University of Twente
Enschede - The Netherlands

Master's Thesis in Financial Engineering & Management

Effects of downgrade momentum on measuring credit migration risks

Thursday, 24 September 2009

VIVIEN DJIAMBOU

Industrial Engineering and Management
School of Management & Governance

University of Twente - Enschede

TABLE OF CONTENTS

Executive Summary

The purpose of this study is to investigate the effects of “downgrade momentum” in credit migration matrices that is observed in rating transition data. The downgrade momentum refers to a non-Markovian effects in which recently downgraded obligors are at increasing risk of experiencing further downgrades. The developed extended transition matrices sensitive to downgrade momentum are used to compute the credit risks for long and short term horizons and later compared to an extended transition matrix insensitive to downgrade momentum. The insensitive extended transition matrix referred as benchmark for comparison has a one-to-one correspondence with the annual S&P’s transition matrix. The first model of the downgrade sensitive extended transition matrix has a one-to-one correspondence with the annual transition matrix and recently downgraded probabilities increased by a factor of three for those that will be downgraded in the next period while the second model represents an adjustment of the first model such that it can be aggregated back to the initial annual S&P’s transition matrix. The calibration of the second model is based on the number of obligors within each rating category. In addition, these extended transition matrices are applied in the measurement of short-term credit migration risks referred in Basel II as Incremental Risk Charge. To maintain a constant level of risk, the portfolio is rebalanced each quarter to match with the initial portfolio with the condition that recently downgraded instruments cannot be sold on the market. We illustrate the significance of the downgrade momentum effects on credit migration matrices based on the algorithm developed in MatLab®. The implication of this research affects both short-term and long-term horizons portfolios which are strongly influenced by the dynamics of the rating transitions. The result of this study is relevant for Basel II because banks that neglect the downgrade momentum might underestimate the risk level of their positions.

Keywords: Credit migration risk, rating drift, downgrade momentum, transition matrices, Value-at-Risk

Acknowledgements

I would like to express my gratitude to all the people who helped me in one way or another in the completion of MSc. Project in Credit Migration Risks.

First, I would like to thank my supervisor Dr. Berend Roorda for his invaluable support. Throughout the project, he has given me a lot of freedom to work on my thesis. His professionalism, knowledge, interest in the research, and guidance has helped me to complete this study within seven months. Special thanks to Co-Supervisor, Mr. Emad Imreizeeq for supervising the project and helping me understand fundamentals of credit risk and his suggestions kept me on track on writing my thesis.

Also, I want to take this opportunity to address gratitude to Dr. Hemo Jorna and also to the Faculty for the help they have been provided to support my effort.

My sincere thanks to my friends Pankaj Chauhan, Avijit Kumar, and Umar Zaghum who have made my study time at the University of Twente very enjoyable and have aided my research by creating a stimulating and productive environment.

Finally, I wish to extend my gratitude to my family for the invaluable sacrifices and support. I cannot describe how thankful I am for the numerous ways in which they have always supported me. Their unconditional support and encouragement has made it possible for me to complete this master program.

List of tables

List of figures

1. Introduction

Over the past decade, banks have developed several techniques and tools to improve the measurement of the financial risks associated with their business activities. Particularly, banks have been interested to better capture the likelihood that borrowers will not be able to meet their financial obligations. The ability for borrowers to repay the interest and principal in a timely manner defined the credit quality of the borrowers. Credit risks refer to risk of default or of reduction in the value of debt instruments (e.g. loans) and credit-related securities due to unexpected changes in the underlying credit quality of the borrowers and counterparties over time, see Duffie and Singleton (2003). This special interest for banks to improve their measurement of credit risks stems from their high exposures to potential losses that may arise in the events of ratings migrations or default. The unprecedented growth of innovative financial products (e.g. Credit Default Swaps CDSs and Collateralized Debt Obligations CDOs) has increased significantly the exposure of banks to its credit portfolio. The inability to adequately measure the credit risks can rapidly lead banks to failure compare to market and operational risks where failure might lead to several small losses. Luigi Zingales (2008) shows how the recent bankruptcy of Lehman Brothers during the financial turmoil where due to the failure to measure the real exposures to CDS products. Lehman Brother sold protections or CDS to investors who will receive a certain payoff if a credit event occurs in particular when the instruments (e.g. mortgages) default in exchange for a series of payments.

Current regulations encourage banks to assess their credit risks through the implementation of the Basel Capital Accord (Basel II), which provides a framework for modeling credit risks and encourages banks to develop internal credit risk models with the agreement from their respective supervisory authorities (Basel Committee on Banking Supervision, 2001). Such a framework helps to further strengthen the soundness and stability of banks and ensure the adequate flow of capitals to individuals and businesses. The supervisory authorities are concerned about the soundness of the financial system because financial products can significantly contribute to the improvement and growth of the entire economy when adequately controlled or can lead to financial turmoil with disastrous consequences to the local and global economies.

Credit risk models differ in their fundamental assumptions. For example, credit losses can be defined based on the default models (e.g. loan defaults) or on the multi-state models with ratings migrations from obligors (typically firms) over time amongst a set of possible states. In the case of default models, banks are concerned with the duration period that the borrowers will take to default. However in a multi-state model, banks that have a portfolio of loans are not only influenced by the potential default of their clients but also their earnings may be severely affected by the rating downgrades or upgrades of their clients. In the present study, the focus will be on credit losses arising from rating migrations that banks experiences due to the evolution of the creditworthiness of the borrowers or counterparties through time.

Credit migration or rating transition matrix is defined as the likelihoods of changes in credit quality of obligors over a specific time period that is usually a year. Such changes lead to ratings upgrade or downgrade that affects the evaluation of the debt instruments as it is mark-to-market. Credit migrations affect the fees that banks charge to its clients and constitute an important source of revenues for banks where clients with high credit quality are charged relatively lesser fees than clients with low credit quality to account for the risks of non-repayments. It also affects the decisions to grant credit requests from different clients because banks aim to have a diversified portfolio. Andersson and Vanini (2008) illustrates that depending on the loans contracts, credit migrations might lead to losses in the event of default, opportunity of losses with the downgrades of obligors to non-default states which decreases the prices at which other banks are willing to purchase the same instruments, or potential gains with the upgrades of obligors which increases the prices at which the same instruments can be sold, The sophistication of the models developed by banks in modeling credit migration risks come from the straightforward implementation of Basel II framework which allow banks to determine the required amount of capitals that banks should hold based on the internal and external credit rating systems. Major providers of external credit migration or transition matrices are Moody's, Standard & Poor's, and Fitch's. These transition matrices are usually published for one, two, or five years and used as inputs to measure potential credit losses incurred or to determine the required level of capitals.

The increasing reliance on rating systems as risk measurement tools emphasized the importance of adequate estimation of credit migration matrices which are major inputs. The downgrade momentum effects in credit risks can critically affect banks revenues and strategic positions. The fluctuations in obligors' creditworthiness over time make that the interests received by banks might be considerably less than current obligors' risk levels. In addition, the fees collected to provide protections when trading credit derivatives may significantly lower due to underestimation the exposures. Regulations about the kind of instruments that some banks and professional investors are allowed to trade means that the effects of borrowers downgraded below certain ratings lead to the termination of some financial contracts.

The outline of the paper is organized as follow. In section 2, we present the literature review on the modeling of credit migrations risks. Section 3 then provides the objective and contribution of the current research. Section 4 gives the description of the data and the sources used in the study. Section 5 describes in detail the extended state models. Section 6 presents the implementation of the algorithms for modeling of rating transition with downgrade momentum using MatLab. Section 7 provides the results of the three different models used to simulate the migration risks caused by the downgrade momentum. Section 8 contains the results of the sensitivity analysis of the simulated random distribution of the initial portfolio. Section 9 gives the implementation of the extended transition matrices in measuring Incremental Risk Charge. Section 10 concludes the study. Appendix A contains the results of the transition matrices for the three different models. Appendix B contains detail description of the algorithms used for the different models.

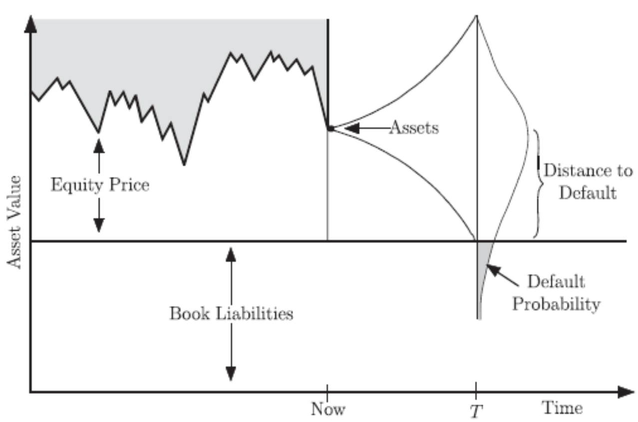
2. Literature review

Literature on the area of credit migration risks are based on the structural and reduced-form models. Earlier classical structural models are represented by circumstances under which a default occurs as soon as the asset value of a firm falls below the face value of its liabilities. Examples of Black and Scholes (1973) and Merton (1974) describe the asset value of the firm as a stochastic process (see Figure 1) under which

$$dV(t) = (\mu - \gamma)\, V(t)dt + \sigma V(t)dW(t)$$

where μ is the drift rate on the assets, γ is proportional cash payout rate, σ is the asset volatility, and $W(t)$ is a standard wiener process.

Figure 1. *The Black-Scholes-Merton structural model of default*



Source: Duffie and Schaefer (2003) in Credit Risk: Pricing, Measurement and Management

Unfortunately, the problems of using structural models come from the determination of the value and volatility of the firm's assets, and to model the stochastic process driving the value of the firm adequately. In addition, the fact that both the drift rate and volatility of the firm's assets can depend on the future economical environments is not considered in their evaluations.

The reduced-form models represent modern approach of credit migration that captures not only the event to default but also ratings change. The first reduced form approach was introduced by Fons (1994) and later extended by several authors, including Jarrow et al (1997), Duffie and Singleton (1999). The discrete-time approach represents the most popular form which usually assumes a time-homogeneous Markov model. The Markov property means that ratings migrations are independent of the past ratings (stable, upgrade, and downgrade). The time-homogeneity means that the transition intensities given in the annual transition matrix are independent of time period considered. These two assumptions clearly affect the estimation of credit migration matrices. Furthermore, the time-homogeneity assumption neglects the cyclical effects of the economy (i.e. during periods of downturn of the economic, the likelihoods of further downgrades exceed those during periods of growth). Nickell et al., (2000), Bangia et al., (2002), Krüger et al., (2005) pointed out that transition matrices are not constant through time and different during periods of expansion and recession of the economy. These are due to the contagion effects in the economy where failure of an important organization threaten the stability and survival of others very sound organizations and might lead them to default as well. During periods of growth (with relative excess of funds) companies can obtain cheap credits from banks to help them fund their business activities and survive. In addition, Trück and Rachec (2005) show substantial effects on risk figures for credit portfolios when credit migration matrices take into account the cyclical effects. Lando and Skødeberg (2002) show that assuming that transition matrices follow a Markov property neglects past ratings movements which are important determinant of the likelihood of a downgrade versus that of an upgrade over a certain period of time.

Discrete-time approach is based on the cohort method which estimate the migration matrices with the probability of migrating from the rating category i at the beginning of the year to the rating category j as

$$\hat{P}_{ij} \equiv \frac{N_{ij}}{N_i}, j \neq i$$

where N_i is the number of firms in the rating class i at the beginning of the year, N_{ij} is the number of firms that started with rating class i and have migrated to the rating class j within the considered time period.

These two assumptions have some useful consequences because annual transition matrices to the n^{th} power generate *n*-year transition matrices. For example, Kreinin and Sidelnikova (2001) show that computing the six-month transition matrix can be performed by taking the square root of annual transition matrix. Unfortunately, despites the attractiveness of such feature in mathematical sense, the result might have little sense in risk assessment. Data quality of ratings transition observed within shorter periods are either not available or too scarce to make a reliable estimate of the migration matrices because obligors usually take more than a year to default and keys financial data are provided at year end in annual reports. The mathematical limitations of fractional power occur when trying to estimate short-term transition matrices for less than a year because there is no guarantee that the transition matrices obtained are stochastic matrices. Transition matrices are stochastic matrices when they satisfied the two conditions: square matrices with nonnegative entries and row sums equal to 1. Typically, fractional powers generate matrices with negative entries despites having row sums still equal to 1, thus the resulting matrices cannot be considered stochastic matrices. In addition to these numerical problems, there might be more than one root to the transition matrices and it is not exactly clear which roots represent the right choice, see Higham and Lin (2009).

To adjust some of the shortcomings of the discrete-time methods, it is possible to transform the discrete-time approach of the credit migration matrices into continuous-time approach through the use generators matrices. The research done by Lando and Skødeberg (2002) shows that continuous-time approach are capable to better capture rare events in rating transition with more realistic non-zero estimates for probabilities of rare events. It gives the advantage that the transition matrices are time-dependent with positive probability values for each rows summing into one. Credit migration matrices *P(t)* for arbitrary time horizons *t* for the continuous-time approach can be defined as

$$P(t) \equiv \exp(\Lambda t), t \geq 0$$

where Λ is the generator of a given annual transition matrix with discrete-time Markov property

However, Israel et al. (2000) show that several discrete transition matrices cannot be transformed into continuous time method because there is no generator at all or a generator that has negative off-diagonal elements exist leading to negative transition probabilities for short time horizon. In most cases, approximation methods are required for generator matrices with negative off-diagonal elements. Israel et al. (2000) show that it is possible more than one generator exists and the transition matrix of the valid or unique generator if it exists must satisfy three conditions: determinant must be greater than 0.5, distinct positive eigenvalues, and distinct real eigenvalues. A preferred approach may be to directly estimate the generator from the data, then to estimate the transition matrices within the required time horizon. An important issue in continuous time approach is that data (e.g. accounting report, external and internal ratings, etc.) are usually reported once a year.

Lando and Skødeberg (2002) showed the non-Markov evolution (for instance, dependence on previous rating) of the transition matrices while still assuming the time homogeneity. Lando et al (2002) demonstrate the advantages of using continuous-time approach to capture the probabilities of rare events (e.g. default of a firm with AAA) that may not have yet occurred but could occur in the future compare to discrete-time approach which assume the probability values of zero. Christensen, et al (2004) introduce a continuous-time hidden Markov chain model to capture the non-Markov effect in which downgraded firms have an increasing probability of experiencing further downgrades.

The present study could not fully summarize all the researches that have been conducted in the field of credit risks. Instead, it has attempted to summarize the most relevant literature that has emerged in the area of credit migration risks and particularly different existing models that capture the non-Markov evolution of transition data.

3. Aims and contribution of this research

In this study we attempt to accomplish four objectives. At first, we aim to provide insights about basic applications of credit migration matrices and its relevance in measuring credit migration risks. Second, to develop algorithms in MatLab that will serve to evaluate the effects of the downgrade momentum in a multi-period setting e.g. downgrade momentum increases the probability of further downgrade by a factor of about three as suggested by Lando and Skødeberg (2002). Third, to analyze the differences between the results of the respective models that will serve the influence of the number of portfolios in each rating. At last, to determine the impact of adjusting transition matrices to capture downgrade momentum in the measurement of Value-at-Risk for long-term portfolios. This adjusts some of the discrepancies observed in the Basel II framework due to lack of dynamics in those models to suggest the rating transition matrices will improve the measure of capital charges.

Specifically, the purpose of the study is to answer the following questions:

- What is the impact of capturing downgrade momentum in the estimate of credit migration matrices for measurement of short and long term portfolios?

- Do the rating transitions matrices improve the ability of the value-at-risk to adequately measure the riskiness of a portfolio?

The research made by Lando and Skødeberg (2002) is of particular relevance to this study. It provides insight into the modeling of credit migration matrices by showing the importance of prior rating in determining the transition intensities of downgrades versus those of stables and upgrades over a given time horizon. There is an apparent downward momentum effect in rating migrations data. This study implements some of the results obtained where it illustrates the significance of downgrade momentum in transition matrices. Their results can be summarize as follow: considering ratings migrations from current states to neighboring states, there is strong downgrade momentum showing that after reaching a rating through a downgrade, an obligor probability of experiencing further downgrades is increased in general by a factor of about three. Also, there is virtually no detectable upgrade momentum showing that after reaching a rating through upgrade, an obligor probability of experiencing further upgrade is unchanged in general. It is assumed that an obligor has reach a rating for a longtime become stable and does not have a momentum to move to a different rating class.

To implement the result, we need to know about the previous ratings of the obligors that we separate into three different states (stable rating, rating upgrade, and rating downgrade). This is a special case of the approach made by Christensen et al., (2004) where they introduced new ratings called "*Excited states*" besides the ratings provided by Moody's which are called "*Normal states*". The excited states refer to obligors that are currently downgraded in the extended state model. After a random amount of time within the excited state with the same rating, the obligor returns to the normal state. However, the disadvantage of that method is that it is not possible to observe the transition of an obligor from the excited state to the corresponding non-excited state due to the random amount of time. Our method avoids the problem of unobserved transition from excited states to the corresponding non-excited state but also gives comprehensive view of all the potential scenarios that can occurs (stable, upgrade, downgrade).

A simple one-to-one extension of the annual *S&P's* transition matrix referred as benchmark represents a standard Markov time-homogeneous transition matrix insensitive to downgrade momentum. The first model of the extended transition matrix triples the transition intensities for recently downgraded obligors that will be further downgraded in the next period resulting in a different from the original *S&P's* annual transition matrix when aggregated back. The calibration of the first model second model involved decrease in the corresponding transition intensities for stable and upgraded states that will be downgraded in the next period such that the aggregation of the extended transition matrix gives the original S&P's transition matrix. In fact, this last construction is dependent on the initial number of obligors in each rating category.

The current study attempts to contribute to the existing literature in three possible ways. First, we implement the statistical research performed by Lando and Skødeberg (2002) in determining the significance of downward momentum that is observed in credit migration matrices. Second, to develop extended transition matrices that capture this downward momentum based on observable ratings representing a special case of the findings of Christensen et al (2004) and the research of Güttler and Raupach (2007). Third, implement the results of the extended transition matrices in the measurement of Incremental Risk Charge, Basel Committees on Banking Supervision (2009).

4. Description of the rating transitions

We will consider in this study the rating transition matrices provided by the Standard and Poor's agency that are classified based on the obligor or counterparty credit quality into one of several discrete credit rating categories (see Table 1).

Table 1. *Classification of Standard and Poor's long-term ratings obligation*

	Ratings	Meanings
Investment grades	AAA	Highest quality, minimal credit risk
	AA	High quality
	A	Strong payment capacity
	BBB	Adequate protection, moderate risk
Speculative grades	BB	Likely to pay, but ongoing uncertainty
	B	High risk but limited margin of safety remains
	CCC	High default risk, depend on sustained favorable economic conditions
	D	In default

The annual transition matrix of the Standard &Poor's covers the period of April 1996. It is estimated based on the cohort approach (see Table 2 below). In the simulations, the rating scales is replaced with an equivalent numerical scales giving for the total of 8 rating categories equal to:

$$P_{ij} = \{AAA, AA, A, BBB, BB, B, CCC, D\} \leftrightarrow \{1, 2, 3, \ldots, 8\} \text{ with } i, j = 1,2,\ldots,8.$$

Table 2. *Probability of migrating from one rating quality to another within 1 year*

Initial rating	Rating at year-end (transition probabilities in %)							
	AAA	**AA**	**A**	**BBB**	**BB**	**B**	**CCC**	**D**
AAA	90.81	8.33	0.68	0.06	0.12	0	0	0
AA	0.70	90.65	7.79	0.64	0.06	0.14	0.02	0
A	0.09	2.27	91.05	5.52	0.74	0.26	0.01	0.06
BBB	0.02	0.33	5.95	86.93	5.30	1.17	1.12	0.18
BB	0.03	0.14	0.67	7.73	80.53	8.84	1.00	1.06
B	0	0.11	0.24	0.43	6.48	83.46	4.07	5.20
CCC	0.2200	0	0.22	1.30	2.38	11.24	64.86	19.79
D	0	0	0	0	0	0	0	100

Source: Standard & Poor's CreditWeek (15th April 1996).

The transition matrix describes all possible migrations for obligors given their rating scales. For example, the notation $P(AA \rightarrow BBB) = 0.06$ represents the transition probability for obligors with AA rating which will migrate to BBB rating in the next year. The rows of the transition matrix correspond to the current ratings for the obligors and columns represent future ratings. The last row with rating D represents the absorbing default state, i.e. the probability for obligors to leave the default state once it is reached equals to zero. The AAA rating represents the highest rating in the first row of the transition matrix. All the elements below the diagonal represent the probabilities for upgrades while the elements above the diagonal represent the probabilities for downgrades. The elements of the last column with D rating represent the default probabilities for the different ratings. Finally, the diagonal elements represent the probabilities for the ratings to be preserved in the next period.

In this study, we faced the problem of lack of data about the history of the ratings migration for the different obligors, for example within the total number of obligors at the beginning of a given year with AA rating; some obligors have reached the AA rating through downgrade from the AAA rating, some obligors have been stable at the AA rating, and finally some obligors have been upgraded to AA rating from one of the following ratings {A, BBB, BB, B, CCC}. It is important to note that such information may be readily available to rating agencies. Some banks may have obtained these data through their internally dealings with clients but it also depends on the management investment strategy.

5. Extended state models

To capture the importance of past dependencies in the transition matrix with increasing probability of further ratings downgrade for obligors that were previously downgraded, we split the ratings into three states {s, u, d}, which respectively correspond for ratings that have been stable, reached through recent upgrade last year, and reached through recent downgrade last year. The normal transition matrix is thus extended into the following:

$$P'_{ij} = \{AAA^s, AAA^u, AA^s, AA^u, AA^d, A^s, A^u, A^d, BBB^s, BBB^u, BBB^d, BB^s, BB^u, BB^d, B^s, B^u, B^d, CCC^s, CCC^d, D\}$$

$$\leftrightarrow \{1, 2, 3, 4, 5, 6, 7, \ldots, 20\} \quad \text{with } i, j = 1,2,\ldots,20.$$

In addition, the rating categories AAA, CCC and D are considered as special rating categories. Therefore, no ratings can directly reach AAA rating through a downgrade because it is the highest possible rating. Also, no ratings can directly reach CCC rating through an upgrade because it is the lowest rating. The last rating D, which is an absorbing state can only be reach through a downgrade and once it is reached, it is not anymore possible to leave that state "absorbing state", thus we do not discriminate between D^s and D^d and therefore simply keep it notations as D. We are focusing the analysis on a sequence of single notch downgrades instead of one multi-notch downgrade as it is the preference of Standard & Poor's.

5.1 Number of firms in the extended states

The determination of the number of firms in the extended states stem from the need to aggregate the transition intensities of the extended transition matrix back to its original form as provided by *S&P's* annual transition matrix. Let's consider the portfolio distribution n_k with $k = 1,2,\ldots,r$ representing the number obligors within the states (AAA, AA, …, CCC, D). The expected number of obligors with rating j within the stable, upgraded, and downgraded states are denotes respectively n_j^s , n_j^u , and n_j^d are assumed to follow a discrete Markov property and computes as follow:

$$n_j^d = \sum_{i=1}^{j-1} n_i * P_{ij} \qquad n_j^u = \sum_{i=j+1}^{r} n_i * P_{ij}$$

$$n_j^s = n_j - n_j^d - n_j^u$$

where P_{ij} represents the *1*-year transition matrix from rating i to j.

The interpretation of these computations is as follow. Having split the ratings into three different states (stable, upgrade, and downgrade), the number of obligors within each rating category are similarly split based on previous ratings. Given that the annual transition matrix P_{ij} provides information about firms' future rating changes, we assumed that the portfolio distribution in the previous section represents an expected average that remains constant through time. To be constant through time, we also assume that banks will repurchase loans with stable ratings to replace those that have defaulted within each rating category.

As a leading example, we consider an expected total portfolio of 295 obligors over the years. The portfolio are subdivided as follow into each of the rating categories: 20 obligors with AAA rating, 45 obligors with AA rating, 45 obligors with A rating, 45 obligors with BBB rating, 45 obligors with BB rating, 45 obligors with B rating, 20 obligors with CCC rating, and no obligor has defaulted. Table 3 below summarizes the result obtained for the different rating categories.

Table 3. *Distribution of the expected number of obligors within each states of the extended transition matrix*

Stable state	**AAA^s**	**AA^s**	**A^s**	**BBB^s**	**BB^s**	**B^s**	**CCC^s**
	19.5780	42.0515	38.2275	38.2840	38.8390	38.0675	17.2010
Upgrade state	AAA^u	AA^u	A^u	BBB^u	BB^u	B^u	-
	0.4220	1.2825	3.1310	3.9320	3.3920	2.2480	-
Downgrade state	-	AA^d	A^d	BBB^d	BB^d	B^d	CCC^d
	-	1.6660	3.6415	2.7840	2.7690	4.6845	2.7990

Let P(i,j) denotes the transition probability from *i* to *j* rating over a period of one year. The computations for the number of obligors with A rating are performed as follows. Let denotes n_A be the number of firms with A rating, n_A^s the number of firms that have held a stable A rating last year, n_A^u the number of firms that have recently been upgraded to A rating last year, and n_A^d the number of firms that have recently been downgraded to A rating last year. $n_A^u = n_{BBB}*P(BBB,A) + n_{BB}*P(BB,A) + n_B*P(B,A) + n_{CCC}*P(CCC,A) = 3.6415$; $n_A^d = n_{AAA}*P(AAA,A) + n_{AA}*P(AA,A) = 3.1310$; $n_A^s = n_A - n_A^u - n_A^d = 38.2275$.

5.2 Transition matrices in the extended state

Let consider a simplified transition matrix denote as "*P*" with four rating categories {*A, B, C, D*} to provide a detail explanation of the procedures to create extended transition matrices P_0, P_1, and P_2. The rating categories A, C, and D are considered special ratings which respectively represent the highest rating, the lowest rating, and the absorbing default state.

P	A	B	C	D
A	0.80	0.10	0.08	0.02
B	0.05	0.85	0.07	0.03
C	0.02	0.13	0.70	0.15
D	0	0	0	1

A positive consequence of this splitting method is that besides at the starting period, it is possible to observe the entrance of an obligor into another rating category for the different splits (stable, upgrade, and downgrade). The decision to split a rating category is based on the nature of this rating. Any rating besides special ratings can be split into three sub-ratings (stable, upgrade, and downgrade).

We defined three possible methods of extension of the normal transition matrix *P* into the extended transition matrices P_0, P_1, and P_2. The extended transition matrix P_0 represents a simple extension of the annual transition matrix that is insensitive to the downgrade momentum effects and will serve as benchmark for the measurement of downgrade momentum. The extended transition matrices P_1 and P_2 are both sensitive to the downgrade momentum effects due to triply of the transition intensities for the rating categories B^d and C^d that will migrate to B^d, C^d, and D in the next period. The stable state of the extended matrices P_0, P_1, and P_2 are readjusted such that the sum of each rows are 1, thus satisfying the condition of stochastic matrices. The structure of the allowed transitions is defined as follow:

- The "**stable state**" defines firms that have kept the same rating categories. Firms within the stable state can either keep the same rating in the next period ($A^s \rightarrow A^s$, $B^s \rightarrow B^s$, $C^s \rightarrow C^s$), or be upgrade in the next period if it does not currently hold the highest rating to ($B^s \rightarrow A^u$, $C^s \rightarrow A^u$, $C^s \rightarrow B^u$), or be downgrade in the next period to ($A^s \rightarrow B^d$, $A^s \rightarrow C^d$, $B^s \rightarrow C^d$), or simply default ($A^s \rightarrow D$, $B^s \rightarrow D$, $C^s \rightarrow D$).

- The "**upgrade state**" defines firms that have recently been upgraded into higher rating categories. Firms within the upgrade state can either keep the same rating categories in the next period ($A^u \rightarrow A^s$, $B^u \rightarrow B^s$), or be upgrade in the next period ($A^u \rightarrow A^u$, $B^u \rightarrow B^u$), or be downgrade ($A^u \rightarrow B^d$, $A^u \rightarrow C^d$, $B^u \rightarrow C^d$), or default ($A^u \rightarrow D$, $B^u \rightarrow D$).

- The "**downgrade state**" defines firms that have recently been downgraded into lower rating categories. Firms within the downgrade state can either keep the same rating categories in the next period ($B^d \rightarrow A^s$, $C^d \rightarrow B^s$), or be upgrade in the next period ($B^d \rightarrow A^u$, $C^d \rightarrow A^u$, $C^d \rightarrow B^u$), or be downgrade ($B^d \rightarrow C^d$), or default ($B^d \rightarrow D$, $C^d \rightarrow D$).

All possible direct rating migrations are reported for the extended transition matrix in Table 11 in Appendix A for rating categories AAA, AA… D on the form (stable, upgrade, and downgrade).

The extended transition matrix P_0 below represents a one-to-one split of the rating categories A, B, C, and D in the transition matrix *P* into rating categories (see Table 12 in Appendix A for the extended transition matrix of the benchmark with rating categories AAA, AA, A, BBB, BB, B, CCC, D).

P_0	A^s	B^s	C^s	A^u	B^u	B^d	C^d	D
A^s	0.80	0	0	0	0	0.10	0.08	0.02
B^s	0	0.85	0	0.05	0	0	0.07	0.03
C^s	0	0	0.70	0.02	0.13	0	0	0.15
A^u	0.80	0	0	0	0	0.10	0.08	0.02
B^u	0	0.85	0	0.05	0	0	0.07	0.03
B^d	0	0.85	0	0.05	0	0	0.07	0.03
C^d	0	0	0.70	0.02	0.13	0	0	0.15
D	0	0	0	0	0	0	0	1

Similarly, we derive the extended transition matrix P_1 below. It is constructed such that P_1 has a one-to-one correspondence or split with the rating categories AAA, AA… D. However, it is sensitive to the downgrade momentum effects due to increase by three[1] times of the transition intensities for recently downgraded firms which will experience further downgrade in the next period. Further, we also assumed that the increase in transition intensities come with decrease in transition intensities for recently downgrade firms that will keep the same rating in the next period. At the same time, the transition intensities for firms that will be upgraded in the next period remain unchanged (see Table 13, Appendix A for the extended transition matrix of Model 1 with rating categories AAA, AA… D).

[1] Result of the research performed by Lando and Skødeberg (2002)

P_1	A^s	B^s	C^s	A^u	B^u	B^d	C^d	D
A^s	0.80	0	0	0	0	0.10	0.08	0.02
B^s	0	0.85	0	0.05	0	0	0.07	0.03
C^s	0	0	0.70	0.02	0.13	0	0	0.15
A^u	0.80	0	0	0	0	0.10	0.08	0.02
B^u	0	0.85	0	0.05	0	0	0.07	0.03
B^d	0	**0.65**	0	0.05	0	0	**0.21**	**0.09**
C^d	0	0	**0.40**	0.02	0.13	0	0	**0.45**
D	0	0	0	0	0	0	0	1

The extended transition matrix P_2 represents a calibrated version of the extended transition matrix P_1 such that its aggregation back into a non-extended transition matrix form is identical to the initial transition matrix P. We also assumed that the transition intensities for firms that will be upgraded in the next period are not affected and remain unchanged as with the extended transition matrix P_0. The increase in transition intensities due to tripling of transition intensities for recently downgrade firms that will be downgraded in the next period are matched with the decrease in transition intensities for recently downgrade firms that will be stable in the next period such that each row sums of the extended transition matrix are 1.

Furthermore, transition intensities both for recently stable and upgraded firms that will be downgrade in the next period have to be decrease based on the proportion of firms recently downgrade to recalibrate the matrix such its aggregation back into non-extended form gives exactly the same transition matrix P. To account for the decrease in transition intensities for recently stable and upgraded that will be downgraded in the next period, the transition intensities for recently stable and upgraded firms that will be stable in the next period have to be increase such that each row sums of the extended transition matrix equal to 1 (see Table 14, Appendix A for the extended transition matrix with rating categories AAA, AA… D).

P_2	A^s	B^s	C^s	A^u	B^u	B^d	C^d	D
A^s	0.8000	0	0	0	0	0.1000	0.0800	0.02
B^s	0	**0.8621**	0	0.0500	0	0	**0.0615**[2]	**0.0264**
C^s	0	0	**0.7762**	0.0200	0.1300	0	0	**0.0738**
A^u	0.8000	0	0	0	0	0.1000	0.0800	0.0200
B^u	0	**0.8621**	0	0.0500	0	0	**0.0615**	**0.0264**
B^d	0	**0.6500**	0	0.0500	0	0	**0.2100**[3]	**0.0900**
C^d	0	0	**0.4000**	0.0200	0.1300	0	0	**0.4500**
D	0	0	0	0	0	0	0	1.0000

To determine the proportion of firms within each rating classes of the extended transition matrix, we provide an example of the computation for the simplified case with 4 rating categories A, B, C, D. Assuming that banks have a portfolio consisting of 4 firms with highest A rating, 7 firms with B rating, 4 firms with C rating, and no firm has defaulted. Similar to Section 5.1, we compute the number of firms within the stable, upgraded, and downgraded state as follows:

$n_{A^u} = n_c*P(C,A)+ n_B*P(B,A) = 0.43$; $n_{A^s} = n_A- n_{A^u} = 3.57$; $n_{B^d} = n_A*P(A,B) = 0.4$; $n_{B^u} = n_c*P(C,B) = 0.52$; $n_{B^s} = n_B- n_{B^d} -n_{B^u} = 6.08$; $n_{C^d} = n_A*P(A,C)+ n_B*P(B,C) = 0.81$; $n_{C^s} = n_C- n_{C^d} = 3.19$. Thus,the proportion of firms within the rating classes A^s, A^u, B^s, B^u, B^d, C^s, C^u, C^d, D are respectively 0.8925, 0.1075, 0.86858, 0.07428, 0.05714, 0.7975, 0.2025, 0.

To sum up, the purpose of the extended transition matrices is to capture the downgrade momentum effects. In addition, to compute the number of firms in each split of ratings the annual transition matrix provides in average a rough estimate for the different rating classes. As we reintroduce the number of firms that have default as new firms that have been purchased with stable rating which means that the portfolio represents a long-term average.

[2] Let ρ_B denotes the proportion of firms with B rating that are recently downgraded such that $\rho_B = 0.05714$
$P(B^s \rightarrow C^d) = P(B^u \rightarrow C^d) = [P(B \rightarrow C)* (1- 3\rho_B)]/(1- \rho_B)$ and

[3] $P(B^d \rightarrow C^d) = 3*P(B \rightarrow C)$

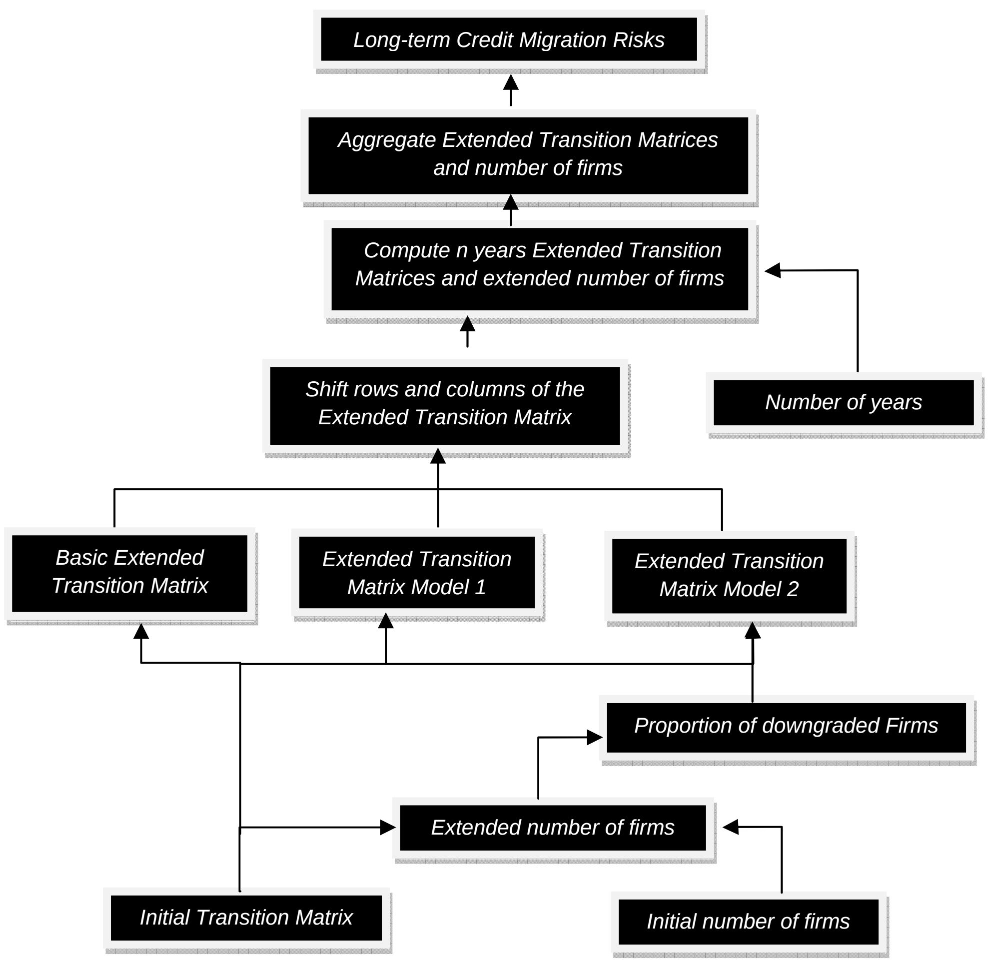
6. Implementation of the algorithm for the rating transitions

In this section, we describe the implementation of the algorithm that will use past dependencies in the transition matrices to capture the downgrade momentum using Matlab[4] coding.

6.1 Long-term credit migration risks

Figure 2 below summarizes with the activity diagram of the different processes involved in the programming.

Figure 2. *Activity diagram for the construction of the long-term horizon credit migration risks*



The initial transition matrix, distribution of firms, and number of years constitutes the basic inputs that the user need. Furthermore, the user need to decide about the output she/he want to obtain which

[4] Matlab® software

refer here as version. Selecting version '0' refers to the routine "*MatrixExt0*" which represent a simple one-to-one extension of the initial annual transition matrix that is insensitive to the downgrade momentum. It serves to illustrate effects of the downgrade momentum and is referred in the report as benchmark (see Syntax 1 in Appendix B). Similarly, selecting version '1' refers to the routine "*MatrixExt1*" which has one-to-one extension or correspondence with the initial annual transition matrix but captures the downgrade momentum by tripling the transition intensities that will be being downgraded in the next period given that they were previously downgraded (see Syntax 2 in Appendix B). The differences between these two distinct extensions provide simple but clear consequences about the effects of downgrade momentum in past ratings. Selecting version '2' refers to the routine "*MatrixExt2*" which represents a more subtle calibration than "version 1" because after extension, aggregating the transition intensities of the extended transition matrix gives the identical transition matrix inputted by the user. The extended matrices are sensitive to the downgrade momentum and based on the proportion of obligors that reached the given rating (see Syntax 3 in Appendix B). The construction of the routine "*MatrxExt2*" necessitates the presence of two sub-routines. The subroutine "*MUExt*" (see Syntax 4 in Appendix B) splits the number of firms within each rating category based on classes (stable, upgraded, downgraded) and the sub-routine "*MUFract*" that generate the proportion of downgraded obligors for each rating categories besides the default (see Syntax 5 in Appendix B).

Extending the initial annual transition matrix segregate states in the extended transition matrix into stable, upgrade, and downgrade classes (e.g. $\{AAA^s, AA^s, A^s, BBB^s, BB^s, B^s, CCC^s\}$; $\{AAA^u, AA^u, A^u, BBB^u, BB^u, B^u\}$; $\{AA^d, A^d, BBB^d, BB^d, B^d, CCC^d\}$; $\{D\}$). The routine "*MatrixBlock*" (see Syntax 6 in Appendix B) shift the rows and columns such that the extended transition matrix is ranked into a single grouped extended transition matrix with credit quality from the highest to the lowest (e.g. $AAA^s, AAA^u, AA^s, AA^u, AA^d, \ldots, CCC^s, CCC^d, D$).

Having obtained the initial extended transition matrix and number of firms within the different rating classes, the sub-routine "*MatrixPowerExt*" compute the power of the extended transition matrix to generate next year extended transition matrix. These extended transition matrices are multiplied with the number of firms of the same year to obtain the number of firms in the next year (see Syntax 7 in Appendix B).

To be able to interpret the results which are in the extended states, we need to aggregate the extended transition matrices and number of firms back into rating categories. The routine "*MatrixShrink*" (see Syntax 8 in Appendix B) serves to resize the extended transition matrices back into transition matrices (e.g. *AAA, AA, A, BBB, BB, B, CCC, D*) by aggregating the transition intensities based on the number of firms in extended state. The routine "*MUShrink*" (See Syntax 9 in Appendix B) aggregate the number of firms from the extended state to rating categories.

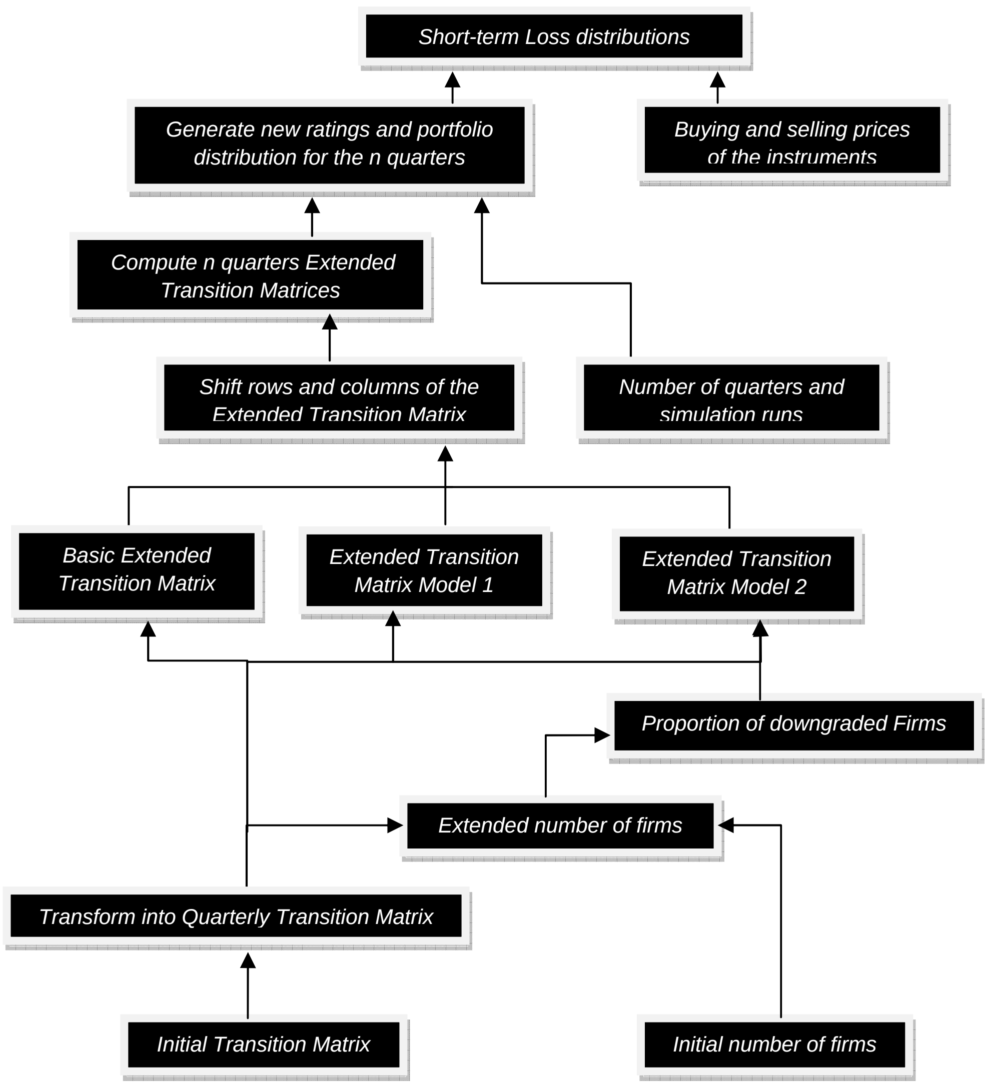
The user has the opportunities to observe all these results above by selecting version '3' and be able to see the effects of the downgrade momentum through comparison of the model 1 and model 2 with the benchmark. Thus, the routine "*MatrixComp*" (see Syntax 10 in Appendix B) contains all the above routines but also an additional to determine the effects of downgrade momentum. The routine for the graphical representation figs and figrs (Syntax 11 and Syntax 12 in Appendix B) give the bars code respectively for the portfolio and the downgrade momentum effects of the portfolio.

The algorithm computes n successive transition matrices and number of firms that are either sensitive or insensitive to the downgrade momentum. The main program named "*MatrixTrans*" (see Syntax 13 in Appendix B) contains all the routines and sub-routines that are selected based on users' decision about the required version. The main program returns the initial extended transition matrix, the aggregated transition matrices and number of firms at the first and n^{th} year, and the bar plot of the distributions of firms which may in some case capture the effects of downgrade momentum.

6.2 Short-term credit migration risks

The Figure 3 below summarizes the activity diagram involving the programming for the computation of the loss distributions.

Figure 3. *Activity diagram for the construction of the short-term horizon credit migration risks*



Similarly like for the long-term migration risks, the extensions of the initial transition matrices are the same. However, before the extension of the initial transition matrix, the fractional power is performed giving the quarterly transition matrix which is referred to the routine "*PowerFract*" (see Syntax 17 in Appendix B). After obtaining the extended portfolio with the routine "*MUExt*", the proportion of downgraded obligors with the subroutine "*MUFract*", the different extended transition matrices with the routines "*MatrxExt0, MatrxExt1, MatrxExt2*", and ranked the segregated extended transition matrices into single grouped with the routine "*MatrixBlock*" similar to the procedure for the long-term horizon. However, the extended transition matrices over the subsequent four quarters remain the same as initially due to the rebalancing. This is performed by the routine "*UnMatrixPowerExt*" (see Syntax 18 in Appendix B). The routine "*UnNewRating*" (see Syntax 19 in Appendix B) determine the evolution of the current portfolio over the different quarters based on obligors' specific risks and common risk factor. The routine "*UnPricesExt*" (see Syntax 20 in Appendix B) simply extend the prices from the rating categories to the extended states. The routine "*UnexpMix*" (see Syntax 21 in Appendix B) compute the loss distributions based on the strategy that only stable and recently upgraded obligors can be rebalanced while recently downgraded obligors cannot be sold unless a significant loss that the banks do not want to incur. The routine "*UnFig*" (see Syntax 22 in Appendix B) gives the graphical representation of the loss distribution. The routine "*UnFigMoment*" (see Syntax 23 in Appendix B) gives the histograms of the effects of downgrade momentum on the loss distributions of the portfolio. Finally, the main routine "*UnexpLoss*" (see Syntax 24 in Appendix B) gives an overview of the different graphical representations of loss distributions.

7. Effects of the downgrade momentum on the transition matrices

In this section, we analyze the effects of the downgrade momentum on the transition matrices in risk management. Due to its construction, the 1-year transition matrix recalculated from the aggregation of the extended transition matrix (similar to extended transition matrix P_0 in Section 5 based on the simplified transition matrix with four ratings A, B, C, and D) derived through simple one-to-one correspondence with the transition matrix provided by Standard & Poor's (see Table 2) are identical with the original S&P's transition matrix. The recalculation of the 1-year transition matrix serves the sole purpose to verify the correctness of the two different models. The 5-year transition matrix is derived by computing the 1-year extended transition matrix to the power of five (see Table 4 below). The 1-year and 4-year transition matrices obtained through this simple one-to-one correspondence with the annual transition matrix provided by Standard & Poor's will serve as 1-year and 4-year transition matrices benchmark or reference in the measurement of the effects of downgrade momentum.

Table 4. *4-year transition matrix insensitive to momentum labeled benchmark using S&P ratings*

	AAA	AA	A	BBB	BB	B	CCC	D
AAA	62.2020	28.5097	7.2989	1.1932	0.5139	0.2044	0.0382	0.0396
AA	2.4412	63.0187	27.2193	5.1526	1.0329	0.7710	0.1681	0.1963
A	0.4442	8.0414	66.3810	17.8493	4.1443	1.9013	0.5134	0.7252
BBB	0.1511	2.0824	19.1755	51.8075	14.1764	6.7093	2.5840	3.3138
BB	0.1284	0.8243	5.1225	19.9074	39.6634	21.9913	3.5852	8.7778
B	0.0697	0.4688	1.5273	4.5261	15.9052	46.1225	6.9652	24.4151
CCC	0.4438	0.2724	1.2532	3.8271	7.2781	19.3415	13.4007	54.1832
D	0	0	0	0	0	0	0	100.0000

The matrix represents the aggregated transition intensities of stable, upgrade, and downgrade states. The 4-year transition matrix is calculated following the discrete time Markov process by taking the fifth power of the initial extended transition matrix insensitive to downgrade momentum and aggregated based on the portfolio distribution.

7.1 Downgrade-sensitive distance measures for transition matrices

We use the methodology developed by Trueck and Rachev (2008) to measure the distances between two different transition matrices. The 1-year transition matrix (see Table 5 below) represents the result of the extended transition matrix of the first model that captures the downgrade momentum. Compared to the 1-year transition matrix of the benchmark, it can be noticed that the transition intensities have shifted from the diagonal elements of the transition matrix to the upper side of the matrix corresponding to the downgrade.

Table 5. *1-year transition sensitive to momentum labeled model 1*

	AAA	AA	A	BBB	BB	B	CCC	D
AAA	90.8100	8.3300	0.6800	0.0600	0.1200	0	0	0
AA	0.7000	90.0095	8.3668	0.6874	0.0644	0.1504	0.0215	0
A	0.0900	2.2700	89.9834	6.4134	0.8598	0.3021	0.0116	0.0697
BBB	0.0200	0.3300	5.9500	84.9686	5.9558	1.3148	1.2586	0.2023
BB	0.0300	0.1400	0.6700	7.7300	79.1886	9.9279	1.1231	1.1905
B	0	0.1100	0.2400	0.4300	6.4800	81.5400	4.9174	6.2826
CCC	0.2200	0	0.2200	1.3000	2.3800	11.2400	59.3108	25.3292
D	0	0	0	0	0	0	0	100.0000

The matrix represents the aggregated transition intensities of stable, upgrade, and downgrade states. The 1-year transition matrix is calculated following the discrete time Markov process by taking the power two of the initial extended transition matrix sensitive to downgrade momentum and aggregated based on the portfolio distribution.

These shifts of transition intensities are significant in the measurement of Value-at-Risk because more firms will be downgrade to lower ratings or simply default compare to the transition matrix of the benchmark. The increases in the migration distances and distances to default from the first model compare to the benchmark indicates higher deterioration of the ratings credit quality.

An important consequence of the shifts for the transition intensities away from the diagonal elements to the downgrade side of the transition matrix will have very severe effects when measuring for long term horizons because despites the fact that migration probabilities far away from the diagonal are small, their rates of increase will be much higher as $t \rightarrow \infty$, leading to more firms ratings migration from investment rating grades to become speculative rating grades.

Table 6. *4-year transition matrix with sensitive to momentum labeled model 1*

	AAA	AA	A	BBB	BB	B	CCC	D
AAA	62.1577	24.3147	9.6675	2.3083	0.7999	0.4527	0.1052	0.1939
AA	2.4288	62.5757	24.5112	6.5462	1.7531	1.2280	0.3008	0.6561
A	0.4429	7.9815	65.6643	15.8183	4.6829	2.8124	0.7121	1.8855
BBB	0.1474	2.0575	18.9035	50.5896	11.8764	7.5617	2.0881	6.7757
BB	0.1268	0.8044	5.0151	19.3992	38.0221	18.8937	2.9323	14.8063
B	0.0580	0.4574	1.4797	4.3642	15.3498	43.8895	4.2107	30.1908
CCC	0.3502	0.2210	1.0485	3.0897	5.8020	14.9628	8.8413	65.6845
D	0	0	0	0	0	0	0	100.0000

The matrix represents the aggregated transition intensities of stable, upgrade, and downgrade states. The 4-year transition matrix is calculated following the discrete time Markov process by taking the fourth power of the initial extended transition matrix sensitive to downgrade momentum and aggregated based on the portfolio distribution.

As expected, the 5-year transition matrix of the first method (see Table 6) is sensitive to downgrade momentum with far higher distances to default and migration distances at the upper side compare to 5-year transition matrix of the benchmark (see Table 4). It interesting to note that correspondingly, the transition intensities at the upgrade side of the matrix are lower than those in the transition matrix of the benchmark. It is important to note that these transition intensities are spread into the upper side of the matrix diagonal particularly for speculative rating grades. This has the severe consequence that it will be more likely for migration changes for ratings that are far in the left of the diagonal than near and also will fewer ratings upgrade compare to the transition matrix of the benchmark.

The 1-year transition matrix derived in the first model is intuitively not very appealing because it should be expected that regrouping the different states (stable, upgrade, and downgrade) of a rating in the 1-year extended transition matrix should provide an identical standard transition matrix as provided by Standard & Poor’s while still capturing the downgrade momentum. A simplified example of an appealing calibration is given for the extended transition matrix P_2 in Section 5. Using this calibration process, the 1-year transition matrix derived in the second model is identical to the 1-year transition matrix of the benchmark (see Table 2). However, the differences observed in the 1-year extended transition matrix leads to significant subtle effects in the long-term.

Table 7. *4-year transition matrix with sensitive to momentum labeled model 2*

	AAA	AA	A	BBB	BB	B	CCC	D
AAA	62.1587	24.5162	9.5896	2.2274	0.7814	0.4401	0.1004	0.1862
AA	2.4596	64.7790	23.3090	5.9213	1.5655	1.1176	0.2667	0.5812
A	0.4505	8.2641	69.3311	13.6002	3.9148	2.3372	0.5786	1.5236
BBB	0.1473	2.1172	19.7454	52.9123	10.7441	6.7072	1.8728	5.7537
BB	0.1275	0.8274	5.2228	20.3853	40.6747	17.5049	2.6018	12.6555
B	0.0522	0.4810	1.5470	4.5452	16.5335	48.8329	3.7102	24.2981
CCC	0.4033	0.2416	1.1840	3.5618	6.7871	18.3610	14.0302	55.4309
D	0	0	0	0	0	0	0	100.0000

The matrix represents the aggregated transition intensities of stable, upgrade, and downgrade states. The 4-year transition matrix is calculated following the discrete time Markov process by taking the fourth power of the initial extended transition matrix sensitive to downgrade momentum and aggregated based on the portfolio distribution.

The 5-year transition matrix in the second model (see Table 7) has some very interesting characteristics compare to the 5-year transition matrix of the benchmark. The distances to default are greater than in the benchmark besides for B rating where the difference is considerably low. The higher distances to default obtained in the second model represent the effects of the downgrade momentum since in the long-term more firms will default that usually predicted by the transition matrix provided by Standard & Poor's. The distances to default are significant for ratings AAA, AA, A, BBB, and BB which are about 1.5 times more than for the distances to default of the benchmark. We can conclude that distances to default are underestimated when using the transition matrix of the benchmark in particular for investment rating grades such as AAA and AA. Speculative grades rating such as B and CCC are well captured in 5-year transition matrix of the benchmark where the downgrade momentum does not have much effect. Migration distances at the lower side of the matrix which correspond to upgrades are relatively higher compared to the benchmark. The migration distances in ratings downgrades are less than in the benchmark for near migrations and higher than in the benchmark for far migrations.

Distance measures

To evaluate the effects of downgrade momentum in model 1 and 2, we need to apply more formal evaluation techniques. Based on the literature, we will select some techniques that are appropriate for in evaluating two migration matrices. Stefan Trück (2004) suggests four interesting evaluation techniques to evaluate two transition matrices.

$$D_1(P,Q) = \sum_{i=1}^{n}\sum_{j=1}^{n} \mathrm{sign}(i-j).|i-j|.(p_{ij}-q_{ij})$$

$$D_2(P,Q) = \sum_{i=1}^{n}\sum_{j=1}^{n} \mathrm{sign}(i-j).\frac{|i-j|.1}{p_{ij}}.(p_{ij}-q_{ij})$$

$$D_3(P,Q) = \sum_{i=1}^{n}\sum_{j=1}^{n} \mathrm{sign}(i-j).|i-j|.sign(p_{ij}-q_{ij}).(p_{ij}-q_{ij})^2$$

$$D_4(P,Q) = \sum_{i=1}^{n}\sum_{j=1}^{n} \mathrm{sign}(i-j).\frac{|i-j|.1}{p_{ij}}.sign(p_{ij}-q_{ij}).(p_{ij}-q_{ij})^2$$

It can be observed from these techniques that far migrations receive a higher weight than near migrations. Greater penalties are imposed on shift of the transition intensities in the downgraded side of the diagonal. The normalized difference imposes greater penalties on changes in small transition intensities corresponding to the downgrade and upgrade intensities of the migration matrices. Default probabilities are very small in particular for investment grade ratings but incorrect estimations of these probabilities have significant impact on the Value-at-Risk. These techniques can be useful in measuring the effects of downgrade momentum (only for cells where $q_{ij} \neq 0$). The results of the different techniques are presented below as follows:

		D_1	D_2	D_3	D_4
Model 1	T = 1	-13.7754	-5.6178	-36.7120	-2.3129
	T = 4	-82.8703	-38.6212	-417.7882	-34.7046
Model 2	T = 4	-19.9112	-28.9205	-184.8164	-26.5333

7.2 Measurement of portfolio exposures

To measure the portfolio risks based on the different transition matrices (benchmark, first model, and second model), we assumed that each of the 265 firms has an exposure of €1 million. Let's consider the state vector $n_k(t)$ with $k = 1, 2\ldots, 8$ as row vector containing the portfolio distribution apportioned by the ratings (AAA, AA, …, CCC, D) at time t. The number of elements n_k corresponding to the different possible credit ratings are arranged in order from the highest to lowest rating with D as the default state. The expected portfolio distribution $n_k(t+1)$ is assumed to follow a discrete Markov process such that $n_k(t+1) = n_k(t)*P$ where P represents the extended transition matrix for the time t. The results of the portfolio distribution in the first year and in the fourth year are presented in Table below.

Table 8. *Portfolio distributions for the different methods at a given period.*

		AAA	AA	A	BBB	BB	B	CCC	D
Benchmark	*t = 1*	18.5840	43.7410	47.7450	45.3845	42.3995	44.4940	15.7690	6.8830
	T = 4	10.0840	34.6302	59.7815	44.8624	30.0559	31.8160	6.6951	47.0750
Model 1	*t = 1*	18.5840	43.4528	47.5246	45.3752	42.1469	44.2038	15.1616	8.5511
	T = 4	9.9952	32.9145	57.0456	42.6385	27.7753	29.3135	4.6885	60.6290
Model 2	*T = 4*	10.0715	34.9120	60.5764	43.4386	29.0140	31.4260	4.8269	50.7346

The number of obligors at time t+1 are calculated by multiplying the expected number of obligors with the transition matrix at time t.

Given that we are either at the first or fourth year, our expositions to the obligors depending on default are as follows: we would expect in the first year losses of € 6.88 million and € 47.08 million in the fourth year corresponding to the transition matrix of the benchmark that is insensitive to downgrade momentum. In the first model, expected losses of € 8.55 million in the first year and € 60.63 million in the fourth year corresponding to the value-at-risk values in the first model. The differences of € 1.67 million in the first year and € 13.55 million in the fourth year represent the effects of the downgrade momentum that is captured in the first model. The expected loss of € 50.73 million in the fifth year corresponds to the value-at-risk in the second model and the difference of €

3.65 million represents the effects of the downgrade momentum that is captured in the second model.

The graphical representations below summarizing the differences between the portfolio distributions apportioned by the ratings AAA, AA…, D for the two different models are as follow:

Figure 4. *Differences in the number of obligors after a year between the model 1 and benchmark*

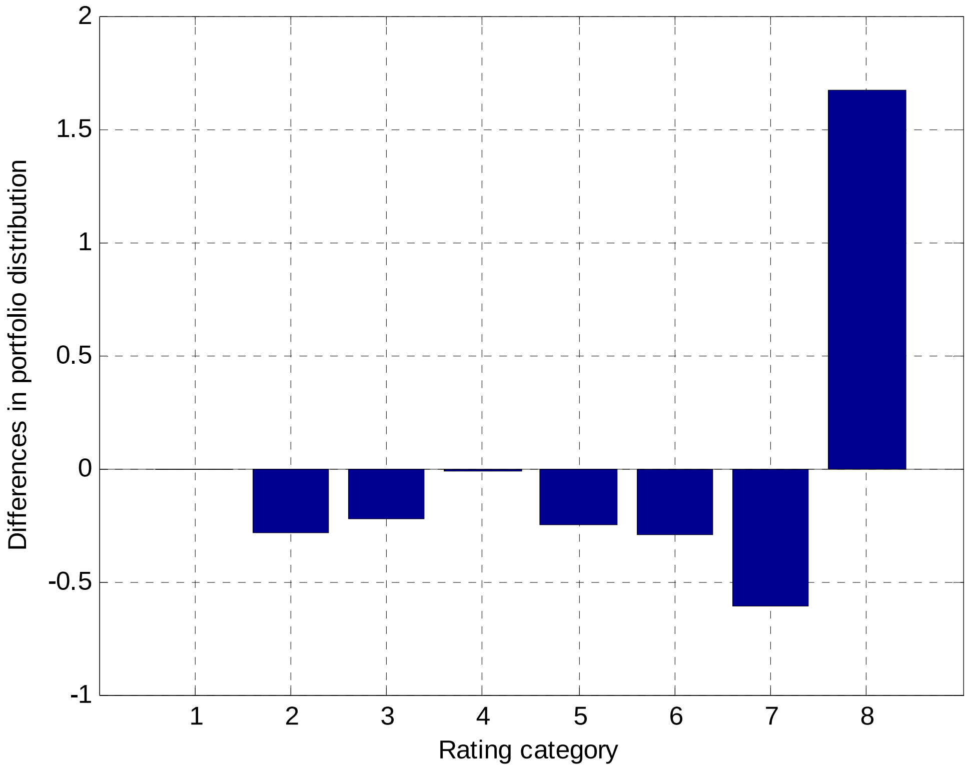


Figure 4 depicts the differences in the number of obligors within each rating category after a year. These differences are measured by comparing the number of obligors obtained in the extended transition matrix of the first model with those of the benchmark. The higher number of obligors that go to default after a year in the first model compare to the benchmark is balanced with the higher number of obligors that migrate to BB, B, and CCC ratings.

Figure 5. *Differences in the number of obligors at the fourth year between model 1 and benchmark*

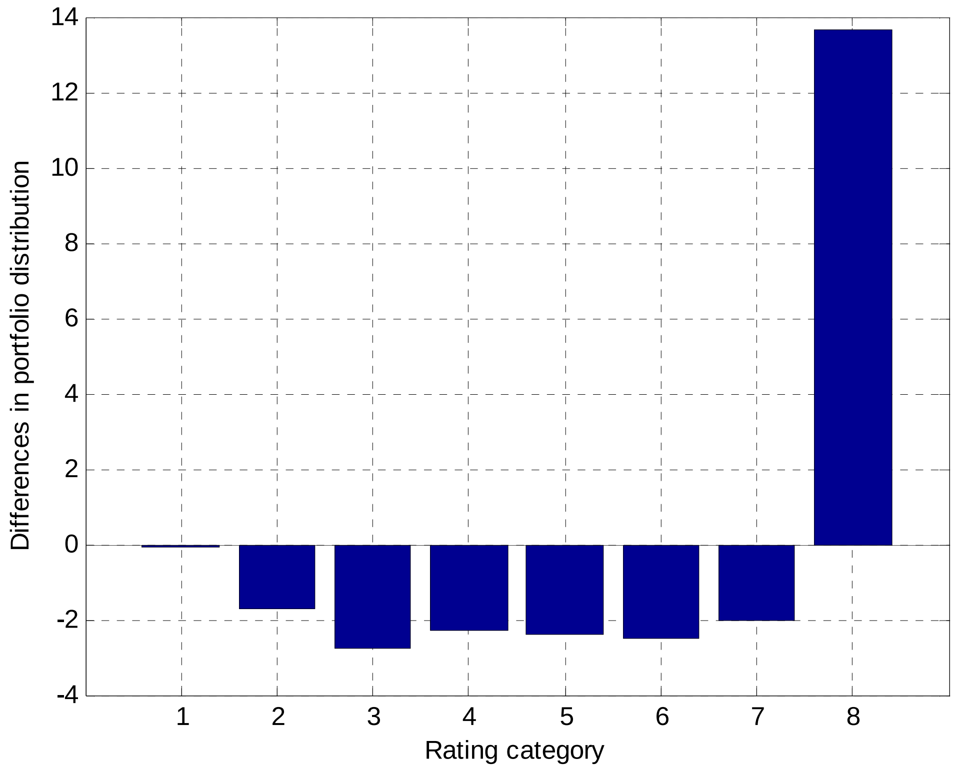


Figure 5 describes the difference in the number of obligors within each rating category in the fifth year between the first model and benchmark. The number of obligors that goes to default in the fifth year is far higher in first model than with the transition matrix of the benchmark. The increases in the number of defaults in the first model are comparatively represented by more migrations to non-default states in the benchmark representing the extreme case.

Figure 6. *Differences in the number of obligors at the fourth year between model 2 and benchmark*

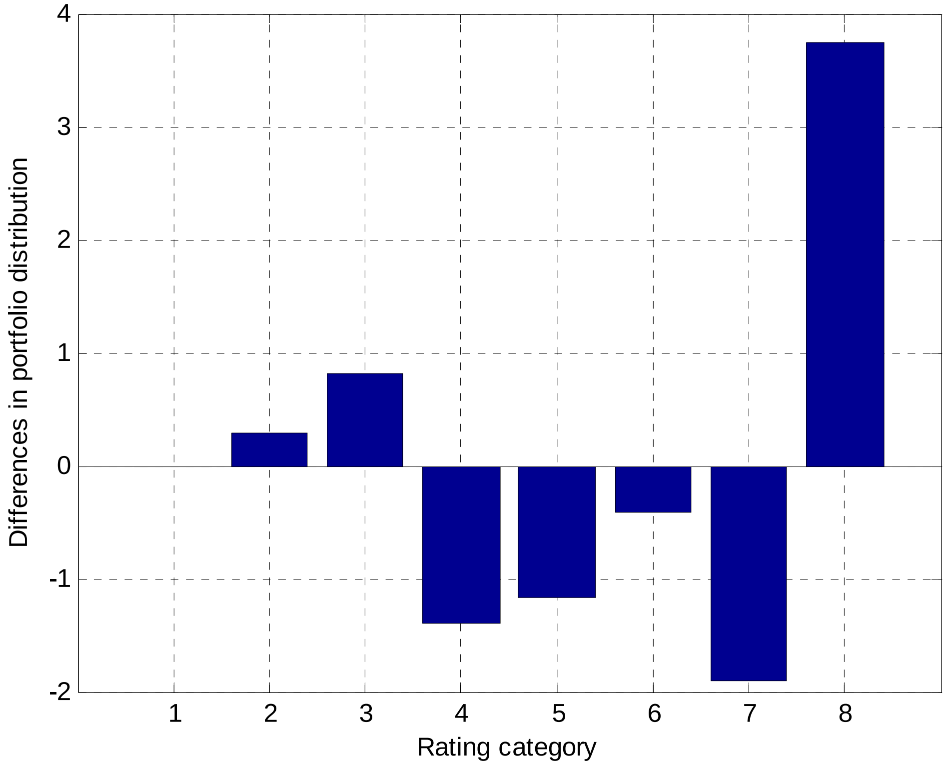


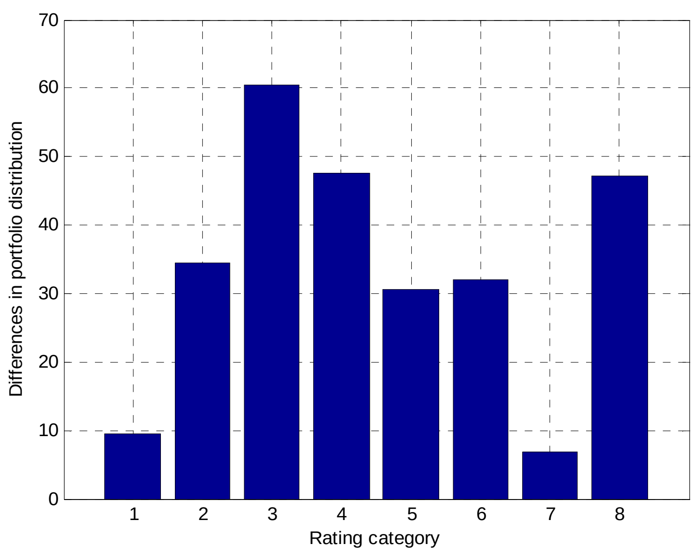
Figure 6 provides the differences in the number of obligors within each rating category in the fifth year between the second model and the benchmark. The number of obligors that go to default in the fifth year in the second model is more than for the benchmark. The effects of the downgrade momentum in the second model can be seen with higher migrations of number of obligors to non-default states in AA and A ratings where obligors with BBB, BB, B, CCC ratings are lower than in the benchmark because more firms with low ratings are downgraded to the default state.

8. Sensitivity to change in the portfolio distribution

In this section, we analyze the influence of the distributions of the number of obligors within each rating category. The variations of the number of obligors within each state of the extended transition matrix will serve to determine the robustness or stability of the results found. We randomly vary the initial number of obligors within each rating class of the extended transition matrix based on normal distribution with mean zero and standard deviation 1.

We first aim to determine the effects of computing the portfolio distribution based on extended transition matrix of the benchmark and then aggregated back into its original form compare to the computation of the portfolio distribution based on S&P's transition matrix (see Figure 7 below). As observed, there are significant differences in the number of obligors obtained between the two computation methods. The computation based on S&P's transition matrix yields higher number of obligors within each rating categories than with the extended transition matrix. This helps us to deduce that when comparing the portfolio distribution between two different extended models, the importance should be based on the proportion of changes rather than on the absolute values of change because of their relatively very low values.

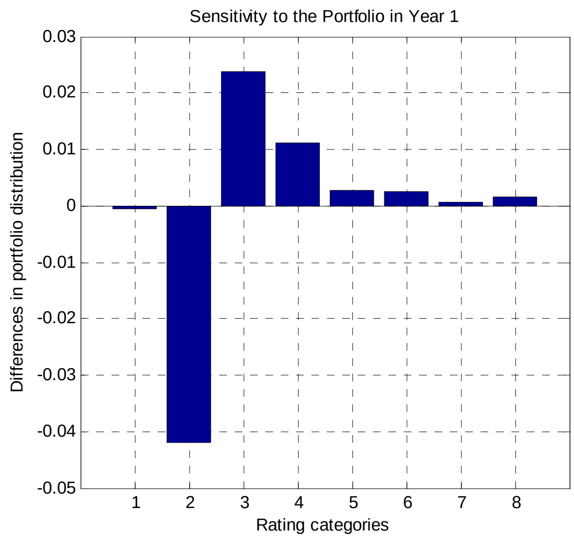
Figure 7. *Variability observes in the number of obligors in each rating category at the fourth year*

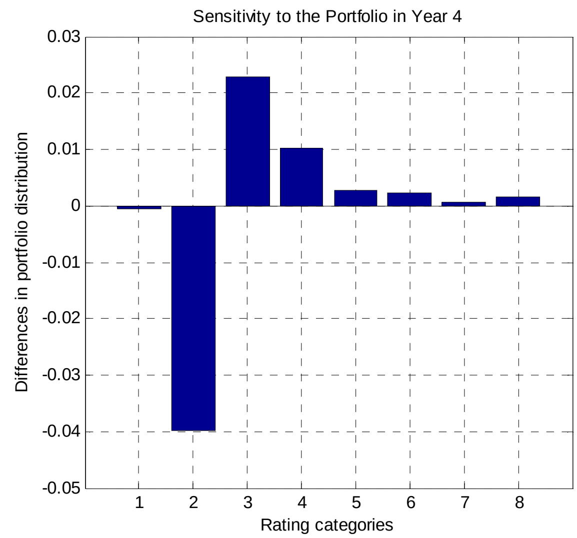


The extended transition matrix of the benchmark insensitive to the downgrade momentum significantly reduces the number of obligors compare to the initial transition matrix. It can be observed that there are far less than 60 obligors in rating A for the extended matrix compare to transition matrix in the fourth year.

Figure 8 summarizes the variations between the average results obtained with random change in the portfolio distribution for the two downgrade momentum sensitive models compare to the benchmark that is insensitive to downgrade momentum. The low values obtained due to the variation in obligors by comparing the number of obligors in the first model and second model to the benchmark after a year and in the fourth year confirm the considerable low impacts of the fluctuation in the distribution of the initial number of obligors in the results. Thus, we can conclude that the variability of the distribution of the initial number of obligors does not have significant effects on the downgrade momentum effects through the extended transition matrices.

Figure 8. *Sensitivities of the extended transition matrices to change in portfolio distribution*





At time t = 1 and t = 4, the variation of initial number of obligors seems to have effects on the AA, A, and BBB ratings.

The numerical values of the differences illustrating the downgrade momentum effects are presented in details in the Table 9 below.

Table 9. Differences in the number of obligors between simulated downgrade sensitive and insensitive

	Model 1-0		Model 2-0	
	Year 1	Year 4	Year 1	Year 4
AAA	0	-0.0004	0	-0.0004
AA	-0.0144	-0.0420	-0.0144	-0.0399
A	0.0121	0.0237	0.0121	0.0229
BBB	0.0017	0.0112	0.0017	0.0103
BB	0.0000	0.0029	0.0000	0.0027
B	0.0005	0.0025	0.0005	0.0024
CCC	0.0001	0.0007	0.0001	0.0006
D	0.0000	0.0015	0.0000	0.0015

9. Measurement of the Incremental Risk

In this section, we attempt to implement some of the ideas developed by the Basel Committee in the assessment of the incremental capital in the trading book for credit instruments. Credit risk-related products represent innovative financial products that can be held by banks to receive interest payments or it can be sold in the market to other financial institutions to fund other business activities. We intend to determine the effects of downgrade momentum in migration risks for a portfolio with short-term horizons through the application of extended transition matrices sensitive or insensitive to the downgrade momentum. To illustrate the effects of downgrade momentum, let us consider the same portfolio and rating distribution as for the long-term horizon. However, the time interval is 3 months or a quarter. We further make the assumption about the selling and buying prices of the different debt instruments depend on the ratings. For each instrument, the selling and buying prices are represented as follows: 100 and 105 euro for rating AAA; 97 and 110 euro for rating AA; 88 and 115 euro for rating A; 84 and 120 euro for rating BBB; 70 and 125 euro for rating BB; 65 and 140 euro for rating B, and 55 and 150 euro for rating CCC.

Expected losses

In the different scenarios developed, the portfolio is rebalanced each quarter. The simple rebalancing is the scenarios in which all the different instruments in the extended state (stable, upgrade, downgrade) are rebalanced to the initial period. The mix rebalancing is the scenario in which only the instruments in the stable and upgrade states are rebalanced. The mix rebalancing is more realistic because it will be very difficult to sell an instrument that has previously been downgraded without incurring significantly losses. We assumed that selling a recently downgraded portfolio is quite difficult without sustaining severe losses and sometime even impossible such as during market turmoil. As expected, the expected losses in the first model are higher in both scenarios compared to those of the benchmark and second model. It can be observed that the expected loss in the mix rebalancing where impose the condition that recently downgraded obligors cannot be sold almost double the total cost compare to the simple rebalancing (see Table 10 below). Moreover, the expected cost for the benchmark and second model are identical in both scenarios because the transition matrices considered only depend on the time interval of a quarter.

Table 10. *Expected losses for the two different scenarios*

	Benchmark	Model 1	Model 2
Simple rebalancing	114.01	123.21	114.01
Mix rebalancing	278.63	315.84	278.63

Value-at-Risk

In the spirit one of the most popular credit risk framework "*CreditMetrics*", we used the threshold models which are an extension of the Merton's (1974) firm-value model to compute the value-at-risk of the portfolio. In Merton's model, a firm *i* is assumed to default if its assets values X_i fall below the value of its liabilities rendering the firm unable to meet its financial obligations at the end of the time period. A firm's asset value X_i is modeled as a one factor model $X_i = Y\sqrt{\rho_i} + Z_i\sqrt{(1-\rho_i)}$ where Y refers to the common factor such as the economical situations, ρ_i refers to the assets correlation, and Z_i refers to the firm specific risk. The random variables Y and Z_i are i.i.d and standard normal.

The threshold models aims to randomly generate a profit and loss distribution (P\L using the Monte Carlo simulation. Just like in Merton's model, we randomly generate the common factor Y and for each firm a specific factor Z_i that are standard normal inverses. The standardized thresholds computed from the standard normal inverses of the cumulative transition intensities of the extended transition matrices (see Table 12, Table 13, and Table 14 in the Appendix B) serve to determine the migrations of the firm's rating to future ratings at the end of different periods or quarters based on the randomly generated firm's values X_i. The upper and lower limits thresholds respectively T_{ij}^{H} and T_{ij}^{L} of the different states are computed as follow:

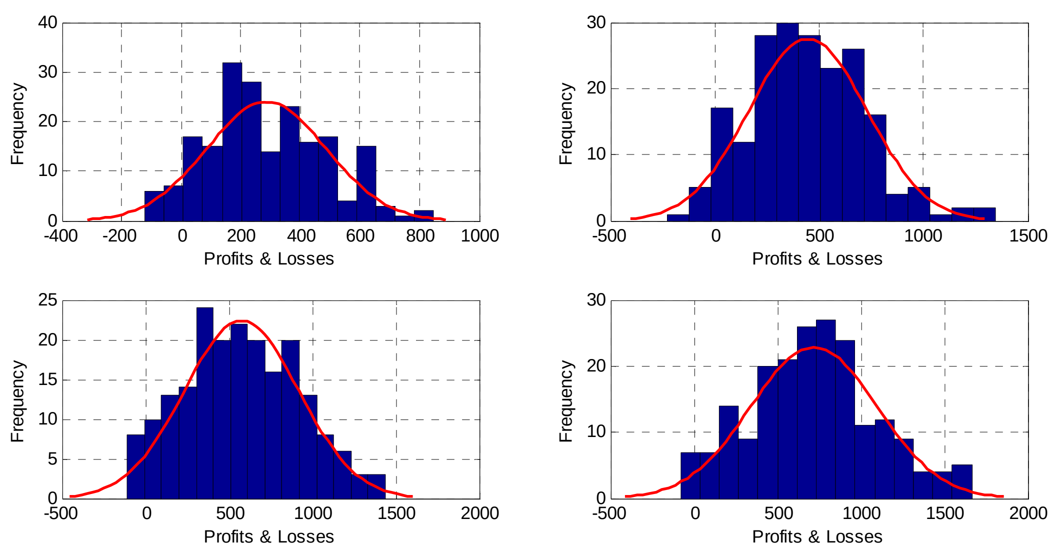
$$T_{ij}^{H} = \phi^{-1}\left(\sum_{i=1}^{j} P_{ij}\right) \quad \text{and} \quad T_{ij}^{L} = \phi^{-1}\left(\sum_{i=1}^{j-1} P_{ij}\right)$$

where *P(i,j)* denotes an extended transition matrix. With $\phi^{-1}(1) = +\infty$ and $\phi^{-1}(0) = -\infty$.

Interpretations of the evolution of firm's value to future credit rating compare to the thresholds are as follow: high generated firm's value compare to the upper limit of the threshold of rating lead to either downgrade or default. Low generated firm's value compare to the lower limit of the threshold of rating lead to upgrade. Less sufficient variations in firm's value compare to the thresholds of rating lead to stable rating. Thus, after the randomly generated firm's value X_i within a given state, the rating within the threshold satisfying X_i corresponds to the firm's new state at the next quarter. It should be note that we assumed that all individual obligors within a rating class k are correlated based on the Basel II condition: $\rho_k = 0.12\left(1 + e^{-50 \times PD_k}\right)$ where PD_k is the probability of default of rating k. Furthermore, we do not take into account the defaulted obligors in the computation of the loss distributions.

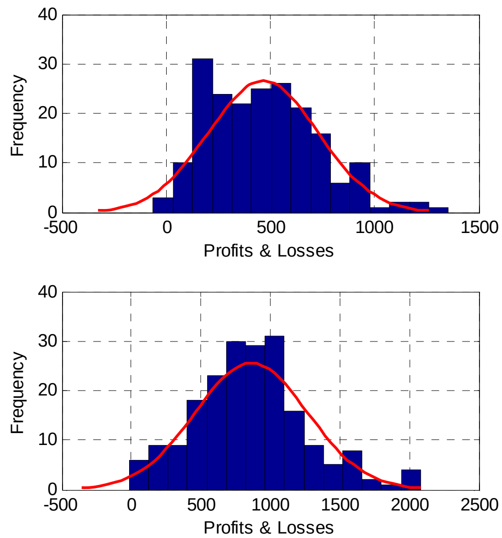
At the portfolio level, each simulation step consists of drawing all asset values according to firm-value of the Merton model above by assigning new ratings using the different thresholds. Computing individual changes in firm's expected values and aggregating them to a single random change of the portfolio's expected value is referred as portfolio's profit and loss (P/L). Repeating this procedure several times enable us to generate the P/L distribution.

Figure 9. *Simulated loss distributions for the benchmark in four successive quarters*



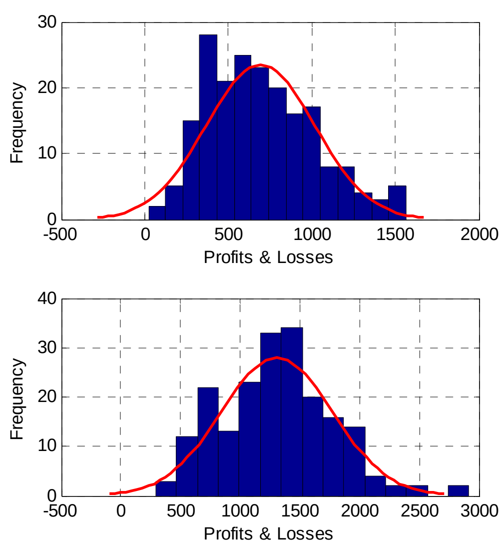
First quarter (up left), second quarter (up right), third quarter (down left), fourth quarter (down right), signs for Loss (+) and Profit (-).

Figure 10. *Simulated loss distributions for the first model in four successive quarters*



First quarter (up left), second quarter (up right), third quarter (down left), fourth quarter (down right); signs for Loss (+) and Profit (-).

Figure 11. *Simulated loss distributions for the second model in four successive quarters*



First quarter (up left), second quarter (up right), third quarter (down left), fourth quarter (down right); signs for Loss (+) and Profit (-)

The following examples in Figure 9, Figure 10, and Figure 11 above correspond to the graphical representation of the simulated loss distributions. Comparing the loss distributions for the successive four quarters in Figure 9, we can conclude that the distributions are very different with increasing cost over the quarter till the maximum loss of about 1500 euro. Similarly, Figure 10 for the first model with a maximum loss of about 2000 euro and Figure 11 for the second model with a maximum loss of about 3500 euro have also very different loss distribution over the four quarters. It is very interesting to note that the losses in the second model yield relatively the highest losses.

Figure 12 and Figure 13 show that not taking into account the downgrade momentum may lead to about 2000 Euro losses based on the first model and about 3000 Euro losses based on the second model in the fourth quarter. We can conclude that banks that do not take into account the downgrade momentum might underestimate their risk levels because the presence of the downgrade momentum in the transition matrices have a substantial impact on the associated credit risks of a portfolio.

Figure 12. *Effects of downgrade momentum in the losses distributions of the first model*

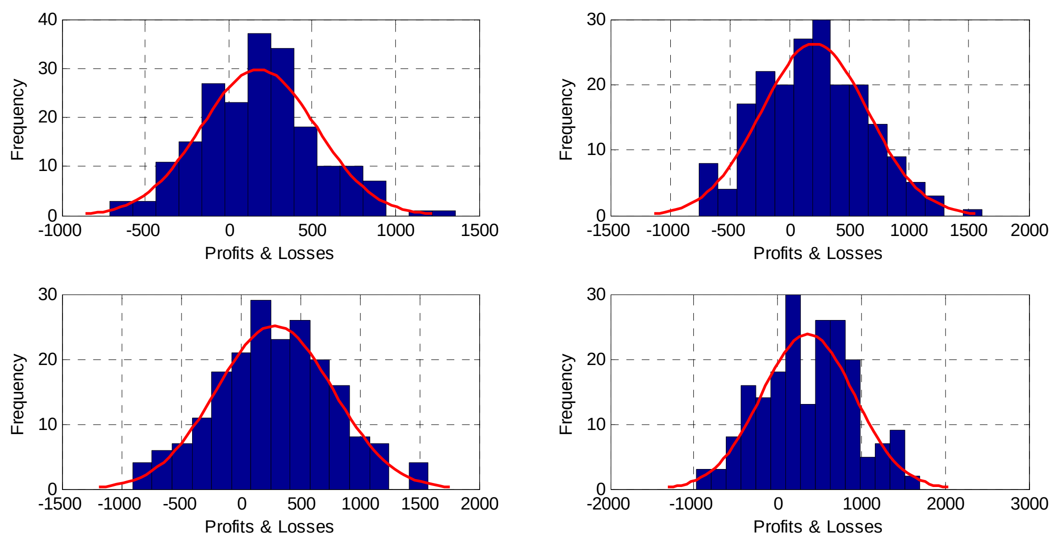
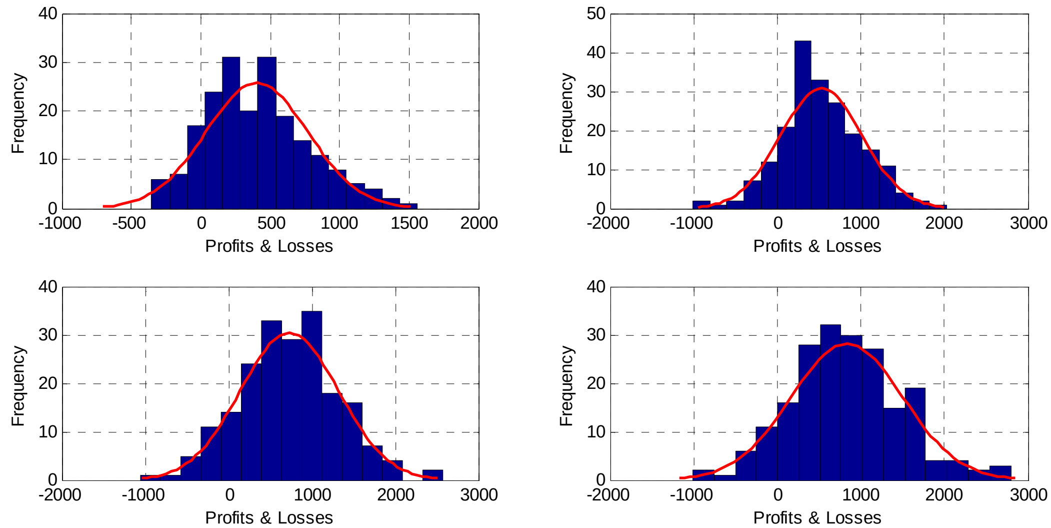


Figure 13. *Effects of downgrade momentum in the losses distributions of the second model*



Due to the limited time, we have kept the same portfolio in the syntax "*UnNewRating*" which may have increased the round off errors and performed 200 simulation runs. Therefore, the current results may be influenced by these factors.

10. Conclusion

In this study we have motivated the importance of credit migration matrices in risk management and in particular following the introduction of the Basel II framework. The Basel II represents an attractive tool that aim to adequately fit banking activities to its risk profile by requiring some capitals cushion.

The present study has attempted to implement the evidence of the effect of downgrade momentum in measuring credit migration risks. The approach presented intends to extend the standard Markov model of rating transition intensities. Assuming a given initial portfolio distribution, we determine an expected portfolio distribution for the extended transition matrix. Using the transition matrix of the Standard & Poor's for the period of 1996, we condition the transition intensities on previous ratings (stable, upgrade, and downgrade) and expected portfolio distribution in order to quantify the effects of the downgrade momentum on the credit migration risks. The transition matrix label benchmark represents a simple extension of the S&P's transition matrix with rating intensities insensitive to the downgrade momentum. The transition matrix label model 1 is an extension of the S&P transition matrix with rating intensities sensitive to the downgrade momentum while model 2 represents a refinement of model 1 that is calibrated to be aggregated as the initial S&P transition matrix.

In both the short and long terms, the results of the transition matrices show that the downgrade momentum affects the credit migration risks. In the long-term, we may incur an additional of 13.55 million euro more that predicted by S&P's transition matrices. Also, in the short-term, losses of about 3000 euro more could be incurred in addition to what is predicted by the transition matrices provided by Standard & Poor's. Investors who do not account for the downgrade momentum underestimate the migration risk compare to investors who do take. However, investors who account for the downgrade momentum based on the model 1 relatively overestimates the migration risk compare to those based on model 2.

11.References

Andersson, A., & Vanini, P. (2008). Credit Migration Risk Modeling. National Centre of Competence in Research Financial Valuation and Risk Management. Working Paper No. 539. SSRN: http://ssrn.com/abstract=1275202

Bangia, A., Diebold, F., Kronimus, A., Schagen, C., & Schuermann, T.(2002). Ratings Migration and the Business Cycle, with Application to Credit Portfolio Stress Testing. *Journal of Banking and Finance 26*, 445-474.

Black, F., & Scholes, M.(1973). The Pricing of Options and Corporate Liabilities. *Journal of Political Economy 81*, 637-654.

Basel Committees on Banking Supervision.(2009). Guidelines for Computing Capital for Incremental Risk in the Trading Book. www.bis.org /publ/bcbs141.htm.

Christensen, J.H.E., Hansen, E., & Lando, D.(2004). Confidence sets for continuous-time rating transition probabilities. *Journal of Banking & Finance*, 28, 2575–2602.

Duffie, D. & Singleton, K.J.(2003). Credit Risk: Pricing, Measurement, and Management. *Princeton University Press*.

Duffie, D. & Singleton, K.J.(1999). Modeling the Term Structures of Defaultable Bonds. *Review of Financial Studies 12*(4), 687-720.

Fons, J.S.(1994). Using Default Rates to Model the Term Structure of Credit Risk. *Financial Analysts Journal Sept-Oct, 25-33*.

Güttler, A., & Raupach, P.(2007).The Impact of Downward Rating Momentum on Credit Portfolio Risk. Working Paper Series. SSRN: http://ssrn.com/abstract=954894

Higham, N.J., & Lin, L.(2009). On p^{th} Roots of Stochastic Matrices. Manchester Institute for Mathematical Sciences, School of Mathematics.

Israel, R., Rosenthal, J., Wei, J.(2000). Finding Generators for Markov Chains via Empirical Transition Matrices, with Application to Credit Ratings. *Mathematical Finance 11*, 245-

265.

Jarrow, R., Lando, D., & Turnbull, D.(1997). A Markov Model for the Term Structure of Credit Spreads. *Review of Financial Studies, 10*(2): 481-523.

Kreinin, A., & Sidelnikova, M.(2001). Regularization Algorithms for Transition Matrices. *Algo Research Quarterly,* vol. 4, nos. 1/2, pp. 23-40.

Lando, D., & Skødeberg, T.M.(2002).Analyzing Ratings Transitions and Rating Drift with Continuous Observations. *Journal of Banking & Finance*, 26, 423–444.

Merton, R.(1974). On the Pricing of Corporate Debt: The Risk Structure of Interest Rates. *Journal of Finance 19*, 449-470.

Nickell, P., Perraudin, W., & Varotto, S.(2000). Stability of Rating Transitions. *Journal of Banking and Finance 1-2*, 203-227.

Trück, S.(2004). Measures for Comparing Transition Matrices from a Value-at-Risk Perspective. *University of Karlsruhe*.

Trück, S., & Rachec, T.(2005). Credit Portfolio Risk and PD Confidence Sets through the Business Cycle. Journal of Credit Risk 2.

Trueck, S., & Rachev, S.T.(2008). Rating Based Modeling of Credit Risk: Theory and Application of Migration Matrices.

Zingales, L.(2008). Causes and Effects of the Lehman Brothers Bankruptcy. *University of Chicago Graduate School of Business*.

12. Appendix A

Table 11. Feasible direct migrations

	AAA^s	AA^s	A^s	BBB^s	BB^s	B^s	CCC^s	AAA^u	AA^u	A^u	BBB^u	BB^u	B^u	AA^d	A^d	BBB^d	BB^d	B^d	CCC^d	D
AAA^s	++													++	++	++	++	++	++	++
AA^s		++						++							++	++	++	++	++	++
A^s			++					++	++							++	++	++	++	++
BBB^s				++				++	++	++							++	++	++	++
BB^s					++			++	++	++	++							++	++	++
B^s						++		++	++	++	++	++							++	++
CCC^s							++	++	++	++	++	++	++							++
AAA^u	++													++	++	++	++	++	++	++
AA^u		++						++							++	++	++	++	++	++
A^u			++					++	++							++	++	++	++	++
BBB^u				++				++	++	++							++	++	++	++
BB^u					++			++	++	++	++							++	++	++
B^u						++		++	++	++	++	++							++	++
AA^d		++						++							++	++	++	++	++	++
A^d			++					++	++							++	++	++	++	++
BBB^d				++				++	++	++							++	++	++	++
BB^d					++			++	++	++	++							++	++	++
B^d						++		++	++	++	++	++							++	++
CCC^d							++	++	++	++	++	++	++							++
D																				++

Stable, upgrade, and downgrade states are tagged respectively with “s”, “u”, and “d”. The double plus sign “++” indicates feasible direct migrations in the initial extended transition matrix. The transition intensities may be positive only in these fields.

Table 12. Momentum insensitive for the initial extended matrix of the benchmark

	AAA^s	AAA^u	AA^s	AA^u	AA^d	A^s	A^u	A^d	BBB^s	BBB^u	BBB^d	BB^s	BB^u	BB^d	B^s	B^u	B^d	CCC^s	CCC^d	D
AAA^s	90.81	0	0	0	8.33	0	0	0.68	0	0	0.06	0	0	0.12	0	0	0	0	0	0
AAA^u	90.81	0	0	0	8.33	0	0	0.68	0	0	0.06	0	0	0.12	0	0	0	0	0	0
AA^s	0	0.70	90.65	0	0	0	0	7.79	0	0	0.64	0	0	0.06	0	0	0.14	0	0.02	0
AA^u	0	0.70	90.65	0	0	0	0	7.79	0	0	0.64	0	0	0.06	0	0	0.14	0	0.02	0
AA^d	0	0.70	90.65	0	0	0	0	7.79	0	0	0.64	0	0	0.06	0	0	0.14	0	0.02	0
A^s	0	0.09	0	2.27	0	91.05	0	0	0	0	5.52	0	0	0.74	0	0	0.26	0	0.01	0.06
A^u	0	0.09	0	2.27	0	91.05	0	0	0	0	5.52	0	0	0.74	0	0	0.26	0	0.01	0.06
A^d	0	0.09	0	2.27	0	91.05	0	0	0	0	5.52	0	0	0.74	0	0	0.26	0	0.01	0.06
BBB^s	0	0.02	0	0.33	0	0	5.95	0	85.93	0	0	0	0	5.30	0	0	1.17	0	1.12	0.18
BBB^u	0	0.02	0	0.33	0	0	5.95	0	85.93	0	0	0	0	5.30	0	0	1.17	0	1.12	0.18
BBB^d	0	0.02	0	0.33	0	0	5.95	0	85.93	0	0	0	0	5.30	0	0	1.17	0	1.12	0.18
BB^s	0	0.03	0	0.14	0	0	0.67	0	0	7.73	0	80.53	0	0	0	0	8.84	0	1.00	1.06
BB^u	0	0.03	0	0.14	0	0	0.67	0	0	7.73	0	80.53	0	0	0	0	8.84	0	1.00	1.06
BB^d	0	0.03	0	0.14	0	0	0.67	0	0	7.73	0	80.53	0	0	0	0	8.84	0	1.00	1.06
B^s	0	0	0	0.11	0	0	0.24	0	0	0.43	0	0	6.48	0	83.47	0	0	0	4.07	5.20
B^u	0	0	0	0.11	0	0	0.24	0	0	0.43	0	0	6.48	0	83.47	0	0	0	4.07	5.20
B^d	0	0	0	0.11	0	0	0.24	0	0	0.43	0	0	6.48	0	83.47	0	0	0	4.07	5.20
CCC^s	0	0.22	0	0	0	0	0.22	0	0	1.30	0	0	2.38	0	0	11.24	0	64.85	0	19.79
CCC^d	0	0.22	0	0	0	0	0.22	0	0	1.30	0	0	2.38	0	0	11.24	0	64.85	0	19.79
D	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100.00

The transition matrix is calculated using S&P credit migration matrix with discrete time Markov process. Transition intensities of obligors with last rating change as stable, upgrade, and downgrade are respectively assigned to the rating classes marked with 's', 'u', and 'd'. The ratings AAA, CCC, and D are exceptions and are considered special ratings. The ratings AAA and CCC cannot be entered respectively through downgrade and upgrade. Also, we assumed the default state D to be absorbing.

Table 13. Momentum sensitive for the initial extended matrix of model 1 using S&P ratings

	AAA^s	AAA^u	AA^s	AA^u	AA^d	A^s	A^u	A^d	BBB^s	BBB^u	BBB^d	BB^s	BB^u	BB^d	B^s	B^u	B^d	CCC^s	CCC^d	D
AAA^s	90.810	0	0	0	8.330	0	0	0.68	0	0	0.060	0	0	0.120	0	0	0	0	0	0
AAA^u	90.810	0	0	0	8.330	0	0	0.68	0	0	0.060	0	0	0.120	0	0	0	0	0	0
AA^s	0	0.700	90.650	0	0	0	0	7.790	0	0	0.640	0	0	0.060	0	0	0.140	0	0.020	0
AA^u	0	0.700	90.650	0	0	0	0	7.790	0	0	0.640	0	0	0.060	0	0	0.140	0	0.020	0
AA^d	0	0.700	73.350	0	0	0	0	23.370	0	0	1.920	0	0	0.180	0	0	0.420	0	0.060	0
A^s	0	0.090	0	2.270	0	91.050	0	0	0	0	5.520	0	0	0.740	0	0	0.260	0	0.010	0.060
A^u	0	0.090	0	2.270	0	91.050	0	0	0	0	5.520	0	0	0.740	0	0	0.260	0	0.010	0.060
A^d	0	0.090	0	2.270	0	77.870	0	0	0	0	16.560	0	0	2.220	0	0	0.780	0	0.030	0.180
BBB^s	0	0.020	0	0.330	0	0	5.950	0	85.930	0	0	0	0	5.300	0	0	1.170	0	1.120	0.180
BBB^u	0	0.020	0	0.330	0	0	5.950	0	85.930	0	0	0	0	5.300	0	0	1.170	0	1.120	0.180
BBB^d	0	0.020	0	0.330	0	0	5.950	0	70.390	0	0	0	0	15.900	0	0	3.510	0	3.360	0.540
BB^s	0	0.030	0	0.140	0	0	0.670	0	0	7.730	0	80.530	0	0	0	0	8.840	0	1.000	1.060
BB^u	0	0.030	0	0.140	0	0	0.670	0	0	7.730	0	80.530	0	0	0	0	8.840	0	1.000	1.060
BB^d	0	0.030	0	0.140	0	0	0.670	0	0	7.730	0	58.730	0	0	0	0	26.520	0	3.000	3.180
B^s	0	0	0	0.110	0	0	0.240	0	0	0.430	0	0	6.480	0	83.470	0	0	0	4.070	5.200
B^u	0	0	0	0.110	0	0	0.240	0	0	0.430	0	0	6.480	0	83.470	0	0	0	4.070	5.200
B^d	0	0	0	0.110	0	0	0.240	0	0	0.430	0	0	6.480	0	64.930	0	0	0	12.210	15.600
CCC^s	0	0.220	0	0	0	0	0.220	0	0	1.300	0	0	2.380	0	0	11.240	0	64.850	0	19.790
CCC^d	0	0.220	0	0	0	0	0.220	0	0	1.300	0	0	2.380	0	0	11.240	0	25.270	0	59.370
D	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100.0

Table 14. Momentum sensitive calibrate with initial extended matrix for the model 2 using S&P ratings

	AAA^s	AAA^u	AA^s	AA^u	AA^d	A^s	A^u	A^d	BBB^s	BBB^u	BBB^d	BB^s	BB^u	BB^d	B^s	B^u	B^d	CCC^s	CCC^d	D
AAA^s	90.810	0	0	0	8.330	0	0	0.680	0	0	0.06	0	0	0.12	0	0	0	0	0	0
AAA^u	90.810	0	0	0	8.330	0	0	0.680	0	0	0.06	0	0	0.12	0	0	0	0	0	0
AA^s	0	0.700	91.315	0	0	0	0	7.191	0	0	0.591	0	0	0.055	0	0	0.129	0	0.018	0
AA^u	0	0.700	91.315	0	0	0	0	7.191	0	0	0.591	0	0	0.055	0	0	0.129	0	0.018	0
AA^d	0	0.700	73.350	0	0	0	0	23.370	0	0	1.920	0	0	0.180	0	0	0.420	0	0.060	0
A^s	0	0.090	0	2.270	0	92.210	0	0	0	0	4.548	0	0	0.609	0	0	0.214	0	0.008	0.049
A^u	0	0.090	0	2.270	0	92.210	0	0	0	0	4.548	0	0	0.609	0	0	0.214	0	0.008	0.049
A^d	0	0.090	0	2.270	0	77.870	0	0	0	0	16.560	0	0	2.220	0	0	0.780	0	0.030	0.180
BBB^s	0	0.020	0	0.330	0	0	5.950	0	86.955	0	0	0	0	4.601	0	0	1.016	0	0.972	0.156
BBB^u	0	0.020	0	0.330	0	0	5.950	0	86.955	0	0	0	0	4.601	0	0	1.016	0	0.972	0.156
BBB^d	0	0.020	0	0.330	0	0	5.950	0	70.390	0	0	0	0	15.900	0	0	3.510	0	3.360	0.540
BB^s	0	0.030	0	0.140	0	0	0.670	0	0	7.730	0	81.959	0	0	0	0	7.681	0	0.869	0.921
BB^u	0	0.030	0	0.140	0	0	0.670	0	0	7.730	0	81.959	0	0	0	0	7.681	0	0.869	0.921
BB^d	0	0.030	0	0.140	0	0	0.670	0	0	7.730	0	58.730	0	0	0	0	26.520	0	3.000	3.180
B^s	0	0	0	0.110	0	0	0.240	0	0	0.430	0	0	6.480	0	85.624	0	0	0	3.124	3.992
B^u	0	0	0	0.110	0	0	0.240	0	0	0.430	0	0	6.480	0	85.624	0	0	0	3.124	3.992
B^d	0	0	0	0.110	0	0	0.240	0	0	0.430	0	0	6.480	0	64.930	0	0	0	12.210	15.600
CCC^s	0	0.220	0	0	0	0	0.220	0	0	1.300	0	0	2.380	0	0	11.240	0	71.291	0	13.349
CCC^d	0	0.220	0	0	0	0	0.220	0	0	1.300	0	0	2.380	0	0	11.240	0	25.270	0	59.370
D	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100.0

13. Appendix B

Syntaxes for the long-term migration risks

Syntax 1. *Expanded matrix(vs.0) one-to-one transition matrix without downgrade momentum*

```
function [Bg] = MatrixExt0(P)

% GOAL:
%   This program aims is to simply expand the annual transition matrix    into an extended transition matrix with a one-to-one correspondence.
%   This extension does not capture the downgrade momentum effect.
%
% INPUT:
%   P          P is square matrix {M x M} with minimum length of 4. Ratings are ranks from highest credit quality to default.
%               Example: {AAA AA A BBB BB CCC D}.
%
% OUTPUT:
%   Bg         Bg returns the extended transition matrix {N x N} that is segregated into 3 classes (stable, upgrade, and downgrade).
%               Within each segregated class; ratings are ranks from the highest credit quality to lowest. Bg is recalibrated such
%                that the sums of each row is 1.
%               Example: {AAA* AA* A* BBB* BB* B* CCC*}; {AAA^ AA^ A^ BBB^BB^ B^}; {AA~ A~ BBB~ BB~ B~ CCC~ D}.
%
%             - star sign * refers to stable rating
%             - power sign ^ refers to upgraded rating
%             - tilde sign ~ refers to downgraded rating
%
% EDITING INFORMATION:
%   Author: Vivien Djiambou (FEM), 2009
%   Last update: September 8, 2009

[mm,nn] = size(P);
if mm ~= nn
   disp('Error: Matrix must be square');
   return,
end;
if mm < 4
   disp('Error: Minimum matrix length is 4');
   return,
end;
l = length(P)-1;
w = 2*l-1;

% compute length of the segregated extended transition matrix Bg
v = 3*l-1;

% create 7x7 upper triangular matrix for ratings that previously held their ratings but will be downgraded in the next period.
% Example: AAA*--> AA~, A*-->BB~, BBB*-->CCC~, B*-->D, CCC*-->D.
for j = 1:l;
   k = j+w;
   m = v;
   while k <= m
      Bg(j,k) = P(j,k-w+1);
      k = k+1;
   end;
end;

% create 6x6 lower triangular matrix for ratings that previously held their ratings but will be upgraded in the next period.
% Example: AA*--> AAA^, A*-->AAA^, BBB*-->A^, B*-->AA^, CCC*-->B^.
for j = 2:l;
   k = l+1;
   m = j+l-1;
   while k <= m
      Bg(j,k) = P(j,k-l);
      k = k+1;
   end;
end;
```

```
% create 7x7 diagonal matrix that represents ratings that have previously held the same ratings and will keep the same ratings in the next period.
% Example: AAA*--> AAA*, AA*-->AA*, A*-->A*, BBB*-->BBB*
for j = 1:l;
   Bg(j,j) = 1-sum(Bg(j,l+1:v));
end;

% create 6x7 upper triangular matrix for ratings that were previously  upgraded and will be downgraded in the next period.
% Example: AAA^ --> AA~, AA^-->A~, A^-->B~, BB^-->A~, B^-->CCC~, BB^-->D
for j = l+1:w;
   k = j+w-l;
   m = v;
   while k <= m
      Bg(j,k) = P(j-l,k-w+1);
      k = k+1;
   end;
end;

% create 5x5 lower triangular matrix for ratings that were previously upgraded and will also be upgraded in the next period.
% Example: AA^ --> AAA^, A^-->AAA^, BBB^-->AA^, BB^-->A^, B^-->BB^
for j = l+1:w;
   for k = l+1:j-1;
      Bg(j,k) = P(j-l,k-l);
   end;
end;

% create 6x6 diagonal matrix for ratings that were previously upgraded but will hold on into their ratings in the next period.
% Example: AAA^ --> AAA*, AA^-->AA*, BBB^-->BBB*, BB^-->BB*, B^-->B*
for j = 1:l-1;
   Bg(j+l,j) = 1-sum(Bg(j+l,l+1:v));
end;

% create 6x6 upper triangular matrix for ratings that were previously downgraded and also will be further downgraded in the next period.
% Example: AA~ --> A~, A~ -->BB~, AA~ -->BBB~, BB~ -->B~, B~ -->CCC~
for j = w+1:v-1;
   k = j+1;
   m = v;
   while k <= m
      Bg(j,k) = P(j-w+1,k-w+1);
      k = k+1;
   end;
end;
Bg(v,v) = 1;

% create 6x6 lower triangular matrix for ratings that were previously downgraded but will be upgraded in the next period.
% Example: AA~ --> AAA^, A~ -->AA^, A~ -->AAA^, BBB~ -->A^, CCC~ -->B^
for j = w+1:v-1;
   k = l+1;
   m = j-w+l;
   while k <= m
      Bg(j,k) = P(j-w+1,k-l);
      k = k+1;
   end;
end;

% create 6x6 upper triangular matrix for ratings that were previously downgraded but will hold on into their ratings in the next period.
% Example: AAA~ --> AAA*, AA~ -->AA*, A~ -->A*, BB~ -->BB*, B~ -->B*
for j = w+1:v-1;
   Bg(j,j-w+1) = 1-sum(Bg(j,l+1:v));
end;
end
```

Syntax 2. *Expanded matrix (vs.1) one-to-one transition matrix with downgrade momentum*

```
function [Bg] = MatrixExt1(P)

% GOAL:
%   This program aims to extend the annual transition matrix into an extended transition matrix based on one-to-one correspondence or split
%   that capture the downgrade momentum effect. Transition intensities of firms that have been downgraded last year and that will be downgraded
%   next year are tripled the transition intensities of the initial annual transition matrix.
%
% INPUT:
%   P          P is square matrix {M x M} with minimum length of 4. Ratings are ranks from highest credit quality to default.
%              Example: {AAA AA A BBB BB CCC D}.
%
% OUTPUT:
%   Bg         Bg returns the extended transition matrix {N x N} that is segregatedinto 3 classes (stable, upgrade, and downgrade). Within each
%               segregated class, ratings are ranks from the highest credit quality to lowest. Bg is recalibrated such that the sums of each row is 1.
%               Example: {AAA* AA* A* BBB* BB* B* CCC*}; {AAA^ AA^ A^ BBB^BB^ B^}; {AA~ A~ BBB~ BB~ B~ CCC~ D}.
%
%            - star sign * refers to stable rating
%            - power sign ^ refers to upgraded rating
%            - tilde sign ~ refers to downgraded rating
%
% EDITING INFORMATION:
%   Author: Vivien Djiambou (FEM), 2009
%   Last update: September 8, 2009

[mm,nn] = size(P);
if mm ~= nn
   disp('Error: Matrix must be square');
   return,
end;
if mm < 4
   disp('Error: Minimum matrix size is 4');
   return,
end;
l = length(P)-1;
w = 2*l-1;

% compute length of the segregated extended transition matrix Bg
v = 3*l-1;

% create 7x7 upper triangular matrix for ratings that previously held their ratings but will be downgraded in the next period.
% Example: AAA*--> AA~,A*-->BB~, BBB*-->CCC~, B*-->D, CCC*-->D.
for j = 1:l;
   k = j+w;
   m = v;
   while k <= m
      Bg(j,k) = P(j,k-w+1);
      k = k+1;
   end;
end;

% create 6x6 lower triangular matrix for ratings that previously held their ratings but will be upgraded in the next period.
% Example: AA*--> AAA^, A*-->AAA^, BBB*-->A^, B*-->AA^, CCC*-->B^.
for j = 2:l;
   k = l+1;
   m = j+l-1;
   while k <= m
      Bg(j,k) = P(j,k-l);
      k = k+1;
   end;
end;

% create 7x7 diagonal matrix that represents ratings that have previously held the same ratings and will keep the same ratings in the next period.
% Example: AAA*--> AAA*, AA*-->AA*, A*-->A*, BBB*-->BBB*
for j = 1:l;
   Bg(j,j) = 1-sum(Bg(j,l+1:v));
end;

% create 6x7 upper triangular matrix for ratings that were previously upgraded and will be downgraded in the next period.
```

```
% Example: AAA^ --> AA~, AA^-->A~, A^-->B~, BB^-->A~, B^-->CCC~, BB^-->D
for j = l+1:w;
   k = j+w-l;
   m = v;
   while k <= m
      Bg(j,k) = P(j-l,k-w+1);
      k = k+1;
   end;
end;

% create 5x5 lower triangular matrix for ratings that were previously upgraded and will also be upgraded in the next period.
% Example: AA^ --> AAA^, A^-->AAA^, BBB^-->AA^, BB^-->A^, B^-->BB^
for j = l+1:w;
   for k = l+1:j-1;
      Bg(j,k) = P(j-l,k-l);
   end;
end;

% create 6x6 diagonal matrix for ratings that were previously upgraded but will keep the same ratings in the next period.
% Example: AAA^ --> AAA*, AA^-->AA*, BBB^-->BBB*, BB^-->BB*, B^-->B*.
for j = 1:l-1;
   Bg(j+l,j) = 1-sum(Bg(j+l,l+1:v));
end;

% create 6x6 upper triangular matrix for ratings that were previously downgraded but will be further downgraded in the next period.
% Example: AA~ --> A~, A~ -->BB~, AA~ -->BBB~, BB~ -->B~, B~ -->CCC~
for j = w+1:v-1;
   k = j+1;
   m = v;
   while k <= m
      Bg(j,k) = 3*P(j-w+1,k-w+1);
      k = k+1;
   end;
end;
Bg(v,v) = 1;

% create 6x6 lower triangular matrix for ratings that were previously downgraded but will be upgraded in the next period.
% Example: AA~ --> AAA^, A~ -->AA^, A~ -->AAA^, BBB~ -->A^, CCC~ -->B^
for j = w+1:v-1;
   k = l+1;
   m = j-w+l;
   while k <= m
      Bg(j,k) = P(j-w+1,k-l);
      k = k+1;
   end;
end;

% create 6x6 upper triangular matrix for ratings that were previously downgraded but will keep the same ratings in the next period.
% Example: AAA~ --> AAA*, AA~ -->AA*, A~ -->A*, BB~ -->BB*, B~ -->B*
for j = w+1:v-1;
   Bg(j,j-w+1) = 1-sum(Bg(j,l+1:v));
end;
end
```

Syntax 3. *Expanded matrix (vs.2) proportional transition matrix with downgrade momentum*

```
function [Bg] = MatrixExt2(P,rhod)

% GOAL:
%   This program aims to extend the annual transition matrix into an extended transition matrix based on the proportions of prior downgraded
%   firms to capture the downgrade momentum effect. Transition intensities of firms that have been downgraded last year and that will be
%   downgraded next year are tripled the transition intensities in the annual transition matrix.
%
% INPUT:
%   P          P is square matrix {M x M} with minimum length of 4. Ratings are ranks from highest credit quality to default.
%              Example: {AAA AA A BBB BB CCC D}
%
```

```
%   rhod        rhod returns the proportions of downgraded firms
%               Example: {AAA~ AA~ A~ BBB~ BB~ B~ CCC~}
%
% OUTPUT:
%   Bg          Bg returns the extended transition matrix {N x N} that is segregated into 3 classes (stable, upgrade, and downgrade). Within each
%               segregated class, ratings are ranks from the highest credit quality to lowest. Bg is recalibrated such that the sums of each row is 1.
%               Example: {AAA* AA* A* BBB* BB* B* CCC*}; {AAA^ AA^ A^ BBB^BB^ B^}; {AA~ A~ BBB~ BB~ B~ CCC~ D}
%
%             - star sign * refers to stable rating
%             - power sign ^ refers to upgraded rating
%             - tilde sign ~ refers to downgraded rating
%
% EDITING INFORMATION:
%   Author: Vivien Djiambou (FEM), 2009
%   Last update: September 8, 2009

[mm,nn] = size(P);
if mm ~= nn
   disp('Error: Matrix must be square');
   return,
end;
if mm < 4
   disp('Error: Minimum matrix size is 4');
   return,
end;
l = length(P)-1;
w = 2*l-1;

% compute length of the segregated extended transition matrix Bg
v = 3*l-1;

dd = 3*(length(rhod)+1)-4;

if v ~= dd
   disp('Error: Portfolio must be equal length as the Matrix length');
   return,
end;

% create 7x7 upper triangular matrix for the proportions of firms that held their ratings previously but will be downgraded in the next period.
% Example: AAA*--> AA~, A*-->BB~, BBB*-->CCC~, B*-->D, CCC*-->D.
for j = 1:l;
   k = j+w;
   m = v;
   while k <= m
      Bg(j,k) = (P(j,k-w+1)*(1-3*rhod(j)))/(1-rhod(j));
      k = k+1;
   end;
end;

% create 6x6 lower triangular matrix for the proportions of firms that have held their ratings previously but will be upgraded in the next period.
% Example: AA*--> AAA^, A*-->AAA^, BBB*-->A^, B*-->AA^, CCC*-->B^.
for j = 2:l;
   k = l+1;
   m = j+l-1;
   while k <= m
      Bg(j,k) = P(j,k-l);
      k = k+1;
   end;
end;

% create 7x7 diagonal matrix for the proportions of firms that have previously held the same ratings and they will keep the same ratings in
% the next period. Example: AAA*--> AAA*, AA*-->AA*, A*-->A*, BBB*-->BBB*
for j = 1:l;
   Bg(j,j) = 1-sum(Bg(j,l+1:v));
end;

% create 6x7 upper triangular matrix for the proportions of firms that were previously upgraded and will be downgraded in the next period.
% Example: AAA^ --> AA~, AA^-->A~, A^-->B~, BB^-->A~, B^-->CCC~, BB^-->D
for j = l+1:w;
   k = j+w-l;
```

```
    m = v;
    while k <= m
        Bg(j,k) = (P(j-l,k-w+1)*(1-3*rhod(j-l)))/(1-rhod(j-l));
        k = k+1;
    end;
end;

% create 5x5 lower triangular matrix for the proportions of firms that were previously upgraded and will also be upgraded in the next period.
% Example: AA^ --> AAA^, A^-->AAA^, BBB^-->AA^, BB^-->A^, B^-->BB^
for j = l+1:w;
    for k = l+1:j-1;
        Bg(j,k) = P(j-l,k-l);
    end;
end;

% create 6x6 diagonal matrix for the proportion of firms that were previously upgraded but will keep the same ratings in the next period.
% Example: AAA^ --> AAA*, AA^-->AA*, BBB^-->BBB*, BB^-->BB*, B^-->B*
for j = 1:l-1;
    Bg(j+l,j) = 1-sum(Bg(j+l,l+1:v));
end;

% create 6x6 upper triangular matrix for the proportions of firms that were previously downgraded but will be further downgraded in the next period.
% Example: AA~ --> A~, A~ -->BB~, AA~ -->BBB~, BB~ -->B~, B~ -->CCC~
for j = w+1:v-1;
    k = j+1;
    m = v;
    while k <= m
        Bg(j,k) = 3*P(j-w+1,k-w+1);
        k = k+1;
    end;
end;
Bg(v,v) = 1;

% create 6x6 lower triangular matrix for the proportion of firms that were previously downgraded but will be upgraded in the next period.
% Example: AA~ --> AAA^, A~ -->AA^, A~ -->AAA^, BBB~ -->A^, CCC~ -->B^
for j = w+1:v-1;
    k = l+1;
    m = j-w+l;
    while k <= m
        Bg(j,k) = P(j-w+1,k-l);
        k = k+1;
    end;
end;

% create 6x6 upper triangular matrix for the proportions of firms that were previously downgraded but will keep the same ratings in the next period.
% Example: AAA~ --> AAA*, AA~ -->AA*, A~ -->A*, BB~ -->BB*, B~ -->B*
for j = w+1:v-1;
    Bg(j,j-w+1) = 1-sum(Bg(j,l+1:v));
end;
end
```

Syntax 4. Extended portfolio distribution

```
function [mmu] = MUExt(P,muu)

% GOAL:
%   This program returns the number of firms that fit extended transition matrix by computing the expected number of firms within each rating
%   class that comes from stable, upgrade, and downgrade.
%
% INPUT:
%   P          P is square matrix {M x M} with minimum length of 4. Ratings are ranks from highest credit quality to default.
%              Example: {AAA AA A BBB BB CCC D}
%
%   muu        muu is number of firms {1 x M} rank from the highest to the lowest rating based on the annual transition matrix P
%
% OUTPUT:
%   mmu        mmu is matrix vector {1 x N} corresponding to the number of firms within the extended transition matrix
%
```

```
% EDITING INFORMATION:
%   Author: Vivien Djiambou (FEM), 2009
%   Last update: September 8, 2009

l = length(P);
m = 3*l-4;

for i = 1:l;
    if i == 1
        mudo(i) = 0;
        muup(i) = 0;
        for j = i+1:l;
            muup(i) = muup(i)+muu(j)*P(j,i);
        end;
        must(i) = muu(i)-mudo(i)-muup(i);

    elseif i>1 && i<l-1
        mudo(i) = 0;
        for j = 1:i-1;
            mudo(i) = mudo(i)+muu(j)*P(j,i);
        end;
        muup(i) = 0;
        for j = i+1:l;
            muup(i) = muup(i)+muu(j)*P(j,i);
        end;
        must(i) = muu(i)-mudo(i)-muup(i);

    elseif i == l-1
        mudo(i) = 0;
        for j = 1:i-1;
            mudo(i) = mudo(i)+muu(j)*P(j,i);
        end;
        muup(i) = 0;
        must(i) = muu(i)-mudo(i)-muup(i);

    elseif i == l
        mudo(i) = 0;
    end;
end;

% Rank number of firm based on the extended transition matrix from  highest credit quality to default
mmu(1,1) = must(1);
mmu(1,2) = muup(1);
s = 3;
for k = 2:l-2;
    mmu(1,s) = must(k);
    mmu(1,s+1) = muup(k);
    mmu(1,s+2) = mudo(k);
    s = s+3;
end;
mmu(1,m-2) = must(l-1);
mmu(1,m-1) = mudo(l-1);
mmu(1,m) = mudo(l);
end
```

Syntax 5. *Vector of downgraded proportion of portfolio distribution*

```
function [rhod] = MUFract(mmu)

% GOAL:
%   This program compute the proportions of downgraded firms within the extended transition matrix.
%
% INPUT:
%   mmu         mmu is matrix vector {1 x N} corresponding to the number of firms within the extended transition matrix.
%
% OUTPUT:
%   rhod        rhod returns the proportions of downgraded firms. Example: {AAA~ AA~ A~ BBB~ BB~ B~ CCC~}
%
```

```
%            - star sign * refers to stable rating
%            - power sign ^ refers to upgraded rating
%            - tilde sign ~ refers to downgraded rating
%
% EDITING INFORMATION:
%   Author: Vivien Djiambou (FEM), 2009
%   Last update: September 8, 2009

% compute the length of the extended state
dd = length(mmu);

d = round(((dd+4)/3))-1;

% compute proportions of firms with the highest ratings (AAA* and AAA^)
for k = 1:2;
   pan = 0;
   for m = 1:2;
      % sum number of firms in the ratings AAA* and AAA^
      pan = pan + mmu(1,m);
   end;
   pt(1,k) = mmu(1,k)/pan;
end;

% compute proportions of firms with non-special ratings. Example: {AA* AA^ AA~}; {A* A^ A~}; {BBB* BBB^ BBB~}; {BB* BB^ BB~}
z = k+1;
while z < dd-2;
   for k = z:z+2;
      pan = 0;
      for m = z:z+2;
         % sum number of firms each rating class
         pan = pan + mmu(1,m);
      end;
      pt(1,k) = mmu(1,k)/pan;
   end;
   z = k+1;
end;

% compute proportions of firms with the lowest ratings (CCC* and CCC~)
for k = dd-2:dd-1;
   pan = 0;
   for m = dd-2:dd-1;
      % sum number of firms in the ratings CCC* and CCC~
      pan = pan + mmu(1,m);
   end;
   pt(1,k) = mmu(1,k)/pan;
end;

% returns the proportions of firms that are downgraded
rhod(:,1) = 0;
xx = 0;
for j = 2:d-1
   rhod(:,j) = pt(:,xx+5);
   xx = xx+3;
end;
rhod(:,d) = pt(:,dd-1);
end
```

Syntax 6. *Block matrix for transforming three group into a single group*

```
function [PB] = MatrixBlock(Bg)

% GOAL:
%   This program shifts rows and columns of the segregated extended transition matrix into a single grouped extended transition matrix with rating
%   classes starting from the highest credit quality to Default.
%
% INPUT:
%   Bg        Bg returns the extended transition matrix {N x N} that is segregated into 3 classes (stable, upgrade, and downgrade).
%             Within each segregated class, ratings are ranks from the
```

```
%            highest credit quality to lowest. Bg is recalibrated such that the sums of each row are 1.
%            Example: {AAA* AA* A* BBB* BB* B* CCC*}; {AAA^ AA^ A^ BBB^  BB^ B^}; {AA~ A~ BBB~ BB~ B~ CCC~ D}
%
% OUTPUT:
%   PB         PB returns the extended transition matrix {N x N} that is single grouped from highest to default. BB^ BB~ CCC* CCC~ D}.
%
%            - star sign * refers to stable rating
%            - power sign ^ refers to upgraded rating
%            - tilde sign ~ refers to downgraded rating
%
% EDITING INFORMATION:
%   Author: Vivien Djiambou (FEM), 2009
%   Last update: September 8, 2009

[mm,nn] = size(Bg);
if mm ~= nn
   disp('Error: Matrix must be square');
   return,
end;
if mm < 8
   disp('Error: Minimum matrix size is 8');
   return,
end;
m = length(Bg);
l = round(((m+4)/3))-1;
jj = (l+1)-4;
kk = 2*((l+1)-4);

% shift matrix columns from highest to lowest ratings
temp = Bg;
Bg(:,2) = temp(:,jj+4);
Bg(:,3) = temp(:,2);
Bg(:,4) = temp(:,jj+5);
Bg(:,5) = temp(:,kk+6);
Bg(:,6) = temp(:,3);
ss = 5;
rr = 6;
uu = 3;
zz = 7;
for u = 1:jj;
   Bg(:,zz) = temp(:,jj+ss+1);
   Bg(:,zz+1) = temp(:,kk+rr+1);
   Bg(:,zz+2) = temp(:,uu+1);
   zz = zz+3;
   ss = ss+1;
   rr = rr+1;
   uu = uu+1;
end;
% shift matrix rows from highest to lowest ratings.
tempo = Bg;
Bg(2,:) = tempo(jj+4,:);
Bg(3,:) = tempo(2,:);
Bg(4,:) = tempo(jj+5,:);
Bg(5,:) = tempo(kk+6,:);
Bg(6,:) = tempo(3,:);
ss = 5;
rr = 6;
uu = 3;
zz = 7;
for u = 1:jj;
   Bg(zz,:) = tempo(jj+ss+1,:);
   Bg(zz+1,:) = tempo(kk+rr+1,:);
   Bg(zz+2,:) = tempo(uu+1,:);
   zz = zz+3;
   ss = ss+1;
   rr = rr+1;
   uu = uu+1;
end;
PB = Bg;
end
```

Syntax 7. *Compute the extended portfolios and extended transition matrices for the n years*

```
function [PB,mmu] = MatrixPowerExt(PB,mmu,n)

% GOAL:
%   This program compute extended transition matrices and the number of firms for n successive years.
%
% INPUT:
%   PB         PB is the initial extended transition matrix {N x N} with min length of 8.
%              Example: {AAA* AAA+ AA* AA+ AA- A* A+ A- BBB* BBB+ BBB- BB*BB+ BB- CCC* CCC- D}
%
%   mmu         mmu is matrix vector {1 x M} at t =0 of the number of firms for the initial extended transition matrix
%
%   n           n is the number of years
%
% OUTPUT:
%   PB         PB returns the extended transition matrices {N x N x n} with min length of 8 for n successive years.
%              Example: {AAA* AAA+ AA* AA+ AA- A* A+ A- BBB* BBB+ BBB- BB* BB+ BB- CCC* CCC- D}
%
%   mmu         mmu is the vectors {1 x M x n} that returns the number of firms for the n years extended transition matrices.
%
%           - star sign * refers to stable rating
%           - power sign ^ refers to upgraded rating
%           - tilde sign ~ refers to downgraded rating
%
% EDITING INFORMATION:
%   Author: Vivien Djiambou (FEM), 2009
%   Last update: September 8, 2009

mmu(:,:,1) = mmu(:,:);
PB(:,:,1) = PB(:,:);
for i = 2:n+1;
   % compute the number of portfolio in each rating for the n-1 years
   mmu(:,:,i) = mmu(:,:,i-1)*PB(:,:,i-1);
   % compute the extended transition matrices for the n-1 years
   PB(:,:,i) = PB(:,:,1)*PB(:,:,i-1);
end;
end
```

Syntax 8. *Transform the extended transition matrices into transition matrices*

```
function [PR] = MatrixShrink(PB,mmu,n)

% GOAL:
%   This program returns the aggregated transition matrices. The transition intensities of the n extended transition matrices are aggregated back
%   into transition matrices.
%
% INPUT:
%   PB        PB are extended transition matrices {N x N x n} with minimum length of 8.
%             Example: {AAA* AAA+ AA* AA+ AA- A* A+ A- BBB* BBB+ BBB- BB*BB+ BB- CCC* CCC- D}.
%
%   mmu       mmu is a vector {1 x M} corresponding to the number of portfolio in the extended transition matrix.
%
%   n          n is the number of years
%
% OUTPUT:
%   PR         PR returns n years aggregated transition matrices {M x M} with minimum length 4 ranked ratings from highest to default
%              Example: {AAA AA A BBB BB CCC D}
%
%           - star sign * refers to stable rating
%           - power sign ^ refers to upgraded rating
%           - tilde sign ~ refers to downgraded rating
%
% EDITING INFORMATION:
%   Author: Vivien Djiambou (FEM), 2009
%   Last update: September 8, 2009
```

```
[mm,nn] = size(PB(:,:,1));
dd = length(mmu(:,:,1));
for i = 2:n+1;
    if mm ~= size(PB(:,:,i),1) || nn ~= size(PB(:,:,i),2)
        disp('Error: All matrices must be equal sizes');
        return,
    end;
    if dd ~= length(mmu(:,:,i));
        disp('Error: All secsiurities vectors must be equal length');
        return,
    end;
end;
if mm ~= nn
    disp('Error: Matrix must be square');
    return,
end;
v = length(PB(:,:,1));
for i = 2:n+1;
    if v ~= length(PB(:,:,i))
        disp('Error: All n-matrices must be equal length');
        return,
    end;
end;
if v < 8
    disp('Error: Minimum matrix size is 8');
    return,
end;
if v ~= dd
    disp('Error: Portfolio must be equal length to the Matrix');
    return,
end;

% Ensure that all values in the matrices are non-negatives
for i = 1:n+1;
    for j = 1:v;
        for k = 1:v;
            if PB(j,k,i)<0
                disp('Error: Matrix components must be positive');
                return,
            end;
        end;
    end;
end;

% compute the length of the transition matrix less default rating
l = round(((v+4)/3))-1;

% resize extended transition matrices into transition matrices
for i = 1:n+1;
    q = 0;
    % reduce matrix rows into its original form (Ex: AAA AA ... CCC D)
    for j = 1:l+1;
        z = 0;
        % reduce matrix columns into its original form (Ex: AAA AA ... CCC D)
        for k = 1:l+1;
            if j == 1

                % reduce ratings AAA* and AAA^ into single rating AAA
                if k == 1
                    int = 0;
                    penn = 0;
                    for m = 1:2;
                        pan = 0;
                        for s = 1:2;
                            % sum each rows of the highest ratings AAA* and AAA^
                            pan = pan + PB(m,s,i);
                        end;
                        % sum numbers of securities in ratings class AAA* AAA^
                        int = int + mmu(1,m,i);
```

```
        penn = penn + pan*mmu(1,m,i);
      end;
      % compute transition probability of AAA -> AAA
      PR(j,k,i) = penn/int;
      z = s+1;

    % reduce ratings {AA* AA^ AA~},{A* A^ A~}... {B* B^ B~} into single ratings AA, A, ..., B.
    elseif k>1 && k<l
      int = 0;
      penn = 0;
      for m = 1:2;
        pan = 0;
        for s = z:z+2;
          % sum sub-ratings of each rating class
          pan = pan + PB(m,s,i);
        end;
        % sum numbers of securities in ratings class AAA* AAA^
        int = int + mmu(1,m,i);
        penn = penn + pan*mmu(1,m,i);
      end;
      % compute transition probabilities of AAA->AA, AAA->A,... AAA->B
      PR(j,k,i) = penn/int;
      z = s+1;

    % reduce ratings CCC* and CCC~ into single rating CCC
    elseif k == l
      int = 0;
      penn = 0;
      for m = 1:2;
        pan = 0;
        for s = z:z+1;
          % sum each rows of the highest ratings CCC* CCC~
          pan = pan + PB(m,s,i);
        end;
        int = int + mmu(1,m,i);
        penn = penn + pan*mmu(1,m,i);
      end;
      % compute transition probability of AAA -> CCC
      PR(j,k,i) = penn/int;

    % default state is unique
    elseif k == l+1
      int = 0;
      penn = 0;
      for m = 1:2;
        % sum numbers of securities in ratings class AAA* AAA^
        int = int + mmu(1,m,i);
        penn = penn + PB(m,v,i)*mmu(1,m,i);
      end;
      % compute transition probability of AAA -> D
      PR(j,k,i) = penn/int;
      q = m+1;
    end;

  elseif j>1 && j<l
    % reduce ratings AAA* and AAA^ into single rating AAA
    if k == 1
      int = 0;
      penn = 0;
      for m = q:q+2;
        pan = 0;
        for s = 1:2;
          pan = pan + PB(m,s,i);
        end;
        % sum numbers of securities in each rating classes. Example: {AA* AA^ AA~},{A* A^ A~}... {B* B^ B~}
        int = int + mmu(1,m,i);
        penn = penn + pan*mmu(1,m,i);
      end;
      % compute transition probabilities of AA->AAA,A->AAA,...,B->AAA
      PR(j,k,i) = penn/int;
```

```
      z = s+1;

    % reduce ratings {AA* AA^ AA~},{A* A^ A~}... {B* B^ B~} into single ratings AA, A, ..., B.
    elseif k>1 && k<l
      int = 0;
      penn = 0;
      for m = q:q+2;
         pan = 0;
         for s = z:z+2;
            pan = pan + PB(m,s,i);
         end;
         % sum numbers of securities in each rating classes. Example: {AA* AA^ AA~},{A* A^ A~}... {B* B^ B~}
         int = int + mmu(1,m,i);
         penn = penn + pan*mmu(1,m,i);
      end;
      % compute transition probabilities for ratings (eg. AA->AA,AA->A,...,AA->B; ...; B->AA,B->A,...,B->B)
      PR(j,k,i) = penn/int;
      z = s+1;

    % reduce ratings {CCC* CCCC~} into single rating CCC
    elseif k == l
      int = 0;
      penn = 0;
      for m = q:q+2;
         pan = 0;
         for s = z:z+1;
            pan = pan + PB(m,s,i);
         end;
         % sum numbers of securities in each rating classes. Example: {AA* AA^ AA~},{A* A^ A~}... {B* B^ B~}
         int = int + mmu(1,m,i);
         penn = penn + pan*mmu(1,m,i);
      end;
      % compute transition probabilities for the ratings (eg. AA->CCC, A->CCC, BBB->CCC, BB->CCC, B->CCC)
      PR(j,k,i) = penn/int;

    % default state has single rating D
    elseif k == l+1
      int = 0;
      penn = 0;
      for m = q:q+2;
         % sum numbers of securities in each rating classes. Example: {AA* AA^ AA~},{A* A^ A~}... {B* B^ B~}
         int = int + mmu(1,m,i);
         penn = penn + PB(m,v,i)*mmu(1,m,i);
      end;
      % compute transition probabilities for ratings (eg. AA->D, A->D, BBB->D, BB->D, B->D)
      PR(j,k,i) = penn/int;
      q = m+1;
    end;

  elseif j == l
    % reduce ratings AAA* and AAA^ into single rating AAA
    if k == 1
      int = 0;
      penn = 0;
      for m = v-2:v-1;
         pan = 0;
         for s = 1:2;
            pan = pan + PB(m,s,i);
         end;
         % sum numbers of securities in ratings class CCC* CCC~
         int = int + mmu(1,m,i);
         penn = penn + pan*mmu(1,m,i);
      end;
      % compute transition probability of CCC->AAA
      PR(j,k,i) = penn/int;
      z = s+1;

    % reduce ratings {AA* AA^ AA~},{A* A^ A~}... {B* B^ B~} into single ratings AA, A, ..., B.
    elseif k>1 && k<l
      int = 0;
```

```
        penn = 0;
        for m = v-2:v-1;
            pan = 0;
            for s = z:z+2;
                pan = pan + PB(m,s,i);
            end;
            % sum numbers of securities in ratings class CCC* CCC~
            int = int + mmu(1,m,i);
            penn = penn + pan*mmu(1,m,i);
        end;
        % compute transition probabilities of CCC->AA,CCC->A,...,CCC->B
        PR(j,k,i) = penn/int;
        z = s+1;

    % reduce ratings {CCC* CCCC~} into single rating CCC
    elseif k == l
        int = 0;
        penn = 0;
        for m = v-2:v-1;
            pan = 0;
            for s = z:z+1;
                pan = pan + PB(m,s,i);
            end;
            % sum numbers of securities in ratings class CCC* CCC~
            int = int + mmu(1,m,i);
            penn = penn + pan*mmu(1,m,i);
        end;
        % compute transition probability of CCC -> CCC
        PR(j,k,i) = penn/int;

    % default state has single rating D
    elseif k == l+1
        int = 0;
        penn = 0;
        for m = v-2:v-1;
            % sum numbers of securities in ratings class CCC* CCC~
            int = int + mmu(1,m,i);
            penn = penn + PB(m,v,i)*mmu(1,m,i);
        end;
        % compute transition probability of CCC -> D
        PR(j,k,i) = penn/int;
    end;

elseif j == l+1
    m = v;
    % reduce ratings AAA* and AAA^ into single rating AAA
    if k == 1
        pan = 0;
        for s = 1:2;
            pan = pan + PB(m,s,i);
        end;
        penn = pan*mmu(1,m,i);
        % compute transition probability of D -> AAA
        PR(j,k,i) = penn;
        z = s+1;

    % reduce ratings {AA* AA^ AA~},{A* A^ A~}... {B* B^ B~} into single ratings AA, A, ..., B.
    elseif k>1 && k<l
        pan = 0;
        for s = z:z+2;
            pan = pan + PB(m,s,i);
        end;
        penn = pan*mmu(1,m,i);
        % compute transition probabilities of D->AA,D->A,...,D->B
        PR(j,k,i) = penn;
        z = s+1;

    % reduce ratings {CCC* CCCC~} into single rating CCC
    elseif k == l
        pan = 0;
```

```
            for s = z:z+1;
                pan = pan + PB(m,s,i);
            end;
            penn = pan*mmu(1,m,i);
            % compute transition probability of D -> CCC
            PR(j,k,i) = penn;

        % default state has single rating D
        elseif k == l+1
            penn = PB(m,v,i);
            % compute transition probability of D -> D
            PR(j,k,i) = penn;
        end;
      end;
    end;
  end;
end;
end
```

Syntax 9. *Transform the extended portfolio distribution into Portfolio distribution*

```
function [musrk] = MUShrink(mmu,n)

% GOAL:
%   This program compute the proportions of downgraded firms within the extended transition matrix.
%
% INPUT:
%   mmu        mmu is matrix vector {1 x N} corresponding to the number of firms within the extended transition matrix
%
%   n           n is the number of years
%
% OUTPUT:
%   musrk       musrk returns the aggregated number of firms within the transition matrices
%               Example: {AAA AA A BBB BB B CCC}
%
% EDITING INFORMATION:
%   Author: Vivien Djiambou (FEM), 2009
%   Last update: September 8, 2009

m = length(mmu(:,:,1));
l = round((m+4)/3);

% Portfolio distribution for the extended transition matrix
for j = 1:n+1;
   for k = 1:m;
       mnmu(k,j) = mmu(1,k,j);
   end;
end;

% Portfolio distribution for the standard transition matrix
for i = 1:n+1;
   for k = 1:l;
       if k == 1;
          rest = 0;
          for s = 1:2;
              rest = rest+mnmu(s,i);
          end;
          musrk(k,i) = rest;
          z= s+1;

       elseif k>1 && k<l-1
          rest = 0;
          for s = z:z+2;
              rest = rest+mnmu(s,i);
          end;
          musrk(k,i) = rest;
          z = s+1;
```

```
        elseif k == l-1
            rest = 0;
            for s = z:z+1;
                rest = rest+mnmu(s,i);
            end;
            musrk(k,i) = rest;
            z = s+1;
        else
            musrk(k,i) = mnmu(z,i);
        end;
    end;
end;
end
```

Syntax 10. *Compute the all the three possible transition matrices and portfolio distributions*

```
function [PBB,PRB,musrk,fig,D1,D2,D3,D4] = MatrixComp(P,muu,n)

% GOAL:
%   This program aims to provide an overview with comparison of the different version. It computes n years extended transition matrices,
%   aggregated number of firms, and transition matrices.
%
% INPUT:
%   P          P is square matrix {M x M} with minimum length of 4. Ratings are ranks from highest credit quality to default.
%              Example: {AAA AA A BBB BB CCC D}
%
%   muu        muu is number of firms {1 x M} ranked from the highest to the lowest rating based on the annual transition matrix P
%
%   n          n is the number of years
%
% OUTPUT:
%   PBB        PBB are extended transition matrices {N x N x n} with minimun length of 8.
%              Example: {AAA* AAA+ AA* AA+ AA- A* A+ A- BBB* BBB+ BBB- BB* BB+ BB- CCC* CCC- D}.
%
%   PRB        PRB returns n years aggregated transition matrices {M x M} with minimun length 4 ranked ratings from highest to default
%              Example: {AAA AA A BBB BB CCC D}
%
%   musrk      musrk returns the aggregated number of firms for n years
%
%   fig        fig returns the figures of the distribution of firms and also the effects of the downgrade momentum
%
%              - star sign * refers to stable rating
%              - power sign ^ refers to upgraded rating
%              - tilde sign ~ refers to downgraded rating
%
% EDITING INFORMATION:
%   Author: Vivien Djiambou (FEM), 2009
%   Last update: September 8, 2009

% version 0: identical with one-to-one extension of the transition matrix
[Bg] = MatrixExt0(P);
[PB] = MatrixBlock(Bg);
[mmu] = MUExt(P,muu);
[PB,mmu] = MatrixPowerExt(PB,mmu,n);
[PR] = MatrixShrink(PB,mmu,n);
PB0(:,:,1) = 100*PB(:,:,1);
PR0(:,:,1) = 100*PR(:,:,1);
PR0(:,:,2) = 100*PR(:,:,n+1);
[musrk0] = MUShrink(mmu,n);

% version 1: one-to-one extension of the matrix and downgrade momentum
[Bg] = MatrixExt1(P);
[PB] = MatrixBlock(Bg);
[mmu] = MUExt(P,muu);
[PB,mmu] = MatrixPowerExt(PB,mmu,n);
[PR] = MatrixShrink(PB,mmu,n);
```

```
PB1(:,:,1) = 100*PB(:,:,1);
PR1(:,:,1) = 100*PR(:,:,1);
PR1(:,:,2) = 100*PR(:,:,n+1);
[musrk1] = MUShrink(mmu,n);

% version 2: proportion of firms in the extended matrix and downgrade momentum
[mmu] = MUExt(P,muu);
[rhod] = MUFract(mmu);
[Bg] = MatrixExt2(P,rhod);
[PB] = MatrixBlock(Bg);
[PB,mmu] = MatrixPowerExt(PB,mmu,n);
[PR] = MatrixShrink(PB,mmu,n);
PB2(:,:,1) = 100*PB(:,:,1);
PR2(:,:,1) = 100*PR(:,:,1);
PR2(:,:,2) = 100*PR(:,:,n+1);
[musrk2] = MUShrink(mmu,n);

PBB(:,:,1) = PB0;
PBB(:,:,2) = PB1;
PBB(:,:,3) = PB2;

PRB(:,:,1,1) = PR0(:,:,1);
PRB(:,:,2,1) = PR0(:,:,2);
PRB(:,:,1,2) = PR1(:,:,1);
PRB(:,:,2,2) = PR1(:,:,2);
PRB(:,:,1,3) = PR2(:,:,1);
PRB(:,:,2,3) = PR2(:,:,2);

musrk(:,:,1) = musrk0(:,:);
musrk(:,:,2) = musrk1(:,:);
musrk(:,:,3) = musrk2(:,:);

[fig] = figrs(musrk0,musrk1,musrk2,n);
[D1,D2,D3,D4] = Tech(PR0,PR1,PR2);
end
```

Syntax 11. *Graph of the portfolio distributions*

```
function [fig] = figs(musrk,n)

l = length(musrk(:,1)); % length of transition matrix

for j = 1:l;
    xx(1,j) = j;
end;

fig1 = figure;
bar(xx,musrk(:,n+1));
grid on;
xlabel('Rating category');
ylabel('Portfolio distribution');
xlim([xx(1,1) xx(1,l)]);

fig = fig1;
end
```

Syntax 12. *Graph of the of the portfolio distributions and the downgrade momentum effects on portfolios*

```
function [fig] = figrs(musrk0,musrk1,musrk2,n)

l = length(musrk0(:,1)); % length of standard transition matrix

for j = 1:l;
    xx(1,j) = j;
end;
```

```
musrk10 = musrk1-musrk0;
musrk20 = musrk2-musrk0;

fig1 = figure; % Portfolio distribution
bar(xx,musrk0(:,n+1));
grid on;
xlabel('Rating categories');
xlim([0 l+1]);
ylabel('Portfolio distribution');
ylim([0 62]);

fig2 = figure;
bar(xx,musrk1(:,n+1));
grid on;
xlabel('Rating categories');
xlim([0 l+1]);
ylabel('Portfolio distribution');

fig3 = figure;
bar(xx,musrk2(:,n+1));
grid on;
xlabel('Rating categories');
xlim([0 l+1]);
ylabel('Portfolio distribution');

% Difference between portfolio distribution
fig4 = figure;
bar(xx,musrk10(:,2));
grid on;
xlabel('Rating categories');
xlim([0 l+1]);
ylabel('Differences in portfolio distribution');

fig5 = figure;
bar(xx,musrk10(:,n+1));
grid on;
xlabel('Rating categories');
xlim([0 l+1]);
ylabel('Differences in portfolio distribution');

fig6 = figure;
bar(xx,musrk20(:,n+1));
grid on;
xlabel('Rating categories');
xlim([0 l+1]);
ylabel('Differences in portfolio distribution');

fig = [fig1;fig2;fig3;fig4;fig5;fig6];
end
```

Syntax 13. *Main program for selecting the outputs*

```
function [PBB,PRB,musrk,fig] = MatrixTrans(P,muu,n,vs)

% GOAL:
%   This program aims to compute n successive years of extended transition matrices, transition matrix, and number of firms.
%
% INPUT:
%   P          P is square matrix {M x M} with minimum length of 4. Ratings are ranks from highest credit quality to default.
%              Example: {AAA AA A BBB BB CCC D}
%
%   muu        muu is number of firms {1 x M} rank from the highest to the lowest rating based on the annual transition matrix P
%
%   vs         vs is the version number {0,1,2,3}. vs = 1 is the one-to-one extended matrix that does not capture past effects; vs = 2
%              is the one-to-one extended matrix that capture downgrade momentum; vs = 3 is based on the proportion firm in the
%              extended matrix that capture downgrade momentum
```

```
%
%   n           n is the number of years
%
% OUTPUT:
%   PBB         PBB are extended transition matrices {N x N x n} with minimum length of 8.
%                Example: {AAA* AAA+ AA* AA+ AA- A* A+ A- BBB* BBB+ BBB- BB* BB+ BB- CCC* CCC- D}.
%
%   PRB         PRB returns n years aggregated transition matrices {M x M} with minimun length 4 ranked ratings from highest to default
%                Example: {AAA AA A BBB BB CCC D}
%
%   musrk       musrk returns the aggregated number of firms for n years
%
%   fig         fig returns the figures of the distribution of firms and also the effects of the downgrade momentum
%
%             - star sign * refers to stable rating
%             - power sign ^ refers to upgraded rating
%             - tilde sign ~ refers to downgraded rating
%
% EDITING INFORMATION:
%   Author: Vivien Djiambou (FEM), 2009
%   Last update: September 8, 2009


switch vs
   case {0, 1, 2}
      % Split the number of firms based on prior ratings
      [mmu] = MUExt(P,muu);
      % version 0: identical with one-to-one extension of the transition matrix
      if vs == 0
         [Bg] = MatrixExt0(P);

      % version 1: one-to-one extension of the transition matrix and downgrade momentum
      elseif vs == 1
         [Bg] = MatrixExt1(P);

      % version 2: proportional extension of the matrix and downgrade momentum
      elseif vs == 2
         [rhod] = MUFract(mmu);
         [Bg] = MatrixExt2(P,rhod);
      end;

      % Transform the partitioned (stable,upgrade,downgrade) or segregated initial extended transition matrix into a single grouped from highest to default
      [PB] = MatrixBlock(Bg);

      % Create n transition matrices and number of firms
      [PB,mmu] = MatrixPowerExt(PB,mmu,n);

      % Aggregate the extended transition matrices into transition matrices
      [PR] = MatrixShrink(PB,mmu,n);

      PBB = 100*PB(:,:,1);
      PRB(:,:,1) = 100*PR(:,:,1);
      PRB(:,:,2) = 100*PR(:,:,n+1);

      % Aggregate back the n extended number of firms
      [musrk] = MUShrink(mmu,n);
      [fig] = figs(musrk,n);

   case {3}
      % Get overview of all results and comparisons
      [PBB,PRB,musrk,fig] = MatrixComp(P,muu,n);

   otherwise
      disp('Error: Select either 0 or 1 or 2 or 3');
      return,
end
end
```

Additional Syntaxes for the short-term migration risks

Syntax 14. Compute *expected losses with simple extended portfolio rebalancing*

```
function [PL_Exp] = SimpleRebalance(PB,mmu,PbuyExt,PsellExt,n)

% GOAL:
%   This program returns the Portfolio Expected Loss of the sum of obligors expected losses in each quarter. A simple rebalancing is performed in
%   which after each quarter, we match extended portfolio with the initial extended portfolio.
%
% INPUT:
%   PB           PB returns the extended transition matrix {v x v} that is single grouped from highest to default.
%                Example: {AAA* AAA^ AA* AA^ AA~ A* A^ A~ BBB* BBB^ BBB~ BB*BB^ BB~ CCC* CCC~ D}
%
%   mmu          mmu is matrix vector {1 x N} corresponding to the number of  firms within the extended transition matrix
%
%   PbuyExt      PbuyExt is extended price vector {1 x v} with min length of 8 at which banks can purchase on the market.
%                Example: {AAA* AAA+ AA* AA+ AA- A* A+ A- BBB* BBB+ BBB- BB* BB+ BB- CCC* CCC- D}
%
%   PsellExt     PsellExt is extended price vector {1 x v} with min length of 8 at which banks can sell on the market.
%                Example: {AAA* AAA+ AA* AA+ AA- A* A+ A- BBB* BBB+ BBB- BB* BB+ BB- CCC* CCC- D}
%
%   n            n is the number of quarters
%
% OUTPUT:
%   PL_Exp       PL_Exp is matrix vector {v x 1} corresponding to the Expected Portfolio Loss with simple rebalancing that is matched to the initial extended portfolio
%
%            star sign * refers to stable rating
%            power sign ^ refers to upgraded rating
%            tilde sign ~ refers to downgraded rating
%
% EDITING INFORMATION:
%   Author: Vivien Djiambou (FEM), 2009
%   Last update: September 8, 2009

v = length(PB);

PB(:,:,1) = PB;
mmu(:,:,1) = mmu;

for i = 2:n+1;
   mmu(:,:,i) = mmu(:,:,1)*PB(:,:,i-1);
   % The transition matrices are considered at the starting of each quarter due to rebalancing
   PB(:,:,i) = PB(:,:,1);
end;

for i = 1:n+1;
   for j = 1:v-1;
      mmur(1,j,i) = mmu(1,j,i)-mmu(1,j,1);
      if mmur < 0
         Buy(1,j,i) = (-1)*mmur(1,j,i)*PbuyExt(1,j);
         Sell(1,j,i) = 0;
      else
         Sell(1,j,i) = mmur(1,j,i)*PsellExt(1,j);
         Buy(1,j,i) = 0;
      end;
      Tol(j,i) = Buy(1,j,i) + Sell(1,j,i);
   end;
   % The expected total profits and losses
   PL_Exp(i,1) = sum(Tol(1:v-1,i));
end;

end
```

Syntax 15. *Compute expected losses with mix extended portfolio rebalancing*

```
function [PL_ExpMix] = MixRebalance(PB,mmu,PbuyExt,PsellExt,n)
```

```
% GOAL:
%   This program returns the Expected Portfolio Loss of the sum of obligors expected losses in each quarter. The portfolio is rebalanced after each quarter,
%   where we match the new extended portfolio with the initial extended portfolio. As rebalancing rule, obligors that have been currently downgraded cannot be sold.
%
% INPUT:
%   PB           PB returns the extended transition matrix {v x v} that is single grouped from highest to default.
%                 Example: {AAA* AAA^ AA* AA^ AA~ A* A^ A~ BBB* BBB^ BBB~ BB* BB^ BB~ CCC* CCC~ D}
%
%   mmu        mmu is matrix vector {1 x N} corresponding to the number of firms within the extended transition matrix
%
%   PbuyExt     PbuyExt is extended price vector {1 x v} with min length of 8 at which banks can purchase on the market.
%                 Example: {AAA* AAA+ AA* AA+ AA- A* A+ A- BBB* BBB+ BBB- BB* BB+ BB- CCC* CCC- D}
%
%   PsellExt    PsellExt is extended price vector {1 x v} with min length of 8 at which banks can sell on the market.
%                Example: {AAA* AAA+ AA* AA+ AA- A* A+ A- BBB* BBB+ BBB- BB* BB+ BB- CCC* CCC- D}
%
%   n            n is the number of quarters
%
% OUTPUT:
%   PL_ExpMix   PL_ExpMix is matrix vector {v x 1} corresponding to the Expected Portfolio Loss with rebalancing where the
%             portfolio should be matched to the initial extended portfolio
%
%             star sign * refers to stable rating
%             power sign ^ refers to upgraded rating
%             tilde sign ~ refers to downgraded rating
%
% EDITING INFORMATION:
%   Author: Vivien Djiambou (FEM), 2009
%   Last update: September 8, 2009

v = length(PB);
l = round((v+4)/3);

PB(:,:,1) = PB;
mmu(1,:,1) = mmu;

for i = 2:n+1;
   mmu(1,:,i) = mmu(1,:,1)*PB(:,:,i-1);
   PB(:,:,i) = PB(:,:,1);
end;

% Sell only stable and currently upgraded loans
for i = 2:n+1;
   PL_ExpMix(i,1) = 0;
   for k = 1:2;
      mmur(1,k,i) = mmu(1,k,i)-mmu(1,k,1);
      if mmur(1,k,i) < 0
         Buy(1,k,i) = mmur(1,k,i)*PbuyExt(1,k);
         Sell(1,k,i) = 0;
      else
         Sell(1,k,i) = mmur(1,k,i)*PsellExt(1,k);
         Buy(1,k,i) = 0;
      end;
      PL_ExpMix(i,1) = PL_ExpMix(i,1)+Buy(1,k,i)+Sell(1,k,i);
   end;
   s = k+1;

   for k = 2:l-2;
      mmur(1,s,i) = mmu(1,s,i)-mmu(1,s,1);
      if mmur(1,s,i) < 0
         Buy(1,s,i) = mmur(1,s,i)*PbuyExt(1,s);
         Sell(1,s,i) = 0;
      else
         Sell(1,s,i) = mmur(1,s,i)*PsellExt(1,s);
         Buy(1,s,i) = 0;
      end;
      PL_ExpMix(i,1) = PL_ExpMix(i,1)+Buy(1,s,i)+Sell(1,s,i);

      mmur(1,s+1,i) = mmu(1,s+1,i)-mmu(1,s+1,1);
```

```
    if mmur(1,s+1,i) < 0
        Buy(1,s+1,i) = mmur(1,s+1,i)*PbuyExt(1,s+1);
        Sell(1,s+1,i) = 0;
    else
        Sell(1,s+1,i) = mmur(1,s+1,i)*PsellExt(1,s+1);
        Buy(1,s+1,i) = 0;
    end;
    PL_ExpMix(i,1) = PL_ExpMix(i,1)+Buy(1,s+1,i)+Sell(1,s+1,i);

    if i == 2
        mmur(1,s+2,i) = mmu(1,s+2,i)-mmu(1,s+2,1);
        if mmur(1,s+2,i) < 0
            Buy(1,s+2,i) = mmur(1,s+2,i)*PbuyExt(1,s+2);
            Sell(1,s+2,i) = 0;
            mmut(1,s+2,i)= 0;
        else
            Buy(1,s+2,i) = 0;
            Sell(1,s+2,i) = 0;
            mmut(1,s+2,i)= mmur(1,s+2,i);
        end;
    else
        mmur(1,s+2,i) = (mmu(1,s+2,i)-mmu(1,s+2,1))+mmut(1,s+2,i-1);
        if mmur(1,s+2,i) < 0
            Buy(1,s+2,i) = mmur(1,s+2,i)*PbuyExt(1,s+2);
            Sell(1,s+2,i) = 0;
            mmut(1,s+2,i) = 0;
        else
            Buy(1,s+2,i) = 0;
            Sell(1,s+2,i) = 0;
            mmut(1,s+2,i) = mmur(1,s+2,i);
        end;
    end;
    PL_ExpMix(i,1) = PL_ExpMix(i,1)+Buy(1,s+2,i)+Sell(1,s+2,i);
    s = s+3;
end;

mmur(1,v-2,i) = mmu(1,v-2,i)-mmu(1,v-2,1);
if mmur(1,v-2,i) < 0
    Buy(1,v-2,i) = mmur(1,v-2,i)*PbuyExt(1,v-2);
    Sell(1,v-2,i) = 0;
else
    Sell(1,v-2,i) = mmur(1,v-2,i)*PsellExt(1,v-2);
    Buy(1,v-2,i) = 0;
end;
PL_ExpMix(i,1) = PL_ExpMix(i,1)+Buy(1,v-2,i)+Sell(1,v-2,i);

if i == 2
    mmur(1,v-1,i) = mmu(1,v-1,i)-mmu(1,v-1,1);
    if mmur(1,v-1,i) < 0
        Buy(1,v-1,i) = mmur(1,v-1,i)*PbuyExt(1,v-1);
        Sell(1,v-1,i) = 0;
        mmut(1,v-1,i)= 0;
    else
        Buy(1,v-1,i) = 0;
        Sell(1,v-1,i) = 0;
        mmut(1,v-1,i)= mmur(1,v-1,i);
    end;
else
    mmur(1,v-1,i) = (mmu(1,v-1,i)-mmu(1,v-1,1))+mmut(1,v-1,i-1);
    if mmur(1,v-1,i) < 0
        Buy(1,v-1,i) = mmur(1,v-1,i)*PbuyExt(1,v-1);
        Sell(1,v-1,i) = 0;
        mmut(1,v-1,i) = 0;
    else
        Buy(1,v-1,i) = 0;
        Sell(1,v-1,i) = 0;
        mmut(1,v-1,i) = mmur(1,v-1,i);
    end;
end;
PL_ExpMix(i,1) = PL_ExpMix(i,1)+Buy(1,v-1,i)+Sell(1,v-1,i);
```

```
end;

end
```

Syntax 16. *Compute the expected losses with mix and simple extended portfolio rebalancing scenarios*

```
function [Pr_min,PL_Exp,PL_ExpMix] = ExpLoss(P,muu,Pbuy,Psell,n)

% GOAL:
%   This program aims to returns the Portfolio Expected Losses in each quarter for simple rebalancing and mix rebalancing rules.
%
% INPUT:
%   P           P is annual transition matrix {M x M} with minimum length of 4. Ratings are ranks from highest credit quality to default.
%               Example: {AAA AA A BBB BB CCC D}
%
%   muu         muu is number of firms {1 x M} rank from the highest to the lowest rating based on the annual transition matrix P
%
%   Pbuy        Pbuy is the price vector {1 x l} with minimum length of 4 at which banks can purchase on the market. Ratings are
%               ranks from highest credit quality to default. Example: {AAA AA A BBB BB CCC D}
%
%   Psell       Psell is the price vector {1 x l} with minimum length of 4 at which banks can sell on the market. Ratings are ranks
%               from highest credit quality to default. Example: {AAA AA A BBB BB CCC D}
%
%   n           n is the number of quarters
%
% OUTPUT:
%   Pr_min      Pr_min returns the lowest negative values in the transition matrix after taking its fraction power (1/n)
%
%   PL_Exp      PL_Exp is matrix vector {v x 1} corresponding to the Expected Portfolio Loss with simple rebalancing where it is
%               matched to the initial extended portfolio
%
%   PL_ExpMix   PL_ExpMix is matrix vector {v x 1} corresponding to the Expected Portfolio Loss with rebalancing where the
%               portfolio should be matched to the initial extended portfolio
%
% EDITING INFORMATION:
%   Author: Vivien Djiambou (FEM), 2009
%   Last update: September 8, 2009

[Pr,Pr_min] = PowerFract(P,n);
[mmu] = MUExt(Pr,muu);
[Bg] = MatrixExt0(Pr);
[PB] = MatrixBlock(Bg);
[PbuyExt,PsellExt] = PricesExt(Pbuy,Psell);
[PL_Exp] = SimpleRebalance(PB,mmu,PbuyExt,PsellExt,n);
PL_Exp0 = PL_Exp;
[PL_ExpMix] = MixRebalance(PB,mmu,PbuyExt,PsellExt,n);
PL_ExpMix0 = PL_ExpMix;

[Pr,Pr_min] = PowerFract(P,n);
[mmu] = MUExt(Pr,muu);
[Bg] = MatrixExt1(Pr);
[PB] = MatrixBlock(Bg);
[PbuyExt,PsellExt] = PricesExt(Pbuy,Psell);
[PL_Exp] = SimpleRebalance(PB,mmu,PbuyExt,PsellExt,n);
PL_Exp1 = PL_Exp;
[PL_ExpMix] = MixRebalance(PB,mmu,PbuyExt,PsellExt,n);
PL_ExpMix1 = PL_ExpMix;

[Pr,Pr_min] = PowerFract(P,n);
[mmu] = MUExt(Pr,muu);
[rhod] = MUFract(mmu);
[Bg] = MatrixExt2(Pr,rhod);
[PB] = MatrixBlock(Bg);
[PbuyExt,PsellExt] = PricesExt(Pbuy,Psell);
[PL_Exp] = SimpleRebalance(PB,mmu,PbuyExt,PsellExt,n);
PL_Exp2 = PL_Exp;
[PL_ExpMix] = MixRebalance(PB,mmu,PbuyExt,PsellExt,n);
PL_ExpMix2 = PL_ExpMix;
```

```
PL_Exp(:,1) = -PL_Exp0;
PL_Exp(:,2) = -PL_Exp1;
PL_Exp(:,3) = -PL_Exp2;

PL_ExpMix(:,1) = -PL_ExpMix0;
PL_ExpMix(:,2) = -PL_ExpMix1;
PL_ExpMix(:,3) = -PL_ExpMix2;

end
```

Syntax 17. *Fractional power transformation of the annual to quarterly transition matrix*

```
function [Pr,Pr_min] = PowerFract(P,n)

% GOAL:
%   This program compute fractional transition matrix from the annual transition matrix. In this example, we compute the quarterly transition matrix with n = 4.
%
% INPUT:
%   P          P is annual transition matrix {M x M} with minimum length of 4. Ratings are ranks from highest credit quality to default.
%              Example: {AAA AA A BBB BB CCC D}
%
%   n          n is the number of quarters
%
% OUTPUT:
%   Pr         Pr is quarterly transition matrix {M x M} with minimum length of 4. Ratings are ranks from highest credit quality to default.
%              Example: {AAA AA A BBB BB CCC D}
%
%   Pr_min     Pr_min is the transition intensity with most negative elements for a quarterly transition matrix.
%
% EDITING INFORMATION:
%   Author: Vivien Djiambou (FEM), 2009
%   Last update: September 8, 2009

l = length(P);

[V,D] = eig(P);
Pr = V*D^(1/n)*inv(V);

% Transform low negative values into zeroes
Pr0 = -10^30;
for j = 1:l;
    for k = 1:l;
        if Pr(j,k)<0
            if Pr(j,k) < Pr0
                Pr0 = Pr(j,k);
            end;
            Pr(j,k) = 0;
        end;
    end;
end;
Pr_min = Pr0;

% Recalibrate rows sum of the transition matrix to 1
for j = 1:l;
    rest = 0;
    rett = 0;
    for k = 1:l;
        if j < k
            rest = rest + Pr(j,k);
        end;
        if j > k
            rett = rett + Pr(j,k);
        end;
    end;
    Pr(j,j) = 1-(rest+rett);
end;
```

```
end
```

Syntax 18. *Compute similar quarterly matrices for the four successive quarters*

```
function [PB] = UnMatrixPowerExt(PB,n)

% GOAL:
%   This program compute extended transition matrices and the number of firms for n successive quarters.
%
% INPUT:
%   PB         PB is the initial extended transition matrix {N x N} with min length of 8.
%              Example: {AAA* AAA+ AA* AA+ AA- A* A+ A- BBB* BBB+ BBB- BB* BB+ BB- CCC* CCC- D}
%
%   n          n is the number of quarters
%
% OUTPUT:
%   PB         PB returns the extended transition matrices {N x N x n} with min length of 8 for n successive quarters.
%              Example: {AAA* AAA+ AA* AA+ AA- A* A+ A- BBB* BBB+ BBB- BB* BB+ BB- CCC* CCC- D}
%
%              star sign * refers to stable rating
%              power sign ^ refers to upgraded rating
%              tilde sign ~ refers to downgraded rating
%
% EDITING INFORMATION:
%   Author: Vivien Djiambou (FEM), 2009
%   Last update: September 8, 2009

PB(:,:,1) = PB;
for i = 2:n+1;
   % The rebalancing at the end of each quarter mean using identical extended transition matrices
   PB(:,:,i) = PB(:,:,1)^(1);
end;

end
```

Syntax 19. *Randomly generate the evolution of the initial portfolio based on the thresholds*

```
function [mmu_new] = UnNewRating(PB,mmu,n,sm)

% GOAL:
%   This program compute simulated number of firms in each rating class of the extended transition matrices for n successive quarters.
%
% INPUT:
%   PB         PB is the initial extended transition matrix {v x v} with minimum length of 8.
%              Example: {AAA* AAA+ AA* AA+ AA- A* A+ A- BBB* BBB+ BBB- BB* BB+ BB- CCC* CCC- D}
%
%   mmu        mmu is the initial extended portfolio distribution {1 x v} with minimum length of 8.
%
%   n          n is the number of years
%
%   sm         sm is the number of simulation runs
%
% OUTPUT:
%   mmu_new    mmu_new return the extended portfolio distribution {1 x v} with minimum length of 8 for the n successive quarters.
%
%              star sign * refers to stable rating
%              power sign ^ refers to upgraded rating
%              tilde sign ~ refers to downgraded rating
%
% EDITING INFORMATION:
%   Author: Vivien Djiambou (FEM), 2009
%   Last update: September 8, 2009


v = length(PB(:,:,1));
```

```
for i = 2:n+1;
    if v ~= length(PB(:,:,i))
        disp('Error: All Extended Transition Matrices must be equal length');
        return,
    end;
end;

% ensure that the random numbers are different from 0 and 1
tt = rand(1);
while tt == 1 || tt == 0
    tt = rand(1);
end;
% common factor
Y = norminv(tt,0,1);

for j = 1:sm;
    mmu_new(v,1,j) = 0;
    for k = 1:v-1;
        % multiply by 100 to reduce round off errors
        mmu_new(k,1,j) = round(100*mmu(1,k));
        for t = 1:mmu_new(k,1,j);
            rating(t,k,1,j) = k;
        end;
    end;
    for i = 2:n+1;
        rett(v,i,j) = 0;
        for k = 1:v-1; % k is the row number
            % correlation for each rating
            rho(k,i,j) = 0.12*(1+exp(-50*PB(k,v,i)));
            for m = 1:v; % compute thresholds for different ratings
                low_limit(k,m,i) = sum(PB(k,1:m-1,i));
                up_limit(k,m,i) = sum(PB(k,1:m,i));
            end;
            for t = 1:mmu_new(k,i-1,j); % the number of companies in rating k at year i
                % ensure that the random numbers are different from 0 and 1
                het = rand(1);
                while het == 1 || het == 0
                    het = rand(1);
                end;
                Zi(t,k,i,j) = norminv(het,0,1); % generate specific factor
                Xi(t,k,i,j) = sqrt(rho(k,i,j))*Y + sqrt(1-rho(k,i,j))*Zi(t,k,i,j);
                for m = 1:v;
                    if Xi(t,k,i,j)>= norminv(low_limit(k,m,i),0,1) && Xi(t,k,i,j)< norminv(up_limit(k,m,i),0,1)
                        rating(t,k,i,j) = m; % Generate new ratings

                        % compute the number of obligors in each rating classes
                        rett(m,i,j) = rett(m,i,j)+1;
                        mmu_new(m,i,j) = rett(m,i,j);
                    end;
                end;
            end;
        end;
    end;
end;

mmu_new = 100*mmu_new; % rescale portfolio to original level

end
```

Syntax 20. *Extension of the prices vectors to fit the extended transition matrices*

```
function [PbuyExt,PsellExt] = UnPricesExt(Pbuy,Psell)

% GOAL:
%   This program extended the different prices matrix to fit the extended transition matrix.
%
% INPUT:
%   Pbuy        Pbuy is the price vector {1 x l} with minimum length of 4 at which banks can purchase on the market. Ratings are ranks from highest credit quality
```

```
%                to default. Example: {AAA AA A BBB BB CCC D}
%
%  Psell      Psell is the price vector {1 x l} with minimum length of 4 at which banks can sell on the market. Ratings are ranks from highest credit quality to default.
%             Example: {AAA AA A BBB BB CCC D}
%
% OUTPUT:
%  PbuyExt    PbuyExt is extended price vector {1 x v} with min length of 8 at which banks can purchase on the market.
%             Example: {AAA* AAA+ AA* AA+ AA- A* A+ A- BBB* BBB+ BBB- BB* BB+ BB- CCC* CCC- D}
%
%  PsellExt   PsellExt is extended price vector {1 x v} with min length of 8 at which banks can sell on the market.
%             Example: {AAA* AAA+ AA* AA+ AA- A* A+ A- BBB* BBB+ BBB- BB* BB+ BB- CCC* CCC- D}
%
%          star sign * refers to stable rating
%          power sign ^ refers to upgraded rating
%          tilde sign ~ refers to downgraded rating
%
% EDITING INFORMATION:
%  Author: Vivien Djiambou (FEM), 2009
%  Last update: September 8, 2009


l = length(Pbuy);
if l ~= length(Psell)
   disp('Error: All Matrices Prices length must be equal');
   return,
end;

for k = 1:2;
   PbuyExt(1,k) = Pbuy(1,1);
   PsellExt(1,k) = Psell(1,1);
end;
s = k+1;
for k = 2:l-2;
   PbuyExt(1,s) = Pbuy(1,k);
   PsellExt(1,s) = Psell(1,k);
   PbuyExt(1,s+1) = Pbuy(1,k);
   PsellExt(1,s+1) = Psell(1,k);
   PbuyExt(1,s+2) = Pbuy(1,k);
   PsellExt(1,s+2) = Psell(1,k);
   s = s+3;
end;
for k = l-1:l-1;
   PbuyExt(1,s) = Pbuy(1,k);
   PsellExt(1,s) = Psell(1,k);
   PbuyExt(1,s+1) = Pbuy(1,k);
   PsellExt(1,s+1) = Psell(1,k);
end;
s = s+2;
PbuyExt(1,s) = Pbuy(1,l);
PsellExt(1,s) = Psell(1,l);

end
```

Syntax 21. *Compute the profits and losses distribution*

```
function [PL] = UnexpMix(mmu_new,PbuyExt,PsellExt,n,sm)

% GOAL:
%  This program aims to compute the profit & loss ditribution that will serve to determine the portfolio value-at-risk. The rebalancing rule
%  concern recently downgraded obligors that cannot to sold in the market.
%
% INPUT:
%  mmu_new    mmu_new return the extended portfolio distribution {1 x v} with minimun length of 8 for the n successive quarters.
%
%  PbuyExt    PbuyExt is extended price vector {1 x v} with min length of 8 at which banks can purchase on the market.
%             Example: {AAA* AAA+ AA* AA+ AA- A* A+ A- BBB* BBB+ BBB- BB* BB+ BB- CCC* CCC- D}
%
```

```
%   PsellExt    PsellExt is extended price vector {1 x v} with min length of 8 at which banks can sell on the market.
%               Example: {AAA* AAA+ AA* AA+ AA- A* A+ A- BBB* BBB+ BBB- BB* BB+ BB- CCC* CCC- D}
%
%   n           n is the number of years
%
%   sm          sm is the number of simulation runs
%
% OUTPUT:
%   PL          PL return the simulated portfolio loss ditributions {n x sm} in after quarter.
%
%               star sign * refers to stable rating
%               power sign ^ refers to upgraded rating
%               tilde sign ~ refers to downgraded rating
%
% EDITING INFORMATION:
%   Author: Vivien Djiambou (FEM), 2009
%   Last update: September 8, 2009

v = length(mmu_new(:,:,1));
% for i = 2:n+1;
%     if v ~= length(mmu_new(:,:,i))
%         disp('Error: All Extended portfolios must be equal length');
%         return,
%     end;
% end;
l = round((v-4)/3);

% Sell only stable and upgraded firms
for j = 1:sm;
    for i = 2:n+1;
        PL(i,j) = 0;
        for k = 1:2;
            mmur(k,i,j) = mmu_new(k,i,j)-mmu_new(k,1,j); % rebalance portfolio to initial level
            if mmur(k,i,j) < 0
                Buy(k,i,j) = mmur(k,i,j)*PbuyExt(1,k);
                Sell(k,i,j) = 0;
            else
                Sell(k,i,j) = mmur(k,i,j)*PsellExt(1,k);
                Buy(k,i,j) = 0;
            end;
            PL(i,j) = PL(i,j)+Buy(k,i,j)+Sell(k,i,j);
        end;
        s = k+1;

        for k = 2:l-2;
            mmur(s,i,j) = mmu_new(s,i,j)-mmu_new(s,1,j); % rebalance portfolio to initial level
            if mmur(s,i,j) < 0
                Buy(s,i,j) = mmur(s,i,j)*PbuyExt(1,s);
                Sell(s,i,j) = 0;
            else
                Sell(s,i,j) = mmur(s,i,j)*PsellExt(1,s);
                Buy(s,i,j) = 0;
            end;
            PL(i,j) = PL(i,j)+Buy(s,i,j)+Sell(s,i,j);

            mmur(s+1,i,j) = mmu_new(s+1,i,j)-mmu_new(s+1,1,j);  % rebalance portfolio to initial level
            if mmur(s+1,i,j) < 0
                Buy(s+1,i,j) = mmur(s+1,i,j)*PbuyExt(1,s+1);
                Sell(s+1,i,j) = 0;
            else
                Sell(s+1,i,j) = mmur(s+1,i,j)*PsellExt(1,s+1);
                Buy(s+1,i,j) = 0;
            end;
            PL(i,j) = PL(i,j)+Buy(s+1,i,j)+Sell(s+1,i,j);

            if i == 2
            mmur(s+2,i,j) = mmu_new(s+2,i,j)-mmu_new(s+2,1,j); % rebalance portfolio to initial level
                if mmur(s+2,i,j) < 0
                    Buy(s+2,i,j) = mmur(s+2,i,j)*PbuyExt(1,s+2);
                    Sell(s+2,i,j) = 0;
```

```
            mmut(s+2,i,j)= 0;
         else
            Buy(s+2,i,j) = 0;
            Sell(s+2,i,j) = 0;
            mmut(s+2,i,j)= mmur(s+2,i,j); % keep downgraded obligors to next period
         end;
      else
         mmur(s+2,i,j) = (mmu_new(s+2,i,j)+mmut(s+2,i-1,j))-mmu_new(s+2,1,j); % rebalance portfolio to initial level
         if mmur(s+2,i,j) < 0
            Buy(s+2,i,j) = mmur(s+2,i,j)*PbuyExt(1,s+2);
            Sell(s+2,i,j) = 0;
            mmut(s+2,i,j) = 0;
         else
            Buy(s+2,i,j) = 0;
            Sell(s+2,i,j) = 0;
            mmut(s+2,i,j) = mmur(s+2,i,j); % keep downgraded obligors to next period
         end;
      end;
      PL(i,j) = PL(i,j)+Buy(s+2,i,j)+Sell(s+2,i,j);
      s = s+3;
   end;

   mmur(v-2,i,j) = mmu_new(v-2,i,j)-mmu_new(v-2,1,j); % rebalance portfolio to initial level
   if mmur(v-2,i,j) < 0
      Buy(v-2,i,j) = mmur(v-2,i,j)*PbuyExt(1,v-2);
      Sell(v-2,i,j) = 0;
   else
      Sell(v-2,i,j) = mmur(v-2,i,j)*PsellExt(1,v-2);
      Buy(v-2,i,j) = 0;
   end;
   PL(i,j) = PL(i,j)+Buy(v-2,i,j)+Sell(v-2,i,j);

   if i == 2
      mmur(v-1,i,j) = mmu_new(v-1,i,j)-mmu_new(v-1,1,j); % rebalance portfolio to initial level
      if mmur(v-1,i,j) < 0
         Buy(v-1,i,j) = mmur(v-1,i,j)*PbuyExt(1,v-1);
         Sell(v-1,i,j) = 0;
         mmut(v-1,i,j)= 0;
      else
         Buy(v-1,i,j) = 0;
         Sell(v-1,i,j) = 0;
         mmut(v-1,i,j)= mmur(v-1,i,j);
      end;
   else
      mmur(v-1,i,j) = (mmu_new(v-1,i,j)+mmut(v-1,i-1,j))-mmu_new(v-1,1,j); % rebalance portfolio to initial level
      if mmur(v-1,i,j) < 0
         Buy(v-1,i,j) = mmur(v-1,i,j)*PbuyExt(1,v-1);
         Sell(v-1,i,j) = 0;
         mmut(v-1,i,j) = 0;
      else
         Buy(v-1,i,j) = 0;
         Sell(v-1,i,j) = 0;
         mmut(v-1,i,j) = mmur(v-1,i,j);
      end;
   end;
   PL(i,j) = PL(i,j)+Buy(v-1,i,j)+Sell(v-1,i,j);
  end;
end;

end
```

Syntax 22. *Graphical representation of the histogram of losses distributions at the different quarters*

```
function [fig] = UnFig(PL,n)

% GOAL:
%   This program aims to provide the graphical representation of the profit & loss ditribution (P\L).
```

```
%
% INPUT:
%   PL          PL return the simulated portfolio loss ditributions {n x sm} in after quarter.
%
%   n           n is the number of years
%
% OUTPUT:
%   fig         fig return the histogram of the simulated portfolio loss ditributions
%
% EDITING INFORMATION:
%   Author: Vivien Djiambou (FEM), 2009
%   Last update: September 8, 2009

% Portfolio distribution
fig1 = figure;
subplot(2,2,1);
histfit(-PL(2,:));
grid on;
xlabel('Profits & Losses');
ylabel('Frequency');

subplot(2,2,2);
histfit(-PL(3,:));
grid on;
xlabel('Profits & Losses');
ylabel('Frequency');

subplot(2,2,3);
histfit(-PL(n,:));
grid on;
xlabel('Profits & Losses');
ylabel('Frequency');

subplot(2,2,4);
histfit(-PL(n+1,:));
grid on;
xlabel('Profits & Losses');
ylabel('Frequency');

fig = fig1;

end
```

Syntax 23. Graphical representation of the downgrade momentum effects on the loss distributions

```
function [fig] = UnFigMomemt(PL0,PL1,PL2,n)

% GOAL:
%   This program aims to provide the graphical representation of the downgrade momentum effects on the loss ditributions (P\L).
%
% INPUT:
%   PL0         PL0 return the simulated portfolio loss ditributions {n x sm} in after quarter based on the benchmark.
%
%   PL1         PL1 return the simulated portfolio loss ditributions {n x sm} in after quarter based on the first model.
%
%   PL2         PL2 return the simulated portfolio loss ditributions {n x sm} in after quarter based on the second model.
%
%   n           n is the number of years
%
% OUTPUT:
%   fig         fig return the histogram of the simulated portfolio downgrade momentum effects on the loss ditributions
%
% EDITING INFORMATION:
%   Author: Vivien Djiambou (FEM), 2009
%   Last update: September 8, 2009

PL10 = PL1-PL0; % Downgrade momemtum effects between the first model and benchmark
```

```
PL20 = PL2-PL0; % Downgrade momemtum effects between the second model and benchmark

% Portfolio distribution
fig1 = figure;
subplot(2,2,1);
histfit(-PL10(2,:));
grid on;
xlabel('Profits & Losses');
ylabel('Frequency');

subplot(2,2,2);
histfit(-PL10(3,:));
grid on;
xlabel('Profits & Losses');
ylabel('Frequency');

subplot(2,2,3);
histfit(-PL10(n,:));
grid on;
xlabel('Profits & Losses');
ylabel('Frequency');

subplot(2,2,4);
histfit(-PL10(n+1,:));
grid on;
xlabel('Profits & Losses');
ylabel('Frequency');

fig2 = figure;
subplot(2,2,1);
histfit(-PL20(2,:));
grid on;
xlabel('Profits & Losses');
ylabel('Frequency');

subplot(2,2,2);
histfit(-PL20(3,:));
grid on;
xlabel('Profits & Losses');
ylabel('Frequency');

subplot(2,2,3);
histfit(-PL20(n,:));
grid on;
xlabel('Profits & Losses');
ylabel('Frequency');

subplot(2,2,4);
histfit(-PL20(n+1,:));
grid on;
xlabel('Profits & Losses');
ylabel('Frequency');

fig = [fig1;fig2];

end
```

Syntax 24. *Graphical representation of the loss distributions and the downgrade momentum effects of the losses*

```
function [Pr_min,fig1,fig2] = UnexpLoss(P,muu,Pbuy,Psell,n,sm)

% GOAL:
%   This program aims to provide an overview with the graphical representation of the Portfolio Expected Losses in each quarter and
%   also of the downgrade momentum effects.
%
% INPUT:
%   P           P is annual transition matrix {M x M} with minimum length of 4. Ratings are ranks from highest credit quality to default.
```

```
%              Example: {AAA AA A BBB BB CCC D}
%
%  muu        muu is number of firms {1 x M} rank from the highest to the lowest rating based on the annual transition matrix P
%
%  Pbuy       Pbuy is the price vector {1 x l} with minimum length of 4 at which banks can purchase on the market. Ratings are
%             ranks from highest credit quality to default. Example: {AAA AA A BBB BB CCC D}
%
%  Psell      Psell is the price vector {1 x l} with minimum length of 4 at which banks can sell on the market. Ratings are ranks
%             from highest credit quality to default. Example: {AAA AA A BBB BB CCC D}
%
%  n          n is the number of quarters
%
%  sm         sm is the number of simulation runs
%
% OUTPUT:
%  Pr_min     Pr_min returns the lowest negative values in the transition matrix after taking its fraction power (1/n)
%
%  fig1       fig1 return the histogram of Portfolio Loss distribution that matches portfolio to the initial extended portfolio
%             based on the rebalancing rule where recently downgrade obligors cannot be sold.
%
%  fig2       fig2 return the histogram of the downgrade momentum effects of the Portfolio Loss distributions that matches portfolio
%             to the initial extended portfolio based on the rebalancing rule where recently downgrade obligors cannot be sold.
%
% EDITING INFORMATION:
%  Author: Vivien Djiambou (FEM), 2009
%  Last update: September 8, 2009

[Pr,Pr_min] = PowerFract(P,n);
[mmu] = MUExt(Pr,muu);
[Bg] = MatrixExt0(Pr);
[PB] = MatrixBlock(Bg);
[PB] = UnMatrixPowerExt(PB,n);
[mmu_new] = UnNewRating(PB,mmu,n,sm);
[PbuyExt,PsellExt] = UnPricesExt(Pbuy,Psell);
[PL] = UnexpMix(mmu_new,PbuyExt,PsellExt,n,sm);
[fig] = UnFig(PL,n);
fig0 = fig;

[Pr,Pr_min] = PowerFract(P,n);
[mmu] = MUExt(Pr,muu);
[Bg] = MatrixExt1(Pr);
[PB] = MatrixBlock(Bg);
[PB] = UnMatrixPowerExt(PB,n);
[mmu_new] = UnNewRating(PB,mmu,n,sm);
[PbuyExt,PsellExt] = UnPricesExt(Pbuy,Psell);
[PL] = UnexpMix(mmu_new,PbuyExt,PsellExt,n,sm);
[fig] = UnFig(PL,n);
PL1 = PL;
fig1 = fig;

[Pr,Pr_min] = PowerFract(P,n);
[mmu] = MUExt(Pr,muu);
[rhod] = MUFract(mmu);
[Bg] = MatrixExt2(Pr,rhod);
[PB] = MatrixBlock(Bg);
[PB] = UnMatrixPowerExt(PB,n);
[mmu_new] = UnNewRating(PB,mmu,n,sm);
[PbuyExt,PsellExt] = UnPricesExt(Pbuy,Psell);
[PL] = UnexpMix(mmu_new,PbuyExt,PsellExt,n,sm);
[fig] = UnFig(PL,n);
PL2 = PL;
fig2 = fig;

fig = [fig0;fig1;fig2];
fig1 = fig;
[fig] = UnFigMomemt(PL0,PL1,PL2,n);
fig2 = fig;

end
```