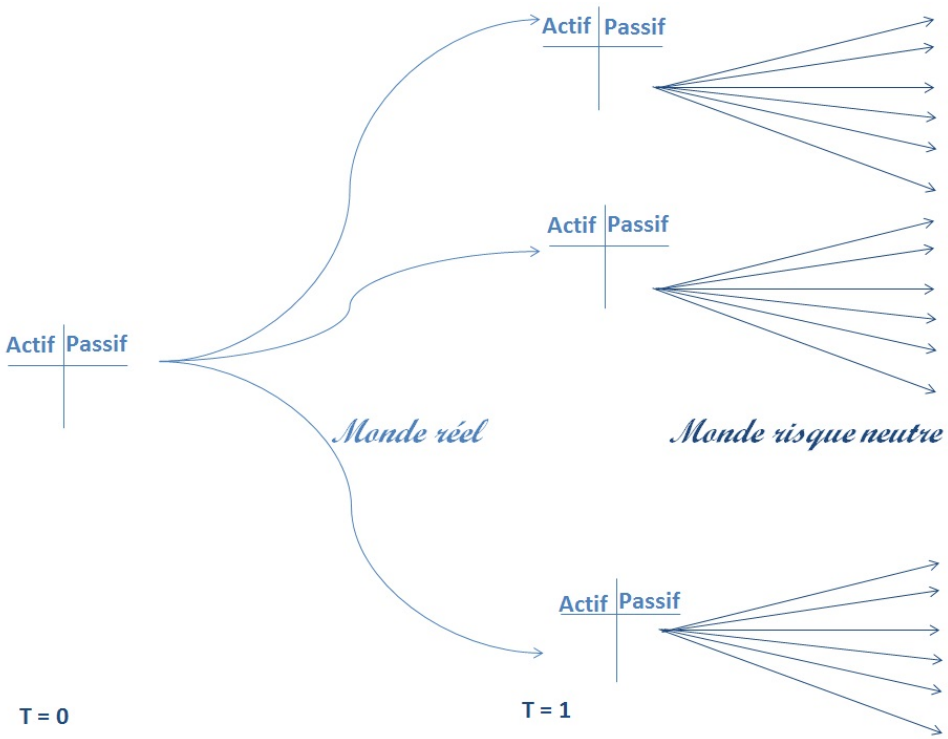


FIGURE 11 – Simulation dans les simulations

l'ensemble des indices des facteurs de risques et f la fonction d'approximation :

$$P = f((Y_j)_{j\in\mathcal{R}}) + \sum_{i\in\mathcal{P}} X_i$$

Les fonctions d'approximations peuvent être polynomiales. La justification théorique de ce choix repose sur le théorème de Weierstrass qui démontre la densité de l'ensemble des fonctions polynomiales dans l'ensemble des fonctions continues sur un compact (voir par exemple le théorème II.1.8 de [PCO]). La continuité des fonctions de pertes est une approximation théorique en l'absence d'autres possibilités pratiques.

3.2.3 L'agrégation de facteurs de risques

Au final, une fois les fonctions d'approximation des passifs calibrées, l'agrégation de facteurs de risques dans un modèle interne est réalisée de la manière suivante :

1. **Modélisation des facteurs de risques** sans tenir compte de la structure de dépendance (calibrage des lois marginales) ;

2. **Modélisation de la dépendance entre ces facteurs de risques** (en utilisant la théorie des copules, voir ci-dessous la section 3.3) ;
3. **Simulation de la réalisation conjointe des facteurs de risques** (grâce aux générateurs de nombres aléatoire, voir la section 3.4) ;
4. Déduction, grâce aux fonctions d'approximation, de la **distribution de pertes agrégées**. Il est ensuite possible d'identifier le quantile à 99,5 (c'est-à-dire l'exigence de capital).

Le calcul d'agrégation par copules présente les avantages et les inconvénients suivants :

Avantages Cette technique d'agrégation permet de modéliser tout type d'agrégation (voir ci-dessous le théorème de Sklar). En particulier, cette approche permet notamment l'étude de dépendances de queue plus fortes que les dépendances de corps (selon le type de copule retenu). La modélisation de dépendances de queue plus fortes peut être expliquée sur un exemple : de manière générale, il est possible de penser que les catastrophes naturelles et les marchés financiers sont peu corrélés dans le corps de la distribution. Par contre, cette corrélation peut éventuellement être plus importante dans la queue de distribution, c'est à dire en cas d'évènements extrêmes. Par ailleurs, cette technique d'agrégation permet de modéliser l'agrégation de facteurs de risques directement (ce que ne permet pas l'approche « var-covar »). Enfin, cette technique d'agrégation permet d'obtenir toute la distribution des pertes. Elle permet une analyse de l'ensemble des scénarios de réalisation des facteurs de risques autour du quantile à 99,5 ;

Inconvénients Tous les gains que les copules permettent sur la finesse de la modélisation doivent être contrabalancés par la complexité qu'elles induisent. En effet, les calculs d'agrégation nécessitent des simulations lourdes numériquement (simulation de chacune des marginales, simulation de la copule etc.). Par ailleurs, le choix de la copule (théoriquement une infinité de possibilités) et le calibrage de ses paramètres demandent des travaux complexes.

3.3 Copules

Depuis son introduction par Abe Sklar en 1959, le concept de copule a joué un rôle important dans la modélisation de la dépendance des variables aléatoire (voir la section 1).

Définition 3.1. *Une copule à n dimensions C est une fonction de* $[0,1]^n \to [0,1]$ *telle que :*

(i) $\forall i \in [\![1;n]\!], \forall (u_j)_{j\in[\![1;n]\!],j\neq i} \in \mathbb{R}^{n-1}, C(u_1,\dots,u_{i-1},0,u_j,\dots,u_n) = 0$ *et* $C(1,\dots,1,u_i,1,\dots,1) = u_i$,

(ii) *C est n-croissante, c'est à dire que pour tout couple de vecteurs de* $\mathbb{R}^n$ *notés respectivement* $(u_1^1 \dots u_n^1)$ *et* $(u_1^2 \dots u_n^2)$ *tel* $\forall i \in [\![1;n]\!], u_i^1 \leq u_i^2$, *on a l'inégalité :* $\sum_{k_1\in\{1,2\}} \dots \sum_{k_n\in\{1,2\}} (-1)^{k_1+\dots k_n} C(u_1^{k_1},\dots,u_n^{k_n}) \geq 0$.

Cette deuxième propriété est plus facilement interprétable en dimension 2 : la 2-croissance d'une fonction g peut se réécrire de la manière suivante : $\forall x \in \mathbb{R}^2, \forall \epsilon > 0, \forall \delta > 0, g(x_1+\epsilon, x_2+\delta) - g(x_1+\epsilon, x_2) \geq g(x_1, x_2+\delta) - g(x_1, x_2)$. Cette inégalité a une interprétation géométrique. En effet, si on considère les quatre points $(x_1+\epsilon, x_2+\delta)$, $(x_1+\epsilon, x_2)$, $(x_1, x_2+\delta)$ et (x_1, x_2) comme les quatre sommets d'un rectangle, alors un fonction 2-croissante pondère plus fortement les points situés sur la diagonale montante ($(x_1+\epsilon, x_2+\delta)$ et (x_1, x_2)) qui expriment une dépendance positive que les points situés sur la diagonale descendante ($(x_1+\epsilon, x_2)$ et $(x_1, x_2+\delta)$), qui expriment une dépendance négative. Le théorème suivant, « théorème de Sklar », justifie l'importance des copules dans la modélisation de la dépendance de variables aléatoires :

Théorème 3.2. *Soit un vecteur aléatoire* $X \in \mathbb{R}^n$ *de lois marginales continues notées* F_i, $\forall i \in [\![1;n]\!]$. *Alors il existe une unique copule C telle que* $\forall x \in \mathbb{R}^n, F_X(x_1, x_2 \dots, x_n) = C(F_1(x_1), F_2(x_2) \dots, F_n(x_n))$. *Réciproquement, si C est une copule, alors la fonction* F_X *de* $\mathbb{R}^n \to \mathbb{R}$ *définie par la relation précédente est une fonction de répartition d'un vecteur aléatoire X dont les lois marginales sont les fonctions* F_i.

Démonstration. Notons $\forall i \in [\![1;n]\!]$, U_i la variable aléatoire définie par $U_i = F_i(X_i)$. Alors, $\forall u_i \in [0,1], P(U_i < u_i) = P(F_i(X_i) < u_i) = P(X_i < F_i^{-1}(u_i)) = F_i(F_i^{-1}(u_i)) = u_i$ donc U_i suit une loi uniforme sur [0,1] (la bijectivité de la fonction F_i a été utilisée pour la démonstration, même si cette hypothèse n'est pas nécessaire, car cela permet de passer outre certaines démonstrations subtiles basées sur le passage à la borne supérieure). On peut définir la fonction $C : [0,1]^n \to [0,1]$ par : $\forall (u_i)_{i\in[\![1;n]\!]} \in \mathbb{R}^n, C(u_1,\dots,u_n) = P(U_1 < u_1, \dots, U_n < u_n)$. Cette fonction vérifie bien les propriétés d'une copule (admis). Par ailleurs, on a bien : $F_X(x_1,\dots,x_n) = P(X_1 < x_1, \dots, X_n < x_n) = P(F_1(X_1) < F_1(x_1), \dots, F_n(X_n) < F_n(x_n)) = P(U_1 < F_1(x_1), \dots, U_n < F_n(x_n)) = C(F_1(x_1), \dots, F_n(x_n))$. □

Il existe de nombreuses familles de copules classiques, qui se situent toutes entre deux bornes (correspondant intuitivement au cas d'une corrélation / d'une anti-corrélation parfaites), dites « bornes de Fréchet » (voir notamment

des travaux du mémoire d'actuariat [DRL]). En pratique, la copule suivante est très utilisée dans les modèles internes des assureurs :

Exemple 3.3. *On appelle copule gaussienne la copule donnée par la formule suivante :* $\forall (u_i)_{i\in[\![1;n]\!]} \in \mathbb{R}^n, C_G(u_1,\dots,u_n) = \Phi_G(\mathcal{N}^{-1}(u_1),\dots,\mathcal{N}^{-1}(u_n))$*, avec* G *une matrice de corrélation,* Φ_G *la fonction de répartition normale multivariée de matrice de corrélation* G*, et* $\mathcal{N}$ *la fonction de répartition de la loi normale centrée réduite.*

Sans rentrer dans des considérations théoriques plus avancées, les figures 12 et 13 représentent de deux manières différentes la densité d'une copule gaussienne (correspondant intuitivement à la densité de la loi jointe (X_1, X_2) de deux marginales uniformes sur $[0, 1]$).

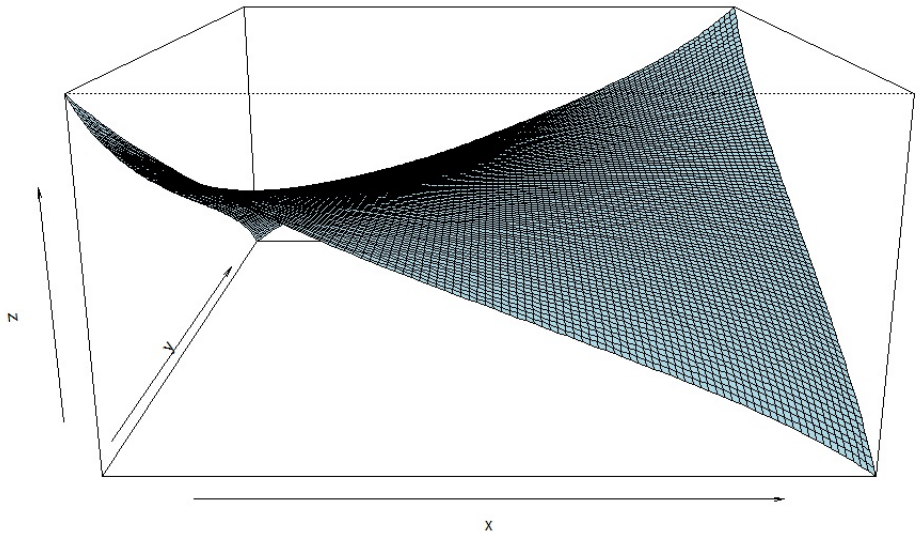


FIGURE 12 – Densité d'une copule gaussienne ($\rho = 0{,}2$) 1/2

Le théorème suivant est nécessaire pour comprendre les méthodes de simulation des copules gaussiennes (et de Student) ainsi que les enjeux autour de la PSDisation des matrices de pseudo-corrélation :

Théorème 3.4. *Si* G *est une matrice définie de* $\mathbb{R}^n$*, alors il existe une unique matrice triangulaire supérieure à coefficient diagonaux strictement positifs tels que* $G = T^T T$*.*

L'intérêt des copules est notamment de permettre des simulations distinctes des marginales et de la dépendance pour simuler un vecteur aléatoire. Au delà de sa définition, il est surtout important de comprendre comment la copule gaussienne peut être simulée.

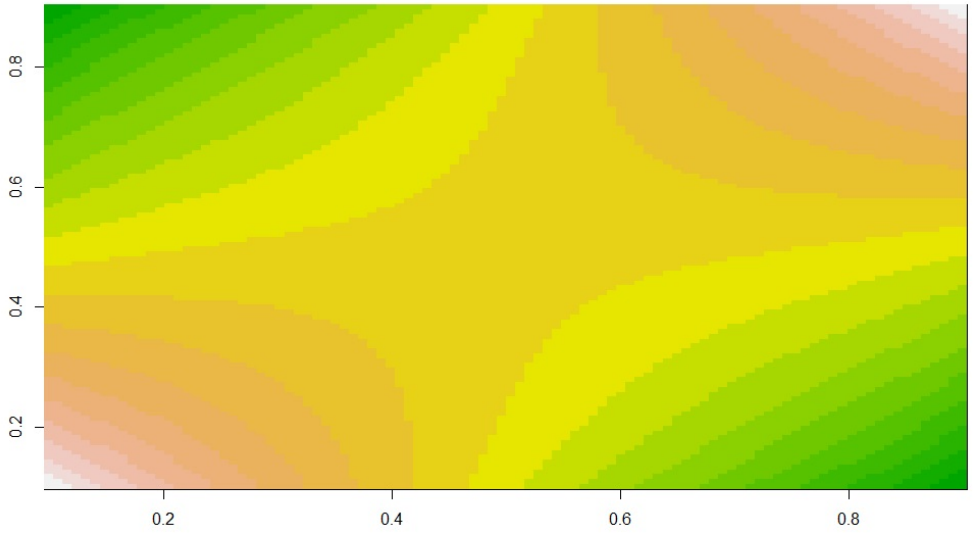


FIGURE 13 – Densité d'une copule gaussienne ($\rho = 0{,}2$) 2/2

Algorithme 3.5. *(Simulation d'une copule gaussienne). Implémenter les étapes suivantes :*

1. *(Simulation de chacune des marginales) En fonction des choix effectués, il est possible de simuler de manière indépendante chacune des n variables que l'on souhaite corréler, quelles que soient leurs natures (loi gaussienne, loi lognormale, loi gamma etc.). On peut recourir pour cela à des méthodes de simulation classiques, comme par exemple la méthode d'inversion, qui repose sur le fait que pour toute variable aléatoire X de fonction de répartition continue, la variable $U = F_X(X)$ suit une loi uniforme sur [0,1] ;*
2. *(Simulation d'un vecteur gaussien corrélé) La simulation d'un vecteur gaussien de matrice de corrélation S est classiquement réalisée par la simulation du vecteur $Y = TX$ avec X un vecteur aléatoire gaussien centré réduit indépendant, et T la matrice apparaissant dans la décomposition de Cholesky de la matrice S ($S = T^T T$). On vérifie facilement que $V(Y) = V(TX) = T^T V(X)T = T^T T = S$;*
3. *(Ordonnancement pour reproduire la structure de dépendance) Cette étape consiste en l'ordonnancement des valeurs des variables aléatoires initiales de la même manière que sont ordonnées les variables aléatoire simulées par le vecteur gaussien. Par exemple, en notant X_1^{5634} la 5634ème valeur de la variable aléatoire classée par ordre croissant, si*

le premier vecteur gaussien simulé représente

$$\begin{pmatrix} X_1^{5952} \\ X_2^{332} \\ \vdots \\ X_i^{1484} \\ \vdots \\ X_n^{9679} \end{pmatrix}$$

alors il faut réordonner les variables aléatoires simulées à la première étape de manière à respecter le même ordre.

Une caractéristique importante des copules, notamment lorsqu'elles sont utilisées pour agréger des capitaux, est la dépendance de queue. Cette notion représente la corrélation des évènements extrêmes. En dimension 2, la dépendance de queue (*upper tail dependency*) est définie par :

$$\lambda_U = \lim_{u \to 1^-} P(X_2 > u | X_1 > u).$$

Dans le cas de la copule gaussienne, en dehors des cas dégénérés (corrélation parfaite entre X_1 et X_2), cette limite est nulle. Ceci signifie que cette famille de copules ne permet pas de représenter un cas de corrélation entre des évènements extrêmes, pourtant utile dans le cas de l'analyse des scénarios extrêmes auxquels un assureur est exposé. Néanmoins, cette famille de copules ne peut pas être rejetée pour autant car la corrélation des évènements du quantile à 99,5 n'est pas nulle. La problématique de la dépendance de queue des copules gaussiennes se pose donc lors du calibrage des corrélations. En pratique, les assureurs recourent également à une autre famille de copules très classique : la copule de Student qui, en dehors des cas dégénérés (anticorrélation parfaite entre X_1 et X_2), permet de modéliser une situation de corrélation de queue (voir par exemple le mémoire d'actuariat [AUB] qui décrit un cas concret menant au choix de cette copule pour la modélisation de la dépendance entre des charges de sinistres entre différentes *lines of business*).

Exemple 3.6. *On appelle copule de Student de paramètre ρ la copule donnée par la formule suivante : $\forall (u_i)_{i \in [\![1;n]\!]} \in \mathbb{R}^n, C_{G,\nu}(u_1, \ldots, u_n) = t_{G,\nu}(t_\nu^{-1}(u_1), \ldots, t_\nu^{-1}(u_n))$, avec G une matrice de corrélation, $t_{G,\nu}$ la fonction de répartition multivariée de Student de matrice de corrélation G, et t_ν la fonction de répartition de la loi de Student à ν degrés de liberté.*

Comme précédemment, les figures 14 et 15 représentent de deux manières

différentes la densité d'une copule de Student.

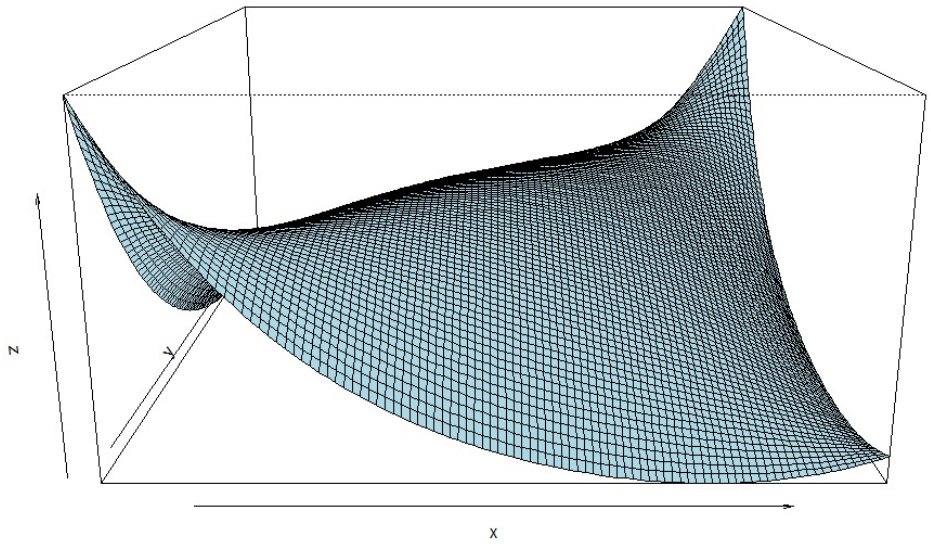


FIGURE 14 – Densité d'une copule de Student ($\rho = 0{,}2$ et $\nu = 3$) 1/2

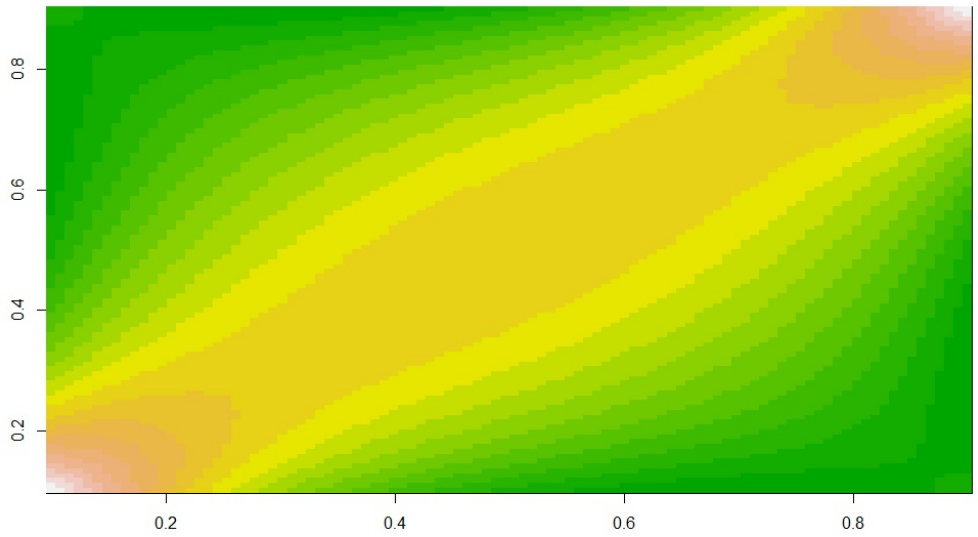


FIGURE 15 – Densité d'une copule de Student ($\rho = 0{,}2$ et $\nu = 3$) 2/2

De même que pour la copule gaussienne, cette copule peut être simulée par un algorithme.

Algorithme 3.7. *(Simulation d'une copule de Student). Implémenter les étapes suivantes :*

1. *(Simulation de chacune des marginales) Implémenter l'étape 1 de l'algorithme 3.5 ;*
2. *(Simulation d'un vecteur de Student corrélé) La simulation d'un vecteur de Student de matrice de corrélation S est classiquement réalisée par la simulation du vecteur $Y = \sqrt{\frac{\nu}{K}} TX$ avec X un vecteur aléatoire gaussien centré réduit indépendant, T la matrice apparaissant dans la décomposition de Cholesky de la matrice S ($S = T^T T$), et K une variable aléatoire indépendante suivant une loi du chi-deux à ν paramètres. Dans ce mémoire, la valeur $\nu = 3$ sera retenue (choix arbitraire pour les exemples numériques) ;*
3. *(Ordonnancement pour reproduire la structure de dépendance) Implémenter l'étape 3 de l'algorithme 3.5.*

Ces algorithmes de simulation 3.5 et 3.7 présentent les avantages et les inconvénients suivants :

Avantages Ces algorithmes permettent de séparer la simulation des marginales (étape 1) de la simulation de la corrélation à proprement dite (étape 2) ;

Inconvénients Néanmoins ces algorithmes, comme tous les algorithmes de simulation, ne permettent pas d'obtenir un résultat parfaitement stable (le résultat est une variable aléatoire). En conséquence, en pratique, afin d'avoir le maximum de stabilité pour la suite des travaux (i.e. afin de réduire la variance de la variable aléatoire simulée), les simulations des marginales seront fixées initialement et un grand nombre de simulations sera effectué afin de simuler la corrélation ($32768 = 2^{15}$ simulations dans les travaux présentés dans ce mémoire, voir la section ci-dessous). De plus, le calcul du quantile à 99,5 sera effectué avec plusieurs simulations successives (4) et la valeur moyenne de ces simulations sera retenue (soit un nombre total de tirages de $2^{17} = 131072$). Ce grand nombre de tirages pour accroitre la stabilité du calcul du quantile à 99,5 est le prix à payer pour pouvoir ordonner les résultats et effectuer une couche d'optimisation supplémentaire (voir sous-section 4.1).

3.4 Générateurs de nombres aléatoires et stabilité des simulations

Comme expliqué dans la section ci-dessus, le calcul du SCR est issu de deux étapes de simulation, ce qui induit donc un bruit autour de la valeur calculée :

— Les marginales sont simulées indépendamment une à une dans une matrice $X \in \mathbb{R}^{n \times ech}$, où chaque ligne $X[i,]$ représente une succession de réalisations indépendantes de la i-ème marginale.
— L'agrégation est simulée (à l'aide des algorithmes 3.5 ou 3.7).

Pour la suite du mémoire, les raisonnements seront systématiquement effectués avec le tirage des marginales fixé (la matrice X est donnée). Néanmoins, la simulation de l'agrégation génère à elle seule de l'incertitude. Schématiquement, le SCR calculé étant une variable aléatoire de type loi normale d'après le théorème de la limite centrale, il est possible d'estimer l'intervalle de confiance autour du SCR calculé comme : $P(|Z - m| < 1,65\,\widehat{\sigma}) = 90\%$ avec $\widehat{\sigma}$ l'écart-type estimé à l'aide de tirages successifs. Pour chacune des matrices, et des marginales utilisées, on dira qu'une déformation est *faible* si l'impact observé sur le SCR est inférieur à $2 \times 1,65 \times \widehat{\sigma}$, et *forte* dans le cas contraire. Le facteur 2 provient du fait qu'il est nécessaire de tenir compte de l'intervalle de confiance autour du SCR initial, puis autour du SCR « après déformation ».

Dans le cas des algorithmes génétiques, le critère retenu doit être plus sévère, car il est fait de nombreuses fois appel aux tirages de nombres aléatoires (au maximum 256 fois). On dira qu'une déformation est *très forte* si l'impact observé sur le SCR est supérieur à $2 \times 2,89 \times \widehat{\sigma}$ car $(P(|Z - m| < 2,89\,\widehat{\sigma}) = 99,8\% = \frac{1}{512})$.

L'objet de ce mémoire n'est pas de rentrer dans une discussion détaillée autour des générateurs de nombres aléatoires retenus, qui représentent un monde à part entière. Néanmoins, lors du recours à des simulations de Monte-Carlo, il est une bonne pratique de considérer plusieurs générateurs différents. Voici donc un point sur les grands principes de deux générateurs de nombres aléatoires très classiques (Mersenne et Sobol, voir par exemple le chapitre 8 de [JAC] pour plus de détails). Les analyses numériques associées sont présentées dans la section 5 (à l'aide de fonctions prédéfinies dans le logiciel R) et un seul des deux générateurs est retenu (celui de Sobol) sur la base de la volatilité observée dans le calcul du SCR.

Le générateur de nombres aléatoires le plus répandu est celui de Mersenne (et utilisé de base par le logiciel de calcul R) qui est défini par les étapes suivantes :

1. Choisir un nombre premier de la forme $p = 2^b - 1$ (dans le logiciel R, b vaut 19937) et deux entiers $(a, c) \in [\![1; p-1]\!]^2$. Sélectionner un nombre $x \leftarrow x_0 \in [\![1; p-1]\!]$;
2. Implémenter $x \leftarrow ax + c$ dans $\mathbb{Z}/p\mathbb{Z}$;
3. Répéter l'étape 2 le nombre de fois voulu, et choisir à chaque étape le

nombre $\frac{x}{p}$ comme nombre aléatoire entre 0 et 1.

La figure 16 illustre sur un exemple un tirage de nombres aléatoires entre 0 et 1, dans le plan, avec un générateur de nombres aléatoires de Mersenne.

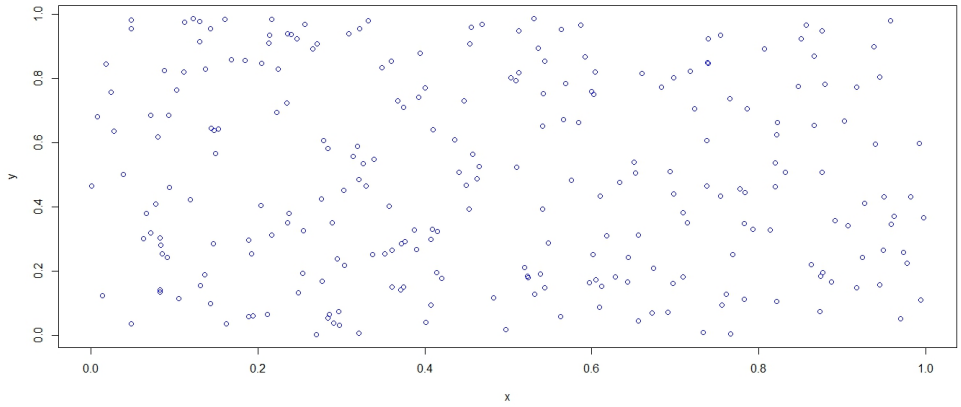


FIGURE 16 – Séquence de nombres générés dans le plan avec un générateur de nombres aléatoires de Mersenne

Le générateur de nombres aléatoires de Sobol dispose de propriétés intéressantes qui peuvent justifier son utilisation lors de simulation de type Monte-Carlo. En effet, il permet de remplir uniformément l'espace (on dit qu'il est « à discrépance faible »), et cette propriété est optimale pour un nombre de tirage de la forme 2^n (voir les résultats empiriques en 8.6 de [JAC]), ce qui justifie les nombres de tirages de $2^{15} = 32768$ et $2^{17} = 131072$ retenus dans ce mémoire. Voici les grandes étapes à suivre pour générer des nombres de Sobol dans d dimensions :

1. De manière préliminaire, il est nécessaire de disposer de d polynômes primitifs modulo 2 notés P_k (i.e. le plus petit polynôme de la forme $X^q - 1$ divisible par P_k dans $\mathbb{Z}/2\mathbb{Z}[X]$ est $X^{deg(P)} - 1$) de la forme $\forall k \in [\![1; d]\!], P_k = \sum_{j=0}^{g_k} a_{kj} X^{g_k - j}$;
2. $\forall k \in [\![1; d]\!]$, choisir arbitrairement une suite de nombres $v_{k1}, v_{k2}, \ldots v_{kg_k}$;
3. $\forall k \in [\![1; d]\!]$, $\forall l > g_k$ générer la suite de ces nombres par la formule suivante :

$$v_{kl} = \frac{v_{k(l-g_k)}}{2^{g_k}} \bigoplus_{j=1}^{g_k} a_{kj} v_{k(l-j)}$$

où le symbole $\bigoplus$ représente l'addition modulo 2 sans retenue [2] ;

4. Une fois les nombres v_{kl} construits, il faut disposer d'une suite injective $n \to \gamma(n)$. La construction du n-ième nombre de Sobol dans la direction $k \in [\![1; d]\!]$ est donnée par :

$$x_{nk} = \bigoplus_{j=1}^{b} v_{kj} \delta_j^{\gamma(n)}$$

avec $\delta_j^{\gamma(n)}$ qui vaut 1 si le j-ième bit de $\gamma(n)$ (en comptant à partir de la droite) vaut 1 et 0 sinon.

5. La division des x_{nk} par 2^b permet de ramener ces nombres dans [0,1].

Au delà de cette construction qui peut s'avérer complexe, il est plus aisé de comprendre le fonctionnement de ce générateur de nombres aléatoires sur un exemple. Ce générateur est dit quasi-aléatoire (en fonction de l'initialisation) car les nombres générés sont répartis uniformément. Par exemple, en utilisant le générateur de nombres aléatoires de Sobol du package *randtoolbox* de R, la commande *sobol(8, dim = 1, init = TRUE)* renverra systématiquement les nombres suivants :

$$\{0,5\,;\ 0,75\,;\ 0,25\,;\ 0,375\,;\ 0,875\,;\ 0,625\,;\ 0,125\,;\ 0,1875\}.$$

En dimension 1, l'exécution successive des commandes *sobol(4, dim = 1, init = TRUE)* puis *sobol(4, dim = 1, init = FALSE)* renverra systématiquement les nombres suivants :

$$\{0,5\,;\ 0,75\,;\ 0,25\,;\ 0,375\}.$$

puis

$$\{0,875\,;\ 0,625\,;\ 0,125\,;\ 0,1875\}.$$

On peut ensuite considérer que les tirages suivants de 4 nombres successifs, à savoir

$$\{0,6875\,;\ 0,9375\,;\ 0,4375\,;\ 0,3125\}$$

$$\{0,81250\,;\ 0,56250\,;\ 0,06250\,;\ 0,09375\}$$

puis

$$\{0,59375\,;\ 0,84375\,;\ 0,34375\,;\ 0,46875\}\ \textit{etc.}$$

représentent (fictivement) des tirages successifs de 4 nombres aléatoires.

---

2. Par exemple, l'addition sans retenue modulo 2 donne $\begin{array}{r} 10011011 \\ \oplus 11000111 \\ \hline 01011100 \end{array}$.

Sur cet exemple, on comprend pourquoi il est nécessaire lors de l'utilisation d'un générateur de nombres aléatoires de Sobol de tirer un nombre de simulations qui soit une puissance de 2. Cela permet à ces suites de Sobol (sans ré-initialisation) d'être à chaque fois « équiréparties » (dans un certain sens que nous ne préciserons pas ici, mais qui est intuitif sur l'exemple précédent). De plus, ces nombres de Sobol ne retombent jamais sur les mêmes valeurs (contrairement au générateur de nombres aléatoires de Mersenne qui dispose d'une certaine période, aussi grande soit-elle).

La figure 17 illustre la propriété de discrépance faible dans le plan. Cette figure présente un caractère beaucoup plus « régulier » que la figure précédente.

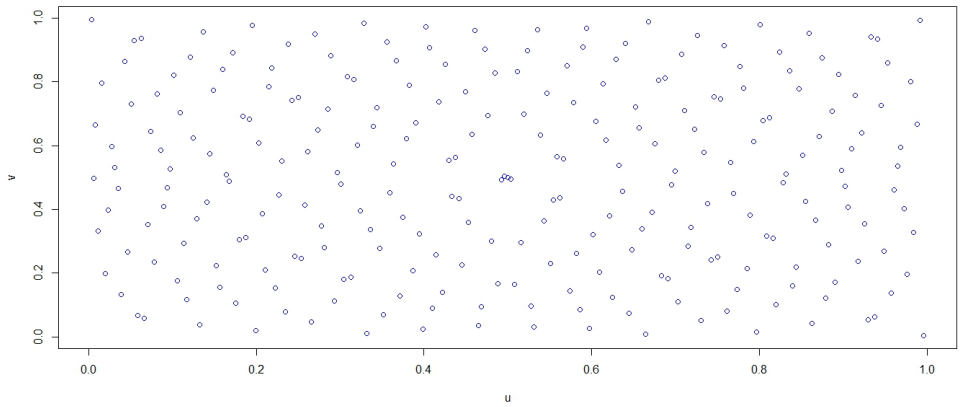


FIGURE 17 – Séquence de nombres générés dans le plan avec un générateur de nombres aléatoires de Sobol

4 Utilisation d'algorithmes génétiques pour étudier la sensibilité autour de l'étape de PSDisation

L'objectif de cette partie est de décrire de manière détaillée les travaux réalisés dans ce mémoire.

4.1 Optimisation du SCR à l'aide d'algorithmes génétiques

Le traitement de la problématique posée dans ce mémoire requiert la mise en place d'une couche supplémentaire d'algorithmes d'optimisation (au-delà des algorithmes d'optimisation déjà nécessaires pour rendre une matrice de pseudo-corrélation PSD, présentés dans la section 2). Ces algorithmes d'optimisation génétique constituent surtout des travaux d'analyse de sensibilité étendus, car il n'y a pas de garantie d'aboutir à la meilleure des solutions.

Comme expliqué ci-dessus, afin d'obtenir un niveau de stabilité suffisant (voir l'analyse détaillée dans la section 3.4), la méthode suivante de calcul du SCR est définie, à simulation fixée des marginales, et à matrice de corrélation fixée. On note $\mathcal{A}$ l'algorithme d'agrégation retenu (selon la copule retenue : voir algorithmes 3.5 et 3.7), S la matrice de corrélation positive semi-définie associée à la copule utilisée et $q_{99,5}^{S,f}(\mathcal{A})$ le quantile à 99,5 de la distribution de la fonction $f((X_i)_{i\in[\![1;n]\!]}) = \sum_{i=1}^{n} X_i$ obtenue lors d'une simulation de l'agrégation :

$$SCR_{f,\mathcal{A}}(S) = \begin{cases} \frac{1}{m}\sum_{i=1}^{m} \xi_i \\ tel\ que\ \forall i \in [\![1;m]\!], \xi_i \leftarrow q_{99,5}^{S,f}(\mathcal{A}) \end{cases} \tag{10}$$

En pratique, dans ce mémoire, afin d'obtenir une bonne stabilité des résultats sans demander des capacités de calculs trop lourdes, on retiendra m = 4. Cette valeur a été définie par tâtonnement, afin de définir un équilibre entre temps de calcul et stabilité des résultats.

4.1.1 Optimisation génétique du SCR avec une permutation des facteurs de pertes

L'objectif de cette sous-section est de mesurer quel impact aurait le choix arbitraire d'ordonnancement des facteurs de pertes par un actuaire de la société d'assurance considérée sur l'exigence de capital de la société. Pour cela, on a recours au formalisme suivant (on dit qu'on fait opérer le groupe symétrique $\mathfrak{S}_n$ sur l'espace vectoriel des matrices $\mathbb{R}^{n\times n}$, en notant $GL(\mathbb{R}^{n\times n})$ l'ensemble des automorphismes de $\mathbb{R}^{n\times n}$) :

$$\rho : \begin{cases} \mathfrak{S}_n \to GL(\mathbb{R}^{n\times n}) \\ \sigma \to \rho(\sigma) = \sigma. \end{cases}$$

avec la définition suivante de l'opération d'une permutation σ sur une matrice $M \in \mathbb{R}^{n\times n}$:

$$\rho(\sigma)(M) = \sigma.M = P_{\sigma^{-1}} M P_\sigma$$

où la matrice P_σ vaut $P_\sigma = [\delta_{i\sigma(j)}]_{(i,j)\in[\![1;n]\!]^2}$ et δ_{ql} représente le symbole de Kronecker. A titre d'illustration, lorsque la permutation considérée est une transposition $\sigma = \tau_{kl}$, cette opération consiste à permuter les colonnes k et l entre elles, ainsi que les lignes k et l. Dans le cas d'une matrice de pseudo-corrélation, ceci constitue un réordonnancement des facteurs de pertes dans la matrice (avant PSDisation). Les algorithmes de PSDisation vont donc converger vers des matrices de corrélation différentes que dans le cas initial, ce qui mènera à une exigence de capital différente.

Remarque : Cette formalisation mathématique permet également de montrer algébriquement pourquoi une permutation des facteurs de pertes ne modifie pas la positivité initiale d'une matrice de corrélation. En effet, la matrice permutée est conjuguée à la matrice initiale (de la forme $P^{-1}AP$), donc elle possède les mêmes valeurs propres.

Commentaire : La conséquence de la remaque ci-dessus est la suivante : la « norme H » étant invariante sous l'action de $\mathfrak{S}_n$, la matrice PSD la plus proche après permutation devrait être en théorie exactement la même qu'avant permutation (à une permutation près). Néanmoins, en pratique, la présence de minima locaux dans l'algorithme de Rebonato rompt cette commutativité théorique (entre permutation et PSDisation).

Enfin, on fait opérer le groupe $\mathfrak{S}_n$ sur l'espace vectoriel des matrices $\mathbb{R}^{n\times ech}$ de la manière suivante :

$$\rho' : \begin{cases} \mathfrak{S}_n \to \mathcal{L}(\mathbb{R}^{n\times ech}) \\ \sigma \to \rho'(\sigma) = \sigma. \end{cases}$$

avec :

$$\rho'(\sigma)(X) = \sigma.X = P_{\sigma^{-1}} X$$

ce qui correspond à permuter les tirages des marginales X (*ech* représente le nombre de tirage de la marginale considérée), de manière à garantir la cohérence avec la permutation de la matrice de pseudo-corrélation.

On cherche donc les valeurs maximale et minimale de la fonction $\bullet \to SCR^{\bullet_3}_{f,\mathcal{A}} \circ PSD_{\bullet_2}(\bullet_1)$ sur l'orbite de la matrice de pseudo-corrélation G et de sa matrice de pondération sous l'action de $\mathfrak{S}_n$, notée $\mathcal{O}_{G,H,X} = \{(\sigma.G, \sigma.H, \sigma.X) | \sigma \in \mathfrak{S}_n\}$. D'un point de vue analytique, afin de mesurer la robustesse du processus de PSDisation d'une matrice de pseudo-corrélation G, on cherche le maximum et le minimum de la fonction g définie par :

$$\forall \sigma \in \mathfrak{S}_n,\ g(\sigma) = SCR^{\sigma.X}_{f,\mathcal{A}}(PSD_{\sigma.H}(\sigma.G)) \tag{11}$$

où, pour rappel :
- X représente un tirage fixé des marginales ($X \in \mathbb{R}^{n \times ech}$) ;
- $\sigma.$ représente l'impact d'une permutation des facteurs de risques et / ou de pertes ;
- G représente la matrice de pseudo-corrélation initiale ;
- $PSD_H(G)$ représente la matrice positive semi-définie la plus proche de G en « norme H » ;
- $SCR^X_{f,\mathcal{A}}$ représente le calcul du SCR par simulation, pour une fonction d'approximation des pertes f donnée, un tirage de marginales X donné, et une copule $\mathcal{A}$ choisie (gaussienne ou Student).

Lorsque n est petit, il est possible de calculer la fonction g sur chacune des permutations de $\mathfrak{S}_n$. Mais très vite, ($10\,! = 3\ 628\ 800$), il est nécessaire d'avoir recours à des alorithmes d'optimisation. Dans notre cas, les algorithmes génétiques sont tous désignés. On va donc mettre en place l'algorithme suivant (sur le modèle des algorithmes génétiques, voir algorithme 1.26) :

Algorithme 4.1. *(Algorithme génétique des permutations). Implémenter les étapes suivantes :*

1. *(Création population initiale) : Tirer une population aléatoire initiale de N permutations, contenant l'identité : $\{\sigma_k \in \mathfrak{S}_n | k \in [\![1; N]\!]\}$. Dans notre cas, la valeur $N = 6$ est retenue.*
2. *(Classement de la population) : Utiliser la fonction g (voir équation 11) pour classer la population.*
3. *(Reproduction de deux individus) : Pour tout couple $(\sigma_1, \sigma_2) \in \mathfrak{S}_n^2$, créer deux individus issus de la composition : $\sigma' = \sigma_1 \circ \sigma_2$ et $\sigma'' = \sigma_2 \circ \sigma_1$. Si les deux permutations commutent, composer l'une d'elle par une transposition quelconque $\sigma' \leftarrow \tau \circ \sigma'$.*
4. *(Mutation) : La mutation d'une permutation correpsond à la composition par une transposition tirée aléatoirement $\tau = (k, k+1)$.*
5. *(Evolution de la population) : La population évolue sans disparition (tous les individus sont conservés jusqu'à la dernière étape). Ceci permet d'obtenir à la fin de l'algorithme à la fois le maximum et le minimum.*

6. *(Sélection de l'individu le plus apte à survivre) : L'algorithme prend fin lorsque le minimum est atteint 4 fois consécutivement par la même permutation, ou lorsqu'un nombre maximal d'itérations est atteint.*

Il est à noter que les chiffres ci-dessus (taille de la population initiale, critère d'arrêt etc.) ont été déterminés empiriquement, au regard du temps de calcul et des résultats obtenus.

4.1.2 Optimisation génétique du SCR en fonction des coefficients de pondération

Le choix des coefficients de pondération des coefficients lors d'une PSDisation en norme H est relativement arbitraire, puisqu'ils sont choisis initialement par l'assureur, sans véritable justification mathématique autre qu'un niveau de confiance dans une corrélation donnée. Il est légitime de se poser la question de la robustesse du calcul du SCR en fonction du choix de la matrice H. Il faut alors voir le SCR comme une fonction de la matrice de pondération H (avec les mêmes notations que dans l'équation 11) :

$$\forall H \in [-1;1]^{n\times n},\ g(H) = SCR_{f,\mathcal{A}}(PSD_H(G)) \tag{12}$$

La question de la robustesse de l'agrégation en fonction de la pondération peut se poser en encadrant la matrice de pondération H entre deux matrices notés H^{min} et H^{max} :

$$\forall (i,j) \in [\![1;n]\!]^2, H_{ij}^{min} \leq H_{ij} \leq H_{ij}^{max}.$$

On peut répondre à cette question en implémentant l'algorithme suivant (sur le modèle des algorithmes génétiques, voir algorithme 1.26) :

Algorithme 4.2. *(Algorithme génétique des pondérations). Implémenter les étapes suivantes :*

1. *(Création population initiale) : Tirer une population aléatoire initiale de N matrices de pondération dans les bornes H^{min} et H^{max} fixées initialement : $\{H_k \in [-1;1]^{n\times n} | k \in [\![1;N]\!]\}$. Dans notre cas, la valeur $N = 8$ est retenue.*
2. *(Classement de la population) : Utiliser la fonction g (voir équation 12) pour classer la population.*
3. *(Reproduction de deux individus) : Pour tout couple $(H_1, H_2) \in ([-1;1]^{n\times n})^2$, créer deux individus H' et H" de la manière suivante (en notant H[,j] la j-ème colonne de la matrice H, et E(x) la partie entière de x) :*

$$\begin{cases} \forall j \leq E(\frac{n}{2}),\ H'[,j] = H_1[,j],\ H'[,j] = H_2[,j] \\ \forall j > E(\frac{n}{2}),\ H'[,j] = H_2[,j],\ H'[,j] = H_1[,j]. \end{cases}$$

4. *(Mutation) : La mutation d'une matrice de pondération correspond à la modification aléatoire d'un coefficient de la matrice H considérée. La mutation consiste à tirer aléatoirement ce coefficient entre 0 et 1.*
5. *(Evolution de la population) : La population évolue sans disparition (tous les individus sont conservés jusqu'à la dernière étape). Ceci permet d'obtenir à la fin de l'algorithme à la fois le maximum et le minimum.*
6. *(Sélection de l'individu le plus apte à survivre) : L'algorithme prend fin lorsque le minimum est atteint 5 fois consécutivement par la même permutation, ou lorsqu'un nombre maximal d'itérations est atteint.*

Comme pour l'algorithme de la section précédente, les chiffres (taille de la population etc.) ont été déterminés empiriquement de manière à trouver un équilibre entre la qualité des résultats et le temps de calcul.

4.2 Autres sensibilités autour de la PSDisation

4.2.1 Ajout d'une dimension vide

Enfin, il est également possible de se poser la question de l'impact de la forme des matrices de pseudo-corrélations retenues. En effet, il est possible que l'assureur dispose dans sa modélisation de facteurs de pertes « faibles » :

$$P = \sum_{i\in\mathcal{P}, i\neq n+1} X_i + \epsilon X_{n+1}$$

avec ϵ un nombre réel très petit. A la limite nous considérerons $\epsilon = 0$, c'est à dire que le facteur de pertes considéré ne comptera pas dans l'évaluation des pertes, mais que les corrélations entre ce facteur de pertes et les autres facteurs de pertes seront listées dans la matrice de pseudo-corrélation.

Sans recourir aux algorithmes génétiques, l'impact de l'ajout d'un facteur de pertes n'intervenant pas dans la fonction de perte et corrélé à 0,1 avec les autres facteurs de pertes de la matrice de pseudo-corrélation sera analysé au travers d'un test dans le cadre de ce mémoire. Mathématiquement, la matrice $G \in \mathbb{R}^{n\times n}$ sera modifiée de la manière suivante, avant l'application de l'algorithme de PSDisation (en conservant la pondération initiale et sans pondérer les coefficients de corrélation additionnels), alors que la fonction de perte sera inchangée :

$$G \in \mathbb{R}^{(n+1)\times(n+1)} \leftarrow \begin{pmatrix} & & & 0,1 \\ & G \in \mathbb{R}^{n\times n} & & \vdots \\ & & & 0,1 \\ 0,1 & \dots & 0,1 & 1 \end{pmatrix}.$$

L'impact de cette transformation sur le SCR sera étudiée.

4.3 Conclusion

En résumé, compte-tenu des très nombreuses sensibilités pouvant être effectuées, il nous a paru utile d'étudier la sensibilité du SCR à trois phénomènes :

- la permutation (arbitraire) de l'ordre dans lequel les facteurs de pertes sont considérés, dont l'impact devrait théoriquement être nul ;
- le choix de pondération des coefficients dans l'algorithme de PSDisation ;
- l'ajout d'une dimension « vide » dans la matrice de pseudo-corrélation.

La section suivante présente les résultats obtenus sur des exemples concernant des matrices de dimension 3, 4, 5 et 10.

5 Interprétation des résultats numériques

L'objectif de cette partie est de présenter les résultats obtenus sur un ensemble d'exemples, d'analyser ces résultats et d'en tirer des conclusions concrètes pour la mise en place de processus internes encadrant l'étape de PSDisation.

5.1 Présentation des exemples choisis

La construction des exemples retenus a été guidée par les principes suivants :

- Analyser les résultats sur plusieurs dimensions (3, 4, 5 et 10) ;
- Faire intervenir des configurations différentes dans les fonctions d'approximation des pertes (termes négatifs, termes croisés etc.) ;
- Utiliser des matrices disposant de valeurs propres négatives ;
- Travailler avec des ordres de grandeur différents selon les exemples ;
- Utiliser des configurations de marginales variées (cas des moyennes et des volatilités de même ordre de grandeur, cas de différences un peu plus significatives entre les moyennes et les volatilités etc.) ;
- Conserver un maximum de diversivité dans les matrices retenues (coefficients positifs, négatifs, proches de 1 ou de -1 etc.) ;
- Travailler avec un exemple de dimension 10, conçu de toutes pièces, mais en lui essayant de lui donner un sens. Les exemples de dimensions 3, 4 et 5 ont été établis de manière purement algébrique.

Ces exemples sont présentés en détail dans l'annexe A. Les notations retenues sont les suivantes, pour une matrice initiale $G_\bullet$ (ou le symbole $\bullet$ représente la dimension suivie du numéro de l'exemple), une matrice de pondération $H_\bullet$ et des matrices d'encadrement $C_\bullet^{min}$ et $C_\bullet^{max}$, on note :

- $S_{\bullet PA}$ la matrice PSD obtenue à partir de $G_\bullet$ à l'aide de l'algorithme des projections alternées sans contraintes ni pondération (algorithme 2.3) ;
- $S_{\bullet N}$ la matrice PSD obtenue à partir de $G_\bullet$ à l'aide de l'algorithme de Newton sans contraintes ni pondération (algorithme 2.5) ;
- $S_\bullet$ la matrice PSD obtenue à partir de $G_\bullet$ à l'aide de l'algorithme hypersphère sans contraintes ni pondération (algorithme 2.9 avec $H = [1]_{(i,j)\in[\![1;n]\!]^2}$) ;
- $S_\bullet^H$ la matrice PSD obtenue à partir de $G_\bullet$ à l'aide de l'algorithme hypersphère sans contraintes avec une pondération correspondant à la matrice H (algorithme 2.9) ;
- et $S_\bullet^C$ la matrice PSD obtenue à partir de $G_\bullet$ à l'aide de l'algorithme hypersphère avec les contraintes fixées par $C_\bullet^{min}$ et $C_\bullet^{max}$ mais sans pondération (algorithme 2.10).

Chacun des exemples est étudié dans différentes configurations (copule gaussienne ou copule de Student, recours à des facteurs de pertes ou à des facteurs de risques). Les paramètres des lois marginales retenues diffèrent lorsque les exemples sont étudiés avec des facteurs de pertes ou avec des facteurs de risques, mais il s'agit toujours de lois lognormales. Les fonctions d'approximation des pertes sont présentées dans l'annexe A.

L'exemple de dimension 10 retenu est basé sur la matrice de pseudo-corrélation suivante :

$$G_{101} = \begin{pmatrix} 1 & -1 & -0,28 & 0 & 0,63 & -0,47 & 0,25 & 0,75 & 0 & 0 \\ -1 & 1 & 0,77 & 0,5 & -0,38 & 0,88 & 0,25 & 0,75 & 0,25 & 0,25 \\ -0,28 & 0,77 & 1 & 0,25 & 0,25 & 0,17 & -0,25 & 0,25 & 0,5 & 0,5 \\ 0 & 0,5 & 0,25 & 1 & 0,25 & 0,25 & 0 & 0,5 & 0,5 & 0,5 \\ 0,63 & -0,38 & 0,25 & 0,25 & 1 & 0,83 & 0,25 & 0,75 & 0 & 0 \\ -0,47 & 0,88 & 0,17 & 0,25 & 0,83 & 1 & 0,75 & 0,75 & 0 & 0 \\ 0,25 & 0,25 & -0,25 & 0 & 0,25 & 0,75 & 1 & 0,5 & 0,25 & 0,25 \\ 0,75 & 0,75 & 0,25 & 0,5 & 0,75 & 0,75 & 0,5 & 1 & 0,25 & 0,25 \\ 0 & 0,25 & 0,5 & 0,5 & 0 & 0 & 0,25 & 0,25 & 1 & 0,75 \\ 0 & 0,25 & 0,5 & 0,5 & 0 & 0 & 0,25 & 0,25 & 0,75 & 1 \end{pmatrix}$$

Cette matrice, déterminée en partie par des mesures sur les marchés (lorsque les coefficients ne sont pas « chiffres ronds »), et en partie par des jugements d'experts (coefficients $\in +/- \{0; 0,25; 0,5; 0,75\}$), représente la corrélation entre les facteurs de risques et de pertes suivants :

- Le premier facteur de risque X_1 représente le risque de baisse des taux ;
- Le deuxième facteur de risque X_2 représente le risque de hausse des taux ;
- Le troisième facteur de risque X_3 représente le risque d'inflation des dépenses ;
- Le quatrième facteur de risque X_4 représente le risque de niveau des dépenses ;
- Le cinquième facteur de risque X_5 représente le risque de hausse des *spread* ;
- Le sixième facteur de risque X_6 représente le risque de baisse des actions ;
- Le septième facteur de risque X_7 représente le risque de longévité ;
- Le huitième facteur de risque X_8 représente le risque de rachat massif ;
- Le neuvième facteur de pertes X_9 représente le risque de réserve en santé ;
- Le dixième facteur de pertes X_{10} représente le risque de réserve en incapacité temporaire de travail.

Les valeurs propres de la matrice G_{101} sont les suivantes :

$$\lambda \in \{3,93\,;\,2,72\,;\,2,00\,;\,1,14\,;\,0,74\,;\,0,57\,;\,0,25\,;\,-0,03\,;\,-0,55\,;\,-0,77\,\}.$$

5.2 Comparaison des algorithmes sans contraintes sur plusieurs exemples

Afin de vérifier la cohérence des résultats obtenus avec chacun des algorithmes de PSDisation sans contraintes (voir algorithmes 2.3, 2.5 et 2.9), des comparaisons de matrices obtenues dans chaque cas avec la norme 2 (correspondant à $H = (1)_{(i,j)\in[\![1;n]\!]^2}$ ou $W = I_n$) sont effectuées. Sur les exemples testés, présentés en Annexe A, les trois algorithmes donnent des solutions similaires (à 3 chiffres après la virgule pour chaque coefficient) dans les exemples avec les indices suivants : 31, 32, 41, 42, 51 et 52 (pour rappel : le premier chiffre de l'indice représente la dimension de la matrice associée à l'exemple, et le deuxième chiffre représente le numéro de l'exemple). Par contre, en dimension 10, les matrices S_{101}, S_{101PA} et S_{101N} présentent des différences notables. La distance entre la matrice d'arrivée et la matrice initiale G_{101} permet de classer les algorithmes par niveau d'efficacité. En l'occurence, l'algorithme des projections alternées fournit le meilleur résultat sur cet exemple :

$$\|S_{101PA} - G_{101}\|_2 = 1,21 \leq \|S_{101} - G_{101}\|_2 = 1,32 \leq \|S_{101N} - G_{101}\|_2 = 1,95.$$

L'algorithme hyperspère est susceptible d'avoir buté dans une zone d'attraction autour d'un minimum local. Cet exemple confirme donc les désavantages liés à cet algorithme et listés dans la section 2. L'algorithme hypersphère sera toutefois retenu dans ce mémoire car il permet de travailler aisément en norme H (ce que l'algorithme des projection alternées ne permet pas), et car il est utilisé de manière extensive par l'industrie pour sa robustesse et sa simplicité.

Il est à noter que l'algorithme de Newton fournit dans ce cas un résultat significativement moins bon que les deux autres algorithmes. Les raisons de ce mauvais résultat ne seront pas investiguées en détail ici, car cet algorithme n'est pas repris dans la suite du mémoire. Il est simplement à noter que la complexité de cet algorithme (passage dans un espace dual) le rend plus difficile à interpréter que les deux autres.

Commentaire : En tout état de cause, et quel que soit l'algorithme utilisé par l'assureur, une analyse détaillée des impacts de l'algorithme (quelles corrélations sont modifiées ? Quel est l'impact sur le SCR ? Quel serait l'impact d'un changement de point initial de l'algorithme de PSDisation ? etc.) est nécessaire de la part de l'utilisateur. La comparaison entre plusieurs algorithmes est une bonne pratique.

5.3 Analyse de la matrice S_{101}^H obtenue

La PSDisation avec l'algorithme de Rebonato pondéré fournit la matrice suivante : $S_{101}^H =$

$$\begin{pmatrix}
1 & -0,734 & -0,286 & -0,117 & 0,573 & -0,321 & 0,111 & 0,343 & 0,044 & 0,044 \\
-0,734 & 1 & 0,670 & 0,272 & -0,166 & 0,671 & 0,245 & 0,318 & 0,291 & 0,291 \\
-0,286 & 0,670 & 1 & 0,410 & 0,000 & 0,256 & -0,272 & 0,393 & 0,456 & 0,456 \\
-0,117 & 0,272 & 0,410 & 1 & 0,327 & 0,352 & 0,064 & 0,387 & 0,529 & 0,529 \\
0,573 & -0,166 & 0,000 & 0,327 & 1 & 0,518 & 0,365 & 0,811 & -0,070 & -0,070 \\
-0,321 & 0,671 & 0,256 & 0,352 & 0,518 & 1 & 0,613 & 0,700 & 0,003 & 0,003 \\
0,111 & 0,245 & -0,272 & 0,064 & 0,365 & 0,613 & 1 & 0,562 & 0,213 & 0,213 \\
0,343 & 0,318 & 0,393 & 0,387 & 0,811 & 0,700 & 0,562 & 1 & 0,310 & 0,310 \\
0,044 & 0,291 & 0,456 & 0,529 & -0,070 & 0,003 & 0,213 & 0,310 & 1 & 0,735 \\
0,044 & 0,291 & 0,456 & 0,529 & -0,070 & 0,003 & 0,213 & 0,310 & 0,735 & 1
\end{pmatrix}$$

Les valeurs propres de la matrice S_{101}^H sont les suivantes :

$$\lambda \in \{3,70\,;\, 2,49\,;\, 1,83\,;\, 1,05\,;\, 0,64\,;\, 0,27\,;\, 0,02\,;\, 0,00\,;\, 0,00\,;\, 0,00\,\}.$$

Les trois dernières valeurs propres (notées 0,00) sont égales à 10^{-5}, qui est le plancher fixé par l'algorithme. On remarque donc que les trois valeurs propres de la matrice initiale G_{101} ont été remplacées par des valeurs aussi faiblement positives que possible par l'algorithme. On note que la valeur propre qui a le plus été modifiée est la valeur propre initiale la plus négative (-0,77). Enfin, la distance (en norme H) de la matrice S_{101}^H à la matrice initiale G_{101} représente :

$$\frac{\|S_{101}^H - G_{101}\|_H}{\|G_{101}\|_H} = 4\%.$$

On observe que certains coefficients sont modifiés de manière significative par l'algorithme, comme par exemple le coefficient de corrélation (8,2) qui passe de 0,75 à 0,318 (corrélation entre le risque de hausse des taux et le risque de rachats massifs). Une analyse des modifications apportées par l'algorithme, coefficient par coefficient, semble donc nécessaire afin de garantir la bonne maîtrise par l'assureur de son modèle interne.

5.4 Impact de la PSDisation sur le SCR

La matrice initiale G_{101} n'étant pas positive semi-définie, il n'est pas possible de calculer l'exigence de capital qui résulterait de son utilisation. Néanmoins, il est possible de donner une évaluation « qualitative » de l'impact de la PSDisation à l'aide de la formule Var-Covar suivante, où X représente le

vecteur des pertes (correspondant au quantile à 99,5 de la distribution simulée) :

$$\frac{\left|X^T S_{101}^H X - X^T G_{101} X\right|}{X^T G_{101} X}$$

Cette analyse (effectuée avec une simulation de Sobol, des facteurs de pertes et une copule gaussienne pour identifier le vecteur de pertes) évalue l'impact de la PSDisation à 5% du SCR.

Commentaire : Ce calcul démontre l'importance des modifications apportées par la PSDisation à la structure de corrélation déterminée initialement par l'assureur. Ceci permet d'ores et déjà de confirmer que l'étape de PSDisation est cruciale pour l'assureur, qui devra porter une attention particulière à l'évolution de la matrice après PSDisation. En particulier, les experts de la société devront comprendre quels coefficients de la matrice ont été modifiés et si ces modifications sont conservatrices ou non par rapport aux valeurs initiales.

5.5 Difficultés rencontrées par l'algorithme avec contraintes inégalités

La problématique de minimum local est encore plus pregnante dans l'extension de l'algorithme hypersphère au cas de contraintes inégalités (voir algorithme 2.10). En effet, sur les exemples testés (31, 41 et 51), on remarque que l'algorithme part d'un point initial correspondant à une solution pondérée en norme H et a tendance à converger autant que possible vers la borne la plus proche fixée par l'encadrement imposé. Néanmoins cette convergence n'est pas toujours possible, surtout lorsque l'algorithme bute sur un minimum local (ce qui est le cas dans l'exemple 51).

Commentaire : L'extension proposée de l'algorithme hypersphère aux cas des contraintes inégalités (voir section 2.4) n'est pas suffisamment robuste du fait de la présence de minima locaux. Néanmoins, le rôle de la pondération (voir l'algorithme hypersphère 2.9) permet déjà de répondre à la contrainte pratique des assureurs (nécessité d'accorder une importance plus grande aux coefficients ayant un impact plus grand sur le SCR et une importance moindre aux coefficients déterminés par jugements d'experts ou par des formules de remplissage automatique de la matrice de pseudo-corrélation).

Remarque : Pour rappel, comme cela a été mentionné précédemment, il existe d'autres algorithmes (de type Newton), qui permettent un encadrement plutôt qu'une pondération. C'est un choix d'avoir axé cet étude sur un algorithme permettant uniquement la pondération (pour les raisons suivantes :

simplicité de la mise en oeuvre et de l'interprétation, algorithme très répandu dans les secteurs de la banque et de l'assurance, et pondération permettant de répondre à la même problématique concrète que l'encadrement).

5.6 Choix du générateur de nombres aléatoires

La comparaison des deux générateurs de nombres aléatoires peut être effectuée en estimant la volatilité du SCR à l'aide de 10 tirages successifs (alternativement avec un générateur de nombres aléatoires de Mersenne puis de Sobol). Les résultats présentés dans l'annexe A montrent que dans 22 cas testés sur 28, l'écart-type du SCR obtenu à l'aide de nombres de Sobol est inférieur à l'écart-type obtenu par des nombres de Mersenne. En conséquence, le générateur de nombres aléatoires de Sobol est utilisé pour les simulations. Il est à noter que l'écart-type du SCR simulé à l'aide de nombres de Sobol est systématiquement inférieur à 1% (rapporté à la moyenne).

5.7 Rappel sur les critères d'évaluation de l'impact d'une transformation

Pour rappel, comme expliqué dans la section 3.4, les critères suivants sont utilisés pour évaluer l'impact d'une transformation sur la matrice initiale de pseuo-corrélation G :

- La déformation sera dite *faible* si l'impact observé sur le SCR est inférieur à $2 \times 1,65 \times \widehat{\sigma}$ (avec $\widehat{\sigma}$ l'écart-type estimé pour le calcul du SCR dans la configuration étudiée) ;
- La déformation sera dite *forte* si l'impact observé sur le SCR est supérieur à $2 \times 1,65 \times \widehat{\sigma}$ et inférieur à $2 \times 2,89 \times \widehat{\sigma}$;
- La déformation sera dite *très forte* si l'impact observé sur le SCR est supérieur à $2 \times 2,89 \times \widehat{\sigma}$.

Dans le cas des algorithmes génétiques, le critère de déformation *très forte* est le plus pertinent, compte-tenu du nombre de calculs des SCR.

5.8 Impact sur le SCR d'une permutation génétique des marginales

Sur les **14 exemples testés**, l'application de l'algorithme d'optimisation génétique des permutations mène **systématiquement** à une déformation **très forte** du SCR, pouvant aller jusqu'à représenter plus de 6% du SCR (voir cas A.77, A.89 et A.105). En théorie, cette transformation ne devrait pourtant pas avoir d'impact sur le SCR.

Les figures 18 et 19 illustrent l'impact d'une permutation des facteurs de pertes sur une distribution de pertes donnée (exemple réalisé avec les matrices G_{52} et H_{52}, et dans le cas de facteurs de pertes).

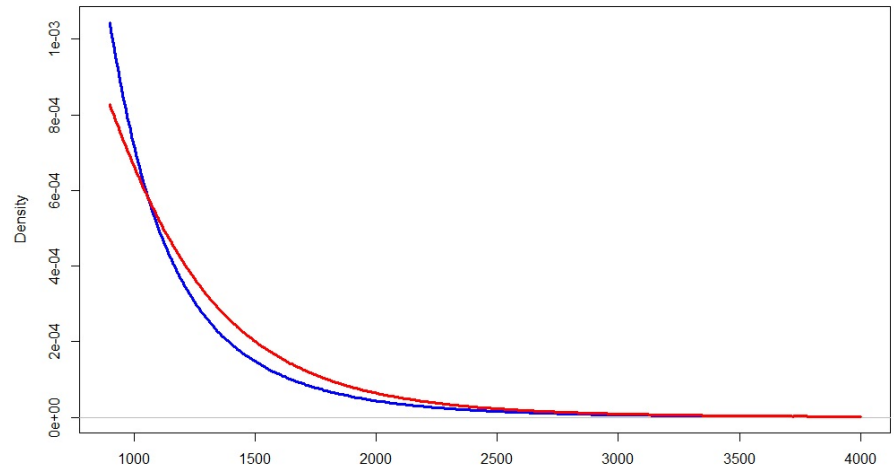


FIGURE 18 – Impact d'une permutation sur la distribution de pertes (1/2)

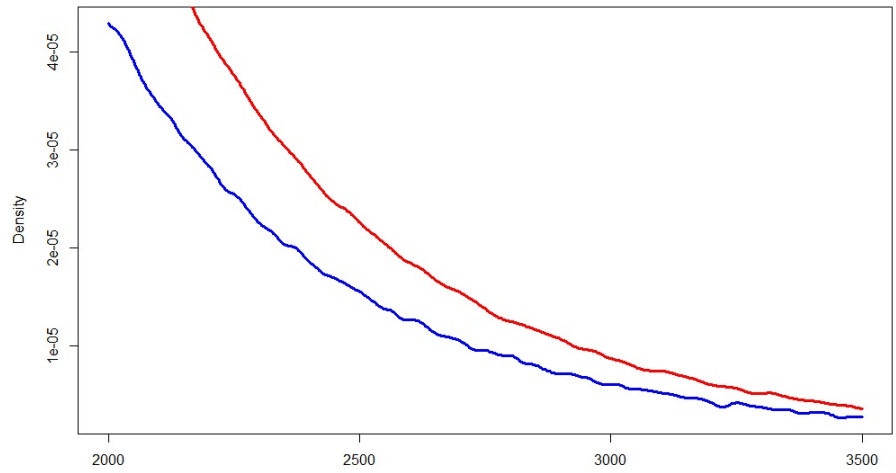


FIGURE 19 – Impact d'une permutation sur la distribution de pertes agrégées (2/2)

Il est à noter qu'en pratique les matrices de pseudo-corrélation sont plutôt constituées de sous-blocs (blocs marché, vie, non-vie etc.), ce qui contraindrait (légèrement) les permutations possibles.

Commentaire : Ces résultats montrent l'impact non-négligeable d'une

transformation pourtant théoriquement neutre de la martrice initiale. Cette absence de commutativité (entre la fonction de calcul du SCR et la fonction de permutation des facteurs de risques) est due à l'algorithme de PSDisation (et à la présence de minima locaux qui changent lors de permutations). Un biais est donc introduit dans le calcul du fait des choix initiaux (arbitraires) de l'assureur, et à cause de la présence de valeurs popres négatives dans la matrice initiale. En pratique, l'assureur peut chercher à définir des processus internes les plus robustes possibles, et permettant de définir autant que possible une matrice initiale PSD (identification des coefficients les « plus incertains » définis à dire d'experts, resollicitation des experts autant que nécessaire, identification de sous-matrices incohérentes etc.).

5.9 Impact sur le SCR d'une modification génétique de la pondération de la PSDisation

Sur les **28 exemples** testés avec un encadrement de la pondération entre $H^{min} = [0]_{(i,j)\in[\![1;n]\!]^2}$ et $H^{max} = [1]_{(i,j)\in[\![1;n]\!]^2}$, l'application de l'algorithme d'optimisation génétique de la pondération mène dans **18 cas** à une déformation **très forte** du SCR, pouvant aller jusqu'à représenter plus de 5% du SCR (voir cas A.78 et A.128).

Sur les **28 exemples** testés avec un encadrement de la pondération entre $H^{min} = H - [0,1]_{(i,j)\in[\![1;n]\!]^2}$ et $H^{max} = H + [0,1]_{(i,j)\in[\![1;n]\!]^2}$, l'application de l'algorithme d'optimisation génétique de la pondération mène dans **8 cas** à une déformation **très forte** du SCR, pouvant aller jusqu'à représenter plus de 4% du SCR (voir cas A.107).

Commentaire : Ces résultats mettent en exergue la sensibilité du SCR aux choix de l'assureur concernant l'algorithme de PSDisation en lui-même. Comme dans la section précédente, la mise en place de processus internes robustes afin d'obtenir une matrice initiale PSD permettrait de s'affranchir de ces difficultés. En tous cas, l'algorithme de PSDisation ne doit pas être considéré comme un boîte noire externalisée par l'assureur, mais doit être remis en cause et faire l'objet de multiples travaux de sensibilités pour retenir les choix jugés les plus appropriés.

5.10 Impact sur le SCR de l'ajout d'une dimension vide

Sur les **28 exemples testés**, l'ajout d'une dimension vide mène dans **10 cas** à une déformation **forte** du SCR, pouvant aller jusqu'à représenter plus de 6% du SCR (voir cas A.96). Néanmoins, en dehors de ce cas exceptionnel, les impacts sur le SCR sont plus modestes (de l'ordre de 1%). Encore une fois,

en théorie, cette transformation ne devrait pas avoir d'impact sur le SCR. Ces analyses conduites sur l'ajout d'une seule dimension vide ne constituent qu'une étude de sensibilité. Elles pourraient être complétées (ajout de plusieurs dimensions vides, modification des coefficients de corrélation de cette dimension vide fixés arbitrairement à 0,1 etc.)

Commentaire : Ces résultats confirment les analyses des deux sections précédentes : le calcul du SCR présente une réelle sensibilité à des choix *a priori* externes et sans impact. En particulier, au moment de prendre la décision d'ajouter un facteur de risques ou de pertes dans son modèle interne (par exemple pour augmenter la granularité des indices financiers modélisés ou des *lines of business* au passif), l'assureur doit analyser les gains associés à cette modélisation (matérialité du facteur de risque ajouté, utilité dans ses processus décisionnels) et les contraintes induites sur la matrice de corrélation (ajout d'une dimension supplémentaire qui peut dégrader le caractère PSD de la matrice initiale). Comme pour de nombreuses décisions, l'assureur doit arbitrer entre d'un côté, le désir de modéliser le plus finement les risques auxquels il est exposé, et de l'autre, la simplicité d'utilisation, la facilité d'interprétation et la maîtrise de son modèle.

5.11 Conclusion

Sur les **98 exemples** testés correspondants à la permutation des facteurs de risques, à l'optimisation de la pondération, et à l'ajout d'une dimension vide dans différentes configurations (facteurs de pertes / de risques, copule gaussienne / de Student), **50 cas** mènent à des déformations statistiquement significatives du SCR, alors que ces sensibilités sont censées être neutres ou à faible impact. Dans certains cas, cet intervalle peut représenter plus de 5% du SCR, ce qui correspond à plusieurs centaines de millions d'euros pour les plus grands assureurs européens.

Ces résultats soulignent donc l'importance du processus de PSDisation lors de l'agrégation des risques dans un modèle interne. Il est donc légitime pour les superviseurs et pour les assureurs que soient prévus des travaux de validation indépendante spécifiques à cette étape mathématique. Ceci conduit à l'ajout de la boucle en rouge ci-dessous par rapport à la figure 1 de l'introduction :

En conclusion, les meilleures pratiques suivantes peuvent être relevées concernant l'étape de PSDisation lors de l'agrégation dans les modèles internes :

— En amont de la PSDisation, prévoir des tests algébriques pour détec-

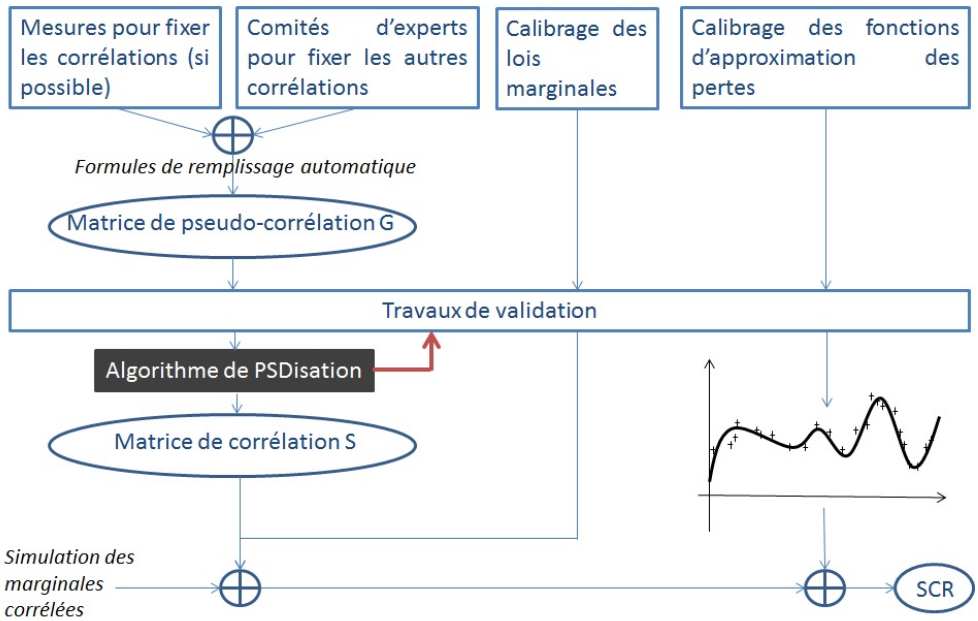


FIGURE 20 – Processus de calcul du SCR en modèle interne avec boucle de validation de la PSDisation

ter les choix incohérents concernant la matrice de pseudo-corrélation et prévoir dans les calendriers de production plusieurs échanges avec les experts de la société pour analyser ces incohérences, identifier les coefficients ayant le moins d'impact sur l'exigence de capital générant des valeurs propres négatives et modifier autant que possible ces choix ;

— Eviter de modéliser un trop grand nombre de facteurs de risques ou de pertes (dans la mesure du possible), surtout si les impacts de ces facteurs de risques sur le SCR et sur les processus décisionnels de la compagnie sont négligeables ;

— Effectuer un nombre important de travaux de sensibilité associé à la PSDisation (permutation des facteurs de risques, changements de pondération etc.)

— Inclure une étape de validation indépendante dédiée à la PSDisation, avec notamment une identification des coefficients ayant subi les fortes modifications au cours de cette transformation mathématique ;

— Enfin, si possible, comparer les résultats issus de plusieurs algorithmes de PSDisation (notamment en vue de détecter des minima locaux et de choisir le meilleur résultat, c'est à dire la matrice la proche de la matrice initiale au sens de la « norme H »).

Conclusion

Les travaux menés dans ce mémoire permettent de confirmer la grande sensibilité des résultats de calcul d'exigence de capital à des modifications de la matrice de pseudo-corrélation même théoriquement sans impact. Par exemple, les résultats obtenus numériquement montrent que l'impact d'une simple permutation des facteurs de risques préalablement à l'étape de PSDisation peut mener à un intervalle d'incertitude autour du SCR supérieur à 5%, ce qui peut représenter des montants conséquents pour certains assureurs systémiques (plusieurs centaines de millions d'euros).

Ceci permet de confirmer :

— d'une part l'importance de l'environnement de contrôle interne autour du processus général d'agrégation des risques dans les modèles internes (y-compris pour l'étape de PSDisation). Cet environnement de contrôle interne requiert des processus clairs, qui font intervenir toutes les personnes nécessaires à la fixation des coefficients de corrélation à dire d'expert, et qui documentent leur décisions. Les algorithmes de remplissage automatiques doivent être prudents, clairement documentés et leurs résultats doivent faire l'objet d'une analyse à chaque nouvelle publication ou utilisation de chiffres sur l'exigence de capital. Enfin, une analyse des coefficients pré et post PSDisation, ainsi que l'impact de ces modifications, est nécessaire. Les coefficients les plus sensibles doivent être identifiés et leur modification encadrée ou fortement pondérée. La robustesse du processus peut être testée à l'aide de modifications « neutres » de la matrice préPSD (permutation des colonnes, ajout de dimensions vides etc.). Enfin, la taille des matrices de pseudo-corrélation doit être réduite autant que possible, de manière à limiter les écarts potentiels à une matrice non PSD ;

— d'autre part, en tout état de cause, si les analyses issues du processus de contrôle interne en démontrent la nécessité, il appartient aux organismes concernés de prendre des marges de prudence adéquates pour garantir une gestion des risques saine. Les travaux de ce mémoire montrent que dans les conditions analysées, le manque de robustesse du processus de PSDisation (dans les conditions étudiées) pourrait conduire à des marges conservatrices représentant jusqu'à 5% du SCR dans certains cas.

Ces travaux pourraient être poursuivis avec par exemple les axes de développement suivants :

— Définir un ensemble de tests algébriques sur la matrice de pseudo-corrélation afin de détecter quels coefficients déterminés par jugements d'experts conduisent à la présence de valeurs propres négatives) ;

— Etendre ces travaux à des matrices de plus grande dimension (1000 x 1000 par exemple, ce qui est plus proche des conditions rencontrées par les assureurs). Ceci nécessitera préalablement une amélioration des algorithmes utilisés ;
— Analyser algébriquement en petite dimension quelles sont les caractéristiques des permutations menant aux SCR minimum et maximum (appartenance à une classe de conjugaison etc.). Ceci pourrait permettre d'en déduire des bonnes pratiques dans l'ordonnancement des facteurs de risques ;
— Développer un algorithme génétique avec l'ajout de plusieurs dimensions vides ;
— Développer d'autres algorithmes de PSDisation avec encadrement (de type Newton) et comparer aux résultats de l'algorithme hypersphère avec encadrement ;
— Analyser quel est le meilleur point de départ possible des divers algorithmes d'optimisation utilisés dans ce mémoire. En particulier, cette analyse pourrait être développée pour les algorithmes d'optimisation sous contraintes (dans ce cas, le point de départ est le couple (λ_0, θ_0) où λ représente le vecteur des multiplicateurs de Lagrange et θ le vecteur angulaire de la décomposition hypersphère) ;
— Analyser quelles contraintes mathématiques pèsent sur les algorithmes d'optimisation avec encadrement. En particulier, dans quelles conditions l'algorithme avec encadrement a-t-il bien une solution ?
— Développer un algorithme hypersphère de convergence quadratique (algorithme de type Newton) et comparer les vitesses de convergence pour des matrices de grandes dimensions ;
— Développer d'autres algorithmes permettant la pondération (voir par exemple [QIS2]) et comparer les résultats avec l'algorithme hypersphère.

Références

[REB] Rebonato, Jäckel. *The most general methodology to create a valid correlation matrix for risk management and option pricing purposes*, Quantitative Research Centre of the NatWest Group. (1999).

[HIG] Higham. *Computing the nearest correlation matrix - a problem from finance*, IMA J Anal (2002) 22 (3) 329 - 343.

[BOR] Borsdorf. *A Newton algorithm for computing the nearest corelation matrix*, MIMS EPrint : 2008.49.

[QIS] Qi, Sun. *A quadratically convergent Newton method for computing the nearest correlation matrix*, SIAM. J. Matrix Anal. & Appl., 28(2), 360–385. (2005).

[GOL] Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley. (1989).

[QIS2] Qi, Sun. *Au augmented Lagrangian dual approach for the H-weighted nearest correlation matrix problem*, IMA J Numer Anal (2011) 31 (2) : 491-511.

[PET] Petit. *Optimisation*. Ecole des Mines de Paris, 2007.

[DEC] Denuit, Charpentier. *Mathématiques de l'assurance non-vie Tome I : Principes fondamentaux de la théorie du risque*. Economica, 2003.

[PCO] Colmez. *Eléments d'analyse et d'algèbre*. Editions de l'école polyechnique, 2012.

[JAC] Jäckel. *Monte Carlo methods in finance*. Wiley and sons. (2002).

[DEC] Decupère. *Agrégation des risques et allocation de capital sous Solvabilité 2*. Mémoire d'actuariat. (2011)

[DRI] Drissi Boutaybi. *Technique d'agrégation des risques*. Mémoire d'actuariat. (2014)

[DRL] Dreyfuss, Lai. *Méthodes avancées d'agrégation des risques sous Solvabilité 2*. Mémoire d'actuariat. (2013)

[AUB] Aubron. *Etude de l'impact des dépendances sur le Capital Economique*. Mémoire d'actuariat. (2009).

[S2] DIRECTIVE 2009/138/CE DU PARLEMENT EUROPÉEN ET DU CONSEIL du 25 novembre 2009 sur l'accès aux activités de l'assurance et de la réassurance et leur exercice (solvabilité II)

[RD] RÈGLEMENT DÉLÉGUÉ (UE) 2015/35 DE LA COMMISSION du 10 octobre 2014 complétant la directive 2009/138/CE du Parlement européen et du Conseil sur l'accès aux activités de l'assurance et de la réassurance et leur exercice (solvabilité II)

A Paramètres et tests effectués

A.1 Matrices utilisées dans ce mémoire

Voici les matrices qui ont été utilisées dans ce mémoire pour les différentes simulations :

$$G_{31} = \begin{pmatrix} 1 & -0,9 & -0,5 \\ -0,9 & 1 & 0,5 \\ -0,5 & 0,5 & 1 \end{pmatrix}, H_{31} = \begin{pmatrix} 1 & 0,9 & 0,8 \\ 0,9 & 1 & 0,1 \\ 0,8 & 0,1 & 1 \end{pmatrix}$$

$$C_{31}^{min} = \begin{pmatrix} 1 & -1 & -0,6 \\ -1 & 1 & -0,9 \\ -0,6 & -0,9 & 1 \end{pmatrix}, C_{31}^{max} = \begin{pmatrix} 1 & -0,8 & -0,4 \\ -0,8 & 1 & -0,1 \\ -0,4 & -0,1 & 1 \end{pmatrix}$$

$$G_{32} = \begin{pmatrix} 1 & -0,6 & 0,5 \\ -0,6 & 1 & 0,9 \\ 0,5 & 0,9 & 1 \end{pmatrix}, H_{32} = \begin{pmatrix} 1 & 0,1 & 0,1 \\ 0,1 & 1 & 0,9 \\ 0,1 & 0,9 & 1 \end{pmatrix}$$

$$G_{41} = \begin{pmatrix} 1 & 0,9 & -0,5 & -0,7 \\ 0,9 & 1 & 0,2 & 0,1 \\ -0,5 & 0,2 & 1 & 0,1 \\ -0,7 & 0,1 & 0,1 & 1 \end{pmatrix}, H_{41} = \begin{pmatrix} 1 & 0,9 & 0,2 & 0,9 \\ 0,9 & 1 & 0,2 & 0,1 \\ 0,2 & 0,2 & 1 & 0,1 \\ 0,9 & 0,1 & 0,1 & 1 \end{pmatrix}$$

$$C_{41}^{min} = \begin{pmatrix} 1 & 0,8 & -0,6 & -0,9 \\ 0,8 & 1 & 0 & 0 \\ -0,6 & 0 & 1 & 0 \\ -0,9 & 0 & 0 & 1 \end{pmatrix}, C_{41}^{max} = \begin{pmatrix} 1 & 1 & -0,4 & -0,5 \\ 1 & 1 & 0,4 & 0,2 \\ -0,4 & 0,4 & 1 & 0,2 \\ -0,5 & 0,2 & 0,2 & 1 \end{pmatrix}$$

$$G_{42} = \begin{pmatrix} 1 & 0,1 & -0,8 & 0,75 \\ 0,1 & 1 & 0,2 & 0,1 \\ -0,8 & 0,2 & 1 & 0,9 \\ 0,75 & 0,1 & 0,9 & 1 \end{pmatrix}, H_{42} = \begin{pmatrix} 1 & 0,1 & 0,9 & 0,2 \\ 0,1 & 1 & 0,1 & 0,1 \\ 0,9 & 0,1 & 1 & 0,9 \\ 0,2 & 0,1 & 0,9 & 1 \end{pmatrix}$$

$$G_{51} = \begin{pmatrix} 1 & -0,8 & -0,9 & -0,9 & 0,2 \\ -0,8 & 1 & 0,9 & -0,7 & 0,6 \\ -0,9 & 0,9 & 1 & 0,2 & -0,6 \\ -0,9 & -0,7 & 0,2 & 1 & -0,1 \\ 0,2 & 0,6 & -0,6 & -0,1 & 1 \end{pmatrix}, H_{51} = \begin{pmatrix} 1 & 0,9 & 0,1 & 0,9 & 0,1 \\ 0,9 & 1 & 0,6 & 0,1 & 0,1 \\ 0,1 & 0,6 & 1 & 2 & 2 \\ 0,9 & 0,1 & 0,2 & 1 & 0,1 \\ 0,1 & 0,1 & 0,2 & 0,1 & 1 \end{pmatrix}$$

$$C_{51}^{min} = \begin{pmatrix} 1 & -0,9 & -1 & -1 & 0 \\ -0,9 & 1 & 0,8 & -1 & 0,2 \\ -1 & 0,8 & 1 & 0 & -1 \\ -1 & -1 & 0 & 1 & -0,2 \\ -0 & 0,2 & -1 & -0,2 & 1 \end{pmatrix}, C_{51}^{max} = \begin{pmatrix} 1 & -0,6 & -0,5 & -0,7 & 0,4 \\ -0,6 & 1 & 0,77 & 2 & 2 \\ -0,5 & 0,77 & 1 & 2 & 2 \\ -0,7 & 0,77 & 2 & 1 & 2 \\ 0,4 & 0,77 & 2 & 2 & 1 \end{pmatrix}$$

$$G_{52} = \begin{pmatrix} 1 & 0,7 & -0,8 & -0,8 & 0,2 \\ 0,7 & 1 & 0,8 & -0,6 & 0,6 \\ -0,8 & 0,8 & 1 & 0,2 & -0,6 \\ -0,8 & -0,6 & 0,2 & 1 & 0,8 \\ 0,2 & 0,6 & -0,6 & 0,8 & 1 \end{pmatrix}, H_{52} = \begin{pmatrix} 1 & 0,1 & 0,9 & 0,9 & 0,1 \\ 0,1 & 1 & 0,1 & 0,9 & 0,1 \\ 0,9 & 0,1 & 1 & 0,1 & 0,9 \\ 0,9 & 0,9 & 0,1 & 1 & 2 \\ 0,1 & 0,1 & 0,9 & 0,1 & 1 \end{pmatrix}$$

$$G_{101} = \begin{pmatrix} 1 & -1 & -0,28 & 0 & 0,63 & -0,47 & 0,25 & 0,75 & 0 & 0 \\ -1 & 1 & 0,77 & 0,5 & -0,38 & 0,88 & 0,25 & 0,75 & 0,25 & 0,25 \\ -0,28 & 0,77 & 1 & 0,25 & 0,25 & 0,17 & -0,25 & 0,25 & 0,5 & 0,5 \\ 0 & 0,5 & 0,25 & 1 & 0,25 & 0,25 & 0 & 0,5 & 0,5 & 0,5 \\ 0,63 & -0,38 & 0,25 & 0,25 & 1 & 0,83 & 0,25 & 0,75 & 0 & 0 \\ -0,47 & 0,88 & 0,17 & 0,25 & 0,83 & 1 & 0,75 & 0,75 & 0 & 0 \\ 0,25 & 0,25 & -0,25 & 0 & 0,25 & 0,75 & 1 & 0,5 & 0,25 & 0,25 \\ 0,75 & 0,75 & 0,25 & 0,5 & 0,75 & 0,75 & 0,5 & 1 & 0,25 & 0,25 \\ 0 & 0,25 & 0,5 & 0,5 & 0 & 0 & 0,25 & 0,25 & 1 & 0,75 \\ 0 & 0,25 & 0,5 & 0,5 & 0 & 0 & 0,25 & 0,25 & 0,75 & 1 \end{pmatrix}$$

$$H_{101} = \begin{pmatrix} 1 & 0,3 & 0,9 & 0,3 & 0,9 & 0,9 & 0,3 & 0,3 & 0,3 & 0,3 \\ 0,9 & 1 & 0,9 & 0,3 & 0,9 & 0,9 & 0,3 & 0,3 & 0,3 & 0,3 \\ 0,3 & 0,9 & 1 & 0,3 & 0,3 & 0,9 & 0,3 & 0,3 & 0,3 & 0,3 \\ 0,3 & 0,3 & 0,3 & 1 & 0,3 & 0,3 & 0,3 & 0,3 & 0,3 & 0,3 \\ 0,9 & 0,9 & 0,3 & 0,3 & 1 & 0,9 & 0,3 & 0,3 & 0,3 & 0,3 \\ 0,9 & 0,9 & 0,9 & 0,3 & 0,9 & 1 & 0,3 & 0,3 & 0,3 & 0,3 \\ 0,3 & 0,3 & 0,3 & 0,3 & 0,3 & 0,3 & 1 & 0,3 & 0,3 & 0,3 \\ 0,3 & 0,3 & 0,3 & 0,3 & 0,3 & 0,3 & 0,3 & 1 & 0,3 & 0,3 \\ 0,3 & 0,3 & 0,3 & 0,3 & 0,3 & 0,3 & 0,3 & 0,3 & 1 & 0,3 \\ 0,3 & 0,3 & 0,3 & 0,3 & 0,3 & 0,3 & 0,3 & 0,3 & 0,3 & 1 \end{pmatrix}$$

De manière générale, pour une matrice initiale $G_\bullet$, une matrice de pondération $H_\bullet$ et des matrices d'encadrement $C_\bullet^{min}$ et $C_\bullet^{max}$, on note :

— $S_{\bullet PA}$ la matrice PSD obtenue à partir de $G_\bullet$ à l'aide de l'algorithme des projections alternées sans contraintes ni pondération (algorithme 2.3) ;

— $S_{\bullet N}$ la matrice PSD obtenue à partir de $G_\bullet$ à l'aide de l'algorithme de Newton sans contraintes ni pondération (algorithme 2.5) ;
— $S_\bullet$ la matrice PSD obtenue à partir de $G_\bullet$ à l'aide de l'algorithme hypersphère sans contraintes ni pondération (algorithme 2.9 avec $H = [1]_{(i,j)\in[\![1;n]\!]^2}$) ;
— $S_\bullet^H$ la matrice PSD obtenue à partir de $G_\bullet$ à l'aide de l'algorithme hypersphère sans contraintes avec une pondération correspondant à la matrice H (algorithme 2.9) ;
— et $S_\bullet^C$ la matrice PSD obtenue à partir de $G_\bullet$ à l'aide de l'algorithme hypersphère avec les contraintes fixées par $C_\bullet^{min}$ et $C_\bullet^{max}$ mais sans pondération (algorithme 2.10).

A.2 Marginales utilisées dans ce mémoire

Voici les paramètres des marginales utilisées dans ce mémoire (toutes les marginales considérées suivent des lois log-normales). De manière générale, on note d'un signe b en indice des vecteurs aléatoires utilisés dans le cas des facteurs de risque (c'est à dire faisant intervenir des fonctions de pertes polynomiales) :

$X_{III} =$

	X_1	X_2	X_3
μ	6	5	4
σ	0,5	0,5	0,1

, $X_{IIIb} =$

	X_1	X_2	X_3
μ	6	3	12
σ	0,8	0,1	2

$X_{IV} =$

	X_1	X_2	X_3	X_4
μ	6	5	4	3
σ	0,5	0,5	0,1	0,9

, $X_{IVb} =$

	X_1	X_2	X_3	X_4
μ	1	2	3	1
σ	5	4	3	4

$X_V =$

	X_1	X_2	X_3	X_4	X_5
μ	6	5	4	4	3
σ	0,5	0,5	0,1	0,9	0,3

, $X_{Vb} =$

	X_1	X_2	X_3	X_4	X_5
μ	1	2	1	2	1
σ	3	2	1	4	3

$X_X =$

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}
μ	6	5	5	6	4	5	5	5	5	6
σ	0,5	0,5	0,2	0,1	0,3	0,6	0,5	0,4	0,3	0,3

$X_{Xb} =$

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}
μ	1,2	2	0,3	2	1,5	3,5	1,5	2,5	5	6
σ	2	1,5	2,5	2	2	1	2	2	2	2

A.3 Fonctions de pertes utilisées dans ce mémoire

Les fonctions de pertes utilisées dans ce mémoire sont listées ci-dessous. On fera référence aux « facteurs de risques »dans le cas de recours à ces fonctions de pertes (par opposition aux « facteurs de pertes »lorsque la fonction de pertes associées est une simple somme des marginales) :

- En dimension 3, $P = f((X_i)_{i\in[\![1;3]\!]}) = 0,5X_1^2 + 2X_2^4 + 0,3X_3 + 10X_1X_2$;
- En dimension 4, $P = f((X_i)_{i\in[\![1;4]\!]}) = 5X_1 + 0,02X_2^2 + 2000X_3 + 5X_4^{\frac{3}{2}}$;
- En dimension 5, $P = f((X_i)_{i\in[\![1;5]\!]}) = X_1 + 0,1X_2^2 + 50X_3 + X_4^{\frac{1}{2}} - 1,2X_5$;
- En dimension 10, $P = f((X_i)_{i\in[\![1;10]\!]}) = 0,5X_1^3 + 0,4X_2^3 + 3X_3^2 + 2X_4^2 + \sum_{i=5}^{7} X_i^2 + 0,5X_8^2 + \sum_{i=9}^{10} X_i$.

A.4 Comparaison des 3 algorithmes sans contraintes

Voici les résultats obtenus avec les 3 algorithmes :

$$S_{31PA} = \begin{pmatrix} 1 & -0,725 & -0,371 \\ -0,725 & 1 & 0,371 \\ -0,371 & 0,371 & 1 \end{pmatrix}, S_{31N} = \begin{pmatrix} 1 & -0,725 & -0,371 \\ -0,725 & 1 & 0,371 \\ -0,371 & 0,371 & 1 \end{pmatrix}$$

$$S_{31} = \begin{pmatrix} 1 & -0,725 & -0,371 \\ -0,725 & 1 & 0,371 \\ -0,371 & 0,371 & 1 \end{pmatrix}$$

$$S_{32PA} = \begin{pmatrix} 1 & -0,436 & 0,343 \\ -0,436 & 1 & 0,695 \\ 0,343 & 0,695 & 1 \end{pmatrix}, S_{32N} = \begin{pmatrix} 1 & -0,436 & 0,343 \\ -0,436 & 1 & 0,695 \\ 0,343 & 0,695 & 1 \end{pmatrix}$$

$$S_{32} = \begin{pmatrix} 1 & -0,436 & 0,343 \\ -0,436 & 1 & 0,695 \\ 0,343 & 0,695 & 1 \end{pmatrix}$$

$$S_{41PA} = \begin{pmatrix} 1 & 0,711 & -0,398 & -0,573 \\ 0,711 & 1 & 0,122 & 0,003 \\ -0,398 & 0,122 & 1 & 0,152 \\ -0,573 & 0,003 & 0,152 & 1 \end{pmatrix}$$

$$S_{41N} = \begin{pmatrix} 1 & 0,711 & -0,398 & -0,573 \\ 0,711 & 1 & 0,122 & 0,003 \\ -0,398 & 0,122 & 1 & 0,152 \\ -0,573 & 0,003 & 0,152 & 1 \end{pmatrix}$$

$$S_{41} = \begin{pmatrix} 1 & 0,711 & -0,398 & -0,573 \\ 0,711 & 1 & 0,122 & 0,003 \\ -0,398 & 0,122 & 1 & 0,152 \\ -0,573 & 0,003 & 0,152 & 1 \end{pmatrix}$$

$$S_{42PA} = \begin{pmatrix} 1 & 0,068 & -0,481 & 0,444 \\ 0,068 & 1 & 0,165 & 0,133 \\ -0,481 & 0,165 & 1 & 0,566 \\ 0,444 & 0,133 & 0,566 & 1 \end{pmatrix}$$

$$S_{42N} = \begin{pmatrix} 1 & 0,068 & -0,481 & 0,444 \\ 0,068 & 1 & 0,165 & 0,133 \\ -0,481 & 0,165 & 1 & 0,566 \\ 0,444 & 0,133 & 0,566 & 1 \end{pmatrix}$$

$$S_{42} = \begin{pmatrix} 1 & 0,068 & -0,481 & 0,444 \\ 0,068 & 1 & 0,165 & 0,133 \\ -0,481 & 0,165 & 1 & 0,566 \\ 0,444 & 0,133 & 0,566 & 1 \end{pmatrix}$$

$$S_{51PA} = \begin{pmatrix} 1 & -0,508 & -0,830 & -0,585 & 0,140 \\ -0,508 & 1 & 0,651 & -0,360 & 0,350 \\ -0,830 & 0,651 & 1 & 0,148 & -0,409 \\ -0,585 & -0,360 & 0,148 & 1 & -0,221 \\ 0,140 & 0,350 & -0,409 & -0,221 & 1 \end{pmatrix}$$

$$S_{51N} = \begin{pmatrix} 1 & -0,508 & -0,830 & -0,585 & 0,140 \\ -0,508 & 1 & 0,651 & -0,360 & 0,350 \\ -0,830 & 0,651 & 1 & 0,148 & -0,409 \\ -0,585 & -0,360 & 0,148 & 1 & -0,221 \\ 0,140 & 0,350 & -0,409 & -0,221 & 1 \end{pmatrix}$$

$$S_{51} = \begin{pmatrix} 1 & -0,508 & -0,830 & -0,585 & 0,140 \\ -0,508 & 1 & 0,651 & -0,360 & 0,350 \\ -0,830 & 0,651 & 1 & 0,148 & -0,409 \\ -0,585 & -0,360 & 0,148 & 1 & -0,221 \\ 0,140 & 0,350 & -0,409 & -0,221 & 1 \end{pmatrix}$$

$$S_{52PA} = \begin{pmatrix} 1 & 0,483 & -0,544 & -0,697 & 0,225 \\ 0,483 & 1 & 0,392 & -0,392 & 0,282 \\ -0,544 & 0,392 & 1 & 0,060 & -0,337 \\ -0,697 & -0,392 & 0,060 & 1 & 0,513 \\ 0,225 & 0,282 & -0,337 & 0,513 & 1 \end{pmatrix}$$

$$S_{52N} = \begin{pmatrix} 1 & 0,483 & -0,544 & -0,697 & 0,225 \\ 0,483 & 1 & 0,392 & -0,392 & 0,282 \\ -0,544 & 0,392 & 1 & 0,060 & -0,337 \\ -0,697 & -0,392 & 0,060 & 1 & 0,513 \\ 0,225 & 0,282 & -0,337 & 0,513 & 1 \end{pmatrix}$$

$$S_{52} = \begin{pmatrix} 1 & 0,483 & -0,544 & -0,697 & 0,225 \\ 0,483 & 1 & 0,392 & -0,392 & 0,282 \\ -0,544 & 0,392 & 1 & 0,060 & -0,337 \\ -0,697 & -0,392 & 0,060 & 1 & 0,513 \\ 0,225 & 0,282 & -0,337 & 0,513 & 1 \end{pmatrix}$$

$S_{101PA} =$

$$\begin{pmatrix} 1 & -0,681 & -0,310 & -0,008 & 0,574 & -0,241 & 0,158 & 0,390 & 0,017 & 0,017 \\ -0,681 & 1 & 0,596 & 0,434 & -0,010 & 0,684 & 0,233 & 0,390 & 0,269 & 0,269 \\ -0,310 & 0,596 & 1 & 0,295 & 0,144 & 0,256 & -0,222 & 0,293 & 0,480 & 0,480 \\ -0,008 & 0,434 & 0,295 & 1 & 0,210 & 0,276 & 0,024 & 0,508 & 0,487 & 0,487 \\ 0,574 & -0,010 & 0,144 & 0,210 & 1 & 0,520 & 0,308 & 0,710 & -0,001 & -0,001 \\ -0,241 & 0,684 & 0,256 & 0,276 & 0,520 & 1 & 0,625 & 0,674 & 0,019 & 0,019 \\ 0,158 & 0,233 & -0,222 & 0,024 & 0,308 & 0,625 & 1 & 0,537 & 0,226 & 0,226 \\ 0,390 & 0,390 & 0,293 & 0,508 & 0,710 & 0,674 & 0,537 & 1 & 0,245 & 0,245 \\ 0,017 & 0,269 & 0,480 & 0,487 & -0,001 & 0,019 & 0,226 & 0,245 & 1 & 0,760 \\ 0,017 & 0,269 & 0,480 & 0,487 & -0,001 & 0,019 & 0,229 & 0,245 & 0,760 & 1 \end{pmatrix}$$

$S_{101N} =$

$$\begin{pmatrix}
1 & -572 & -0,264 & 0,029 & 0,315 & -0,184 & 0,134 & 0,400 & -0,020 & -0,020 \\
-0,572 & 1 & 0,604 & 0,390 & -0,157 & 0,495 & 0,202 & 0,451 & 0,235 & 0,235 \\
-0,264 & 0,604 & 1 & 0,838 & 0,001 & 0,196 & -0,154 & 0,236 & 0,400 & 0,400 \\
0,029 & 0,390 & 0,838 & 1 & 0,392 & 0,302 & 0,088 & 0,431 & 0,672 & 0,672 \\
0,315 & -0,157 & 0,001 & 0,392 & 1 & 0,420 & 0,270 & 0,454 & 0,015 & 0,015 \\
-0,184 & 0,495 & 0,196 & 0,302 & 0,420 & 1 & 0,457 & 0,426 & 0,051 & 0,051 \\
0,134 & 0,202 & -0,154 & 0,088 & 0,270 & 0,457 & 1 & 0,414 & 0,178 & 0,178 \\
0,400 & 0,451 & 0,236 & 0,431 & 0,454 & 0,426 & 0,414 & 1 & 0,200 & 0,200 \\
-0,020 & 0,235 & 0,400 & 0,672 & 0,015 & 0,051 & 0,178 & 0,200 & 1 & 1,000 \\
-0,020 & 0,235 & 0,400 & 0,672 & 0,015 & 0,051 & 0,178 & 0,200 & 1,000 & 1
\end{pmatrix}$$

$S_{101} =$

$$\begin{pmatrix}
1 & -0,706 & -0,291 & -0,099 & 0,610 & -0,244 & 0,171 & 0,374 & 0,043 & 0,043 \\
-0,706 & 1 & 0,647 & 0,304 & -0,102 & 0,697 & 0,263 & 0,345 & 0,304 & 0,304 \\
-0,291 & 0,647 & 1 & 0,384 & 0,139 & 0,253 & -0,297 & 0,359 & 0,449 & 0,449 \\
-0,099 & 0,304 & 0,384 & 1 & 0,293 & 0,311 & 0,101 & 0,411 & 0,514 & 0,514 \\
0,610 & -0,102 & 0,139 & 0,293 & 1 & 0,494 & 0,293 & 0,828 & -0,052 & -0,052 \\
-0,244 & 0,697 & 0,253 & 0,311 & 0,494 & 1 & 0,640 & 0,734 & -0,008 & -0,008 \\
0,171 & 0,263 & -0,297 & 0,101 & 0,293 & 0,640 & 1 & 0,604 & 0,196 & 0,196 \\
0,374 & 0,345 & 0,359 & 0,411 & 0,828 & 0,734 & 0,604 & 1 & 0,306 & 0,306 \\
0,043 & 0,304 & 0,449 & 0,514 & -0,052 & -0,008 & 0,196 & 0,306 & 1 & 0,737 \\
0,043 & 0,304 & 0,449 & 0,514 & -0,052 & -0,008 & 0,196 & 0,306 & 0,737 & 1
\end{pmatrix}$$

A.5 Résultats des PSDisations avec pondération

$$S_{31}^H = \begin{pmatrix} 1 & -0,848 & -0,505 \\ -0,848 & 1 & -0,029 \\ -0,505 & -0,029 & 1 \end{pmatrix}, S_{32}^H = \begin{pmatrix} 1 & -0,278 & 0,187 \\ -0,278 & 1 & 0,892 \\ 0,187 & 0,892 & 1 \end{pmatrix}$$

$$S_{41}^H = \begin{pmatrix} 1 & 0,844 & -0,401 & -0,707 \\ 0,844 & 1 & -0,029 & -0,329 \\ -0,401 & -0,029 & 1 & 0,216 \\ -0,707 & -0,329 & 0,216 & 1 \end{pmatrix}$$

$$S_{42}^H = \begin{pmatrix} 1 & 0,041 & -0,614 & 0,041 \\ 0,041 & 1 & 0,129 & 0,144 \\ -0,614 & 0,129 & 1 & 0,763 \\ 0,041 & 0,144 & 0,763 & 1 \end{pmatrix}$$

$$S_{51}^{H} = \begin{pmatrix} 1 & -0,822 & -0,795 & -0,811 & 0,132 \\ -0,822 & 1 & 0,834 & 0,353 & 0,155 \\ -0,795 & 0,834 & 1 & 0,371 & -0,412 \\ -0,811 & 0,353 & 0,371 & 1 & -0,193 \\ 0,132 & 0,155 & -0,412 & -0,193 & 1 \end{pmatrix}$$

$$S_{52}^{H} = \begin{pmatrix} 1 & 0,569 & -0,714 & -0,776 & 0,310 \\ 0,569 & 1 & 0,049 & -0,555 & 0,206 \\ -0,714 & 0,049 & 1 & 0,213 & -0,610 \\ -0,776 & -0,555 & 0,213 & 1 & 0,339 \\ 0,310 & 0,206 & -0,610 & 0,339 & 1 \end{pmatrix}$$

$S_{101}^{H} =$

$$\begin{pmatrix} 1 & -0,734 & -0,286 & -0,117 & 0,573 & -0,321 & 0,111 & 0,343 & 0,044 & 0,044 \\ -0,734 & 1 & 0,670 & 0,272 & -0,166 & 0,671 & 0,245 & 0,318 & 0,291 & 0,291 \\ -0,286 & 0,670 & 1 & 0,410 & 0,000 & 0,256 & -0,272 & 0,393 & 0,456 & 0,456 \\ -0,117 & 0,272 & 0,410 & 1 & 0,327 & 0,352 & 0,064 & 0,387 & 0,529 & 0,529 \\ 0,573 & -0,166 & 0,000 & 0,327 & 1 & 0,518 & 0,365 & 0,811 & -0,070 & -0,070 \\ -0,321 & 0,671 & 0,256 & 0,352 & 0,518 & 1 & 0,613 & 0,700 & 0,003 & 0,003 \\ 0,111 & 0,245 & -0,272 & 0,064 & 0,365 & 0,613 & 1 & 0,562 & 0,213 & 0,213 \\ 0,343 & 0,318 & 0,393 & 0,387 & 0,811 & 0,700 & 0,562 & 1 & 0,310 & 0,310 \\ 0,044 & 0,291 & 0,456 & 0,529 & -0,070 & 0,003 & 0,213 & 0,310 & 1 & 0,735 \\ 0,044 & 0,291 & 0,456 & 0,529 & -0,070 & 0,003 & 0,213 & 0,310 & 0,735 & 1 \end{pmatrix}$$

A.6 Problème des minima locaux dans le cas de l'algorithme avec contraintes

Les matrices obtenues ci-dessous permettent d'illustrer les problèmes de minima locaux obtenus avec l'algorithme de PSDisation avec encadrement (algorithme 2.10) :

$$S_{31}^{C} = \begin{pmatrix} 1 & -0,800 & -0,400 \\ -0,800 & 1 & -0,230 \\ -0,400 & -0,230 & 1 \end{pmatrix}, S_{41}^{C} = \begin{pmatrix} 1 & 0,802 & -0,400 & -0,501 \\ 0,802 & 1 & 0,021 & 0,001 \\ -0,400 & 0,021 & 1 & 0,197 \\ -0,501 & 0,001 & 0,197 & 1 \end{pmatrix}$$

$$S_{51}^{C} = \begin{pmatrix} 1 & -0,528 & -0,781 & -0,622 & 0,067 \\ -0,528 & 1 & 0,801 & -0,335 & 0,210 \\ -0,781 & 0,801 & 1 & 0,010 & -0,326 \\ -0,622 & -0,335 & 0,010 & 1 & -0,201 \\ 0,067 & 0,210 & -0,326 & -0,201 & 1 \end{pmatrix}$$

A.7 Test de la stabilité de la simulation du SCR avec un générateur de nombres aléatoires de Mersenne

L'objectif de cette section est de montrer les résultats de différentes simulations du calcul du SCR dans plusieurs cas (voir section 3). Tous les chiffres suivants sont calculés avec 32768 tirages, un calcul du SCR comme moyenne de 4 tirages répétés (voir équation 10), et un générateur de nombres aléatoires de Mersenne.

A.7.1 Cas d'une copule gaussienne et de facteurs de pertes (Mersenne)

Cas A.1. *Avec une copule gaussienne, la matrice de corrélation S_{31}^H et le vecteur aléatoire X_{III} (cas de facteurs de pertes), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*2557,637*	*2549,221*	*2549,722*	*2549,712*	*2551,858*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*2552,376*	*2553,433*	*2552,754*	*2552,190*	*2552,665*
$\widehat{\sigma}$	*0,09%*				

Cas A.2. *Avec une copule gaussienne, la matrice de corrélation S_{32}^H, et le vecteur aléatoire X_{III} (cas de facteurs de pertes), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*2695,369*	*2680,462*	*2674,988*	*2684,170*	*2676,699*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*2683,950*	*2694,912*	*2685,042*	*2689,176*	*2681,128*
$\widehat{\sigma}$	*0,24%*				

Cas A.3. *Avec une copule gaussienne, la matrice de corrélation S_{41}^H et le vecteur aléatoire X_{IV} (cas de facteurs de pertes), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*3381,565*	*3376,039*	*3376,572*	*3390,494*	*3365,265*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*3372,681*	*3351,733*	*3363,568*	*3376,566*	*3373,964*
$\widehat{\sigma}$	*0,30%*				

Cas A.4. *Avec une copule gaussienne, la matrice de corrélation S_{42}^H, et le vecteur aléatoire X_{IV} (cas de facteurs de pertes), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*2839,677*	*2841,983*	*2850,508*	*2853,123*	*2854,843*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*2843,823*	*2857,203*	*2866,569*	*2855,908*	*2845,855*
$\widehat{\sigma}$	*0,27%*				

Cas A.5. *Avec une copule gaussienne, la matrice de corrélation S_{51}^H et le vecteur aléatoire X_V (cas de facteurs de pertes), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*2623,786*	*2631,848*	*2628,463*	*2636,843*	*2639,505*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*2629,949*	*2633,930*	*2635,054*	*2631,067*	*2630,630*
$\widehat{\sigma}$	*0,16%*				

Cas A.6. *Avec une copule gaussienne, la matrice de corrélation S_{52}^H, et le vecteur aléatoire X_V (cas de facteurs de pertes), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*3174,707*	*3169,502*	*3186,008*	*3168,038*	*3161,862*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*3166,071*	*3165,342*	*3155,591*	*3169,788*	*3159,680*
$\widehat{\sigma}$	*0,25%*				

Cas A.7. *Avec une copule gaussienne, la matrice de corrélation S_{101}^H et le vecteur aléatoire X_X (cas de facteurs de pertes), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*5852,700*	*5819,220*	*5861,433*	*5862,688*	*5811,300*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*5839,532*	*5863,868*	*5843,832*	*5846,574*	*5879,442*
$\widehat{\sigma}$	*0,34%*				

A.7.2 Cas d'une copule de Student et de facteurs de pertes (Mersenne)

Cas A.8. *Avec une copule de Student, la matrice de corrélation* S_{31}^{H} *et le vecteur aléatoire* X_{III} *(cas de facteurs de pertes), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*2639,489*	*2628,388*	*2633,545*	*2623,164*	*2634,899*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*2624,184*	*2622,280*	*2623,339*	*2629,817*	*2623,361*
$\widehat{\sigma}$	*0,22%*				

Cas A.9. *Avec une copule de Student, la matrice de corrélation* S_{32}^{H}*, et le vecteur aléatoire* X_{III} *(cas de facteurs de pertes), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*2828.449*	*2851.922*	*2838.048*	*2852.348*	*2857.174*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*2843.416*	*2855.228*	*2844.018*	*2859.524*	*2838.013*
$\widehat{\sigma}$	*0,33%*				

Cas A.10. *Avec une copule de Student, la matrice de corrélation* S_{41}^{H} *et le vecteur aléatoire* X_{IV} *(cas de facteurs de pertes), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*3376,523*	*3351,283*	*3382,232*	*3374,002*	*3369,996*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*3365,377*	*3379,148*	*3351,038*	*3415,119*	*3369,776*
$\widehat{\sigma}$	*0,51%*				

Cas A.11. *Avec une copule de Student, la matrice de corrélation* S_{42}^{H}*, et le vecteur aléatoire* X_{IV} *(cas de facteurs de pertes), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*2988,592*	*2989,214*	*2985,350*	*2996,761*	*2985,418*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*3002,304*	*3005,437*	*3016,524*	*2966,953*	*2986,457*
$\widehat{\sigma}$	*0,43%*				

Cas A.12. *Avec une copule de Student, la matrice de corrélation S_{51}^H et le vecteur aléatoire X_V (cas de facteurs de pertes), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*2694,418*	*2692,356*	*2703,021*	*2685,915*	*2699,194*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*2708,711*	*2687,560*	*2700,239*	*2691,793*	*2691,507*
$\widehat{\sigma}$	*0,25%*				

Cas A.13. *Avec une copule de Student, la matrice de corrélation S_{52}^H, et le vecteur aléatoire X_V (cas de facteurs de pertes), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*3268,688*	*3293,614*	*3291,226*	*3299,022*	*3283,987*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*3282,746*	*3299,090*	*3289,102*	*3270,211*	*3285,327*
$\widehat{\sigma}$	*0,30%*				

Cas A.14. *Avec une copule de Student, la matrice de corrélation S_{101}^H et le vecteur aléatoire X_X (cas de facteurs de pertes), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*6733,744*	*6695,332*	*6704,649*	*6727,365*	*6707,386*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*6713,511*	*6718,506*	*6773,358*	*6727,674*	*6670,891*
$\widehat{\sigma}$	*0,38%*				

A.7.3 Cas d'une copule gaussienne et de facteurs de risques (Mersenne)

Cas A.15. *Avec une copule gaussienne, la matrice de corrélation S_{31}^H et le vecteur aléatoire X_{IIIb} (cas de facteurs de risques), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*11680627*	*11659646*	*11700655*	*11721160*	*11756991*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*11702598*	*11685595*	*11651260*	*11724827*	*11704996*
$\widehat{\sigma}$	*0,26%*				

Cas A.16. *Avec une copule gaussienne, la matrice de corrélation S_{32}^H, et le vecteur aléatoire X_{IIIb} (cas de facteurs de risques), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*13550628*	*13402836*	*13446224*	*13681863*	*13462006*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*13447101*	*13567149*	*13506136*	*13642674*	*13545137*
$\widehat{\sigma}$	*0,63%*				

Cas A.17. *Avec une copule gaussienne, la matrice de corrélation S_{41}^H et le vecteur aléatoire X_{IVb} (cas de facteurs de risques), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*3712280*	*3700939*	*3713618*	*3714385*	*3714770*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*3701429*	*3711803*	*3675144*	*3718875*	*3679241*
$\widehat{\sigma}$	*0,39%*				

Cas A.18. *Avec une copule gaussienne, la matrice de corrélation S_{42}^H, et le vecteur aléatoire X_{IVb} (cas de facteurs de risques), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*3736667*	*3734244*	*3739559*	*3737599*	*3743335*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*3732951*	*3740026*	*3732363*	*3743215*	*3751692*
$\widehat{\sigma}$	*0,15%*				

Cas A.19. *Avec une copule gaussienne, la matrice de corrélation S_{51}^H et le vecteur aléatoire X_{Vb} (cas de facteurs de risques), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*10047,889*	*10062,587*	*10055,736*	*10087,915*	*9997,026*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*10062,734*	*10186,984*	*10043,865*	*9947,782*	*10202,554*
$\widehat{\sigma}$	*0,73%*				

Cas A.20. *Avec une copule gaussienne, la matrice de corrélation S_{52}^H, et le vecteur aléatoire X_{Vb} (cas de facteurs de risques), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*9157,407*	*9132,778*	*9135,022*	*9050,715*	*9091,203*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*9076,050*	*9084,894*	*9100,913*	*9068,398*	*9126,265*
$\widehat{\sigma}$	*0,36%*				

Cas A.21. *Avec une copule gaussienne, la matrice de corrélation* S^H_{101} *et le vecteur aléatoire* X_{Xb} *(cas de facteurs de risques), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*4256631*	*4300993*	*4277165*	*4176858*	*4275264*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*4273748*	*4235778*	*4253672*	*4227811*	*4221247*
$\widehat{\sigma}$	*0,80%*				

A.7.4 Cas d'une copule de Student et de facteurs de risques (Mersenne)

Cas A.22. *Avec une copule de Student, la matrice de corrélation* S^H_{31} *et le vecteur aléatoire* X_{IIIb} *(cas de facteurs de risques), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*12823707*	*12593934*	*12648260*	*12658250*	*12652118*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*12658708*	*12644894*	*12785277*	*12721624*	*12747215*
$\widehat{\sigma}$	*0,54%*				

Cas A.23. *Avec une copule de Student, la matrice de corrélation* S^H_{32}*, et le vecteur aléatoire* X_{IIIb} *(cas de facteurs de risques), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*14921613*	*14885242*	*15032997*	*15067289*	*14808685*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*14741574*	*15111194*	*14965739*	*14749696*	*14890094*
$\widehat{\sigma}$	*0,82%*				

Cas A.24. *Avec une copule de Student, la matrice de corrélation* S^H_{41} *et le vecteur aléatoire* X_{IVb} *(cas de facteurs de risques), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*3659251*	*3652228*	*3658376*	*3660083*	*3662670*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*3650573*	*3676024*	*3641355*	*3672273*	*3640851*
$\widehat{\sigma}$	*0,30%*				

Cas A.25. *Avec une copule de Student, la matrice de corrélation* S_{42}^{H}*, et le vecteur aléatoire* X_{IVb} *(cas de facteurs de risques), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*3649236*	*3651283*	*3655315*	*3643951*	*3648499*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*3665510*	*3653840*	*3630430*	*3664706*	*3642775*
$\widehat{\sigma}$	*0,27%*				

Cas A.26. *Avec une copule de Student, la matrice de corrélation* S_{51}^{H} *et le vecteur aléatoire* X_{Vb} *(cas de facteurs de risques), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*8710,538*	*8818,717*	*8744,018*	*8604,836*	*8753,493*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*8629,639*	*8666,959*	*8823,890*	*8680,887*	*8730,539*
$\widehat{\sigma}$	*0,80%*				

Cas A.27. *Avec une copule de Student, la matrice de corrélation* S_{52}^{H}*, et le vecteur aléatoire* X_{Vb} *(cas de facteurs de risques), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*7867,636*	*8178,273*	*8033,017*	*7960,792*	*8098,870*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*8023,719*	*7877,464*	*8010,706*	*8034,642*	*8045,276*
$\widehat{\sigma}$	*1,11%*				

Cas A.28. *Avec une copule de Student, la matrice de corrélation* S_{101}^{H} *et le vecteur aléatoire* X_{Xb} *(cas de facteurs de risques), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*4656914*	*4705820*	*4683908*	*4707367*	*4725976*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*4752452*	*4684354*	*4763138*	*4733783*	*4655332*
$\widehat{\sigma}$	*0,75%*				

A.8 Test de la stabilité de la simulation du SCR avec un générateur de nombres aléatoires de Sobol

L'objectif de cette section est de montrer les résultats de différentes simulations du calcul du SCR dans plusieurs cas (voir section 3). Tous les chiffres suivants sont calculés avec 32768 tirages, un calcul du SCR comme moyenne de 4 tirages répétés (voir équation 10), et un générateur de nombres aléatoires de Sobol.

A.8.1 Cas d'une copule gaussienne et de facteurs de pertes (Sobol)

Cas A.29. *Avec une copule gaussienne, la matrice de corrélation S_{31}^H et le vecteur aléatoire X_{III} (cas de facteurs de pertes), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*2565,379*	*2564,802*	*2566,630*	*2565,653*	*2568,714*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*2564,247*	*2567,879*	*2565,654*	*2563,555*	*2567,912*
$\widehat{\sigma}$	*0,06%*				

Cas A.30. *Avec une copule gaussienne, la matrice de corrélation S_{32}^H, et le vecteur aléatoire X_{III} (cas de facteurs de pertes), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*2695,138*	*2691,238*	*2693,338*	*2690,833*	*2695,587*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*2693,460*	*2693,434*	*2692,304*	*2698,252*	*2695,224*
$\widehat{\sigma}$	*0,08%*				

Cas A.31. *Avec une copule gaussienne, la matrice de corrélation S_{41}^H et le vecteur aléatoire X_{IV} (cas de facteurs de pertes), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*3319,530*	*3317,546*	*3323,359*	*3325,865*	*3315,343*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*3319,010*	*3313,739*	*3319,054*	*3317,254*	*3318,951*
$\widehat{\sigma}$	*0,10%*				

Cas A.32. *Avec une copule gaussienne, la matrice de corrélation S_{42}^H, et le vecteur aléatoire X_{IV} (cas de facteurs de pertes), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*2805,754*	*2803,571*	*2816,311*	*2807,605*	*2802,711*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*2809,330*	*2810,695*	*2807,530*	*2808,719*	*2813,868*
$\widehat{\sigma}$	*0,14%*				

Cas A.33. *Avec une copule gaussienne, la matrice de corrélation S_{51}^H et le vecteur aléatoire X_V (cas de facteurs de pertes), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*2607,746*	*2606,772*	*2602,754*	*2602,007*	*2609,491*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*2602,844*	*2603,220*	*2605,669*	*2605,289*	*2603,931*
$\widehat{\sigma}$	*0,09%*				

Cas A.34. *Avec une copule gaussienne, la matrice de corrélation S_{52}^H, et le vecteur aléatoire X_V (cas de facteurs de pertes), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*3132,310*	*3134,633*	*3135,268*	*3139,569*	*3137,111*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*3145,796*	*3142,966*	*3140,471*	*3139,920*	*3134,543*
$\widehat{\sigma}$	*0,13%*				

Cas A.35. *Avec une copule gaussienne, la matrice de corrélation S_{101}^H et le vecteur aléatoire X_X (cas de facteurs de pertes), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*5825.113*	*5824.996*	*5828.068*	*5833.347*	*5826.476*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*5845.889*	*5828.170*	*5827.728*	*5831.393*	*5828.625*
$\widehat{\sigma}$	*0,10%*				

A.8.2 Cas d'une copule de Student et de facteurs de pertes (Sobol)

Cas A.36. *Avec une copule de Student, la matrice de corrélation S_{31}^H et le vecteur aléatoire X_{III} (cas de facteurs de pertes), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*2575,835*	*2571,828*	*2576,098*	*2576,829*	*2575,682*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*2576,712*	*2573,595*	*2570,008*	*2578,894*	*2579,832*
$\widehat{\sigma}$	*0,11%*				

Cas A.37. *Avec une copule de Student, la matrice de corrélation S_{32}^H, et le vecteur aléatoire X_{III} (cas de facteurs de pertes), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*2807,059*	*2806,073*	*2801,715*	*2818,420*	*2805,969*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*2816,841*	*2806,291*	*2814,504*	*2802,457*	*2812,953*
$\widehat{\sigma}$	*0,20%*				

Cas A.38. *Avec une copule de Student, la matrice de corrélation S_{41}^H et le vecteur aléatoire X_{IV} (cas de facteurs de pertes), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*3379,131*	*3375,076*	*3381,567*	*3374,727*	*3389,255*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*3373,700*	*3371,637*	*3373,049*	*3376,895*	*3383,676*
$\widehat{\sigma}$	*0,16%*				

Cas A.39. *Avec une copule de Student, la matrice de corrélation S_{42}^H, et le vecteur aléatoire X_{IV} (cas de facteurs de pertes), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*2999,033*	*2994,696*	*2989,813*	*2988,709*	*3009,094*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*2987,187*	*3008,903*	*2986,686*	*3006,533*	*2988,720*
$\widehat{\sigma}$	*0,29%*				

Cas A.40. *Avec une copule de Student, la matrice de corrélation S_{51}^H et le vecteur aléatoire X_V (cas de facteurs de pertes), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*2669,309*	*2677,280*	*2670,282*	*2679,502*	*2671,161*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*2670,527*	*2677,807*	*2682,871*	*2676,855*	*2668,105*
$\widehat{\sigma}$	*0,18%*				

Cas A.41. *Avec une copule de Student, la matrice de corrélation S_{52}^H, et le vecteur aléatoire X_V (cas de facteurs de pertes), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*3261,020*	*3263,028*	*3264,608*	*3249,922*	*3259,357*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*3269,826*	*3252,369*	*3260,237*	*3241,949*	*3281,678*
$\widehat{\sigma}$	*0,32%*				

Cas A.42. *Avec une copule de Student, la matrice de corrélation S_{101}^H et le vecteur aléatoire X_X (cas de facteurs de pertes), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*6752,426*	*6731,193*	*6727,590*	*6767,236*	*6718,165*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*6757,712*	*6720,474*	*6719,525*	*6703,919*	*6717,530*
$\widehat{\sigma}$	*0,29%*				

A.8.3 Cas d'une copule gaussienne et de facteurs de risques (Sobol)

Cas A.43. *Avec une copule gaussienne, la matrice de corrélation S_{31}^H et le vecteur aléatoire X_{IIIb} (cas de facteurs de risques), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*12137226*	*12146959*	*12119354*	*12140340*	*12130864*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*12095677*	*12136256*	*12124984*	*12084653*	*12124868*
$\widehat{\sigma}$	*0,16%*				

Cas A.44. *Avec une copule gaussienne, la matrice de corrélation S_{32}^H, et le vecteur aléatoire X_{IIIb} (cas de facteurs de risques), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*13836469*	*13782049*	*13764322*	*13792507*	*13837308*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*13797267*	*13823492*	*13834278*	*13846354*	*13808174*
$\widehat{\sigma}$	*0,19%*				

Cas A.45. *Avec une copule gaussienne, la matrice de corrélation S_{41}^H et le vecteur aléatoire X_{IVb} (cas de facteurs de risques), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*3537185*	*3536431*	*3532464*	*3527741*	*3530110*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*3532563*	*3529727*	*3528026*	*3529547*	*3536223*
$\widehat{\sigma}$	*0,10%*				

Cas A.46. *Avec une copule gaussienne, la matrice de corrélation S_{42}^H, et le vecteur aléatoire X_{IVb} (cas de facteurs de risques), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*3559408*	*3550435*	*3561652*	*3576282*	*3555246*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*3580639*	*3579402*	*3580684*	*3563414*	*3577709*
$\widehat{\sigma}$	*0,31%*				

Cas A.47. *Avec une copule gaussienne, la matrice de corrélation S_{51}^H et le vecteur aléatoire X_{Vb} (cas de facteurs de risques), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*9296,852*	*9390,663*	*9337,779*	*9358,991*	*9484,868*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*9274,548*	*9344,234*	*9427,308*	*9386,335*	*9458,495*
$\widehat{\sigma}$	*0,68%*				

Cas A.48. *Avec une copule gaussienne, la matrice de corrélation S_{52}^H, et le vecteur aléatoire X_{Vb} (cas de facteurs de risques), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*8302,769*	*8277,269*	*8319,585*	*8322,120*	*8271,515*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*8327,549*	*8292,115*	*8271,857*	*8311,247*	*8259,983*
$\widehat{\sigma}$	*0,28%*				

Cas A.49. *Avec une copule gaussienne, la matrice de corrélation* S^H_{101} *et le vecteur aléatoire* X_{Xb} *(cas de facteurs de risques), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*4154699*	*4168607*	*4249724*	*4205292*	*4177377*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*4183560*	*4193940*	*4162373*	*4207178*	*4205619*
$\widehat{\sigma}$	*0,63%*				

A.8.4 Cas d'une copule de Student et de facteurs de risques (Sobol)

Cas A.50. *Avec une copule de Student, la matrice de corrélation* S^H_{31} *et le vecteur aléatoire* X_{IIIb} *(cas de facteurs de risques), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*12515095*	*12520866*	*12520480*	*12456886*	*12385371*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*12502990*	*12506723*	*12511055*	*12482894*	*12473295*
$\widehat{\sigma}$	*0,32%*				

Cas A.51. *Avec une copule de Student, la matrice de corrélation* S^H_{32}*, et le vecteur aléatoire* X_{IIIb} *(cas de facteurs de risques), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*14805282*	*14616655*	*14790069*	*14731247*	*14609093*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*14520079*	*14864021*	*14534222*	*14472087*	*14641261*
$\widehat{\sigma}$	*0,86%*				

Cas A.52. *Avec une copule de Student, la matrice de corrélation* S^H_{41} *et le vecteur aléatoire* X_{IVb} *(cas de facteurs de risques), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*3534568*	*3545427*	*3559742*	*3551746*	*3538232*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*3560331*	*3531344*	*3532617*	*3525870*	*3553384*
$\widehat{\sigma}$	*0,33%*				

Cas A.53. *Avec une copule de Student, la matrice de corrélation S_{42}^H, et le vecteur aléatoire X_{IVb} (cas de facteurs de risques), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*3575878*	*3544984*	*3563328*	*3571786*	*3559739*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*3547176*	*3562081*	*3558125*	*3554434*	*3588290*
$\widehat{\sigma}$	*0,35%*				

Cas A.54. *Avec une copule de Student, la matrice de corrélation S_{51}^H et le vecteur aléatoire X_{Vb} (cas de facteurs de risques), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*9524,135*	*9410,141*	*9369,674*	*9374,792*	*9522,308*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*9462,634*	*9411,329*	*9433,945*	*9396,633*	*9362,759*
$\widehat{\sigma}$	*0,59%*				

Cas A.55. *Avec une copule de Student, la matrice de corrélation S_{52}^H, et le vecteur aléatoire X_{Vb} (cas de facteurs de risques), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*8611,025*	*8515,357*	*8419,983*	*8725,143*	*8603,455*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*8481,669*	*8593,835*	*8484,237*	*8531,153*	*8627,808*
$\widehat{\sigma}$	*0,99%*				

Cas A.56. *Avec une copule de Student, la matrice de corrélation S_{101}^H et le vecteur aléatoire X_{Xb} (cas de facteurs de risques), on obtient :*

Tirage	*1*	*2*	*3*	*4*	*5*
SCR	*4246555*	*4165044*	*4262772*	*4206620*	*4225241*

Tirage	*6*	*7*	*8*	*9*	*10*
SCR	*4190035*	*4212289*	*4244048*	*4283079*	*4268568*
$\widehat{\sigma}$	*0,84%*				

A.9 Résultats d'optimisation obtenus et sensibilités

A.9.1 Hypothèses de travail pour la suite

Suite aux tests de stabilités effectués ci-dessus, les hypothèses suivantes seront conservées pour la suite du mémoire :

— un nombre de 32768 tirages pour chaque calcul de SCR ;
— un calcul du SCR comme moyenne de 4 tirages répétés (voir équation 10) ;
— un générateur de nombres aléatoires de Sobol.

A.9.2 Rappels sur les critères d'évaluation d'impact

L'intervalle de confiance autour du SCR simulé est défini par : $P(|Z-m| < 1,65\,\widehat{\sigma}) = 90\%$ avec $\widehat{\sigma}$ l'écart-type estimé à l'aide de tirages successifs. Pour chacune des matrices, et des marginales utilisées, on dira qu'une déformation est *faible* si l'impact observé sur le SCR est inférieur à $2 \times 1,65 \times \widehat{\sigma}$, et *forte* dans le cas contraire. Par ailleurs, on dira qu'une déformation est *très forte* si l'impact observé sur le SCR est supérieur à $2 \times 2,89 \times \widehat{\sigma}$.

A.9.3 Cas d'une copule gaussienne et de facteurs de pertes

Cas A.57. *Avec la matrice de pseudocorrélation G_{31}, sa matrice de pondération H_{31}, le vecteur aléatoire X_{III} (cas des facteurs de pertes), l'application de l'algorithme de permutation 4.1 mène à un intervalle de* ***0,40%*** *autour du SCR (min = 2562,644 et max = 2572,78). La déformation est donc* ***très forte*** *(écart-type estimé à 0,06%).*

Cas A.58. *Avec la matrice de pseudocorrélation G_{31}, sa matrice de pondération H_{31}, et l'encadrement $H_{min} = 0$ et $H_{max} = 1$, le vecteur aléatoire X_{III} (cas des facteurs de pertes), l'application de l'algorithme d'optimisation de la pondération 4.2 mène à un intervalle de* ***0,53%*** *autour du SCR (min = 2564,11 et max = 2577,688). La déformation est donc* ***très forte*** *(écart-type estimé à 0,06%).*

Cas A.59. *Avec la matrice de pseudocorrélation G_{31}, sa matrice de pondération H_{31}, et l'encadrement $H_{min} = H_{31} - 0,1$ et $H_{max} = H_{31} + 0,1$, le vecteur aléatoire X_{III} (cas des facteurs de pertes), l'application de l'algorithme d'optimisation de la pondération 4.2 mène à un intervalle de* ***0,23%*** *autour du SCR (min = 2560,194 et max = 2565,962). La déformation est donc* ***forte*** *(écart-type estimé à 0,06%).*

Cas A.60. *Avec la matrice de pseudocorrélation G_{31}, sa matrice de pondération H_{31}, le vecteur aléatoire X_{III} (cas des facteurs de pertes), l'ajout d'une*

dimension vide (voir section 4.2.1) mène à un intervalle de ***0,99%*** *autour du SCR (après modification = 2593,969 et initial = 2568,932). La déformation est donc* ***très forte*** *(écart-type estimé à 0,06%).*

Cas A.61. *Avec la matrice de pseudocorrélation G_{32}, sa matrice de pondération H_{32}, le vecteur aléatoire X_{III} (cas des facteurs de pertes), l'application de l'algorithme de permutation 4.1 mène à un intervalle de* ***4,66%*** *autour du SCR (min = 2635,26 et max = 2758,19). La déformation est donc* ***très forte*** *(écart-type estimé à 0,08%).*

Cas A.62. *Avec la matrice de pseudocorrélation G_{32}, sa matrice de pondération H_{32}, et l'encadrement $H_{min} = 0$ et $H_{max} = 1$, le vecteur aléatoire X_{III} (cas des facteurs de pertes), l'application de l'algorithme d'optimisation de la pondération 4.2 mène à un intervalle de* ***1,44%*** *autour du SCR (min = 2627,977 et max = 2665,767). La déformation est donc* ***très forte*** *(écart-type estimé à 0,08%).*

Cas A.63. *Avec la matrice de pseudocorrélation G_{32}, sa matrice de pondération H_{32}, et l'encadrement $H_{min} = H_{32} - 0,1$ et $H_{max} = H_{32} + 0,1$, le vecteur aléatoire X_{III} (cas des facteurs de pertes), l'application de l'algorithme d'optimisation de la pondération 4.2 mène à un intervalle de* ***0,68%*** *autour du SCR (min = 2675,383 et max = 2693,569). La déformation est donc* ***très forte*** *(écart-type estimé à 0,08%).*

Cas A.64. *Avec la matrice de pseudocorrélation G_{32}, sa matrice de pondération H_{32}, le vecteur aléatoire X_{III} (cas des facteurs de pertes), l'ajout d'une dimension vide (voir section 4.2.1) mène à un intervalle de* ***0,55%*** *autour du SCR (min = 2709,628 et max = 2694,791). La déformation est donc* ***très forte*** *(écart-type estimé à 0,08%).*

Cas A.65. *Avec la matrice de pseudocorrélation G_{41}, sa matrice de pondération H_{41}, le vecteur aléatoire X_{IV} (cas des facteurs de pertes), l'application de l'algorithme de permutation 4.1 mène à un intervalle de* ***1,93%*** *autour du SCR (min = 3293,269 et max = 3356,767). La déformation est donc* ***très forte*** *(écart-type estimé à 0,10%).*

Cas A.66. *Avec la matrice de pseudocorrélation G_{41}, sa matrice de pondération H_{41}, et l'encadrement $H_{min} = 0$ et $H_{max} = 1$, le vecteur aléatoire X_{IV} (cas des facteurs de pertes), l'application de l'algorithme d'optimisation de la pondération 4.2 mène à un intervalle de* ***3,56%*** *autour du SCR (min = 3121,059 et max = 3232,057). La déformation est donc* ***très forte*** *(écart-type estimé à 0,10%).*

Cas A.67. *Avec la matrice de pseudocorrélation G_{41}, sa matrice de pondération H_{41}, et l'encadrement $H_{min} = H_{41} - 0,1$ et $H_{max} = H_{41} + 0,1$, le vecteur aléatoire X_{IV} (cas des facteurs de pertes), l'application de l'algorithme d'optimisation de la pondération 4.2 mène à un intervalle de* ***0,69%*** *autour du SCR*

(min = 3297,227 et max = 3320,037). La déformation est donc ***très forte*** *(écart-type estimé à 0,10%).*

Cas A.68. *Avec la matrice de pseudocorrélation G_{41}, sa matrice de pondération H_{41}, le vecteur aléatoire X_{IV} (cas des facteurs de pertes), l'ajout d'une dimension vide (voir section 4.2.1) mène à un intervalle de* ***0,17%*** *autour du SCR (min = 3228,748 et max = 3323,747). La déformation est donc* ***faible*** *(écart-type estimé à 0,10%).*

Cas A.69. *Avec la matrice de pseudocorrélation G_{42}, sa matrice de pondération H_{42}, le vecteur aléatoire X_{IV} (cas des facteurs de pertes), l'application de l'algorithme de permutation 4.1 mène à un intervalle de* ***2,37%*** *autour du SCR (min = 2759,38 et max = 2824,889). La déformation est donc* ***très forte*** *(écart-type estimé à 0,14%).*

Cas A.70. *Avec la matrice de pseudocorrélation G_{42}, sa matrice de pondération H_{42}, et l'encadrement $H_{min} = 0$ et $H_{max} = 1$, le vecteur aléatoire X_{IV} (cas des facteurs de pertes), l'application de l'algorithme d'optimisation de la pondération 4.2 mène à un intervalle de* ***3,25%*** *autour du SCR (min = 2814,421 et max = 2905,759). La déformation est donc* ***très forte*** *(écart-type estimé à 0,14%).*

Cas A.71. *Avec la matrice de pseudocorrélation G_{42}, sa matrice de pondération H_{42}, et l'encadrement $H_{min} = H_{42} - 0,1$ et $H_{max} = H_{42} + 0,1$, le vecteur aléatoire X_{IV} (cas des facteurs de pertes), l'application de l'algorithme d'optimisation de la pondération 4.2 mène à un intervalle de* ***0,90%*** *autour du SCR (min = 2803,703 et max = 2828,836). La déformation est donc* ***très forte*** *(écart-type estimé à 0,14%).*

Cas A.72. *Avec la matrice de pseudocorrélation G_{42}, sa matrice de pondération H_{42}, le vecteur aléatoire X_{IV} (cas des facteurs de pertes), l'ajout d'une dimension vide (voir section 4.2.1) mène à un intervalle de* ***0,47%*** *autour du SCR (après modification = 2801,376 et initial = 2814,497). La déformation est donc* ***forte*** *(écart-type estimé à 0,14%).*

Cas A.73. *Avec la matrice de pseudocorrélation G_{51}, sa matrice de pondération H_{51}, le vecteur aléatoire X_V (cas des facteurs de pertes), l'application de l'algorithme de permutation 4.1 mène à un intervalle de* ***0,75%*** *autour du SCR (min = 2607,652 et max = 2627,147). La déformation est donc* ***très forte*** *(écart-type estimé à 0,09%).*

Cas A.74. *Avec la matrice de pseudocorrélation G_{51}, sa matrice de pondération H_{51}, et l'encadrement $H_{min} = 0$ et $H_{max} = 1$, le vecteur aléatoire X_V (cas des facteurs de pertes), l'application de l'algorithme d'optimisation de la pondération 4.2 mène à un intervalle de* ***0,94%*** *autour du SCR (min = 2637,397 et max = 2662,247). La déformation est donc* ***très forte*** *(écart-type estimé à 0,09%).*

Cas A.75. *Avec la matrice de pseudocorrélation G_{51}, sa matrice de pondération H_{51}, et l'encadrement $H_{min} = H_{51} - 0,1$ et $H_{max} = H_{51} + 0,1$, le vecteur aléatoire X_V (cas des facteurs de pertes), l'application de l'algorithme d'optimisation de la pondération 4.2 mène à un intervalle de* ***0,34%*** *autour du SCR (min = 2607,383 et max = 2616,209). La déformation est donc* ***forte*** *(écart-type estimé à 0,09%).*

Cas A.76. *Avec la matrice de pseudocorrélation G_{51}, sa matrice de pondération H_{51}, le vecteur aléatoire X_V (cas des facteurs de pertes), l'ajout d'une dimension vide (voir section 4.2.1) mène à un intervalle de* ***0,42%*** *autour du SCR (après modification = 2626,374 et initial = 2615,5). La déformation est donc* ***forte*** *(écart-type estimé à 0,09%).*

Cas A.77. *Avec la matrice de pseudocorrélation G_{52}, sa matrice de pondération H_{52}, le vecteur aléatoire X_V (cas des facteurs de pertes), l'application de l'algorithme de permutation 4.1 mène à un intervalle de* ***6,76%*** *autour du SCR (min = 2952,791 et max = 3152,476). La déformation est donc* ***très forte*** *(écart-type estimé à 0,13%).*

Cas A.78. *Avec la matrice de pseudocorrélation G_{52}, sa matrice de pondération H_{52}, et l'encadrement $H_{min} = 0$ et $H_{max} = 1$, le vecteur aléatoire X_V (cas des facteurs de pertes), l'application de l'algorithme d'optimisation de la pondération 4.2 mène à un intervalle de* ***5,45%*** *autour du SCR (min = 2929,411 et max = 3088,922). La déformation est donc* ***très forte*** *(écart-type estimé à 0,13%).*

Cas A.79. *Avec la matrice de pseudocorrélation G_{52}, sa matrice de pondération H_{52}, et l'encadrement $H_{min} = H_{52} - 0,1$ et $H_{max} = H_{52} + 0,1$, le vecteur aléatoire X_V (cas des facteurs de pertes), l'application de l'algorithme d'optimisation de la pondération 4.2 mène à un intervalle de* ***0,67%*** *autour du SCR (min = 3120,478 et max = 3141,468). La déformation est donc* ***forte*** *(écart-type estimé à 0,13%).*

Cas A.80. *Avec la matrice de pseudocorrélation G_{52}, sa matrice de pondération H_{52}, le vecteur aléatoire X_V (cas des facteurs de pertes), l'ajout d'une dimension vide (voir section 4.2.1) mène à un intervalle de* ***0,22%*** *autour du SCR (après modification = 3145,946 et initial = 3139,025). La déformation est donc* ***faible*** *(écart-type estimé à 0,13%).*

Cas A.81. *Avec la matrice de pseudocorrélation G_{101}, sa matrice de pondération H_{101}, le vecteur aléatoire X_X (cas des facteurs de pertes), l'application de l'algorithme de permutation 4.1 mène à un intervalle de* ***2,23%*** *autour du SCR (min = 5755,021 et max = 5883,115). La déformation est donc* ***très forte*** *(écart-type estimé à 0,10%).*

Cas A.82. *Avec la matrice de pseudocorrélation G_{101}, sa matrice de pondération H_{101}, et l'encadrement $H_{min} = 0$ et $H_{max} = 1$, le vecteur aléatoire*

*X_X (cas des facteurs de pertes), l'application de l'algorithme d'optimisation de la pondération 4.2 mène à un intervalle de **1,68%** autour du SCR (min = 5803,957 et max = 5901,507). La déformation est donc **très forte** (écart-type estimé à 0,10%).*

Cas A.83. *Avec la matrice de pseudocorrélation G_{101}, sa matrice de pondération H_{101}, et l'encadrement $H_{min} = H_{101} - 0,1$ et $H_{max} = H_{101} + 0,1$, le vecteur aléatoire X_X (cas des facteurs de pertes), l'application de l'algorithme d'optimisation de la pondération 4.2 mène à un intervalle de **0,82%** autour du SCR (min = 5786,699 et max = 5834,237). La déformation est donc **très forte** (écart-type estimé à 0,10%).*

Cas A.84. *Avec la matrice de pseudocorrélation G_{101}, sa matrice de pondération H_{101}, le vecteur aléatoire X_X (cas des facteurs de pertes), l'ajout d'une dimension vide (voir section 4.2.1) mène à un intervalle de **1,37%** autour du SCR (après modification = 5769,44 et initial = 5849,554). La déformation est donc **très forte** (écart-type estimé à 0,10%).*

A.9.4 Cas d'une copule de Student et de facteurs de pertes

Cas A.85. *Avec la matrice de pseudocorrélation G_{31}, sa matrice de pondération H_{31}, le vecteur aléatoire X_{III} (cas des facteurs de pertes), l'application de l'algorithme de permutation 4.1 mène à un intervalle de **0,95%** autour du SCR (min = 2565,318 et max = 2589,811). La déformation est donc **très forte** (écart-type estimé à 0,11%).*

Cas A.86. *Avec la matrice de pseudocorrélation G_{31}, sa matrice de pondération H_{31}, et l'encadrement $H_{min} = 0$ et $H_{max} = 1$, le vecteur aléatoire X_{III} (cas des facteurs de pertes), l'application de l'algorithme d'optimisation de la pondération 4.2 mène à un intervalle de **1,27%** autour du SCR (min = 2575,71 et max = 2608,432). La déformation est donc **très forte** (écart-type estimé à 0,11%).*

Cas A.87. *Avec la matrice de pseudocorrélation G_{31}, sa matrice de pondération H_{31}, et l'encadrement $H_{min} = H_{31} - 0,1$ et $H_{max} = H_{31} + 0,1$, le vecteur aléatoire X_{III} (cas des facteurs de pertes), l'application de l'algorithme d'optimisation de la pondération 4.2 mène à un intervalle de **0,41%** autour du SCR (min = 2565,049 et max = 2575,553). La déformation est donc **forte** (écart-type estimé à 0,11%).*

Cas A.88. *Avec la matrice de pseudocorrélation G_{31}, sa matrice de pondération H_{31}, le vecteur aléatoire X_{III} (cas des facteurs de pertes), l'ajout d'une dimension vide (voir section 4.2.1) mène à un intervalle de **0,08%** autour du SCR (après modification = 2578,867 et initial = 2576,769). La déformation est donc **faible** (écart-type estimé à 0,11%).*

Cas A.89. *Avec la matrice de pseudocorrélation G_{32}, sa matrice de pondération H_{32}, le vecteur aléatoire X_{III} (cas des facteurs de pertes), l'application de l'algorithme de permutation 4.1 mène à un intervalle de* ***6,09%*** *autour du SCR (min = 2726,016 et max = 2891,999). La déformation est donc* ***très forte*** *(écart-type estimé à 0,20%).*

Cas A.90. *Avec la matrice de pseudocorrélation G_{32}, sa matrice de pondération H_{32}, et l'encadrement $H_{min} = 0$ et $H_{max} = 1$, le vecteur aléatoire X_{III} (cas des facteurs de pertes), l'application de l'algorithme d'optimisation de la pondération 4.2 mène à un intervalle de* ***1,62%*** *autour du SCR (min = 2705,997 et max = 2749,83). La déformation est donc* ***très forte*** *(écart-type estimé à 0,20%).*

Cas A.91. *Avec la matrice de pseudocorrélation G_{32}, sa matrice de pondération H_{32}, et l'encadrement $H_{min} = H_{32} - 0,1$ et $H_{max} = H_{32} + 0,1$, le vecteur aléatoire X_{III} (cas des facteurs de pertes), l'application de l'algorithme d'optimisation de la pondération 4.2 mène à un intervalle de* ***0,70%*** *autour du SCR (min = 2784,256 et max = 2803,878). La déformation est donc* ***forte*** *(écart-type estimé à 0,20%).*

Cas A.92. *Avec la matrice de pseudocorrélation G_{32}, sa matrice de pondération H_{32}, le vecteur aléatoire X_{III} (cas des facteurs de pertes), l'ajout d'une dimension vide (voir section 4.2.1) mène à un intervalle de* ***0,12%*** *autour du SCR (après modification = 2805,6 et initial = 2802,372). La déformation est donc* ***faible*** *(écart-type estimé à 0,20%).*

Cas A.93. *Avec la matrice de pseudocorrélation G_{41}, sa matrice de pondération H_{41}, le vecteur aléatoire X_{IV} (cas des facteurs de pertes), l'application de l'algorithme de permutation 4.1 mène à un intervalle de* ***2,25%*** *autour du SCR (min = 3354,108 et max = 3429,738). La déformation est donc* ***très forte*** *(écart-type estimé à 0,16%).*

Cas A.94. *Avec la matrice de pseudocorrélation G_{41}, sa matrice de pondération H_{41}, et l'encadrement $H_{min} = 0$ et $H_{max} = 1$, le vecteur aléatoire X_{IV} (cas des facteurs de pertes), l'application de l'algorithme d'optimisation de la pondération 4.2 mène à un intervalle de* ***1,14%*** *autour du SCR (min = 3288,278 et max = 3325,722). La déformation est donc* ***très forte*** *(écart-type estimé à 0,16%).*

Cas A.95. *Avec la matrice de pseudocorrélation G_{41}, sa matrice de pondération H_{41}, et l'encadrement $H_{min} = H_{41} - 0,1$ et $H_{max} = H_{41} + 0,1$, le vecteur aléatoire X_{IV} (cas des facteurs de pertes), l'application de l'algorithme d'optimisation de la pondération 4.2 mène à un intervalle de* ***0,56%*** *autour du SCR (min = 3360,153 et max = 3378,962). La déformation est donc* ***forte*** *(écart-type estimé à 0,16%).*

Cas A.96. *Avec la matrice de pseudocorrélation G_{41}, sa matrice de pondération H_{41}, le vecteur aléatoire X_{IV} (cas des facteurs de pertes), l'ajout d'une dimension vide (voir section 4.2.1) mène à un intervalle de* ***6,19%*** *autour du SCR (après modification = 3572,287 et initial = 3364,128). La déformation est donc* ***très forte*** *(écart-type estimé à 0,16%).*

Cas A.97. *Avec la matrice de pseudocorrélation G_{42}, sa matrice de pondération H_{42}, le vecteur aléatoire X_{IV} (cas des facteurs de pertes), l'application de l'algorithme de permutation 4.1 mène à un intervalle de* ***3,50%*** *autour du SCR (min = 2909,999 et max = 3011,722). La déformation est donc* ***très forte*** *(écart-type estimé à 0,29%).*

Cas A.98. *Avec la matrice de pseudocorrélation G_{42}, sa matrice de pondération H_{42}, et l'encadrement $H_{min} = 0$ et $H_{max} = 1$, le vecteur aléatoire X_{IV} (cas des facteurs de pertes), l'application de l'algorithme d'optimisation de la pondération 4.2 mène à un intervalle de* ***2,19%*** *autour du SCR (min = 2995,144 et max = 3060,68). La déformation est donc* ***très forte*** *(écart-type estimé à 0,29%).*

Cas A.99. *Avec la matrice de pseudocorrélation G_{42}, sa matrice de pondération H_{42}, et l'encadrement $H_{min} = H_{42} - 0,1$ et $H_{max} = H_{42} + 0,1$, le vecteur aléatoire X_{IV} (cas des facteurs de pertes), l'application de l'algorithme d'optimisation de la pondération 4.2 mène à un intervalle de* ***2,14%*** *autour du SCR (min = 2933,728 et max = 2996,47). La déformation est donc* ***très forte*** *(écart-type estimé à 0,29%).*

Cas A.100. *Avec la matrice de pseudocorrélation G_{42}, sa matrice de pondération H_{42}, le vecteur aléatoire X_{IV} (cas des facteurs de pertes), l'ajout d'une dimension vide (voir section 4.2.1) mène à un intervalle de* ***0,10%*** *autour du SCR (après modification = 2999,268 et initial = 2996,243). La déformation est donc* ***faible*** *(écart-type estimé à 0,29%).*

Cas A.101. *Avec la matrice de pseudocorrélation G_{51}, sa matrice de pondération H_{51}, le vecteur aléatoire X_V (cas des facteurs de pertes), l'application de l'algorithme de permutation 4.1 mène à un intervalle de* ***1,56%*** *autour du SCR (min = 2658,804 et max = 2700,284). La déformation est donc* ***très forte*** *(écart-type estimé à 0,18%).*

Cas A.102. *Avec la matrice de pseudocorrélation G_{51}, sa matrice de pondération H_{51}, et l'encadrement $H_{min} = 0$ et $H_{max} = 1$, le vecteur aléatoire X_V (cas des facteurs de pertes), l'application de l'algorithme d'optimisation de la pondération 4.2 mène à un intervalle de* ***1,75%*** *autour du SCR (min = 2697,89 et max = 2745,082). La déformation est donc* ***très forte*** *(écart-type estimé à 0,18%).*

Cas A.103. *Avec la matrice de pseudocorrélation G_{51}, sa matrice de pondération H_{51}, et l'encadrement $H_{min} = H_{51} - 0,1$ et $H_{max} = H_{51} + 0,1$, le*

*vecteur aléatoire X_V (cas des facteurs de pertes), l'application de l'algorithme d'optimisation de la pondération 4.2 mène à un intervalle de **0,52%** autour du SCR (min = 2660,282 et max = 2674,215). La déformation est donc **faible** (écart-type estimé à 0,18%).*

Cas A.104. *Avec la matrice de pseudocorrélation G_{51}, sa matrice de pondération H_{51}, le vecteur aléatoire X_V (cas des facteurs de pertes), l'ajout d'une dimension vide (voir section 4.2.1) mène à un intervalle de **0,21%** autour du SCR (après modification = 2665,322 et initial = 2670,862). La déformation est donc **faible** (écart-type estimé à 0,18%).*

Cas A.105. *Avec la matrice de pseudocorrélation G_{52}, sa matrice de pondération H_{52}, le vecteur aléatoire X_V (cas des facteurs de pertes), l'application de l'algorithme de permutation 4.1 mène à un intervalle de **6,00%** autour du SCR (min = 3113,237 et max = 3300,162). La déformation est donc **très forte** (écart-type estimé à 0,32%).*

Cas A.106. *Avec la matrice de pseudocorrélation G_{52}, sa matrice de pondération H_{52}, et l'encadrement $H_{min} = 0$ et $H_{max} = 1$, le vecteur aléatoire X_V (cas des facteurs de pertes), l'application de l'algorithme d'optimisation de la pondération 4.2 mène à un intervalle de **2,96%** autour du SCR (min = 3148,103 et max = 3241,263). La déformation est donc **très forte** (écart-type estimé à 0,32%).*

Cas A.107. *Avec la matrice de pseudocorrélation G_{52}, sa matrice de pondération H_{52}, et l'encadrement $H_{min} = H_{52} - 0,1$ et $H_{max} = H_{52} + 0,1$, le vecteur aléatoire X_V (cas des facteurs de pertes), l'application de l'algorithme d'optimisation de la pondération 4.2 mène à un intervalle de **4,75%** autour du SCR (min = 3100,957 et max = 3248,255). La déformation est donc **très forte** (écart-type estimé à 0,32%).*

Cas A.108. *Avec la matrice de pseudocorrélation G_{52}, sa matrice de pondération H_{52}, le vecteur aléatoire X_V (cas des facteurs de pertes), l'ajout d'une dimension vide (voir section 4.2.1) mène à un intervalle de **1,09%** autour du SCR (après modification = 3228,873 et initial = 3264,483). La déformation est donc **forte** (écart-type estimé à 0,32%).*

Cas A.109. *Avec la matrice de pseudocorrélation G_{101}, sa matrice de pondération H_{101}, le vecteur aléatoire X_X (cas des facteurs de pertes), l'application de l'algorithme de permutation 4.1 mène à un intervalle de **2,38%** autour du SCR (min = 6644,305 et max = 6802,299). La déformation est donc **très forte** (écart-type estimé à 0,29%).*

Cas A.110. *Avec la matrice de pseudocorrélation G_{101}, sa matrice de pondération H_{101}, et l'encadrement $H_{min} = 0$ et $H_{max} = 1$, le vecteur aléatoire X_X (cas des facteurs de pertes), l'application de l'algorithme d'optimisation*

*de la pondération 4.2 mène à un intervalle de **2,62%** autour du SCR (min = 6632,395 et max = 6806,281). La déformation est donc **très forte** (écart-type estimé à 0,29%).*

Cas A.111. *Avec la matrice de pseudocorrélation G_{101}, sa matrice de pondération H_{101}, et l'encadrement $H_{min} = H_{101} - 0,1$ et $H_{max} = H_{101} + 0,1$, le vecteur aléatoire X_X (cas des facteurs de pertes), l'application de l'algorithme d'optimisation de la pondération 4.2 mène à un intervalle de **1,11%** autour du SCR (min = 6668,115 et max = 6742,007). La déformation est donc **forte** (écart-type estimé à 0,29%).*

Cas A.112. *Avec la matrice de pseudocorrélation G_{101}, sa matrice de pondération H_{101}, le vecteur aléatoire X_X (cas des facteurs de pertes), l'ajout d'une dimension vide (voir section 4.2.1) mène à un intervalle de **0,04%** autour du SCR (après modification = 6710,545 et initial = 6713,349). La déformation est donc **faible** (écart-type estimé à 0,29%).*

A.9.5 Cas d'une copule gaussienne et de facteurs de risques

Cas A.113. *Avec la matrice de pseudocorrélation G_{31}, sa matrice de pondération H_{31}, et l'encadrement $H_{min} = 0$ et $H_{max} = 1$, le vecteur aléatoire X_{IIIb} (cas des facteurs de risques), l'application de l'algorithme d'optimisation de la pondération 4.2 mène à un intervalle de **0,32%** autour du SCR (min = 12081399 et max = 12120187). La déformation est donc **faible** (écart-type estimé à 0,16%).*

Cas A.114. *Avec la matrice de pseudocorrélation G_{31}, sa matrice de pondération H_{31}, et l'encadrement $H_{min} = H_{31} - 0,1$ et $H_{max} = H_{31} + 0,1$, le vecteur aléatoire X_{IIIb} (cas des facteurs de risques), l'application de l'algorithme d'optimisation de la pondération 4.2 mène à un intervalle de **0,52%** autour du SCR (min = 12072651 et max = 12135251). La déformation est donc **faible** (écart-type estimé à 0,16%).*

Cas A.115. *Avec la matrice de pseudocorrélation G_{31}, sa matrice de pondération H_{31}, le vecteur aléatoire X_{IIIb} (cas des facteurs de risques), l'ajout d'une dimension vide (voir section 4.2.1) mène à un intervalle de **0,96%** autour du SCR (après modification = 12249386 et initial = 12132660). La déformation est donc **très forte** (écart-type estimé à 0,16%).*

Cas A.116. *Avec la matrice de pseudocorrélation G_{32}, sa matrice de pondération H_{32}, et l'encadrement $H_{min} = 0$ et $H_{max} = 1$, le vecteur aléatoire X_{IIIb} (cas des facteurs de risques), l'application de l'algorithme d'optimisation de la pondération 4.2 mène à un intervalle de **3,07%** autour du SCR (min = 12447404 et max = 12829070). La déformation est donc **très forte** (écart-type estimé à 0,19%).*

Cas A.117. *Avec la matrice de pseudocorrélation G_{32}, sa matrice de pondération H_{32}, et l'encadrement $H_{min} = H_{32} - 0,1$ et $H_{max} = H_{32} + 0,1$, le vecteur aléatoire X_{IIIb} (cas des facteurs de risques), l'application de l'algorithme d'optimisation de la pondération 4.2 mène à un intervalle de* ***1,62%*** *autour du SCR (min = 12765514 et max = 12971883). La déformation est donc* ***très forte*** *(écart-type estimé à 0,19%).*

Cas A.118. *Avec la matrice de pseudocorrélation G_{32}, sa matrice de pondération H_{32}, le vecteur aléatoire X_{IIIb} (cas des facteurs de risques), l'ajout d'une dimension vide (voir section 4.2.1) mène à un intervalle de* ***0,93%*** *autour du SCR (après modification = 13249432 et initial = 13127451). La déformation est donc* ***forte*** *(écart-type estimé à 0,19%).*

Cas A.119. *Avec la matrice de pseudocorrélation G_{41}, sa matrice de pondération H_{41}, et l'encadrement $H_{min} = 0$ et $H_{max} = 1$, le vecteur aléatoire X_{IVb} (cas des facteurs de risques), l'application de l'algorithme d'optimisation de la pondération 4.2 mène à un intervalle de* ***0,46%*** *autour du SCR (min = 3516332 et max = 3532352). La déformation est donc* ***forte*** *(écart-type estimé à 0,10%).*

Cas A.120. *Avec la matrice de pseudocorrélation G_{41}, sa matrice de pondération H_{41}, et l'encadrement $H_{min} = H_{41} - 0,1$ et $H_{max} = H_{41} + 0,1$, le vecteur aléatoire X_{IVb} (cas des facteurs de risques), l'application de l'algorithme d'optimisation de la pondération 4.2 mène à un intervalle de* ***0,36%*** *autour du SCR (min = 3520161 et max = 3532792). La déformation est donc* ***forte*** *(écart-type estimé à 0,10%).*

Cas A.121. *Avec la matrice de pseudocorrélation G_{41}, sa matrice de pondération H_{41}, le vecteur aléatoire X_{IVb} (cas des facteurs de risques), l'ajout d'une dimension vide (voir section 4.2.1) mène à un intervalle de* ***0,15%*** *autour du SCR (après modification = 3534685 et initial = 3529310). La déformation est donc* ***faible*** *(écart-type estimé à 0,10%).*

Cas A.122. *Avec la matrice de pseudocorrélation G_{42}, sa matrice de pondération H_{42}, et l'encadrement $H_{min} = 0$ et $H_{max} = 1$, le vecteur aléatoire X_{IVb} (cas des facteurs de risques), l'application de l'algorithme d'optimisation de la pondération 4.2 mène à un intervalle de* ***0,90%*** *autour du SCR (min = 3519944 et max = 3551629). La déformation est donc* ***faible*** *(écart-type estimé à 0,31%).*

Cas A.123. *Avec la matrice de pseudocorrélation G_{42}, sa matrice de pondération H_{42}, et l'encadrement $H_{min} = H_{42} - 0,1$ et $H_{max} = H_{42} + 0,1$, le vecteur aléatoire X_{IVb} (cas des facteurs de risques), l'application de l'algorithme d'optimisation de la pondération 4.2 mène à un intervalle de* ***0,70%*** *autour du SCR (min = 3540453 et max = 3565096). La déformation est donc* ***faible*** *(écart-type estimé à 0,31%).*

Cas A.124. *Avec la matrice de pseudocorrélation G_{42}, sa matrice de pondération H_{42}, le vecteur aléatoire X_{IVb} (cas des facteurs de risques), l'ajout d'une dimension vide (voir section 4.2.1) mène à un intervalle de* ***0,08%*** *autour du SCR (après modification = 3561905 et initial = 3564593). La déformation est donc* ***faible*** *(écart-type estimé à 0,31%).*

Cas A.125. *Avec la matrice de pseudocorrélation G_{51}, sa matrice de pondération H_{51}, et l'encadrement $H_{min} = 0$ et $H_{max} = 1$, le vecteur aléatoire X_{Vb} (cas des facteurs de risques), l'application de l'algorithme d'optimisation de la pondération 4.2 mène à un intervalle de* ***4,57%*** *autour du SCR (min = 8541,412 et max = 8931,55). La déformation est donc* ***très forte*** *(écart-type estimé à 0,68%).*

Cas A.126. *Avec la matrice de pseudocorrélation G_{51}, sa matrice de pondération H_{51}, et l'encadrement $H_{min} = H_{51} - 0,1$ et $H_{max} = H_{51} + 0,1$, le vecteur aléatoire X_{Vb} (cas des facteurs de risques), l'application de l'algorithme d'optimisation de la pondération 4.2 mène à un intervalle de* ***1,38%*** *autour du SCR (min = 9201,858 et max = 9328,433). La déformation est donc* ***faible*** *(écart-type estimé à 0,68%).*

Cas A.127. *Avec la matrice de pseudocorrélation G_{51}, sa matrice de pondération H_{51}, le vecteur aléatoire X_{Vb} (cas des facteurs de risques), l'ajout d'une dimension vide (voir section 4.2.1) mène à un intervalle de* ***0,03%*** *autour du SCR (après modification = 9336,614 et initial = 9333,506). La déformation est donc* ***faible*** *(écart-type estimé à 0,68%).*

Cas A.128. *Avec la matrice de pseudocorrélation G_{52}, sa matrice de pondération H_{52}, et l'encadrement $H_{min} = 0$ et $H_{max} = 1$, le vecteur aléatoire X_{Vb} (cas des facteurs de risques), l'application de l'algorithme d'optimisation de la pondération 4.2 mène à un intervalle de* ***5,09%*** *autour du SCR (min = 8230,702 et max = 8649,879). La déformation est donc* ***très forte*** *(écart-type estimé à 0,28%).*

Cas A.129. *Avec la matrice de pseudocorrélation G_{52}, sa matrice de pondération H_{52}, et l'encadrement $H_{min} = H_{52} - 0,1$ et $H_{max} = H_{52} + 0,1$, le vecteur aléatoire X_{Vb} (cas des facteurs de risques), l'application de l'algorithme d'optimisation de la pondération 4.2 mène à un intervalle de* ***1,75%*** *autour du SCR (min = 8180,515 et max = 8323,612). La déformation est donc* ***très forte*** *(écart-type estimé à 0,28%).*

Cas A.130. *Avec la matrice de pseudocorrélation G_{52}, sa matrice de pondération H_{52}, le vecteur aléatoire X_{Vb} (cas des facteurs de risques), l'ajout d'une dimension vide (voir section 4.2.1) mène à un intervalle de* ***0,55%*** *autour du SCR (après modification = 8337,67 et initial = 8291,983). La déformation est donc* ***faible*** *(écart-type estimé à 0,28%).*

Cas A.131. *Avec la matrice de pseudocorrélation G_{101}, sa matrice de pondération H_{101}, et l'encadrement $H_{min} = 0$ et $H_{max} = 1$, le vecteur aléatoire X_{Xb} (cas des facteurs de risques), l'application de l'algorithme d'optimisation de la pondération 4.2 mène à un intervalle de* ***1,67%*** *autour du SCR (min = 4104065 et max = 4172504). La déformation est donc* ***faible*** *(écart-type estimé à 0,63%).*

Cas A.132. *Avec la matrice de pseudocorrélation G_{101}, sa matrice de pondération H_{101}, et l'encadrement $H_{min} = H_{101} - 0,1$ et $H_{max} = H_{101} + 0,1$, le vecteur aléatoire X_{Xb} (cas des facteurs de risques), l'application de l'algorithme d'optimisation de la pondération 4.2 mène à un intervalle de* ***1,47%*** *autour du SCR (min = 4110377 et max = 4170802). La déformation est donc* ***faible*** *(écart-type estimé à 0,63%).*

Cas A.133. *Avec la matrice de pseudocorrélation G_{101}, sa matrice de pondération H_{101}, le vecteur aléatoire X_{Xb} (cas des facteurs de risques), l'ajout d'une dimension vide (voir section 4.2.1) mène à un intervalle de* ***1,29%*** *autour du SCR (après modification = 4162216 et initial = 4216574). La déformation est donc* ***faible*** *(écart-type estimé à 0,63%).*

A.9.6 Cas d'une copule de Student et de facteurs de risques

Cas A.134. *Avec la matrice de pseudocorrélation G_{31}, sa matrice de pondération H_{31}, et l'encadrement $H_{min} = 0$ et $H_{max} = 1$, le vecteur aléatoire X_{IIIb} (cas des facteurs de risques), l'application de l'algorithme d'optimisation de la pondération 4.2 mène à un intervalle de* ***1,16%*** *autour du SCR (min = 12326216 et max = 12469140). La déformation est donc* ***forte*** *(écart-type estimé à 0,32%).*

Cas A.135. *Avec la matrice de pseudocorrélation G_{31}, sa matrice de pondération H_{31}, et l'encadrement $H_{min} = H_{31} - 0,1$ et $H_{max} = H_{31} + 0,1$, le vecteur aléatoire X_{IIIb} (cas des facteurs de risques), l'application de l'algorithme d'optimisation de la pondération 4.2 mène à un intervalle de* ***0,84%*** *autour du SCR (min = 12382841 et max = 12486700). La déformation est donc* ***faible*** *(écart-type estimé à 0,32%).*

Cas A.136. *Avec la matrice de pseudocorrélation G_{31}, sa matrice de pondération H_{31}, le vecteur aléatoire X_{IIIb} (cas des facteurs de risques), l'ajout d'une dimension vide (voir section 4.2.1) mène à un intervalle de* ***1,07%*** *autour du SCR (après modification = 12568778 et initial = 12435526). La déformation est donc* ***forte*** *(écart-type estimé à 0,32%).*

Cas A.137. *Avec la matrice de pseudocorrélation G_{32}, sa matrice de pondération H_{32}, et l'encadrement $H_{min} = 0$ et $H_{max} = 1$, le vecteur aléatoire X_{IIIb} (cas des facteurs de risques), l'application de l'algorithme d'optimisation de*

la pondération 4.2 mène à un intervalle de ***2,12%*** *autour du SCR (min = 13758628 et max = 14050626). La déformation est donc* ***faible*** *(écart-type estimé à 0,86%).*

Cas A.138. *Avec la matrice de pseudocorrélation G_{32}, sa matrice de pondération H_{32}, et l'encadrement $H_{min} = H_{32} - 0,1$ et $H_{max} = H_{32} + 0,1$, le vecteur aléatoire X_{IIIb} (cas des facteurs de risques), l'application de l'algorithme d'optimisation de la pondération 4.2 mène à un intervalle de* ***3,70%*** *autour du SCR (min = 14133464 et max = 14656943). La déformation est donc* ***forte*** *(écart-type estimé à 0,86%).*

Cas A.139. *Avec la matrice de pseudocorrélation G_{32}, sa matrice de pondération H_{32}, le vecteur aléatoire X_{IIIb} (cas des facteurs de risques), l'ajout d'une dimension vide (voir section 4.2.1) mène à un intervalle de* ***0,55%*** *autour du SCR (après modification = 14623516 et initial = 14704558). La déformation est donc* ***faible*** *(écart-type estimé à 0,86%).*

Cas A.140. *Avec la matrice de pseudocorrélation G_{41}, sa matrice de pondération H_{41}, et l'encadrement $H_{min} = 0$ et $H_{max} = 1$, le vecteur aléatoire X_{IVb} (cas des facteurs de risques), l'application de l'algorithme d'optimisation de la pondération 4.2 mène à un intervalle de* ***0,73%*** *autour du SCR (min = 3515774 et max = 3541524). La déformation est donc* ***faible*** *(écart-type estimé à 0,33%).*

Cas A.141. *Avec la matrice de pseudocorrélation G_{41}, sa matrice de pondération H_{41}, et l'encadrement $H_{min} = H_{41} - 0,1$ et $H_{max} = H_{41} + 0,1$, le vecteur aléatoire X_{IVb} (cas des facteurs de risques), l'application de l'algorithme d'optimisation de la pondération 4.2 mène à un intervalle de* ***0,68%*** *autour du SCR (min = 3517889 et max = 3541680). La déformation est donc* ***faible*** *(écart-type estimé à 0,33%).*

Cas A.142. *Avec la matrice de pseudocorrélation G_{41}, sa matrice de pondération H_{41}, le vecteur aléatoire X_{IVb} (cas des facteurs de risques), l'ajout d'une dimension vide (voir section 4.2.1) mène à un intervalle de* ***0,21%*** *autour du SCR (après modification = 3541166 et initial = 3533674). La déformation est donc* ***faible*** *(écart-type estimé à 0,33%).*

Cas A.143. *Avec la matrice de pseudocorrélation G_{42}, sa matrice de pondération H_{42}, et l'encadrement $H_{min} = 0$ et $H_{max} = 1$, le vecteur aléatoire X_{IVb} (cas des facteurs de risques), l'application de l'algorithme d'optimisation de la pondération 4.2 mène à un intervalle de* ***1,30%*** *autour du SCR (min = 3509471 et max = 3555086). La déformation est donc* ***forte*** *(écart-type estimé à 0,35%).*

Cas A.144. *Avec la matrice de pseudocorrélation G_{42}, sa matrice de pondération H_{42}, et l'encadrement $H_{min} = H_{42} - 0,1$ et $H_{max} = H_{42} + 0,1$, le*

*vecteur aléatoire X_{IVb} (cas des facteurs de risques), l'application de l'algorithme d'optimisation de la pondération 4.2 mène à un intervalle de **0,97%** autour du SCR (min = 3521590 et max = 3555627). La déformation est donc **faible** (écart-type estimé à 0,35%).*

Cas A.145. *Avec la matrice de pseudocorrélation G_{42}, sa matrice de pondération H_{42}, le vecteur aléatoire X_{IVb} (cas des facteurs de risques), l'ajout d'une dimension vide (voir section 4.2.1) mène à un intervalle de **0,56%** autour du SCR (après modification = 3529009 et initial = 3549029). La déformation est donc **faible** (écart-type estimé à 0,35%).*

Cas A.146. *Avec la matrice de pseudocorrélation G_{51}, sa matrice de pondération H_{51}, et l'encadrement $H_{min} = 0$ et $H_{max} = 1$, le vecteur aléatoire X_{Vb} (cas des facteurs de risques), l'application de l'algorithme d'optimisation de la pondération 4.2 mène à un intervalle de **4,33%** autour du SCR (min = 8768,985 et max = 9148,675). La déformation est donc **très forte** (écart-type estimé à 0,59%).*

Cas A.147. *Avec la matrice de pseudocorrélation G_{51}, sa matrice de pondération H_{51}, et l'encadrement $H_{min} = H_{51} - 0,1$ et $H_{max} = H_{51} + 0,1$, le vecteur aléatoire X_{Vb} (cas des facteurs de risques), l'application de l'algorithme d'optimisation de la pondération 4.2 mène à un intervalle de **2,82%** autour du SCR (min = 9136,763 et max = 9394,289). La déformation est donc **forte** (écart-type estimé à 0,59%).*

Cas A.148. *Avec la matrice de pseudocorrélation G_{51}, sa matrice de pondération H_{51}, le vecteur aléatoire X_{Vb} (cas des facteurs de risques), l'ajout d'une dimension vide (voir section 4.2.1) mène à un intervalle de **0,51%** autour du SCR (après modification = 9409,494 et initial = 9457,665). La déformation est donc **faible** (écart-type estimé à 0,59%).*

Cas A.149. *Avec la matrice de pseudocorrélation G_{52}, sa matrice de pondération H_{52}, et l'encadrement $H_{min} = 0$ et $H_{max} = 1$, le vecteur aléatoire X_{Vb} (cas des facteurs de risques), l'application de l'algorithme d'optimisation de la pondération 4.2 mène à un intervalle de **4,62%** autour du SCR (min = 8444,656 et max = 8834,731). La déformation est donc **forte** (écart-type estimé à 0,99%).*

Cas A.150. *Avec la matrice de pseudocorrélation G_{52}, sa matrice de pondération H_{52}, et l'encadrement $H_{min} = H_{52} - 0,1$ et $H_{max} = H_{52} + 0,1$, le vecteur aléatoire X_{Vb} (cas des facteurs de risques), l'application de l'algorithme d'optimisation de la pondération 4.2 mène à un intervalle de **2,88%** autour du SCR (min = 8339,173 et max = 8579,715). La déformation est donc **faible** (écart-type estimé à 0,99%).*

Cas A.151. *Avec la matrice de pseudocorrélation G_{52}, sa matrice de pondération H_{52}, le vecteur aléatoire X_{Vb} (cas des facteurs de risques), l'ajout d'une*

*dimension vide (voir section 4.2.1) mène à un intervalle de **2,44%** autour du SCR (après modification = 8680,028 et initial = 8473,53). La déformation est donc **faible** (écart-type estimé à 0,99%).*

Cas A.152. *Avec la matrice de pseudocorrélation G_{101}, sa matrice de pondération H_{101}, et l'encadrement $H_{min} = 0$ et $H_{max} = 1$, le vecteur aléatoire X_{Xb} (cas des facteurs de risques), l'application de l'algorithme d'optimisation de la pondération 4.2 mène à un intervalle de **2,34%** autour du SCR (min = 4139819 et max = 4236750). La déformation est donc **faible** (écart-type estimé à 0,84%).*

Cas A.153. *Avec la matrice de pseudocorrélation G_{101}, sa matrice de pondération H_{101}, et l'encadrement $H_{min} = H_{101} - 0,1$ et $H_{max} = H_{101} + 0,1$, le vecteur aléatoire X_{Xb} (cas des facteurs de risques), l'application de l'algorithme d'optimisation de la pondération 4.2 mène à un intervalle de **2,09%** autour du SCR (min = 4138668 et max = 4225342). La déformation est donc **faible** (écart-type estimé à 0,84%).*

Cas A.154. *Avec la matrice de pseudocorrélation G_{101}, sa matrice de pondération H_{101}, le vecteur aléatoire X_{Xb} (cas des facteurs de risques), l'ajout d'une dimension vide (voir section 4.2.1) mène à un intervalle de **1,06%** autour du SCR (après modification = 4227908 et initial = 4273245). La déformation est donc **faible** (écart-type estimé à 0,84%).*

B Notice du code informatique (sous R)

Voici la description des principales commandes du code informatique développé pour obtenir les résultats présentés dans ce mémoire (voir Annexe A) :

- La commande **PA(G)** renvoit la matrice PSD la plus proche de G, sans pondération ni encadrement, à l'aide de l'algorithme 2.3 des projections alternées ;
- La commande **Newton(G)** renvoit la matrice PSD la plus proche de G, sans pondération ni encadrement, à l'aide de l'algorithme 2.5 de PSDisation de Newton ;
- La commande **Rebbis(G,H)** renvoit la matrice PSD la plus proche de G, avec les poids associés à la matrice H, mais sans encadrement, à l'aide de l'algorithme 2.9 hypersphère ;
- La commande **Rebcont(G, C^{min}, C^{max})** renvoit la matrice PSD la plus proche de G, sans pondération mais encadrée par les contraintes fixées par les inégalités suivantes : $\forall (i,j) \in [\![1;n]\!]^2 | i > j, C_{ij}^{min} < S_{ij} < C_{ij}^{max}$, en notant S la matrice PSD la plus proche de G (voir algorithme 2.10). On remarque donc que seules les contraintes sur les parties triangulaires inférieures strictes des matrices C^{min} et C^{max} entrent en jeu dans l'algorithme ;
- La commande **Logno(σ, m, ech)** renvoit une matrice de taille $n \times ech$ telle que chaque ligne i corresponde à ech tirages d'une variable aléatoire lognormale de paramètres μ = m[i,1] et $\sigma = \sigma$[i,i]. Les données d'entrée sont m un vecteur de dimension $n \times 1$ et σ une matrice de dimension $n \times n$. Seuls les nombres sur la diagonale de σ interviennent ;
- La commande **VaRbis(S,Y)** renvoit le résultat de la simulation du SCR à l'aide d'une copule dont la corrélation est représentée par la matrice S. La matrice S doit nécessairement être symétrique définie positive pour que l'algorithme puisse renvoyer un résultat (sans cela, la décomposition de Cholesky échouera). Il est à noter que chaque ligne de Y représente les valeurs de chacune des marginales simulées indépendamment, avant agrégation. Le nombre de colonnes de Y représente donc le nombre de tirages (noté ech = 32768 dans le code ci-dessous). Si l'utilisateur souhaite augmenter le nombre de tirages, il devra veiller à modifier également manuellement la valeur du paramètre ech dans la fonction **VaRbis(S,Y)**. Enfin, le code prévoit 2 versions de cette fonction, correspondant à une agrégation par copule gaussienne ou par copule de Student. L'utilisateur devra veiller à bien entrer dans R la fonction **VaRbis(S,Y)** avec laquelle il souhaite que l'agrégation se fasse. Dans ce code, le paramètre de la loi du chi-2 associée à la simulation de la copule de Student est noté nu = 3 (à changer manuellement dans la commande **Student** si nécessaire) ;
- La commande **Teststab(S, Y)** renvoit q = 10 calculs consécutifs du SCR, vu comme une moyenne de m = 4 calculs successifs du SCR (la fonction **VaRbis(S,Y)** est appelée 4 fois à chaque fois). Le changement éventuel des paramètres doit être fait manuellement dans le corps de la fonction (m représentant le nombre moyen de calculs du SCR, et q le nombre de fois où le calcul est réeffectué) ;
- La commande **OptimPermutGen(G,H,Y)** renvoit les deux permutations pour lesquels le minimum et la maximum de la fonction : $\sigma \to SCR_{f,\mathcal{A}}^{\sigma.Y}(PSD_{\sigma.H}(\sigma.G))$ sont atteints. Il est à noter que cette fonction ne simule à chaque étape que l'agrégation (les simulations des marginales sont fixées indépendamment dans la matrice Y de taille $n \times ech$, avec n le nombre de marginales et ech le nombre de tirages). Le nombre m de calculs du SCR effectués à chaque itération (pour le calcul de la moyenne, voir équation 10) doit être fixé à la main en dur dans la fonction **Permut**. Le choix de la fonction **VaRbis(S,Y)** permet de sélectionner le type de copule (Student ou Gauss). Par ailleurs, le SCR est calculé en considérant la fonction de perte comme une somme des marginales. Pour calculer

le SCR sur des facteurs de risques, il faut modifier préalablement la fonction **Pertes**. Par exemple, si on souhaite que la première marginale de Y intervienne sous la forme $0,03\,Y_1^4$, il faudra effectuer préalablement la transformation suivante sur la première ligne de la matrice Y au sein de la fonction **Pertes** : $Y[,1] \leftarrow 0,03\,Y[,1]^4$;

— La commande **OptimH(G,H^{min}, H^{max}, Y)** renvoit les deux matrices de pondération (comprises entre H^{min} et H^{max}) pour lesquels le minimum et la maximum de la fonction : $H \rightarrow SCR_{f,\mathcal{A}}(PSD_H(G))$ sont atteints. Les mêmes remarques que pour la fonction **OptimPermutGen(G,H,Y)** s'appliquent (choix du nombre de tirages ech, de la copule, utilisation des facteurs de risques etc.). En particulier, le nombre nbm utilisé pour calculer la moyenne du SCR (voir équation 10) doit être fixé en dur dans la fonction **OptimH** ;

— La commande **Ajoutvide(G,H,Y)** renvoit les deux SCR avant et après ajout d'une dimension vide dans les modalités décrites dans la section 4.2.1. De même, le nombre m utilisé pour calculer la moyenne du SCR (voir équation 10) doit être fixé en dur dans la fonction **Ajoutvide** ;

— Remarque 1 : Pour le code s'exécute entièrement, il est nécessaire de charger les packages R suivants : *permute*, *rngWELL* et *randtoolbox* ;

— Remarque 2 : Lors du recours à un générateur de nombres aléatoires de Sobol, une attention particulière doit être apportée à l'initialisation. Celle-ci dépend de la copule utilisée. En effet, dans le cas d'une copule gaussienne, seules n dimensions (où n représente la nombres de marginales) doivent être initialisées (soit l'éxecution préalable de la commande **initsobol <- sobol(ech, dim = n, init = TRUE)** avec la variable initisobol comme variable muette). Cette initialisation est également nécessaire avant de simuler des marginales non-corrélées. Par contre, à cause de la loi du chi-2 qui intervient dans la simulation de la copule de Student (voir 3.7), il est nécessaire d'initialiser n+1 dimensions dans le cas d'utilisation de la copule de Student (soit l'éxecution préalable de la commande **initsobol <- sobol(ech, dim = n+1, init = TRUE)**).

— Exemple : après avoir paramétré les commandes de manière à utiliser un générateur de nombres aléatoires de Sobol, et après avoir paramétré la commande **VaRbis** de manière à recourir à la copule de Student, il faut exécuter les commandes suivantes pour analyser la stabilité :

```
ech <- 32768
n <- 3
initsobol <- sobol(ech,n)
sigma3 <- diag(c(0.5,0.5,0.1))
m3 <- matrix(c(6,5,4),3,1)
Y3 = Logno(sigma3,m3,ech)
initsobol <- sobol(ech,n+1)
Teststab(S31H,Y3,4)
```