

Pricing Models for Collateralized Debt Obligations

by
Julie D. Hurley, B.Sc.

A thesis submitted to the University of Limerick for the
Degree of Master of Science.

Department of Mathematics and Statistics,
University *of* Limerick

Supervisor: Prof. James P. Gleeson

May 2011

Author's Declaration

This thesis is presented in fulfilment of the requirements for the degree of a Masters of Science. It is entirely my own work and has not been submitted to any other University or higher education institution, or for any other academic award in this University. Where use has been made of the work of other people it has been fully acknowledged and fully referenced.

Signature -----
Julie Hurley

Abstract

One of the most controversial and innovative financial products in recent years has been collateralised debt obligations (CDOs). Much of the blame for the current credit crisis is being attributed to the mathematical models and quantitative methods associated with these credit derivatives. In recent years, there has been rapidly growing research on credit derivatives and correlated defaults and in this thesis we examine the possible replacement of current copula based approaches with intuitive contagion models for percolation on finite networks. We propose that modelling the probability of default in a correlated portfolio is similar to modelling the probability of default of contagion spreading in a network.

In the first part of our thesis we review current models from the literature that have been suggested to price CDOs. From the literature review we noted that the critical input in the pricing of a CDO is an estimate of the default dependence (default correlation) between the underlying names in a portfolio. Dependency modelling with copula functions, introduced by Li (2000), has become a market standard in the pricing of CDOs. We compute the default distribution for both the Gaussian and Student t4 copulas by implementing both a Monte Carlo and theoretical approach. We then compare these copulas ability to fit market data.

In the second part of our thesis we begin by introducing some of the earliest network models proposed by Paul Erdős and Alfréd Rényi. We show how similar to component sizes on a network being related to contagion on a network, it is possible then to compute the probability of default for a portfolio of names using network percolation. We present the finite and infinite network models used to model the default dependency between names in a portfolio and thus extend the infinite network model to compute the probability of default for a CDO portfolio. We finally compare the percolation model, Gaussian and Student t4 copulas to market data and from our results determine that the percolation model provides a better fit to market data than the Gaussian and Student t4 copulas.

Acknowledgements

My utmost gratitude goes to my thesis supervisor, Prof. James Gleeson for allowing me to join his team, for his expertise, kindness, and most of all, for his patience. One simply could not wish for a better supervisor.

I would like to express my thanks to Dr. Sergey Melnik for his helpful advice and comments during the past two years and a big thank you to Dr. Bernard Murphy from the Kemmy Business School for his help on the finance aspect of my thesis.

I thank the academic and administrative staff of the Department of Mathematics and Statistics, UL for providing the environment to facilitate this work and especially Dr. Patrick Johnson for his L^AT_EX expertise.

I gratefully acknowledge funding provided by Science Foundation Ireland under programmes 06/IN.1/I366 and MACSI 06/MI/005.

Lastly, I would like to thank my boyfriend Kieran and my family for their endless support, love and encouragement. For my parents who have supported and encouraged me to do my best in all matters of life, to them I dedicate this thesis.

Contents

Author's Declaration	i
Abstract	i
Acknowledgements	i
List of Figures	iv
List of Tables	ix
1 Introduction	1
1.1 General introduction	1
1.1.1 Introduction to CDOs	2
1.1.2 How a CDO works	3
1.1.3 Types of CDOs	7
1.2 Literature review	8
1.2.1 Copula model	8
1.2.2 Factor copula model	9
1.2.3 Alternatives to copula models	14
1.2.4 Correlation smile and base correlations	15
2 One factor copula model	18
2.1 The factor copula approach	19
2.1.1 Large homogeneous pool	20
2.2 The distribution of defaults	20
2.2.1 Threshold value K	21

2.2.2	Conditional default probability	21
2.2.3	Default probability distribution P_n	22
3	One factor copula models with different underlying distributions	25
3.1	The one factor Gaussian copula	26
3.1.1	Threshold value K	27
3.1.2	Monte Carlo techniques to compute P_n	28
3.1.3	Theoretical default distribution when $\rho = 0$	31
3.1.4	Theoretical default distribution for $\rho > 0$	33
3.1.5	Comparing Monte Carlo and theoretical approaches	34
3.1.6	Shortcomings of the Gaussian copula	36
3.2	Student t copula	37
3.2.1	Threshold value K	39
3.2.2	Monte Carlo methods to compute P_n	42
3.2.3	Theoretical default distribution	43
3.2.4	Comparing Monte Carlo and theoretical approaches	46
3.3	Comparing Gaussian and Student t4 copulas	48
3.4	Base correlations	51
3.4.1	Implied correlation	54
3.4.2	Base correlation	54
3.5	Expected loss	57
3.6	Comparing models	58
4	Network models	61
4.1	Introduction to network theory	61
4.2	Infinite network model	64
4.2.1	Degree distribution	64
4.2.2	Probability generating functions	65
4.2.3	The distribution of component sizes	67
4.2.4	Infinite network model	68
4.3	Finite network model	70
4.3.1	Component structure	70

4.4	Comparing infinite and finite networks models	73
5	The percolation model	80
5.1	The default distribution averaging over seed values	81
5.2	The percolation model Q_n	84
5.3	Market data	89
5.3.1	The Large Pool Model	89
5.4	Percolation model parameters	94
5.5	Constraining the bond percolation parameter	97
5.6	Comparing the percolation model and copula models to market data	103
6	Summary and future work	107
7	Appendix A	112
7.1	Comparing Student t4 and Gaussian copulas	112
7.2	Computing base correlations from expected losses	114
8	Appendix B	117
8.1	A function to generate the N by N matrix $B_{1,n}$	117
8.2	A function to generate the best ρ_0 and <i>pperc</i> parameters .	118
8.3	A function to generate a, b, c given <i>pperc</i> and ρ_0 . This function also outputs the percolation model expected losses	121

List of Figures

1.1	The construction of a tranching CDO from a number of individual credits.	5
3.1	Plot of the default probability distribution P_n for $N = 125$ names where $\rho = 0$ and $p = 0.05$ using Monte Carlo simulation for a Gaussian copula model. As results between 60 and 125 are extremely small we only plot the default probability distribution P_n up to $n = 60$	30
3.2	Plot of the default probability distribution P_n for $N = 125$ names where $\rho = 0.4$ and $p = 0.05$ using Monte Carlo simulation for a Gaussian copula model.	30
3.3	Plot of the default probability distribution P_n for $N = 125$ names in a portfolio where $\rho = 0$ and $p = 0.05$ using the binomial distribution for a Gaussian copula model.	32
3.4	Plot of the default probability distribution P_n for $N = 125$ names in a portfolio where $\rho = 0.4$ and $p = 0.05$ using the theoretical approach for a Gaussian copula model.	35
3.5	Plot of the probability of default P_n for $N = 125$ names in a portfolio where $\rho = 0$ and $p = 0.05$ using the Binomial distribution and a Monte Carlo approach.	36
3.6	Plot of the probability of default P_n for $N = 125$ names in a portfolio where $\rho = 0.4$ and $p = 0.05$ using the theoretical default distribution and a Monte Carlo approach.	36

3.7	As the number of degrees of freedom in a Student t distribution is increased from 3 through 4 to 5 (red, green and yellow curves) it is seen that the probability density functions begin to closer resemble that of a Gaussian distribution (blue).	38
3.8	Plot of the probability of default P_n for $N = 125$ names in a portfolio where $\rho = 0$ and $p = 0.05$ using Monte Carlo simulation for a Student t4 copula model.	43
3.9	Plot of the probability of default P_n for $N = 125$ names in a portfolio where $\rho = 0.4$ and $p = 0.05$ using Monte Carlo simulation for a Student t4 copula model.	44
3.10	Plot of the probability of default P_n for $N = 125$ names in a portfolio where $\rho = 0$ and $p = 0.05$ using the theoretical default distribution for a Student t4 copula.	45
3.11	Plot of the probability of default P_n for $N = 125$ names in a portfolio where $\rho = 0.4$ and $p = 0.05$ using the theoretical default distribution for a Student t4 copula.	46
3.12	Plot of the probability of default P_n for $N = 125$ names in a portfolio where $\rho = 0$ and $p = 0.05$ using the theoretical default distribution and Monte Carlo approach.	47
3.13	Plot of the probability of default P_n for $N = 125$ names in a portfolio where $\rho = 0.4$ and $p = 0.05$ using the theoretical default distribution and Monte Carlo approach.	47
3.14	Plot of the probability of default P_n for $N = 125$ names in a portfolio where $\rho = 0.1$ and $p = 0.05$ using the Gaussian and Student t4 copulas.	49
3.15	Plot of the probability of default P_n for $N = 125$ names in a portfolio where $\rho = 0.4$ and $p = 0.05$ using the Gaussian and Student t4 copulas.	49
3.16	Plot of the probability of default P_n for $N = 125$ names in a portfolio where $\rho = 0.7$ and $p = 0.05$ using the Gaussian and Student t4 copulas.	50

3.17	Plot of the probability of default P_n for $N = 125$ names in a portfolio where $\rho = 0.9$ and $p = 0.05$ using the Gaussian and Student t4 copulas.	50
3.18	Plot of the probability of default P_n for $N = 125$ names in a portfolio where $\rho = 0.1$ and $p = 0.15$ using the Gaussian and Student t4 copulas.	52
3.19	Plot of the probability of default P_n for $N = 125$ names in a portfolio where $\rho = 0.1$ and $p = 0.45$ using the Gaussian and Student t4 copulas.	52
3.20	Plot of the probability of default P_n for $N = 125$ names in a portfolio where $\rho = 0.1$ and $p = 0.75$ using the Gaussian and Student t4 copulas.	53
3.21	Plot of the probability of default P_n for $N = 125$ names in a portfolio where $\rho = 0.1$ and $p = 0.95$ using the Gaussian and Student t4 copulas.	53
3.22	Plot of the expected loss for the five tranches of the iTraxx Europe on the 12 th July 2007. Included in the plot are the best fitting (single correlation) Gaussian and Student t4 copula cases. Note the poor agreement of the Gaussian and good agreement (except for senior tranches) of the Student t4 copula.	60
4.1	A network is a collection of vertices connected by edges.	62
4.2	Comparing Newman's finite and infinite models when $N = 10$ and probability of default $p = 0.09$	74
4.3	Comparing Newman's finite and infinite models when $N = 40$ and probability of default $p = 0.024$. Note the errors are due to numerical calculation of binomial coefficients - this could be improved by using a more careful method of computing the binomial coefficients. We don't pursue this in our thesis but it could be an area for further work.	76
4.4	Comparing Newman's finite and infinite models when $N = 70$ and probability of default $p = 0.0143$	77

4.5	Comparing Newman’s finite and infinite models when $N = 125$ and probability of default $p = 0.008$	79
5.1	The fraction of optimists a that minimises the error function for the percolation model. The market data is made up of 5Yr, 7Yr and 10Yr iTraxx Europe tranche fixings.	97
5.2	The fraction of percolationists b that minimises the error function.	97
5.3	The fraction of catastrophists c that minimises the error function.	98
5.4	The bond percolation parameter $pperc$ that minimises the error function. Note, the relationship between $pperc$ and z is $z = (pperc)(N)$, as given in Section 5.1.	98
5.5	The parameter ρ_0 that minimises the error function. Note, the relationship between ρ_0 and z_1 is $z_1 = (\rho_0)(N - 1)$, as given in Section 5.1.	99
5.6	The function MSE is the sum of the squares of the difference between the market expected loss EM_j and the percolation model expected losses E_j for each tranche j . The smaller the value of MSE the better the fit between the percolation model and market data base correlations. Here we graph the minimum MSE value for all iTraxx tranche fixings.	99
5.7	The fraction of optimists a that minimises the function MSE with the bond percolation parameter $pperc$ constrained. The market data comprises of 5Yr, 7Yr and 10Yr iTraxx Europe tranche fixings.	101
5.8	The fraction of percolationists b that minimises the error function MSE with the bond percolation parameter $pperc$ constrained.	101
5.9	The fraction of catastrophists c that minimises the function MSE with the bond percolation parameter $pperc$ constrained.	102

5.10	The bond percolation parameter $pperc$ that minimises the function MSE with the bond percolation parameter $pperc$ constrained. Note, the relationship between $pperc$ and z is $z = (pperc)(N)$ as given in Section 5.1.	102
5.11	The $rho0$ parameter that minimises the function MSE with the bond percolation parameter $pperc$ constrained. Note, the relationship between ρ_0 and z_1 is $z_1 = (\rho_0)(N - 1)$ as given in Section 5.1.	103
5.12	The function MSE is the sum of the squares of the difference between the market expected loss EM_j and the percolation model expected losses E_j for each tranche j . The smaller the value of MSE the better the fit between the percolation model and market data base correlations. Here we graph the minimum error function MSE with the bond percolation parameter $pperc$ constrained.	104
5.13	Comparing the percolation model, Gaussian and Student t4 copulas to market data. This market data set is an iTraxx Europe tranching CDO maturing in 5 years.	105
5.14	Comparing the percolation model, Gaussian and Student t4 copulas to market data. This market data set is an iTraxx Europe tranching CDO maturing in 7 years.	105
5.15	Comparing the percolation model, Gaussian and Student t4 copulas to market data. This market data set is an iTraxx Europe tranching CDO maturing in 10 years.	106

List of Tables

4.1	Polynomials $P(k 10)$ for values for k up to 10	72
4.2	Results for the infinite and finite network models varying the default probability p when $N = 10$	74
4.3	Results for the infinite and finite network models varying the default probability p when $N = 40$	75
4.4	Results for the infinite and finite network models varying the default probability p when $N = 70$	76
4.5	Results for the infinite and finite network models varying the default probability p when $N = 125$	78

Chapter 1

Introduction

1.1 General introduction

In recent years the market for credit derivatives has developed rapidly with the introduction of new products. Along with the introduction of new products comes the issue of how to price them. Credit derivatives are among the most risky and complex products and have been criticised as a contributing factor in the recent credit crisis (Crouhy et al., 2008). Much of the blame for the current credit crisis is being attributed to the mathematical models and quantitative methods associated with credit derivatives. Collateralised Debt Obligations (CDOs) are a particular type of credit derivative whose value and payments are derived from a portfolio of underlying assets. Developing alternative pricing models of CDOs is the main focus of this thesis.

In this thesis we review two standard methodologies for pricing CDO tranches. These are both factor copula models, one with a Gaussian distri-

bution and the other based on a Student t distribution. We then propose a new percolation model based on network theory for use in pricing CDO tranches. We introduce this fundamental model and then fit the model to market data. We examine the possible replacement of current copula based approaches with intuitive contagion models for percolation on finite networks.

The idea of using percolation theory to model the probability of default in credit derivative type products, i.e. that defaults trigger other defaults, is similar to the spread of diseases in epidemiology where diseased individuals are in contact with other individuals and disease spreads. Epidemiology is a well established theory in the network community and in this thesis we examine the use of percolation theory to match to empirical data of a CDO, where the idea of defaults triggering other defaults is analogous to disease-spread on a network.

This thesis is structured as follows: first we introduce CDOs and review previous literature on the pricing of these products; then we review both the Gaussian and Student t copula models; next we introduce our new pricing framework and formulae; finally we present our results and discuss our conclusions.

1.1.1 Introduction to CDOs

A collateralised debt obligation (CDO) is a security backed by a pool of assets (bonds, loans, credit default swaps, etc.) which are packed together as a portfolio and then tranced (where tranching is the structuring of

the assets into a number of different classes of notes, ranking them by seniority). CDOs contain a pool of underlying instruments (called collateral) against which notes of debt are issued with varying cash flow priority. These notes differ in credit quality depending on the subordination level. When each note is issued, it usually receives a rating from an independent agency (Standard & Poor's, Moody's, etc.). Typically, the collateral of a CDO is a portfolio of corporate bonds or bank loans or other financial products (lending, leasing, residential or commercial mortgages, even other credit derivatives, etc.). As such, CDOs form a customised asset class by allowing a variety of investors to share the risk and return of an underlying portfolio of debt obligations (Meneguzzo and Vecchiato, 2004).

The main economic reason for the existence of CDOs is to address market imperfections. Firstly, the CDO market frees regulatory capital by allowing bankers and other financial institutions to sell some portion of their assets, therefore reducing the amount of regulatory capital they must hold. Secondly, individual bonds or loans may be illiquid and this leads to a reduction in their market values. Securitization may improve liquidity, and thereby raise the total valuation to the issuer of the CDO structure (Duffie and Garleanu, 2001).

1.1.2 How a CDO works

A Special Purpose Vehicle (SPV) is set up to hold the portfolio of assets and issues notes with different subordination, called tranches that it sells to investors. CDOs are generally issued by SPVs to allow for the

transfer of risk from the originator to an entity. ¹These SPVs are usually bankruptcy remote, thinly capitalised and isolated from any credit risk related to the originator. The SPV of a CDO issues multiple classes of equity and debt that are tranching with respect to seniority in bankruptcy and timing of repayment (Lucas, 2001). Each tranche has an attachment point/percentage A and detachment point/percentage D . For simplicity of discussion we will present an example of a CDO portfolio made up of bonds only. If the cumulative percentage loss of the portfolio of bonds reaches the attachment percentage of a given tranche then investors in that tranche start to lose their principal and when the cumulative percentage loss of principal reaches the detachment percentage, the investors in the tranche lose their entire principal and no further loss can occur to them. These tranches are ordered so that losses in principal or interest of the collateral are absorbed by the junior (lowest) tranche first and then in order by the next tranche and so on. The equity tranche, also called the first loss position, is the most junior tranche in the CDO's capital structure and is the first to be affected by a default in the portfolio. It supports the risk of payment delays and credit losses first in order to make more senior tranches less credit-risky (Lucas, 2001). Figure 1.1 shows the basic structure of a CDO backed by a portfolio of bonds. Senior CDO notes are paid before mezzanine and lower subordinated notes, with any

¹We note that SPVs, along with the credit rating agencies, have been implicated as contributing factors in the recent credit crisis and that perhaps instead of decreasing risk, these CDOs may have increased the credit risk, with these effects not being picked up by the credit rating agencies. This is an area that would benefit from further work, however possible causes of the credit crisis are not the focus of this thesis.

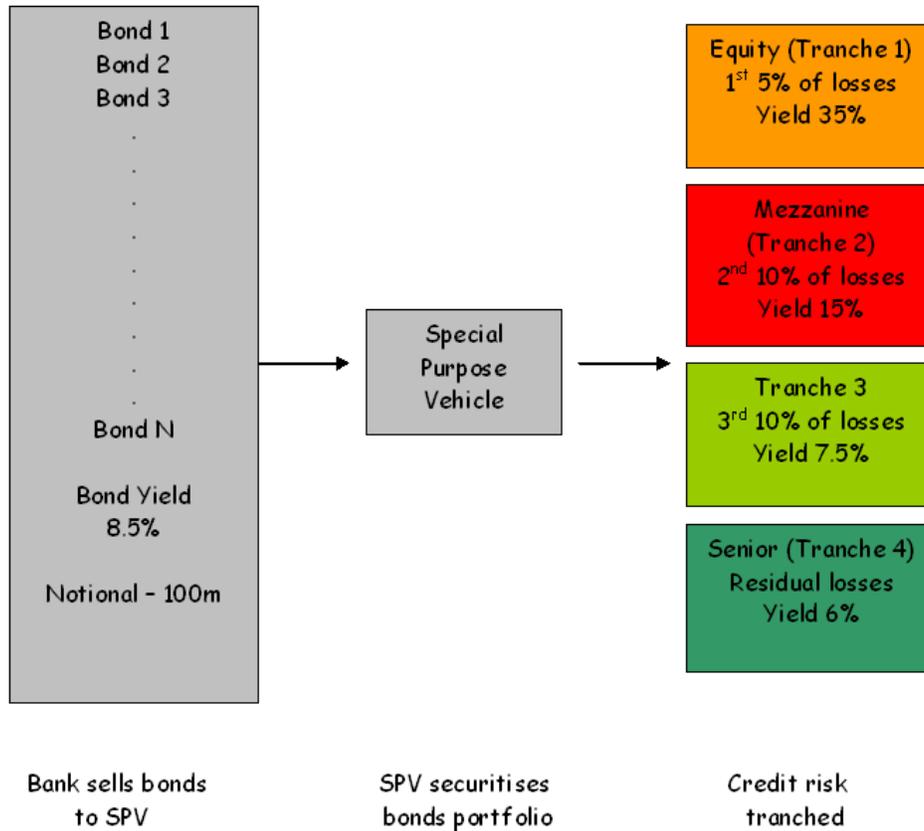


Figure 1.1: The construction of a tranching CDO from a number of individual credits.

residual cash flow being received by the equity tranche. In practice we can say that the CDO investors are selling protection by selecting a tranche according to their risk appetite and return preference.

Subordinated CDO debt tranches protect more senior debt tranches against credit losses, are less risky and therefore moving down the CDO's tranches from senior to equity, the risk level increases. As discussed previously, each note except for the equity, is rated by credit rating agencies,

with senior notes usually having Triple-A or Double-A ratings. In such a way CDOs create a customised asset class by allowing various investors to share the risk and return of an underlying portfolio of debt obligations.

As an example consider a CDO with four tranches created from a portfolio of bonds. Suppose that the total notional value is 100 million with the capital structures as shown in Figure 1.1. The first/equity tranche of securities has 5% of the total bond portfolio and absorbs all credit losses from the portfolio during the life of the CDO up to the value of 5 million. If at maturity the total default loss is less than 5 million, only the equity tranche is affected. By definition, the equity tranche has an attachment percentage at 0% and is the most junior tranche in a CDO portfolio. The second tranche has 10% of the principal and covers the cumulative loss during the life of a CDO in excess of 5% (its attachment point) and up to a maximum of 15% (its detachment point). This tranche is called the mezzanine tranche as it is placed between the equity tranche and subsequent more senior tranches. The mezzanine tranche absorbs all losses up to maturity in excess of 5 million and up to 15 million. If the total default loss is between 5 and 15 million the equity tranche holders do not get their principal back and the mezzanine tranche holders get only part of theirs. The third tranche has 10% of the principal and absorbs all losses in excess of 15% and up to 25% of the CDO's portfolio. If the loss is between 15 and 25 million then the equity and mezzanine tranche holders do not get their principal back and the third tranche holders receive only part of theirs back. The fourth tranche has 75% of the principal and

absorbs losses in excess of 25 million. As described above, junior tranches in the CDO absorb the first losses to occur, whereas defaults on the bond portfolio must exceed 25% before the holders of the senior tranche are responsible for any credit losses (Hull and White, 2004).

The yields in Figure 1.1 are the rates of interest paid to tranche holders. These rates are paid on the balance of the principal remaining in the tranche after losses have been paid. In Figure 1.1 it can be seen that as we move from low risk (senior) tranches to high risk (equity) tranches the interest rates on the tranches increase.

In this thesis we don't explicitly examine mechanisms that cause defaults in a CDO portfolio but possible examples could be the idea that firms hold bonds or assets that are linked in a network or another example could be the idea that demand for products would drop when linked firms would go bankrupt. This an area that would benefit from further work.

1.1.3 Types of CDOs

In financial markets, depending on how the SPV gains credit risk, CDOs are categorised into two types: cashflow CDOs and synthetic CDOs. A CDO is referred to as a cashflow CDO if the SPV owns the underlying debt obligations, in other words if the SPV acquires the credit risk exposure by purchasing debt obligations. In contrast the CDO is referred to as a synthetic CDO if the SPV does not own the debt obligations and instead obtains the credit risk exposure by selling credit default swaps (CDSs) on the debt obligations of the names in the portfolio. In this thesis the models

we review and the new model we propose can be applied to both cashflow CDOs and synthetic CDOs.

1.2 Literature review

There is rapidly growing research on credit derivatives and correlated defaults and in the following review we will introduce some of the predominant models used to price CDOs. The key input into pricing CDO tranches is an approximation of the default dependence between the underlying assets in the CDO portfolio. We review several types of models used to price CDOs and begin with the current market standard.

1.2.1 Copula model

Hull and White (2004) reveal that an important problem of CDO pricing is modelling default correlation between the underlying assets of a portfolio. Prior to this Li (2000) introduced the Gaussian copula model to CDO pricing to capture “the time-until-default” and applied Sklar’s theorem (Sklar, 1959) to link together two marginal distributions to form a joint distribution. This joint distribution is a copula function. This model allows one to calculate the loss distribution of every tranche in a portfolio. Li’s Gaussian copula model computes the default correlation between the assets in the portfolio but requires Monte Carlo simulation techniques for the default correlation structure to be completely general. However, the computational complexity of Monte Carlo simulations in calculating the

Gaussian copula model increases as the number of names in the portfolio increases. To deal with this problem the factor copula method (Andersen et al. (2003), Hull and White (2004) and Gregory and Laurent (2005)) was introduced.

1.2.2 Factor copula model

The general theory behind the one factor copula model is that credit events of different names in a portfolio are independent conditional on a common factor. In the case of a one factor Gaussian (Normal) copula this leads to integrals in the pricing formula being computed analytically (O’Kane and Schloegl, 2005). Due to the simplicity of the one factor Gaussian copula it has since become the market standard (Burtschell et al., 2007).

Burtschell et al. (2009) examine a fundamental problem of the one factor Gaussian copula. If one calculates the correlations that are implied by the market prices of tranches of the same CDO using the Large Homogeneous Pool (LHP) approach (Vasicek, 1987), one does not get the same correlation over the whole structure. Instead there is a reported “correlation smile” in the CDO market analogous to the concept of volatility smile of the Black-Scholes option pricing model. The cause of the correlation smile generated by the Gaussian copula is because the model cannot produce a single correlation parameter that makes all tranche spreads equal to the market quotes. The root cause of this problem was firstly identified in the copula function literature as the lacking of tail dependence in the Gaussian copula (Gregory and Laurent, 2004). Tail dependence in a

copula can describe the increased joint probability of defaults and can be measured by a tail index. The Gaussian copula model has a low tail index of zero whereas other copulas, such as the Student t , have positive tail indices. Many researchers propose alternative approaches to the Gaussian copula to bring about more tail dependence in the model (Kalemanova et al., 2007).

Andersen and Sidenius (2005) present two analytically tractable extensions to the Gaussian copula with the aim of matching their results to market data. They extend the Gaussian one factor model by randomising recovery rates and allowing for correlation between recovery rates and defaults. In a second extension, they introduce random factor loadings to allow for higher correlation in economic depressions. From their results, these extensions are capable of producing correlation skews similar to those observed in the market. Burtschell et al. (2007) present another extension to the one factor Gaussian copula model by way of a simple class of stochastic correlation models.

One other natural extension to the one factor Gaussian copula is to use a distributional assumption that produces heavy tails. The Student t distribution has fatter tails than the Gaussian distribution and is known to generate tail dependence in the joint distribution for the factors. These are two properties that are also observed in the market and hence the Student t copula is presented as an extension to the Gaussian copula (Andersen et al. (2003) and Schloegl and O’Kane (2005)). We will review the Gaussian and Student t copula models in Section 3.1 and 3.2 respectively. Hull and

White (2004) consider a one factor double t copula for the pricing of CDOs. The underlying aspect of this approach relies on combining Student's t random variables in a copula model. Hull and White find that the one factor double t copula model fits market prices reasonably well for most tranches in a CDO when using the Student's t distribution with 4 degrees of freedom. The Marshall-Olkin copula is yet another class of copula and stems from the multivariate compound Poisson process (Andersen and Sidenius (2005) and Giesecke (2003)). In this model individual defaults are constructed from a series of independent common shocks and has led to a dramatic fattening of the tail.

Burtschell et al. (2009) present a comparative analysis of the Gaussian copula, Student t copula model, double t factor model, stochastic correlation extensions to Gaussian copula, Clayton copula and the Marshall-Olkin copula to model dependency between default times. They provide their results for calibrating these copula approaches to market spreads of DJ iTraxx data and show that Student t and Clayton copula give results very similar to the Gaussian copula model. Burtschell et al. (2009) find that the Marshall-Olkin copula led to dramatic tail fattening and their results from the double t factor model, stochastic correlation and the factor loading were closer to market prices. Note that the fundamental idea here is that there is a link between default probability and market prices for CDOs and it is this link that is critical to the pricing of CDO portfolios. We examine further the relationship between pricing and default probabilities in Chapter 3.

A set of other factor models based on different distributions have been proposed. Moosbrucker (2006) uses a one factor Variance Gamma model, Kalemanova et al. (2007) works with a Normal Inverse Gaussian (NIG) factor model and Baxter (2006) introduce the Brownian Variance Gamma (BVG) model. Moosbrucker (2006) results show that the model leads to a consistent way of pricing bespoke tranches on CDO portfolios. Kalemanova et al. (2007) develop a factor copula approach based on NIG distributions. The employment of the NIG distribution not only brings considerable improvement with respect to computation times but also more flexibility in the modelling of the dependence structure in a portfolio.

Hull and White (2006) present the implied copula approach, a simple and appealing alternative to the Gaussian copula model. This approach shows how CDO quotes can be used to imply a probability distribution for the hazard rate over the life of a CDO. Basically, instead of implying the copula correlations from market prices they imply the copula itself. As an extension to Hull and White (2006), Hull and White (2008b) develop a parametric version of this implied copula model that can be used to value bespoke CDOs. Rosen and Saunders (2009) provide a valuation framework based on the application of multifactor credit models in conjunction with Monte Carlo techniques. The framework can be seen as an extension of the implied copula methodology of Hull and White (2006), where more general multi-factor credit portfolio models can be used. These multifactor models allow for the inclusion of sector and geographical concentrations and potentially other risk factors such as recoveries and prepayments.

Glasserman and Suchintabandid (2007) propose a reduced form model for pricing CDOs through a series of independent-obligor model, i.e. models involving only independent defaults. This approximation uses the multifactor Normal Copula model to capture the default correlation between obligators. Their approach reduces the problem of pricing a model with correlated defaults to calculations involving only independent defaults. As an alternative to the market standard copula, Hofert and Scherer (2008) present a default model based on nested Archimedean copulas. This copula is able to capture the hierarchical dependence structures among the obligators in a credit portfolio. It describes the dependence between names in the portfolio by an inner copula and the sectors in a portfolio via an outer copula.

Further extensions to the Gaussian copula have been presented by Schönbucher and Schubert (2001). They present a new approach to incorporating dynamic dependency in intensity based default risk models. The model uses an arbitrary default dependency structure which is specified by the copula of times to default. This approach is combined with individual intensity-based models for the defaults of the obligators without loss of the calibration of the individual default intensity based models. These features of the model make it ideally suited to the analysis of the pricing and hedging of CDOs. The factor copula models address some of the shortcomings of the market standard Gaussian copula. Further, there has been much research into alternative methods for pricing CDOs and some are presented in the following section.

1.2.3 Alternatives to copula models

To model default dependency, several varieties of structural models have been proposed. Hull et al. (2005) present a structural model similar to that in Black and Cox (1976) for valuing correlation-dependent credit derivatives and suppose that the correlation between the asset values of different companies is driven by one or more common factors. A base case model is introduced where the asset correlation and recovery rate are both constant. They extend the base case in the following three ways: firstly they allow the asset correlation to be stochastic and correlated with default rates, secondly they assume that the recovery rate is stochastic and correlated with default rates, and thirdly a mixture of these two processes. This leads to both the common factors and idiosyncratic factors having distributions with heavy tails.

The use of saddlepoint approximation is recommended by Yang (2006) and Antonov et al. (2005) in the context of CDO pricing. Yang (2006) presents a second order saddlepoint method which offers generally superior performance in computing the loss tranche function and CDO prices for large credit portfolios. Hull and White (2008a) propose a simple dynamic model that assumes the hazard rate (default rate) of a company has a deterministic drift with periodic impulses. The impulse size plays a similar role to default correlation in the Gaussian copula model. The model is computed with ease and can be represented by a binomial tree. Sidenius et al. (2008) present the so-called SPA model, a framework to modelling the dynamics of portfolio default losses specified by a two layer process.

In the absence of information about default time, the 1st layer models the dynamics of portfolio loss distributions. The 2nd layer models the loss process itself as a Markov process and is conditional on the path taken by the background process.

Intensity based models have also been introduced for pricing CDO tranches. The modelling framework was first proposed by Duffie and Garleanu (2001) and the estimated model allows heterogeneous default intensity dynamics as in Mortensen (2006). Brigo et al. (2007) present a model based on a generalised Poisson process. In contrast to alternative approaches to loss modelling, this model allows for more than one default in small time intervals. Papageorgiou and Sircar (2009) also present framework based on the intensity based model. They propose improvements via incorporating fast mean reverting stochastic volatility in the default intensity process and by considering homogeneous groups within the original set of firms. Their model is calibrated to market data and includes market data during the credit crunch crisis.

1.2.4 Correlation smile and base correlations

The liquidity of standardised CDO indices prior to the credit crisis led to the market convention of pricing tranches in terms of implied correlations. This was to facilitate a comparison of prices across tranches. The implied compound correlation of a tranche is the correlation value that makes the tranche value, computed by the standard market model, equal to its observed market value. In general, the implied correlation is found to be

higher in equity and senior tranches than for mezzanine tranches. This is known as the “correlation smile”. Implied correlations suffer two other major problems (McGinty et al., 2004b). Since mezzanine tranches are not monotonic in correlation, implied correlations might not be unique. Also, implied correlations depend on the upper and lower tranche attachment points and as a consequence they cannot be interpolated to price other tranches with different attachment points.

McGinty et al. (2004b) propose the so-called Base Correlations as a way of solving such problems and this has since become an industry standard method. Base correlations are the correlation inputs required for a sequence of first loss tranches that give the tranche values consistent with quoted spreads of a standardised CDO structure using a Large Homogeneous Pool Model. Due to its monotonicity it can be easily interpolated and used to price off market CDO tranches with the same collateral pool. In contrast to the implied correlation approach, the base correlation framework does not display the “smile” but a meaningful and well defined correlation skew. We further discuss the base correlation framework in Section 3.4.

The thesis is organised as follows: the second chapter presents the one factor copula model. The third chapter introduces the Gaussian copula and Student t4 copula for use in pricing CDOs. Further we compare these copulas in their ability in pricing CDO tranches. The fourth section includes network theory and the finite and infinite network models. The fifth chapter focuses on our new percolation model and compares this

model to other copula models in its ability to match to market data. The last chapter concludes.

Chapter 2

One factor copula model

As previously discussed in Section 1.2 the critical input in the pricing of a CDO is an estimate of the default dependence (default correlation) between the underlying names in a portfolio. Dependency modelling with copula functions, introduced by Li (2000), has become a market standard in the pricing of CDOs and was further extended with the introduction of the factor copula model (Andersen et al. (2003), Hull and White (2004) and Gregory and Laurent (2005)).

A one factor copula models default dependence through linear correlation of random variables x_i , where x_i represents the value of an individual name in a portfolio, with a single global factor M . In this chapter we provide a general description of the one factor copula function and present the methodology needed to compute the default probability distribution P_n , where P_n is the probability of n defaults of the N names in the portfolio up to time T .

2.1 The factor copula approach

In a one factor copula framework we model the value of name i (x_i) as a linear combination of two random variables,

$$x_i = \sqrt{\rho_i}M + \sqrt{1 - \rho_i}Z_i, \quad i = 1, 2, \dots, N. \quad (2.1.1)$$

In this equation, M is the global factor common to all N names in the portfolio and Z_i is the idiosyncratic factor specific to name i . Both M and Z_i have variance equal to 1 and the covariances are in fact correlations, thus explaining why we later call ρ_i the correlation factor for each $i = 1, 2, \dots, N$. Also both M and Z_i have independent zero mean distributions such that idiosyncratic factors Z_i for example could represent innovations of a firm or the quality of the management of a company.

The correlation factor ρ_i satisfies $0 \leq \rho_i \leq 1$ and the default correlation between x_i and x_j is $\sqrt{\rho_i \rho_j}$ for $(i \neq j)$.² The correlation value ρ_i determines the extent to which x_i is correlated with the movements of the global factor M and the relative sizes of the idiosyncratic and global factors are controlled by the correlation value ρ_i . For example, if $\rho_i = 0$ then the business cycle of a particular name is not influenced by the global factor M , whereas if $\rho_i = 1$ then the global factor M is the only driver of defaults for name i .

²Note only positive correlation is used in the model and this is so because empirically only positive correlations are observed in the market. Therefore the fact that this model does not allow for negative correlation is not a major drawback.

2.1.1 Large homogeneous pool

By making assumptions on the nature of the underlying names in the portfolio, the framework previously presented can be easily used to price CDOs. We consider a large homogeneous portfolio (LHP) approximation on a factor copula model where the portfolio consists of N identical names (where the limit $N \rightarrow \infty$ may be considered as a special case). This LHP limit approximation, employing the Law of Large Numbers, was first proposed by Vasicek (1987). Under the LHP approximation all names have the same marginal probability of default (p) (and consequently identical threshold values ($K_i = K$)), correlation value ($\rho_i = \rho$), notional (ntl) and recovery rate (R). This simplifies Eq. (2.1.1) to,

$$x_i = \sqrt{\rho}M + \sqrt{1 - \rho}Z_i. \quad (2.1.2)$$

From this point onwards we will adopt the LHP approach.

To price CDOs using this one factor copula model presented previously we first need to calculate the probability distribution for the losses in a given portfolio. In Section 3.5 we use these loss probabilities to calculate the expected losses in any tranche of the portfolio.

2.2 The distribution of defaults

In this section we will first explain what we consider to be a default when dealing with a name's value and then set out the probability distribution of default conditional on a common factor M . Using these instruments we

can compute the default probability distribution P_n .

2.2.1 Threshold value K

A default is triggered when a name's value x_i drops below a certain threshold value K , that is $x_i < K$ (Lehrert et al., 2005). The probability of default is then $Pr(x_i < K)$. The value of K is determined with reference to $C_x(K) = Pr(x_i < K)$, the cumulative distribution function of x_i . We calibrate $C_x(K)$ to be exactly p , the marginal probability that name i has defaulted, such that,

$$C_x(K) = p, \tag{2.2.1}$$

where p is given. We then solve Eq. (2.2.1) to compute the threshold value K , such that $K = C_x^{-1}(p)$. Now that we have a model for the correlated movements of the value x_i for each name and a criterion to determine whether a default is deemed to have happened given those movements, we can then write down the conditional default probability function d_M .³

2.2.2 Conditional default probability

Similar to Hull and White (2004) we look to compute d_M , the conditional probability that a name i has defaulted given a value for the global factor M . Given the value for M , we can write the probability of default of the

³Note that generally speaking the inverse of the cumulative distribution C_x does not have to be injective. In this thesis the threshold value K will be the appropriate end point of the interval given by $C_x^{-1}(p)$.

individual name as,

$$\begin{aligned} Pr(x_i < K \mid M) &= Pr(\sqrt{\rho}M + \sqrt{1-\rho}Z_i < K \mid M) \\ &= Pr\left(Z_i < \frac{K - \sqrt{\rho}M}{\sqrt{1-\rho}} \mid M\right). \end{aligned} \quad (2.2.2)$$

Eq. (2.2.2) can be explained by the fact that given a value for the global factor M a default occurs when the idiosyncratic factor Z_i goes below a certain level. Then from Eq. (2.2.2) the cumulative distribution function of the probability that an individual name will default given the level of the global factor can be written as,

$$d_M = C_Z\left(\frac{K - \sqrt{\rho}M}{\sqrt{1-\rho}}\right), \quad (2.2.3)$$

where C_Z is the cumulative distribution function of the Z_i 's.

2.2.3 Default probability distribution P_n

The final step in implementing the one factor copula model requires the unconditional default probability distribution. Having previously defined the conditional probability of default, we can deduce the number of conditional defaults among the N names given different values of the common factor M . The number of conditional defaults, where trials are identical, is given by the binomial distribution. The total defaults that have occurred up to time T conditional on the global factor M is given by,

$$\binom{N}{n} d_M^n (1 - d_M)^{(N-n)}. \quad (2.2.4)$$

The expression in Eq. (2.2.4) is integrated with respect to the probability measure of M to obtain P_n , the probability that there will be exactly n defaults by time T ,

$$P_n = \int \binom{N}{n} d_M^n (1 - d_M)^{(N-n)} P_M(M) dM, \quad (2.2.5)$$

where $P_M(M)$ is the probability density function of the random variable M . The integral is carried out using numerical approximation with the M range being truncated and divided into intervals of equal length ΔM ,

$$\begin{aligned} P_n &= \binom{N}{n} \int d_M^n (1 - d_M)^{(N-n)} P_M(M) dM \\ &= \binom{N}{n} \sum_j d_{M_j}^n (1 - d_{M_j})^{(N-n)} (C_M(M_{j+1}) - C_M(M_j)) \\ &\approx \binom{N}{n} \Delta M \sum_j d_{M_j}^n (1 - d_{M_j})^{(N-n)} P_M(M_j). \end{aligned} \quad (2.2.6)$$

for $n = 0, \dots, N$ and where the approximation decreases in error as the number of intervals in the M range are increased.⁴ Eq. (2.2.6) gives us the default probability distribution P_n , the probability that there are n defaults up to time T .

In this chapter we have introduced the fundamental ideas behind the one factor copula approach and discussed how the calculations for the model can be computed using a large homogeneous pool approximation. In the next chapter we review the market standard one factor Gaussian and Student t copula models where the M and Z_i are factors with Gaussian

⁴It may be possible to give an error bound on this approximation but this is outside the scope of this current work.

and Student's t probability distributions respectively. Then using the probability of default distribution P_n for each distribution we compute the base correlations and expected losses for each tranche in the portfolio.

Chapter 3

One factor copula models with different underlying distributions

In this chapter we review the Gaussian and Student t copula models for use in modelling default dependence in a CDO portfolio. To compute the probability default distribution for both the Gaussian and Student t copulas we implement both a Monte Carlo and theoretical approach. We then compare the copulas in their ability to fit to market data.

Note that to compute the default probabilities in the copula models we use the pricing measure as our standard approximation. The pricing measure used is not the same as the physical probability measure with the main difference being that the physical probability measure is a historical measure of CDO prices whereas the pricing measure is implied from the

CDS spreads of the names in the CDO portfolio. Therefore the calculations we present in this chapter are implied from the CDS spreads and in turn give the market-implied probabilities of default for each name in the CDO portfolio, effectively using the pricing measure.

Next we present the base correlation approach as an alternative to quoting implied correlations tranche by tranche. We use the default distribution P_n for the Gaussian and Student t copulas to compute the expected value of each tranche in a CDO portfolio.

3.1 The one factor Gaussian copula

In the previous chapter we presented the methodology for a one factor copula model without specifying which distributions might be used for the random variables M and Z_i . Li (2000) was the first to suggest that the Gaussian copula can be employed in credit risk modelling to estimate the correlation default. In the one factor Gaussian copula framework we model the name i value as,

$$x_i = \sqrt{\rho}M + \sqrt{1 - \rho}Z_i, \quad i = 1, 2, \dots, N, \quad (3.1.1)$$

where both the global factor M and idiosyncratic factor Z_i are Gaussian random variables such that:

- M is a normally distributed random variable with zero mean and unit variance, i.e. $M \sim N(0, 1)$.

- Z_i are independent normally distributed random variables independent of M , i.e. $Z_i \sim N(0, 1)$.
- Hence, $x_i \sim N(0, 1)$. Due to the stability of the normal distribution under convolution, x_i also follows a standard normal distribution.

The Gaussian copula became popular in the pricing of CDOs as it is easy to draw random samples from it and correlated Gaussian random variables are well known. Also, the dependency structure in the Gaussian copula is well understood and is to some degree tractable. For these reasons this model has become the market standard in the pricing of CDOs. Details on the implementation of the one factor Gaussian copula using both Monte Carlo techniques and a theoretical approach are discussed in Sections 3.1.2 and 3.2. In order to implement both approaches we first need to compute the threshold value K for a Gaussian copula model.

3.1.1 Threshold value K

As discussed in Section 2.2.1, default of name i is considered to have occurred at some time up to T if the stochastic variable x_i lies below a threshold value K . To compute the threshold K for the Gaussian copula requires the normal cumulative distribution function. The cumulative distribution function of the standard normal distribution is denoted by Φ and is expressed as,

$$\Phi(x) = \frac{1}{2} \left[1 + \operatorname{erf} \left(\frac{x}{\sqrt{2}} \right) \right], \quad (3.1.2)$$

where $\text{erf}\left(\frac{x}{\sqrt{2}}\right)$ is the error function of $\left(\frac{x}{\sqrt{2}}\right)$. We calibrate Eq. (3.1.2) to reflect the individual probability of default p and do this by setting the threshold value K to a level which replicates the given probability of default such that,

$$\Phi(K) = p.$$

In other words,

$$\frac{1}{2} \left[1 + \text{erf} \left(\frac{K}{\sqrt{2}} \right) \right] = p.$$

We let $f(K)$ be the function,

$$f(K) = \frac{1}{2} \left[1 + \text{erf} \left(\frac{K}{\sqrt{2}} \right) \right] - p, \quad (3.1.3)$$

and for a given value of p , we solve for K when $f(K) = 0$. In MATLAB we use the `fzero` command to compute the value of K when Eq. (3.1.3) changes sign. Having computed the threshold value K , we now can compute P_n , the default probability distribution for a Gaussian copula. We will first implement the Gaussian copula model using Monte Carlo techniques.

3.1.2 Monte Carlo techniques to compute P_n

Monte Carlo methods are a class of computational algorithms that rely on repeated random sampling to compute results. The idea is that when the number of simulations is sufficiently large, it gives a good approximation of the true value of average quantities. Monte Carlo simulation techniques

can be used to estimate the default distribution P_n for a Gaussian copula. These are based on repeated trials of random defaults with an assumed correlation structure for the copula model.

In MATLAB we use the `randn` command to generate pseudorandom numbers drawn from a standard normal distribution. The `randn` function generates a random variable common to all names in the portfolio for the global factor M and generates an array of size N of random variables for the idiosyncratic factors Z_i . The decision as to whether a name in the portfolio has defaulted or not can be modelled by drawing a value for the random variable x_i and checking whether x_i lies above or below the threshold value K . Default occurs if x_i lies below the threshold K and does not occur if it lies above.

From the results we show later, 5000 simulations is sufficient to give results matching with reasonable accuracy to the analytical solutions. We compute the number of occurrences of the values that fall between 0 and N . Therefore the probability of 0 defaults, 1 default, 2 defaults, etc. are the number of occurrences of each event divided by the total number of simulations such that,

$$\sum_{n=0}^N P_n = 1,$$

where P_0 is the probability of 0 defaults, P_1 is the probability of 1 default, P_2 is the probability of 2 defaults, etc. We choose N to be 125 for our portfolio as one of the most widely traded indices is the iTraxx Europe index which comprises of 125 reference names. In Figures 3.1 and 3.2 we plot the default probability distribution results for a Gaussian copula

model using Monte Carlo simulation.

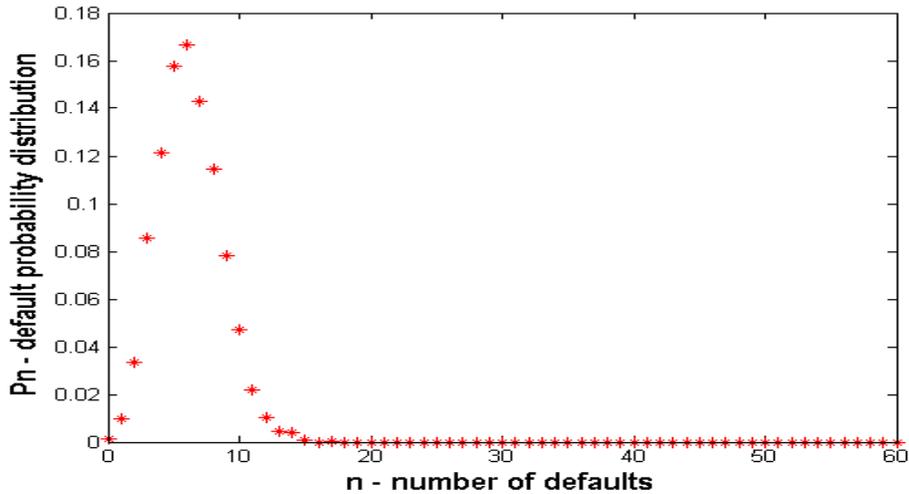


Figure 3.1: Plot of the default probability distribution P_n for $N = 125$ names where $\rho = 0$ and $p = 0.05$ using Monte Carlo simulation for a Gaussian copula model. As results between 60 and 125 are extremely small we only plot the default probability distribution P_n up to $n = 60$

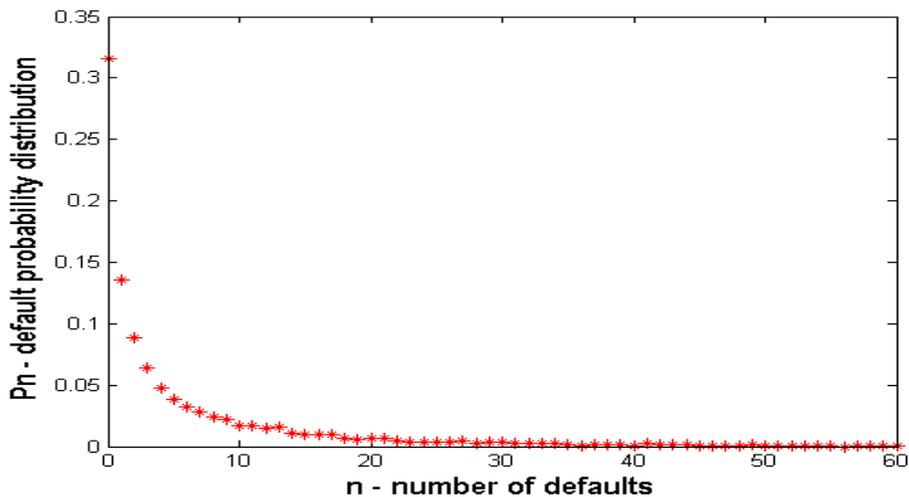


Figure 3.2: Plot of the default probability distribution P_n for $N = 125$ names where $\rho = 0.4$ and $p = 0.05$ using Monte Carlo simulation for a Gaussian copula model.

Shortcomings of Monte Carlo simulation

Monte Carlo simulation is computationally expensive, especially when simulations are large. For results to be repeatable we need the number of realizations here to be at least 5000. Due to Monte Carlo method shortcomings it is useful to compare with the theoretical approach to computing the probability default distribution P_n . We first consider the case where the correlation between names in a portfolio is zero and then extend the approach described in Section 3.1.4 to include a Gaussian distribution when $\rho > 0$.

3.1.3 Theoretical default distribution when $\rho = 0$

If the correlation value between names in a portfolio is zero ($\rho = 0$) then each name in the portfolio defaults independently of the others. In this case, defaults of different names in the portfolio are independent and occur with probability p , thus for a portfolio of N names the loss distribution for the portfolio can be described by the binomial distribution.

An experiment that follows a binomial distribution will satisfy the following conditions:

- The experiment consists of N identical trials.
- Each trial has two possible outcomes, success (default) or failure (no default).
- The trials are independent so that the outcome of one trial has no effect on the outcome of another.

Hence, the binomial distribution perfectly models the probability of default in a CDO portfolio where each name defaults independently of the others. Then, considering computing the probability of default for N names in a portfolio, let x be the number of defaults observed having default probability p . The binomial distribution gives the probability of observing $n \leq N$ defaults as,

$$P[x = n] = P_n = \binom{N}{n} p^n (1 - p)^{(N-n)}. \quad (3.1.4)$$

This equation gives the theoretical probability of default P_n when the

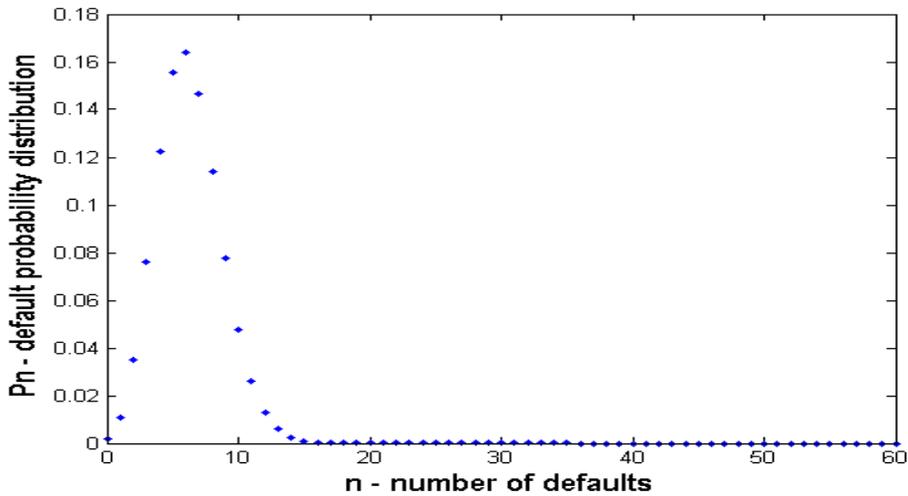


Figure 3.3: Plot of the default probability distribution P_n for $N = 125$ names in a portfolio where $\rho = 0$ and $p = 0.05$ using the binomial distribution for a Gaussian copula model.

correlation value between names in a portfolio is zero. In Figure 3.3 we plot the probability of default results for a one factor Gaussian copula using the Binomial distribution when $\rho = 0$. In order for us to compute the probability of default in the general case when $\rho > 0$ we generalise this

$\rho = 0$ method by computing a binomial distribution such that defaults are independent conditional on the global factor M .

3.1.4 Theoretical default distribution for $\rho > 0$

To compute the theoretical default distribution P_n when $\rho > 0$, we use the fact that defaults are independent conditional on the global factor M . As previously described in Section 2.2, to compute the default probability distribution P_n we first compute the conditional default probability d_M and probability density function $P_M(M)$ for the global factor M .

In a Gaussian copula the conditional default probability d_M that a name i has defaulted given a value for M is given by $\Phi\left(\frac{K - \sqrt{\rho}M}{\sqrt{1-\rho}}\right)$ where Φ is the cumulative normal distribution previously defined in Eq. (3.1.2). The probability density function $P_M(M)$ for a Gaussian distribution is,

$$P_M(M) = \frac{1}{\sqrt{2\pi}} \exp\left(\frac{-M^2}{2}\right). \quad (3.1.5)$$

Then given a value for M (say $\in [-4, 4]$) and having computed d_M , the probability of having exactly n of N names default is given by the binomial distribution,

$$\binom{N}{n} d_M^n (1 - d_M)^{(N-n)}, \quad i = 0, 1, \dots, N. \quad (3.1.6)$$

The use of the binomial distribution here is due to the default independence after conditioning on M . To obtain the probability of default P_n , the expression in Eq. (3.2.7) is integrated with respect to the probability

measure of M . This integral is carried out using numerical approximation with the M range being divided into intervals of equal length ΔM . The probability default distribution for a Gaussian copula model is given by

$$\begin{aligned}
P_n &= \int \binom{N}{n} d_M^n (1 - d_M)^{(N-n)} \frac{1}{\sqrt{2\pi}} \exp\left(\frac{-M^2}{2}\right) dM \\
&= \binom{N}{n} \sum_j d_M^n (1 - d_M)^{(N-n)} (\Phi(M_{j+1}) - \Phi(M_j)) \\
&\approx \binom{N}{n} \frac{1}{\sqrt{2\pi}} \Delta M \sum_j d_{M_j}^n (1 - d_{M_j})^{(N-n)} \exp\left(\frac{-M_j^2}{2}\right) \quad (3.1.7)
\end{aligned}$$

for $n = 0, \dots, N$ and where the approximation above is good and decreases in error as the number of intervals in the M range is increased.⁵ Thus, Eq. (3.1.7) gives us the theoretical default distribution P_n for a Gaussian copula model. See Figure 3.4 for results using this theoretical approach to computing the probability of default, P_n .

3.1.5 Comparing Monte Carlo and theoretical approaches

Using the steps outlined previously we implement both the Monte Carlo and theoretical approaches to computing the probability of default P_n in a Gaussian copula using MATLAB. We compare the Monte Carlo and theoretical case for various default probabilities p and correlation values ρ . As expected, the theory and Monte Carlo approaches match closely, see Figure 3.6. We also look at the case when correlation $\rho = 0$ and compare

⁵It may be possible to give an error bound on the approximation but this is outside the scope of the current work.

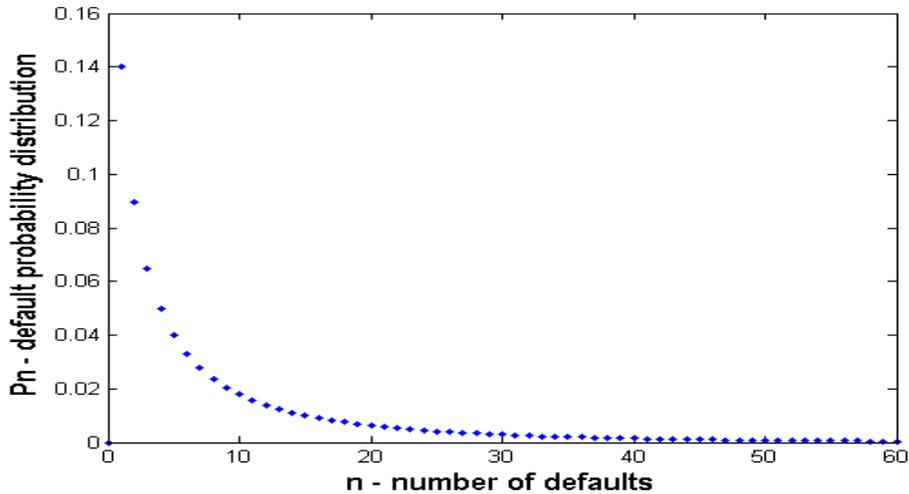


Figure 3.4: Plot of the default probability distribution P_n for $N = 125$ names in a portfolio where $\rho = 0.4$ and $p = 0.05$ using the theoretical approach for a Gaussian copula model.

the Monte Carlo approach to the Binomial case and note they display good fit for varying p , see Figure 3.5. From our results it is clear that theoretical default and Monte Carlo approaches match closely for varying p and ρ values, provided that the number of realisations used in the Monte Carlo method is sufficiently large.

As mentioned in Section 3.1.2, Monte Carlo simulation is expensive in terms of computational resources and time and to achieve good accuracy for the Gaussian copula model we need simulations to have 5000 realisations or more. For this reason, continuing onwards we will use the theoretical approach as our preferred method to model the Gaussian copula. While the Gaussian copula is convenient and intuitive it has a number of shortcomings as a model for real CDOs.

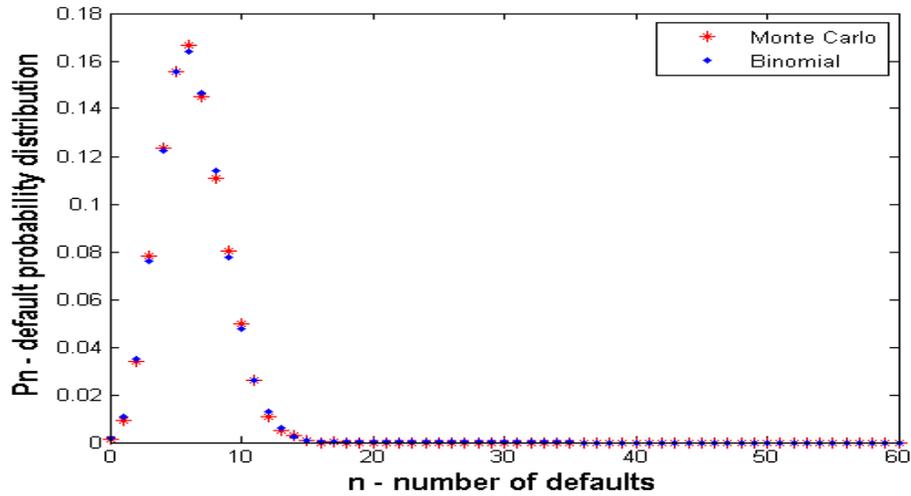


Figure 3.5: Plot of the probability of default P_n for $N = 125$ names in a portfolio where $\rho = 0$ and $p = 0.05$ using the Binomial distribution and a Monte Carlo approach.

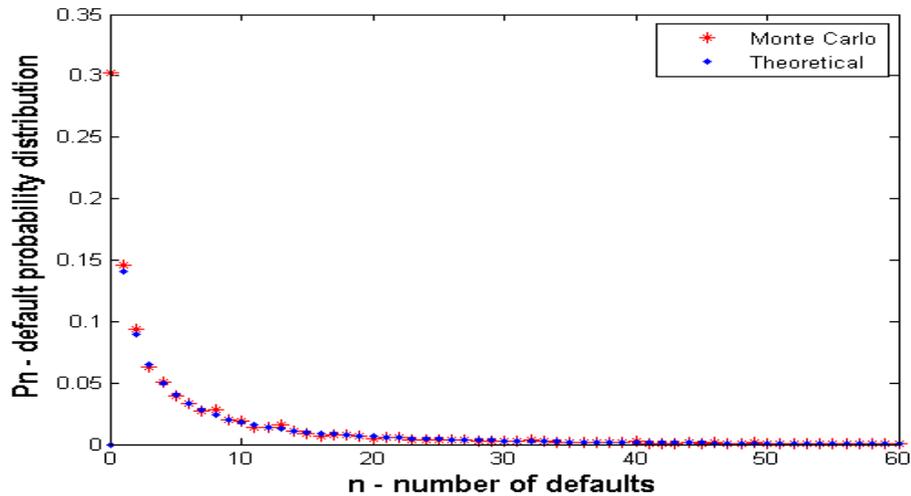


Figure 3.6: Plot of the probability of default P_n for $N = 125$ names in a portfolio where $\rho = 0.4$ and $p = 0.05$ using the theoretical default distribution and a Monte Carlo approach.

3.1.6 Shortcomings of the Gaussian copula

The one factor Gaussian copula fails to fit the prices of different CDO tranches simultaneously and instead there is a reported “correlation smile”.

Many researchers explain this phenomenon with the lack of tail dependence of the Gaussian copula (Kalemanova et al., 2007). Hull and White (2004) suggest the Student t copula as an improvement on the Gaussian copula because it has heavier tails than the normal distribution and better fits to market data than the Gaussian copula. In this thesis we assume a homogeneous portfolio in the pricing of CDOs but note that this LHP assumption can be relaxed to allow for default probabilities p_i different for each name in the portfolio i.e. the heterogeneous case, see Hull and White (2004).

In this section we have introduced and implemented the Gaussian copula model for use in CDO pricing. Following Hull and White's suggestion, we introduce in the next section the Student t copula as a better fit to market data.

3.2 Student t copula

Since the Gaussian copula was introduced to CDO pricing many attempts have been made to improve the fitness of the model to pricing CDO tranches. As mentioned previously in Section 1.2 the Student t copula is one of the most popular copulas studied as an alternative to the Gaussian copula and the main reason for this is the ability of the Student t copula to better capture the phenomenon of dependent extreme values, which is often observed in financial return data (Hull and White, 2004).

The Student t distribution makes up a family of distributions parameterised by a degree of freedom and possessing power-law tails. These are heavier tails than those of the Gaussian distribution and it is these heavier

tails that capture better the phenomenon of dependent extreme values. As the number of degrees of freedom increase, the Student t distribution approaches a Gaussian distribution. Thus the Student t distribution allows one to see how the output of the factor model is affected as one incrementally moves from Gaussian to heavier tailed distributions (see Figure 3.7). For our Student t copula model we look at a Student t copula with four

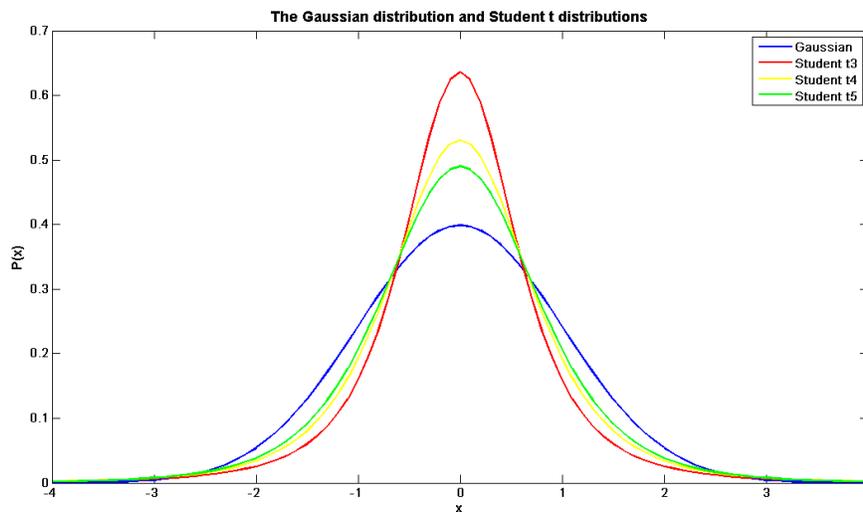


Figure 3.7: As the number of degrees of freedom in a Student t distribution is increased from 3 through 4 to 5 (red, green and yellow curves) it is seen that the probability density functions begin to closer resemble that of a Gaussian distribution (blue).

degrees of freedom and denote this by Student t4. We choose four degrees of freedom as it was highlighted in Hull and White (2004) that this value of t replicates the market values better than other choices. In a one factor copula model the variable x_i (whose value is compared to a threshold value

K to determine whether name i has defaulted) is given by,

$$x_i = \sqrt{\rho}M + \sqrt{1 - \rho}Z_i, \quad i = 1, 2, \dots, N. \quad (3.2.1)$$

In a Student t4 copula both the global factor M and idiosyncratic factor Z_i are Student t4 random variables such that:

- M is a Student t4 distributed random variable with zero mean and unit variance.
- Z_i are zero mean, unit variance Student t4 distributed random variables independent of M .

We seek to compute P_n , the probability distribution for the number of names having defaulted by time T and to do this we firstly need to compute the threshold value K for a Student t4 copula.

3.2.1 Threshold value K

Computing the threshold K for a Student t4 copula involves the Student t4 cumulative distribution function $C_x(K)$. To compute $C_x(K)$, the probability that a random variable x_i is less than the threshold K , we need:

- P_M , the unit variance probability density function for the market factor M from a Student t4 distribution;
- P_Z , the unit variance probability density function for the random variables Z_i from a Student t4 distribution;

- $C_z(Z)$, the Student t4 cumulative distribution function for the random variables Z_i .

For a Student t4 copula the probability density function $P_M(M)$ with unit variance is,

$$P_M(M) = \frac{3}{(2 + M^2)^{\frac{5}{2}}}, \quad (3.2.2)$$

and similarly for Z_i ,

$$P_Z(Z) = \frac{3}{(2 + Z^2)^{\frac{5}{2}}}. \quad (3.2.3)$$

We want to compute the probability that x_i is less than the threshold K with M fixed, i.e. $Pr(x_i < K | M)$. Then substituting Eq. (3.2.1) for x_i ,

$$\begin{aligned} Pr(x < K | M) &= Pr(\sqrt{\rho}M + \sqrt{1-\rho}Z < K | M) \\ &= Pr\left(Z < \frac{K - \sqrt{\rho}M}{\sqrt{1-\rho}} \middle| M\right) \\ &= C_Z\left(\frac{K - \sqrt{\rho}M}{\sqrt{1-\rho}}\right) \\ &= C_Z(Z), \end{aligned}$$

where $Z = \frac{K - \sqrt{\rho}M}{\sqrt{1-\rho}}$. We compute the cumulative distribution function C_Z by integrating Eq. (3.2.3),

$$\begin{aligned} C_Z(Z) &= \int_{-\infty}^Z P_Z(Z) dZ \\ &= \frac{1}{2} + \frac{1}{2} \left(\frac{3Z + Z^3}{(2 + Z^2)^{\frac{3}{2}}} \right). \end{aligned}$$

Then, to compute the probability that x_i is less than the threshold K $Pr(x_i < K)$ we average over all possible M values to get,

$$Pr(x_i < K) = \int_{-\infty}^{\infty} C_Z(Z)P_M(M)dM.$$

The integral is carried out using numerical approximation with the M range being truncated and divided into intervals of equal length ΔM ,

$$C_x(K) = Pr(x_i < K) \approx \sum_j C_Z(Z)P_M(M_j)\Delta M. \quad (3.2.4)$$

We calibrate $C_x(K)$ to reflect the individual probability of default (p) and do this by setting the threshold K to a level which replicates the given probability of default such that,

$$C_x(K) = p.$$

We let $f(K)$ be the function,

$$f(K) = C_x(K) - p, \quad (3.2.5)$$

where we solve for K when $f(K) = 0$ for a given p and ρ value. Having computed the threshold K , we now look to compute P_n , the probability default distribution for a Student t4 copula. We will first implement the Student t4 copula using Monte Carlo techniques.

3.2.2 Monte Carlo methods to compute P_n

As previously stated in Section 3.1.2, Monte Carlo simulations are a class of computational algorithms that rely on repeated random sampling to compute its results. Monte Carlo simulation techniques can be used to estimate the default distribution P_n for a Student t4 copula based on repeated trials of random defaults with an assumed correlation structure for the model.

In MATLAB, we use the `trnd` function to generate random numbers from a Student t distribution specifying four degrees of freedom for a Student t4 distribution. The `trnd` function generates a random variable common to all names in the portfolio for the global factor M and generates an array of size N of random variables for the idiosyncratic factors Z_i . The global and idiosyncratic variables are to have unit variance and we scale the outputs of the `trnd` function by $\frac{1}{\sqrt{2}}$ to account for this.

The decision as to whether a name in the portfolio has defaulted or not can be modelled by drawing a value for the random variable x_i and checking whether the value of x_i lies above or below the critical threshold K . Default occurs if x_i lies below the threshold K and does not occur if x_i lies above K . From the results we show later, 5000 simulations is sufficient to give results matching reasonably accurately to analytical results.⁶ We then compute the number of occurrences of the values that fall between 0 and N where the probability of 0 defaults, 1 default, 2 defaults, etc. are

⁶It is also possible to derive a confidence interval to motivate the number of drawings used in the Monte Carlo simulations but this is outside the scope of this current work.

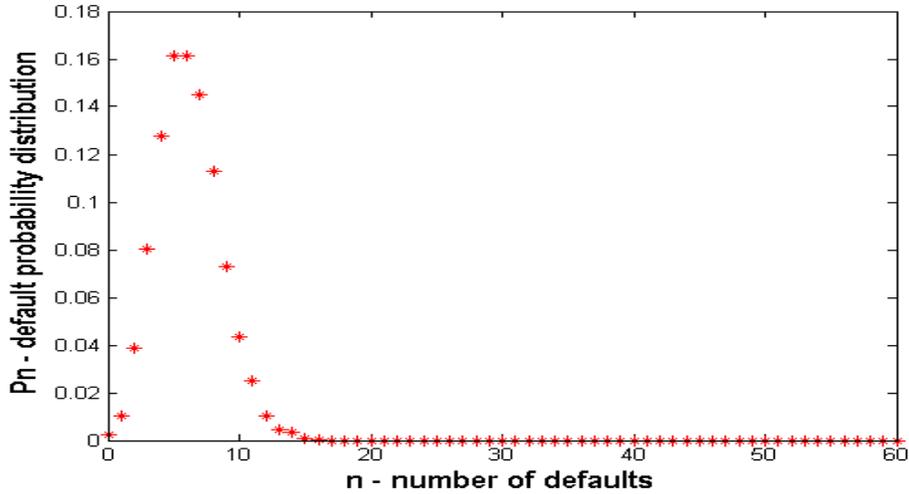


Figure 3.8: Plot of the probability of default P_n for $N = 125$ names in a portfolio where $\rho = 0$ and $p = 0.05$ using Monte Carlo simulation for a Student t4 copula model.

the occurrences divided by the total number of simulations such that,

$$\sum_{n=0}^N P_n = 1.$$

See Figure 3.8 and 3.9 for the probability of default results using Monte Carlo simulations for a Student t4 copula model.

As discussed in Section 3.1.2, Monte Carlo simulation is computationally expensive and therefore the theoretical approach to computing the probability default distribution P_n is described below.

3.2.3 Theoretical default distribution

As previously described in Section 2.2 to compute the default probability distribution P_n we first compute the conditional default probability d_M

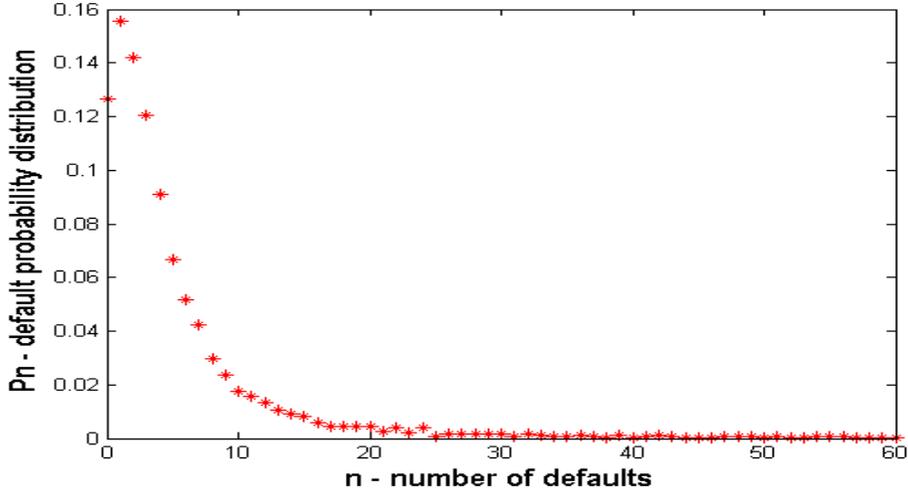


Figure 3.9: Plot of the probability of default P_n for $N = 125$ names in a portfolio where $\rho = 0.4$ and $p = 0.05$ using Monte Carlo simulation for a Student t4 copula model.

and probability density function $P_M(M)$ for the global factor M . In a Student t4 copula, d_M the conditional default probability that a name i has defaulted given a value for M is given by,

$$d_M = \frac{1}{2} + \frac{1}{2} \left(\frac{3Z + Z^3}{(2 + Z^2)^{\frac{3}{2}}} \right), \quad (3.2.6)$$

where $Z = \frac{K - \sqrt{\rho}M}{\sqrt{1-\rho}}$. The probability density function $P_M(M)$ for a Student t4 copula is given by Eq. (3.2.2).

Then given a value for M (say $\in [-20, 20]$) and having computed d_M , the probability of having exactly n of N names default is given by the binomial distribution,

$$\binom{N}{n} d_M^n (1 - d_M)^{(N-n)}, \quad i = 0, 1, \dots, N. \quad (3.2.7)$$

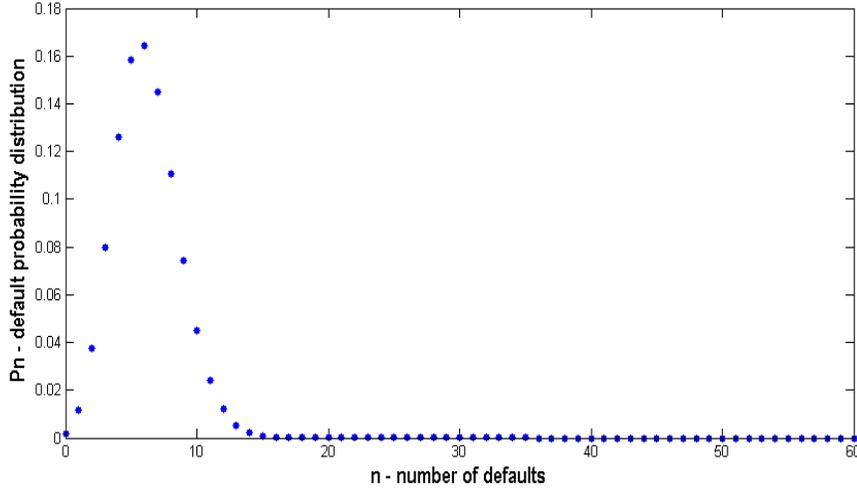


Figure 3.10: Plot of the probability of default P_n for $N = 125$ names in a portfolio where $\rho = 0$ and $p = 0.05$ using the theoretical default distribution for a Student t4 copula.

We range M for a Student t4 copula over a larger range than the Gaussian copula because of the Student t4 distribution having fatter tails. Then we compute P_n similarly to Eq. (2.2.6) such that,

$$\begin{aligned}
 P_n &= \binom{N}{n} \sum_j d_{M_j}^n (1 - d_{M_j})^{(N-n)} \frac{1}{2} \left(\frac{(3M_{j+1} + M_{j+1}^3)}{(2 + M_{j+1}^2)^{\frac{3}{2}}} - \frac{(3M_j + M_j^3)}{(2 + M_j^2)^{\frac{3}{2}}} \right) \\
 &\approx \binom{N}{n} \Delta M \sum_j d_{M_j}^n (1 - d_{M_j})^{(N-n)} \left(\frac{3}{(2 + M_j^2)^{\frac{5}{2}}} \right), \quad (3.2.8)
 \end{aligned}$$

for a Student t4 copula, where the approximation decreases in error as the number of intervals in the M range is increased⁷. Eq. (3.2.8) gives us the theoretical default distribution P_n for a Student t4 copula model. See Figure 3.10 and 3.11 for results using this theoretical approach to

⁷It may be possible to give an error bound on the approximation but this is outside the scope of this current work.

computing the probability of default P_n .

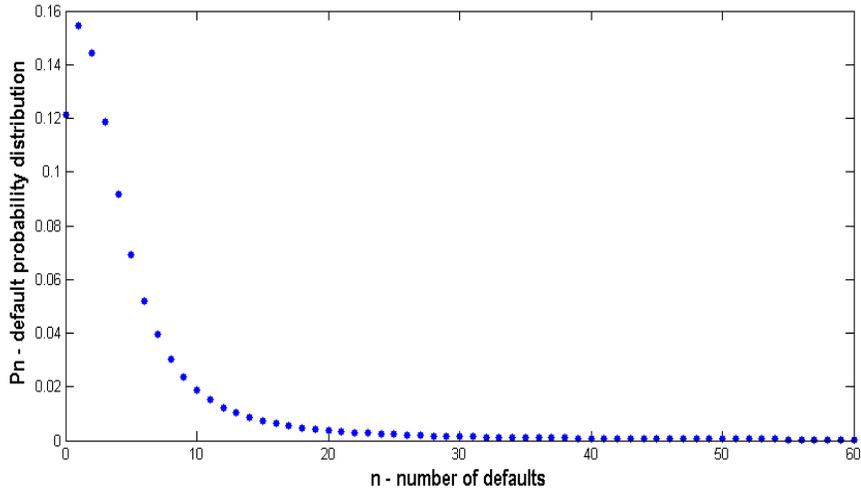


Figure 3.11: Plot of the probability of default P_n for $N = 125$ names in a portfolio where $\rho = 0.4$ and $p = 0.05$ using the theoretical default distribution for a Student t4 copula.

3.2.4 Comparing Monte Carlo and theoretical approaches

Using the steps outlined above, we implement both the Monte Carlo and theoretical approaches to computing the probability of default in a Student t4 copula in MATLAB. We compare both the Monte Carlo and theoretical approach for various default probabilities p and correlation values ρ . As expected, the two approaches match closely, see Figure 3.12 and 3.13.

We discussed previously in Section 3.1.2 that Monte Carlo simulation is computationally expensive and to achieve good accuracy for the Student t4 copula model we need simulations to have 5000 or more realisations.

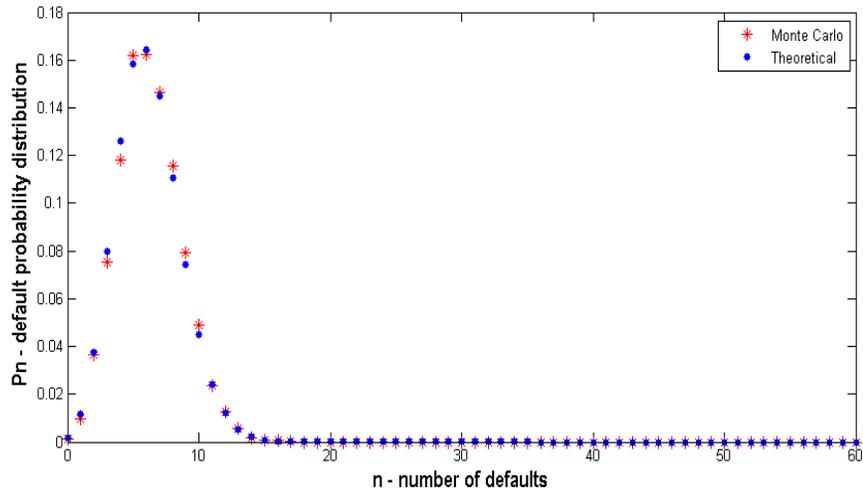


Figure 3.12: Plot of the probability of default P_n for $N = 125$ names in a portfolio where $\rho = 0$ and $p = 0.05$ using the theoretical default distribution and Monte Carlo approach.

For this reason, continuing onwards we will use the theoretical approach as our preferred method to implement the Student t4 copula.

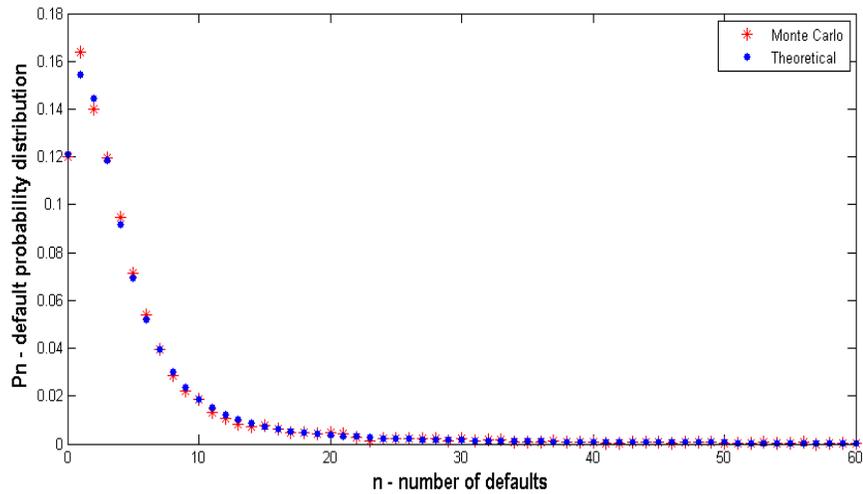


Figure 3.13: Plot of the probability of default P_n for $N = 125$ names in a portfolio where $\rho = 0.4$ and $p = 0.05$ using the theoretical default distribution and Monte Carlo approach.

In this section, we have introduced the Student t4 copula model for use in pricing CDOs as an alternative to the Gaussian copula. In the next section, we compare the Gaussian copula and Student t4 copula and investigate the effect of default probabilities p and correlation ρ on the results for P_n .

3.3 Comparing Gaussian and Student t4 copulas

In this section we compare the Gaussian and Student t4 copula models for different values of p and ρ . Details the algorithm employed are provided in Appendix A. We consider first a CDO portfolio of 125 names and keeping the probability of default constant at 5% we examine the effect of increasing the correlation value for the copula models. From Figure 3.14 to 3.17 it can be seen that as we increase the correlation in the portfolio the probability of one or more defaults declines and the probability of 10 or more increases. In the extreme situation when default correlation between names is one, the names in the portfolio are essentially the same and the global factor M is the driver of defaults. Either they all default (with probability 5%) or none of them default (with probability 95%).

Next we increase the probability of default p , which increases the probability of a given number of names in the portfolio defaulting i.e. $Pr(x_i < K) = p$. Increasing the value of p therefore increases the threshold value K and this results in more names in the portfolio defaulting, see

Figure 3.14 to 3.21.

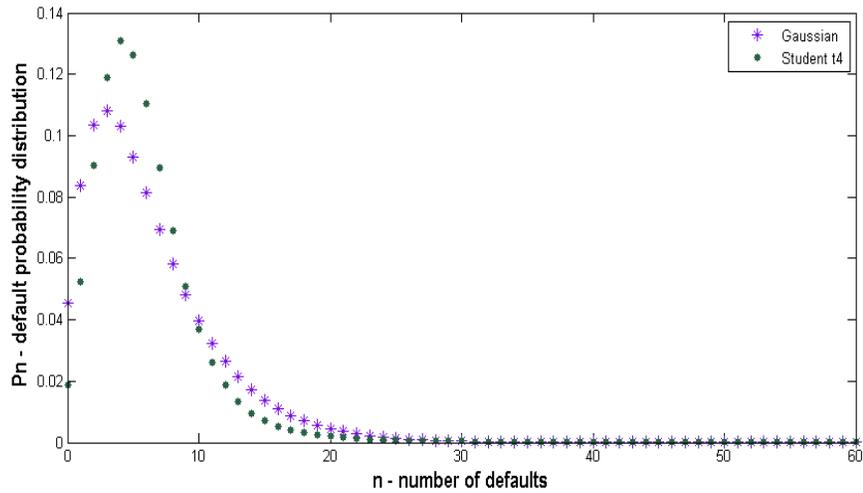


Figure 3.14: Plot of the probability of default P_n for $N = 125$ names in a portfolio where $\rho = 0.1$ and $p = 0.05$ using the Gaussian and Student t4 copulas.

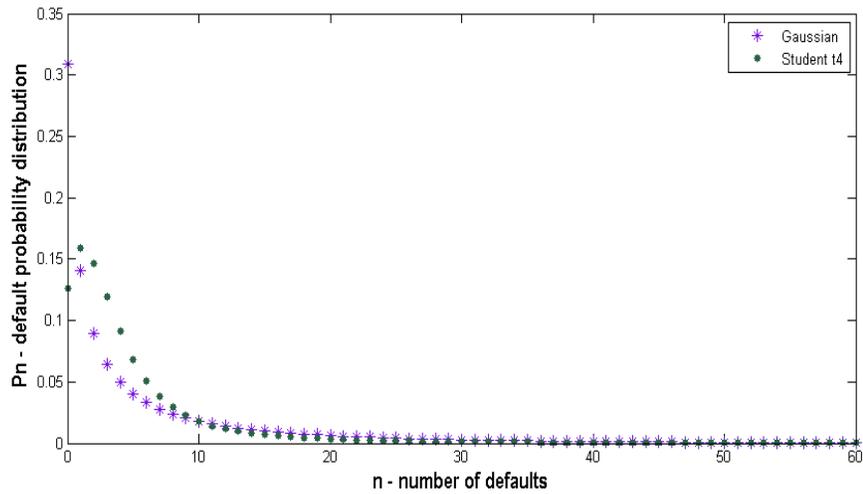


Figure 3.15: Plot of the probability of default P_n for $N = 125$ names in a portfolio where $\rho = 0.4$ and $p = 0.05$ using the Gaussian and Student t4 copulas.

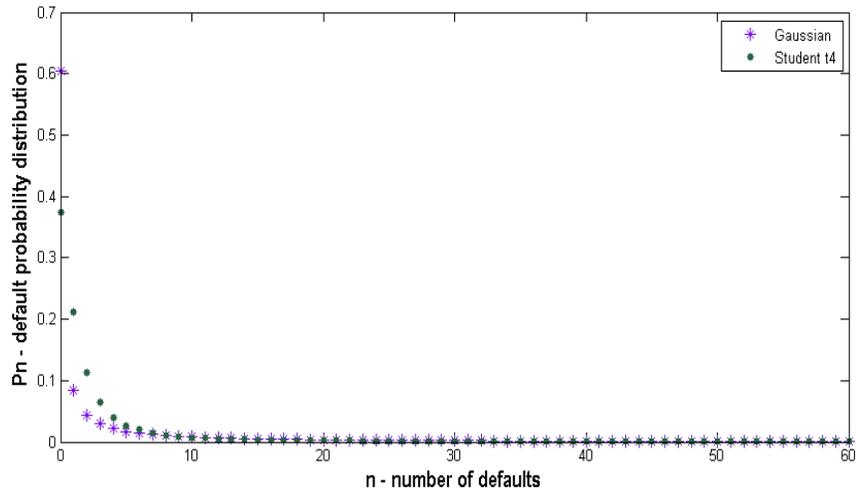


Figure 3.16: Plot of the probability of default P_n for $N = 125$ names in a portfolio where $\rho = 0.7$ and $p = 0.05$ using the Gaussian and Student t4 copulas.

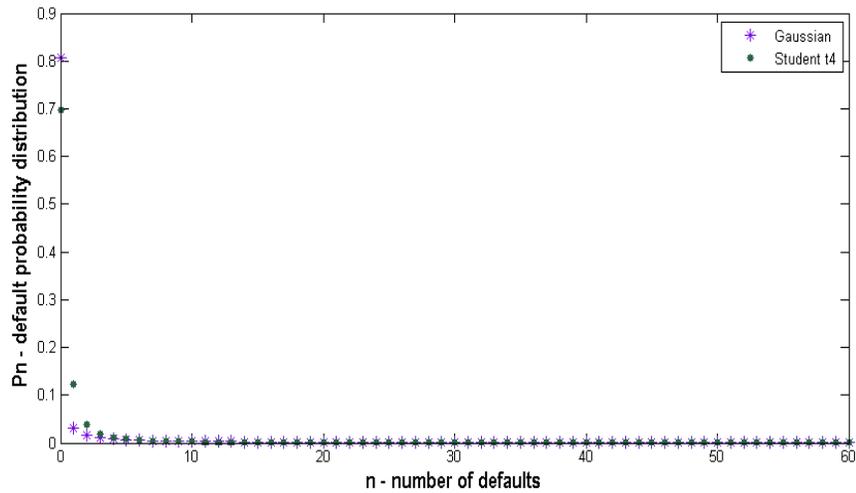


Figure 3.17: Plot of the probability of default P_n for $N = 125$ names in a portfolio where $\rho = 0.9$ and $p = 0.05$ using the Gaussian and Student t4 copulas.

The fundamental difference between the two copula models is the ability of the Student t4 copula to better model tail dependence in the port-

folio. Also, discussed previously in Section 1.2 that a shortcoming of a Gaussian copula model is that when calculating the correlations that are implied by the market prices of tranches of the same CDO using the LHP approach, we do not get the same correlation over the whole structure but observe a “correlation smile” (Kalemanova et al., 2007). The main explanation of this problem is the lack of tail dependence into the model. The Student t4 distribution has fatter tails than the Gaussian distribution and it better generates tail dependence in the portfolio, see Figure 3.14- 3.21. These are two properties that are observed in the market. Thus incorporating tail dependence into the one factor model through the Student t copula model yields significant improvement.

In the following section we introduce the base correlation framework proposed by McGinty et al. (2004b). It has been introduced into the CDO framework to compute correlations embedded in tranche prices and an improved alternative to using implied correlations where there is a reported “correlation smile”.

3.4 Base correlations

Since the coming of standardised tranche CDOs, there has been a need for a commonly agreed method of quoting the implied correlation between the names in the respective portfolios (O’Kane and Livesey, 2004). In this section we review implied correlation methodology and discuss its shortcomings as a method for quoting correlation between names in a portfolio. To avoid the difficulties associated with the implied correlation,

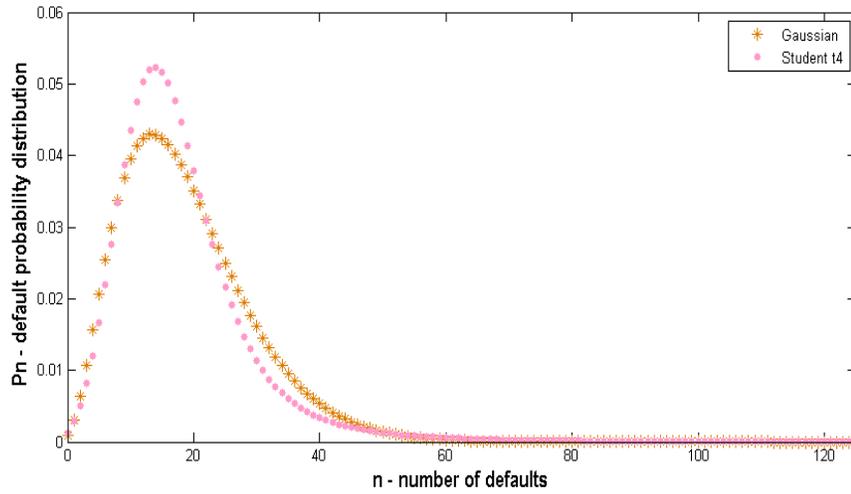


Figure 3.18: Plot of the probability of default P_n for $N = 125$ names in a portfolio where $\rho = 0.1$ and $p = 0.15$ using the Gaussian and Student t4 copulas.

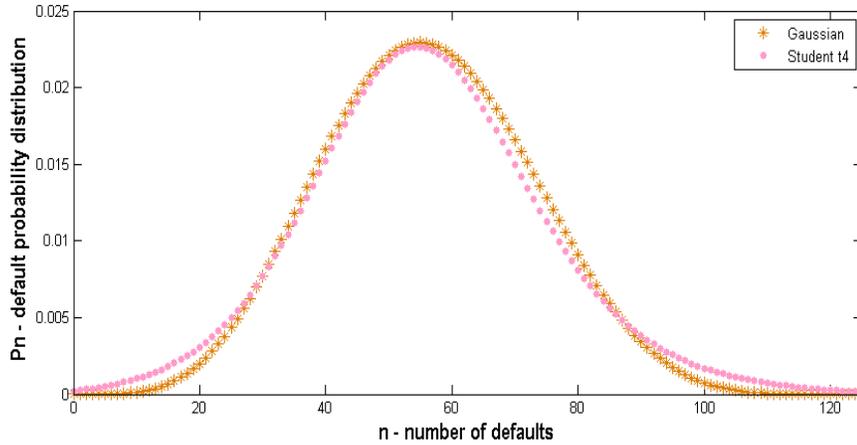


Figure 3.19: Plot of the probability of default P_n for $N = 125$ names in a portfolio where $\rho = 0.1$ and $p = 0.45$ using the Gaussian and Student t4 copulas.

we introduce the modelling framework proposed by JPMorgan to compute the so-called “Base Correlation” (McGinty et al., 2004b) and outline the

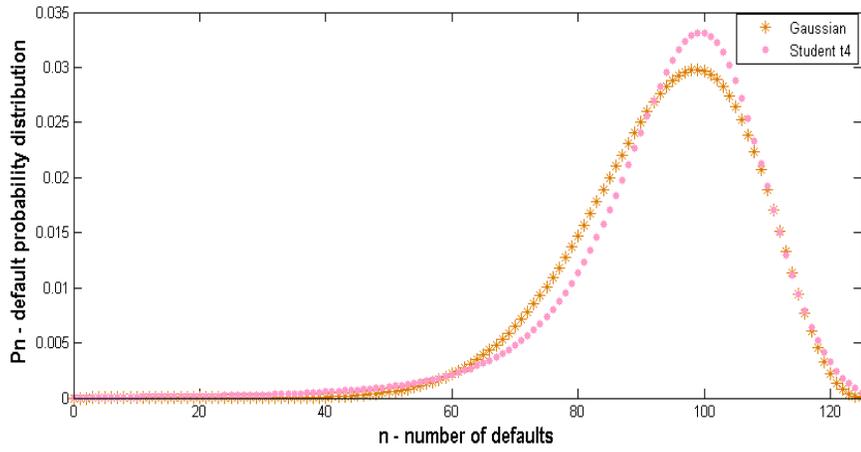


Figure 3.20: Plot of the probability of default P_n for $N = 125$ names in a portfolio where $\rho = 0.1$ and $p = 0.75$ using the Gaussian and Student t4 copulas.

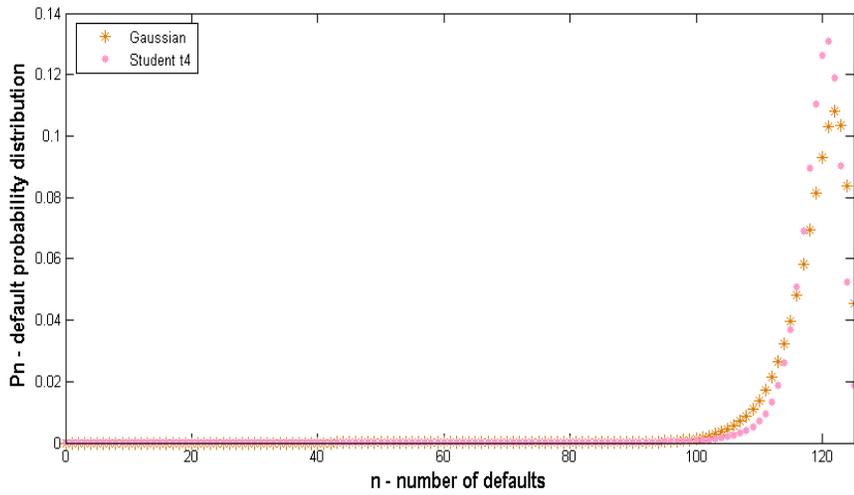


Figure 3.21: Plot of the probability of default P_n for $N = 125$ names in a portfolio where $\rho = 0.1$ and $p = 0.95$ using the Gaussian and Student t4 copulas.

steps in calculating base correlations for CDO tranches.

3.4.1 Implied correlation

The implied correlation of a tranche is defined as the correlation value in a one factor Gaussian copula model that makes the theoretical value of a tranche equal to its market quote. The basic assumption of the one factor copula model is that all names in a portfolio are correlated with the same constant correlation parameter. If this assumption were true then the implied correlation calibrated from market quotes of different tranches would be the same, since each tranche is exposed to the same reference portfolio of names. However, implied correlation for market quoted CDOs are observed to be higher in equity and more senior tranches than for mezzanine tranches and this leads to a “correlation smile” (Hull and White, 2004). This implied correlation approach has two major drawbacks: firstly, calculating implied correlations for mezzanine tranches is problematic as their spread is not a monotonic function of correlation and secondly, it is difficult to price bespoke tranches as implied correlation is not a monotonic function of attachment and detachment points (McGinty et al., 2004a). To avoid the difficulties associated with quoting correlation tranche by tranche the market has chosen base correlations as its quotation convention.

3.4.2 Base correlation

Base correlations are the correlation inputs required for a sequence of first loss tranches to give tranche values consistent with quoted spreads of a standardised CDO structure using a Large Homogeneous Pool Model

(McGinty et al., 2004b). The fundamental idea behind the concept of base correlation is that we can break down all tranches into combinations of base tranches, where a base tranche is simply another name for an equity tranche. The ‘base’ comes from the fact that the subordination of a base tranche is always zero, i.e. it is attached to the base of the loss distribution. The expected loss for an equity tranche detaching at D_i and with attachment point $A_i = 0$ is given by Eq. (3.5.2) (see Section 3.5). Our objective, given market values for E_j is to find the correlation parameter which reproduces Eq. (3.5.2) in a Gaussian copula; this is the base correlation value.

To obtain the base correlation for each first loss tranche we need to use a bootstrapping process. Consider an example with tranche attachment points A_1, A_2, A_3, \dots (3%, 6%, 9%,... for tranching iTraxx Europe) where the base correlation for the first loss $[0, A_1]$ (where 0 is the attachment point and A_1 is the detachment point) tranche is the same as its tranche implied correlation (O’Kane and Livesey, 2004). The process for computing the expected loss (E) for each first loss tranche is as follows:

$$E[0, A_j] = E[0, A_{j-1}] + E[A_{j-1}, A_j] \text{ for } j \geq 2,$$

where $E[0, A_1]$ is the expected loss for the first loss tranche with attachment point 0 and detachment A_1 and $E[A_{j-1}, A_j]$ comes from the market spread on the $[A_{j-1}, A_j]$ tranche. For example, the expected loss on the $[0\%, 6\%]$ tranche is calculated by summing the expected loss on the $[0\%, 3\%]$ tranche and the expected loss on the $[3\%, 6\%]$ tranche. Once the

expected losses for the sequence of first loss tranches are computed, we can iterate to find the single base correlation for each tranche in the portfolio (McGinty et al., 2004a). Details of the algorithm employed is provided in Appendix A.

McGinty et al. (2004b) present the key reasons why base correlations offer advantages over implied correlations. Firstly, embedded in the base correlation framework is a convenient method for pricing off-market CDO tranches with the same collateral pool, secondly, base correlations are unique across the capital structure and thirdly, they define a more meaningful skew than implied correlations. McGinty (2004) implements the base correlations through the LHP model and provides a description of the base correlation model as a spreadsheet. We use this spreadsheet in Section 5.3.1 when computing base correlation values for market data. This framework is also useful for computing base correlations when comparing pricing algorithms. We can do this by taking the Student t4 base correlations and Gaussian base correlations and compare them to the market base correlations. In doing this we seek out copulas whose base correlation curves fit better to those observed by the market and in Chapter 5 we propose a new model based on network contagion. In the following section we use the default probability distributions P_n computed for the Gaussian and Student t copulas to calculate the expected loss for each tranche j .

3.5 Expected loss

The expected loss for tranche j , E_j , is computed by summing the probability of loss x occurring by time T times the loss to tranche j due to portfolio loss of x . Then the expected loss for tranche j with attachment point A_j and detachment point D_j at time T is

$$E_j = \int_{A_j}^N f(x)P(x)dx,$$

where $f(x)$ is the loss to tranche j due to portfolio loss of x such that

$$f(x) = \begin{cases} x - A_j & A_j < x < D_j \\ D_j - A_j & D_j < x < N \\ 0 & \text{otherwise.} \end{cases}$$

and $P(x)$ is the probability of loss x occurring by time T . Therefore the expected loss for tranche j is

$$E_j = \int_{A_j}^{D_j} P(x)(x - A_j)dx + (D_j - A_j) \int_{D_j}^N P(x)dx. \quad (3.5.1)$$

Then using a Large Homogenous Pool approximation, we replace the continuous loss probability distribution $P(x)$ by a discrete distribution P_n where n is the number of defaults and assume the recovery rate R is zero and let N be the total number of names in a portfolio. As mention previously, losses included in a tranche j with an attachment point A_j and detachment point D_j are when $n > A_jN$ and $n \leq D_jN$ i.e. in the range

$A_j N < n \leq D_j N$. Then using Riemann's sum we simplify Eq. (3.5.1) to

$$E_j \approx \sum_{n=\lfloor A_j N \rfloor + 1}^{\lfloor D_j N \rfloor} P_n (n - A_j N) + (D_j - A_j) N \sum_{n=\lfloor D_j N \rfloor + 1}^N P_n. \quad (3.5.2)$$

Now, let us assume that assets have the same recovery rate R . Then, the total loss of the equity tranche of A_j will occur only when assets of the total amount $\frac{A_j}{1-R}$ have defaulted. Thus allowing for homogeneous recovery rate R in Eq. (3.5.2), the expected loss for a tranche j attaching at A_j and detaching at D_j is,

$$E_j = (1 - R) \sum_{n=\lfloor \frac{A_j N}{1-R} \rfloor + 1}^{\lfloor \frac{D_j N}{1-R} \rfloor} P_n \left(n - \frac{A_j N}{1-R} \right) + (D_j - A_j) N \sum_{n=\lfloor \frac{D_j N}{1-R} \rfloor + 1}^N P_n. \quad (3.5.3)$$

3.6 Comparing models

To compare the LHP models introduced in Section 3.1 and 3.2 we consider a tranching iTraxx Europe CDO with 5 year maturity. The iTraxx Europe index is composed of the most liquid 125 credit default swaps referencing European investment grade credits and consists of tranches with attachment/detachment points at 3%, 6%, 9%, 12% and 22%. The market quotes of iTraxx tranches on the 12th July 2007 are considered with a constant recovery rate assumed to be 40%.

To compute the Student t4 copula model that best fits to market expected losses we seek the single correlation value that replicates the market

expected losses. For the iTraxx portfolio, we use a grid search to find the best fit to the market expected losses and for the Student t4 case this is the correlation value 0.2421. We repeat this grid search process to compute the Gaussian copula expected losses that best fit to market expected losses and for this iTraxx portfolio the expected loss for this Gaussian case is computed when the correlation value is 0.2440. The expected losses for the iTraxx tranches and the best fitting Gaussian and Student t4 copula cases are summarised in Figure 3.22.

From Figure 3.22 it can be seen that the Student t4 copula produces a good fit to market data overall except for the most senior tranches and matches better to the market values. The Gaussian copula performed badly, especially for the more senior tranches in the portfolio.

In the next chapter we introduce a new percolation model based on network theory for use in pricing CDO tranches. We introduce this fundamental model and fit the new model to market data, showing that it can fit even senior tranche expected losses.

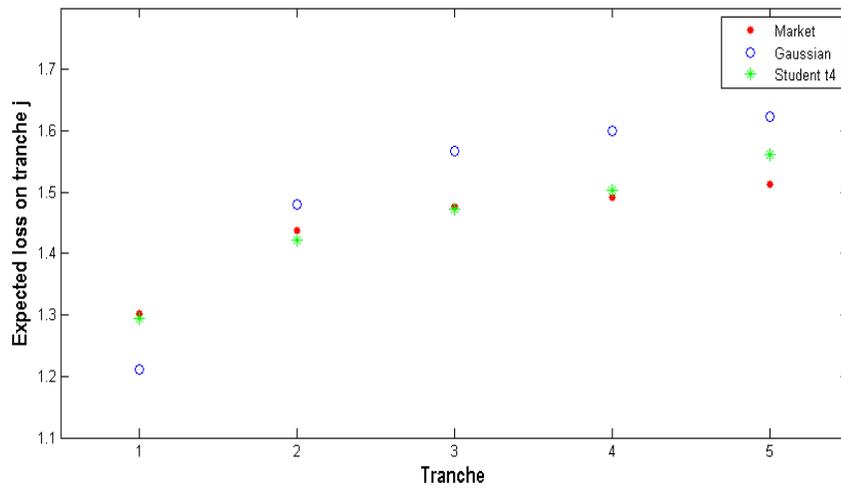


Figure 3.22: Plot of the expected loss for the five tranches of the iTraxx Europe on the 12th July 2007. Included in the plot are the best fitting (single correlation) Gaussian and Student t4 copula cases. Note the poor agreement of the Gaussian and good agreement (except for senior tranches) of the Student t4 copula.

Chapter 4

Network models

In this chapter we begin by introducing some of the earliest network models proposed by Paul Erdős and Alfréd Rényi and show how similar to relative component sizes on a network being related to contagion on a network we can compute the probability of default for a portfolio of names using network percolation.

We introduce two approaches to computing the component size distribution of a network by means of the finite and infinite network models and compare these models in their ability to model a portfolio of N names.

4.1 Introduction to network theory

In this section we closely follow Bornholdt and Schuster (2003) in describing some early network models. One of the first theoretical models of a network was the random graph which was introduced and studied by Paul Erdős and Alfréd Rényi in a series of seminal papers in the 1950s and

1960s.

The basic random graph consists of N vertices, where an edge is placed between pairs of vertices chosen uniformly at random. The most frequently studied version of the Erdős-Rényi random graph is the $G_{N,p}$ model where each edge is included in the graph with independent probability p , and is absent with probability $1 - p$. This $G_{N,p}$ model can not only be expressed in terms of p but also in terms of the average degree of a vertex z , where the degree of a vertex is the number of edges connected to that vertex. From Bornholdt and Schuster (2003) the average degree of a vertex is,

$$z = \frac{N(N-1)p}{N} = (N-1)p \simeq Np, \quad (4.1.1)$$

where the approximation $z \approx Np$ is good for large N ($N \rightarrow \infty$) where p is a parameter between 0 and 1.

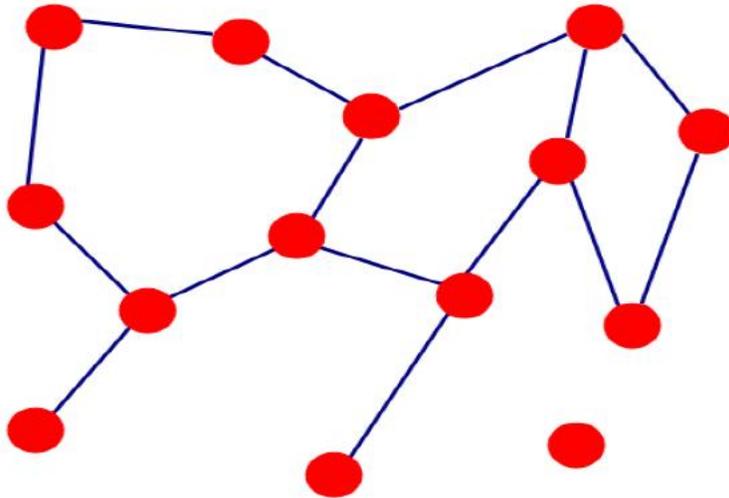


Figure 4.1: A network is a collection of vertices connected by edges.

The Erdős-Rényi random graph has a number of advantageous properties as a model of a network. For instance, the random graph has an interesting feature in that the model shows a phase transition with increasing z at which a giant component forms, where a component is a subset of vertices in the graph each of which can be reached from other vertices by some path through the network. When the average degree of a vertex z is small (i.e. few edges in the graph), most of the vertices are disconnected from one another and components are small. However, there is a critical value of z above which the one largest component in the graph contains a fraction S of the total number of vertices. The largest component in a graph is called the giant component. The phase transition at which the giant component occurs is precisely $z = 1$ and the formation of a giant component in the random graph is similar to the behaviour of many real-world networks (Bornholdt and Schuster, 2003).

The relative component sizes in a network are related to contagion on a network and depending on how many names a given name (i.e. seed default) is connected to and assuming contagion spreads along an edge with probability p , we can compute the probability of default using network percolation. In the following sections we introduce the infinite and finite network models which we use as two approaches in computing the component size distribution of a network.

4.2 Infinite network model

In this section we introduce the infinite network model and employ the model to compute the probability that a given vertex belongs to a component of a given size. We first present the degree distribution for the network model and then apply probability generating function techniques to compute the sizes of connected components in the network.

4.2.1 Degree distribution

Newman (2003) describes the number of edges connected to any particular vertex as the degree k of that vertex, having a probability distribution p_k given by,

$$\begin{aligned} p_k &= \binom{N-1}{k} p^k (1-p)^{N-1-k} \\ &\approx \frac{z^k e^{-z}}{k!}, \end{aligned} \tag{4.2.1}$$

where the last approximation equality becomes exact in the limit for $N \rightarrow \infty$ where p is a parameter between 0 and 1. Here p_k is the degree distribution of the infinite network model (Newman et al., 2001). Then if instead of a vertex we choose an edge and follow it to the vertex at one of its ends, the number of other edges emerging from that vertex follows the distribution q_k . This number is one less than the total degree of the vertex and is distributed as

$$q_k = \frac{(k+1)p_{k+1}}{z}, \tag{4.2.2}$$

where $z = \sum_{k=0}^{\infty} kp_k$ (for $N \rightarrow \infty$) is the average degree in the network (Bornholdt and Schuster (2003) and Newman (2007)). An alternative representation of the degree distribution is the probability generating function and in the following we introduce the probability generating functions for p_k and q_k .

4.2.2 Probability generating functions

Following closely Bornholdt and Schuster (2003), the corresponding generating function for p_k , the probability that a randomly chosen vertex has degree k is defined as,

$$G_0(x) = \sum_{k=0}^{\infty} p_k x^k, \quad (4.2.3)$$

We recover p_k from $G_0(x)$ by differentiation according to

$$p_k = \frac{1}{k!} \left. \frac{d^k G_0}{dx^k} \right|_{x=0}.$$

We say that $G_0(x)$ “generates” the probability distribution p_k and the distribution p_k is assumed correctly normalized such that

$$G_0(1) = \sum_{k=0}^{\infty} p_k = 1.$$

The generating function for the distribution q_k introduced in Eq. (4.2.2) is given by,

$$G_1(x) = \sum_{k=0}^{\infty} q_k x^k = \frac{\sum_{k=0}^{\infty} (k+1)p_{k+1}x^k}{\sum_{k=0}^{\infty} kp_k} = \frac{\sum_{k=0}^{\infty} kp_k x^{k-1}}{\sum_{k=0}^{\infty} kp_k} = \frac{G'_0(x)}{z}, \quad (4.2.4)$$

where z is the average degree of the vertex.

Taking the infinite Erdős-Rényi random graph with a Poisson degree distribution, i.e. substituting Eq. (4.2.1) into Eq. (4.2.3), we get,

$$G_0(x) = e^{-z} \sum_{k=0}^{\infty} \frac{z^k}{k!} x^k = e^{z(x-1)}. \quad (4.2.5)$$

Eq. (4.2.5) is the generating function for the Poisson distribution and the generating function for $G_1(x)$ is given by,

$$G_1(x) = \frac{G'_0(x)}{z} = e^{z(x-1)}. \quad (4.2.6)$$

Hence for the Poisson distribution case $G_0(x) = G_1(x)$ (Newman et al., 2001). As a result of having computed the generating functions p_k and q_k we can now compute the distribution of the component sizes for our network model.

4.2.3 The distribution of component sizes

We follow Bornholdt and Schuster (2003) in applying the generating functions presented previously to describe the distribution of component sizes. We first consider the distribution of the sizes of connected components in the graph. Let $H_1(x)$ be the generating function that generates the distribution of the sizes of components, reached by choosing a random edge and following it to one of its ends. Here “size” refers to the number of vertices in the component. Defined previously was q_k , the distribution of the number of edges k emanating from a vertex other than the one along which we came along. The distribution of the sum of the sizes of the k clusters that it can lead to is generated by,

$$H_1(x) = x \sum_{k=0}^{\infty} q_k [H_1(x)]^k. \quad (4.2.7)$$

However, q_k is nothing other than the coefficient of x^k in the generating function $G_1(x)$ and hence,

$$H_1(x) = xG_1(H_1(x)). \quad (4.2.8)$$

Subsequently, we look to compute the distribution of the sizes of the clusters to which a randomly chosen vertex belongs. The number of edges emanating from a vertex is distributed according to the degree distribution p_k and each edge leads to a cluster whose size is drawn from the distribution generated by the function $H_1(x)$. Hence the generating function for the

size of the whole component is,

$$H_0(x) = x \sum_{k=0}^{\infty} p_k [H_1(x)]^k = xG_0(H_1(x)). \quad (4.2.9)$$

We compute closed-form expressions for the generating functions up to an finite order in x by iteration of Eq. (4.2.8). And so, beginning with the expression $H_1(x) = q_0x$ which is known to be correct for order x^1 we substitute it into Eq. (4.2.8) and iterate. On each iteration we generate an expression for $H_1(x)$ that is accurate to one order higher. After n iterations, we will have an expression in which the coefficients for all orders up to and including x^{N+1} are exactly correct (Bornholdt and Schuster, 2003). Having described the general methodology in computing the distribution of component sizes, in the following section we implement this model with a Poisson degree distribution.

4.2.4 Infinite network model

For the infinite network model, we take the Erdős-Rényi random graph with a Poisson degree distribution whereby $G_0(x) = G_1(x) = e^{z(x-1)}$ and noting that $q_0 = e^{-z}$ for this specific case, we find the first few iterations

to be as follows,

$$\begin{aligned}
zH_1^{(1)}(x) &= xze^{-z} + O(x^2), \\
zH_1^{(2)}(x) &= xze^{-z} + (xze^{-z})^2 + O(x^3), \\
&\vdots \\
zH_1^{(5)}(x) &= xze^{-z} + (xze^{-z})^2 + \frac{3}{2}(xze^{-z})^3 + \frac{8}{3}(xze^{-z})^4 + \\
&\quad + \frac{125}{24}(xze^{-z})^5 + O(x^6). \tag{4.2.10}
\end{aligned}$$

Continuing with this procedure, the probability P_s of a randomly chosen vertex belonging to components of size $s = 1, 2, 3, \dots$ is,

$$\begin{aligned}
P_1 &= e^{-z}, & P_2 &= ze^{-2z}, & P_3 &= \frac{3}{2}z^2e^{-3z}, \\
P_4 &= \frac{8}{3}z^3e^{-4z}, & P_5 &= \frac{125}{24}z^4e^{-5z}, & \dots
\end{aligned}$$

With Maple we implement an iterative procedure to compute P_s the component size distribution of the Poisson random graph up to order x^{126} . Equally, Newman (2007) shows that these values have the general formula

$$P_s = \frac{e^{-zs}(zs)^{s-1}}{s!}, \tag{4.2.11}$$

for $s = 1, 2, \dots, \infty$, where P_1 corresponds to the is the probability of 1 individual defaulting, P_2 is the probability of 2 individual defaulting, and so on. Here the probability of belonging to a component of a given size reflects the probability of default, and an (implicit) correlation.

In the following section we introduce the finite network model, an alter-

native method to computing the component size distribution of a network.

4.3 Finite network model

As an alternative method to compute the distribution of component sizes we present a finite network model based on Newman (2003) which provides us with the distribution of component sizes, assuming two individuals are connected with probability p . This model incorporates non-Poisson degree distributions and non trivial clustering, two properties that are often observed in real world networks (Bornholdt and Schuster, 2003).

4.3.1 Component structure

The finite model is set up as follows: suppose a person A belongs to a group of N people. We are looking to compute how many individuals within that group A is connected to, whether directly via a single edge or indirectly via any path through other members of the group. From Newman (2003), $P(k|N)$ is the probability that vertex A belongs to a connected cluster of k vertices in the group including itself such that,

$$P(k|N) = \binom{N-1}{k-1} q^{k(N-k)} P(k|k). \quad (4.3.1)$$

Eq. (4.3.1) follows since by taking a graph of k vertices, all of which A is connected to, and adding $N - k$ other vertices to the graph, which can be done in $\binom{N-1}{k-1}$ distinct ways, each with probability $q^{k(N-k)}$, we can make an appropriate graph of n labeled vertices where $q^{k(N-k)}$ is the probability that

no newly added vertices connects to any of the k old vertices (Newman, 2003). Note $q = 1 - p$.

The probability $P(k|k)$ from Eq. (4.3.1) can be written in the form,

$$P(k|k) = \sum_{l=0}^s M_l^k p^l q^{s-l}, \quad (4.3.2)$$

a polynomial in p of order $s = \frac{1}{2}k(k-1)$ and M_l^k is the number of labelled connected graphs with k vertices and l edges. A disadvantage of this form is that it can be complex to evaluate analytically and for this reason we simply observe from Newman (2003) that

$$P(k|k) = 1 - \sum_{l=0}^{k-1} P(l|k). \quad (4.3.3)$$

This equation in combination with Eq. (4.3.1) allows us to evaluate $P(k|N)$ iteratively, given an initial condition $P(1|1) = 1$.

Iteratively, we compute $P(1|N)$, $P(2|N)$, $P(3|N) \dots P(N|N)$ where $P(1|N)$ is the probability of 1 individual defaulting out of a possible N individuals, $P(2|N)$ is the probability of 2 individuals defaulting out of a possible N individuals, etc. In Table 4.1, we describe the polynomials $P(k|10)$ for k up to 10.

In the following section we compare the two models presented previously in their ability to compute the distribution of component sizes and their ability in modelling the probability of default.

Table 4.1: Polynomials $P(k|10)$ for values for k up to 10

k	$P(k 10)$
1	$(1 - p)^9$
2	$9(1 - p)^{16}p$
3	$36(-1 + p)^{21}p^2(-3 + 2p)$
4	$-84(-1 + p)^{24}p^3(-16 + 33p - 24p^2 + 6p^3)$
5	$-126(-1 + p)^{25}p^4(570p^4 - 180p^5 + 24p^6 - 980p^3 + 125 - 528p + 970p^2)$
6	$126(-1 + p)^{24}p^5(-63195p^3 + 86110p^4 - 81840p^5 + 31080p^2 + 54780p^6 - 25440p^7 + 7830p^8 - 1440p^9 + 120p^{10} + 1296 - 9300p)$
7	$84(-1 + p)^{21}p^6(-965160p^2 + 3209430p^3 - 7527471p^4 + 103320p^{13} - 12600p^{14} + 720p^{15} + 183810p - 4894680p^{10} + 1869840p^{11} - 526890p^{12} + 13150032p^5 - 17633945p^6 + 18448710p^7 - 15159120p^8 + 9770810p^9 - 16807)$
8	$-36(-1 + p)^{16}p^7(636845160p^{15} - 205732800p^{16} + 52319190p^{17} - 10086720p^{18} + 4068456p - 30802240p^2 + 150657080p^3 - 5536123600p^{12} + 3258291120p^{13} + 1386000p^{19} - 120960p^{20} - 1590501640p^{14} - 262144 + 9324568001p^9 - 9345271992p^{10} + 7856193296p^{11} - 532354536p^4 + 1441519296p^5 - 3098951072p^6 + 5410382880p^7 - 7786027816p^8 + 5040p^{21})$
9	$-9(-1 + p)^9p^8(4782969 - 6952118544p^3 + 1032288516p^2 + 31699531080p^2 - 7276802400p^2 - 1337084280p^2 - 189090720p^2 - 100143792p + 2256642859464p^8 - 4160649753288p^9 + 6604921452864p^{10} - 866124834680p^{19} + 340283488860p^{20} + 19323360p^{26} - 1270080p^{27} - 113494862880p^{21} - 1044267039720p^7 + 5841996257286p^{16} - 3565563257088p^{17} + 1890561025584p^{18} - 9087542239104p^{11} + 10886907107538p^{12} - 11390487268104p^{13} + 10424251015920p^{14} - 8346992077872p^{15} + 34273176588p^4 - 131447465856p^5 + 407158789500p^6 + 40320p^{28})$
10	$-15300758477189520p^{22} + 21469710851551800p^{23} - 27011077082801580p^{24} + 30532209914200806p^{25} + 17388982649046960p^{29} - 11596879696617600p^{30} - 234748765440p^{40} + 32833495800p^{41} + 100000000p^9 - 2719892160p^{10} + 2782630494934920p^{19} - 5532426738592740p^{20} - 3567715200p^{42} - 31050312703343640p^{26} + 28429756177413360p^{27} + 9748958193896580p^{21} + 6939551178972720p^{31} - 3716558335019880p^{32} + 1775448575926410p^{33} - 753273866698920p^{34} + 282287441908080p^{35} + 282592800p^{43} - 14515200p^{44} - 159642667620135p^{16} + 477077447178540p^{17} - 1232671556293800p^{18} + 362880p^{45} + 26521978127400p^{37} - 6517548349200p^{38} - 92792729053500p^{36} + 1357020856800p^{39} + 36628300800p^{11} - 324496267200p^{12} + 2121183237600p^{13} - 10884316965480p^{14} + 45558310696800p^{15} - 23437879996999860p^{28}$

4.4 Comparing infinite and finite networks models

In this section we compare the component size distributions for the finite and infinite networks for differing number of names in the network and graph our results. We graph the results of increasing default probability p from 0 up to the critical threshold value $z = 1$ and examine the graphs for similarities. In a network, the phase transition is the point at which the giant component forms and this occurs precisely at $z = 1$. We constrain the bond percolation parameter so that z is at or below the threshold value to ensure we remain in the regime where there is only finite size components i.e. no giant component.

First we consider a network of 10 vertices i.e. a portfolio of 10 reference names. We compute the default probability p corresponding to the average degree of a vertex 1 i.e. $z = 1$ such that,

$$p = \frac{z}{N - 1} = \frac{1}{9} = 0.1111.$$

We examine the probability of default P_n when the default probability p is below this threshold value for all infinite and finite networks. We note our results in Table 4.2. From this table it can be seen that both the finite and infinite networks match well for all default probability values. When comparing the models, we also check to confirm whether the $\sum_{n=1}^{\infty} P_n$ for the infinite network models is approximately close to 1. Note $\sum_{n=1}^{\infty} P_n$ will

slightly deviate from 1 since we do not include P_0 . Decreasing the critical

Table 4.2: Results for the infinite and finite network models varying the default probability p when $N = 10$.

p	z	Infinite $-\sum_{n=1}^N P_n \approx$	Finite $-\sum_{n=1}^N P_n \approx$	Good match for n btw
0.11	0.99	0.762008	1	1 – 10
0.09	0.81	0.884082	1	1 – 10
0.07	0.63	0.963569	1	1 – 10
0.05	0.45	0.994808	1	1 – 10
0.03	0.27	0.999863	1	1 – 10
0.01	0.09	0.999999	1	1 – 10
0.005	0.045	0.99999	1	1 – 10
0.004	0.036	1	1	1 – 10
0.003	0.027	1	1	1 – 10
0.001	0.009	1	1	1 – 10

value from 0.11 to 0 we note improvements in the fitting of the finite and infinite network probability of defaults, see Figure 4.2.

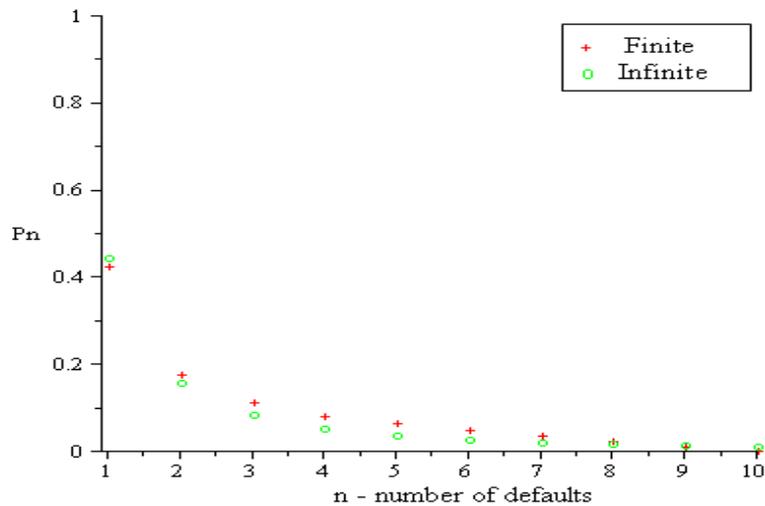


Figure 4.2: Comparing Newman’s finite and infinite models when $N = 10$ and probability of default $p = 0.09$.

Next we consider a larger network where the number of names in the portfolio is $N = 40$. We compute the default probability p corresponding to an average degree of a vertex $z = 1$ such that,

$$p = \frac{z}{N - 1} = \frac{1}{39} = 0.02564.$$

We examine the behaviour of the probability of default P_n when the default probability p lies below this value and note our results in Table 4.3. The

Table 4.3: Results for the infinite and finite network models varying the default probability p when $N = 40$.

p	z	Infinite $-\sum_{n=1}^N P_n \approx$	Finite $-\sum_{n=1}^N P_n \approx$	Good match for n btw
0.024	0.936	0.924577	1	1 – 12
0.02	0.78	0.988695	1	1 – 10
0.0175	0.6825	0.988245	1	1 – 11
0.015	0.585	0.999872	1	1 – 9
0.01	0.39	0.999999	1	1 – 8
0.006	0.234	0.999999	1	1 – 8
0.003	0.117	1	1	1 – 7
0.001	0.039	0.999999	1	1 – 10

finite and infinite network models match well for $1 \leq n \leq 10$. The $\sum_{n=1}^N P_n$ for the finite and infinite models are approximately close to 1. Note the poor agreement between the models when $n > 10$, see Figure 4.3.

Increasing the number of names in the network further, we examine the probability of default when $N = 70$. We compute the default probability p corresponding to the threshold value,

$$p = \frac{z}{N - 1} = \frac{1}{69} = 0.01449,$$

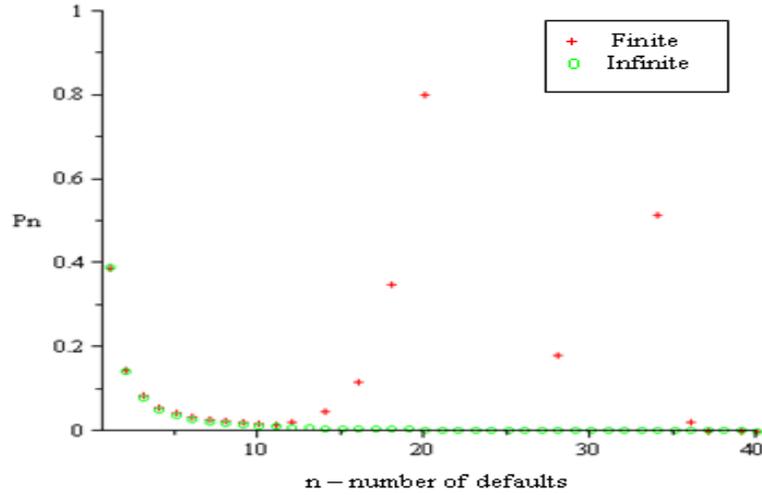


Figure 4.3: Comparing Newman’s finite and infinite models when $N = 40$ and probability of default $p = 0.024$. Note the errors are due to numerical calculation of binomial coefficients - this could be improved by using a more careful method of computing the binomial coefficients. We don’t pursue this in our thesis but it could be an area for further work.

and examine the probability of default P_n when the default probability p lies below this threshold value. We note our results in Table 4.4. From our

Table 4.4: Results for the infinite and finite network models varying the default probability p when $N = 70$.

p	z	Infinite $-\sum_{n=1}^N P_n \approx$	Finite $-\sum_{n=1}^N P_n \approx$	Good match for n btw
0.0143	0.9867	0.916664	1	1 – 10
0.011	0.759	0.998526	0	1 – 9
0.008	0.552	0.999999	0	1 – 8
0.004	0.276	0.999999	0	1 – 7
0.0017	0.1173	1	0	1 – 8
0.0005	0.0345	1	0	1 – 7

analysis the infinite and finite network models match well when $1 \leq n \leq 8$

and fails to match when $n \geq 8$. We compute the $\sum_{n=1}^N P_n$ for the infinite network model and the results are approximately close to 1. In contrast, the $\sum_{n=1}^N P_n$ for the finite network model is approximately 0. The finite network no longer has $\sum_{n=1}^N P_n$ being approximately 1 and this failure is due to the numerical round off errors when computing large binomial coefficients, see Figure 4.4. The numerical errors here are sufficiently serious such that the outcome no longer makes sense.

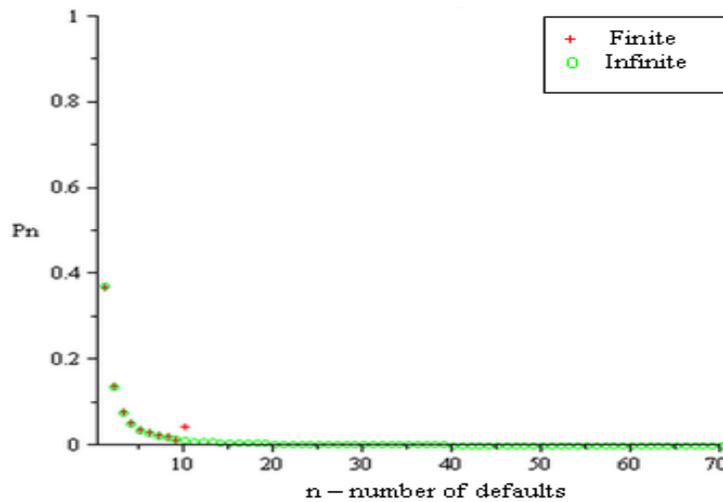


Figure 4.4: Comparing Newman's finite and infinite models when $N = 70$ and probability of default $p = 0.0143$.

Finally we examine a large portfolio network similar to that of the size of an iTraxx Europe tranching CDO with $N = 125$ reference names and compute the default probability p corresponding to an average degree of

a vertex $z = 1$ such that,

$$p = \frac{z}{N-1} = \frac{1}{124} = 0.0080645.$$

We examine the behaviour of the probability of default P_n when the default probability p lies below the threshold value and note our results in Table 4.5. The finite and infinite models match well when $1 \leq n \leq 7$ and

Table 4.5: Results for the infinite and finite network models varying the default probability p when $N = 125$.

p	z	Infinite $-\sum_{n=1}^N P_n \approx$	Finite $-\sum_{n=1}^N P_n \approx$	Good match for n btw
0.008	0.992	0.93601	0	1 – 8
0.00645	0.7998	0.999416	0	1 – 7
0.0049	0.6076	0.99999	0	1 – 7
0.0032	0.3968	1	0	1 – 6
0.0016	0.1984	1	0	1 – 5
0.0008	0.0992	1	0	1 – 4
0.0004	0.0496	1	0	1 – 5

the $\sum_{n=1}^N P_n$ is approximately close to 1 for the infinite network model. For

the finite network model the $\sum_{n=1}^N P_n \approx 0$, this is due to the round off errors highlighted previously and as the network get infinitely large solving the probability of default P_n for the finite network becomes increasingly computationally expensive. This is due to the round off numerical errors generated by the multiplication of large binomial coefficients with smaller terms, see Figure 4.5. Therefore we only get accurate results for $\sum_{n=1}^N P_n$ when we limit the number of names in the portfolio to $N = 30$ for the finite case. This is a major disadvantage considering the typical size of

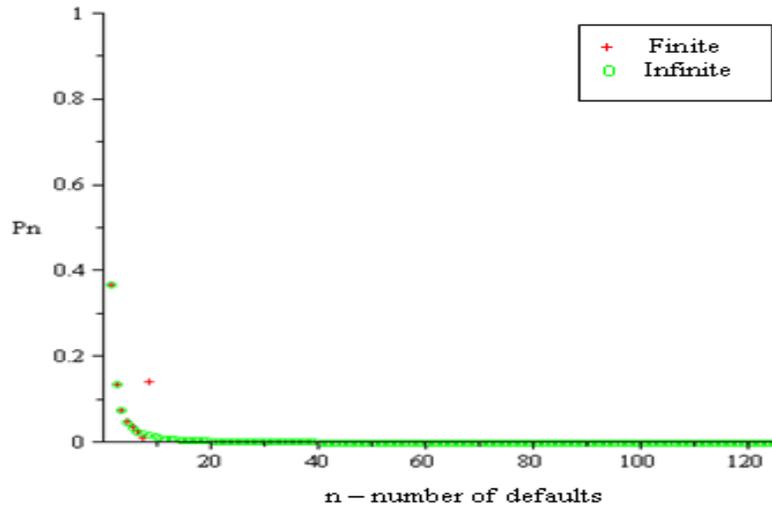


Figure 4.5: Comparing Newman’s finite and infinite models when $N = 125$ and probability of default $p = 0.008$.

a CDO portfolio consists of 125 reference names (for CDX and iTraxx Europe indices) and up to 600 reference names for bespoke CDOs. For this reason and since the results for the infinite network case and the corresponding finite networks cases match well, as our preferred network model continuing onwards we will use the infinite network model as our standard network model.

In the following chapter we extend the infinite network model presented here and introduce our percolation model Q_n . We then compare this percolation model, the Gaussian and Student t4 copulas to market data and note our results.

Chapter 5

The percolation model

In this chapter we propose a new percolation model based on network theory for use in CDO pricing. To model default dependence between names in a portfolio we examine the possible replacement of current copula based approaches with intuitive contagion models for percolation on networks.

We extend the infinite network model introduced in Chapter 4 to compute the probability of default of a CDO portfolio. Note that to compute the percolation model we use the pricing measure to imply the bond percolation parameter from the CDS spreads similar to the use of the pricing measure in the copula case in Chapter 3.

In this chapter we firstly compute the default distribution averaging over seed values and from this set up our percolation model Q_n . We extract percolation parameters from the model using iTraxx data sets and then compare the percolation model to other models in its ability to match to market data.

5.1 The default distribution averaging over seed values

In this section we further develop the infinite network model introduced in the previous chapter to compute the default distribution \underline{P}_n . Firstly, we let $P_n(m)$ represent the probability distribution for number n of defaults assuming m initial seeds, where a seed is a vertex in the network. Then, depending on how many other vertices a seed or seeds are connected to, we are able to model the spread of contagion given an initial number of seeds, such that $m = 1, 2, \dots, N$.

From Newman (2007), the probability that a single seed in a network results in n defaults is given by,

$$P_n(1) = \frac{e^{-zn}(zn)^{n-1}}{n!} = \frac{1}{z}(ze^{-z})^n \frac{n^{n-1}}{n!}, \quad (5.1.1)$$

where $z = (pperc)(N)$ and $pperc$ is the bond percolation parameter i.e. the parameter p from Section 4.1 and for $n = 1, \dots, N$. We simplify Eq. (5.1.1) to

$$P_n(1) = \frac{1}{z}(ze^{-z})^n A_{1,n}, \quad (5.1.2)$$

where $A_{1,n} = \frac{n^{n-1}}{n!}$. Next we consider a probability density function of two cascade sizes on an infinite network such that

$$P_n(2) = \sum_{r=1}^{n-1} P_r(1)P_{n-r}(1). \quad (5.1.3)$$

Then, rewriting Eq. (5.1.3) in matrix form, see Appendix B, keeping z dependence separate as much as possible results in

$$P_n(2) = \sum_{r=1}^{n-1} \frac{1}{z} (ze^{-z})^r \frac{1}{z} (ze^{-z})^{n-r} \left[\frac{r^{r-1} (n-r)^{n-r-1}}{r! (n-r)!} \right],$$

and simplifying,

$$P_n(2) = \frac{1}{z^2} (ze^{-z})^n A_{2,n}, \quad (5.1.4)$$

where $A_{2,n}$ contains numerical factors only, independent of z i.e. $A_{2,n} = \sum_{r=1}^{n-1} \left[\frac{r^{r-1} (n-r)^{n-r-1}}{r! (n-r)!} \right]$. Therefore, the general formula for the probability of i seeds leading to n defaults is given by

$$P_n(i) = \sum_{r=1}^{n-1} P_r(1) P_{n-r}(i-1), \quad (5.1.5)$$

and this can simply be written as

$$P_n(i) = \frac{1}{z^i} (ze^{-z})^n A_{i,n}, \quad (5.1.6)$$

where $A_{1,n} = \sum_{r=1}^{n-1} A_{1,r} A_{(i-1),(n-r)}$.

For our percolation model we assume that the numbers of seeds have a Poisson-type density function \widetilde{Q}_i , where \widetilde{Q}_i is the distribution of the seed sizes and that the expected number of seed defaults is given by $z_1 = \rho_0(N-1)$. In this case the density function can be written as

$$\widetilde{Q}_i = \frac{z_1^{i-1}}{(i-1)!} e^{-z_1}, \quad (5.1.7)$$

where \widetilde{Q}_i is normalised such that

$$\sum_{i=1}^{\infty} \widetilde{Q}_i = 1,$$

and its mean is

$$\sum_{i=1}^{\infty} i \widetilde{Q}_i = z_1 + 1.$$

Then, having computed the distribution of seed sizes we can compute the default distribution \underline{P}_n after averaging over the seeds sizes, such that

$$\underline{P}_n = \sum_{i=1}^N \widetilde{Q}_i P_n(i), \quad (5.1.8)$$

where $P_n(i)$ is the probability distribution for n number of defaults assuming i initial seeds. We substitute Eq. (5.1.3) and Eq. (5.1.7) into the above equation resulting in,

$$\underline{P}_n = \sum_{i=1}^N \frac{z_1^{i-1}}{(i-1)!} e^{-z_1} \frac{1}{z^i} (ze^{-z})^n A_{i,n}.$$

This simplifies the default distribution \underline{P}_n to

$$\underline{P}_n = \frac{1}{z_1} e^{-z_1} \sum_{i=1}^N \left(\frac{z_1}{z}\right)^i \frac{1}{(i-1)!} A_{i,n} (ze^{-z})^n. \quad (5.1.9)$$

We generate \underline{P}_n in MATLAB by,

$$\underline{P}_n = \frac{1}{z_1} e^{-z_1} \sum_{i=1}^N \left(\frac{z_1}{z}\right)^i B_{i,n} (ze^{-z})^n, \quad (5.1.10)$$

where $B_{i,n} = \frac{1}{(i-1)!}A_{i,n}$, see Appendix B.

In this section we have computed the default distribution \underline{P}_n having averaged over seed sizes. Using this result in the following section we can begin to compute the probability of default for the percolation model.

5.2 The percolation model Q_n

In this section we introduce our percolation model used to compute the probability of default for N names in a CDO portfolio. We model the probability of default Q_n in the network model where names in the portfolio are correlated. Modelling the probability of default in a correlated portfolio is similar to modelling the probability of default of contagion spreading in a network.

The basic percolation model is made up of agents, who are classed as optimists, percolationists or catastrophists. Then as a result of a group of agents' expectations of the network, one can model the price of bonds in a CDO portfolio and draw conclusions from it. As the name suggests, optimists are the fraction of agents that expect no defaults to occur in the network, catastrophists are the fraction that expect N defaults to occur in the network and percolationists are the fraction that expect defaults in between these extreme events. For the percolation model, we first ensure that the total fraction of catastrophists, percolationists and optimists sums

to 1, so we let

$$\widetilde{P}_n = \frac{P_n}{\sum_{n=1}^N P_n},$$

such then that $\sum_{n=1}^N \widetilde{P}_n = 1$. The distribution of the number of defaults Q_n is thus given by

$$Q_n = \begin{cases} a & n = 0, \\ b\widetilde{P}_n(z, z_1) & 1 \leq n \leq N - 1, \\ c & n = N, \end{cases}$$

where a is the fraction of optimists, b is the fraction of the population who are percolationists and c is the fraction who are catastrophists. Optimists and catastrophists improve the fit to the market data, because percolationists always assume at least one default hence the parameters a and c are introduced. Having outlined the percolation model in Eq. (5.2), we now set out the constraints of the model and use these constraints to compute the unknown parameters of the model by fitting to data.

The constraints of the model are as follows: firstly, the total distribution of the number of defaults must sum to 1, as a result,

$$\sum_{n=0}^N Q_n = 1 \implies a + b + c = 1,$$

and rewriting in terms of the fraction of optimists,

$$a = 1 - b - c. \tag{5.2.1}$$

Aside from this condition, we include a check in our MATLAB code to ensure that the fraction of optimist $a \geq 0$, the fraction of percolationists $b \geq 0$, and the fraction of catastrophists $c \geq 0$. Secondly, the calculation of the expected loss (EL) is given by

$$EL = \sum_{n=0}^N nQ_n,$$

such then that

$$EL = b \sum_{n=1}^{N-1} n\tilde{P}_n + cN,$$

and this provides a condition for the fraction of catastrophists c ,

$$c = \frac{EL - b \sum_{n=1}^{N-1} n\tilde{P}_n}{N}. \quad (5.2.2)$$

Since having computed a condition for a (the fraction of optimists) and a condition for c (the fraction of catastrophists) we now need to compute a value for the fraction of percolationists b . Then, for a given z and z_1 and using Eq (5.2.2) for c , we choose the best fit b to match to market base correlation tranche values. The calculation of the expected loss on tranche j , attaching at $A_j = 0$ i.e. the base tranche, is given by,

$$E_j = (1 - R) \sum_{k=1}^{\lfloor \frac{D_j N}{1-R} \rfloor} k p_k + D_j N \sum_{k=\lfloor \frac{D_j N}{1-R} \rfloor + 1}^N p_k.$$

Consequently, the expected loss for the percolation model when $A_j = 0$ is,

$$E_j = (1 - R) \sum_{n=1}^{\lfloor \frac{D_j N}{1-R} \rfloor} n \tilde{P}_n + D_j N \sum_{n=\lfloor \frac{D_j N}{1-R} \rfloor + 1}^N b \tilde{P}_n + D_j N c. \quad (5.2.3)$$

Replacing c , the fraction of catastrophists, with Eq. (5.2.2) results in,

$$E_j = (1 - R)b \sum_{n=1}^{\lfloor \frac{D_j N}{1-R} \rfloor} n \tilde{P}_n + b D_j N \sum_{n=\lfloor \frac{D_j N}{1-R} \rfloor + 1}^N \tilde{P}_n - b D_j \sum_{n=1}^N n \tilde{P}_n + D_j N E L.$$

Simplifying we get,

$$E_j = b m_j + D_j N E L \quad (5.2.4)$$

where

$$m_j = (1 - R) \sum_{n=1}^{\lfloor \frac{D_j N}{1-R} \rfloor} n \tilde{P}_n + D_j N \sum_{n=\lfloor \frac{D_j N}{1-R} \rfloor + 1}^N \tilde{P}_n - D_j \sum_{n=1}^N n \tilde{P}_n.$$

Then, in order to solve for b , the fraction of percolationists, we allow $f(b)$ be the function

$$f(b) = \sum_{j=1}^5 (E M_j - E_j)^2, \quad (5.2.5)$$

the sum of the squares of the difference between the market expected loss $E M_j$ and the percolation model expected losses E_j for each tranche. We seek to compute a value of b that minimises the difference between market expected losses and the percolation model expected losses. We rewrite Eq.

(5.4.1), using Eq. (5.2.4) as

$$f(b) = \sum_{j=1}^5 (F_j - bm_j)^2, \quad (5.2.6)$$

where $F_j = EM_j - D_jNEL$, and minimise the function $f(b)$ by computing the gradient

$$\frac{df}{db} = \sum_{j=1}^5 2(F_j - bm_j)(-m_j).$$

Setting $\frac{df}{db} = 0$, the best fit fraction of percolationists b to market data is given by

$$\hat{b} = \frac{\sum_{j=1}^5 m_j F_j}{\sum_{j=1}^5 m_j^2}. \quad (5.2.7)$$

Note that the estimates a , b and c are dependent on $pperc$ and ρ_0 . Then for the percolation model, we now can compute a the fraction of optimists, b the fraction of percolationists and c the fraction of catastrophists for a given $pperc$ and ρ_0 and in addition the percolation model expected loss EM_j for each tranche in the portfolio - see Appendix B. We measure the fit between market tranche values and percolation tranche values using a least square calculation given a $pperc$ and ρ_0 value and next convert the percolation model expected losses to base correlation values using the process discussed previously in Section 3.4.

In this section we outlined the parameters of the percolation model and the methodology of how the fraction of percolationists, catastrophists and optimists are computed. Prior to examining some of the features of

the percolation model and extracting implied percolation parameters, we firstly use market data from www.creditfixings.com to compute market expected loss values for each tranche j and the default probability of the CDO portfolio. Then we use the market data as inputs into the percolation model and extract percolation parameters from it.

5.3 Market data

In this section we use market data from www.creditfixings.com to compute the default probability of the portfolio and the expected loss for each tranche in the portfolio. We link market data to the Large Pool Model to which we then apply various different models e.g. copula, percolation etc.

We closely follow McGinty (2004) to implement the Large Pool Model and this model allows us to input quoted market data and outputs the portfolio default probability and the expected loss for each tranche in the portfolio. Using a bootstrapping process we compute the tranche base correlation values for the CDO portfolio and using these outputs we can compute parameters for our percolation model detailed previously in Section 5.2.

5.3.1 The Large Pool Model

Previously in Section 3.4 we discussed the Homogeneous Large Pool Gaussian Copula Model (Large Pool Model) spreadsheet in McGinty (2004) that we use to compute tranche expected loss values from market data inputs.

This Large Pool Model is used as a method for translating market quotes to an implied correlation and works on the assumption that the portfolio is infinite in size and homogeneous.

In this section we link market data to the Large Pool Model to which we can then apply various different models e.g. copula, percolation etc. and using this method we compute the portfolio default probability and the expected loss for each tranche j in the portfolio using the Large Pool Model.

In this section, we closely follow McGinty (2004) in describing the Large Pool Model. Firstly, to compute the default probability p , we consider a CDO portfolio with normalised total notional $notional = 1$, average recovery rate R and an index spread $IndSpd$ quoted in basis points. The period of interest for valuation is calculated as,

$$horizon = \frac{MaturityDate - ValueDate}{360},$$

and the portfolio clean spread

$$CleanSpread = \frac{IndSpd}{1 - R}.$$

Having outlined the $horizon$ and $CleanSpread$ above, the portfolio default probability is given as

$$p(horizon) = 1 - e^{-\frac{(CleanSpread)(horizon)}{10000}}. \quad (5.3.1)$$

We now have an equation to compute the default probability p . We next need to compute the expected loss for each tranche in the portfolio. Previously in Section 3.5, we described that in order to calculate the expected loss for each tranche in the portfolio, a model where the loss distribution and the default probability are inputs is used. This method is similar to that of the Large Pool Model. From McGinty (2004), we compute the expected loss on a tranche j from the quoted market spreads such that,

$$E_j(horizon) = (D_j - A_j)[1 - ES_j(horizon)], \quad (5.3.2)$$

where D_j and A_j are the detachment and attachment points of tranche j respectively and $ES_j(horizon)$ is the expected survival of tranche j at horizon. We convert the expected survival to a survival rate, $SRate$, and use these as risky discount factors,

$$ES_j(t) = \left[1 + \frac{SRate_j}{4} \right]^{-4t},$$

where the $SRate_j$ is the survival rate for tranche j discounted quarterly such that this equation holds for $t = horizon$. This simplifies the expected survival to

$$ES_j(t) = [ES_j(horizon)]^{\frac{t}{horizon}}, \quad (5.3.3)$$

and with the aid of Eq. (5.3.2) it can be simplified further to,

$$ES_j(t) = 1 - \frac{E_j(t)}{D_j - A_j}. \quad (5.3.4)$$

From McGinty (2004) the spread s on tranche j , i.e. the spread at which the present value of the contingent leg is equal to the present value of the fee leg is given by

$$s_j = \frac{\sum_{k=1}^n PD_j(t_k, t_{k-1})DF(t_k)(D_j - A_j)}{(RiskyDuration)(D_j - A_j)}, \quad (5.3.5)$$

where $DF(t_k)$ is the discount factor for the time period t_k and PD_j is the expected incremental tranche loss such that

$$PD_j(t_k, t_{k-1}) = ES_j(t_{k-1}) - ES_j(t_k).$$

We simplify Eq. (5.3.5) to,

$$s_j = \frac{\sum_{k=1}^n [ES_j(t_{k-1}) - ES_j(t_k)]DF(t_k)}{\sum_{k=1}^n DF(t_k)ES_j(t_k)DayCount(t_k)}, \quad (5.3.6)$$

where the risky period is,

$$RiskyDuration = \sum_{k=1}^n DF(t_k)ES_j(t_k)DayCount(t_k),$$

and the fraction of days in the calculation period is

$$DayCount(t_k) = \frac{t_k - t_{k-1}}{360}.$$

Further simplifying, we substitute in Eq. (5.3.4), such that,

$$s_j = \frac{\sum_{k=1}^n \left[\frac{E_j(t_k)}{D_j - A_j} - \frac{E_j(t_{k-1})}{D_j - A_j} \right] DF(t_k)}{\sum_{k=1}^n DF(t_k) \left[1 - \frac{E_j(t_k)}{D_j - A_j} \right] DayCount(t_k)}, \quad (5.3.7)$$

and by substituting in Eq. (5.3.3) the spread for tranche j of a CDO portfolio is

$$s_j = \frac{\sum_{k=1}^n \left(\left[1 - \frac{E_j(horizon)}{D_j - A_j} \right]^{\frac{t_{k-1}}{horizon}} - \left[1 - \frac{E_j(horizon)}{D_j - A_j} \right]^{\frac{t_k}{horizon}} \right) DF(t_k)}{\sum_{k=1}^n \left[1 - \frac{E_j(horizon)}{D_j - A_j} \right]^{\frac{t_k}{horizon}} DF(t_k) DayCount(t_k)}. \quad (5.3.8)$$

Here in Eq. (5.3.8) we now have the spread for tranche j in terms of the expected loss for tranche j . We use market data spread values for each tranche as inputs for s_j and invert Eq. (5.3.8) to compute the expected loss for each tranche $E_j(horizon)$. Furthermore, since Eq. (5.3.8) links spreads s_j to expected losses $E_j(horizon)$ it is possible to simplify the time argument i.e. we abbreviate $E_j(horizon)$ to E_j . Moreover, the Large Pool Model described above also computes the base correlation values for each tranche j , similar to that of Section 3.4, where base correlations are defined as the correlation inputs required for a series of equity tranches that give tranche values consistency with quoted spreads using the standardised Large Pool Model. In the next section, we use the base correlation values computed from market data to imply percolation parameters for

our percolation model.

5.4 Percolation model parameters

In this section we apply the Large Pool percolation model introduced in Section 5.2 with a view to finding the single ρ_0 and $pperc$ value which best reproduces the market base correlation values for each tranche j in a CDO portfolio. The market data we use to compute the percolation model parameters are monthly iTraxx tranche fixing datasets from www.creditfixings.com.

From the market data, with the aid of the Large Pool Model, we extract the expected loss of each tranche j and the portfolio default probability p . This is required to implement the percolation model and extract percolation parameters from it. We use a recovery rate of $R = 40\%$ and a spread s_j equal to the mid level of an equivalently quoted iTraxx for each tranche j with a discount rate of 0. The market datasets we use to implement the model are quoted with attachment points - $\{0, 3, 6, 9, 12\}$ with corresponding detachment points - $\{3, 6, 9, 12, 22\}$. The percolation model is implemented with a market spread equal to the mid level of the equivalently quoted iTraxx where the $[0\% - 3\%]$ tranche is quoted as an upfront with a fixed 500 basis point spread. The value date of the portfolio is the date of issue of the quoted iTraxx portfolio and maturity date is equal to the value date plus 5, 7 or 10 years depending on the duration of the CDO portfolio.

The known parameters outlined above are input into the JP Morgan

spreadsheet of McGinty (2004) and it is this spreadsheet that outputs the portfolio default probability p , and for each tranche j , a base correlation and expected loss value. We have computed the market values needed to implement the percolation model in the Large Pool Model and the next step is to code the percolation model detailed in Section 5.2 in MATLAB. We code this model in MATLAB with the aim at finding the best fitting percolation model expected losses to market data expected losses for a given z and z_1 , input via $pperc$ and ρ_0 . We compute the mean square error function for a given range of $pperc$ and ρ_0 values and grid search over these values with the aim at finding the minimum mean square error for a unique $pperc$ and ρ_0 . Here, the mean square error (MSE) is

$$MSE = \sum_{j=1}^5 (EM_j - E_j)^2, \quad (5.4.1)$$

and is a function of $pperc$ and ρ_0 . This grid search recovers the best fitting percolation expected loss to market expected loss for each tranche j . Then, for each optimal $pperc$ and ρ_0 i.e. the values that minimise the mean square error, we compute the fraction of optimists a , the fraction of percolationists b , and the fraction of catastrophists c that make up the percolation model. We summarise our results in Figures 5.1 - 5.6.

In these figures we plot the percolation model parameters a , b , c , ρ_0 , $pperc$ and MSE for each of the iTraxx datasets. Firstly, examining Figure 5.1 - 5.6, we can see some trends in the data. Observing the behaviour of the bond percolation parameter $pperc$, the optimal parameter value for 5 and 7 year iTraxx data is approximately 0.015 for all iTraxx fixings while

for 10 year iTraxx data it is approximately 0.005. Secondly, note that the percolation model itself comprises mainly of percolationists, then a smaller fraction of optimists and the smallest fraction being catastrophists. The fraction of percolationists are generally in the range of $0.7 - 1$, while fraction of optimists in the range $0 - 0.3$ and the fraction of catastrophists in the range $0 - 0.14$. The ρ_0 parameter is generally in the range $0 \leq \rho_0 \leq 0.08$. The sum of the squares of the difference between the market expected loss EM_j and the percolation model expected losses E_j for each tranche j is less than 0.1 for all iTraxx fixings. In this section we have computed all unknown percolation model parameters. In the next section we examine the effect of constraining the bond percolation parameter below the phase transition level described previously in Section 4.1. We examine the effects that the constraining of $pperc$ has on other parameters in the network and its effect if any on MSE , the difference between the market expected loss EM_j and the percolation model expected losses E_j for each tranche j .

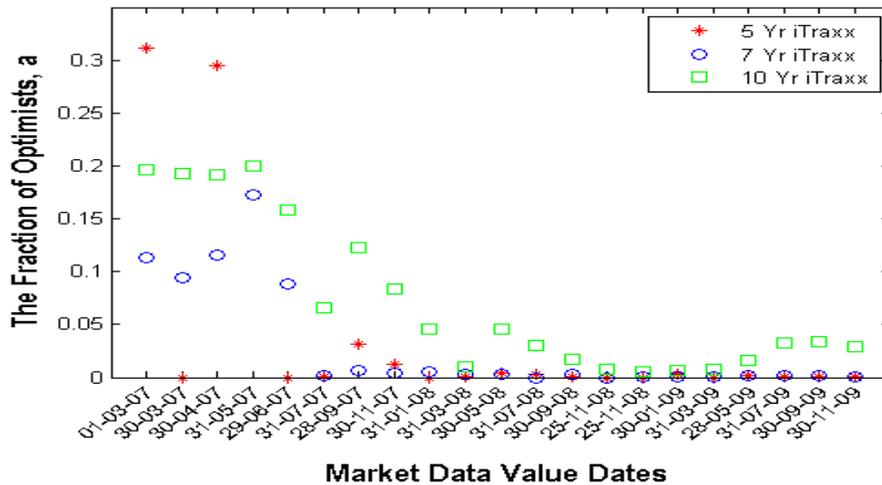


Figure 5.1: The fraction of optimists a that minimises the error function for the percolation model. The market data is made up of 5Yr, 7Yr and 10Yr iTraxx Europe tranche fixings.

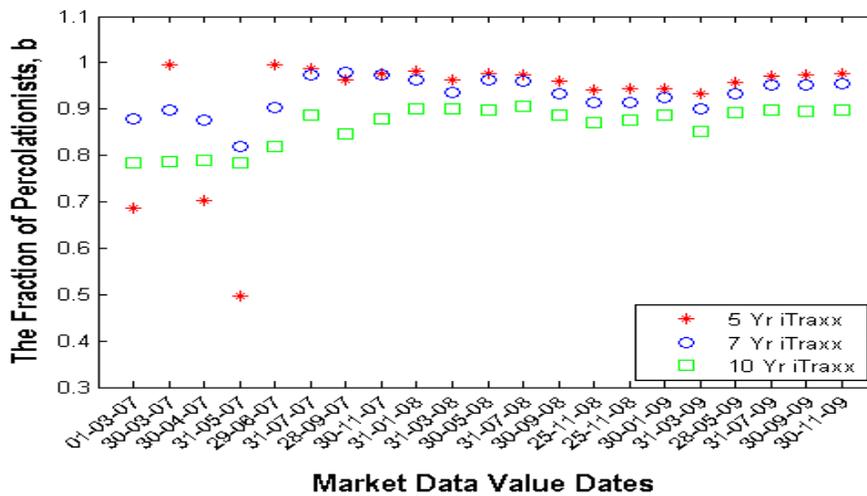


Figure 5.2: The fraction of percolationists b that minimises the error function.

5.5 Constraining the bond percolation parameter

In Section 4.1, we described the phase transition at which the giant component occurs in a network model and it is precisely $z = 1$. The formation

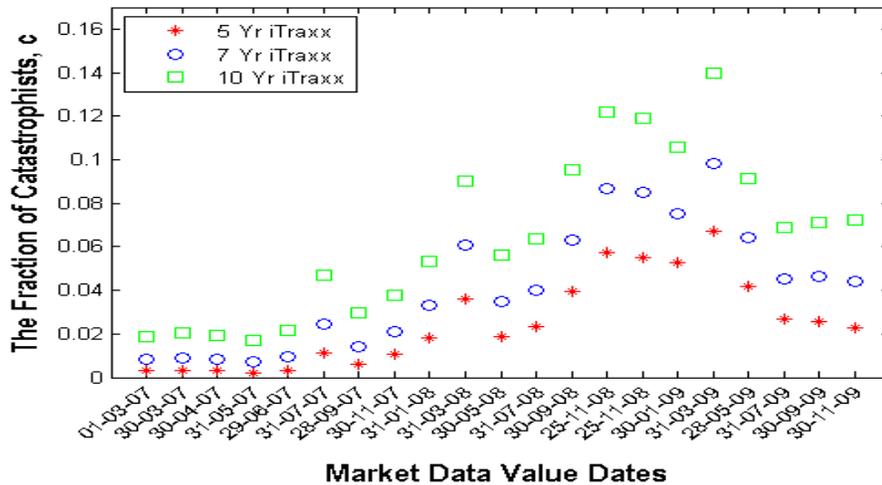


Figure 5.3: The fraction of catastrophists c that minimises the error function.

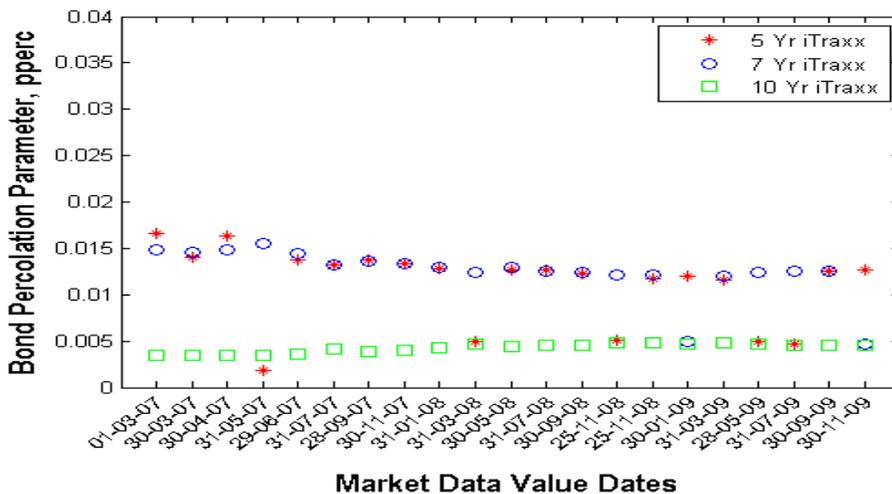


Figure 5.4: The bond percolation parameter $pperc$ that minimises the error function. Note, the relationship between $pperc$ and z is $z = (pperc)(N)$, as given in Section 5.1.

of a giant component in the random graph is similar to the behaviour of many real world networks. In the percolation model described previously

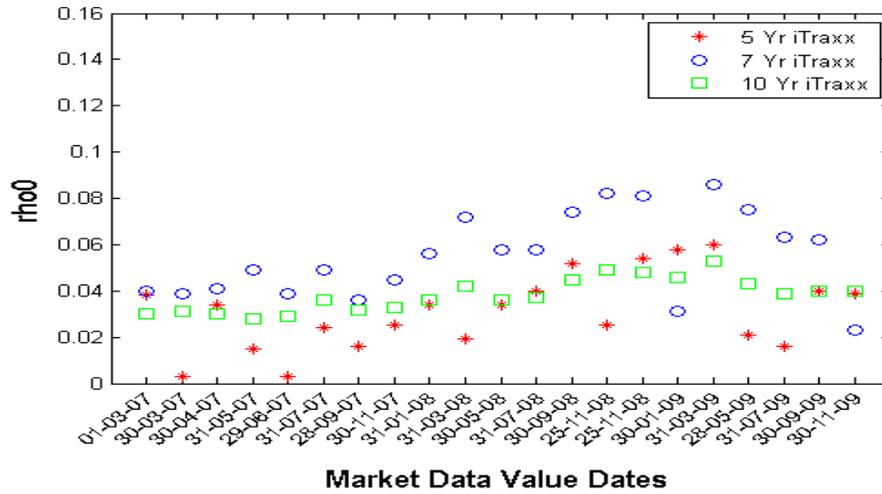


Figure 5.5: The parameter ρ_0 that minimises the error function. Note, the relationship between ρ_0 and z_1 is $z_1 = (\rho_0)(N - 1)$, as given in Section 5.1.

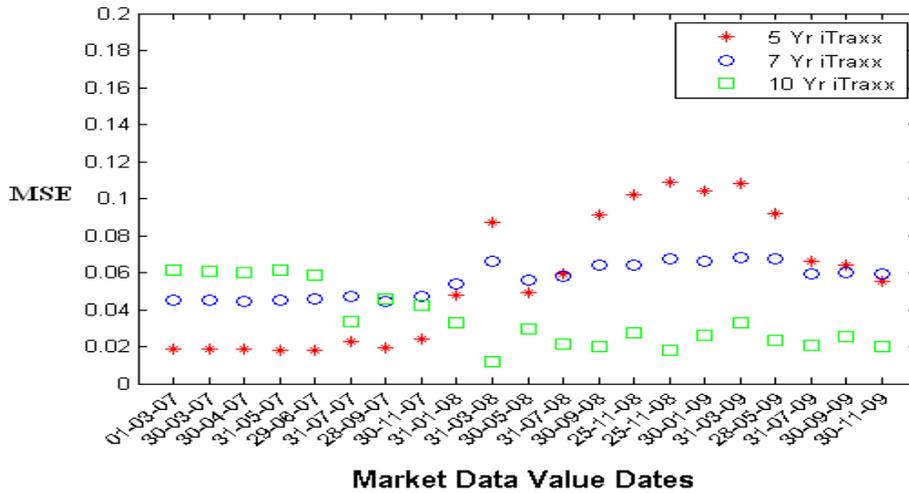


Figure 5.6: The function MSE is the sum of the squares of the difference between the market expected loss EM_j and the percolation model expected losses E_j for each tranche j . The smaller the value of MSE the better the fit between the percolation model and market data base correlations. Here we graph the minimum MSE value for all iTraxx tranche fixings.

we introduce a constraint on this model such that $pperc$, i.e. the bond percolation parameter, ranges in values between 0 and $pperc = \frac{z}{N-1}$. We constrain the bond percolation parameter below a threshold value to ensure we remain in the regime where theory is valid (i.e. no giant connected component). The phase transition level is 0.008 for our iTraxx Europe tranche fixings as there are 125 names in the market data portfolios.

We repeat the steps outlined previously to compute the best fitting percolation parameters to match the percolation model expected losses to market data expected losses when $z \leq 1$, see Section 5.4. In Figures 5.7 - 5.12 we plot our results for the unknown percolation parameters of the constraint bond percolation model.

The constraint $pperc$ percolation model shows similar behaviour to the unconstrained case in computing the fraction of optimists a , the fraction of percolationists b , and the fraction of catastrophists c . In addition, the ρ_0 and the function $f(b)$ show similar behaviour to the unconstrained case. All data for MSE generally lie below 0.1 and constraining the $pperc$ parameter does not drastically influence the function $f(b)$. Observing the bond percolation parameter $pperc$ all iTraxx tranche fixings for 5, 7 and 10 year datasets are approximately 0.0035. We apply the constraint $pperc$ condition with the aim of being able to generalise the value of the parameters in the percolation model e.g. the bond percolation parameter $pperc$. In the following section we compare the Gaussian copula, Student t4 copula and percolation model to market data for a series of iTraxx tranche fixings.

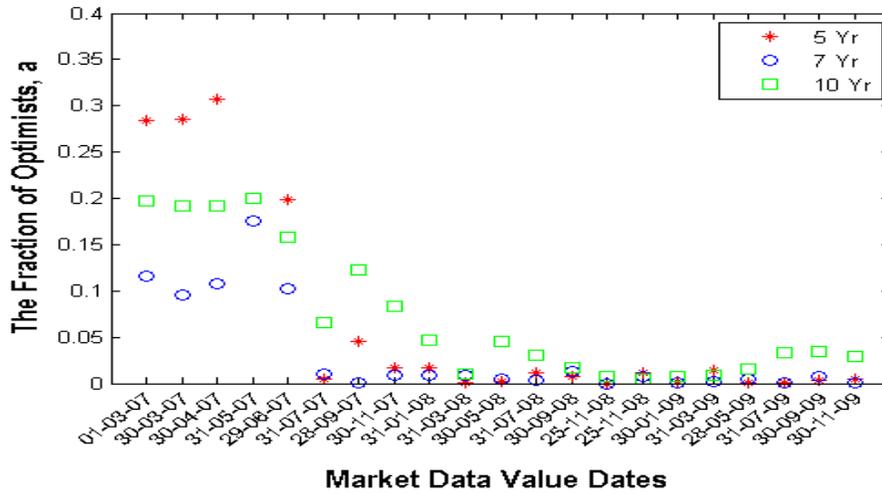


Figure 5.7: The fraction of optimists a that minimises the function MSE with the bond percolation parameter $pperc$ constrained. The market data comprises of 5Yr, 7Yr and 10Yr iTraxx Europe tranche fixings.

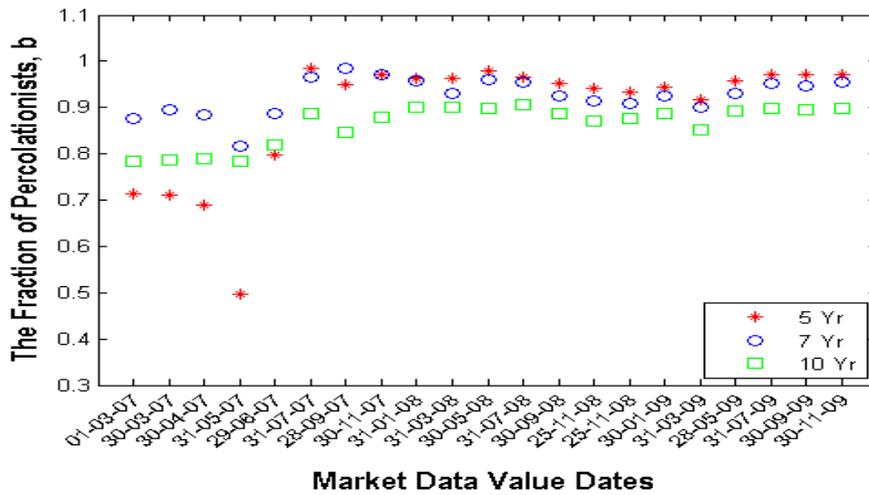


Figure 5.8: The fraction of percolationists b that minimises the error function MSE with the bond percolation parameter $pperc$ constrained.

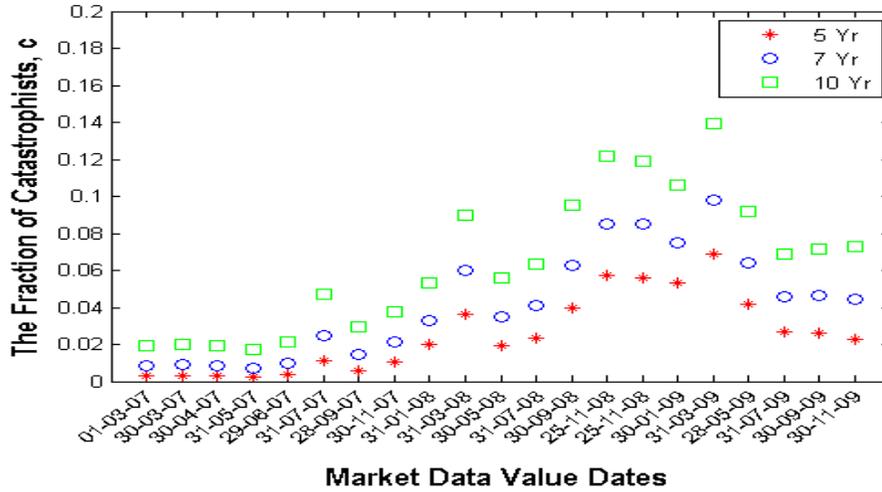


Figure 5.9: The fraction of catastrophists c that minimises the function MSE with the bond percolation parameter $pperc$ constrained.

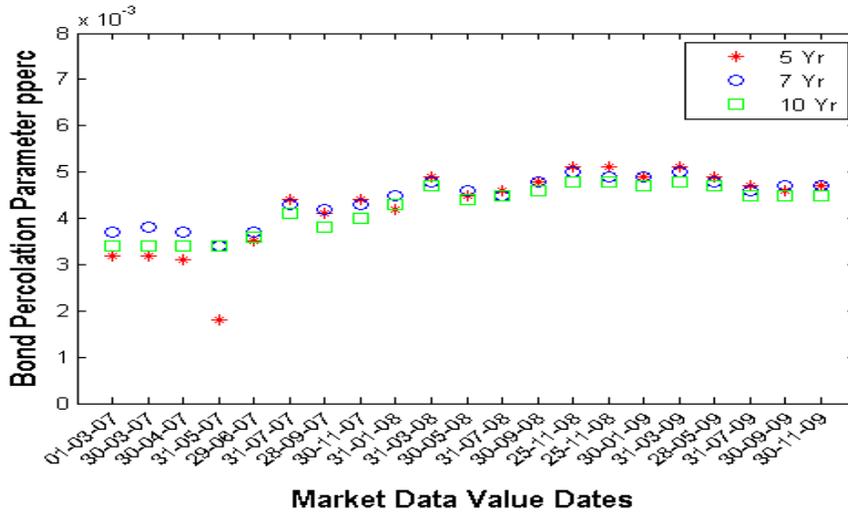


Figure 5.10: The bond percolation parameter $pperc$ that minimises the function MSE with the bond percolation parameter $pperc$ constrained. Note, the relationship between $pperc$ and z is $z = (pperc)(N)$ as given in Section 5.1.

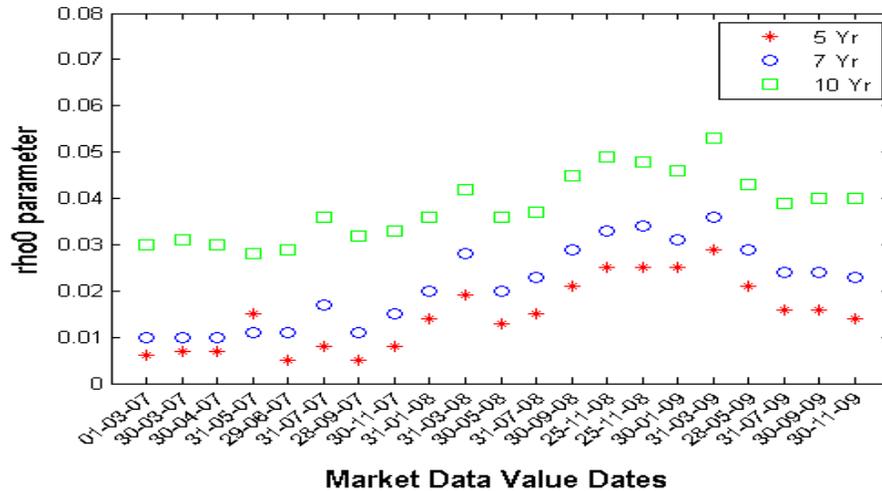


Figure 5.11: The ρ_0 parameter that minimises the function MSE with the bond percolation parameter $pperc$ constrained. Note, the relationship between ρ_0 and z_1 is $z_1 = (\rho_0)(N - 1)$ as given in Section 5.1.

5.6 Comparing the percolation model and copula models to market data

In order to compare the three large portfolio models introduced in Chapters 3 and 5 an example of a tranching iTraxx Europe is considered. We choose an iTraxx Europe portfolio with different maturity dates i.e. 5 years, 7 years and 10 years to maturity. The empirical testing of the three models is done with a data set from www.creditfixings.com.

The reference portfolio consists of 125 names with tranches having attachment/detachment at 3%, 6%, 9%, 12% and 22%. The investor of the equity tranche receives an upfront fee that is quoted in the market and an annual spread of 500bps quarterly. The settlement date of the series of

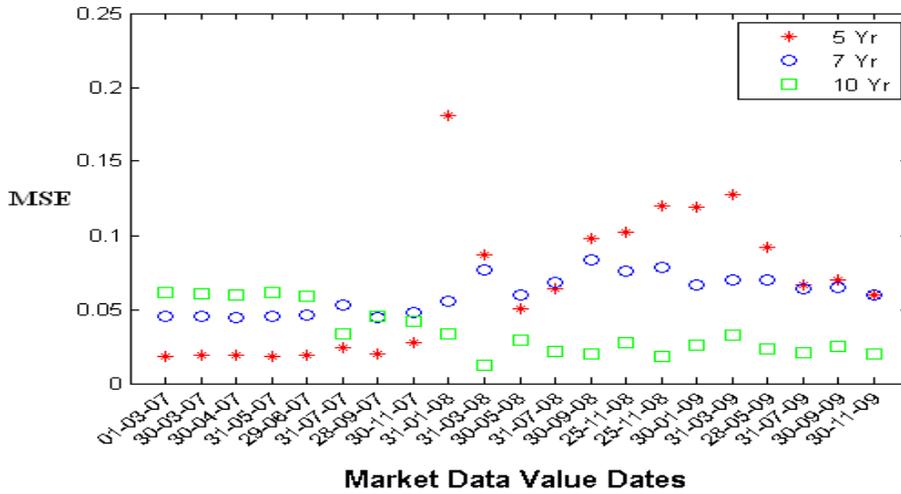


Figure 5.12: The function MSE is the sum of the squares of the difference between the market expected loss EM_j and the percolation model expected losses E_j for each tranche j . The smaller the value of MSE the better the fit between the percolation model and market data base correlations. Here we graph the minimum error function MSE with the bond percolation parameter $pperc$ constrained.

this index is 31 July 2009 and matures in 2014, 2016, 2019 depending on the portfolio duration. The constant recovery rate is assumed to be 40%.

In Figures 5.13 - 5.15 we compare the three models outlined in this thesis to market data sets. It can be seen that the Student t4 copula produces a good fit to market data overall except for the most senior tranches and the Gaussian copula performed badly, especially for the more senior tranches in the portfolio. The percolation model matches to market data better than the two copula models and has the ability to fit well to senior tranches. It is evident from the figures that the our percolation model has the best fit to market data.

In this chapter we have extended the infinite network model presented

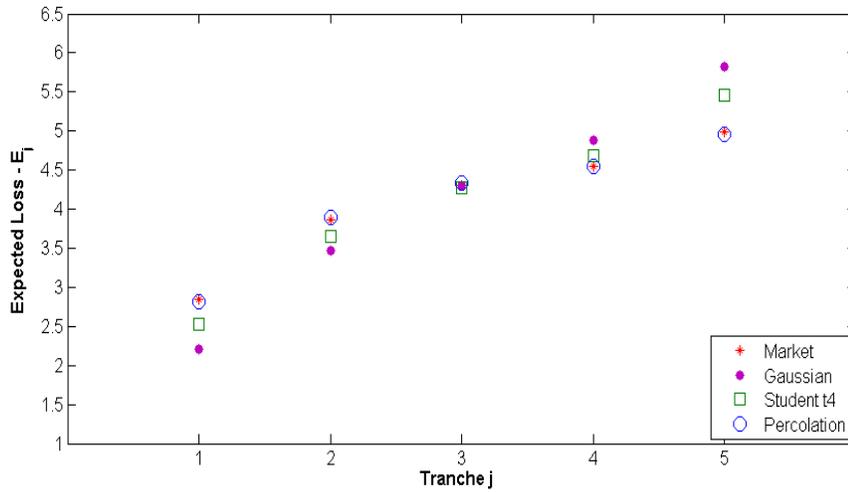


Figure 5.13: Comparing the percolation model, Gaussian and Student t4 copulas to market data. This market data set is an iTraxx Europe tranching CDO maturing in 5 years.

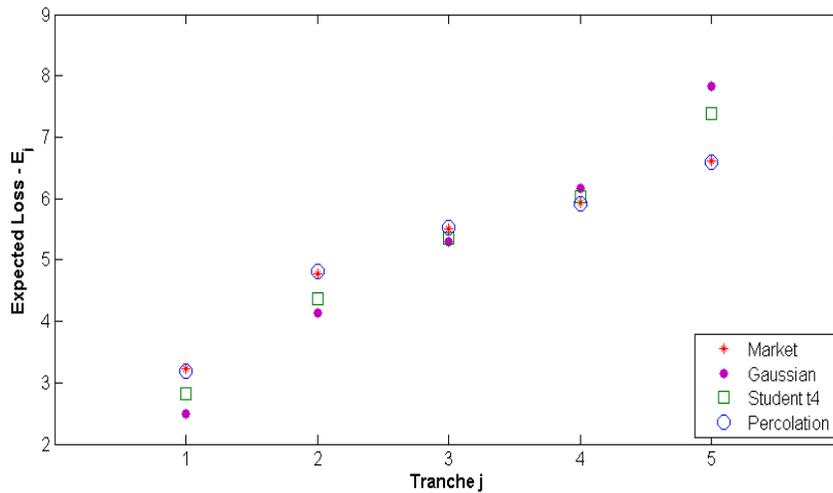


Figure 5.14: Comparing the percolation model, Gaussian and Student t4 copulas to market data. This market data set is an iTraxx Europe tranching CDO maturing in 7 years.

in Chapter 4 to compute the probability of default Q_n for a Poisson random graph. We outlined how to compute the percolation model parameters for

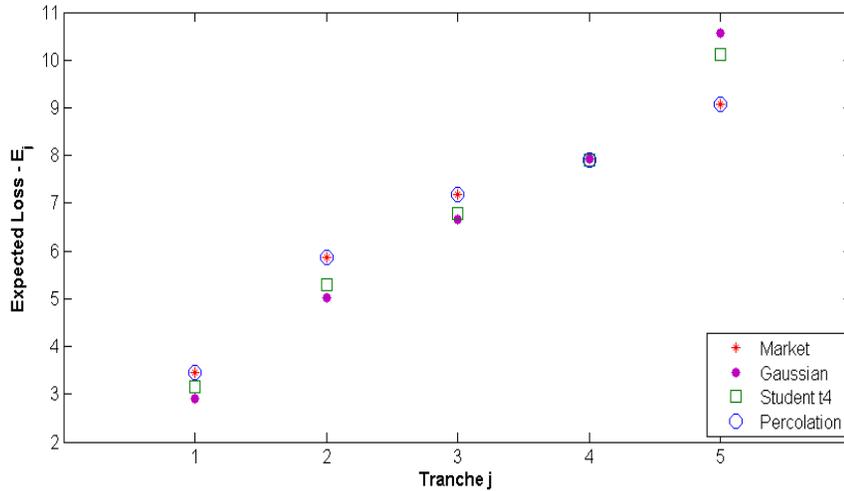


Figure 5.15: Comparing the percolation model, Gaussian and Student t4 copulas to market data. This market data set is an iTraxx Europe tranching CDO maturing in 10 years.

the network model and fitted the percolation model to market data to compute these percolation parameters for our data sets.

In the constrained percolation model case we looked to compute the percolation parameter values for the model when the bond percolation parameter $pperc \leq 0.008$. It is at this value and higher that the giant component of a network forms.

Comparing the percolation model, Gaussian and Student t4 copulas to market data we observed that the percolation model gave the best fit to the market data of CDO portfolios. The Gaussian and Student t4 copulas didn't match as well to the market data especially for more senior tranches.

Chapter 6

Summary and future work

Beginning at Chapter 1 we reviewed some of the predominant models in the literature used to price CDO tranches. We identified that in order to price CDOs portfolios a key input is an approximation of the default dependency (default correlation) between the underlying names in the CDO portfolio. Using these default dependencies we can simply transform them into expected losses for each tranche j . The liquidity of standardised CDO indices prior to the credit crisis led to the market convention of pricing tranches in terms of implied correlations. In general, the implied correlation is found to be higher in equity and senior tranches than for mezzanine tranches. This is known as the “correlation smile”. McGinty et al. (2004b) proposed base correlations as a way of solving such problems and this has since become an industry standard method.

As previously discussed, the critical input in the pricing of a CDO is an estimate of the default dependence between the underlying names in the portfolio. In Chapter 2 we reviewed the one factor copula model

and extended this model in Chapter 3 by specifying a Gaussian and a Student t4 distribution for the copula model. The Gaussian copula model introduced by Li (2000) is the market standard for pricing CDOs due to its simplicity and tractability. As an improvement, Hull and White (2004) suggested the Student t copula because of its heavier tails than the normal distribution and its better fit to market data than the Gaussian copula. To compare the LHP models introduced previously we considered a tranching iTraxx Europe CDO with 5 year maturity. From our results it was seen that the Student t4 copula produced a good fit to market data overall except for the most senior tranches and matches better to tranche market values. The Gaussian copula performed badly, especially for the more senior tranches in the portfolio. As an alternative method to computing the default distribution for a CDO portfolio we investigated the possible replacement of copula methods by network based models.

In Chapter 4 we presented the finite and infinite network models used to model the default dependency between names in a portfolio. We link the copula default distribution to network theory by suggesting that the relative component sizes in a network is related to contagion on a network. Depending on how many names a given name is connected to (component size) and assuming contagion spreads along an edge with probability p , we can compute the probability of default using network percolation. We found that in the finite network model round off numerical errors were generated by the multiplication of large binomial coefficients with smaller terms and due to these errors continuing onwards we used the

infinite network model as our standard network model. In Chapter 5 we extended the infinite network model to compute the probability of default Q_n for a CDO portfolio. We outlined the parameters of the percolation model and also outlined the methodology to compute the fraction of percolationists, catastrophists and optimists. We linked market data to the Large Pool Model to which we can then applied various different models e.g. copula, percolation etc. Using the market data for a series of iTraxx Europe tranches we implied the best fitting percolation model parameters from the model. We then constrained the bond percolation parameter below a threshold value ($z = 1$) to ensure we remain where theory is valid (i.e. no giant connected component) and again implied percolation model parameters for the model.

Finally in order to compare the percolation, Gaussian and Student t4 models a tranced iTraxx Europe CDO was considered. In summary our results showed that the Student t4 copula produces a good fit to market data overall, except for the most senior tranches. The Gaussian copula model performed badly, especially for the more senior tranches in the portfolio. The percolation model proposed matched to market data better than the two copula models and had the ability to match well to senior tranches where Student t4 and Gaussian copulas failed to. From our analysis, the percolation model was the best fit to market data.

To summarise, we have in this thesis

- reviewed the factor copula model and alternatives models in the literature that have been suggested to price CDO portfolios.

- introduced the one factor large homogeneous copula model as a method to computing P_n the probability of default, without specifying the default distribution.
- applied the Gaussian distribution and Student t4 distribution to the factor copula model and computed the probability of default P_n using Monte Carlo and theoretical methods. We compared these two copula models in their ability to fit to market tranche expected losses E_j .
- presented some early network models proposed by Paul Erdős and Alfréd Rényi and computed the distribution of component sizes in an infinite and finite network.
- extended the infinite network model to compute Q_n , the probability of default distribution.
- determined that the percolation model provides a better fit to market data than the Student t4 and market standard Gaussian copula.

The aim of this thesis was to develop an alternative model based on network theory to price a tranching CDO portfolio. Haldane (2009) highlighted the importance of networks in finance and the percolation model proposed in this thesis is a first step in the use of network theory in the modelling of CDO prices.

At present the number of parameters in the percolation model proposed is greater than the number of parameters in copula based models. This could be seen as an unfair comparison and further work is needed in

reducing the number of parameters in the percolation model. One method could be computing the average parameter values on a large and diverse data set which could possibly lead to certain parameters having constant values - see, for example, Figure 5.10. Alternatively using information theory one could correct for this by using an Akaike information criterion (AIC).

In conclusion, we hope our discussion and analysis of the percolation model has benefited the reader by suggesting network models as a possible approach to pricing correlated financial products.

Chapter 7

Appendix A

Here we reproduce some of the key elements of the MATLAB code underpinning the copula models introduced.

7.1 Comparing Student t4 and Gaussian copulas

```
%comparingstudenttandgaussiancopula
%studentt4 copula

N=125;                               %names in a portfolio
p=0.95;                               %input probability
a=0.1;                                %correlation
delM=0.1;                             %M divided into intervals of equal length
minM=-20;
```

```

maxM=20;
K=fzero(@(x) myfun(x,p),1); %extracted from the myfun.m file
sumofpnm=zeros(1,N+1);
M=minM:delM:maxM; %M ranges from -4 to +4
[b,c] = size(M); %the size of M
for i=1:1
    for j=1:c
        Z(j)=(K-((sqrt(a))*M(j)))/sqrt(1-((a)));
        dM(i,j)=(0.5)+(0.5*(((Z(j))*3+((Z(j))^2))/((2+((Z(j))^2))^(3/2))));
        probM(i,j)=((3)/((2+((M(j))^2))^(5/2)));
        for u=0:N
            term1 = prod( ((u+1):(N)) ./ (1:(N-u)) ); % nchoosek (N,u)
            pnm(1,u+1)=(term1)*((dM(j))^(u))*((1-(dM(j)))^(N-u));
        end
        sumofpnm=sumofpnm+ pnm*(delM)*(probM(j));
    end
end
for ry=1:126
    rr(ry,1)=sumofpnm(1,ry);
end
%gaussian copula
minM2=-4;
maxM2=4;
xbar=fzero(@(x) f(x,p),1);
sumofpnm2=zeros(1,N+1);
M2=minM2:delM:maxM2;
[d,r] = size(M2);
for i2=1:1
    for j2=1:r

```

```

dM2(i2, j2)=0.5+(0.5)*(erf((xbar-((sqrt(a))*(M2(j2))))/
((sqrt(2))*(sqrt(1-(a))))));
probM2(i2, j2)=(1/(sqrt(2*pi)))*(exp(-(M2(j2))^2/2));
for u2=0:N
    term2 = prod( (u2+1):(N) ./ (1:(N-u2)) );% nchoosek (N,u2)
    pnm2(1,u2+1)=(term2)*((dM2(j2))^(u2))*((1-(dM2(j2)))^(N-u2));
end
sumofpnm2=sumofpnm2 + pnm2*(delM)*(probM2(j2));
end
end
for ty=1:126
    br(ty,1)=sumofpnm2(1,ty);
end

```

7.2 Computing base correlations from expected losses

```

function base corr values=E to base(E1, E2, E3, E4, E5, N, p);
% A function to generate base correlation values. Given E values
% for a given N and p.
rho=0:0.001:0.99;
[le,re]=size(rho);
R=0.4; %recovery rate(common to all names)
D=[0.03 0.06 0.09 0.12 0.22]; %detachment points
sumofpnm = zeros(re,N+1);
for jk=1:re
    K=fzero(@(x) f(x,p),1); %threshold value

```

```

delM=0.1;
minM=-4;
maxM=4;
M=minM:delM:maxM;
[b,c] = size(M);
edges = 0:1:N;
marketvalue=[E1 E2 E3 E4 E5];
probn=zeros(1,N+1);
    for j=1:c
        dM(j)=0.5+(0.5)*(erf((K-((rho(jk))*M(j))))/
            ((sqrt(2))*(sqrt(1-((rho(jk))^2))))));
        probM(j)=((1/(sqrt(2*pi)))*(exp((-M(j))^2/2)));
        for u=0:N
            term1 = prod( ((u+1):(N)) ./ (1:(N-u)) ); % nchoosek(N,u)
            pnm(1,u+1) = (term1)*((dM(j))^(u))*((1-(dM(j)))^(N-u));
        end
        probn=probn + pnm*(delM)*(probM(j));
    end
sumofpnm(re,:) = sumofpnm(jk,:);
sumofpnm(jk,:)= probn;
for itr=1:5
    fDN(itr) = floor(((D(itr))*N)/(1-R));
    p1(jk,itr)=0;
    p2(jk,itr)=0;
    for k1= 1:fDN(itr)
        p1(jk,itr)=p1(jk,itr)+((1-R)*(sumofpnm(jk,k1+1))*(k1));
        %sumofpnm(k1+1) as our matrix is running from 1-126
    end
    for k2= fDN(itr)+1:N

```

```

    p2(jk,itr)=p2(jk,itr)+((((D(itr))*N)*(sumofpnm(jk,k2+1)));
    %sumofpnm(k2+1) as our matrix is running from 1-126
end
Evalue(jk,itr)=p1(jk,itr)+p2(jk,itr);
end
for ir=1:5
    etor(jk,ir)= (abs((Evalue(jk,ir))-(marketvalue(ir))));
end
[minvalue,index]=min(etor);
bestrhoindex=rho(index);

```

Chapter 8

Appendix B

Here we reproduce some of the key elements of the MATLAB code underpinning our percolation model.

8.1 A function to generate the N by N matrix $B_{1,n}$

```
function Bmat=create Bmat(N);
sv=1:N;
row1=(sv.(sv-1))./factorial(sv);
Amat=zeros(N,N);g
Amat(1,:)=row1;
Bmat=Amat;
for i=2:N
    rowabove=Amat(i-1,:);
```

```

for n=i:N
    sumval=0;
    for r=1:n-1
        sumval = sumval+row1(r) * rowabove(n-r);
    end % for r
    Amat(i,n)=sumval;
end % for n
Bmat(i,:)=Amat(i,:)/factorial(i-1);
end % for i

```

8.2 A function to generate the best ρ_0 and *pperc* parameters

```

function [bestrho0,bestpperc]=fit_perc_model_3_rho0_pperc_diff(bc1, bc2,
bc3, bc4, bc5, N, pval);
%-----
% Optimises over pperc
% and rho0 parameters and returns the value
% for pperc and rho0
% which minimizes the error term.
% Input : N, recovery rate, BC values, tranche detachment points
% Output : - rho0 and pperc value which minimises the error term.
%
%
%
%-----
Bmat=create_Bmat(N); % only needs to be run once for a given N

```

```

rec=0.4; % recovery rate
bcinput=[bc1 , bc2 , bc3 , bc4 , bc5 ];
Evalue=base to E(bcinput(1), bcinput(2) , bcinput(3) , bcinput(4)
, bcinput(5) , N, pval);
losslist= [Evalue(1) , Evalue(2), Evalue(3), Evalue(4),
Evalue(5)]; %$E-j$ for each tranche, from market data
detachments=[ 3,6,9,12,22 ];
% detachment points of the tranches

%input a range of rho0 and pperc values
rho0=0.001:0.001:0.1;%0.9;
[le1,re1]=size(rho0);
pperc=0.001:0.001:0.06;%0.9;
[le2,re2]=size(pperc);

errorterm=zeros(re1,re2); %the size of the error matrix

for jk1=1:re1
for jk2=1:re2

z1=rho0(jk1)*(N-1);
z=pperc(jk2)*N;
histin=histgenfn(N, (pperc(jk2))*N, rho0(jk1)*(N-1), Bmat);
pkorg=histin/sum(histin);
kv=1:length(pkorg);
d=detachments/100;
individualtranche = 1:1:5;
for tranche=1:5
j=floor(d(tranche)*N/(1-rec));

```

```

Fj(tranchenum)=losslist(tranchenum) - d(tranchenum)*N*pval;
cj(tranchenum) = (1-rec)*sum(kv(1:j).*pkorg(1:j))
+ d(tranchenum)*N*sum(pkorg(j+1:N)) - d(tranchenum)*(pkorg*kv');
end % for tranchenum
avall=(cj*Fj')/(cj*cj');
aval=min(1,avall); % ensure $aval \leq 1$
cval1=pval-aval/N*(pkorg*kv'); % determine cval from Expected Loss
% (via pval)
cval=min(1-aval,cval1); % ensure $cval \leq 1$

pk=aval*pkorg;
pk(N)=pk(N)+cval;
pk0=1-sum(pk); % sets fraction of optimists to be remainder

probn=pk ;

for tranchenum=1:5
    j=floor(d(tranchenum)*N/(1-rec));
    Eval(tranchenum)=(1-rec)*sum(kv(1:j).*pk(1:j))
    + d(tranchenum)*N*sum(pk(j+1:N));
end % for tranchenum

errL2=sqrt( sum( (Eval-losslist) . 2 ) );
errorterm(jk1,jk2)=errL2;
ELmat(jk1,jk2)=(1-rec)*pk*kv'/N;
end
end

```

```

[Val,Ind]=min2d(errorterm);
%min2d finds the min error value in the 2d matrix and outputs
%the minimum value and its index in the matrix
bestrho0=rho0(Ind(1));
bestpperc=pperc(Ind(2));
%best rho0 value and pperc value that minimizes the error term
aval;
cval;
pk0;

```

8.3 A function to generate a , b , c given $pperc$ and ρ_0 . This function also outputs the percolation model expected losses

```

function Eval=fit_perc_model(rho0, pperc, Gbc1, Gbc2, Gbc3,
Gbc4, Gbc5, Sbc1, Sbc2, Sbc3, Sbc4, Sbc5, N, pval);
% Calculate best-fit a and c values, given pperc and rho0.

Bmat=create_Bmat(N); % only needs to be run once for a given N

rec=0.4; % recovery rate
bcinput=[Gbc1 , Gbc2 , Gbc3 , Gbc4 , Gbc5 ];
Evalue=base_to_E(bcinput(1), bcinput(2) , bcinput(3)
, bcinput(4) , bcinput(5) , N, pval);
losslist= [Evalue(1) , Evalue(2), Evalue(3), Evalue(4),
Evalue(5)]; %Ej for each tranche, from market data

```

```

detachments=[ 3,6,9,12,22 ]; % detachment points of the tranches

z1=rho0*(N-1);
z=pperc*N;
histin=histgenfn(N,pperc*N,rho0*(N-1),Bmat);
pkorg=histin/sum(histin);
kv=1:length(pkorg);
d=detachments/100;
individualtranche = 1:1:5;
for tranchenum=1:5
j=floor(d(tranchenum)*N/(1-rec));
Fj(tranchenum)=losslist(tranchenum) - d(tranchenum)*N*pval;
cj(tranchenum) = (1-rec)*sum(kv(1:j).*pkorg(1:j))
+ d(tranchenum)*N*sum(pkorg(j+1:N)) - d(tranchenum)*(pkorg*kv');
end % for tranchenum
avall=(cj*Fj')/(cj*cj');
aval=min(1,avall) % ensure aval  $\leq 1$ 
cval1=pval-aval/N*(pkorg*kv');
% determine cval from Expected Loss (via pval)
cval=min(1-aval,cval1) % ensure cval  $\leq 1$ 

pk=aval*pkorg;
pk(N)=pk(N)+cval;
pk0=1-sum(pk)
% sets fraction of optimists to be remainder

for tranchenum=1:5
j=floor(d(tranchenum)*N/(1-rec));
Eval(tranchenum)=(1-rec)*sum(kv(1:j).*pk(1:j))

```

```
        + d(tranchenum)*N*sum(pk(j+1:N));
end % for tranchenum

errL2=sqrt( sum( (Eval-losslist).^2 ) )

base corr values=E to base(Eval(1), Eval(2),
Eval(3), Eval(4), Eval(5), N, pval);
base corr values;
Eval;
```

Bibliography

- Andersen, L. and J. Sidenius (2005). Extensions to the Gaussian copula: Random recovery and random factor loadings. *Journal of Credit Risk* 1(1), 29–70.
- Andersen, L., J. Sidenius, and S. Basu (2003). All your hedges in one basket. *Risk*, 67–72.
- Antonov, A., S. Mechkovy, and T. Misirpashaevz (2005). Analytical techniques for synthetic CDOs and credit default risk measures. Working paper, Numerix.
- Baxter, M. (2006). Levy process dynamic modelling of single name credits and CDO tranches. Working Paper, Nomura Fixed Income Quant Group.
- Black, F. and J. Cox (1976). Valuing corporate securities: Some effects of bond indenture provisions. *Journal of Finance* 31(2), 351–67.
- Bornholdt, S. and H. G. Schuster (2003). *Handbook of Graphs and Networks*. Wiley-VCH.
- Brigo, D., A. Pallavicini, and R. Torresetti (2007). Calibration of CDO tranches with the dynamical generalized-Poisson loss model. *Risk* 20(5), 70–75.
- Burtschell, X., J. Gregory, and J. Laurent (2007). Beyond the Gaussian copula: Stochastic and local correlation. *Journal of Credit Risk* 3(1), 31–62.

- Burtschell, X., J. Gregory, and J. Laurent (2009). A comparative analysis of CDO pricing models under the factor copula framework. *Journal of Derivatives* 16(4), 9–37.
- Crouhy, M., R. Jarrow, and S. Turnhill (2008). The subprime credit crisis of 2007. *Journal of Derivatives* 16(1), 81–110.
- Duffie, D. and N. Garleanu (2001). Risk and valuation of collateralized debt obligations. *Financial Analysts Journal* 57(1), 41–59.
- Giesecke, K. (2003). A simple exponential model for dependent defaults. *Journal of Fixed Income* 13(3), 74–83.
- Glasserman, P. and S. Suchintabandit (2007). Correlation expansions for CDO pricing. *Journal of Banking and Finance* 31(3), 1375 – 1398.
- Gregory, J. and J. Laurent (2004). In the core of correlation. *Risk* 17(10), 87–91.
- Gregory, J. and J. Laurent (2005). Basket default swaps, CDOs, and factor copulas. *Journal of Risk* 7(4), 103–122.
- Haldane, A. G. (2009). Rethinking the financial network. <http://www.bankofengland.co.uk/publications/speeches/2009/speech386.pdf>.
- Hofert, M. and M. Scherer (2008). CDO pricing with nested Archimedean copulas. <http://www.mathematik.uni-ulm.de/numerik/preprints/2008/CDOpricingAC.pdf>.
- Hull, J., M. Predescu, and A. White (2005). The valuation of correlation-dependent credit derivatives using a structural model. SSRN Electronic paper collection, <http://ssrn.com/abstract=686481>.
- Hull, J. and A. White (2004). Valuation of a CDO and an n^{th} to default CDS without Monte Carlo simulation. *Journal of Derivatives* 12(2), 8–23.

- Hull, J. and A. White (2006). Valuing credit derivatives using an implied copula approach. *Journal of Derivatives* 14(2), 8–28.
- Hull, J. and A. White (2008a). Dynamic models of portfolio credit risk: A simplified approach. *Journal of Derivatives* 15(4), 9–28.
- Hull, J. and A. White (2008b). An improved implied copula model and its application to the valuation of bespoke CDO tranches. *Journal of Investment Management*. Forthcoming, Third Quarter 2010.
- Kalemanova, A., B. Schmid, and R. Werner (2007). The normal inverse Gaussian distribution for synthetic CDO pricing. *Journal of Derivatives* 13(3), 80–93.
- Lehrert, N., F. Altrock, S. Rachev, S. Truck, and A. Wilch (2005). Implied correlations in CDO tranches. http://www.pstat.ucsb.edu/research/papers/cdo_paper.pdf.
- Li, D. (2000). On default correlation: A copula approach. *Journal of Fixed Income* 9(4), 43–54.
- Lucas, D. (2001). CDO handbook. JP Morgan Technical Paper.
- McGinty, L. (2004). A model for base correlation calculation. JP Morgan Technical Paper.
- McGinty, L., R. Ahluwalia, M. Watts, and E. Beinstein (2004a). Credit correlation: A guide. JP Morgan Technical Paper.
- McGinty, L., R. Ahluwalia, M. Watts, and E. Beinstein (2004b). Introducing base correlations. JP Morgan Technical Paper.
- Meneguzzo, D. and W. Vecchiato (2004). Copula sensitivity in collateralized debt obligations and basket default swaps. *The Journal of Futures Markets* 24(1), 37–70.

- Moosbrucker, T. (2006). Pricing CDOs with correlated variance gamma distributions. Department of Banking, University of Cologne. Research Report.
- Mortensen, A. (2006). Semi-analytical valuation of basket credit derivatives in intensity-based models. *Journal of Derivatives* 13(4), 8–26.
- Newman, M. E. J. (2003). Properties of highly clustered networks. *Physical Review E* 68(026121).
- Newman, M. E. J. (2007). Component sizes in networks with arbitrary degree distributions. *Physical Review E* 76(045101-3).
- Newman, M. E. J., S. H. Strogatz, and D. J. Watts (2001). Random graphs with arbitrary degree distributions and their applications. *Physical Review E* 64(026118).
- O’Kane, D. and M. Livesey (2004). Base correlation explained. Fixed Income Quantitative Research, Lehman Brothers.
- O’Kane, D. and L. Schloegl (2005). An analytical portfolio credit model with tail dependence. Quantitative Credit Research, Lehman Brothers.
- Papageorgiou, E. and R. Sircar (2009). Multiscale intensity models and name grouping for valuation of multi-name credit derivatives. *Applied Mathematical Finance* 16(4), 353–383.
- Rosen, D. and D. Saunders (2009). Valuing CDOs of bespoke portfolios with implied multi-factor models. *Journal of Credit Risk* 5(3), 3–36.
- Schloegl, L. and D. O’Kane (2005). A note on the large homogeneous portfolio approximation with the Student t copula. *Finance and Stochastics* 9(4), 577–584.
- Schönbucher, P. and D. Schubert (2001). Copula-dependent default risk in intensity models. Bonn University. Working Paper.

- Sidenius, J., V. Pittenbarg, and L. Andersen (2008). A new framework for dynamic credit portfolio loss modelling. *International Journal of Theoretical and Applied Finance* 11(2), 163–197.
- Sklar, A. (1959). Fonctions de répartition à n dimensions et leurs marges. *Publications de l'Institut de Statistique de l'Université de Paris* 8, 229–231.
- Vasicek, O. (1987). Probability of loss on loan portfolio. KMV Corporation.
- Yang, J. (2006). Saddlepoint approximation method for pricing CDOs. *Journal of Computational Finance* 10(1), 1–20.