Pricing CDOs and Other Credit Derivatives in Multifactor Models

Sira Suchintabandid

Submitted in partial fulfillment of the Requirement for the degree of Doctor in Philosophy under the Executive Committee of the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY
2007
UMI Number: 3266682

Copyright 2007 by
Suchintabandid, Sira

All rights reserved.

INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

UMI

UMI Microform 3266682
Copyright 2007 by ProQuest Information and Learning Company.
All rights reserved. This microform edition is protected against unauthorized copying under Title 17, United States Code.

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Abstract

Pricing CDOs and Other Credit Derivatives in Multifactor Models

Sira Suchintabandid

We develop numerical methods for measuring credit risks and pricing credit basket derivatives, such as the increasingly popular collateralized debt obligations (CDOs), which offer a way to create new classes of securities based on multi-name portfolios. A key issue in the modelling of credit portfolios is how to capture the dependence among the defaults of obligors in the portfolios. In practice, the correlation among obligors are often modelled by factor structures, with the single-factor structure being the most convenient, albeit unrealistic and too simplistic to explain the skew in the implied correlation curves. Once we venture into the more realistic case of multifactor models, the difficulty becomes the computation of model outputs, as traditional methods often require computing time that is exponential in the number of factors. The purpose of this work is to develop pricing methods that are non-simulation based, and are less sensitive to the number of factors. We work within the Normal Copula model, which is the industry standard, and propose two methods for approximating CDO prices and portfolio loss distributions. The first approach is based on a power series expansion in a parameter that scales the correlation among obligors; we express the CDO price in a multifactor model as a series of prices in independent-obligor models. Thus, pricing in a model with correlated defaults is reduced to calculations involving only independent defaults. The second approach is based on the Laplace inversion method, which has become a popular tool for pricing credit derivatives. Due to the lack of explicit formulae for the Laplace transform of portfolios with dependent obligors, one faces the challenge of approximating said transforms efficiently and accurately. We develop a closed-form approximant for the Laplace transforms that is robust to the number of factors; then, we use the approximant along with the readily available Laplace inversion formulae to approximate the loss distribution, or the derivative prices. We also prove some results relating to the speed and accuracy of both approaches.
Contents

1 Background 1
   1.1 Introduction ................................................................. 1
   1.2 Background on CDOs ...................................................... 3
       1.2.1 Basic Structure ..................................................... 3
       1.2.2 Portfolio Loss Approach in Pricing CDOs ............. 6
   1.3 Normal Copula Model for Credit Risk ............................. 7

2 Approximation by Correlation Expansions 10
   2.1 The Perturbation Formula .............................................. 11
   2.2 Computing Gaussian Joint Probabilities ......................... 13
   2.3 Application in CDO Pricing: Weak Correlation Case ......... 19
   2.4 Extension to Strong Correlation Case ............................. 25
   2.5 Sensitivity Analysis .................................................... 35
       2.5.1 Sensitivity with respect to correlation .................. 37
       2.5.2 Sensitivity with respect to default probability ...... 37
   2.6 Concluding Remarks ................................................... 39

3 Quadratic Transform Approximation 40
   3.1 Pricing through Laplace Transform ............................... 41
   3.2 Approximating $\phi(s)$ .................................................... 44
       3.2.1 Approximation Method ......................................... 45
       3.2.2 Geometrical Interpretation ................................... 49
       3.2.3 Convergence Theorem ......................................... 52

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Chapter 1

Background

1.1 Introduction

The market for multi-name (portfolio) credit derivatives has been one of the fastest growing derivatives markets over the past decade. While credit default swaps have facilitated trading in individual sources of credit risk, collateralized debt obligations (CDOs) and other multi-name credit derivatives have provided new mechanisms for the transfer of credit risk in an entire portfolio.

An important aspect of the valuation and risk management of multi-name credit derivatives is the modelling of dependence among sources of credit risk. On one hand, risk models that ignore this dependence and presuppose that obligors default independently are very tractable, but they omit a crucial feature of a portfolio view of credit risk. On the other hand, capturing the dependence among obligors comes at a price of increased complexity, both in the modelling and in the computation of the model's output, be it the price of a CDO, the value-at-risk of a portfolio, or the price of some other basket credit derivative.

Factor models of dependence — in which defaults become independent conditional on a set of underlying factors — lie at an intermediate level of complexity between independent-obligor models and models that allow arbitrary dependence. In particular, in a single-factor model, CDO prices can be computed efficiently using semi-analytical methods and numerical integration (see, for example, Andersen, Sidenius, and Basu...
2003, and Laurent and Gregory 2003). However, the single factor structure is too simplistic to be credible in risk management. More specifically, the skew in the implied correlation curves clearly contradicts the single-factor assumption. The purpose of this work, therefore, is to address models in which the number of underlying factors is substantially smaller than the number of names in the portfolio, but not so small as to allow use of the types of semi-analytic methods used in the single-factor models. We focus on the problem of valuing credit risks and pricing credit basket derivative, using alternative methods to the traditional use of numerical integration or Monte Carlo simulation.

We work within the Normal Copula framework, which is the industry standard, and develop two methods of approximating CDO tranche prices (or other credit basket derivatives) under the multifactor assumption. In the first approximation method, we express a CDO tranche price in a dependent-obligor model as a series of easy-to-compute prices in independent-obligor models. To be more specific, we scale the obligors’ correlation matrix and expand the desired tranche price as a power series of the scaling parameter. We then show that each term in this expansion can be expressed as a weighted finite sum of independent-obligor prices. Thus, approximating prices in a multifactor model reduces to a series of calls to an independent-obligor pricing routine. This method takes full advantage of the fact that independent-obligor prices are quick and easy to compute. We also show how the proposed approximation can be applied to compute quantities related to sensitivity analysis.

The second approximation scheme is based on the Laplace inversion method, which has been enjoying popularity (see, for example, Laurent and Gregory (2003), Gregory and Laurent (2004), Lord and Kahl (2006), etc.) as a tool for pricing credit derivatives in the single-factor models. Once we move on to the multifactor models, however, computing the Laplace transform of a pool of obligors becomes a challenge: the transform can be expressed as an integral in multidimensional space, depending on the number of risk factors, which is problematic because computing time increases exponentially with the number of factors. Our method is specifically designed to avoid this problem; rather than using traditional methods such as numerical integration or Monte-Carlo simulation to evaluate the multidimensional integral, we approximate the integrand in such a way.
that the integral simplifies itself to a closed form. As a result, we show that the Laplace transform of the portfolio can be approximated accurately using a simple analytical formula, whose computation is fast and, more importantly, robust to the number of risk factors. This approximation formula, combined with the readily developed inversion methods, provides a quick and accurate way of pricing CDO tranches and other credit derivatives.

The dissertation’s organization is as follows. In the rest of this chapter, we describe the context of our work and provide necessary background. Section 1.2 describe the CDO technology, and the approach we use to price CDO tranches. Section 1.3 reviews the Normal Copula model for credit risk. Section 2.3 describes the approximation regime for CDO prices for portfolios with weakly-correlated obligors. The remaining two chapters — the main contribution of this work — are devoted to the description of the proposed approximation methods: Chapter 2 is devoted to the discussion of the first approximation method (entitled “Correlation Expansions”), and Chapter 3 to the second method (entitled “Quadratic Transform Approximation”). All technical proofs are deferred to the appendices. The main theorem from Chapter 2 is proved formally in Appendix A. The proofs of theorems from Chapter 3 are given in Appendix B.

1.2 Background on CDOs

1.2.1 Basic Structure

Collateralized debt obligations (CDOs) have become an increasingly popular class of securities to emerge in the credit derivatives market. The CDO technology offers a way to redistribute the credit risk of a pool of debt instruments and create a family of securities with widely different risk profiles, ranging from very safe to highly speculative. For a comprehensive background on CDOs, see for example Hull and White (2003) or Schönbucher (2003). We now describe the fundamental idea. In its most basic form, a CDO structure may be explained through Figure 1.1.

A CDO structure is constructed based on a reference pool of assets, which can consist of a diversified group of debt instruments — let us consider them bonds for the purpose of our illustration. The number of obligors generally varies between 50 to 150. These
bonds are subject to credit risk; in other words, they may default — a situation that, depending on the recovery rate after default, can result in minor or major losses within the collateral pool.

CDO tranches offer the opportunity to buy and sell protection from certain a fraction of these losses. In Figure 1.1, a CDO structure is created by “tranching out” credit losses on the reference pool into three types of securities: 1) an equity tranche, whose upper attachment point is 7% of the maximum loss in the reference portfolio; 2) a mezzanine tranche, whose attachment points are 7%-25%; and 3) a senior tranche, whose lower attachment point is 25%.

The equity tranche absorbs the first 7% of the losses in the underlying pool. The holders of the equity tranche are paid a fixed rate specific to this tranche — 35% according to Figure 1.1 — based on the outstanding nominal on the tranche. That is to say, the equity tranche holders initially are paid a return of 35% of the whole amount they invested (7% of the total bond principal). But suppose that, at time $T_1$, losses of 2% have been incurred (the gray stripe in Figure 1.1), then at the subsequent coupon date the equity tranche holders will be paid 35% of the remaining tranche notional (the unflooded portion in the equity tranche).
Similarly, the mezzanine tranche periodically receives coupon payments of 15% on the stochastically decaying tranche notional (initially set to $25\% - 7\% = 18\%$ of the total bond portfolio) until the losses reach 25% of the portfolio principal, at which point the contract is written down. The senior tranche is responsible for all losses in excess of the 25% absorbed by the equity and mezzanine tranches and, being the safest tranche, receives the lowest coupon rate among the three tranches. Figure 1.2 displays the loss absorbed by each tranche as a function of the aggregated loss in the reference collateral pool.

The CDO structure in Figure 1.1 is referred to as a cash CDO. There are natural extensions to this idea — liability structures with more (or fewer) than three tranches, combinations of fixed- and floating-coupon assets, and liability asset classes other than bonds. An increasingly popular variation of the CDO technology is the synthetic CDOs, which are very similar to cash CDOs, except that the bonds are replaced by credit default swaps; the arranger of a synthetic CDO passes the default risk on to the CDO tranches, thereby allowing investors to participate at different risk levels.
1.2.2 Portfolio Loss Approach in Pricing CDOs

We will focus on pricing the CDO tranches using the so-called portfolio loss approach, in which all cashflows during the life of the CDO are expressed in terms of the cumulative losses in the reference portfolio at prespecified coupon dates. Thus, the price of a CDO tranche can be written as an expectation with respect to the distribution of the underlying portfolio loss process. The detailed explanation of this approach can be found, for example, in Andersen et al. (2003) and Laurent and Gregory (2003). We now briefly explain the approach.

As seen from the payoff scheme described in the previous section, the value of a CDO tranche equals the default-free cashflows (calculated as a fixed coupon rate on the entire tranche notional) less the losses resulting from defaults in the reference pool. The fixed (default-free) leg is non-stochastic, and is valued like a stream of fixed cashflows. The main issue, therefore, is valuing the default payment leg.

Let us decompose the default payment leg by considering the contribution from each coupon period. Suppose that up to the time of the \(i\)th coupon payment \(T_i\), the reference pool has accumulated default losses of \(L_{T_i}\). Then, by time \(T_i\), the tranche notional will have depleted by \((L_{T_i} - A)^+ - (L_{T_i} - B)^+\), where \(A\) and \(B\) are the lower and upper attachment points, respectively (see Figure 1.3). Hence, the expected loss to be experienced by holders of this tranche during the \(i\)th coupon period is \(\text{E}(L_{T_i} - A)^+ - \text{E}(L_{T_i} - B)^+\) times the coupon rate. Discounting this amount to time zero and summing over all coupon periods during the life of the CDO gives the value of the default payment leg.

Thus, the problem of pricing a CDO tranche is reduced to that of evaluating expec-
tations of the form $E(L-y)^+$, where $L$ is the cumulative loss in the underlying portfolio at a specific coupon date, and $y$ is an attachment point. In this work, the snapshot of the loss portfolio $L$ at a given coupon date is modelled within the Normal Copula framework, described in the following section. We then spend the rest of the chapter discussing the computation of $E(L-y)^+$.

### 1.3 Normal Copula Model for Credit Risk

In this section, we describe the widely used Normal Copula model initiated by Li (2000), which has become an industry standard for pricing. We consider a CDO constructed from a reference portfolio of $M$ obligors. As explained in the previous section, in order to value a CDO contract we only need the marginal distribution of the aggregated loss of the reference portfolio at coupon dates. Let $L$ be the accumulated portfolio loss at a fixed coupon date. To specify the distribution of $L$, we introduce the following notation:

- $I_i = \text{default indicator for } i\text{th obligor}$
- $I_i = \begin{cases} 1 & \text{if } i\text{th obligor defaults,} \\ 0 & \text{otherwise;} \end{cases}$
- $p_i = P(I_i=1) = \text{marginal probability that the } i\text{th obligor defaults;}$
- $Y_i = \text{loss resulting from the default of } i\text{th obligor.}$

The marginal default probabilities $p_i$ are assumed known (e.g., from credit spreads for the prices of credit default swaps). The losses given default $Y_i$ are independent random variables, whose distributions are assumed to be known. Then, the portfolio loss at the given coupon date is

$$L = Y_1 I_1 + Y_2 I_2 + \ldots + Y_M I_M . \quad (1.1)$$

To model dependence among obligors we need to introduce dependence among the default indicators $I_1, \ldots, I_M$. In the Normal Copula model, dependence is introduced by letting

$$I_i = 1\{X_i > x_i\}, \quad i = 1, 2, \ldots, M , \quad (1.2)$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
where \((X_1, \ldots, X_M)\) are correlated \(N(0,1)\) random variables. The “default boundary” \(x_i\) is chosen to match the default probability, that is
\[
p_i = \Phi(x_i) .
\]
(1.3)

where \(\Phi(x) = 1 - \Phi(x)\), with \(\Phi\) the cumulative normal distribution function.

While the correlation among the \(X_i\) can be specified by their correlation matrix, it is a common practice to introduce the correlation structure through a set of “factors” \(Z_1, Z_2, \ldots, Z_d\) \((d \ll M)\),
\[
\begin{bmatrix}
X_1 \\
X_2 \\
X_3 \\
\vdots \\
X_M
\end{bmatrix}
= \begin{bmatrix}
a_{11} & a_{12} & a_{1d} & e_1 & e_1 \\
a_{21} & a_{22} & a_{2d} & e_2 & e_2 \\
a_{31} & a_{32} & a_{3d} & e_3 & e_3 \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
a_{M1} & a_{M2} & a_{Md} & e_M & e_M
\end{bmatrix}
\begin{bmatrix}
Z_1 \\
Z_2 \\
\vdots \\
Z_d
\end{bmatrix} + \begin{bmatrix}
e_1 \\
e_2 \\
e_3 \\
\vdots \\
e_M
\end{bmatrix} .
\]
(1.4)

Here, \(e_1, \ldots, e_M\) and \(Z_1, \ldots, Z_d\) are independent \(N(0,1)\) random variables. The real constants \(a_{ij}\) and \(e_i\) satisfy \(a_{i1}^2 + \ldots + a_{id}^2 + e_i^2 = 1\), so as to ensure that the variance of \(X_i\) is one. Each \(e_i\) \((1 \leq i \leq M)\) represents the idiosyncratic risk affecting only the \(i\)th obligor, while \(Z_1, \ldots, Z_d\) are risk factors that affect more than one obligor. The factors \(Z_1, \ldots, Z_d\) are sometimes given economic interpretations (as industry or regional risk factors, for example), or they may simply be the result of fitting the correlation matrix of \(X_1, \ldots, X_M\) with a factor structure using, for example, principle component analysis.

The matrix \(A := [a_{ij}]_{M \times d}\) is referred to as the loading matrix, and it completely determines the correlation structure among \(X_i\)’s, as can be explained in the following. Let \(a_i = [a_{i1} a_{i2} \ldots a_{id}]^T\) be the \(d \times 1\) column vector whose elements are taken from the \(i\)th row of the loading matrix \(A\). Note, first, that \(e_i = \sqrt{1 - a_{ii}^2} a_i\). Therefore, one can rewrite (1.4) as
\[
X_i = a_i^T Z + \sqrt{1 - a_{ii}^2} a_i e_i , \quad i = 1, \ldots, M
\]
where \(Z = [Z_1 Z_2 \ldots Z_d]^T\) is the \(d \times 1\) column vector of risk factors. Using the above representation for \(X_i\), we can compute the correlation coefficient \(\rho_{ik} := \text{E}[X_i X_k]\). As
the result,

\[ \rho_{ik} = \mathbf{a}_i^T \mathbf{a}_k = \sum_{j=1}^{d} a_{ij} a_{kj}, \quad \forall i \neq k. \]  

(1.5)

This equation is referred to as the \textit{factorization} of the correlation structure. Correspondingly, the covariance matrix \( \mathbf{C} = [\rho_{ij}]_{M \times M} \) is said to be of \textit{d-factor structure}, where \( d \) is the number of columns of the loading matrix \( \mathbf{A} \).

As explained in Section 1.2.2, pricing CDO tranches involves computing expectations of the form \( \mathbb{E}(L - y)^+ \), \( y \in \mathbb{R} \). This expectation may be viewed as the default payment leg of a single-period CDO tranche with lower attachment point \( y \). In subsequent chapters, which represent the main contribution of this work, we develop two approximation methods for approximating \( \mathbb{E}(L - y)^+ \), where \( L \) follows the Normal Copula model described above. The first approximation method, entitled \textit{Correlation Expansions}, is discussed in Chapter 2. The second method, \textit{Quadratic Transform Approximation}, is described in Chapter 3.
Chapter 2

Approximation by Correlation Expansions

The pricing method presented in this chapter is based on computing joint probabilities of correlated Gaussian random variables; i.e., probabilities of the form

\[ P(\mathbf{X} > \mathbf{x}) \]

where \( \mathbf{X} \) are correlated \( \mathcal{N}(0,1) \) random variables, and \( \mathbf{x} \) are real numbers. Our approach builds on Kibble's (1945) generalized tetrachoric series for multivariate normal random variables, in which the above joint probability is expanded as an infinite series of correlation coefficients. While various methods exist for special cases of the multivariate normal (e.g., bivariate (Ağca and Chance 2003, Vasicek 1998), trivariate (Kendall 1941), equal correlation (Somerville 1954)), Kibble's approximation is, in principle, general. However, Kibble's series in its original form contains a large number of terms and is computationally impractical. The approach we develop in this chapter leads to an alternative and more efficient way for evaluating Kibble's series when the underlying correlation matrix has a factor structure. This gives rise to a method of approximating tranche prices of CDOs and, more generally, expectations of other functions of the loss distribution in the Normal Copula model.

Within the context of the credit model described in Section 1.3, \( p_i = P(X_i > x_i) \) is interpreted as the marginal default probability of the \( i \)th obligor, and the joint probability
above is interpreted as the probability that a number of obligors default simultaneously. We propose an approximation of the form:

\[ P(X_1 > x_1, \ldots, X_M > x_M) \approx \sum_J \text{const. } \tilde{p}^{(J)}_1 \ldots \tilde{p}^{(J)}_M \tag{2.1} \]

where the summation contains a finite number of terms. For each label \( J \), the probability \( \tilde{p}^{(J)}_i \) is computed from \( p_i \) using a perturbation formula, which will be described shortly (Section 2.1). Note that the product \( \tilde{p}^{(J)}_1 \ldots \tilde{p}^{(J)}_M \) can be interpreted as the probability that a group of independent obligors, each having a marginal default probability of \( \tilde{p}^{(J)}_i \), default simultaneously. In this connection, the above approximation may be viewed as approximating a correlated model by a series of independent-obligor models.

Based on the above method of computing joint probabilities, we develop a method for approximating CDO tranche prices, i.e., the expectation of the form \( E(L - y)^+ \). Analogous to (2.1), the approximation is of the form:

\[ E(L - y)^+ \approx \sum_J \text{const. } \tilde{E}_J(L - y)^+ \tag{2.2} \]

where, for each label \( J \), the expectation \( \tilde{E}_J(L - y)^+ \) is the tranche price of a credit portfolio whose obligors are independent, each with default probability \( \tilde{p}^{(J)}_i \). To sum up (2.2) in words: a portfolio whose obligors are correlated can be approximated by a series of portfolios in which obligors are independent. Hence, computing the tranche price in a correlated-obligor model is reduced to simpler calculations of prices in independent-obligor models. (This method takes full advantage of the fact that prices in independent-obligor models are easy to compute.) The reason behind the title “correlation expansion” will be clear once we describe the method in full detail.

The organization of this chapter is as follows. We first describe the perturbation formula (Section 2.1) and discuss, in more detail, the problem of computing Gaussian joint probabilities (Section 2.2). Then, as the main discussion of this chapter, the subsequent sections are devoted to applications in pricing credit derivatives.

### 2.1 The Perturbation Formula

Where \( X_1, \ldots, X_M \) are standard Gaussian random variables, let \( \rho_{ik} \) denote the correlation coefficient between \( X_i \) and \( X_k \). Assume that \( \rho_{ik} \) have a generalized d-factor
structure; that is, for all \(i, k = 1, \ldots, M\),
\[
\rho_{ik} = \sum_{j=1}^{d} \omega_j a_{ij} a_{kj}
\]
(2.3)
where \(\omega_j\) and \(a_{ij}\) are real scalars. Note that (1.5) is a special case of (2.3) where \(\omega_j\) are one. We write \(a_{ij}\) and \(a(i, j)\) interchangeably. It is useful to have \(a_{ij}\) admit nonpositive index \(j\) by defining the convention \(a(i, -j) = -a(i, j)\) and \(a(i, 0) = 0\). Where \(d\) is the dimension of the factorization (2.3), define \(D := \{0, \pm 1, \pm 2, \ldots, \pm d\}\). Let \(D^n\) denote the Cartesian product
\[
D^n := \frac{D \times D \times \cdots \times D}{\text{n folds}} = \left\{ (j_1, \ldots, j_n) : j_1, \ldots, j_n \in D \right\}
\]
(2.4)
Let \(D^0 = \{0\}\) by convention.

Given a coordinate \(J = (j_1, \ldots, j_n)\) from the set \(D^n\) and a sequence of real numbers \(\theta_1, \ldots, \theta_n\) — which will later be called the perturbation parameters — we define \(v_j(i, k)\), where \(1 \leq i \leq M\) and \(k\) is an integer, through the following recursion: for \(J = (j_1, \ldots, j_n) \in D^n\),
\[
v_j(i, k) = v_{j'}(i, k) + \theta_n a(i, j_n) v_{j'}(i, k-1)
\]
(2.5)
where \(J' = (j_1, \ldots, j_{n-1}) \in D^{n-1}\) is the truncation of \(J\) by deleting the \(n^{th}\) (last) coordinate. The boundary condition for the above recursion is as follows: for \(J \in D^n\),
\[
v_j(i, 0) := 1\) and \(v_j(i, k) := 0\) whenever \(k > n\). Note that as \(\theta_1, \ldots, \theta_n \to 0\), \(v_j(i, k) \to 0\) for all \(J \in D^n\) and \(k > 0\).

**Definition 2.1 (Perturbed Probabilities)** For a given nonnegative integer \(n\) and a set of perturbation parameters \(\theta_1, \ldots, \theta_n\), and for \(J \in D^n\) and \(i = 1, \ldots, M\), we define \(\tilde{p}_i^{(J)}\) — the perturbed probability of \(p_i = P(X_i > x_i) = \tilde{\Phi}(x_i)\) — by
\[
\tilde{p}_i^{(J)} := p_i + \varphi(x_i) \sum_{k=1}^{n} v_j(i, k) H_{k-1}(x_i)
\]
(2.6)
where \(v_j(i, k)\) are scalars computed (2.5), \(\varphi(x)\) is the standard Gaussian density function, and \(H_n(x)\) denotes the Hermite polynomial of degree \(n\); that is, \(H_0(x) = 1\), \(H_1(x) = x\), and \(H_{n+1}(x) = x H_n(x) - n H_{n-1}(x)\) for \(n \geq 1\). □
Because the coefficients $v_j(i,k)$ in (2.6) converge to zero as $\theta_1, \ldots, \theta_n \to 0$, one can see that $\tilde{p}_{ij}^{(J)}$ defined in (2.6) is a bonafide probability, provided that $0 < p_i < 1$ and the perturbation parameters are sufficiently small.

As will be seen in subsequent sections, the perturbation formula (2.6) is useful in computing joint distribution of Gaussian random variables, and also in pricing credit derivatives under the Normal Copula model.

\section{Computing Gaussian Joint Probabilities}

Based on the perturbation formula in Definition 2.1, this section develops a method for computing joint probabilities of the form

$$P(X_1 > x_1, \ldots, X_M > x_M)$$  \hspace{1cm} (2.7)

where $X_1, \ldots, X_M$ are standard Gaussian random variables, with $\rho_{ik}$ as the correlation coefficient between $X_i$ and $X_k$. The method can be described as follows.

We begin with a parameterization scheme of the correlation structure. For $t \in [0,1]$, let $P_t$ be the probability measure under which $X_1, \ldots, X_M$ are zero-mean Gaussian
random variables whose covariance matrix is

\[ C_t := \begin{bmatrix} 1 & t\rho_{12} & t\rho_{13} & \ldots & t\rho_{1M} \\ t\rho_{21} & 1 & t\rho_{23} & t\rho_{2M} \\ t\rho_{31} & t\rho_{32} & 1 & t\rho_{3M} \\ \vdots & \vdots & \vdots & \ddots \\ t\rho_{M1} & t\rho_{M2} & t\rho_{M3} & \ldots & 1 \end{bmatrix} \] (2.8)

It follows that \( X_1, \ldots, X_M \) are independent under \( P_0 \). At \( t = 1 \), \( P_t \) coincides with the original probability measure \( P \). It is known (Kibble, 1945) that \( P_t \) is analytical in \( t \); in other words, there exist real numbers \( b_0, b_1, \ldots \) such that for all \( t \in [0,1] \),

\[ P_t(X_1 > x_1, \ldots, X_M > x_M) = b_0 + b_1 t + b_2 t^2 + \cdots \] (2.9)

where the series is convergent. Kibble gives the expression for the coefficient \( b_n \) as

\[ b_n = \sum \rho_{k_1 \ell_1} \cdots \rho_{k_n \ell_n} B_{h_1}(x_1) \cdots B_{h_M}(x_M) \] (2.10)

Here, the summation is over all integers \( k_1, \ell_1, \ldots, k_n, \ell_n \) taking values between 1 and \( M \) and subject to the condition \( k_1 < \ell_1, \ldots, k_n < \ell_n \). In each term of the summation, \( h_i \) \((1 \leq i \leq M)\) denotes the number of times "\( i \)" occurs among \( k_1, \ell_1, \ldots, k_n, \ell_n \). The function \( B_k(x) \), where \( k \) is a nonnegative integer, are as defined as \( B_0(x) := \Phi(x) \) and \( B_k(x) := \varphi(x) H_{k-1}(x) \) for \( k \geq 1 \).

In principle, (2.9) can be used to approximate the desired probability (2.7) by computing the first few coefficients \( b_n \)'s through (2.10) and then substitute \( t = 1 \). Unfortunately, one can show that in general the number of terms in the summation (2.10), even if we do not count duplicate terms, is

\[ \left( n + (M - 1)(M + 2)/2 \right)^n \] (2.11)

Obviously, even for moderate \( M \), the number of terms in the summation (2.10) is not manageable. So, Kibble's formula is not practical for computing joint Gaussian probabilities. Instead, we propose a more efficient method based on Theorem 2.1 below.
In stating the theorem, it is useful to define the following notation: where $D^n$ is the set displayed in (2.4) and $\omega_1, \ldots, \omega_d$ are the scalars in the factorization (2.3), let $\omega_0 := -2(\omega_1 + \ldots + \omega_d)$ and define $\omega_J$, for all $J = (j_1, \ldots, j_n)$ in the set $D^n$, as $\omega_J := \omega_{|j_1|} \ldots \omega_{|j_n|}$. We are now ready to state the theorem.

**Theorem 2.1** Suppose that $\rho_{ik}$'s in (2.8) have the generalized $d$-factor structure (2.3). Let $n$ be a nonnegative integer. For $J \in D^n$, let $\tilde{p}_i^{(J)}$ be the perturbed probabilities of $p_i = P(X_i > x_i)$, as described in Definition 2.1, with perturbation parameters $\theta_1 = \ldots = \theta_n = \theta$. Then, the coefficient $b_n$ in the expansion (2.9) can be expressed as the limit:

$$
\sum_{J \in D^n} w_J \tilde{p}_1^{(J)} \ldots \tilde{p}_M^{(J)} \rightarrow_{\theta \to 0} b_n \tag{2.12}
$$

where the weight $w_J$ is given as follows: $w_J := 1$ if $n = 0$; otherwise $w_J := \omega_J / (2\theta^2)^n$.

This theorem suggests a way to compute $b_n$. Using a perfect (no roundoff error) computer, we set the perturbation parameters $\theta_1 = \ldots = \theta_n = \theta$ to be infinitesimally small, calculate the perturbed probabilities $\tilde{p}_i^{(J)}$, then compute the weighted sum (2.12) to obtain $b_n$. In practice, we found that the sum (2.12) converges to $b_n$ very quickly as $\theta \to 0$; therefore, to obtain an accurate value of $b_n$, we can set $\theta$ to be only moderately small. After we have obtained the coefficients $b_k$, we can then use, say, the first $N$ terms of the expansion (2.9) to approximate the joint probability. Before we discuss the speed of this method, note the following important points:

i) It may appear that, in using (2.12) to compute the first $n+1$ coefficients $b_0, \ldots, b_n$, one needs to calculate $\tilde{p}_i(J)$ for all $J$ in $D^0, D^1, \ldots, D^n$. It suffices, however, to calculate the perturbed probabilities $\tilde{p}_i(J)$ only for $J \in D^n$. This is because for all $k$ and for all $J \in D^k$, one can show (see (2.5-2.6)) that $\tilde{p}_i(J) = \tilde{p}_i(J^*)$, where $J^* = (J, 0) \in D^{k+1}$; in other words, computing $\tilde{p}_i(J^*)$ for all $J^* \in D^{k+1}$ will automatically give you $\tilde{p}_i(J)$ for all $J \in D^k$.

ii) The summation in (2.12) seemingly contains $|D^n| = (2d+1)^n$ terms. Luckily, thanks to the symmetry of the perturbation formula, there are many repeated terms: if $J^* = (j_1^*, \ldots, j_n^*)$ is a permutation of $J = (j_1, \ldots, j_n)$, then it can be shown that

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Therefore, to calculate the weighted sum in (2.12), it suffices to compute \( \tilde{p}^{(J)}_i \) only for \( J \) in a subset of \( D^n \) in which the coordinates are arranged monotonically:

\[
\Lambda_n := \{ (j_1, \ldots, j_n) \in D^n : j_1 \leq j_2 \leq \ldots \leq j_n \}
\]

(2.13)

Obviously, \( \Lambda_n \) is a much smaller set than \( D^n \). In geometrical terms, \( D^n \) can be visualized as a hypercube in the \( n \)-dimensional space, while \( \Lambda_n \) can be visualized as a tetrahedron in the \( n \)-dimensional space. The following figure depicts the reduction in size from \( D^n \) to \( \Lambda_n \).

From i) and ii), we conclude that for all \( k = 0, 1, \ldots, n \), the coefficient \( b_k \) can be computed through a weighted sum over \( J \in \Lambda_n \) of the product \( \tilde{p}^{(J)}_1 \cdots \tilde{p}^{(J)}_n \). Consequently, the \( n \)th-order approximation of (2.9) at \( t = 1 \) is given by

\[
P(X_1 > x_1, \ldots, X_M > x_M) \approx b_0 + b_1 + b_2/2 + \ldots + b_n/n!
= \sum_{J \in \Lambda_n} \text{const.} \tilde{p}^{(J)}_1 \cdots \tilde{p}^{(J)}_n
\]

(2.14)

If we view the product \( \tilde{p}^{(J)}_1 \cdots \tilde{p}^{(J)}_n \) as the joint probability of a group of independent random variables, each having a tail probability of \( \tilde{p}^{(J)}_i \), then the above approximation may be loosely interpreted as approximating the joint distribution of correlated random variables by a series of joint distributions of independent ones.

Now, let us analyze the computing time of this approximation method. Obviously, the main effort in computing \( b_0, \ldots, b_n \) through Theorem 2.1 is spent in calculating the perturbed probabilities \( \tilde{p}^{(J)}_i \)'s for all \( J \in \Lambda_n \). It can be shown that, ignoring the one-time cost of computing \( \varphi(x_1), \ldots, \varphi(x_M) \) in (2.6), there is a way to implement the perturbation formula so that the whole task is accomplished within \( 4n(\binom{n+2d}{n}/n)M \) basic operations (addition and multiplication). Obviously, this is much faster than computing
$b_0, \ldots, b_n$ directly using (2.10), which, as explained earlier, requires an effort of the complexity shown in (2.11).

Next, let us compare our approximation with two other methods of computing joint Gaussian probabilities, namely Monte Carlo simulation and numerical integration. If $X_1, \ldots, X_M$ follow the $d$-factor model (1.4), then

$$P(X_1 > x_1, \ldots, X_M > x_M | Z) = \prod_{i=1}^{M} \Phi \left( \frac{x_i - a_i^T Z}{\sqrt{1 - a_i^T a_i}} \right)$$

(2.15)

Monte Carlo simulation is then employed to uncondition (2.15) and obtain the estimate of the desired probability. Note that, for one sampling of $Z$, computing (2.15) requires $2dM$ basic operations and $M$ evaluations of the normal cdf. Numerical integration, when used in lieu of Monte Carlo simulation to uncondition (2.15), spends the same effort per $Z$. In summary, the following table compares the computing time of our method with that of plain Monte Carlo simulation (i.e., without variance reduction techniques).

<table>
<thead>
<tr>
<th></th>
<th>Plain Monte-Carlo</th>
<th>Expansion (2.9) with $b_0, \ldots, b_n$ from (2.12)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of operations</td>
<td>$2dMN$</td>
<td>$4n \left( \frac{n+2d}{n} \right)M$</td>
</tr>
<tr>
<td>• basic operations$^\dagger$</td>
<td></td>
<td>$MN$</td>
</tr>
<tr>
<td>• normal cdf $\Phi(\cdot)$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Accuracy</td>
<td>$\mathcal{O}(1/\sqrt{N})$</td>
<td>$\mathcal{O}(\rho^{n+1})$</td>
</tr>
</tbody>
</table>

$^\dagger$ Basic operations comprise addition and multiplication.

Table 2.1: Comparing the complexity of two methods

As seen from the table, the method on the last column is, apart from the one-time evaluation of $\varphi(\nu_1), \ldots, \varphi(\nu_M)$ in (2.6), purely algebraic (i.e., only additions and multiplications). This feature gives the method an edge over Monte Carlo simulation and numerical integration, where repeated calculations of $\Phi(\cdot)$ contribute to longer computing time. (A single call to $\Phi(\cdot)$ can take as much as 40 times longer than a basic binary operation.) On top of that, simulation often requires many replications, while numerical integration requires computing time which grows exponentially with $d$. So, the proposed method is quite competitive.
It must be noted, however, that our method is not always appropriate. If the magnitude of \( \rho_{ik} \)'s in (2.8) are large, then the correlation structure \( C_t \) at \( t = 1 \) deviates significantly from the correlation structure at \( t = 0 \). As a result, when \( t = 1 \) the series (2.9) converges very slowly. This implies that one needs high-order approximation (large \( n \)) and, consequently, longer computing time.

Example 2.1 Let \( X_1, \ldots, X_{10} \) be \( \mathcal{N}(0,1) \) random variables. The marginal probabilities \( p_i = P(X_i > x_i) \) are given as follows.

\[
0.8913, 0.7621, 0.4565, 0.0185, 0.8214, 0.4447, 0.6154, 0.7919, 0.9218, 0.7382
\]

Note that if \( X_i \) are independent, \( P(X_1 > x_1, \ldots, X_{10} > x_{10}) = p_1 \ldots p_{10} = 6.95 \times 10^{-4} \).

Suppose, now, that the correlation structure of \( X_i \)'s are given by the following 2-factor loading matrix:

\[
A = \begin{bmatrix}
0.932 & 0.1664 \\
0.0837 & 0.1006 \\
0.1692 & 0.1419 \\
0.1050 & 0.0858 \\
0.0405 & 0.0609 \\
0.1344 & 0.0379 \\
0.1676 & 0.0387 \\
0.0039 & 0.1364 \\
0.1363 & 0.0606 \\
0.0759 & 0.1083
\end{bmatrix}
\]

The objective of this example is to contrast the two methods from Table 2.1 in computing the joint probability \( P(X_1 > x_1, \ldots, X_{10} > x_{10}) \).

Plain Monte Carlo with 10,000 replications gives the estimate \( (9.61 \pm 0.19) \times 10^{-4} \) with 95% confidence level; with 1,000,000 replications, the 95% confidence interval is \( (9.72 \pm 0.02) \times 10^{-4} \). Using (2.12), the second-order approximation is \( b_0 + b_1 + b_2/2 = 9.716 \times 10^{-4} \). Compared to the Monte Carlo simulation, the second-order approximation gets us within \( 10^{-6} \) of the true value at a fraction of the time. It is interesting to note that while the correlations are very light, the joint probability when \( X_i \) are correlated differs from the joint probability when \( X_i \) are independent by almost 40%.

\( \square \)
2.3 Application in CDO Pricing: Weak Correlation Case

In this section, we turn our attention to the problem of CDO pricing under the Normal Copula model. More specifically, we focus on the problem of computing the following expectation for any given real number $y$.

$$E(L - y)^+ , \text{ where } L = \sum_{i=1}^{M} Y_i 1\{X_i > x_i\}, \quad \rho_{ik} = \text{Corr}(X_i, X_k) \quad (2.16)$$

Here, $L$ represents the loss of a credit portfolio of $M$ obligors, and the expectation corresponds to the price of a CDO tranche whose lower attachment point is $y$. The random variable $Y_i$ represents the loss resulting from the default of the $i$th obligor, and the correlation among the defaults is modeled by the correlated $\mathcal{N}(0,1)$ random variables $X_1, \ldots, X_M$.

As the main contribution of this section, we develop a numerical method for computing $E(L - y)^+$ based on the findings from Section 2.2. Using the result from the previous section, we will show that the tranche price $E(L - y)^+$ in (2.16), which is difficult to compute because obligors are correlated, can be expressed as a series of prices in independent-obligor models, which are much easier to compute. The detail explanation of this method is as follows.

We begin, as we did previously in Section 2.2, by parameterizing the correlation structure of the random variables $X_1, \ldots, X_M$. We let $E_t$ denote the expectation under which $X_1, \ldots, X_M$ are zero-mean Gaussian random variables whose covariance structure is the matrix $C_t$ given in (2.8). It follows that, as $t$ varies from 0 to 1, the value of $E_t(L - y)^+$ progresses from a price in an independent-obligor model to the desired price in (2.16). Note that the marginal default probability of the $i$th obligor is $p_i = \Phi(x_i)$, constant under the measure $E_t$ for all $t \in [0,1]$. The following theorem, which is clearly analogous to Theorem 2.1, suggests the use of the perturbed probabilities $\tilde{p}_i^{(J)}$ from Definition 2.1 in computing $E_t(L - y)^+$.

**Theorem 2.2** For $t \in [0,1]$, the tranche price $E_t(L - y)^+$ of the portfolio $L$ is analytical in $t$; that is, $E_t(L - y)^+$ admits the infinite expansion

$$E_t(L - y)^+ = \alpha_0 + \alpha_1 t + \alpha_2 \frac{t^2}{2!} + \ldots \quad (2.17)$$
Furthermore, if $\rho_{ik}$'s in (2.8) have the generalized $d$-factor structure (2.3), then the coefficient $\alpha_n$ in the above expansion can be characterized as follows. For each label $J$ in the set $D^n$ (defined in (2.4)), let $\bar{E}_J$ denote the expectation under which each obligor $i$ defaults independently with probability $\bar{p}_i^{(J)}$, perturbed from $p_i = \bar{p}(x_i)$ using perturbation parameters $\theta_1 = \ldots = \theta_n = \theta$. Then, we have

$$\sum_{J \in D^n} w_J \bar{E}_J(L - y)^+ \xrightarrow{\theta \downarrow 0} \alpha_n$$

(2.18)

where the weight $w_J$ is as given in Theorem 2.1.

**Proof** Let $S = \{1, \ldots, M\}$. Let $\bar{P}_J$ denote the probability measure under which each obligor $i$ defaults independently with probability $\bar{p}_i^{(J)}$. We will first argue that

$$\left. \frac{d^n}{dt^n} \right|_{t=0} P_t(\text{names } i \in S \text{ default}) = \lim_{s \to 0} \sum_{J \in D^n} w_J \bar{P}_J(\text{names } i \in S \text{ default}) .$$

(2.19)

By differentiating the expansion (2.9) with respect to $t$, and by interpreting the event $\{X_t > x_1\}$ as the event that name $i$ defaults, one can see that the left-hand side of (2.19) simplifies to $b_n$. But, according to (2.12), the right-hand side of (2.19) simplifies to $b_n$ as well. So, the equality (2.19) is established. Furthermore, since the expansion (2.9) and Theorem 2.1 are valid for an arbitrary group of random variables $X_i$, one can deduce that the equality (2.19) holds, not only for $S = \{1, \ldots, M\}$, but for all subsets $S$ of $\{1, \ldots, M\}$. The expansion (2.9) also implies that $P_t(\text{names } i \in S \text{ default})$ is analytical in $t$ for all $S \subset \{1, \ldots, M\}$. Let $E_S$ denote the event that obligors $i \in S$ default but the rest (obligors $\ell \in \{1, \ldots, M\} \setminus S$) do not. Using the inclusion-exclusion formula from set theory, it is easy to argue that $P_t(E_S)$ is analytical in $t$ and, similarly to (2.19),

$$\left. \frac{d^n}{dt^n} \right|_{t=0} P_t(E_S) = \lim_{s \to 0} \sum_{J \in D^n} w_J \bar{P}_J(E_S)$$

(2.20)

for all $S \subset \{1, \ldots, M\}$. For every set $S$ of indices, define a random variable $Y_S := \sum_{i \in S} Y_i$. It is easy to see that

$$E_t(L - y)^+ = \sum_{S \subset \{1, \ldots, M\}} E(Y_S - y)^+ P_t(E_S) .$$

(2.21)

(The summation is over all possible sets of defaulting obligors, and $E(Y_S - y)^+$ is the payoff upon the event $E_S$.) It follows that, since $P_t(E_S)$ is analytical in $t$, so is $E_t(L - y)^+$;
this justifies the expansion (2.17). Now, if we multiply both sides of (2.20) by $E(Y_S-y)^+$
(this expectation is taken with respect to the distributions of $Y_i$'s, which are the same
under both $P_t$ and $\tilde{P}_J$) and then sum over all $S \subset \{1, \ldots, M\}$, we obtain

$$\left. \frac{d^n}{dt^n} \right|_{t=0} E_t(L-y)^+ = \lim_{s \to 0} \sum_{J \in D^n} w_J \tilde{E}_J(L-y)^+$$

(2.22)

Note that, according to (2.17), the left-hand side simplifies to $\alpha_n$. This proves the limit
(2.18). Thus, the proof of Theorem 2.2 is complete.

Since every coefficient $\alpha_n$ in the expansion (2.17) can be expressed as the weighted
sum (2.18), the effect of Theorem 2.2 is that it expands the tranche price $E_t(L-y)^+$,
which is difficult to compute when $t > 0$ (because obligors are correlated), into a series
of prices $\tilde{E}_J(L-y)^+$ in independent-obligor models, which are easy to compute. In
this respect, the desired tranche price $E(L-y)^+$ in (2.16) can be approximated in the
following manner. Fixing a small value of $\theta$, we compute the weighted sum (2.18) to
obtain the coefficient $\alpha_n$. (In practice, we have found that the convergence (2.18) is
quite fast, so a moderately small $\theta$ is sufficient to give an accurate value of $\alpha_n$.) Once
we obtain the necessary coefficients, we use the first few terms of the expansion (2.17),
with $t$ substituted by 1, to approximate $E(L-y)^+$. Let us now analyze the computing
time consumed by this approximation method.

In computing the first $n+1$ coefficients $\alpha_0, \ldots, \alpha_n$ using the formula (2.18), it seems
that one needs to compute $\tilde{E}_J(L-y)^+$ for every $J \in D^0, \ldots, D^n$. Luckily, because
the summation (2.18) contains many duplicate terms thanks to the symmetry of the
perturbation formula, we only need to compute $\tilde{E}_J(L-y)^+$ for a much smaller set of $J$.
In fact, by following the same line of arguments that leads to the expression (2.14) in
Section 2.2, one can argue that for all $k = 0, \ldots, n$, $\alpha_k$ can be expressed as a weighted
sum of $\tilde{E}_J(L-y)^+$ over $J \in \Lambda_n$, where $\Lambda_n$ is the much smaller subset of $D^n$ given
in (2.13). So, analogous to (2.14), the $n$th-order approximation of $E(L-y)^+$ can be
expressed as

$$E(L-y)^+ \approx \alpha_0 + \alpha_1 + \alpha_2/n + \ldots + \alpha_n/n!$$

$$= \sum_{J \in \Lambda_n} \text{const.} \tilde{E}_J(L-y)^+$$
Define **one unit of computing time** as the amount of time it takes to compute the tranche price in an independent-obligor model. It follows that computing the above \( n \)-order approximation of \( \mathbb{E}(L - y)^+ \) requires

\[
|A_n| = \binom{n + 2d}{n} \text{ units of computing time},
\]

where \( d \) is the dimension in the factorization (2.3). Table 2.2 shows the amount of time required for computing the \( n \)th order approximation for some sample values of \( d \).

<table>
<thead>
<tr>
<th>( d )</th>
<th>1st order</th>
<th>2nd order</th>
<th>3rd order</th>
<th>4th order</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>7</td>
<td>28</td>
<td>84</td>
<td>210</td>
</tr>
<tr>
<td>5</td>
<td>11</td>
<td>66</td>
<td>286</td>
<td>1,001</td>
</tr>
<tr>
<td>7</td>
<td>15</td>
<td>120</td>
<td>680</td>
<td>3,060</td>
</tr>
<tr>
<td>10</td>
<td>21</td>
<td>231</td>
<td>1,771</td>
<td>10,626</td>
</tr>
</tbody>
</table>

Table 2.2: Computing time for the expansion (2.17). One unit of computing time equals the amount of time it takes to compute a tranche price in an independent-obligor model.

The duration of one unit of computing time depends on the method employed to compute tranche prices in independent-obligor models. In our numerical experiments, we use the recursive algorithm in Andersen et al. (2003), which builds the loss distribution by successively adding one obligor per iteration. This recursion is particularly attractive for us because it yields the **exact** value of the prices \( \mathbb{E}_i(L - y)^+ \) in independent-obligor models. This ensures that the summation (2.18) converges to the right limit as we send the perturbation parameter \( \theta \) to zero. Straightforward implementation of this recursion on MATLAB takes about 0.6 seconds for a loss portfolio with 200 obligors.

Now, let us discuss the choice of \( n \) — the order of the approximation. As seen in (2.8), the magnitude of the \( \rho_{ij} \)'s determine how much \( C_t \) changes as \( t \) varies from 0 to 1. Consequently, the rate at which the series (2.17) converges depends on the magnitude of \( \rho_{ij} \); the stronger the correlation among obligors, the higher the order \( n \) we need in order to ensure that the truncated series is accurate. For this reason, the approximation (2.17) is not appropriate if \( X_i \)'s are highly correlated. We will explain how to deal with heavier correlation in the next section.
As seen in Table 2.2, the approximation is attractive when the number of factors $d$ is small. However, if $d$ is very small ($d \leq 2$, say), then numerical integration is a very viable option, and there would be no point in doing any kind of approximation. Our approximation is competitive when $d$ is of moderate size, as it is less sensitive to $d$ than, say, numerical integration, whose computing time grows exponentially in $d$. For example, if $d = 3$, then, according to Table 2.2, 210 units of time allow us to compute up to the 4th-order approximation, which can give good results. In contrast, numerical integration would give poor results given the same amount of computing time, because we have only $\sqrt{210} \approx 6$ units of time (6 gridpoints in the trapezoidal rule) allocated to integrate each of the three factors.

If $d$ is too large, the approximation becomes impractical, as it requires too many units of computing time. In this case, a remedy is to fit the correlation matrix $C$ with a lower factor structure. A 10-factor structure usually suffice for mimicking an empirical correlation matrix.

The following example demonstrates the implementation of Theorem 2.2. A more realistic example will be given in the next section, when we allow heavier correlation.

**Example 2.2** Consider a portfolio of $M = 50$ names. The default probabilities are $p_i = 0.02$ for all obligors. The exposure of obligor $k$ is given by $c_k = k$, $k = 1, \ldots, M$. The loading matrix $A = [a_{ij}]_{50 \times 5}$ is a sparse matrix whose nonzero elements are all equal to 0.2. The nonzero elements in the first column are $a_{11}, \ldots, a_{12,1}$; in the 2nd column $a_{9,2}, \ldots, a_{22,2}$; in the 3rd column $a_{19,3}, \ldots, a_{32,3}$; in the 4th column $a_{29,4}, \ldots, a_{42,4}$; and in the 5th column $a_{39,5}, \ldots, a_{50,5}$. This factor structure results in sparse (and light) correlation matrix.

Suppose we want to approximate $E_t(L - y)^+$ at $t = 1$ and $y = 200$ (this is about 16% the total exposure), using no more than 500 units of computing time. According to Table 2.2, this allows us to use up to the 3rd order term in the expansion (2.17). We need, in total, $|A_3| = 286$ different values of $E_J(L - y)^+$, each of which is computed with the perturbed default probabilities $\tilde{p}_i^{(J)}$ from formula (2.6). Note that the perturbation parameter $\theta$ should be small, so that the summation (2.18) converges to the right limit. However, in practice, we have found that $\theta$ need not be very small. Figure 2.2 below
shows the value of the weighted sum (2.18), with \( n = 3 \), as a function of \( \theta \). One can see

\[
0.00260 - 0.00258 - 0.00256 - 0.00254 - 0.00252 - 0.00250
\]

Figure 2.2: The summation (2.18) as a function of \( \theta \)

that, at \( \theta = 0.5 \), the weighted sum (2.18) starts to converge already. In this example, and in every example that follows, we consistently use perturbation parameter \( \theta = 0.1 \). As the results, the coefficients \( \alpha_n \) are as follows.

<table>
<thead>
<tr>
<th>( \alpha_0 )</th>
<th>( \alpha_1 )</th>
<th>( \alpha_2 )</th>
<th>( \alpha_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0012</td>
<td>0.0016</td>
<td>0.0022</td>
<td>0.0026</td>
</tr>
</tbody>
</table>

Therefore, the 3rd-order approximation for \( E_t(L - 200)^+ \) is

\[
(1.2 + 1.6 t + 2.2 t^2/2! + 2.6 t^3/3!) \times 10^{-3}.
\]

Figure 2.3 compares the first, second, and third order approximation. The true values are represented with "o". These are computed using Monte Carlo simulation. One can see that at \( t = 1 \), the 3rd-order approximation is quite accurate.

To end this section, it is worth noting that, while we have focused exclusively on approximating \( E_t(L - y)^+ \), the same method can be used to approximate \( E_t[f(L)] \), where \( f \) is an arbitrary function of the loss portfolio. We can show that for any \( f : \mathbb{R} \rightarrow \mathbb{R} \), \( E_t[f(L)] \) admits an infinite-series expansion in \( t \), and each coefficient in the expansion can be calculated through a weighted sum of the expectation \( \tilde{E}_t[f(L)] \), under which obligors are independent. The formal statement is as follows.
Theorem 2.3 (Generalization of Theorem 2.2) Let $L$ be the loss of a credit portfolio modeled in (2.16). Let $E_t$ denote the expectation under which the random variables $X_1, \ldots, X_M$ have Gaussian distribution with zero-mean and with covariance structure $C_t$ given in (2.8). For any mapping $f : \mathbb{R} \rightarrow \mathbb{R}$, $E_t[f(L)]$ is analytical in $t$; that is, $E_t[f(L)]$ admits the infinite expansion

$$E_t[f(L)] = b_0 + b_1 t + b_2 \frac{t^2}{2!} + \ldots$$

Furthermore, under the same additional assumptions as Theorem 2.2, we have

$$\sum_{J \in D^n} w_J \tilde{E}_J[f(L)] \xrightarrow{\theta \downarrow 0} b_n$$

where, for each $J \in D^n$, $\tilde{E}_J$ denotes the expectation under which each obligor $i$ defaults independently with probability $p_i^{(J)}$ perturbed from $p_i = \bar{p}(x_i)$ using perturbation parameters $\theta_1 = \ldots = \theta_n = \theta$. The weight $w_J$ is as given in Theorem 2.1. \qed

The proof of this theorem is analogous to that of Theorem 2.2. Note that, because $L$ has finite support, we do not need to impose any regularity conditions on $f$. Theorem 2.3 opens up a wide range of application. For example, by choosing $f(L) = 1\{L > y\}$, Theorem 2.3 can be used to approximate the loss distribution; this is useful in calculating the value-at-risk.

### 2.4 Extension to Strong Correlation Case

The approximation scheme described in the previous section performs well when the correlation among obligors is not too strong. This section explains how we can modify
the method in Section 2.3 to cover the case of heavy correlation.

The main modification lies in the parameterization scheme. Let \( C = [\rho_{ij}] \) be the correlation matrix of the zero-mean Gaussian random variables \( X_1, \ldots, X_M \) in (2.16). Instead of defining \( C_t \) as in (2.8), where \( C_0 \) corresponds to obligors' being independent, we now define \( C_t \) so that \( C_0 \) corresponds to obligors' having a reference correlation structure \( R \); that is,

\[
C_t := (1-t)R + tC , \quad t \in [0,1] ,
\]

where the reference covariance matrix \( R \) is chosen to have an \( r \)-factor structure, with \( r \) being a small integer. We shall discuss the issue of selecting \( R \) later; at this point, we regard \( R \) as an \( r \)-factor approximation of \( C \). It is important that \( r \) should be significantly smaller than \( d \) — the number of factors in the targeted correlation structure \( C \). (Remember, if \( d \) is already very small, then there is no point in doing any kind of approximation, as numerical integration can be carried out efficiently.) In the following analysis, we assume that \( r = 1 \). The extension to \( r > 1 \) follows in an obvious manner.

Let \( L \) be the loss of a credit portfolio modeled in (2.16). Let \( E_t \) denote the expectation under which the random variables \( X_1, \ldots, X_M \) in (2.16) have Gaussian distribution with zero-mean and with correlation structure \( C_t \) in (2.23). Suppose that the reference covariance matrix \( R \) is chosen to have a single-factor structure. That is, for some real scalars \( \gamma_1, \ldots, \gamma_M \),

\[
(i,j)\text{th element of } R = \begin{cases} 1, & i = j \\ \gamma_i \gamma_j, & i \neq j \end{cases} .
\]

(2.24)

It follows that, as \( t \) increases from 0 to 1, the value of \( E_t(L-y)^+ \) progresses from the tranche price in a single-factor model to the desired tranche price of the targeted model (2.16). We propose approximations of the form

\[
E_t(L-y)^+ = \zeta_0 + \zeta_1 t + \zeta_2 \frac{t^2}{2!} + \ldots + \zeta_n \frac{t^n}{n!} (2.25)
\]

Presently, we will specify the coefficients \( \zeta_i \)'s through a conditioning argument and an application of the result from Section 2.3.
First, we observe that, if $X_1, \ldots, X_M$ are $\mathcal{N}(0,1)$ random variables with the covariance structure $C_t$ given by (2.23), and if the reference correlation matrix $R$ has the single-factor structure (2.24), then $X_1, \ldots, X_M$ can be decomposed as

$$X_i = \gamma_i Z + \sqrt{1-\gamma_i^2} \tilde{X}_i ,$$

(2.26)

where $Z \sim \mathcal{N}(0,1)$, and $\tilde{X}_1, \ldots, \tilde{X}_M$ are $\mathcal{N}(0,1)$ random variables, independent of $Z$, having the following covariance structure:

$$
\begin{bmatrix}
1 & t \rho_{12} & t \rho_{13} & \ldots & t \rho'_{1M} \\
 t \rho_{21} & 1 & t \rho_{23} & \ldots & t \rho'_{2M} \\
 t \rho_{31} & t \rho_{32} & 1 & \ldots & t \rho'_{3M} \\
 \vdots & \vdots & \vdots & \ddots & \vdots \\
t \rho'_{M1} & t \rho'_{M2} & t \rho'_{M3} & \ldots & 1
\end{bmatrix}
$$

(2.27)

where $\rho'_{ik} = (\rho_{ik} - \gamma_i \gamma_k) / \sqrt{(1-\gamma_i^2)(1-\gamma_k^2)}$. We will refer to the random variable $Z$ as the market factor, since it generates the reference correlation matrix $R$. In this respect, the $\rho'_{ij}$ may be viewed as the deviations from the reference correlation matrix $R$. It follows that, conditioned on the market factor $Z$, the correlation among the obligors becomes (2.27). Consequently, from Section 4,

$$E_t [(L-y)^+ | Z=z] = \alpha_0(z) + \alpha_1(z) t + \alpha_2(z) \frac{t^2}{2!} + \ldots$$

(2.28)

where the argument $z$ in $\alpha_n(z)$ signifies that the coefficients depend on the value at which the market factor is conditioned. For any given $z$, $\alpha_n(z)$ can be computed using the method described in Section 2.3. Comparing (2.25) and (2.28) gives

$$\zeta_k = \mathbb{E}[\alpha_k(Z)] = \int_{-\infty}^{\infty} \alpha_k(z) \varphi(z) \, dz .$$

(2.29)

Using numerical integration, $\zeta_0, \ldots, \zeta_n$ can then be computed from $\alpha_0(z), \ldots, \alpha_n(z)$ to give the nth order approximation (2.25) for $E_t(L-y)^+$. 

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Note that computing $E(L-y)^+$ through other methods (Monte Carlo simulation or numerical integration) involves some form of integrating in $d$-dimensional space, where $d$ is the number of factors in the structure of the correlation matrix $C$. The virtue of (2.29), therefore, is that it reduces a $d$-dimensional problem to one-layer integration.

The computational effort consumed by this method can be analyzed in units of time as in Section 2.3. That is, the amount of time required to compute $\alpha_n(z)$ for a given $z$ depends on $n$ (the order of the approximation) and the factor-structure of the matrix (2.27) of deviations. Note that the method in this section requires extra time for evaluating the integral (2.29). If the deviations $\rho'_{ij}$ in (2.27) have a $d'$-factor structure, and if we use $K$ values of $z$ in evaluating the integral (2.29), then the method in this section requires

$$K \times \left( \frac{n + 2d'}{n} \right) \text{ units of time}$$

(2.30)

to compute the $n$th order approximation. Through numerical experiments, we have found that, as a function of $z$, $\alpha_k(z)$ is quite well-behaved and predictable. (See Figure 2.4 in the following example for the typical shapes of the curves $\alpha_1(z)$ and $\alpha_2(z)$.)

In practice, we have found that $K = 10n$ typically gives good results.

Obviously, the choice of the order $n$ depends on the rate at which the series (2.25) converges. Unlike the expansion (2.17) in Section 2.3, where $t = 0$ always corresponds to the price in an independent-obligor model, we now have the freedom to choose the reference matrix $R$ as the correlation structure at $t = 0$. If $R$ successfully captures the most important features of $C$, the progression of $(t, E_t(L-y)^+)$ from $t = 0$ to $t = 1$ will nearly be a straight line, in which case the linear or quadratic approximations will give good results.

Just as $R$ has the interpretation of being the market-wide correlation, the deviations $\rho'_{ij}$ in (2.27) can be interpreted as the excess correlation within and among groups of obligors. (It is not uncommon to find that the matrix (2.27) has a block structure, such as that described in Gregory and Laurent (2004).) In this connection, $d'$ — the number of factors in the structure of the deviations — has the interpretation of being the number of sectors in the market. Alternatively, $d'$ can simply be the dimension used in fitting a factor structure to the deviations (2.27) through principal component analysis. In doing so, keep in mind that $d'$ should be kept small, so that the computing time (2.30) is...
manageable. A reasonable range of $d'$ is displayed in Table 2.3. The computing time is as given in (2.30), with $K = 10n$.

<table>
<thead>
<tr>
<th>$d'$</th>
<th>1st order</th>
<th>2nd order</th>
<th>3rd order</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>70</td>
<td>560</td>
<td>2,520</td>
</tr>
<tr>
<td>4</td>
<td>90</td>
<td>900</td>
<td>4,950</td>
</tr>
<tr>
<td>6</td>
<td>130</td>
<td>1,820</td>
<td>13,650</td>
</tr>
<tr>
<td>8</td>
<td>170</td>
<td>3,060</td>
<td>29,070</td>
</tr>
</tbody>
</table>

Table 2.3: Computing time of expansion (2.25). One unit of computing time equals the amount of time it takes to compute an independent-model price.

Now, consider the computing time required by other methods. In plain Monte Carlo simulation, we require $N$ units of time to generate $N$ replications. To achieve reasonable precision, the required number of replications can easily be in the tens of thousands. Comparing this with the computing time in Table 2.3, one can see that the 2nd-order (or even 3rd-order) approximation can be competitive for a wide range of $d'$. (This comparison will be made concrete later in a numerical example.)

In numerical integration, the computing time increases exponentially with $d'$, as can be explained as follows. Suppose that we use the trapezoidal rule with $K$ gridpoints to numerically integrate with respect to the market factor $Z$. This takes one unit of time per gridpoint. But we still have to integrate with respect to $d'$ more factors (from the deviation matrix (2.27)). Thus, in total, numerical integration consumes $K^{d'+1}$ units of computing time. To be more concrete, consider $d' = 3$, for example. While 2520 units of time allow us to compute up to the third-order approximation (see Table 2.3), the same amount of time allows the trapezoidal rule only $\sqrt[3]{2520} \approx 7$ gridpoints per factor, which is likely to give poor results. In other words, the curse of dimensionality is more pronounced in numerical integration than in our method.

Before we give a numerical example, let us discuss briefly the means of selecting the reference correlation matrix $R$. To ensure that the expansion (2.25) converges within the first few terms, $R$ should be chosen so that it approximates the targeted covariance matrix $C$ as closely as possible. Andersen, Sidenius, and Basu (2003) give an efficient algorithm for fitting a low-factor structure to a given covariance matrix. In numerical
experiments, we have found this method to be quite satisfactory for determining $R$. Alternatively, since the covariance matrix $C$ is often specified through the copula (1.4), a convenient way of choosing the single-factor correlation structure $R$ is to pick one of the factors $Z_1, \ldots, Z_d$ as the market factor (the rest of the factors will determine the deviations $\rho_{ik}$ in (2.27)). This is appropriate especially when one of factors $Z_1, \ldots, Z_d$ carries heavier loadings than the others, and is, therefore, a clear choice for the market factor.

In general, one can choose the reference correlation matrix $R$ to be of $r$-factor structure, where $r > 1$. In that case, instead of having one market factor $Z$ in the decomposition (2.26), we will have $r$ market factors in its place; subsequently, the expectation (2.29) will become an $r$-layer integral. Since the effort required to evaluate the integral is exponential in $r$, we should resort to this measure only when the single-factor structure fails to capture the nature of $C$.

**Example 2.3** We consider a portfolio of $M = 200$ names. For obligor $k = 1, \ldots, M$, the default probability and the loss given default are

$$p_k = 0.02 \left( 1 + \sin \frac{8\pi k}{M} \right);$$

$$c_k = \left( \frac{8k}{M} \right)^2.$$

The covariance structure among the obligors is represented by the following 7-factor loading matrix:

$$A = \begin{bmatrix} 
\begin{array}{ccccccc}
\hline
& & & & & & \\
1 & & & & & & \\
\hline
& g & & e & & e & \\
& g & & e & & e & \\
& h & g & e & e & e & e \\
& h & g & e & e & e & e \\
\end{array}
\end{bmatrix}$$

where $h$ is a column vector of 200 entries, all equal to 0.6; $g$ is a column vector of 50 entries, all equal to 0.4; and $e$ is a column vector of 25 entries, all equal to 0.3. We want
to approximate $E(L - y)^+$ at $y = 700$ (this is about 8 times the expected loss $E[L]$ and about 14% of the maximum loss) using the 2nd-order approximation ($n = 2$).

We choose the reference single-factor matrix $R$ to be the correlation structure generated by the first column of $A$. Consequently, the residual correlations $\rho'_{ij}$ in (2.27) will have a 6-factor structure. Let $\alpha_1(z)$ and $\alpha_2(z)$ be the coefficients in the expansion (2.28), calculated using the method from Section 2.3. Figure 2.4 shows $\alpha_1(z)$ and $\alpha_2(z)$ for a range of $z$. In this example, 20 values of $z$ are enough to define the shape of these curves.

![Figure 2.4: $\alpha_1(z)$ and $\alpha_2(z)$ for $z \in [1,3]$.](image)

Through several numerical examples, we have found that the curve $\alpha_1(z)$ is generally unimodal, $\alpha_2(z)$ has three extrema, and $\alpha_3(z)$ five extrema. (Presumably, in general $\alpha_k(z)$ has $2k - 1$ extrema, though we have no theoretical result to substantiate this.) This information can help save the number of points used in evaluating the integral (2.29) by putting them in strategic spots.

Using the curves $\alpha_1(z)$ and $\alpha_2(z)$ in Figure 2.4, we compute $\zeta_1$ and $\zeta_2$ from the integral (2.29). (The integration is carried out by using the trapezoidal rule.) As the result, the first curve integrates to $\zeta_1 = 5.19$, and the second curve integrates to $\zeta_2 = 2.77$. The price of the reference single-factor model is $\zeta_0 = E_0(L - 700)^+ = 6.11$. Thus,
the 2nd-order approximation for the desired price $E(L-y)^+ \approx \zeta_0 + \zeta_1 + \frac{1}{2} \zeta_2 = 12.69$. Plain Monte Carlo simulation with $10^6$ replications gives $12.86 \pm 0.07$ with 95% confidence level.

Figure 2.5 shows the true values of $E_t(L-y)^+$ at $t = 0, 0.2, 0.4, 0.6, 0.8, 1$. These are marked by “o”. The 1st-order approximation is shown as the dotted line, the 2nd-order approximation as the solid line. Note that $\zeta_1$ and $\zeta_2$ represent, respectively, the slope and the convexity at $t = 0$.

![Figure 2.5: Approximation for $E_t(L-y)^+$, $t \in [0,1]$.]

Now, let us compare this method with Monte Carlo simulation. Table 2.4 shows the computing time and the outputs of each approach. We take the estimate in the last

<table>
<thead>
<tr>
<th>Approximation (2.25)</th>
<th>Computing time*</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st order</td>
<td>260</td>
<td>11.30</td>
</tr>
<tr>
<td>2nd order</td>
<td>1,820</td>
<td>12.69</td>
</tr>
<tr>
<td>Plain Monte Carlo</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2,000 replications</td>
<td>2,000</td>
<td>11.28 ± 5.18†</td>
</tr>
<tr>
<td>$10^7$ replications</td>
<td>$10^7$</td>
<td>12.86 ± 0.07†</td>
</tr>
</tbody>
</table>

* One unit of computing time equals the amount of time it takes to compute an independent-obligor price.
† 95% confidence interval.

Table 2.4: Comparing computing time
line as the representative of the true value. Evidently, the variance of the Monte Carlo estimate is quite high; at 2000 replications, the resulting confidence interval carries little to no information. On the other hand, the 2nd-order approximation requires straightforward computation of only 1,820 independent-obligor prices, and gives a fairly accurate result.

It should be noted that, while simulation characterizes error in terms of confidence intervals, our approximation does not specify errors. A simple way to check the accuracy of the 2nd-order approximation is to compute the 3rd-order term and see that it is negligible (this, of course, comes with extra computational cost, and does not provide a rigorous error bound.)

Figure 2.6 shows the approximation for a range of $y$. The true values are shown by "o". The dots on the left plot show the 1st-order approximation, and the right plot shows the 2nd-order approximation. One can see that the 1st-order approximation performs well for a wide range of $y$. This means that, within this range of $y$, the progression of $E_t(L-y)^+$ with $t$ is almost linear. As a rule of thumb, higher $y$ requires higher order of approximation.

**Example 2.4** The purpose of this example is to demonstrate the importance of the choice of the reference correlation matrix $R$. As explained earlier, if $R$ successfully capture the feature of the targeted correlation matrix $C$, then $C_t$ defined in (2.23) will change only by a little as $t$ varies from 0 to 1; consequently, the series (2.25) will converge quickly (i.e., we need only the first few terms to get a good approximation).
Consider the CDO structure from the previous example. Let $C = [\rho_{ik}]$ be the correlation matrix of the obligors, computed from the given loading matrix. Before we apply our approximation method, let us first use the algorithm in Andersen et. al. (2003) to fit the covariance matrix $C$ with a 7-factor structure. The loading matrix resulting from this algorithm \textit{may not be the same} as the one given in the previous example; indeed, the algorithm returns the loading matrix displayed on page 36. Nevertheless, this new loading matrix, by construction, is equivalent to the one in the previous example, in the sense that they both generate the same correlation structure. Thus, we can characterize the same portfolio from the previous example using this new loading matrix instead.

Suppose, once again, that we want to approximate the tranche price $E(L - 700)^+$ using the expansion (2.25). This time, however, we choose the reference matrix $R$ to be the correlation matrix generated by the first column of the loading matrix on page 36. One can see that, because the factor loadings in the other columns are quite small, this new reference matrix $R$ is a much better approximation of $C$ than the one we used in the previous example. Therefore, we can expect improvement in the accuracy of first-order approximation. The following figure shows the first-order (dotted line) and the second-order (solid line) approximations for $E_t(L - 700)^+$, $t \in [0,1]$. (In this example, the coefficients in the expansion (2.25) are $\zeta_0 = 10.28$, $\zeta_1 = 1.76$, and $\zeta_2 = 1.19$.) The true values at $t = 0, .2, .4, .6, .8, 1$ are shown as the circles.

![Figure 2.7: Approximation for $E_t(L - 700)^+$, $t \in [0,1]$.](image)

Compare Figure 2.7 with Figure 2.5 from the previous example. Note that the starting point at $t = 0$ for each plot is different (because we use different $R = C_0$). One
can see that, this time, the first order approximation performs better than the previous example.

At this point, we have computed $\zeta_0$, $\zeta_1$, and $\zeta_2$ — three derivatives of $E_t(L-700)^+$ with respect to $t$ at $t = 0$. As an alternative to the polynomial approximation (2.25), we can use approximation of the form

$$E_t(L-700)^+ \approx h + \frac{r}{1 + vt}$$

(2.31)

where $h$, $r$, and $v$ are real numbers chosen so that the 0th, 1st, and 2nd derivatives approximation match those of $E_t(L-700)^+$. In other words,

$$v = -\frac{\zeta_2}{2\zeta_1}, \quad r = -\frac{\zeta_1}{v}, \quad h = \zeta_0 - r$$

The following figure compares the 2nd-order polynomial approximation (2.25) with the above rational approximation (2.31):

![Figure comparing polynomial and rational approximations](image)

2.5 Sensitivity Analysis

For purposes of hedging and risk management, it is often necessary to calculate the sensitivity of the tranche price $E(L-y)^+$ with respect to default probabilities and correlation parameters. The methods in the previous sections are quite useful in sensitivity analysis, as described in the following.
<table>
<thead>
<tr>
<th></th>
<th>( \alpha_1 )</th>
<th>( \alpha_2 )</th>
<th>( \alpha_3 )</th>
<th>( \alpha_4 )</th>
<th>( \alpha_5 )</th>
<th>( \alpha_6 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>row 1 to 25</td>
<td>.6671</td>
<td>-2121</td>
<td>-1.155</td>
<td>.3186</td>
<td>-0.0718</td>
<td>-0.0001</td>
</tr>
<tr>
<td>row 26 to 50</td>
<td>.6671</td>
<td>2121</td>
<td>-1.155</td>
<td>.3186</td>
<td>-0.0718</td>
<td>-0.0001</td>
</tr>
<tr>
<td>row 51 to 75</td>
<td>.6671</td>
<td>-2121</td>
<td>-1.155</td>
<td>-.0972</td>
<td>.3118</td>
<td>-0.0001</td>
</tr>
<tr>
<td>row 76 to 100</td>
<td>.6671</td>
<td>2121</td>
<td>-1.155</td>
<td>-.0972</td>
<td>.3118</td>
<td>-0.0001</td>
</tr>
<tr>
<td>row 101 to 125</td>
<td>.6671</td>
<td>-2121</td>
<td>-1.155</td>
<td>-.2215</td>
<td>-2.4000</td>
<td>-0.0001</td>
</tr>
<tr>
<td>row 126 to 150</td>
<td>.6671</td>
<td>2121</td>
<td>-1.155</td>
<td>-.2215</td>
<td>-2.4000</td>
<td>-0.0001</td>
</tr>
<tr>
<td>row 151 to 175</td>
<td>.6671</td>
<td>-2121</td>
<td>.3464</td>
<td>.0000</td>
<td>.0000</td>
<td>.0002</td>
</tr>
<tr>
<td>row 176 to 185</td>
<td>.6671</td>
<td>2121</td>
<td>.3464</td>
<td>.0000</td>
<td>.0000</td>
<td>.0004</td>
</tr>
<tr>
<td>row 186</td>
<td>.6671</td>
<td>2121</td>
<td>.3464</td>
<td>.0000</td>
<td>.0000</td>
<td>-.0160</td>
</tr>
<tr>
<td>row 187 to 195</td>
<td>.6671</td>
<td>2121</td>
<td>.3464</td>
<td>.0000</td>
<td>.0000</td>
<td>.0017</td>
</tr>
<tr>
<td>row 196</td>
<td>.6671</td>
<td>2121</td>
<td>.3464</td>
<td>.0000</td>
<td>.0000</td>
<td>.0138</td>
</tr>
<tr>
<td>row 197 to 200</td>
<td>.6671</td>
<td>2121</td>
<td>.3464</td>
<td>.0000</td>
<td>.0000</td>
<td>.0004</td>
</tr>
</tbody>
</table>

Loading matrix for Example 2.4
2.5.1 Sensitivity with respect to correlation

By definition, the coefficient $\alpha_1$ in (2.17) (and $\zeta_1$ in (2.25)) is the rate at which the tranche price changes as the correlation structure deviates from an independent (single-factor) model. That is to say, $\alpha_1$ ($\zeta_1$) represents the "directional sensitivity" of the tranche price $E(L-y)^+$ with respect to the change of the correlation matrix in a certain direction. The sensitivity with respect to an individual correlation coefficient, say $\rho_{12}$, can be obtained by setting $\rho_{12} = 1$ and all other $\rho_{ij} = 0$ in the matrix (2.27). Thus, our approach lends itself naturally to the calculation of these sensitivities. Gregory and Laurent (2004) provide an alternative approach to sensitivity analysis, in which the effect of a perturbation of correlation matrix is analyzed through characteristic functions.

2.5.2 Sensitivity with respect to default probability

If obligors are independent, computing the sensitivity with respect to a default probability is easy (see also Andersen, Sidenius, and Basu, 2003):

$$\frac{\partial}{\partial p_i} E(L-y)^+ = \frac{\partial}{\partial p_i} [(1 - p_i) E(L_i-y)^+ + p_i E(L_i-y + c_i)^+]$$

$$= E(L_i-y + c_i)^+ - E(L_i-y)^+, \quad (2.32)$$

where $L_i$ is the reduced portfolio obtained by removing the $i$th obligor. When obligors are correlated, however, computing sensitivity with respect to a default probability $p_i$ is not as simple, because the dependence among obligors forbids us to separate $p_i$ and split the expectation in the manner of (2.32). But, according to the previous sections, a dependent-obligor model can be expressed as a weighted sum of independent-obligor models. Therefore, the problem of computing $p$-sensitivity in the dependent-obligor case reduces to that in the independent-obligor case. We now explain in more detail.

We first consider the case where the correlation is weak. Taking the first $n+1$ terms from the expansion (2.17) and differentiating with respect to $p_i$ yields the approximation

$$\frac{\partial}{\partial p_i} E_i(L-y)^+ \approx u_0 + u_1 t + \ldots + u_n \frac{t^n}{n!},$$
where $u_n$ is the derivative of $\alpha_n$ with respect to $p_i$. It follows from (2.18) that

$$
\sum_j w_j \frac{\partial}{\partial p_i} \tilde{E}_j(L-y)^+ \xrightarrow{s \downarrow 0} u_n
$$

(2.33)

We will calculate $u_n$ from this weighted sum, with each term computed in the following manner. Recall that each $\tilde{E}_j(L-y)^+$ is the tranche price in an independent-obligor model with default probabilities $\tilde{p}_1^{(j)}, \ldots, \tilde{p}_M^{(j)}$. To compute the derivative of $\tilde{E}_j(L-y)^+$ with respect to $p_i$, we simply use the chain rule: we first differentiate $\tilde{E}_j(L-y)^+$ with respect to $\tilde{p}_i^{(j)}$ — this yields an expression of the form (2.32) — then multiply the result by an extra factor $\partial \tilde{p}_i^{(j)}/\partial p_i$, whose analytical expression can be easily obtained by differentiating the equation (2.6). Evidently, computing $u_n$ is no more difficult than computing $\alpha_n$.

Now, we move on to the case where obligors are heavily correlated, with the covariance structure parameterized as in (2.23). Differentiating (2.25) yields

$$
\frac{\partial}{\partial p_i} E_t(L-y)^+ \approx U_0 + U_1 t + \ldots + U_n \frac{t^n}{n!}
$$

(2.34)

Through the same conditioning argument as in Section 2.3, we can show that

$$
U_k = E[u_k(Z)] = \int u_k(z) \varphi(z) \, dz
$$

(2.35)

where $u_k(z)$ is the coefficient in the expansion of $(\partial/\partial p_i) E_t[(L-y)^+|Z=z]$, computed in the same manner as (2.33). The integral (2.35) can then be evaluated numerically to yield $U_k$.

**Example 2.5** Consider the portfolio in the previous example. Suppose that we want to compute the 2nd-order approximation for $(\partial/\partial p_M) E(L-1000)^+$. Again, we use the first column of the loading matrix to generate the reference correlation structure $R$ in (2.23). We follow the above procedure (2.34–2.35) and obtain $U_0 = 4.3$, $U_1 = 4.1$, and $U_2 = 2.3$. Thus, the 2nd-order approximation for $(\partial/\partial p_M) E_t(L-1000)^+$ is $4.3 + 4.1 t + 2.3 t^2/2$. This approximation is shown as the solid line in Figure 2.8. The true values of the sensitivity at $t = 0, .2, .4, .6, .8, 1$ are marked with "o". They are computed (using...
Monte Carlo simulation) from the identity

\[
\frac{\partial}{\partial \rho_i} E_{\mathcal{N}(0, \Sigma)} (L-y)^+ = E_{\mathcal{N}(x_i, \Sigma, -c_i \Sigma^T)} [(L-y+c_i)^+ - (L-y)^+] .
\]

(The subscript of the expectation denotes the distribution of the latent variable \( X_i \)'s, and \( C_i \) denotes the \( i \)th column of \( C \).)

\[ \Box \]

2.6 Concluding Remarks

We have developed a method for approximating CDO prices in the multifactor Normal Copula model. This method expresses the CDO price of a multifactor model as a series of prices in independent-obligor models, thereby exploiting the fact that independent-obligor prices are easy to compute. We also demonstrate how the proposed approximation can be used to compute quantities relating to sensitivity analysis; namely, sensitivity with respect to correlation structure and sensitivity with respect to default probabilities. Additionally, in its most basic form, the method can be used to compute joint probabilities of multivariate normal random variables.
Chapter 3

Quadratic Transform Approximation

Laplace inversion methods have been a popular tool in valuing credit risk and pricing credit basket derivatives (see, for example, Laurent and Gregory (2003), Gregory and Laurent (2004), Lord and Kahl (2006), etc.). The reason behind the method’s popularity in pricing multiname credit derivatives is that the Laplace transform of the underlying portfolio is often easy to obtain. If obligors are independent, computing the Laplace transform is very simple (because the portfolio is simply a sum of independent random variables). By extension, for a single-factor model, one can make use of the fact that obligors become independent upon conditioning on the factor, and subsequently compute the conditional Laplace transform. Then, the process of unconditioning can be carried out simply by numerical integration to obtain the Laplace transform of the correlated obligors’ pool. Once we obtain the Laplace transform, we can use the Laplace inversion formulae, which are readily available in various forms, to invert the transform and yield the desired prices.

However, as we venture into models with large number of factors, the Laplace transforms are not so easy to obtain; this is because numerical integration, efficient as it may be in the single-factor case, now requires computing time that grows exponentially with the number of factors. The approximation scheme presented in this chapter is specifically designed to address this problem. In fact, our method bypasses the pro-
cess of numerical integration altogether, and results in a closed-form, easy-to-compute approximation for the Laplace transforms in models with multiple risk factors. This approximation, when used in place of the true Laplace transforms in the inversion formulae, offers a fast and accurate way of pricing CDOs and other credit derivatives in the multifactor Normal Copula model.

The organization of this chapter is as follows. In Section 3.1, we review the method of Laplace inversion in risk measurement and derivative pricing. Section 3.2 is devoted to describing our approximation method, which is shown to be quite accurate when the correlation among obligors is weak. Then, Section 3.3 explains how one can extend the method to cover the case of strong correlation among obligors. We analyze the speed and accuracy through numerical examples.

3.1 Pricing through Laplace Transform

Let \( i = \sqrt{-1} \) denote the imaginary unit. For a complex number \( s \), we denote its real part and its imaginary part by \( \text{Re} s \) and \( \text{Im} s \), respectively, so that \( s = \text{Re} s + i \text{Im} s \).

Let \( C^+ \) be the set of all complex number \( s \) such that \( \text{Re} s > 0 \). The Laplace transform of the loss portfolio \( L \) is defined as the mapping

\[
\phi(s) := \mathbb{E}[e^{-sL}], \quad s \in C^+.
\]

This mapping is very versatile, as it can be used to generate the Laplace transform of many quantities of interest: for example,

\[
\text{Transform of } P(L > y) := \int_0^\infty e^{-sy} P(L > y) \, dy = \frac{1 - \phi(s)}{s} \quad \text{(3.1)}
\]

\[
\text{Transform of } E(L \wedge y) := \int_0^\infty e^{-sy} E(L \wedge y) \, dy = \frac{1 - \phi(s)}{s^2} \quad \text{(3.2)}
\]

\[
\text{Transform of } E(L - y)^+ := \int_0^\infty e^{-sy} E(L - y)^+ \, dy = \frac{1 - \phi(s) - s\mathbb{E}[L]}{s^2} \quad \text{(3.3)}
\]
(In (3.2), we write \( L \wedge y \) to mean \( \min\{L, y\} \).) These quantities and their Laplace transforms correspond in a one-to-one manner through the inversion integral:

\[
P(L > y) = \frac{1}{i\pi} \int_{a-i\infty}^{a+i\infty} \text{Re}(e^{sy}) \text{Re}\left(\frac{1 - \phi(s)}{s}\right) \, ds
\]

(3.4)

\[
E(L \wedge y) = \frac{1}{i\pi} \int_{a-i\infty}^{a+i\infty} \text{Re}(e^{sy}) \text{Re}\left(\frac{1 - \phi(s)}{s^2}\right) \, ds
\]

(3.5)

\[
E(L - y)^+ = -\frac{1}{i\pi} \int_{a-i\infty}^{a+i\infty} \text{Re}(e^{sy}) \text{Re}\left(\frac{1 - \phi(s) - sE[L]}{s^2}\right) \, ds
\]

(3.6)

where the contour of integration is the vertical line \( \text{Re} s = a \) on the complex plane, where \( a \) is any positive real number. Assuming that \( \phi(s) \) is known, these integrals can be evaluated numerically to retrieve quantities on the left-hand side.

It is worth noting that the formulae (3.5) and (3.6), in fact, imply each other. To show this, we first rewrite (3.6) by breaking the fraction in the parenthesis into two parts:

\[
E(L - y)^+ = \frac{E[L]}{i\pi} \int_{a-i\infty}^{a+i\infty} \text{Re}(e^{sy}) \text{Re}\left(\frac{1}{s}\right) \, ds - \frac{1}{i\pi} \int_{a-i\infty}^{a+i\infty} \text{Re}(e^{sy}) \text{Re}\left(\frac{1 - \phi(s)}{s^2}\right) \, ds
\]

(3.7)

In arriving at this last equality, we have used the fact that, since \( 1/s \) is the Laplace transform of the Heaviside step function, the first integral — the inversion integral of \( 1/s \) — simplifies to one. One can see that (3.7) and (3.5) are consistent with the relation \( E(L - y)^+ = E[L] - E(L \wedge y) \); in this connection, the two formulae (3.5) and (3.6) are equivalent.

Note that the integrand in (3.7) is not continuous at \( s = 0 \) (because \( 1 - \phi(s) = \mathcal{O}(s) \)). However, there is an easy way to modify the inversion formula (3.7) so that the integrand is continuous everywhere on the complex plane. (This property will be useful later in Section 3.2.3.) First, note that, by substituting \( y = 0 \) in (3.7),

\[
0 = \frac{1}{i\pi} \int_{a-i\infty}^{a+i\infty} \text{Re}\left(\frac{1 - \phi(s)}{s^2}\right) \, ds
\]

Next, we multiply this equation by \( e^{sy} \), then subtract it from (3.7); as the result, we
have
\[ E(L - y)^+ = E[L] - \frac{1}{i\pi} \int_{-\infty}^{\infty} \text{Re}(e^{sv}) \frac{1 - \phi(s)}{s^2} \, ds \]  
(3.8)

Here, if we define the integrand to be zero at \( s = 0 \), then the integrand is continuous for all \( s \in \mathbb{C}^+ \), and the formula is valid for all \( a \geq 0 \).

All the inversion integrals above can be evaluated numerically using many discretization schemes. For example, if we use the trapezoidal rule with stepsize \( h \) to compute the integral (3.8) on the contour \( \text{Re} s = a = 0 \), we obtain
\[ E(L - y)^+ \approx E[L] - \frac{2h}{\pi} \sum_{k=1}^{\infty} \text{Re}(1 - \phi(ikh)) \frac{1 - \cos(khy)}{k^2h^2} \]  
(3.9)

This is analogous to the method of inverting characteristic functions described in Abate and Whitt (1992).

A sophisticated discretization scheme is given in Abate and Whitt (1995), where the trapezoidal rule with stepsize \( h = \pi/2y \) is applied to the integral (3.6), on a contour \( a = A/y \) (where \( A \) is a positive real number), to give the nearly-alternating series:
\[ E(L - y)^+ \approx E[L] - \frac{e^A y}{2} \sum_{k=-\infty}^{\infty} (-1)^k \text{Re} \frac{1 - \phi(A/y + i\pi k/y)}{(A + i\pi k)^2} \]  
(3.10)

The feature of this discretization scheme is that the discretization error decreases exponentially with \( A \) (see Abate and Whitt (1995) for more detail).

Assuming that \( \phi(s) \) is known, we can use the discretized versions of the inversion formulae to compute the tranche price \( E(L - y)^+ \). The appeal of pricing through inversion formulae is that the Laplace transform \( \phi(s) \) of the portfolio loss \( L \) is easy to compute, provided that the correlation among obligors has a low-factor structure, say, \( d \leq 3 \), where \( d \) is the length of the vector \( Z \) of the Gaussian risk factors. To obtain \( \phi(s) \), we take advantage of the factor structure (1.4), noting that obligors become independent upon conditioning on the factors \( Z \). The conditional probability that the \( i \)th obligor defaults is
\[ p_j(Z) := P(X_j > x_j | Z) = P\left( \sqrt{1 - a_j^2} a_j \varepsilon_j + a_j^T Z > x_j | Z \right) \]
\[ = \frac{1}{\phi\left( \frac{x_j - a_j^T Z}{\sqrt{1 - a_j^2 a_j}} \right)} \]  
(3.11)
It follows that the transform of the loss contributed by the $j$ obligor is

$$p_j(Z) \mathbb{E}[e^{-sY_j}] + (1 - p_j(Z))e^{-s^0} = 1 + (\psi_j(s) - 1) p_j(Z)$$

where $\psi_j(s) = \mathbb{E}[e^{-sY_j}]$ is the Laplace transform of $Y_j$, which we assumed to be known explicitly for all $s \in \mathbb{C}^+$. Because obligors are independent conditioned on $Z$, the transform of the entire portfolio loss is simply the product of individual names:

$$\mathbb{E}[e^{-sL(Z)}] = \prod_{j=1}^M \left( 1 + (\psi_j(s) - 1) p_j(Z) \right)$$  \hspace{1cm} (3.12)

The Laplace transform $\phi(s) = \mathbb{E}[e^{-sL}]$ is then obtained by unconditioning the above expression. If the number of factors is low, the process of unconditioning can be carried out efficiently by numerical integration. The difficulty arises, however, when the number of factor is high, in which case numerical integration becomes impractical, as the computing time increases exponentially with the number of factors. Likewise, Monte Carlo simulation may not be an appropriate tool for unconditioning (3.12), because the estimate produces large variance. Evidently, as the inversion formulae involve $\phi(s)$ at many values of $s$, there is a need for a fast and accurate method for computing the Laplace transform. As our main contribution, and as the focus of the rest of this chapter, we develop an analytical method for approximating the Laplace transform $\phi(s)$ which, combined with the aforementioned inversion formulae, offers a fast and accurate way of pricing CDO tranches.

### 3.2 Approximating $\phi(s)$

In this section, we propose a fast, analytical method for approximating the Laplace transform $\phi(s)$. The description of the method is given in Section 3.2.1 (for a preview, see Step 1–3 at the end of Section 3.2.1). Then, Section 3.2.2 analyzes the accuracy of the approximation through geometrical interpretation of the method. In Section 3.2.3, we state some convergence properties of the approximation.
3.2.1 Approximation Method

We will spend this section describing how we arrive at a procedure for approximating
the Laplace transform \( \phi(s) \) for any given \( s \in \mathbb{C}^+ \). The procedure will be summarized at
the end of the section.

For a complex scalar \( x \neq 0 \), let \( \log x := \log |x| + i(-1)^{\text{Im} x < 0} \arccos(\Re x / |x|) \), so
that \( \exp(\log x) = x \). For \( j = 1, \ldots, M \), let \( G_j \) be the mapping defined as

\[
G_j(t) := \log \left( 1 + (\psi_j(s)-1) \Phi(t) \right), \quad t \in \mathbb{R}
\]

where \( \psi_j(s) \) denotes the Laplace transform of the random loss \( Y_j \). Strictly speaking,
the notation on the left-hand side of the above definition should read \( G_j(t, s) \), since
its value depends on both \( t \in \mathbb{R} \) and \( s \in \mathbb{C}^+ \). However, because this section focuses on
approximating \( \phi(s) \) for a fixed \( s \), we may suppress the argument \( s \) throughout the section
so as to simplify notation. Later on, we will occasionally use the notation \( G_j(t, s) \),
whenever we want to stress that the mapping \( G_j \) also takes the argument \( s \).

Let \( U_1, \ldots, U_M \) be random variables defined as

\[
U_j := \frac{x_j - a_j^T Z}{\sqrt{1 - a_j^T a_j}}
\]

where \( Z \) is the vector of risk factors. Note that, since \( Z \) is a vector of \( \mathcal{N}(0,1) \) random
variables, \( U_j \) has a Gaussian distribution. By rewriting (3.11-3.12) in terms of \( G_j \) and
\( U_j \), and then unconditioning the factors, we have

\[
\phi(s) = \mathbb{E} \left[ \prod_{j=1}^M \exp G_j(U_j) \right] = \mathbb{E} \left[ e^{S(Z)} \right], \quad \text{where} \quad S(Z) := \sum_{j=1}^M G_j(U_j).
\]

The key idea behind our approximation method is as follows. We will approximate
\( S(Z) \) by a quadratic function of \( Z \), so that the above expectation, with \( S(Z) \) replaced
by its quadratic approximation, simplifies to a closed form. Thus, without having to
integrate with respect to the \( d \)-dimensional factors \( Z \), we obtain \( \phi(s) \) through the closed-
form approximation. We now proceed by describing the method of fitting \( S(Z) \) with a
quadratic function of \( Z \).

Suppose that \( G_j(U_j) \) in (3.15) admit the following approximation:

\[
G_j(U_j) \approx \beta_j + \gamma_j U_j + \eta_j U_j^2
\]
where the scalars $\beta_j$, $\gamma_j$, and $\eta_j$ are complex-valued. (We will describe how to obtain these coefficients later.) Using this approximation, it follows from the definition (3.15) of $S(Z)$ that

$$S(Z) \approx \sum_{j=1}^{M} \left( \beta_j + \gamma_j U_j + \eta_j U_j^2 \right) = b + g^T Z + Z^T H Z =: \hat{S}(Z) \quad (3.17)$$

Here, the scalar $b$, the vector $g$, and the matrix $H$ are the result of substituting $U_j$ from (3.14). (Because $U_j$'s are linear in $Z$, the above summation simplifies to a quadratic form of $Z$.) The explicit expressions for $b$, $g$, and $H$ are

$$b = \sum_{j=1}^{M} \left( \beta_j + \frac{\gamma_j x_j}{\sqrt{1-a_j^2}} + \frac{\eta_j x_j^2}{1-a_j^2} \right) \quad (3.18)$$

$$g = -\sum_{j=1}^{M} \frac{2\eta_j x_j + \gamma_j \sqrt{1-a_j^2}}{1-a_j^2} a_j \quad (3.19)$$

$$H = \sum_{j=1}^{M} \frac{\eta_j a_j a_j^T}{1-a_j^2} \quad (3.20)$$

At this stage, we have successfully approximated $S(Z)$ by the quadratic function $\hat{S}(Z)$, specified by (3.17-3.20). The Laplace transform $\phi(s) = \mathbb{E}[e^{s S(Z)}]$ can then be approximated by

$$\hat{\phi}(s) := \mathbb{E}[e^{\hat{S}(Z)}] = \mathbb{E}[e^{b + g^T Z + Z^T H Z}] \quad (3.21)$$

To evaluate this expectation, we refer to the following proposition:

**Proposition 3.1**  The expectation (3.21) simplifies to

$$\hat{\phi}(s) = \frac{1}{\sqrt{\det(I-2H)}} e^{b + g^T (I-2H)^{-1} g/2} \quad (3.22)$$

This formula, with $b$, $g$, and $H$ computed from (3.18-3.20) (assuming that $\beta_j$, $\gamma_j$, $\eta_j$ are known), provides a closed-form approximation for the Laplace transform $\phi(s)$. To complete our narrative, it is left only to explain how to obtain the coefficients $\beta_j$, $\gamma_j$, $\eta_j$ in the approximation (3.16).

We use the least-squares method as the criteria for choosing the triplet $(\beta_j, \gamma_j, \eta_j)$; the "best fit" is chosen to minimize the errors of (3.16) at certain gridpoints, say,
$U_j = t_1, \ldots, t_K$. That is to say, we choose $\beta_j$, $\gamma_j$, and $\eta_j$ that solve the following minimization problem:

$$
\min \sum_{k=1}^K w_k \left| G_j(t_k) - \beta_j - \gamma_j t_k - \eta_j t_k^2 \right|^2
$$

(3.23)

where the weights $w_1, \ldots, w_K > 0$ add up to one. Obviously, the gridpoints $t_k$ and the weights $w_k$ should be specified in such a way that they reflect the distribution of $U_j$ — larger weights should be assigned to the gridpoints around which $U_j$ is more likely to take a value. Since $U_j$ is a Gaussian random variable, it is natural to specify the gridpoints $t_k$ by

$$
t_1 = \mu_j + \lambda_1 \sigma_j, \; \ldots, \; t_K = \mu_j + \lambda_K \sigma_j
$$

(3.24)

where $\mu_j$ and $\sigma_j$ are the mean and the standard deviation of $U_j$, respectively, and $\lambda_1, \ldots, \lambda_K$ are real constants. Because $U_j$ rarely takes a value of more than two standard deviations away from the mean, $\lambda_1, \ldots, \lambda_K$ may be specified as ranging from $-2$ to $2$ (for example, $0, \pm 0.5, \pm 1, \ldots, \pm 2$). As for the weights, we suggest that $w_k$ be exponentially decreasing in $|\lambda_k|^2$, so as to reflect the Gaussian distribution of $U_j$.

The advantage of using the least-squares method to determine $\beta_j$, $\gamma_j$, and $\eta_j$ is that the optimization problem (3.23–3.24) has a unique, closed-form solution. To characterize the solution, first note from (3.14) that the mean $\mu_j$ and the standard deviation $\sigma_j$ of $U_j$ are

$$
\mu_j = \frac{x_j}{\sqrt{1 - \lambda_j^2 \sigma_j}} \quad \text{and} \quad \sigma_j = \frac{\sqrt{\lambda_j^2 \sigma_j}}{\sqrt{1 - \lambda_j^2 \sigma_j}}
$$

(3.25)

For a matrix $U = [u_{k\ell}]$ and for a positive integer $n$, let $U^n$ be the matrix (of the same dimension as $U$) defined by $U^n := [u_{k\ell}^n]$. In addition, where $G_j$ is the mapping defined in (3.13), let $G_j(U)$ denote the matrix of the same dimension as $U$ given by $G_j(U) := [G_j(u_{k\ell})]$. Where $\lambda_k$ are the constants that specify the gridpoints in (3.24), define a column vector $\mathbf{A}_j = [\lambda_1 \ldots \lambda_K]^T$. Note that $\mathbf{A}_j$ need not be the same for all $j$, as we are free to use different sets of gridpoints for different names $j$. As for the weights, let $w_j = [w_1 \ldots w_K]^T$ and let $W_j$ be the $K$-by-$K$ diagonal matrix whose $k$th element on the main diagonal is $w_k$. Lastly, define

$$
B_j = \left( [\mathbf{1} \mathbf{A}_j \mathbf{A}^2_j]^T W_j [\mathbf{1} \mathbf{A}_j \mathbf{A}^2_j] \right)^{-1} [\mathbf{1} \mathbf{A}_j \mathbf{A}^2_j]^T W_j
$$

(3.26)
where \( \mathbf{1} \) denotes the column vector of the same length as \( \mathbf{A}_j \) whose elements are all one.

The unique solution \((\beta_j, \gamma_j, \eta_j)\) to the minimization problem (3.23–3.24) is given by

\[
\begin{bmatrix}
\beta_j \\
\gamma_j \\
\eta_j
\end{bmatrix} = 
\begin{bmatrix}
1 & \mu_j & \mu_j^2 \\
0 & \sigma_j & 2\mu_j\sigma_j \\
0 & 0 & \sigma_j^2
\end{bmatrix}^{-1} 
\begin{bmatrix}
\mathbf{B}_j \\
\mathbf{G}_j(\mu_j + \mathbf{A}_j\sigma_j)
\end{bmatrix}
\quad (3.27)
\]

(Here, we write \( \mu_j + \mathbf{A}_j\sigma_j \) to mean \( \mu_j\mathbf{1} + \mathbf{A}_j\sigma_j \), where \( \mathbf{1} \) is the vector of the same length as \( \mathbf{A}_j \) whose elements are all one.)

At this point we have given the full description of the method of approximating the Laplace transform \( \phi(s) \). We now summarize the approximation method as a 3-step procedure:

**Summary** To approximate the Laplace transform \( \phi(s) \),

1. **Step 1** For each \( j = 1, \ldots, M \), compute \( \beta_j, \gamma_j, \) and \( \eta_j \) from (3.27).
2. **Step 2** Compute \( \mathbf{b}, \mathbf{g} \) and \( \mathbf{H} \) from (3.18–3.20).
3. **Step 3** The approximant for \( \phi(s) \) is

\[
\hat{\phi}(s) = \frac{1}{\sqrt{\det(\mathbf{I} - 2\mathbf{H})}} \mathbf{e}^\mathbf{b} + \mathbf{g}^\top(\mathbf{I} - 2\mathbf{H})^{-1} \mathbf{g}/2
\]

This 3-step procedure provides a fast, analytical way of approximating the Laplace transform \( \phi(s) \). The approximant \( \hat{\phi}(s) \) can then be used in place of \( \phi(s) \) in the inversion formulae (from Section 3.1) to compute an approximation for the tranche price \( \mathbb{E}(L-y)^+ \) or other quantities of interest.

To end this section, we note that \( \hat{\phi}(s) \), written as the expectation in (3.21), will blow up unless the matrix \( \mathbf{H} \) satisfies certain conditions. A convenient way to ensure finiteness of \( \hat{\phi}(s) \) is to impose that \( \text{Re} \eta_j \leq 0 \) for all \( j \); then, as seen in (3.20), \( \text{Re} \mathbf{H} \) is negative semidefinite, and, consequently, the exponential function inside the expectation (3.21) is integrable. So, to ensure that \( \text{Re} \eta_j \leq 0 \) for all \( j \), we modify the above procedure by adding the following intervening step. After Step 1, for all \( j \in \{1, \ldots, M\} \) such that \( \eta_j \) already satisfies \( \text{Re} \eta_j \leq 0 \), there is no need to make any alterations. For every \( j \) such that \( \text{Re} \eta_j > 0 \), however, we re-set \( \text{Re} \eta_j = 0 \) and re-solve the least-squares problem (3.23), which, as before, has a closed-form solution; that is,
Extra step (inserted right after Step 1) For every $j \in \{1, \ldots, M\}$ such that $\Re \eta_j > 0$, keep $\Im \beta_j$, $\Im \gamma_j$, and $\Im \eta_j$ unaltered, but re-set $\Re \eta_j = 0$ and re-calculate $\Re \beta_j$ and $\Re \eta_j$ using the following formula:

$$\begin{bmatrix} \Re \beta_j \\ \Re \gamma_j \end{bmatrix} = \begin{bmatrix} 1 & \mu_j \\ 0 & \sigma_j \end{bmatrix}^{-1} \begin{bmatrix} 1 & W_j^T A_j \\ W_j^T A_j & W_j^T A_j^2 \end{bmatrix}^{-1} \begin{bmatrix} w_j \\ W_j A_j \end{bmatrix}^T G_j \left( \mu_j + A_j \sigma_j \right)$$

(3.28)

where $w_j$, $W_j$, and $A_j$ are as defined in the paragraph above equation (3.26).

Although we have taken this precaution to ensure that we enter Step 2 with $\Re \eta_j \leq 0$ for all $j$, we will see later (Section 3.2.2) that, typically, the $\eta_j$'s produced by Step 1 already satisfy $\Re \eta_j \leq 0$, so the extra step above is often automatically skipped.

It should be noted that the method described so far can, with appropriate modification, be applied to a generalization of the Normal Copula model referred to as the Random Factor Loadings (RFL) models. For detailed description of the RFL models, we refer to Andersen and Sidenius, 2004.

In the next few sections, we will attempt to answer the question of how accurate the approximant $\hat{\phi}(s)$ is, and what model parameters determine its accuracy. Then, we will provide numerical examples in Section 3.2.4.

### 3.2.2 Geometrical Interpretation

By giving the method in Section 3.2.1 a geometrical interpretation, this section provides an intuitive way of analyzing the method's accuracy.

To begin, note that the approximation (3.16) can be decomposed as:

$$\Re G_j(U_j) \approx \Re \beta_j + U_j \Re \gamma_j + U_j^2 \Re \eta_j$$

(3.29)

$$\Im G_j(U_j) \approx \Im \beta_j + U_j \Im \gamma_j + U_j^2 \Im \eta_j$$

(3.30)

Correspondingly, the optimization problem (3.23) can be decomposed, using the identity $|x|^2 = (\Re x)^2 + (\Im x)^2$, into two separate optimization problems: one of minimizing the error of (3.29), the other the error of (3.30). (The solutions to these two problems are, respectively, the real-part and the imaginary-part of (3.27).)
Recall that the approximant $\tilde{\phi}(s)$ is based on the approximation $S(Z) \approx \tilde{S}(Z)$ in (3.17), which in turn is based on the approximation (3.16). Therefore, the accuracy of $\tilde{\phi}(s)$ hinges entirely on the quality of the approximation (3.16), which consists of the two parts displayed in (3.29–3.30). That is to say, if both $\text{Re} G_j(U_j)$ and $\text{Im} G_j(U_j)$ are approximately quadratic in $U_j$, then $\tilde{\phi}(s)$ is an accurate approximant of $\phi(s)$. Conversely, if there exists no quadratic curves that fit $\text{Re} G_j(U_j)$ and $\text{Im} G_j(U_j)$ satisfactorily, then $\tilde{\phi}(s)$ is not an appropriate approximation for $\phi(s)$. In this connection, to analyze the accuracy of $\tilde{\phi}(s)$ it is important to investigate the shape of the curves $\text{Re} G_j(t)$ and $\text{Im} G_j(t)$ ($t \in \mathbb{R}$). The following proposition summarizes their properties.

**Proposition 3.2** If $\psi_j(s) = 1$ then $G_j(t) = 0$ for all $t$. Assume that $\psi_j(s) \neq 1$.

(i) $\lim_{t \to -\infty} G_j(t) = \log \psi_j(s)$

(ii) $|G_j(t)| < 3 \Phi(t)$ for all $t > 2$.

(iii) $\text{Im} G_j(t)$ is monotone in $t$.

(iv) If $\text{Re} \psi_j(s) \geq |\psi_j(s)|^2$, then $\text{Re} G_j(t)$ is nondecreasing in $t$. Otherwise, $\text{Re} G_j(t)$ is unimodal, attaining its unique minimum at $t = \Phi^{-1}(\text{Re} 1/(\psi_j(s) - 1))$.

(v) The curve $\text{Re} G_j(t)$ is concave at all $t$ satisfying

$$\text{Re} \frac{t}{\Phi(t)-(1-\psi_j(s))^{-1}} < \text{Re} \frac{\varphi(t)}{[\Phi(t)-(1-\psi_j(s))^{-1}]^2}$$

In particular, $\text{Re} G_j(t)$ is concave for all $t > 0.9$.

This proposition allows us to visualize the shape of the curves $\text{Re} G_j(t)$ and $\text{Im} G_j(t)$. Note that Item (i) specifies the limit as $t \to -\infty$ of $\text{Re} G_j(t)$ and $\text{Im} G_j(t)$. Item (ii) implies that both $\text{Re} G_j(t)$ and $\text{Re} G_j(t)$ are practically flat when $t > 2$. The following figures give a visual interpretation of the approximations (3.29–3.30).

Since $U_j$ is Gaussian with mean $\mu_j$ and variance $\sigma_j^2$, the shaded region, spanning from $\mu_j - 2\sigma_j$ to $\mu_j + 2\sigma_j$, indicates the important range of $U_j$. To say that the approximations (3.29–3.30) of $\text{Re} G_j(U_j)$ and $\text{Im} G_j(U_j)$ are accurate means, in geometrical terms, that the solid curves are approximately parabolic in the shaded stripe. The dotted parabolas represent the best fit obtained from the least-squares optimization (3.23). Note that, since we assign larger weights $w_k$ to the gridpoints nearer to $\mu_j$ in the minimization problem (3.23), the parabolas fit better around the center of the shaded stripe.
Figure 3.1: Illustration of the approximation (3.29–3.30).

Figure 3.1 above depicts the desirable scenario: the stripe’s midpoint ($\mu_j$) lies where both $\text{Re} \, G_j(U_j)$ and $\text{Im} \, G_j(U_j)$ are smooth, and the stripe’s coverage ($4\sigma_j$) is not too wide; as a result, the dotted parabolas fit the curves quite well in the shaded region. In general, because the curves $\text{Re} \, G_j(t)$ and $\text{Im} \, G_j(t)$ are smooth, it is not so much the position of stripe but its width that determines the quality of the fit; small $\sigma_j$ limits the range of $U_j$, thereby allowing for better approximation. Now, note from (3.25) that $\sigma_j$ increases proportionately to $\sqrt{a_j^* a_j}$. Hence, smaller values of $\sqrt{a_j^* a_j}$ ($j = 1, \ldots, M$) result in smaller $\sigma_j$ and, as explained above, imply better accuracy of the approximations (3.29–3.30). Using the fact that $\sqrt{a_j^* a_j} \leq |a_{j1}| + \ldots + |a_{jd}| \leq \|A\|_\infty$, where $\|A\|_\infty$ is the $\infty$-norm of the loading matrix (the $\infty$-norm of a matrix $U = [u_{ij}]$ is defined
by \( \|U\|_\infty := \max_i \sum_j |u_{ij}| \), we conclude that \( \hat{\phi}(s) \) is an accurate approximation for \( \phi(s) \) when \( \|A\|_\infty \) is small. Because the \( \infty \)-norm of matrices appear frequently in our subsequent analysis, we write \( \|A\| \) to mean \( \|A\|_\infty \) for the sake of simplicity.

Before we end this section, let us revisit the procedure summarized at the end of Section 3.2.1 — particularly the extra step (3.28), to be executed after Step 1 to ensure that \( \Re \eta_j \leq 0 \) for all \( j \). We mentioned that, in general, this extra step is automatically skipped because \( \eta_j \), as calculated from (3.27), already satisfies \( \Re \eta_j \leq 0 \). We now explain the reason behind this assertion. Note that \( \mu_j \) is in reverse proportion to the default probability \( p_j \) (see (3.25) and recall that \( p_j = \Phi(x_j) \)). Because the default probabilities are typically small, \( \mu_j \) — the center of the shaded area — usually lies to the far right on the horizontal axis. Since the curve \( \Re G_j(t) \) tends to be concave in this area (see Proposition 3.2), it follows that the fitting parabola will also be concave. This implies that, typically, \( \Re \eta_j \leq 0 \). Thus, the extra step (3.28), while necessary, is rarely executed.

### 3.2.3 Convergence Theorem

As explained in the previous section, \( \hat{\phi}(s) \) is a good approximation of the Laplace transform \( \phi(s) \) when \( \|A\| \) is small. In fact, it is easy to show that \( |\phi(s) - \hat{\phi}(s)| \to 0 \) as \( \|A\| \to 0 \); in other words, the approximant \( \hat{\phi}(s) \) is exact when \( \|A\| \to 0 \). This section poses the question: how fast does \( \hat{\phi}(s) \) converge to \( \phi(s) \)? Theorem 3.1 and Theorem 3.2 below answer this question by giving the rate of convergence. Both theorems assume that the gridpoints \( A_j \) and the weights \( w_j \) used in the formula (3.26-3.27) are the same for every name \( j \); that is, \( A_1 = \ldots = A_M = A \) and \( w_1 = \ldots = w_M = w \), where \( A \) and \( w \) are constant vectors. In stating the theorems, the following notations are useful. For a positive integer \( n \), let \( \lambda^n \) denote the scalar constant defined by \( \lambda^n := w^T \Lambda^n \). Where \( G_j \) is the mapping defined in (3.13) and \( x_j = \Phi^{-1}(p_j) \), define

\[
D_n(s) := \lim_{t \to 0} \frac{\partial^n}{\partial t^n} \begin{bmatrix} G_1(x_1 + t, s) \\ \vdots \\ G_M(x_M + t, s) \end{bmatrix} = \begin{bmatrix} G_1^{(n)}(x_1, s) \\ \vdots \\ G_M^{(n)}(x_M, s) \end{bmatrix} \quad (3.31)
\]
We will often suppress the argument $s$ in $D_n(s)$ to simplify notation. Next, let $\phi_0(s)$ denote the Laplace transform of the loss portfolio $L$ assuming that obligors are independent; that is,

$$\phi_0(s) = \prod_{j=1}^{M} \left( 1 + (\psi_j(s)-1)p_j \right)$$

Lastly, for a square matrix $U$, let $\text{diag}(U)$ denote the square matrix of the same dimension as $U$ whose elements on the main diagonal are those of $U$.

**Theorem 3.1** Assume that the default probabilities $p_1, \ldots, p_M$ are less than 0.5. Let $\phi(s)$ be the Laplace transform of $L$, and let $\hat{\phi}(s)$ be its approximant obtained from the 3-step procedure in Section 3.2.1. Suppose that the gridpoints $\Lambda$ and the weights $w$ are arranged symmetrically around zero (so that $\Lambda_n = 0$ whenever $n$ is odd). Then, there exist real constants $\delta$ and $C$ such that for all loading matrices $A$ with $\|A\| < \delta$,

$$|\phi(s) - \hat{\phi}(s)| < C \|A\|^4, \quad \forall s \in C^+$$

(3.32)

In other words, $\hat{\phi}(s)$ converges to $\phi(s)$ at the rate of $\|A\|^4$. For a fix $s \in C^+$, the rate of convergence is given explicitly by

$$\frac{\phi(s) - \hat{\phi}(s)}{\phi_0(s)} = r D_1^T \text{diag}(AA^T) 1 + v D_1^T \text{diag}(AA^T)AA^T D_1 + O(\|A\|^6)$$

(3.33)

where $1$ is the column vector of length $M$ whose elements are all one. The scalars $r$ and $v$ are given by $24r = 3 - \lambda^3 + (\lambda^2 - 1)(\lambda^6 - \lambda^2 \lambda^4)/\lambda^4$ and $4v = 3 - \lambda^3/\lambda^2$. □

The proof of Theorem 3.1 is given in the appendix. The next result (Theorem 3.2 below) shows that $\hat{\phi}(s)$ converges at an even faster rate provided that we impose stronger conditions on $A$ and $w$. Its proof, likewise, is deferred to the appendix.

**Theorem 3.2** Same premise as Theorem 3.1. Suppose, in addition, that $\lambda^2$, $\lambda^4$, and $\lambda^6$ match the second, fourth, and sixth moments, respectively, of the standard normal random variable; that is to say, $\lambda^2 = 1$, $\lambda^4 = 3$, and $\lambda^6 = 15$. Then, there exists real constant $\delta$ and $C$ such that for all loading matrix $A$ with $\|A\| < \delta$,

$$|\phi(s) - \hat{\phi}(s)| < C \|A\|^6, \quad \forall s \in C^+$$

(3.34)
In other words, \( \hat{\phi}(s) \) converges to \( \phi(s) \) at the rate of \( \|A\|^6 \). For a fix \( s \in \mathbb{C}^+ \), the rate of convergence is given explicitly by

\[
\frac{\phi(s) - \hat{\phi}(s)}{\phi_0(s)} = \frac{1}{2} D_3^T \text{diag}(AA^T)AA^TD_1 + \frac{1}{12} D_3^T (AA^T)^3 D_3 + \mathcal{O}(\|A\|^7)
\]

(3.35)

where \( \Pi \) is the sum of the square matrix \( D_1D_1^T/3 \) and the \( M \)-by-\( M \) diagonal matrix whose \( j \)th element on the main diagonal is \( G_j''(x_j,s) \).

Next, we will show that the convergence rates in Theorem 3.1 and Theorem 3.2 are preserved throughout the process of Laplace transform inversion. Let \( F(y) \) be the approximant of \( E(L - y)^+ \) obtained by replacing \( \phi(s) \) in (3.8) with \( \hat{\phi}(s) \); that is,

\[
F(y) := E[L] - \frac{1}{i\pi} \int_{a-i\infty}^{a+i\infty} \text{Re}(e^{sy} - e^{sy}) \text{Re}\left(\frac{1 - \hat{\phi}(s)}{s^2}\right) \, ds
\]

**Corollary 3.1** Under the assumptions of Theorem 3.1, \( F(y) \) converges to \( E(L - y)^+ \) at the rate of \( \|A\|^4 \) for all real number \( y \). If, in addition, \( \lambda^2 = 1, \lambda^4 = 3, \) and \( \lambda^6 = 15 \), then \( F(y) \) converges to \( E(L - y)^+ \) at the rate of \( \|A\|^6 \) for all real number \( y \).

**Proof** Assume the premise of Theorem 3.1. From the definition of \( F(y) \) and from (3.8),

\[
|F(y) - E(L - y)^+| = \left| \frac{1}{i\pi} \int_{a-i\infty}^{a+i\infty} \text{Re}(e^{sy} - e^{sy}) \text{Re}\left(\frac{\phi(s) - \hat{\phi}(s)}{s^2}\right) \, ds \right|
\]

\[
< \frac{1}{\pi} \int_{-\infty}^{\infty} e^{ay} (1 - \cos\omega y) \frac{C\|A\|^4}{a^2 + \omega^2} \, d\omega
\]

\[
= C\|A\|^4 \frac{e^{ay} - 1}{a}
\]

The second line results from the change of variable \( s = a + i\omega \). Also, we have used the fact that for all complex numbers \( x \) and \( y \), \( |\text{Re}(x/y)| \leq |x|/|y| \). Here, \( C \) is the constant from Theorem 3.1. Thus, we have proved the \( \|A\|^4 \) rate of convergence. The convergence rate of \( \|A\|^6 \) can be shown in the same manner.

**3.2.4 Numerical Examples**

**Example 3.1** Consider a credit portfolio of \( M = 50 \) names. Each obligor belongs to one of the following five groups, depending on its default probability: \( p_i = 2\% \) for
\( i = 1, \ldots, 10; \ p_i = 4\% \text{ for } i = 11, \ldots, 20; \ p_i = 6\% \text{ for } i = 21, \ldots, 30; \ p_i = 8\% \text{ for } i = 31, \ldots, 40; \text{ and } p_i = 10\% \text{ for } i = 41, \ldots, 50. \) The correlation among the obligors is represented by the following 5-factor loading matrix:

\[
\begin{bmatrix}
4 & 3 & 4 & 3 & 4 \\
3 & 4 & 3 & 4 & 3 \\
4 & 3 & 4 & 3 & 4 \\
3 & 4 & 3 & 4 & 3 \\
4 & 3 & 4 & 3 & 4 \\
\end{bmatrix}
\]

The losses-given-default \( Y_1, \ldots, Y_M \) are discrete-valued random variables given by \( Y_j = (1 + T_j)/10, \text{ where } T_1, \ldots, T_M \text{ are independent Poisson random variables with mean } 10. \) Note that the Laplace transform of \( Y_j \) is

\[
\psi_j(s) = \exp(-.1s - 10 + 10e^{-1s})
\]

Let the random variable \( L \) be the loss of this portfolio. We want to approximate the Laplace transform \( \phi(s) = \mathbb{E}[e^{-sL}]. \)

Using the 3-step procedure described above, we can approximate \( \phi(s) \) for any given \( s \in \mathbb{C}^+. \) The following table compares the approximation with the true values for some sample \( s. \) One can see that the approximation method performs quite well (at least, up to the second decimal).

<table>
<thead>
<tr>
<th>Specimen</th>
<th>True value of ( \phi(s) )</th>
<th>Approximant ( \tilde{\phi}(s) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( s = 0.61 + 0.13i )</td>
<td>.270 - .046i</td>
<td>.267 - .046i</td>
</tr>
<tr>
<td>( s = 0.23 - 1.67i )</td>
<td>.019 + .057i</td>
<td>.017 + .055i</td>
</tr>
<tr>
<td>( s = 0.72i )</td>
<td>-.047 - .318i</td>
<td>-.055 - .317i</td>
</tr>
</tbody>
</table>

| Table 3.1: Showing the performance of the 3-step method. |

When \( \text{Re } s = 0, \) the Laplace transform \( \phi(s) \) has a specific meaning: the mapping \( \omega \mapsto \phi(-i\omega) \) is referred to as the characteristic function of \( L. \) Figure 3.2 shows the performance of the 3-step method, when used to approximate the characteristic function.
Figure 3.2: Characteristic function of $L$.

The solid line represents the true value, while the dotted line represents the approximation. As can be seen, the approximation coincides almost perfectly with the true value.

Note that, as a property of any characteristic functions, the real part is an even function of $\omega$, while the imaginary part is an odd function. Also, the discreteness of a random variable has a particular influence over the behavior of its characteristic function. In this example, because of the way $Y_j$'s are defined, the portfolio loss $L$ only takes values that are multiples of 0.1. This results in the characteristic function being periodic, with the period of $20\pi$ (this means, for instance, that the next maximum of the real part occurs at $\omega = 20\pi$). In general, if a random variable is discretized to be multiples of $h$, its characteristic function will have a period of $2\pi/h$.

Even though these behaviors of $\phi(s)$ as a function of $s = -j\omega$ are worth mentioning, they do not, however, affect the performance of the 3-step approximation method, which computes the approximant $\tilde{\phi}(s)$ one $s$ at a time. (The parameter that does affect the accuracy of the method directly is, as explained in Section 3.2.2, the norm of the loading matrix $A$.) But the knowledge of the periodicity is helpful in drawing $\phi(i\omega)$ for all
\( \omega \in \mathbb{R} \), for we need to approximate only one period of the characteristic function.

Once we obtain the approximant \( \tilde{\phi} \) for characteristic function of \( L \), we can use it in place of \( \phi \) in the inversion integral (3.8) to obtain an approximation for the tranche price \( \mathbb{E}(L - y)^+ \). (The integration can be carried out numerically using the trapezoidal rule.) Figure 3.3 below shows the approximation versus the true value of the tranche price \( \mathbb{E}(L - y)^+ \) for a range of \( y \). Evidently, the approximation performs well for a wide range of \( y \). However, since \( \tilde{\phi} \) is accurate only for two decimal places (see Table 3.1), the approximation for \( \mathbb{E}(L - y)^+ \) is a bit off the mark for \( y > 8 \), where the price drops near \( 10^{-2} \).

\[ \square \]

**Example 3.2** Consider the same portfolio as the previous example, but instead of \( \omega \) having a discrete distribution, we let \( \omega \) be independent, continuous random variables having the Gamma distribution with parameters \((11, 10.31)\) (therefore, the mean is \( 11/10.31 \approx 1.067 \)). This distribution may be viewed as a continuous counterpart to the discrete distribution in the previous example. Figure 3.2 shows the probability mass of \( \omega \) from the previous example (shown as black dots) versus the histogram of the Gamma\((11, 10.31)\) distribution (the gray bars). The Laplace transform of the
newly defined $Y_j$ is

$$
\psi_j(s) = \left( \frac{10.31}{s + 10.31} \right)^{11} \quad (3.36)
$$

Let $L$ be the loss of this portfolio. As we did in the previous example, we will compute the approxim atation for the characteristic function of $L$. The approximation procedure is, apart from inputting new $\psi_j(s)$'s, exactly the same as the previous example's. The following figure shows the resulting approximation (the dotted curve), which coincides almost perfectly with the true characteristic function (the solid curve).

Because the distribution of $Y_i$ in this example is similar to that in the previous example, it comes as no surprise that the characteristic function shown here (we have omitted the plot of the imaginary part) is almost identical to that in Figure 3.2. But, unlike the previous example, the characteristic function in this example is not periodic; instead it converges to a limit as $\omega \to \pm \infty$ (here, the convergence starts when $|\omega| > 10$). To understand this phenomenon, note from (3.36) that $\psi_j(-i\omega)$ converges to zero very quickly as $\omega \to \pm \infty$; hence, the characteristic function is practically independent of $\omega$. 

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
once $|\omega| > 10$. This tail behavior, at any rate, does not have any particular impact on the accuracy of the approximation, as the method performs just as well here as in the previous example. The following plot shows the result of using the approximated characteristic function in the inversion formula (3.8) to obtain the tranche price.

Just as in the previous example, the approximation here is only accurate up to the second decimal.

3.3 Strong Correlation Case

As explained in Section 3.2.2, if $\|A\|$ is large, or, in other words, if the correlation among the obligors are strong, then the method in Section 3.2 may not give an accurate approximation for the Laplace transform $\phi(s)$. In response to this problem, this section presents two remedies that, once incorporated into the method from Section 3.2, allows us to approximate $\phi(s)$ in the case of strong correlation. These remedies take advantage of the fact that when obligors are highly correlated, the correlation matrix can often be factorized so that the loading matrix $A$, while multicolumn, has only a few columns that carry most of the loadings. The factor associated with the column that carries the heaviest load is often referred to as the market factor. The two remedies, both are ways
of manipulating the market factor, are as follows.

3.3.1 Conditioning on the Market Factor

In this section, we demonstrate how a simple conditioning argument can be used in conjunction with the procedure in Section 3.2.1 to approximate \( \phi(s) \) in the case where \( \|A\| \) is large.

Assume, without loss of generality, that the factor \( Z_1 \) associated with the first column of the loading matrix \( A \) is the market factor. Let \( \phi_z(s) := E[ e^{-sL} | Z_1 = z] \) be the Laplace transform of \( L \) conditioned on \( Z_1 = z \). It follows that

\[
\phi(s) = \int_{-\infty}^{\infty} \phi_z(s) \varphi(z) \, dz \tag{3.37}
\]

Note that, since \( L \) follows a Normal Copula model, for fix \( z \) the conditioned loss portfolio \( (L|Z_1 = z) \), likewise, follows a Normal Copula model (with default probabilities and factor loadings modified from those of the unconditioned model). While the \( \infty \)-norm of the loading matrix \( A \) of the unconditioned model may be large, the \( \infty \)-norm of that of the conditioned model, on the other hand, is significantly smaller because the conditioning process takes away the factor that carries most of the loadings. The procedure in Section 3.2.1 can then be used straightforwardly to approximate the conditional Laplace transform \( \phi_z(s) \) for any given \( z \). The remaining step of unconditioning, namely (3.37), can be carried out by numerical integration to obtain an approximation for \( \phi(s) \).

**Example 3.3** Consider a CDO structure with 100 names and five-year maturity. The obligors have equal unit nominal and are are divided into four groups according to their spreads: 15 basis points for name 1 to 25, 70 basis points for name 26 to 50, 150 basis points for name 51 to 75, and 230 basis points for name 76 to 100. The correlation
structure among the names is specified by the following loading matrix:

\[
A = \begin{bmatrix}
0.8 & 0.4 \\
0.6 & 0.4 \\
0.4 & 0.4 \\
0.2 & 0.4 \\
0.1 & 0.4 \\
\end{bmatrix}
\]

where each block represents a column vector with 25 elements. The recovery rates are assumed to be deterministic and equal to 30%. We will approximate the Laplace transform \( \phi(s) \) of this credit portfolio using i) the method from Section 3.2 with no conditioning argument, versus ii) the combination of the conditioning argument (3.37) and the method from Section 3.2.

Note that for \( j = 1, \ldots, 50, \sqrt{a_j^* a_j} \) are quite large, and consequently so is \( \|A\| \). As a result, one would not expect the method in Section 3.2 to perform well on its own. But, this shortfall can be overcome, as seen in the following table, by adding the remedial step of conditioning on the first (market) factor before applying the approximation method.

<table>
<thead>
<tr>
<th>Specimen ( s )</th>
<th>True value of ( \phi(s) )</th>
<th>Using the method in Section 3.2.1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( .245 + .085i )</td>
<td>( .220 + .089i )</td>
</tr>
<tr>
<td></td>
<td>( .223 - .405i )</td>
<td>( .185 - .435i )</td>
</tr>
<tr>
<td></td>
<td>( -.026 - .149i )</td>
<td>( -.066 - .129i )</td>
</tr>
</tbody>
</table>

Table 3.2: Illustrate the effect of the conditioning argument.

The improvement seen in the last column comes with a price, however. In this example, we evaluate the integral (3.37) by the trapezoidal rule using 31 values of \( z \). This means that, in order to obtain each value on the last column of Table 3.2, we must call to the 3-step procedure in Section 3.2 multiple times to approximate \( \phi_z(s) \) for different values of \( z \). In contrast, each approximant on the second-to-last column
requires only one call to the approximation procedure. Fortunately, this increase in computer time is not significant, since each call to the procedure takes very little amount of time.

To see the performance on approximating $\phi(s)$ for a larger sample of $s$, consider Figure 3.5 below. The left plot shows the performance of the procedure in Section 3.2 in approximating $\phi(i\omega)$, $\omega \in [.2,.8]$. The second plot shows the improvement resulting from incorporating the conditioning argument.

![Figure 3.5: Compare two methods in approximating $\phi(i\omega)$](image)

Now that we can approximate $\phi(i\omega)$ accurately, we substitute $\phi(i\omega)$ by its approximation in the inversion formula (3.9) to obtain an approximation for the tranche price $E(L - y)^+$. Table 3.3 shows the result for some values of $y$. As one might expect, accuracy in approximating the Laplace transform transfers to accuracy in approximating the tranche prices.

<table>
<thead>
<tr>
<th>$y$</th>
<th>True $E(L - y)^+$</th>
<th>Approximation</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.7419</td>
<td>0.7416</td>
</tr>
<tr>
<td>15</td>
<td>0.3071</td>
<td>0.3047</td>
</tr>
<tr>
<td>20</td>
<td>0.1389</td>
<td>0.1312</td>
</tr>
</tbody>
</table>

Table 3.3: Approximating tranche prices

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
3.3.2 Segmenting the Market Factor

To describe the idea of segmenting the market factor, we begin by explaining how the procedure in Section 3.2.1 can be applied to approximate expectation of the form

$$E\left[ e^{S(Z)} 1\{ z < Z_1 \leq \bar{z} \} \right] ,$$

where $S(Z)$ are given in (3.15) and $z < \bar{z}$ are real numbers. Following the same line of arguments in Section 3.2.1, we approximate this expectation by replacing $S(Z)$ with its quadratic approximation $\tilde{S}(Z) = b + g^T Z + Z^T H Z$, so that the expectation simplifies to a closed form (we will provide the closed-form expression presently). Recall that $b$, $g$, and $H$ are computed, using (3.18-3.20), from the coefficients $\beta_j$, $\gamma_j$, and $\eta_j$ ($1 \leq j \leq M$) in the approximation (3.16). Back then, $[\beta_j \gamma_j \eta_j]$ are calculated (see (3.27)) from $\mu_j$ and $\sigma_j$ — the mean and standard deviation of $U_j$ (defined in (3.14)). But now, since the desired expectation (3.38) is taken upon the event $z < Z_1 \leq \bar{z}$, we will execute the formula (3.27) with $\mu_j$ and $\sigma_j$ replaced by

$$\tilde{\mu}_j := E[U_j | z < Z_1 \leq \bar{z}] = \mu_j + \frac{a_{j1}}{\sqrt{1 - a_j^2 \sigma_j^2}} \frac{\Phi(\bar{z}) - \Phi(z)}{\Phi(z) - \Phi(z)}$$

(3.39)

$$\tilde{\sigma}_j^2 := \text{Var}(U_j | z < Z_1 \leq \bar{z}) = \sigma_j^2 - (\mu_j - \tilde{\mu}_j)^2 + \frac{a_{j1}^2}{1 - a_j^2 \sigma_j^2} \frac{\Phi(\bar{z}) - \Phi(z)}{\Phi(z) - \Phi(z)}$$

(3.40)

where $\mu_j$ and $\sigma_j$ are as given in (3.25), and $\phi'(x) = -x \phi(x)$ is the derivative of $\phi(x)$. The derivation of (3.39-3.40) is straightforward and is left to the readers. Note that, under the event that $Z_1 \in (z, \bar{z}]$, the Gaussian assumption of $U_j$ no longer holds. However, if the interval $(z, \bar{z}]$ is narrow, then $U_j$ remains approximately normal. Therefore, in computing $[\beta_j \gamma_j \eta_j]$ from (3.26-3.27) (now with $\mu_j$ and $\sigma_j$ replaced by $\tilde{\mu}_j$ and $\tilde{\sigma}_j$), we may choose $A_j$ and $w_j$ in the same manner as when $U_j$ is Gaussian. In conclusion, the procedure for approximating the expectation (3.38) can be summarized as follows.

1. For $j = 1, \ldots, M$, compute $[\beta_j \gamma_j \eta_j]$ from (3.26-3.27), with $\mu_j$ and $\sigma_j$ replaced by $\tilde{\mu}_j$ and $\tilde{\sigma}_j$ given in (3.39-3.40).
2. Compute $b$, $g$, and $H$ from $[\beta_j \gamma_j \eta_j]$ using (3.18-3.20).
3. Approximate the expectation (3.38) by

$$E\left[ e^{\tilde{S}(Z)} 1\{ z < Z_1 \leq \bar{z} \} \right] = E\left[ e^b + g^T Z + \tilde{Z}^T H Z \right] 1\{ z < Z_1 \leq \bar{z} \}$$

(3.41)
To evaluate this last expectation, we invoke the following result, the proof of which is omitted here.

**Proposition 3.3** The expectation (3.41) simplifies to

$$
\frac{e^{b} + g^{T}(I-2H)^{-1}g/2}{\sqrt{\det(I-2H)}} \left[ \Phi \left( \frac{z - \nu}{\varsigma} \right) - \Phi \left( \frac{z - \nu}{\varsigma} \right) \right]
$$

where $\nu$ is the first element of $(I-2H)^{-1}g$, and $\varsigma^2$ is the first element of $(I-2H)^{-1}$. (The first element of a matrix is defined as the one at the top left corner.) For a complex number $x$, $\Phi(x)$ is given by

$$
\Phi(x) := \frac{1}{2} + \frac{1}{\sqrt{2\pi}} \int_{0}^{x} e^{-t^2/2} dt = \frac{1}{2} + \frac{1}{\sqrt{2\pi}} \sum_{k=0}^{\infty} \frac{x^{2k+1}}{k!(2k+1)}
$$

(3.42) Because the series (3.42) converges rapidly, the approximant (3.41) of the expectation (3.38) can be computed quickly.

Now, to approximate $\phi(s)$, we use the fact that

$$
\phi(s) = \mathbb{E}\left[ e^{S(Z)} \right] = \sum_{\ell=1}^{m} \mathbb{E}\left[ e^{S(Z)} 1\{ u_{\ell-1} < Z_1 \leq u_{\ell} \} \right]
$$

(3.43) where $-\infty = u_0 < u_1 < \ldots < u_m = \infty$. To obtain an approximation for $\phi(s)$, we simply use the above procedure to approximate each summand. That is to say, we partition the market factor $Z_1$ into $m$ segments (hence the name of this section), and then approximate each segment separately.

**Example 3.4** Consider the CDO structure from the previous example. The purpose of this example is to compare the method in this section with that in Section 3.3.1 ("Conditioning on the Market Factor"). Equivalent to comparing the methods' speed, this example compares the accuracy that the two methods produce within the same amount of computer time. To maintain equal computing time for both methods, we first recall that, for the method in Section 3.3.1, the 3-step procedure in Section 3.2 needs to be executed several times, depending on how many values of $z$ are used in evaluating the integral (3.37). Similarly, in approximating $\phi(s)$ via the decomposition (3.43) with $m$ being the number of segments, we require $m$ repeated calls to the modified 3-step procedure (see the previous page), which consumes the same amount of time per
call as does the procedure in Section 3.2 (assuming that the time it takes to calculate $\tilde{\mu}_j$, $\tilde{\sigma}_j$, and $\Phi(\cdot)$ is negligible). Therefore, for fair comparison, the number of segments $m$ should equal the number of different values of $z$ used to evaluate the integral (3.37).

In this example, we use $m = 5$, with the partitions $u_k$ in (4.43) chosen to be

$$u_k = -\infty, -1.683, -0.507, 0.507, 1.683, \infty$$

To give the method in Section 3.3.1 equal amount of time to compute its output, we will use only five values of $z$ in evaluating the integral (3.37). The five values of $z$ are chosen to be $z = 0, \pm 1.026, \pm 2.476$.

Using the idea of segmenting the market factor, we obtain an approximation of $\phi(s)$ via the decomposition (3.43), where each summand is approximated using Step 1-3 on the previous page. The approximation of $\phi(s)$ is then used in the inversion formula (3.9) to obtain an approximation for the tranche price $E(L-y)^+$. The result, as shown in the left plot below, is quite accurate.

![Graph](image)

The right plot shows the approximation of $E(L-y)^+$ obtained through the same inversion formula (3.9), but with $\phi(s)$ substituted by its approximation obtained from the method in Section 3.3.1. Because we only use five values of $z$ in evaluating the integral (3.37), the resulting approximation for $\phi(s)$ is not very accurate. As a consequence, the approximation on the right plot breaks down when $y > 15$. Contrast this with the previous example, where we use 31 values of $z$ to evaluate the integral (3.37). As seen
from Table 3.3 in the previous example, the approximation for the tranche price stays accurate even for $y > 15$.

Let us conclude this example by showing the implied correlation curve (Figure 3.6). The solid line represents the true implied correlation, while the dotted line represents the correlation induced by the 5-segment approximation. Evidently, since the approximated price is only accurate up to $y = 30\%$, the dotted line breaks away from the implied correlation curve when $y > 30\%$.

3.4 Application in Control Variate Simulation

The approximation methods described so far provide a way of evaluating the expectation $\phi(s) = \mathbb{E}[e^{sZ}]$ without using Monte Carlo simulation. However, Monte Carlo simulation does have its appeal: unlike approximation errors, the variance of Monte Carlo estimates can be made arbitrarily small by increasing the number of replications. Therefore, one can make a case for the use of simulation in computing the Laplace transform. If one so wishes, the approximants from the previous sections can be used in conjunction with Monte Carlo simulation as a mean of variance reduction. We now describe the use of the approximant from Section 3.2.1 as the control variate. (By ex-
tension, the approximants from Section 3.3 can also be used as control variates in a similar fashion.)

Our objective is to compute \( \phi(s) = \mathbb{E}[e^{S(Z)}] \) using Monte Carlo simulation. The procedure is simple: we sample the factors \( Z_1, \ldots, Z_d \) from the standard Gaussian distribution, compute \( e^{S(Z)} \) from (3.15), and then average over a number of replications to get an estimate for \( \phi(s) \). However, because the variance of this estimate can be large, one needs a large number of replications in order to shrink the confidence interval. As a remedy, we suggest using the approximant from Section 3.2.1 as the control variate, so as to reduce the variance of the Monte Carlo estimate. Consider \( S(Z) \) in (3.17). Since \( S(Z) \approx \tilde{S}(Z) \), we propose a Monte Carlo estimate of the form:

\[
e^{S(Z)} - \theta \left( e^{\tilde{S}(Z)} - \tilde{\phi}(s) \right)
\]

where \( \theta \) is a constant. Firstly, recall that \( \tilde{\phi}(s) \) is known explicitly. Secondly, note that, since \( \mathbb{E}[e^{\tilde{S}(Z)}] = \tilde{\phi}(s) \), the above expression is an unbiased estimate of \( \phi(s) \). It can be shown that if \( \theta = \text{Cov}(e^{S(Z)}, e^{\tilde{S}(Z)}) / \text{Var}(e^{\tilde{S}(Z)}) \), then the above estimate achieves a variance reduction ratio of \( 1 - \rho^2 \), where \( \rho \) is the correlation coefficient between \( e^{S(Z)} \) and \( e^{\tilde{S}(Z)} \). Since \( e^{\tilde{S}(Z)} \) proves to be quite an accurate approximation of \( e^{S(Z)} \), their correlation coefficient \( \rho \) can be close to one. Therefore, we can expect significant variance reduction.

**Example 3.5** Consider the portfolio from Example 3.1. Suppose that, instead of using \( \tilde{\phi}(s) = \mathbb{E}[e^{\tilde{S}(Z)}] \) as the approximation for \( \phi(s) \), we estimate \( \phi(s) \) using Monte Carlo simulation with 900 replications and with \( e^{\tilde{S}(Z)} \) as the control variate, as described above. We then substitute \( \phi(s) \) in the inversion formula by its Monte-Carlo estimate, and obtain the approximation for the trache price \( \mathbb{E}(L - y)^+ \). Figure 3.7 shows the resulting approximation. Note the slight improvement over Figure 3.3 around the tail area.

\[\square\]

### 3.5 Concluding Remarks

We propose an analytical method for approximating the Laplace transform of the credit loss in the Multifactor Gaussian copula model. This method, which requires small computing time and yields satisfyingly accurate approximation, provides an attractive
alternative to numerical integration or Monte Carlo simulation, especially in models with high number of risk factors. The approximation method, used in conjunction with the Laplace inversion formulae, offers a fast and accurate way of pricing CDO tranches and other basket credit derivatives.

Figure 3.7: Result of using $e^{S(z)}$ as the control variate.
Appendix A

Proof of Theorem 2.1

We use the following notation throughout the proof. For a positive integer $k$, let $I_k := \{1, \ldots, k\}$ and let $I_k^n := \{(j_1, \ldots, j_n) : j_1, \ldots, j_n \in I_k\}$.

The backbone of Theorem 2.1 is Kibble’s (1945) generalized tetrachoric series, as shown in (2.9–2.10). We begin our proof by seeking a compact representation for the summation (2.10).

Lemma A.1 Let $C^{(1)}$, $\ldots$, $C^{(n)}$ be $M$-by-$M$ covariance matrices, each having a single-factor structure. That is, for all $r = 1, \ldots, n$,

$$
C^{(r)} = \begin{bmatrix} \rho^{(r)}_{k\ell} \end{bmatrix}_{M \times M}, \quad \text{where } \rho^{(r)}_{k\ell} = a^{(r)}_k a^{(r)}_\ell \quad (k \neq \ell).
$$

Let $\xi^{(i)}_i$ $(i = 1, \ldots, M$ and $r = 0, \ldots, n)$ be any real constants. Consider

$$
\sum_{1 \leq k_1 < k_2 \leq M} \cdots \sum_{1 \leq h_1 < h_2 \leq n} \rho^{(1)}_{k_1\ell_1} \cdots \rho^{(n)}_{h_n\ell_n} \xi^{(1)}_{h_1} \cdots \xi^{(M)}_{h_M}, \quad (A.1)
$$

where, in each term in the summation, $h_i$ denotes the number of times “$i$” occurs among the subscripts $k_1, \ell_1, \ldots, h_n, \ell_n$. Then, we have that the summation (A.1) is equal to

$$
\frac{1}{2^n} \frac{\partial^{2n}}{\partial \theta^2_1 \cdots \partial \theta^2_n} \bigg|_{\theta_1 = \ldots = \theta_n = 0} P_1 \cdots P_M, \quad (A.2)
$$

where $P_i$ $(i = 1, \ldots, M)$ is a polynomial in $\theta_1, \ldots, \theta_n$ defined as

$$
P_i := \sum_{T \subseteq I_n} \xi^{(i)}_{[T]} \prod_{r \in T} \theta_r a^{(r)}_i. \quad (A.3)
$$

□
Proof Note that $P_1, \ldots, P_M$ consist of $2^n$ terms each, all of whom distinct. Let $P$ be the result of expanding the product $P_1 P_2 \ldots P_M$, so that $G$ comprises $2^nM$ distinct terms. The operation in (A.2) kills all terms of $P$ except any term of the type $\theta_1^2 \theta_2^2 \ldots \theta_n^2 \times constant$, from which the operation extracts the constant. Therefore, in order to prove that the expression in (A.2) equals the summation (A.1), it suffices to show that A: "if a term in $P$ is of the form $\theta_1^2 \ldots \theta_n^2 \times constant$, then the constant matches a term in the summation (A.1)"; and B: "each term in the summation (A.1), once multiplied by $\theta_1^2 \ldots \theta_n^2$, equals a term in $P$".

Let us first paint the picture of $P$. Suppose that $A$ is one of the $2^nM$ terms in $P$. Since $P$ is the product of $P_1, \ldots, P_M$, it follows that $A = A_1 \ldots A_M$, where $A_i$ is one of the $2^n$ terms of $P_i$. Evidently from (A.3), $A_i$ is characterized by the following conditions: (I) $A_i$ contains a factor $\xi_q^{(i)}$, where $q$ equals the number of accompanying $a_i$'s; (II) $a_i^{(r)}$ always appears alongside $\theta_r$ in $A_i$, and vice versa; and (III) $A_i$ contains no $\theta_m^n$ with $m > 1$. We now split our proof in two parts: verifying Statement A and verifying Statement B.

Verify Statement A Suppose that $A = A_1 \ldots A_M$ contains exactly two $\theta_1$’s. Because of (III), the two $\theta_1$’s must come from two different $A_i$’s — say, $A_{k_1}$ and $A_{\ell_1}$ (assume $k_1 < \ell_1$ without loss of generality). It follows from (II) that $A$ must contain $a_{k_1}^{(1)}$ and $a_{\ell_1}^{(1)}$, and no other $a_i^{(1)}$’s. By repeating this argument, we conclude that if $A$ contains exactly two $\theta_1$’s, exactly two $\theta_2$’s, and so on, then for some $k_1 < \ell_1$, $k_2 < \ell_2$, ..., 

$$A = \theta_1^2 \ldots \theta_n^2 \left( a_{k_1}^{(1)} a_{\ell_1}^{(1)} \ldots a_{k_n}^{(n)} a_{\ell_n}^{(n)} \right) \xi_{q_1}^{(1)} \ldots \xi_{q_M}^{(M)},$$

where the factors $\xi_q^{(i)}$ appear as a consequence of (I). Obviously, the expression in the paranthesis simplifies to $\rho_{k_1\ell_1}^{(1)} \ldots \rho_{k_n\ell_n}^{(n)}$, where the number of times "$i$" occurs among $k_1, \ell_1, \ldots, k_n, \ell_n$ is the count of $a_i^{(1)}$’s which, according to (I), equals to $q_i$. That is, $h_i = q_i$. It follows the expression that comes behind $\theta_1^2 \ldots \theta_n^2$ in $A$ is a term in the summation (A.1). Thus, the proof of this part is complete.

Verify Statement B Pick a term in the summation (A.1) and multiply it by $\theta_1^2 \ldots \theta_n^2$, we obtain

$$\frac{\theta_1 a_{k_1}^{(1)}}{Cluster 1} \frac{\theta_2 a_{k_2}^{(1)}}{Cluster 2} \ldots \frac{\theta_n a_{k_n}^{(n)}}{Cluster 2n-1} \frac{\theta_n a_{\ell_n}^{(n)}}{Cluster 2n} \xi_{h_1}^{(1)} \ldots \xi_{h_M}^{(M)} := A$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
The above product can be regrouped as $A = A_1 \ldots A_M$, where $A_i$ collects $\xi_{h_i}^{(i)}$ and every cluster in which the subscript of $a$ is $i$. One can easily see that $A_i$ satisfies (I), (II), and (III). Therefore, $A$ is a term in $G$. Thus, we have verified Statement B, and the proof of this lemma is completed. □

**Proposition A.1** Consider the coefficient $b_n$ given in (2.10). Suppose that $\rho_{tk}$ have the generalized d-factor structure (2.3). Let $D^n$ be the set defined in (2.4). For $J \in D^n$, let $\hat{p}_i^{(J)}$ be the perturbed probabilities of $p_i = P(X_i > x_i)$ described in Definition 2.1, with perturbation parameters $\theta_1, \ldots, \theta_n$. Then, as an alternative to (2.10), $b_n$ can be expressed as

$$b_n = \frac{1}{2^n} \sum_{J \in D^n} \omega_J \left( \frac{\partial^{2n}}{\partial \theta_1^2 \cdots \partial \theta_n^2} \right) p_1^{(J)} \ldots p_M^{(J)},$$

where $\omega_J$ is as defined right above Theorem 2.1 in Section 2.2. □

**Proof** Let $v_J(i,k)$ be the coefficient defined through the recursion (2.5). For $J = (j_1, \ldots, j_n) \in D^n$, it can be shown by induction that $v_J(i,k)$ equals the summation of $\prod_{T \in T, r \in T} \theta_r a(i,j_r)$ over all $T \subset \{1, \ldots, n\}$, subject to the condition $|T| = k$. As a consequence, one can rewrite (2.6) as

$$\hat{p}_i^{(J)} := \sum_{T \subset I_n} B_{|T|}(x_i) \prod_{r \in T} \theta_r a(i,j_r).$$

where $B_0(x) = \tilde{\Phi}(x)$ and $B_k(x) = \Phi(x) H_{k-1}(x)$ for $k \geq 1$. (Remember, $H_n(x)$ denotes the Hermite polynomial of degree $n$.) From (2.3), one can write $\rho_{kt}$ as

$$\rho_{kt} = \sum_{j=1}^d \omega_j \rho_{k,j},$$

where $\rho_{k,j} = a_{kj} a_{tj}.$

Using (A.5), we rewrite $\rho_{k_1t_1}, \ldots, \rho_{k_n\ell_n}$ in (2.10) in the following manner. In rewriting $\rho_{k_1t_1}$ using (A.5), we use $j_1$ as the running index of the summation (A.5). Similarly, we use $j_2$ as the running index in rewriting $\rho_{k_2t_2}$ using (A.5), and so on. As a result, (2.10) becomes

$$\sum_{j_1, \ldots, j_n \in I_d} \omega_{j_1} \ldots \omega_{j_n} \sum_{1 \leq k_1 < \ell_1 \leq M \atop 1 \leq k_n < \ell_n \leq M} \rho_{k_1\ell_1}^{(j_1)} \ldots \rho_{k_n\ell_n}^{(j_n)} B_{h_1}(x_1) \ldots B_{h_M}(x_M).$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
To finish the proof, we rewrite the inner summation using Lemma A.1 with $a_i^{(r)} := a(i, j_r)$ and $\xi_r^{(i)} := B_r(x_i)$.

We are now ready to prove Theorem 2.1. Let $g := \tilde{p}_1^{(J)} \cdots \tilde{p}_M^{(J)}$. To stress the fact that $g$ is a function of the index $j_n$ and the perturbation parameter $\theta_n$, let us write $g = g(s_n, j_n)$. Using the definition of the perturbed probabilities, one can show that $g(-\theta_n, j_n) = g(s_n, -j_n)$ and $g(0, j_n) = g(\theta_n, 0)$. From the finite difference formula,

$$
\frac{d^2}{d\theta^2} g(\theta, j) = \lim_{\theta \to 0} \frac{g(\theta, j) + g(-\theta, j) - 2g(0, j)}{\theta^2} = \lim_{\theta \to 0} \frac{g(\theta, j) + g(\theta, -j) - 2g(\theta, 0)}{\theta^2} .
$$

(A.6)

(Here, to simplify notation, we have suppressed the subscript $n$ in $\theta_n$ and $j_n$.) Multiplying (A.6) by $\omega_j$ and then summing over $j \in I_d$, we yield

$$
\frac{1}{2} \sum_{j \in I_d} \omega_j \frac{d^2}{d\theta^2} g(\theta, j) = \lim_{\theta \to 0} \frac{1}{2\theta^2} \sum_{j=1}^d \omega_j \left[ g(\theta, j) + g(\theta, -j) - 2g(\theta, 0) \right]
\frac{1}{2\theta^2} \sum_{j=1}^d \omega_j g(\theta, j)
\frac{1}{2\theta^2} \sum_{j=1}^d \omega_j g(\theta, j) .
$$

The last equality follows from the definition of $\omega_j$ for nonpositive index $j$ (see the paragraph above Theorem 2.1 in Section 2.2). Now, write $g$ as $g(\theta_1, \ldots, \theta_n; J)$, so that the dependence on $\theta_1, \ldots, \theta_n$ and $J$ is made explicit. By repeating the above argument $n$ times, we have

$$
\frac{1}{2^n} \sum_{J \in I_d^n} \omega_J \frac{\partial^{2n} g}{\partial \theta_1^2 \cdots \partial \theta_n^2} \bigg|_{\theta_1 = \cdots = \theta_n = 0} = \lim_{\theta \to 0} \frac{1}{(2\theta^2)^n} \sum_{J \in D^n} \omega_J g(\theta, \ldots, \theta; J)
$$

The left-hand side, according to Proposition A.1, is $b_n$. To prove the statement (2.18), note that $\omega_J/(2s^2)^n = w_J$, and that $g(\theta, \ldots, \theta; J)$ is the product of the perturbed probabilities $\tilde{p}_1^{(J)}, \ldots, \tilde{p}_M^{(J)}$ when the perturbation parameters are $\theta_1 = \ldots = \theta_n = \theta$.

Thus, Theorem 2.1 is established. □
Appendix B

Proof of Theorem 3.1 and Theorem 3.2

Before we begin our proof, let us revisit some notations from the main text. Consider the mapping $G_j(t)$ defined in (3.13); from now on, we will alter its notation to $G_j(t,s)$ to signify that, as noted below its definition, the mapping $G_j$ takes as its argument, in addition to $t$, the Laplace variable $s$. In the same spirit, $S(Z)$ and $\tilde{S}(Z)$ (defined in (3.15) and (3.17)) will henceforth be referred to as $S(A,s,Z)$ and $\tilde{S}(A,s,Z)$, respectively, to signify dependence on the Laplace variable $s$ and the loading matrix $A$. To express $S(A,s,Z)$ explicitly in terms of its three arguments, we combine (3.14) and (3.15) to yield

$$S(A,s,Z) = \sum_{j=1}^{M} G_j \left( \frac{x_j - \frac{a_j^T Z}{\sqrt{1-a_j^T a_j}}, s} \right) \quad (B.1)$$

As for $\tilde{S}(A,s,Z)$, we rewrite the summand in (3.17) as the matrix product of $[1 \ U_j \ U_j^2]$ and the coefficients $[\beta_j \ \gamma_j \ \eta_j]^T$, for which we then substitute the formula (3.27), with $\mu_j$ and $\sigma_j$ replaced by their definitions (3.25). This yields

$$\tilde{S}(A,s,Z) = \sum_{j=1}^{M} \left[ 1 - \frac{a_j^T Z}{\sqrt{a_j^2 a_j}} \right] B G_j \left( \frac{x_j 1 + \Lambda \frac{a_j^T Z}{\sqrt{1-a_j^2 a_j}}, s} \right) \quad (B.2)$$

where $B = B_1 = \ldots = B_M$ and $1$ denote the column vector of the same length as $\Lambda$ whose elements are all one. The 1-by-3 matrix in the summand is the result of multiplying $[1 \ U_j \ U_j^2]$ by the 3-by-3 matrix displayed in equation (3.27), with $U_j$
substituted by its definition (3.14).

Proposition B.1

(i) There exist a real number \( \varepsilon_0 > 0 \) and an integrable function \( g(Z) \) such that for all nonnegative integer \( n \leq 8 \) and for all \( M \)-row matrices \( A \) with \( \|A\| \leq 1 \),

\[
\sup_{0 < \varepsilon < \varepsilon_0, s \in C^+} \left| \frac{\partial^n}{\partial \varepsilon^n} e^{S(\varepsilon A, s, Z)} \right| < g(Z) \tag{B.3}
\]

(ii) There exist a real number \( \varepsilon^* > 0 \) and an integrable function \( \tilde{g}(Z) \) such that for all nonnegative integer \( n \leq 8 \) and for all \( M \)-row matrices \( A \) with \( \|A\| \leq 1 \),

\[
\sup_{0 < \varepsilon < \varepsilon^*, s \in C^+} \left| \frac{\partial^n}{\partial \varepsilon^n} e^{\tilde{S}(\varepsilon A, s, Z)} \right| < \tilde{g}(Z) \tag{B.4}
\]

Proof The following notations are useful not only in proving this proposition but also in proving the subsequent lemmas.

\[
\mu_j = \frac{x_j}{\sqrt{1 - a_j^2}}, \quad \sigma_j = \frac{\sqrt{a_j^2 a_j}}{\sqrt{1 - a_j^2}}, \quad \text{and} \quad Q_j = \frac{x_j}{\sqrt{a_j^2 a_j}} \tag{B.5}
\]

Then, from (B.1) and (B.2),

\[
S(\varepsilon A, s, Z) = \sum_{j=1}^{M} G_j \left( \mu_j(\varepsilon) + Q_j \sigma_j(\varepsilon), s \right) \tag{B.6}
\]

\[
\tilde{S}(\varepsilon A, s, Z) = \sum_{j=1}^{M} \left[ 1 - Q_j Q_j^2 \right] B G_j \left( \mu_j(\varepsilon), 1 + \Lambda \sigma_j(\varepsilon), s \right) \tag{B.7}
\]

Proof of (i) Because \( X_j \)'s are assumed nonnegative, \( |\psi_j(s)| \leq 1 \) for all \( s \in C^+ \). Let \( \mu_j^{(n)}(\varepsilon) = (\partial^n/\partial \varepsilon^n) \mu_j(\varepsilon) \) and \( \sigma_j^{(n)}(\varepsilon) = (\partial^n/\partial \varepsilon^n) \sigma_j(\varepsilon) \). It is easy to verify that

\[
\sup_{0 < \varepsilon < 0.9} |\mu_j^{(n)}(\varepsilon)| < \infty \quad \text{and} \quad \sup_{0 < \varepsilon < 0.9} |\sigma_j^{(n)}(\varepsilon)| < \infty \tag{B.8}
\]

Let \( \Phi^{(n)}(t) := (\partial^n/\partial t^n) \Phi(t) \). It is known that \( \Phi^{(n)}(t) \) is bounded for all \( t \in R \) and \( n \). In fact, \( |\Phi^{(n)}(t)| < 15 \) for all \( t \in R \) and \( n \leq 8 \). From (B.6) and (3.13),

\[
e^{S(\varepsilon A, s, Z)} = \prod_{j=1}^{M} \left( 1 + (\psi_j(s) - 1) \bar{\Phi}(\mu_j(e) + Q_j \sigma_j(e)) \right)
\]

It follows (Chain Rule) that \( (\partial^n/\partial s^n)e^{S(\varepsilon A, s, Z)} \) equals a polynomial in \( \psi_j(s), Q_j, \mu_j^{(k)}(e), \sigma_j^{(k)}(e), \Phi^{(k)}(\mu_j(\varepsilon) + Q_j \sigma_j(\varepsilon)) \) (\( 1 < k \leq n \)). All of these, except \( Q_j \), are bounded by a constant for all \( \varepsilon < 0.9 \) and \( s \in C^+ \). Hence, \( (\partial^n/\partial s^n)e^{S(\varepsilon A, s, Z)} \) is bounded by a polynomial in \( Q_j \)'s, which is integrable. So the proof of (i) is completed.
Proof of (ii) Fix $\delta^* \in (0, 1)$. Let $x_{\min} := \min\{x_1, \ldots, x_M\} > 0$ and let $t^* = x_{\min} \delta^*$. We will first prove that there exists a real constant $C$ such that
\[
\left| \frac{\partial^n}{\partial t^n} G_j(t, s) \right| < C \quad \text{for all } t > t^*, \; s \in C^+, \; n \leq 8, \; j = 1, \ldots, M. \tag{B.9}
\]
Define $g_j(p, s) := \log(1 - p + p \Psi_j(s))$. Using the fact that $|\Psi_j(s)| \leq 1$ and that $|\log x| \leq \pi + |\log|x|| for all complex number $x$, one can show that $|g_j(p, s)| \leq \pi + |\log|1 - 2p||$. It follows that $G_j(t, s) = g_j(\Phi(t), s)$ is bounded for all $t > t^*$ and for all $s \in C^+$. Thus, in showing (B.9) we may assume that $n \neq 0$. Using the Chain Rule, one can prove (B.9) by showing that $(\partial^n/\partial p^n)g_j(p, s)$ is bounded for all $p < \Phi(t^*), \; s \in C^+, \; \text{and } n \leq 8$. Indeed,
\[
\left| \frac{\partial^n}{\partial p^n} g_j(p, s) \right| \leq \frac{(n-1)!2^n}{(1-p+p\Psi_j(s))^n} < \frac{(n-1)!2^n}{|1-2p|^n}, \; \forall p < \Phi(t^*), \; s \in C^+.
\]
Thus, (B.9) is established. Where $C$ is the constant in (B.9), fix $\varepsilon^* \in (0, 1)$ such that
\[
\left| x_{\min} - \lambda_{\max} \frac{\varepsilon^*}{1-\varepsilon^*} \right| > t^* \quad \text{and} \quad \left( \frac{\varepsilon^*}{1-\varepsilon^*} \right)^2 < \frac{\delta^*}{MC\|A\|^2\|B\|} \tag{B.10}
\]
Using the fact that $\mu_j(\varepsilon) > x_j$ and $\sigma_j(\varepsilon) < \varepsilon/(1-\varepsilon)$, we can deduce from the first part of (B.10) that $\mu_j(\varepsilon) > t^*$ and $\mu_j(\varepsilon) + \Lambda \sigma_j(\varepsilon) > t^*$ for all positive number $\varepsilon < \varepsilon^*$. And since (B.9) implies that $G_j(t, s)$ is twice continuously differentiable at $t > t^*$, the Mean Value Theorem ensures that there exists a diagonal matrix $\Sigma_j$ whose elements on the main diagonal are $G''(1, s), \ldots, G''(t^*, s)$, where $\zeta_1, \ldots, \zeta_k > t^*$, that satisfies
\[
G_j(\mu_j(\varepsilon)\mathbf{1} + \Lambda \sigma_j(\varepsilon), s) = G_j(\mu_j(\varepsilon), s)\mathbf{1} + \sigma_j(\varepsilon)G_j''(\mu_j(\varepsilon), s)\Lambda + \frac{1}{2} \sigma_j^2(\varepsilon)\Sigma_j \Lambda^2 \tag{B.11}
\]
One can view this as a second-order expansion in $\Lambda$. Note for later use that $\|\Sigma_j\| < C$, where $C$ is the constant from (B.9). If we front-multiply both sides by $[0 \; 0 \; 1]$, the first two terms on the right vanishes because $B\mathbf{1} = [1 \; 0 \; 0]^T$ and $B\Lambda = [0 \; 1 \; 0]^T$ (to see this, note from the definition of $B$ that $B[1 \; \Lambda \; \Lambda^2]$ equals the identity matrix). As a result, from (B.7),
\[
|\tilde{S}(\varepsilon\mathbf{A}, s, Z)| \leq \sum_{j=1}^M \left( [1 \; Q_j \; 0] + Q_j^2[0 \; 0 \; 1] \right) B G_j(\mu_j(\varepsilon)\mathbf{1} + \Lambda \sigma_j(\varepsilon), s) \left| + \sum_{j=1}^M \frac{Q_j^2}{2} \Sigma_j^2(\varepsilon) \right| B \|\Xi_j\| \Lambda^2 \tag{B.12}
\]
for all $\varepsilon < \varepsilon^*$ and $s \in C^+$. Here, we have used the fact that $|Q_j| \leq \sqrt{\Sigma_j^2}$ (see (B.5)). Combining (B.12) with the second part of (B.10), we have
\[
|e^{i\tilde{S}(\varepsilon\mathbf{A}, s, Z)}| \leq e^{\frac{1}{2}|\tilde{S}(\varepsilon\mathbf{A}, s, Z)|} < \exp\left(MC\|B\|\left(1 + \sqrt{\Sigma_j^2} + \frac{\varepsilon^*}{2}\right)^2\right) \tag{B.13}
\]
for all $\varepsilon < \varepsilon^*$ and $s \in C^+$. The last step of the proof is to differentiate (B.7) with respect to $\varepsilon$, and then take $\infty$-norm:

$$
\left| \frac{\partial^n}{\partial \varepsilon^n} \hat{S}(\varepsilon A, s, Z) \right| \leq \sum_{j=1}^{M} (1 + |Q_j| + Q_j^2) \|B\| \left\| \frac{\partial^n}{\partial \varepsilon^n} G_j(\mu_j(\varepsilon)1 + A\sigma_j(\varepsilon), s) \right\|
$$

$$
< 2M \left( 1 + Z^T Z \right) \|B\|D, \forall \varepsilon < \varepsilon^*, s \in C^+, n \leq 8
$$

(B.14)

where $D$ is some constant resulting from combining (B.9) and (B.8). According to the Chain Rule, $(\partial^n/\partial \varepsilon^n)e^{\hat{S}(\varepsilon,A,s,Z)}$ equals $e^{\hat{S}(\varepsilon,A,s,Z)}$ times a polynomial in $(\partial^k/\partial \varepsilon^k)\hat{S}(\varepsilon,A,s,Z)$ ($1 \leq k \leq n$). By combining (B.14) and (B.13), we conclude that $(\partial^n/\partial \varepsilon^n)e^{\hat{S}(\varepsilon,A,s,Z)}$ is bounded by an integrable function for all $\varepsilon < \varepsilon^*$, $s \in C^+$, and $n \leq 8$. □

Before we move on to the next lemma, let us define a matrix of constants:

$$
\begin{bmatrix}
c_{30} & c_{40} & c_{50} & c_{60} \\
c_{31} & c_{41} & c_{51} & c_{61} \\
c_{32} & c_{42} & c_{52} & c_{62}
\end{bmatrix}
=: 
\begin{bmatrix}
1 & \lambda & \lambda^2 & \lambda^3 \lambda^4 & \lambda^5 & \lambda^6 \\
\lambda & \lambda^2 & \lambda^3 & \lambda^4 & \lambda^5 & \lambda^6 \\
\lambda^2 & \lambda^3 & \lambda^4 & \lambda^5 & \lambda^6 & \lambda^7 \\
\lambda^3 & \lambda^4 & \lambda^5 & \lambda^6 & \lambda^7 & \lambda^8
\end{bmatrix}^{-1}
$$

(B.15)

These constants are useful in stating the next lemma. To understand the meaning of these constants, we note that the matrix defined above is the unique solution to the following optimization problem:

$$
\min_{c_{ij}} \sum_{k=1}^{K} w_k f(\lambda_k)^T f(\lambda_k)
, \quad f(x) :=
\begin{bmatrix}
x^3 \\
x^4 \\
x^5 \\
x^6
\end{bmatrix}
- 
\begin{bmatrix}
c_{30} & c_{31} & c_{32} \\
c_{40} & c_{41} & c_{42} \\
c_{50} & c_{51} & c_{52} \\
c_{60} & c_{61} & c_{62}
\end{bmatrix}
\begin{bmatrix}
1 \\
x \\
x^2
\end{bmatrix}
$$

According to this minimization problem, the scalars $c_{40}$, $c_{41}$, and $c_{42}$, for example, have the interpretation of being the coefficients such that $x^4 \approx c_{40} + c_{41}x + c_{42}x^2$. Furthermore, it can be shown that if $\lambda_k$ and $w_k$ are arranged symmetrically around zero (so that $\lambda^n = 0$ whenever $n$ is odd), then the solution to the above minimization problem satisfies $c_{30} = c_{32} = c_{41} = c_{50} = c_{52} = c_{61} = 0$; this means, for example, that $c_{50}x$ is the optimal approximation of $x^5$ (note that both are odd functions), and that $c_{60} + c_{62}x^2$ is the optimal approximation of $x^6$ (both are even functions).

Now that we know how to interpret the constants defined in (B.15), we are ready to state the next lemma.
Lemma B.1 Let \( A \) be an \( M \times d \) matrix such that \( \| A \| \leq 1 \). Let \( U := \sqrt{\text{diag}(AA^T)} \).

Let \( D_n \) be given by (3.31). Let \( X \) be the diagonal matrix whose \( j \)th element on the main diagonal is \( x_j \). For an integer \( n \), let \( T_n \) and \( R_n \) be \( 1 \times M \) row vectors defined by
\[
T_n := D_n^T U^n \quad \quad (B.16)
\]
\[
R_n := \binom{n}{2} T_{n-1} U X + 3 \binom{n}{3} T_{n-2} U^2 \quad \quad (B.17)
\]

Let \( V_n := (\partial^n/\partial \varepsilon^n)|_{\varepsilon=0} \tilde{S}(\varepsilon A, s, Z) \) and let \( \tilde{V}_n := (\partial^n/\partial \varepsilon^n)|_{\varepsilon=0} \tilde{S}(\varepsilon A, s, Z) \). Define a random vector \( \Omega := U^{-1} A Z \), and note that \( \Omega \) is an \( M \times 1 \) vector of correlated \( \mathcal{N}(0,1) \) random variables. Then,
\[
V_1 = T_1 \Omega \quad \quad (B.18)
\]
\[
V_2 = T_2 \Omega^2 + R_2 1 \quad \quad (B.19)
\]
\[
V_3 = T_3 \Omega^3 + R_3 \Omega \quad \quad (B.20)
\]
\[
V_3 - \tilde{V}_3 = T_3 \left( \Omega^3 - c_{31} \Omega \right) \quad \quad (B.21)
\]
\[
V_4 - \tilde{V}_4 = T_4 \left( \Omega^4 - c_{40} - c_{42} \Omega^2 \right) \quad \quad (B.22)
\]
\[
V_5 - \tilde{V}_5 = T_5 \left( \Omega^5 - c_{50} \Omega \right) + R_5 \left( \Omega^3 - c_{31} \Omega \right) \quad \quad (B.23)
\]
\[
V_6 - \tilde{V}_6 = T_6 \left( \Omega^6 - c_{60} - c_{62} \Omega^2 \right) + R_6 \left( \Omega^4 - c_{40} - c_{42} \Omega^2 \right) \quad \quad (B.24)
\]

where \( 1 \) denotes the vector of length \( M \) whose elements are all one. (Technically, the terms \( c_{40} \) and \( c_{60} \) in (B.22) and (B.24) should be written as \( c_{40} 1 \) and \( c_{60} 1 \), respectively, so that their dimensions agree with the surrounding matrices. But, upon this understanding, we omit the \( 1 \) to simplify notation.)

Proof Here, we will suppress the argument "\( s \)" from \( G_j(\cdot, s) \) to simplify notation. Let \( \mu_j(\varepsilon) \) and \( \sigma_j(\varepsilon) \) be as given in (B.5). Define \( f(t) := \sqrt{1-t} \). We will first show that for all \( \lambda \in \mathbb{R} \),
\[
\frac{\partial^k}{\partial v^k} \bigg|_{v=0} G_j(\mu_j(\varepsilon)+\lambda \sigma_j(\varepsilon)) = (a_j^T a_j)^{\frac{k}{2}} \lambda^k \sum_{t=0}^{k} \alpha_{jkt} \lambda^t = \sum_{t=0}^{k} \alpha_{jkt} \lambda^t \quad \quad (B.25)
\]

where \( \alpha_{jkt} \)'s are some constants. The first equality results from the change of variable \( v = \varepsilon \sqrt{a_j^T a_j} \) so that \( \mu_j(\varepsilon) = x_j/f(v^2) \) and \( \sigma_j(\varepsilon) = v/f(v^2) \). The second equality follows from the fact that, for any mappings \( g(v) \) and \( h(v) \) that are \( k \)-time differentiable at \( v = 0 \) with \( h(0) = 0 \) and for any mapping \( G(t) \) that is \( k \)-time differentiable at \( t = g(0) \), we have that
\[(\partial^k/\partial v^k)_{v=0}G(\varphi(v)+\lambda h(v))\] equals a polynomial in \(\lambda\) of degree \(k\). The coefficients \(\alpha_{jkl}\)'s of the polynomial can be obtained by differentiating the left-hand side of the second equality:

\[
\alpha_{jkl} = \frac{(a_j^T a_j)^{\frac{k}{2}}}{\ell!} \frac{\partial^{k-\ell}}{\partial v^{k-\ell}} \left|_{v=0} \right. G_j \left( \left. \frac{x_j + \lambda v}{f(v^2)} \right) \right.
\]

To prove the last equality in (B.25), we will show that \(\alpha_{jkl} = 0\) whenever \(k - \ell\) is odd. Using the identity \((\partial^k/\partial v^k)_{v=0}(v^j g(v)) = \ell! \left( \frac{k}{\ell} \right) g^{(k-\ell)}(0)\), the above becomes

\[
\alpha_{jkl} = \frac{k! (a_j^T a_j)^{\frac{k}{2}}}{\ell! (k-\ell)!} \frac{\partial^{k-\ell}}{\partial v^{k-\ell}} \left|_{v=0} \right. G_j \left( \left. \frac{x_j/f(v^2)}{[f(v^2)]^{\ell}} \right) \right.
\]

Because, for any mapping \(g(x)\) that is \(n\)-time differentiable at \(x = 0\), \((\partial^n/\partial v^n)_{v=0} g(v^2) = 0\) whenever \(n\) is odd, we conclude that \(\alpha_{jkl}\) above is zero whenever \(k - \ell\) is odd. Thus, (B.25) is established in its entirety. Now, let \(\Gamma_j\) and \(L\) be matrices defined below (in the definition of \(L\) below, \(1\) denotes the vector of the same length as \(A\) whose elements are all one). In addition, where \(v\) is a column vector (of any length) whose elements are functions of \(\varepsilon\), let \(D\) be the operator defined beneath.

\[
\Gamma_j = \begin{bmatrix}
\alpha_{j00} & 0 & 0 & 0 & 0 & 0 \\
0 & \alpha_{j11} & 0 & 0 & 0 & 0 \\
0 & \alpha_{j20} & \alpha_{j22} & 0 & 0 & 0 \\
0 & \alpha_{j31} & 0 & \alpha_{j33} & 0 & 0 \\
0 & \alpha_{j40} & \alpha_{j42} & 0 & \alpha_{j44} & 0 \\
0 & \alpha_{j51} & 0 & \alpha_{j53} & 0 & \alpha_{j55} \\
0 & \alpha_{j60} & \alpha_{j62} & 0 & \alpha_{j64} & 0 & \alpha_{j66}
\end{bmatrix}
\]

\[
L := \begin{bmatrix} 1 & A & A^2 & \ldots & A^6 \end{bmatrix}
\]

\[
D \varepsilon := \begin{bmatrix} \varepsilon & \frac{\partial \varepsilon}{\partial \varepsilon} & \frac{\partial^2 \varepsilon}{\partial \varepsilon^2} & \ldots & \frac{\partial^6 \varepsilon}{\partial \varepsilon^6} \end{bmatrix}_{\varepsilon=0}
\]

Using these matrix notations, equation (B.25) can be written as

\[
D G_j(\mu_j(\varepsilon)1 + A \sigma_j(\varepsilon)) = L \Gamma_j^T.
\]

Then, applying the operator \(D\) to (B.7), noting that \(D \hat{S}(\varepsilon A, s, Z) = [\hat{V}_0 \ldots \hat{V}_6]\) (definition of \(\hat{V}_n\)), we have

\[
[\hat{V}_0 \ldots \hat{V}_6] = \sum_{j=1}^{M} [Q_j Q_j^2] \cdot D G_j(\mu_j(\varepsilon)1 + A \sigma_j(\varepsilon)) = \sum_{j=1}^{M} [Q_j Q_j^2] \cdot B L \Gamma_j^T \quad (B.27)
\]

Similarly, from the definition of \(V_n\) and from (B.6),

\[
[V_0 \ldots V_6] = D S(\varepsilon A, s, Z) = \sum_{j=1}^{M} D G_j(\mu_j(\varepsilon)1 + Q_j \sigma_j(\varepsilon)) = \sum_{j=1}^{M} [Q_j Q_j^2 \ldots Q_j^6] \Gamma_j^T
\]

\[(B.28)\]

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
where \( Q_j \) is as defined in (B.5). Equations (B.27–B.28), with \( B \) given by (3.26) and with \( L \) and \( F_j \) defined above, completely characterize \( V_n \) and \( \dot{V}_n \) \((0 \leq n \leq 6)\). We can further simplify the last summation in (B.27) by using the definition (3.26) of \( B_j \) to show that

\[
BL = \begin{bmatrix}
1 & \overline{\lambda} & \overline{\lambda^2} \\
\overline{\lambda} & \overline{\lambda^2} & \overline{\lambda^3} \\
\overline{\lambda^2} & \overline{\lambda^3} & \overline{\lambda^4}
\end{bmatrix}^{-1}
\begin{bmatrix}
1 & \overline{\lambda} & \ldots & \overline{\lambda^6} \\
\overline{\lambda} & \overline{\lambda^2} & \ldots & \overline{\lambda^6} \\
\overline{\lambda^2} & \overline{\lambda^3} & \ldots & \overline{\lambda^6}
\end{bmatrix} = \begin{bmatrix}
1 & 0 & 0 & c_{30} & c_{40} & c_{50} & c_{60} \\
0 & 1 & 0 & c_{31} & c_{41} & c_{51} & c_{61} \\
0 & 0 & 1 & c_{32} & c_{42} & c_{52} & c_{62}
\end{bmatrix}
\] (B.29)

where the second equality follows from (B.15). To finish the proof, consider \( T_n \) and \( R_n \) defined in (B.16–B.17), and consider \( \alpha_{jk} \) given in (B.26). It can be verified that the \( j \)th element of \( T_k \) equals \( \alpha_{jk} \), and the \( j \)th element of \( R_k \) equals \( \alpha_{i,k,k-2} \). Using this observation and noting that \( \Omega = [Q_1 \ldots Q_M]^T \), one can easily show that (B.27–B.29) imply (B.18–B.24).

The following notation is useful in the rest of the proof.

\[
R_n(A, s, Z) := \frac{\partial^n}{\partial \varepsilon^n} \left|_{\varepsilon=0} \right. \frac{e^{S(\varepsilon A, s, Z)} - e^{S(0, s, Z)}}{\phi_0(s)}
\] (B.30)

**Lemma B.2** Let \( A \) be an \( M \times d \) matrix such that \( \|A\| \leq 1 \).

(i) Suppose that the gridpoints \( \lambda_k \) and the associated weights \( w_k \) are arranged symmetrically around zero, so that \( \overline{\lambda^n} = 0 \) whenever \( n \) is odd. Then, \( \mathbb{E}[R_n(A, s, Z)] = 0 \) for \( n = 1, 2, 3, 5 \). And,

\[
\mathbb{E}[R_4(A, s, Z)] = 24r \, D_4^T \text{diag}^2(AA^T) \, 1 + 24v \, D_3^T \text{diag}((AA^T)AA^T) \, D_1
\] (B.31)

where \( 1 \) denotes the column vector of length \( M \) whose elements are all one. The scalars \( r \) and \( v \) are as given in Theorem 1.

(ii) Suppose, in addition, that \( \overline{\lambda^2} = 1, \overline{\lambda^4} = 3, \) and \( \overline{\lambda^6} = 15 \). Then,

\[
\mathbb{E}[R_6(A, s, Z)] = 360 \, D_3^T \text{diag}(AA^TAA^T) \, AA^T \, D_1 + 60 \, D_3^T \text{diag}(AA^T)^3 \, D_3
\]

where \( \Pi \) is as given in Theorem 1.
Proof Using the Chain Rule, \( R_n(A, s, Z) \) can be expressed in terms of \( V_k \) and \( \hat{V}_k \) (1 \( \leq k \leq n \)) given in Lemma 1. Specifically, \( R_1(A, s, Z) = R_2(A, s, Z) = 0 \), and

\[
R_3(A, s, Z) = V_3 - \hat{V}_3 \quad \text{(B.32)}
\]

\[
R_4(A, s, Z) = V_4 - \hat{V}_4 + 4V_1(V_3 - \hat{V}_3) \quad \text{(B.33)}
\]

\[
R_5(A, s, Z) = V_5 - \hat{V}_5 + 5V_1(V_4 - \hat{V}_4) + 10(V_3 - \hat{V}_3)(V_2 - \hat{V}_2) \quad \text{(B.34)}
\]

\[
R_6(A, s, Z) = V_6 - \hat{V}_6 + 6V_1(V_5 - \hat{V}_5) + 15(V_4 - \hat{V}_4)(V_2 - \hat{V}_2) + 10(V_3 - \hat{V}_3)(V_3 + \hat{V}_3 + 6V_1V_2 + 2V_3^2) \quad \text{(B.35)}
\]

Proof of (i) Because \( \Omega \) is a Gaussian vector with mean zero, it follows that for every triplet \((\ell, m, n)\) of integers whose sum is odd, \( E[\Omega^\ell \Omega^m \Omega^n] = 0 \). Using this fact, one can easily see from substituting (B.18-B.23) into (B.32) and (B.34) that \( E[R_3(A, s, Z)] = E[R_5(A, s, Z)] = 0 \). Now, let \( \Sigma \) be the correlation structure of \( \Omega \). Then, \( E[\Omega \Omega^T] = \Sigma \), and \( E[\Omega^3 \Omega^T] = 3 \Sigma \).

It follows from (B.33) that

\[
E[\chi(0, s, Z)] = E[V_4 - \hat{V}_4] + 4E[V_1(V_3 - \hat{V}_3)]
\]

\[
= T_4 E[\Omega^4 - c_{40} - c_{42} \Omega^2] + 4T_3 E[(\Omega^3 - c_{31} \Omega) \Omega^T] T_1^T
\]

\[
= (3c_{40} - c_{42}) T_4 1 + 4(3c_{31}) T_3 \Sigma T_1^T
\]

By writing \( c_{31}, c_{40}, \) and \( c_{42} \) in terms of \( \lambda_n \)'s using (B.15), one can show that \( 3c_{31} = 6v \) and \( 3c_{40} - c_{42} = 24r \). Recall that \( T_n = D_n \tilde{U}^n \), where \( \tilde{U} = \sqrt{\text{diag}(AA^T)} \) (see definition (B.16)).

The proof of (i) is then completed by noting that and \( \Sigma = U^{-1}AA^T U^{-1} \).

Proof of (ii) Two random vectors \( v_1 \) and \( v_2 \) of the same length are said to be orthogonal (symbol \( v_1 \perp v_2 \)) if both \( E[v_1 v_2] \) and \( E[v_1^T v_1] \) are zero matrices. Since \( \Omega \) is a vector of \( \mathcal{N}(0, 1) \) random variables, one can show that \( \Omega \) is orthogonal both to \( \Omega^3 - 3 \) and to \( \Omega^5 - 15 \). Also, \( \Omega^4 - 6 \Omega^2 + 3 \) is orthogonal both to \( \Omega^2 \) and to \( \Omega^5 \), where \( v \) is any constant vector of length \( M \). Now, from (B.15), one can show that if \( \lambda^2 = 1, \lambda^2 = 3, \lambda^2 = 15, \) and \( \lambda^2 = 0 \) for \( n \) odd, then \( c_{31} = 3, c_{40} = -3, c_{42} = 6, c_{51} = 15, \) and \( c_{60} + c_{62} = 15 \). Consequently, it is obvious from (B.24) that \( E[V_6 - \hat{V}_6] = 0 \). And, because of the orthogonality of \( \Omega \), it follows from (B.18-B.23) that \( V_4 \perp V_5 - \hat{V}_3 \) and \( V_2 \perp V_4 \perp V_5 - \hat{V}_5 \). Moreover, by writing \( V_1^2 = T_1 \Omega \Omega^T T_1^T \) and use the aforementioned fact that \( \Omega^3 \Omega^T T_1^T \perp \Omega^4 - 6 \Omega^2 + 3 \), one can see that \( V_1^2 \perp V_4 \perp V_5 - \hat{V}_5 \). Thus, we have shown that the first line on the right side of (B.35) has expected value zero. We now consider the second line. From (B.20-B.21), \( \hat{V}_3 = V_4 - (V_5 - \hat{V}_3) = (3T_3 + R_3) \Omega \). It follows that, since \( \Omega \) is orthogonal to \( \Omega^3 - 3 \), we have \( \hat{V}_3 \perp V_3 - \hat{V}_3 \). Using the fact that \( E[\Omega^3 (\Omega^3 - 3) \Omega^T] = 6 \Sigma^3 \) and that \( E[V_3 - \hat{V}_3] = 0 \), one can see from (B.20-B.21) that \( E[V_5(V_5 - \hat{V}_5)] = 6 T_3 \Sigma^3 T_3^T \). Finally,
from (B.18–B.19), one can show that

$$6V_2 + 2V_3^2 = 6\Omega^T\Xi\Omega + 6 R_2 1$$

where $\Xi$ is the sum of the square matrix $T_1^T T_1 / 3$ and the $M$-by-$M$ diagonal matrix whose $j$th element on the main diagonal is the $j$th element of $T_2$. It follows that, since $V_1 = T_1 \Omega$ and $V_3 - \tilde{V}_3 = (\Omega^3 - 3\Omega)^T T_3$, and $E[V_1(V_3 - \tilde{V}_3)] = 0$,

$$E\left[(6V_1 V_2 + 2V_3^2)(V_3 - \tilde{V}_3)\right] = E\left[V_1(6V_2 + 2V_3^2)(V_3 - \tilde{V}_3)\right] = 6 T_1 E[\Omega^T \Xi \Omega (\Omega^3 - 3\Omega)^T] T_3$$

Note the fact that $E[\Omega^T \Xi \Xi (\Omega^3 - 3\Omega)^T] = 6 \Xi \text{diag}(\Xi \Xi \Xi)$. We conclude that

$$E\left[(V_3 + \tilde{V}_3 + 6V_1 V_2 + 2V_3^2)(V_3 - \tilde{V}_3)\right] = 6 T_3 \Xi^T T_3^T + 36 T_1 \Xi \text{diag}(\Xi \Xi \Xi) T_3 \tag{B.36}$$

To complete the proof of (ii), we note the relation $\Xi = U \Pi U$ and recall that $T_n = D_n^T U^n$, where $U = \sqrt{\text{diag}(AA^T)}$ (see definition (B.16)), and $\Sigma = U^{-1} A A^T U^{-1}$. □

We are now ready to prove Theorem 3.1 and Theorem 3.2 from Section 3.2.3. Where $\varepsilon_0$ and $\varepsilon^*$ are the real numbers in the statement of Proposition B.1, let $\delta = \min\{\varepsilon_0, \varepsilon^*\}$. It can be implied from Proposition B.1 that $E[e^{S(\varepsilon A, s, Z)}]$ is 7-time continuously differentiable in $\varepsilon$ for all $\varepsilon < \delta$, $\|A\| \leq 1$, and $s \in C^\dagger$. Consequently, for every $n \leq 7$ and for every $\varepsilon < \delta$, there exists $\tilde{\varepsilon} \in [0, \varepsilon]$ such that

$$E[e^{S(\varepsilon A, s, Z)}] = \sum_{k=0}^{n-1} \frac{\varepsilon^k}{k!} \frac{\partial^k}{\partial \varepsilon^k} \left|_{\varepsilon=0} E[e^{S(\varepsilon A, s, Z)}] + \frac{\varepsilon^n}{n!} \frac{\partial^n}{\partial \varepsilon^n} \left|_{\varepsilon=\tilde{\varepsilon}} E[e^{S(\varepsilon A, s, Z)}]\right]$$

Note that, because of the regularity of $e^{S(\varepsilon A, s, Z)}$ established in Proposition B.1, one can interchange the derivatives with the expectations in the above equation. Note, also, that if we replace the matrix $A$ above by $\hat{A} = A/\|A\|$ and replace $\varepsilon$ by $\|A\|$, the left-hand side of the above equation simplifies to the Laplace transform $\phi(s)$ (see (3.15)). Following the same line of arguments, we conclude that $\hat{\phi}(s)$ admits a similar expansion.

It follows that, with the notation (B.30), we can write the difference $\phi(s) - \hat{\phi}(s)$ as

$$\frac{\phi(s) - \hat{\phi}(s)}{\phi_0(s)} = \sum_{k=0}^{n} \frac{\|A\|^k}{k!} E[R_k(\hat{A}, s, Z)]$$

$$+ \frac{\|A\|^n}{n! \phi_0(s)} \left( \frac{\partial^n}{\partial \varepsilon^n} E[e^{S(\varepsilon A, s, Z)}] - \frac{\partial^n}{\partial \varepsilon^n} E[e^{S(\varepsilon \hat{A}, s, Z)}] \right) \tag{B.37}$$

where $\tilde{\varepsilon}, \check{\varepsilon} < \|A\| < \delta$ and $n \leq 7$. Note that, according to Proposition B.1, the last parenthesis in (B.37) can be bounded by $E[g(Z)] + E[\hat{g}(Z)]$, where $g(Z)$ and $\hat{g}(Z)$ is the
integrable functions in Proposition B.1. To establish the bound (3.32), we take \( n = 4 \) in (B.37) and note that, according to Part (i) of Lemma B.2, the summation on the first line of (B.37) is zero. To prove the convergence rate (3.33), we take \( n = 6 \) in (B.37) and use, once again, Part (i) of Lemma B.2. To prove the bound (3.34), we take \( n = 6 \) in (B.37) and use Part (i) of Lemma B.2, noting that the constants \( r \) and \( v \) in (B.31) are zeros when \( \lambda^2 = 1 \) and \( \lambda^4 = 3 \). Finally, to prove the convergence rate (3.35), we take \( n = 7 \) in (B.37) and use Part (i) and (ii) of Lemma B.2. Thus, Theorem 3.1 and Theorem 3.2 are proven in their entirety. \( \square \)
References


