Modelling dependence in collateralied debt obligations with copulas

by Christopher Linley

Supervisor: Prof. R Becker

Submitted to the University of Cape Town in partial fulfilment of the degree of Master of Science in Financial Mathematics

May 19, 2010

Abstract

In this paper we provide a review of credit derivatives, and some of the tools used to model them. We give a basic introduction to copulas and how they are used to model the depedence between single name credit derivatives. We then investigate various features of Gaussian and t copula dependence using numerical results obtained from Monte-Carlo simulation.

Contents

1 Introduction

1.1 Outline

In this project, we shall examine some dependence structures in Collateralised Debt Obligations (CDO's) modelled using copulas.

Section 2 provides an introduction to modelling credit risk. It provides a brief overview of firm value and first passage time models and a more thorough discussion of intensity models (we will be using intensity models in the modelling of CDO's). There is also a worked example of calibrating the intensity to market data. It ends with an introduction to basket credit derivatives.

Section 3 introduces copulas and some of the important results in the theory. It also describes how copulas may be used to model the dependence in basket credit derivatives. Section 4 motivates for some particular dependence models we will consider in our numerical modelling. Section 5 describes the numerical implementation of the model. The results of the model are given in Section 6, and a brief conclusion in Section 7. Finally, there is an appendix on Poisson processes.

1.2 The development of credit derivatives

Credit risk is ubiquitous in finance. Defaults may be uncommon, but when they do occur the repurcussions are usually substantial. Thus it is important that market participants have efficient tools to hedge against (and others to speculate on) such events. The development of credit derivatives has contributed significantly towards providing such tools.

Credit derivatives have only seen significant trading volumes in the recent past. Since large default events are uncommon, opportunities to test the modelling of such products by analysts have been scant. In the wake of the financial crisis of 2007-2009, credit derivatives have been subject to much scrutiny. In particular, the Gaussian Copula model for basket contracts, as developed by David Li [19] and widely adopted by the market [5], has attracted criticism [5].

Estimates for the size of the credit derivatives market have repeatedly surpassed predictions [3]. Table 1 provides estimates for the total notional value of credit derivatives:

Year	USD(bn)
1996	180
1997	-
1998	350
1999	586
2000	893
2001	1,189
2002	1,952
2003	3,548
2004	5,021
2005	-
2006	20,207
2007	45, 460

Table 1: Size of the credit derivatives market (source [3, 31])

Note that many of the notional amounts will never be paid out since the reference credit event does not always occur during the period of the contract. Thus, the total settlements that occur will be significantly smaller than the amounts given above.

The growth of credit derivatives has exceeded that of other over-the-counter (OTC) derivatives for many years. The notional share of the credit derivatives market in the global OTC markets grew from 2.5% in 2004 to 6.9% in 2006 [23]. Credit derivatives volumes grew 73% in 2007 compared to a total OTC derivative growth of 38% [32]. Hedge funds have been increasingly involved in the credit derivatives market, as their share of the trading volume doubled from 2004 to 2006 [3]. As the market for credit derivatives has developed, it has also been streamlined — physical settlement has been decreasing, with the balance taken up by cash settlement [3].

However, such rapid growth does not seem set to continue. The notional value of credit derivatives held by US Commerical Banks decreased by 8% in the second quarter of 2009 [34].

There are many types of credit derivatives being traded. A partial breakdown of the credit derivatives market into some of its products is given in Table 2:

Type	2000	2002	2004	2006
Basket products	6.0%	6.0%	4.0%	1.8%
Single-name credit default swaps	38.0%	45.0%	51.0%	32.9%
Synthetic CDOs - full capital	n/a	n/a	6.0%	3.7%
Synthetic CDOs - partial capital	n/a	n/a	10.0%	12.6%
Full index trades	n/a	n/a	9.0%	30.1%
Tranched index trades	n/a	n/a	2.0%	7.6%

Table 2: Partial breakdown of credit derivatives market (source [3])

Note the change in composition over the years. One reason is that the credit derivatives market is still fairly young, and new products are constantly being developed. In particular, we note the development of index trades.

2 Credit Derivatives

2.1 Credit risk

Default risk, or *counterparty risk,* is a factor in virtually all financial transactions. Often the net value of the contractual obligations at some time t will be positive for one party, say A, and negative for the other party, B. Should the contract fail to close out properly A will incur a loss.

As a simple example, we may consider the case of a zero-coupon bond. Suppose that at time $t = 0$, party A buys a bond from party B which matures at time T. Throughout the interval $[0, T]$, the contract has positive value in A 's portfolio (and negative in B's portfolio). If, during the interim period $[0, T]$ party **B** encounters financial difficulty, it may be unable to honour its contractual obligation to A at time T. We say B has *defaulted.* A has incurred a loss caused by the default of B — thus during the period of the contract, A was exposed to *default risk.*

Default occurrences are important since the losses involved are a significant proportion of (typically large) notional amounts. As a result, there are a number of indicators of the risk perceived by the market such as credit spreads and credit ratings.

Investors purchasing risky bonds face a high risk of default loss, and expect a greater return in compensation. Hence, the riskier the bond, the higher the return. The difference between the return on a risky bond and the a riskless bond (typically a government bond) is known as the *credit spread* [6]. The credit spread may be given in terms of yield to maturity or the instantaneous forward rate [6]. Larger credit spreads are indicative of higher default risk.

Important bonds will typically receive a *credit rating* from ratings agencies such as Moody's, Standard and Poor, and Fitch. These are letter grades, such as AAA (very low risk) and B- (very high risk).

Global Corporate Cumulative Average Default Rates (1981 - 2008) (%)						
Rating	Years					
	1	**3**	**5**	**7**	**10**	**15**
AAA	0.00	0.09	0.27	0.40	0.55	0.65
AA	0.03	0.14	0.34	0.56	0.83	1.20
A	0.08	0.34	0.72	1.21	1.94	2.91
BBB	0.24	1.17	2.43	3.59	5.16	7.70
BB	0.99	5.07	9.07	12.41	16.02	19.33
B	4.51	14.43	20.58	24.46	28.41	33.14
CCC/C	25.67	39.25	44.93	47.45	50.33	52.93

Table 3: S&P ratings and associated default rates (source [35])

Even if default has not yet occurred, the threat of default can cause an investor to incur a loss. If an investor has bought corporate bonds from a firm and new information suggests this firm is more likely to default, the credit spread may increase - resulting in a loss even if he sells the bonds to another investor before the (likely) default. We use the term *credit risk* as a generic term to describe the risk associated with default events, or the risk associated with changes in credit spreads or credit ratings.

A *credit derivative* is derivative whose value depends on some reference credit risk, whether it be a default event, a change in credit spread, or a change in rating. Note that the credit derivative itself may be subject to counterparty risk.

2.2 Single name credit default swaps

One of the most popular credit derivatives is the single name *credit default swap* (from here on we will refer to them as CDS's). We see in Table 2 that CDS's make up a large portion of the credit derivatives market. CDS's have been around in a variety of forms for many years, but it is only recently that CDS's have become a fairly standard OTC contract. CDS's are very liquid, thus models are not required to value them but to be consistently calibrated to market CDS quotes [8] - later we will give an example of calibrating an intensity model to CDS quotes. Before we describe CDS's, we briefly mention some properties of defaultable bonds.

Defaultable bonds may be *corporate bonds,* issued by a firm, or *sovereign bonds,* issued by a national government (a notable sovereign default was that of Russia in 1998 [12]). When a firm or government defaults, it is unable to fully repay it's debt and thus, in general, all its creditors will not be fully repaid. In this project we will differentiate between the quality of the debt, but not the nature of the underlying.

We again consider the zero-coupon bond example given in Section 2.1, where A owns a bond issued by B. If B becomes insolvent it may be unable to fully repay its debt to A. The proportion of its debt that it repays is called the recovery rate, $REC \in [0, 1]$. If $REC = 1$ then A incurs no loss at all , while $REC = 0$ signifies a complete loss to A. The proportional loss is called the *loss given default* $LGD = 1 - REC$.

A *credit default swap* (CDS) is an instrument that provides insurance against the loss incurred by creditors when a firm defaults on its debt. Two parties: A (a protection seller) and B, (a protection buyer), enter into a contract based on the risk of default of a reference firm C. B will pay A a (usually quarterly) coupon at a certain rate, (called the *spread* or *rate)* on a notional amount K. In return, if C defaults, B will receive a payout from A equal to $(1 - REC_C)K = LGD_C K$. The quarterly amounts paid by B to A is called the *premium leg,* while the amount paid by A to B on the event of default is called the *protection leg.* A is said to be short protection on C, while B is long protection on C.

In the above situation, B was able to trade away some it's risk exposure on C to A, and paid A a premium in compensation. Thus CDS's provide a facility for market participants to *trade risk*. Due their liquidty, CDS's have become an efficient tool for trading risk.

There are a number of stipulations in a CDS contract to avoid ambiguity. We list some of the main variables.

- Reference entity
- The time to maturity of the contract
- The rate/spread that is paid by the protection buyer
- The currency in which payments are to be made
- The determination of the recovery rate
- What constitutes a credit event (triggering the protection payment)
- When will the protection payment be made (assuming a default occurs)

We will speak of the *payment dates* $\{T_1 < T_2 < \ldots < T_n\}$, which are the dates (in years) at which the premium amounts are paid. In this discussion we will assume the initial date of the contract to be $T_0 = 0$. The date of the last payment T_n is generally the date of maturity: $T_n = T$. The rate may be paid quarterly over the entire period of the contract, or until default, this will be stipulated in the contract. Sometimes the expected discounted premium amounts are converted into an upfront lump sum. This is especially true for riskier CDS's. There are other variations as well, where the quarterly rate is a fixed 1% coupon, and an upfront settlement is made to render the contract fair at inception.

The currency of payment is usually USD (US dollars) or EU (Euros). There are a number of possible specifications of what constitutes the credit event, for a full list of those mentioned in the ISDA 2003 Agreement see the Markit Data Guide [21]. Here we simply mention two credit events: *bankruptcy* and *restructuring.* As mentioned in [21], the definitions of what constitutes restructuring are controversial and thus the stipulations need to be very precise. When modelling defaults in this discussion, we will consider only one type of event as representative of default.

We will be modelling the recovery rate as a given constant. In the market, the recovery rate may be random. After a firm has defaulted, its assets are usually sold in an auction conducted by the banks. Note some banks may have a net long or short protection on a recently defaulted firm. If such a bank were to participate in the auction, it would have a vested interest in the recovery rate. If it is short protection it would want the recovery rate to be as high as possible, and vice versa.

The money recovered from the auction is transferred to creditors at some later date. As an example we detail the recovery rates detailed on three CDS's after their respective credit events: The Federal National Mortgage Association (FNMA); The Federal Home Loan Mortgage Group (FHMLC), and Washington Mutual Inc. (WM).

CDS	EDD	Auction Date	Settlement Date	Recovery
FHLMC SNRFOR USD XR	2008/08/09	2008/10/06	2008/10/15	0.94
FNMA SNRFOR USD XR	2008/08/09	2008/10/06	2008/10/15	0.92
WM SNRFOR USD XR	2008/09/29	2008/10/23	2008/11/07	0.57

Table 4: Recovery rates for market firms (EDD stands for *Event Default Date)*

The code *SNRFOR* stands for *Senior Unsecured Debt, Foreign Currency Sovereign Debt* which specifies the type of the underlying. The code *XR* is a code that specifices what is deemed a credit event. There are four codes: *XR, CR, MM* and *MR.* The more events that fall under a certain code, the more likely a credit event, so the probability of a protection is payment is higher. Thus the corresponding rates are also higher.

Since the credit crisis, there has been much deliberation on how to improve the trading of CDS's. At the moment most trades take place directly between counterparties, and thus scope for complete standardisation and monitoring is limited. It has been proposed that the trading of CDS's should become more centralised. We refer the reader to Casey [9].

Like most market quotes, CDS's are subject to bid/offer spreads that are usually determined by the liquidity. We will be examining fairly liquid products, so bid/offer spreads will be ignored.

Brigo *Sz* Mercurio [8] discuss a few CDS variations that we shall repeat here: running CDS (R-CDS), postponed payments running CDS (1) (PR1-CDS), postponed payments CDS (2) (PR2-CDS). We assume, as before, that A is the protection seller and B is the protection buyer on a notional amount K, with recovery rate REC, maturity T and the usual payments grid. For the three running CDS's, if default does not occur during the contract then the respective cash flows are identical, namely B pays A a coupon at a predefined rate at the dates T1, $T2$, $Tn = T$. Since there is no default, A does not make any payments to B. If default occurs at a grid point T1, then B pays A the premium on each T1, T2, , Ti and A pays B the protection leg (1 — $REC)K$ at T. If default occurs at time τ where Ti < < Ti+1 then, the contracts look exactly the same up to T. after this we have the following variations in cash flows (time is measured in years)

- R-CDS: At τ, A pays B amount $(1 - REC)K$, B pays A amount $(\tau - T_i)R$
- PR1-CDS: at T_{i+1}, A pays B amount $(1 - REC)K$, B makes no further payments.
- PR2-CDS: at T_{i+1}, A pays B amount $(1-REC)K$, at T_{i+1}, B makes one final payment $(T_{i+1} - T_i)R$

Of course, cashflows on the same day can be netted. Finally, in an upfront payment CDS (U-CDS), the present value of the premium leg is calculated and paid by B to A at the start of the contract.

We will be modelling CDS's using intensity models. Before we do that, we briefly mention two other methods for modelling default risk.

2.3 Single entity models

2.3.1 Merton firm value approach

Merton's Firm Value Model [24] was one of the first important attempts at modelling defaultable bonds [2, 10]. This model is based on the general Black-Scholes assumptions [24 no transactional costs or taxes; continuous trading; infinite divisibility of assets; unlimited borrowing and lending at a common interest rate so that each investor may buy as sell as much of any asset as possible; short selling is permissible; the term structure of interest rates is a flat rate r; the value of the firm is modelled as a diffusion process:

$$dV = \mu V dt + \sigma V dW \tag{1}$$

We would like to estimate the parameters in (1). The total value of the firm is not actively traded in the market, but the equity of the firm is, and we may use the share price to infer some behaviour on $V(t)$.

We consider the evolution of firm's value $V(t)$ and it's debt $L(t)$ over time. If at time T the firm is required to repay its liabilities due on that date, $L(T)$, but its total value is too low: $V(T) < L(T)$), then we say the firm has defaulted. Thus, at time T, the firm pays to creditors min(V(T), $L(T)$). This is equivalent to saying that when the firm issues debt, it retains a call option to buy back the debt for the value of the firm at the future date T. We may also formulate this by assuming the firm buys a put option from its creditors allowing it to sell its total assets at T to creditors for the price of the total debt is has issued[2]. If we ignore seniority of debt and assume the firm is able to liquidate all its assets at the fair price, we will have a recovery rate $REC =$ Y,M.

We use this put option characterisation to model the value of a defaultable bond $Pd(t, T)$ (the value at t of a unit notional defaultable bond issued by the firm) using the usual Black-Scholes option pricing formula. The value of the defaultable bond Pd is given by:

$$P^d(t,T) = P(t,T) - p(t)$$

$P(t, T)$ is the value of the riskless unit bond at t that matures at T, while $p(t)$ is the value of european put option with maturity T on a unit notional of the firm value V with strike $K = 1$. According to the Black-Scholes formula we have:

$$p(t) = V\Phi(-d) - e^{-r(T-t)}\Phi(-d + \sigma\sqrt{T-t})$$

where ø(x) is the value of the cumulative standard normal distribution at x, and d is given by:

$$d = \frac{\log(\frac{V}{1}) + (r + \frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{T-t}}$$

The basic Merton model is quite limited. It assumes that all debt has the same maturity date, and that default may only be decided on this date. It does not make allowances for debt seniority. Furthermore, even if a firm has sufficient assets to cover its debt, liquidity problems may render it unable to repay its creditors timeously [2].

In his paper [24], Merton extended the model to include callable coupon bonds. A number of other extensions have been made over the years, we refer the reader to Amman [2] and Bielecki Sz Rutkowski [6] for more details.

2.3.2 First passage time models

First passage time models are a fairly natural extension of Merton's original model, and were introduced by Black and Cox [7]. Again we consider the firm value *V(t),* and now we introduce a time dependent barrier *b(t).* The assumption is that default occurs on the first occasion that *V(t)* crosses the barrier *b(t).* We see that this easily allows us to model an arbitrary default time. Each model also specifies a recovery rate that can be made to fit bond covenants, banktruptcy costs and taxes[6]. We can obtain Merton's model from the first passage time framework by setting:

$$b(t) = \begin{cases} \infty & if \quad t < T \\ L(T) & if \quad t \geq T \end{cases}$$

There have been a great many first passage time models discussed in the literature, any reasonable guess for the barrier process and the recovery rate leads to a new model. For list of some first passage time models and their respective parameterisations we refer the reader to Amman [2] and Bielecki & Rutkowski [6].

Firm value and first passage time models are classified as *structual models.* They model the asset/liability structure of the firm parametrically, and once they have estimated the parameters of the asset/liability dynamics they are used to make inferences on the likelihood of default. Intensity models — introduced in the next chapter, and the models we will be focusing on in this discussion — take another approach. Instead of focusing on the specifics of the firm, they assume that default is driven by an exogenous variable that is completely hidden from the market. Given this exogenous variable, intensity models proceed to add dependence on the market data by a suitable transformation.

2.3.3 Intensity models

In *intensity models,* the time of default of a firm is driven by a jump process. Usually it is modelled as the first jump time of a *Poisson process.* We assume a basic familiarity with Poisson processes here. For an outline of Poisson processes the reader may consult the appendix.

The first formal continuous-time intensity model appears in Jarrow and Turnbull [2], where they start by modelling default in discrete time and extend it to the continuous setting [18]. As mentioned in the last section, the variable driving the time of default $\tau > 0$ is assumed to be completely exogenous to observable economic variables such as interest rates and share prices. Formally, we define the default time as:

$$\tau = \inf_{t \geq 0}(N_{\lambda(t)}(t) > 0) \tag{2}$$

where *NA(t)(t)* is a Poisson process with *intensity,* or *hazard rate A(t)* > 0 defined Vt > 0 (we will always be assuming that *t* > 0). Note how our initial step is to bypass the firm dynamics of *V(t).* For this reason, intensity models are also referred to as *reduced form models.*

A useful function related to the intensity process A(t) is the *cumulated intensity* process, or *cumulated hazard rate, A(t)* given by:

$$\Lambda(t) = \int_0^t \lambda(s)ds$$

Note that since we assumed A(t) is strictly positive, it follows that A(t) is strictly increasing. We also assume that A([O, oo)) = [0, oo) (so ***T* <** *oo a.^* ***).*** In this discussion we will assume A(t) is piecewise continuous, and thus A(t) will be piecewise differentiable. The intensity process A(t) is an indicator of how likely default is at time *t* — for small *h* the probability of default during *[t, t* h] is approximately equal to *A(t)h,* hence the term 'hazard rate'. The cumulated intensity process A(t) is a measure of the likelihood that default occurs before *t.*

The only concern when setting up a first jump time intensity model is to determine the dynamics of A(t). Once we have specified the dynamics of A(t) the distribution of the default time is uniquely determined (not always in closed form). As mentioned earlier, since CDS's are very liquid, we usually need to calibrate the intensity process so that the implied fair rates match the market quotes. Perhaps the simplest choice for the intensity process (that can still be calibrated to the market) is to make it piecewise constant. We shall provide an example of such a calibration in the next section.

The distribution function for ***T*** is given by:

$$\Pr(\tau \leq t) \;=\; 1 - e^{-\Lambda(t)} \tag{3}$$

Sometimes people speak of the *survival probability* $S(t)$:

$$\begin{aligned} S(t) &= \Pr(\tau > t) \\ &= 1 - \Pr(\tau \leq t) \\ &= e^{-\Lambda(t)} \end{aligned}$$

Since A(t) is strictly increasing and has range [0, oo), it has a unique inverse A'(t). If we set = AN, then e has a standard exponential distribution:

$$\begin{aligned}
\Pr(\xi \leq t) &= \Pr(\Lambda(\tau) \leq t) \\
&= \Pr((\tau \leq \Lambda^{-1}(t)) \\
&= 1 - e^{-\Lambda(\Lambda^{-1}(t))} \\
&= 1 - e^{-t}
\end{aligned}$$

Thus if we wish to simulate a random sample from ***T*** we can simulate a standard exponential random variable and transform it appropriately:

$$\tau = \Lambda^{-1}(\xi) \tag{4}$$

The right hand side of (3) is differentiable almost everywhere and induces a measure tt, that is absolutely continuous with respect to the usual Lebesgue measure. Since we have already used A as our intensity process, we let 7 denote the Lebesgue measure on R The density function/Radon-Nikodym derivative is given by:

$$\begin{aligned}
\frac{d\mu_\tau}{d\gamma} &= \frac{d}{dt}\left[1 - e^{-\Lambda(t)}\right] \\
&= e^{-\Lambda(t)}\lambda(t)
\end{aligned} \tag{5}$$

which is uniquely defined almost everywhere, so that for A c [0, oo) and ***f*** defined on A:

$$\begin{aligned}
\int_A f(s)d\mu_\tau &= \int_A f(s)\frac{d\mu_\tau}{d\gamma}d\gamma \\
&= \int_A f(s)e^{-\Lambda(s)}\lambda(s)ds
\end{aligned} \tag{6}$$

(6) will be our main tool in pricing CDS's. Now assume ***r*** is a random variable on the usual probability space (Q, ***,F,*** IP). Recall the no-arbitrage pricing formula for a contingent claim:

- Given a contingency claim ***X*** paying an amount ***XT*** at T, the value of ***X*** at time 0 is given by:

$$\frac{V_X(0)}{S_0} = \mathbb{E}_S\left[\frac{X_T}{S_T}\right]$$

where (St)0<t<7, is a numeraire and Es[.] is the expectation under the associated equivalent martingale measure Q.

Usually the numeraire chosen is the risk free asset Bt where Bo = 1, so that:

$$V_X(0) = \mathbb{E}\left[\frac{X_T}{B_T}\right]$$

In this discussion, the expectation $\mathbb{E}[\cdot]$ (without a subscript) will be the expectation under the risk-neutral measure $\mathbb{Q}$:

$$\mathbb{E}[X(\omega)] = \int X(\omega)d\mathbb{Q}$$

If we let

$$D(0,T) = \frac{B_0}{B_T}$$

denote the discount factor, we have $D(0,0) = 1$ and:

$$V_X(0) \;=\; \mathbb{E}[D(0,T)X_T] \tag{7}$$

We will be making use of the risk neutral pricing formula (7) to value the premium and protection legs of a running single name CDS (R-CDS). We make the following assumptions:

- We have the expected values of the discount factors E[D(0, *t)] = P(0, t)* for all 0 < *t* < *T* where *T* is the maturity of the CDS we are pricing.
- We have the risk neutral distribution of the default time 7- — i.e. we are modelling A(t) under the risk neutral measure
- **T** is independent of interest rates

We let lA denote the indicator function, i.e.

$$\mathbf{1}_A(x) = \begin{cases} 1 & if \quad x \in A \\ 0 & if \quad x \notin A \end{cases}$$

We will determine the present value of the protection and premium legs, *X* and Y of a RCDS on a notional of *K* with constant recovery rate *REC,* premium rate *R,* and a payments grid [0 = To < T1 < ... < TT, = T]. We start with the protection leg.

If default occurs at time $\tau < T$, then the protection leg payment K(1 — *REC)* is also made at time r. If $\tau > T$ then no payment is made. Thus the (undiscounted) payoff X' is given by

$$X'(\omega) = K(1 - REC)\mathbf{1}_{[0,T]}(\tau(\omega))$$

Taking the discounted expectation of X' with respect to the risk neutral measure we have:

$$\begin{aligned} X &= \mathbb{E}[D(0,\tau(\omega))X'(\omega)] \\ &= K(1-REC)\mathbb{E}[D(0,\tau(\omega))\mathbf{1}_{[0,T]}(\tau(\omega))] \end{aligned}$$

(using the linearity of E). We let ***6x(A)*** denote the Dirac measure:

$$\delta_x(A) = \begin{cases} 1 & if \quad x \in A \\ 0 & if \quad x \notin A \end{cases}$$

and we have the usual distributional relation for 6x and ***x*** E A:

$$\int_A f(y)\delta_x(y)dy = f(x)$$

Following Brigo & Mercurio [8], we integrate for s over $[0,\infty)$:

$$\begin{aligned} X &= K(1-REC)\mathbb{E}[D(0,\tau(\omega))\mathbf{1}_{[0,T]}(\tau(\omega))] \\ &= K(1-REC)\mathbb{E}\left[\int_0^\infty D(0,s)\mathbf{1}_{[0,T]}(s)\delta_{\tau(\omega)}(s)d\gamma(s)\right] \\ &= K(1-REC)\mathbb{E}\left[\int_0^T D(0,s)\delta_{\tau(\omega)}(s)d\gamma(s)\right] \\ &= K(1-REC)\int_0^T \mathbb{E}[D(0,s)\delta_{\tau(\omega)}(s)]d\gamma(s) \\ &= K(1-REC)\int_0^T \mathbb{E}[D(0,s)]\mathbb{E}[\delta_{\tau(\omega)}(s)]d\gamma(s) \end{aligned}$$

We can change the order of integration by Fubini's theorem since (Q,,F,P) and ([0, oo), B([0, oo), 7) are a-finite, and the integrand is non-negative. Also, we make use of our assumption that default time is independent of the interest rate to factorise our expectations. Note that for the interval *[s, s ± ds)* there is an *s** E *[s,* ***s*** *± ds)* such that:

$$
\begin{aligned}
\mathbb{E}[\delta_{\tau(\omega)}([s, s+ds)] &= \mathbb{Q}(\tau^{-1}([s, s+ds))) \\
&= \mu_\tau([s, s+ds)) \\
&= \frac{d\mu_\tau}{d\gamma}(s^*)ds \\
&= e^{-\Lambda(s^*)}\lambda(s^*)ds
\end{aligned}
$$

where the last line follows from (5). So we have:

$$
\begin{aligned}
X &= K(1-REC)\int_0^T \mathbb{E}[D(0,s)]e^{-\Lambda(s)}\lambda(s)ds \\
&= K(1-REC)\int_0^T P(0,s)e^{-\Lambda(s)}\lambda(s)ds \qquad (8)
\end{aligned}
$$

where P(0, ***t)*** is the current expected value of unit notional, risk free, zero coupon bond with maturity ***t.*** Extending this analysis to the premium leg requires a change in the indicator function, and an appropriate weighting of the annual rate ***R.*** If τ E Ti+1], we make the full rate payments K(1 — *REC)R(Tk — Tk_i)* for all ***Tk*** < Ti and one final payment *R(T* — Ti). So the undiscounted payoff Y' is given by:

$$
\begin{aligned}
Y'(\omega) &= K(1-REC)R\sum_{i=1}^{n}\Big[(T_i - T_{i-1})\mathbf{1}_{[T_i,\infty)}(\tau(\omega)) \\
&\quad + (\tau(\omega) - T_{i-1})\mathbf{1}_{[T_{i-1},T_i]}(\tau(\omega))\Big]
\end{aligned}
$$

We use the same argument as before, remembering that terms of the form

$$
(T_i - T_{i-1})\mathbf{1}_{[T_i,\infty)}(\tau(\omega))
$$

must be discounted back from T_i, not τ_ω.

$$
\begin{aligned}
Y &= \mathbb{E}\Bigg[K(1-REC)R\sum_{i=1}^{n}\Big[D(0,T_i)(T_i-T_{i-1})\mathbf{1}_{[T_i,\infty)}(\tau(\omega)) \\
&\quad +D(0,\tau(\omega))(\tau(\omega)-T_{i-1})\mathbf{1}_{[T_{i-1},T_i]}(\tau(\omega))\Big]\Bigg] \\
&\vdots \\
&= \sum_{i=1}^{n}\Bigg[P(0,T_i)(T_i-T_{i-1})e^{-\Lambda(T_i)}+ \\
&\quad +\int_{T_{i-1}}^{T_i}P(0,s)(s-T_{i-1})e^{-\Lambda(s)}\lambda(s)ds\Bigg] \qquad (9)
\end{aligned}
$$

Similar calculations may be used for determining the fair rates for the postponed running CDS's (PR1-CDS, PR2-CDS). We shall use (8) and (9) when applying our model to market data.

2.3.4 Calibrating the intensity to CDS market data

We now consider the problem of using the market rates of a CDS to obtain a risk neutral intensity process. We list the assumptions made:

- We have a running CDS.
- The notional is 1.
- We assume the contract starts at $T_0 = 0$, and we have a payments grid $[0 = T_0 < T_1 < T_2, \ldots < T_n = T]$, where T is the date of maturity of the contract.
- The fair rates R_i, $1 \leq i \leq m$ have been quoted for the CDS over the periods $[0, S_1], [0, S_2], \ldots, [0, S_m]$ for $S_1 < S_2 < \ldots < S_m$ and $S_m = T$, where each $S_i = T_j$ for some j.
- We have the bond prices $P(0, t)$ for all $0 \leq t \leq T$.
- The recovery rate REC is constant
- The distribution of the default time follows a Poisson process with piecewise one-parameter intensity so that

$$
\lambda(t) = \overline{\lambda}_1(t)\mathbf{1}_{[0,S_1]}(t) + \overline{\lambda}_2(t)\mathbf{1}_{[S_1,S_2]}(t) + \ldots + \overline{\lambda}_m(t)\mathbf{1}_{[S_{m-1},S_m]}(t)
$$

where $\overline{\lambda}_i(t)$ is a function of t and the i^{th} parameter λ_i^*. We have the usual cumulated density $\Lambda(t) = \int_0^t \lambda(t)dt$. We let $\Lambda_i(t)$ and $\lambda_i(t)$ denote the restriction of $\Lambda(t)$ and $\lambda(t)$ to the interval $[0, S_i]$, so that

$$\Lambda_i(t) = \Lambda_i(t; \lambda_1^*, \lambda_2^*, \ldots, \lambda_i^*)$$

and

$$\lambda_i(t) = \lambda_i(t; \lambda_1^*, \lambda_2^*, \ldots, \lambda_i^*)$$

Having listed all the assumptions, we work inductively, determining λ_1^*, then the subsequent λ_i^*'s.

- For the first period quoted period $[0, S_1]$ we calculate the present value over the period $[0, S_1]$ of the protection leg X_1 and the premium leg Y_1. Using the theory from the previous section the present value of the protection leg is given by:

$$\begin{aligned} X_1 = (1 - REC) \int_0^{S_1} P(0,s)e^{-\Lambda_1(s)}\lambda_1(s)ds & \\ &= X_1(\lambda_1^*) \end{aligned}$$

and the present value of the premium leg is:

$$\begin{aligned} Y_1 &= R_1 \sum_{0<T_k\leq S_1} \left[P(0,T_k)e^{-\Lambda_1(T_i)} + \int_{T_{k-1}}^{T_k} (s - T_k)P(0,s)e^{-\Lambda_1(s)}\lambda_1(s)ds \right] \\ &= Y_1(\lambda_1^*) \end{aligned}$$

We wish to equate these two values. We set:

$$\begin{aligned} F_1 &= X_1(\lambda_1^*) - Y_1(\lambda_1^*) \\ &= F_1(\lambda_1^*) \end{aligned}$$

Then we solve for λ_1^* using a root finding method.

- Now we assume we have determined the values of $\lambda_1^*, \lambda_2^*, \ldots \lambda_i^*$, and we wish to determine the value of λ_{i+1}^*. We have:

$$\begin{aligned} X_{i+1} &= (1-REC)\left\{\sum_{k=1}^{i}\left[\int_{S_{k-1}}^{S_k} P(0,s)e^{-\Lambda_i(s)}\lambda_i(s)ds\right] + \int_{S_i}^{S_{i+1}} P(0,s)e^{-\Lambda_{i+1}(s)}\lambda_{i+1}(s)ds\right\} \\ &= X_{i+1}([\lambda_1^*, \lambda_2^*, \ldots, \lambda_i^*, \ldots, \lambda_i], \lambda_{i+1}^*) \end{aligned}$$

and:

$$\begin{aligned} Y_{i+1} &= R_{i+1}\left\{\sum_{0<T_k\leq S_i}\left[P(0,T_k)e^{-\Lambda_i(T_k)} + \int_{T_{k-1}}^{T_k}(s-T_k)P(0,s)e^{-\Lambda_i(s)}\lambda_i(s)ds\right]\right. \\ &= \left. + \sum_{S_i<T_k\leq S_{i+1}}\left[P(0,T_k)e^{-\Lambda_{i+1}(T_k)} + \int_{T_{k-1}}^{T_k}(s-T_k)P(0,s)e^{-\Lambda_{i+1}(s)}\lambda_{i+1}(s)ds\right]\right\} \\ &= Y_{i+1}([\lambda_1^*, \lambda_2^*, \ldots, \lambda_i^*, \ldots, \lambda_i], \lambda_{i+1}^*) \end{aligned}$$

We assumed that $\lambda_1^*, \lambda_2^*, \ldots, \lambda_i^*$ have already been determined, is λ_{i+1}^*. Working as before we have:

$$\begin{aligned} F_{i+1} &= X_{i+1}(\lambda_{i+1}^*) - Y_{i+1}(\lambda_{i+1}^*) \\ &= F_{i+1}(\lambda_{i+1}^*) \end{aligned}$$

and we solve for λ_{i+1}^*.

This process will usually be computationally intensive. A root finding method will need to evaluate the function F a number of times to find an approximation of the root, and each call it makes will require the numerical evaluation of the definite integrals. Depending on the interpolation method of the term structure and the specification of the Yi-:(t) 's, evaluating the integrands (which themselves include the definite integral from the cumulated intensity) may also be costly.

One obvious method to speed computation is to keep a running total of the Xi 's, and the factors of the Ri's so that we don't recalculate the same integrals each time the root finding method evaluates the difference Xi — Yi Web should also choose (t) 's that are easy to integrate.

Intensity processes were calibrated for single name CDS data. The details are as follows:

- The intensity processes were assumed to be piecewise constant

- Quotes were obtained for some running CDS's that formed part of the CDX.NA.IG.9-V3 index on 30 October 2009
- The zero coupon bond prices were also obtained on this date
- The term structure was calculated using simple linear interpolation on the bond prices
- The recovery rate is a determinstic 40%
- Settlement occurs on the same day as default
- The day count was the actual calendar days (the daycount convention will make little different to our calculations)

The (risk neutral) calibrated intensities obtained, in basis points, are listed below:

Start Date	End Date	Raytheon Co	MBI InsCorp	IBM	Time Warner Inc	Wal Mart Stores Inc
2009/10/30	**2010/06/21**	33.7	5292.17	21.76	27	54.92
2010/06/21	**2010/12/20**	36.79	5468.55	22.15	37.52	58.41
2010/12/20	**2011/12/20**	50.25	6766.78	33.39	54.98	74.32
2011/12/20	**2012/12/20**	60.11	7179.39	41.88	63.92	79.44
2012/12/20	**2013/12/20**	69.3	7182.67	49.67	75.25	81.4
2013/12/20	**2014/12/22**	77.06	7106.81	56.17	91.51	89.61
2014/12/22	**2016/12/20**	83.53	6739.12	63.75	101.89	99.43
2016/12/20	**2019/12/20**	88.77	6346.18	69.64	115.22	108.22
2019/12/20	**2024/12/20**	91.82	6168.38	73.61	122.68	115.19
2024/12/20	**2029/12/20**	94.07	6199.91	57.88	112.34	120.22
2029/12/20	**2039/12/20**	95.46	5985.68	73.01	109.69	128.01

Table 5: Piecewise risk neutral constant intensities (bps) calibrated to 30 October 2009

For low risk companies, the intensity tends to increase over time. We may account for this by noting that the further into the future we look, the more uncertain the state of the company. The clear outlier in the above table is MBI-InsCorp. It appears a default is imminent. If a firm is about to default, trade on the long term CDS's may cease all together, so that current CDS prices are not up to date. This may account for the decline in intensitiy of MBI-InsCorp after 2012.

2.4 Basket credit derivatives

While a single name CDS references a single debt instrument, a basket credit derivative references multiple debt instruments. Perhaps the simplest example of a basket product is a *first to default swap* (FtD). Again, we have A selling protection, and ***B*** buying protection, but this time we have a number of reference entities C1, C2, • • • , *Cn* with default times 7-1, y2, , **Tn.** The cash flows are exactly the same as those for a single name CDS except that the default event is defined as the first time that any one of the firms in the basket defaults:

$$\tau_{\text{FtD}} = \min(\tau_i) \quad i = 1, 2, \ldots n$$

If ***B*** has exposure to debt issued by any (or all) of the Ci's, then a FtD allows ***B*** to hedge against this risk.

We can say immediately that the fair rate for the first to default swap **RFtD** should be at least as high as the individual CDS rates:

$$R_{\text{FtD}} \geq \max(R_{C_i}) \quad i = 1, 2, \ldots n$$

since the protection leg payment of the FtD is always greater than or equal to the protection leg payment of each of the component CDS's. Clearly **RFID** depends on the joint distribution of the ri's. If there is perfect dependence between the ri's, so that knowing any one of them implies we know the rest, then for each w E
would have a minimum default time Tn, E (r1, 72, • • •

$$\tau_m(\omega) \leq \tau_i(\omega) \quad i = 1, 2, \ldots, n$$

and **RFID** = ***Rm.*** If dependence is negative between the default times, then we are likely to have at least one default occurring early, so that **RFtr)** will be much higher than the individual Ri's. If the dependence is positive, then the individual default times will be similar, and **RFtp** will be similar to the Ri's. Now we consider *collateralised debt obligations (CDO's).*

A CDO is an asset-backed security formed by taking a number of bonds from reference entities **C1, C2, . , Cn** with respective notionals N1, **N2, . , Nn** and pooling them into a large composite portfolio where each bond has a weighting wi:

$$w_i = \frac{N_i}{N_{\text{CDO}}}$$

where **NcDo =** Ni is the total notional of the CDO. Shares in the CDO are then sold off to investors, so that each unit notional **(kip°** owned may be decomposed as:

$$U_{\text{CDO}} = \sum_{i=1}^{n} w^i U_i$$

where Ui is a unit notional of the debt issued by Ci Thus (assuming no other costs) the investor will receive coupons/notional amounts from each Ci that are weighted by wi, similarly a unit notional of the CDO will have a default exposure:

$$\beta = \sum_{i=1}^{n} w_i(1 - REC_i)$$

where *RECi* is the recovery rate of Ci (we assume it is constant). Furthermore, a seniority structure is usually imposed on the CDO by *tranching* it into several portions, each with a different risk to exposure. This is done by considering a set of attachment points 0 = ao < al < < æm 1, and the tranches are defined by the intervals [ao, ad, [ai, a2], , am].

Assume we have the ordered default times y1 < T2 < ...Tn. We let *i(Tk)* denote the index of the kth reference entity with to default, so that Ti(Tk) Tk. In this setting, investors with a unit notional share in the first (most junior) tranche [ao, ail will incur losses:

$$l_1 = \frac{1}{\alpha_1 - \alpha_0}\left[[1 - REC_{i(\tau^1)}]w_{i(\tau^1)} + [1 - REC_{i(\tau^2)}]w_{i(\tau^2)} + \dots\right]$$

at each yi until the tranche is completely wiped out i.e. until /1 = 1. After the first tranche has been wiped out, the second tranche will incur losses with each subsequent default. If we let l denote the total loss to the ith tranche by the kth default, then:

$$\begin{aligned} l_2 &= \frac{1}{\alpha_2 - \alpha_1}\left[\max\left([1 - REC_{i(\tau^1)}]w_{i(\tau^1)} - l_1^0 - \alpha_1, 0\right)\right. \\ &\quad + \max\left([1 - REC_{i(\tau^2)}]w_{i(\tau^1)} + l_1^1 - \alpha_2, 0\right) \\ &\quad \left. + \dots\right] \end{aligned}$$

until l2 = 1. An example is a bit more transparent. We consider the total cumulated loss per tranche given the total cumulated loss to the portfolio at times

	Total Loss	0%-3% Tranche	3% - 7% Tranche	7% - 10% Tranche	10%-15% Tranche	15%-30% Tranche	30%-100% Tranche
T_1	.02	.67	.00	.00	.00	.00	.00
T_2	.06	1.00	.75	.00	.00	.00	.00
T_3	.15	1.00	1.00	1.00	1.00	.00	.00
T_4	.18	1.00	1.00	1.00	1.00	.20	.00
T_5	.37	1.00	1.00	1.00	1.00	1.00	.10

Table 6: Example of losses per tranche

The junior tranches are exposed to more risk, since they are the first to absorb default losses. In contrast, the senior tranches are only affected if there are a large number of defaults in the portfolio. Thus investors in the junior tranches will receive a higher rate of return in compensation for the exta risk.

More recently, *synthetic* CDO's have been created. Here there is no underlying pool of assets. Instead, its cashflows are determined by a hypothetical mix of single name CDS's. One party receives premium payments on each tranche, and in return it will make protection payments on the proportional losses to each tranche.

The last few years has seen the rapid development of index trades (see Table 2). Here standardised mixes of CDS's are set up managed by the index company, and market participants can trade on the index. Market participants seeking protection will pay a premium amount on a certain index, while participants looking to sell protection will receive the premiums. If there is a loss to the tranche the protection sellers will have to make payments to the protection buyers. The two main CDO indices are the CDX (North America) and iTraxx indices (Europe), managed by the Markit Group.

3 Copulas

3.1 History

The word *copula* is taken from latin where it means 'link' or 'join', and is used by linguists to denote a verb that joins together the subject and complement of a sentence. Early traces of copulas can be found in the 1940's. Wassily Hoeffding introduced standard bivariate distributions on [-1, x '2--] as early as 1940 [26]. Many results about copula's arose from the study of probability metrics and t-norms, as suggested by Karl Menger as a generalisation of metric spaces in 1942 [28]. Some t-norms may be used as copulas on R2 and vice versa.

The term 'copula' was first used in a mathematical context by Abe Sklar to describe a class of functions featured in his eponymous theorem in 1959. He used the term to describe functions that could be used to couple univariate distribution functions on R to create multivariate distribution functions on R.

3.2 Outline

We will list some of the important results in the theory of copulas. Statements of these results may be found in most introductory texts, such as Nelsen [26] and Cherubini [11]. For an introduction, the reader is advised to consult Nelsen [26] who provides proofs of these results for the 2-dimensional case — from which a good intuition may be developed.

Since copulas defined on the unit square [0,1] x [0, **1]** admit easy visualisation as surfaces in R3, the text provides such figures where appropriate.

We be working with random variables assuming values in R i.e. their support is the whole real line. Random variables will be written in uppercase, e.g. *X,* and the associated measure and distributions function as *btx* and *Fx* respectively. Vectors will be written in **bold,** e.g. x, with **0** and **1** representing the zero and unit vectors respectively. A vector of random variables will be denoted **X.** Although we have already used 1A(x) as the indicator function, the context will remove any ambiguity.

As we will often encounter the limiting behaviour of functions *Fx(x)* as *x* ±oo, we introduce the extended real line R = R U {—oo, opal (we may also consider the extended real plane R2 etc.). We extend *Fx* in a natural fashion:

$$F_X(-\infty) = \lim_{x \to -\infty} F_X(x)$$

and

$$F_X(\infty) = \lim_{x \to \infty} F_X(x)$$

The image/range of a function *F* well be denoted *F(S)* where *S* is the domain of *F.*

When working with general points in ṙin we shall typically use **x** and **y.** When considering vectors restricted to the unit n-hypercube (which is the domain of the n-dimensional copula) we shall typically use **u** and **v,** since u and v are commonly used to denote percentiles in the literature.

We also intoduce some convenient notation as used by Schoenbiicher Siz Schubert [30]. We let:

$$H(\mathbf{x}_{-i}, y) := H(x_1, x_2, \ldots, x_{i-1}, y, x_{i+1}, \ldots, x_n)$$

I.e. we simply replace the i^{th} component of $\mathbf{x}$ by y.

3.3 Distribution functions on $\overline{\mathbb{R}}$

We briefly revise cumulative distribution functions, or just *distribution functions* on $\mathbb{R}$. We will be working in the usual probability space $(\Omega, \mathcal{F}, \mathbb{P})$. Given a random variable $X : (\Omega, \mathcal{F}, \mathbb{P}) \to (\mathbb{R}, \mathcal{B}(\mathbb{R}))$, we may obtain a function $F_X : \mathbb{R} \to [0, 1]$ given by $F_X(x) = \mu_X((-\infty, x]) = \mathbb{P}(X^{-1}((-\infty, x])$.

$F_X(x)$ represents the probability that X is less than or equal to x. μ_X is a finite measure on $\mathbb{R}$, and the the set of intervals $\{(-\infty, x], x \in R\}$ forms a Π-system that generates $\mathcal{B}(\mathbb{R})$ (the Borel sigma algebra). Furthermore, $\mathbb{R}$ is an element of this Π-system, and an application of Dynkin's Lemma shows that μ_X is uniquely characterised by F_X (exercise in Ouwehand [27]). Thus we may use the distribution function to find the probability of any X-measurable event. For example, given $a \leq b$ we have:

$$\Pr(X \in (a, b]) = \Pr(X \in (-\infty, b] \setminus (-\infty, a]) = F(b) - F(a)$$

Since the probability of this event is non-negative, we see that F is increasing.

We define a univariate distribution functions as follows:

Definition 3.1. *A function $F : \overline{\mathbb{R}} \to [0, 1]$ is a (univariate) distribution function iff*

1. $F(-\infty) = 0$
2. $F(\infty) = 1$
3. *F is monotone increasing*

To characterise the distribution of a random variable, we need only give it's distribution function. We provide a simple example of the standard uniform distribution:

Example 3.2. *A random variable X is said to have a uniform distribution on $[0, 1]$ ($X \sim \mathcal{U}(0, 1)$) if it has a distribution function given by:*

$$F_X(x) = \begin{cases} 0 & if \quad x \leq 0 \\ x & if \quad 0 < x \leq 1 \\ 1 & if \quad 1 < x \end{cases}$$

A plot of this distribution function is given in Figure 1. This distribution function is of particular importance in the study of copulas, as we shall see later.

The uniform random variable is a an example of a *continuous* random variable:

Definition 3.3. *We say that a random variable X is continuous if and only if the associated distribution function F_X is continuous (under the usual Euclidian norm).*

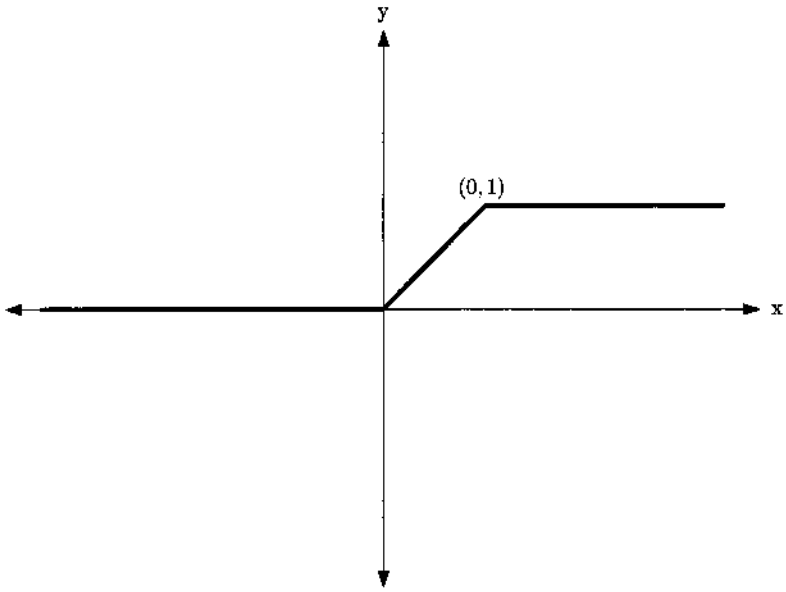

Figure 1: Distribution function for the random variable $X \sim \mathcal{U}(0,1)$

Equivalently, the distribution function F_X has range $[0,1]$ i.e. $F_X(\overline{\mathbb{R}}) = [0,1]$. Since $F_X : \overline{\mathbb{R}} \to [0,1]$ is surjective, it has a right inverse F^{-1} such that $F(F^{-1}(u)) = u$ for all $u \in \mathbb{R}$. Note that for a continuous random variable X, it's associated measure μ_X has no atoms so that sets differing by a countable number of points have the same probability. In particular, for $a \leq b$ we have:

$$\Pr(X \in [a,b]) = \Pr(X \in (a,b]) = \Pr(X \in [a,b)) = \Pr(X \in (a,b))$$

As used in the intensity models section, if μ_X is absolutely continuous with respect to the usual Lebesgue measure λ on $\mathbb{R}$, i.e. $\mu_X \ll \lambda$ it has a *probability density function* (the Radon-Nikodym derivatiave) $f_X = \frac{d\mu_X}{d\lambda}$ such that

$$\Pr(X \in A) = \mu_X(A) = \int_A f_X d\lambda$$

Which will typically admit evaluation (not necessarily in closed form) by Riemann integration:

$$\Pr(X \in A) = \int_A f_X(x)dx$$

3.4 Some useful results

We now define some useful properties of functions on $\mathbb{R}$. Since we will be dealing with functions with similar domain and range, we define this class of functions for brevity.

Definition 3.4. *Let $\emptyset \neq (S_i) \subseteq \overline{\mathbb{R}}$ for $1 \leq i \leq n$. A function*

$$H : \prod_{i=1}^{n} S_i \to \overline{\mathbb{R}}$$

is called an n-place real function.

Definition 3.5. *Suppose H is an n-place real function, and that $B = \prod_{i=1}^{n}[x_i, y_i]$ is an n-box in $\overline{\mathbb{R}}^n$ such that $x_i, y_i \in S_i$ and $x_i \leq y_i$. This n-box has 2^n vertices, where each vertex $\mathbf{v}_j$ is an n-vector such that each component is either an x_i or a y_i. Let $x(\mathbf{v_j})$ denote the number of x_i's appearing in $\mathbf{v_j}$. The H-volume of B is defined as:*

$$V_H(B) = \sum_{j=1}^{2^n} H(\mathbf{v_j})(-1)^{x(\mathbf{v_j})}$$

If we consider the x_i's as left-hand endpoints, then the H-volume is obtained by adding H at the vertices with an even number of left-hand endpoints, and subtracting H at those with an odd number of left-hand endpoints. Later we shall see the importance of the H-volume to distribution functions.

Definition 3.6. *An n-place real function H is n-increasing if and only if $V_H(B) \geq 0$ for all n-boxes B with vertices in $\prod_{i=1}^{n} S_i$.*

In the simplest case where $n = 1$, we have a function $H : S(\subseteq \overline{\mathbb{R}}) \to \overline{\mathbb{R}}$. The H-volume of $[x, y]$ is simply $V_H([x, y]) = H(y) - H(x)$. So having $V_H([x, y]) > 0$ is equivalent to saying the function H is increasing.

When $n = 2$, the H-volume of $[x_1, y_1] \times [x_2, y_2]$ is given by:

$$V_H([x_1, y_1] \times [x_2, y_2]) = H(x_2, y_2) - H(x_1, y_2) - H(x_2, y_1) + H(x_1, y_1)$$

In general, H may be n-increasing, and yet not increasing in each argument, and vice versa (except for the case $n = 1$). However, with an additional property we may obtain a useful result.

Definition 3.7. *Suppose H is an n-place real function such that each S_i has a least element a_i. We say H is grounded if and only if $H(\mathbf{x}_{-j}, a_j) = 0$ for all $\mathbf{x} \in \prod_{i=1}^{n} S_i$ and $1 \leq j \leq n$.*

With the grounded property, we find that the n-increasing property implies the function is increasing in each argument:

Lemma 3.8. *Let H be a grounded, n-increasing function. Then H is increasing in each argument – i.e. $H(\mathbf{x}_{-j}, t)$ is increasing in t for all (fixed) $\mathbf{x} \in \prod_{i=1}^{n} S_i$ and $t \in S_j$.*

Proof. Suppose $t_1 \leq t_2$, both in S_j. We have:

$$\begin{aligned} H(\mathbf{x}_{-j}, t_2) - H(\mathbf{x}_{-j}, t_1) &= V_H([a_1, x_1] \ldots \times [a_{j-1}, x_{j-1}] \times [t_1, t_2] \\ &\quad \times [a_{j+1}, x_{j+1}] \times \ldots \times [a_n, x_n]) \\ &\geq 0 \end{aligned}$$

since H is zero at each vertex of the n-box above except at $(\mathbf{x}_{-j}, t_1)$ and $(\mathbf{x}_{-j}, t_2)$. □

When we deal with a vector of random variables, we may want to consider the distribution of a few random variables in isolation. This is possible using the notion of margins:

Definition 3.9. *Let H be an n-increasing function, and suppose each S_i has a maximum element b_i – then we say H has margins. We may obtain a k-dimensional margin by fixing $\mathbf{x}_i = b_j$ for $n - k$ distinct values of i and considering H as a k-place real function.*

For each $1 \leq k \leq n$ we have $\binom{n}{k}$ margins. Setting $k = n$ corresponds to the trivial case of H itself, while setting $k = 1$ yields the 1-dimensional margins $H_1, H_2, \ldots, H_n$ which are:

$$H_i(x) = H(\mathbf{b}_{-i}, x)$$

Where $\mathbf{b} = (b_1, b_2, \ldots, b_n)$.

Margins provide an 'upper bound' on the discontinuity of grounded increasing functions:

Lemma 3.10. *Let H be a grounded, n-increasing function with margins, and let $\mathbf{x}, \mathbf{y} \in \prod_{i=1}^{n} S_i$. Then:*

$$|H(\mathbf{x}) - H(\mathbf{y})| \leq \sum_{i=1}^{n} |H_i(x_i) - H_i(y_i)|$$

Proof. See Schweizer and Sklar [29] □

Since the number of margins is finite, if they are all continuous it follows that H will also be continuous. So if we have a vector of continuous random variables, their joint distribution function will be continuous.

We now formalise distribution functions on re.

3.5 Distribution functions on $\overline{\mathbb{R}}^n$

A univariate distribution function on $\overline{\mathbb{R}}$ can completely characterise one random variable. Similarly, joint distribution functions can completely characterise a vector of random variables.

Given an n-vector of random variables $\mathbf{X} = (X_1, X_2, \ldots, X_n)$, each defined on the same probability space: $X_i : (\Omega, \mathcal{F}, \mathbb{P}) \to (\mathbb{R}, \mathcal{B}(\mathbb{R}))$ for $1 \leq i \leq n$, we are interested in the joint distribution of the X_i's. We may form a joint distribution function $H : \overline{\mathbb{R}}^n \to [0,1]$ such that

$$H(\mathbf{x}) = \mu_{\mathbf{X}}(\prod_{i=1}^{n}(-\infty, x_i]) = \Pr(X_1 \leq x_1, X_2 \leq x_2, \ldots, X_n \leq x_n)$$

As in the case with 1-dimensional distribution functions, we will want to determine the probabilities for arbitrary events. Again we note that sets of the form $\prod_{i=1}^{n}(-\infty, x_i]$ form a Π system that generates $\mathcal{B}(\mathbb{R}^n)$, so that H completely characterises the distribution of $\mathbf{X}$.

We require non-negative values for probabilities of the form $\Pr(\mathbf{X} \in \prod_{i=1}^{n}(x_i, y_i])$ where $x_i \leq y_i$. This probability is exactly the H-volume of the n-box $\prod_{i=1}^{n}[x_i, y_i]$). (Recall we are not distinguishing between sets that differ by a finite (or countable) number of points)

Definition 3.11. *An n-place real function H is a joint distribution function if and only if*

1. $H(\mathbf{x}_{-i}, -\infty) = 0$ *for all* $\mathbf{x} \in \prod_{i=1}^{n} S_i$ *and* $1 \leq i \leq n$
2. $H(\mathbf{1} \cdot \infty) = 1$
3. *H is n-increasing*

We are now ready to introduce copulas.

3.6 Copulas

Definition 3.12. *We say an n-place real function C' is an n-subcopula if and only if*

1. $C' : \prod_{i=1}^{n} S_i \to \mathbb{R}$ *where each* S_i *is a subset of* $I = [0, 1]$.
2. C' *is grounded and n-increasing*
3. *For each 1-dimensional margin* C'_i *, we have* $C'_i(u) = u$ *for all* $u \in S_i$.

By Lemma 3.8, C' is increasing in each argument. Since C' is grounded and each margin is bounded above by 1, it follows that $0 \leq C'(\mathbf{u}) \leq 1$ for all $\mathbf{u} \in \prod_{i=1}^{n} S_i$.

If we restrict ourselves to those subcopulas that are defined on the whole n-hypercube we obtain copulas:

Definition 3.13. *An n-copula is an n-subcopula with domain the entire n-hypercube* I^n.

In fact, each copula is the joint distribution function of a set of uniform random variables on $[0, 1]$ restricted to the unit n-hypercube. Each 1-dimensional margin of a copula is exactly the distribution function in Figure 1 restricted to $[0, 1]$. As it's name suggests, it joins together these random variables to form a joint distribution. A copula may be extended to a distribution function H by setting $H(\mathbf{x}) = C(\min(\max(x_i, 0), 1))$ for $i = 1, 2, \ldots, n$. (We are mapping x_i's greater than 1 to 1, and all x_i's less than 0 to 0).

Using the uniform margins of a (sub)copula we have the following result from Lemma 3.10.

Lemma 3.14. *Let C' be an n-subcopula and* $\mathbf{u}, \mathbf{v}$ *be points in the n-cube. Then:*

$$|C'(\mathbf{u}) - C'(\mathbf{v})| \leq \sum_{i=1}^{n} |u_i - v_i|$$

Since

$$\sum_{i=1}^{n} |u_i - v_i| \leq n \cdot ||\mathbf{u} - \mathbf{v}||$$

where $||.||$ is the usual Euclidian norm, it follows that n is a Lipschitz constant for C'. Hence C' is Lipschitz continuous, and thus absolutely continuous. It follows that C' is differentiable almost everywhere. In this discussion we shall be dealing exclusively with differentiable copulas except when we use the Fréchet-Hoeffding bounds. A differentiable n-copula C(u) admits a density so that:

$$C(\mathbf{v}) = \int_0^{v_n} \cdots \int_0^{v_2} \int_0^{v_1} \frac{\partial^n C}{\partial u_1 \partial u_2 \ldots \partial u_n} du_1 du_2 \ldots du_n$$

The next result provides a lower and upper bound for subcopulas:

Theorem 3.15.

$$W(\mathbf{u}) \quad \leq C'(\mathbf{u}) \quad \leq M(\mathbf{u})$$

Where:

$$W(\mathbf{u}) = \max(\sum_{i=1}^{n} u_i - n + 1, 0)$$

and

$$M(\mathbf{u}) = \min(u_1, u_2, \ldots, u_n)$$

Proof. C' is increasing in each argument, so for each j

$$\begin{aligned} C'(\mathbf{u}) &\leq C'(\mathbf{1}_{-j}, u_j) \\ &= u_j \end{aligned}$$

whence the right hand side of the inequality. For the left, we note that first that trivially $C'(\mathbf{u}) \geq 0$. Now suppose C' is defined at $\mathbf{1}$ so that $C'(\mathbf{1}) = 1$ (otherwise define it as such), then by Lemma 3.14

$$\begin{aligned} |C'(\mathbf{1}) - C'(\mathbf{u})| &\leq \sum_{i=1}^{n} |1 - u_i| \\ 1 - C'(\mathbf{u}) &\leq \sum_{i=1}^{n} (1 - u_i) \\ \sum_{i=1}^{n} u_i - n + 1 &\leq C'(\mathbf{u}) \end{aligned}$$

□

The bounds on C' in Theorem 3.15 are called the *Fréchet-Hoeffding* lowerbound and *Fréchet-Hoeffding* upperbound, and are denoted W(u) and M(u) respectively. We provide illustations of these surfaces for the 2-copula in figures 2 and 3.

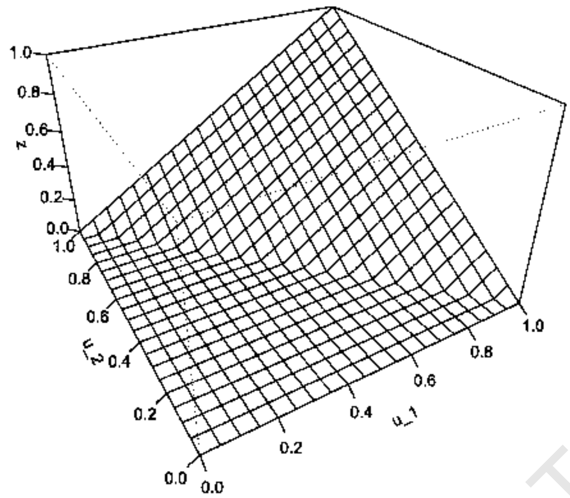

Figure 2: The Frechet-Hoeffding lower bound z = W(ui, u2)

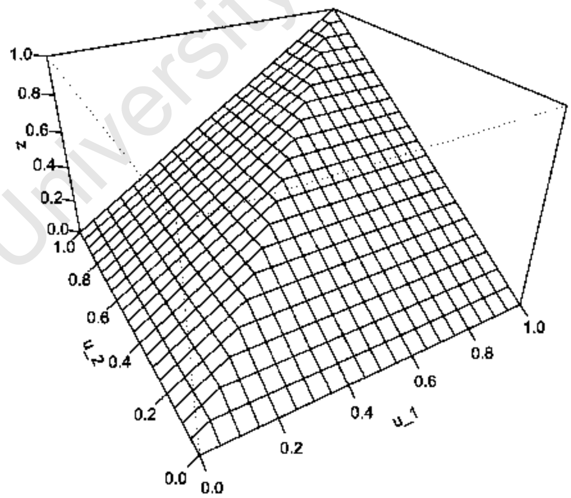

Figure 3: The Frechet-Hoeffding upper bound z = M(ui, u2)

For the arbitrary n-dimensional case, M has support on the diagonal 1 $.t$ (0 < t < 1) which corresponds to perfect positive dependence between random variables. In the two-dimensional case, W has support on the diagonal $(t, 1-t)$, 0 < t < 1 which corresponds to perfect negative dependence. We note that negative dependence cannot be arbitrarily extended to higher dimensions due to alternating signs. M is a copula for all n, while W is only a copula for n = 2. However, for n > 2, for each **U** in the n-hypercube there exists a an n-copula C such that W(u) = C(u) [26].

Another important copula is the product copula II(u) = Fr 1 ui which corresponds to independence of random variables [26] — see figure 4.

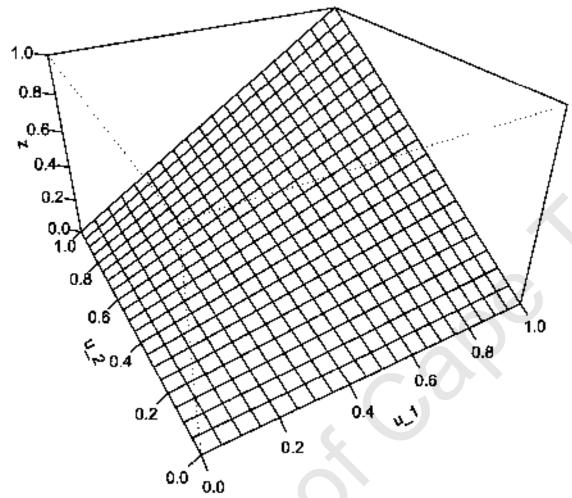

Figure 4: The product copula z = II(ui, u2)

3.7 Sklar's Theorem

Sklar's Theorem is a central result in the study of copulas. It demonstrates a correspondence between copulas and joint distribution functions.

Theorem 3.16. (Sklar)

1. *Suppose we have n 1-dimensional distribution functions F_i, $1 \leq i \leq n$, a vector $\mathbf{x} \in \overline{\mathbb{R}}^n$, and an n-subcopula C' defined on $A \supseteq \prod_{i=1}^n F_i(\overline{\mathbb{R}})$. Then the function $H : \overline{\mathbb{R}}^n \to [0,1]$ defined by*

$$H(\mathbf{x}) = C'(F_1(x_1), F_2(x_2), \ldots, F_n(x_n))$$

is a joint distribution function on lin *with 1-dimensional margins F1,* F2 , F.

2. *Conversely, suppose that H is an n-dimensional distribution function with margins F1,* F2 , **Fn** . *Then there exists a unique subcopula C' : fl Fi* (R) -4 [0, 1] *such that*

$$H(\mathbf{x}) = C'(F_1(x_1), F_2(x_2), \ldots, F_n(x_n))$$

If all the Fi's are continuous, then C' is a copula. Otherwise, C' is a subcopula that can be extended to a copula C such that C (u) = C' (u) for all **u** *in 117_1* Fi(R).

Proof. See Moore & Spruill [25]

The first part of Sklar's theorem allows us to create joint distribution functions given an arbitrary set of n continuous random variables, each with its own marginal distribution: Given a continuous random variable X, we use the *probability integral transform* to set Y = *Fx (X).* Then we have Y takes on values in [0, **1]**, and for each right inverse Fj-c-1 of *Fx:*

$$\begin{aligned}
\Pr(Y \le u) &= \Pr(F_X(X) \le u) \\
&= \Pr(X \le F_X^{-1}(u)) \\
&= F(F^{-1}(u)) \\
&= u
\end{aligned}$$

Which means Y U(0, **1).** So we apply the respective distribution function to each random variable, and we join the resulting U(0, 1) random variables using a copula. In this way we can couple any combination of continuous random variables, be they normal, exponential, log-normal etc. This will prove useful in the Monte-Carlo section, where we need to produce draws from a joint distribution function.

The second part of Sklar's theorem shows that given a joint distribution function, we know that we can always decompose it into the individual margins and a dependence structure as given by the associated copula. The copula is independent of the margins, so that we may think of a joint distribution as being made up of two parts: the dependence structure (the copula), and the univariate margins.

3.8 Examples

We have already mentioned some important copulas, such as the Frechet-Hoeffding upperbound M, and the lowerbound W (only a copula in the 2-dimensional case), as well as the product copula **H.** There are also many copula families that are used extensively. These families are usually indexed by one more parameters. Here we list some important copula families that are widely used.

Perhaps the most recognisable copula in finance is the *Guassian copula* — one of the early proponents of its application to modelling default dependence was Li [19].

Example 3.17. *Let $\Phi(x)$ represent the standard univariate normal distribution, and $\Phi^{\Sigma}(\mathbf{x})$ represent the n-dimensional multivariate normal distribution with standard normal margins (mean $\boldsymbol{\mu} = \mathbf{0}$) and (symmetric positive definite) covariance matrix Σ ($\sigma_{ii} = 1$).*

The Gaussian copula is given by:

$$C^{\mathrm{Ga}}(\mathbf{u}) = \Phi^{\Sigma}(\Phi^{-1}(u_1), \Phi^{-1}(u_2), \ldots, \Phi^{-1}(u_n))$$

We are free to specify the off-diagonal entries of the symmetric matrix Σ that represent the correlations (the same as the covariances since each u_i is distributed as $\mathcal{N}(0,1)$). So for an n-dimensional Gaussian copula we are free to adjust $\frac{n^2-n}{2}$ parameters.

We give plots of the density of the 2-Gaussian copula for correlations of $\rho = 0.6$ and $\rho = -0.6$ in figures 5 and 6. Note the support for positive correlation is centered around the diagonal (t,t), while the the support for negative correlation is centered around the diagonal $(t, 1-t)$. This behaviour becomes more prominent as $|\rho|$ approaches 1.

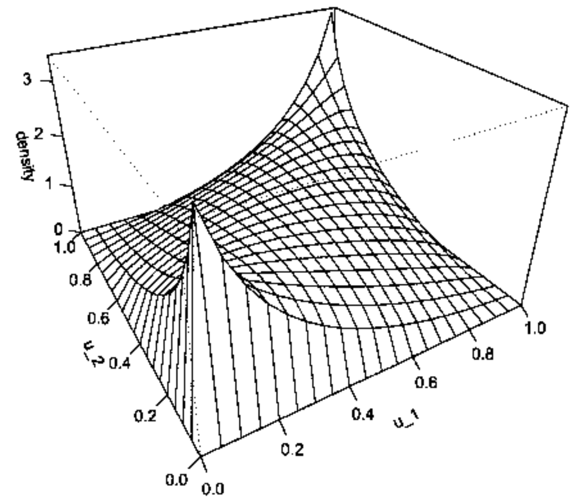

Figure 5: Density for the 2-Gaussian copula with $p = 0.6$

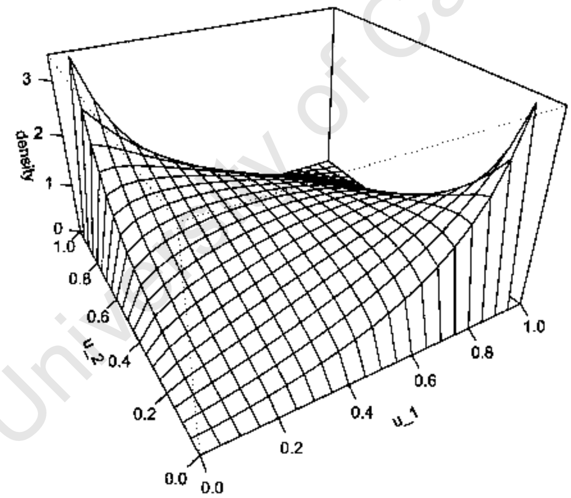

Figure 6: Density for the 2-Gaussian copula with $p = -0.6$

Closely related to the Gaussian copula is the t copula, based on the Student-t distribution.

Example 3.18. *Let $t(x)$ represent the standard univariate t distribution with v* *degrees of freedom, and $t_\nu^\Sigma(\mathbf{x})$ represent the multivariate t distribution with mean $\mu = \mathbf{0}$ and with Σ as in the Guassian example.*

The student t copula is given by:

$$C^t(\mathbf{u}) = t_\nu^\Sigma(t_\nu^{-1}(u_1), t_\nu{-}1(u_2), \ldots, t_\nu^{-1}(u_n))$$

We see Gaussian and t copulas are created using the respective multivariate distribution functions. There is a useful symmetry property here that we mention.

Definition 3.19. *A vector of random variables* $\mathbf{X}$ *is said to be radially symmetric about the point* $\mathbf{a}$ *if and only if the distribution of* $\mathbf{X}-\mathbf{a}$ *is the same as the distribution of* $\mathbf{a} - \mathbf{X}$.

We consider the multivariate normal distribution centered at $\boldsymbol{\mu}$. It has density function:

$$\phi_{\boldsymbol{\mu}}^\Sigma(\mathbf{x}) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} e^{-(\mathbf{x}-\boldsymbol{\mu})'\frac{1}{2}\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})}$$

so that

$$\Pr(\mathbf{X} \in A) = \int_A \phi_{\boldsymbol{\mu}}^\Sigma(\mathbf{x}) dx_1 dx_2 \ldots dx_n$$

The distribution function is:

$$\Phi_{\boldsymbol{\mu}}^\Sigma(\mathbf{x}) = \int_{-\infty}^{x_n} \cdots \int_{-\infty}^{x_2} \int_{-\infty}^{x_1} \phi_{\boldsymbol{\mu}}^\Sigma(\mathbf{s}) ds_1 ds_2 \ldots ds_n$$

Also note that $\phi_{\boldsymbol{\mu}}^\Sigma(\mathbf{x}+\boldsymbol{\mu}) = \phi_{\boldsymbol{\mu}}^\Sigma(-(\mathbf{x}+\boldsymbol{\mu}))$.

$$
\begin{aligned}
\Pr(\mathbf{X}-\boldsymbol{\mu} \leq \mathbf{x}) &= \Pr(\mathbf{X} \leq \mathbf{x}+\boldsymbol{\mu}) \\
&= \int_{-\infty}^{x_n+\mu_n} \cdots \int_{-\infty}^{x_2+\mu_2} \int_{-\infty}^{x_1+\mu_1} \phi_{\boldsymbol{\mu}}^{\Sigma}(\mathbf{t}) dt_1 dt_2 \ldots dt_n \\
&= \int_{-\infty}^{x_n} \cdots \int_{-\infty}^{x_2} \int_{-\infty}^{x_1} \phi_{\boldsymbol{\mu}}^{\Sigma}(\mathbf{t}+\boldsymbol{\mu}) dt_1 dt_2 \ldots dt_n \\
&= (-1) \int_{-x_n}^{\infty} \cdots (-1) \int_{-x_2}^{\infty} (-1) \int_{-x_1}^{\infty} \phi_{\boldsymbol{\mu}}^{\Sigma}(-(\mathbf{t}+\boldsymbol{\mu}))(-1)^n dt_1 dt_2 \ldots dt_n \\
&= \int_{-x_n}^{\infty} \cdots \int_{-x_2}^{\infty} \int_{-x_1}^{\infty} \phi_{\boldsymbol{\mu}}^{\Sigma}(\mathbf{t}+\boldsymbol{\mu}) dt_1 dt_2 \ldots dt_n \\
&= \int_{-x_n+\mu_n}^{\infty} \cdots \int_{-x_2+\mu_2}^{\infty} \int_{-x_1+\mu_2}^{\infty} \phi_{\boldsymbol{\mu}}^{\Sigma}(\mathbf{t}) dt_1 dt_2 \ldots dt_n \\
&= \Pr(\mathbf{X} \geq -\mathbf{x}+\boldsymbol{\mu}) \\
&= \Pr(\boldsymbol{\mu}-\mathbf{X} \leq \mathbf{x})
\end{aligned}
$$

so that $\mathbf{X}$ is radially symmetric about $\boldsymbol{\mu}$. The Gaussian copula density is given by

$$
\begin{aligned}
c^{\text{Ga}} &= \frac{\partial^n C^{\text{Ga}}}{\partial u_1 \partial u_2 \ldots \partial u_n} \\
&= \frac{\partial^n}{\partial u_1 \partial u_2 \ldots \partial u_n}\left[\Phi^{\Sigma}(\Phi^{-1}(u_1), \Phi^{-1}(u_2), \ldots \Phi^{-1}(u_n))\right]
\end{aligned}
$$

If we let $x_i = \Phi^{-1}(u_1)$ then since each x_i depends only on u_i we have:

$$
\begin{aligned}
c^{\text{Ga}} &= \frac{\partial^n}{\partial x_1 \partial x_2 \ldots \partial x_n}\left[\Phi^{\Sigma}(x_1, x_2, \ldots, x_n)\right] \prod_{i=1}^{n} \frac{\partial x_i}{\partial u_i} \\
&= \phi^{\Sigma}(x_1, x_2, \ldots, x_n) \prod_{i=1}^{n} \frac{1}{\frac{\partial u_i}{\partial x_i}} \\
&= \phi^{\Sigma}(x_1, x_2, \ldots, x_n) \prod_{i=1}^{n} \frac{1}{\phi(x_i)} \\
&= \frac{1}{|\Sigma|^{1/2}} e^{-\mathbf{x}\frac{1}{2}(\Sigma^{-1}-I)\mathbf{x}}
\end{aligned}
$$

Which is even in $\mathbf{x}$. Also, $\Phi^{-1}(1-u) = -\Phi^{-1}(u)$. So $c^{\text{Ga}}(\mathbf{1}-\mathbf{u}) = c^{\text{Ga}}(\mathbf{u})$ for $\mathbf{u} \in [0,1]$. Using the same integration argument as before, we can show

$$\mathbf{u} \sim \mathbf{1} - \mathbf{u} \tag{10}$$

Or, $\mathbf{u} - \frac{1}{2} \sim \frac{1}{2} - \mathbf{u}$. So the Gaussian copula is radially symmetric about $\frac{1}{2}$. A similar argument holds for the t copula. We will use (10) when implementing our Monte Carlo simulation.

We provide a level curve plot for the density of the 2-Gaussian and 2-t copulas copula with correlation $\rho = 0.5$. Note the symmetry of the contours around the point $\frac{1}{2}$.

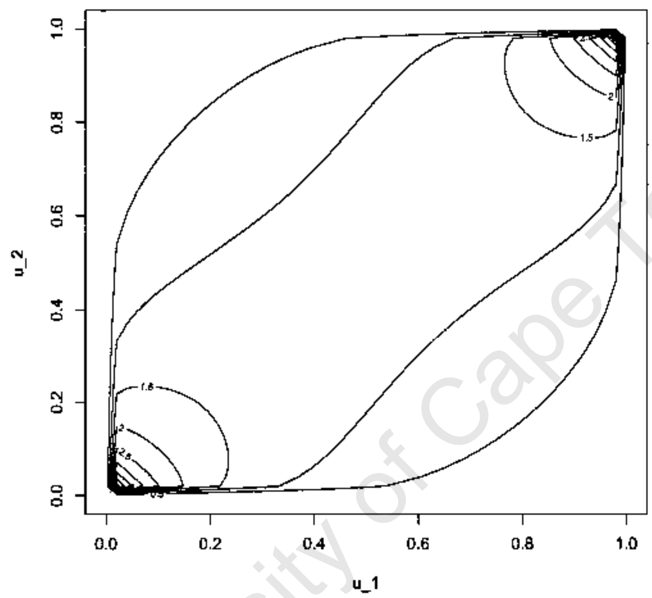

Figure 7: Level curves for the 2-Gaussian copula density with $p = 0.5$

The multivariate normal and t distributions are examples of a *elliptical distributions.* In the n-dimensional case, the level surfaces of the multvariate densities are hyper-ellipses in **Rn.** Copulas constructed from elliptical distributions, such as the Gaussian and t copulas, are called *elliptical copulas.* See Embrechts et al [14] for more details.

Finally we mention an important class of copulas — the so-called *Archimedean copulas.* Before discussing these copulas we introduce some necessary concepts.

Definition 3.20. *A function f :R R is said to be completely monotonic if*

$$(-1)^k \frac{d^k f}{dx^k} \geq 0$$

for $k = 1, 2, 3, \ldots$

Theorem 3.21. *Let $\phi : [0,1] \to [0,\infty]$ be a continuous, surjective, strictly decreasing function such that $\phi(1) = 0$. Then it has an inverse $\phi^{-1} : [0,\infty] \to [0,1]$, and we define an n-place real function C with domain the n-hypercube by:*

$$C(\mathbf{x}) = \phi^{-1}(\phi(x_1) + \phi(x_2) + \ldots + \phi(x_n))$$

C is an n-copula for all $n \geq 2$ if and only if ϕ^{-1} is completely monotonic on $[0,\infty)$.

Proof. See Schweizer & Sklar [29] □

So for each appropriate generator ϕ we may obtain an n-copula.

Often the generator is given in terms of a parameter θ, so that $\phi = \phi(x;\theta)$. This gives rise to a family of copulas. We give examples of some common families here:

Example 3.22. *The Clayton n-copula has generator*

$$\phi(u) = u^{-\theta} - 1$$

yielding

$$\phi^{-1}(x) = (x+1)^{-1/\theta}$$

with $\theta > 0$.

The final form is given by:

$$C(\mathbf{u}) = \left[\sum_{i=1}^{n} u_i^{-\theta} - n + 1\right]^{-1/\theta}$$

Example 3.23. *The Frank family has generator*

$$\phi(u) = \log\left(\frac{e^{-\theta u} - 1}{e^{-\theta} - 1}\right)$$

and

$$\phi^{-1}(x) = -\frac{\log(1 + e^{x}(e^{-\theta} - 1))}{\theta}$$

with $\theta > 0$.

The final form is given by:

$$C(\mathbf{u}) = -\frac{1}{\theta}\log\left\{1 + \frac{\prod_{i=1}^{n}(e^{-\theta u_i} - 1)}{(e^{-\theta} - 1)^{n-1}}\right\}$$

Example 3.24. *The Gumbel family has generator*

$$\phi(u) = (-\log(u))^{\theta}$$

yielding

$$\phi^{-1}(x) = e^{x^{1/\theta}}$$

with

$\theta > 1$.

The final form is given by:

$$C(\mathbf{u}) = \exp\left\{-\left[\sum_{i=1}^{n}(-\log(u_i))^{\theta}\right]^{\frac{1}{\theta}}\right\}$$

3.9 Tail dependence

Much work has been done in financial modelling on extreme events. Often the distributions of financial variables have heavy tails, so that extreme events occur relatively frequently. Given two random variables X1 and **X2,** the tail dependence is an indicator of how likely we are to observe an extreme value in the one variable, given that we have already observed an extreme value in the other variable.

We define bivariate tail dependence as follows:

Definition 3.25. *Let (X1, X2) be continuous random variables with distribution functions Fxi and* ***FX2*** *respectively. A bivariate distribution is said to have upper tail depedence if and only if*

$$\lim_{t\to 1^-} \Pr[F_{X_2}(X_2) > t | F_{X_1}(X_1) > t] = \lambda_U > 0$$

and lower tail dependence if and only if

$$\lim_{t\to 0^+} \Pr[F_{X_2}(X_2) \leq t | F_{X_1}(X_1) \leq t] = \lambda_L > 0$$

Where Au is called the *coefficient of upper tail dependence* and AL the *coefficient of lower tail dependence.* We see that bivariate tail dependence is independent of the margins since we are using the probability integral transform of each variable. It follows that bivariate tail dependence is completely determined by the copula of X1 and **X2.** We have the following formula for calculating bivariate tail dependence:

Theorem 3.26. *Let (X_1, X_2) be continuous random variables with distribution functions F_{X_1} and F_{X_2} respectively and associated copula C, and let $\delta_C(t) = C(t,t)$ be the $\mathbf{0} \to \mathbf{1}$ diagonal section of C. Then the coefficients of tail dependence are given by:*

$$\begin{aligned}\lambda_U &= 2 - \lim_{t\to 1^-} \frac{1 - C(t,t)}{1-t} \\ &= 2 - \left.\frac{d\delta_C(t)}{dt}\right|_{1^-}\end{aligned}$$

$$\begin{aligned}\lambda_L &= \lim_{t\to 0^+} \frac{C(t,t)}{t} \\ &= \left.\frac{d\delta_C(t)}{dt}\right|_{0^+}\end{aligned}$$

Where $\left.\frac{d\delta_C(t)}{dt}\right|_{1^-}$ *denotes the left derivative of* δ_C *at* 1 *and* $\left.\frac{d\delta_C(t)}{dt}\right|_{0^+}$ *denotes the right derivative of* δ_C *at 0.*

Proof. See Nelsen [26] □

Example 3.27. *We calculate the lower tail dependence of the 2-Clayton copula. For* $0 < t \ll 1$*:*

$$\begin{aligned}C(t,t) &= (t^{-\theta} + t^{-\theta} - 1)^{-1/\theta} \\ &= (2t^{-\theta} - 1)^{-1/\theta}\end{aligned}$$

And:

$$\begin{aligned}\lambda_L &= \left.\frac{dC}{dt}\right|_{0^+} \\ &= \lim_{t\to 0^+} -\frac{1}{\theta}(2t^{-\theta} - 1)^{-(1+\theta)/\theta}(-2\theta)t^{-(\theta+1)} \\ &= \lim_{t\to 0^+} 2(t^{\theta}(2t^{-\theta} - 1))^{-(1+\theta)/\theta} \\ &= \lim_{t\to 0^+} 2(2 - t^{\theta})^{-(1+\theta)/\theta} \\ &= 2 \cdot 2^{-(1+\theta)/\theta} \\ &= 2^{-1/\theta}\end{aligned}$$

The 2-gaussian copula has no tail dependence (except in the degenerate case when $p = 1)$, while the t copula has both upper and lower tail dependence which is decreasing in the degrees of freedom [14]. Indeed, as we increase the degrees of freedom, the t distribution approaches the normal distribution, so we would expect this to happen. As a practical example, we provide a plot of random draws from the Gaussian and t copulas:

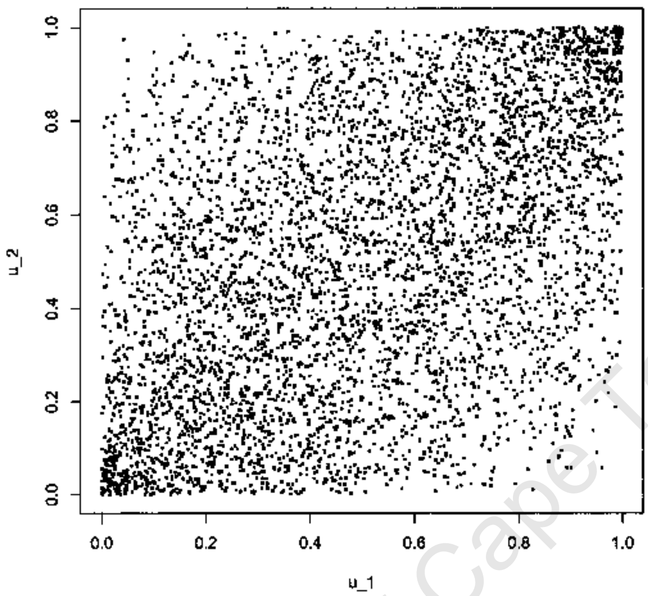

Figure 8: 5000 random draws from the 2-Gaussian copula with $p = 0.5$

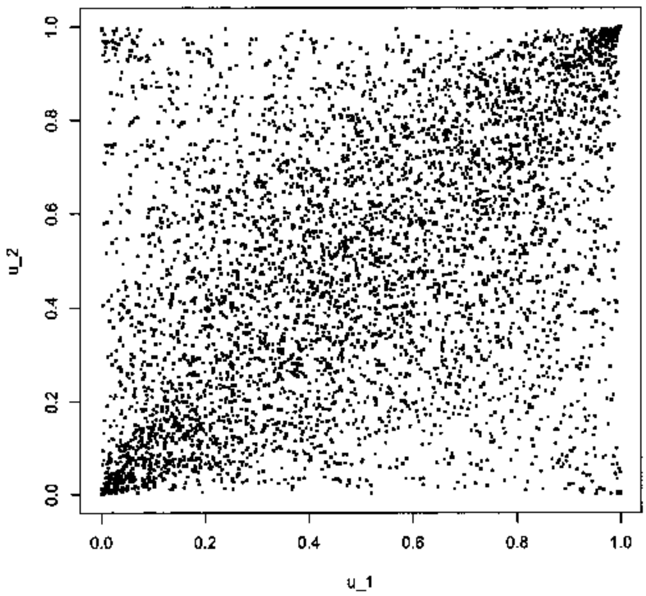

Figure 9: 5000 random draws from the 2-t copula with v = 2 and $p = 0.5$

Note how Figure 9 exhibits clustering near (0, 0) and (1, 1) when compared to Figure 8, indicative of the upper and lower tail dependence of the t copula.

Tail dependence in the 2-dimensional case has been discussed extensively in statistics literature [20]. Suggestions have been made for an extension of tail dependence to the general n-dimensional case. We mention one suggestion given by Li [20]:

Definition 3.28. *Let* $\mathbf{X}$ *be a vector of continuous random variables with marginals* $F_1, F_2, \ldots F_n$ *and copula* C.

1. $\mathbf{X}$ *is said to be upper-orthant tail dependent if for some subset* $\emptyset \neq J \subset 1, 2, \ldots, n$

$$\lim_{u \to 1^-} \Pr(F_j(X_j) > u, \forall j \notin J | F_i(X_i) > u, \forall i \in J) = \lambda_{U_J}^C > 0$$

If for all $\emptyset \neq J \subset \{1, \ldots, n\}, \lambda_{U_J}^C = 0$*, then we say* $\mathbf{X}$ *is upper-orthant tail independent.*

2. $\mathbf{X}$ *is said to be lower-orthant tail dependent if for some subset* $\emptyset \neq J \subset 1, 2, \ldots, n$

$$\lim_{u \to 0^+} \Pr(F_j(X_j) \leq u, \forall j \notin J | F_i(X_i) \leq u, \forall i \in J) = \lambda_{L_J}^C > 0$$

If for all $\emptyset \neq J \subset \{1, \ldots, n\}, \lambda_{L_j}^C = 0$, then we say $\mathbf{X}$ is lower-orthant tail independent.

So we have multivariate orthant tail dependence if we can condition on some non-empty subset $A \subset \{X_1, X_2, \ldots, X_n\}$'s so that the conditional probability of observing extreme values in all of $\mathbf{X} \setminus A$ does not vanish. Since we are taking the probability integral transform of the marginals, we see that this generalisation of tail dependence is also dependent only on the copula.

Since we will not be investigating the theory of multivariate tail dependence, we perform simple numerical test to see how frequently extreme values tend to occur together.

We consider the 50-Gaussian and 50-t copulas with all the elements off the main diagonal of E set to a common value $p = 0.5$. We consider the distribution of the number of the number of extreme scalars observed in each vector draw. We ignore the cases of 0 or 1 extreme observations since these will affect the scale of the plots too much. We take as our definition of extreme event as the probability that one of the components of our vector is less than 0.05. The results are in Figure 10.

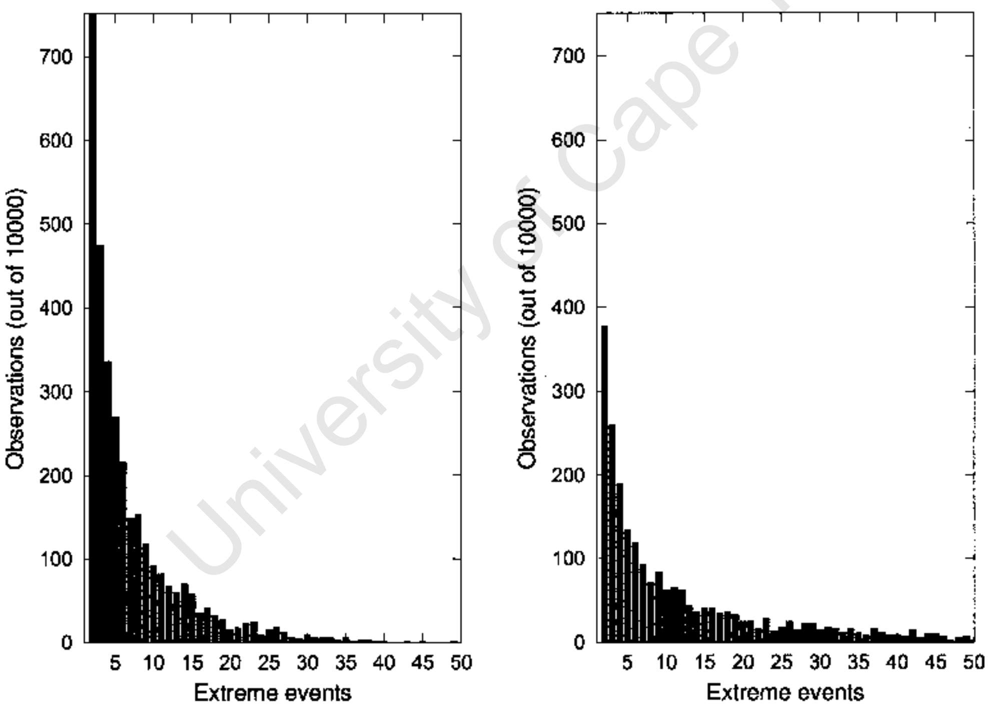

Figure 10: Distribution of number of components less than the 0.05 percentile for the Gaussian (left) and t (right) copulas for $\rho = 0.5$

We see the distribution for the Gaussian copula concentrated on the left with a small tail on the right. In constrast, the distribution for the t copula has a heavy tail.

3.10 Modelling basket credit risk

Now suppose we have a basket credit derivative who's cashflows depend on n reference credits with default times T = (71, 72, • • • 77,) with continuous distribution functions F1, F2, . , Fn. For modeling basket credit risk, we need to model the joint distribution ***HT*** of the default times of the basket components. Sklar's theorem allows us a shortcut if we set:

$$H_\tau = C(F_1(\tau_1), F_2(\tau_2), \ldots, F_n(\tau_n))$$

for any n-dimensional copula C. So the problem of modelling basket credit risk can be divided in two:

1. Model the individual default times independently of one another
2. Create a a joint distribution using a copula

This is exactly the approach we shall take in our model. Note that applying the copula leaves the individual default distributions intact. So that if we have calibrated some single name CDS's, we will not lose this calibration if we join them using a copula.

4 The model

4.1 Setup

We will consider tranches of various seniority in a CDO, where the individual default times (unconditioned on any extra information) are similar. We will investigate the properties of each tranche as we vary the dependence structure as modelled by an appropriate copula.

A strong dependence in the default times may be caused by macroeconomic variables such as commodity prices, interest rates, or housing prices.

4.2 Monte-Carlo approach

To determine the fair rates for the tranches, we use Monte Carlo (MC) simulation. Important papers on analytic and semi-analytic methods have been developed for pricing certain copula models [17]. Compared to these methods, MC simulation is very computationally intensive. However the MC method is very general, so that we may extend a basic implementation to model almost any dependence structure and marginal distributions (assuming we have the appropriate simulation methods).

Furthermore, a full MC implementation provides us with an error estimate for our results. The results of other (non-analytic) faster algorithms with no error estimates are often tested against MC results to see if they are producing accurate estimates.

Using the copula approach, generating random draws from a joint distribution can be performed in two steps. First we draw a point $\mathbf{u}$ in the n-hypercube from our copula – which models the dependence between the random variables $(X_1, X_2, \ldots, X_n)$ independently of the their margins. Then we transform each percentile u_i to obtain a draw from the marginal distribution of X_i.

4.3 Generating draws from the copula

Since MC simulation is computationally intensive, we would like efficient techniques to produce draws from our multivariate distribution.

We saw in Section 3.8 that the Gaussian and t copulas are constructed using their respective multivariate univariate normal margins. Generating draws from the elliptical copulas mentioned in Section 3.8 is straightforward, the following methods (and for other copulas) may be found in Cherubini et al [11]. For the Gaussian copula with correlation matrix Σ:

- Generate a column vector $\mathbf{u}$ of n independent draws $(u_1, u_2, \ldots, u_n)$ from the standard normal distribution
- Find the Cholesky factorization L of the symmetric and positive definite matrix Σ so that $LL^T = \Sigma$
- Apply the one dimensional standard normal distribution to each element of $L\mathbf{u}$ to obtain the percentiles

For the t copula with ν degrees of freedom:

- Generate a column vector $\mathbf{u}$ of n independent draws $(u_1, u_2, \ldots, u_n)$ from the standard normal distribution
- Find the Cholesky factorization L of the symmetric and positive definite matrix Σ so that $LL^T = \Sigma$
- Generate a random draw s from the χ^2_ν distribution
- Apply the one dimensional standard t_ν distribution to each element of $\sqrt{\frac{\nu}{s}}L\mathbf{u}$ to obtain the percentiles

Once determined for a particular dependence structure, the Cholesky decomposition $\boldsymbol{L}$ remains the same for all the simulations, thus it makes sense to determine $\boldsymbol{L}$ outside of the main MC loop. This may be achieved in Octave by making the Cholesky decomposition a *persistent* variable in the corresponding function, and resetting the variable after each new set of MC parameters. In Section 3.8 we mentioned that for the Gaussian and t copulas, U and 1 — U are identically distributed. Thus for each draw u we obtain from these copulas, we immediately have another draw 1 — u.

There are more general methods for inverting copulas, such as the conditional distribution approach. Generally, the more general the method the slower it is, and for MC simulation we need efficient methods. One efficient method for generating random draws from particular multivariate distributions based on Laplace transforms was developed by Marshall *Sz* Olkin [22].

4.4 Obtaining the default times from copula percentiles

The next step is to invert the percentiles obtained in Section 4.3 to obtain the default times. If the intensity process has constant intensity $\lambda > 0$, then a standard exponential inverse function may be applied. If $\lambda(t)$ is piecewise constant, the cumulated intensity $\Lambda(t)$ will be piecewise linear. Once we have determined the endpoints of each linear section and stored them in a vector, we can use simple linear interpolation to find $\Lambda^{-1}(t)$. As the complexity of $\Lambda(t)$ increases, so the difficulty of inverting it increases.

4.5 Reliability of estimates

MC simulation can also provide us with an error estimate. We are looking to find the mean of some payoff function $f(\mathbf{X})$ by generating draws from the random variable. Each simulation we run yields a draw $\mathbf{X}_i$ and an associated estimate $f(\mathbf{X}_i)$, and we estimate $\mathbb{E}[f(\mathbf{X})]$ as:

$$\begin{aligned}\mathbb{E}[f(\mathbf{X})] &\approx \frac{1}{n}\sum_{i=1}^{n} f(\mathbf{X}_i)\\ &= \mu_n\end{aligned}$$

By the law of large numbers, the expression on the right converges to the true mean of $f(\mathbf{X})$. An unbiased estimate for the sample variance is:

$$\begin{aligned} s^2(n) &= \frac{1}{n-1}\sum_{i=1}^{n}(f(\mathbf{X}_i)-\mu_n)^2 \\ &= \frac{1}{n-1}\left[\sum_{i=1}^{n} f(\mathbf{X}_i)^2 - \frac{1}{n}\left(\sum_{i=1}^{n} f(\mathbf{X}_i)\right)^2\right] \end{aligned}$$

Thus, if for each iteration of our MC simulation we store the running totals of $\sum f(\mathbf{X}_i)$ and $\sum f(\mathbf{X}_i)^2$, we can calculate the sample variance. Under suitable conditions, $\mu(n)$ is approximately normally distributed. The standard error of the mean is $s(n)/\sqrt{N}$ and can be used to estimate σ (the true variance) [4]. Thus we can find confidence intervals for our estimate.

For the Gaussian and t copulas, we can using the draws $\mathbf{u}$ and $\mathbf{1}-\mathbf{u}$ which admit antithetic variance reduction:

When we invert these draws to obtain default time vectors $\boldsymbol{\tau}^+$ and $\boldsymbol{\tau}^-$, the (τ_i^+, τ_i^-) pairs will have negative covariance since they are monotonic transformations of the copula percentiles (assuming $\lambda(t) > 0$). Furthermore, the present values of the payment and protection legs in a synthetic CDO are monotonic in τ. Thus the two estimates we obtain will have negative covariance, and:

$$\mathrm{var}\left(\frac{f(\tau^+)+f(\tau^-)}{2}\right) = \frac{\mathrm{var}(f(\tau^+)) + \mathrm{var}(f(\tau^-))}{4} + \frac{1}{2}\mathrm{cov}(f(\tau^+), f(\tau^-))$$

Since $\mathrm{cov}(f(\tau^+), f(\tau^-)) < 0$, $\mathrm{var}\left(\frac{f(\tau^+)+f(\tau^-)}{2}\right) < \frac{1}{2}\mathrm{var}(f(\tau^+))$.

4.6 Program specification

A MC CDO model was implemented in Octave, an open source interpreted language that supports most Matlab functions. Usually, vector implementations run faster than explicit *for* and *while* loops. However, for large MC simulations, it becomes impractical to store all the scenarios in one array due to memory constraints.

The MC implementation allows the user to specify:

- The number of obligors
- A vector of notionals
- A vector of recovery rates
- A function for producing the n-dimensional percentile draws

- A vector function for inverting the percentile to obtain default times
- A discount factor function
- A vector of tranche attachment points
- Maturity of the contract
- The number of payments per year

On completion the function returns

- The mean discounted loss per tranche and standard error
- The mean fair rates and standard error

5 Results

In our setup, the following parameters remain constant for each model tested:

- The term structure is assumed to be a flat 3%
- The maturity of the contract was fixed at 5 years
- We have 50 obligors each with a constant hazard rate of 2% and recovery rate 50%

The dependence was modelled using 50-Gaussian copulas and 50-t copulas with 2 degrees of freedom. The correlation coefficient was varied and the results recorded.

The most notable difference between figures 11 and 12 is the expected loss on the most junior tranche (0-3%) for small correlations. We see that at $p = 0$ *it* is approximately 0.9 in Figure 11, and 0.55 in Figure 12. From the fixed intensity, we have for each default time:

$$\begin{aligned}\Pr(\tau \leq 5) &= 1 - e^{-(0.02)5} \\ &\approx 0.095\end{aligned}$$

Thus, default during the contract is approaching an extreme event, so we would expect the t copula to have more such events due to it's tail dependence. However, the fact that the tranche is so junior is important.

Since the most junior tranche only absorbs the first 3% of the loss, it is only affected by the first 3 defaults (each obligor has 0.02 share in the CDO, and the recovery rate is 0.5). If we look at Figure 10 in Section 3.9, we see that while the t copula has a heavier tail to the right, the Gaussian copula has more samples in bins 2 and 3 (we did not include bins 0 and 1 in that plot). So we expect the events of 2 or 3 defaults (and probably 1 as well), to occur more frequently for the Guassian than the t-copula. Hence the larger expected loss. Also, the flat 3% term structure will not devalue future payoffs a great deal in a 5 year contract.

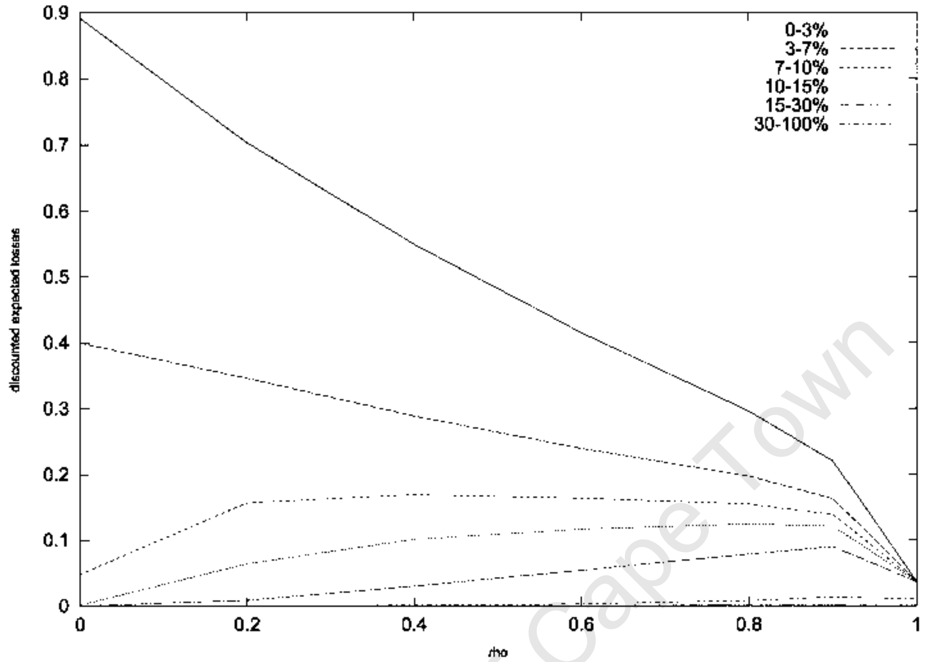

Figure 11: Discounted expected tranche losses for the Gaussian copula (10 000 simulations)

Note that for the more senior tranches, the behaviour is similar for each copula. In the degenerate case where $p = 1$, we have perfect dependence, and the tranches face the exact same risk, except for the most senior tranche. Since the recovery rate is 50%, the largest possible loss the most senior tranche can suffer is 20/70 r'z-, 0.29, which is the ratio between its own loss and that of the other tranches at $p = 1$.

We see quite different behaviour in the fair rates for the 0 — 3% and 15 — 30% tranches in figures 13 and 14. The fact that the rate for the 15 — 30% tranche only starts to significantly increase after about $p = 0.6$ may be an indicator that we need to pass some minimum threshold of dependence before significant numbers (16 or more) of defaults occur that can impact the 15 — 30% tranche.

The standard errors of the fair rate calculations were fairly high (even for large numbers of simulations) so we should be cautious when analysing these results. These large standard errors may be attributed to the fact that an early loss to a tranche results in the notional decreasing very quickly, so the rate payment is only made on the full notional for a short space of time. So there is a very strong inverse relationship between the time of tranche loss and the fair rate. A few very early

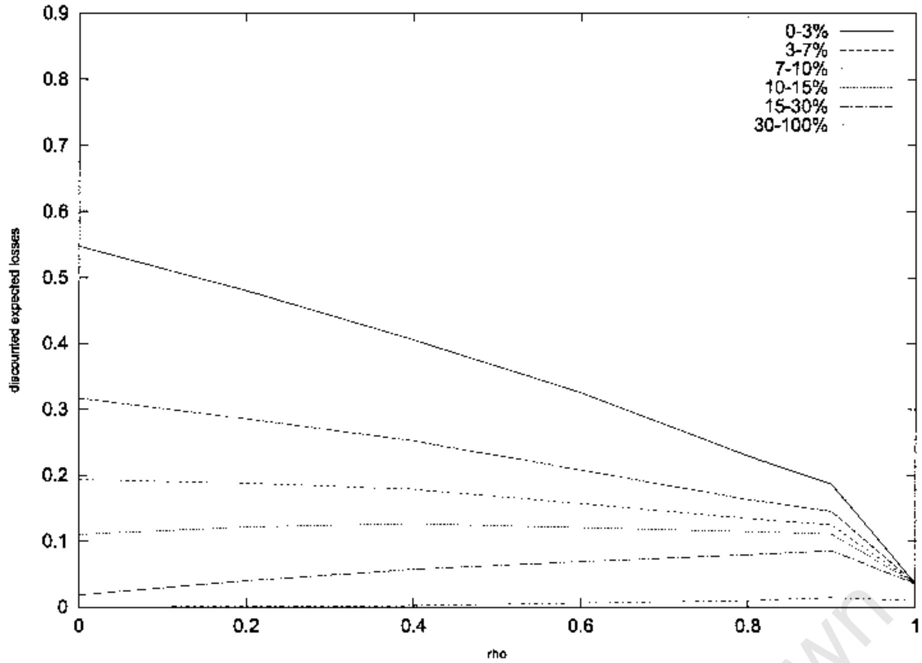

Figure 12: Discounted expected tranche losses for the t copula (10 000 simulations)

default times can increase the sample variance significantly, while extremely for large default times we are rounding down to 5 years, so that tail of the distribution is trimmmed.

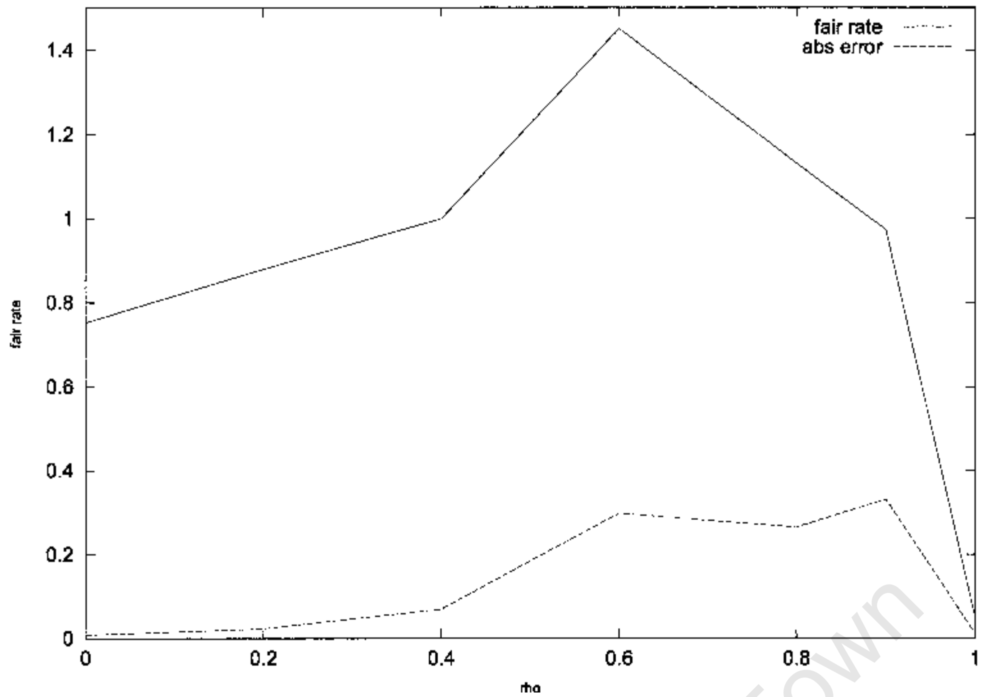

Figure 13: Fair rate and absolute error on the 0-3% tranche for the Gaussian copula (10 000 simulations)

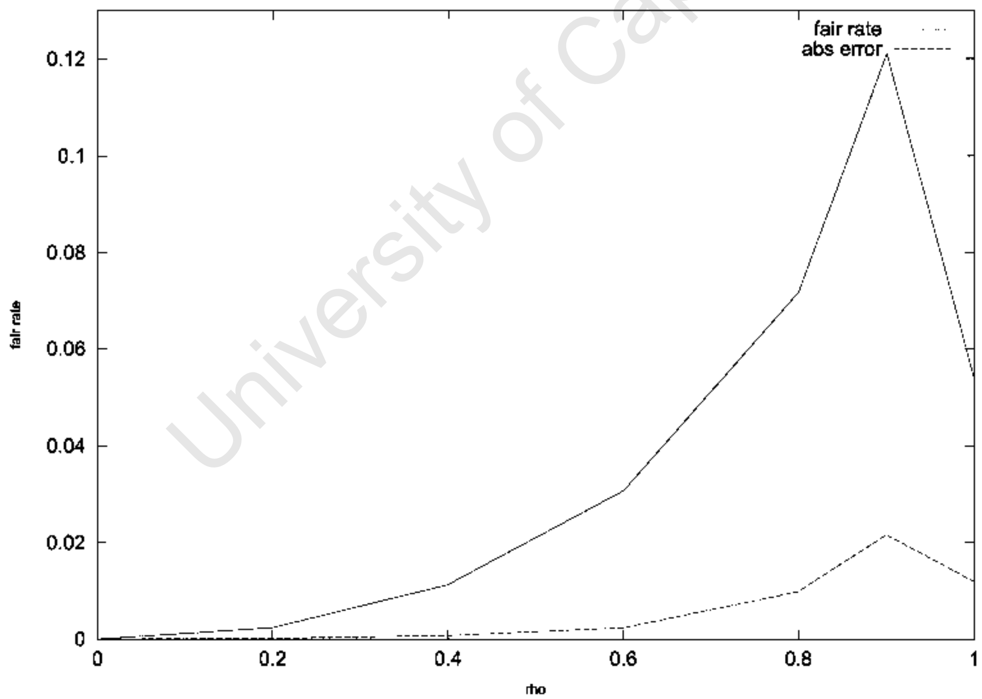

Figure 14: Fair rate and absolute error on the 15-30% tranche for the Gaussian copula (10 000 simulations)

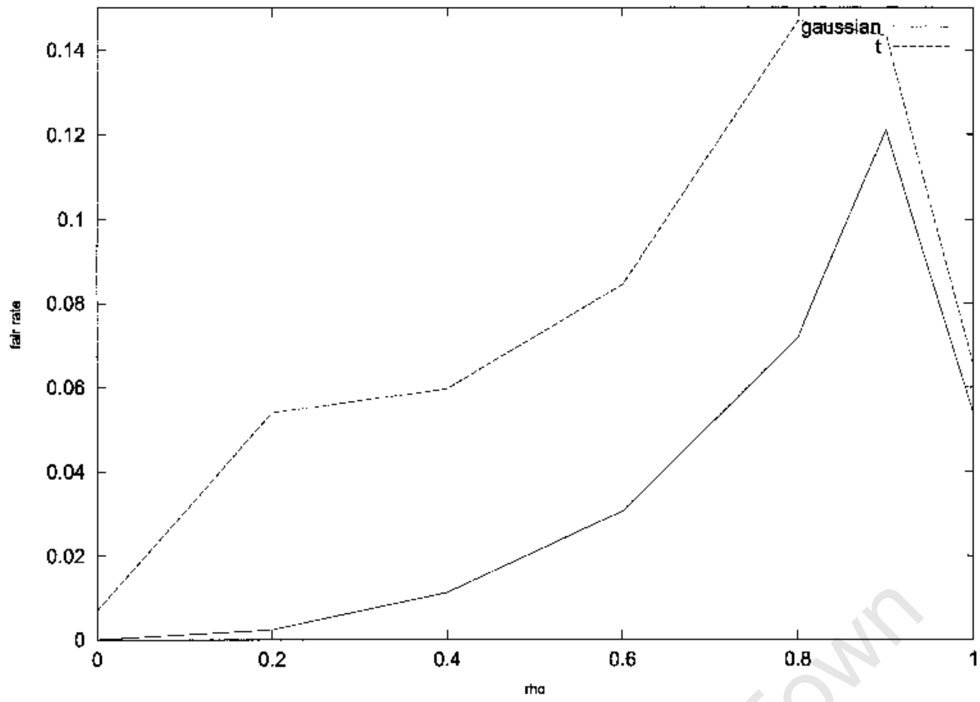

Figure 15: Fair rates on the 15-30% tranche for the Gaussian and t copulas (10 000 simulations)

Figure 15 shows the t copula has a higher rate for the 15 — 30% tranche. Again, if we refer to Figure 10, we see that from bin 16 onwards the t copula has more samples. Furthermore, the t copula is more likely to produce very early default times, which as mentioned earlier correspond to very high rates.

6 Conclusion

We have seen how the different characteristics of copulas, such as tail dependence, are significant factors when modelling CDO's, especially the more senior tranches. Investors must be wary that they do not underestimate the dependence structure in a CDO, since strong dependence may yield even very senior tranches vulnerable to credit risk. The numerical implementation used in this project can be easily extended to include:

- Random recovery rates
- Different interest rate models (and relaxing the assumption of independence between default and interest rates)
- The restructuring of the CDO after defaults (as is typical of CDO indices)
- Modelling delay of the protection leg payment

- Stochastic intensity processes

Furthermore, we don't have to use one copula for the entire dependence structure. We could model the dependence of some obligors using one copula, and those of the others using another. This could be used to characterise a CDO where the underlyings fall into fairly distinct market segments.

One weakness of the MC approach was the high standard error often witnessed with the fair rates, and a lengthy computation time.

One problem with the underlying static copula approach used in this discussion is that it does not allow for dynamic updating of default probabilities. Extensions and alternatives to the basic copula approach may be found in Schöenbucher and Schubert [30], Graziano and Roger [16].

7 Appendix A - Poisson processes

The purpose of this appendix is to provide a brief outline of Poisson processes and their properties, since they feature prominently in intensity models.

Definition 7.1. *Given a sequence of events occuring at various times, we may form the associated counting process N(t) (t > 0) that represents the number of events occuring before t. We list the properties of N(t):*

1. $N([0,\infty)) = \mathbb{N}$
2. *If* $s \leq t$ *then* $N(s) \leq N(t)$
3. *If* $s < t$ *then* $N(t) - N(s)$ *is the number of events occurring in* $(s, t]$.

Definition 7.2. *Let* $T_n = \sum_{i=1}^{n} \tau_i$ *where* $(\tau_i)_{i\geq 1}$ *is a sequence of independent exponential random variables with parameter* $\lambda > 0$. *The Poisson process with constant intensity* λ, *is given by*

$$N_\lambda(t) = \sum_{i=1}^{\infty} \mathbf{1}_{[T_i,\infty)}(t) \quad \forall t \geq 0$$

We let $N_1(t) = N(t)$ *denote the standard Poisson process.*

Since an exponential random variable is greater than 0 *a.s.*, we have that $N_\lambda(0) = 0$ *a.s.*

Thus a Poisson process is a counting process where the distribution of the time between each event is given by an exponential random variable with parameter λ. The distribution of the Poisson process $N_\lambda(t)$ may be described by the *Poisson distribution*:

Proposition 7.3. *Let* $N_\lambda(t)$ *be Poisson process with constant intensity* λ, *then* $N_\lambda(t) \sim \mathcal{P}(\lambda t)$. *I.e.*

$$\Pr(N_\lambda(t) = n) = e^{-\lambda t}\frac{(\lambda t)^n}{n!}$$

Proof. See Cont & Tankov [13] □

We now introduce some properties of the Poisson process.

Definition 7.4. *Suppose* $s \leq t \leq u \leq v$. *A process* $A(t)$ $(t \geq 0)$ *is said to have independent increments if and only if* $A(t) - A(s)$ *is independent of* $A(v) - A(u)$

Proposition 7.5. *Let $N_\lambda(t)$ be a Poisson process with constant intensity λ. Then $N_\lambda(t)$ has independent increments.*

Proof. See Cont & Tankov [13] □

Definition 7.6. *A process $A(t)$ $(t \geq 0)$ is said to have stationary increments if and only if whenever $(t-s) = (v-u)$ we have $A(t) - A(s) \sim A(v) - A(u)$.*

Proposition 7.7. *Let $N_\lambda(t)$ be a Poisson process with constant intensity λ. Then $N_\lambda(t)$ has stationary increments.*

Proof. See Cont & Tankov [13] □

It follows that for $0 \leq s \leq t$:

$$\begin{aligned} N_\lambda(t) - N_\lambda(s) &\sim N_\lambda(t-s) - N_\lambda(0) \\ &\sim \mathcal{P}(\lambda \cdot (t-s)) \end{aligned}$$

Note that the a Brownian motion $B(t)$ has independent increments. Furthermore, $B(t) - B(s) \sim \mathcal{N}(\mu(t-s), \sigma(t-s))$, so that the increments are also stationary. One may think of the time homogeneous Poisson process as a jump-process analog of Brownian motion. The time-homogeneous Poisson process and Brownian motion are examples of *Lévy processes*:

Definition 7.8. *A process $X(t)$, $t \geq 0$ is a Lévy process if and only if*

1. $X(0) = 0$
2. *X(t) has independent increments*
3. *X(t) has stationary increments*
4. *X(t) possesses stochastic continuity:* $\forall \epsilon > 0$, $\lim_{h \to 0} \Pr(|X(t+h) - X(t)| \geq \epsilon) = 0$

We refer the reader to [13] for a thorough discussion of Lévy processes.

We may use Proposition (7.3) to examine some asymptotic behaviour of the Poisson process. We have:

$$\begin{aligned} \lim_{h \to 0} \left\{ \frac{1}{h} \Pr(N_\lambda(t+h) - N_\lambda(t) = 1) \right\} &= \lim_{h \to 0} e^{-\lambda h} \frac{(\lambda h)^1}{h} \\ &= \lim_{h \to 0} \frac{\lambda h + O(h^2)}{h} \\ &= \lambda \end{aligned}$$

Thus for $h << 1$, the probability of one jump occuring during the period $[t, t+h]$ is approximately λh, and this approximation becomes more accurate as h tends to 0. Conversely, the chance of no jump occuring during $[t, t+h]$ is simply $1-\lambda h$. Now we consider the probability of having more than one jump during a small interval:

$$\begin{aligned}\lim_{h\to 0}\left\{\frac{1}{h}\Pr(N_\lambda(t+h)-N_\lambda(t)\geq 2)\right\} &= \lim_{h\to 0}\frac{1}{h}\sum_{i=2}^{\infty}e^{-\lambda h}\frac{(\lambda h)^i}{i!}\\ &= \lim_{h\to 0}\frac{O(h^2)}{h}\\ &= 0\end{aligned}$$

Practically this means that we can't have more than two jumps at the same time.

So far we have only considered the case where λ is a constant. Now we introduce time dependence, so that $\lambda = \lambda(t) \geq 0$, and $\lambda(t)$ is determistic. We assume that $\lambda(t)$ is piecewise continuous, and thus Riemann integrable. We set:

$$\Lambda(t) = \int_0^t \lambda(s)ds$$

If $\lambda(t) > 0$ then $\Lambda(t)$ is strictly increasing.

Definition 7.9. *Suppose we have $\lambda(t) \geq 0$ and $\Lambda(t)$ as above. The time inhomogeneous process $N_{\lambda(t)}(t)$ is defined as:*

$$N_{\lambda(t)}(t) = N(\Lambda(t))$$

So changing the intensity of a Poisson process yields a time dilation. Note that if at some point $\Lambda(t)$ is negative, the time inhomogeneous process would be undefined, so we restrict ourselves to the case where $\Lambda(t) \geq 0$. If $\lambda(t)$ is negative for some open interval, then there will be regions where $\Lambda(t)$ is decreasing, and the process $N_{\lambda(t)}(t)$ will not necessarily be increasing. It would be unrealistic to model a default time with a partially negative intensity, since this would imply that there exist $t_1 < t_2$ such that the probability the firm survives till t_2 will be greater than the probability it survives till t_1. Finally, if $\lambda(t) = 0$ for some open interval then $\Lambda(t)$, and therefore our process will be constant. This would mean that there would be a period $[t_1, t_2]$ where default could not occur. This is unlikely to be a realistic situation for most companies.

By Proposition 7.3 we have $N_{\lambda(t)}(t) = N_1(\Lambda(t)) \sim \mathcal{P}(\Lambda(t))$. Again we consider the asymptotic behaviour:

$$
\begin{aligned}
\lim_{h\to 0}\left\{\frac{1}{h}\Pr(N_{\lambda(t)}(t+h)-N_{\lambda(t)}(t)=1)\right\} &= \lim_{h\to 0}\left\{\frac{1}{h}\Pr(N(\Lambda(t+h))-N(\Lambda(t))=1)\right\} \\
&= \lim_{h\to 0} e^{-(\Lambda(t+h)-\Lambda(t))}\frac{(\Lambda(t+h)-\Lambda(t))^1}{h} \\
&= \lim_{h\to 0}\frac{\Lambda(t+h)-\Lambda(t)+O(h^2)}{h} \\
&= \frac{d\Lambda}{dt} \\
&= \lambda(t)
\end{aligned}
$$

Since we assumed $\lambda(t)$ was piecewise continuous, $\frac{d\Lambda}{dt}$ exists almost everywhere. So in the time inhomogeneous case, the probability of having one jump in a small interval $[t, t+h]$ is $\lambda(t)h$. A similar calculation to earlier shows we may not have two or more jumps in an arbitrarily small interval. This suggests a method for simulating a Poisson process. We simply divide our interval into smaller intervals of width $h \ll 1$, generate a $\mathcal{U}(0,1)$ draw u and if $\lambda(t)h \leq u$ we increase the value by 1, otherwise we leave it constant. However, is not a very efficient way to simulate a Poisson process.

In credit risk modelling we are usually concerned with the first jump time τ of a Poisson process, since this represents the time of default of a firm. Using Definition 7.2, we see that the first jump time ξ_1 of the standard poisson process has a standard exponential distribution. We use this to obtain the distribution of τ:

$$
\begin{aligned}
\Pr(\tau \leq t) &= \Pr(N_{\lambda(t)}(t) > 1) \\
&= \Pr(N(\Lambda(t)) > 1) \\
&= \Pr(\xi_1 \leq \Lambda(t)) \\
&= 1 - e^{-\Lambda(t)}
\end{aligned}
$$

If we let $\xi = \Lambda(\tau)$. Then we have:

$$
\begin{aligned}
\Pr(\xi \leq t) &= \Pr(\Lambda(\tau)) \leq t) \\
&= \Pr(\tau \leq \Lambda^{-1}(t)) \\
&= 1 - e^{\Lambda(\Lambda^{-1}(t))} \\
&= 1 - e^{-t}
\end{aligned}
$$

So by transforming the first jump time by the cumulated density Λ, we obtain a standard exponential random variable. Thus we may find the distribution for τ by taking $\tau = \Lambda^{-1}(\xi)$ where ξ is a standard exponential random variable. If Λ^{-1} has a simple form, then generating random samples of the default time is easy. Indeed, this is the case in the model used in this discussion.

We mentioned in Section 2.3.3 that the Poisson process driving default is assumed to be indepedent of the observable market variables (which are typically driven by Brownian motions). The following result guarantees independence of Poisson processes and Brownian motions defined on the same probability space:

Proposition 7.10. *Let $B(t)$ and $N(t)$ be a Brownian motion and a Poisson process respectively, defined on the same filtered probability space $(\Omega, \mathcal{F}_\infty, \mathbb{P})$. Then $B(t)$ and $N(t)$ are independent.*

Proof. See Bielecki & Rutkowski[6] □

We can extend our use of Poisson processes in a few ways. If we allow the intensity $\lambda(t)$ to be stochastic, we obtain a *Cox*, or *doubly stochastic* process. As mentioned earlier, we would like the intensity process to be strictly positive. We already have a rich supply of strictly positive processes from interest rate theory, such as Vasićek, Cox-Ingersoll-Ross, Dothan, Hull-White (extended Vasićek) etc. [27]. Furthermore, we could correlate our intensity process with interest rates, share prices etc., and thus explicitly introduce dependence between the default time and market variables.

Finally, we mention the *compound* Poisson process, where the jump sizes are no longer determinstic, but are randomly distributed. A more general approach to modelling default could therefore define the default time as the first time the compound process reaches a certain barrier – a similar setting to the first passage time models.

8 Appendix B - Calibration functions

```
%f_XYDiff.m: Returns the difference in present discounted value of
the two legs as a function of lambda
%used for single name cds calibration

function retval = f_XYDiff(lambda)

%Global variables used by this function (lots!)
global GLOBAL_lambda_VEC;
global GLOBAL_R_I;
global GLOBAL_S_VEC;
global GLOBAL_R_VEC;
global GLOBAL_T_VEC;
global GLOBAL_LAMBDA_VEC;
global GLOBAL_T_START;
```

```
global GLOBAL_X_TOTAL;
global GLOBAL_Y_TOTAL;
global GLOBAL_X_CURRENT;
global GLOBAL_Y_CURRENT;
global GLOBAL_F_DIS = @f_DiscountFactorMarket;
global GLOBAL_REC_RATE;
R_CURRENT = GLOBAL_R_VEC(GLOBAL_R_I);

%the net period of this quote (taking away the period of the previous quote)
START =  GLOBAL_S_VEC(GLOBAL_R_I);
MATURITY = GLOBAL_S_VEC(GLOBAL_R_I+1);

T_i = 1;

%Make T_i the first payment point during our net quote period
while GLOBAL_T_VEC(T_i) <= START + 0.0001;
T_i++;
endwhile

%Each time we try a new lambda we need to
update the intensity and the cumulated intensity
GLOBAL_lambda_VEC(GLOBAL_R_I) = lambda;
GLOBAL_LAMBDA_VEC = f_ConstInt(GLOBAL_S_VEC, GLOBAL_lambda_VEC)';

%find the value of the protection leg for the current net quote period
GLOBAL_X_CURRENT = (1-GLOBAL_REC_RATE)*quad(@f_ProtLeg, START, MATURITY,
 [0, 1e-4]);

%now find the value of the premium leg for the current net quote period
GLOBAL_Y_CURRENT = 0;

%when looping over floating points we put in a tol to be safe
while T_i <= length(GLOBAL_T_VEC) && GLOBAL_T_VEC(T_i) <= MATURITY + 0.0001
%find the starting time for the period, since it is part of the integrand
GLOBAL_T_START = GLOBAL_T_VEC(T_i - 1);
T_END = GLOBAL_T_VEC(T_i);
GLOBAL_Y_CURRENT += (T_END - GLOBAL_T_START)*exp(-f_LAMBDA(T_END))
*f_DiscountFactorMarket(T_END) + quad(@f_PremLeg, GLOBAL_T_START, T_END,
[0, 1e-4]);
T_i++;
endwhile

%%we find the difference, scaling the Y value by the current R quote
retval = (GLOBAL_X_TOTAL + GLOBAL_X_CURRENT) - R_CURRENT*(GLOBAL_Y_TOTAL
+ GLOBAL_Y_CURRENT);
```

```
endfunction


%f_ProtLeg.m: Returns the integrand for protection leg for input value s
%used for single name cds calibration

function retval = f_ProtLeg(s)
retval = f_DiscountFactorMarket(s) * exp(-f_LAMBDA(s)) * f_Lambda(s);
endfunction

%f_PremLeg.m: Returns the integrand for premium leg for input value s
%used for single name cds calibration

function retval = f_PremLeg(s)
global GLOBAL_T_START; %we need to use this in the integrand, and we can't
pass parameters to quad, so we use a global variable

retval = (s - GLOBAL_T_START) * f_DiscountFactorMarket(s) * exp(-f_LAMBDA(s))
* f_Lambda(s);
endfunction

%f_LAMBDA.m: returns the value of the cumulated density at time s
%used for single name cds calibration

function retval = f_LAMBDA(s)
global GLOBAL_S_VEC; %the vector of quoted maturities
global GLOBAL_LAMBDA_VEC; %the vector hazard rates

%use linear interpolation for piecewise linear function
retval = interp1(GLOBAL_S_VEC, GLOBAL_LAMBDA_VEC, s);
endfunction

%f_Lambda.m: returns the value of the density at time s
%used for single name cds calibration

function retval = f_Lambda(s)
global GLOBAL_S_VEC; %the vector of quoted maturities
global GLOBAL_lambda_VEC; %the vector hazard rates

count = 1;
%iterate till we find the right period
while GLOBAL_S_VEC(count+1) < s
count++;
endwhile
```

```
retval = GLOBAL_lambda_VEC(count);
endfunction

%f_ConstInt.m: Determines the integral of a piecewise constant function
%Used to determine cumulated hazard function given time boundaries and
associated constant intensities for single name cds calibration
%Also used to calculate cumulative tranche notionals between the payment
dates for cdo modelling
%If T has n points, lambda has n-1 pieces

function rd_Gamma = f_ConstInt(rd_T, rd_Lambda)
rd_Gamma = zeros(1, length(rd_T));
for i = 2:length(rd_T)
rd_Gamma(i) = rd_Gamma(i-1) + (rd_T(i) - rd_T(i-1))*rd_Lambda(i-1);
endfor
endfunction

%f_DiscountFactorMarket: function to return the discount factor based
on a constant interest rate r
%used in single name cds calibration

function vd_DiscountFactor = f_DiscountFactorMarket(t)
%read the market data
 persistent pers_MarketDiscountFacs = load("./MarketData
 /market_DiscountFactors.data");
 persistent pers_MarketTimes = load("./MarketData
 /market_DiscountTimes.data");

%use simple linear interpolation
vd_DiscountFactor = interp1(pers_MarketTimes,
pers_MarketDiscountFacs, t);
endfunction

%%f_PieceWiseInvert: Generates a time from the poisson process with
piecewise constant intensity
%The columns of GLOBAL_GAMMAVEC must be the cumulated hazard rates on the
payments grid
%used in monte carlo simulation of cdo's

function cd_Tau = f_PieceWiseInvert(vd_U)

%we need to access the cumulated intensity functions
global GLOBAL_GAMMAVEC;
global GLOBAL_TVEC;
```

```
cd_Tau = zeros(1, length(vd_U));

%find the default times:
for i = 1:length(vd_U)
d_Xi = expinv(vd_U(i),1);
if d_Xi > max(GLOBAL_GAMMAVEC) %if xi is outside the
 range of gamma we just set it to inf
cd_Tau(i) =inf;
else
cd_Tau(i) = interp1(GLOBAL_GAMMAVEC,
GLOBAL_TVEC, d_Xi);
endif
endfor
endfunction
```

9 Appendix C - Monte Carlo functions

```
%f_GaussDraw.m: Returns a column vector draw from the gaussian copula
given correlation matrix Sigma
%used for monte carlo simulation

function cd_Draw = f_GaussDraw(md_Sigma)
%make the cholesky factorisation global, so that we don't have
to recalculate for each MC simulation
global md_Chol;
cd_Draw = randn(rows(md_Sigma), 1);
cd_Draw = (md_Chol')*cd_Draw;
cd_Draw = normcdf(cd_Draw, 0, 1); %return column vector
endfunction
```

```
%f_StudentDraw.m: Returns a column vector draw from the Student-t copula given
correlation matrix Sigma, and degrees of freedom dof
%used for monte carlo simulation

function cd_Draw = f_StudentDraw(md_Sigma, d_Dof)
%make the cholesky factorisation persistent, so that we don't have to
recalculate for each MC simulation
persistent md_Chol = (chol(md_Sigma)');
cd_Draw = randn(rows(md_Sigma), 1);
cd_Draw = (md_Chol')*cd_Draw;
```

```
cd_Draw = sqrt(d_Dof/chi2rnd(d_Dof, 1))*cd_Draw;
cd_Draw = tcdf(cd_Draw, d_Dof); %return the column vector
endfunction
```

```
%f_DiscountR: Returns the discount factor for a given time based on a
constant interest rate r
%used in monte carlo simulation of a cdo
function DiscountFactor = f_DiscountFactorR(T)
global GLOBAL_R;
DiscountFactor = exp(-GLOBAL_R*T);
endfunction
```

```
%f_FairRates.m: Determines the fair rate R for each tranche given
nominal tranche losses at
each default time, the the discount factors at the payment grid
times, the default times,
the grid of payment times, and the maturity of the contract

function rd_FairRates = f_FairRates(md_NominalTrancheLosses, rd_TrancheLosses,
cd_RunningDiscountFactors, cd_DefaultTimes, cd_PaymentTimes, d_T)

i_NumTranches = length(rd_TrancheLosses);
i_NumDefaults = length(cd_DefaultTimes);
i_NumPayments = length(cd_PaymentTimes);

%First we determine the running notional amounts between the n d
efault times
%At the start of the contact, every tranche has a full notional
and running
notionals have n+1 pieces
md_RunningNotionals = [ones(1, i_NumTranches);
1-cumsum(md_NominalTrancheLosses)];

%There will be n+2 points for our cumulative notional functions (the n
default times plus the 2 end points)
md_CumDefaultNotionals = zeros(i_NumDefaults + 2, i_NumTranches);
cd_DefaultsGrid = [0; cd_DefaultTimes; d_T]

%Then we integrate to find the cumulative running notionals
```

```
at the default times
for i = 1:i_NumTranches
md_CumDefaultNotionals(:,i) = f_ConstInt(cd_DefaultsGrid,
md_RunningNotionals(:,i));
endfor

%Then we interpolate between default times to get the cumulative
running notionals
at payment dates
md_CumPaymentNotionals = zeros(i_NumPayments + 1, i_NumTranches);
cd_PaymentsGrid = [0; cd_PaymentTimes];

for i = 1:i_NumTranches
md_CumPaymentNotionals(:,i) = interp1(cd_DefaultsGrid,
md_CumDefaultNotionals(:,i),
 cd_PaymentsGrid);
endfor


%Now we find the 1/PaymentsPerYear*(average notional) over
each payment period
md_MeanPaymentNotionals = zeros(i_NumPayments, i_NumTranches);

for i = 1:i_NumTranches
md_MeanPaymentNotionals(:,i) = md_CumPaymentNotionals(
2:i_NumPayments + 1, i) -
md_CumPaymentNotionals(1: i_NumPayments, i);
endfor

md_MeanPaymentNotionals

%We get the R coefficients by taking the scalar product of the mean
 notionals
 and the discount factors
%md_MeanPaymentNotionals = 0.25*ones(i_NumPayments, i_NumTranches);
rd_RCoeffs = (cd_RunningDiscountFactors')*md_MeanPaymentNotionals;

%Finally we take pairwise division of the expected discounted losses by
coefficients
rd_FairRates = rd_TrancheLosses./rd_RCoeffs;
endfunction



% f_CDOMonteCarloAnti.m: Returns antithetic Monte Carlo determination of
```

expected loss and fair rates for CDO for each tranche %Takes in a row vector of notional amounts, row vector of associated recovery rates, row vector of tranche attachment points, a copula function for random draws from the hypercube (column vector), an inverse transform function for default times, a discount factor function, time till maturity (years), number of payments per year, absolute tolerance level, max number of simulations

```
function [rd_MeanTrancheLosses rd_AbsErrTrancheLosses rd_MeanFairRates
rd_AbsErrFairRates] = f_CDOMonteCarloAnti(rd_Notionals, rd_RECs,
rd_TranchePoints, f_CopulaDraw, f_TimeInvert,
  f_DiscountFactor, d_T, i_PaymentsPerYear, TOL, MAX_N)

%Set up obligor constants
i_Obligors = length(rd_Notionals);
rd_Weights = rd_Notionals/sum(rd_Notionals);

%Set up tranche information
i_NumTranches = length(rd_TranchePoints) - 1;
rd_TrancheWidths = rd_TranchePoints(2: i_NumTranches + 1) - r
d_TranchePoints(
1: i_NumTranches);

%Set up payment dates and the associated discount factors
cd_PaymentTimes = linspace(0 + 1/i_PaymentsPerYear, d_T,
d_T*i_PaymentsPerYear)';
cd_RunningDiscountFactors = f_DiscountFactor(cd_PaymentTimes);

%Calculate the loss incurred for each obligor default
rd_DefaultLossesFixed = (1 - rd_RECs).*rd_Weights;

%Create variables to store the totals from for each tranche
%First row is for the sum, the second row is for the sum of
squares (used for variance estimate)
cd_TotalGlobalLoss = zeros(2,1);
md_TotalTrancheLosses = zeros(2, i_NumTranches);
md_TotalFairRates = zeros(2, i_NumTranches);

%Create variables to store the absolute errors
d_AbsErrGlobalLoss = 1;
rd_AbsErrTrancheLosses = ones(1, i_NumTranches);
rd_AbsErrFairRates = ones(1, i_NumTranches);

for i = 1:MAX_N/2
rd_TempLosses = zeros(1, i_NumTranches);
```

```
rd_TempRates = zeros(1, i_NumTranches);
d_TempGlobalLoss = 0;

cd_U =  f_CopulaDraw(i_Obligors);
for j = 1:2
%If we have reached an acceptable tolerance
 level for each estimate we break
%We also don't want any flukes, so we ensure
i > 20
if max(rd_AbsErrTrancheLosses) < TOL && max(
rd_AbsErrFairRates)
 < TOL && d_AbsErrGlobalLoss < TOL && i > 20
disp(["poesed number: ", num2str(i)]);
break;
endif

%If we don't reach our tolerance level :(
if i == MAX_N
disp("Warning: Simulation limit reached");
endif

 %generate the random times from the quantiles
cd_DefaultTimes = f_TimeInvert(cd_U);

%Isolate those defaults (times and amounts)
that occurred  during the contract
vbool_Defaults = cd_DefaultTimes <= d_T;
cd_DefaultTimes = cd_DefaultTimes(vbool_Defaults);
cd_DefaultLosses = (rd_DefaultLossesFixed(
vbool_Defaults))';

%If we have no defaults, we continue
if length(cd_DefaultTimes) == 0
continue;
endif

%Sort the defaults (times and amounts)
[cd_DefaultTimes, v_I] = sort(cd_DefaultTimes);
cd_DefaultLosses = cd_DefaultLosses(v_I);

%Find the discount factors for the default
times
cd_DefaultDiscountFactors = f_DiscountFactor(
cd_DefaultTimes);
```

```
%Determine the total discounted loss for the
whole CDO (tranche independent)
d_TempGlobalLoss += (cd_DefaultLosses')*
cd_DefaultDiscountFactors;

%Calculate the discounted tranche losses
md_TempNominalLosses = f_NominalTrancheLosses(
cd_DefaultLosses,
 rd_TranchePoints, rd_TrancheWidths);
rd_TempLosses += (cd_DefaultDiscountFactors')
*md_TempNominalLosses;

%Calculate the associated fair rates
rd_TempRates += f_FairRates(rd_TempNominalLosses,
rd_TrancheLosses,
 cd_RunningDiscountFactors, cd_DefaultTimes,
 cd_PaymentTimes, d_T);
%take the antithetic path
cd_U = 1 - cd_U;
endfor

%now take the mean of the antithetic results:
cd_TotalGlobalLoss(1) += d_TempGlobalLoss/2;
cd_TotalGlobalLoss(2) += d_GlobalLoss^2;

md_TotalTrancheLosses(1,:) += (rd_TempLosses/2);
md_TotalTrancheLosses(2,:) += (rd_TempLosses/2).^2;


md_TotalFairRates(1,:) += rd_TempRates/2;
md_TotalFairRates(2,:) += (rd_TempRates/2).^2;

%[d_MeanGlobalLoss d_AbsErrGlobalLoss] = f_Stats(
cd_TotalGlobalLoss, i);
[rd_MeanTrancheLosses rd_AbsErrTrancheLosses] =
f_Stats(md_TotalTrancheLosses, i);
[rd_MeanFairRates rd_AbsErrFairRates] = f_Stats(
md_TotalFairRates, i);

endfor
endfunction



% f_CDOMonteCarlo.m: Returns Monte Carlo determination of expected
```

```
loss and fair rates for CDO for each tranche

%Takes in a row vector of notional amounts, row vector of associated
recovery rates, row vector of tranche attachment points, a copula
function for random draws from the hypercube (column vector), an
 inverse transform function for default times, a discount factor
  function, time till maturity (years), number of payments per year,
 absolute tolerance level, max number of simulations

function [rd_MeanTrancheLosses rd_AbsErrTrancheLosses rd_MeanFairRates
rd_AbsErrFairRates] = f_CDOMonteCarlo(rd_Notionals, rd_RECs,
rd_TranchePoints, f_CopulaDraw, f_TimeInvert, f_DiscountFactor, d_T,
 i_PaymentsPerYear, TOL, MAX_N)

%Set up obligor constants
i_Obligors = length(rd_Notionals);
rd_Weights = rd_Notionals/sum(rd_Notionals);

%Set up tranche information
i_NumTranches = length(rd_TranchePoints) - 1;
rd_TrancheWidths = rd_TranchePoints(2: i_NumTranches + 1) -
rd_TranchePoints(1:
i_NumTranches);

%Set up payment dates and the associated discount factors
cd_PaymentTimes = linspace(0 + 1/i_PaymentsPerYear, d_T,
d_T*i_PaymentsPerYear)';
cd_RunningDiscountFactors = f_DiscountFactor(cd_PaymentTimes);

%Calculate the loss incurred for each obligor default
rd_DefaultLossesFixed = (1 - rd_RECs).*rd_Weights;

%Create variables to store the totals from for each tranche
%First row is for the sum, the second row is for the sum of
squares (used for
variance estimate)
cd_TotalGlobalLoss = zeros(2,1);
md_TotalTrancheLosses = zeros(2, i_NumTranches);
md_TotalFairRates = zeros(2, i_NumTranches);

%Create variables to store the absolute errors
d_AbsErrGlobalLoss = 1;
rd_AbsErrTrancheLosses = ones(1, i_NumTranches);
rd_AbsErrFairRates = ones(1, i_NumTranches);
```

```
for i = 1:MAX_N
%If we have reached an acceptable tolerance level
for each estimate
 we break
%We also don't want any flukes, so we ensure i > 20
if max(rd_AbsErrTrancheLosses) < TOL && max(
rd_AbsErrFairRates)
< TOL && d_AbsErrGlobalLoss < TOL && i > 20
disp(["poesed number: ", num2str(i)]);
break;
endif

%If we don't reach our tolerance level :(
if i == MAX_N
disp("Warning: Simulation limit reached");
endif

cd_U =  f_CopulaDraw(i_Obligors);

 %generate the random times from the quantiles
cd_DefaultTimes = f_TimeInvert(cd_U);

%Isolate those defaults (times and amounts) that
occurred during
the contract
vbool_Defaults = cd_DefaultTimes <= d_T;
cd_DefaultTimes = cd_DefaultTimes(vbool_Defaults);
cd_DefaultLosses = (rd_DefaultLossesFixed(
vbool_Defaults))';

%If we have no defaults, we continue
if length(cd_DefaultTimes) == 0
continue;
endif

%Sort the defaults (times and amounts)
[cd_DefaultTimes, v_I] = sort(cd_DefaultTimes);
cd_DefaultLosses = cd_DefaultLosses(v_I);

%Find the discount factors for the default times
cd_DefaultDiscountFactors = f_DiscountFactor(
cd_DefaultTimes);

% cd_DefaultLosses
% cd_DefaultTimes
```

```
% pause;

%Determine the total discounted loss for the whole CDO
(tranche independent)
d_GlobalLoss = (cd_DefaultLosses')
*cd_DefaultDiscountFactors;
cd_TotalGlobalLoss(1) += d_GlobalLoss;
cd_TotalGlobalLoss(2) += d_GlobalLoss^2;

%Calculate the discounted tranche losses
md_NominalTrancheLosses = f_NominalTrancheLosses(
cd_DefaultLosses, rd_TranchePoints, rd_TrancheWidths);
rd_TrancheLosses = (cd_DefaultDiscountFactors')
*md_NominalTrancheLosses; md_TotalTrancheLosses(1,:)
+= rd_TrancheLosses;
md_TotalTrancheLosses(2,:) += rd_TrancheLosses.^2;

%Calculate the associated fair rates
rd_FairRates = f_FairRates(md_NominalTrancheLosses,
rd_TrancheLosses,
cd_RunningDiscountFactors, cd_DefaultTimes, cd_PaymentTimes,
d_T);
md_TotalFairRates(1,:) += rd_FairRates;
md_TotalFairRates(2,:) += rd_FairRates.^2;

[d_MeanGlobalLoss d_AbsErrGlobalLoss] = f_Stats(
cd_TotalGlobalLoss, i);
[rd_MeanTrancheLosses rd_AbsErrTrancheLosses] = f_Stats(
md_TotalTrancheLosses,
 i);
[rd_MeanFairRates rd_AbsErrFairRates] = f_Stats(
md_TotalFairRates, i);
endfor

d_AbsErrGlobalLoss*10000;
rd_AbsErrTrancheLosses*10000;
rd_AbsErrFairRates*10000;
endfunction



%f_trancheLoss.m: Function to determine the nominal loss per tranche at each
default time
function md_NominalTrancheLosses = f_NominalTrancheLosses(cd_DefaultLosses,
rd_TranchePoints, rd_TrancheWidths)
```

```
%Get some basic info
i_NumTranches = length(rd_TranchePoints) - 1;
i_NumDefaults = length(cd_DefaultLosses);

%Matrix to hold the loss incurred in each tranche for each default
md_NominalTrancheLosses = sparse(i_NumDefaults, i_NumTranches);
md_NominalTrancheRemaining = ones(i_NumDefaults, i_NumTranches);

%Cumulated nominal loss points for before and after the current
 default
d_LowNomTotal = 0;
d_HighNomTotal = 0;

%%Start off with most junior tranche, and get its bounds
i_TrancheIndex = 1;
d_LBound = rd_TranchePoints(i_TrancheIndex);
d_UBound = rd_TranchePoints(i_TrancheIndex + 1);

%%Iterate over each default - if there are no defaults the loop
will not execute, and the values have already been set to 0
for i = 1:i_NumDefaults
d_HighNomTotal += cd_DefaultLosses(i);

  %if we haven't reached the lowest tranche yet we keep
  adding   defaults
if d_HighNomTotal < d_LBound
continue;

%when the lower sum has gone past the last tranche there is no
 point in continuing
elseif  d_LowNomTotal > rd_TranchePoints(length(
rd_TranchePoints))
return;
else
if d_LowNomTotal < d_LBound  %first part of default
 does  not lie in this tranche
d_LowNomTotal = d_LBound;
endif

%We keep shifting the lower point up till we
reach the higher point
%Once we have reach the higher point we have processed
the default completely
 while d_LowNomTotal != d_HighNomTotal
```

```
if d_HighNomTotal <= d_UBound  %The rest of the
loss lies in this tranche
md_NominalTrancheLosses(i,
i_TrancheIndex) = d_HighNomTotal
 - d_LowNomTotal;
d_LowNomTotal = d_HighNomTotal;
else %Some of the loss lies in the next tranche
md_NominalTrancheLosses(i, i_TrancheIndex)
 = d_UBound - d_LowNomTotal;
d_LowNomTotal = d_UBound;
d_LBound = d_UBound;
if i_TrancheIndex < i_NumTranches %Make
sure we don't go past the last tranche
i_TrancheIndex++;
d_UBound = rd_TranchePoints(
i_TrancheIndex+1);
else
break;
endif
endif
endwhile
endif
endfor

%Now scale the losses by the tranchewidths
for i = 1:i_NumDefaults
md_NominalTrancheLosses(i,:) = md_NominalTrancheLosses(i,:)
./rd_TrancheWidths;
endfor
endfunction



%f_Stats.m: Returns the mean and absolulte error statistics given sum, sum of
squares and number of scenarios
%first row of vd_Totals contains the sum, the second row contains the sum of
squares

function [rd_Means, rd_AbsErr] = f_Stats(md_Totals, N)
rd_Means = md_Totals(1,:)/N;

if N == 1 || max(rd_Means)  == 0%if we only have one simulation or
only zero mean we can't get the error
rd_AbsErr = ones(size(md_Totals(1,:)));
else
```

```
rd_Var = 1/(N-1)*(md_Totals(2,:) - N*rd_Means.^2);
rd_AbsErr = sqrt(rd_Var/N);
endif
endfunction


%main_CDO.m: MAIN FUNCTION: Generates data for graphs using MC simulation of the CDO

clear all;

%Set up basic contract parameters
d_T = 5; %maturity of contract
i_Obligors = 31;
%d_Notionals = 100; %scalar of notional amounts (used if all the same)
rd_Notionals = ones(1,i_Obligors); %tester
rd_Notionals(31) = 20;
d_REC = 0.5; %recovery rate (used if all the same)
rd_RECs = ones(1, i_Obligors)*d_REC; %vector of recovery rates
vd_FracTranchePoints = [0 0.03 0.07 0.1 0.15 0.30 1]; %vector of tranche attachment
 points
i_PaymentsPerYear = 4; %number of protection leg payments for the year
BASISPOINT = 1e4;


%Set up statistical stuff
global GLOBAL_R = 0.03; %value for constant interest rate
%global GLOBAL_RHO = 0.8; %correlation coefficient
global GLOBAL_DOF = 2;
global GLOBAL_LAMBDA = 0.02; %hazard rate
global GLOBAL_RHO = 0.4;
global GLOBAL_CHOLESKY

%Set up monte-carlo parameters
TOL =1e-4;
MAX_N = 20000;

%randn("state", 1); %seed the random vector for testing
%run the monte carlo

%set the ranges to vary our parameters
vd_RECVec = 0;%linspace(0, 0.9, 3);

%create 3-D arrays to hold the fair rates and mean losses for each tranche
i_NumTranches = length(vd_FracTranchePoints) - 1;
```

```
%create our choices for rho
rho_vec = [0 0.2 0.4 0.6 0.8 0.9 1];
i_Rhos = length(rho_vec);


%Set up variables to store the data for gaussian case
md_FairRatesGauss = zeros(i_Rhos, i_NumTranches);
md_ExpectedLossesGauss = zeros(i_Rhos, i_NumTranches);
md_AbsErrLossesGauss = zeros(i_Rhos, i_NumTranches);
md_AbsErrRatesGauss = zeros(i_Rhos, i_NumTranches);

tic
for i = 1:i_Rhos
GLOBAL_RHO = rho_vec(i);
[md_ExpectedLossesGauss(i,:) md_AbsErrLossesGauss(i,:) md_FairRatesGauss(
i,:) md_AbsErrRatesGauss(i,:)]  = f_CDOMonteCarlo(rd_Notionals, rd_RECs,
vd_FracTranchePoints, @f_GaussRhoDraw, @f_ExpInvert, @f_DiscountFactorR,
d_T, i_PaymentsPerYear, TOL, MAX_N);
clear f_GaussDraw;
endfor


%Set up variables to store the data for student t case
md_FairRatesStudent = zeros(i_Rhos, i_NumTranches);
md_ExpectedLossesStudent = zeros(i_Rhos, i_NumTranches);
md_AbsErrLossesStudent = zeros(i_Rhos, i_NumTranches);
md_AbsErrRatesStudent = zeros(i_Rhos, i_NumTranches);

for i = 1:i_Rhos
GLOBAL_RHO = rho_vec(i);
[md_ExpectedLossesStudent(i,:) md_AbsErrLossesStudent(i,:)  md_FairRatesStudent(
i,:) md_AbsErrRatesStudent(i,:)]  = f_CDOMonteCarlo(rd_Notionals, rd_RECs,
vd_FracTranchePoints, @f_StudentRhoDraw, @f_ExpInvert, @f_DiscountFactorR,
d_T, i_PaymentsPerYear, TOL, MAX_N);
clear f_StudentDraw;
endfor
toc
```

References

[1] Alexander C., *Market Risk Analysis: Practical Financial Econometrics Vol. 2.* John Wiley and Sons, 2008.

[2] Amman, M., *Credit Risk Valuation – Methods Models and Applications.* Springer-Verlag, 2001.

[3] Barret R. et al., *BBA Credit Derivatives Report 2006* British Bankers Association Enterprises Limited, 2006.

[4] Becker, R. *Numerical Modelling Course Notes*

[5] Bhansali V. et al., *Systematic credit risk: What is the market telling us?.* 2008.

[6] Bielecki T.R. Rutkowski M., *Credit Risk: Modelling, Valuation and Hedging.* Springer Science and Business Media, 2006.

[7] Black F. Siz Cox, J.C., *Valuing Corporate Securities: Some Effects of Bond Indenture Provisions* The Journal of Finance, Vol.31 No. 2, 1976.

[8] Brigo, D. & Mercurio, F., *Interest Rate Models - Theory and Practice - With Smile, Inflation and Credit.* Springer-Verlag, 2006.

[9] Casey, 0. *The CDS Big Bang.* The Markit Magazine, Spring 2009.

[10] Chatiras M., Mukherjee B., *Capital Structure Arbitrage: An Empirical Investigation using Stocks and High Yield Bonds.* Working Paper, University of Masschusetts, Amherst, 2004.

[11] Cherubini U. et.. al, *Copula Methods in Finance.* John Wiley Sz Sons, 2004.

[12] Chiodo A. Sz Owyang M., *A Case Study of a Currency Crisis: The Russian Default of 1998.* Federal Reserve Bank of St. Louis Review Vol, 86(2), 2004.

[13] Cont, R. Sz Tankov P., *Financial Modelling with Jump Processes.* Chapman Sz Hall, 2003

[14] Embrechts, P. et al. *Modelling Dependence with Copulas and Applications to Risk Management.* ETH Zurich, Working paper, 2001.

[15] Fusai G. Sz Roncoroni A., *Implementing Models in Quantitative Finance.* Springer-Verlag, 2008.

[16] Graziano, D.G., Sz Rogers, L.C.G, *A dynamic approach to the modelling of correlation credit derivatives using Markov chains* Statistical laboratorym, University of Cambridge working paper, November 2006.

[17] Kalemanova, A. et al *The Normal Inverse Gaussian Distribution for Synthetic CDO pricing.* Journal of derivatives, Spring 2007.

[18] R. Jarrow, Turnbull, S., *Pricing Derivatives on Financial Securities Subject to Credit Risk.* The Journal of Finance, Vol. 50, No. 1., March 1995

[19] Li D.X., *On Default Correlation: A Copula Function Approach.* RiskMetrics Working Paper, September 1999.

[20] Li H., *Tail Dependence Comparison of Survival Marshall‐Olkin Copulas.* Methodology and Computing in Applied Probability, 2007.

[21] Markit Data Guide Logical Information Machines, Inc., July 24, 2007.

[22] Marshall A.W.Sz Olkin I., *Families of Multivariate Distributions.* Journal of the American Statistical Association, Vol. 83, No. 403, September 1988.

[23] McKenzie, D. *Derivatives 2007.* International Financial Services, London, 2007

[24] Merton, R.C., *On the Pricing of Corporate Debt: The Risk Structure of Interest Rates.* The Journal of Finance, Vol. 29, No. 2, 1974.

[25] Moore, S.D., Spruill, M.C., *Unified Large‐Sample Theory of General Chi-Squared Statistics for Tests of Fit.* The Annals of Statistics, Vol. 3, No. 3, pp. 599-616, May, 1975.

[26] Nelsen R.B., *An Introduction to Copulas, Second Edition.* Springer‐Verlag, 2002.

[27] Ouwehand, P., *Foundations of Stochastic Finance.* University of Cape Town, working notes, 2004.

[28] Quesada-Molina J.J. et. al, *What are Copulas?.* Monografias del Semin. Matem. Garcia de Galdeano, 2003.

[29] Schweizer B., and Sklar A., *Probabilistic Metric Spaces.* New York: North Holland, 1983.

[30] Schöenbucher P.J., Sz Schubert D., *Copula‐Dependent Default Risk in Intensity Models.* JEL 19, 2001.

[31] ISDA News Release. International Swaps and Derivatives Association, Inc., 26 September, 2006.

[32] ISDA 2008 Operations Benchmarking Survey. International Swaps and Derivatives Association, Inc., 2008.

[33] *Markit Website.* www.markit.com

[34] The Office of the Comptroller of Currency *OCCs Quarterly Report on Bank Trading and Derivatives Activities - Second Quarter 2009.* 2009.

[35] Sweeney Table. www2.standardandpoors.com/spf/pdf/media/SweeneyTab1e1.pdf