

Insurance: Mathematics and Economics manuscript No. (will be inserted by the editor)
--

Sequential Monte Carlo Samplers for capital allocation under copula-dependent risk models

Rodrigo S. Targino · Gareth W. Peters · Pavel V. Shevchenko

January 7, 2015

Abstract In this paper we assume a multivariate risk model has been developed for a portfolio and its capital derived as a homogeneous risk measure. The Euler (or gradient) principle, then, states that the capital to be allocated to each component of the portfolio has to be calculated as an expectation conditional to a rare event, which can be challenging to evaluate in practice. We exploit the copula-dependence within the portfolio risks to design a Sequential Monte Carlo Samplers based estimate to the marginal conditional expectations involved in the problem, showing its efficiency through a series of computational examples.

Keywords Risk Management · Capital Allocation · Sequential Monte Carlo (SMC) · Copula Models

MSC 2010: 65C40, 65C05, 62P05

JEL Classification: C15, G21, G22

1 Introduction

Since financial institutions are in the business of managing and reallocating risks, an important part of their internal risk management is to have an appropriate level of capital as a buffer against unexpected losses. Typically, retail banks, investment banks and insurance companies must satisfy their local jurisdiction version of capital adequacy, which are usually specified by the local regulatory authorities according to some version of either the Basel II/III banking supervision guides or the Solvency II insurance guides.

In this context capital may refer to two different quantities: Economic Capital (EC) or Regulatory Capital (RC). The first is the capital that would have been chosen in the absence of regulation. It represents the amount the institution estimates in order to remain solvent at a given confidence level and fixed time horizon and is set in order to meet some credit risk rating. The Regulatory Capital, in turn, reflects the needs given by regulatory guidance and rules. It is important to note that the capital actually held by the institution (henceforth referred to as *capital*) will always be the maximum between the EC and the RC.

Both for banks and insurance companies, regulation has evolved towards Regulatory Capital based on risk measures (see Section 2). For banks, the Basel II accord set the standard to the Value at Risk (still in use at Basel III) while insurance directives such as Solvency II and the Swiss Solvency Test diverge on the risk measure to be used (the first suggests the usage of Value at Risk and the former Expected Shortfall). It is important to note that although the RC and the EC will differ in most of the financial institutions, both quantities are usually based on the same class of risk measures, differing only on the confidence level.

Rodrigo S. Targino
University College London, London, UK,
E-mail: r.targino.12@ucl.ac.uk

Gareth W. Peters
University College London, London, UK,

Pavel V. Shevchenko
CSIRO, Sydney, Australia

Business line		Event type	
1	Corporate finance	1	Internal fraud
2	Trading and sales	2	External fraud
3	Retail banking	3	Employment practices and workplace safety
4	Commercial banking	4	Clients, products and business practices
5	Payment and settlement	5	Damage to physical assets
6	Agency services	6	Business disruption and system failures
7	Asset management	7	Execution, delivery and process management
8	Retail brokerage		

Table 1.1 Basel II business lines (left) and event types (right) – see [5], Annexes 8 and 9.

This work is focused on studying the problem of capital allocation, with particular focus on Operational Risk (OpRisk) capital – see [42] for an text-book introduction to OpRisk modelling. In order to solely study this aspect, we will assume that the parametric risk models have been selected and the parameter estimations performed in each business unit or division for all the relevant risk types. Then using these models the bank or insurance company has obtained an estimate of the total capital from the model. Based on this capital figure, our work aims to study a problem that follows the calculation of the capital to be held, namely the second order problem of the capital allocation to different divisions and business units as a capital charge (see Table 1.1). Once the overall capital is calculated, the financial institution faces the problem of how to allocate this given capital among different risk sources, in order to understand how much each risk cell contributes to the total risk (capital) and in order to asses their risk management controls and performance, as part of the process discussed in the Pillar III of Basel II/III. Put another way there is a capital charge that must be allocated to each division and business unit which must reflect commensurately the risk profile of the given business unit. This continues to provide an incentive for banks following the Advanced Measurement Approach (AMA) to carefully model their dependence structures.

Apart from the fact that losses in some of the these risk cells may be dependent, the Basel II accord [5] §657 ensures that the capital estimate can have diversification benefits if dependency modelling is approved by the local regulator. In other words, the bank may be authorized to set aside less Regulatory Capital if they can demonstrate evidence for dependence features in their loss processes between each business line or between risk types within a business line.

We will also assume the dependency among all risk cells in the portfolio is known from the first phase of model selection and estimation. More precisely, we will assume that the bank’s portfolio consists of d individual losses (in a risk cell level) denoted by X_1, \dots, X_d , each one modelled as random variables (rv’s) with continuous cumulative distribution function (cdf) given by F_i , $i = 1, \dots, d$.

The dependence structure of the losses will be given by a (known) copula $C(u_1, \dots, u_d)$ (see Appendix B for the definition and some results regarding copulas), leading to a joint distribution of the losses given by

$$F_{\mathbf{X}}(\mathbf{x}) = C(F_1(x_1), \dots, F_d(x_d)),$$

where $\mathbf{X} = (X_1, \dots, X_d)$, and $\mathbf{x} = (x_1, \dots, x_d)$.

In the recent years many academic works were devoted to the joint modelling of operational losses and its impact on capital calculation. Recently, [7] introduced a zero-inflated dependence model, which is then coupled using different copulas (Archimedean, elliptical, individual Student’s t and vine). Previously, [23] used α -stable marginal distributions and Student’s t copulas (both symmetric and skewed ones); [6] derived approximations for the operational Value at Risk assuming generalized Pareto distributions and Lévy copulas. In the context of OpRisk, [39] developed a dynamic OpRisk model with copula dependence structures between the frequency, severity and annual loss as possible model structures. In addition, a detailed account of dependence modelling and capital estimation can be found in [14] and [38].

The remainder of the paper is structured as follows. In Section 2 we present results related to risk contributions, and special attention is drawn to the Euler allocation rule and its duality with expectations conditional to rare events. After understanding the relationship between the Euler principle and conditional expectations, Section 3 presents different methods to estimate these expectations: from a simple Monte Carlo scheme to recently developed Importance Sampling approaches. To be able to discuss the Sequential Monte Carlo Sampler (SMCS) algorithm proposed, Section 4 briefly discusses how to design a sequence of probability densities converging to the conditional distribution involved in the allocation problem. The formal sequential procedure is, then, carefully described in Section 5 and the algorithm’s ingredients necessary for the allocation problem designed in Section 6. We conclude the paper with some simulation examples on Section 7 and final remarks on Section 8.

2 Risk contributions and capital allocations

This section recalls a few key results that have been developed primarily relating to coherent capital allocation principles. A brief overview is provided for the relevant theory of risk contributions, with a focus on Euler allocation principles, for which a more detailed introduction is provided in [33], Section 6.3, [45] and [41], Section 3.

In general one may consider the notation in which X_1, \dots, X_d denotes the returns of d different assets in a portfolio. In the case of OpRisk modelling, this notation will correspond to losses from d different business unit and risk type combinations within divisions of a banking or insurance institution.

If the weights of these assets in the portfolio are given by $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_d) \in A \subset \mathbb{R}^d \setminus \{\mathbf{0}\}$ then we will denote the portfolio-wide loss by

$$X(\boldsymbol{\lambda}) = \sum_{i=1}^d \lambda_i X_i. \quad (2.1)$$

In particular, if $\boldsymbol{\lambda} = (1, \dots, 1)$ we will write $X = X(\boldsymbol{\lambda})$. In the case of OpRisk modelling this aggregate loss amount given by $\sum_{i=1}^d \lambda_i X_i$ would represent the institution-wide total annual loss and typically the weights would be unity.

If all the losses X_1, \dots, X_d are defined in a common probability space $(\Omega, \mathcal{F}, \mathbb{P})$, then we denote $\rho : \Omega \rightarrow \mathbb{R}$ a generic risk measure. Given a specific risk measure ρ , the function $r_\rho : A \rightarrow \mathbb{R}$ such that $r_\rho(\boldsymbol{\lambda}) = \rho(X(\boldsymbol{\lambda}))$ will be called risk-measure function.

In the context of OpRisk, the Basel II/III guidelines specify clearly the recommended regulatory requirements for capital adequacy standards. For instance in the European Union (EU) region the Capital Requirements Directives (CRD) for the financial services industry have developed a supervisory framework to reflect its guidelines on capital measurement and capital standards. In the EU region the CRD-IV package entered into force on 17 July 2013 and reflects the Basel III standards expanding existing Basel II regulatory EU directives (2013/36/EU) and EU regulation (575/2013). These EU directives, like their counterparts in other jurisdictions include specifications on the amounts of Tier I and Tier II capital, liquidity ratios and other relevant capital adequacy criteria. In this paper we are not so interested in specific break up of capital components, instead in this regard, throughout this paper we will refer to capital rather loosely as corresponding to generically the total amount of assets that a financial institution must hold to mitigate their yearly loss exposure obtained from quantification of a risk measure based on the institutions OpRisk models.

From now on, let us assume the bank's capital is defined as a function of a given risk measure ρ , defined as either the standard deviation, the Value at Risk or the Expected Shortfall (see Definition 2.1). For example, for regulatory purposes the OpRisk capital to be held is given by $\rho(X(\boldsymbol{\lambda})) = VaR_\alpha(X(\boldsymbol{\lambda}))$, with $\alpha = 0.999$ for the Regulatory Capital and $\alpha = 0.9995$ for the Economic Capital, for example.

Definition 2.1 (Particular choices of risk measures) If X_1, \dots, X_d are continuous random variables, and $X = \sum_{i=1}^d X_i$ three of the most popular choices of risk measures are given by

1. Standard deviation: $\rho(X) = \sqrt{Var(X)}$;
2. Value at Risk: $\rho(X) = VaR_\alpha(X) := \inf\{x \in \mathbb{R} : F_X(x) = \alpha\}$;
3. Expected Shortfall: $\rho(X) = ES_\alpha(X) := \mathbb{E}[X | X \geq VaR_\alpha(X)]$.

Assuming the risk measure ρ has been chosen and that $\rho(X(\boldsymbol{\lambda}))$, the risk (capital) of the portfolio (institution) has already been calculated, the allocation process consists of understanding how much of the risk (capital) is due to each of the constituents of the portfolio. In the case of OpRisk, this involves understanding what each divisions total capital requirement would be across all the Basel III risk types, as well as what each individual business units and risk types combinations capital requirement should be, based on a "fair" risk based capital allocation from the institutions total capital requirement. This involves the disaggregation of the total institutions capital back down the business unit/risk type structure that will be specific for a given institution but can be generically represented for instance by the 56 business unit/risk type categories proposed in Basel II/III.

More formally, let us denote by $\mathcal{C}_i^\rho(\boldsymbol{\lambda})$ the capital allocated to one unit of X_i when the portfolio's loss is given by $X(\boldsymbol{\lambda})$. For the sake of simplicity, to derive the Euler allocation we will accept the following set of assumptions.

Assumptions 2.2 *If the individual and portfolio losses are given, respectively by X_1, \dots, X_d and Equation (2.1) then we assume that*

- (i) *the capital allocated to the position $\lambda_i X_i$ is given by $\lambda_i \mathcal{C}_i^\rho(\boldsymbol{\lambda})$;*

(ii) the overall risk capital $r_\rho(\boldsymbol{\lambda})$ is fully allocated to the individual positions in the portfolio:

$$\sum_{i=1}^d \lambda_i \mathcal{C}_i^\rho(\boldsymbol{\lambda}) = r_\rho(\boldsymbol{\lambda}). \quad (2.2)$$

The interpretation of the first assumption is that we require proportional positions to have proportional capital shares and the second one ensures the total capital will be allocated to the individual positions. As a result of these assumptions, we only need to calculate $\mathcal{C}_i^\rho(\boldsymbol{\lambda})$, for $i = 1, \dots, d$.

Following the nomenclature in [33], we will say $\mathcal{C}_i^\rho : \Lambda \rightarrow \mathbb{R}^d$ is a per-unit capital allocation principle if (2.2) is satisfied for all $\boldsymbol{\lambda} \in \Lambda$.

So far we have not imposed or assumed any specific requirement on the risk measure used as a base for the allocation principle, but to introduce the popular Euler allocation principle we need to restrict ourselves to the class of positive homogeneous (of degree one) risk measures.

Definition 2.3 (Positive homogeneity) A risk measure ρ is said to be positive homogeneous (of degree one) if $\rho(\lambda X) = \lambda \rho(X)$, for any random variable X and $\lambda > 0$.

It is straightforward to verify that Value at Risk (VaR) is positive homogeneous, as well as any coherent risk measure (in the sense of [2]). For positive homogeneous risk measures it is also trivial to show that the associated risk measure function r_ρ satisfies $r_\rho(t\boldsymbol{\lambda}) = tr_\rho(\boldsymbol{\lambda})$. Therefore, applying Euler's homogeneous function theorem (see Appendix A) on r_ρ , we have that

$$r_\rho(\boldsymbol{\lambda}) = \sum_{i=1}^d \lambda_i \frac{\partial r_\rho}{\partial \lambda_i}(\boldsymbol{\lambda}) \quad (2.3)$$

The combination of (2.2) and (2.3) leads to the so-called Euler allocation principle (sometimes referred as allocation by the gradient), where the capital allocated to the i -th component of the portfolio is given by the partial derivatives, with $\mathcal{C}_i^\rho(\boldsymbol{\lambda}) := \frac{\partial r_\rho}{\partial \lambda_i}(\boldsymbol{\lambda})$.

The Euler allocation principle arises in different contexts in the literature. For example, in [19] and [29] the Euler principle is motivated by two (different) sets of axioms, leading to coherent allocation principles (for a relationship between coherent risk measures and coherent capital allocations see [8]).

Assuming that X_1, \dots, X_d are continuous random variables at the point at which the risk measure is evaluated, we now present some explicit forms of the Euler contributions, based on the different risk measures presented in 2.1.

Proposition 2.4 If $X = \sum_{i=1}^d X_i$ and $\mathbf{X} = (X_1, \dots, X_d)$ has a joint continuous density, then the Euler allocation takes the following form

1. *Standard deviation*: $\rho(X) = \sqrt{\text{Var}(X)} \implies \mathcal{C}_i^\sigma(X_i) = \frac{\text{Cov}(X_i, X)}{\sqrt{\text{Var}(X)}}$;
2. *Value at Risk*: $\rho(X) = \text{VaR}_\alpha(X) \implies \mathcal{C}_i^{\text{VaR}}(X_i) = \mathbb{E}[X_i | X = \text{VaR}_\alpha(X)]$;
3. *Expected Shortfall*: $\rho(X) = \text{ES}_\alpha(X) \implies \mathcal{C}_i^{\text{ES}}(X_i) = \mathbb{E}[X_i | X \geq \text{VaR}_\alpha(X)]$.

Proof See [33], Section 6.3 and references therein.

Remark 2.5 Proposition 2.4 is still valid even if the distribution of \mathbf{X} is not continuous but other technical conditions should be satisfied (see [44] and [26]).

2.1 Euler allocation in a hierarchical structure

Here we briefly extend the concept of Euler allocations to a Bank structure divided, for example, in Business Units and Event types, as in Table 1.1.

Let us assume a bank has a structure given as in Figure 2.1, comprising of K Business Units (B.U.'s) and d_l Event Types (E.T's) in each of its B.U.'s ($l = 1, \dots, K$). In this context we define $d = \sum_{l=1}^K d_l$ as the total number of cells for which capital should be allocated, $X = \sum_{i=1}^d X_i$ the bank loss and $X_{[l]} = \sum_{m=1}^{d_l} X_m$ for $l = 1, \dots, K$ the loss in the l -th B.U..

Assuming the bank capital is given by $\text{ES}_\alpha(X)$ (as in Definition 2.1) then the Euler principle states that the capital at the B.U. level should be given by $\mathbb{E}[X_{[l]} | X > \text{VaR}_\alpha(X)]$ for each B.U. $l = 1, \dots, K$. To

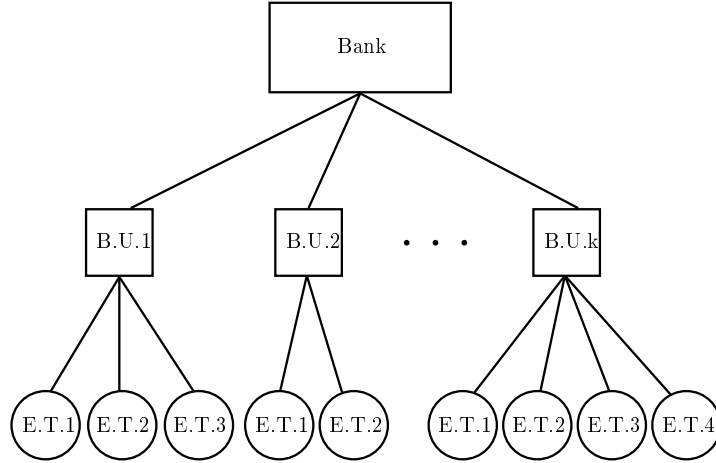


Fig. 2.1 Hierarchical bank structure, with k B.U.'s.

allocate the capital calculated for the l -th unit to its E.T.'s we can assume this capital is a (homogeneous) risk measure defined as

$$\rho(X_{[l]}) = \mathbb{E}[X_{[l]} | X > VaR_\alpha(X)].$$

Then, following the Euler principle it is easy to check the allocation for the m -th E.T. in the l -th B.U. is given by

$$\mathbb{E}[X_m | X > VaR_\alpha(X)].$$

The reader should note that the same allocations could have been derived “heuristically” in the following way. First the total capital $ES_\alpha(X)$ is allocated directly to each of the d E.T.'s. Then for the l -th B.U. the capital is computed as the sum of the allocations at the d_l E.T.'s in this unit. Although the result would be the same, we emphasize the first method, as it is a direct application of the Euler principle (twice).

3 From capital allocation to conditional expectations

In this section we explore the fact that, using the two most important risk measures for OpRisk, the capital allocation framework just described under the Euler allocation principle can be redefined as the calculation of conditional expectations. This is practically a very appealing result as it means that the allocation of the capital can be estimated using specialized Monte Carlo sampling solutions, presented in detail below.

Throughout it will be assumed that the marginal loss processes are continuous random variables. Therefore, one has that all the marginal inverse distribution functions (quantile functions) F_i^{-1} are well defined and are also continuous. Moreover, one may apply Sklar's theorem to state that the dependence structure of the loss vector \mathbf{X} will be uniquely determined by a copula function C (see Appendix B for some results on copula theory). Formally the joint cdf and pdf of the vector \mathbf{X} can be written as

$$F_{\mathbf{X}}(\mathbf{x}) = \mathbb{P}[X_1 \leq x_1, \dots, X_d \leq x_d] = C(F_1(x_1), \dots, F_d(x_d)) \text{ and}$$

$$f_{\mathbf{X}}(\mathbf{x}) = c(F_1(x_1), \dots, F_d(x_d)) \prod_{i=1}^d f_i(x_i),$$

where C and c are the copula and copula density, respectively.

From Proposition 2.4 we can see both the allocation based on VaR and on ES can be calculated as an expectation of the form

$$C_i^\rho(X_i) = \mathbb{E}[h(\mathbf{X}) | g(\mathbf{X}) \in A], \quad (3.1)$$

for the following choices of h , g and A :

1. For VaR: $h(\mathbf{X}) = X_i$, $g(\mathbf{X}) = \sum_{i=1}^d X_i$, $A = [VaR_\alpha(\sum_{i=1}^d X_i), VaR_\alpha(\sum_{i=1}^d X_i)]$;
2. For ES: $h(\mathbf{X}) = X_i$, $g(\mathbf{X}) = \sum_{i=1}^d X_i$, $A = [VaR_\alpha(\sum_{i=1}^d X_i), +\infty)$.

Note that in any practical scenario α will be chosen to be close to one and both conditioning events will have very small probabilities, ie, $\mathbb{P}[g(\mathbf{X}) \in A] \approx 0$. More precisely, the event will have probability $1 - \alpha$ for the ES case and probability zero (regardless of the choice of α) in the VaR case, since the set A is just a point. Therefore in practice, in the VaR case one would work instead with an ϵ approximation by setting the conditioning event as the ϵ -ball of A given by $\mathcal{B}_\epsilon(A)$ specified according to $\mathcal{B}_\epsilon(A) = [VaR_\alpha(\sum_{i=1}^d X_i) - \epsilon, VaR_\alpha(\sum_{i=1}^d X_i) + \epsilon]$ for some small positive ϵ in the neighbourhood of zero, $\epsilon \in n.e.(0^+)$. This type of approach was advocated in [25] and [18].

From a risk management perspective, many other choices of h and g can be of interest. For example, if one is interested in measuring the impact of marginal tail events in the portfolio, one could calculate expectations of the form $\mathbb{E}[\sum_{i=1}^d X_i | X_k > VaR_\alpha(X_k)]$, where the choices of h , g and A are trivial.

In practice, apart from very particular choices of dependence structure and marginal distributions (eg, [3]), analytic representations for the expectation in (3.1) will not be available and we will need to resort in some numerical approximation to evaluate such quantities in practice. There are several classes of algorithm of relevance to undertake this task so, before presenting our Sequential Monte Carlo (SMC) Samplers based solution, we review some of the most recent proposals in the literature.

3.1 Simulation methods to calculate conditional expectations of rare-events

In order to fix the notation to be used throughout this work, in this section we describe a generic Monte Carlo based estimator, before detailing a few specialized approaches to be compared with our proposed method. It is important to recognize the difference between Monte Carlo, Importance Sampling and Sequential Monte Carlo Sampler based solutions. These are different categories of solution technique as will be detailed below.

The generic expectation $\mathbb{E}[h(\mathbf{X}) | g(\mathbf{X}) \in A]$ can be approximated via a Monte Carlo simulation through the use of a set of N weighted samples $\{\mathbf{x}^{(j)}, W^{(j)}\}_{j=1}^N$, with $\sum_{j=1}^N W^{(j)} = 1$, from the conditional distribution $f_{\mathbf{X}}(\mathbf{x} | g(\mathbf{x}) \in A)$. The approximation for the expectation is, then, given by

$$\mathbb{E}[h(\mathbf{X}) | g(\mathbf{X}) \in A] \approx \sum_{j=1}^N W^{(j)} h(\mathbf{x}^{(j)}).$$

For example, if we can directly sample i.i.d. realizations from the distribution of $\mathbf{X} | g(\mathbf{X}) \in A$ (eg, using a rejection scheme) we would have $W^{(j)} = 1/N$. In general, though, samples from the above conditional distribution are not readily available and we need to use an importance sampling distribution (coupled with a rejection step), calculating the weights accordingly to remove bias. In both cases a rejection mechanism can be used, accepting only those particles satisfying the conditioning event. However, if done naively this would result in very large rejection rates that would behave poorly as the conditioning event becomes rarer or the dimension d gets large.

In this setup, the conditioning event we are interested in can be defined as $[\mathbf{X} \in \mathcal{G}_{\mathbf{X}}]$, where

$$\mathcal{G}_{\mathbf{X}} := \{\mathbf{x} \in \mathbb{R}^d : g(\mathbf{x}) \in A\}, \quad (3.2)$$

since $g(\mathbf{X}) \in A \iff \mathbf{X} \in \mathcal{G}_{\mathbf{X}}$. Given that our multivariate loss model is uniquely characterized by a copula (either explicitly or implicitly, through a parametric joint distribution) the region $\mathcal{G}_{\mathbf{X}}$ in \mathbb{R}^d holds a close relationship with some region in $[0, 1]^d$. Formally, if we define

$$\mathcal{G}_{\mathbf{U}} := \{\mathbf{u} \in [0, 1]^d : (F_1^{-1}(u_1), \dots, F_d^{-1}(u_d)) \in \mathcal{G}_{\mathbf{X}}\} \quad (3.3)$$

then it holds that

$$\mathbf{x} \in \mathcal{G}_{\mathbf{X}} \iff \mathbf{u} \in \mathcal{G}_{\mathbf{U}}.$$

Therefore, similarly to the simulation of an unconditional multivariate distribution, to sample $\mathbf{x} = (x_1, \dots, x_d)$ from the distribution of $(\mathbf{X} | g(\mathbf{X}) \in A)$ we can

1. Produce a weighted sample $\{\mathbf{u}^{(j)}, W^{(j)}\}_{j=1}^N$ from C such that $\mathbf{u}^{(j)} = (u_1^{(j)}, \dots, u_d^{(j)}) \in \mathcal{G}_{\mathbf{U}}$ for all $j = 1, \dots, N$;
2. Return the weighted sample $\{\mathbf{x}^{(j)}, W^{(j)}\}_{j=1}^N$ where $x_i^{(j)} = F_i^{-1}(u_i^{(j)})$, for $i = 1, \dots, d, j = 1, \dots, N$.

Note that one can calculate conditional expectations with respect to \mathbf{X} as follows

$$\mathbb{E}[h(\mathbf{X}) | g(\mathbf{X}) \in A] = \mathbb{E}\left[h(F_1^{(-1)}(u_1), \dots, F_d^{(-d)}(u_d)) | \mathbf{u} \in \mathcal{G}_{\mathbf{U}}\right] \quad (3.4)$$

$$\approx \sum_{j=1}^N W^{(j)} h(F_1^{(-1)}(u_1^{(j)}), \dots, F_d^{(-d)}(u_d^{(j)})). \quad (3.5)$$

Clearly, if all the marginal quantile functions F_i^{-1} are known, then the difficulty of the proposed approach is to sample from the constrained copula. The idea of performing the sampling procedure in the constrained copula space has been independently developed by Arbenz, Cambou and Hofert in [1] (see an overview of their algorithm in section 3.3), where an importance sampling distribution is designed to target the distribution of $\mathbf{u} | \mathbf{u} \in \mathcal{G}_{\mathbf{U}}$.

In this paper, instead of targeting the rare region $\mathbf{u} \in \mathcal{G}_{\mathbf{U}}$ we propose to sequentially target less rare regions, in a specially designed SMC Sampler procedure that is made precise in Sections 4 and 5.

Before presenting these specialized algorithms, it is informative to briefly comment on alternative Importance Sampling (IS) based approaches in the actuarial literature. At this stage we observe that there are many different types of rare-event simulation algorithms available, and the choice of a particular type will depend principally on how one defines the notion of a ‘‘rare-event’’ in the sample space. Although the following brief discussions on alternative IS based solutions are not directly targeting the same type of multivariate rare-event problems as faced in the case of capital allocation, they are informative to discuss especially with regard to the concept of relative error.

We also note that there are classes of asymptotic approximation results available for approximation of capital allocations. For instance, in order to estimate expectations of the form (3.1) in a bivariate set-up, [9] assume that large values of $g(\mathbf{X})$ correspond to high values of $h(\mathbf{X})$ (in their case $h(\mathbf{X}) = X_i$). Under these constraints, the authors use results from Extreme Value Theory (EVT) to derive an estimator of (3.1) and study some of its properties. We refer the interested reader to references in [9] for further background on such asymptotic approximations and instead we continue to focus upon sampling based solutions.

3.2 Related Monte Carlo and Importance Sampling approaches

In the particular case of Gaussian copulas [25] presents an IS scheme to approximate conditional expectations. For the same family of models, [43] more recently proposed a method based on Fourier transforms to compute marginal risk contributions. In [32] the author develops a class of IS based estimators that satisfy a condition of bounded relative error. In particular they estimate tail events in a univariate framework via IS based distributions constructed from exponential families which will guarantee bounded relative error in estimated tail functionals. The class of methods they develop revolves around exponential tilting of the tails of the target distribution, also known in the actuarial literature as the Esscher transform or tempering. In addition, it is argued in [32] that for tail events, the variance or standard error is a less desirable quantity to consider in assessing the performance of algorithm, compared to a version scaled by the mean such as the relative error. Therefore, [32] proposed consideration of the relative error which is simply the ratio of the estimators standard error deviation to its mean. In Section 5.3 we present some discussion of this concept in the context of the SMC algorithms proposed.

The approach proposed in [32] selects the IS distribution to minimize the relative error of the rare-event probability by rewriting the problem as the solution to the minimization of the Renyi generalized divergence. In several simple univariate examples one can prove the relative error can be bounded if selected in this manner. This is interesting as it is contrary to other IS based approaches which seek to minimize the importance sampling weights variance. The closest class of IS based solution to our proposed SMC Sampler solution has been developed recently by [1]. We present this briefly before detailing our approach.

3.3 Arbenz-Cambou-Hofert algorithm

In common with the proposed sampling method in this paper, the approach recently developed in [1] involves sampling from a target distribution given by a constrained copula. This is particularly relevant as it leads to convenient bounded state spaces for sampling, since the support of the copula when unconstrained is $[0, 1]^d$ and consequently the constrained copula will be on a sub-space of this hypercube.

Unlike our proposed solution, the approach of [1] involves developing an Importance Sampling (IS) scheme targeting the constrained copula distribution. However, in contrast to our approach their method does not involve any intermediate sequence of constrained regions leading smoothly up to the rare-event constraint. Instead, they try to directly approximate the optimal importance sampling proposal. This is in general a very challenging task and they have some interesting insight. For this reason we briefly present their methodology below as it will form a direct comparison with the approach we develop.

In [1] the aim of their work is to generate a sample from the unconditional copula with most of the particles satisfying a condition such as (3.3). In order to generate these samples, an importance sampling distribution $F_{\mathbf{V}}$ is designed as a mixture of conditional copulas. More formally, the IS distribution is defined as

$$F_{\mathbf{V}}(\mathbf{u}) = \int_0^1 C^{(\lambda)}(\mathbf{u}) dF_A(\lambda), \quad (3.6)$$

where $C^{(\lambda)}$ is the distribution of \mathbf{U} conditional on the event that at least one of its components exceeds λ , ie,

$$C^{(\lambda)}(\mathbf{u}) = \mathbb{P}[U_1 \leq u_1, \dots, U_d \leq u_d \mid \max\{U_1, \dots, U_d\} > \lambda].$$

In the main algorithm presented in [1], samples from the importance distribution are generated by rejection, but a ‘‘conditional sampling algorithm’’ is also presented. Overall, an appealing aspect of their proposed method is that it does not make use of the copula density explicitly. This can be advantageous in settings in which the copula density is computationally expensive to be calculated or even unknown. However, as with all Monte Carlo methods, there are also drawbacks to the proposed approach that we will argue can be overcome through development not of an IS solution but instead via a Sequential Monte Carlo Sampler (SMC Sampler) solution in the constrained copula space.

Under simplifying assumptions on the joint behaviour of \mathbf{U} , an optimal distribution for F_A is presented, but in general the only restriction on the choice of the mixing distribution F_A is that $\mathbb{P}[A = 0] > 0$. To sample from $\mathbf{X} \mid \mathbf{X} > B$ one of the algorithms proposed is given as follows.

Algorithm 1: IS-ACH algorithm from [1].

Inputs: N : desired sample size for $F_{\mathbf{V}}$; F_A : mixing distribution (as in (3.6)) ;

for $j = 1, \dots, N$ **do**

 Sample $\Lambda^{(j)} \sim F_A$;

 Sample $\mathbf{u}^{(j)} \sim C$ until $\max\{u_1^{(j)}, \dots, u_d^{(j)}\} > \Lambda^{(j)}$;

 Define $x_i^{(j)} := F_i^{-1}(u_i^{(j)})$ for $i = 1, \dots, d$;

 Compute the importance weight $w^{(j)} := w(\mathbf{u}^{(j)})$ as in [1], Section 5 ;

 Compute the normalized importance weight $W^{(j)} = \frac{w^{(j)}}{\sum_{j=1}^N w^{(j)}}$;

 Define $\{\tilde{\mathbf{u}}^{(jk)}\}_{k=1}^{\tilde{N}}$ as the sub-set of $\{\mathbf{u}^{(j)}\}_{j=1}^N$ such that $\sum_{i=1}^d \tilde{x}_i^{(jk)} > B$;

end

Result: Weighted random samples (of random sample size \tilde{N}): $\{\mathbf{u}^{(j)}, W^{(j)}\}_{j=1}^{\tilde{N}}$;

Some points need to be stressed about Algorithm 1. First, let $\mathbb{E}[N_{\mathbf{V}}]$ denote the expected number of draws from C in order to have a sample satisfying $\max\{u_1, \dots, u_d\} > \Lambda$. It can be easily shown that (see [1], Lemma 4.2)

$$\mathbb{E}[N_{\mathbf{V}}] = \int_0^1 \frac{1}{1 - C(\lambda \mathbf{1})} dF_A(\lambda),$$

where $\mathbf{1} = (1, \dots, 1) \in \mathbb{R}^d$. Therefore, to generate a sample of fixed size N from $F_{\mathbf{V}}$ it is necessary to sample (on average) $N \times \mathbb{E}[N_{\mathbf{V}}]$ times from C .

Another important aspect of this algorithm pertains to an understanding of the number of ‘‘particles’’ (samples) which are obtained with non-zero weight. Since $p_0 := \mathbb{P}[A = 0]$ is necessarily positive, we can ensure some of the N samples from $F_{\mathbf{V}}$ will be actually from the unconditional copula C , meaning that $p_0 \times 0.99 \times 100\%$ of the particles are expected not to satisfy the condition $\sum_{i=1}^d X_i > VaR_{0.99}(\sum_{i=1}^d X_i)$. For $\lambda > 0$ the same behaviour is expected, leading, in practice, to \tilde{N} (as defined in the last step of Algorithm 1) being smaller than N , and in cases of relevance to capital allocation, this difference can be significant, with $\tilde{N} \ll N$. In capital allocation problems such cases can prove to be a serious problem in terms of computational cost and efficiency for this IS based approach as will be discussed in Section 7.

Next we will present a completely different class of methods to the IS based solutions discussed. These will be based on a class of algorithms that extends IS solutions to sequential settings, known in statistics literature as Sequential Monte Carlo Samplers (SMC Samplers). To understand SMC Samplers we first recall the SMC algorithm before showing its generalization to the class of SMC Sampler algorithms.

4 Reaching rare-events through sequences of intermediate sets

The idea of using intermediate sets to approximate the conditional density $f_{\mathbf{X}|g(\mathbf{X}) \in A}(\mathbf{x})$ is to start sampling from the unconditional distribution $f_{\mathbf{X}}(\mathbf{x})$ and move the weighted particles towards the rare conditioning set through “not so rare” sets.

Using the notation from the previous section, for a fixed function g and set A , let $\{A_t\}_{t=1}^T$ be a sequence of nested sets shrinking to A , ie, $A_t \subset A_{t-1}$ and $A_t \downarrow A$, when $t \rightarrow T$. This sequence of sets defines a sequence of regions (as before):

$$\begin{aligned}\mathcal{G}_{\mathbf{X}_t} &:= \{\mathbf{x}_t \in \mathbb{R}^d : g(\mathbf{x}_t) \in A_t\}, \\ \mathcal{G}_{\mathbf{U}_t} &:= \{\mathbf{u}_t \in [0, 1]^d : (F_1^{-1}(u_{t,1}), \dots, F_d^{-1}(u_{t,d})) \in \mathcal{G}_{\mathbf{X}_t}\}.\end{aligned}$$

Although it is true that

$$\mathbf{x}_t \in \mathcal{G}_{\mathbf{X}_t} \iff \mathbf{u}_t \in \mathcal{G}_{\mathbf{U}_t},$$

with $\mathbf{u}_t = (u_{t,1}, \dots, u_{t,d}) := (F_1(x_{t,1}), \dots, F_d(x_{t,d}))$ we will see in the sequel that working in the bounded space $[0, 1]^d$ we will have some advantages in the design of the algorithm. In this set-up our goal will be ultimately to have (weighted) samples $\left\{ \mathbf{u}_T^{(j)}, W_T^{(j)} \right\}_{j=1}^N$ from the conditional copula $c(\mathbf{u}_T | \mathbf{u}_T \in \mathcal{G}_{\mathbf{U}_T})$ which would be then transformed through the marginal inverse cdf's in order to get a weighted sample from $f_{\mathbf{X}}(\mathbf{x} | \mathbf{x} \in \mathcal{G}_{\mathbf{X}_T})$.

Following the notation to be used in Section 5.2 we define our target distribution at each time step (level) $t = 1, \dots, T$ as

$$\pi_t(\mathbf{u}_t) := \frac{c(\mathbf{u}_t) \mathbb{1}_{\{\mathbf{u}_t \in \mathcal{G}_{\mathbf{U}_t}\}}(\mathbf{u}_t)}{\mathbb{P}[\mathbf{U}_t \in \mathcal{G}_{\mathbf{U}_t}]}. \quad (4.1)$$

4.1 Copula Constrained Geometry

Before we formalize the algorithm to sample from the constrained copula we will study some properties of the restricted region in the copula space, defined in (3.2) and (3.3), for the particular case where $g(\mathbf{X}) = \sum_{i=1}^d X_i$ and $A = [B, +\infty)$. The idea is that the knowledge of the restricted region can help us to design more efficient sampling schemes. Similar analyses can be performed for different restrictions.

In \mathbb{R}^d our interest is to study points such that $\sum_{i=1}^d x_i = B$ which turn out to be equivalent to points in $[0, 1]^d$ such that $\sum_{i=1}^d F_i^{-1}(u_i) = B$. It is easy to see that each of these curves (in \mathbb{R}^d or in $[0, 1]^d$) will lie in a $d - 1$ dimensional space. Formally, these curves are defined through the following mappings

$$\begin{aligned}\tilde{\mathcal{G}}_{\mathbf{X}} &:= \left\{ (x_1, \dots, x_{d-1}) \in \mathbb{R}^{d-1} : (x_1, \dots, x_{d-1}, B - \sum_{i=1}^{d-1} x_i) \right\}, \\ \tilde{\mathcal{G}}_{\mathbf{U}} &:= \left\{ \mathbf{u}_{-d} := (u_1, \dots, u_{d-1}) \in [0, 1]^{d-1} : (u_1, \dots, u_{d-1}, r(\mathbf{u}_{-d})) \right\},\end{aligned} \quad (4.2)$$

where $r(\mathbf{u}_{-d}) := F_d\left(B - \sum_{i=1}^{d-1} F_i^{-1}(u_i)\right)$.

First of all, note that if g is a generic continuous function and all the marginal cdfs F_1, \dots, F_d are continuous then the curve $\tilde{\mathcal{G}}_{\mathbf{U}}$ (defined similarly to (4.2)) will be continuous. Moreover, the region $\mathcal{G}_{\mathbf{U}}$ in (3.3) will not be the union of disjoint set, but only one continuous region. Some other properties of these regions may be derived in particular cases. For example, we know that in the linear case ($g(\mathbf{X}) = \sum_{i=1}^d X_i$) the curve in $[0, 1]^d$ will pass through the points $(F_1(B), 0, \dots, 0)$, $(0, F_2(B), 0, \dots, 0)$, \dots , $(0, \dots, 0, F_d(B))$.

Another interesting information about the curve $\tilde{\mathcal{G}}_{\mathbf{U}}$ is given by its curvature, as seen in the next Proposition.

Proposition 4.1 *The curve $\tilde{\mathcal{G}}_{\mathbf{U}}$ defined in (4.2) is convex at $(u_1, \dots, u_{d-1}, r(\mathbf{u}_{-d}))$ if*

$$\langle \mathbf{u}_{-d}, \nabla^2 r(\mathbf{u}_{-d}) \mathbf{u}_{-d} \rangle > 0, \quad \forall \mathbf{u}_{-d} \in \mathbb{R}^{d-1},$$

where $\langle \mathbf{x}, \mathbf{y} \rangle$ is the inner product of \mathbf{x} and \mathbf{y} and $\nabla^2 f$ is the Hessian matrix of f .

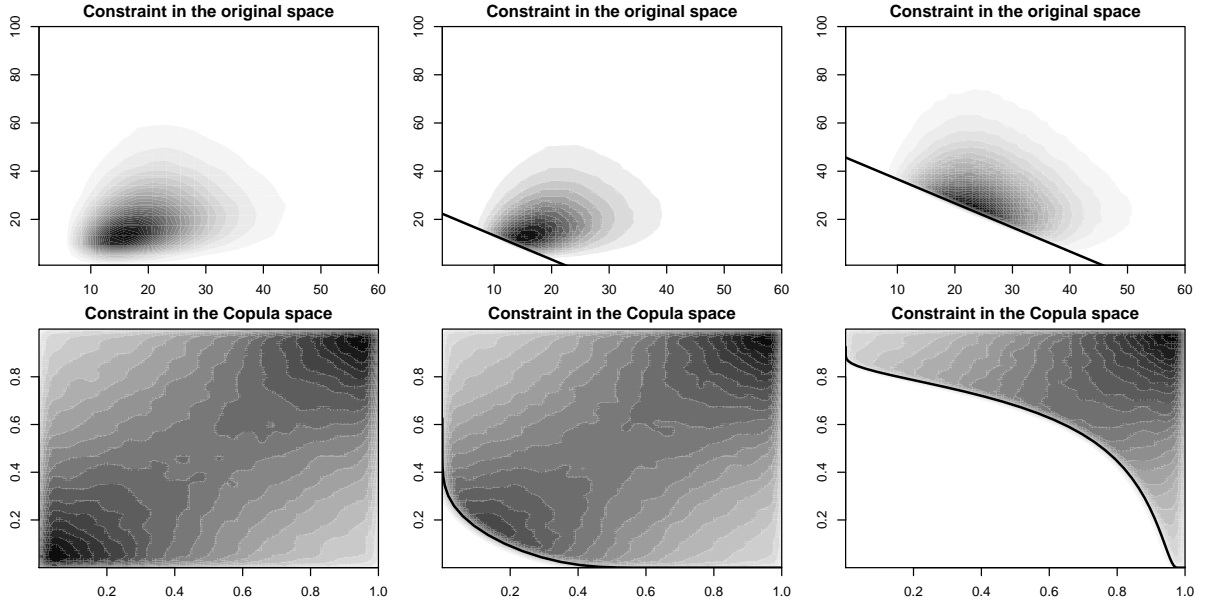


Fig. 4.1 Frank copula with parameter 2, Log-Normal marginal cdf's, both with same $\mu = 3$, and $\sigma_1 = 0.4$, $\sigma_2 = 0.6$. (Top row) Constraint in the original space, for $B = 0, 23, 46$ (Bottom row) Constraint in the Copula space, $[0, 1]^2$, for equivalent levels.

In the particular case where $r(\mathbf{u}_{-d}) := F_d\left(B - \sum_{i=1}^{d-1} F_i^{-1}(u_i)\right)$ the general terms of the Hessian matrix are given by

$$\begin{aligned} \frac{\partial^2 r}{\partial u_j \partial u_k}(\mathbf{u}_{-d}) &= \frac{f'_d\left(B - \sum_{i=1}^{d-1} x_i\right)}{f_j(x_j) f_k(x_k)}, \quad \forall j \neq k, \quad j, k = 1, \dots, d-1 \\ \frac{\partial^2 r}{\partial u_j^2}(\mathbf{u}_{-d}) &= \frac{f'_d\left(B - \sum_{i=1}^{d-1} x_i\right) f_j(x_j) + f_d\left(B - \sum_{i=1}^{d-1} x_i\right) f'_j(x_j)}{[f_j(x_j)]^3}, \\ &\quad \forall j = k, \quad j = 1, \dots, d-1 \end{aligned}$$

where, once again, we use the notation $x_i = F_i^{-1}(u_i)$ to make the above formulas more appealing.

From Proposition 4.1, in the very particular case where $d = 2$ and $X_1, X_2 \geq 0$ (representing losses, for example) the concavity of $\tilde{\mathcal{G}}_{\mathbf{U}}$ is determined only by the sign of

$$f'_2(B - x_1) f_1(x_1) + f_2(B - x_1) f'_1(x_1).$$

This is due to the fact that the denominator is the power of a density function (non-negative) and that x_1 is non-negative.

On Figure 4.1 we can see that for different constraint levels the curve in $[0, 1]^2$ presents different shapes, continuously varying from a convex to a concave region.

5 Sequential Monte Carlo Methods (SMC)

In this section we will introduce the general class of algorithms known as Sequential Monte Carlo (SMC) and an important variant for rare-event simulation, the SMC Samplers classes of methods. This family of Monte Carlo algorithms has been developed to approximate sequences of integrals constructed from a sequence of probability density functions. Of course, adjustments are possible when the interest lies only in one distribution, such as the terminal distribution in a sequence of intermediate increasingly rare events, such as was shown in the ideas presented in Section 4).

SMC methods have emerged out of the fields of engineering, probability and statistics in recent years. Variants of the methods sometimes appear under the names of particle filtering or interacting particle systems e.g. [40], [20], [16], and their theoretical properties have been extensively studied in [13], [16], [11], [30]. For a recent survey in the topic, with focus on economics, finance and insurance applications the reader is referred to [12] and [18]. For an application to rare events in a financial context, see [10].

The general context of a standard SMC method is that one wants to approximate a (often naturally occurring) sequence of probability density functions (pdf's) $\{\tilde{\pi}_t\}_{t \geq 1}$ such that the support of every function in this sequence is defined as $\text{supp}(\tilde{\pi}_t) = E_t$ and the dimension of E_t forms an increasing sequence, i.e., $\dim(E_{t-1}) < \dim(E_t)$. For example, the reader can think of $E_1 = \mathbb{R}^d, \dots, E_t = \mathbb{R}^{d \times t}$, which will be precisely the sequence to be used throughout this work.

We may also assume that $\tilde{\pi}_t$ is only known up to a normalizing constant,

$$\tilde{\pi}_t(\mathbf{x}_{1:t}) = Z_t^{-1} \tilde{f}_t(\mathbf{x}_{1:t}),$$

where $\mathbf{x}_{1:t} := (\mathbf{x}_1, \dots, \mathbf{x}_t) \in E_t = \mathbb{R}^{d \times t}$. As in section 3.1, the approximation for $\tilde{\pi}_t$ is given by a weighted sum of random samples (also known as ‘‘particles’’).

Procedurally, we initialize the algorithm sampling a set of N particles from the distribution $\tilde{\pi}_1$ and set the normalized weights to $W_1^{(j)} = 1/N$, for all $j = 1, \dots, N$. If it is not possible to sample directly from $\tilde{\pi}_1$, one should sample from an importance distribution \tilde{q}_1 and calculate its weights accordingly (see Algorithm 2). Then the particles are sequentially propagated thorough each distribution $\tilde{\pi}_t$ in the sequence via three main processes: mutation, correction (incremental importance weighting) and resampling. In the first step (mutation) we propagate particles from time $t - 1$ to time t and in the second one (correction) we calculate the new importance weights of the particles.

This method can be seen as a sequence of IS steps, where the target distribution at each step t is $\tilde{f}_t(\mathbf{x}_{1:t})$ (the unnormalized version of $\tilde{\pi}_t$) and the importance distribution is given by

$$\tilde{q}_t(\mathbf{x}_{1:t}) = \tilde{q}_1(\mathbf{x}_1) \prod_{j=2}^t K_j(\mathbf{x}_{j-1}, \mathbf{x}_j), \quad (5.1)$$

where $K_j(\mathbf{x}_{j-1}, \cdot)$ is the mechanism used to propagate particles from time $t - 1$ to t , known as the mutation stage. The algorithm works in the following way:

Algorithm 2: Standard SMC algorithm.

Inputs: IS density \tilde{q}_1 , (forward) mutation kernels $\{K_t(\mathbf{x}_{t-1}, \mathbf{x}_t)\}_{t=1}^T$;
for $j = 1, \dots, N$ **do**
 Sample $\mathbf{x}_1^{(j)}$ from $\mathbf{X}_1 \sim \tilde{q}_1(\cdot)$ (*Mutation step*);
 Calculate the weights $w_1^{(j)} = \frac{\tilde{f}_1(\mathbf{x}_1^{(j)})}{\tilde{q}_1(\mathbf{x}_1^{(j)})}$;
end
Calculate the normalized weights $W_1^{(j)} = \frac{w_1^{(j)}}{\sum_{j=1}^N w_1^{(j)}}$ (*Correction step*);
for $t = 2, \dots, T$ **do**
 for $j = 1, \dots, N$ **do**
 Sample $\mathbf{x}_t^{(j)}$ from $\mathbf{X}_t | \mathbf{X}_{t-1} = \mathbf{x}_{t-1}^{(j)} \sim K_t(\mathbf{x}_{t-1}^{(j)}, \cdot)$ (*Mutation step*);
 Create the vector $\mathbf{x}_{1:t}^{(j)} := (\mathbf{x}_{1:(t-1)}^{(j)}, \mathbf{x}_t^{(j)})$;
 Calculate the weights $w_t^{(j)} = \frac{\tilde{f}_t(\mathbf{x}_{1:t}^{(j)})}{\tilde{q}_t(\mathbf{x}_{1:t}^{(j)})} = w_{t-1}^{(j)} \underbrace{\frac{\tilde{f}_t(\mathbf{x}_{1:t}^{(j)})}{\tilde{f}_{t-1}(\mathbf{x}_{1:t-1}^{(j)}) K_t(\mathbf{x}_{t-1}^{(j)}, \mathbf{x}_t^{(j)})}}_{\text{incremental weight: } \tilde{\alpha}(\mathbf{x}_{1:t}^{(j)})}$;
 end
 Calculate the normalized weights $W_t^{(j)} = \frac{w_t^{(j)}}{\sum_{j=1}^N w_t^{(j)}}$ (*Correction step*).
end
Result: Weighted random samples $\{\mathbf{x}_{1:t}^{(j)}, W_t^{(j)}\}_{j=1}^N$ approximating $\tilde{\pi}_t$, for all $t = 1, \dots, T$;

If $\{\mathbf{x}_{1:t}^{(j)}, W_t^{(j)}\}_{j=1}^N$ is a set of weighted particles returned by the SMC algorithm then

$$\sum_{j=1}^N W_t^{(j)} \varphi(\mathbf{x}_{1:t}^{(j)}) \longrightarrow \mathbb{E}_{\tilde{\pi}_t}[\varphi(\mathbf{X}_{1:t})] := \int_{E_t} \varphi(\mathbf{x}_{1:t}) \tilde{\pi}_t(\mathbf{x}_{1:t}) d\mathbf{x}_{1:t}, \quad (5.2)$$

$\tilde{\pi}_t$ -almost surely as $N \rightarrow +\infty$, for any test function φ such that the expectation of φ under $\tilde{\pi}_t$ exists.

Remark 5.1 The reader should note that the knowledge of $\tilde{\pi}_t$ up to a normalizing constant is sufficient for the implementation of a generic SMC algorithm, since the normalized version of the weights would be the same for both $\tilde{\pi}_t$ and f_t .

The optimal selection of the mutation kernel (SMC importance distribution) for SMC methods is widely studied and a good tutorial review on the optimal choice minimizing the variance of the incremental importance sampling weights is overviewed in [21]. There are also a range of known probabilistic properties of the SMC algorithm available in the literature, for a tutorial in the insurance context on these properties see [18]. This includes details on central limit theorem results for SMC algorithms along with asymptotic variance expressions, finite sample bias decompositions and propagation of chaos as well as finite sample concentration inequality bounds. There are also tutorials available on SMC algorithms in general such as [21] and the book length coverage of [16].

5.1 Resampling and Moving particles

In practice the generic algorithm presented in the previous section will eventually (as t increases) be based only in a few distinct particles, in the sense that almost all the other particles will have negligible weights. This phenomenon is known as particle degeneracy.

To overcome this problem, when the system is too degenerate one can resample all the particles $x_{1:t}$ after the correction step, choosing the j -th one with probability proportional to $W_t^{(j)}$. In [31] it was suggested using the Effective Sample Size (ESS) to measure the sample degeneracy, where

$$ESS_t := \left[\sum_{j=1}^N (W_t^{(j)})^2 \right]^{-1}$$

and resample steps should be performed only when $ESS_t < M$ – as a rule of thumb we can set $M = N/2$. It is important to note that after this step we need to set $W_t^{(j)} = 1/N$ for all particles, since they are all identically distributed.

Although the resample step alleviates the degeneracy problem, its successive reapplication, at each stage of the sampler, produces the so-called sample impoverishment, where the number of distinct particles is extremely small. In [24] it was proposed to add a move with any kernel such that the target distribution is invariant with respect to it to rejuvenate the system. This kernel may be, for example, a Markov Chain kernel, which would begin with equally weighted samples from the target distribution and then perturb them under a single step of a Metropolis Hastings accept-reject mechanism. This would preserve the target distribution and add diversity to the particle cloud.

More precisely, we can apply any kernel $M(\mathbf{x}_{1:t}, \mathbf{x}_{1:t}^*)$ that leaves $\tilde{\pi}_t$ invariant to move particle $\mathbf{x}_{1:t}$ to $\mathbf{x}_{1:t}^*$ (the star will denote particles *after* the “move” step), ie,

$$\tilde{\pi}_t(\mathbf{x}_{1:t}^*) = \int M(\mathbf{x}_{1:t}, \mathbf{x}_{1:t}^*) \tilde{\pi}_t(\mathbf{x}_{1:t}) d\mathbf{x}_{1:t}.$$

Two of the simplest ways to construct such a kernel M are to use a Gibbs sampler or a Metropolis-Hastings (M-H) algorithm. To use a Gibbs sampler algorithm, the full conditional distributions $\tilde{\pi}_t(\mathbf{x}_{1:t,i} | \mathbf{x}_{1:t,1}, \dots, \mathbf{x}_{1:t,i-1}, \mathbf{x}_{1:t,i+1}, \dots, \mathbf{x}_{1:t,d})$, for $i = 1, \dots, d$, must be known up to proportionality, while for the M-H they are not necessary. On the other hand, in the M-H algorithm one needs to design a proposal density $Q(\mathbf{x}_{1:t}, \mathbf{x}_{1:t}^*)$ that moves the particle $\mathbf{x}_{1:t}$ to $\mathbf{x}_{1:t}^*$ or some component of it such as \mathbf{x}_t to \mathbf{x}_t^* . The Gibbs sampler is presented as Algorithm 3. For the Metropolis-Hastings and/or more details on MCMC methods, see, for example, [22].

Algorithm 3: Gibbs Sampler algorithm.

Inputs: Full conditional pdf's: $\tilde{\pi}_t(\mathbf{x}_{1:t,1} | \mathbf{x}_{1:t,2}, \dots, \mathbf{x}_{1:t,d}), \dots, \tilde{\pi}_t(\mathbf{x}_{1:t,d} | \mathbf{x}_{1:t,1}, \dots, \mathbf{x}_{1:t,d})$; Sample from $\tilde{\pi}_t$:
 $\mathbf{x}_{1:t} = (\mathbf{x}_{1:t,1}, \dots, \mathbf{x}_{1:t,d})$;
 Sample $\mathbf{x}_{1:t,1}^* \sim \tilde{\pi}_t(\mathbf{x}_{1:t,1} | \mathbf{x}_{1:t,2}, \dots, \mathbf{x}_{1:t,d})$;
 Sample $\mathbf{x}_{1:t,2}^* \sim \tilde{\pi}_t(\mathbf{x}_{1:t,2} | \mathbf{x}_{1:t,1}^*, \mathbf{x}_{1:t,3}, \dots, \mathbf{x}_{1:t,d})$;
 \vdots
 Sample $\mathbf{x}_{1:t,d}^* \sim \tilde{\pi}_t(\mathbf{x}_{1:t,d} | \mathbf{x}_{1:t,1}^*, \dots, \mathbf{x}_{1:t,d-1}^*)$;
Result: New sample from $\tilde{\pi}_t$: $\mathbf{x}_{1:t}^* = (\mathbf{x}_{1:t,1}^*, \dots, \mathbf{x}_{1:t,d}^*)$;

Including both the resampling and the “move” steps into the generic SMC algorithm leads to the “Resample-Move” algorithm, first presented in [24] and subsequently widely used in the SMC literature.

The generic class of SMC algorithms whilst widely used in practice can not be directly applied to the problems addressed in this work, since all the distributions in the sequence (4.1) are defined over the same support, ie, $E_t = E$ and not $E_t = E \times \dots \times E$ as required by the SMC algorithms just described.

To overcome this problem a specialized variation of this method, named SMC Samplers is introduced in the next section.

5.2 SMC Samplers

Having presented the SMC class of algorithms, we now present in contrast to these the class of SMC Sampler algorithms which involve the same mechanism as the SMC algorithm also using a mutation, correction and resampling stage at each iteration. However, the class of SMC Sampler algorithm is importantly different in the space that the sequence of distributions being sampled from are defined upon. Differently from Section 5, our interest now is to approximate a generic sequence of probability distributions $\{\pi_t\}_{t=1}^T$ such that $\text{supp}(\pi_t) = \text{supp}(\pi_{t-1}) = E$ for all $t = 1, \dots, T$ (once again one may think of $E = \mathbb{R}^d$). Here we may also assume our target distribution is only known up to a normalizing constant, ie, $\pi_t(\mathbf{x}_t) = Z_t^{-1} f_t(\mathbf{x}_t)$. For notational clarity, functions in the enlarged space will be denoted with a upper tilde, as $\tilde{f}_t : E^t \rightarrow \mathbb{R}$.

The idea presented on [37] and [17] is to transform this problem into one resembling the usual SMC formulation, where the sequence of target distributions $\{\tilde{f}_t\}_{t=1}^T$ is defined on the product space, i.e., $\text{supp}(\tilde{f}_t) = E \times E \times \dots \times E = E^t$.

The construction of \tilde{f}_t (the density in the *path* space) is carried out as:

$$\tilde{f}_t(\mathbf{x}_{1:t}) = f_t(\mathbf{x}_t) \tilde{f}_t(\mathbf{x}_{1:t-1} | \mathbf{x}_t), \text{ for } t = 2, \dots, T \quad (5.3)$$

where $\tilde{f}_t(\mathbf{x}_{1:t-1} | \mathbf{x}_t)$ is a probability distribution on the space E^{t-1} , for all $\mathbf{x}_t \in E$. Similarly to (5.1) we can carry out the construction of the importance distribution at time t .

As noticed in [37] and [17], a wise choice for $\tilde{f}_t(\mathbf{x}_{1:t-1} | \mathbf{x}_t)$ is given by

$$\tilde{f}_t(\mathbf{x}_{1:t-1} | \mathbf{x}_t) = \prod_{s=1}^{t-1} L_s(\mathbf{x}_{s+1}, \mathbf{x}_s),$$

where the each L_s is the density of an (artificial) backward Markov kernel. It is important to note that, by construction, $\tilde{f}_t(\mathbf{x}_{1:t})$ admits $f_t(\mathbf{x}_t)$ as a marginal, since

$$\int \tilde{f}_t(\mathbf{x}_{1:t}) d\mathbf{x}_{1:t-1} = f_t(\mathbf{x}_t) \int \prod_{s=1}^{t-1} L_s(\mathbf{x}_{s+1}, \mathbf{x}_s) d\mathbf{x}_{1:t-1} = f_t(\mathbf{x}_t), \quad \forall t > 1.$$

Moreover, provided that \tilde{f}_t admits f_t as a marginal the normalizing constant of the enlarged density will be the same as the original density:

$$\int \tilde{f}_t(\mathbf{x}_{1:t}) d\mathbf{x}_{1:t} = \int \int \tilde{f}_t(\mathbf{x}_{1:t}) d\mathbf{x}_{1:t-1} d\mathbf{x}_t = \int f_t(\mathbf{x}_t) d\mathbf{x}_t = Z_t.$$

Now that we are back to the SMC framework from last section, we can easily write the SMC Sampler algorithm (Algorithm 4). Moreover, the Resample-Move strategy from Section 5.1 can still be utilized.

5.2.1 Backward kernels selection

The introduction of the sequence of kernels $\{L_{t-1}\}_{t=2}^T$ creates a new degree of freedom in SMC samplers when compared with usual SMC algorithms, where only the forward mutation kernels $\{K_t\}_{t=1}^T$ should be designed. In this section we will discuss how to, given the kernels $\{K_t\}_{t=1}^T$, optimize the choice of backward kernels $\{L_{t-1}\}_{t=2}^T$.

Denote by $q_t(\mathbf{x}_t)$ the marginal importance distribution at time t , which is given by

$$q_t(\mathbf{x}_t) = \int \tilde{q}_t(\mathbf{x}_{1:t}) d\mathbf{x}_{1:t-1} = \int q_1(\mathbf{x}_1) \prod_{j=2}^t K_j(\mathbf{x}_{j-1}, \mathbf{x}_j) d\mathbf{x}_{1:t-1}. \quad (5.4)$$

In the case in which we know how to calculate q_t in exact form we can simply approximate the target distribution f_t by a weighted sample $\{\mathbf{x}_t^{(j)}, W_t^{(j)}\}$, where $\mathbf{x}_t \sim q_t$ and W_t is the normalized version of $w_t := \frac{f_t(\mathbf{x}_t)}{q_t(\mathbf{x}_t)}$. From the definition of q_t we can see that sampling from q_t is simple if it is easy to sample from q_1 and from all the kernels K_t . On the other hand, the density of q_t will only be tractable if we are

Algorithm 4: SMC Sampler algorithm.

Inputs: IS density q_1 , (forward) mutation kernels $\{K_t(\mathbf{x}_{t-1}, \mathbf{x}_t)\}_{t=1}^T$, (artificial) backward kernels $\{L_{t-1}(\mathbf{x}_t, \mathbf{x}_{t-1})\}_{t=1}^T$, move kernel $M_t(\tilde{\mathbf{x}}_t, \mathbf{x}_t)$;

for $j = 1, \dots, N$ **do**

Sample $\mathbf{x}_1^{(j)}$ from $\mathbf{X}_1 \sim \tilde{q}_1(\cdot)$ (*Mutation step*);

Calculate the weights $w_1^{(j)} = \frac{\tilde{f}_1(\mathbf{x}_1^{(j)})}{\tilde{q}_1(\mathbf{x}_1^{(j)})}$;

end

Calculate the normalized weights $W_1^{(j)} = \frac{w_1^{(j)}}{\sum_{j=1}^N w_1^{(j)}}$ (*Correction step*);

for $t = 2, \dots, T$ **do**

for $j = 1, \dots, N$ **do**

Sample $\mathbf{x}_t^{(j)}$ from $\mathbf{X}_t | \mathbf{X}_{t-1} = \mathbf{x}_{t-1}^{(j)} \sim K_t(\mathbf{x}_{t-1}^{(j)}, \cdot)$ (*Mutation step*);

Using (5.3) and (5.1), calculate the weights $w_t^{(j)} = \frac{\tilde{f}_t(\mathbf{x}_{1:t}^{(j)})}{\tilde{q}_t(\mathbf{x}_{1:t}^{(j)})} = w_{t-1}^{(j)} \underbrace{\frac{f_t(\mathbf{x}_t^{(j)})L_{t-1}(\mathbf{x}_t^{(j)}, \mathbf{x}_{t-1}^{(j)})}{f_{t-1}(\mathbf{x}_{t-1}^{(j)})K_t(\mathbf{x}_{t-1}^{(j)}, \mathbf{x}_t^{(j)})}}_{\text{incremental weight: } \alpha(\mathbf{x}_{t-1}^{(j)}, \mathbf{x}_t^{(j)})}$;

end

Calculate the normalized weights $W_t^{(j)} = \frac{w_t^{(j)}}{\sum_{j=1}^N w_t^{(j)}}$ (*Correction step*). ;

if $ESS_t < N/2$ **then**

for $j=1, \dots, N$ **do**

Resample $\tilde{\mathbf{x}}_t^{(j)} = \mathbf{x}_t^{(k)}$ with prob. $W_t^{(k)}$ (*Resample step*) ;

Sample $\mathbf{x}_t^{(j)} \sim M_t(\tilde{\mathbf{x}}_t^{(j)}, \cdot)$ (*Move step*);

Set $W_t^{(j)} = 1/N$;

end

end

end

Result: Weighted random samples $\{\mathbf{x}_t^{(j)}, W_t^{(j)}\}_{j=1}^N$ approximating π_t , for all $t = 1, \dots, T$;

able to solve the marginalization integral (in $t - 1$ dimensions) – which, in practice will hardly ever be the case.

The introduction of backward kernels $\{L_{t-1}\}_{t=2}^T$ helps us (in most of the practical cases) to avoid the computation of q_t . On the other hand, since \tilde{f}_t and \tilde{q}_t admits, respectively, f_t and q_t as marginals, Lemma 5.2 tells us the price we need to pay: *an increase in the variance of the importance weights*. Fortunately, the same Lemma provides us some insights on how to optimally choose the backward kernels.

Lemma 5.2 *Let $f(\mathbf{x}_1, \mathbf{x}_2)$ and $g(\mathbf{x}_1, \mathbf{x}_2)$ be two probability densities with $\text{supp}(f) \subset \text{supp}(g)$. Then*

$$\text{Var}_g \left(\frac{f(\mathbf{X}_1, \mathbf{X}_2)}{g(\mathbf{X}_1, \mathbf{X}_2)} \right) \geq \text{Var}_g \left(\frac{f_1(\mathbf{X}_1)}{g_1(\mathbf{X}_1)} \right),$$

where $f_1(\mathbf{x}_1) = \int f(\mathbf{x}_1, \mathbf{x}_2) d\mathbf{x}_2$ and $g_1(\mathbf{x}_1) = \int g(\mathbf{x}_1, \mathbf{x}_2) d\mathbf{x}_2$.

Proof From the variance decomposition, we have that

$$\begin{aligned} \text{Var}_g \left(\frac{f(\mathbf{X}_1, \mathbf{X}_2)}{g(\mathbf{X}_1, \mathbf{X}_2)} \right) &= \text{Var}_g \left(\mathbb{E}_g \left[\frac{f(\mathbf{X}_1, \mathbf{X}_2)}{g(\mathbf{X}_1, \mathbf{X}_2)} \middle| \mathbf{X}_1 = \mathbf{x}_1 \right] \right) \\ &\quad + \mathbb{E}_g \left[\text{Var}_g \left(\frac{f(\mathbf{X}_1, \mathbf{X}_2)}{g(\mathbf{X}_1, \mathbf{X}_2)} \middle| \mathbf{X}_1 = \mathbf{x}_1 \right) \right] \\ &\geq \text{Var}_g \left(\mathbb{E}_g \left[\frac{f(\mathbf{X}_1, \mathbf{X}_2)}{g(\mathbf{X}_1, \mathbf{X}_2)} \middle| \mathbf{X}_1 = \mathbf{x}_1 \right] \right), \end{aligned}$$

since $f, g \geq 0$ (they are densities).

The result follows from the fact that the ratio of marginal densities can be rewritten as the following conditional expectation:

$$\begin{aligned} \frac{f_1(\mathbf{x}_1)}{g_1(\mathbf{x}_1)} &= \int \frac{f(\mathbf{x}_1, \mathbf{x}_2)}{g_1(\mathbf{x}_1)g_{2|1}(\mathbf{x}_2|\mathbf{x}_1)} g_{2|1}(\mathbf{x}_2|\mathbf{x}_1) d\mathbf{x}_2 \\ &= \mathbb{E}_g \left(\frac{f(\mathbf{X}_1, \mathbf{X}_2)}{g(\mathbf{X}_1, \mathbf{X}_2)} \middle| \mathbf{X}_1 = \mathbf{x}_1 \right). \end{aligned}$$

□

As mentioned previously, Proposition 5.3 shows how to design the backward kernels $\{L_{t-1}\}_{t=2}^T$ in order to minimize the variance of the importance weights.

Proposition 5.3 (Optimal backward kernel) *The kernel*

$$L_t^{opt}(\mathbf{x}_{t+1}, \mathbf{x}_t) := \frac{q_t(\mathbf{x}_t)K_{t+1}(\mathbf{x}_t, \mathbf{x}_{t+1})}{q_{t+1}(\mathbf{x}_{t+1})} \text{ is optimal in the sense that } \text{Var}_{\tilde{q}_t}(w_t^{opt}(\mathbf{X}_{1:t})) \leq \text{Var}_{\tilde{q}_t}(w_t(\mathbf{X}_{1:t})),$$

where $w_t^{opt}(\mathbf{x}_{1:t}) = \frac{f_t(\mathbf{x}_t)}{q_t(\mathbf{x}_t)}$.

Proof If we substitute the optimal kernel into the definition of importance weights we have that

$$\begin{aligned} w_t^{opt}(\mathbf{x}_{1:t}) &= \frac{\tilde{f}_t(\mathbf{x}_{1:t})}{\tilde{q}_t(\mathbf{x}_{1:t})} \\ &= \frac{f_t(\mathbf{x}_t) \prod_{s=1}^{t-1} L_s^{opt}(\mathbf{x}_{s+1}, \mathbf{x}_s)}{q_1(\mathbf{x}_1) \prod_{j=2}^t K_j(\mathbf{x}_{j-1}, \mathbf{x}_j)} \\ &= \frac{f_t(\mathbf{x}_t) \prod_{s=1}^{t-1} \frac{q_t(\mathbf{x}_t)K_{t+1}(\mathbf{x}_t, \mathbf{x}_{t+1})}{q_{t+1}(\mathbf{x}_{t+1})}}{q_1(\mathbf{x}_1) \prod_{j=2}^t K_j(\mathbf{x}_{j-1}, \mathbf{x}_j)} \\ &= \frac{f_t(\mathbf{x}_t)}{q_t(\mathbf{x}_t)}. \end{aligned}$$

The result, then, follows from Lemma 5.2.

From Proposition 5.3 we can see that if we know how to sample from $q_t(\mathbf{x}_t)$ then the SMC sampler algorithm reduces to a sequence of Importance Sampling steps, where at each time t we sample particles from $q_t(\mathbf{x}_t)$ and correct the bias through the weights $w_t(\mathbf{x}_t) = \frac{f_t(\mathbf{x}_t)}{q_t(\mathbf{x}_t)}$.

Independent kernel One situation where we know how to calculate $q_t(\mathbf{x}_t)$ is when $K_t(\mathbf{x}_{t-1}, \mathbf{x}_t)$ does not depend on \mathbf{x}_{t-1} , making the mutation step completely memory-less. As an abuse of notation, let $K_t(\mathbf{x}_t) := K_t(\mathbf{x}_{t-1}, \mathbf{x}_t)$. This choice is not always recommended to be used in practice due to difficulties in designing an appropriate kernel. In this case it is easy to see that it is possible to perform a sequence of Importance Sampling steps, since

$$\begin{aligned} q_t(\mathbf{x}_t) &= \int q_1(\mathbf{x}_1) \prod_{j=2}^t K_j(\mathbf{x}_{j-1}, \mathbf{x}_j) d\mathbf{x}_{1:t-1} \\ &= \int q_1(\mathbf{x}_1) \prod_{j=2}^t K_j(\mathbf{x}_j) d\mathbf{x}_{1:t-1} \\ &= \left(\int q_1(\mathbf{x}_1) d\mathbf{x}_1 \right) \left(\prod_{j=2}^{t-1} \int K_j(\mathbf{x}_j) d\mathbf{x}_{1:t-1} \right) \left(\int K_t(\mathbf{x}_t) d\mathbf{x}_t \right) \\ &= K_t(\mathbf{x}_t). \end{aligned}$$

Approximations of the optimal kernel Various approximations of the optimal backward kernel have been proposed in the literature (see, for example [17], Section 3.3.2) but here we will discuss only one of them.

If we rewrite the optimal backward kernel from Proposition 5.3 as

$$L_t^{opt}(\mathbf{x}_{t+1}, \mathbf{x}_t) = \frac{q_t(\mathbf{x}_t)K_{t+1}(\mathbf{x}_t, \mathbf{x}_{t+1})}{\int q_t(\mathbf{x}_t)K_{t+1}(\mathbf{x}_t, \mathbf{x}_{t+1})d\mathbf{x}_t}$$

it suggests that a sensible approximation for this kernel is to use π_t instead of q_t . In this case,

$$L_t^{opt}(\mathbf{x}_{t+1}, \mathbf{x}_t) \approx \frac{f_t(\mathbf{x}_t)K_{t+1}(\mathbf{x}_t, \mathbf{x}_{t+1})}{\int f_t(\mathbf{x}_t)K_{t+1}(\mathbf{x}_t, \mathbf{x}_{t+1})d\mathbf{x}_t}, \quad (5.5)$$

since the normalizing constants of π cancel out.

Although the integral in the denominator of (5.5) is (usually) not analytically tractable we can use the weighted sample $\{\mathbf{x}_{1:t}^{(j)}, w_t^{(j)}\}_{j=1}^N$ from π_t generated by the SMC sampler procedure to approximate L_t^{opt} as

$$L_t(\mathbf{x}_{t+1}, \mathbf{x}_t) = \frac{f_t(\mathbf{x}_t) K_{t+1}(\mathbf{x}_t, \mathbf{x}_{t+1})}{\sum_{j=1}^N w_t^{(j)} K_{t+1}(\mathbf{x}_t^{(j)}, \mathbf{x}_{t+1})}. \quad (5.6)$$

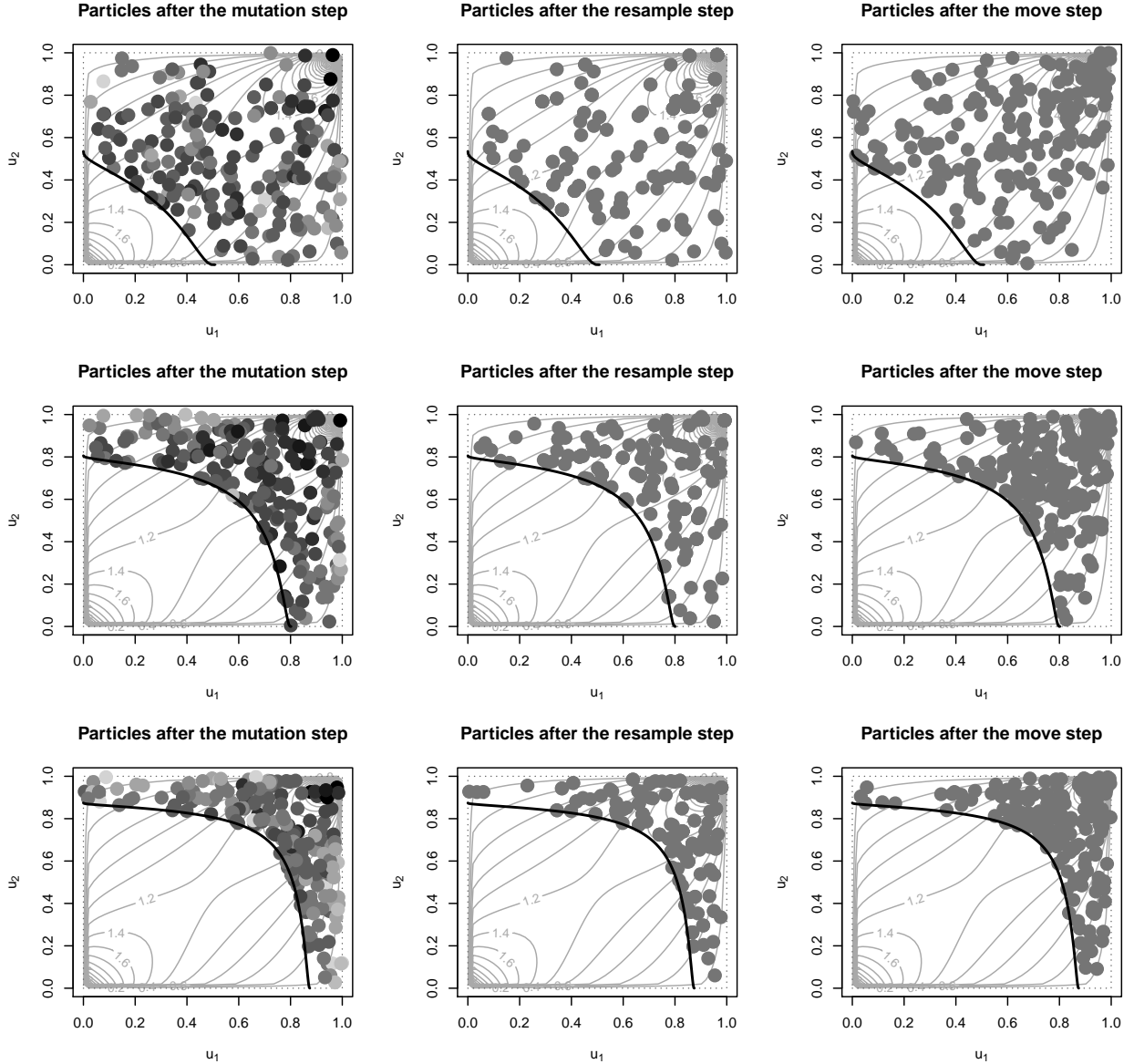


Fig. 5.1 Evolution of particles in the SMC algorithm, with marginals $X_{t,i} \sim LN(10 - 0.1i, 1 + 0.2i)$, $i = 1, 2$, $t = 1, \dots, T$; Gumbel($\theta = 1.5$); $N = 200$ particles and $T = 20$ equidistant levels from $B_1 = 0, \dots, B_T = 100,000$; (top) $t = 4$, (middle) $t = 10$, (bottom) $t = 14$. The level sets in the background represent the structure of the base, unconstrained copula.

5.2.2 Forward kernels selection

So far we have discussed how to design backward kernels L_t that are optimal for specific choices of forward kernels K_t . We now present possible choices of K_t in order to have simple forms of importance weights when used with the optimal choice of backward kernel.

In this sense, one convenient choice for the forward kernels K_t is to assume they are such that $K_t(\mathbf{x}_{t-1}, \mathbf{x}_t)$ has π_t as invariant density, ie,

$$\pi_t(\mathbf{x}_{t+1}) = \int K_{t+1}(\mathbf{x}_t, \mathbf{x}_{t+1}) \pi_{t+1}(\mathbf{x}_t) d\mathbf{x}_t.$$

Proceeding this way, we can choose the backward kernel as follows,

$$L_t(\mathbf{x}_{t+1}, \mathbf{x}_t) = \frac{\pi_{t+1}(\mathbf{x}_t)K_{t+1}(\mathbf{x}_t, \mathbf{x}_{t+1})}{\pi_{t+1}(\mathbf{x}_{t+1})}, \quad (5.7)$$

which is a reasonable approximation for the optimal backward kernel from Proposition 5.3. It also worth noticing the kernel defined on (5.7) is the reversal Markov kernel associated with K_{t+1} .

For theoretical purposes, we may also assume the forward kernel mixes perfectly, ie, $K_{t+1}(\mathbf{x}_t, \mathbf{x}_{t+1}) = \pi_{t+1}(\mathbf{x}_{t+1})$. This choice of kernels is obviously not feasible in practice, since $\pi_{t+1}(\mathbf{x}_{t+1})$ is precisely the density we are trying to sample from, but it can provide us interesting insights. In this case, the incremental weights of the SMC sampler algorithm (see Algorithm 4) are given by

$$\alpha(\mathbf{x}_{t-1}, \mathbf{x}_t) = \frac{f_t(\mathbf{x}_t)L_{t-1}(\mathbf{x}_t, \mathbf{x}_{t-1})}{f_{t-1}(\mathbf{x}_{t-1})K_t(\mathbf{x}_{t-1}, \mathbf{x}_t)} \propto \frac{\pi_t(\mathbf{x}_{t-1})}{\pi_{t-1}(\mathbf{x}_{t-1})},$$

which makes the weights at time t independent of the particles sampled at time t .

5.3 Generalizing the relative error expression to the path-space IS and SMC Sampler

As in Equation (3.4), throughout this section we will assume we are interested in estimating conditional expectations such as $\mathbb{E}[h(\mathbf{X}) | \mathbf{X} \in \mathcal{G}_{\mathbf{X}}] = \mathbb{E}[\varphi(\mathbf{U}) | \mathbf{U} \in \mathcal{G}_{\mathbf{U}}]$ where $h(\mathbf{X}) = h(F_1^{(-1)}(u_1), \dots, F_d^{(-d)}(u_d)) =: \varphi(\mathbf{U})$. Moreover, let us denote by $p_t = \mathbb{P}[\mathbf{X}_t \in \mathcal{G}_{\mathbf{X}_t}] = \mathbb{P}[\mathbf{U}_t \in \mathcal{G}_{\mathbf{U}_t}]$. Throughout this section we will denote by $\mathbb{E}_{\pi_T}[\cdot]$ the expectation with respect to the density π_T .

In general, given $\widehat{\varphi}$ an estimate for $\mathbb{E}[\varphi(\mathbf{U}) | \mathbf{U} \in \mathcal{G}_{\mathbf{U}}]$ a reasonable efficiency measure is given by its variance (the smaller the variance the better). In many cases, however, both the quantity of interest and the variance of the estimator are so small that using the variance as a measure efficiency is meaningless. For example, let us assume the quantities of interest are given by $z_s = \mathbb{P}[\mathbf{U} \in A_s]$, for some sequence of sets $A_s \downarrow \emptyset$ when $s \rightarrow \infty$. It is trivial to see that if the estimator of the probability via simple Monte Carlo is denoted by $\widehat{\varphi}_s$ then $\text{Var}(\widehat{\varphi}_s) \rightarrow 0$ when $s \rightarrow \infty$.

In such scenarios, as discussed in [4], Chapter VI, the relevant performance measure is to verify if the (unbiased) estimators possess bounded relative error, ie, if

$$\limsup_{s \rightarrow \infty} \frac{\text{Var}(\widehat{\varphi}_s)}{z_s^2} < \infty.$$

In this section we present the generic variance expression for three different estimators, targeting $\mathbb{E}_{\pi_t}[\varphi(\mathbf{U}_t)]$, namely: Importance Sampling (IS), Path space Importance Sampling (PIS) and Sequential Monte Carlo Sampler (SMCS). The reader should note, from (5.8), that the last two algorithm are *asymptotically* unbiased, but may have some finite sample bias. For some specific quantities, [1] show that their estimator will produce bounded relative error and the same is valid for the Importance Sampling algorithm from [32], when targeting small probabilities.

For SMC Samplers [17] provide two Central Limit Theorem (CLT) results in “extreme” cases: the first one deals with the case when no resampling is performed and the second one for the case of resampling (using multinomial resample) at each iteration of the algorithm. These two cases will be briefly reviewed in the sequel, in order for us to have interpretable results for the asymptotic variances.

In both cases, under suitable integrability conditions discussed in [17], Proposition 2, the following convergence in distribution is valid when the number of particles tends to infinity,

$$\sqrt{N} \left(\sum_{j=1}^N W_t^{(j)} \varphi(\mathbf{U}^{(j)}) - \mathbb{E}_{\pi_t}[\varphi(\mathbf{U}_t)] \right) \xrightarrow{N \rightarrow +\infty} N(0, \sigma_t^2), \quad \forall t = 1, \dots, T \quad (5.8)$$

where the normalized weights $W_t^{(j)}$ calculated in the SMC Sampler algorithm (see Algorithm 4) and the limiting variance σ_t^2 are dependent on the resampling strategy.

5.3.1 Importance Sampling (IS)

Let us assume we use a density $q_0(\mathbf{u}_0)$ as an importance distribution targeting $\pi_T(\mathbf{u}) = \pi(\mathbf{u})\mathbb{1}_{\{\mathbf{u} \in \mathcal{G}_{\mathbf{U}_T}\}}(\mathbf{u})/p_T$ (as in 4.1)). If we denote the Importance Sampling estimator for $\varphi(\mathbf{U})$ by

$$\widehat{\varphi}_{IS} := \frac{1}{N} \sum_{j=1}^N \varphi(\mathbf{u}_0^{(j)}) w_{IS}(\mathbf{u}_0^{(j)}), \quad (5.9)$$

where $w_{IS}(\mathbf{u}_0) := w_{IS}(\mathbf{u}_0) = \frac{\pi_T(\mathbf{u}_0)}{q_0(\mathbf{u}_0)}$ then we have that

$$\text{Var}(\widehat{\varphi}_{IS}) = \frac{1}{N} \left[\int \frac{\pi_T^2(\mathbf{u})}{q_0^2(\mathbf{u})} \varphi(\mathbf{u}) q_0(\mathbf{u}) d\mathbf{u} - \left(\int \frac{\pi_T(\mathbf{u})}{q_0(\mathbf{u})} \varphi(\mathbf{u}) q_0(\mathbf{u}) d\mathbf{u} \right)^2 \right].$$

In particular, if $\pi_T \geq q_0$ π_T -almost surely (eg, the IS distribution is the unconditional distribution π), then

$$\text{Var}(\widehat{\varphi}_{IS}) \geq \frac{1}{N} \text{Var}_{\pi_T}(\varphi(\mathbf{U})). \quad (5.10)$$

In the capital allocation problem we know the probability of the conditioning event is chosen to be $p_T = 1 - \alpha$, meaning that the target distribution π_T will be known (including its proportionality constant) if π is perfectly known. In this scenario, if the importance distribution is the unconditional density, ie, $q_0 \equiv \pi$ then the importance weights can be perfectly calculated. However, in many interesting cases either π or q_0 may only be known up to a constant. In this case it is necessary to use the “self-normalized” version of the importance weights, namely,

$$w_{IS-SN}^{(i)} = \frac{\pi_T(\mathbf{u}^{(i)})/q_0(\mathbf{u}^{(i)})}{\sum_{j=1}^N \pi_T(\mathbf{u}^{(j)})/q_0(\mathbf{u}^{(j)})}. \quad (5.11)$$

When the self-normalized importance weights are used to estimate $\varphi(\mathbf{U})$ it is easy to show the variance of the new estimator will be larger than the one defined in (5.9). More formally, if we define $\widehat{\varphi}_{IS-SN} := \sum_{j=1}^N \varphi(\mathbf{u}_0^{(j)}) w_{IS-SN}(\mathbf{u}_0^{(j)})$, then

$$\frac{1}{N} \text{Var}_{\pi_T}(\varphi(\mathbf{U})) \leq \text{Var}(\widehat{\varphi}_{IS}) \leq \text{Var}(\widehat{\varphi}_{IS-SN}).$$

5.3.2 Path space Importance Sampling (PIS)

When no resampling is performed in the SMC Sampler algorithm, it is easy to see that, at each time step, it collapses to an Importance Sampling algorithm in the Path space (PIS). Since the particles have no interaction, the (unnormalized) weights in (5.8) will be given by

$$w_{PIS,t}(\mathbf{U}_t) = \frac{\widetilde{\pi}_t(\mathbf{u}_{1:t})}{\widetilde{q}_t(\mathbf{u}_{1:t})},$$

where $\widetilde{q}_t(\mathbf{u}_{1:t}) = \widetilde{q}_1(\mathbf{u}_1) \prod_{j=2}^t K_j(\mathbf{u}_{j-1}, \mathbf{u}_j)$ and $\widetilde{\pi}_t(\mathbf{u}_{1:t}) = \pi_t(\mathbf{u}_t) \prod_{s=1}^{t-1} L_s(\mathbf{u}_{s+1}, \mathbf{u}_s)$. In this case the following result is valid.

Proposition 5.4 (Path Importance Sampling (PIS) CLT) *Under the integrability assumptions given in [17], Proposition 2, the asymptotic variance, as the number of particles increases to infinity, in (5.8) is given by*

$$\sigma_t^2 = \sigma_{PIS,t}^2 := \int \frac{\widetilde{\pi}_t^2(\mathbf{u}_{1:t})}{\widetilde{q}_t(\mathbf{u}_{1:t})} \{\varphi(\mathbf{u}_t) - \mathbb{E}_{\pi_t}[\varphi(\mathbf{U}_t)]\}^2 d\mathbf{u}_{1:t} \quad (5.12)$$

In the particular case where the forward kernel is perfectly mixing and the backward kernels are given by (5.7) then it is easy to show that

$$\sigma_{PIS,t}^2 = \int \frac{\pi_1^2(\mathbf{u}_0)}{q_0(\mathbf{u}_0)} d\mathbf{u}_0 \prod_{k=2}^t \int \frac{\pi_k^2(\mathbf{u}_{k-1})}{\pi_{k-1}(\mathbf{u}_{k-1})} d\mathbf{u}_{k-1} \text{Var}_{\pi_t}(\varphi(\mathbf{U})). \quad (5.13)$$

Moreover, let us assume the sequence of distributions is given by (4.1). Then, each integral can be simplified to

$$\begin{aligned} \int \frac{\pi_k^2(\mathbf{u}_{k-1})}{\pi_{k-1}(\mathbf{u}_{k-1})} d\mathbf{u}_{k-1} &= \int \frac{\pi^2(\mathbf{u}_{k-1}) \mathbb{1}_{\{\mathbf{u}_{k-1} \in \mathcal{G}_{\mathbf{U}_k}\}}(\mathbf{u}_{k-1})}{\pi(\mathbf{u}_{k-1}) \mathbb{1}_{\{\mathbf{u}_{k-1} \in \mathcal{G}_{\mathbf{U}_{k-1}}\}}(\mathbf{u}_{k-1})} \frac{p_{k-1}}{p_k^2} d\mathbf{u}_{k-1} \\ &= \frac{p_{k-1}}{p_k} \int \frac{\pi(\mathbf{u}_{k-1}) \mathbb{1}_{\{\mathbf{u}_{k-1} \in \mathcal{G}_{\mathbf{U}_k}\}}(\mathbf{u}_{k-1})}{p_k} d\mathbf{u}_{k-1} \\ &= \frac{p_{k-1}}{p_k}. \end{aligned}$$

But since the events become rarer the larger the index k , then $\frac{p_{k-1}}{p_k} > 1$ (and the same argument holds for the first integral, if, as in the IS algorithm, we start sampling from the unconditional distribution) and for N sufficiently large,

$$\text{Var}(\widehat{\varphi}_{PIS,T}) \approx \frac{1}{N} \sigma_{PIS,t}^2 > \frac{1}{N} \text{Var}_{\pi_T}(\varphi(\mathbf{U})).$$

5.3.3 Sequential Monte Carlo Sampler (SMCS) – resampling at every time step

When resampling is performed at every step of the SMC algorithm we can still calculate the variance σ^2 from (5.8) through the Proposition 5.14. The estimate calculated through this scheme will be denoted here $\widehat{\varphi}_{SMCS}$.

Proposition 5.5 (Always-Resampling SMC Sampler (SMCS) CLT) *Under the integrability assumptions given in [17], Proposition 2, the asymptotic variance in (5.8) is given by*

$$\begin{aligned} \sigma_t^2 = \sigma_{SMCS,t}^2 &:= \int \frac{\widetilde{\pi}_t^2(\mathbf{u}_1)}{\mu_t(\mathbf{u}_1)} \left(\int \varphi(\mathbf{u}_t) \widetilde{\pi}(\mathbf{u}_t | \mathbf{u}_1) d\mathbf{u}_t - E_{\pi_t}(\varphi(\mathbf{U})) \right)^2 d\mathbf{u}_1 \\ &+ \sum_{k=2}^{t-1} \int \frac{(\widetilde{\pi}_t(\mathbf{u}_k) L_{k-1}(\mathbf{u}_k, \mathbf{u}_{k-1}))^2}{(\pi_{k-1}(\mathbf{u}_{k-1}) K_k(\mathbf{u}_{k-1}, \mathbf{u}_k))} \left(\int \varphi(\mathbf{u}_t) \widetilde{\pi}(\mathbf{u}_t | \mathbf{u}_k) d\mathbf{u}_t - E_{\pi_t}(\varphi(\mathbf{U})) \right)^2 d\mathbf{u}_{k-1} d\mathbf{u}_k \\ &+ \int \frac{(\pi_t(\mathbf{u}_t) L_{t-1}(\mathbf{u}_t, \mathbf{u}_{t-1}))^2}{(\pi_{t-1}(\mathbf{u}_{t-1}) K_t(\mathbf{u}_{t-1}, \mathbf{u}_t))} (\varphi(\mathbf{u}_t) - E_{\pi_t}(\varphi))^2 d\mathbf{u}_{t-1} d\mathbf{u}_t, \end{aligned} \quad (5.14)$$

where $\widetilde{\pi}_n(\mathbf{u}_k) := \int \widetilde{\pi}_n(\mathbf{u}_{1:n}) d\mathbf{u}_{1:k-1} d\mathbf{u}_{k+1:n}$ and $\widetilde{\pi}_n(\mathbf{u}_n | \mathbf{u}_k) := \int \widetilde{\pi}_n(\mathbf{u}_{1:n}) d\mathbf{u}_{1:k-1} d\mathbf{u}_{k+1:n} / \widetilde{\pi}_n(\mathbf{u}_k)$.

As in the previous section, if we assume the mutation kernel is perfectly mixing and if we use the approximation of the optimal kernel given by (5.7) then the variance in (5.14) can be simplified to

$$\sigma_{SMCS,t}^2 = \int \frac{\pi_t^2(\mathbf{u}_{t-1})}{\pi_{t-1}(\mathbf{u}_{t-1})} d\mathbf{u}_{t-1} \text{Var}_{\pi_t}(\varphi(\mathbf{U})), \quad (5.15)$$

from which it becomes clear from (5.13) and (5.15) that $\sigma_{PIS,t}^2 > \sigma_{SMCS,t}^2$, which implies that

$$\text{Var}(\widehat{\varphi}_{SMCS,T}) < \text{Var}(\widehat{\varphi}_{PIS,T}).$$

6 Design of a SMC sampler with linear constraints for capital allocation

In this section we return to the problem of sampling from the distribution of $(\mathbf{X} | \sum_{i=1}^d X_i > B)$ producing samples from $\mathbf{U} \in \mathcal{G}_{\mathbf{U}}$, as explained in section 3.1. To use the algorithm specified in Section 5.2 we still need to design: (1) the forward kernels $K_t(u_{t-1}, u_t)$, (2) the backward kernels $L_{t-1}(u_t, u_{t-1})$ and (3) a Markov Chain move kernel M (in the spirit of Section 5.1). For the backward kernel we will use the approximation to the optimal one presented in section 5.2.1; the forward kernel and the “move” kernel will be presented, respectively in sections 6.1 and 6.2

6.1 Forward Kernel

For the forward kernel $K_t(\mathbf{u}_{t-1}, \mathbf{u}_t)$, if we can guarantee that any move from \mathbf{u}_{t-1} to \mathbf{u}_t will already be in $\mathcal{G}_{\mathbf{U}_t}$ then we will not loose any particle in the mutation step, improving the efficiency of the algorithm. Since we are developing the sampling procedure in $[0, 1]^d$ then, under the assumption that we can precisely characterize the constraint region $\mathcal{G}_{\mathbf{U}_t}$ (ie, we can calculate F_i^{-1} for all $i = 1, \dots, d$) then we can propose a “slice-sampling” procedure for K_t .

The idea of this type of kernel is that we can first sample $d - 1$ coordinates of the \mathbf{u}_t vector (chosen randomly) and then, conditional on these values, sample the last component constrained to an interval that will ensure that $\sum_{i=1}^d x_i > B_t$. In general these kernels will look like

$$K_t(\mathbf{u}_{t-1}, \mathbf{u}_t) = \sum_{m=1}^d \left[K_t^{(-m)}(\mathbf{u}_{t-1, -m}, \mathbf{u}_{t, -m}) K_t^{(m)}(u_{t-1, m}, u_{t, m}) \right] p_m, \quad (6.1)$$

where p_m is the probability of the m -th coordinate being the last one to be chosen,

$\mathbf{u}_{t, -m} = (u_{t, 1}, \dots, u_{t, m-1}, u_{t, m+1}, \dots, u_{t, d})$ is the vector \mathbf{u}_t without its m -th coordinate and $K_t^{(-m)}$ is the kernel that moves the $d - 1$ dimensions of $\mathbf{u}_{t-1, -m}$ to time t . Similarly, $K_t^{(m)}$ denotes the kernel that moves $u_{t-1, m}$ to $u_{t, m}$ ensuring that $\sum_{i=1}^d x_{t, i} > B_t$.

To guarantee the condition is satisfied, $K_t^{(m)}$ needs to be defined over $[B_t^u(m), 1]$, where

$$B_t^u(m) := F_m^{-1}(B_t^x(m)) \quad (6.2)$$

with

$$B_t^x(m) := \max \left\{ 0, B_t - \sum_{\substack{i=1 \\ i \neq m}}^d F_i(u_{t, i}) \right\}. \quad (6.3)$$

For simplicity, we can choose the last move to be uniformly distributed in $[B_t^u(m), 1]$, leading to

$$K_t^{(m)}(u_{t-1, m}, u_{t, m}) = \frac{u_{t, m}}{1 - B_t^u(m)} \mathbb{1}_{\{u_{t, m} \in [B_t^u(m), 1]\}}(u_{t, m}).$$

Again, for the sake of simplicity, we will only discuss the case where $K_t^{(-m)}$ consists of independent moves in each dimension, ie,

$$K_t^{(-m)}(\mathbf{u}_{t-1}, \mathbf{u}_t) = \prod_{\substack{i=1 \\ i \neq m}}^d K_t^{(-m, i)}(u_{t-1, i}, u_{t, i}). \quad (6.4)$$

Moreover, it will be assumed that $p_m = 1/d$, for all $m = 1, \dots, d$.

Uniform moves in $\mathcal{G}_{\mathbf{U}}$ The first (naïve) idea is to define the move in each component of \mathbf{u} as uniform, leading to a marginal kernels $K_t^{(-m, i)}(u_{t-1, i}, u_{t, i}) = u_{t, i} \mathbb{1}_{\{u_{t, i} \in [0, 1]\}}(u_{t, i})$ and, consequently,

$$K_t(\mathbf{u}_{t-1}, \mathbf{u}_t) = \frac{1}{d} \sum_{m=1}^d \left(\prod_{\substack{i=1 \\ i \neq m}}^d u_{t, i} \mathbb{1}_{\{u_{t, i} \in [0, 1]\}}(u_{t, i}) \right) \left(\frac{u_{t, m}}{1 - B_t^u(m)} \mathbb{1}_{\{u_{t, m} \in [B_t^u(m), 1]\}}(u_{t, m}) \right).$$

As we can see from the construction of this kernel, it is clearly independent of u_{t-1} and the comments by the end of Section 5.2.1 apply, meaning that the problem reduces to a series of Importance Sampling problems.

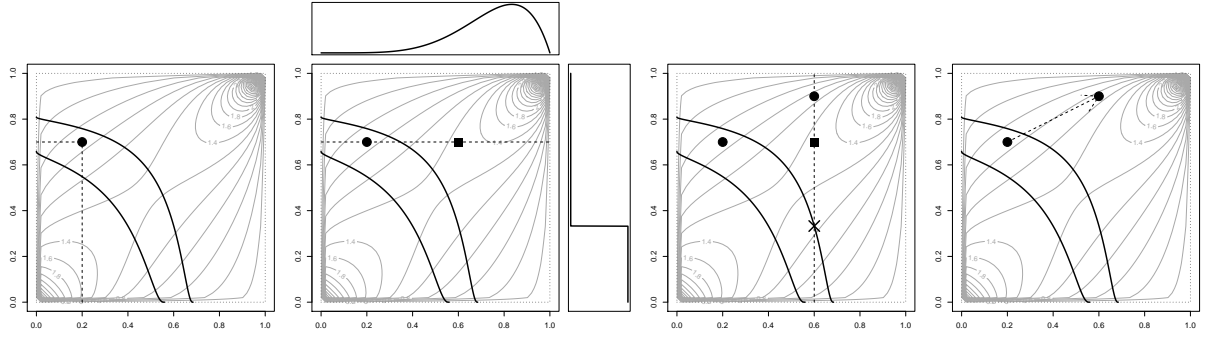


Fig. 6.1 Example of the Global Beta kernel for a Gumbel(1.5) copula with Log-Normal marginals: $(\mu_1 = 0.6, \sigma_1 = 1.4)$, $(\mu_2 = 0.4, \sigma_2 = 1)$. The boundary are such that $F_1^{-1}(u_1) + F_2^{-1}(u_2) > 2.25$ and 3.57 .

Global adaptive Beta moves in $[0, 1]^{d-1}$ One strategy to use the information contained in \mathbf{u}_{t-1} in the mutation step is to use the whole set of weighted particles at $t-1$ to estimate the parameters of the mutation kernel (subject to some restriction).

Since our kernels are defined in $[0, 1]$ a reasonable idea is to use a global Beta kernel, in the sense that all particles at time $t-1$ will be mutated through the same kernel. To select the parameters of the Beta distribution we match the first two moments of the Beta distributions with its sample moments at time $t-1$. Formally, let us denote $\{\mathbf{u}_{t-1}^{(j)}, W_{t-1}^{(j)}\}_{j=1}^N$ the set of N weighted particles at time $t-1$ and $K_t^{(-m,i)}(u_{t-1,i}, u_{t,i}) = \text{Beta}(u_{t,i}; \alpha_{t-1,i}, \beta_{t-1,i})$, where the RHS term denotes the density of a random variable $Y_{t-1,i}$ Beta distributed with parameters $\alpha_{t-1,i}$ and $\beta_{t-1,i}$ evaluated at $u_{t,i}$. Then matching the first two moments we have

$$\begin{aligned} \mathbb{E}[Y_{t-1,i}] &= \frac{\alpha_{t-1,i}}{\alpha_{t-1,i} + \beta_{t-1,i}} = \sum_{j=1}^N W_{t-1}^{(j)} u_{t-1,i}^{(j)} =: \hat{\mu}_{t-1,i}, \\ \text{Var}(Y_{t-1,i}) &= \frac{\alpha_{t-1,i} \beta_{t-1,i}}{(\alpha_{t-1,i} + \beta_{t-1,i})^2 (\alpha_{t-1,i} + \beta_{t-1,i} + 1)} \approx \hat{\mu}_{t-1,i}^2 - \sum_{j=1}^N W_{t-1}^{(j)} \left(u_{t-1,i}^{(j)}\right)^2 =: \hat{\sigma}_{t-1,i}^2 \end{aligned}$$

and after some algebra we find

$$\begin{aligned} \hat{\alpha}_{t-1,i} &= \frac{(1 - \hat{\mu}_{t-1,i})}{\hat{\sigma}_{t-1,i}^2} \hat{\mu}_{t-1,i}^2 \\ \hat{\beta}_{t-1,i} &= \hat{\alpha}_{t-1,i} \left(\frac{1}{\hat{\mu}_{t-1,i}} - 1 \right). \end{aligned}$$

Therefore an approximation for the mutation kernel at time t and dimension i – as in (6.4) – is given by

$$K_t^{(-m,i)}(u_{t-1,i}, u_{t,i}) \approx \text{Beta}(u_{t,i}; \hat{\alpha}_{t-1,i}, \hat{\beta}_{t-1,i}). \quad (6.5)$$

Remark 6.1 It is important to emphasize the kernel in (6.5), when plugged into (6.4) and (6.1), does not require any tuning and is not independent of $u_{t-1,i}$, since the parameters of the Beta distribution depend on these values.

Figure 6.1 exemplifies the mutation of one particle $u_{t-1} = (0.2, 0.7)$ (which is in the $(t-1)$ -th level set) to $u_t = (0.6, 0.9)$ (which is in the t -th level set). The mutation starts moving the first coordinate of u_{t-1} through a Beta distribution and then the second coordinate is moved following a uniform distribution defined in the appropriate region.

6.2 Markov Chain move kernel

Since the forward kernels designed in Section 6.1 ensure the new particles will satisfy the condition at level t one possibility is to use the same kernel as a proposal in a M-H algorithm. The drawback would be that in higher dimensions the acceptance rate of the M-H would be extremely small. Instead, here we propose the usage of a Gibbs sampling algorithm, that should be always preferred when the full conditional densities are known.

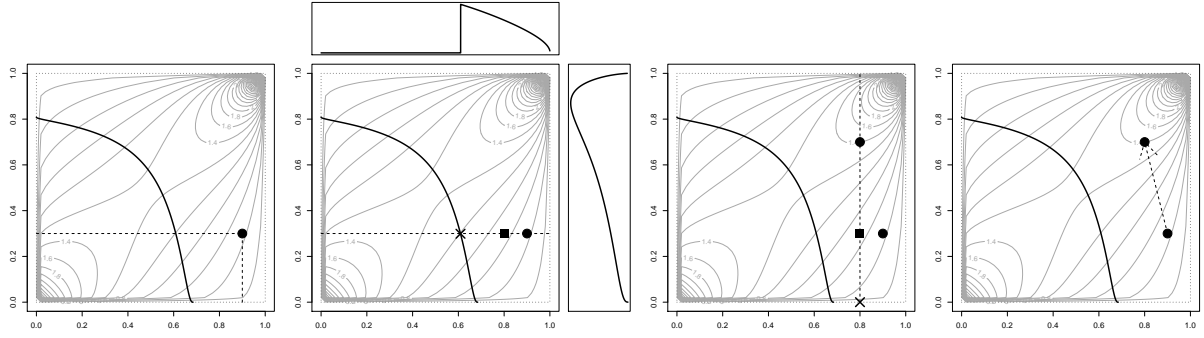


Fig. 6.2 Example of the move kernel for a Gumbel(1.5) copula with Log-Normal marginals: $(\mu_1 = 0.6, \sigma_1 = 1.4)$, $(\mu_2 = 0.4, \sigma_2 = 1)$. The boundary is such that $F_1^{-1}(u_1) + F_2^{-1}(u_2) > 3.57$.

Suppressing the dependence in t in the vector \mathbf{u} and denoting $v^*(m) := (u_1^*, \dots, u_m^*, u_{m+1}, \dots, u_d)$ we have that the full conditional for a generic $m = 1, \dots, d$ can be written as

$$\begin{aligned} \pi_t(u_m^* | u_1^*, \dots, u_{m-1}^*, u_{m+1}, \dots, u_d) &= \frac{\pi_t(u_1^*, \dots, u_m^*, u_{m+1}, \dots, u_d)}{\pi_t(u_1, \dots, u_{m-1}, u_{m+1}, \dots, u_d)} \\ &\propto \pi_t(v^*(m)) \\ &\propto c(v^*(m)) \mathbb{1}_{\{v^*(m) \in \mathcal{G}_{\mathbf{U}_t}\}}(v^*(m)). \end{aligned}$$

Note that the full conditional distribution for the m -th coordinate of \mathbf{u} is a probability distribution for u_m^* . On the other hand, since $u_1^*, \dots, u_{m-1}^*, u_{m+1}, \dots, u_d$ are fixed, we can rewrite the condition $[v^*(m) \in \mathcal{G}_{\mathbf{U}_t}]$ as $u_m^* \in [B^u(m), 1]$, with $B^u(m)$ as in (6.2).

To sample u_m^* from its full conditional distribution one can use a univariate slice sampler algorithm (see [35]), which only requires the full conditional target up to a normalizing constant. In Figure 6.2 we present an example of such a Markov Chain move. On the RHS, the initial point is $(u_1, u_2) = (0.9, 0.3)$. First the support of the full conditional distribution is calculated, ie, $B^u(1) = 0.6$ and plotted as a cross on the second figure. Then, a value $u_1^* = 0.8$ is sampled from $\pi(u_1 | u_2 = 0.3)$ (a square in the second plot). On top of the second plot we present the full conditional distribution (truncated on the left at $B^u(1) = 0.6$). For this value we find that $B^u(2) = 0$ and the support of the next full conditional distribution is $[0, 1]$ (the actual density is plotted vertically). The second coordinate $u_2^* = 0.7$ is then sampled from $\pi(u_2 | u_1^* = 0.8)$. In the last plot we have the final move, from $(u_1, u_2) = (0.9, 0.3)$ to $(u_1^*, u_2^*) = (0.8, 0.6)$.

7 Case Studies

In this section we present some simulation examples of the performance of the proposed Copula-Constrained SMC Sampler algorithm. For all the simulations and density calculations we made extensive use of the R-package `copula` [28].

The aim of all methods presented here will be to calculate conditional expectations of the form

$$\mathbb{E} \left[X_k | \sum_{i=1}^d X_i > VaR_\alpha \left(\sum_{i=1}^d X_i \right) \right], \quad \text{for } k = 1, \dots, d \quad (7.1)$$

where the α quantile (VaR_α) is assumed to be perfectly known (see comments below) and $\alpha \in (0.1, 0.2, \dots, 0.9, 0.95, 0.99, 0.995, 0.999, 0.9995, 0.9999, 0.99995)$. Moreover, as in [1], we will assume the marginal distributions of \mathbf{X} are Log-Normal with

$$X_i \sim LN(10 - 0.1i, 1 + 0.2i), \quad i = 1, \dots, d.$$

For all the examples presented here, since we are not able to express the VaR_α of the aggregated process in a closed form, the first step is to calculate a reliable approximation of this quantity, for each level α of interest. This is done through a Monte Carlo simulation of the loss vector $\mathbf{X} = (X_1, \dots, X_d)$ – from which we can compute the aggregate loss – of fixed size $N_q = 10,000,000$. Given a particular sample of size N_q , all the quantiles of the aggregate loss can be calculated. This process is, then, repeated for $N_{rep} = 500$ times, and the α -quantile is set as the average of the α -quantiles over all the 500 runs. The

reader should note that the estimate of extreme quantiles (for example $\alpha = 0.9999$) will be less precise than the estimate of lower quantiles (such as $\alpha = 0.3$), but for the purpose of comparing the proposed algorithm with competing strategies this is irrelevant, as long as the quantile used in the conditioning argument of (7.1) is the same for all methods.

After calculating the quantiles for all levels α (which are, from now on, assumed to be exact) the baseline comparison values for the expectations in (7.1) are calculated as follows. For each level α we sample as many loss vectors \mathbf{X} as necessary in order to have a sample of size $N_{MC} = 1,000$ satisfying the condition $\sum_{i=1}^d X_i > VaR_\alpha(\sum_{i=1}^d X_i)$. At this point we note that this naive Monte Carlo sampling strategy is very inefficient and would never be utilized in practice, due to the huge computational cost, but it provides us reference comparison to our more efficient SMC Sampler. To perform these simulations we required the usage of hundreds of cores from UCL Legion High Performance Computing Facility.

The expectations in (7.1) are, then, estimated as

$$\widehat{\varphi}_{MC} = \frac{1}{N_{MC}} \sum_{j=1}^{N_{MC}} X_i^{(j)},$$

where $\mathbf{X}^{(j)}$, $j = 1, \dots, N_{MC}$ are the samples satisfying the condition. This procedure is repeated $N_{rep} = 500$ and the Monte Carlo estimate $\widehat{\varphi}_{MC}$ is taken as the average over these N_{rep} repetitions, as follows

$$\bar{\varphi}_{MC} = \frac{1}{N_{rep}} \sum_{k=1}^{N_{rep}} \widehat{\varphi}_{MC}^{(k)},$$

where $\widehat{\varphi}_{MC}^{(k)}$ stands for the estimate (using N_{MC} particles) from the k -th run (out of N_{rep}). Analogously we can also define the variance of the MC estimator, $Var(\widehat{\varphi}_{MC})$. Moreover, we will denote by N_{SMC} the number of particles used in the SMC algorithm, and by $Var(\widehat{\varphi}_{SMC})$ the variance of its estimate, also calculated using N_{rep} runs.

One may observe that the expected number of samples in the Monte Carlo scheme in order to have N_{MC} samples satisfying the α condition is equal to $M_{MC} = N_{MC}/(1 - \alpha)$, which can be prohibitive if α is very close to 1.

For all the examples, the efficiency of the algorithm will be measured with respect to the Variance Reduction when compared with a simple Monte Carlo scheme (properly normalized). More formally, if the SMC algorithm uses T levels to approximate (7.1) then we will denote by Variance Reduction the ratio:

$$\text{Variance Reduction} = N_{MC} \times Var(\widehat{\varphi}_{MC}) \Big/ T \times N_{SMC} \times Var(\widehat{\varphi}_{SMC}). \quad (7.2)$$

We note this is a conservative measure of the Variance Reduction, as typically practitioners may only use in the denominator $N_{SMC} \times Var(\widehat{\varphi}_{SMC})$. In addition, the Variance Reduction must be analysed in conjunction with the estimation bias. For this purpose we will study the Relative Bias, defined as the relative difference from the SMC estimate to the MC estimate (assumed to be the truth, due to the very large sample sizes taken):

$$\text{Relative Bias} = \frac{\widehat{\varphi}_{SMC} - \widehat{\varphi}_{MC}}{\widehat{\varphi}_{MC}}$$

If the level of interest of the expectations in (7.1) is, for example, $\alpha = 0.999$ then, the SMC algorithm designed here will use as intermediate levels the quantiles $\alpha = 0.1, 0.2, \dots, 0.9, 0.95, 0.99, 0.995$. Although expectations conditional on quantiles at lower levels, such as $0.1, \dots, 0.9$ are not of direct interest for risk managers, as a by-product of the SMC algorithm, weighted samples from all the intermediate levels will be created and all the conditional expectations can be estimated.

7.1 Clayton copula dependence between risk cells

In this first study we assume we have a simple business unit and risk cell structure in which it is assumed that the dependence is on the annual losses and given simply by a Clayton copula. We first study a representative simple case, with dimension $d = 5$ (see Definition B.4) and investigate the behaviour of the proposed algorithm for different parameters values with fixed number of particles $N_{SMC} = 250$. To choose the parameters of interest we set the multivariate coefficient of (lower) tail dependence λ_l (see [15] or Definition B.5 and Figure 7.1) to be approximately equals to 0.25, 0.5, 0.75 and 0.9, which led to

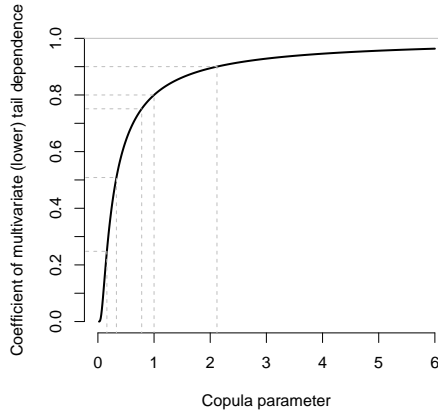


Fig. 7.1 Coefficient of multivariate lower tail dependence for a 5-dimensional Clayton copula.

$\theta = 0.16, 0.33, 0.78$ and 2.12 (see Figure 7.1). To compare with the results presented in [1], the parameter value $\theta = 1$ is also considered.

On Figure 7.2, we present the Relative Bias (top row) and Variance Reduction (bottom row) for all the range of different copula parameters and quantile levels. For ease of presentation, only three expectations are shown, namely the first marginal ($i = 1$), the last marginal ($i = 5$) and the sum of all marginal expectations (which is precisely the Expected Shortfall for the aggregated loss). From the Relative Bias analysis one can see that, regardless of the copula parameter and quantile levels, the estimation error is always smaller than 4% (in absolute value).

Since the estimates are unbiased, it makes sense to look at the Variance Reduction set of plots (bottom row of Figure 7.2). In the vertical axis of the plot the $\log_{10}(\text{Variance Reduction})$ is presented, meaning that, for example, the variance of the SMC algorithm is $10^{1.17} \approx 15$ times smaller than the MC scheme when $\theta = 2.12$ and $\alpha = 0.999$. The horizontal line at 0 defines the threshold where the SMC method outperforms a simple Monte Carlo: when the Variance Reduction is below the line the MC variance is smaller. As one should expect, for lower quantile levels a simple MC scheme should be preferred over the SMC method, since the condition in (7.1) can be easily satisfied with a reasonably small sample size. On the other hand, as soon as the conditioning event becomes rarer, the variance of the simple Monte Carlo scheme starts to increase polynomially fast when compared with the variance of the proposed SMC algorithm.

The rarer the conditioning event the more computationally efficient it becomes to use the SMC Sampler method proposed. As previously mentioned, when $\alpha \approx 1$ the number of Monte Carlo samples required in order to generate one sample satisfying the conditioning event increases like $1/(1 - \alpha)$. On the contrary, the SMC sampler is constrained to always use a fixed number of particles, independently of the rareness of the event. This is a significant advantage of such an approach.

For the ACH Importance Sampling algorithm of [1], discussed in Section 3.3, we follow the suggestion proposed in the original work involving the use of a discrete version of the optimal mixing distribution F_A (see (3.6)) with mass concentrated on the following 20 points: $x_k = 1 - 0.5^k$, $k = 1, \dots, 20$. For a fixed quantile level α and parameter θ , the calibration of F_A follows the proposed procedure in [1], Section 6.1 which uses the stop-loss as the objective function

$$\tilde{\Psi}(\mathbf{u}) = \max \left\{ \sum_{i=1}^d F_i^{-1}(u_i) - VaR_\alpha \left(\sum_{i=1}^d F_i^{-1}(u_i) \right), 0 \right\}.$$

Whilst the SMC algorithm is asymptotically unbiased (although it can be seen from Figure 7.2 the bias can be negligible even for finite N_{SMC}) the IS - ACH is unbiased for any finite sample size $N_{IS} < \infty$. Therefore it is not necessary to analyse the Relative Bias of the method.

On the other hand, following the notation on Section 3.3 for a fixed parameter θ a new efficiency measure can be studied as a function of α . We will denote by $P_{IS}(\alpha)$ the “percentage of particles with non-zero weight” for the α quantile. Formally this quantity is defined as

$$P_{IS}(\alpha) = \frac{\mathbb{E}[\tilde{N}]}{\mathbb{E}[N_{\mathbf{V}}]N_{IS}}, \quad (7.3)$$

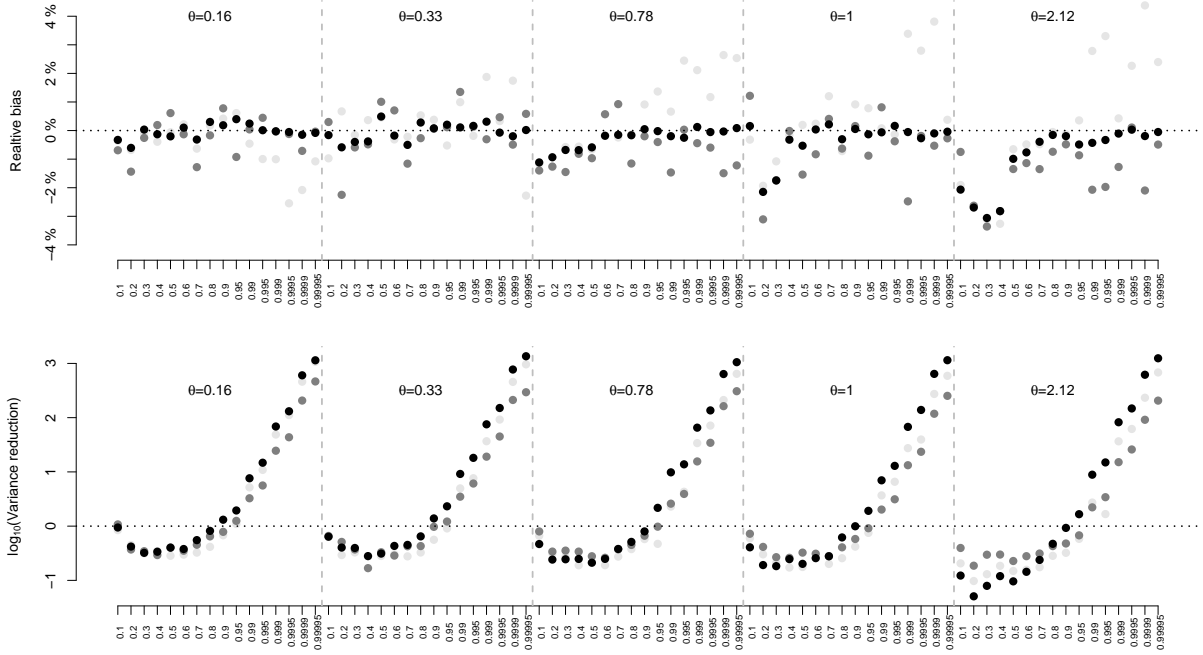


Fig. 7.2 Relative Bias (top) and Variance Reduction (bottom) for the 5-dimensional Clayton copula using the SMC algorithm. Using the notation from (7.1), \bullet Marginal for $i = 1$, \bullet Marginal for $i = 5$, \bullet Sum of all the marginal conditional expectations (Expected Shortfall).

where N_{IS} is the desired sample size of the algorithm and \tilde{N} and $\mathbb{E}[N_{\mathbf{V}}]$ are, respectively, the number of particles with non-zero weight and the number of draws in the rejection algorithm in order to have one sample from $F_{\mathbf{V}}$ (see Section 3.3). Intuitively we should expect some Variance Reduction if, and only if, the quantity $P_{IS}(\alpha)$ is larger than $1 - \alpha$.

As in the analysis made for the SMC algorithm, for the IS-ACH we also look at the (rescaled) Variance Reduction. To take into account the rejection steps in the algorithm we worked with the following Variance Reduction formula

$$\text{Variance Reduction} = N_{MC} \times \text{Var}(\hat{\varphi}_{MC}) \Big/ \mathbb{E}[N_{\mathbf{V}}] \times N_{IS} \times \text{Var}(\hat{\varphi}_{IS}). \quad (7.4)$$

From Figure 7.3 (top) we can see that the percentage of particles with non-zero weight, $P_{IS}(\alpha)$, is always smaller than the $1 - \alpha$, indicating an inefficiency of the IS - ACH. This inefficiency is verified in the bottom of the same figure, where the scaled Variance Reduction (as of 7.4) is presented. As in the SMC case, the Variance Reduction factor increases as a function of α , but in the IS-ACH case it is always smaller than 1. Although this is the case, we can expect the method to be efficient (in the Variance Reduction sense) as we get even closer to $\alpha = 1$.

7.2 Gumbel copula dependence between risk cells

In the second example we analyse the impact of the dimension in the estimation of conditional expectations when the copula is assumed to be from the Gumbel family (see Definition B.4). In this case we have chosen one parameter value ($\theta = 1.25$) in order to have values for the coefficient of multivariate upper tail dependence (λ_u from [15] and Definition B.5) ranging from very mild dependence ($\lambda_u \approx 0.25$) up to very strong dependence ($\lambda_u \approx 0.9$) in a highly constrained copula density, ie, a single Gumbel copula in up to 7 dimensions.

The SMC algorithm was studied for examples including dimensions $d = 2, 3, \dots, 7$ with $N_{SMC} = 250$ particles and the results are presented on Figure 7.5. From the top row we can see the Relative Bias of the conditional expectations for low dimensional copulas (eg, $d = 2$ or $d = 3$) is well behaved, being at most 5% of the true (Monte Carlo) value when $d = 2$ for all quantiles, but when the dimensionality of the problem increases a larger bias is observed. This is expected, as a single Gumbel copula in 7 dimensions, for instance, is highly constrained and its mass is mostly concentrated in a small area of the

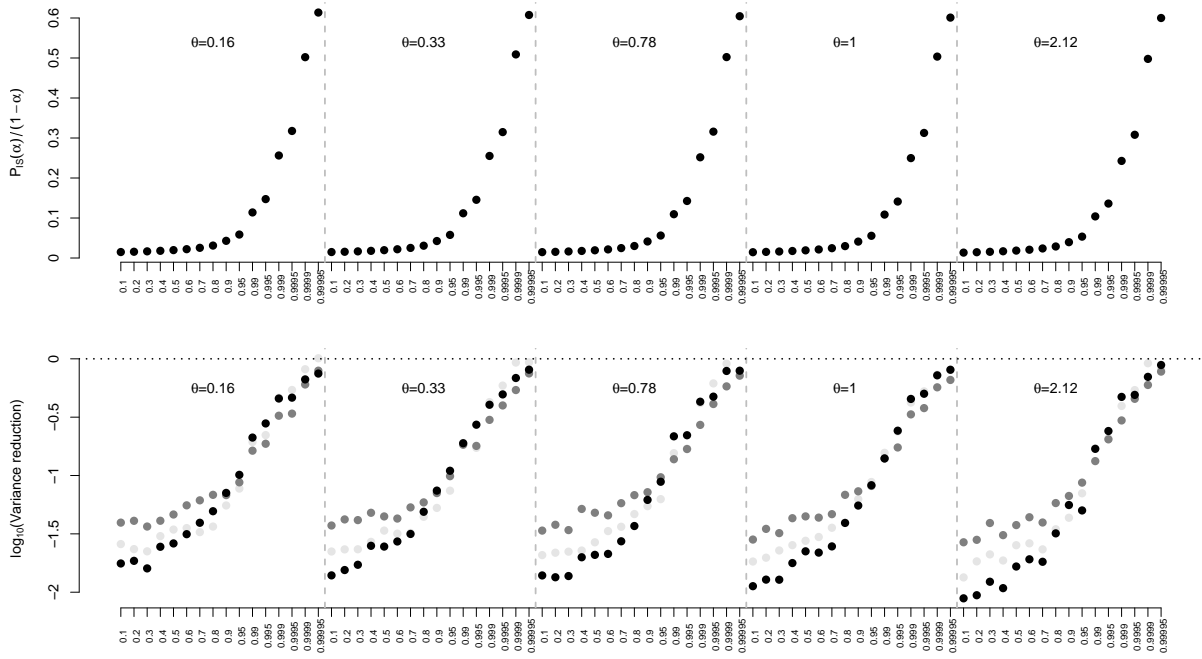


Fig. 7.3 Ratio between the percentage of particles with non-zero weight and $1 - \alpha$ (top) and Variance Reduction (bottom) for the 5-dimensional Clayton copula using the IS - ACH algorithm. Using the notation from (7.1), \bullet Marginal for $i = 1$, \bullet Marginal for $i = 5$, \bullet Sum of all the marginal conditional expectations (Expected Shortfall).

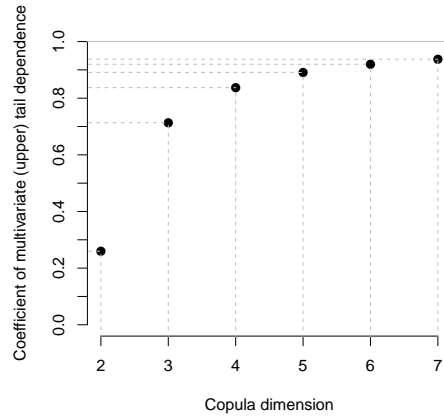


Fig. 7.4 Coefficient of multivariate upper tail dependence for a Gumbel(1.25) copula.

upper right quadrant of the 7-dimensional hypercube $[0, 1]^7$. In the worst case, for the first marginal of a d dimensional copula the Relative Bias reaches more than 40% of the true value. To reduce this bias, one must increase the number of particles in the SMC Sampler. Here we have selected a very conservative set of $N_{SMC} = 250$ particles. Next we studied the bias reduction as the number of particles increases, verifying the asymptotic unbiasedness of the SMC Sampler when $N_{SMC} \rightarrow \infty$.

From the bottom row of Figure 7.5 we can see that in all dimensions presented the SMC method is highly effective regarding decreasing the variance of the estimates when the quantile is larger than 0.999 but, as in the Clayton case, it is less efficient than a simple MC when the quantile is low.

Even though the bias involved in the SMC procedure is large for dimensions larger than $d = 3$, Figure 7.6 shows that one can decrease the absolute bias by increasing the sample size used in the SMC algorithm. For example, for the first marginal in 6 dimensions the Relative Bias goes from -35% to -30% when the number of particles increases from $N_{SMC} = 250$ to $N_{SMC} = 1,000$. The drawback of the increase in the sample size is that the method gets less effective in the Variance Reduction sense, although even in the case where $d = 6$ and $N_{SMC} = 1,000$ we still observe some humble improvement in the variance.

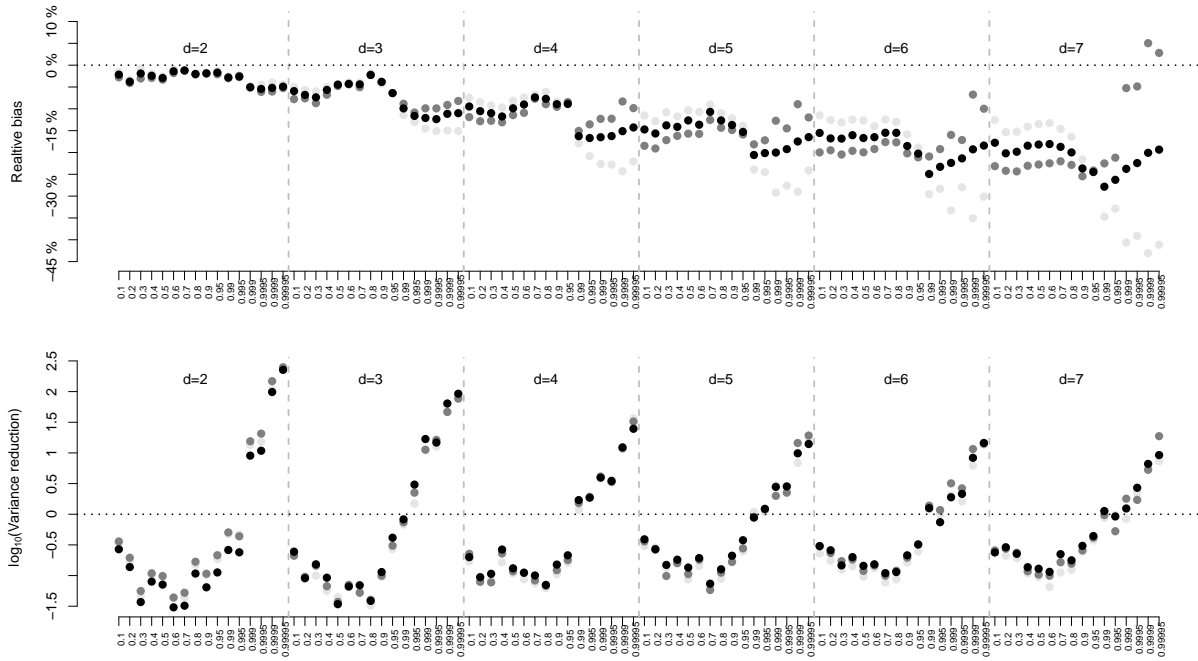


Fig. 7.5 Relative Bias (top) and Variance Reduction (bottom) for the Gumbel(1.25) copula using the SMC algorithm. Using the notation from (7.1), \bullet Marginal for $i = 1$, \bullet Marginal for $i = d$, \bullet Sum of all the marginal conditional expectations (Expected Shortfall).

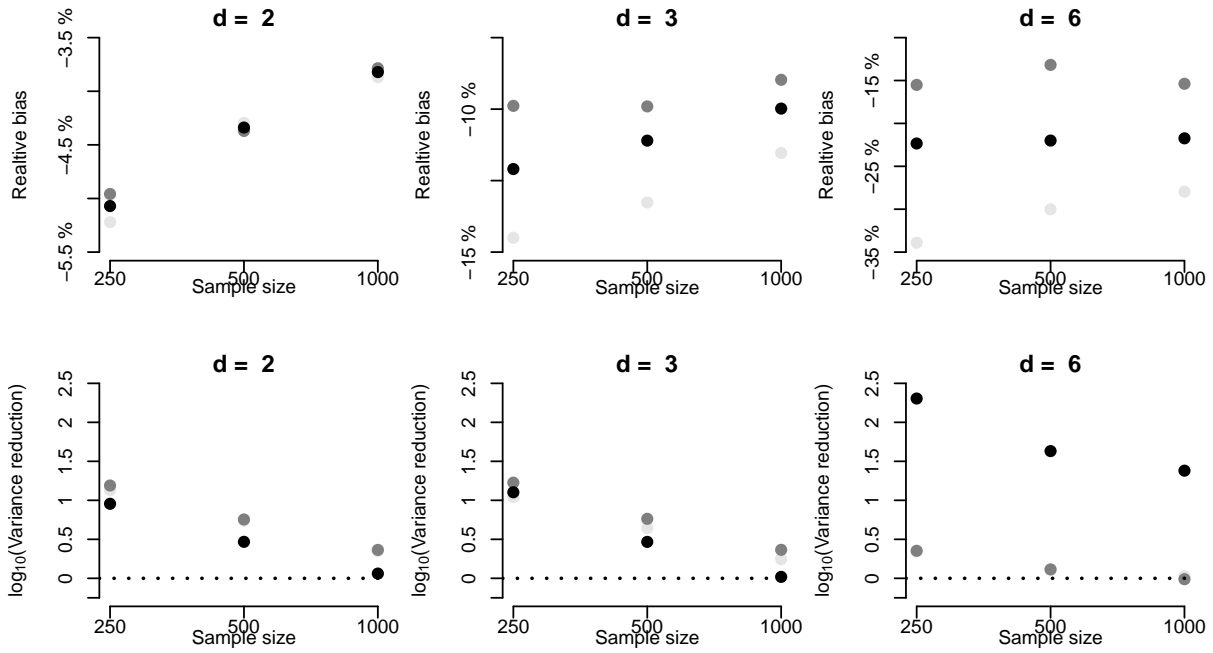


Fig. 7.6 Relative Bias (top) and Variance Reduction (bottom) for the Gumbel(1.25) copula using the SMC algorithm with $N_{SMC} = 250, 500$ and $1,000$ particles. Using the notation from (7.1), $\alpha = 0.999$, \bullet Marginal for $i = 1$, \bullet Marginal for $i = d$, \bullet Sum of all the marginal conditional expectations (Expected Shortfall).

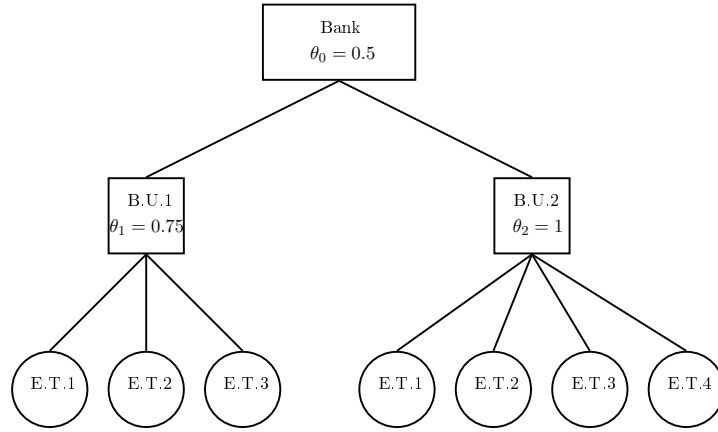


Fig. 7.7 Hierarchical Clayton Copula.

It is important to note that the estimation of expectations of the form (7.1) in the Gumbel model is extremely challenging, specially due to the fact that, differently from the Clayton copula, the Gumbel copula possess an intricate dependence structure near the upper right corner of the unit cube. In this case the exploration of the $[0, 1]^d$ needs to be done in a (even more) careful way, in order to avoid regions with low probability density. In practice it will also be important to consider the design of the mutation kernel in the SMC Sampler algorithm if higher dimensions are considered.

7.3 Hierarchical Clayton copula dependence between Business Units and Event Types

As a final example, we show the utilization of the SMC Sampler method in a hierarchical allocation process (as in Section 2.1). As a toy model, we assume a bank is divided in two different Business Units (B.U.). For the first one it is assumed that Operational Losses are due to three different Event Types (E.T.), while for B.U.2 losses may come from four different E.T.. For the simulation and also density calculation, in this example we made use the R-package HAC [36].

This bank structure can be conveniently modelled with the help of a Hierarchical Archimedean Copula (HAC), also known as Nested Archimedean Copula (see [36] and references therein). For this example we have chosen a Hierarchical Archimedean Copula as in Figure 7.7. The dependence of the three E.T.'s on B.U.1 is given by a Clayton copula with parameter $\theta_1 = 0.75$, while within the 4 E.T.'s in B.U.2 the dependence is modelled through a Clayton copula with parameter $\theta_2 = 1$. Moreover, any loss on B.U.1 is related to losses in B.U.2 through a Clayton copula with parameter $\theta_0 = 0.5$. The copula for this model is given by

$$C(\mathbf{u}) = C_0(C_1(\mathbf{u}_1; \theta_1), C_2(\mathbf{u}_2; \theta_2); \theta_0),$$

where $C(\cdot; \theta)$ denotes a Clayton copula with parameter θ and $\mathbf{u} = (u_1, \dots, u_7)$, $\mathbf{u}_1 = (u_1, u_2, u_3)$, $\mathbf{u}_2 = (u_4, u_5, u_6, u_7)$. The reader should note that $C_0(\cdot; \theta_0)$ is not a copula between aggregated losses. It is also important to stress the fact that this choice of parameters will ensure the Hierarchical Copula is a well defined copula, since all the members are from the same family and the parameters are decreasing from the highest to the lowest level (see, for example, [27]).

As in the non-nested Clayton case, from Figure 7.8 we can see the SMC Sampler procedure is unbiased for $N_{SMC} = 250$, with Relative Bias smaller than 5% in absolute terms. The method also decreases the variance of the estimates when the quantile level in the conditional event is larger than $\alpha = 0.99$, from where we can state its effectiveness.

8 Conclusion and Final Remarks

With focus on the capital allocation problem for copula-dependent risks, we presented a Sequential Monte Carlo (SMC) Sampler algorithm to calculate conditional expectations where the conditioning event is rare. We exploit the copula structure to design a SMC algorithm whose efficiency is analysed through the Variance Reduction (see Equation 7.2) when compared to a simple Monte Carlo scheme.

If, in addition to the Variance Reduction, the computational time is to be computed, one should be extremely careful, as it will be strongly dependent the programming language used, the hardware and

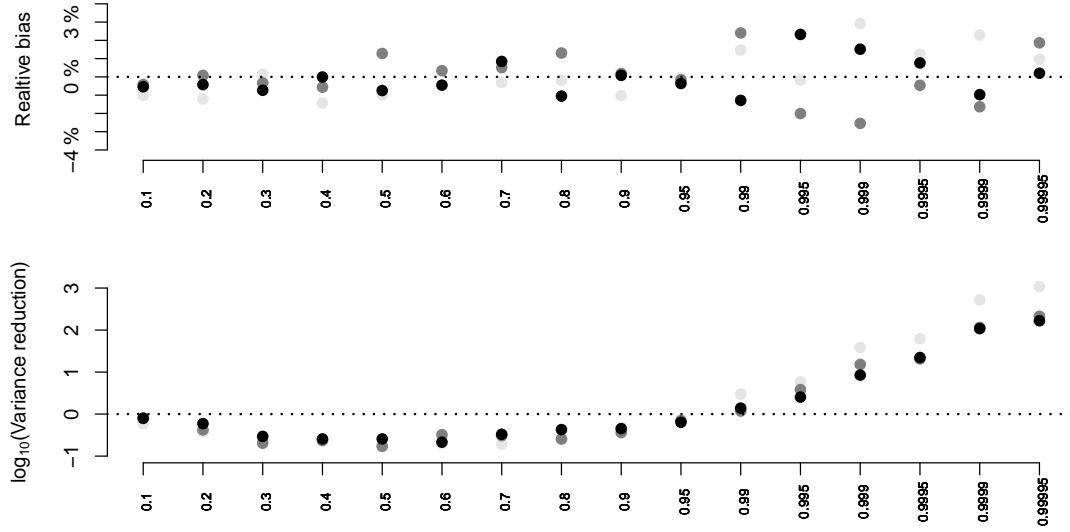


Fig. 7.8 Relative Bias (top) and Variance Reduction (bottom) for the Hierarchical Clayton copula from Figure 7.7 using the SMC algorithm with $N_{SMC} = 250$ particles. Using the notation from (7.1), \circ Marginal for $i = 1$, \bullet Marginal for $i = 7$, \blacklozenge Sum of all the marginal conditional expectations (Expected Shortfall).

also the actual implementation of the algorithms. For example, in R most of the basic functions (including sampling from simple distributions) can be used in a vectorial form, meaning that applying the function to a vector will be much faster than calculating the function values serially, ie, inside a `for` loop. For the sake of simplicity, both in the simple MC and in the SMC schemes we sampled one value at a time and did not make use of any vectorial form.

Due to the nature of the allocation problem, there is no need for the algorithm presented to be run *online*. In most of the cases it will be used once a month or even once a year. Nevertheless, if the performance of the SMC algorithm needs to be improved specialized libraries such as the *LibBi* (see [34]) can be used.

Although the code used for this paper has not been thoroughly optimised, some computational analysis can still be done. For the Clayton example from Section 7.1 (with parameter $\theta = 1$), for each quantile level, Table 8 presents (1) the time (in minutes) necessary to run the SMC algorithm; (2) how many times the SMC algorithm is slower than the simple Monte Carlo; (3) the Variance Reduction factor (as of (7.2)) for the quantities in (7.1), where ES denotes the sum of the expectations (Expected Shortfall).

Quantile	SMC time	$\frac{\text{SMC time}}{\text{MC time}}$	Variance Reduction		
			k=1	k=5	ES
0.1	0.32	53.83	0.34	1.95	0.46
0.2	0.64	68.97	0.27	1.16	0.29
0.3	0.97	73.94	0.11	0.14	0.37
0.4	1.30	75.04	0.07	1.89	0.19
0.5	1.63	73.09	0.19	0.16	0.18
0.6	1.97	68.15	0.35	0.03	0.28
0.7	2.31	61.41	0.42	0.21	0.78
0.8	2.65	52.55	0.30	0.38	0.21
0.9	3.00	39.28	0.44	2.94	1.83
0.95	3.35	26.34	0.29	0.33	4.10
0.99	3.70	9.64	2.79	5.36	3.54
0.995	4.04	4.56	2.98	6.54	21.87
0.999	4.39	1.28	16.56	7.20	71.37
0.9995	4.74	0.56	94.74	88.12	172.51
0.9999	5.08	0.15	224.82	154.54	272.74
0.99995	5.42	0.06	1891.81	205.48	593.32

Table 8.1 Computational time (in minutes) and Variance Reduction for the SMC algorithm when compared to a simple Monte Carlo scheme.

On this particular implementation of the algorithm, for low quantiles we can see that the MC scheme is considerably faster for the same accuracy of the SMC method. For example, when $\alpha = 0.4$ the MC method is 75 times faster than the SMC and the Variance Reduction on the first marginal expectation is 0.07, meaning that the MC variance is around 14 times smaller than the SMC variance.

On the other hand, for higher quantiles, in particular after $\alpha = 0.99$, the SMC method starts to become more appealing, since the Variance Reduction gets larger than the difference in time. It's worth noticing the proposed SMC method has been designed to be used on extreme quantiles where lower quantiles are only used as intermediate steps.

The computing times presented on Table 8 were measured using R 3.1.0 in a Intel Xeon E5-1650, 3.20GHz and 16GB RAM. For each quantile level the algorithms were run 10 times and the values presented are an average over these 10 runs.

The R code used is available upon request from the authors.

Acknowledgements The authors would like to thank Prof. Mario Wüthrich (ETH, Switzerland) and Prof. Pierre Del Moral (UNSW, Australia) for fruitful discussions related to this work. The authors also acknowledge the use of the UCL Legion High Performance Computing Facility (Legion@UCL), and associated support services. RST has been funded by the CNPq (Brazil) through a Ciência sem Fronteiras scholarship and part of this project has been developed while he was an Endeavour Research Fellow at CSIRO, Australia. GWP acknowledges the support of the Institute of Statistical Mathematics, Japan; the Oxford-Man Institute, UK and the Systemic Risk Center, London School of Economics, UK.

A Euler's homogeneous function theorem

Definition A.1 (Homogeneous function) A function $f : U \subset \mathbb{R}^d \rightarrow \mathbb{R}$ is said to be homogeneous of degree τ if, for all $h > 0$ and $u \in U$ with $hu \in U$ the following equation holds:

$$f(hu) = h^\tau f(u).$$

Theorem A.2 (Euler's homogeneous function theorem) Let $U \subset \mathbb{R}^n$ be an open set and $f : U \rightarrow \mathbb{R}$ be a continuously differentiable function. Then f is homogeneous of degree τ if, and only if, it satisfies the following equation:

$$\tau f(\mathbf{u}) = \sum_{i=1}^n u_i \frac{\partial f}{\partial u_i}, \quad \mathbf{u} = (u_1, \dots, u_n) \in U, h > 0$$

B Copulas and Sklar's theorem

Definition B.1 (Copula) A d-dimensional copula is a distribution function on $[0, 1]^d$ with uniform marginal distributions.

Theorem B.2 (Sklar) Let $F_{\mathbf{X}}$ be a joint distributions with margins F_1, \dots, F_d . Then there exists a copula $C : [0, 1]^d \rightarrow [0, 1]$ such that

$$F_{\mathbf{X}}(\mathbf{x}) = C(F_1(x_1), \dots, F_d(x_d)), \quad \forall \mathbf{x} = (x_1, \dots, x_d) \in \overline{\mathbb{R}}^d. \quad (\text{B.1})$$

If the margins are continuous then C is unique, given by

$$C(u_1, \dots, u_d) = F_{\mathbf{X}}(F_1^{-1}(u_1), \dots, F_d^{-1}(u_d))$$

Conversely, if C is a copula and F_1, \dots, F_d are univariate distributions, then the F defined in (B.1) is a joint distribution function with margins F_1, \dots, F_d .

Moreover, if we assume that F_1, \dots, F_d are differentiable, then the joint density function of \mathbf{X} can be written as

$$f_{\mathbf{X}}(\mathbf{x}) = c(F_1(x_1), \dots, F_d(x_d)) \prod_{i=1}^d f_i(x_i),$$

where

$$c(u_1, \dots, u_d) = \frac{\partial^d C(u_1, \dots, u_d)}{\partial u_1 \dots \partial u_d}$$

and f_i is the density of X_i .

Definition B.3 (Archimedean generator) An Archimedean generator is a continuous, decreasing function $\psi : [0, \infty] \rightarrow [0, 1]$ that satisfies $\psi(0) = 1$, $\lim_{t \rightarrow \infty} \psi(t) = 0$ and is strictly decreasing on $[0, \inf\{t : \psi(t) = 0\}]$.

Definition B.4 (Archimedean copulas) A d -dimensional copula is called Archimedean if it is of

$$C(\mathbf{u}; \psi) = \psi(\psi^{-1}(u_1) + \psi^{-1}(u_d)), \quad \mathbf{u} = (u_1, \dots, u_d) \in [0, 1]^d, \quad (\text{B.2})$$

where ψ is the Archimedean generator.

Family	Parameter	Generator $\psi(t)$
Clayton	$\theta \in (0, \infty)$	$(1+t)^{-1/\theta}$
Gumbel	$\theta \in [1, \infty)$	$\exp -t^{1/\theta}$

Table B.1 Commonly used Archimedean generators

Definition B.5 (Multivariate coefficients of tail dependence) For the copulas defined in Table B.1 the multivariate coefficients of upper and lower tail dependence are defined, respectively, as

$$\begin{aligned} \lambda_u &= \lim_{u \rightarrow 1^-} \mathbb{P}[U_1 > u \mid U_2 > u, \dots, U_d > u] \\ &= \lim_{t \rightarrow 0^+} \frac{\sum_{i=1}^d \binom{n}{n-i} i (-1)^i \psi'(it)}{\sum_{i=1}^{n-1} \binom{n-1}{n-1-i} i (-1)^i \psi'(it)} \end{aligned}$$

$$\begin{aligned} \lambda_l &= \lim_{u \rightarrow 0^+} \mathbb{P}[U_1 \leq u \mid U_2 \leq u, \dots, U_d \leq u] \\ &= \frac{d}{d-1} \lim_{t \rightarrow \infty} \frac{\psi'(dt)}{\psi'((d-1)t)} \end{aligned}$$

References

- Arbenz, P., Cambou, M., Hofert, M.: An Importance Sampling Algorithm for Copula Models in Insurance. arXiv preprint arXiv:1403.4291 (2014)
- Artzner, P., Delbaen, F., Eber, J.M., Heath, D.: Coherent measures of risk. *Mathematical finance* **9**(3), 203–228 (1999)
- Asimit, A.V., Vernic, R., Zitikis, R.: Evaluating Risk Measures and Capital Allocations Based on Multi-Losses Driven by a Heavy-Tailed Background Risk: The Multivariate Pareto-II Model. *Risks* **1**(1), 14–33 (2013)
- Asmussen, S., Glynn, P.W.: *Stochastic Simulation: Algorithms and Analysis: Algorithms and Analysis*, vol. 57. Springer (2007)
- BCBS: International Convergence of Capital Measurement and Capital Standards - A Revised Framework (Comprehensive Version). Tech. rep., Bank for International Settlements (2006). URL <http://www.bis.org/publ/bcbs128.pdf>
- Böcker, K., Klüppelberg, C.: Modelling and measuring multivariate operational risk with Lévy copulas. *The Journal of Operational Risk* **3**(2), 3–27 (2008)
- Brechmann, E., Czado, C., Paterlini, S.: Flexible dependence modeling of operational risk losses and its impact on total capital requirements. *Journal of Banking & Finance* **40**, 271–285 (2014)
- Buch, A., Dorfleitner, G.: Coherent risk measures, coherent capital allocations and the gradient allocation principle. *Insurance: Mathematics and Economics* **42**(1), 235–242 (2008)
- Cai, J., Einmahl, J.H., De Haan, L., Zhou, C.: Estimation of the marginal expected shortfall: the mean when a related variable is extreme. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* (forthcoming)
- Carmona, R., Fouque, J.P., Vestal, D.: Interacting particle systems for the computation of rare credit portfolio losses. *Finance and Stochastics* **13**(4), 613–633 (2009)
- Chopin, N.: Central limit theorem for sequential Monte Carlo methods and its application to Bayesian inference. *The Annals of Statistics* **32**(6), 2385–2411 (2004)
- Creal, D.: A survey of sequential Monte Carlo methods for economics and finance. *Econometric Reviews* **31**(3), 245–296 (2012)
- Crisan, D., Doucet, A.: A survey of convergence results on particle filtering methods for practitioners. *Signal Processing, IEEE Transactions on* **50**(3), 736–746 (2002)
- Cruz, M.G., Peters, G.W., Shevchenko, P.V.: *Fundamental Aspects of Operational Risk Modeling and Insurance Analytics: A Handbook of Operational Risk*. John Wiley & Sons (2014)
- De Luca, G., Rievecchio, G.: Multivariate tail dependence coefficients for archimedean copulae. In: *Advanced Statistical Methods for the Analysis of Large Data-Sets*, pp. 287–296. Springer (2012)
- Del Moral, P.: *Feynman-Kac Formulae*. Springer (2004)
- Del Moral, P., Doucet, A., Jasra, A.: Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **68**(3), 411–436 (2006)

18. Del Moral, P., Peters, G.W., Vergé, C.: An Introduction to Stochastic Particle Integration Methods: With Applications to Risk and Insurance. In: Monte Carlo and Quasi-Monte Carlo Methods 2012, pp. 39–81. Springer (2013)
19. Denault, M.: Coherent allocation of risk capital. *Journal of risk* **4**(1), 1–34 (2001)
20. Doucet, A., De Freitas, N., Gordon, N.: Sequential Monte Carlo methods in practice. Springer (2001)
21. Doucet, A., Johansen, A.M.: A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of Nonlinear Filtering* **12**, 656–704 (2009)
22. Gamerman, D., Lopes, H.F.: Markov chain Monte Carlo: stochastic simulation for Bayesian inference. CRC Press (2006)
23. Giacometti, R., Rachev, S., Chernobai, A., Bertocchi, M.: Aggregation issues in operational risk. *The Journal of Operational Risk* **3**(3), 3–23 (2008)
24. Gilks, W.R., Berzuini, C.: Following a moving target Monte Carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **63**(1), 127–146 (2001)
25. Glasserman, P.: Measuring marginal risk contributions in credit portfolios. *Journal of Computational Finance* **9**(2), 1 (2005)
26. Gouriéroux, C., Laurent, J.P., Scaillet, O.: Sensitivity analysis of values at risk. *Journal of Empirical Finance* **7**(3), 225–245 (2000)
27. Hofert, J.M.: Sampling Nested Archimedean Copulas: With Applications to CDO Pricing. Suedwestdeutscher Verlag fuer Hochschulschriften (2010)
28. Hofert, M., Kojadinovic, I., Maechler, M., Yan, J.: copula: Multivariate Dependence with Copulas (2014). URL <http://CRAN.R-project.org/package=copula>. R package version 0.999-9.
29. Kalkbrener, M.: An axiomatic approach to capital allocation. *Mathematical Finance* **15**(3), 425–437 (2005)
30. Künsch, H.R.: Recursive Monte Carlo filters: algorithms and theoretical analysis. *Annals of Statistics* **33**(5), 1983–2021 (2005)
31. Liu, J.S., Chen, R.: Sequential Monte Carlo methods for dynamic systems. *Journal of the American statistical association* **93**(443), 1032–1044 (1998)
32. McLeish, D.L.: Bounded relative error importance sampling and rare event simulation. *Astin Bulletin* **40**(1), 377–398 (2010)
33. McNeil, A.J., Frey, R., Embrechts, P.: Quantitative Risk Management: Concepts, Techniques, and Tools. Princeton university press (2010)
34. Murray, L.M.: Bayesian state-space modelling on high-performance hardware using libbi. arXiv preprint arXiv:1306.3277 (2013)
35. Neal, R.M.: Slice sampling. *Annals of statistics* **31**(3), 705–741 (2003)
36. Okhrin, O., Ristig, A.: Hierarchical archimedean copulae: The hac package. *Journal of Statistical Software* **58**(4), 1–20 (2014). URL <http://www.jstatsoft.org/v58/i04>
37. Peters, G.W.: Topics in sequential Monte Carlo samplers. M. Sc., University of Cambridge, Department of Engineering (2005)
38. Peters, G.W., Shevchenko, P.V.: Advances in Heavy Tailed Risk Modeling: A Handbook of Operational Risk. John Wiley & Sons (2014)
39. Peters, G.W., Shevchenko, P.V., Wüthrich, M.V.: Dynamic operational risk: modeling dependence and combining different sources of information. *The Journal of Operational Risk* **4**(2), 69–104 (2009)
40. Ristic, B., Arulampalam, S., Gordon, N.: Beyond the Kalman filter: Particle filters for tracking applications. Artech house (2004)
41. Rosen, D., Saunders, D.: Risk factor contributions in portfolio credit risk models. *Journal of Banking & Finance* **34**(2), 336–349 (2010)
42. Shevchenko, P.V.: Modelling Operational Risk Using Bayesian Inference. Springer (2011)
43. Siller, T.: Measuring marginal risk contributions in credit portfolios. *Quantitative Finance* **13**(12), 1915–1923 (2013)
44. Tasche, D.: Risk contributions and performance measurement. Report of the Lehrstuhl für mathematische Statistik, TU München (1999)
45. Tasche, D.: Capital allocation to business units and sub-portfolios: the Euler principle. In: Pillar II in the New Basel Accord: The Challenge of Economic Capital, pp. 423–453. Risk Books (2008)