A Novel Methodology for Credit Portfolio Analysis : Numerical Approximation Approach *

Yasushi Takano † Jiro Hashiba ‡

April 24, 2008

Mizuho-DL Financial Technology,
1-3, Otemachi 1-chome, Chiyoda-ku, Tokyo 100-0004, Japan

Abstract

This paper proposes a novel numerical methodology for quantifying credit portfolio risk, based on the multi-factor Merton model. This methodology consists of two steps. The first step is a numerical algorithm for computing moment generating function very quickly. The second one is a fast Laplace inversion algorithm first introduced by de Hoog et al. The moment generating function computed in the first step is transformed into a loss distribution function through the second step. It is demonstrated that the risk measures such as VaR and CVaR obtained by this methodology are sufficiently accurate, for a wide range of portfolios. Furthermore, computation time depends on portfolio size quite moderately in this methodology. In fact, by using an ordinary personal computer, we can compute the credit risk of a portfolio with 1 million obligors only in a few minutes. We also present a fast algorithm for computing the risk contributions of obligors to VaR and CVaR.

JEL classification : C63, G21.

Keywords : multi-factor model, credit portfolio risk, credit concentration, risk contribution, Laplace inversion.

*The views expressed in this paper are those of the authors, and do not necessarily reflect those of their institution.

†yasushi-takano@fintec.co.jp

‡jiro-hashiba@fintec.co.jp

1 Introduction

It is important for a bank to manage the risks originating from its business activities. In particular, the credit risk underlying its credit portfolio is often the largest risk in a bank. Therefore, credit risk management is an important function in a bank to maintain its financial soundness. In order to quantitatively assess to what extent a bank is exposed to the risk of bankruptcy, it is necessary to measure the credit risk at portfolio level. The measured credit risk is then used to assign risk capital, which a bank should hold to absorb potential losses arising from its credit portfolio.

One of the major origins of credit risk at portfolio level is concentration risk, which can be grouped into two categories : *name concentration* and *sector concentration* [1]. Recently, quantification of these concentration risks has been a central issue both for supervisors and bank practitioners. This is partly because the asymptotic single risk factor (ASRF) model adopted in Basel II is based on the assumption that a bank's portfolio is completely diversified and the systematic factor is univariate [2]. Name concentration and sector concentration to a specific industry or region can not be taken into account under this assumption. In order not to underestimate the credit risk at portfolio level, these concentration risks must be appropriately evaluated.

Monte Carlo simulation is a standard method for measuring credit portfolio risk. However, this method is time-consuming in general. This is problematic if a bank attempts to rebalance its credit portfolio actively and frequently. To rebalance a portfolio, we usually need information about how individual transactions such as asset sales or credit derivatives will affect the total credit risk. In principle, this can be done by directly comparing the total risks before and after executing the transactions, or appropriately allocating the total risk to each transaction. To frequently obtain these information for many transactions, computation must be done very quickly.

In summary, credit risk managers in a bank are always faced with the following challenges.

- How can concentration risks be quantified?
- How can risk measures be computed in a short time?
- How can the contribution of individual transactions to total risk be computed?

Although it is difficult to satisfy all these requirements at the same time, many theoretical advances have been made in recent years by academics, practitioners, and supervisory authorities. Some of these methodologies have proposed analytic expressions for portfolio risk information such as loss distribution function or VaR. As compared to familiar Monte Carlo simulation, these analytic methods have an advantage that risk measures can be computed very quickly.

[1] For a comprehensive survey on these concentration risks, see BCBS (2006).
[2] See BCBS (2005).

For example, the granularity adjustment due to Gordy (2003) takes into account name concentration effects, by considering the second order correction to the ASRF model. To include sector concentration effects in addition, Pykhtin (2004) has generalized this method to multi-factor cases. In this generalization, multivariate systematic factors are replaced with a linear combination of them, yielding a model that incorporates only one factor effectively. Martin et al. (2001) has applied the saddle-point method, which is familiar in physics, to portfolio risk quantification. In this method, an approximate expression of the Laplace inversion of the moment generating function (MGF) is obtained. This plain saddle-point method has been further improved by Martin and Ordovás (2006), based on conditional independence models. The main feature of this improvement is that the saddle-point approximation is applied to the *conditional* MGF, not to the *unconditional* MGF. This modification requires additional computation time, but the approximation becomes much better. Analytic calculation of the risk contributions of obligors has been studied in Emmer and Tasche (2003) for single-factor cases, and in Tasche (2006) for multi-factor cases.

Not only analytical approaches as above, but improvements of Monte Carlo simulation have also been considered. In Higo (2006), a credit portfolio is segmented into two groups, namely a sub-portfolio with large exposures and the other one with the remaining smaller exposures. Then, the idiosyncratic risks of the smaller exposures are neglected. Computation time is substantially reduced in this method, because random numbers for idiosyncratic risks of many smaller exposures need not be generated. Glasserman (2005) has computed risk contributions obtained as the probability of default conditional on portfolio losses close to VaR. In this study, importance sampling is found effective to generate loss scenarios close to VaR.

This paper proposes a numerical algorithm for quantifying credit portfolio risk, based on the multi-factor Merton model. This algorithm consists of two steps. In the first step, the MGF of loss distribution is computed. As reported in BCBS (2006), the number of systematic factors employed in world's major banks varies from 7 to 110. This implies that Monte Carlo integration is necessary to obtain the MGF, in conditional independence models. In a simple implementation of Monte Carlo integration, the computation time of the MGF is proportional to $N_I \times N$, where N_I is the number of sample integration points, and N is the number of obligors. The first step uses a trick to make this dependence much milder, so that the MGF can be computed in a much shorter time. In the second step, we use the numerical Laplace inversion algorithm proposed by de Hoog et al. (1982). This step is further decomposed into two algorithms. The first algorithm approximates the Bromwich inversion integral by an infinite series with the trapezoidal rule. However, this algorithm alone does not work, since the convergence of the resulting infinite series is extremely slow. Thus, in the second algorithm, the infinite series is converted into a more rapidly converging sequence, i.e., continued fraction expansion. This idea is an application of the so-called Padé approximation, which approximates any function by a rational function. We find that the convergence speed is significantly improved by the second algorithm.

As mentioned above, importance sampling has been extensively studied recently to

compute the risk contributions of obligors. However, this method is time-consuming especially for large portfolios, since they rely on Monte Carlo simulation. In this paper, we present an efficient numerical algorithm for computing the risk contributions to VaR and CVaR. A slight modification of the above two steps can be straightforwardly applied to this computation. By this algorithm, it becomes possible to quantitatively evaluate name concentration to obligors with large exposures. Furthermore, the risk contributions of *all* obligors can be computed in a short time, even for a portfolio with millions of obligors.

In this paper, a series of algorithms explained in the last two paragraphs is collectively referred to as *the fast numerical approximation*. The fast numerical approximation resolves some drawbacks of analytical methods and Monte Carlo simulation. In fact, some of the existing analytical methods are based on strong assumptions about credit portfolio (e.g. mild concentration or strong default correlation), to obtain analytically closed expressions. On the other hand, this paper does not seek analytically closed forms, but derives final results through a series of simple algorithms. As a result, the fast numerical approximation is applicable to a broader range of portfolios. For instance, the accuracy of risk measures is affected very weakly by the granularity of portfolio, the probability of default, and the strength of default correlation. Furthermore, all simple algorithms constituting the fast numerical approximation are highly optimized, so that computation time remains sufficiently shorter than that of Monte Carlo simulation.

This paper is organized as follows. In section 2, we review Laplace transformation and the Merton model, and derive some mathematical expressions for portfolio risk measures. In section 3, we review de Hoog algorithm used as the second step of the fast numerical approximation. In section 4, we propose a novel numerical algorithm for computing MGF, which is the first step of the fast numerical approximation. We show numerical examples in section 5 for some sample portfolios, and show that the fast numerical approximation works quite well for a wide range of portfolios. The last section is devoted to conclusions.

2 Credit Risk and Laplace Transformation

2.1 Laplace Transformation and Inversion

We begin with the definition of Laplace transformation. Suppose that $f(x)$ is a real-valued function defined along the positive real axis. Then, the Laplace transform $\hat{f}(\lambda)$ of $f(x)$ is defined by

$$\hat{f}(\lambda) = \int_0^\infty e^{-\lambda x} f(x) dx, \tag{1}$$

where λ is a complex-valued auxiliary variable. This relation can be inverted by the Bromwich integral as [3]

$$f(x) = \frac{1}{2\pi i} \int_{\gamma - i\infty}^{\gamma + i\infty} e^{\lambda x} \hat{f}(\lambda) d\lambda, \tag{2}$$

where the integration path specified by a real number γ must be on the right side of all singularities of $\hat{f}(\lambda)$.

If $f(x)$ is the pdf of a random variable $\tilde{X}$, its Laplace transform is identical with the moment generating function (MGF) of the distribution:

$$\begin{aligned} \hat{f}(\lambda) &= \int_0^\infty e^{-\lambda x} f(x) dx, \\ &= \mathbf{E}\left[\exp\left(-\lambda \tilde{X}\right)\right]. \end{aligned} \tag{3}$$

Laplace inversion is quite useful, since in some cases it is much easier to compute MGF than to compute pdf directly. The Merton model explained in the next subsection is an example of such cases.

2.2 Multi-factor Merton Model

The Merton model is a well-established framework for quantifying credit portfolio risk with default correlation among obligors. In this subsection, we review this model for multi-factor cases.

Let $\tilde{L}$ denote the portfolio loss given by

$$\tilde{L} = \sum_{i=1}^{N} E_i \tilde{D}_i, \tag{4}$$

where the symbols on the r.h.s. are defined as follows:

- E_i : The exposure of obligor i [4].
- $\tilde{D}_i$: The default indicator of obligor i taking the following values:

$$\tilde{D}_i = \begin{cases} 0 & \text{if obligor } i \text{ is not in default,} \\ 1 & \text{if obligor } i \text{ is in default.} \end{cases} \tag{5}$$

- N : The number of obligors.

[3] In this paper, i represents $\sqrt{-1}$.

[4] We assume $\sum_{i=1}^{N} E_i = 1$ here without loss of generality. LGD is implicitly assumed to be non-stochastic, and set to be 100%.

To describe the obligors' default and its correlation structure, we assign each obligor a random variable called *firm-value* hereafter. The firm-value $\tilde{Z}_i$ of obligor i is represented by

$$\tilde{Z}_i = \boldsymbol{\alpha}_i \cdot \tilde{\mathbf{X}} + \sqrt{1 - |\boldsymbol{\alpha}_i|^2}\tilde{\varepsilon}_i, \tag{6}$$

where the symbols on the r.h.s. are defined as follows:

- $\tilde{\mathbf{X}} = (\tilde{X}_1, \tilde{X}_2, \ldots, \tilde{X}_{N_f})$: The systematic risk factors, each of which is a standard normal random variable.
- N_f : The number of systematic risk factors.
- $\boldsymbol{\alpha}_i = (\alpha_{i,1}, \alpha_{i,2}, \ldots, \alpha_{i,N_f})$: The *loading vector* of obligor i with $0 < |\boldsymbol{\alpha}_i| < 1$.
- $\tilde{\varepsilon}_i \sim N(0,1)$: The idiosyncratic risk factor of obligor i.

In addition, we assume that the random variables $\tilde{X}_j$ and $\tilde{\varepsilon}_i$ are all independent. Firm-values (6) then become also standard normal, and dependent of each other through the systematic factors. In fact, the correlation between obligors i and j takes a non-vanishing value as

$$\mathrm{corr}\left(\tilde{Z}_i, \tilde{Z}_j\right) = \boldsymbol{\alpha}_i \cdot \boldsymbol{\alpha}_j. \tag{7}$$

Let us explain in more detail the meaning of the systematic and idiosyncratic risk factors in (6). The systematic factors $\tilde{X}_j$ can be viewed as variables that represent macro-economic conditions and affect the creditworthiness of all obligors simultaneously. As will be discussed later in section 4.2, the obligors in the same industrial or regional sector are usually assigned a common loading vector $\boldsymbol{\alpha}_i$. This vector characterizes how the industry or region is influenced by the macro-economic conditions. On the other hand, the idiosyncratic factors $\tilde{\varepsilon}_i$ represent conditions unique to each obligor i. This is why they are assumed to be independent of each other.

In banks' standard risk management practice, each obligor is assigned a credit rating and probability of default (PD) is estimated for each rating from empirical default data. Let PD_r be the PD estimate for rating r, and $r(i)$ be obligor i's rating. In the Merton model, obligor i defaults if its firm-value falls below the threshold level C_i defined by

$$C_i \equiv \Phi^{-1}(\mathrm{PD}_{r(i)}), \tag{8}$$

where $\Phi(x)$ is the standard normal cumulative distribution function, and $\Phi^{-1}(x)$ denotes its inverse function. It is straightforward to verify that the probability of $\tilde{Z}_i$ being lower than C_i is $\mathrm{PD}_{r(i)}$. The probability of obligor i's default conditional on $\tilde{\mathbf{X}} = \mathbf{x}$ is given by

$$\begin{aligned} p_i(\mathbf{x}) &\equiv \Pr\left(\tilde{Z}_i < C_i \,\middle|\, \tilde{\mathbf{X}} = \mathbf{x}\right), \\ &= \Phi\left(\frac{C_i - \boldsymbol{\alpha}_i \cdot \mathbf{x}}{\sqrt{1 - |\boldsymbol{\alpha}_i|^2}}\right). \end{aligned} \tag{9}$$

Thus, the conditional PD depends on the systematic factors, reflecting the fact that macro-economic conditions affect the possibility of an obligor's default.

2.3 MGF and pdf in Merton Model

We showed in the last subsection that default correlation can be incorporated by the systematic factors. Another advantage of the systematic factors is that they enable us to obtain a nearly analytic expression of MGF. If the systematic factors are fixed, default occurs independently because the only remaining uncertainty is the idiosyncratic risk. In the multi-factor Merton model, the MGF conditional on $\tilde{\mathbf{X}}$ is thus given by the product of each obligor's MGF as

$$\begin{aligned} M_L(\lambda;\tilde{\mathbf{X}}) &\equiv \mathbf{E}\left[\exp\left(-\lambda\tilde{L}\right) \mid \tilde{\mathbf{X}}\right], \\ &= \prod_{i=1}^{N} \mathbf{E}\left[e^{-\lambda E_i \tilde{D}_i} \mid \tilde{\mathbf{X}}\right], \\ &= \prod_{i=1}^{N} \left\{1 - p_i(\tilde{\mathbf{X}}) + p_i(\tilde{\mathbf{X}})e^{-\lambda E_i}\right\}. \end{aligned} \tag{10}$$

Taking the expectation value of this conditional MGF yields the unconditional MGF

$$\begin{aligned} M_L(\lambda) &\equiv \mathbf{E}\left[\exp\left(-\lambda\tilde{L}\right)\right], \\ &= \mathbf{E}\left[\mathbf{E}\left[\exp\left(-\lambda\tilde{L}\right) \mid \tilde{\mathbf{X}}\right]\right], \\ &= \mathbf{E}\left[M_L(\lambda;\tilde{\mathbf{X}})\right], \\ &= \mathbf{E}\left[\prod_{i=1}^{N} \left\{1 - p_i(\tilde{\mathbf{X}}) + p_i(\tilde{\mathbf{X}})e^{-\lambda E_i}\right\}\right]. \end{aligned} \tag{11}$$

As explained before, pdf can be obtained from the corresponding MGF through Laplace inversion. Hence the pdf $f_L(l)$ for portfolio losses is represented by

$$f_L(l) = \frac{1}{2\pi i}\int_{\gamma-i\infty}^{\gamma+i\infty} e^{\lambda l} M_L(\lambda) d\lambda, \tag{12}$$

$$= \frac{1}{2\pi i}\int_{\gamma-i\infty}^{\gamma+i\infty} e^{\lambda l}\mathbf{E}\left[\prod_{i=1}^{N} \left\{1 - p_i(\tilde{\mathbf{X}}) + p_i(\tilde{\mathbf{X}})e^{-\lambda E_i}\right\}\right] d\lambda. \tag{13}$$

However, for the purpose of computing quantiles, it is more straightforward to compute cdf directly instead of pdf. Using the Laplace transformation formula for the integral of a function, we obtain the cdf as

$$F_L(l) = \frac{1}{2\pi i}\int_{\gamma-i\infty}^{\gamma+i\infty} \frac{e^{\lambda l}}{\lambda} M_L(\lambda) d\lambda, \tag{14}$$

$$= \frac{1}{2\pi i}\int_{\gamma-i\infty}^{\gamma+i\infty} \frac{e^{\lambda l}}{\lambda}\mathbf{E}\left[\prod_{i=1}^{N} \left\{1 - p_i(\tilde{\mathbf{X}}) + p_i(\tilde{\mathbf{X}})e^{-\lambda E_i}\right\}\right] d\lambda. \tag{15}$$

Finally, let us mention mathematically subtle issues in (13) and (15). Since the loss can take only a finite number ($= 2^N$ at most) of discrete values in the current model, the pdf $f_L(l)$ does not exist in the strict sense [5] . Or equivalently, the cdf $F_L(l)$ is a discontinuous function. However, as will be explained later, we approximate the Bromwich integral in (14) by a trigonometric series (see (38) below), and then truncate it into a finite sum. The cdf $F_L(l)$ becomes a continuous and differentiable function through this approximation.

The remaining issue is whether $F_L(l)$ is monotonically increasing or not. If not, $f_L(l)$ becomes negative and can not be regarded as pdf. In general, a sum of trigonometric functions can not be monotonic. However, since its convergence is very slow, the trigonometric series is further converted to a sequence of functions that converges sufficiently fast to the true cdf. It will be shown later that, for a wide range of portfolios, $F_L(l)$ obtained this way becomes monotonically increasing if relevant parameters are appropriately chosen.

Given the above rationale, we assume from now on that $F_L(l)$ is a continuous, differentiable and monotonically increasing function. This assumption makes the arguments in the next subsection quite simple.

2.4 Risk Measures and Risk Contributions

Consider two familiar portfolio risk measures, i.e., VaR and CVaR defined by

$$\begin{aligned} \mathrm{VaR}(\alpha) &\equiv \inf\{l|\, \Pr(\tilde{L} \geq l) \leq 1-\alpha\}, && (16) \\ \mathrm{CVaR}(\alpha) &\equiv \mathbf{E}\left[\tilde{L} \;\middle|\; \tilde{L} \geq \mathrm{VaR}(\alpha)\right], && (17) \end{aligned}$$

where α is the confidence level. This definition of VaR is equivalent to a more intuitive one implicitly given by

$$F_L(\mathrm{VaR}(\alpha)) = \alpha, \tag{18}$$

under the assumption that $F_L(l)$ is continuous and monotonically increasing. Using the same assumption and the existence of $f_L(l)$, we adopt an alternative definition of CVaR given by

$$\mathrm{CVaR}(\alpha) = \frac{1}{1-\alpha} \int_{\mathrm{VaR}(\alpha)}^{\infty} l f_L(l) dl \tag{19}$$

from now on, instead of (17). These risk measures are computed from the cdf given by (15), and represent the total risk of a credit portfolio.

Next, we consider how to decompose these total risks into individual transactions. In this paper, we study the allocation principle simply given by the partial derivative of the risk measures with respect to the exposure of an obligor [6]. Let us start with VaR, and

[5] It is possible to express $f_L(l)$ by a sum of delta functions, but they are of course not ordinary functions.

[6] See Tasche (2000) for the properties of this allocation principle concerning the performance measurement of a credit portfolio.

define the *risk contribution* of obligor i by

$$\mathrm{RC}_i \equiv E_i \cdot \frac{\partial \mathrm{VaR}(\alpha)}{\partial E_i}. \tag{20}$$

This definition of risk contribution naturally satisfies the additivity

$$\sum_{i=1}^{N} \mathrm{RC}_i = \mathrm{VaR}(\alpha), \tag{21}$$

which follows from Euler's homogeneous function theorem. In the remainder of this subsection, we use the notation

$$F_L(E_i, l) = F_L(l),$$

to clarify the dependence of the cdf on exposure E_i. By differentiating

$$F_L(E_i, \mathrm{VaR}(\alpha)) = \alpha$$

with respect to E_i, we obtain

$$\mathrm{RC}_i = -E_i \cdot \frac{\dfrac{\partial F_L(E_i, \mathrm{VaR}(\alpha))}{\partial E_i}}{\left.\dfrac{\partial F_L(E_i, l)}{\partial l}\right|_{l=\mathrm{VaR}(\alpha)}}. \tag{22}$$

From (15), the numerator in (22) reads

$$\frac{\partial F_L(E_i, \mathrm{VaR}(\alpha))}{\partial E_i} = \frac{1}{2\pi i} \int_{\gamma - i\infty}^{\gamma + i\infty} \frac{e^{\lambda \mathrm{VaR}(\alpha)}}{\lambda} \frac{\partial M_L(\lambda)}{\partial E_i} d\lambda, \tag{23}$$

$$\frac{\partial M_L(\lambda)}{\partial E_i} = -\lambda \mathbf{E}\left[p_i(\lambda; \tilde{\mathbf{X}}) M_L(\lambda; \tilde{\mathbf{X}}) \right], \tag{24}$$

where we have defined *twisted conditional PD* by

$$p_i(\lambda; \mathbf{x}) \equiv \frac{p_i(\mathbf{x}) \exp(-\lambda E_i)}{1 - p_i(\mathbf{x}) + p_i(\mathbf{x}) \exp(-\lambda E_i)}. \tag{25}$$

The denominator in (22) is equal to the pdf $f_L(\mathrm{VaR}(\alpha))$, which can be computed using the additivity condition (21).

Let us turn to the risk contributions to CVaR. The obligor i's risk contribution to CVaR is defined by

$$\mathrm{RC}_i^{\mathrm{CVaR}(\alpha)} \equiv E_i \cdot \frac{\partial \mathrm{CVaR}(\alpha)}{\partial E_i}. \tag{26}$$

Similarly to the VaR case, this definition satisfies the additivity condition

$$\sum_{i=1}^{N} \mathrm{RC}_i^{\mathrm{CVaR}(\alpha)} = \mathrm{CVaR}(\alpha). \tag{27}$$

To compute (26), we rewrite the definition of CVaR as

$$\begin{aligned}
\mathrm{CVaR}(\alpha) &= \frac{1}{1-\alpha}\int_{\mathrm{VaR}(\alpha)}^{\infty} l f_L(l)dl, \\
&= \frac{1}{1-\alpha}\left\{\mathrm{EL} - \int_0^{\mathrm{VaR}(\alpha)} l f_L(l)dl\right\}, \\
&= \frac{1}{1-\alpha}\left\{\mathrm{EL} - \alpha\mathrm{VaR}(\alpha) + \int_0^{\mathrm{VaR}(\alpha)} F_L(E_i, l)dl\right\},
\end{aligned}$$

where EL is the expected loss of the portfolio,

$$\mathrm{EL} = \sum_{i=1}^{N} E_i \cdot \mathrm{PD}_{r(i)} = \int_0^{\infty} l f_L(l)dl.$$

Hence, we have

$$\begin{aligned}
\frac{\partial \mathrm{CVaR}(\alpha)}{\partial E_i} &= \frac{1}{1-\alpha}\left\{\mathrm{PD}_{r(i)} - \alpha\frac{\partial \mathrm{VaR}(\alpha)}{\partial E_i}\right. \\
&\quad \left. + \frac{\partial \mathrm{VaR}(\alpha)}{\partial E_i}F_L(E_i, \mathrm{VaR}(\alpha)) + \int_0^{\mathrm{VaR}(\alpha)} \frac{\partial F_L(E_i, l)}{\partial E_i}dl\right\}, \\
&= \frac{1}{1-\alpha}\left\{\mathrm{PD}_{r(i)} + \int_0^{\mathrm{VaR}(\alpha)} \frac{\partial F_L(E_i, l)}{\partial E_i}dl\right\}. \qquad (28)
\end{aligned}$$

From (23), the integral on the r.h.s. reads

$$\int_0^{\mathrm{VaR}(\alpha)} \frac{\partial F_L(E_i, l)}{\partial E_i}dl = \frac{1}{2\pi i}\int_{\gamma-i\infty}^{\gamma+i\infty} \frac{e^{\lambda \mathrm{VaR}(\alpha)}}{\lambda^2}\frac{\partial M_L(\lambda)}{\partial E_i}d\lambda. \qquad (29)$$

In summary, the expressions for the risk contributions to VaR and CVaR are given as follows:

$$\mathrm{RC}_i = CE_i \cdot \frac{1}{2\pi i}\int_{\gamma-i\infty}^{\gamma+i\infty} \frac{e^{\lambda \mathrm{VaR}(\alpha)}}{\lambda}\frac{\partial M_L(\lambda)}{\partial E_i}d\lambda, \qquad (30)$$

$$\mathrm{RC}_i^{\mathrm{CVaR}(\alpha)} = E_i \cdot \frac{1}{1-\alpha}\left\{\mathrm{PD}_{r(i)} + \frac{1}{2\pi i}\int_{\gamma-i\infty}^{\gamma+i\infty} \frac{e^{\lambda \mathrm{VaR}(\alpha)}}{\lambda^2}\frac{\partial M_L(\lambda)}{\partial E_i}d\lambda\right\}, \qquad (31)$$

where constant C is determined to satisfy (21).

Later in this paper, we test the accuracy of the risk contributions numerically computed by (30) and (31). In this test, it is helpful to use alternative expressions given by

$$\mathrm{RC}_i \simeq \mathrm{RC}_i^{\Delta} \equiv -\frac{(1-\alpha_+)\mathrm{RC}_i^{\mathrm{CVaR}(\alpha_+)} - (1-\alpha_-)\mathrm{RC}_i^{\mathrm{CVaR}(\alpha_-)}}{\alpha_+ - \alpha_-}, \qquad (32)$$

$$\mathrm{RC}_i^{\mathrm{CVaR}(\alpha)} = E_i \cdot \mathbf{E}\left[\tilde{D}_i \mid \tilde{L} \geq \mathrm{VaR}(\alpha)\right], \qquad (33)$$

where Δ is a constant sufficiently smaller than VaR, and $\alpha_\pm$ is defined by

$$\alpha_\pm \equiv F_L(E_i, \mathrm{VaR}(\alpha) \pm \Delta/2).$$

The proofs of (32) and (33) are given in appendix A.

3 Numerical Laplace Inversion : the Second Step

As explained in the last section, risk measures and risk contributions can be computed by Laplace inversion, if the MGF and its partial derivatives are given. In this section, we assume the MGF to be given already, and review a numerical Laplace inversion algorithm first studied by de Hoog et al. (1982). This is the second step of the fast numerical approximation following the first step explained in the next section.

3.1 Review of related works

Application of Laplace inversion to portfolio risk quantification has already been studied in the literature. The most successful approach would be the saddle-point approximation and its extensions (Martin et al. (2001) and Martin and Ordovás (2006)). In these approximations, one finds a saddle-point of the integrand in (12) or (14), and integrates it along the steepest descent. This method gives a good estimate of the cdf, if the loss distribution is unimodal. However, the approximation becomes worse for skewed distributions. For instance, consider the single-factor model studied in Martin et al. (2001), in which the factor can take only four discrete values. The loss distribution is a weighted sum of four conditional distributions in this model. If the four conditional EL depending on the factor are considerably different from each other, the distribution can have four distinct peaks. Even if the factor is continuous, the distribution can be highly skewed for a concentrated portfolio with several dominant exposures. Anyway, in the plain saddle-point method, VaR can be biased for skewed distributions [7].

Numerical Laplace inversion can be used as another approach to the current problem. Abate et al. (2000) has proposed a Laplace inversion methodology, which consists of two algorithms. First, the Bromwich integral is approximated by an infinite series using the trapezoidal rule. Second, the convergence of the infinite series is accelerated by a method called Euler summation. Glasserman and Ruiz-Mata (2007) has applied this methodology to the single-factor Merton model. They have shown that the cdf is comparatively accurate in small loss region, whereas the accuracy worsens in tail region. This is because the infinite series obtained by the Euler summation is an alternating series, each term of which has a very large absolute value. A high precision calculation would thus be required to achieve sufficient accuracy in this methodology.

[7] To address this, Huang et al. (2007) has studied a method called the adaptive saddle-point approximation. In this method, obligors are classified into two groups to express the conditional tail probability as $\Pr(\tilde{L} > l|\tilde{\mathbf{X}}) = 1 - \Pr(\tilde{L} - \sum_{i\in S_{\mathrm{con}}} E_i\tilde{D}_i \le l|\tilde{\mathbf{X}}) \prod_{i\in S_{\mathrm{con}}} \Pr(\tilde{D}_i = 0|\tilde{\mathbf{X}})$, where S_{con} is the group of obligors i with $E_i > l$. This gives accurate distributions even in the presence of extremely large exposures.

In this section, we review an approach due to de Hoog et al. (1982) and Ahn et al. (2003), who have studied a methodology similar to Abate et al. (2000). We first use the Poisson algorithm to approximate the Bromwich integral by an infinite series. Second, in order to accelerate the slow convergence of the infinite series, we use the Quotient-Difference (QD) algorithm first studied by Rutishauser (1954). The combination of these algorithms will be referred to as *de Hoog algorithm* from now on. The numerical examples presented later will reveal that, in contrast with the Euler summation technique, de Hoog algorithm is quite efficient in measuring tail probability.

3.2 Infinite Series Expansion : the Poisson Algorithm

The notion of the Poisson algorithm is quite simple. We divide the infinite integration path in (14) into the small intervals with *discretization width* h, and evaluate the integral using the trapezoidal rule. The resulting infinite series reads

$$\begin{aligned} F_L^h(l) &\equiv \frac{h}{2\pi}\sum_{k=-\infty}^{\infty}\exp\{(\gamma+ikh)l\}\frac{M_L(\gamma+ikh)}{\gamma+ikh}, \\ &= \frac{h}{\pi}\exp(\gamma l)\left[\frac{M_L(\gamma)}{2\gamma}+\sum_{k=1}^{\infty}\mathrm{Re}\left\{\frac{M_L(\gamma+ikh)}{\gamma+ikh}\exp(ikhl)\right\}\right], \\ &= \frac{h}{\pi}\exp(\gamma l)\mathrm{Re}\left(\sum_{k=0}^{\infty}s_k z_l^k\right), \end{aligned} \tag{34}$$

where we have defined

$$\begin{aligned} &s_0 \equiv \frac{M_L(\gamma)}{2\gamma}, \quad s_k \equiv \frac{M_L(\gamma+ikh)}{\gamma+ikh} \quad (k=1,2,\ldots), \\ &z_l \equiv \exp(ihl). \end{aligned} \tag{35}$$

By truncating this series into a finite sum, we obtain

$$\begin{aligned} F_L(l) &\simeq F_L^{h,N_t}(l), \\ &\equiv \frac{h}{\pi}\exp(\gamma l)\mathrm{Re}\left(\sum_{k=0}^{2N_t}s_k z_l^k\right), \end{aligned} \tag{36}$$

where we have introduced an integer N_t called *truncation parameter*.

The numerical error $e(l) \equiv F_L^{h,N_t}(l) - F_L(l)$ in this approximation is decomposed as $e(l) = e_d(l) + e_t(l)$, where discretization error $e_d(l)$ and truncation error $e_t(l)$ are defined by

$$\begin{aligned} e_d(l) &\equiv F_L^h(l) - F_L(l), \\ e_t(l) &\equiv F_L^{h,N_t}(l) - F_L^h(l). \end{aligned}$$

Of these errors, the discretization error can easily be adjusted by changing parameters γ and h. To determine γ and h, an upper bound formula for the discretization error,

$$|e_d(l)| \leq \frac{\exp(-2\pi\gamma/h)}{1-\exp(-2\pi\gamma/h)} \quad \left(0 \leq l \leq \frac{2\pi}{h}\right), \tag{37}$$

is useful [8]. In principle, the truncation error can also be adjusted by the truncation parameter N_t. However, the convergence of the series (34) is very slow, since s_k converges to zero very slowly as $\mathcal{O}(k^{-1})$. Therefore, N_t must be very large to sufficiently reduce the truncation error.

This slow convergence is unavoidable as long as we deal with discontinuous distribution function. To see this, we rewrite (34) as a trigonometric series

$$F_L^h(l) = \frac{h}{\pi}\exp(\gamma l)\sum_{k=0}^{\infty}\left\{a_k\cos(khl) + b_k\sin(khl)\right\}, \tag{38}$$

where real coefficients a_k and b_k are defined by $a_k - ib_k \equiv s_k$. Expression (38) implies that the Poisson algorithm approximates the cdf by a superposition of trigonometric functions with period $2\pi/kh$. If the cdf is sufficiently smooth, the coefficients a_k and b_k converge to zero very fast, since in general a smooth function can be created by summing up mildly varying functions only. However, if the cdf has discontinuities, highly oscillatory trigonometric functions are necessary to create the discontinuities. This is the origin of the very slow convergence of (34), which is resolved in the next subsection.

3.3 Continued Fraction Expansion : the QD Algorithm

In order to improve the convergence speed of an infinite series, *convergence acceleration methods* are often used. Continued fraction expansion is presumably the most efficient acceleration method that is suitable for the current problem. The QD algorithm is used to convert an infinite series into a continued fraction expansion, which converges drastically faster than the original series does.

Let us begin with a brief review of continued fraction. A continued fraction is a quantity of the form

$$b_0 + \cfrac{a_1}{b_1 + \cfrac{a_2}{b_2 + \cfrac{a_3}{b_3 + \cdots}}}.$$

[8] This upper bound is derived from the Poisson resummation formula that justifies the algorithm's name. For proof, see Abate et al. (2000).

The following notations are also used to represent continued fractions:

$$b_0 + \cfrac{a_1}{b_1 + \cfrac{a_2}{b_2 + \cfrac{a_3}{b_3 + \cdots + \cfrac{a_n}{b_n}}}} = b_0 + \left[\frac{a_k}{b_k}\right]_{k=1}^{n},$$
$$b_0 + \cfrac{a_1}{b_1 + \cfrac{a_2}{b_2 + \cfrac{a_3}{b_3 + \cdots}}} = b_0 + \left[\frac{a_k}{b_k}\right]_{k=1}^{\infty},$$

where $[a_k/b_k]_{k=1}^{n}$ is assumed to be zero if $n \leq 0$.

To prepare for the arguments below, we consider a complex valued sequence s_i that satisfies the following assumption.

Assumption 1 *For all $i \geq 0$ and $k \geq 0$,*

$$|H_k^{(i)}| \neq 0,$$

where $H_k^{(i)}$ is a Hankel matrix

$$H_k^{(i)} = \begin{pmatrix} s_i & s_{i+1} & \cdots & s_{i+k-1} \\ s_{i+1} & s_{i+2} & \cdots & s_{i+k} \\ \vdots & \vdots & \ddots & \vdots \\ s_{i+k-1} & s_{i+k} & \cdots & s_{i+2k-2} \end{pmatrix}, \tag{39}$$

and $|H_k^{(i)}|$ is its determinant with $|H_0^{(i)}| = 1$.

We are now ready to derive a continued fraction representation of an infinite series from the following three lemmas. Proofs of all lemmas are given in appendix C.

Lemma 1 *Let $S(z)$ be a formal power series whose coefficients are given by s_i, namely*

$$S(z) = \sum_{k=0}^{\infty} s_k z^k, \tag{40}$$

and define a sequence c_k by

$$c_0 = s_0,\ c_{2k-1} = -\frac{|H_{k-1}^{(0)}||H_k^{(1)}|}{|H_k^{(0)}||H_{k-1}^{(1)}|},\ c_{2k} = -\frac{|H_{k+1}^{(0)}||H_{k-1}^{(1)}|}{|H_k^{(0)}||H_k^{(1)}|}\quad (k = 1, 2, \ldots). \tag{41}$$

Then, a continued fraction

$$C_n(z) \equiv c_0 \Big/ \left(1 + \left[\frac{c_k z}{1}\right]_{k=1}^{n}\right) \tag{42}$$

is the $[\lfloor n/2 \rfloor / \lfloor (n+1)/2 \rfloor]$ Padé approximation to $S(z)$ [9] *. Or equivalently, $C_n(z)$ is a rational function whose numerator and denominator are polynomials of degree $\lfloor n/2 \rfloor$ and $\lfloor (n+1)/2 \rfloor$ respectively, and satisfies*

$$S(z) - C_n(z) = \mathcal{O}(z^{n+1}).$$

[9] $\lfloor x \rfloor$ denotes the floor function, i.e., the largest integer smaller than or equal to x. See appendix B for the definition of Padé approximation.

		$q_1^{(0)}$	$\rightarrow$	$e_1^{(0)}$		$q_2^{(0)}$	$e_2^{(0)}$	$q_3^{(0)}$
	$\nearrow$		$\nearrow$					$\vdots$
$e_0^{(1)}$	$\rightarrow$	$q_1^{(1)}$		$e_1^{(1)}$	$\rightarrow$	$q_2^{(1)}$	$e_2^{(1)}$	
			$\nearrow$		$\nearrow$		$\vdots$	
$e_0^{(2)}$		$q_1^{(2)}$	$\rightarrow$	$e_1^{(2)}$		$q_2^{(2)}$		
						$\vdots$		
$e_0^{(3)}$		$q_1^{(3)}$		$e_1^{(3)}$				
				$\vdots$				
$e_0^{(4)}$		$q_1^{(4)}$						
$\vdots$		$\vdots$						

Table 1: QD algorithm

This lemma enables us to obtain a continued fraction approximating $S(z)$, from Hankel determinants. However, it requires much computation time to compute the Hankel determinants directly. Instead, we use the QD algorithm presented below.

Lemma 2 (QD algorithm) *Let $e_k^{(i)}$ and $q_k^{(i)}$ be the sequences defined by*

$$e_0^{(i)} = 0,\ q_1^{(i)} = s_{i+1}/s_i \quad (i = 0, 1, 2, \ldots), \tag{43}$$

$$\begin{cases} e_k^{(i)} = e_{k-1}^{(i+1)} + q_k^{(i+1)} - q_k^{(i)} \\ q_{k+1}^{(i)} = q_k^{(i+1)} e_k^{(i+1)} / e_k^{(i)} \end{cases} \quad (i = 0, 1, 2, \ldots; k = 1, 2, \ldots). \tag{44}$$

Then, they satisfy

$$e_k^{(i)} = \frac{|H_{k+1}^{(i)}||H_{k-1}^{(i+1)}|}{|H_k^{(i)}||H_k^{(i+1)}|}, \quad q_k^{(i)} = \frac{|H_{k-1}^{(i)}||H_k^{(i+1)}|}{|H_k^{(i)}||H_{k-1}^{(i+1)}|} \quad (i = 0, 1, 2 \ldots; k = 1, 2, \ldots). \tag{45}$$

Recurrence relation (44) can be understood as a "parallelogram rule" exhibited in Table 1. In this table, initial values s_{i+1}/s_i are assigned to $q_1^{(i)}$, and the remaining sequences are successively determined from left to upper right. The resultant sequences $e_k^{(0)}$ and $q_k^{(0)}$ are used as the coefficients of the desired continued fraction. Obviously, by this algorithm, c_k can be computed in polynomial time with respect to the dimension of Hankel matrices. Finally, we can quickly compute the desired continued fraction by the following lemma.

Lemma 3 *Let $A_n(z)$ and $B_n(z)$ be the sequences of polynomials defined by*

$$A_{-1}(z) = 0,\ A_0(z) = c_0,\ B_{-1}(z) = 1,\ B_0(z) = 1, \tag{46}$$

$$\begin{cases} A_{n+2}(z) = A_{n+1}(z) + c_{n+2} z A_n(z) \\ B_{n+2}(z) = B_{n+1}(z) + c_{n+2} z B_n(z) \end{cases} \quad (n = -1, 0, 1, 2, \ldots). \tag{47}$$

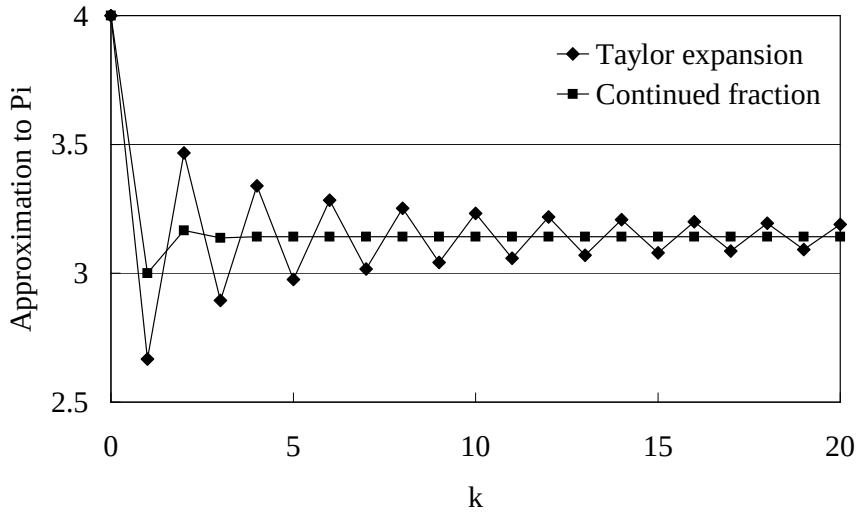

Figure 1: Convergence behavior of two approximations to π.

Then, $A_n(z)$ and $B_n(z)$ are polynomials of degree $\lfloor n/2 \rfloor$ and $\lfloor (n+1)/2 \rfloor$ with $B_n(0) = 1$, and satisfy

$$C_n(z) = \frac{A_n(z)}{B_n(z)}, \tag{48}$$

where $C_n(z)$ is defined by (42).

Let us consider numerical computation of π to illustrate the performance of continued fraction expansion. First, we expand $\tan^{-1}(x)$ as a Taylor series,

$$\begin{aligned} \tan^{-1}(x) &= \sum_{k=0}^{\infty} \frac{(-1)^k x^{2k+1}}{2k+1}, \\ &= x \sum_{k=0}^{\infty} \frac{(-1)^k (x^2)^k}{2k+1}. \end{aligned} \tag{49}$$

Setting $s_i = (-1)^i/(2i+1)$ in (43), we obtain

$$e_k^{(i)} = -\frac{4k^2}{(2i+4k-1)(2i+4k+1)}, \quad q_k^{(i)} = -\frac{(2i+2k-1)^2}{(2i+4k-3)(2i+4k-1)},$$

from (44) after a little algebra. Hence, for a small x,

$$\tan^{-1}(x) = \cfrac{x}{1 + \left[\cfrac{\{k^2/(2k-1)(2k+1)\}x^2}{1}\right]_{k=1}^{\infty}} \tag{50}$$

follows from lemma 1 and 2. Let us compare these two expansions using an identity $\pi = 4\tan^{-1}(1)$. The convergence behavior shown in Figures 1 and 2 indicates that

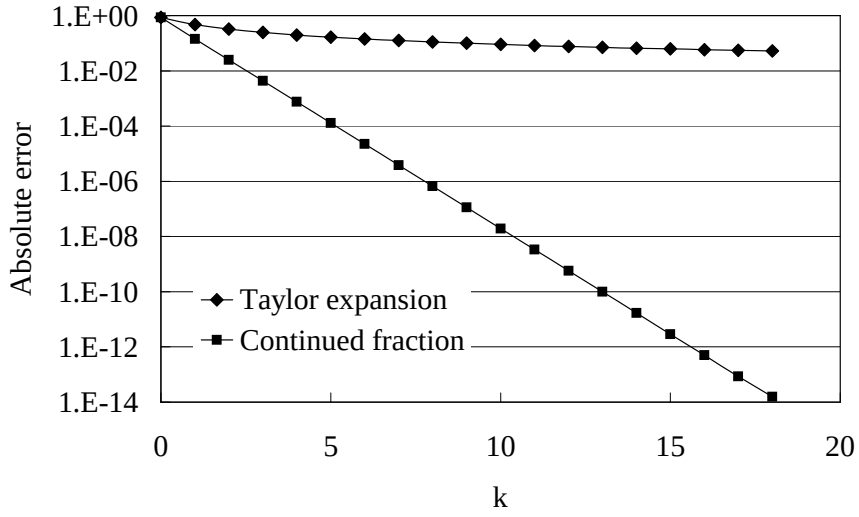

Figure 2: Absolute error of two approximations to π.

continued fraction drastically outperforms Taylor expansion. For instance, truncating (49) at $k = 1,000,000$ yields

$$\pi \simeq 3.1415936\cdots,$$

which is correct only to five decimal places. In contrast, truncating (50) at $k = 18$, we obtain

$$\pi \simeq 3.14159265358981\cdots,$$

which is correct to twelve decimal places.

For $x = 1$, the k-th term in Taylor series (49) converges as $\mathcal{O}(k^{-1})$, which is the same behavior as that of (34). We can thus expect that continued fraction expansion is also useful in the current portfolio model.

3.4 Summary

In this subsection, we summarize de Hoog algorithm explained in the preceding subsections. The cdf can be computed by the following four steps [10].

L-1. Computation of infinite series coefficients:

Compute

$$s_0 = \frac{M_L(\gamma)}{2\gamma}, \quad s_k = \frac{M_L(\gamma + ikh)}{\gamma + ikh} \quad (k = 1, 2, \ldots, 2N_t),$$

using the algorithm explained in the next section.

[10] An example of matlab function implementing de Hoog algorithm is given in Hollenbeck (1998).

L-2. Computation of continued fraction coefficients:

Compute

$$c_0 = s_0,\ c_{2k-1} = -q_k^{(0)},\ c_{2k} = -e_k^{(0)} \quad (k = 1, 2, \ldots, N_t),$$

using the QD algorithm

$$e_0^{(i)} = 0,\ q_1^{(i)} = s_{i+1}/s_i \quad (i = 0, 1, 2, \ldots),$$
$$\begin{cases} e_k^{(i)} = e_{k-1}^{(i+1)} + q_k^{(i+1)} - q_k^{(i)} \\ q_{k+1}^{(i)} = q_k^{(i+1)} e_k^{(i+1)} / e_k^{(i)} \end{cases} \quad (i = 0, 1, 2, \ldots; k = 1, 2, \ldots).$$

L-3. Computation of continued fraction:

Let S_L be the *set of cdf valuation points*, i.e., the set of loss levels where we wish to compute the cdf. Then, for all $l \in S_L$, compute

$$C_{2N_t}(z_l) = \frac{A_{2N_t}(z_l)}{B_{2N_t}(z_l)}, \quad z_l = \exp(ihl),$$

using the recurrence relation

$$A_{-1}(z) = 0,\ A_0(z) = c_0,\ B_{-1}(z) = 1,\ B_0(z) = 1,$$
$$\begin{cases} A_{n+2}(z) = A_{n+1}(z) + c_{n+2} z A_n(z) \\ B_{n+2}(z) = B_{n+1}(z) + c_{n+2} z B_n(z) \end{cases} \quad (n = -1, 0, 1, \ldots, 2N_t - 2).$$

L-4. Computation of cdf:

Compute

$$F_L(l) \simeq \frac{h}{\pi} \exp(\gamma l) \mathrm{Re}\{C_{2N_t}(z_l)\},$$

for all $l \in S_L$.

Let us explain here an advantage of de Hoog algorithm concerning computational speed. As will become clear in the next section, it is very time-consuming to compute the MGF, particularly for large portfolios with many systematic factors. However, this computation is necessary only in step L-1. Once the infinite series coefficients are given in step L-1, we can obtain the whole distribution function very quickly by steps L-3 and L-4 [11].

The risk contributions to VaR and CVaR are computed by repeating the following four steps for all obligors i. We assume that VaR has already been computed from the cdf obtained by step L-4.

[11] In contrast, the Euler summation technique in Abate et al. (2000) requires re-calculation of the MGF for each loss level l.

L-5. Computation of infinite series coefficients:

Choose an obligor i and compute

$$\begin{cases} s_0^i = \dfrac{1}{2\gamma}\dfrac{\partial M_L(\gamma)}{\partial E_i}, \quad s_k^i = \dfrac{1}{\gamma + ikh}\dfrac{\partial M_L(\gamma + ikh)}{\partial E_i} & \text{(for VaR)}, \\ s_0^i = \dfrac{1}{2\gamma^2}\dfrac{\partial M_L(\gamma)}{\partial E_i}, \quad s_k^i = \dfrac{1}{(\gamma + ikh)^2}\dfrac{\partial M_L(\gamma + ikh)}{\partial E_i} & \text{(for CVaR)}, \end{cases}$$

for $k = 1, 2, \ldots, 2N_t$, where $\partial M_L(\lambda)/\partial E_i$ is computed by the algorithm explained in the next section.

L-6. Computation of continued fraction coefficients:

Compute

$$c_0^i = s_0^i, \ c_{2k-1}^i = -q_k^{(0)}, \ c_{2k}^i = -e_k^{(0)} \quad (k = 1, 2, \ldots, N_t),$$

using the QD algorithm

$$\begin{aligned} & e_0^{(j)} = 0, \ q_1^{(j)} = s_{j+1}^i / s_j^i \quad (j = 0, 1, 2, \ldots), \\ & \begin{cases} e_k^{(j)} = e_{k-1}^{(j+1)} + q_k^{(j+1)} - q_k^{(j)} \\ q_{k+1}^{(j)} = q_k^{(j+1)} e_k^{(j+1)} / e_k^{(j)} \end{cases} \quad (j = 0, 1, 2, \ldots; k = 1, 2, \ldots). \end{aligned}$$

L-7. Computation of continued fraction:

Compute

$$C_{2N_t}^i(z) = \frac{A_{2N_t}(z)}{B_{2N_t}(z)}, \quad z = \exp(ih\mathrm{VaR}(\alpha)),$$

using the recurrence relation

$$\begin{aligned} & A_{-1}(z) = 0, \ A_0(z) = c_0^i, \ B_{-1}(z) = 1, \ B_0(z) = 1, \\ & \begin{cases} A_{n+2}(z) = A_{n+1}(z) + c_{n+2}^i z A_n(z) \\ B_{n+2}(z) = B_{n+1}(z) + c_{n+2}^i z B_n(z) \end{cases} \quad (n = -1, 0, 1, \ldots, 2N_t - 2). \end{aligned}$$

L-8. Computation of risk contributions:

Compute

$$\begin{aligned} \mathrm{RC}_i &\simeq CE_i \cdot \frac{h}{\pi} \exp(\gamma \mathrm{VaR}(\alpha)) \mathrm{Re}\{C_{2N_t}^i(z)\}, \\ \mathrm{RC}_i^{\mathrm{CVaR}(\alpha)} &\simeq E_i \cdot \frac{1}{1-\alpha} \left[\mathrm{PD}_{r(i)} + \frac{h}{\pi} \exp(\gamma \mathrm{VaR}(\alpha)) \mathrm{Re}\{C_{2N_t}^i(z)\} \right], \end{aligned}$$

where C is determined to satisfy $\sum_{i=1}^N \mathrm{RC}_i = \mathrm{VaR}(\alpha)$.

4 Computation of MGF : the First Step

4.1 Generalities on Numerical Integration

Since the expectation values required to obtain the MGF can not be analytically calculated, they must be computed by numerical integration. In this subsection, we briefly review the idea of numerical integration methods. Most of the well-known numerical integration algorithms take the form of

$$\int_{\mathbf{x}\in I} g(\mathbf{x})d\mathbf{x} \simeq \sum_{k=1}^{N_I} w_k g(\mathbf{x}_k), \tag{51}$$

where we have introduced the notations as

- I : D-dimensional integration domain.
- $\mathbf{x}_k$: *Integration points* in I.
- w_k : *Integration weights* associated with $\mathbf{x}_k$.
- N_I : The number of the integration points.

Approximation (51) implies that numerical integration can be considered as a procedure for appropriately determining $(w_k, \mathbf{x}_k)$.

In general, computational accuracy depends on the form of the integrand $g(\mathbf{x})$, even if the same $(w_k, \mathbf{x}_k)$ is used. It is important to choose appropriate integration methods suitable for the integrand. For instance, suppose that $g(\mathbf{x})$ is of the form $g(\mathbf{x}) = h(\mathbf{x})\phi_D(\mathbf{x})$, where $h(\mathbf{x})$ is an arbitrary function and $\phi_D(\mathbf{x})$ is the D-dimensional standard normal pdf. Then, we have the following choices.

Gauss-Hermite quadrature

If $D = 1$, the Gauss-Hermite quadrature can be used to determine $\mathbf{x}_k$ and w_k.

Good Lattice Points

For $D = 2, 3, 4$, good lattice points can be used as $\mathbf{x}_k$.

Monte Carlo integration

For multi-dimensional cases with $D \gtrsim 4$, Monte Carlo integration is presumably the most efficient algorithm. In Monte Carlo integration, standard normal random numbers are used as $\mathbf{x}_k$, and (51) is computed by replacing $g(\mathbf{x}_k)$ on the r.h.s. with $h(\mathbf{x}_k)$, and setting $w_k = 1/N_I$.

Note that the expectation values in (15) and (24) are integrations with respect to standard normal systematic factors, and hence we can use one of the above methods

depending on N_f. However, an appropriate choice of integration methods is not sufficient for the current problem, in particular for large portfolios. In these cases, we have to consider how to reduce the computation time of the integrand at a *single* integration point. This is the main issue discussed in the following subsections.

4.2 Merton Model Revisited

First, let us consider how to estimate loading vectors $\boldsymbol{\alpha}_i$ used in the multi-factor Merton model. In principle, the loading vector of obligors can be different from each other. This means that pairwise correlations can be different among all possible pairs of obligors. However, the data needed to estimate the correlations are limited, especially for private firms for which we can not observe their market stock values. Moreover, even if the pairwise correlations could be appropriately estimated, it would be difficult to generate random firm-values with pairwise correlations for a large portfolio.

A standard practice in banks is to decompose a portfolio into industrial or regional sectors, and assume the loading vectors to depend only on the sectors, not on individual obligors. For instance, one first estimates the correlations among stock market sector indices, which are mapped somehow to the sectors internally used in a bank. Then, the loading vectors are determined so that they reproduce the estimated correlations among the indices. This practical assumption is a key to a drastically fast algorithm for computing MGF.

This standard practice is described as follows. Let $\boldsymbol{\alpha}_S$ be the loading vector for sector S, and rewrite $\boldsymbol{\alpha}_i$ as

$$\boldsymbol{\alpha}_i = \boldsymbol{\alpha}_{S(i)}, \tag{52}$$

where $S(i)$ is the sector to which obligor i belongs. Then, the firm-value reads

$$\tilde{Z}_i = \beta_{S(i)} \tilde{Y}^{S(i)} + \sqrt{1 - \beta_{S(i)}^2}\, \tilde{\varepsilon}_i, \tag{53}$$

where we have defined *loading scalar* β_S, sectoral factor $\tilde{Y}^S$ and *unit loading vector* $\mathbf{e}_S$ by

$$\begin{aligned} \beta_S &\equiv \left|\boldsymbol{\alpha}_S\right|, \\ \tilde{Y}^S &\equiv \mathbf{e}_S \cdot \tilde{\mathbf{X}}, \\ \mathbf{e}_S &\equiv \boldsymbol{\alpha}_S / \left|\boldsymbol{\alpha}_S\right|. \end{aligned} \tag{54}$$

It is important that $\tilde{Y}^{S(i)}$ is uniquely determined by obligor i's sector [12]. This means that a sub-portfolio consisting only of the obligors in a specific sector is described by the single-factor Merton model.

[12] The practical assumption here can be relaxed by allowing the norm of $\boldsymbol{\alpha}_i$ to depend on i. Then, we have $\boldsymbol{\alpha}_i = |\beta_i| \mathbf{e}_{S(i)}$ and $\tilde{Z}_i = \beta_i \tilde{Y}^{S(i)} + \sqrt{1 - \beta_i^2}\tilde{\varepsilon}_i$. Even under this relaxed condition, the fast computation idea explained in the next subsection works.

Given this reformulation, we decompose $\tilde{L}$ into the sum of the losses $\tilde{L}_S$ arising from sector S as

$$\tilde{L} = \sum_{S=1}^{N_S} \tilde{L}_S, \tag{55}$$

where N_S is the number of sectors [13]. Then, the conditional MGF for $\tilde{L}_S$ is given by

$$M_{L_S}(\lambda; \mathbf{x}) = \prod_{i \text{ s.t. } S(i)=S} \left\{1 - p_i(\mathbf{x}) + p_i(\mathbf{x}) e^{-\lambda E_i}\right\}, \tag{56}$$

which we refer to as *sector MGF* below. By multiplying $M_{L_S}(\lambda; \mathbf{x})$ for all sectors, we obtain the total conditional MGF as

$$M_L(\lambda; \mathbf{x}) = \prod_{S=1}^{N_S} M_{L_S}(\lambda; \mathbf{x}). \tag{57}$$

If numerical integration weights and points $(w_k, \mathbf{x}_k)$ are given, an approximation to the unconditional MGF

$$\begin{aligned} M_L(\lambda) &\simeq \sum_{k=1}^{N_I} w_k M_L(\lambda; \mathbf{x}_k), \\ &= \sum_{k=1}^{N_I} w_k \prod_{S=1}^{N_S} M_{L_S}(\lambda; \mathbf{x}_k) \end{aligned} \tag{58}$$

is derived from (11). Here, note that sector MGF $M_{L_S}(\lambda; \mathbf{x}_k)$ depends only on a scalar $\mathbf{e}_S \cdot \mathbf{x}_k$, i.e., the projection of $\mathbf{x}_k$ on axis $\mathbf{e}_S$. It is thus convenient to define

$$M'_{L_S}(\lambda; y) \equiv \prod_{i \text{ s.t. } S(i)=S} \left\{1 - p'_i(y) + p'_i(y) e^{-\lambda E_i}\right\}, \tag{59}$$

where a new function denoting conditional PD is defined by

$$\begin{aligned} p'_i(y) &\equiv \Pr\left(\tilde{Z}_i < C_i \,\middle|\, \tilde{Y}^{S(i)} = y\right), \\ &= \Phi\left(\frac{C_i - \beta_{S(i)} y}{\sqrt{1 - \beta_{S(i)}^2}}\right). \end{aligned}$$

Using (59), we obtain

$$M_L(\lambda) \simeq \sum_{k=1}^{N_I} w_k \prod_{S=1}^{N_S} M'_{L_S}(\lambda; \mathbf{e}_S \cdot \mathbf{x}_k). \tag{60}$$

The above argument is just an algebraic manipulation, and does not improve computational speed at all. However, a further approximation applied to (60) significantly reduces computation time. This will be discussed in the next subsection.

[13] From now on, we assume $N_S \geq N_f$ without loss of generality.

4.3 Idea of Fast Computation

Suppose that there exist $N_f \gtrsim 4$ systematic factors, which require Monte Carlo integration according to the criteria given in subsection 4.1. To compute the MGF in this case, at least $N_I \sim 10^6$ integration points are necessary to sufficiently reduce the statistical error due to Monte Carlo integration. Accordingly, every sector MGF must be computed at least 1 million times if we naively use expression (60). This lengthy computation can be avoided by noting the following two points. First, a bank's portfolio consists of single factor sub-portfolios under the practical assumption of the last subsection. Second, it has been empirically found that $N_I \sim 100$ integration points are sufficient to obtain sufficiently accurate loss distribution for a single factor portfolio. These two points have led us to find an algorithm explained below.

The main idea of fast computation is *discretization of sectoral factors*. First, we put a finite number of points on the real axis as

$$G \equiv \{g_1, g_2, \ldots, g_{N_g}\} \quad (g_1 < g_2 < \cdots < g_{N_g}). \tag{61}$$

We refer to g_m and N_g as *discretized valuation point* and *discretization parameter*, respectively. Next, we define

$$M_{L_S}^{(m)}(\lambda) \equiv M'_{L_S}(\lambda; g_m), \tag{62}$$

which represents the sector MGF valuated at discretized valuation point g_m. Then, our final approximation is given by

$$\begin{aligned} M_L(\lambda) &\simeq \sum_{k=1}^{N_I} w_k \prod_{S=1}^{N_S} M'_{L_S}(\lambda; \mathbf{e}_S \cdot \mathbf{x}_k), \\ &\simeq \sum_{k=1}^{N_I} w_k \prod_{S=1}^{N_S} M_{L_S}^{(m_k^S)}(\lambda), \end{aligned} \tag{63}$$

where an integer m_k^S specifies a discretized valuation point $g_{m_k^S}$ close to $\mathbf{e}_S \cdot \mathbf{x}_k$. The result (63) gives a good approximation to (60), provided that $N_g \sim 100$ and discretized valuation points are appropriately chosen. This is because $M'_{L_S}(\lambda; y)$ is a continuous function of y, and $\tilde{Y}^S = \mathbf{e}_S \cdot \tilde{\mathbf{X}}$ has a very small probability of taking large absolute values. In this paper, we adopt the following simple setup:

- G : A set of points equally separated in the interval $[-5, 5]$,
  i.e., $\{g_m = -5 + 10(m-1)/(N_g - 1) \mid m = 1, 2, \ldots, N_g\}$.
- m_k^S : An integer m minimizing $\left|g_m - \mathbf{e}_S \cdot \mathbf{x}_k\right|$.

Let us compare two expressions (60) and (63), focusing on their computational speed. The overall computational complexity of (60) is $\mathcal{O}(N_t N_I N)$ [14]. It is thus unrealistic

[14] Recall that $M_L(\lambda)$ must be computed for $2N_t + 1$ different values of λ in step L-1 of subsection 3.4.

to simply use (60) in large banks employing $N_f \gtrsim 4$ systematic factors. However, this problem is resolved in (63). Note that $M_{L_S}^{(m)}(\lambda)$ does not depend on integration points, and can be computed only from a set of discretized valuation points G. This allows us to compute $M_{L_S}^{(m)}(\lambda)$ before generating $\mathbf{x}_k$ and store the results in RAM. When we compute (63), we just find an appropriate m_k^S for each index k and sector S, read the corresponding values of $M_{L_S}^{(m_k^S)}(\lambda)$ from RAM, and multiply them for all sectors. Since at most $N_S \sim 100$ as mentioned in Introduction, this can be done in a relatively short time. In summary, the computational complexity has been reduced to $\mathcal{O}(N_t(N_g N + N_I N_S))$ in (63), where $N_g N$ and $N_I N_S$ correspond to the loops for computing the sector MGF and total MGF, respectively. Given the typical values of N_g, N, N_I, and N_S presented above, it is obvious that this complexity reduction dramatically reduces computation time.

Furthermore, approximation (63) outperforms Monte Carlo simulation with N_I loss scenarios in its computational accuracy. The N_I integration points in (63) are in one-to-one correspondence with *conditional cdf*, which are summed up to obtain the total cdf. On the other hand, Monte Carlo simulation generates loss scenarios, which are in one-to-one correspondence with *points in the total cdf*. Therefore, the risk measures computed by (63) are obviously more accurate than those obtained by Monte Carlo simulation with N_I loss scenarios.

4.4 Application to Risk Contributions

In this subsection, we apply the fast numerical algorithm presented in the last subsection to the computation of risk contributions. In order to compute risk contributions, we have to compute $\partial M_L(\lambda)/\partial E_i$ used in (30) and (31). A naive numerical integration of (24) reads

$$\begin{aligned}
\frac{\partial M_L(\lambda)}{\partial E_i} &\simeq -\lambda \sum_{k=1}^{N_I} w_k p_i(\lambda; \mathbf{x}_k) M_L(\lambda; \mathbf{x}_k), \\
&= -\lambda \sum_{k=1}^{N_I} w_k p_i(\lambda; \mathbf{x}_k) \prod_{S=1}^{N_S} M'_{L_S}(\lambda; \mathbf{e}_S \cdot \mathbf{x}_k), \\
&\simeq -\lambda \sum_{k=1}^{N_I} w_k p_i(\lambda; \mathbf{x}_k) \prod_{S=1}^{N_S} M_{L_S}^{(m_k^S)}(\lambda). \qquad (64)
\end{aligned}$$

However, even if we could temporarily store the value of $\prod_{S=1}^{N_S} M_{L_S}^{(m_k^S)}(\lambda)$ in RAM after computing the MGF by (63), it would still require an overall complexity $\mathcal{O}(N_t N_I N)$ to compute (64) for all obligors. It is thus impractical to use (64) for a large portfolio with many systematic factors.

To obtain an expression more suitable for numerical computation, we return to (24) and rewrite it as

$$\frac{\partial M_L(\lambda)}{\partial E_i} = -\lambda \sum_{m=1}^{N_g} \mathbf{E}\left[p_i(\lambda; \tilde{\mathbf{X}}) M_L(\lambda; \tilde{\mathbf{X}}) 1_{\mathbf{e}_{S(i)} \cdot \tilde{\mathbf{X}} \in I_m} \right], \qquad (65)$$

where I_m is the interval containing g_m defined by

$$I_m \equiv \begin{cases} (-\infty, (g_1+g_2)/2] & (m=1), \\ ((g_{m-1}+g_m)/2, (g_m+g_{m+1})/2] & (m=2,3,\ldots,N_g-1), \\ ((g_{N_g-1}+g_{N_g})/2, \infty) & (m=N_g). \end{cases}$$

Note that $\mathbf{e}_S \cdot \mathbf{x}_k \in I_m$ is equivalent to $m_k^S = m$. To evaluate (65), it is convenient to define

$$p_i'(\lambda; y) \equiv \frac{p_i'(y)\exp(-\lambda E_i)}{1 - p_i'(y) + p_i'(y)\exp(-\lambda E_i)}, \tag{66}$$

since $p_i(\lambda; \tilde{\mathbf{X}})$ depends only on a scalar $\mathbf{e}_{S(i)} \cdot \tilde{\mathbf{X}}$. Then, we can approximate the expectation value in (65) using discretized valuation points by

$$\mathbf{E}\left[p_i(\lambda;\tilde{\mathbf{X}}) M_L(\lambda;\tilde{\mathbf{X}}) 1_{\mathbf{e}_{S(i)}\cdot\tilde{\mathbf{X}} \in I_m}\right] \simeq p_i'(\lambda; g_m)\mathbf{E}\left[M_L(\lambda;\tilde{\mathbf{X}}) 1_{\mathbf{e}_{S(i)}\cdot\tilde{\mathbf{X}} \in I_m}\right]. \tag{67}$$

Using integration points $\mathbf{x}_k$, the expectation value on the r.h.s. of (67) can be further approximated as

$$\begin{aligned} \mathbf{E}\left[M_L(\lambda;\tilde{\mathbf{X}}) 1_{\mathbf{e}_{S(i)}\cdot\tilde{\mathbf{X}} \in I_m}\right] &\simeq \sum_{k=1}^{N_I} w_k 1_{\mathbf{e}_{S(i)}\cdot\mathbf{x}_k \in I_m} \prod_{S=1}^{N_S} M_{L_S}(\lambda; \mathbf{x}_k), \\ &= \sum_{k \in K_m^{S(i)}} w_k \prod_{S=1}^{N_S} M_{L_S}'(\lambda; \mathbf{e}_S \cdot \mathbf{x}_k), \\ &\simeq \sum_{k \in K_m^{S(i)}} w_k \prod_{S=1}^{N_S} M_{L_S}^{(m_k^S)}(\lambda), \end{aligned} \tag{68}$$

where we have defined a subset K_m^S of the integration point index by

$$K_m^S \equiv \{k | k = 1, 2, \ldots, N_I, m_k^S = m\}.$$

Combining (67) and (68) with (65), we finally obtain

$$\frac{\partial M_L(\lambda)}{\partial E_i} \simeq -\lambda \sum_{m=1}^{N_g} p_i^{(m)}(\lambda) P_m^{S(i)}(\lambda), \tag{69}$$

where we have defined

$$\begin{aligned} p_i^{(m)}(\lambda) &\equiv p_i'(\lambda; g_m) \\ P_m^S(\lambda) &\equiv \sum_{k \in K_m^S} w_k \prod_{S'=1}^{N_S} M_{L_{S'}}^{(m_k^{S'})}(\lambda). \end{aligned} \tag{70}$$

Expression (69) is superior to (64) for the following reason. First, note that a factor $\prod_{S=1}^{N_S} M_{L_S}^{(m_k^S)}(\lambda)$ appears in (63) and (70) in common, as mentioned previously. Therefore,

it requires only a complexity $\mathcal{O}(N_t N_I N_S)$ to compute $P_m^S(\lambda)$ and store the results in RAM, if this computation is done in parallel with (63). When we finally compute (69), the value of $P_m^S(\lambda)$ is read from RAM. This computation requires a complexity $\mathcal{O}(N_t N_g N)$. In summary, the overall complexity has been reduced to $\mathcal{O}(N_t(N_I N_S + N_g N))$ in (69), where $N_I N_S$ and $N_g N$ correspond to the loops for computing $P_m^S(\lambda)$ and $\partial M_L(\lambda)/\partial E_i$, respectively.

In principle, the above algorithm is sufficient to compute risk contributions via (30) and (31). However, (31) often gives inaccurate results because of the error due to numerical integration. This can be remedied by rewriting $\mathrm{PD}_{r(i)}$ as

$$\begin{aligned}
\mathrm{PD}_{r(i)} &= \mathbf{E}\left[p_i'(\mathbf{e}_{S(i)} \cdot \tilde{\mathbf{X}})\right] \\
&= \sum_{m=1}^{N_g} \mathbf{E}\left[p_i'(\mathbf{e}_{S(i)} \cdot \tilde{\mathbf{X}}) 1_{\mathbf{e}_{S(i)} \cdot \tilde{\mathbf{X}} \in I_m}\right] \\
&\simeq \sum_{m=1}^{N_g} p_i'(g_m) \mathbf{E}\left[1_{\mathbf{e}_{S(i)} \cdot \tilde{\mathbf{X}} \in I_m}\right] \\
&\simeq \sum_{m=1}^{N_g} p_i'(g_m) \left(\sum_{k=1}^{N_I} w_k 1_{\mathbf{e}_{S(i)} \cdot \mathbf{x}_k \in I_m}\right) \\
&= \sum_{m=1}^{N_g} p_i'(g_m) \left(\sum_{k \in K_m^{S(i)}} w_k\right). \qquad (71)
\end{aligned}$$

Replacing $\mathrm{PD}_{r(i)}$ in (31) with its approximated version (71), we can "cancel out" the numerical error contained in (71) and $\partial M_L(\lambda)/\partial E_i$. This trick improves to a considerable degree the accuracy of the risk contributions to CVaR.

As mentioned in Introduction, it is helpful for portfolio management if we can know how the total risk of a portfolio is increased (or decreased) by individual transactions. This can be done instantaneously by applying the above algorithm, if we restrict ourselves to the case where the exposure of only *one* obligor changes. Suppose that the exposure of a specific obligor i changes as $E_i \to E_i^{\mathrm{new}}$. It is a straightforward exercise to verify that the approximated MGF (63) then changes as

$$M_L(\lambda) \to M_L^{\mathrm{new}}(\lambda) \simeq \sum_{m=1}^{N_g} q_i^{(m)}(\lambda) P_m^{S(i)}(\lambda), \qquad (72)$$

where $q_i^{(m)}(\lambda)$ is defined by

$$q_i^{(m)}(\lambda) \equiv \frac{1 - p_i'(g_m) + p_i'(g_m)\exp(-\lambda E_i^{\mathrm{new}})}{1 - p_i'(g_m) + p_i'(g_m)\exp(-\lambda E_i)}. \qquad (73)$$

Expression (72) can be computed instantaneously, if a program code is implemented so that the value of $P_m^S(\lambda)$ remains in RAM after the first risk measurement is finished [15].

[15] This algorithm can easily be extended to the case where the rating of only one obligor changes.

In addition, it is possible to calculate higher order derivatives of $M_L(\lambda)$ with respect to E_i, using (73) as follows. Let us regard $q_i^{(m)}(\lambda)$ as a function of $E_i^{\rm new}$, and write down its Taylor expansion around E_i as

$$q_i^{(m)}(\lambda) = 1 + \sum_{n=1}^{\infty} \frac{1}{n!}(-\lambda)^n p_i^{(m)}(\lambda)(E_i^{\rm new} - E_i)^n,$$

where we have used

$$\left.\frac{\partial^n q_i^{(m)}(\lambda)}{(\partial E_i^{\rm new})^n}\right|_{E_i^{\rm new}=E_i} = \frac{(-\lambda)^n p_i'(g_m)\exp(-\lambda E_i)}{1 - p_i'(g_m) + p_i'(g_m)\exp(-\lambda E_i)} = (-\lambda)^n p_i^{(m)}(\lambda).$$

Then, from (72) we have

$$\frac{\partial^n M_L(\lambda)}{\partial E_i^n} \simeq (-\lambda)^n \sum_{m=1}^{N_g} p_i^{(m)}(\lambda) P_m^{S(i)}(\lambda). \tag{74}$$

This can be used to compute higher order derivatives of VaR and CVaR along the same lines as subsection 2.4. In particular, the second order derivative is analogous to "Gamma risk", which is familiar in market risk. Finite difference (72) and its infinitesimal version (74) give valuable information about the sensitivity of credit portfolio risk to individual transactions.

4.5 Summary

In this subsection, we summarize the algorithm for computing MGF and its partial derivative with respect to exposures. The MGF can be computed by the following two steps.

M-1. Computation of sector MGF:

Compute

$$M_{L_S}^{(m)}(\lambda) = \prod_{i \text{ s.t. } S(i)=S} \left\{1 - p_i'(g_m) + p_i'(g_m)e^{-\lambda E_i}\right\},$$

for $S = 1, 2, \ldots, N_S$ and $m = 1, 2, \ldots, N_g$.

M-2. Computation of $P_m^S(\lambda)$ and MGF:

Set $P_m^S(\lambda) = 0$ for all $S = 1, 2, \ldots, N_S, m = 1, 2, \ldots, N_g$, and $M_L(\lambda) = 0$. Then, repeat (a) and (b) given below for $k = 1, 2, \ldots, N_I$, to finally obtain [16]

$$\begin{aligned} P_m^S(\lambda) &= \sum_{k \in K_m^S} w_k \prod_{S'=1}^{N_S} M_{L_{S'}}^{(m_k^{S'})}(\lambda) \quad (S = 1, 2, \ldots, N_S, m = 1, 2, \ldots, N_g), \\ M_L(\lambda) &\simeq \sum_{k=1}^{N_I} w_k \prod_{S=1}^{N_S} M_{L_S}^{(m_k^S)}(\lambda). \end{aligned}$$

[16] If there is no need to compute risk contributions, computation of $P_m^S(\lambda)$ can be omitted.

(a) Compute $\prod_{S=1}^{N_S} M_{L_S}^{(m_k^S)}(\lambda)$, using $M_{L_S}^{(m)}(\lambda)$ obtained in step M-1.

(b) Add $w_k \prod_{S=1}^{N_S} M_{L_S}^{(m_k^S)}(\lambda)$ to $P_{m_k^S}^S(\lambda)$ for all $S = 1, 2, \ldots, N_S$, and to $M_L(\lambda)$.

The partial derivative of MGF with respect to E_i can be computed by the following two steps.

M-3. Computation of twisted conditional PD:

Compute

$$p_i^{(m)}(\lambda) = \frac{p_i'(g_m)\exp(-\lambda E_i)}{1 - p_i'(g_m) + p_i'(g_m)\exp(-\lambda E_i)},$$

for $m = 1, 2, \ldots, N_g$.

M-4. Computation of $\partial M_L(\lambda)/\partial E_i$:

Using $P_m^S(\lambda)$ obtained in step M-2, compute

$$\frac{\partial M_L(\lambda)}{\partial E_i} \simeq -\lambda \sum_{m=1}^{N_g} p_i^{(m)}(\lambda) P_m^{S(i)}(\lambda).$$

Needless to say, various elementary coding techniques must be used in order for the above steps to work effectively. For instance, the data used many times in a loop (e.g. $e^{-\lambda E_i}$ in step M-1) must be computed outside the loop. If this and other techniques are completely implemented, computation time eventually comes to be dominated by memory accessing, not by computation itself.

It should also be pointed out that this algorithm significantly saves memory space. For instance, the sizes of memory space required to temporarily store the value of $M_{L_S}^{(m)}(\lambda)$, $M_L(\lambda)$, and $P_m^S(\lambda)$ in steps M-1 and M-2 are respectively proportional to $N_g N_S$, N_t, and $N_t N_g N_S$. Remarkably, these numbers do not depend on N and N_I. Thus, this algorithm can handle a large portfolio with many systematic factors by an ordinary personal computer.

5 Numerical Examples

In this section, we examine the performance of the fast numerical approximation (FNA) explained in the preceding sections, for some sample portfolios [17].

[17] All measurement results including computation time in this section are obtained by a parallel program, which is written in C++ using OpenMP and compiled by Intel® C++ Compiler for Windows. We use a personal computer with Windows 2000 OS, Intel® Core™2 Duo CPU 2.66 GHz, and 3.12 GB RAM.

Portfolio	N	Exposure E_i	Rating $r(i)$	Loading scalar β_S
D1	1,000	$E_i = 1/N$	Random	0.5
D2	10,000	$E_i = 1/N$	Random	0.5
D3	1,000,000	$E_i = 1/N$	Random	0.5
C1	1,000	$E_i = c/i$	Random	0.5
C2	10,000	$E_i = c/i$	Random	0.5
C2-(a)	10,000	$E_i = c/i$	$r(i) = 4$ (PD = 0.1%)	0.5
C2-(b)	10,000	$E_i = c/i$	Random	0.1
C2-(c)	10,000	$E_i = c/i$	Random	0.9
C3	1,000,000	$E_i = c/i$	Random	0.5

Table 2: Nine sample portfolios. Constant c is determined so that $\sum_{i=1}^{N} E_i = 1$.

r	1	2	3	4	5	6	7	8	9	10
PD_r	0.01%	0.03%	0.05%	0.1%	0.12%	0.15%	0.2%	0.25%	0.3%	0.4%
r	11	12	13	14	15	16	17	18	19	
PD_r	0.5%	0.65%	0.8%	1.0%	1.2%	1.4%	1.75%	3.5%	10.0%	

Table 3: PD for 19 ratings

5.1 Sample Portfolios and Parameters

We consider nine sample portfolios, as shown in Table 2. The exposure distribution indicates that portfolios D1 to D3 are completely diversified, whereas all other portfolios are severely concentrated according to a power law distribution. We consider a rating system with 19 ratings, and assume their PD to be given by Table 3. In all portfolios except C2-(a), the rating $r(i)$ of obligor i is chosen randomly with $\Pr(r(i) = k) = 1/19$ for all k $(= 1, 2, \ldots, 19)$ [18]. The number of systematic factors and sectors are both set as $N_f = N_S = 33$, which is the number of TOPIX sector indices. Unit loading vectors $\mathbf{e}_S$ are determined so that the correlation matrix

$$\mathrm{corr}(\tilde{Y}^S, \tilde{Y}^{S'}) = \mathbf{e}_S \cdot \mathbf{e}_{S'} \tag{75}$$

is equal to that estimated from 33 TOPIX sector indices [19]. Similarly to the random rating allocation explained above, the sector $S(i)$ of obligor i is chosen randomly with $\Pr(S(i) = k) = 1/33$ for all k $(= 1, 2, \ldots, 33)$. The value of loading scalar β_S is identical throughout the 33 sectors, in all sample portfolios.

We refer to all other parameters as *algorithm parameters*, which determine the performance of the fast numerical approximation but do not affect the risk profile of portfolios. These algorithm parameters are listed in Table 4. First, let us explain how to determine

[18] The rating of the largest exposure in portfolio C2 is intentionally set to be $r(1) = 13$, namely PD = 0.8%.

[19] We use monthly data of TOPIX sector indices from 2002 to 2007.

Algorithm Parameters	
Real part of Bromwich integration path	$\gamma = -\dfrac{\ln 10^{-14}}{4l_{\max}}$
Discretization width	$h = \dfrac{\pi}{2l_{\max}}$
Truncation parameter	$N_t = 100$
Discretization parameter	$N_g = 128$
Number of integration points	$N_I = 2.5 \times 10^6$
Set of cdf valuation points	$S_L = \left\{ \dfrac{k}{1000} l_{\max} \;\middle|\; k = 1, 2, \ldots, 1000 \right\}$

Table 4: Basic setting of the algorithm parameters

γ and h. Recall the upper bound formula (37) for the discretization error $e_d(l)$. If γ and h are chosen as in Table 4, $e_d(l)$ satisfies

$$|e_d(l)| \lesssim 10^{-14} \quad (0 \leq l \leq 4l_{\max}), \tag{76}$$

where $l_{\max}$ is the maximum loss level where the cdf is computed. Therefore, the discretization error is sufficiently small in S_L, if S_L is defined as in Table 4 [20]. Next, we have to determine $l_{\max}$. If $l_{\max}$ is too small, we can not compute quantiles larger than $l_{\max}$. Conversely, if $l_{\max}$ is too large, the numerical approximation worsens because the cdf valuation points in S_L become sparse. To avoid these problems, we set $l_{\max}$ to be 1.1 times the 99.9999% VaR computed in advance using a parameter setting

$$N_t = 25,\ N_g = 32,\ N_I = 1 \times 10^5,\ l_{\max} = \sum_{i=1}^{N} E_i = 1. \tag{77}$$

This preliminary computation can be done in a relatively short time. By setting $l_{\max}$ in this way, we can compute VaR at confidence levels lower than 99.9999%.

The remaining algorithm parameters are set as follows. Since N_f is very large, we adopt Monte Carlo integration to compute the expectation values over systematic factors. We use $N_I = 2.5 \times 10^6$ integration points $\mathbf{x}_k$ generated by standard normal random numbers. For most of the analyses given below, the seed of these random numbers is fixed at a specific value. However, Monte Carlo integration involves statistical error. This statistical error is evaluated for portfolios C2, C2-(b) and C2-(c), by comparing risk measures computed by different seeds. The truncation and discretization parameters are set as $N_t = 100$ and $N_g = 128$. We mainly use the algorithm parameters given by Table 4 in the following examples, unless otherwise stated.

[20] Although $|e_d(l)| \lesssim 10^{-14}$ is satisfied also in $[l_{\max}, 4l_{\max}]$, we restrict S_L to the interval $[0, l_{\max}]$, since the authors have empirically observed that the truncation error $e_t(l)$ increases for $l \sim 4l_{\max}$.

5.2 Risk Measures and Tail Probability

In this subsection, we examine the performance of the fast numerical approximation by computing risk measures for the nine sample portfolios. Tail loss probabilities are also exhibited to show that the fast numerical approximation works quite well for a wide range of loss levels. All Monte Carlo simulations used as benchmarks in this subsection are performed with 10 million loss scenarios.

We first present VaR and CVaR measurement results obtained by the fast numerical approximation and Monte Carlo simulation in Table 5. Most of the relative errors are smaller than 1%. However, for portfolio C2-(a), the 99.9% VaR of the fast numerical approximation is 0.102, larger than that of Monte Carlo simulation by more than 10%. Which result is correct? To answer this question, note that all obligors have PD=0.1% in this portfolio. Therefore, 99.9% VaR can not be less than the largest exposure $E_1 \simeq 0.10$. This indicates that the result of the fast numerical approximation is closer to the true value. We will come back to this point soon later.

In Figures 3-5, we depict the tail loss probabilities of diversified portfolios D1 to D3. The two methods are almost indistinguishable for loss levels smaller than the 99.99 percentile point. The reason for the discrepancy in far tail region is explained as follows. In a well diversified portfolio, portfolio loss is almost uniquely determined by the systematic factors, due to the law of large numbers. This means that the loss can approximately take only N_I discrete values, in the fast numerical approximation with N_I integration points. Therefore, loss distribution becomes very sparse in far tail region. To obtain more accurate tail probability, we only have to increase N_I. As an example, we compute the tail probability of portfolio D3 for $N_I = 2.5 \times 10^7$ case with all other algorithm parameters unchanged. Figure 5 shows that the accuracy of the fast numerical approximation is improved, as expected.

The tail loss probabilities of concentrated portfolios C1, C2, C3, and C2-(a) are shown in Figures 6-9. The two methods are almost indistinguishable everywhere, in contrast with diversified portfolios. It is remarkable that the graph of C2 has a small flat region near the tail probability $\sim 1\%$. In other words, the loss has a very small probability of taking values slightly lower than 0.1. This is due to the largest exposure with $E_1 \simeq 0.10$ and PD=0.8%. In general, it is difficult to accurately compute quantiles near such region. However, as we have already seen in Table 5, the 99% VaR obtained by the fast numerical approximation is sufficiently accurate. The tail probability of portfolio C2-(a) is highly skewed, since all obligors have PD=0.1% in this portfolio. In particular, there exists a wide stationary region near 99.9% VaR due to the largest exposure. This is the reason why 99.9% VaR is underestimated by Monte Carlo simulation due to statistical error.

We stress that most of the tail probabilities are monotonically decreasing in the fast numerical approximation. This means that the trigonometric series (38) can be sufficiently smoothed by the QD algorithm. However, the tail probabilities of portfolios D3 with $N_I = 2.5 \times 10^6$ and C2-(a) are *not* monotonic. In portfolio D3, the tail probability slightly oscillates in far tail region, whereas in portfolio C2-(a) the tail probability is

slightly increasing near 99.9% VaR. In principle, non-monotonicity can be suppressed by increasing N_t in general cases, or increasing N_I especially for diversified portfolios. However, the non-monotonicity in the present cases causes no serious problem in practice, since it is almost invisible as the figure indicates.

Next, we compute risk measures obtained by ten different seeds of numerical integration, for portfolios C2, C2-(b) and C2-(c). The results are summarized in Tables 6-8. It is found that the statistical errors increase as β_S increases. This is because the systematic risk arising from the uncertainty of systematic factors is roughly proportional to β_S. However, the statistical errors are smaller than 1%, even for the strong correlation case $\beta_S = 0.9$.

The stability of the fast numerical approximation under the change of N_t and N_g is examined as follows. Table 9 presents the risk measurement results for various N_t, with all other algorithm parameters set as in Table 4. We regard $N_t = 200$ cases as benchmarks, and compute the relative errors of $N_t = 50$ and $N_t = 100$ cases. All relative errors are smaller than 1%, indicating that continued fraction expansion is suficiently convergent. Table 10 presents the risk measurement results for various N_g, with all other algorithm parameters set as in Table 4. Similarly to Table 9, $N_g = 64$ and $N_g = 128$ cases are compared with $N_g = 256$ cases as benchmarks. All relative errors are smaller than 1% again, indicating that discretized valuation points are appropriately chosen.

Portfolio	VaR			CVaR		
	99%	99.9%	99.97%	99%	99.9%	99.97%
Fast Numerical Approximation						
D1	0.055 (0.38%)	0.090 (1.17%)	0.110 (0.64%)	0.070 (0.43%)	0.107 (0.14%)	0.127 (−0.50%)
D2	0.054 (0.18%)	0.088 (0.38%)	0.107 (−0.08%)	0.068 (0.22%)	0.104 (−0.15%)	0.124 (−0.87%)
D3	0.053 (0.32%)	0.087 (0.31%)	0.106 (−0.22%)	0.067 (0.30%)	0.103 (−0.21%)	0.122 (−0.78%)
C1	0.068 (0.13%)	0.114 (0.18%)	0.140 (−0.42%)	0.088 (0.10%)	0.135 (−0.19%)	0.163 (−0.48%)
C2	0.081 (0.25%)	0.148 (−0.06%)	0.170 (−0.43%)	0.120 (−0.12%)	0.166 (−0.24%)	0.188 (−0.35%)
C2-(a)	0.013 (0.53%)	0.102 (11.29%)	0.105 (0.15%)	0.034 (0.10%)	0.106 (0.08%)	0.111 (0.14%)
C3	0.067 (0.22%)	0.105 (0.02%)	0.127 (−0.22%)	0.083 (0.17%)	0.123 (−0.27%)	0.146 (−0.48%)
Monte Carlo (10 million loss scenarios)						
D1	0.055	0.089	0.109	0.070	0.106	0.127
D2	0.054	0.087	0.107	0.068	0.104	0.125
D3	0.053	0.087	0.106	0.067	0.103	0.123
C1	0.068	0.113	0.140	0.088	0.136	0.163
C2	0.081	0.148	0.170	0.120	0.166	0.189
C2-(a)	0.013	0.092	0.105	0.034	0.106	0.111
C3	0.067	0.105	0.127	0.083	0.124	0.146

Table 5: Risk measurement results of the fast numerical approximation and Monte Carlo simulation. The numerical errors of the fast numerical approximation relative to Monte Carlo simulation are shown in parentheses.

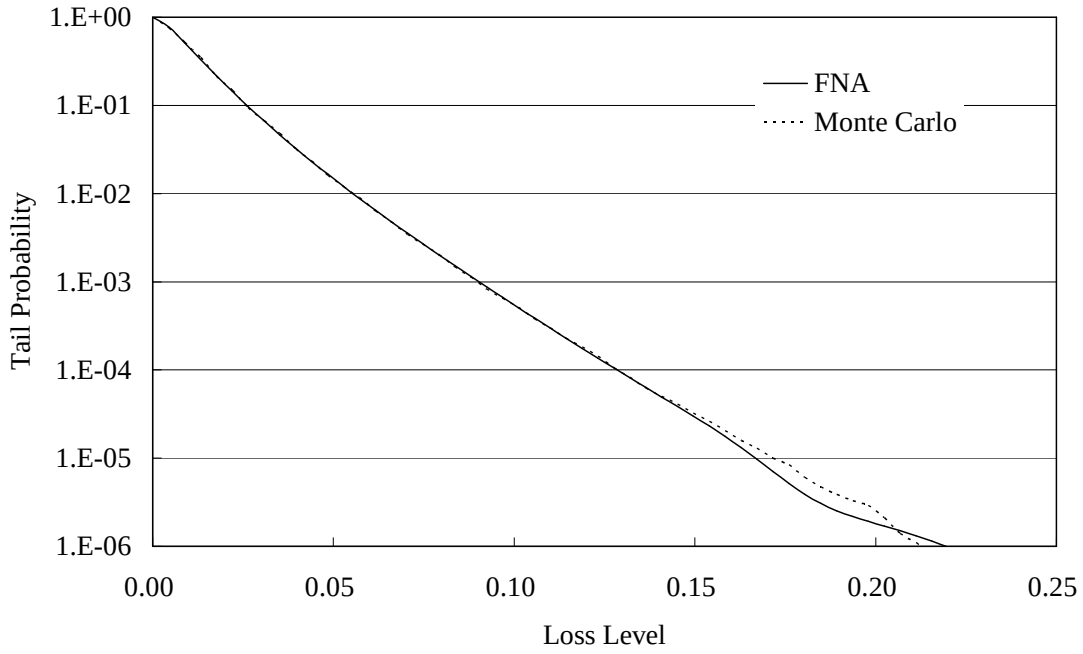

Figure 3: Tail loss probability of portfolio D1.

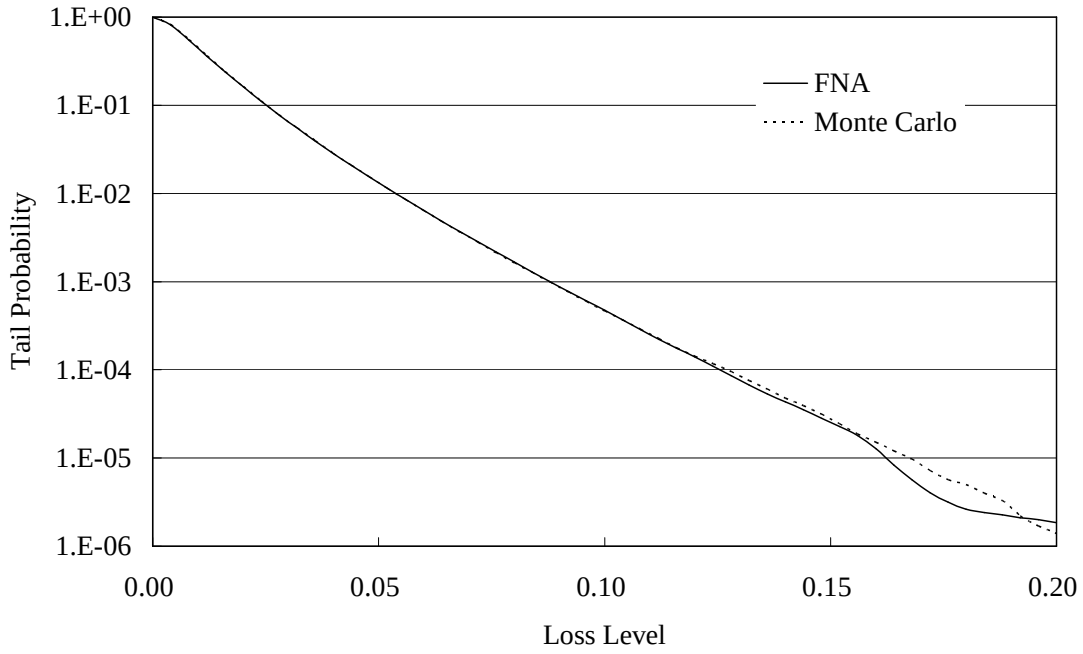

Figure 4: Tail loss probability of portfolio D2.

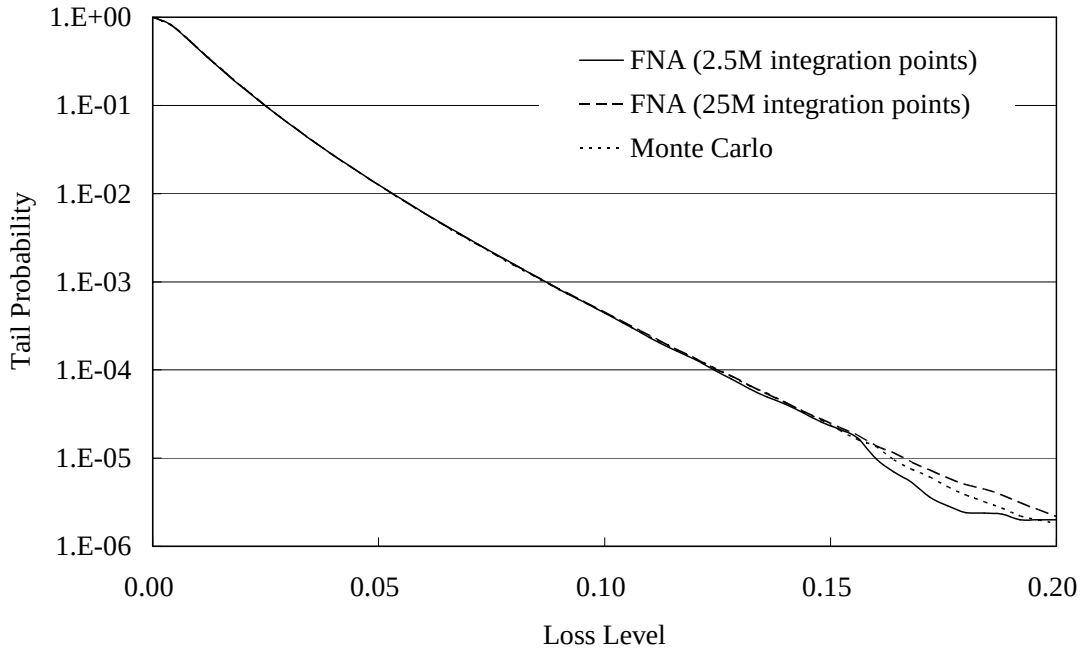

Figure 5: Tail loss probability of portfolio D3.

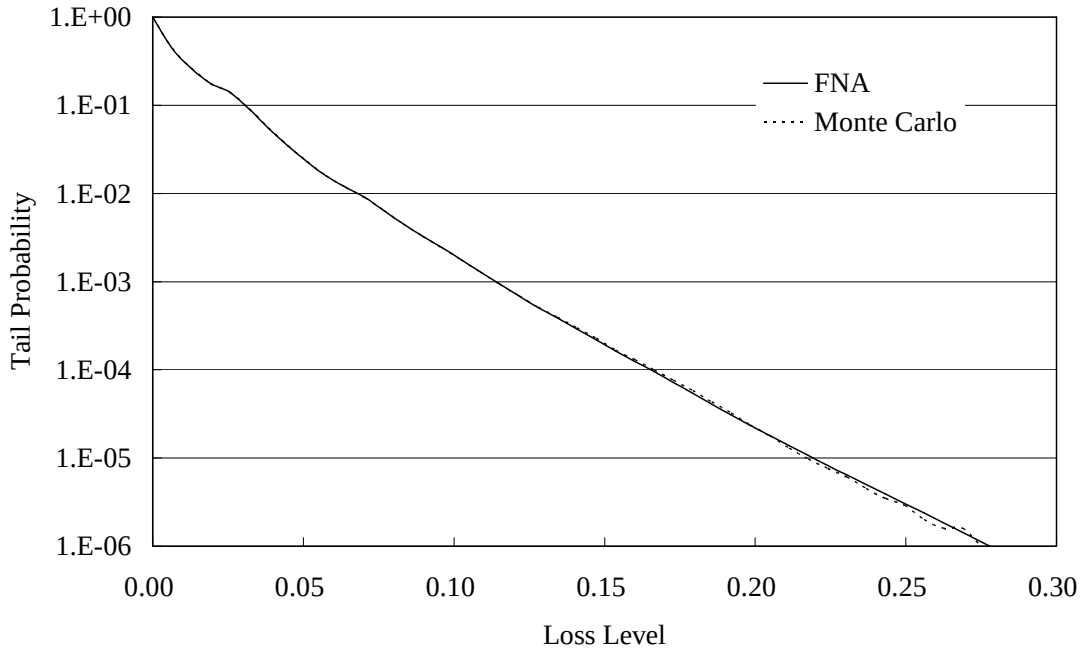

Figure 6: Tail loss probability of portfolio C1.

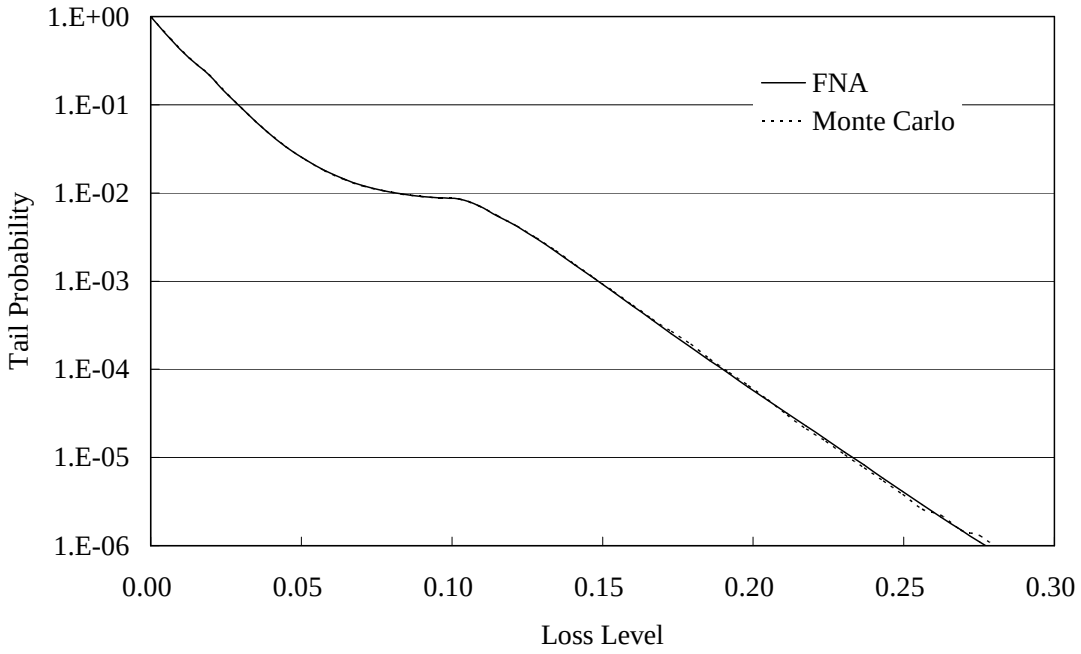

Figure 7: Tail loss probability of portfolio C2. A small flat region near 99% VaR is due to the largest exposure with $E_1 \simeq 0.10$ and PD=0.8%.

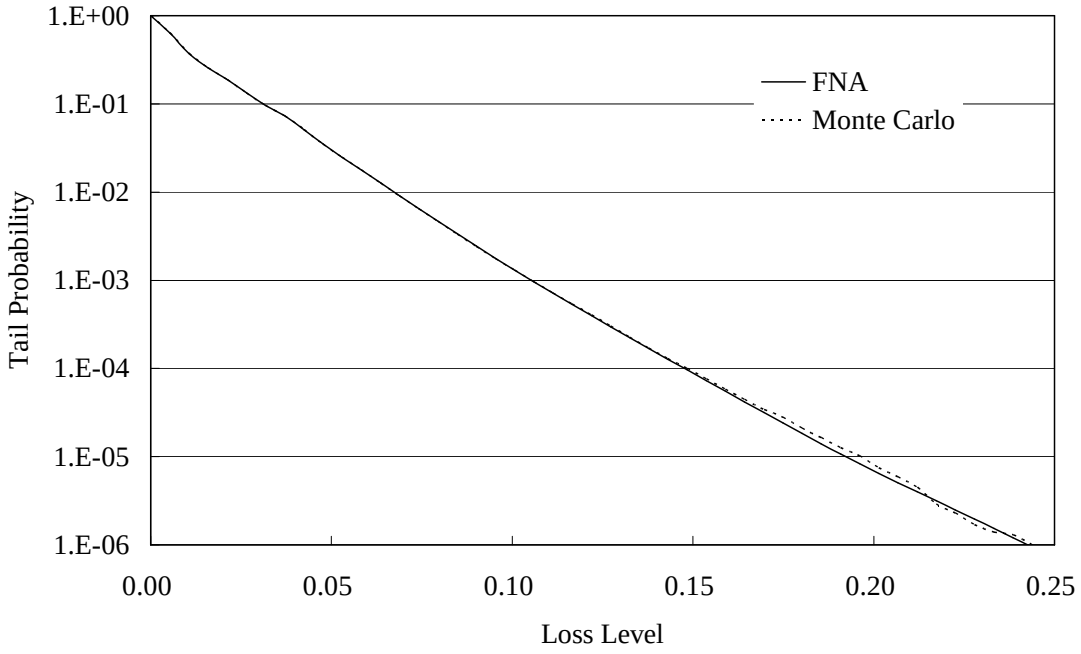

Figure 8: Tail loss probability of portfolio C3.

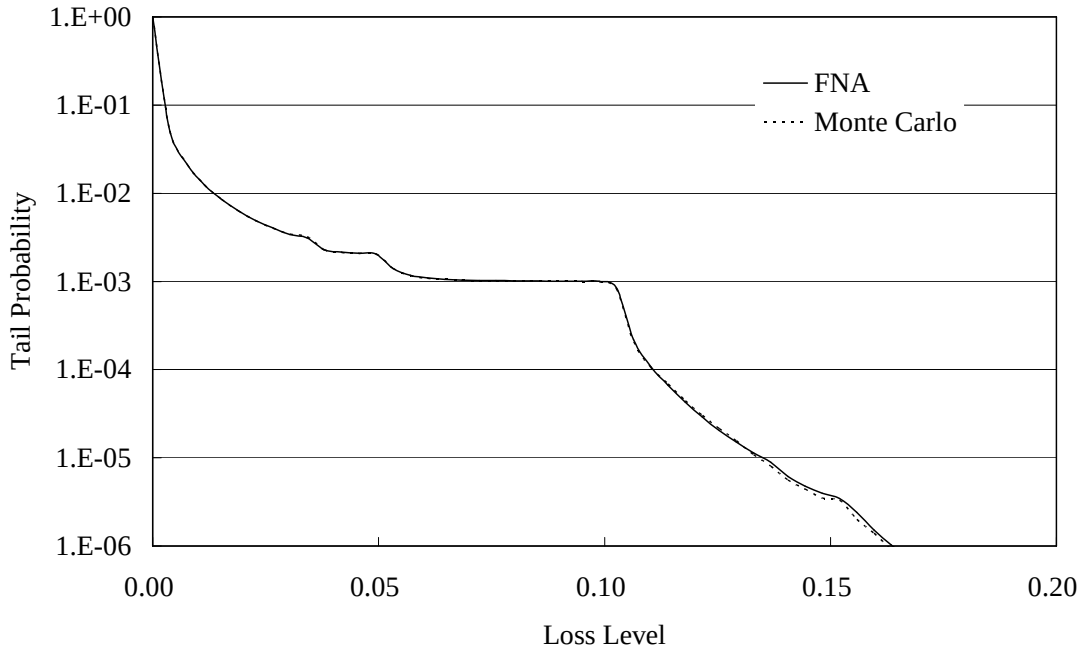

Figure 9: Tail loss probability of portfolio C2-(a). A flat region near 99.9% VaR is due to the largest exposure with $E_1 \simeq 0.10$.

	VaR			CVaR		
Measurement #	99%	99.9%	99.97%	99%	99.9%	99.97%
1	0.0563	0.1245	0.1317	0.1042	0.1308	0.1388
2	0.0563	0.1245	0.1317	0.1042	0.1308	0.1388
3	0.0563	0.1245	0.1317	0.1042	0.1308	0.1388
4	0.0563	0.1245	0.1317	0.1042	0.1308	0.1388
5	0.0563	0.1245	0.1317	0.1042	0.1308	0.1388
6	0.0563	0.1245	0.1317	0.1042	0.1308	0.1388
7	0.0563	0.1245	0.1317	0.1042	0.1308	0.1388
8	0.0563	0.1245	0.1316	0.1042	0.1308	0.1388
9	0.0563	0.1245	0.1317	0.1042	0.1308	0.1388
10	0.0563	0.1245	0.1317	0.1042	0.1308	0.1388
Mean	0.0563	0.1245	0.1317	0.1042	0.1308	0.1388
StDev	1.775E-6	4.393E-6	7.148E-6	2.185E-6	2.151E-6	4.875E-6
StDev/Mean	0.003%	0.004%	0.005%	0.002%	0.002%	0.004%

Table 6: Risk measurement results of the fast numerical approximation for portfolio C2-(b) with $\beta_S = 0.1$, obtained by ten different seeds of numerical integration.

	VaR			CVaR		
Measurement #	99%	99.9%	99.97%	99%	99.9%	99.97%
1	0.0802	0.1478	0.1691	0.1198	0.1656	0.1871
2	0.0804	0.1479	0.1693	0.1199	0.1658	0.1875
3	0.0806	0.1481	0.1696	0.1200	0.1661	0.1881
4	0.0807	0.1480	0.1694	0.1201	0.1659	0.1877
5	0.0804	0.1479	0.1692	0.1199	0.1657	0.1874
6	0.0805	0.1480	0.1694	0.1201	0.1660	0.1879
7	0.0806	0.1480	0.1694	0.1201	0.1659	0.1878
8	0.0807	0.1480	0.1696	0.1201	0.1660	0.1879
9	0.0805	0.1479	0.1694	0.1200	0.1659	0.1879
10	0.0806	0.1480	0.1695	0.1201	0.1661	0.1881
Mean	0.0805	0.1480	0.1694	0.1200	0.1659	0.1877
StDev	0.0001	0.0001	0.0002	0.0001	0.0002	0.0003
StDev/Mean	0.17%	0.05%	0.09%	0.07%	0.10%	0.17%

Table 7: Risk measurement results of the fast numerical approximation for portfolio C2 with $\beta_S = 0.5$, obtained by ten different seeds of numerical integration.

	VaR			CVaR		
Measurement #	99%	99.9%	99.97%	99%	99.9%	99.97%
1	0.1441	0.2869	0.3694	0.2054	0.3547	0.4360
2	0.1441	0.2869	0.3673	0.2055	0.3537	0.4354
3	0.1445	0.2886	0.3716	0.2064	0.3575	0.4410
4	0.1440	0.2874	0.3709	0.2056	0.3559	0.4379
5	0.1441	0.2895	0.3750	0.2063	0.3589	0.4406
6	0.1441	0.2864	0.3678	0.2052	0.3528	0.4320
7	0.1441	0.2866	0.3685	0.2052	0.3548	0.4372
8	0.1441	0.2882	0.3719	0.2057	0.3566	0.4383
9	0.1442	0.2879	0.3707	0.2058	0.3556	0.4368
10	0.1443	0.2849	0.3661	0.2050	0.3522	0.4339
Mean	0.1442	0.2873	0.3699	0.2056	0.3553	0.4369
StDev	0.0001	0.0013	0.0026	0.0004	0.0021	0.0028
StDev/Mean	0.10%	0.45%	0.71%	0.22%	0.59%	0.63%

Table 8: Risk measurement results of the fast numerical approximation for portfolio C2-(c) with $\beta_S = 0.9$, obtained by ten different seeds of numerical integration.

		VaR			CVaR		
Portfolio	N_t	99%	99.9%	99.97%	99%	99.9%	99.97%
D1	50	0.055 (0.00%)	0.090 (0.00%)	0.110 (0.00%)	0.070 (0.00%)	0.107 (0.00%)	0.127 (0.00%)
	100	0.055 (0.00%)	0.090 (0.00%)	0.110 (0.00%)	0.070 (0.00%)	0.107 (0.00%)	0.127 (0.00%)
	200	0.055	0.090	0.110	0.070	0.107	0.127
D2	50	0.054 (0.00%)	0.088 (−0.02%)	0.107 (0.01%)	0.068 (0.00%)	0.104 (0.00%)	0.124 (0.00%)
	100	0.054 (0.00%)	0.088 (0.00%)	0.107 (0.00%)	0.068 (0.00%)	0.104 (0.00%)	0.124 (0.00%)
	200	0.054	0.088	0.107	0.068	0.104	0.124
D3	50	0.053 (0.02%)	0.087 (0.03%)	0.106 (0.03%)	0.067 (0.00%)	0.103 (0.00%)	0.122 (−0.01%)
	100	0.053 (0.01%)	0.087 (0.04%)	0.106 (−0.09%)	0.067 (0.00%)	0.103 (0.00%)	0.122 (0.00%)
	200	0.053	0.087	0.106	0.067	0.103	0.122
C1	50	0.068 (−0.12%)	0.114 (0.00%)	0.140 (0.01%)	0.088 (0.00%)	0.135 (0.00%)	0.163 (0.00%)
	100	0.068 (−0.02%)	0.114 (0.00%)	0.140 (0.00%)	0.088 (0.00%)	0.135 (0.00%)	0.163 (0.00%)
	200	0.068	0.114	0.140	0.088	0.135	0.163
C2	50	0.081 (−0.01%)	0.148 (0.00%)	0.170 (0.00%)	0.120 (0.00%)	0.166 (0.00%)	0.188 (0.00%)
	100	0.081 (0.00%)	0.148 (0.00%)	0.170 (0.00%)	0.120 (0.00%)	0.166 (0.00%)	0.188 (0.00%)
	200	0.081	0.148	0.170	0.120	0.166	0.188
C2-(a)	50	0.013 (−0.32%)	0.102 (−0.09%)	0.105 (0.07%)	0.034 (0.00%)	0.106 (−0.02%)	0.111 (0.01%)
	100	0.013 (0.09%)	0.102 (−0.28%)	0.105 (0.01%)	0.034 (−0.01%)	0.106 (0.00%)	0.111 (0.00%)
	200	0.013	0.102	0.105	0.034	0.106	0.111
C3	50	0.067 (−0.00%)	0.105 (0.00%)	0.127 (0.00%)	0.083 (0.00%)	0.123 (0.00%)	0.146 (0.00%)
	100	0.067 (0.00%)	0.105 (0.00%)	0.127 (0.00%)	0.083 (0.00%)	0.123 (0.00%)	0.146 (0.00%)
	200	0.067	0.105	0.127	0.083	0.123	0.146

Table 9: Risk measurement results for various N_t. For $N_t = 50$ and $N_t = 100$, the numerical errors relative to $N_t = 200$ cases are shown in parentheses.

		VaR			CVaR		
Portfolio	N_g	99%	99.9%	99.97%	99%	99.9%	99.97%
D1	64	0.055 (0.07%)	0.090 (0.06%)	0.110 (0.02%)	0.070 (0.06%)	0.107 (0.05%)	0.127 (0.05%)
	128	0.055 (0.02%)	0.090 (0.02%)	0.110 (0.02%)	0.070 (0.02%)	0.107 (0.02%)	0.127 (0.02%)
	256	0.055	0.090	0.110	0.070	0.106	0.127
D2	64	0.054 (0.07%)	0.088 (0.07%)	0.107 (−0.02%)	0.068 (0.06%)	0.104 (0.05%)	0.124 (0.06%)
	128	0.054 (0.02%)	0.088 (0.02%)	0.107 (0.03%)	0.068 (0.02%)	0.104 (0.02%)	0.124 (0.03%)
	256	0.054	0.088	0.107	0.068	0.104	0.123
D3	64	0.053 (0.08%)	0.087 (0.07%)	0.106 (−0.08%)	0.067 (0.07%)	0.103 (0.05%)	0.122 (0.06%)
	128	0.053 (0.02%)	0.087 (0.03%)	0.106 (−0.06%)	0.067 (0.02%)	0.103 (0.03%)	0.122 (0.03%)
	256	0.053	0.087	0.106	0.067	0.103	0.122
C1	64	0.068 (0.08%)	0.114 (0.05%)	0.140 (0.05%)	0.088 (0.06%)	0.135 (0.05%)	0.163 (0.05%)
	128	0.068 (0.04%)	0.114 (0.02%)	0.140 (0.02%)	0.088 (0.02%)	0.135 (0.02%)	0.163 (0.02%)
	256	0.068	0.114	0.140	0.088	0.135	0.163
C2	64	0.081 (0.18%)	0.148 (0.04%)	0.170 (0.04%)	0.120 (0.07%)	0.166 (0.04%)	0.188 (0.03%)
	128	0.081 (0.04%)	0.148 (0.01%)	0.170 (0.01%)	0.120 (0.01%)	0.166 (0.01%)	0.188 (0.01%)
	256	0.081	0.148	0.170	0.120	0.166	0.188
C2-(a)	64	0.013 (0.20%)	0.102 (0.06%)	0.105 (0.02%)	0.034 (0.09%)	0.106 (0.01%)	0.111 (0.03%)
	128	0.013 (0.03%)	0.102 (0.03%)	0.105 (0.00%)	0.034 (0.00%)	0.106 (0.00%)	0.111 (0.00%)
	256	0.013	0.102	0.105	0.034	0.106	0.111
C3	64	0.067 (0.06%)	0.105 (0.05%)	0.127 (0.05%)	0.083 (0.05%)	0.123 (0.05%)	0.146 (0.05%)
	128	0.067 (0.01%)	0.105 (0.02%)	0.127 (0.02%)	0.083 (0.01%)	0.123 (0.02%)	0.146 (0.02%)
	256	0.067	0.105	0.127	0.083	0.123	0.146

Table 10: Risk measurement results for various N_g. For $N_g = 64$ and $N_g = 128$, the numerical errors relative to $N_g = 256$ cases are shown in parentheses.

Portfolio	N	$N_t = 50$	$N_t = 100$
C1	1,000	20sec	40sec
C2	10,000	21sec	42sec
C3	1,000,000	1min43sec	3min45sec

Table 11: Computation time of whole distribution function in the fast numerical approximation.

Finally, we examine the computational speed of the fast numerical approximation. In Table 11, the computation time for $N_t = 50$ and $N_t = 100$ cases are shown (other algorithm parameters are unchanged from Table 4). The computation of $P_m^S(\lambda)$ in step M-2 is omitted here, but Table 11 includes the time to obtain the values of the cdf at 1,000 loss levels in S_L [21]. Remarkably, the dependence of computation time on portfolio size is quite moderate. As a consequence, it takes only a few minutes even for a portfolio with 1 million obligors. This is in great contrast with Monte Carlo simulation, which requires computation time roughly proportional to portfolio size.

5.3 Risk Contributions

We first explain how to examine the accuracy of the risk contributions computed by the fast numerical approximation. Since the r.h.s. of (33) is the expectation value of $\tilde{D}_i$ conditional on $\tilde{L} \geq \mathrm{VaR}(\alpha)$, it can be computed by Monte Carlo simulation with a large number of scenarios. Therefore, we can test the risk contributions to CVaR by comparing (31) with (33). On the other hand, the risk contributions to VaR are indirectly tested by the following rationale. Since the risk contributions to CVaR have already been tested by the above comparison, the accuracy of RC_i^{Δ} given by (32) is guaranteed. Thus, by comparing (30) with (32), we can indirectly test the accuracy of RC_i.

In Figures 10-12, we plot the relative errors of $\mathrm{RC}_i^{\mathrm{CVaR}(\alpha)}$ between the fast numerical approximation and Monte Carlo simulation, for all obligors in portfolio C2. Here, the relative error is defined by

$$\frac{\mathrm{RC}_i^{\mathrm{CVaR}(\alpha)}(\text{Monte Carlo}) - \mathrm{RC}_i^{\mathrm{CVaR}(\alpha)}(\text{Numerical Algorithm})}{\mathrm{RC}_i^{\mathrm{CVaR}(\alpha)}(\text{Numerical Algorithm})},$$

where two sets of Monte Carlo simulation are performed with 100 million and 1 billion loss scenarios, respectively. The horizontal axis in Figures 10-12 represents the conditional PD of obligor i given by

$$\frac{\partial \mathrm{CVaR}(\alpha)}{\partial E_i} = \mathbf{E}\left[\tilde{D}_i \,\middle|\, \tilde{L} \geq \mathrm{VaR}(\alpha)\right].$$

Table 12 shows the standard deviation of the relative error for all obligors. Two observations immediately follow from these figures and table. First, the relative error uniformly

[21] On the other hand, Table 11 does not include the preliminary computation time of $l_{\max}$, which is about 10 seconds in portfolio C3, and less than 1 second in C1 and C2.

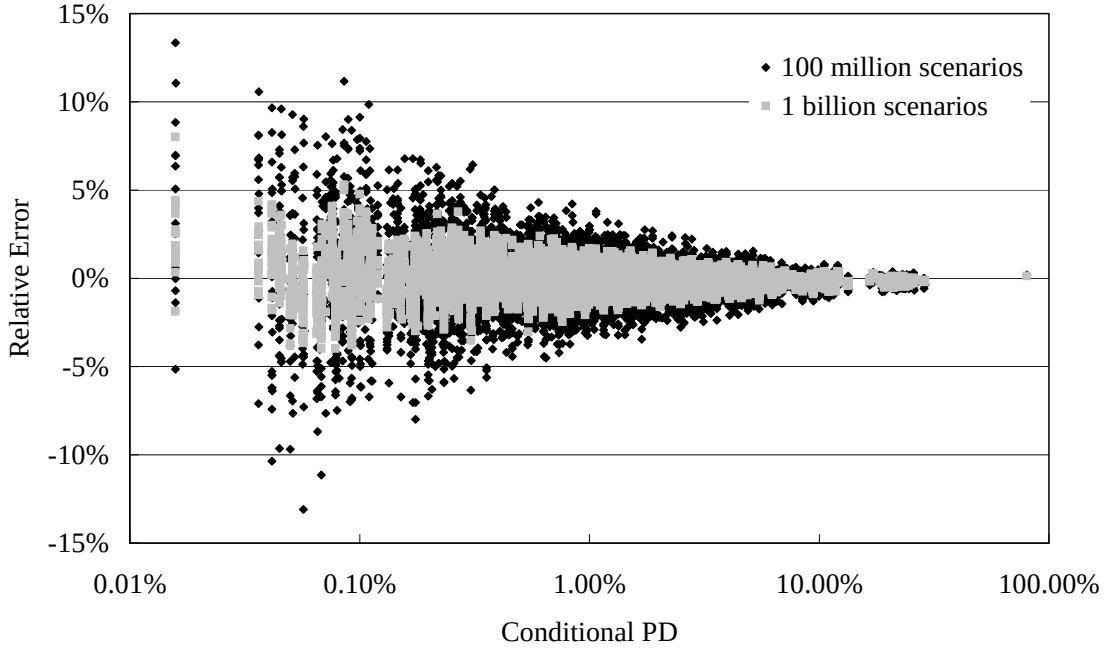

Figure 10: Relative error of risk contributions to 99% CVaR between the fast numerical approximation and Monte Carlo simulation for portfolio C2.

decreases regardless of the conditional PD, as the number of loss scenarios increases. Second, the relative error decreases as the conditional PD increases. Hence, we conclude that the relative error is mainly caused by the statistical error in Monte Carlo simulation, and the results of the fast numerical approximation are not biased. Figures 13-15 show the largest 100 risk contributions to CVaR obtained by the fast numerical approximation and Monte Carlo simulation with 100 million loss scenarios. The two methods give almost indistinguishable results.

Table 13 shows the relative discrepancy between RC_i and RC_i^{Δ} for portfolio C2 defined by

$$(\mathrm{RC}_i^{\Delta} - \mathrm{RC}_i)/\mathrm{RC}_i,$$

with the statistics (maximum, minimum and standard deviation) taken over all obligors except the largest one [22]. We set $\Delta = \mathrm{VaR}(\alpha)/1000$ for all confidence levels. All statistics of the relative discrepancy are smaller than 1%, indirectly assuring the accuracy of RC_i.

Similarly to the last subsection, we evaluate the statistical error of risk contributions due to numerical integration. To do this, we first compute risk contributions ten times for all obligors in portfolio C2, changing only the seed of numerical integration. Next, the average (Ave) and standard deviation (StDev) of ten results are computed for each obligor. The results are summarized in Table 14 and Figures 16-21. Table 14 shows the relative statistical error defined by StDev/Ave, with the mean and maximum taken over

[22] We exclude the largest exposure here, because its risk contribution to 99% VaR trivially vanishes.

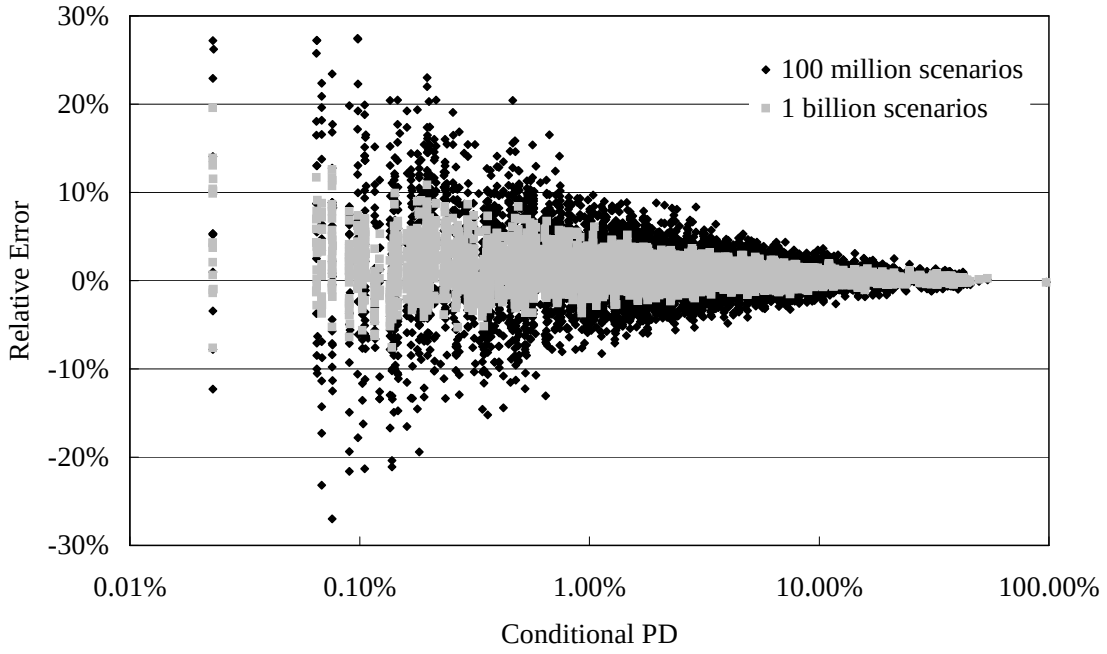

Figure 11: Relative error of risk contributions to 99.9% CVaR between the fast numerical approximation and Monte Carlo simulation for portfolio C2.

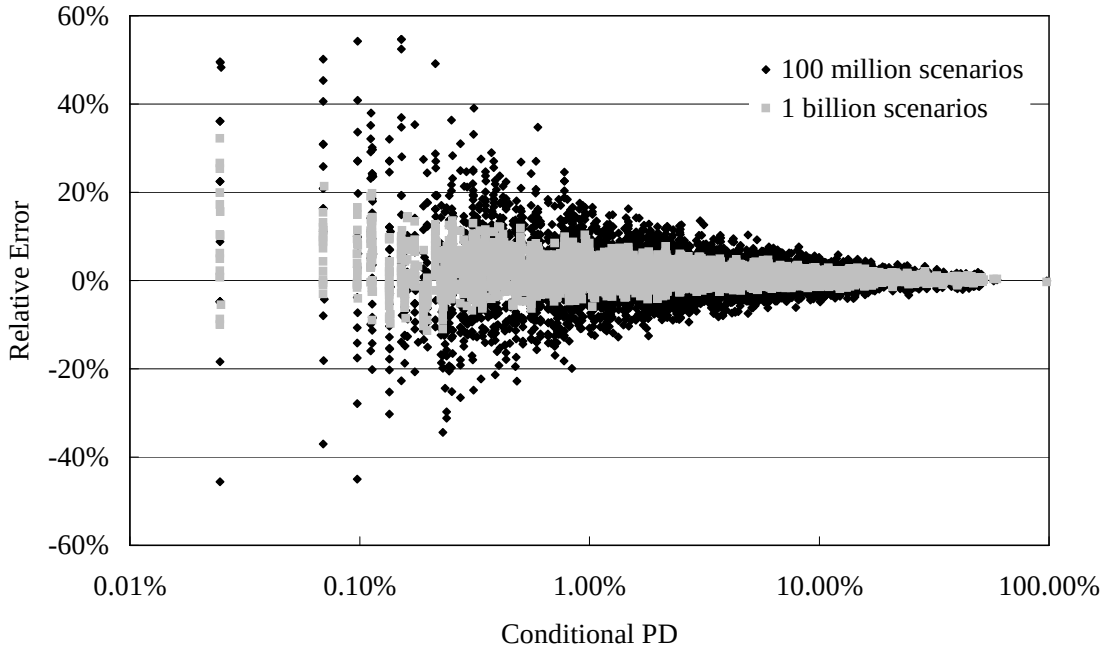

Figure 12: Relative error of risk contributions to 99.97% CVaR between the fast numerical approximation and Monte Carlo simulation for portfolio C2.

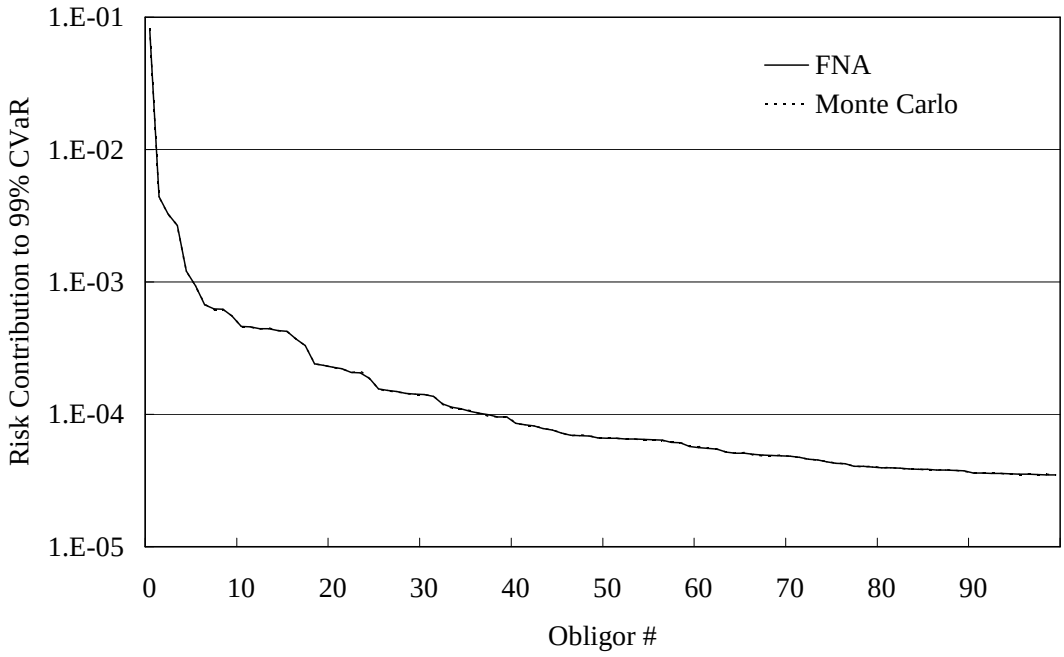

Figure 13: The largest 100 risk contributions to 99% CVaR for portfolio C2.

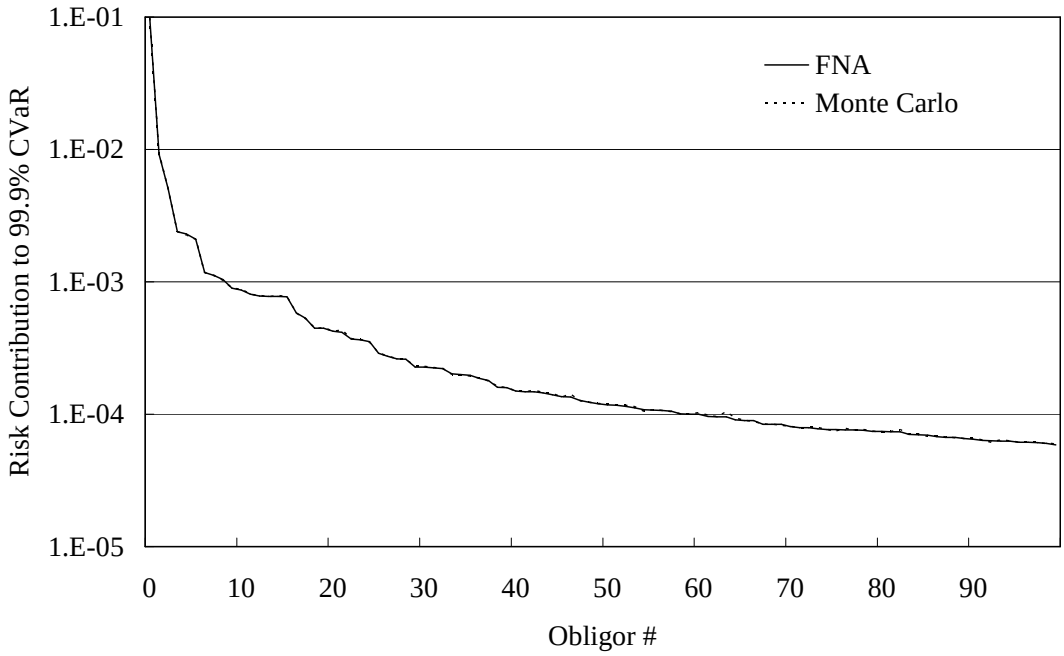

Figure 14: The largest 100 risk contributions to 99.9% CVaR for portfolio C2.

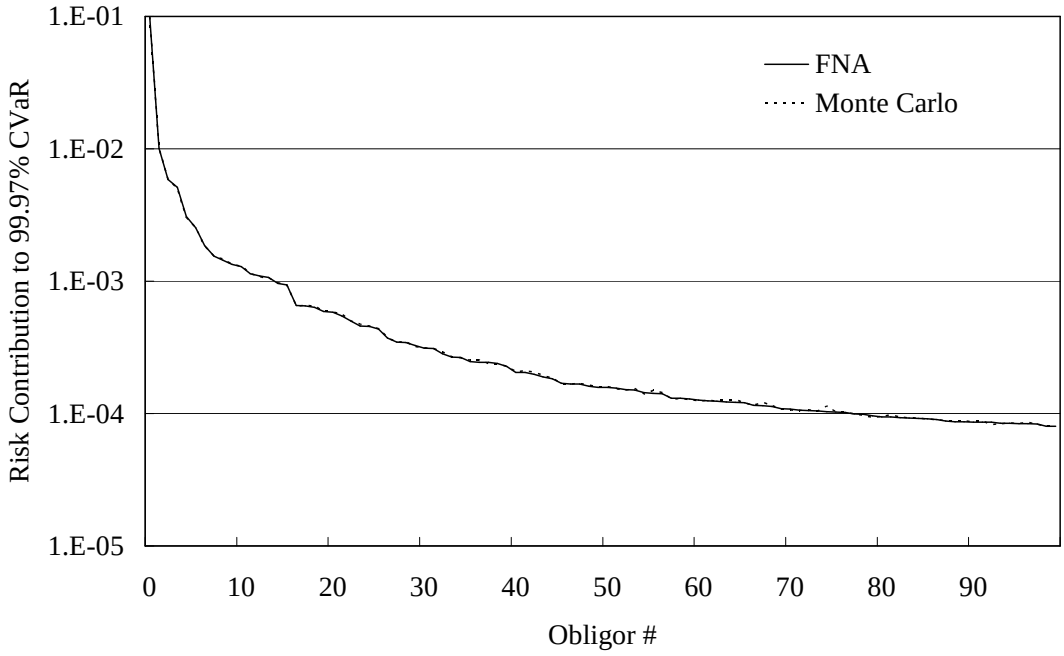

Figure 15: The largest 100 risk contributions to 99.97% CVaR for portfolio C2.

	CVaR		
	99%	99.9%	99.97%
100 million scenarios	1.55%	3.58%	5.41%
1 billion scenarios	0.81%	1.47%	2.17%

Table 12: Comparison of the risk contributions to CVaR between the fast numerical approximation and Monte Carlo simulation for portfolio C2.

	VaR		
	99%	99.9%	99.97%
Max	0.124%	0.600%	0.031%
Min	−0.115%	−0.810%	−0.039%
StDev	0.004%	0.015%	0.002%

Table 13: Statistics of the relative discrepancy between RC_i and RC_i^{Δ} for portfolio C2.

	VaR			CVaR		
	99%	99.9%	99.97%	99%	99.9%	99.97%
Mean	0.76%	0.77%	1.09%	0.66%	1.20%	1.85%
Max	3.56%	2.71%	3.67%	2.13%	4.19%	6.69%

Table 14: Mean and maximum of the relative statistical error of risk contributions for portfolio C2.

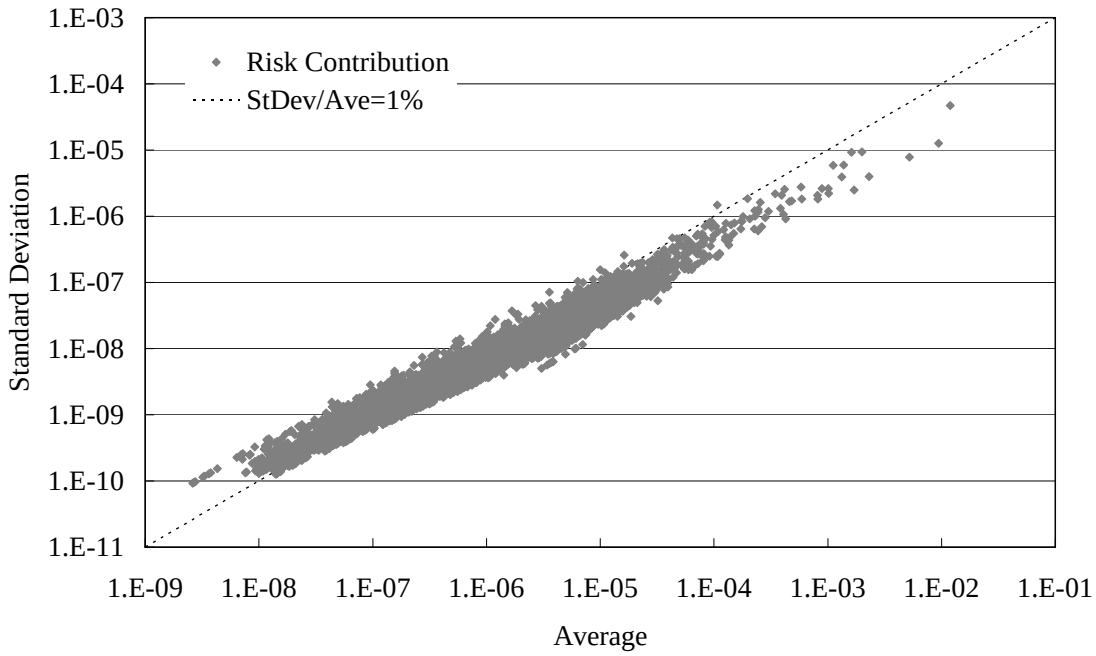

Figure 16: Average and standard deviation of risk contributions to 99% VaR for portfolio C2.

all obligors except the largest one. The relative statistical error gradually increases as the confidence level increases. However, as shown in Figures 16-21, the relative statistical error is smaller than 1%, for most of the obligors with relatively large risk contribution. This is because the risk contributions of these obligors are explained mainly by name concentration effect, not by sector concentration effect.

Next, we analyze the stability of risk contributions under the change of N_t and N_g. This is similar to the stability analysis of risk measures performed previously. The upper (lower) half of Table 15 shows the errors of the risk contributions relative to $N_t = 200$ ($N_g = 256$) cases. The maximum and minimum are taken over all obligors, and all algorithm parameters except N_t and N_g are set as in Table 4. All relative errors are smaller than 1%, indicating that continued fraction expansion and discretized valuation points work quite well.

Finally, we examine the computational speed of the fast numerical approximation in

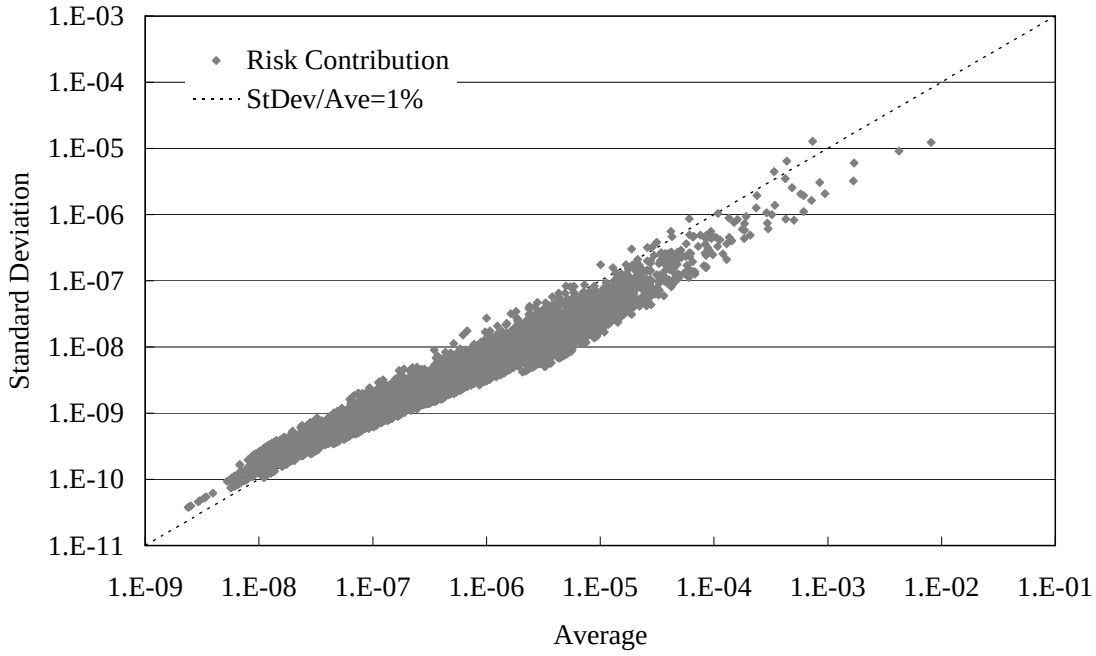

Figure 17: Average and standard deviation of risk contributions to 99.9% VaR for portfolio C2.

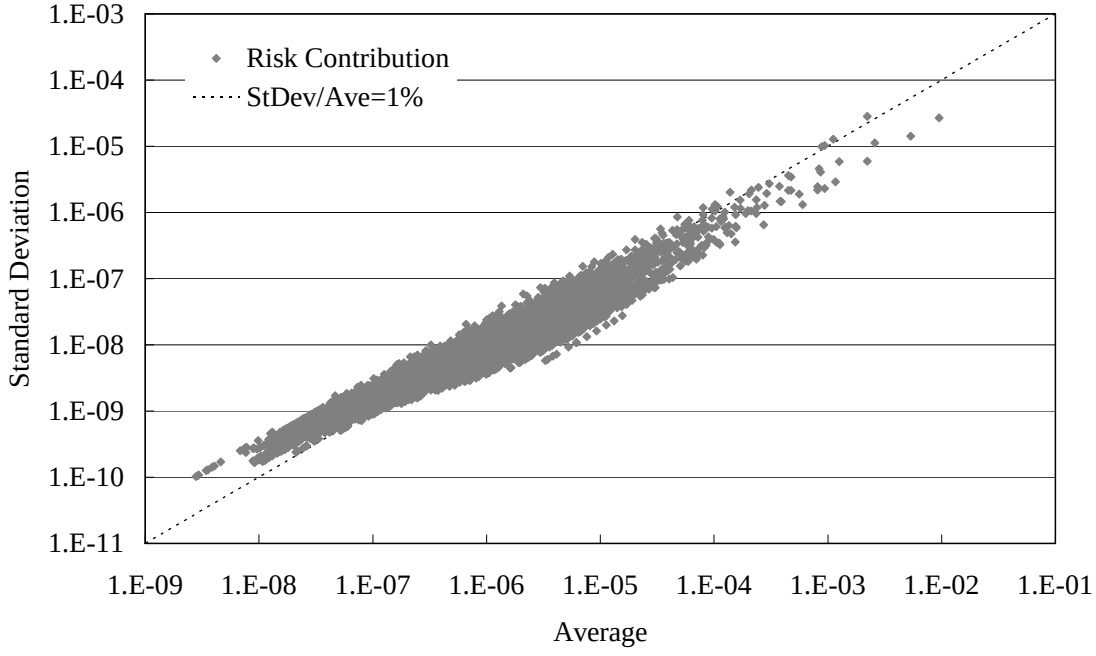

Figure 18: Average and standard deviation of risk contributions to 99.97% VaR for portfolio C2.

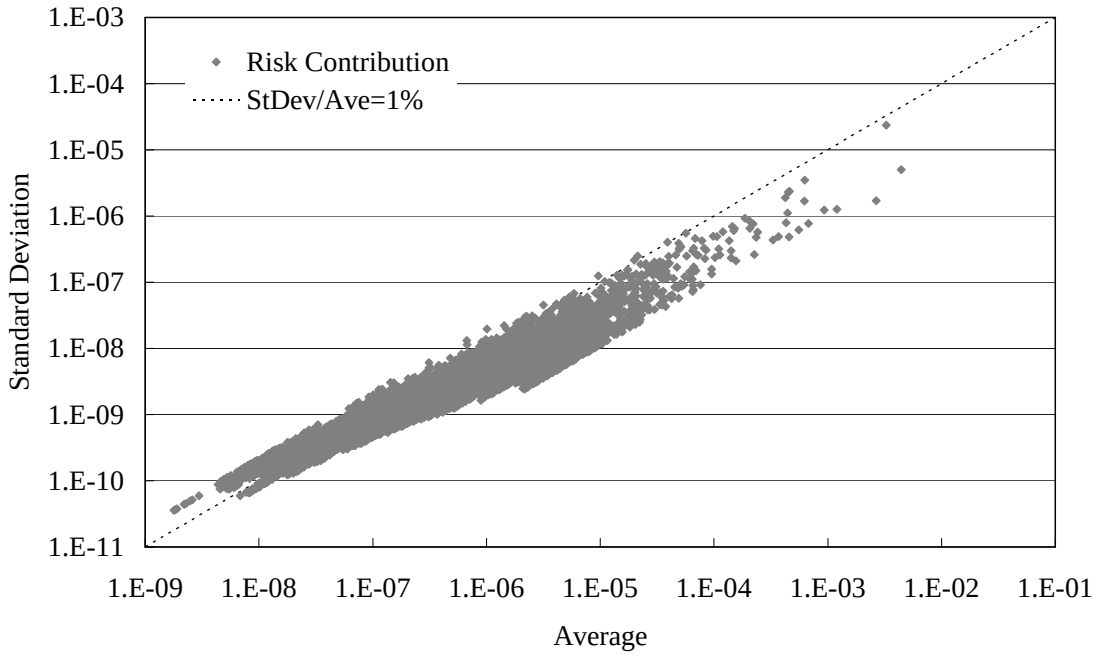

Figure 19: Average and standard deviation of risk contributions to 99% CVaR for portfolio C2.

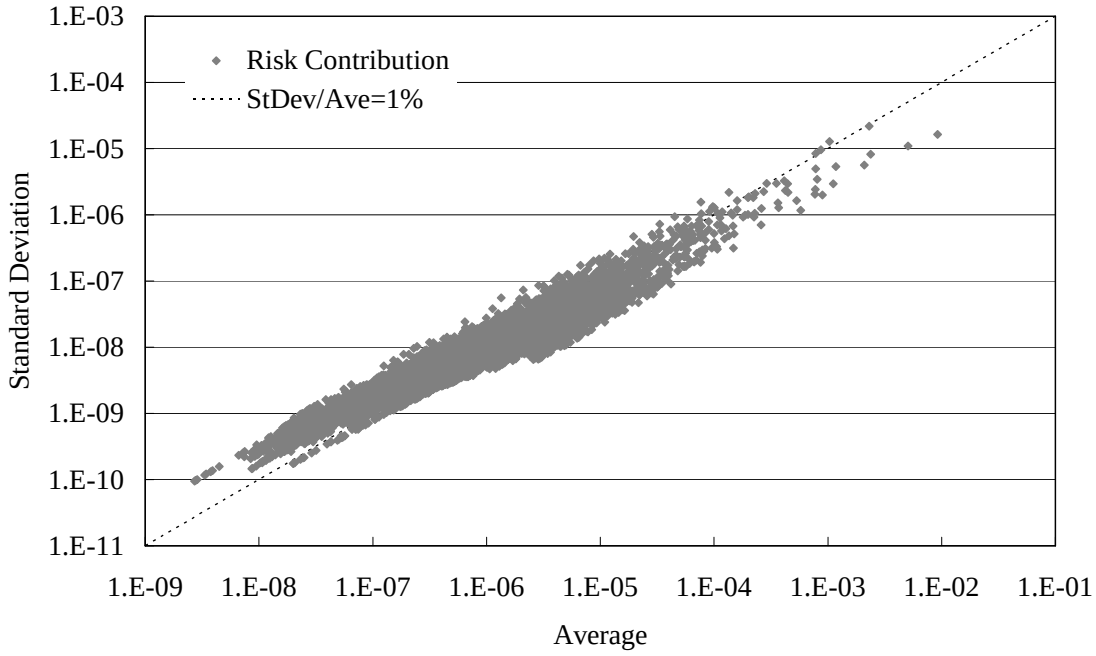

Figure 20: Average and standard deviation of risk contributions to 99.9% CVaR for portfolio C2.

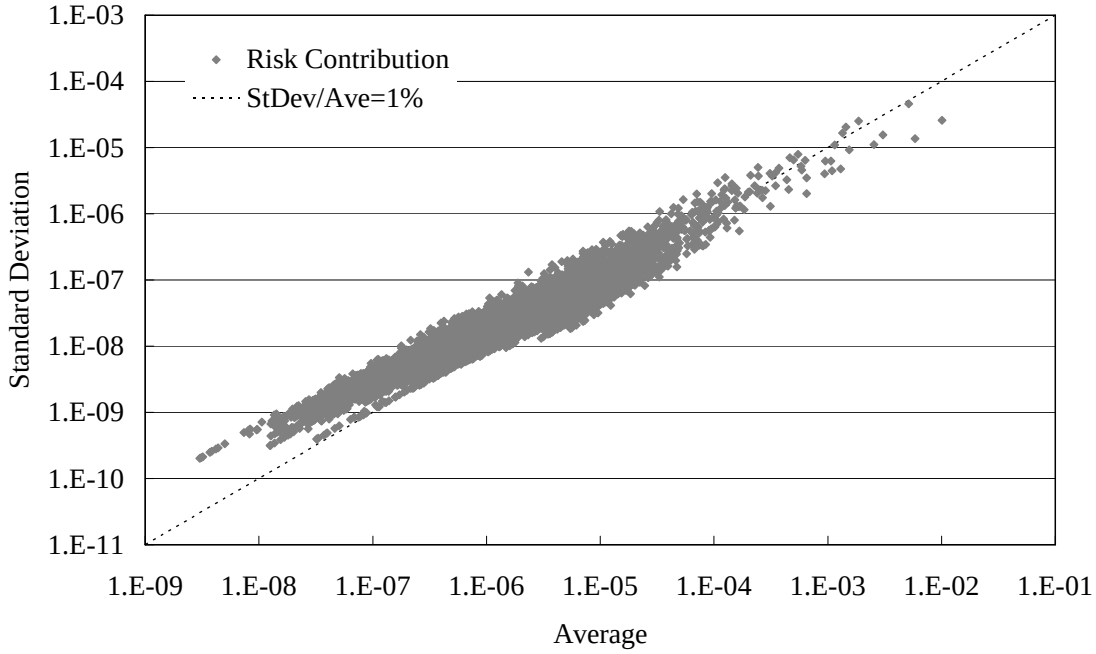

Figure 21: Average and standard deviation of risk contributions to 99.97% CVaR for portfolio C2.

		VaR			CVaR		
		99%	99.9%	99.97%	99%	99.9%	99.97%
Truncation Parameter							
$N_t = 50$	Max	0.223%	0.085%	0.371%	0.071%	0.272%	0.017%
	Min	−0.250%	−0.996%	−0.287%	−0.037%	−0.027%	−0.044%
$N_t = 100$	Max	0.039%	0.035%	0.007%	0.002%	0.007%	0.000%
	Min	−0.101%	−0.025%	−0.009%	−0.001%	−0.003%	−0.000%
Discretization Parameter							
$N_g = 64$	Max	0.759%	0.971%	0.651%	0.402%	0.757%	0.776%
	Min	0.045%	−0.010%	−0.197%	−0.379%	−0.045%	−0.168%
$N_g = 128$	Max	0.322%	0.271%	0.348%	0.133%	0.357%	0.409%
	Min	−0.089%	−0.053%	−0.079%	−0.108%	−0.083%	−0.276%

Table 15: Stability of risk contributions under the change of truncation and discretization parameters for portfolio C2.

Portfolio	N	$N_t = 50$	$N_t = 100$
C1	1,000	1min03sec	2min43sec
C2	10,000	1min05sec	2min51sec
C3	1,000,000	7min18sec	17min43sec

Table 16: Total computation time for computing whole distribution function and the risk contributions of all obligors.

computing risk contributions. Table 16 shows the total time required to obtain the cdf and the risk contributions of all obligors to 99% VaR and CVaR. As in the last subsection, we consider the cases with $N_t = 50$ and $N_t = 100$ (and other algorithm parameters unchanged). The additional time to compute risk contributions is longer than that for computing the cdf, for all portfolios. However, the total time remains sufficiently short even for a portfolio with 1 million obligors.

6 Conclusions

The fast numerical approximation presented in this paper consists of two steps. In the first step, moment generating function is computed efficiently with sufficient accuracy. Discretized valuation points play a crucial role to reduce computation time in this step. In the second step, the moment generating function is transformed into a loss distribution, through de Hoog algorithm. Remarkably, continued fraction expansion is used here to improve the slow convergence of an infinite series obtained by discretizing Laplace inversion integral. The risk contribution of transactions can also be computed only by slightly modifying these two steps.

We have studied the performance of the fast numerical approximation for some sample portfolios. As a result, the algorithm has been shown to be applicable to a wide range of realistic portfolios. In particular, whole distribution function is kept uniformly accurate even in the presence of exceptionally large exposures. The largest source of numerical error is the statistical error arising from Monte Carlo integration. However, this statistical error is kept small enough even for strongly correlated portfolios. Moreover, risk measures are stable under the change of important algorithm parameters. It is also remarkable that the dependence of computation time on portfolio size is moderate in the fast numerical approximation. This property is advantageous particularly for large commercial banks. In addition, we would like to stress that the fast numerical approximation can easily be implemented by an ordinary personal computer.

Another advantage of the fast numerical approximation is that the risk contribution of all transactions can be computed quickly and accurately. Tasche (2000) has proved that the definition of risk contribution in Section 2 is the only allocation method that is suitable for performance measurement. The fast numerical approximation thus serves as an effective tool for evaluating the risk/return profile of a credit portfolio. This performance

measurement is a first step to proactive credit portfolio management. In addition, the fast numerical approximation can also support decision-making concerning how a bank's total risk capital should be allocated into its business lines.

We enumerate possible extensions of the fast numerical approximation as follows. First, it is favorable that the algorithm can be applied not only to traditional loans but also to more complicated debt instruments. For instance, to handle securitization exposures characterized by tranched structure, it would be necessary to develop more sophisticated algorithm. Second, it is interesting to consider the case where LGD is stochastic. This extension is easily done by a slight refinement of the algorithm, provided that the stochastic LGD model is not so complicated. Finally, we stress that the fast MGF computation algorithm in section 4 works well regardless of the probability distribution of the systematic factors. Therefore, the fast numerical approximation can be extended to deal with systematic factors described by more general marginal distribution and copula function.

A Derivation of Risk Contribution Expressions

In this section, we derive the expressions for risk contributions, (32) and (33). To show (32), we replace partial derivatives in (22) with finite differences as

$$\mathrm{RC}_i \simeq -E_i \cdot \frac{F_L(E_i + \Delta/2, \mathrm{VaR}(\alpha)) - F_L(E_i - \Delta/2, \mathrm{VaR}(\alpha))}{F_L(E_i, \mathrm{VaR}(\alpha) + \Delta/2) - F_L(E_i, \mathrm{VaR}(\alpha) - \Delta/2)}.$$

From (15), the numerator is written as

$$\begin{aligned}
&F_L(E_i + \Delta/2, \mathrm{VaR}(\alpha)) - F_L(E_i - \Delta/2, \mathrm{VaR}(\alpha)) \\
&= \frac{1}{2\pi i} \int_{\gamma - i\infty}^{\gamma + i\infty} \frac{e^{\lambda \mathrm{VaR}(\alpha)}}{\lambda} \mathbf{E}\left[\frac{p_i(\tilde{\mathbf{X}}) \{ e^{-\lambda(E_i + \Delta/2)} - e^{-\lambda(E_i - \Delta/2)} \}}{1 - p_i(\tilde{\mathbf{X}}) + p_i(\tilde{\mathbf{X}}) \exp(-\lambda E_i)} M_L(\lambda; \tilde{\mathbf{X}}) \right] d\lambda. \\
&= \frac{1}{2\pi i} \int_{\gamma - i\infty}^{\gamma + i\infty} \frac{e^{\lambda(\mathrm{VaR}(\alpha) - \Delta/2)} - e^{\lambda(\mathrm{VaR}(\alpha) + \Delta/2)}}{\lambda} \mathbf{E}\left[p_i(\lambda; \tilde{\mathbf{X}}) M_L(\lambda; \tilde{\mathbf{X}}) \right] d\lambda. \\
&= \frac{1}{2\pi i} \int_{\gamma - i\infty}^{\gamma + i\infty} \frac{e^{\lambda(\mathrm{VaR}(\alpha) + \Delta/2)} - e^{\lambda(\mathrm{VaR}(\alpha) - \Delta/2)}}{\lambda^2} \frac{\partial M_L(\lambda)}{\partial E_i} d\lambda. \\
&= \left\{ (1 - \alpha_+) \mathrm{RC}_i^{\mathrm{CVaR}(\alpha_+)} - (1 - \alpha_-) \mathrm{RC}_i^{\mathrm{CVaR}(\alpha_-)} \right\} / E_i,
\end{aligned}$$

Therefore, we obtain an approximate relation between RC_i and $\mathrm{RC}_i^{\mathrm{CVaR}(\alpha)}$ as

$$\mathrm{RC}_i \simeq -\frac{(1 - \alpha_+) \mathrm{RC}_i^{\mathrm{CVaR}(\alpha_+)} - (1 - \alpha_-) \mathrm{RC}_i^{\mathrm{CVaR}(\alpha_-)}}{\alpha_+ - \alpha_-}.$$

Next we turn to the proof of (33). First, note that the r.h.s. of (33) is written as

$$\begin{aligned}
\mathbf{E}\left[\tilde{D}_i \,\middle|\, \tilde{L} \geq \mathrm{VaR}(\alpha) \right] &= \frac{\Pr(\{\tilde{D}_i = 1\} \wedge \{\tilde{L} \geq \mathrm{VaR}(\alpha)\})}{\Pr(\tilde{L} \geq \mathrm{VaR}(\alpha))}, \\
&= \frac{1}{1 - \alpha} \left\{ \Pr(\tilde{D}_i = 1) - \Pr(\{\tilde{D}_i = 1\} \wedge \{\tilde{L} < \mathrm{VaR}(\alpha)\}) \right\}, \\
&= \frac{1}{1 - \alpha} \left\{ \mathrm{PD}_{r(i)} - \Pr(\{\tilde{D}_i = 1\} \wedge \{\tilde{L} < \mathrm{VaR}(\alpha)\}) \right\}. \qquad (78)
\end{aligned}$$

The joint probability in (78) reads

$$\begin{aligned}
\Pr(\{\tilde{D}_i = 1\} \wedge \{\tilde{L} < \mathrm{VaR}(\alpha)\}) &= \mathbf{E}\left[\Pr(\{\tilde{D}_i = 1\} \wedge \{\tilde{L} < \mathrm{VaR}(\alpha)\} | \tilde{\mathbf{X}}) \right], \\
&= \mathbf{E}\left[\Pr(\{\tilde{D}_i = 1\} \wedge \{\tilde{L}^{i-} < \mathrm{VaR}(\alpha) - E_i\} | \tilde{\mathbf{X}}) \right], \\
&= \mathbf{E}\left[\Pr(\tilde{D}_i = 1 | \tilde{\mathbf{X}}) \Pr(\tilde{L}^{i-} < \mathrm{VaR}(\alpha) - E_i | \tilde{\mathbf{X}}) \right],
\end{aligned}$$

where we have defined $\tilde{L}^{i-} \equiv \tilde{L} - E_i \tilde{D}_i$, and used conditional independence on the third line. Since the relation

$$\Pr(\tilde{L}^{i-} < \mathrm{VaR}(\alpha) - E_i | \tilde{\mathbf{X}}) = \frac{1}{2\pi i} \int_{\gamma - i\infty}^{\gamma + i\infty} \frac{e^{\lambda(\mathrm{VaR}(\alpha) - E_i)} M_L(\lambda; \tilde{\mathbf{X}})}{\lambda \left\{ 1 - p_i(\tilde{\mathbf{X}}) + p_i(\tilde{\mathbf{X}}) e^{-\lambda E_i} \right\}} d\lambda$$

holds in analogy with (14), we obtain

$$\begin{aligned}
&\Pr(\{\tilde{D}_i = 1\} \wedge \{\tilde{L} < \mathrm{VaR}(\alpha)\}) \\
&\quad = \frac{1}{2\pi i} \int_{\gamma - i\infty}^{\gamma + i\infty} \frac{e^{\lambda(\mathrm{VaR}(\alpha) - E_i)}}{\lambda} \mathbf{E}\left[\frac{p_i(\tilde{\mathbf{X}}) M_L(\lambda; \tilde{\mathbf{X}})}{1 - p_i(\tilde{\mathbf{X}}) + p_i(\tilde{\mathbf{X}}) e^{-\lambda E_i}} \right] d\lambda, \\
&\quad = \frac{1}{2\pi i} \int_{\gamma - i\infty}^{\gamma + i\infty} \frac{e^{\lambda \mathrm{VaR}(\alpha)}}{\lambda} \mathbf{E}\left[p_i(\lambda; \tilde{\mathbf{X}}) M_L(\lambda; \tilde{\mathbf{X}}) \right] d\lambda, \\
&\quad = -\frac{1}{2\pi i} \int_{\gamma - i\infty}^{\gamma + i\infty} \frac{e^{\lambda \mathrm{VaR}(\alpha)}}{\lambda^2} \frac{\partial M_L(\lambda)}{\partial E_i} d\lambda.
\end{aligned} \tag{79}$$

Hence, (33) is derived from (31) by replacing the joint probability in (78) with (79).

B Padé Approximation

In Padé approximation, a function is approximated by a rational function. It often gives a much better result than that obtained by Taylor series approximation. Let us consider a function $S(z)$ that admits a formal power series expansion as

$$S(z) = \sum_{k=0}^{\infty} s_k z^k, \tag{80}$$

and suppose that two polynomials

$$P_M(z) \equiv \sum_{k=0}^{M} p_k z^k, \quad Q_N(z) \equiv 1 + \sum_{k=1}^{N} q_k z^k$$

are given. Then, a rational function $P_M(z)/Q_N(z)$ is said to be the $[M/N]$ *Padé approximation to* $S(z)$, if it satisfies

$$S(z) - \frac{P_M(z)}{Q_N(z)} = \mathcal{O}(z^{M+N+1}). \tag{81}$$

Let us write down the condition (81) more explicitly in terms of s_k, p_k and q_k, for two cases $N = M$ and $N = M + 1$. Since (81) is equivalent to

$$S(z) Q_N(z) - P_M(z) = \mathcal{O}(z^{M+N+1}), \tag{82}$$

we have

$$\begin{cases}
s_k + \displaystyle\sum_{l=1}^{k} s_{k-l} q_l - p_k = 0 & (k = 0, 1, \ldots, M), \\
s_k + \displaystyle\sum_{l=1}^{M} s_{k-l} q_l = 0 & (k = M+1, M+2, \ldots, 2M),
\end{cases} \tag{83}$$

for $M = N$, and

$$\begin{cases} s_k + \displaystyle\sum_{l=1}^{k} s_{k-l} q_l - p_k = 0 & (k = 0, 1, \ldots, M), \\ s_k + \displaystyle\sum_{l=1}^{M+1} s_{k-l} q_l = 0 & (k = M+1, M+2, \ldots, 2M+1), \end{cases} \tag{84}$$

for $N = M + 1$. Solving these equations, we obtain

$$p_k = \frac{\begin{vmatrix} & & s_{M+1} \\ & H_M^{(1)} & \vdots \\ & & s_{2M} \\ s_{k-M} & \cdots\ s_{k-1} & s_k \end{vmatrix}}{|H_M^{(1)}|} \quad (k = 0, 1, \ldots, M), \tag{85}$$

$$q_k = \frac{(C_{M+1}^{(1)})_{M+1,M-k+1}}{|H_M^{(1)}|} \quad (k = 1, 2, \ldots, M), \tag{86}$$

for $M = N$, and

$$p_k = \frac{\begin{vmatrix} & & s_{M+1} \\ & H_{M+1}^{(0)} & \vdots \\ & & s_{2M+1} \\ s_{k-M-1} & \cdots\ s_{k-1} & s_k \end{vmatrix}}{|H_{M+1}^{(0)}|} \quad (k = 0, 1, \ldots, M), \tag{87}$$

$$q_k = \frac{(C_{M+2}^{(0)})_{M+2,M-k+2}}{|H_{M+1}^{(0)}|} \quad (k = 1, 2, \ldots, M+1), \tag{88}$$

for $N = M + 1$. Here, $H_n^{(i)}$ is a Hankel matrix defined by (39), $(C_n^{(i)})_{i,j}$ denotes the (i, j)-cofactor of $H_n^{(i)}$, and s_i with $i < 0$ is assumed to be zero.

From (86) and (88), we can explicitly compute the coefficient of the first order term in (82) as

$$\begin{aligned} s_{2M+1} + \sum_{k=1}^{M} s_{2M+1-k} q_k &= \frac{|H_{M+1}^{(1)}|}{|H_M^{(1)}|} \quad (N = M), \\ s_{2M+2} + \sum_{k=1}^{M+1} s_{2M+2-k} q_k &= \frac{|H_{M+2}^{(0)}|}{|H_{M+1}^{(0)}|} \quad (N = M+1). \end{aligned}$$

Hence, we obtain

$$S(z)Q_N(z) - P_M(z) = \frac{|H_{M+1}^{(1)}|}{|H_M^{(1)}|} z^{2M+1} + \mathcal{O}(z^{2M+2}) \quad (N = M), \tag{89}$$

$$S(z)Q_N(z) - P_M(z) = \frac{|H_{M+2}^{(0)}|}{|H_{M+1}^{(0)}|} z^{2M+2} + \mathcal{O}(z^{2M+3}) \quad (N = M+1). \tag{90}$$

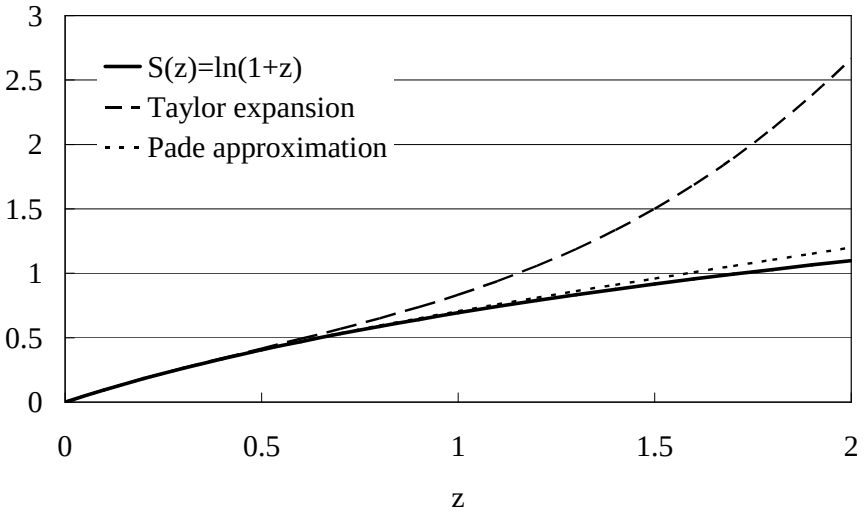

Figure 22: Comparison of two approximations to $S(z) = \ln(1+z)$.

This result is helpful in proving lemma 1 in the next section.

Example

Let us consider the case $S(z) = \ln(1+z)$, to demonstrate how well Padé approximation works. The third order Taylor expansion of $S(z)$ reads

$$S(z) = z - \frac{1}{2}z^2 + \frac{1}{3}z^3 + \mathcal{O}(z^4).$$

On the other hand, the [1/2] Padé approximation of $S(z)$ is given by

$$S(z) = \frac{z}{1 + z/2 - z^2/12} + \mathcal{O}(z^4).$$

The performance of these approximations is shown in Figure 22. In contrast with a large discrepancy between the Taylor expansion and $S(z)$ for $z > 1$, the Padé approximation gives a good result almost everywhere in $z < 2$.

C Proof of Lemmas

This section is devoted to proving the lemmas presented in section 3. We start with lemma 3, and then turn to lemma 1 and 2.

Proof of Lemma 3

Since it is trivial to compute the degrees of $A_n(z), B_n(z)$ and show $B_n(0) = 1$, we restrict ourselves to proving (48). It is straightforward to show (48) for $n = 1$. Next,

let us assume (48) to hold for $n = k$ (≥ 1). Since $C_{k+1}(z)$ is obtained by replacing c_k in $C_k(z)$ with $c_k/(1 + c_{k+1}z)$, we have

$$C_{k+1}(z) = \frac{A'_k(z)}{B'_k(z)},$$

where we have defined

$$\begin{aligned} A'_k(z) &\equiv A_{k-1}(z) + \frac{c_k z}{1 + c_{k+1}z} A_{k-2}(z), \\ B'_k(z) &\equiv B_{k-1}(z) + \frac{c_k z}{1 + c_{k+1}z} B_{k-2}(z). \end{aligned}$$

Hence, we obtain

$$\begin{aligned} C_{k+1}(z) &= \frac{(1 + c_{k+1}z)A_{k-1}(z) + c_k z A_{k-2}(z)}{(1 + c_{k+1}z)B_{k-1}(z) + c_k z B_{k-2}(z)}, \\ &= \frac{\{A_{k-1}(z) + c_k z A_{k-2}(z)\} + c_{k+1}z A_{k-1}(z)}{\{B_{k-1}(z) + c_k z B_{k-2}(z)\} + c_{k+1}z B_{k-1}(z)}, \\ &= \frac{A_k(z) + c_{k+1}z A_{k-1}(z)}{B_k(z) + c_{k+1}z B_{k-1}(z)}, \\ &= \frac{A_{k+1}(z)}{B_{k+1}(z)}, \end{aligned}$$

and the lemma is proved by induction.

Proof of Lemma 1

Note that, from lemma 3, $A_n(z)$ and $B_n(z)$ defined by (47) are polynomials of degree $\lfloor n/2 \rfloor$ and $\lfloor (n+1)/2 \rfloor$ respectively, and satisfy

$$\begin{cases} C_n(z) = A_n(z)/B_n(z), \\ B_n(0) = 1. \end{cases} \qquad (n = 0, 1, 2, \ldots)$$

Therefore, we expect that $C_n(z)$ gives the $[\lfloor n/2 \rfloor / \lfloor (n+1)/2 \rfloor]$ Padé approximation to $S(z)$, if coefficients c_k are suitably chosen. This is indeed the case for $n = 0$ and $n = 1$, if we choose

$$c_0 = s_0, \quad c_1 = -\frac{s_1}{s_0} = -\frac{|H_0^{(0)}||H_1^{(1)}|}{|H_1^{(0)}||H_0^{(1)}|}$$

in accordance with (41).

Now we assume that, for an integer $k \geq 0$, $C_{2k}(z)$ and $C_{2k+1}(z)$ are respectively the $[k/k]$ and $[k/k+1]$ Padé approximations to $S(z)$. Then, it follows from (89) and (90) that

$$\begin{aligned} S(z)B_{2k}(z) - A_{2k}(z) &= \frac{|H_{k+1}^{(1)}|}{|H_k^{(1)}|} z^{2k+1} + \mathcal{O}(z^{2k+2}), \\ S(z)B_{2k+1}(z) - A_{2k+1}(z) &= \frac{|H_{k+2}^{(0)}|}{|H_{k+1}^{(0)}|} z^{2k+2} + \mathcal{O}(z^{2k+3}). \end{aligned}$$

Hence,

$$\begin{aligned} S(z)B_{2k+2}(z) - A_{2k+2}(z) &= S(z)\{B_{2k+1}(z) + c_{2k+2}zB_{2k}(z)\} \\ &\quad -\{A_{2k+1}(z) + c_{2k+2}zA_{2k}(z)\}, \\ &= S(z)B_{2k+1}(z) - A_{2k+1}(z) \\ &\quad +c_{2k+2}z\{S(z)B_{2k}(z) - A_{2k}(z)\}, \\ &= \left(\frac{|H_{k+2}^{(0)}|}{|H_{k+1}^{(0)}|} + \frac{|H_{k+1}^{(1)}|}{|H_k^{(1)}|}c_{2k+2}\right) z^{2k+2} + \mathcal{O}(z^{2k+3}), \\ &= \mathcal{O}(z^{2k+3}), \end{aligned}$$

showing $C_{2k+2}(z) = A_{2k+2}(z)/B_{2k+2}(z)$ to be the $[k+1/k+1]$ Padé approximation.

In a similar way, we can verify that $C_{2k+3}(z)$ is the $[k+1/k+2]$ Padé approximation to $S(z)$, if $C_{2k+1}(z)$ and $C_{2k+2}(z)$ are respectively the $[k/k+1]$ and $[k+1/k+1]$ Padé approximations to $S(z)$ for an integer $k \geq 0$. The proof is thus completed.

Proof of Lemma 2

This lemma can be proved by induction with respect to integer k in (45). For $k = 1$, we can show (45) as

$$\begin{aligned} q_1^{(i)} &= \frac{s_{i+1}}{s_i} = \frac{|H_0^{(i)}||H_1^{(i+1)}|}{|H_1^{(i)}||H_0^{(i+1)}|}, \\ e_1^{(i)} &= e_0^{(i+1)} + q_1^{(i+1)} - q_1^{(i)} = \frac{s_{i+2}}{s_{i+1}} - \frac{s_{i+1}}{s_i} = \frac{s_i s_{i+2} - s_{i+1}^2}{s_i s_{i+1}} = \frac{|H_2^{(i)}||H_0^{(i+1)}|}{|H_1^{(i)}||H_1^{(i+1)}|}. \end{aligned}$$

Let us assume (45) to hold for all $k \leq l$ for an integer $l \geq 1$. Then, we can verify the second equation in (45) for $k = l + 1$ as

$$\begin{aligned} q_{l+1}^{(i)} &= q_l^{(i+1)} e_l^{(i+1)} / e_l^{(i)}, \\ &= \frac{|H_{l-1}^{(i+1)}||H_l^{(i+2)}|}{|H_l^{(i+1)}||H_{l-1}^{(i+2)}|} \cdot \frac{|H_{l+1}^{(i+1)}||H_{l-1}^{(i+2)}|}{|H_l^{(i+1)}||H_l^{(i+2)}|} \Big/ \frac{|H_{l+1}^{(i)}||H_{l-1}^{(i+1)}|}{|H_l^{(i)}||H_l^{(i+1)}|}, \\ &= \frac{|H_l^{(i)}||H_{l+1}^{(i+1)}|}{|H_{l+1}^{(i)}||H_l^{(i+1)}|}. \end{aligned}$$

It is more difficult to prove the first equation in (45) for $k = l + 1$. Note first that

$$C_{2k+1}(z) = c_0 \Big/ \left(1 + \left[\frac{c_m z}{1}\right]_{m=1}^{2k+1}\right)$$

is the $[k/k+1]$ Padé approximation to $S_{i+1}(z) \equiv \sum_{k=0}^{\infty} s_{i+1+k} z^k$ as shown by lemma 1, if $k \leq l$ and

$$\begin{aligned} &c_0 = s_{i+1}, \quad c_1 = -q_1^{(i+1)}, \\ &c_{2k} = -e_k^{(i+1)}, \quad c_{2k+1} = -q_{k+1}^{(i+1)} \quad (k = 1, 2, \ldots, l). \end{aligned}$$

Therefore, if we write down the polynomial $B_{2k+1}(z)$ $(k \leq l)$ defined by (47) explicitly as

$$B_{2k+1}(z) = 1 + \sum_{m=1}^{k+1} q_m z^m,$$

its coefficients q_m must satisfy

$$q_m = \frac{(C_{k+2}^{(i+1)})_{k+2,k-m+2}}{|H_{k+1}^{(i+1)}|}$$

as derived from (88), since $B_{2k+1}(z)$ is the denominator of the Padé approximation $C_{2k+1}(z)$. Thus, comparing the terms proportional to z in

$$\begin{aligned} B_{2l+1}(z) &= B_{2l}(z) + c_{2l+1} z B_{2l-1}(z), \\ &= B_{2l-1}(z) + c_{2l} z B_{2l-2}(z) + c_{2l+1} z B_{2l-1}(z), \end{aligned}$$

we obtain

$$\frac{(C_{l+2}^{(i+1)})_{l+2,l+1}}{|H_{l+1}^{(i+1)}|} = \frac{(C_{l+1}^{(i+1)})_{l+1,l}}{|H_l^{(i+1)}|} + c_{2l} + c_{2l+1}.$$

Hence

$$e_l^{(i+1)} + q_{l+1}^{(i+1)} = \frac{(C_{l+1}^{(i+1)})_{l+1,l}}{|H_l^{(i+1)}|} - \frac{(C_{l+2}^{(i+1)})_{l+2,l+1}}{|H_{l+1}^{(i+1)}|}. \tag{91}$$

Next, let us redefine c_k by

$$\begin{aligned} &c_0 = s_i, \quad c_1 = -q_1^{(i)}, \\ &c_{2k} = -e_k^{(i)}, \quad c_{2k+1} = -q_{k+1}^{(i)} \quad (k = 1, 2, \ldots, l), \\ &c_{2l+2} = -\frac{|H_{l+2}^{(i)}||H_l^{(i+1)}|}{|H_{l+1}^{(i)}||H_{l+1}^{(i+1)}|}. \end{aligned}$$

Then, the corresponding continued fraction

$$C_{2k+2}(z) = c_0 \Big/ \left(1 + \left[\frac{c_m z}{1}\right]_{m=1}^{2k+2}\right)$$

with $k \leq l$ is the $[k+1/k+1]$ Padé approximation to $S_i(z) = \sum_{k=0}^{\infty} s_{i+k} z^k$ as shown by lemma 1. Similarly to the above argument, the coefficients of

$$B_{2k+2}(z) = 1 + \sum_{m=1}^{k+1} q_m z^m$$

defined by (47) satisfy

$$q_m = \frac{(C_{k+2}^{(i+1)})_{k+2,k-m+2}}{|H_{k+1}^{(i+1)}|},$$

as a result of (86). If we compare again the terms proportional to z in

$$\begin{aligned} B_{2l+2}(z) &= B_{2l+1}(z) + c_{2l+2} z B_{2l}(z), \\ &= B_{2l}(z) + c_{2l+1} z B_{2l-1}(z) + c_{2l+2} z B_{2l}(z), \end{aligned}$$

it follows that

$$\frac{(C_{l+2}^{(i+1)})_{l+2,l+1}}{|H_{l+1}^{(i+1)}|} = \frac{(C_{l+1}^{(i+1)})_{l+1,l}}{|H_{l}^{(i+1)}|} + c_{2l+1} + c_{2l+2}.$$

Hence

$$q_{l+1}^{(i)} + \frac{|H_{l+2}^{(i)}||H_{l}^{(i+1)}|}{|H_{l+1}^{(i)}||H_{l+1}^{(i+1)}|} = \frac{(C_{l+1}^{(i+1)})_{l+1,l}}{|H_{l}^{(i+1)}|} - \frac{(C_{l+2}^{(i+1)})_{l+2,l+1}}{|H_{l+1}^{(i+1)}|}. \tag{92}$$

Subtracting (92) from (91), we obtain

$$\begin{aligned} e_{l+1}^{(i)} &= e_{l}^{(i+1)} + q_{l+1}^{(i+1)} - q_{l+1}^{(i)} \\ &= \frac{|H_{l+2}^{(i)}||H_{l}^{(i+1)}|}{|H_{l+1}^{(i)}||H_{l+1}^{(i+1)}|}, \end{aligned}$$

which completes the proof.

References

Abate, J., Choudhury, G.L., and Whitt, W. (2000). An introduction to numerical transform inversion and its application to probability models. In *Computational Probability*, ed. W.K. Grassman, Kluwer, Norwell, MA.

Ahn, J., Kang, S., and Kwon, Y. (2003). A flexible inverse Laplace transform algorithm and its application. *Computing* **71**. 115-131.

Basel Committee on Banking Supervision (BCBS). (2005). International convergence of capital measurement and capital standards : a revised framework.

Basel Committee on Banking Supervision. (2006). Studies on credit risk concentration. Working Paper.

de Hoog, F., Knight, J., and Stokes, A. (1982). An improved method for numerical inversion of Laplace transforms. *SIAM Journal on Scientific and Statistical Computing* **3**. 357-366.

Emmer, S. and Tasche, D. (2003). Calculating credit risk capital charges with the one-factor model. arXiv:cond-mat/0302402. (available at http://arxiv.org/)

Gordy, M. (2003). A risk-factor model foundation for ratings-based bank capital rules. *Journal of Financial Intermediation* **12**. 199-232.

Glasserman, P. (2005). Measuring marginal risk contributions in credit portfolios. *Journal of Computational Finance* **9**. 1-41.

Glasserman, P. and Ruiz-Mata, J. (2006). Computing the credit loss distribution in the Gaussian copula model: a comparison of methods. *Journal of Credit Risk* **3**. 33-66.

Higo, H. (2006). An efficient Monte Carlo method for a large and nongranular credit portfolio. Bank of Japan Working Paper Series No.06-E-19.

Hollenbeck, K. (1998). INVLAP.M : A matlab function for numerical inversion of Laplace transforms by de Hoog algorithm. (available at http://www.mathworks.com/)

Huang, X., Oosterlee, C., and Mesters, M. (2007). Computation of VaR and VaR contribution in the Vasicek portfolio credit loss model: a comparative study. Working paper, Delft University of Technology.

Martin, R. and Ordovás, R. (2006). An indirect view from the saddle. *Risk* **19**. 94-9.

Martin, R., Thompson, K., and Browne, C. (2001). Taking to the saddle. *Risk* **14**, 91-4.

Pykhtin, M. (2004). Multi-factor adjustment. *Risk* **17**, 85-90.

Rutishauser, H. (1954). Der quotienten-differenzen-algorithmus. *Zeitschrift für Angewandte Mathematik und Physik* **5**, 233-251.

Tasche, D. (2000). Risk contributions and performance measurement. Working paper, Munich University of Technology.

Tasche, D. (2006). Measuring sectoral diversification in an asymptotic multi-factor framework. *Journal of credit risk* **2**. 33-55.