Adaptive integration for multi-factor portfolio credit loss models

Xinzheng Huang[a,b,*], Cornelis. W. Oosterlee[a,c]

[a] *Delft Institute of Applied Mathematics, Delft University of Technology,*
Mekelweg 4, 2628CD, Delft, the Netherlands

[b] *Group Risk Management, Rabobank,*
Croeselaan 18, 3521CB, Utrecht, the Netherlands

[c] *CWI - National Research Institute for Mathematics and Computer Science,*
Kruislaan 413, 1098 SJ, Amsterdam, the Netherlands

October 5, 2007

Abstract

We propose algorithms of adaptive integration for calculation of the tail probability in multi-factor credit portfolio loss models. We first devise the classical Genz-Malik rule, a deterministic multiple integration rule suitable for portfolio credit models with number of factors less than 8. Later on we arrive at the adaptive Monte Carlo integration, which simply replaces the deterministic integration rule by pseudo-random numbers. The latter can not only handle higher-dimensional models but is also able to provide reliable probabilistic error bounds. Both algorithms are asymptotic convergent and consistently outperform the plain Monte Carlo method.

1 Introduction

We consider the computation of the tail probability of credit portfolio loss L in a multi-factor model like CreditMetrics (Gupton et al. 1997). In such latent factor models the obligors are assumed to be independent conditional on some d latent factors, denoted by $\mathbf{Y^d}$. We are interested in the estimation of the tail probability

$$P(L > x) = \int P\left(L > x \,|\, \mathbf{Y^d}\right) dP(\mathbf{Y^d}),$$

[*] Corresponding author; E-mail: X.Huang@ewi.tudelft.nl

especially for extreme losses x. This is essential for the determination of the portfolio Value at Risk (VaR). The integrand $P\left(L > x \,|\, \mathbf{Y^d}\right)$ can be approximated with ease since conditional on $\mathbf{Y^d}$, the portfolio loss L reduces to a sum of independent random variables. Various approximations exist and prove to work very well, for example, the recursive method due to Andersen et al. (2003), the normal approximation method as in Martin (2004) and the saddlepoint approximation presented in Huang et al. (2007). In a one-factor model ($d = 1$), the calculation of the integral can be handled efficiently by Gaussian quadrature.

When one wishes to take into account the effects of different industries and geographical regions, multiple factors become necessary. A multi-factor model is certainly able to capture the correlation structure among obligors (or defaults) better. Meanwhile the computation of the tail probability $P(L > x)$ in a multi-factor model is much more involved. The product quadrature rule becomes impractical because the number of function evaluations grows exponentially with d and the so-called *curse of dimensionality* arises.

In this article we deal with the high-dimensionality and show that *adaptive integration algorithms* are very well suited for the calculation of the tail probability. An adaptive integration algorithm successively divides the integration region into subregions, detects the subregions where the integrand is most irregular, and places more points in those subregions. We first devise the Genz-Malik rule (Genz & Malik 1980), a deterministic multiple integration rule suitable for portfolio credit models with a number of factors less than 8. Later on we arrive at the adaptive Monte Carlo integration, which simply replaces the deterministic integration rule by pseudo-random numbers.

The rest of the article is organized as follows. We give in section 2 an introduction into a multi-factor portfolio credit loss model and derive certain properties of the conditional tail probability as a function of the common factors. Section 3 gives a motivation for adaptive integration by means of a one-factor model example. In section 4 we briefly review the globally adaptive integration algorithm and the Genz-Malik rule. Section 5 presents a tailor-made Genz-Malik rule for the computation of tail probability in the context of portfolio credit loss, followed by some numerical results in section 6. We then discuss the adaptive Monte Carlo integration in section 7 and show some numerical results in section 8. Section 9 concludes.

2 Multi-factor portfolio credit loss model

Consider a credit portfolio consisting of n obligors with exposure w_i, $i = 1, \ldots, n$. Assume that obligor i defaults if its standardized log asset value X_i is less than some default threshold γ_i after a fixed time horizon. The event of default can be modeled as a Bernoulli random variable $D_i = 1_{\{X_i < \gamma_i\}}$ with known default probability $p_i = P(X_i < \gamma_i)$. It follows that the loss L_i due to obligor i is simply $w_i D_i$ and the portfolio loss is given by

$$L = \sum_{i=1}^{n} L_i = \sum_{i=1}^{n} w_i D_i. \tag{1}$$

A key issue in portfolio credit loss modeling is the modeling of the default dependence among obligors. It is common practice to utilize a latent factor model. The correlations among L_i are specified implicitly by a factor model of asset correlations such that

$$X_i = \alpha_{i1} Y_1 + \cdots + \alpha_{id} Y_d + \beta_i Z_i, \tag{2}$$

where $(Y_1 \ldots Y_d)$ and Z_i are independent for all i. $(Y_1 \ldots Y_d)$ are thought as systematic factors that affect more than one obligor and Z_i is an idiosyncratic part that only affects an obligor itself. In case that $d = 1$ the model reduces to a one-factor model, in which Y can be interpreted as the "state of economy". A well-known example of the one-factor model is the Vasicek (2002) model. More factors are necessary if one wishes to take the effects of different industries and geographical regions into account. The resulting multi-factor model offers a better solution to identifying the correlations among individual obligors. Write

$$\mathbf{Y^d} = (Y_1, \ldots, Y_d) \quad \text{and} \quad \alpha_i = (\alpha_{i1}, \ldots, \alpha_{id}).$$

It is easily deduced that X_i and X_j are conditionally independent given the realization of $\mathbf{Y^d}$. This implies that L_i and L_j are also conditionally independent given $\mathbf{Y^d}$.

In this article we are interested in the estimation of tail probability

$$P(L > x) = \int P\left(L > x \,|\, \mathbf{Y^d}\right) dP(\mathbf{Y^d}), \tag{3}$$

especially for extreme losses x. This is essential for the determination of the portfolio Value at Risk (VaR). For now we consider the widely used Gaussian factor model as in CreditMetrics (Gupton et al. 1997), where $Y_1 \ldots Y_d$ and all Z_i are i.i.d. standard normal random variables. $\alpha_{i1}^2 + \cdots + \alpha_{id}^2 + \beta_i^2 = 1$ so that the X_i are also standard normally distributed. We further assume that all α_{ik} and β_i are nonnegative.

Under this setup the probability of default of obligor i conditional on the common factor $\mathbf{Y^d}$ is given by

$$p_i\left(\mathbf{Y^d}\right) = P\left(D_i = 1 | \mathbf{Y^d}\right) = P\left(X_i < \gamma_i | \mathbf{Y^d}\right) = \Phi\left(\frac{\Phi^{-1}(p_i) - \alpha_i \cdot \mathbf{Y^d}}{\beta_i}\right), \tag{4}$$

where Φ denotes the cumulative distribution function of the standard normal distribution. Equation (4) shows that the individual conditional default probability is non-increasing in $\mathbf{Y^d}$. An important consequence is that the conditional tail probability of portfolio loss $P\left(L > x \,|\, \mathbf{Y^d}\right)$ is also non-decreasing in $\mathbf{Y^d}$. Without loss of generality, we prove the following proposition.

Proposition 1. *The mapping*

$$y_k \longmapsto P(L > x | Y_1 = y_1, Y_2 = y_2, \dots, Y_d = y_d), \quad k = 1, \cdots, d,$$

is non-increasing in y_k.

Proof. Let us write

$$L = \sum_{i=1}^{n} w_i \mathbf{1}_{\{X_i < \gamma_i\}} = \sum_{i=1}^{n} w_i \mathbf{1}_{\{\alpha_{i1} y_1 + \cdots + \alpha_{id} y_d + \beta_i Z_i < \gamma_i\}}.$$

The conditional tail probability can be reformulated to be

$$P(L > x | Y_1 = y_1, Y_2 = y_2, \dots, Y_d = y_d) = P\left(\sum_{i=1}^{n} w_i \mathbf{1}_{\{\alpha_{i1} y_1 + \cdots + \alpha_{id} y_d + \beta_i Z_i < \gamma_i\}} > x \right).$$

The indicator function

$$\mathbf{1}_{\{\alpha_{i1} y_1 + \cdots + \alpha_{id} y_d + \beta_i Z_i < \gamma_i\}} = \mathbf{1}_{\{Z_i < \frac{1}{\beta_i}(\gamma_i - \alpha_{i1} y_1 - \cdots - \alpha_{id} y_d)\}}$$

is non-increasing in y_k for all k when α_{ik} and β_i are nonnegative for all i. It follows that

$$\sum_{i=1}^{n} w_i \mathbf{1}_{\{\alpha_{i1} y_1 + \cdots + \alpha_{id} y_d + \beta_i Z_i < \gamma_i\}}$$

is also non-increasing in y_k for all k. This immediately leads to the assertion. □

In addition it is not difficult to derive that $P(L > x | -\infty, \dots, -\infty) = 1$ and $P(L > x | +\infty, \dots, +\infty) = 0$.

The rest of this article hinges strongly on the validity of Proposition 1. Note that Proposition 1 is a quite general result. Its proof is not contingent on the assumption that $Y_1, \dots Y_d$ are independent. The distributions of $\mathbf{Y^d}$ and Z_i, $i = 1, \dots, n$ are not relevant either. The only two necessary conditions for the monotonicity are

1. $\mathbf{Y^d}$ and Z_i, $i = 1 \dots n$ are independent, and
2. the factor loadings, α_{ik}, $i = 1, \cdots, n$, $k = 1, \cdots, d$ are all nonnegative.

Proposition 2. *$P(L > x | Y_1, Y_2, \dots, Y_d)$ is continuous and differentiable with respect to Y_k, $k = 1, \cdots, d$.*

Proof. Denote by $\varepsilon = (\varepsilon_1, \dots, \varepsilon_n) = \{0, 1\}^n$ a realization of $(D_1, \dots, D_n)$ and write $w = (w_1, \dots, w_n)$. The conditional tail probability is given by

$$P(L > x | \mathbf{Y^d}) = \sum_{\varepsilon : w \cdot \varepsilon > x} P\left(D_i = \varepsilon_i, i = 1, \dots, n | \mathbf{Y^d}\right)$$

As D_i and D_j are independent conditional on $\mathbf{Y^d}$, we get

$$P\left(D_i = \varepsilon_i, i = 1, \ldots, n | \mathbf{Y^d}\right) = \prod_{i=1}^{n} \left[p_i\left(\mathbf{Y^d}\right)\right]^{\varepsilon_i} \left[1 - p_i\left(\mathbf{Y^d}\right)\right]^{1-\varepsilon_i}.$$

Since $p_i\left(Y_1, Y_2, \ldots, Y_d\right)$ is continuous and differentiable in Y_k for all k, so is the tail probability $P(L > x | Y_1, Y_2, \ldots, Y_d)$.

□

3 Motivation for adaptive integration: a one-factor model

We start with a Gaussian one-factor model, motivating the need for adaptivity in the numerical integration. For integration we employ a straightforward N-point Gauss-Legendre quadrature rule. We truncate the domain of the common factor Y to the interval $[-5, 5]$ so that the probability of Y falling out of this interval is merely 5.7×10^{-7}. Denote the Gauss nodes and weights by Y_k with $Y_1 > Y_2 > \ldots > Y_N$ and u_k, $k = 1, \ldots, N$, respectively. The tail probability $P(L > x)$ is then approximated by

$$P(L > x) \approx \int_{-5}^{5} P(L > x | Y) dP(Y) \approx \sum_{k=1}^{N} P(L > x | Y_k) \phi(Y_k) u_k, \tag{5}$$

where ϕ denotes the probability distribution function of the standard normal distribution.

Take as an example a homogeneous portfolio A consisting of 1000 obligors with $w_i = 1$, $p_i = 0.0033$ and $\alpha_i = \sqrt{0.2}$, $i = 1, \ldots, 1000$. The integrand $P(L > x|Y)$ with $x = 100$ is illustrated in Figure 1. It is a non-increasing function of Y. Furthermore, it decreases rapidly from its upper bound 1 to its lower bound 0 for Y in a narrow band (between the two dashed vertical lines in Figure 1) much smaller than the domain of Y. Note that the band will move toward the left tail of Y as the loss level x increases. Moreover the width of the band should further decrease as the number of the obligors n increases. Asymptotically, as $n \to \infty$, $P(L > x|Y)$ approaches a Heaviside step function. Due to the law of large numbers, $L(Y) \to \sum w_i p_i(Y)$ *a.s.* and $P(L > x|Y) \to \mathbf{1}_{\{\sum w_i p_i(Y) > x\}}$.

Generally an N-point quadrature rule demands N integrand evaluations. However since in our problem the integrand is monotone and bounded in $[0, 1]$, significantly fewer evaluations are necessary with an adaptive integration algorithm for the same accuracy. Below we give a simple procedure that utilizes the nodes of an N-point Gauss-Legendre quadrature rule. It produces identical results for the integral (5) as the N-point quadrature but it substantially reduces the number of integrand evaluations. For simplicity, we write $f(Y)$ instead of $P(L > x|Y)$ and denote $I = P(L > x)$. Basically the algorithm first identifies the smallest node y_1 giving $f(y_1) = 0$. It then discards all nodes larger than

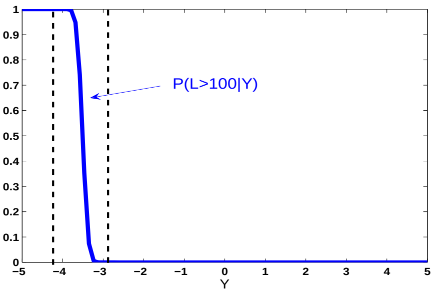


Figure 1: The integrand $P(L > 100|Y)$ as a function of the common factor Y for portfolio A, which consists of 1000 obligors with $w_i = 1$, $p_i = 0.0033$ and $\alpha_i = \sqrt{0.2}$, $i = 1, \ldots, 1000$.

y_1 and proceeds sequentially with decreasing Y until we find a y_2 such that $f(y_2) = 1$. For all $Y < y_2$ we set $f(Y) = 1$.

Algorithm 1 1D adaptive integration

```
Generate the $N$-degree Gaussian nodes $Y_1, \ldots, Y_N$ and weights $u_1, \ldots, u_N$
Find $Y_i = \min\{Y_k | f(Y_k) = 0, k = 1, \ldots, N\}$
$j = i + 1$, $I = 0$
while $j \leq N, f(Y_j) < 1$ do
    $I = I + f(Y_j) \cdot \phi(Y_j) \cdot u_j$
    $j = j + 1$
end while
$I = I + \sum_{k=j}^{N} \phi(Y_k) \cdot u_k$
```

For the above example with $N = 100$ this algorithm results in less than 20 integrand evaluations. It is evident that an adaptive integration algorithm is able to effectively reduce the amount of computations in a one-factor model. Unfortunately the above algorithm cannot be extended to a multi-factor model by a simple product rule.

4 Globally adaptive algorithms for numerical integration

Consider now a general integral over a d-dimensional rectangular region $\mathcal{C}^d$

$$I(f) = \int \cdots \int_{\mathcal{C}^d} f(\mathbf{x}) g(\mathbf{x}) dx_1 dx_2 \cdots dx_d, \tag{6}$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_d)$ and $g(\cdot)$ is a weight function.

Monte Carlo simulation and quasi-Monte Carlo methods are the prevailing methods used to solve multi-dimensional problems in finance. Both methods do not suffer from the dimensional effect. The Monte Carlo method is known to be only accurate with a tremendous amount of scenarios since its rate of convergence is $O\left(1/\sqrt{N}\right)$. Quasi-Monte Carlo methods use deterministic sequences that have better uniform properties measured by discrepancy. They are usually superior to the Monte Carlo method as they have a convergence rate of $O\left((\log N)^d/N\right)$. The advantage of the Monte Carlo method is however that it gives practical probabilistic error bounds. As for quasi-Monte Carlo methods even though they are able to provide a deterministic error bound, the bounds can be unrealistically pessimistic (cf. Spanier & Maize 1994, Cools 2002).

Recall that integration with both Monte Carlo and quasi-Monte Carlo methods require a transformation of integration region into the unit cube $[0,1]^d$. Pseudo-random numbers or quasi-random sequences are then generated uniformly in the $[0,1]^d$ cube. This can however be inefficient if most of the points are placed outside the regions which are significant for the evaluation of the integral. In this respect the better uniform properties of quasi-Monte Carlo sequences over Monte Carlo simulation can be meaningless. In addition, both Monte Carlo and quasi-Monte Carlo methods are not able to take advantage of the regularity of the integrand.

4.1 Preliminaries

An adaptive integration algorithm differs fundamentally from Monte Carlo and quasi-Monte Carlo methods in that it successively divides the integration region into subregions, detects the subregions where the integrand is most irregular, and places more points in those subregions.

We restrict ourselves to the globally adaptive algorithms for multidimensional integration, which typically have a structure that consists of the following steps:

1. Choose a subregion from a collection of subregions and subdivide the chosen subregion.

2. Apply an integration rule to the resulting new subregions; update the collection of subregions.

3. Update the global integral and error estimate; check whether a predefined termination criterion is met; if not, go back to step 1 .

The two important ingredients of an adaptive algorithm are

1. an integration rule for estimating the integral in each subregion.

2. an error estimate for each subregion.

Definition 1. *An integration rule for the cube $[-1,1]^d$ is **fully symmetric** if, whenever the rule contains a point $\mathbf{x} = (x_1, x_2, ..., x_d)$ with associated weight u, it also contains all points that can be generated from $\mathbf{x}$ by permutations and/or sign-changes of the coordinates with the same associated weight.*

Example 1. *If a fully symmetric integration rule for the square* $[-1,1]^2$ *contains* (x_1, x_2)*, with* $x_1 \neq x_2$*, then it also contains the following points,* $(x_1, -x_2)$*,* $(-x_1, x_2)$*,* $(-x_1, -x_2)$*,* (x_2, x_1)*,* $(x_2, -x_1)$*,* $(-x_2, x_1)$*,* $(-x_2, -x_1)$*.*

A variety of fully symmetric polynomial interpolation rules for multiple integration in a d-rectangle are available. An integration rule has *polynomial degree* m if it integrates exactly all *monomials* $x_1^{k_1} x_2^{k_2} \dots x_n^{k_d}$ with $\sum k_i \leq m$ and fails to integrate exactly at least one monomial of degree $m+1$. For a comprehensive review see Stroud (1971), Cools & Rabinowitz (1993), Cools (1999).

An error estimate ϵ is generally taken to be the difference of two quadrature rules with different degrees of exactness m_1 and m_2,

$$\epsilon = I_{m_1} - I_{m_2}, \quad m_1 > m_2.$$

It is expected that I_{m_1} is a better approximation to I than I_{m_2}, i.e.,

$$|I - I_{m_1}| \leq |I_{m_1} - I_{m_2}|, \quad m_1 > m_2 \tag{7}$$

so that $|I_{m_1} - I_{m_2}|$ acts as a conservative estimate of the integration error.

The use of the error estimate is usually two-fold. The subregions with the largest error estimates in absolute value will be chosen for subdivision. Besides, the (local) error estimates for each subregion can be aggregated over the whole integration region $\mathcal{C}^d$ to attain a global error estimate. The global absolute/relative error can serve as a termination criterion for subdivision. Subdivision will continue until either the global absolute/relative error falls below a level or a maximum number of function evaluations has been reached, or a combination of them.

4.2 The Genz-Malik rule

The Genz & Malik (1980) rule is a fully symmetric degree 7 rule. It gives an integration rule in the square $[-1,1]^d$ but can be readily generalized to any rectangular region by an affine transformation. The degree 7 rule is given as follows

$$\begin{aligned} I_7(f) =& u_1 f(0,0,\dots,0) + u_2 \sum_{FS} f(\lambda_2, 0, 0, \dots, 0) + u_3 \sum_{FS} f(\lambda_3, 0, 0, \dots, 0) + \\ &+ u_4 \sum_{FS} f(\lambda_4, \lambda_4, 0, 0, \dots, 0) + u_5 \sum_{FS} f(\lambda_5, \lambda_5, \dots, \lambda_5), \end{aligned} \tag{8}$$

where $\sum_{FS}$ denotes a fully symmetric summation over all permutations of coordinates including sign changes and

$$\lambda_2^2 = \frac{9}{70}, \lambda_3^2 = \lambda_4^2 = \frac{9}{10}, \lambda_5^2 = \frac{9}{19},$$

$$u_1 = 2^d(12824 - 9120d + 400d^2)/19683,$$

$$u_2 = 2^d(980/6561),$$

$$u_3 = 2^d(1820 - 400d)/19683,$$
$$u_4 = 2^d(200/19683),$$
$$u_5 = 6859/19683.$$

All integration nodes are inside the integration domain. The degree 7 integration rule requires $2^d + 2d^2 + 2d + 1$ integrand evaluations for a function of d variables and is thus known to be most advantageous for problems with $d \leq 8$. We remark that, by contrast, a Gauss-Legendre quadrature rule of degree 7 would require 4^d integration evaluations, which is significantly larger for $d \geq 3$.

Example 2. *The Genz-Malik rule in the square* $[-1, 1]^2$ *evaluates a function at the following 17 points,* $(0, 0)$*,* $(\pm\sqrt{\frac{9}{70}}, 0)$*,* $(0, \pm\sqrt{\frac{9}{70}})$*,* $(\pm\sqrt{\frac{9}{10}}, 0)$*,* $(0, \pm\sqrt{\frac{9}{10}})$*,* $(\pm\sqrt{\frac{9}{10}}, \pm\sqrt{\frac{9}{10}})$*,* $(\pm\sqrt{\frac{9}{19}}, \pm\sqrt{\frac{9}{19}})$*.*

The Genz-Malik rule distinguishes itself from other multiple integration rules in that it has an embedded degree 5 rule for error estimation. The degree 5 rule uses a subset of points of the degree 7 rule, which means that no additional integrand evaluations are necessary. This is highly desirable for multidimensional problems. The embedded degree 5 rule is given by

$$\begin{aligned} I_5(f) = &u'_1 f(0, 0, \ldots, 0) + u'_2 \sum_{FS} f(\lambda_2, 0, 0, \ldots, 0) + u'_3 \sum_{FS} f(\lambda_3, 0, 0, \ldots, 0) + \\ &+ u'_4 \sum_{FS} f(\lambda_4, \lambda_4, 0, 0, \ldots, 0), \end{aligned} \tag{9}$$

with

$$u'_1 = 2^d(729 - 950d + 50d^2)/729,$$
$$u'_2 = 2^d(245/486),$$
$$u'_3 = 2^d(265 - 100d)/1458,$$
$$u'_4 = 2^d(25/729).$$

As pointed out in the preceding section, an error approximation for each subregion is simply the difference of these two rules, i.e.,

$$\epsilon = I_7 - I_5. \tag{10}$$

Starting from the whole integration region, in every step the (sub)region with the largest error estimate in absolute value will be chosen for subdivision. The division rule used to determine along which direction to divide is due to van Dooren & de Ridder (1976). To avoid an exponential explosion in the number of subregions, the chosen region is not divided into 2^d subregions but only into two. In particular, the direction that has the largest fourth divided difference is halved. Five points are used in the direction $i = 1, \ldots, d$,

$$x_i = -\lambda_3, -\lambda_2, 0, \lambda_2, \lambda_3, \quad \text{and } x_j = 0 \text{ for } j \neq i.$$

And the fourth divided differences are given by

$$Dif_i = [f(-\lambda_3) - 2f(0) + f(\lambda_3)] - \frac{\lambda_2^2}{\lambda_3^2}[f(-\lambda_2) - 2f(0) + f(\lambda_2)]. \tag{11}$$

Note that no additional integrand evaluations are required here either.

It follows that after $K-1$ subdivisions, the integration region $\mathcal{C}^d$ is divided into K non-overlapping rectangular subregions. For any subregion k, the Genz-Malik rule gives a local integral estimate $I_7^{(k)}$, a local error estimate $\epsilon^{(k)}$ and a direction $s^{(k)}$ that has the largest fourth divided difference given by (11), which is then chosen for the next subdivision. Aggregating the local information over $\mathcal{C}^d$ we obtain a global integral estimate to $I(f)$ as follows,

$$I_7(f) = \sum_{k=1}^{K} I_7^{(k)}(f), \tag{12}$$

where $I_7^{(k)}(f)$ is calculated by (8) with suitable affine transformation. Meanwhile the K local error estimates sum to a global error estimate i.e., $\epsilon = \sum_{k=1}^{K} \epsilon^{(k)}$.

A remark is that when a region is subdivided, integrand values evaluated in this region are discarded and the integration rule is applied in both new subregions. Roughly this means that in the long run only half of the integrand evaluations is used for the calculation of the integral, the other half is wasted in the process of subdivision.

5 A tailor-made adaptive Genz-Malik rule

We restate our problem as calculating

$$I(f) = \int \cdots \int_{\mathcal{C}^d} f(\mathbf{Y^d})\phi(\mathbf{Y^d}) dY_1 \cdots dY_d, \tag{13}$$

where $f(\mathbf{Y^d}) = P(L > x|\mathbf{Y^d})$ is bounded in $[0,1]$ and $\phi(\mathbf{Y^d})$ is the probability density function of d-dimensional normal distribution with zero mean and identity covariance matrix.

A major problem with the Genz-Malik rule is that the weights u_i can be negative: $u_1 < 0$ for $2 \leq d \leq 21$ and $u_3 < 0$ for $d \geq 5$. Consequently even though our integrand is always positive in some subregions a straightforward Genz-Malik rule may give negative results for the integral. This however can be rather easily dealt with in our context. Recall from Prop. 1 that $f(\mathbf{Y^d})$ should be bounded in any rectangular (sub)region $[a_1, b_1] \times [a_2, b_2] \ldots \times [a_d, b_d]$, more specifically,

$$f(b) \leq f(\mathbf{Y^d}) \leq f(a), \tag{14}$$

where $a = (a_1, a_2, \ldots, a_d)$ and $b = (b_1, b_2, \ldots, b_d)$. As a result we have for $I^{(k)}(f)$, superscript (k) indicating subregion k, both an upper bound and a lower bound, i.e.,

$$f(b^{(k)})\prod_{i=1}^{d}\left(\Phi(b_i^{(k)})-\Phi(a_i^{(k)})\right)\le I^{(k)}(f)\le f(a^{(k)})\prod_{i=1}^{d}\left(\Phi(b_i^{(k)})-\Phi(a_i^{(k)})\right). \tag{15}$$

Denote by $\mathcal{U}^{(k)}$, $\mathcal{L}^{(k)}$ the upper bound and lower bound respectively for subregion k. Positivity of the integrand can be easily preserved by the following correction,

$$I_7^{(k)}(f)=I_7^{(k)}(f)1_{\{\mathcal{L}^{(k)}\le I_7^{(k)}(f)\le\mathcal{U}^{(k)}\}}+\mathcal{L}^{(k)}1_{\{I_7^{(k)}(f)<\mathcal{L}^{(k)}\}}+\mathcal{U}^{(k)}1_{\{I_7^{(k)}(f)>\mathcal{U}^{(k)}\}}. \tag{16}$$

The last term in Eq. (16) in addition corrects possible overshooting of the integration rule to some extent. More importantly, the local bounds over all subregions can be aggregated to a global upper bound and a global lower bound for the whole integration region $\mathcal{C}^d$. It follows that the estimate to the integral should asymptotically converge to $I(f)$ if we continue the subdivision until the global upper bound and lower bound coincide.

It is also important to recognize that the integral can be calculated exactly for subregions where the integrand is constantly 0 or 1. These subregions can be identified by simply evaluating the integrand at the end points $a^{(k)}$ and $b^{(k)}$. By bounded monotonicity we have

$$I^{(k)}(f)=\begin{cases}0 & \text{if } f(a^{(k)})=0,\\ \prod_{i=1}^{d}\left(\Phi(b_i^{(k)})-\Phi(a_i^{(k)})\right) & \text{if } f(b^{(k)})=1.\end{cases} \tag{17}$$

In these subregions we should set $\epsilon^{(k)}=0$.

We are now in a position to present our adaptive integration algorithm based on a tailor-made Genz-Malik rule. It is presented as Algorithm 2. For clarity in notation we use superscript l for local estimates in any subregion. s denotes the subdivision direction of a subregion.

The error estimate ϵ deserves further investigation. Typically it not only determines the region for subdivision in each step, but it is also used to check whether the termination criteria are met. According to Lyness & Kaganove (1976), Berntsen (1989), error estimates based on differences of two rules can in general be unreliable. The inequality (7) is not necessarily satisfied, thus it is possible that, while the actual error is very large, the estimated error is marginal. Schürer (2001) shows that in particular the Genz-Malik rule performs rather poor in terms of error estimation. Various ways of improving the reliability of error estimates can be found in Berntsen (1989), Berntsen et al. (1991), among which a simple approach is to use more than two integration rules for error estimation. Following this we take a parsimonious change by including the degree 1 midpoint rule for the square $[-1,1]^d$,

$$I_1=f(0,0,\ldots,0),$$

Algorithm 2 adaptive integration for $2 \leq d < 10$

```
Apply the GM rule over the integration region,
return $I_7^l$, $\epsilon^l$ and subdivision direction $s$, impose (16)
while termination criteria not met do
  Choose the (sub)region with largest $\epsilon^l$ and divide along direction $s$.
  Compute $f(a)$ and $f(b)$ for the resulting two subregions.
  if $f(a) = 0$ or $f(b) = 1$ then
    Apply (17), let $\epsilon^l = 0$.
  else
    Apply the GM rule to both subregions, return $I_7^l$, $\epsilon^l$ and $s$, impose (16).
  end if
  Update $I_7$, $\epsilon$ and the subregion collection.
end while
```

which is also embedded in the degree 7 rule, as a second check on error. Thus the error estimate is defined to be

$$\epsilon = (I_7 - I_5)\mathbf{1}_{\{|I_7 - I_5| \geq |I_7 - I_1|\}} + (I_7 - I_1)\mathbf{1}_{\{|I_7 - I_5| < |I_7 - I_1|\}}. \tag{18}$$

The new error estimate is more reliable but also conservative. In fact $I_7 - I_5$ is already too conservative an error estimate for the degree 7 rule since it is rather an error estimate for the degree 5 rule. A stop rule based on such absolute/relative errors can consequently be ineffective. It may well happen that while the integration rule is giving accurate results, the error estimate remains above a given precision level and the subdivision carries on more than necessary, see e.g. Genz & Kass (1997). Hence a stop rule that does not rely on ϵ is desirable. In return, when an error estimate is not used as a termination criterion for subdivision, it no longer needs to provide a good approximation for $I(f) - I_7(f)$. Instead, an error estimate is of sufficient quality if for any two subregions k_1 and k_2, $\epsilon^{(k_1)} > \epsilon^{(k_2)}$ implies

$$\left| I^{(k_1)}(f) - I_7^{(k_1)}(f) \right| > \left| I^{(k_2)}(f) - I_7^{(k_2)}(f) \right|$$

with a great probability. In this sense (18) is likely to outperform (10) as the former will more often than not magnify the error estimate for subregions in which the integrand varies substantially but provides little change in smooth subregions.

An alternative termination criterion is suggested in Genz & Kass (1997). They propose to compare the integral estimate after every h subdivision and the algorithm is stopped when there is negligible change in successive results. This is not difficult to understand. We have a converging sequence of integral estimates in the process of subdivision. For $K' < K$, the relative error estimate $|\hat{I}_{K'} - \hat{I}_K|/\hat{I}_K$ provides an indication whether the approximation has converged. When the changes become sufficiently small it is reasonable to stop further division. A similar criterion is also adopted by Paskov & Traub (1995) to control the generation of quasi-Monte Carlo sequences. The algorithm however has the disadvantage that it can only be stopped after a multiple of h divisions. To make the stop rule more flexible it is possible to define a moving window with bandwidth h and replace $|I_K - I_{K-h}|$ by

$$\max_{K-h \le K' < K} |I_K - I_{K'}| ,$$

where K need not to be a multiple of h. In this way the algorithm can be terminated for all $K > h$, and is also somehow more robust since the maximum of the differences is used.

Variants of the stop rule can be further explored. Unfortunately these stop rules share a common disadvantage that a practical error bound is lacking. Therefore we would rather stay with the simplest termination criterion: maximum number of integrand evaluations or similarly, maximum number of subdivisions. This also saves us from the delicate problem of choosing suitable parameters such as h, which can be quite arbitrary.

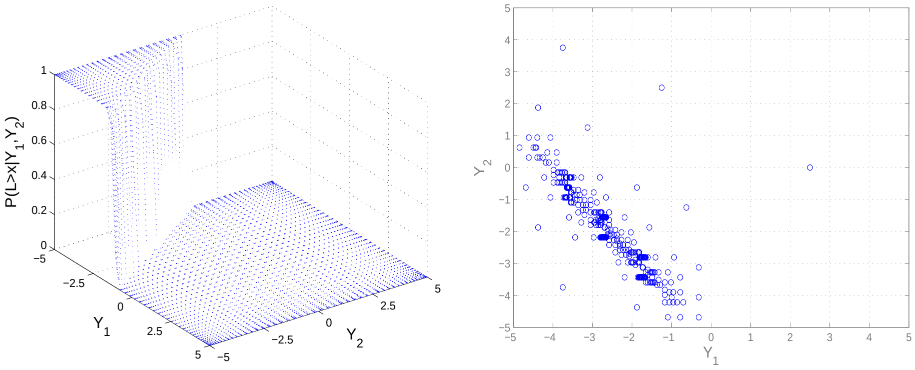
6 Numerical results I

Here we first illustrate by a two-factor model example how the adaptive integration algorithm works. For some arbitrary portfolio and suitable loss level x, Figure 2(a) gives the conditional tail probability $P(L > x|Y_1, Y_2)$ for (Y_1, Y_2) truncated to the square $[-5, 5]^2$. The integrand turns out to contribute nothing to the integral value in almost 7/8 of the area, which suggests that an adaptive algorithm should be favored. Figure 2(b) shows a scatterplot of the subregion centers generated by the adaptive integration algorithm 2. It is clearly seen that the adaptive algorithm does focus its integrand evaluation in the subregions where the integrand values vary rapidly.

Let us consider again credit portfolio A with 1000 obligors with $w_i = 1$, $p_i = 0.0033$, $i = 1, \dots, 1000$. However, we now move to a five-factor model such that the obligors are grouped into 5 buckets of 200 obligors. Within each bucket, the obligors have identical factor loadings

$$\alpha_i = \begin{cases} \left(\frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}\right), i = 1, \dots, 200, \\ \left(\frac{1}{\sqrt{5}}, \frac{1}{\sqrt{5}}, \frac{1}{\sqrt{5}}, \frac{1}{\sqrt{5}}, 0\right), i = 201, \dots, 400, \\ \left(\frac{1}{\sqrt{4}}, \frac{1}{\sqrt{4}}, \frac{1}{\sqrt{4}}, 0, 0\right), i = 401, \dots, 600, \\ \left(\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, 0, 0, 0\right), i = 601, \dots, 800, \\ \left(\frac{1}{\sqrt{2}}, 0, 0, 0, 0\right), i = 800, \dots, 1000. \end{cases}$$

We compute tail probabilities over a wide range of 20 loss levels from 75 to 550, with an increment of 25. These losses correspond to quantiles of the portfolio loss distribution roughly from 99% to 99.99%. As a benchmark we



(a) integrand $P(L > x|Y_1, Y_2)$

(b) scatterplot subregion centers

Figure 2: Adaptive integration for a two-factor model. (a) integrand $P(L > x|Y_1, Y_2)$; (b) centers of the subregions generated by adaptive integration.

use simulation with a tremendous amount of scenarios. Integrand evaluation is accomplished by the normal approximation and is considered to be exact. We compare the results obtained by the adaptive Genz-Malik rule (ADGM), the Monte Carlo (MC) and quasi-Monte Carlo (QMC) methods with a similar number of integrand evaluations, denoted by N. For the quasi-Monte Carlo method we choose the SOBOL sequence. The sequence is generated by the GSL library, which is based on Antonov & Saleev (1979).

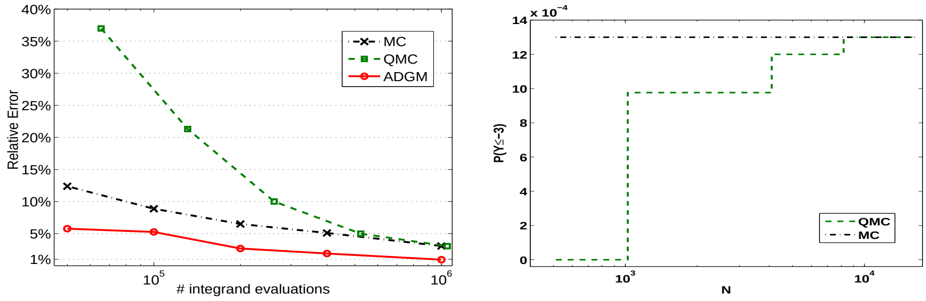
We control the number of integrand evaluations rather than computation time in the course of subdivision because the latter can vary substantially for different portfolios, different methods for integrand evaluation and different data structures of the subregion collection. The approximation error is measured by the unsigned relative error (RE) defined as

$$\frac{|\hat{I}(f) - I(f)|}{I(f)},$$

where $I(f)$ is the result given by the benchmark and $\hat{I}(f)$ denotes any estimate to $I(f)$. The relative errors reported for the Monte Carlo method are averaged over 100 different runs. Alongside the mean error we also show the 95% confidence interval of the signed relative error. Since Monte Carlo simulation gives an unbiased estimate, the confidence interval is simply ±1.96 times of the error standard deviation.

We first show in Figure 3(a) the average performance of each method over all 20 loss levels with different numbers of integrand evaluations N ranging from $50,000$ to 2^{20}. Note that for the adaptive integration, these correspond roughly to K, the number of subregions, from 250 to $5,000$ because the Genz-Malik rule samples in five dimensions around 100 points in each subregion. Apparently the adaptive integration consistently outperforms both Monte Carlo and the quasi-Monte Carlo methods for all levels of N. Its relative errors are around one-third of those obtained from plain Monte Carlo given the same amount of integrand evaluations. Contrary to common knowledge, the quasi-Monte Carlo method is here inferior to the Monte Carlo method, especially for low $N \leq 2^{18}$. The quasi-Monte Carlo method does show a higher convergence rate than the plain Monte Carlo method, but that is only because it is so deviant for small N. For example, the average relative error of quasi-Monte Carlo with $65,536$ $(= 2^{16})$ evaluations is more than 35%, while plain Monte Carlo and the adaptive integration with around $50,000$ evaluations yield 12.4% and 5.8% respectively.

The poor performance of the quasi-Monte Carlo method, especially for small N, is not unexpected. Let's go back to the one-factor model example in section 3 and consider the binary van der Corput sequence, a one-dimension quasi-Monte Carlo sequence, with the total number of points $N = 2^s - 1$ for some $s \in Z^+$. We concentrate on the interval $Y \leq -3$ as this is the interval where the integrand makes the most contribution according to Figure 1. In Figure 3(b) we show for $9 \leq s \leq 14$ the percentages of quasi-Monte Carlo points falling in this interval compared to Monte Carlo points. Note the more points are in this interval, the more accurate the integration result will be. The percentage with the Monte Carlo method is constantly $\Phi^{-1}(-3) = 0.0013$ by expectation. The van der Corput sequence has no points at all in the interval with $s = 9$ and only catches up with the Monte Carlo method for $s \geq 13$.



(a) (b)

Figure 3: (a) Estimation relative errors of the adaptive Genz-Malik rule (ADGM), Monte Carlo (MC) and quasi-Monte Carlo (QMC) methods over 20 loss levels. The number of total integrand evaluations N ranges from $50,000$ to 2^{20}. (b) The percentages of Monte Carlo (MC) and quasi-Monte Carlo (QMC) points that fall in the interval $Y \leq -3$. The total number of points is $N = 2^s - 1$ with $9 \leq s \leq 14$.

With around $N = 10^6$ evaluations, it seems that all three methods produce satisfactory results. Relative errors are, respectively, 0.9% (ADGM), 3.0% (QMC) and 3.1% (MC). Figure 4 further compares the performance of the different methods with around 10^6 evaluations for various loss levels. Monte Carlo and quasi-Monte Carlo methods are quite accurate for low loss levels but deteriorate notably as the loss level increases. An upward trend in the relative error is conspicuous for both methods. In particular, for the loss level $x = 550$, Monte Carlo has an error 8.8% and quasi-Monte Carlo gives 12.8%. By contrast, the relative error of the adaptive integration for the same loss level is merely 0.5%. Even though at some low loss levels adaptive integration is not superior to the other two methods, it dominates its two opponents for loss levels larger than 300. The adaptive integration is remarkably distinct from Monte Carlo and quasi-Monte Carlo methods in that it is not sensitive to the portfolio loss level of interest. As a consequence, the adaptive integration becomes more and more advantageous compared to Monte Carlo and quasi-Monte Carlo methods for increasing loss levels. This is especially attractive for the purpose of determining the portfolio VaR, which always involves large loss levels.

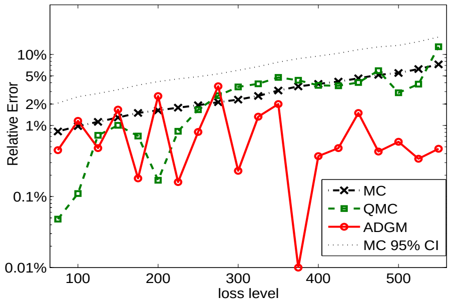
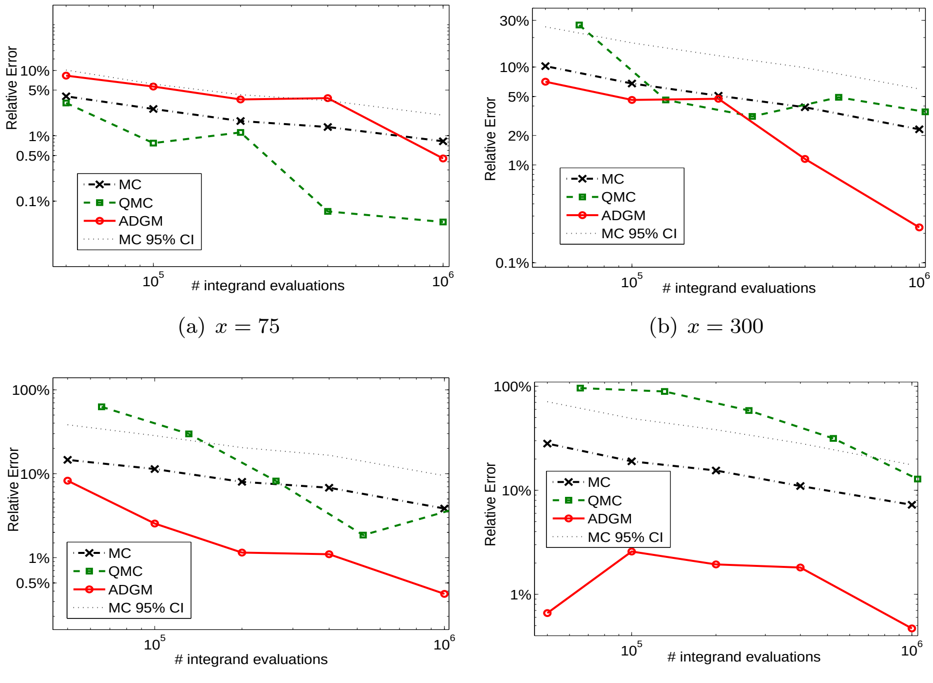


Figure 4: Estimation relative errors of the adaptive Genz-Malik rule (ADGM), Monte Carlo (MC) and quasi-Monte Carlo (QMC) methods with around $N = 10^6$ evaluations for various loss levels.

A close-up look to the three methods for different loss levels is presented in Figure 5. We show results for four loss levels, $x = 75$, 300, 400, 550, which correspond to roughly quantiles 99%, 99.9%, 99.95% and 99.99%, respectively. Two issues of the adaptive integration need to be addressed. First, increasing the number of subregions K, generally improves the quality of an approximation but it is not guaranteed. Second, as the accuarcy of an approximation is not sensitive to the portfolio loss level, a stop rule seems to be less important. For example, any K between $1,000$ and $2,000$ can serve as a reasonable termination criterion.

7 Adaptive Monte Carlo integration

We have shown that adaptive integration based on the Genz-Malik rule provides an efficient tool for calculating credit portfolio loss distribution in a multi-factor framework. It is particularly advantageous in the tail of the loss distribution. However the adaptive Genz-Malik rule still suffers from two problems. First, the integration rule is only able to handle models with relatively low dimension, say $d \leq 8$. This is due to the fact that the number of integrand evaluations is fully



(a) $x = 75$

(b) $x = 300$

(c) $x = 400$

(d) $x = 550$

Figure 5: Relative estimation error of $P(L > x)$ by all methods for four different loss levels x. PD$= 0.0033$, $\rho = 0.2$, $d = 5$.

determined by d and grows exponentially. Second, no practical error bounds are available for the estimates. The second problem also applies to other multiple integration techniques such as quasi-Monte Carlo methods and sparse grids.

A natural alternative that does not suffer from the above two problems is Monte Carlo integration. A Monte Carlo integration *embedded in a globally adaptive algorithm* is able to provide an unbiased estimate of the integral and also probabilistic error bounds for the estimate. In the mean-time it has higher accuracy and faster convergence than the plain Monte Carlo integration. The idea of adaptive Monte Carlo integration is not new. Two well-known algorithms can be found in Press & Farrar (1990) and Lepage (1990, 1980). It has however, to our knowledge, never been used in the context of credit portfolio loss modeling.

Our adaptive Monte Carlo integration replaces the degree 7 Genz-Malik rule with uniform random numbers as the integration rule. Let us go back to Eq. (13) and write $\xi = f \cdot \phi$. The tail probability as in Eq. (12) can then be approximated by

$$\hat{I}(\xi) = \sum_{k=1}^{K} \hat{I}^{(k)}(\xi) = \sum_{k=1}^{K} v^{(k)} \sum_{j=1}^{M} \frac{\xi_j^{(k)}}{M}, \tag{19}$$

where K is the number of subregions, M is number of points in each subregion and $v^{(k)}$ denotes the volume of subregion k. This estimate $\hat{I}(\xi)$ is unbiased since it is an independent sum of unbiased Monte Carlo estimates. The variance of $\hat{I}(\xi)$ is given by

$$Var\left(\hat{I}(\xi)\right) = \sum_{k=1}^{K} Var\left(v^{(k)} \sum_{j=1}^{M} \frac{\xi_j^{(k)}}{M}\right) = \sum_{k=1}^{K} \frac{\left(v^{(k)}\right)^2}{M} Var\left(\xi^{(k)}\right), \tag{20}$$

where $Var\left(\xi^{(k)}\right)$ can be estimated from the simulated sample. If we use the unbiased version of sample variance for each subregion, Eq. (20) gives an unbiased estimate as well.

Additionally an upper bound for the variance can be derived. Recall that for any subregion both an upper bound and a lower bound for the integral are available. We denote them by $\mathcal{U}^{(k)}$, $\mathcal{L}^{(k)}$ respectively for subregion k and let $\delta^{(k)} = \mathcal{U}^{(k)} - \mathcal{L}^{(k)}$. It is immediate to see that

$$Var\left(\xi^{(k)}\right) = E\left(\xi^{(k)} - E\left(\xi^{(k)}\right)\right)^2 \leq E\left(\mathcal{U}^{(k)} - \mathcal{L}^{(k)}\right)^2 = \left(\delta^{(k)}\right)^2.$$

It follows that the upper bound for the variance is

$$Var\left(\hat{I}(\xi)\right) \leq \sum_{k=1}^{K} \frac{\left(v^{(k)}\delta^{(k)}\right)^2}{M}. \tag{21}$$

To reduce the variance we minimize its upper bound. This is achieved by simply choosing in each step the subregion with the largest $v\delta$ for subdivision. A large $v\delta$ generally implies a large variance, but the converse does not hold due to simulation noise in the sample variance, esp. for small M. In this sense it is more robust to rely on $v\delta$ than on the estimated variance. In particular, given any collection of subregions, the subregion chosen for the next subdivision is deterministic and requires no simulation at all. Furthermore, the upper bound of variance given by (21) is strictly decreasing in the process of subdivision but this is not necessarily the case for the estimated variance. Similar to the adaptive Genz-Malik rule, the integral estimate should asymptotically converge to $I(f)$ if we continue the subdivision until the upper bound and lower bound of $I(f)$ in $\mathcal{C}^d$ coincide and the variance vanishes.

We still need a subdivision rule replacing the fourth divided differences as in (11), since simulated samples cannot be fully symmetric. Consider a subregion centered at the origin. Let y_i denote the element of y in direction i. A convenient substitute in the spirit of divided differences is the following

$$Dif_i = \left|\sum \xi(y) \left(\mathbf{1}_{\{y_i>0\}} - \mathbf{1}_{\{y_i<0\}}\right)\right|, \tag{22}$$

if along each direction, a simulated sample always contains the same number of points in the positive and negative axes. For this purpose we generate random numbers *antithetically* rather than randomly. Since antithetic variates are no longer independent, the variance estimated needs a slight modification. Suppose that ξ and $\bar{\xi}$ are obtained from antithetic pairs, then the variance should be estimated by $M/2$ pairs of averaged antithetic pairs $(\xi+\bar{\xi})/2$, i.e.,

$$Var\left(\hat{I}(\xi)\right)=\sum_{k=1}^{K}\frac{\left(v^{(k)}\right)^2}{M/2}Var\left(\frac{\xi^{(k)}+\bar{\xi}^{(k)}}{2}\right). \tag{23}$$

It is well-known that the variance is reduced by antithetic variates if $\xi(y)$ is monotonic. In our case ξ is a product of a monotonic function f and a unimodal density ϕ and it is thus not monotonic in the whole integration region $\mathcal{C}^d$. As a consequence variance reduction is not theoretically guaranteed, although it is usually found to be achieved in most subregions. This is mainly due to the local monotonicity of ξ.

We should now be able to summarize the algorithm of adaptive Monte Carlo integration for the calculation of tail probability in a multi-factor credit portfolio loss model. This is presented as Algorithm 3. Note that constraint (15) used in Algorithm 2 is dropped to ensure that (19) gives an unbiased estimate.

Algorithm 3 adaptive Monte Carlo integration

```
Generate $M$ antithetic uniform random variables over the integration region,
return $I^l$, $Var(I^l)$, $v^l$, $\delta^l$ and subdivision direction $i_s$
while termination criteria not met do
  Choose the (sub)region with largest $v^l\delta^l$ and divide along direction $i_s$.
  Compute $f(a)$ and $f(b)$ for the resulting two subregions.
  if $f(a)=0$ or $f(b)=1$ then
    Apply (17), let $Var(I^l)=\delta^l=0$.
  else
    Generate $M$ antithetic uniform random variables in both subregions,
    return $I^l$, $v^l$, $\delta^l$ and $i_s$.
  end if
  Update $I$, $Var(I)$ and the subregion collection.
end while
```

Unlike the deterministic Genz-Malik rule, Monte Carlo integration allows flexibility in the choice of M, the number of sample points in each subregion. Suppose we would like to double the total number of integrand evaluations, we can double either M or K.

If we double M, then the new variance is

$$Var\left(\hat{I}(\xi)\right)=\sum_{k=1}^{K}Var\left(v^{(k)}\sum_{j=1}^{2M}\frac{\xi_j^{(k)}}{2M}\right)=\sum_{k=1}^{K}\frac{\left(v^{(k)}\right)^2}{2M}Var\left(\xi^{(k)}\right), \tag{24}$$

the convergence rate is thus $O\left(1/\sqrt{N}\right)$, the same as the plain Monte Carlo simulation.

It is more difficult to derive a variance estimate if we keep M unchanged and double K. The new collection of subregions can vary from problem to problem since the subregion for subdivision is chosen dynamically in each subdivision step. The worst case in terms of error convergence seems to be that all subregions are divided exactly once regardless of their v and δ. Suppose subregion k is divided into two subregions k_1 and k_2. Its new variance becomes

$$\begin{aligned} Var\left(\hat{I}^{(k)}(\xi)\right) &= \frac{\left(v^{(k_1)}\right)^2}{M} Var\left(\xi^{(k_1)}\right) + \frac{\left(v^{(k_2)}\right)^2}{M} Var\left(\xi^{(k_2)}\right) \\ &= \frac{\left(v^{(k)}\right)^2}{4M}\left[Var\left(\xi^{(k_1)}\right) + Var\left(\xi^{(k_2)}\right)\right] \\ &= \frac{\left(v^{(k)}\right)^2}{2M}\left[Var\left(\xi^{(k)}\right) - \frac{1}{4}\left(E\xi^{(k_1)} - E\xi^{(k_2)}\right)^2\right]. \end{aligned} \tag{25}$$

The last equality shows that subdivision gives an error convergence rate at least $1/\sqrt{N}$. This is also a standard result on stratified sampling, see e.g., Glasserman et al. (1999). It follows that

$$Var\left(\hat{I}(\xi)\right) \leq \sum_{k=1}^{K} \frac{\left(v^{(k)}\right)^2}{2M} Var\left(\xi^{(k)}\right). \tag{26}$$

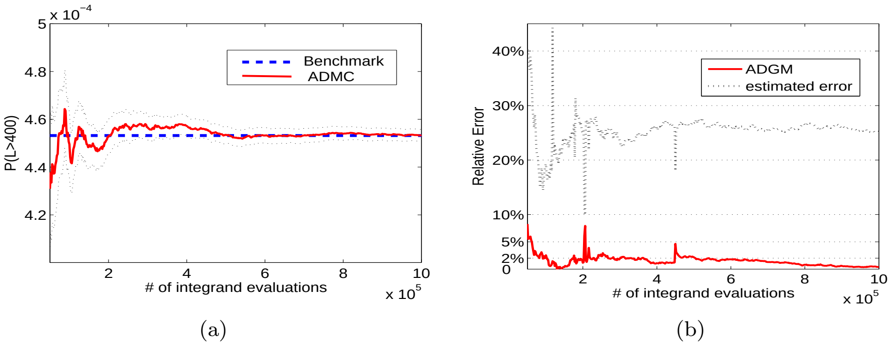
We emphasize that by employing a subdivision rule based on (22) we always divide a region along the direction that is expected to give the largest $\left(E\xi^{(k_1)} - E\xi^{(k_2)}\right)^2$, hence leading to maximal variance reduction. Consequently, increasing the number of subregions is always more favorable than increasing the number of samples in all subregions.

We should finally remark that, in terms of accuracy, the adaptive Monte Carlo integration is not necessarily inferior to the adaptive algorithm based on fully symmetric interpolation rules like the Genz-Malik rule, although the latter is supposed to provide more accurate approximation for smooth integrands. Since, with a fixed number of samples $N = MK$, the adaptive Monte Carlo integration may choose an M much less than the samples required for the Genz-Malik rule and may therefore obtain many more subregions K.

8 Numerical results II

We continue our numerical experiments with the five-factor model for portfolio A in section 6 and compare adaptive Monte Carlo integration to plain Monte Carlo integration. Rather than the relative error, we report the standard deviation normalized by benchmark, i.e., $\sqrt{Var(\hat{I}(\xi))/I}$.

Figure 6(a) shows the estimated tail probability for the loss level $x = 400$ by adaptive Monte Carlo integration along with the corresponding 95% confidence interval. It is evident that the adaptive Monte Carlo integration indeed gives a convergent estimate with reliable error bounds. By contrast, the error estimate given by adaptive Genz-Malik rule (based on Eq. (10)) can be less reliable. It is shown by Figure 6(b) that for the same loss level, although the relative error of the tail probability estimate given by the adaptive Genz-malik rule is only around 2%, the estimated error by Eq. (10) is more than 20%.



(a)

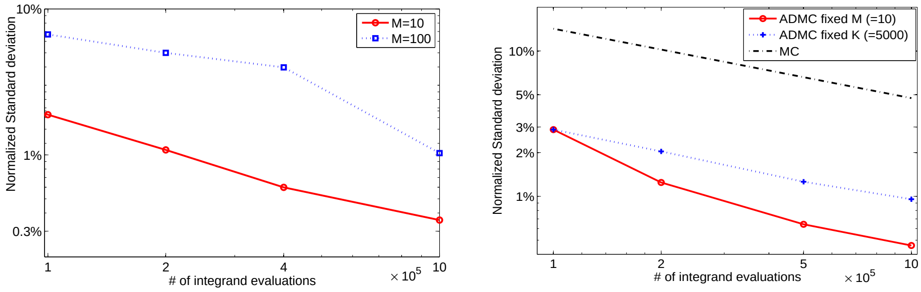
(b)

Figure 6: (a) Tail probability $P(L > 400)$ computed by adaptive Monte Carlo integration and their corresponding 95% confidence intervals (dotted lines). The dashed line is our Benchmark. (b) Relative errors of the adaptive Genz-Malik rule for $P(L > 400)$ compared to its associated error estimates (dotted lines) based on Eq. (10). The number of integrand evaluations ranges from $50,000$ to 10^6.

We have pointed out that Monte Carlo integration allows flexibility in the choice of M, the number of sample points in each subregion. Figure 7(a) compares performance of the adaptive Monte Carlo integration with $M = 10$ and $M = 100$. It confirms that with a fixed number of samples $N = MK$, a large K is favored over a large M. In Figure 7(b) we present the error convergence of adaptive Monte Carlo integration by doubling M and doubling K. We also include the error convergence rate of plain Monte Carlo method for reference. The error convergence rate by doubling M and with K fixed at 5000 is similar to the plain Monte Carlo method, whereas doubling K and fixing $M = 10$ displays a better convergence than $1/\sqrt{N}$. These results are in line with our analysis in the previous section. After all, adaptive Monte Carlo integration consistently outperforms plain Monte Carlo integration in terms of standard deviation.

We further demonstrate in Figure 8 the performance of the adaptive Monte Carlo integration with $M = 10$ for four different loss levels as in section 6. It comes with no surprise that, just like the adaptive Genz-Malik rule, the adaptive Monte Carlo integration is not sensitive to the portfolio loss level. At the loss level $x = 300$, the adaptive Monte Carlo integration with around 50 thousand integrand evaluations is already comparable to the plain Monte Carlo integration with 1 million integrand evaluations, which is a reduction of a factor of 20.

Finally, we would like to point out that the grid generated by the adaptive Monte Carlo integration may also provide a good basis for the calculation of the marginal VaR contributions (VaRC), i.e., $w_i E(D_i|L = x)$. As an example we



(a)

(b)

Figure 7: Adaptive Monte Carlo integration for loss level $x = 400$. (a) Normalized standard deviation obtained from Adaptive Monte Carlo integration with $M = 10$ and $M = 100$; (b) Error convergence of adaptive Monte Carlo integration by doubling M and doubling K.

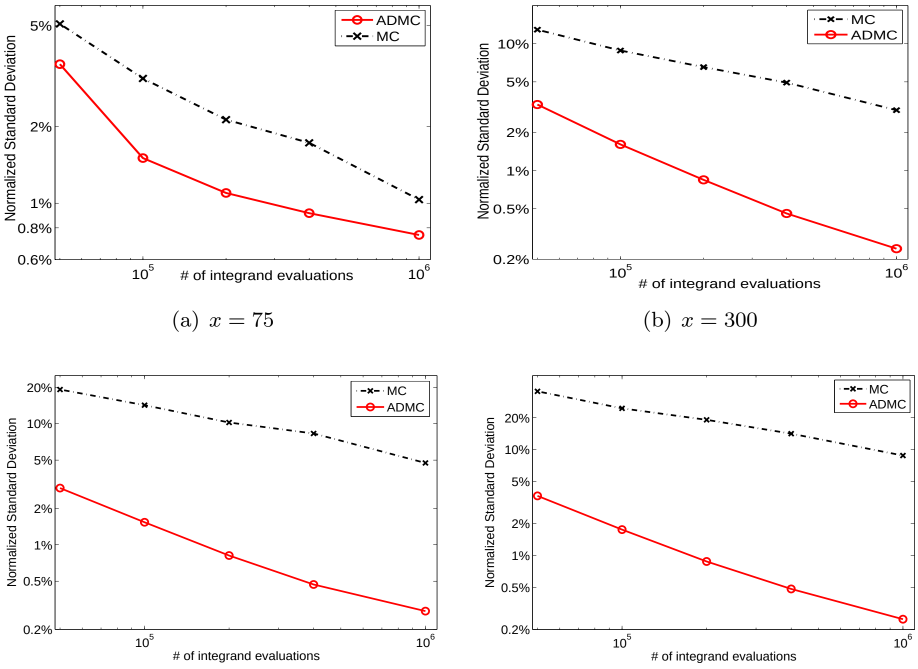
present in Table 1 the VaRC of the obligors in different buckets for the loss level $x = 300$. The estimates obtained from the adaptive Monte Carlo integration are based on 50 thousand integrand evaluations. The standard deviations (std) are calculated with 20 independent trials and in parentheses are the standard deviations as a percentage of their corresponding benchmark. Both the VaRC estimates and standard deviations are similar to those given by plain Monte Carlo integration with 1 million integrand evaluations. This is in line with the performance regarding the tail probability.

bucket	Benchmark	MC	std	ADMC	std
1	0.4331	0.4258	0.0239 (5.5%)	0.4293	0.0163 (3.8%)
2	0.4498	0.4504	0.0141 (3.1%)	0.4489	0.0127 (2.8%)
3	0.3467	0.3526	0.0167 (4.8%)	0.3475	0.0120 (3.5%)
4	0.2022	0.2037	0.0129 (6.4%)	0.2076	0.0157 (7.8%)
5	0.0683	0.0676	0.0089 (13.0%)	0.0667	0.0069 (10.1%)

Table 1: The VaR contributions of the obligors in different buckets for the loss level $x = 300$. The adaptive Monte Carlo (ADMC) integration uses 50 thousand integrand evaluations and plain Monte Carlo (MC) integration uses 1 million integrand evaluations. The standard deviations (std) are calculated with 20 independent trials and in parentheses are the standard deviations normalized by benchmark.

9 Conclusions

In this article we proposed algorithms of adaptive integration for the calculation of the tail probability in multi-factor credit portfolio loss models. The



(a) $x = 75$

(b) $x = 300$

(c) $x = 400$

(d) $x = 550$

Figure 8: Standard deviations of the tail probability estimates given by plain Monte Carlo (MC) and Adaptive Monte Carlo (ADMC) for four loss levels. Standard deviations are reported as a percentage of the respective tail probabilities. For plain MC standard deviations are computed based on 100 independent runs of simulation and for Adaptive MC, standard deviations are estimated by Eq. (20.)

problem is important as the tail probabilities are essential for the determination of the portfolio VaR. We showed that under mild conditions, the conditional tail probability, as a function of the common factors, is monotone and differentiable. Starting with an algorithm in one dimension, we devise an adaptive Genz-Malik rule suitable for portfolio credit models with a number of factors $2 \leq d \leq 8$. The algorithm based on the Genz-Malik rule is asymptotically convergent and particularly attractive for large loss levels. An adaptive algorithm differs fundamentally from Monte Carlo or quasi-Monte Carlo methods in that it successively divides the integration region into subregions, detects the difficult subregions for integration, i.e., those where the integrand is most irregular, and places more points in those subregions. It consistently outperforms the plain Monte Carlo and quasi-Monte Carlo methods in terms of approximation error. Finally we arrive at the adaptive Monte Carlo integration, which simply replaces the Genz-Malik rule by pseudo-random numbers. The algorithm is advantageous in that it can handle higher-dimensional models and is able to provide reliable probabilistic error bounds. The error convergence rate of the adaptive Monte Carlo integration is shown to be at worst $O\left(1/\sqrt{N}\right)$. In summary, especially for higher-dimensional problems the adaptive Monte Carlo method seems the clear favorite, whereas for lower-dimensional problems both adaptive methods, the deterministic and the Monte Carlo version, work very well.

References

Andersen, L., Sidenius, J. & Basu, S. (2003), 'All your hedges in one basket', *RISK* (November), 67–72.

Antonov, I. A. & Saleev, V. M. (1979), 'An economic method of computing LP$_\tau$-sequences', *USSR Computational Mathematics and Mathematical Physics* **19**, 252–256.

Berntsen, J. (1989), 'Practical error estimation in adaptive multidimensional quadrature routine', *Journal of Computational and Applied Mathematics* **25**(3), 327–340.

Berntsen, J., Espelid, T. O. & Genz, A. (1991), 'An adaptive algorithm for the approximate calculation of multiple integrals', *ACM Transactions on Mathematical Software* **17**(4), 437–451.

Cools, R. (1999), 'Monomial cubature rules since Stroud: A compilation - part 2', *Journal of Computational and Applied Mathematics* **112**(1-2), 21–27.

Cools, R. (2002), 'Advances in multidimensional integration', *Journal of Computational and Applied Mathematics* **149**(1), 1–12.

Cools, R. & Rabinowitz, P. (1993), 'Monomial cubature rules since Stroud: A compilation', *Journal of Computational and Applied Mathematics* **48**(3), 309–326.

Genz, A. & Kass, R. (1997), 'Subregion adaptive integration of functions having a dominant peak', *Journal of Computational and Graphical Statistics* **6**(1), 92–111.

Genz, A. & Malik, A. (1980), 'An adaptive algorithm for numerical integration over an n-dimensional rectangular region', *Journal of Computational and Applied Mathematics* **6**(4), 295–302.

Glasserman, P., Heidelberger, P. & Shahabuddin, P. (1999), 'Asymptotically optimal importance sampling and stratification for pricing path-dependent options', *Mathematical Finance* **9**(2), 117–152.

Gupton, G., Finger, C. & Bhatia, M. (1997), *CreditMetrics – Technical document*, J.P.Morgan New York.

Huang, X., Oosterlee, C. W. & van der Weide, J. A. M. (2007), 'Higher order saddlepoint approximations in the Vasicek portfolio credit loss model', *Journal of Computational Finance* **11**(1), 93–113.

Lepage, G. P. (1980), Vegas: An adaptive multi-dimensional integration routine., Technical Report CLNS-80/447, Newman Laboratory of Nuclear Studies, Cornell University.

Lepage, G. P. (1990), 'A new algorithm for adaptive multidimensional integration', *Journal of Computational Physics* **27**, 192–203.

Lyness, J. N. & Kaganove, J. J. (1976), 'Comments on the nature of automatic quadrature routines', *ACM Transactions on Mathematical Software* **2**(1), 65–81.

Martin, R. (2004), *Credit portfolio modeling handbook*, Credit Suisse Financial products.

Paskov, S. & Traub, J. (1995), 'Faster valuation of financial derivatives', *The Journal of Portfolio Management* **22**(1), 113–120.

Press, W. H. & Farrar, G. R. (1990), 'Recursive stratified sampling for multidimensional Monte Carlo integration', *Computers in Physics* **4**(2), 190–195.

Schürer, R. (2001), High-dimensional numerical integration on parallel computers, Master's thesis, University of Salzburg, Austria.

Spanier, J. & Maize, E. (1994), 'Quasi-random methods for estimating integrals using relatively small samples', *SIAM Review* **36**(1), 18–44.

Stroud, A. (1971), *Approximate Calculation of Multiple Integrals*, Prentice Hall.

van Dooren, P. & de Ridder, L. (1976), 'An adaptive algorithm for numerical integration over an n-dimensional cube', *Journal of Computational and Applied Mathematics* **2**(3), 207–217.

Vasicek, O. (2002), 'Loan portfolio value', *RISK* (December), 160–162.