



HAL
open science

Operational choices for risk aggregation in insurance: PSDization and SCR sensitivity

Xavier Milhaud, Victorien Poncelet, Clément Saillard

► **To cite this version:**

Xavier Milhaud, Victorien Poncelet, Clément Saillard. Operational choices for risk aggregation in insurance: PSDization and SCR sensitivity. *Risks*, 2018, 6 (2), pp.36. 10.3390/risks6020036. hal-01711196

HAL Id: hal-01711196

<https://hal.science/hal-01711196>

Submitted on 16 Feb 2018

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

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

Operational choices for risk aggregation in insurance: PSDization and SCR sensitivity

Xavier Milhaud¹, Victorien Poncelet², Clément Saillard^{3*}

¹Université de Lyon, Université Claude Bernard Lyon1, ISFA, LSAF, F-69007, LYON, France

²Banque de France, 61 rue Taitbout, 75009 Paris, France

³BNP Paribas Cardif, RISK, 10 rue du Port, 92000 Nanterre, France

February 16, 2018

Abstract

This paper answers crucial questions about the robustness of the *PSDization process* for applications in insurance. *PSDization* refers to the process that forces a matrix to become positive semi-definite. For companies using copulas to aggregate risks in their internal model, *PSDization* occurs when working with correlation matrices to compute the Solvency Capital Requirement (SCR). We study how classical operational choices concerning the modelling of risk dependence impacts the SCR during PSDization. These operations refer to permutations of risks (or business lines) in the correlation matrix, addition of a new risk, and introduction of confidence weights given to the correlation coefficients. Using genetic algorithms, it is shown that theoretically neutral transformations of the correlation matrix can surprisingly lead to significant sensitivities of the SCR (up to 6%). This highlights the need for a very strong internal control around the PSDization step.

Keywords: Solvency II, risk aggregation, positive semi-definite, Rebonato-Jäckel.

*Corresponding author: clement.saillard@bnpparibas.com. The views expressed herein reflect solely those of their authors.

1 Motivation

When measuring the insurer’s exposure to numerous risks, in particular to determine the own funds requirement (or Solvency Capital Requirement in insurance, denoted further by SCR^g , and representing the 99.5th percentile of the aggregated loss distribution), one of the most sensitive steps is the modelling of the dependence between those risks.

This question is of course important and, quite deservedly, has recently attracted attention from the scientific community, see for instance the works by Georgescu et al. (2017), Cifuentes and Charlin (2016), Bernard et al. (2014), Cheung and Vanduffel (2013), Devineau and Loisel (2009), Filipovic (2009), Denuit et al. (1999), and references therein. This aggregation step allows to take into account mitigation, or the potentiality of those individual risks to occur simultaneously. According to the European Directive Solvency II, there exist two main approaches to compute aggregated risk measures considering the dependence structure between risks. In the first case, this aggregation is performed through a *variance-covariance* approach via the “Standard Formula”, that is

$$SCR^g = \sqrt{\sum_{i \in \llbracket 1, n \rrbracket} \sum_{j \in \llbracket 1, n \rrbracket} \rho_{ij} \times SCR_i \times SCR_j}, \quad (1.1)$$

where SCR_i is the 99.5th percentile of the loss associated to risk i , and ρ_{ij} is the linear correlation such that $\rho_{ij} = Cov(SCR_i, SCR_j) / \sqrt{Var(SCR_i)Var(SCR_j)}$. This technique was shown to be valid for elliptical loss vectors, which is not the case in full generality¹.

Another possibility for insurers is to calculate SCR^g thanks to their internal model, once the latter has been approved by supervisors. In this case, insurers usually work with copulas for the aggregation of risk factors in order to obtain the full distribution of losses. Indeed, copulas can model the most general situation for dependence, as shown by the well known Sklar’s theorem. In practice, internal models require the implementation of these successive steps:

1. calibration of marginal distributions for each risk factor (e.g. equity, interest rates);
2. aggregation of risk factors by modelling their dependence through a copula;
3. fitting of proxy functions to link the risk factors to their associated loss (e.g. through curve-fitting in ALM forecasts), which leads to get the entire distribution of the aggregated loss. Taking the 99.5th percentile of this distribution then allows to evaluate the aggregated SCR (SCR^g with our notations).

¹As the CEIOPS (ex-EIOPA) admitted in its Solvency II calibration paper of April 2010 (SEC-10-40).

The aggregation thus requires a correlation matrix as input, whatever the technique. This matrix, whose dimensions can be huge in practice (e.g. 1000x1000, i.e. with around 500 000 different values), is full of correlation coefficients that can result from empirical statistical measures, expert judgments or automatic formulas. By the way, it is thus rarely positive semi-definite (PSD): this is what is commonly called a *pseudo-correlation matrix*. Unfortunately, this matrix cannot be used directly to aggregate risks. Indeed, both *variance-covariance* and *copula* aggregation techniques require the correlation matrix to be PSD (that is with all eigenvalues ≥ 0) for the following main reasons:

- Coherence: it is a well-known property that correlation matrices are PSD. The presence of negative eigenvalues points a logical mistake made while establishing the coefficients of the matrix. For instance, consider the case of a 3x3 matrix : if the coefficients indicate a strong positive correlation between the first and second risk factors, a strong positive correlation between the second and third risk factors, but a strong negative correlation between the first and third risk factors, this will generate a negative eigenvalue corresponding to the coherence mistake made;
- Prudence: taking more risk could decrease the insurer's SCR^g if there exists one negative eigenvalue associated to an eigenvector with positive coefficients. For instance, consider the loss vector (100M€, 10M€, 40M€) and the correlation matrix

$$\rho = \begin{pmatrix} \rho_{11} & \rho_{12} & \rho_{13} \\ \rho_{21} & \rho_{22} & \rho_{23} \\ \rho_{31} & \rho_{32} & \rho_{33} \end{pmatrix} = \begin{pmatrix} 1 & -0.9 & -0.5 \\ -0.9 & 1 & -0.5 \\ -0.5 & -0.5 & 1 \end{pmatrix}.$$

Using the variance-covariance approach, the riskier situation with losses (106.20M€; 16.20M€; 44.81M€) leads to a lower SCR^g (70.48M€ against 74.16M€)!

- Ability to perform simulations: in the copula approach, the input correlation matrix has to be PSD to apply Choleski decomposition. This is necessary for Gaussian or Student vectors, which are the most famous cases for such tasks in practice.

Using PSD correlation matrices is therefore crucial, which explains why it is explicitly required by Delegated Rules (EU) 2015/35 (Annex XVIII) of the Solvency II regulation. In this view, insurers apply algorithms on their pseudo-correlation matrix in order to make it become PSD: this is the so-called *PSDization process*. Most common algorithms can be separated into three categories: Alternating Projections, Newton and Hypersphere

algorithms. One focuses in this paper on the Rebonato-Jäckel algorithm, which belongs to the latter family (see Section 3 for further details about the motivation of this choice).

Given this framework, one wonders how the SCR is impacted by standard operations on the correlation matrix used for risk aggregation. Our interest lies in studying operational choices such as weighting the correlation coefficients during *PSDization* (to reflect the confidence experts may have on these coefficients), permutating some columns in the matrix (equivalently reorder the risks before aggregation), or adding one dimension (analogously add a business line, with the constraint that it should not impact SCR^g at the end). Numerical examples support the main idea of the paper: transformations of the matrix, with low or null theoretical impact on SCR^g , sometimes lead to unexpected changes of this global SCR. For large insurance companies, this is all the more important that a 1% change of SCR^g can cost millions of euros in terms of capital. This would strongly affect the Return On Equity index, an essential profitability indicator for investors.

The paper is organized as follows: Section 2 introduces the pseudo-correlation matrices to be considered hereafter. Section 3 describes most common PSD algorithms, and motivates our choice. Thanks to some toy examples, Section 4 illustrates to which extent *PSDization* leads to modify the initial pseudo-correlation coefficients. The cases of higher risk matrix dimensions, weighted correlation coefficients and risk permutations are studied. Finally, real-life sensitivities are assessed in Section 5 thanks to the use of genetic algorithms and simulations, and provides with some interesting results concerning the aforementioned operations and their impact on the global SCR of the company.

2 Pseudo-correlation matrices under study

2.1 Correlation matrices of the Standard Formula

Before studying the algorithms which enable to make a matrix be PSD when using an internal model, it is verified that the matrices defined by the standard formula are already PSD. As a reminder, the Standard Formula given by the regulation states that risks should be aggregated in a bottom-up approach, with a tree-based structure. This means that individual SCRs first have to be assessed, each one corresponding to a module (Life, Non Life, and so on). Solvency II texts then define several correlation matrices for each level of aggregation. Table 1 shows that the eigenvalues of these matrices are all positive, meaning that they are PSD. Except the global matrix which can be found in the Directive, all matrices are described in the Delegated Acts, see references in the first column of Table 1.

Module	Dimension	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7	λ_8	λ_9	λ_{10}	λ_{11}	λ_{12}	PSD
Global (Ann. IV)	5	1.92	1.16	0.75	0.75	0.40								Yes
Market up (Art. 164)	6	2.47	1.18	1.00	0.68	0.50	0.15							Yes
Market down (Art. 164)	6	2.89	1.00	0.87	0.57	0.50	0.15							Yes
Life (Art. 136)	7	2.18	1.51	1.07	0.81	0.70	0.57	0.12						Yes
Health SLT (Art. 151)	6	2.04	1.43	1.00	0.81	0.58	0.12							Yes
Health non SLT (Ann. XV)	4	3	0.5	0.5	0.5									Yes
Health (Art. 144)	3	1.68	0.81	0.5										Yes
Non Life (Art. 114)	3	1.25	1	0.75										Yes
Prem. Reserve (Ann. IV)	12	4.91	1.45	1.09	0.97	0.73	0.68	0.61	0.48	0.38	0.33	0.20	0.12	Yes

Table 1: Eigen values of correlation matrices in Solvency II regulation.

2.2 Notations and correlation matrices under study

Throughout the paper, G_{xy} denotes the initial pseudo-correlation matrix to be *PSDized*, where x refers to the matrix dimension and y is the number of the example. When using weights to apply to the correlation coefficients during *PSDization*, a weighting matrix H_{xy} is defined. Then, we denote by S_{xy}^{PA} the *PSDized* matrix obtained from G_{xy} using the Alternated Projections algorithm; S_{xy}^N the *PSDized* matrix with the Newton algorithm; S_{xy} the *PSDized* matrix with the Hypersphere algorithm and S_{xy}^H the *PSDized* matrix with the Hypersphere algorithm using a weighting matrix H . The use of a weighting matrix enables to give more importance to one or several correlation coefficients with respect to other coefficients when making the correlation matrix PSD. This is very useful for insurance companies, since some coefficients can have an bigger impact on the final capital requirement, and the aim would be that *PSDization* modifies these coefficients as little as possible. It must be noted that some algorithms (Newton, Hypersphere) can be extended to use constraints on the correlation coefficients, such as $\rho_{ij}^{min} \leq \rho_{ij} \leq \rho_{ij}^{max}$. However, these extensions are left for future research because of two main reasons: very few practitioners use them, and they cause non-trivial theoretical and practical issues (there could be no solution, i.e. *PSDized* matrix, according to the constrained set).

Our examples are built with the same simple idea: assess the impact of *PSDization* combined to *classical operational choices* on diverse situations to reflect the real world. We thus consider correlation matrices of various dimensions, with positive and negative eigen values, and with high heterogeneity regarding their individual correlation coefficients (positive, negative, far or close to extreme values -1 or +1).

The seven examples studied all along the paper are listed in Appendix A, including the 10-

dimension correlation matrix that has been created manually. This example was designed with the aim of having a certain coherence with the reality of risk aggregation in insurance. The first risk factor, say X_1 , refers to the risk that interest rates decrease (X_2 represents the risk of interest rates rising). X_3 corresponds to unexpected high expenses, X_4 relates to the bad assessment of the level of expenses, X_5 is the risk of spreads increasing. The risk that market stocks drop is given by X_6 , X_7 accounts for the longevity risk, X_8 is the mass lapse risk, X_9 corresponds to the reserve risk in Health business line ; and X_{10} is the reserve risk in short-term disability.

Correlation coefficients were determined either by statistical measures built on historical data on financial markets, or by expert opinions. To read appropriately the matrix, the linear correlation between X_1 and X_2 is the coefficient ρ_{12} (or ρ_{21}), located on line 1 column 2 (of course, the coefficient equals -1 in this case). Finally, our pseudo-correlation matrix looks like

$$G_{101} = \begin{pmatrix} 1 & -1 & -0.28 & 0 & 0.63 & -0.47 & 0.25 & 0.75 & 0 & 0 \\ -1 & 1 & 0.77 & 0.5 & -0.38 & 0.88 & 0.25 & 0.75 & 0.25 & 0.25 \\ -0.28 & 0.77 & 1 & 0.25 & 0.25 & 0.17 & -0.25 & 0.25 & 0.5 & 0.5 \\ 0 & 0.5 & 0.25 & 1 & 0.25 & 0.25 & 0 & 0.5 & 0.5 & 0.5 \\ 0.63 & -0.38 & 0.25 & 0.25 & 1 & 0.83 & 0.25 & 0.75 & 0 & 0 \\ -0.47 & 0.88 & 0.17 & 0.25 & 0.83 & 1 & 0.75 & 0.75 & 0 & 0 \\ 0.25 & 0.25 & -0.25 & 0 & 0.25 & 0.75 & 1 & 0.5 & 0.25 & 0.25 \\ 0.75 & 0.75 & 0.25 & 0.5 & 0.75 & 0.75 & 0.5 & 1 & 0.25 & 0.25 \\ 0 & 0.25 & 0.5 & 0.5 & 0 & 0 & 0.25 & 0.25 & 1 & 0.75 \\ 0 & 0.25 & 0.5 & 0.5 & 0 & 0 & 0.25 & 0.25 & 0.75 & 1 \end{pmatrix}.$$

Table 2 sums up the eigen values of our seven pseudo-correlation matrices: notice that none of the considered matrices is PSD. However, some of them are not very far from having this property.

Example #	Dim.	Notation	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7	λ_8	λ_9	λ_{10}	PSD
1	3	G_{31}	1.90	1.39	-0.29								No
2	3	G_{32}	1.91	1.44	-0.35								No
3	4	G_{41}	2.17	1.27	0.88	-0.32							No
4	4	G_{42}	1.95	1.74	0.96	-0.64							No
5	5	G_{51}	2.80	1.98	1.16	-0.17	-0.77						No
6	5	G_{52}	2.44	1.97	1.67	-0.19	-0.88						No
7	10	G_{101}	3.93	2.72	2.00	1.14	0.74	0.57	0.25	-0.03	-0.55	-0.77	No

Table 2: Eigen values in our examples, before *PSDization*.

3 Selection of an adequate algorithm for PSDization

PSDization algorithms aim to build the nearest PSD matrix to an initial non-PSD matrix, where the notion of “nearest” is detailed in the sequel. This section briefly presents the main families of *PSDization* algorithms, and justifies our choice to work with the Hypersphere (or Rebonato-Jäckel) algorithm. Interesting recent works on this topic and more details can be found in Cutajar et al. (2017).

3.1 Families of algorithms

3.1.1 Alternating Projections

The Alternating Projections (AP) algorithm, introduced by Higham (2002), enables to find the nearest correlation matrix under the W-norm, defined by

$$\forall A \in \mathbb{R}^{n \times n}, \quad \|A\|_W = \|W^{\frac{1}{2}}AW^{\frac{1}{2}}\|_2,$$

where $\forall A \in \mathbb{R}^{n \times n}$, $\|A\|_2^2 = \sum_{(i,j) \in \llbracket 1;n \rrbracket^2} a_{ij}^2 = \text{tr}(AA^T)$. This norm is also called the Frobenius norm, and W is a matrix with positive coefficients.

The AP algorithm corresponds to a linear optimization, projecting alternatively the matrix obtained at each step on two convex closed subsets of the matrix space $\mathbb{R}^{n \times n}$. It enables in particular to show the uniqueness of the solution under this type of norm.

However, the W-norm does not in general correspond to the norm insurers may be interested in. The H-norm, defined by $\forall A \in \mathbb{R}^{n \times n}$, $\|A\|_H^2 = \sum_{(i,j) \in \llbracket 1;n \rrbracket^2} h_{ij}a_{ij}^2$ where $H \in \mathbb{R}^{n \times n}$, offers more flexibility to weight coefficients according to their materiality on SCR^g , or according to the confidence level one has on the coefficients. The W-norm and the H-norm only coincide when W is diagonal ($W = \text{diag}(w_i)_{i \in \llbracket 1;n \rrbracket}$) and H is a rank-1 matrix ($H = [(w_i w_j)^{\frac{1}{2}}]$). Thus, the AP algorithm lacks flexibility in that it does not enable to use another general matrix to weight freely each individual correlation coefficient.

3.1.2 Newton algorithm

Newton algorithms with such applications were introduced by Qi and Sun (2006). They were initially designed for the traditional 2-norm, and are therefore computationally quicker than the Alternating Projections. However, their extension to the most general case (i.e. H-norm) requires optimization techniques such as the Uzawa method. Unfortunately, Uzawa method implies optimization within optimization, and makes the overall algorithm much more time consuming, as well as much more complex to interpret.

3.1.3 Rebonato-Jäckel algorithm

Belonging to the family of Hypersphere algorithms, Rebonato-Jäckel method was initially introduced in Rebonato and Jäckel (1999). Since then, it has been extensively studied in the literature, and many papers have proved its efficiency in reaching a robust solution. Its large success follows from the following theorem (see the proof in Jäckel (2002)): any correlation matrix ρ can be written as

$$\rho = BB^T$$

where the coefficients of the matrix $B \in \mathbb{R}^{n \times n}$ can be written as:

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

In addition, the angular vector θ is unique if :

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

The Hypersphere algorithm thus consists in looking for the solution matrix under the above-mentioned form. It offers several advantages; in particular it is simple to use, easily understandable, and is the most widely used algorithm in the bank and insurance sectors. Moreover, it allows the use of the H-norm and converges fairly quickly. However, its main weakness lies in that it sometimes converges to local minima, and therefore does not guarantee that the output is the nearest *PSDized* correlation matrix. This drawback has to be kept in mind, since it may cause other side effects. Let us mention for instance the fact that the order in which risk factors are considered in the correlation matrix matters, although it should not (see Sections 4 and 5).

3.2 Choice of the algorithm for the rest of the paper

To check for the robustness when performing *PSDization* with these algorithms, one first compares the distances between the initial pseudo-correlation matrix and its *PSDized* version in the three different cases. Of course, the lower this distance the better the algorithm. We use the Frobenius-norm (see 3.1.1), since it is common for all algorithms.

Results about *PSDized* versions for each example are detailed in Appendix B : notice that the *PSDized* correlation matrices are the same when the dimension remains low, whatever the algorithm under consideration. More precisely, the three algorithms give

very similar results with *PSDized* versions of G_{31} , G_{32} , G_{41} , G_{42} , G_{51} , and G_{52} (same coefficients, up to 10^{-4}). On the contrary, the *PSDized* versions of G_{101} are slightly different depending on the algorithm used. As an illustration, we get the following distances:

$$\|S_{101}^{PA} - G_{101}\|_2 = \|S_{101}^N - G_{101}\|_2 = 1.211 \leq \|S_{101} - G_{101}\|_2 = 1.213.$$

In this example, it seems that the two first algorithms give better results. The Rebonato-Jäckel algorithm is likely to have selected a locally-optimal solution. Despite not being the best technique in this particular case, we will use the latter for three main reasons in coming analyses : i) distances do not seem to be significantly different from one method to another, ii) the Rebonato-Jäckel algorithm enables to easily integrate confidence weights to the individual correlation coefficients in practice (through the H-norm), which is a key point, iii) the Rebonato-Jäckel algorithm is fast and easy to interpret. Indeed, experts know that some initial coefficient values can be particularly reliable, or they can anticipate that some of them will have a significant impact on the global SCR (SCR^g).

3.3 One *PSDization* example: the 10-dimensional matrix

Trying to replicate the conditions in which insurers use *PSDization* algorithms, it is clearly more appropriate to consider the H-norm. Indeed, it allows to integrate confidence weights given to correlation coefficients during the *PSDization* process. *PSDization* of G_{101} using the Rebonato-Jäckel algorithm and the weighting matrix H_{101} (see Appendix A) leads to a new correlation matrix S_{101}^H (disclosed in Appendix B), whose eigen values are

$$\lambda \in \{3.70; 2.49; 1.83; 1.05; 0.64; 0.27; 0.02; 0.00; 0.00; 0.00\}.$$

The three last values equal 10^{-5} , which is the lower bound defined in the algorithm. This means that the three negative eigen values of G_{101} (see Table 2) have been replaced by the lowest possible value. One then measures the standardized distance between the initial pseudo-correlation matrix G_{101} and its *PSDized* version S_{101}^H :

$$D_{101}^H = \frac{\|S_{101}^H - G_{101}\|_H}{\|G_{101}\|_H} = 19,7\%.$$

It thus seems that *PSDization* globally had a great impact on the pseudo-correlation matrix. Some coefficients were strongly modified, see for instance ρ_{65} (fictional correlation between equity and spreads). Indeed, ρ_{65} equals 0.83 at the beginning in G_{101} , but is close to 0.54 after *PSDization* in S_{101}^H . Such an example highlights the need for the insurer to check all the modifications, to gain control and monitor her internal model.

4 Sensitivity of the matrices to *PSDization*

In this section, one would like to illustrate how the correlation matrix can be modified when performing *PSDization* by the Rebonato-Jackël algorithm, with toy examples. In particular, we investigate how the coefficients of G_{31} (Appendix A) change during the sole *PSDization*. We also investigate the evolution of individual correlation coefficients when considering other classical operations for practitioners: permutations of some coefficients before *PSDization*, change of matrix dimension (before *PSDization*), or weights given to the correlation coefficients during *PSDization*. In this view, we consider the following initial weighting matrix

$$H_{31}^{init} = \begin{pmatrix} 1 & 0.1 & 0.9 \\ 0.1 & 1 & 0.5 \\ 0.9 & 0.5 & 1 \end{pmatrix}.$$

These operations mainly correspond to the decisions actuaries have to make when developing internal models for risk aggregation. Note that the impact on the capital requirement can be substantially different from the impact in terms of matrix norm, according to the respective importance of the loss marginals. This impact will be studied in Section 5.

4.1 Impact of permutations

To study how permutations of risks defining the correlation matrix impacts the standard *PSDization process*, one first considers the permutation σ such that

$$\begin{pmatrix} 1 & 2 & 3 \\ \downarrow & \downarrow & \downarrow \\ 2 & 3 & 1 \end{pmatrix}.$$

As a result, one obtains the following modifications on the correlation coefficients:

Coefficient	Before <i>PSDization</i>	Direct <i>PSDization</i>	<i>PSDization</i> after permutation
ρ_{12}	-0.9	-0.585	-0.605
ρ_{13}	-0.5	-0.473	-0.476
ρ_{23}	-0.5	-0.437	-0.411

Examining the coefficients, we notice that they are significantly modified: first by the *PSDization* process itself, but also by the permutation of risks. This last result is surprising, since there should be no theoretical impact with this operation. However, because our

algorithm presents local minima issues, the choice of the order of risk factors (arbitrarily made by the insurer) matters when performing *PSDization*.

To figure out more comprehensively the impact of permutations, the best would be to look at the exhaustive list of permutations for a given pseudo-correlation matrix. Recall that a D-dimensional matrix admits D! permutations, and let us consider the example G_{51} . Figure 1 shows the Frobenius distance to the initial matrix for the 5! permutations of G_{51} , knowing that this distance between G_{51} and S_{51}^H initially equals 1.68 without any permutation. The Frobenius norm is clearly not the norm that the algorithm optimizes, but it simply illustrates to which extent the solution matrix S_{51}^H is modified. Two remarks can be made here. First, the distances follow a block pattern which is due to the order of the permutations. Second, the permutations do not always lead to increase the distance to the initial pseudo-correlation matrix.

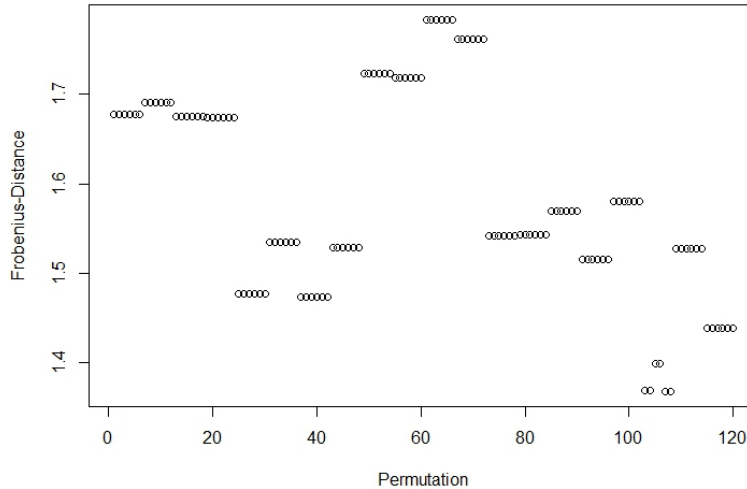


Figure 1: Impact of permutations on the Frobenius norm, in the case of G_{51} and H_{51} .

4.2 Adding a risk: higher the matrix dimension

Another arbitrary element chosen by the actuaries of the company is the number of risk factors to be aggregated. In some cases, it might be necessary to model many risk factors, according to the use of the internal model that is made by the business units (need to model many lines of business when modelling the reserving loss factor for instance). These choices have to be made for risk factors which are not material at the Group level, even if they can be important for the concerned subsidiaries. Nevertheless, these choices will impact the final SCR (SCR^g) through the modification brought to the overall correlation

matrix during *PSDization*. Still based on the same example, let us consider the following case:

$$\begin{pmatrix} 1 & -0.9 & -0.5 \\ -0.9 & 1 & -0.5 \\ -0.5 & -0.5 & 1 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & -0.9 & -0.5 & 0.1 \\ -0.9 & 1 & -0.5 & 0.1 \\ -0.5 & -0.5 & 1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 1 \end{pmatrix}$$

As can be noticed, the correlation is low between the added risk and others.

Coef.	Before <i>PSDization</i>	Direct <i>PSDization</i>	<i>PSDization</i> with higher dimension
ρ_{12}	-0.9	-0.585	-0.577
ρ_{13}	-0.5	-0.473	-0.471
ρ_{23}	-0.5	-0.437	-0.435

Direct *PSDization* obviously gives the same results, whereas correlation coefficients are slightly modified after *PSDization* when introducing the new risk. Changes on these coefficients are hard to anticipate, but it seems that the impact is lower in this case than with permutations. A natural question would be to understand whether the value of the correlation coefficient that was added is key to explain the modifications obtained in the PSDized correlation matrix. Figure 2 shows this impact on the Frobenius norm, with a new risk whose correlation coefficients vary from 0.1 to 1. Results are intuitive: the bigger the coefficients added, the larger the Frobenius norm. Indeed, *PSDization* is a whole process that takes into account every coefficient, including the one that was added.

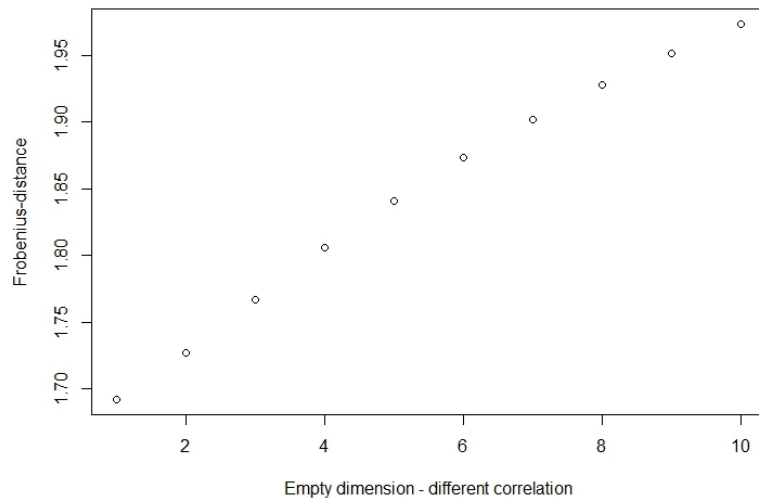


Figure 2: Impact of adding a new dimension on the Frobenius norm (through modified *PSDized* coefficients), with various correlation coefficients corresponding to the new risk.

4.3 Impact of confidence weights

Finally, the choice of the weights associated to the terms of the correlation matrix, which somewhat represents the confidence level given by experts to the individual correlation coefficients, can also have a significant impact on the *PSDized* matrix. To illustrate this, still keeping in mind the example G_{31} with the initial weights listed in H_{31}^{init} , we consider the new following weights:

$$\begin{pmatrix} \omega_{11} & \omega_{12} & \omega_{13} \\ \omega_{21} & \omega_{22} & \omega_{23} \\ \omega_{31} & \omega_{32} & \omega_{33} \end{pmatrix} = \begin{pmatrix} 1 & 0.2 & 0.8 \\ 0.2 & 1 & 0.4 \\ 0.8 & 0.4 & 1 \end{pmatrix}.$$

The *PSDized* matrix now reads

Coef.	Before <i>PSDization</i>	<i>PSDized</i> (initial weights)	<i>PSDization</i> with new weights
ρ_{12}	-0.9	-0.585	-0.642
ρ_{13}	-0.5	-0.473	-0.463
ρ_{23}	-0.5	-0.437	-0.381

As it can be seen, correlation coefficients significantly vary. Notice that the modifications are all the more far from the standard *PSDized* coefficient that the weight is low, which is in line with intuition. Indeed, more weight on some correlation coefficients means more importance in the *PSDization* process, and thus less modification for them (so as to minimize the Frobenius norm). To generalize and understand deeper to which extent the weights could impact the correlation coefficients after *PSDization*, Figure 3 shows the Frobenius norm between the solution S_{51}^H and the initial matrix G_{51} , with weights varying from H_{51} to a limit weighting matrix given by

$$H_{51}^{limit} = \begin{pmatrix} 1 & 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 1 & 0.2 & 0.5 & 0.5 \\ 0.5 & 0.2 & 1 & 0.6 & 0.6 \\ 0.5 & 0.5 & 0.6 & 1 & 0.1 \\ 0.5 & 0.5 & 0.6 & 0.5 & 1 \end{pmatrix}.$$

This limit weighting matrix is used for illustration purposes only. It corresponds to increase linearly by +0.4 the lowest weights of the matrix H_{51} , while decreasing highest weights by the same factor. Figure 3 shows that the Frobenius norm decreases as the weighting matrix is distorted towards the limit H_{51}^{limit} , and a closer analysis reveals that this phenomenon is mainly due to the correlation coefficients ρ_{24} , ρ_{25} , ρ_{34} and ρ_{35}

of G_{51} (and their transposed coefficients) which are much less modified than with the initial weighting matrix since their weights are significantly increased: from 0.1 to 0.5 and respectively from 0.2 to 0.6.

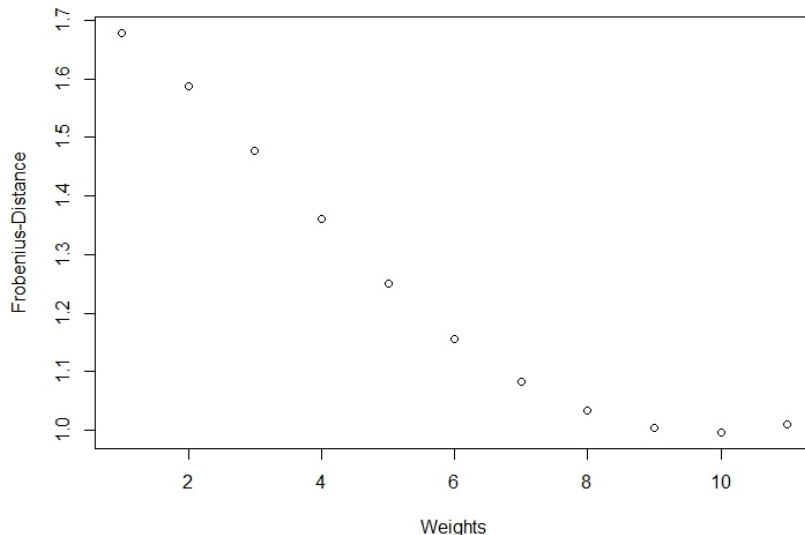


Figure 3: Impact of weights on the Frobenius norm, in the case of the example G_{51} .

4.4 Summary

To put it in a nutshell, Figure 4 shows the impact of permutations and weights (the two most prominent operations) on the Frobenius norm, in the conditions stated above. It shows that the impact on the Frobenius norm is more important when weights vary than when the order of risk factors is changed. After these primary illustrations, we now move to the analysis of such transformations on the capital requirement, SCR^g .

5 Analysis of SCR sensitivity

One has just highlighted the importance of *PSDization* on the final correlation matrix to be used to assess the insurer's own funds requirement. The correlation coefficients chosen by the experts, or even those defined by statistical means can be significantly modified. The aim of this section is to provide with some real-life sensitivities concerning the computation of the global SCR thanks to internal models. One would like to see SCR^g as a function of the main parameters in the actuary's hand.

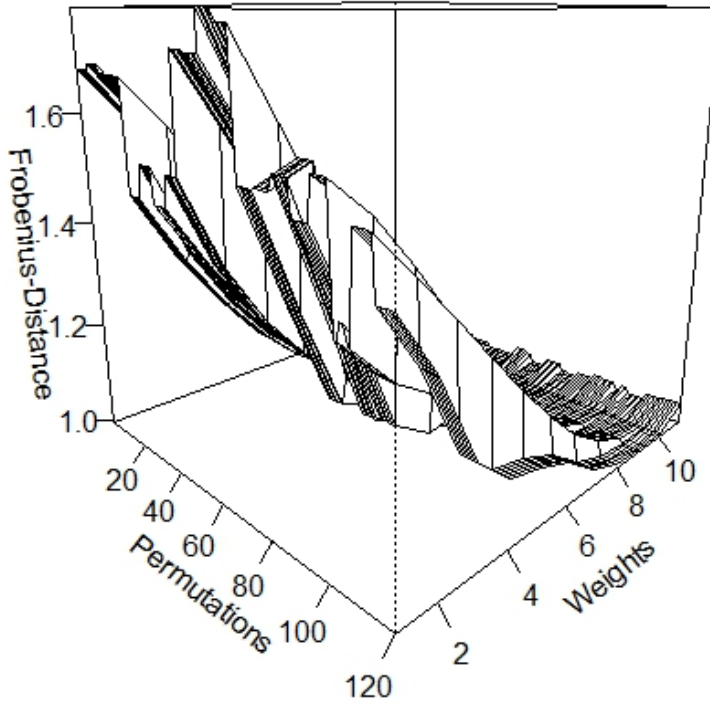


Figure 4: 3D plot showing the impact of permutations and weights on the Frobenius norm, with G_{51} and H_{51} .

To make it, one applies genetic algorithms to find a range $[min, max]^2$ of values to which the SCR belongs; given a copula, realizations of risk factors, and proxy functions (more details further). The implemented algorithms are disclosed in Appendices D.1 and D.2, respectively for the case of permutations and weights. They correspond to an adaptation of the Rebonato-Jackël algorithm that incorporate these operations. At the end, the range is obtained for a given pseudo-correlation matrix G , a given weighting matrix H , and possibly a given permutation σ . Hence we want to evaluate the function g such that $g : (G, H, \sigma) \rightarrow g(G, H, \sigma) = SCR(PSD_H(\sigma.G))$, where $\sigma.G$ stands for the effect of σ on the risk factors represented in G , and $PSD_H(G)$ represents the nearest (from G) PSD matrix obtained using the Rebonato-Jackël algorithm with weights H . The considered permutation σ is the one presented at the beginning of Section 4.1.

²Recall that the genetic algorithms must be seen as “clever” sensitivities rather than fully convergent optimization algorithms. There is no guarantee that the convergence to a minimum is obtained.

5.1 Loss factors or risk factors?

In full generality, it is a very hard task to estimate the loss generated by the occurrence of some given risk. Much easier is to describe the behavior of risk factors, through marginal distributions. For instance if the interest rates rise, the potential loss for the insurer depends on impacts on both assets (e.g. value of obligations drops) and liabilities (contract credited rates may vary, which should modify expected lapse rates). To compute the loss associated to the variation of some risk factors, one thus needs a (very) complex transformation. Practically speaking, and to save computation time, simple functions (polynomial form) approximate these losses. However, the insurer can sometimes directly evaluate the loss related to one given loss factor: this is the case for example when considering the reserve risk, which can be modeled by classical statistical methods (bootstrap). The insurer's total loss, P , thus reads

$$P = \mathbb{1}_{\mathcal{R} \neq \emptyset} f((X_i)_{i \in \mathcal{R}}) + \mathbb{1}_{\mathcal{P} \neq \emptyset} \sum_{i \in \mathcal{P}} X_j,$$

where f is a given (proxy) function, X_i and X_j are random variables, \mathcal{P} is the set of loss factors and \mathcal{R} is the set of risk factors.

For our coming analysis, Table 3 gives the different functional forms depending on the risk dimension and the risk factors X_i . For the sake of simplicity, one considers that all our marginals (X_i and X_j) follow the same distribution, but with different parameters. This common distribution is lognormal $\mathcal{LN}(\mu, \lambda)$, since it is widely used in insurance for prudential reasons. Table 4 sums up the parameters involved in the eight different cases under study: vectors X_k^a (where k refers to the dimension of the vector) will be used to compute the global loss in the case of loss factors aggregation (meaning that $\mathcal{R} = \emptyset$), whereas vectors X_k^b will be the input of proxy functions defined in Table 3 for risk factors aggregation ($\mathcal{P} = \emptyset$). We distinguish these two configurations to see whether taking into account proxy functions gives very different SCR sensibilities as compared to only aggregate risk factors.

5.2 Variance-covariance or copula approach, pros and cons

Except *PSDization* step itself that generates different results, another important choice is the aggregation approach. Here, one would like to detail the reasons for choosing one of them (i.e. copula or variance-covariance). Let us consider the simplest framework: the loss P only depends on loss factors (no need to apply proxy functions that link risk factors to loss factors). It is then possible to model this loss as following: $P = \sum_{i \in \mathcal{P}} X_i$.

Dimension	Functional form under consideration
3	$P = f((X_i)_{i \in [1,3]}) = 0.5X_1^2 + 2X_2^4 + 0.3X_3 + 10X_1X_2$
4	$P = f((X_i)_{i \in [1,4]}) = 5X_1 + 0.02X_2^2 + 2000X_3 + 5X_4^{3/2}$
5	$P = f((X_i)_{i \in [1,5]}) = X_1 + 0.1X_2^2 + 50X_3 + X_4^{1/2} - 1.2X_5$
10	$P = f((X_i)_{i \in [1,10]}) = 0.5X_1^3 + 0.4X_2^3 + 3X_3^2 + 2X_4^2 + \sum_{i=5}^7 X_i^2 + 0.5X_8^2 + \sum_{i=9}^{10} X_i$

Table 3: Polynomials used as proxy functions to get the overall insurer's loss.

Individual loss factors have to be modeled and estimated by the actuaries for internal models, or come from standardized shocks if using the Standard Formula. Fortunately, it is likely that extreme events corresponding to the 99.5th percentile of every loss factors do not occur at the same time: there is thus a mitigation effect, that generally yields to

$$q_{99.5\%}(P) \leq \sum_{i \in \mathcal{P}} q_{99.5\%}(X_i).$$

As already mentioned in Section 1, the regulation states that the variance-covariance approach can be used to aggregate risks, with the given correlation matrices. This method has some advantages, but also some drawbacks (Embrechts et al. (2013)). Of course, it is the easiest way to aggregate risks: the formula is quickly implemented (which allows to compute sensitivities without too much effort), and easy to understand. However it does not provide with the entire distribution of the aggregated loss, knowing that the insurer is sometimes interested in other risk measures than the unique 99.5th percentile.

Name	Dim.	Parameters	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}
X_3^a	3	(μ, λ)	(6,0.5)	(5,0.5)	(4,0.1)							
		$Q_{99.5}$	1463	538	71							
X_3^b	3	(μ, λ)	(6,0.8)	(3,0.1)	(12,2)							
		$Q_{99.5}$	3167	26	28110637							
X_4^a	4	(μ, λ)	(6,0.5)	(5,0.5)	(4,0.1)	(3,0.9)						
		$Q_{99.5}$	1463	538	71	204						
X_4^b	4	(μ, λ)	(1,5)	(2,4)	(3,3)	(1,4)						
		$Q_{99.5}$	1065704	220426	45592	81090						
X_5^a	5	(μ, λ)	(6,0.5)	(5,0.5)	(4,0.1)	(4,0.9)	(3,0.3)					
		$Q_{99.5}$	1463	538	71	555	43					
X_5^b	5	(μ, λ)	(1,3)	(2,2)	(1,1)	(2,4)	(1,3)					
		$Q_{99.5}$	6170	1276	36	220426	6170					
X_{10}^a	10	(μ, λ)	(6,0.5)	(5,0.5)	(5,0.2)	(6,0.1)	(4,0.3)	(5,0.6)	(5,0.5)	(5,0.4)	(5,0.3)	(6,0.3)
		$Q_{99.5}$	1463	538	248	522	118	696	538	416	321	874
X_{10}^b	10	(μ, λ)	(1,2,2)	(2,1,5)	(0,3,2,5)	(2,2)	(1,5,2)	(3,5,1)	(1,5,2)	(2,5,2)	(5,2)	(6,2)
		$Q_{99.5}$	573	352	845	1276	774	435	774	2104	25633	69679

Table 4: Marginals for each risk factor and corresponding 99.5th percentile.

Moreover, this approach is not adequate for modeling non-linear correlations, which is often the case when considering the tails of loss distributions. It means that it is very tricky to calibrate the correlation matrix so as to ensure that we can effectively estimate the 99.5th percentile of the aggregated loss. Finally, the variance-covariance approach is too restrictive since it does not allow to correlate risk factors, but only correlate losses. This makes the interpretation of scenarios generating a huge aggregated loss very difficult.

For all these reasons, internal models are generally developed using copulas: they enable to simulate a large number of joint replications of the risk factors, before applying proxy functions (most of time). By this technique, insurers obtain the full distribution of P , and thus a richer information among which the quantile of interest (Embrechts and Puccetti (2010), Lescouret and Robert (2006)). Very familiar copulas in the insurance industry (Gaussian and Student copulas) are parameterized by the linear correlation matrix, whose marginals are the risk and loss factors. This is linked to the main property of copulas: they allow to define the correlation structure and the marginals separately. For example, aggregation with the Gaussian copula can be simulated with the following steps:

- simulation of the marginals stand-alone (stored in the vector $X \in \mathbb{R}^{n \times B}$, where n stands for the number of risk factors and B the number of random samples);
- simulation of a Gaussian vector Y through the expression $Y = TZ$, where T represents the Choleski decomposition of the correlation matrix $\rho = (\rho_{ij})_{(i,j) \in \llbracket 1, n \rrbracket}$ and Z is an independent Gaussian vector of size n ;
- ordering X in the same order as Y to ensure that $\forall j \in \{1, \dots, B\}, \forall i \in \{1, \dots, n\}, q(X_{ij}) = q(Y_{ij})$ ($q(x)$ stands for the quantile corresponding to x).

5.3 Results using a simplified internal model

Applications presented hereafter were designed to consider a wide range of operational situations in which the insurer aims to estimate its global SCR, i.e. SCR^g .

For a given PSD correlation matrix S_{xy}^H , and given values for the vector of risk factors (simulated with lognormal distributions), one performs $Q = 131072 = 2^{17}$ simulations for the aggregation of risk factors (dependence structure). As a matter of fact, there are two sources of uncertainty explaining the variation of SCR^g values ($SCR^g = (SCR_q^g)_{q=1, \dots, Q}$). First, the genetic algorithm itself is likely to have reached different local solutions depending on the simulation. Second, the simulation of the Gaussian or Student vectors

to model the correlation through copulas may change. In order to focus the study on correlation, it must be noted that marginals were simulated initially and then kept fixed. Roughly, it can be assumed that the confidence interval of the global SCR is similar to that of a Gaussian distribution ($SCR_q^g \sim SCR^g \sim \mathcal{N}(m, \sigma)$) because of the Central Limit Theorem and of the independence of the simulations of each SCR, i.e.

$$\mathbb{P}(|SCR^g - m| < 1.65 \sigma) = 90\%,$$

where σ stands for the standard deviation of SCR^g , and m its mean. Of course, the estimation of these parameters is made simple using their empirical counterparts, denoted by $\hat{m} = (1/Q) \sum_q SCR_q^g$ and $\hat{\sigma}^2 = (1/(Q-1)) \sum_q (SCR_q^g - \hat{m})^2$. Table 5 summarizes the estimated quantities for each case in our framework.

Then, we study the impact of our transformations (permutation, weights varying, and higher dimension) as compared to this standard deviation $\hat{\sigma}$. More precisely, we consider one operation, perform the same number of simulations, and store the minimum and maximum values of the vector $(SCR_q^g)_{q=1, \dots, Q}$. This way, it is possible to define a normalized range (NR) for these values, whose expression follows

$$NR = \frac{\max(SCR^g) - \min(SCR^g)}{\hat{m}}.$$

If NR is lower than $(2 \times 1.65 \times \hat{\sigma})$, the transformation is said to have a limited impact on SCR^g . Otherwise, it is considered as a significant impact. The worst cases correspond to situations where NR is greater than $(2 \times 2.89 \times \hat{\sigma})$. The multiplier 2 enables to take into account the fact that there are two sources of uncertainty (genetic algorithm and simulated dependence structure). All the results are stored in Table 6, where column ‘Imp.’ describes the strength of the impact of the transformation under study : ‘0’ refers to a limited impact

Example	Dim. k	Gaussian copula				Student copula (3 d.f.)			
		Loss factors X_k^a		Risk factors X_k^b		Loss factors X_k^a		Risk factors X_k^b	
		\hat{m}	$\hat{\sigma}$	\hat{m}	$\hat{\sigma}$	\hat{m}	$\hat{\sigma}$	\hat{m}	$\hat{\sigma}$
S_{31}^H	3	2566	0.06%	12119354	0.16%	2576	0.11%	12473296	0.32%
S_{32}^H	3	2694	0.08%	13808174	0.19%	2813	0.20%	14641263	0.86%
S_{41}^H	4	3319	0.10%	3532563	0.10%	3379	0.16%	3555426	0.33%
S_{42}^H	4	2809	0.14%	3563414	0.31%	2997	0.29%	3578124	0.35%
S_{51}^H	5	2605	0.09%	9359	0.68%	2677	0.18%	9411	0.59%
S_{52}^H	5	3139	0.13%	8303	0.28%	3260	0.32%	8531	0.99%
S_{101}^H	10	5825	0.26%	4174500	0.73%	6717	0.31%	4234635	1.14%

Table 5: Mean and standard deviation of the global SCR, SCR^g ($Q = 2^{17}$ simulations).

($NR < 2 \times 1.65\hat{\sigma}$), ‘+’ means a significant impact ($NR \in [2 \times 1.65\hat{\sigma}, 2 \times 2.89\hat{\sigma}]$), and ‘++’ a very strong impact ($NR > 2 \times 2.89\hat{\sigma}$).

5.3.1 Impact of permutations on the global SCR

On the 14 examples under study (7 pseudo-correlation matrices G_{xy} , times Gaussian or Student copula), the permutation systematically causes a very strong impact on the global SCR. This change can represent up to 6.7% in practice, although it should have no theoretical impact. This is mainly due to the *PSDization* process that leads to select different local minima after permutation is made. This highlights two phenomenons: the need to control the bias coming from the initial choice of the insurer concerning the order of risk factors, and the need to initially define PSD correlation matrices (revisiting the experts’ opinions, and identifying incoherent correlation submatrices).

To have a more comprehensive view of this impact, Figure 5 illustrates it on the total loss distribution (rather than the sole 99.5th percentile), with G_{52} , H_{52} and considering the aggregation of loss factors. The red curve corresponds to the loss distribution after applying the permutation ($1 \mapsto 5; 4 \mapsto 1; 5 \mapsto 4$) to G_{52} in the case of a Gaussian copula, whereas the blue one corresponds to the permutation ($2 \mapsto 3; 3 \mapsto 4; 4 \mapsto 5; 5 \mapsto 2$).

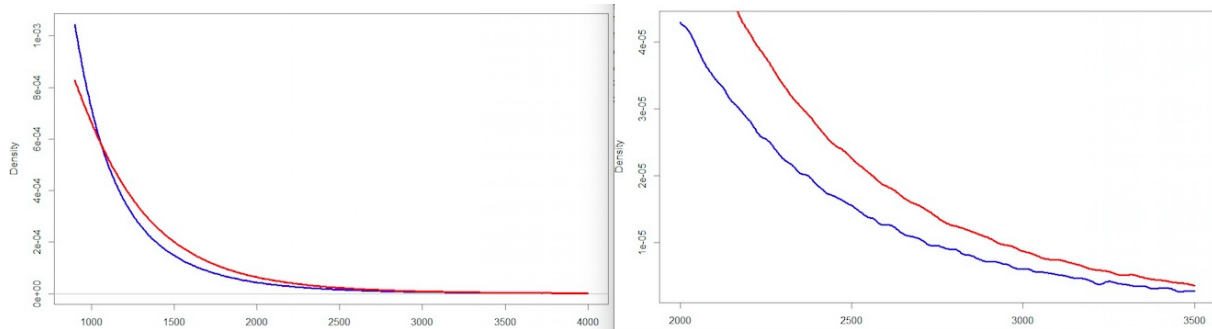


Figure 5: Illustration of the impact of the permutation on the loss distribution, with a focus on the right on the area around the capital requirement (99.5 quantile)

5.3.2 Impact of the modification of weights on SCR

Our sensitivities relate to weighting coefficients varying in a given range. This range is defined by weights between $H^{min} = [0]_{(i,j) \in [1,n]^2}$ and $H^{max} = [1]_{(i,j) \in [1,n]^2}$. This sensitivity is denoted by ‘W Sensi1’ in Table 6. On the 28 examples analyzed here (7 pseudo-correlation matrices G_{xy} , times (Gaussian or Student, copula), times (risk or loss factors)),

k	Operation	Gaussian copula						Student copula										
		Loss factors X_k^a			Risk factors X_k^b			Loss factors X_k^a			Risk factors X_k^b							
		SCR^a min	SCR^a max	NR	Imp.	SCR^b min	SCR^b max	NR	Imp.	SCR^a min	SCR^a max	NR	Imp.	SCR^b min	SCR^b max	NR	Imp.	
G_{31}	3	Permut.	2562.64	2572.78	0.4%	0.4%	++	++	++	12081399	12120187	0.32%	0	0	2565.31	2589.81	0.95%	++
		W Sens1	2564.11	2577.68	0.53%	0.53%	++	++	++	12072651	12135251	0.52%	0	0	2575.71	2608.43	1.27%	++
		W Sens2	2560.19	2565.96	0.23%	0.23%	+	+	+	initial : final :	2575.05	2575.55	0.41%	+	+	2575.05	2575.55	0.41%
G_{32}	3	Permut.	2593.96	2568.93	0.99%	0.99%	++	++	++	12132660	12249386	0.96%	++	++	2576.77	2578.86	0.08%	0
		W Sens1	2635.26	2758.19	4.66%	4.66%	++	++	++	12447404	12829070	3.07%	++	++	2726.01	2891.99	6.09%	++
		W Sens2	2627.97	2665.76	1.44%	1.44%	++	++	++	12765514	12971883	1.62%	++	++	2705.99	2749.83	1.62%	++
G_{41}	4	Permut.	3293.26	3356.76	1.93%	1.93%	++	++	++	13127451	13249432	0.93%	+	+	2802.37	2805.6	0.12%	0
		W Sens1	3121.05	3232.06	3.56%	3.56%	++	++	++	3516332	3532352	0.46%	+	+	3288.28	3325.72	1.14%	++
		W Sens2	3297.22	3320.04	0.69%	0.69%	++	++	++	3520161	3532792	0.36%	+	+	3360.15	3378.96	0.56%	+
G_{42}	4	Permut.	2759.38	2824.89	2.37%	2.37%	++	++	++	3529310	3534685	0.15%	0	0	2909.99	3011.72	3.50%	++
		W Sens1	2814.42	2905.75	3.25%	3.25%	++	++	++	3519944	3551629	0.90%	0	0	2995.14	3060.68	2.19%	++
		W Sens2	2803.70	2828.836	0.90%	0.90%	++	++	++	3540453	3565096	0.70%	0	0	2983.73	2996.47	2.14%	++
G_{51}	5	Permut.	2607.65	2627.15	0.75%	0.75%	++	++	++	3564593	3561905	0.08%	0	0	2996.24	2999.27	0.10%	0
		W Sens1	2637.39	2662.24	0.94%	0.94%	++	++	++	8541.412	8931.55	4.57%	++	++	2658.80	2700.28	1.56%	++
		W Sens2	2607.38	2616.21	0.34%	0.34%	+	+	+	9201.858	9328.433	1.38%	0	0	2697.89	2745.08	1.75%	++
G_{52}	5	Permut.	2952.79	3152.47	6.76%	6.76%	++	++	++	9333.50	9336.61	0.03%	0	0	2670.86	2665.32	0.21%	0
		W Sens1	2929.41	3088.92	5.45%	5.45%	++	++	++	8230.70	8649.88	5.09%	++	++	3113.23	3300.16	6.00%	++
		W Sens2	3120.47	3141.46	0.67%	0.67%	+	+	+	8180.51	8323.61	1.75%	++	++	3100.95	3248.25	4.75%	++
G_{101}	10	Permut.	5767.08	5887.43	2.09%	2.09%	++	++	++	4096899	4166343	1.69%	0	0	6657.96	6837.03	2.69%	++
		W Sens1	5716.66	5864.03	2.58%	2.58%	++	++	++	4086983	4181583	2.31%	0	0	6631.49	6737.25	1.60%	+
		W Sens2	5765.11	5816.17	0.89%	0.89%	+	+	+	4226948	4192248	0.82%	0	0	6633.39	6728.64	1.44%	+
	dim+1	5800.75	5811.27	0.18%	0.18%	0	0	0	6701.83	6753.39	0.76%	0	0	4224386	4362900	3.27%	0	

Table 6: Sensitivities of the global SCR to transformations on the correlation matrix (permutation, two different cases for weighting the correlation coefficients, and addition of a new risk); depending on copula, and risk factors / loss factors aggregation.

17 cases lead to a very strong impact on SCR^g . The range $[SCR^{min}; SCR^{max}]$ can represent more than 5.4% of the SCR, which is really huge in practice.

The same analysis with stronger constraints (weights belonging to an interval of width 0.2 around the initial weights, i.e. $H^{min} = H - [0.1]_{(i,j) \in [1,n]^2}$ and $H^{max} = H + [0.1]_{(i,j) \in [1,n]^2}$, see ‘W Sensi2’ in Table 6) shows that on 28 examples, 7 cases lead to a very significant impact on the final SCR, with a range likely to represent more than 4.7% of the SCR! Once more, this shows the necessity to properly define correlation coefficients at the very beginning.

5.3.3 Impact of adding a dimension to the correlation matrix

In practice, the insurer’s global loss often incorporates some negligible loss factor. In the simple case where there are only loss factors affecting the global loss, it means that

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

where ϵ thus tends to 0. The limit case would be $\epsilon = 0$, which means that the $(n+1)^{th}$ risk factor would have no impact on the insurer’s loss, but still plays a role through its presence in the correlation matrix and its impact in the *PSDization* process. The correlation between this risk factor and others is fixed to 0.1 (as in Section 4.2). We measure SCR^g value before and after adding this dimension.

On the 28 examples analyzed, almost one third (9 cases exactly) leads to a statistically significant impact on the final SCR (strong or very strong impact on SCR^g). However, except one particular case involving an impact value around 6%, most of the impacts seem to be lower than with other operations. Once again, it is important to realize that this transformation should have no theoretical impact. Of course, it suggests that it would be worth conducting deeper analyses on this aspect, especially about adding more than one dimension and changing the correlation coefficients of the added risk factor.

Concluding remarks and discussion

Insurers using internal models, as well as supervisors, are legitimate to ask themselves about the robustness of their *PSDization* algorithm. Our study shows and highlights the importance of *PSDization* through quantified answers to very practical questions on a series of real-life examples. On the 98 ($3 \cdot 28 + 14$) examples based on various configurations (different copulas and ways to consider risks, see Table 6), approximately one half

(exactly 47) generates significant impacts on the global SCR (up to 6%) when studying sensitivities to our three tuning parameters (weights given to individual correlation coefficients, permutations, and addition of a fictive business line). Knowing that these transformations are either theoretically neutral, or should not lead to big effects on the global capital requirement, this underlines that practitioners' choices (risk managers, actuaries) are fundamental when performing risk aggregation in internal models. A strong control of *PSDization* by supervisors thus makes sense. Proxy functions do not seem to change conclusions here: SCR^g sensitivity is similar when considering only loss factors.

The following best practices were identified: i) develop a sound internal control framework on both the triggers generating negative eigenvalues (e.g. expert judgments) and the *PSDization* step itself, ii) assess the need for adding a new risk (e.g. new business line) in terms of its impact on the correlation matrix and thus on the global SCR. On the former point, independent validations and systematic reviews of the modifications brought to the correlation matrix by the algorithm should be analyzed, and a wide number of sensitivities has to be implemented to challenge the results. Concerning the dimension of the risk matrix, there seems to be a trade-off to find: adding business lines allows to increase granularity when describing the correlation between risks, but tend to cause more disturbance on the individual correlation coefficients during *PSDization*. As usual, the best choice lies in an intermediate dimension for the risk matrix.

Finally, this work could be extended in several ways, among which the definition of algebraic tests to detect in advance inconsistencies in the experts' choices; and a deeper understanding of the permutations leading to the minimum or maximum values of the SCR. In particular, if these permutations show some similar features, it would be possible to define best practices when ordering risk factors.

Acknowledgment

This work is partially supported by BNP Paribas Cardif, through the Research Chair DAMI. We also would like to thank two researchers for useful and constructive comments, Hansjörg Albrecher and Léonard Vincent.

References

- Bernard, C., Jiang, X., and Wang, R. (2014). Risk aggregation with dependence uncertainty. *Insurance: Mathematics and Economics*, 54(1):93–108.
- Cheung, K. and Vanduffel, S. (2013). Bounds for Sums of Random Variables when the Marginal Distributions and the Variance of the Sum are Given. *Scandinavian Actuarial Journal*, 13(2):103–118.
- Cifuentes, A. and Charlin, V. (2016). Operational risk and the Solvency II capital aggregation formula: implications of the hidden correlation assumptions. *Journal of Operational Risk*, 11(4):23–33.
- Cutajar, S., Smigoc, H., and O’Hagan, A. (2017). Actuarial Risk Matrices: The Nearest Positive Semidefinite Matrix Problem. *North American Actuarial Journal*, 21(4):552–564.
- Denuit, M., Genest, C., and Marceau, E. (1999). Stochastic bounds on sums of dependent risks. *Insurance: Mathematics and Economics*, 25(1):85–104.
- Devineau, L. and Loisel, S. (2009). Risk aggregation in Solvency II: How to converge the approaches of the internal models and those of the standard formula? *Bull. Français d’Actuariat*, 9(18):107–145.
- Embrechts, P. and Puccetti, G. (2010). *in Copula Theory and Its Applications*. Wiley Finance. Springer.
- Embrechts, P., Puccetti, G., and Rüschendorf, L. (2013). Model uncertainty and VaR aggregation. *Journal of Banking & Finance*, 37(8):2750–2764.
- Filipovic, D. (2009). Multi-level Risk Aggregation. *ASTIN Bulletin*, 39(2):565–575.
- Georgescu, D., Higham, N., and Peters, G. (2017). Explicit Solutions to Correlation Matrix Completion Problems, with an Application to Risk Management and Insurance. Technical report, Univ. of Manchester, Manchester Institute for Mathematical Sciences.
- Higham, N. (2002). Computing the nearest correlation matrix - a problem from finance. *IMA Journal of Numerical Analysis*, 22(3):329–343.
- Jäckel, P. (2002). *Monte Carlo methods in finance*. Wiley Finance. Wiley and sons.

Lescourret, L. and Robert, C. (2006). Extreme dependence of multivariate catastrophic losses. *Scandinavian Actuarial Journal*, 4:203–225.

Qi, H. and Sun, D. (2006). A quadratically convergent Newton method for computing the nearest correlation matrix. *SIAM Journal on Matrix Analysis and Applications*, 28(2):360–385.

Rebonato, R. and Jäckel, P. (1999). The most general methodology to create a valid correlation matrix for risk management and option pricing purposes.

A Pseudo-correlation matrices under study

Let us present the pseudo-correlation matrices, but also the weighting matrices coming from expert judgments to be taking into account during *PSDization*.

Example #1:

$$G_{31} = \begin{pmatrix} 1 & -0.9 & -0.5 \\ -0.9 & 1 & -0.5 \\ -0.5 & -0.5 & 1 \end{pmatrix} \quad H_{31} = \begin{pmatrix} 1 & 0.9 & 0.8 \\ 0.9 & 1 & 0.1 \\ 0.8 & 0.1 & 1 \end{pmatrix}$$

Example #2:

$$G_{32} = \begin{pmatrix} 1 & -0.6 & 0.5 \\ -0.6 & 1 & 0.9 \\ 0.5 & 0.9 & 1 \end{pmatrix} \quad H_{32} = \begin{pmatrix} 1 & 0.1 & 0.1 \\ 0.1 & 1 & 0.9 \\ 0.1 & 0.9 & 1 \end{pmatrix}.$$

Example #3:

$$G_{41} = \begin{pmatrix} 1 & 0.9 & -0.5 & -0.7 \\ 0.9 & 1 & 0.2 & 0.1 \\ -0.5 & 0.2 & 1 & 0.1 \\ -0.7 & 0.1 & 0.1 & 1 \end{pmatrix} \quad H_{41} = \begin{pmatrix} 1 & 0.9 & 0.2 & 0.9 \\ 0.9 & 1 & 0.2 & 0.1 \\ 0.2 & 0.2 & 1 & 0.1 \\ 0.9 & 0.1 & 0.1 & 1 \end{pmatrix}$$

Example #4:

$$G_{42} = \begin{pmatrix} 1 & 0.1 & -0.8 & 0.75 \\ 0.1 & 1 & 0.2 & 0.1 \\ -0.8 & 0.2 & 1 & 0.9 \\ 0.75 & 0.1 & 0.9 & 1 \end{pmatrix} \quad H_{42} = \begin{pmatrix} 1 & 0.1 & 0.9 & 0.2 \\ 0.1 & 1 & 0.1 & 0.1 \\ 0.9 & 0.1 & 1 & 0.9 \\ 0.2 & 0.1 & 0.9 & 1 \end{pmatrix}.$$

Example #5:

$$G_{51} = \begin{pmatrix} 1 & -0.8 & -0.9 & -0.9 & 0.2 \\ -0.8 & 1 & 0.9 & -0.7 & 0.6 \\ -0.9 & 0.9 & 1 & 0.2 & -0.6 \\ -0.9 & -0.7 & 0.2 & 1 & -0.1 \\ 0.2 & 0.6 & -0.6 & -0.1 & 1 \end{pmatrix} \quad H_{51} = \begin{pmatrix} 1 & 0.9 & 0.1 & 0.9 & 0.1 \\ 0.9 & 1 & 0.6 & 0.1 & 0.1 \\ 0.1 & 0.6 & 1 & 0.2 & 0.2 \\ 0.9 & 0.1 & 0.2 & 1 & 0.1 \\ 0.1 & 0.1 & 0.2 & 0.1 & 1 \end{pmatrix}$$

Example #6:

$$G_{52} = \begin{pmatrix} 1 & 0.7 & -0.8 & -0.8 & 0.2 \\ 0.7 & 1 & 0.8 & -0.6 & 0.6 \\ -0.8 & 0.8 & 1 & 0.2 & -0.6 \\ -0.8 & -0.6 & 0.2 & 1 & 0.8 \\ 0.2 & 0.6 & -0.6 & 0.8 & 1 \end{pmatrix} \quad H_{52} = \begin{pmatrix} 1 & 0.1 & 0.9 & 0.9 & 0.1 \\ 0.1 & 1 & 0.1 & 0.9 & 0.1 \\ 0.9 & 0.1 & 1 & 0.1 & 0.9 \\ 0.9 & 0.9 & 0.1 & 1 & 0.1 \\ 0.1 & 0.1 & 0.9 & 0.1 & 1 \end{pmatrix}.$$

Example #7:

$$G_{101} = \begin{pmatrix} 1 & -1 & -0.28 & 0 & 0.63 & -0.47 & 0.25 & 0.75 & 0 & 0 \\ -1 & 1 & 0.77 & 0.5 & -0.38 & 0.88 & 0.25 & 0.75 & 0.25 & 0.25 \\ -0.28 & 0.77 & 1 & 0.25 & 0.25 & 0.17 & -0.25 & 0.25 & 0.5 & 0.5 \\ 0 & 0.5 & 0.25 & 1 & 0.25 & 0.25 & 0 & 0.5 & 0.5 & 0.5 \\ 0.63 & -0.38 & 0.25 & 0.25 & 1 & 0.83 & 0.25 & 0.75 & 0 & 0 \\ -0.47 & 0.88 & 0.17 & 0.25 & 0.83 & 1 & 0.75 & 0.75 & 0 & 0 \\ 0.25 & 0.25 & -0.25 & 0 & 0.25 & 0.75 & 1 & 0.5 & 0.25 & 0.25 \\ 0.75 & 0.75 & 0.25 & 0.5 & 0.75 & 0.75 & 0.5 & 1 & 0.25 & 0.25 \\ 0 & 0.25 & 0.5 & 0.5 & 0 & 0 & 0.25 & 0.25 & 1 & 0.75 \\ 0 & 0.25 & 0.5 & 0.5 & 0 & 0 & 0.25 & 0.25 & 0.75 & 1 \end{pmatrix}$$

$$H_{101} = \begin{pmatrix} 1 & 0.3 & 0.9 & 0.3 & 0.9 & 0.9 & 0.3 & 0.3 & 0.3 & 0.3 \\ 0.3 & 1 & 0.9 & 0.3 & 0.9 & 0.9 & 0.3 & 0.3 & 0.3 & 0.3 \\ 0.9 & 0.9 & 1 & 0.3 & 0.3 & 0.9 & 0.3 & 0.3 & 0.3 & 0.3 \\ 0.3 & 0.3 & 0.3 & 1 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 \\ 0.9 & 0.9 & 0.3 & 0.3 & 1 & 0.9 & 0.3 & 0.3 & 0.3 & 0.3 \\ 0.9 & 0.9 & 0.9 & 0.3 & 0.9 & 1 & 0.3 & 0.3 & 0.3 & 0.3 \\ 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 1 & 0.3 & 0.3 & 0.3 \\ 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 1 & 0.3 & 0.3 \\ 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 1 & 0.3 \\ 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 1 \end{pmatrix}.$$

B *PSDized* matrices, without weighting coefficients

We recall here that except for the highest dimension under consideration (dimension 10), the three algorithms give the same solution.

Examples #1 and #2: $S_{31}^{PA} = S_{31}^N = S_{31}$, and $S_{32}^{PA} = S_{32}^N = S_{32}$, where

$$S_{31} = \begin{pmatrix} 1 & -0.725 & -0.371 \\ -0.725 & 1 & -0.370 \\ -0.371 & -0.370 & 1 \end{pmatrix}, \quad S_{32} = \begin{pmatrix} 1 & -0.436 & 0.343 \\ -0.436 & 1 & 0.695 \\ 0.343 & 0.695 & 1 \end{pmatrix}.$$

Examples #3 and #4: $S_{41}^{PA} = S_{41}^N = S_{41}$, and $S_{42}^{PA} = S_{42}^N = S_{42}$, where

$$S_{41} = \begin{pmatrix} 1 & 0.711 & -0.398 & -0.573 \\ 0.711 & 1 & 0.122 & 0.003 \\ -0.398 & 0.122 & 1 & 0.152 \\ -0.573 & 0.003 & 0.152 & 1 \end{pmatrix}, \quad S_{42} = \begin{pmatrix} 1 & 0.068 & -0.481 & 0.444 \\ 0.068 & 1 & 0.165 & 0.133 \\ -0.481 & 0.165 & 1 & 0.566 \\ 0.444 & 0.133 & 0.566 & 1 \end{pmatrix}.$$

Examples #5 and #6: $S_{51}^{PA} = S_{51}^N = S_{51}$, and $S_{52}^{PA} = S_{52}^N = S_{52}$, where

$$S_{51} = \begin{pmatrix} 1 & -0.508 & -0.830 & -0.585 & 0.140 \\ -0.508 & 1 & 0.651 & -0.360 & 0.350 \\ -0.830 & 0.651 & 1 & 0.148 & -0.409 \\ -0.585 & -0.360 & 0.148 & 1 & -0.221 \\ 0.140 & 0.350 & -0.409 & -0.221 & 1 \end{pmatrix}, \quad S_{52} = \begin{pmatrix} 1 & 0.483 & -0.544 & -0.697 & 0.225 \\ 0.483 & 1 & 0.392 & -0.392 & 0.282 \\ -0.544 & 0.392 & 1 & 0.060 & -0.337 \\ -0.697 & -0.392 & 0.060 & 1 & 0.513 \\ 0.225 & 0.282 & -0.337 & 0.513 & 1 \end{pmatrix}.$$

Example #7: on this example, the *PSDized* versions of the initial pseudo-correlation matrix differ depending on the algorithm used.

$$S_{101}^{PA} = \begin{pmatrix} 1 & -0.681 & -0.310 & -0.008 & 0.574 & -0.241 & 0.158 & 0.390 & 0.017 & 0.017 \\ -0.681 & 1 & 0.596 & 0.434 & -0.010 & 0.684 & 0.233 & 0.390 & 0.269 & 0.269 \\ -0.310 & 0.596 & 1 & 0.295 & 0.144 & 0.256 & -0.222 & 0.293 & 0.480 & 0.480 \\ -0.008 & 0.434 & 0.295 & 1 & 0.210 & 0.276 & 0.024 & 0.508 & 0.487 & 0.487 \\ 0.574 & -0.010 & 0.144 & 0.210 & 1 & 0.520 & 0.308 & 0.710 & -0.001 & -0.001 \\ -0.241 & 0.684 & 0.256 & 0.276 & 0.520 & 1 & 0.625 & 0.674 & 0.019 & 0.019 \\ 0.158 & 0.233 & -0.222 & 0.024 & 0.308 & 0.625 & 1 & 0.537 & 0.226 & 0.226 \\ 0.390 & 0.390 & 0.293 & 0.508 & 0.710 & 0.674 & 0.537 & 1 & 0.245 & 0.245 \\ 0.017 & 0.269 & 0.480 & 0.487 & -0.001 & 0.019 & 0.226 & 0.245 & 1 & 0.760 \\ 0.017 & 0.269 & 0.480 & 0.487 & -0.001 & 0.019 & 0.226 & 0.245 & 0.760 & 1 \end{pmatrix};$$

$$S_{101}^N = S_{101}^{PA}.$$

$$S_{101} = \begin{pmatrix} 1 & -0.671 & -0.313 & -0.018 & 0.563 & -0.243 & 0.159 & 0.396 & 0.010 & 0.010 \\ -0.671 & 1 & 0.592 & 0.422 & -0.110 & 0.676 & 0.233 & 0.392 & 0.262 & 0.262 \\ -0.313 & 0.592 & 1 & 0.296 & 0.146 & 0.259 & -0.223 & 0.294 & 0.479 & 0.479 \\ -0.018 & 0.422 & 0.296 & 1 & 0.217 & 0.281 & 0.025 & 0.507 & 0.486 & 0.486 \\ 0.563 & -0.110 & 0.146 & 0.217 & 1 & 0.521 & 0.310 & 0.704 & -0.000 & -0.000 \\ -0.243 & 0.676 & 0.259 & 0.281 & 0.521 & 1 & 0.625 & 0.669 & 0.018 & 0.018 \\ 0.159 & 0.233 & -0.223 & 0.025 & 0.310 & 0.625 & 1 & 0.5350.224 & 0.224 & \\ 0.396 & 0.392 & 0.294 & 0.507 & 0.704 & 0.669 & 0.535 & 1 & 0.249 & 0.249 \\ 0.010 & 0.262 & 0.479 & 0.486 & -0.000 & 0.018 & 0.224 & 0.249 & 1 & 0.761 \\ 0.010 & 0.262 & 0.479 & 0.486 & -0.000 & 0.018 & 0.224 & 0.249 & 0.761 & 1 \end{pmatrix}.$$

C *PSDized* matrices with weighted coefficients

Since the only method that allows to integrate weights when looking for the closest *PSDized* matrix is the Rebonato-Jäckel algorithm, we have here seven results coming from our seven examples.

Examples #1 and #2:

$$S_{31}^H = \begin{pmatrix} 1 & -0.848 & -0.505 \\ -0.848 & 1 & -0.029 \\ -0.505 & -0.029 & 1 \end{pmatrix}, \quad S_{32}^H = \begin{pmatrix} 1 & -0.278 & 0.187 \\ -0.278 & 1 & 0.892 \\ 0.187 & 0.892 & 1 \end{pmatrix}.$$

Examples #3 and #4: $S_{41}^{PA} = S_{41}^N = S_{41}$, and $S_{42}^{PA} = S_{42}^N = S_{42}$, where

$$S_{41}^H = \begin{pmatrix} 1 & 0.844 & -0.401 & -0.707 \\ 0.844 & 1 & -0.029 & -0.329 \\ -0.401 & -0.029 & 1 & 0.216 \\ -0.707 & -0.329 & 0.216 & 1 \end{pmatrix}, \quad S_{42}^H = \begin{pmatrix} 1 & 0.041 & -0.614 & 0.041 \\ 0.041 & 1 & 0.129 & 0.144 \\ -0.614 & 0.129 & 1 & 0.763 \\ 0.041 & 0.144 & 0.763 & 1 \end{pmatrix}.$$

Examples #5 and #6: $S_{51}^{PA} = S_{51}^N = S_{51}$, and $S_{52}^{PA} = S_{52}^N = S_{52}$, where

$$S_{51}^H = \begin{pmatrix} 1 & -0.822 & -0.795 & -0.811 & 0.132 \\ -0.822 & 1 & 0.834 & 0.353 & 0.155 \\ -0.795 & 0.834 & 1 & 0.371 & -0.412 \\ -0.811 & 0.353 & 0.371 & 1 & -0.193 \\ 0.132 & 0.155 & -0.412 & -0.193 & 1 \end{pmatrix}, \quad S_{52}^H = \begin{pmatrix} 1 & 0.569 & -0.714 & -0.776 & 0.310 \\ 0.569 & 1 & 0.049 & -0.555 & 0.206 \\ -0.714 & 0.049 & 1 & 0.213 & -0.610 \\ -0.776 & -0.555 & 0.213 & 1 & 0.339 \\ 0.310 & 0.206 & -0.610 & 0.339 & 1 \end{pmatrix}.$$

Example #7:

$$S_{101}^H = \begin{pmatrix} 1 & -0.709 & -0.294 & -0.016 & 0.547 & -0.313 & 0.116 & 0.366 & 0.012 & 0.012 \\ -0.709 & 1 & 0.647 & 0.389 & -0.158 & 0.660 & 0.236 & 0.355 & 0.291 & 0.291 \\ -0.294 & 0.647 & 1 & 0.326 & 0.006 & 0.251 & -0.226 & 0.334 & 0.468 & 0.468 \\ -0.016 & 0.389 & 0.326 & 1 & 0.200 & 0.299 & 0.034 & 0.496 & 0.479 & 0.479 \\ 0.547 & -0.158 & 0.006 & 0.200 & 1 & 0.539 & 0.355 & 0.719 & -0.008 & -0.008 \\ -0.313 & 0.660 & 0.251 & 0.299 & 0.539 & 1 & 0.582 & 0.657 & 0.029 & 0.029 \\ 0.116 & 0.236 & -0.226 & 0.034 & 0.355 & 0.582 & 1 & 0.561 & 0.220 & 0.220 \\ 0.366 & 0.355 & 0.334 & 0.496 & 0.719 & 0.657 & 0.561 & 1 & 0.257 & 0.257 \\ 0.012 & 0.264 & 0.468 & 0.479 & -0.008 & 0.029 & 0.220 & 0.257 & 1 & 0.769 \\ 0.012 & 0.264 & 0.468 & 0.479 & -0.008 & 0.029 & 0.220 & 0.257 & 0.769 & 1 \end{pmatrix}$$

D Genetic algorithms used in this paper

D.1 Algorithm with permutations

(Genetic algorithm for permutations). Implement the following steps:

1. (Creation of the initial population): Simulate a random population of N permutations, with the identity: $\{\sigma_k \in \mathfrak{S}_n | k \in \llbracket 1; N \rrbracket\}$. We have chosen to use $N = 6$ for stability purposes and computational feasibility.
2. (Ranking of the population): Use the function which associates to each permutation the associated SCR (for fixed marginals, fixed weighting matrix, given loss function and copula) to rank the individuals of the population.
3. (Reproduction of two individuals): For any couple $(\sigma_1, \sigma_2) \in \mathfrak{S}_n^2$, create two new individuals by the following permutation composition : $\sigma' = \sigma_1 \circ \sigma_2$ and $\sigma'' = \sigma_2 \circ \sigma_1$. If the two permutation commute, compose one of them by any transposition $\sigma' \leftarrow \tau \circ \sigma'$.
4. (Mutation) : The mutation of a permutation corresponds to its composition by a random transposition $\tau = (k, k + 1)$.
5. (Evolution of the population) : The population evolves without any of its individuals disappearing, which enables to obtain at the end of the algorithm both a minimum and a maximum (not corresponding necessarily to the absolute extrema - but only the results of the simulation).
6. (End of the algorithm) : Terminates when a maximum number of iterations is obtained (5).

D.2 Algorithm with confidence weights

(Genetic algorithm for weights). Implement the following steps:

1. (Creation of the initial population): Simulate a random population of N weighting matrices between H^{min} and H^{max} initially chosen: $\{H_k \in [-1; 1]^{n \times n} | k \in \llbracket 1; N \rrbracket\}$. We have chosen to use $N = 8$ for stability purposes and computational feasibility.
2. (Ranking of the population): Use the function which associates to each weighting matrices the associated SCR (for fixed marginals, given loss function and copula) to rank the individuals of the population.
3. (Reproduction of two individuals): For any couple $(H_1, H_2) \in ([-1; 1]^{n \times n})^2$, create two new individuals H' et H'' in the following manner (with $H[:,j]$ designating the j -th column of the matrix H , and $E(x)$ the integer part of 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) : The mutation of a weighting matrix corresponds to the random modification of a coefficient of the matrix H considered. The mutation consists in simulating a random coefficient between 0 and 1.
5. (Evolution of the population) : The population evolves without any of its individuals disappearing, which enables to obtain at the end of the algorithm both a minimum and a maximum (not corresponding necessarily to the absolute extrema - but only the results of the simulation).
6. (End of the algorithm) : The algorithm terminates when a maximum number of iterations is obtained (5).