Alma Mater Studiorum · Università di Bologna

SCUOLA DI SCIENZE

Corso di Laurea Magistrale in Matematica

# On the computation of short-term credit transition matrices with scenario-dependence

Relatore:
Chiar.mo Prof.
Stefano Pagliarani

Presentata da:
Alessandro Borgia

Correlatore:
Gent.mo Dott.
Diego Perrotta

Anno Accademico 2022/23

# Introduction

The computation of short-term credit transition matrices is a problem for which different solution approaches have been proposed over the years ([19], [20]). In this thesis, carried out with the support of Prometeia S.p.A., we give a full overview of these methods with a main focus on the mathematical foundation; moreover, we include a study on the scenario-dependence.

A credit transition matrix (CTM) is a stochastic matrix that represents the probabilities of migration from a rating class to another; it refers to a given time interval and considers a specific class of debtors. Following the path already traced by the literature, we assume that the credit rating migrations can be described through a discrete time-homogeneous Markov model ([20]). Basic concepts of credit risk ([1]) and details about the credit transition matrices and the Markov model ([26]) are in Section 1.1. Many annual CTMs are published by rating agencies ([15]), but in credit risk framework banks usually compute their own matrices that only consider their operations. In both cases, the smallest time interval in which a CTM is available is usually one year but for some financial applications a short-term CTM (e.g., monthly/quarterly CTM) is needed ([19]). Thanks to the time-homogeneity assumption and to the semigroup property, we have that short-term CTMs can be obtained as roots of the annual CTM. However, given a stochastic matrix $A \in \mathbb{S}^{n\times n}$, we need to understand how it is possible to compute $A^{\frac{1}{p}}$, for $p \in \mathbb{N}$. For this reason, in Section 1.2 we discuss functions of matrices ([13]). In this section, we give two equivalent definitions of function of matrix and observe that knowing $\log(A)$ we can compute each $p$-th root as

$A^{\frac{1}{p}} = \exp(\frac{1}{p}\log(A))$.

Once we are able to compute $A^{\frac{1}{p}}$, the problem is not solved at all since some entries could be complex, which is nonsensical from a probabilistic point of view. We solve this problem at the end of Chapter 1: consistently with the literature ([17]), we assume that a credit transition matrix is strongly diagonally dominant (this assumption is the result of empirical observations on annual CTMs published by rating agencies) and thanks to this assumption it is possible to prove that a credit transition matrix always admit a real logarithm and, consequently, a real $p$-th root for each $p \in \mathbb{N}$ ([11]). However, a real root is not sufficient, since we need the transition matrix to be also stochastic. Indeed, the exact root can have some negative entries or, more rarely, a row whose sum is not one. For this reason, we need to find a stochastic matrix $X \in \mathbb{S}^{n\times n}$ that is a good approximation of $A^{\frac{1}{p}}$.

In Chapter 2, we report three mathematical methods and some other widely-spread methods for doing this approximation. Following the general approach in literature, we consider $\|X^p - A\|$ as the measure of the error ([20]). In Section 2.1 we analyse the quasi optimization of the root (QOM) method ([19], [29]), which finds the nearest stochastic matrix to the exact root, i.e. an $X$ such that $\|X - A^{\frac{1}{p}}\|$ is minimized. We provide a lean algorithm with detailed proof, for which some convex optimization results are required. In 2.2, we study the quasi optimization of the generator (QOG) method ([15], [22]). This method looks for a continuous-time Markov chain of the form $X(t) = \exp(tG)$ that fits well with $A$; $G \in \mathbb{M}^{n\times n}$ is called generator and it has the property that $\exp(tG)$ is stochastic for each $t \geq 0$. We prove that most of the CTMs do not admit a generator and then provide an original algorithm, similar to the QOM one, which finds the generator $G$ that minimizes $\|G - \log(A)\|$. In Section 2.3, we discuss the nonlinear optimization approach ([20], [5]), which is the most accurate but it is not given much credit in literature because of its complexity and low efficiency.

In Chapter 3 we provide numerical results. We tested the algorithms also on a credit transition matrix that Prometeia received from an Italian bank: this matrix is much bigger than those published by rating agencies and for this reason does not meet the empirical assumption of strong diagonal dominance.

Chapter 4 is about the inclusion of the scenario-dependence ([27], [4]). In fact, credit transition matrices are estimated on historical data but, if we want to use them to forecast the future migrations, we definitely need to include the macroeconomic forecasts of influencing variables ([24], [16]). In 4.1 we describe a simplified approach for the inclusion of the scenario-dependence, while in 4.2 we propose two strategies for the inclusion of the scenario-dependence in quarterly credit transition matrices.

# Contents

**Introduction** **i**

**1 Financial and mathematical context** **1**

1.1 An introduction to short-term credit transition matrices . . . 1

1.2 Function of matrices . . . . . . . . . . . . . . . . . . . . . . . 5

1.2.1 Jordan decomposition approach . . . . . . . . . . . . . 6

1.2.2 Hermite approach . . . . . . . . . . . . . . . . . . . . . 8

1.2.3 Real roots and logarithm of a CTM . . . . . . . . . . . 12

**2 Stochastic approximation of the p-th root of a credit transition matrix** **15**

2.1 QOM - Quasi optimization of the root matrix . . . . . . . . . 15

2.1.1 Karush - Kuhn - Tucker conditions . . . . . . . . . . . 16

2.1.2 Algorithm . . . . . . . . . . . . . . . . . . . . . . . . . 20

2.2 QOG - Quasi optimization of the generator . . . . . . . . . . . 23

2.2.1 Generator of a Markov chain . . . . . . . . . . . . . . . 23

2.2.2 Algorithm . . . . . . . . . . . . . . . . . . . . . . . . . 27

2.3 BAM - Best approximating matrix . . . . . . . . . . . . . . . 32

2.3.1 Sequential Quadratic Programming . . . . . . . . . . . 33

2.3.2 BAG - Best approximating generator . . . . . . . . . . 35

2.4 Popular methods . . . . . . . . . . . . . . . . . . . . . . . . . 35

2.4.1 Clip method . . . . . . . . . . . . . . . . . . . . . . . . 36

2.4.2 Diagonal Adjustment and Weighted Adjustment . . . . 36

2.5 Example of application of Clip, QOM, QOG . . . . . . . . . . 37

**3 Numerical results 41**

3.1 Accuracy results for standard CTMs . . . . . . . . . . . . . . . 41

3.2 Accuracy results for a non-standard CTM . . . . . . . . . . . . 44

**4 Conditioning quarterly CTMs to a macroeconomic scenario 49**

4.1 Scenario-dependent credit transition matrix . . . . . . . . . . . 49

4.2 Including the macroeconomic scenario in quarterly CTMs . . . 57

**Conclusions and future perspectives 63**

**A Review of Markov chains 65**

**Bibliography 69**

# Glossary and notations

We provide definitions and notations that will be valid throughout the thesis, unless there are duly noted exceptions.

- $\mathbb{M}^{n\times n}$ denotes the set of $n$-dimensional matrices with complex entries;
- $\mathbb{R}_{\geq 0}$ is the set of non-negative real numbers;
- $\mathbb{R}_{<0}$ is the set of negative real numbers;
- $n_{Jor}$ is the number of Jordan blocks of a given matrix;
- $\|\cdot\|$ is the Frobenius norm;
- $x^*$ is the optimal solution of an optimization problem of the type

  $$\min_{x\in D} / \max_{x\in D} f(x),$$

  with $f : D \subseteq \mathbb{R}^n \to \mathbb{R}^n$;
- Bold numbers/letters are vectors where every component is equal to that number/letter. In some cases, the dimension is given as a subscript. For instance:

  $$\mathbf{0} = (0, ..., 0) \in \mathbb{R}^n,$$

  $$\mathbf{1}_n = (1, ..., 1) \in \mathbb{R}^n;$$
- Given $x, y$ in $\mathbb{R}^n$, $x > y$ means $x_i > y_i$ for any $i = 1, ..., n$; the same holds for non-strict inequalities. Wordings like "$x$ is non-negative" mean that $x \geq \mathbf{0}$;

- Given a sequence of square matrices $(B_k)_{k\leq m}$, we denote with $\text{diag}(B_k)_{k\leq m}$ a block diagonal matrix in which the $k$-th block is equal to $B_k$. Please note that the matrices can be of different dimensions;
- $\mathbb{S}^n$ is the probability simplex, namely the set of real vectors with non-negative entries and sum equal to one:

$$\mathbb{S}^n \coloneqq \{x \in \mathbb{R}^n | x^\top \mathbf{1} = 1, x \geq \mathbf{0}\};$$

- $\mathbb{S}^{n\times n}$ is the set of all the stochastic matrices, namely matrices such that each row is in the probability simplex.

# Chapter 1

# Financial and mathematical context

In this chapter we provide some knowledge about credit transition matrices and credit risk. Moreover, we give the definition of function of matrices and provide results about the $p$-th root of a credit transition matrix.

## 1.1 An introduction to short-term credit transition matrices

**Credit risk.** Bank of Italy defines credit risk as "the contingency for the creditor that a financial obligation is not fulfilled either at maturity or subsequently" ([2]). Throughout this thesis, it is possible to simplify the financial aspect by thinking about the creditor as a bank and about obligations as loans to privates.

**Rating systems.** In credit risk management, information about the financial health of the debtors is essential. For this reason, debtors are divided in *rating classes* that, decreasingly from the best one to the worst, represent how likely it is that the debtor will not be able to pay off debts. There are different rating systems with different notations, but the choice is totally irrelevant for the purpose of the thesis, since each class (except for the worst) is treated in the same way and therefore the only important variable is the dimension of the problem.

**Non-performing loans.** We will follow the Italian standards ([3], [1]) that differentiate between:

- *Non-performing loans*, divided into:
  - *Bad loans: "are exposures to debtors that are insolvent or in substantially similar circumstances".* We will consider a specific credit class named DEF3;
  - *Unlikely-to-pay exposures: "aside from those included among bad loans, are those in respect of which banks believe the debtors are unlikely to meet their contractual obligations in full unless action such as the enforcement of guarantees is taken".* We will consider a specific credit class named DEF2;
  - *Overdrawn and/or past-due exposures: "aside from those classified among bad loans and unlikely-to-pay exposures, are those that are overdrawn and/or past-due by more than 90 days and for above a predefined amount".* We will consider a specific credit class named DEF1;
- *Performing loans*: operations not included in the previous classes, that are not showing particular anomalies. These loans will be divided into $n-3$ credit classes called BO1, BO2,...,BO(n-3) in decreasing order of goodness.

In this thesis we will use both acronyms and numbers to refer to rating classes in decreasing order of goodness; in this sense, $\{1, ..., 4, ..., n\}$ correspond to {DEF3,...,BO(n-3),...,BO1}.

**Credit transition matrices.** While performing a credit risk analysis, a creditor needs to make forecasts of the amount of unpaid conceded loans in a given period and for this reason needs to keep track of the migrations from a credit class to another. Let us consider a rating system with $n$ classes; a *credit transition matrix (CTM)* is an $n$-dimensional stochastic matrix $A$ such that the element $a_{ij}$ represents the probability to move from class $i$ to class $j$. We will assume that the stochastic process representing the rating class of a certain debtor is a time-homogeneous Markov chain and for this reason we have that the probability of migration does not depend on the history of the debtor. Therefore, we can describe the entire stochastic process if we know the starting class and the CTM.

**Absorbing DEF3 assumption.** We will assume that the last class is *absorbing* in the sense of Markov chains, i.e. a DEF3 debtor will never pay and has probability 1 of remaining in DEF3. This is a classical assumption, see any of the references about CTMs.

**Applying CTMs to exposures.** We define an exposure as an $n$-dimensional vector in which the $i$-th component represents the amount of loans globally conceded to debtors in the $i$-th credit class:

$$x = (x_{BO1}, ..., x_{DEF1}, x_{DEF2}, x_{DEF3}).$$

Given the actual exposure and the one year CTM, we can provide a forecast of the exposure in 1 year from now by computing the matrix product of the transposed exposure and the CTM. Naming the new exposure $\tilde{x}$ we have that

$$\tilde{x}_i = \sum_{j=1}^{n} x_j p_{ji}.$$

For example, consider a Markov chain with 6 states and transition matrix $A$ in Table (1.1). If a bank has a current exposure of $x =$ (60, 20, 0, 20, 0, 0) we expect that in one year the new exposure will be

$$\tilde{x} = x^{\top} A = (51.2,\ 22.6,\ 9.6,\ 10.8,\ 3.8,\ 2.0).$$

| | BO1 | BO2 | BO3 | DEF1 | DEF2 | DEF3 |
|---|---|---|---|---|---|---|
| BO1 | 0.80 | 0.12 | 0.06 | 0.02 | 0.00 | 0.00 |
| BO2 | 0.14 | 0.63 | 0.10 | 0.08 | 0.04 | 0.01 |
| BO3 | 0.06 | 0.16 | 0.53 | 0.09 | 0.08 | 0.08 |
| DEF1 | 0.02 | 0.14 | 0.20 | 0.40 | 0.15 | 0.09 |
| DEF2 | 0.00 | 0.05 | 0.10 | 0.15 | 0.50 | 0.20 |
| DEF3 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 |

Table 1.1: Example of a CTM with 3 bonis and 3 default states.

**Strong diagonal dominance** There are some properties which are met by most of the credit transition matrices available online. These matrices are also published by rating agencies such as Moody's and Standard & Poor's, and sometimes we will refer to them as standard CTMs.
A complete list of the empirically derived properties can be found in [17] but we are mainly interested in the most popular one: we will assume that a CTM is strongly diagonally dominant, i.e. for each $i, j$ we have that

$$|a_{ii}| > \sum_{j \neq i} |a_{ij}|;$$

we notice that, since $A$ is stochastic, this condition is satisfied if and only if

$$a_{ii} > 0.5.$$

**The importance of the p-th root.** The shortest time interval in which a CTM is available is usually one year; indeed, considering a shorter time interval would lead to a matrix too influenced by fluctuations. In many financial applications a shorter time interval is needed ([19]): for example, if we want to study the future wallet of a bank every three months, we need to know the forecasts of the quarterly probability of migrations. If we assume that the Markov chain is time-homogeneous, i.e. setting the time span the

CTM is always the same, thanks to the semigroup property (Definition 2.2.1) we have that the quarterly CTM is the 4-th root of the annual CTM. In the same way, the monthly matrix is the 12-th root.

**Stochastic approximation.** However, it is not always possible to compute the root of a matrix in the real sense ([20]) and complex numbers are nonsensical from a probabilistic point of view. Furthermore, even if the root is real, we have no guarantee that stochasticity holds since we could have some negative entries and, more rarely, row-sums other than 1. For these reasons, we want to find the stochastic matrix that best approximates the exact root: we will refer to it as the best stochastic approximation of the $p$-th root.

*Remark* 1.1.1. We say that the approximation is stochastic in the sense that we are looking for an approximation in the space of the stochastic matrices: all the algorithms that we are about to see are deterministic.

## 1.2 Function of matrices

We have discussed the $p$-th root of a matrix previously, and we will delve into other matrix functions in subsequent sections. We need to clarify what it means to apply a complex function to a matrix. We follow the approach by [13].

**Polynomial case.** Applying a polynomial function $p$ to a matrix $A \in \mathbb{M}^{n\times n}$ is straightforward: a simple substitution works. In case of a rational polynomial $r = \frac{p}{g}$, if $g(A)$ is invertible we define $r(A) := p(A)g(A)^{-1}$. This approach can also be used for functions with a power series representation with infinite radius of convergence. For example, the exponential

$$\exp(tA) = I + \sum_{n\geq 1} \frac{t^n A^n}{n!},$$

where $t \in \mathbb{R}$.

However, we need a definition that is consistent with this approach but applicable to non-polynomial functions. We see two approaches in the next subsections.

### 1.2.1 Jordan decomposition approach

**Definition 1.2.1.** Consider $f : \mathbb{C} \to \mathbb{C}$ and $A \in \mathbb{M}^{n\times n}$ with $s$ distinct eigenvalues. We say that $f$ is defined on the spectrum of $A$ if:

$$f^{(j)}(\lambda_i),\ j = 0, ..., n_i - 1,\ i = 1, ..., s$$

exists for every $\lambda_i$ eigenvalue of $A$, with $n_i$ representing the dimension of the highest-order Jordan block in which $\lambda_i$ appears.

**Definition 1.2.2** (Matrix function via Jordan canonical form)**.** Consider $f : \mathbb{C} \to \mathbb{C}$ defined on the spectrum of $A \in \mathbb{M}^{n\times n}$. Suppose that $A$ has Jordan canonical form $A = ZJZ^{-1}$, then

$$f(A) := Zf(J)Z^{-1} = Z\,\mathrm{diag}(f(J_k)_{k=1,\dots,n_{Jor}})Z^{-1},$$

where $J_k \in \mathbb{M}^{m_k\times m_k}$ is the Jordan block relative to $\lambda_k$.

$$f\left(J_k\right) := \begin{bmatrix} f\left(\lambda_k\right) & f'\left(\lambda_k\right) & \cdots & \frac{f^{(m_k-1)}(\lambda_k)}{(m_k-1)!} \\ & f\left(\lambda_k\right) & \ddots & \vdots \\ & & \ddots & f'\left(\lambda_k\right) \\ & & & f\left(\lambda_k\right) \end{bmatrix}.$$

*Example* 1.2.1. Consider $f(x) = x^2$ and the matrix

$$A = \begin{bmatrix} 0.5 & 0.5 & 0 \\ 0 & 0.5 & 0.5 \\ 0 & 0 & 1 \end{bmatrix}.$$

$A = ZJZ^{-1}$ with

$$Z = \begin{bmatrix} 1 & -0.5 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & 0 \end{bmatrix},$$

$$J_1 = [1]\,, \qquad J_2 = \begin{bmatrix} 0.5 & 1 \\ 0 & 0.5 \end{bmatrix}, \qquad J = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.5 & 1 \\ 0 & 0 & 0.5 \end{bmatrix},$$

$$f(J_1) = [1]\,, \quad f(J_2) = \begin{bmatrix} 0.25 & 1 \\ 0 & 0.25 \end{bmatrix}, \quad f(J) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.25 & 1 \\ 0 & 0 & 0.25 \end{bmatrix},$$

so we can compute the square of $A$ in the following way:

$$A^2 = Zf(J)Z^{-1} = \begin{bmatrix} 0.25 & 0.5 & 0.25 \\ 0 & 0.25 & 0.75 \\ 0 & 0 & 1 \end{bmatrix},$$

that is exactly the same result obtained with polynomial substitution.

*Remark* 1.2.1. In the case of a multi-branch complex function, it is important to always apply the same branch to different Jordan blocks, otherwise the definition becomes dependent on the choice of the Jordan normal form. We will see an example later in this chapter.

*Remark* 1.2.2. Given a diagonal matrix $D \in \mathbb{M}^{n\times n}$, we have that $f(D)$ is also diagonal with non-negative entries $f(\lambda_i)_{i=1,\dots,n}$. If $A$ is diagonalizable, then $f(A) = Zf(D)Z^{-1}$, so $f(A)$ has the same eigenvectors as $A$ with eigenvalues $f(\lambda_i)_{i=1,\dots,n}$.

**Mathematical motivation behind the definition.** Observe that, writing $J_k = \lambda_k I + N_k$, we have that $N_k$ is composed of zeros except for the superdiagonal of ones:

$$N_k = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & 0 \\ 0 & \cdots & 0 & 0 & 1 \\ 0 & \cdots & 0 & 0 & 0 \end{bmatrix}.$$

Note that $N_k$ is nilpotent of order $m_k$. Considering the formal Taylor series,

$$f(x) = \sum_{n \geq 0} \frac{f^{(n)}(\lambda_k)(x - \lambda_k)^n}{n!},$$

substituting $J_k$ for $x$ we find

$$f(J_k) = \sum_{n=0}^{m_k - 1} \frac{f^{(n)}(\lambda_k)(N_k)^n}{n!},$$

that is the same as Definition 1.2.2.

Now we provide another approach that does not rely on Taylor expansion.

### 1.2.2 Hermite approach

Let us call $\psi_A$ the minimal polynomial of $A \in \mathbb{M}^{n \times n}$; we have that $\psi_A$ divides any polynomial $p$ such that $p(A) = 0$. To verify this, write $p = \psi_A q + r$ with $\deg(r) < \deg(\psi_A)$ and note that $0 = p(A) = r(A)$, so $r$ must be identically null due to the minimality of $\psi_A$. It is a classical result of linear algebra that

$$\psi_A(x) = \prod_{i=1}^{s} (x - \lambda_i)^{n_i}, \tag{1.1}$$

where the notation is the same as in the previous approach.

Recalling that the polynomial functions of $A$ are defined by substitution, we study an interesting property that we want to keep valid for a generic function.

**Lemma 1.2.1.** *Let $p, q$ be polynomials and consider $A \in \mathbb{M}^{n \times n}$. Then $p(A) = q(A)$ if and only if $p^{(j)}(\lambda_i) = q^{(j)}(\lambda_i)$ for $i = 1, ..., s$, $j = 0, ..., n_i - 1$. If the latter is verified, we say that $p$ and $q$ coincide on the spectrum of $A$.*

*Proof.* Let $\psi_A$ be the minimal polynomial of $A$. If $p(A) = q(A)$ we have that $(p-q)(A) = 0$ and so $\psi_A$ divides $p-q$. On the other hand, if $p$ and $q$ coincide on the spectrum of $A$ the same result comes directly from the explicit formula in (1.1). Hence, in both cases $p - q = \psi_A r$ and:

- $p(A) - q(A) = \psi_A(A)r(A) = 0$

- $p^{(j)}(\lambda_i) - q^{(j)}(\lambda_i) = \sum_{k=0}^{j} \binom{j}{k} \psi_A^{(k)}(\lambda_i) r^{(j-k)}(\lambda_i) = 0$
  for $i = 1, ..., s$, $j = 0, ..., n_i - 1$

so the statement holds. □

We have that the value of $p(A)$ is completely defined by the value of $p$ on the spectrum; to maintain this good property valid for a generic $f$, we provide a definition with polynomial interpolation.

**Definition 1.2.3** (Hermite interpolating polynomial)**.** Given $f : \mathbb{C} \to \mathbb{C}$ defined on the spectrum of $A \in \mathbb{C}^{n \times n}$, the Hermite interpolating polynomial is the polynomial $p$ such that:

- $\deg(p) < \deg(\psi_A)$,

- $p^{(j)}(\lambda_i) = f^{(j)}(\lambda_i)$, $j = 0, ..., n_i - 1$, $i = 1, ..., s$.

Such a polynomial always exists and showing its uniqueness is not a difficult task.

**Existence.** The Hermite interpolating polynomial can be represented with an explicit formula:

$$p(x) = \sum_{i=1}^{s} \left( \left( \sum_{j=0}^{n_i - 1} \frac{1}{j!} \phi_i^{(j)}(\lambda_i)(x - \lambda_i)^j) \right) \prod_{j \neq i} (x - \lambda_j)^{n_j} \right),$$

where $\phi_i(x) = \frac{f(x)}{\prod_{j \neq i}(x - \lambda_j)^{n_j}}$. For a matrix with distinct eigenvalues this formula reduces to the Lagrange interpolating polynomial:

$$p(x) = \sum_{i=1}^{n} \left( f(\lambda_i) \prod_{j=1, j \neq i}^{n} \frac{x - \lambda_j}{\lambda_i - \lambda_j} \right).$$

**Uniqueness.** Let us consider two polynomials $p$ and $q$ respecting the interpolation conditions. We consider $d = p - q$ which is a polynomial of order lower than $\max\{\deg(p), \deg(q)\}$. Note that each eigenvalue is a root of $d$ with multiplicity $n_i$, so we have that $d$ has to be exactly zero because

$$\deg(d) < \max\{\deg(p), \deg(q)\} < \deg(\psi_A) = \sum_{i=1}^{s} n_i.$$

*Example* 1.2.2. Let us solve the problem in the Example 1 with the Hermite interpolation approach. We divide the sum in two components, one for each eigenvalue:

$$p(x) = p_1(x) + p_{0.5}(x).$$

For what concerns the first one,

$$\phi_1(x) = \frac{x^2}{(x-0.5)^2}$$

and so

$$p_1(x) = 4(x-0.5)^2.$$

For the second we also need the derivative:

$$\phi_{0.5}(x) = \frac{x^2}{(x-1)}, \qquad \phi'_{0.5}(x) = \frac{x^2-2x}{(x-1)^2},$$

from which

$$p_{0.5}(x) = (-0.5 - 3(x-0.5))(x-1).$$

Then we have that

$$p(x) = 4(x-0.5)^2 - 0.5(x-1) - 3(x-1)(x-0.5)$$

and substituting $x$ with $A$ we find the same result of Example 1.2.1.

*Remark* 1.2.3. If $f$ is given by a convergent power series, this definition yields that $f(A)$ can be written as a polynomial in $A$ of degree at most $n-1$. The same result can be obtained as a consequence of the Cayley-Hamilton theorem.

**Theorem 1.2.1** (Equivalence of definitions). *The definitions of function of matrices obtained by the Jordan canonical form and the Hermite interpolation are equivalent.*

*Proof.* Using the definition obtained with Hermite interpolation, for a certain polynomial depending on the matrix $A$ it holds that $f(A) = p_A(A)$. Applying simple properties of matrix polynomials, we find that

$$f(A) = p_A(A) = p_A(ZJZ^{-1}) = Zp_A(J)Z^{-1} = Z\,\mathrm{diag}(p_A(J_k)_{k=1,\dots,n_{jor}})Z^{-1}.$$

We can actually derive the expression for $f(J_k)$ in Definition 1.2.2 from Definition 1.2.3, observing that since $J_k$ is a Jordan block it has only one eigenvalue $\lambda_k$ with algebraic multiplicity equal to the dimension of the block. The properties of $N_k$ do the rest and we find $p_{J_k}$ such that $f(J_k) = p_{J_k}(J_k)$. Notice that $p_{J_k}$ and $p_A$ coincide on the spectrum of $J_k$ and so for Lemma 1.2.1 $p_A(J_k) = p_{J_k}(J_k)$, that concludes the proof. □

*Remark* 1.2.4. In general, $p_A$ will be of higher order than $p_{J_k}$; anyway, this does not affect the ability of both to reproduce $f(J_k)$. Similarly, it is always possible to ask for additional conditions when looking for the interpolating polynomial, but asking for additional conditions will lead to an increase of the degree of the polynomial.

**Non-primary matrix functions.** The matrix functions obtained with the previous definitions are called *primary matrix functions.* We explain the existence of *non-primary matrix functions* through an example. Consider the identity matrix $I \in \mathbb{R}^{2\times 2}$ and $f(x) = \sqrt{x}$: we have that the Hermite interpolation approach finds $I$ and $-I$ as square roots. Applying Definition 1.2.2, we can consider different branches of the complex square root and find other results such as

$$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix},$$

that of course are still valid. We quickly pass to the conclusion that there are infinite possible square roots of the identity matrix by writing the Householder reflections

$$H(\theta) = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ \sin(\theta) & -\cos(\theta) \end{bmatrix}, \ \theta \in [0, 2\pi].$$

All these roots are called non-primary and occur when there is at least one multiple eigenvalue that appears in more than a single Jordan block. Some of these roots can be found from Definition 1.2.2 by applying different branches of the function $f$ to different Jordan blocks referring to the same eigenvalue. We recall that, in this way, $f(A)$ becomes dependent on the choice of the Jordan form and for this reason most of the theoretical results in literature refers to primary matrix functions.

### 1.2.3 Real roots and logarithm of a CTM

Let $A \in \mathbb{S}^{n\times n}$ be a CTM and recall that we assume $A$ to be strongly diagonally dominant. We start providing a classical result about the eigenvalues of a matrix (see [25]).

**Theorem 1.2.2** (Gershgorin circle theorem). *Let $A \in \mathbb{M}^{n\times n}$. For each $i = 1, ..., n$ we define*

$$\mathcal{C}_i = \{z \in \mathbb{C} : |z - a_{ii}| \leq \sum_{j=1, j\neq i}^{n} |a_{ij}|\}.$$

*Let $\Lambda$ be the set of all eigenvalues of $A$. Then, we have that*

$$\Lambda \subseteq \cup_{i=1}^{n} \mathcal{C}_i.$$

**Corollary 1.2.1.** *Let $A \in \mathbb{S}^{n\times n}$ be strongly diagonally dominant. Then we have that* $\mathrm{Re}(\lambda) > 0$ *for every $\lambda$ eigenvalue of $A$. In particular, $A$ is invertible.*

**Theorem 1.2.3.** *Given $A \in \mathbb{C}^{n\times n}$ with no eigenvalue on $\mathbb{R}_{<0}$ and $\alpha \in [-1, 1]$, we have that*

$$\log(A^{\alpha}) = \alpha \log(A).$$

Thanks to this result, whose proof can be found in [13, Theorem 11.2], we find a strong relationship between the logarithm of a matrix and the $p$-th roots. In fact, if we are able to find $\log(A)$, then

$$A^{\frac{1}{p}} = \exp\left(\log\left(A^{\frac{1}{p}}\right)\right) = \exp\left(\frac{1}{p}\log\left(A\right)\right).$$

In particular, if the logarithm of $A$ is real, then also the $p$-th root is real. This is important since all the methods we are about to see need an input matrix whose entries are all real.

Now we present a necessary condition for the logarithm of a matrix to be real. This result is proven in [11] and the assumptions are satisfied by any credit transition matrix.

**Theorem 1.2.4.** *Let $A \in \mathbb{R}^{n\times n}$ be invertible. If $A$ has no eigenvalues on $\mathbb{R}_{<0}$, then $A$ has a real logarithm.*

**Corollary 1.2.2.** *Given $A \in \mathbb{S}^{n\times n}$ that is strongly diagonally dominant, we have that $A$ has a real logarithm. In particular, $A$ admits a real $p$-th root for each $p \in \mathbb{N}$.*

# Chapter 2

# Stochastic approximation of the p-th root of a credit transition matrix

This chapter is about the stochastic approximation of the $p$-th root of a credit transition matrix $A \in \mathbb{S}^{n\times n}$, that is, we want to find a matrix $X \in \mathbb{S}^{n\times n}$ that is a good approximation of the exact root $A^{\frac{1}{p}}$. We will go through three different approaches: QOM method, which works on the exact root; QOG method, which works on the exact logarithm to find a generator for an homogeneous continuous-time Markov chain compatible with $A$; nonlinear optimization methods, in particular BAM, which do not require the computation of the exact root but are computationally expensive. At the end of the chapter we report some popular methods, that are trivially not competitive with QOM, QOG and BAM.

## 2.1 QOM - Quasi optimization of the root matrix

In this section the exact root is assumed to be already computed with one of the approaches seen in Chapter 1: we will focus on finding the stochastic matrix that best approximates the exact root (in case it is not already a transition matrix). Consider the problem

$$\min_{X \in \mathbb{S}^{n \times n}} \frac{1}{2}\|X - A^{\frac{1}{p}}\|^2 \tag{QOM}$$

that can be solved with a row by row approach

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \; & \frac{1}{2}\|x - a\|^2 \\ \text{s.t.} \; & x^\top \mathbf{1} = 1 \\ & x \geq \mathbf{0}, \end{aligned}$$

where $x$ is the row of the matrix $X$ that will be found projecting the row $a$ of the matrix $A$ onto the probability simplex. We will discuss the algorithm proposed in [29] because of its simple proof, for which some convex optimization results are required.

*Remark* 2.1.1. We are looking for the closest vector in the probability simplex, but we are proposing the problem in this form to emphasize the fact that it is a constrained problem in $\mathbb{R}^n$.

Before going into details, let us recall some important results of optimization theory that will be useful to prove the correctness of the algorithm.

### 2.1.1 Karush - Kuhn - Tucker conditions

Consider a more general convex optimization problem

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \; & f(x) \\ \text{s.t.} \; & Cx = b \\ & x \geq \mathbf{0}, \end{aligned} \tag{P}$$

where $C \in \mathbb{R}^{n \times m}$, with $m$ being the number of linear equality constraints. The function $f : \mathbb{R}^n \to \mathbb{R}$ is named *objective* or *target function* and is assumed to be convex and differentiable; moreover, in this case, $\mathbb{R}^n$ is the *search space.*

We define $L(x, \lambda, \beta) \coloneqq f(x) - \lambda^\top(Cx - b) - \beta^\top x$ and $g(\lambda, \beta) \coloneqq \min_{x \in \mathbb{R}^n} L(x, \lambda, \beta)$ and consider the dual problem of (P):

$$\begin{aligned} \max_{(\lambda,\beta) \in \mathbb{R}^m \times \mathbb{R}^n} \; & g(\lambda, \beta) \\ \text{s.t.} \; & \beta \geq \mathbf{0}. \end{aligned} \tag{D}$$

**Definition 2.1.1** (Feasible set)**.** We define the feasible set $\mathcal{F}$ as the set of points in the search space that meet the constraints. In case of problem (P) we have

$$\mathcal{F} \coloneqq \{x \in \mathbb{R}^n \text{ s.t. } Cx = b, x \geq \mathbf{0}\}.$$

**Definition 2.1.2** (Bounded problem)**.** We say that an optimization problem is lower/upper bounded if the objective function $f$ is lower/upper bounded over the feasible set.

**Theorem 2.1.1** (Weak duality)**.** *Supposing that the optimization problem* (P) *is lower bounded, we have that*

$$g(\lambda^*, \beta^*) \leq f(x^*),$$

*that is, the optimal dual solution represents a lower bound for the primal problem. This property is called weak duality and the difference between the two solutions is called duality gap.*

*Proof.*

$$g(\lambda, \beta) = \min_x L(x, \lambda, \beta) \leq L(x^*, \lambda, \beta) = f(x^*) - \lambda^\top(Cx^* - b) - \beta^\top x^* \leq f(x^*),$$

where the last inequality comes from the feasibility of $x^*$. □

*Remark* 2.1.2. The convexity of the function is not necessary for the weak duality.

**Definition 2.1.3** (Strong duality)**.** Consider the optimization problem (P) and its dual (D). We say that strong duality holds if all the following are true:

- (P) admits optimal solution $x^*$ if and only if (D) admits optimal solutions $\lambda^*, \beta^*$,
- $f(x^*) = g(\lambda^*, \beta^*)$ (zero duality gap).

Unlike weak duality, strong duality does not always hold. However, we provide a classical sufficient condition for convex problems proved by [6].

**Theorem 2.1.2** (Slater's condition). *Consider the convex optimization problem* (P). *Then strong duality holds if and only if there exists a strictly feasible* $\overline{x}$*, that is,* $\overline{x}$ *such that* $C\overline{x} = b$ *and* $\overline{x} > \mathbf{0}$.

*Proof.* Without any loss of generality, we assume $C$ to be full rank and the primal problem to be lower bounded. Define the sets

- $\mathcal{A} = \{(u, v, t) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} : \exists x \in \mathbb{R}^n, -x \leq u, b - Cx = v, f(x) \leq t\}$,
- $\mathcal{B} = \{(0, 0, s) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} : s < f(x^*)\}$.

These sets have empty intersection, since the opposite would imply the existence of a point $x'$ such that $x'$ is feasible and $f(x') < f(x^*)$. Moreover, both sets are convex, so we can apply the separating hyperplane theorem (see [6], 2.5.1) which guarantees the existence of a triplet $(\overline{\beta}, \overline{\lambda}, \overline{\mu})$ such that

- $\overline{\beta}^\top u + \overline{\lambda}^\top v + \overline{\mu} t \geq \alpha \ \forall (u, v, t) \in \mathcal{A}$,
- $\overline{\mu} t \leq \alpha \ \forall t$ such that $(0, 0, t) \in \mathcal{B}$,

for some value $\alpha \in \mathbb{R}$. We can suppose $\overline{\beta}$ and $\overline{\mu}$ to be non-negative, otherwise $\alpha$ would not be a lower bound since $t$ and $u$ can become arbitrarily large. Moreover, we observe that $\forall x \in \mathbb{R}^n$ we have that $(-x, b - Cx, f(x)) \in \mathcal{A}$ and so

$$-\overline{\beta}^\top x - \overline{\lambda}^\top (Cx - b) + \overline{\mu} f(x) \geq \alpha \geq \overline{\mu} f(x^*) \ \forall x \in \mathbb{R}^n,$$

where the latter follows from $\overline{\mu} t \leq \alpha \ \forall t < f(x^*)$ (since $(0, 0, t) \in \mathcal{B}$).

If $\overline{\mu} \neq 0$, dividing by $\overline{\mu}$ we get that:

$$L(x, \lambda, \beta) = f(x) - \lambda^\top (Cx - b) - \beta^\top x \geq f(x^*),$$

where $\lambda = \frac{\overline{\lambda}}{\overline{\mu}}, \beta = \frac{\overline{\beta}}{\overline{\mu}}$. In particular this holds for the $x$ that minimizes $L$, which means that $g(\lambda^*, \beta^*) \geq g(\lambda, \beta) \geq f(x^*)$. By weak duality we have $g(\lambda^*, \beta^*) \leq f(x^*)$ and so the equality holds. Now we show that $\overline{\mu}$ cannot be zero. Supposing that $\overline{\mu} = 0$ we find that

$$-\overline{\beta}^\top x - \overline{\lambda}^\top (Cx - b) \geq 0.$$

By assumption $C\overline{x} = b$ and so $-\overline{\beta}^\top \overline{x} \geq 0$, but since $\overline{x}$ is positive and $\overline{\beta}$ is non-negative it has to be $\overline{\beta} = \mathbf{0}$. Then we have that $\forall x \in \mathbb{R}^n$:

$$-\overline{\lambda}^\top (Cx - b) \geq 0.$$

By the separing hyperplane theorem $(\overline{\beta}, \overline{\lambda}, \overline{\mu}) \neq (0, 0, 0)$, that implies $\overline{\lambda} \neq 0$. Then we have that

$$-\overline{\lambda}^\top (C\overline{x} - b) = 0$$

and since $C$ is full rank there is an $x$ such that $-\overline{\lambda}^\top (Cx - b) < 0$, that is absurd. □

**Theorem 2.1.3** (KKT conditions). *Consider the convex optimization problem* (P) *and its dual* (D) *and let us assume that strong duality holds. Then* $x^*, \lambda^*, \beta^*$ *are the optimal solutions, of* (P) *and* (D) *respectively, if and only if they satisfy all the following conditions:*

*1)* $Cx^* = b$, $x^* \geq \mathbf{0}$ *(primal feasibility),*

*2)* $\nabla f(x^*) - C^\top \lambda^* - \beta^* = 0$ *(stationarity condition),*

*3)* $\beta^* \geq \mathbf{0}$ *(dual feasibility),*

*4)* $x_i^* \beta_i^* = 0$ *for* $i = 1, ..., n$, *(complementarity condition).*

*Proof.* Suppose $x^*, \lambda^*, \beta^*$ are optimal solutions for (P) and (D) respectively. For strong duality:

$$\begin{aligned} f(x^*) =& g(\lambda^*, \beta^*) = \min_{x \in \mathbb{R}^n} f(x) - \lambda^{*\top}(Cx - b) - \beta^{*\top} x \\ & \leq f(x^*) - \lambda^{*\top}(Cx^* - b) - \beta^{*\top} x^* \leq f(x^*), \end{aligned} \qquad (*)$$

where the last inequality follows from the feasibility conditions. Then we have that all the inequalities above are equalities and so

$$x^* = \operatorname*{argmin}_{x \in \mathbb{R}^n} f(x) - \lambda^{*\top}(Cx - b) - \beta^{*\top} x,$$

from which stationarity condition holds. Furthermore, from (*) we have that $f(x^*) - \beta^{*\top} x^* = f(x^*)$ and so

$$\beta^{*\top} x^* = 0.$$

Since both have non-negative entries it has to be $x_i^* \beta_i^* = 0$ for $i = 1, ..., n$, i.e. complementarity condition holds.
We now prove sufficiency. From conditions 1) and 3) $x^*, \lambda^*, \beta^*$ are feasible. The convexity of $f$ and the stationarity condition imply that $x^*$ is a global minimum for $L(x, \lambda^*, \beta^*)$. Therefore

$$g(\lambda^*, \beta^*) = f(x^*) - \lambda^{*\top}(Cx^* - b) - \beta^{*\top} x^* = f(x^*),$$

where the last equality follows from 1) and 4). This equality implies the optimality of the points. To see this, suppose by contradiction that $x^*, \lambda^*, \beta^*$ are not optimal and let $\overline{x}, \overline{\lambda}, \overline{\beta}$ be optimal points. Then we have that

$$f(\overline{x}) \leq f(x^*) = g(\lambda^*, \beta^*) \leq g(\overline{\lambda}, \overline{\beta})$$

and $f(\overline{x}) = g(\overline{\lambda}, \overline{\beta})$ for strong duality. This completes the proof. □

*Remark* 2.1.3. The convexity of the objective function was used only for sufficiency; necessity holds in general.

### 2.1.2 Algorithm

We move back to the row by row minimization problem

$$\min_{x \in \mathbb{S}^n} \frac{1}{2} \|x - a\|^2 \quad , \tag{RM}$$

where we recall that $\mathbb{S}^n$ is the probability simplex.
We begin by describing the algorithm and then provide the proof of the correctness.

```
Algorithm 1 Projection onto the probability simplex
Input $a \in \mathbb{R}^n$
  1 Define $u = a$ sorted in descending order
  2 Find $\rho = \max\{i = 1, ..., n : u_i + \frac{1}{i}(1 - \sum_{j=1}^{i} u_j) > 0\}$
  3 Define $\lambda = \frac{1}{\rho}(1 - \sum_{i=1}^{\rho} u_i)$
Output $x \in \mathbb{S}^{n \times n}$ s.t. $x_i = max\{a_i + \lambda, 0\}$ for $i = 1, ..., n$.
```

**Theorem 2.1.4.** *Algorithm 1 finds the optimal solution.*

*Proof.* (RM) is a convex optimization problem and Slater's condition holds trivially, so we can apply the KKT conditions to find the optimal point.

1) $\sum_{i=1}^{n} x_i^* = 1$, $x^* \geq \mathbf{0}$,

2) $x_i^* - a_i - \lambda^* - \beta_i^* = 0$, for $i = 1, ...n$,

3) $\beta^* \geq \mathbf{0}$,

4) $x_i^* \beta_i^* = 0$, for $i = 1, ..., n$,

where $\lambda^*$ is a scalar since we have only one equality constraint.
If $x_i^* = 0$ then $\beta_i^* \geq 0$ and $a_i + \lambda^* = -\beta_i^* \leq 0$.
If $x_i^* > 0$ then $\beta_i^* = 0$ and $x_i^* = a_i + \lambda^* > 0$.
It is clear that $x_i^*$ is zero for the lowest values of $a_i$, so we can suppose $x^*$ and $a$ to be sorted. There is no loss of generality since we can find the optimal point for the sorted vector and then apply the inverse permutation: this will lead to the optimal solution, indeed the KKT conditions still hold. Define $\tilde{\rho}$ as the number of positive entries in the optimal solution, so that

$$x_i^* > 0 \text{ for } i = 1, ..., \tilde{\rho} \text{ and } x_i^* = 0 \text{ for } i = \tilde{\rho} + 1, ..., n.$$

For the primal feasibility we have

$$1 = \sum_{i=1}^{\tilde{\rho}} x_i^* = \sum_{i=1}^{\tilde{\rho}} (a_i + \lambda^*).$$

It follows that

$$\lambda^* = \frac{1}{\tilde{\rho}}(1 - \sum_{i=1}^{\tilde{\rho}} a_i),$$

so we only need to prove that $\tilde{\rho} = \max\{i = 1, ..., n : a_i + \frac{1}{i}(1 - \sum_{j=1}^{i} a_j) > 0\}$. To prove the equality we observe that

$$a_{\tilde{\rho}} + \frac{1}{\tilde{\rho}}(1 - \sum_{j=1}^{\tilde{\rho}} a_j) = a_{\tilde{\rho}} + \lambda^* = x^*_{\tilde{\rho}} > 0$$

and if $i > \tilde{\rho}$

$$\begin{aligned} a_i + \frac{1}{i}(1 - \sum_{j=1}^{i} a_j) &= \frac{1}{i}(ia_i + 1 - \sum_{j=1}^{i} a_j) \\ &= \frac{1}{i}(ia_i + 1 - \sum_{j=1}^{\tilde{\rho}} a_j - \sum_{j=\tilde{\rho}+1}^{i} a_j) \\ &= \frac{1}{i}(ia_i + \tilde{\rho}\lambda^* - \sum_{j=\tilde{\rho}+1}^{i} a_j) \\ &= \frac{1}{i}(\tilde{\rho}(a_i + \lambda^*) + \sum_{j=\tilde{\rho}+1}^{i} (a_i - a_j)) < 0. \end{aligned}$$

□

We have proved that the algorithm finds the optimal solution for (RM), that is a row by row formulation of (QOM). Anyway, it is not trivial that the stochastic matrix closest to the exact root is the best one. Indeed, the most used error measure is $\|X^p - A\|$. For this reason, we provide a relationship between the quantities $\|X - A^{\frac{1}{p}}\|$ and $\|X^p - A\|$ that was found in [20] and holds for credit transition matrices.

**Theorem 2.1.5.** *Consider $A \in \mathbb{M}^{n\times n}$ and suppose that $A$ has no eigenvalues on the closed negative real axis.*
*If $\|X - A^{\frac{1}{p}}\| = \epsilon\|A^{\frac{1}{p}}\|$, then*

$$\|X^p - A\| \leq \|A^{\frac{1}{p}}\|^p((1 + \epsilon)^p - 1).$$

*Proof.* Let $B = A^{\frac{1}{p}}$ and $E = X - A^{\frac{1}{p}}$. We have that

$$X^p = (B+E)^p = B^p + (B^{p-1}E + B^{p-2}EB + ... + EB^{p-1}) + ... + E^p.$$

It follows that:

$$\begin{aligned}\|X^p - B^p\| &\leq p\|B\|^{p-1}\|E\| + \frac{p(p-1)}{2}\|B\|^{p-2}\|E\|^2 + ... + \|E\|^p \\ &= \left(\|B\|^p + p\|B\|^{p-1}\|E\| + ... + \|E\|^p\right) - \|B\|^p \\ &= \|B\|^p\left((1+\epsilon)^p - 1\right),\end{aligned}$$

since $\epsilon = \frac{\|E\|}{\|B\|}$. □

## 2.2 QOG - Quasi optimization of the generator

In this section, we consider an alternative formulation of the problem. We try to find an homogeneous continuous-time Markov chain whose transition matrix at time one is exactly equal to the stochastic matrix $A$ in input. To do that, we introduce the concept of generator, inspect the theoretical background and provide an algorithm similar to Algorithm 1 for finding the generator that best approximate the logarithm of a given stochastic matrix. The theoretical results come mainly from [22].

### 2.2.1 Generator of a Markov chain

Suppose that we want to find best stochastic approximation of the $p$-th root of a stochastic matrix $A \in \mathbb{S}^{n\times n}$ and that we are able to find a matrix $G$ such that $\exp(G) = A$ exactly. In this case we have that $A^{\frac{1}{p}} = \exp(\frac{1}{p}G)$ and so, if this exponential is stochastic, the problem is immediately solved. We will show that $P : \mathbb{R}_{\geq 0} \to \mathbb{R}^{n\times n}$, $P(t) := \exp(tG)$ is an homogeneous continuous-time Markov chain if and only if $G$ is a generator; in case $G$ is not a generator, $\exp(\frac{1}{p}G) = A^{\frac{1}{p}}$ is not stochastic.

**Definition 2.2.1** (Transition semigroup)**.** Given a discrete state space $E \subset \mathbb{R}^n$, we say that $P : \mathbb{R}_{\geq 0} \to E$ is a transition semigroup if, for each $s, t \geq 0$, all the following hold:

1) $P(t)$ is a stochastic matrix,

2) $P(0) = I$,

3) $P(t+s) = P(t)P(s)$.

We refer to 3) as *semigroup property*, also known in Markov chains theory as *Chapman - Kolmogorov identity.*

**Definition 2.2.2** (Generator)**.** A square matrix $G \in \mathbb{R}^{n \times n}$ is said to be a generator if

- $G$ has row-sums constantly equal to zero,
- $g_{ij} \geq 0$ for each $i, j \in E$, $i \neq j$.

**Lemma 2.2.1.** *Consider $A, B \in \mathbb{M}^{n \times n}$. Then we have that*

$$\exp(tA + tB) = \exp(tA)\exp(tB) \textit{ for all } t > 0 \textit{ if and only if } AB = BA.$$

*In particular,* $\exp(tA) = \exp(A)^t$ *for every* $A \in \mathbb{M}^{n \times n}$.

*Proof.* If $AB = BA$ the proof is the same as in the scalar case, since Newton's formula for binomial powers is still valid. Vice versa, writing both members of the equality and asking for $t^2$ coefficients to be equal we find $\frac{(AB+BA)}{2} = AB$, hence the statement. □

We now prove some useful results that hold for the exponential of any square matrix.

**Lemma 2.2.2.** *Consider $Q \in \mathbb{R}^{n \times n}$ and $P : \mathbb{R}_{\geq 0} \to \mathbb{R}^{n \times n}$, $P(t) := \exp(tQ)$. Then $P$ has the following properties:*

1. *$P$ is the unique solution of the Kolmogorov backward Cauchy problem:*

$$\frac{d}{dt}P(t) = QP(t) \qquad\qquad P(0) = I \tag{KB}$$

2. *$P$ is the unique solution of the Kolmogorov forward Cauchy problem:*

$$\frac{d}{dt}P(t) = P(t)Q \qquad\qquad P(0) = I \tag{KF}$$

*3.* $\frac{d^k}{dt^k}P(0) = Q^k$ *for any* $k \in \mathbb{N}$.

*Proof.* The exponential series has infinite radius of convergence, so its derivative has infinite radius of convergence. It is obtained by term-by-term differentiation:

$$P'(t) = \sum_{k=1}^{\infty} \frac{t^{k-1}Q^k}{(k-1)!} = P(t)Q = QP(t),$$

from which we have that $P$ solves Kolmogorov differential systems and *3.* holds. For what concerns the uniqueness, assuming that $M(t)$ is a solution of (KB) we find that

$$\begin{aligned}\frac{d}{dt}\left(M(t)\exp(-tQ)\right) = & \left(\frac{d}{dt}M(t)\right)\exp(-tQ) + M(t)\left(\frac{d}{dt}\exp(-tQ)\right) \\ = & M(t)Q\exp(-tQ) + M(t)(-Q)\exp(-tQ) = 0\end{aligned}$$

and so $M(t)\exp(-tQ)$ is constantly equal to the initial data, that means $M = P$. A similar argument proves (KF). □

**Theorem 2.2.1.** *Consider* $G \in \mathbb{R}^{n\times n}$ *and* $P : \mathbb{R}_{\geq 0} \to \mathbb{R}^{n\times n}$, $P(t) := \exp(tG)$. *Then* $P$ *is a transition semigroup if and only if* $G$ *is a generator.*

*Proof.* First of all we prove that if $P$ is a transition semigroup then $G$ is a generator. We have that for $t \to 0$

$$P(t) = I + tG + O(t^2),$$

and so $p_{ij}(t) \approx tg_{ij}$ if $i \neq j$ and $t$ is small enough. So we have that $p_{ij}(t) \geq 0$ for all $t$ implies that $g_{ij} \geq 0$. About the row-sums, since $\sum_{j=1}^n p_{ij}(t) = 1$ for all $t \geq 0$, then

$$\sum_{j=1}^n g_{ij} = \sum_{j=1}^n \frac{dp_{ij}}{dt}(0) = \frac{d}{dt}\left[\sum_{j=1}^n p_{ij}\right](0) = 0.$$

Now suppose that $G$ is a generator. Observe that from Lemma 2.2.1 we have that $P(t+s) = P(t)P(s)$, from which $P(t) = P(\frac{t}{n})^n$ and $p_{ij}(t) \geq p_{ij}(\frac{t}{n})^n$ for every integer $n$. Since $P$ is continuous, if we fix $t$ we can choose $n(t)$ big enough so that $p_{ij}(\frac{t}{n})^n \geq 0$, i.e. $p_{ij}(t)$ is non-negative for every $t \geq 0$. With the same reasoning we find that $p_{ii}(t) \geq 0$ whatever the sign of $g_{ii}$ is.

For what concerns the row-sums, we use the following lemma.

**Lemma 2.2.3.** *Given $A, B \in \mathbb{R}^{n \times n}$ with row-sums that are constantly equal to $\alpha$ and $\beta$ respectively, we have that:*

- *$A + B$ has row-sums constantly equal to $\alpha + \beta$;*
- *$AB$ has row-sums constantly equal to $\alpha\beta$.*

*Proof.* The first one is trivial, while for the second we have that:

$$\sum_{j=1}^{n}\sum_{k=1}^{n} a_{ik}b_{kj} = \sum_{k=1}^{n} a_{ik}\left(\sum_{j=1}^{n} b_{kj}\right) = \alpha\beta,$$

where the first equality follows from Lemma 2.2.2. □

Then we have that $P(t) = exp(tG) = I + \sum_{n\geq 1}\frac{t^n}{n!}G^n$ has row-sums constantly equal to 1 and so $P$ is a transition semigroup. □

*Example* 2.2.1.

$$G = \begin{bmatrix} -0.3 & 0.3 & 0 \\ 0.4 & -0.6 & 0.2 \\ 0 & 0 & 0 \end{bmatrix}$$

is a generator for the stochastic matrix with absorbing state

$$A = \begin{bmatrix} 0.781 & 0.196 & 0.023 \\ 0.261 & 0.586 & 0.153 \\ 0 & 0 & 1 \end{bmatrix}.$$

Then we have that $P : \mathbb{R}_{\geq 0} \to \mathbb{S}^{n \times n}$, $P := \exp(tG)$ is the transition semigroup of a continuous-time Markov chain such that $P(1) = A$. Finally,

$$\exp\left(\frac{1}{12}G\right) = \begin{bmatrix} 0.976 & 0.024 & 0 \\ 0.032 & 0.952 & 0.016 \\ 0 & 0 & 1 \end{bmatrix}$$

is a stochastic 12-th root of $A$.

### 2.2.2 Algorithm

Let us go back to the original problem; now we know that we are looking for a generator of the given stochastic matrix $A = P(1)$. A matrix $A$ that admits an exact generator is usually called *embeddable* and there are many theoretical results concerning both embeddability and uniqueness of generators, for example in [15] and [26]. We report a necessary condition for the existence of an exact generator that is often violated by credit transition matrices.

In case a generator does not exist, *regularization* of the logarithm of the given stochastic matrix is required, where regularization means finding the generator closest to the exact logarithm.

**Theorem 2.2.2.** *Let $P$ be a transition semigroup which admits a generator $G$. Then for any couple of states $i, j \in E$ we have that if $p_{ij}(\bar{t}) = 0$ for some $\bar{t} > 0$, then $p_{ij}(t) = 0$ for all $t > 0$.*

*Proof.* Notice that $p_{ij}(t)$ is an analytic function and therefore, according to analytic continuation theory, either the function is exactly null or the set of all its zeros is a set of isolated points. For each state $i$, we have that $p_{ii}(t) \to 1$ for $t \to 0$ and so, for each $t > 0$, and in particular for $\bar{t}$ that is the zero of $p_{ij}$, there exists a big $\overline{n}(\bar{t})$ such that $p_{ii}\left(\frac{\bar{t}}{\overline{n}}\right) > 0$. Thanks to the monotonicity of probability measures, we find

$$p_{ij}(t) \geq p_{ii}\left(\frac{t}{n}\right)^{n-1} p_{ij}\left(\frac{t}{n}\right)$$

for every $t \geq 0$, $n \in \mathbb{N}$. If we consider $\bar{t}$ and $n \geq \overline{n}$ it has to be $p_{ij}\left(\frac{\bar{t}}{n}\right) = 0$ (or $\bar{t}$ would not be a zero for $p_{ij}$). In conclusion, we found that $\left(p_{ij}\left(\frac{\bar{t}}{n}\right)\right)_{n \geq \overline{n}}$ is a convergent sequence of points in the set of zeros and so, since $p_{ij}$ is analytical, $p_{ij} \equiv 0$ and thesis holds. □

*Remark* 2.2.1. In most cases, given a credit transition matrix $A$ there is not an embeddable homogeneous continuous-time Markov chain compatible with $A$. In fact, we usually find that the probability that a debtor in the best rating class BO1 migrates to the absorbing default DEF3 is zero, but transition to DEF3 is almost always possible with a path made of more than one time step. This means that there is zero probability to go from BO1 to DEF3 in one year but considering a higher amount of years (i.e., a higher $t$) this transition is not negligible anymore. Then, the necessary condition in Theorem 2.2.2 is not met.

For example, the CTM in Table 1.1 has zero probability to move from BO1 to DEF3, but DEF3 is accessible from BO1 by using the DEF1 state as an intermediate step (BO1 → DEF1 → DEF3).

Motivated by this result, we present a method to regularize the exact logarithm of an input matrix; the idea is similar to Algorithm 1 and the logarithm of the matrix is assumed to be already computed. The problem is

$$\begin{aligned} &\min_{G\in\mathbb{R}^{n\times n}} \frac{1}{2}\|G-\log(A)\|^2 \\ \text{s.t.} &\sum_{j=1}^{n} g_{ij} = 0 \text{ for all } i=1,..,n \\ &g_{ij} \geq 0 \text{ for all } i,j=1,...,n,\ i\neq j. \end{aligned} \tag{QOG}$$

(QOG) can be solved with a row by row approach, but we need to keep track of the non-positive element's index $i$, which represents the position of the row vector in the input matrix.

$$\begin{aligned} &\min_{g\in\mathbb{R}^{n}} \frac{1}{2}\|g-a\|^2 \\ \text{s.t. } &g^\top \mathbf{1} = 0 \\ &g_j \geq 0 \text{ for } j\neq i. \end{aligned} \tag{QOGr}$$

In order to solve this set of problems, we present an original algorithm obtained by following the same procedure exposed for Algorithm 1.

**Algorithm 2** Regularization algorithm

```
Input $a \in \mathbb{R}^n$, $i \in \{1, ...n\}$
1 If $a_i \geq \max\{a_j : j \neq i\}$ output 0
2 Define $u = a$ sorted in descending order except for $u_n = a_i$
3 Find $\rho = \max\{j = 1, ..., n-1 : u_j - \frac{1}{j+1}(\sum_{l=1}^{j} u_l + u_n) > 0\}$
4 Define $\lambda = -\frac{1}{\rho+1}(\sum_{l=1}^{\rho} u_l + u_n)$
Output g s.t. $g_i = a_i + \lambda$, $g_j = max\{a_j + \lambda, 0\}$, for $j = 1, ..., n$, $j \neq i$.
```

**Lemma 2.2.4.** *Consider the problem* (QOGr)*. There is a non-zero optimal solution if and only if* $a_i < \max\{a_j : j \neq i\}$.

*Proof.* Suppose that $a_i < \max\{a_j : j \neq i\}$ and suppose by contradiction that **0** is the only optimal solution. Let us call $\overline{j}$ the smallest index such that

$$a_{\overline{j}} = \max\{a_j : j = 1, ..., n\}.$$

We have that the vector $g$ such that

$$g_j := \begin{cases} 0 & \text{if } j \neq i, \overline{j} \\ a_{\overline{j}} - a_i & \text{if } j = \overline{j} \\ a_i - a_{\overline{j}} & \text{if } j = i \end{cases}$$

is feasible and it is an optimal solution. Indeed

$$\begin{aligned} \|g - a\|^2 &= (\sum_{j \neq i, \overline{j}} |a_j|^2) + |a_{\overline{j}} - a_i - a_{\overline{j}}|^2 + |a_i - a_{\overline{j}} - a_i|^2 \\ &= (\sum_{j \neq i, \overline{j}} |a_j|^2) = \|a\|^2 \end{aligned}$$

and then we have a contradiction.

On the other hand, suppose that $a_i \geq \max\{a_j : j \neq i\}$: we need to prove that **0** is the only optimal solution. Let us observe that **0** is the only point in which $g_i$ can be zero, while moving to another point means having a negative $i$-th component for $g$. Moreover, assume that $i = 1$ to simplify the notation; a generic movement from **0** can be written as

$$\tilde{g} = (-\epsilon, \lambda_2 \epsilon, ..., \lambda_n \epsilon)$$

with $\epsilon > 0$, $\sum_{j=2}^{n} \lambda_j = 1$.
Then we have that

$$\|\tilde{g} - a\|^2 = (a_1 + \epsilon)^2 + \sum_{j=2}^{n} (a_j + \lambda_j \epsilon)^2 = \|a\|^2 + \epsilon^2 + \sum_{j=2}^{n} \lambda_j^2 \epsilon^2 + 2\epsilon(a_1 - \sum_{j=2}^{n} \lambda_j a_j)$$

and this quantity is clearly bigger than $\|a\|^2$, since the first two other addends are positive and the last one (for which the assumption is crucial) is non-negative. □

**Theorem 2.2.3.** *Algorithm 2 finds the optimal solution of problem* (QOGr).

*Proof.* The proof is strongly related to the proof of Algorithm 1. Recall that $i$ represents the index of the element that can be negative in the solution. Let us first assume that $a_i < \max\{a_j : j \neq i\}$ : since (QOGr) is a convex optimization problem and Slater's condition holds trivially, we can apply the KKT conditions to find the optimal point.

1) $\sum_{j=1}^{n} g_j^* = 0$, $g_j^* \geq 0$ for $j \neq i$,

2a) $g_j^* - a_j - \lambda^* - \beta_j^* = 0$ for $j \neq i$,

2b) $g_i^* - a_i - \lambda^* + \beta_i^* = 0$,

3) $\beta^* \geq \mathbf{0}$,

4) $g_i^* \beta_i^* = 0$, for $i = 1, ..., n$,

where $\lambda^*$ is a scalar since we have only one equality constraint.
For $j \neq i$ we have that:

- if $g_j^* = 0$ then $\beta_j^* \geq 0$ and $a_j + \lambda^* = -\beta_j^* \leq 0$,

- if $g_j^* > 0$ then $\beta_j^* = 0$ and $g_j^* = a_j + \lambda^* > 0$.

Otherwise we find that:

- if $g_i^* = 0$ then $\beta_i^* \geq 0$ and $a_i + \lambda^* = \beta_i^* \geq 0$,

- if $g_i^* < 0$ then $\beta_i^* = 0$ and $g_i^* = a_i + \lambda^* < 0$.

It is clear that, except for $j = i$, $g_j^*$ is zero in correspondence of the lowest components of $a$. We therefore suppose that $g^*$ and $a$ are sorted except for the $i$-th element. For simplicity, we insert the $i$-th element at the end of the sorted vector and for the observations in the proof of Algorithm 1, there is no loss of generality.

We define $\tilde{\rho}$ as the number of positive entries in the optimal solution and we observe that $\tilde{\rho} > 0$ thanks to our assumption and Lemma 2.2.4. We have that

$$g_j^* > 0 \text{ for } j = 1, ..., \tilde{\rho} \text{ and } g_j^* = 0 \text{ for } j = \tilde{\rho} + 1, ..., n-1, g_n^* \leq 0.$$

For the primal feasibility we have

$$0 = \sum_{i=1}^{\tilde{\rho}} g_i^* + g_n^* = \sum_{i=1}^{\tilde{\rho}} (a_i + \lambda^*) + (a_n + \lambda^*).$$

It follows that

$$\lambda^* = -\frac{1}{\tilde{\rho}+1}(\sum_{i=1}^{\tilde{\rho}} a_i + a_n)$$

so we only need to prove that $\rho = \tilde{\rho}$, where we recall that $\rho := \max\{j = 1, ..., n-1 : u_j - \frac{1}{j+1}(\sum_{l=1}^{j} u_l + u_n) > 0\}$. Let us first observe that $\rho$ is well-defined thanks to our assumption, in fact $a_1 - \frac{1}{2}(a_1 + a_n) > a_1 - \frac{1}{2}(a_1 + a_1) = 0$ and so the set is not empty. For what concerns the equality:

$$a_{\tilde{\rho}} - \frac{1}{\tilde{\rho}+1}\left(\sum_{k=1}^{\tilde{\rho}} a_k + a_n\right) = a_{\tilde{\rho}} + \lambda^* = g_{\tilde{\rho}}^* > 0$$

and for any $\eta > \tilde{\rho}$:

$$\begin{aligned} a_\eta - \frac{1}{\eta+1}\left(\sum_{k=1}^{\eta} a_k + a_n\right) &= \frac{1}{\eta+1}\left((\eta+1)a_\eta - \sum_{k=1}^{\eta} a_k - a_n\right) \\ &= \frac{1}{\eta+1}\left((\eta+1)a_\eta - \sum_{k=1}^{\tilde{\rho}} a_k - \sum_{k=\tilde{\rho}+1}^{\eta} a_k - a_n\right) \\ &= \frac{1}{\eta+1}\left((\eta+1)a_\eta + (\tilde{\rho}+1)\lambda^* - \sum_{k=\tilde{\rho}+1}^{\eta} a_k\right) \\ &= \frac{1}{\eta+1}\left((\tilde{\rho}+1)(a_\eta+\lambda^*) + \sum_{k=\tilde{\rho}+1}^{\eta} (a_\eta - a_k)\right) \leq 0. \end{aligned}$$

□

We conclude the section providing a relationship between the solution of problem (QOG) and the most common way to compute the error made in the approximation of a generator, that is $\|exp(G) - A\|$. Proof in [9].

**Theorem 2.2.4.** *Consider $A \in \mathbb{M}^{n\times n}$ with no eigenvalues on $\mathbb{R}_{\leq 0}$. Define*

$$\|A\|_s := \{\|Av\|_\infty : v \in \mathbb{R}^n, \|v\|_\infty \leq 1\}.$$

*If $\|G - log(A)\|_s = \epsilon$, then*

$$\|A - \exp(G)\|_s \leq \min\{2, 2\epsilon\}.$$

## 2.3 BAM - Best approximating matrix

In this section, we study a completely different approach that does not require the computation of the exact root. However, the algorithm is iterative, so we need a starting point and we will use the output of one of the previous methods. We consider a nonlinear optimization problem on the space of the stochastic matrices

$$\min_{X\in\mathbb{S}^{n\times n}} \frac{1}{2}\|X^p - A\|^2. \tag{BAM}$$

This problem cannot be solved row by row, but it is possible to apply the sequential quadratic programming (SQP) method, whose theory is extremely complex and will be inspected superficially. The theoretical results mainly come from [30] and [18], which is the official reference in the documentation of the minimize function of the SciPy python library, which will be used in the implementation of the BAM method.

### 2.3.1 Sequential Quadratic Programming

Sequential quadratic programming is a well-known technique used to solve nonlinear constrained optimization problems. In general, the problem can be written as:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \ & f(x) \\ \text{s.t.} \ & Cx = b \\ & x \geq \mathbf{0}, \end{aligned} \tag{NLP}$$

where $b \in \mathbb{R}^n$ and $C \in \mathbb{R}^{m \times n}$ is the linear equality constraint matrix. Moreover, we ask that $f : \mathbb{R}^n \to \mathbb{R}$ is twice continuously differentiable.

SQP algorithm is iterative and starts from a given vector $x_0$ to provide new values following the rule

$$x_{k+1} = x_k + \alpha_k d_k,$$

where $\alpha_k > 0$ and $d_k \in \mathbb{R}^n$ are named *step length* and *search direction.* The step length can be chosen in different ways depending on the problem, while the search direction is the key of the algorithm and is found solving a quadratic sub-problem (from which the name SQP). It is always possible to solve a quadratic problem with a finite amount of iterations ([30]); however, some characteristics of the problem have a big influence on solvers' efficiency. A generic quadratic problem is

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \ & \frac{1}{2} x^\top G x + x^\top c \\ \text{s.t.} \ & Cx = b \\ & Hx = 0, \end{aligned}$$

and we differentiate between *convex problems*, in which the matrix $G$ is positive semidefinite, *strictly convex problems*, where $G$ is positive definite and *non-convex problems*, in which the matrix is indefinite. The sub-problem to be solved has to represent the local characteristics of the main problem and in our case we consider

$$\begin{aligned} \min_{d\in\mathbb{R}^n} \ & \frac{1}{2}d^\top \nabla^2_{xx} f(x_k)d + \nabla f(x_k)d \\ \text{s.t.} \ & Cd + Cx_k = 2b \\ & d + x_k \geq 0. \end{aligned} \tag{QP}$$

In general, (NLP) can be solved with nonlinear constraints and (QP) is our adaptation to (BAM) of the general form that can be found both in [18] and [30]. For the theoretical motivation behind the choice of the sub-problem, see one of the references above.

If $x_k$ is close to the optimal solution then (QP) is convex, since $B_k$ is positive definite; otherwise non-convex techniques need to be used. In our application, the output of QOM should work well thanks to Theorem 2.1.5.

*Remark* 2.3.1. Even if the theoretical results are in $\mathbb{R}^n$, we can apply this method to (BAM), where the feasible set is made of $n \times n$ matrices. We just need to write all the elements of the matrix into a $n^2 \times 1$ column vector $x$ and the target function is still well-defined and differentiable. The constraints are still linear and can be expressed in the following vectorial form:

$$\begin{aligned} x &\geq \mathbf{0}_{n^2} \\ Cx &= \mathbf{1}_n, \end{aligned}$$

where $A$ is a $n \times n^2$ matrix that is basically the identity matrix where each scalar is replaced by the correspondent $n$-dimensional vector:

$$C = \begin{bmatrix} \mathbf{1}_n & \mathbf{0}_n & \mathbf{0}_n & \cdots & \mathbf{0}_n \\ \mathbf{0}_n & \mathbf{1}_n & \mathbf{0}_n & \cdots & \mathbf{0}_n \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0}_n & \mathbf{0}_n & \cdots & \mathbf{0}_n & \mathbf{1}_n \end{bmatrix}.$$

At the end of the process we reshape the optimal vector to obtain the optimal matrix $X$.

**Global optimum** Since the objective function is not convex, we can guarantee the convergence to a local minimum only. To obtain convergence to a global optimum our main attempt was to use stochastic optimization algorithms such as simulated annealing and genetic algorithms. However, the sample space is too big so our attempts have proved inefficient and never able to find a better solution than the output of BAM.

### 2.3.2 BAG - Best approximating generator

We can use the SQP method to find the best approximating generator for a given stochastic matrix. It suffices to change the target function and the constraints and study the problem

$$\begin{aligned} \min_{G \in \mathbb{R}^{n \times n}} & \| \exp(G) - A \| \\ \text{s.t.} & \sum_{j=1}^{n} g_{ij} = 0 \text{ for all } i = 1, .., n \\ & g_{ij} \geq 0 \text{ for all } i, j = 1, ..., n, \ i \neq j. \end{aligned}$$

All the observations about the BAM approach are still valid, and for what concerns the starting point it is possible to use the output of QOG thanks to Theorem 2.2.4.

## 2.4 Popular methods

Despite being less accurate than the previous methods, there are some algorithms that are widely used in applications due to their simplicity. We believe that they're worth a mention, and that is why we present three of them in this section.

### 2.4.1 Clip method

The reference for this method is a Prometeia's internal source. Consider a stochastic matrix $A \in \mathbb{S}^{n\times n}$ and suppose that we want to find a stochastic matrix $X \in \mathbb{S}^{n\times n}$ that approximates $A^{\frac{1}{p}}$. We can use the row by row approach and work individually on a row $a$. Define $a^+$ as the positive part of $a$, which is

$$a_j^+ = max\{0,\, a_j\}, \text{ for } j = 1, ..., n.$$

The output of the clip method is the positive part normalized:

$$x_j = \frac{a_j^+}{\sum_{k=1}^n a_k^+}$$

and it is clearly an element of the probability simplex.

*Remark* 2.4.1. Notice that this method is not applicable to non-positive vectors. However, this is not really a problem in the financial applications, since regularizing a non-positive vector would be senseless (any result would be a poor approximation).

### 2.4.2 Diagonal Adjustment and Weighted Adjustment

Diagonal adjustment (DA) and weighted adjustment (WA) are two methods for the regularization of the logarithm of a credit transition matrix. Recall that $A \in \mathbb{S}^{n\times n}$ is the input matrix and $a$ is a row of $\log(A)$, $i$ is the index of the element that can be negative (that is, we are working on the $i$-th row of the generator). Both the methods work row by row and start setting to zero all the negative components except for the $i$-th; at this point

- in the diagonal adjustment we only modify the $i$-th element so that the sum of the elements is zero,
- in the weighted adjustment all the non-zero elements are modified so that the sum of the elements is zero.

In particular:

**DA** $g_j = \max\{0, a_j\}$ for $j \neq i$ and $g_i = -\sum_{j=1, j\neq i}^{n} g_j$

**WA** $\tilde{g}_j = \max\{0, a_j\}$ for $j \neq i$, $\tilde{g}_i = a_i$

$g_j = \tilde{g}_j - |\tilde{g}_j| \frac{\sum_{j=1}^{n} \tilde{g}_j}{\sum_{j=1}^{n} |\tilde{g}_j|}$ for $j = 1, ..., n$.

For additional results, see [19] and [15].

## 2.5 Example of application of Clip, QOM, QOG

We apply the Clip method to regularize the square root of the CTM in Table 1.1. The exact root is

$$A^{\frac{1}{2}} = \begin{bmatrix} 0.8906 & 0.0692 & 0.0341 & 0.0097 & -0.0027 & -0.0009 \\ 0.0824 & 0.7831 & 0.0591 & 0.0525 & 0.0213 & 0.0016 \\ 0.0321 & 0.1010 & 0.7152 & 0.0593 & 0.0507 & 0.0417 \\ 0.0057 & 0.0880 & 0.1418 & 0.6123 & 0.1077 & 0.1135 \\ -0.0028 & 0.0233 & 0.0588 & 0.1111 & 0.6961 & 0.1135 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Thinking row by row, we need to regularize only the first and the fifth row. For the first row $a^1$ we have that

$$a^{1^+} = (0.8906,\ 0.0692,\ 0.0341,\ 0.0097,\ 0,\ 0)$$

and

$$x^1 = \frac{1}{1.0036} a^{1^+} = (0.8874,\ 0.0689\, 0.0340,\ 0.0097,\ 0,\ 0)$$

that is in the probability simplex. Similarly

$$x^5 = \frac{1}{1.0028} a^{5^+} = (0,\ 0.0232\, 0.0587,\ 0.1108,\ 0.6941,\ 0.1132)$$

and so the proposed solution is

$$X_{Clip} = \begin{bmatrix} 0.8874 & 0.0689 & 0.0340 & 0.0097 & 0 & 0 \\ 0.0824 & 0.7831 & 0.0591 & 0.0525 & 0.0213 & 0.0016 \\ 0.0321 & 0.1010 & 0.7152 & 0.0593 & 0.0507 & 0.0417 \\ 0.0057 & 0.0880 & 0.1418 & 0.6123 & 0.1077 & 0.1135 \\ 0 & 0.0232 & 0.0587 & 0.1108 & 0.6941 & 0.1132 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Now we try to apply the QOM method to regularize the same two rows. First of all, we sort the rows

$$u^1 = (0.8906,\ 0.0692,\ 0.0341,\ 0.0097, -0.0009, -0.0027)$$

$$u^5 = (0.6961,\ 0.1135,\ 0.1111,\ 0.0588, 0.0233, -0.0028)$$

and then we find

$$\rho_1 = \max\{i = 1, ..., 6\ :\ {u^1}_i + \frac{1}{i}(1 - \sum_{j=1}^{i} {u^1}_j) > 0\} = 4$$

$$\rho_5 = \max\{i = 1, ..., 6\ :\ {u^5}_i + \frac{1}{i}(1 - \sum_{j=1}^{i} {u^5}_j) > 0\} = 5.$$

We have that

$$\lambda_1 = \frac{1}{5}(1 - \sum_{j=1}^{5} {u^1}_j) = -0.0009,$$

$$\lambda_5 = \frac{1}{4}(1 - \sum_{j=1}^{4} {u^5}_j) = -0.0007,$$

and so

$$x^1 = (0.8897,\ 0.0683,\ 0.0332,\ 0.0088,\ 0,\ 0),$$

$$x^5 = (0,\ 0.0228,\ 0.0582,\ 0.1105,\ 0.6955,\ 0.1130),$$

from which we get the proposed solution

$$X_{QOM} = \begin{bmatrix} 0.8897 & 0.0683 & 0.0332 & 0.0088 & 0 & 0 \\ 0.0824 & 0.7831 & 0.0591 & 0.0525 & 0.0213 & 0.0016 \\ 0.0321 & 0.1010 & 0.7152 & 0.0593 & 0.0507 & 0.0417 \\ 0.0057 & 0.0880 & 0.1418 & 0.6123 & 0.1077 & 0.1135 \\ 0 & 0.0228 & 0.0582 & 0.1105 & 0.6955 & 0.1130 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

The QOG method needs to be applied to the logarithm of the matrix

$$G = \log(A) = \begin{bmatrix} -0.2408 & 0.1605 & 0.0783 & 0.0181 & -0.0129 & -0.0032 \\ 0.1959 & -0.5150 & 0.1392 & 0.1422 & 0.0431 & -0.0052 \\ 0.0681 & 0.2589 & -0.7030 & 0.1596 & 0.1294 & 0.0870 \\ -0.0030 & 0.2253 & 0.4136 & -1.0371 & 0.3171 & 0.0841 \\ -0.0121 & 0.0361 & 0.1331 & 0.3371 & -0.7561 & 0.2618 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

in particular the regularization is required for rows $g_1$, $g_2$, $g_4$, $g_5$. We do the computation for $g_1$ and present the proposed solution.

We note that the first component (i.e. the one that can be negative) is the minimum of the vector, and therefore we only need to sort $g_1$

$$u^1 = (0.1605,\ 0.0783,\ 0.0181,\ -0.0032,\ -0.0129,\ -0.2408)$$

and compute

$$\rho_1 = \max\{i = 1, ..., 5 : {u^1}_i - \frac{1}{i+1}(\sum_{j=1}^{i} {u^1}_j + {u^1}_6) > 0\} = 3,$$

$$\lambda_1 = -\frac{1}{4}(\sum_{j=1}^{3} {u^1}_j + {u^1}_6) = -0.0040.$$

Finally we find

$$x^1 = (-0.2448,\ 0.1565,\ 0.0743,\ 0.0141,\ 0,\ 0)$$

and doing the computation for the other rows

$$X_{QOG} = \begin{bmatrix} -0.2448 & 0.1565 & 0.0743 & 0.0141 & 0.0000 & 0.0000 \\ 0.1948 & -0.5159 & 0.1381 & 0.1411 & 0.0421 & 0.0000 \\ 0.0681 & 0.2589 & -0.7030 & 0.1596 & 0.1294 & 0.0870 \\ 0.0000 & 0.2247 & 0.4130 & -1.0377 & 0.3165 & 0.0835 \\ 0.0000 & 0.0337 & 0.1307 & 0.3347 & -0.7585 & 0.2594 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \end{bmatrix}.$$

To find the root we compute

$$\exp\left(\frac{1}{2}X_{QOM}\right) = \begin{bmatrix} 0.8887 & 0.0673 & 0.0324 & 0.0085 & 0.0022 & 0.0010 \\ 0.0819 & 0.7825 & 0.0585 & 0.0521 & 0.0211 & 0.0039 \\ 0.0322 & 0.1009 & 0.7151 & 0.0592 & 0.0507 & 0.0418 \\ 0.0071 & 0.0877 & 0.1415 & 0.6120 & 0.1074 & 0.0443 \\ 0.0020 & 0.0225 & 0.0579 & 0.1101 & 0.6951 & 0.1124 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 1.0000 \end{bmatrix}.$$

# Chapter 3

# Numerical results

In this chapter, we present numerical results and comparisons among the methods previously analysed. All the experiments were conducted in Python using NumPy and SciPy python libraries. The clip method is included in the comparison because it is widely used, even if it is poor from a mathematical point of view. Furthermore, we provide results of an application to a realistic non diagonally dominant credit transition matrix.

## 3.1 Accuracy results for standard CTMs

In this section, we consider two different dasasets of randomly generated credit transition matrices. We compute the error made approximating the 12-th root with each of the studied methods using three measures: Frobenius norm of the error, 1-norm of the error and the maximum absolute error on a single element. In the absence of other specifications, we call error the matrix

$$E = X^p - A$$

or its norm, where in this analysis $p = 12$. We observe that with the generators methods this quantity is exactly equal to

$$E_g = \exp(G) - A$$

and does not depend on $p$.

For both BAM and BAG the tolerance was set to $10^{-6}$ and the maximum number of iterations (never reached) at 1000. We recall that in iterative methods, in particular for BAM and BAG, the tolerance is on the step-size: the algorithm stops when the maximum number of iterations is reached or when the step size becomes less than the tolerance ($\|x_{k+1} - x_k\| < tol$). We have used QOM/QOG output as starting point for BAM/BAG: for this reason, we already know that the optimization methods (BAM and BAG) will be more accurate than the projection methods (QOM and QOG) since, in the worst case, they will produce the same result. However, nonlinear optimization is computationally expensive, so we are interested in whether there is an improvement and in the extent of it. Furthermore, we will try to understand if the generators approach is competitive with respect to the exact root approach.

We recall that the results of the clip method are inserted to give an idea of the improvement that the studied methods could provide with respect to the widely spread and most intuitive method; however, clip will never give a better result than QOM since QOM finds the stochastic matrix that best approximates $A^{\frac{1}{p}}$.

**Results for random $8 \times 8$ CTMs** We consider a set composed by 10000 randomly generated CTMs. We can see in Table 3.1 that the mean of all the errors in Frobenius norm is smaller for QOM than for QOG and, furthermore, QOM performed better than QOG for 79.9% of the matrices. The improvement obtained with BAM and BAG is negligible, with a mean percentage improvement of 0.6% and 0.3% and a maximum percentage improvement of 8.22% and 26.46%. For nearly six thousands matrices, BAM produced no improvement at all.

It is interesting to observe that BAM and BAG performed worse in 1-norm; this suggests us that the choice of the measure of the error is not trivial. However, 1-norm cannot be used for nonlinear optimization since, even squared, it is not differentiable.

On the other hand, nonlinear optimization methods required a very small number of iterations (1.06 on average with a maximum of 6 iterations) and so there was no issue with the efficiency.

Clip performance is the worst on average.

| | Clip | QOM | QOG | BAM | BAG |
|---|---|---|---|---|---|
| Frob norm (mean) | 0.002310 | 0.001842 | 0.002162 | 0.001820 | 0.002153 |
| Max on s.e. (mean) | 0.001651 | 0.001525 | 0.001770 | 0.001498 | 0.001758 |
| 1-norm (mean) | 0.005174 | 0.004959 | 0.005976 | 0.005130 | 0.006090 |
| Frob norm (max) | 0.0296132 | 0.0217305 | 0.0239416 | 0.020655 | 0.022526 |

Table 3.1: Errors for 8 × 8 standard CTMs

**Results for $8 \times 8$ matrices with an exact stochastic root.** We consider a set of matrices with an exact stochastic root. Trivially, all the methods that work on the root are able to find the exact root (error below 1e-14) since they have to do nothing besides the computation of the root itself: it would be a nice stability property if also QOG and BAG were able to do this. However, as you can see in Table 3.2 this statement turns out to be false. Notice that doing the same thing with embeddable matrices (i.e. matrices with an exact generator) we would have that all the methods trivially find an exact stochastic root, so QOM meets this stability requirement with respect to embeddable matrices.

| <1e-12 | <1e-8 | <1e-5 | <1e-3 | <1e-2 |
|---|---|---|---|---|
| 23.3% | 23.3% | 24.2% | 95% | 100% |

Table 3.2: Percentage of matrices such that the error of the output of the QOG method is lower than the thresholds.

**Influence of $p$ on the error** We already observed that, using the generators approach, the algorithm (and then the error) does not depend on $p$.

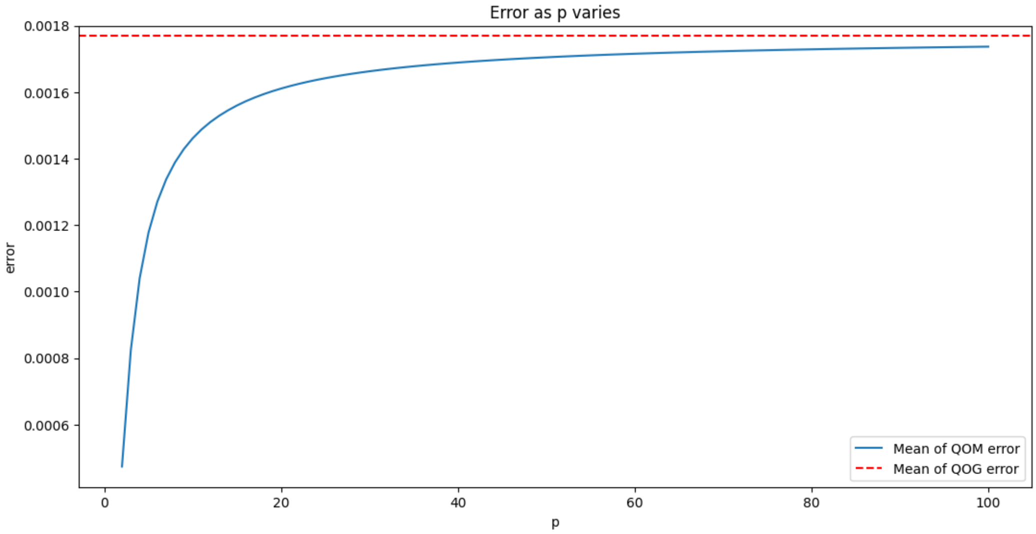

Figure 3.1: Evolution of the error as p increases

However, this is not true for the exact root approach, where the size of $p$ influences the accuracy of the algorithm. In Figure 3.1 there is the result of the application of QOM to 100 randomly generated CTMs for every $p$ from 2 to 100: the mean error of QOM is always increasing and it looks like it is converging from below to the mean error of QOG.

*Remark* 3.1.1. In general, it is not true that QOM always gives a better result than QOG (there are counterexamples in our dataset as underlined before).

## 3.2 Accuracy results for a non-standard CTM

The assumption that a CTM is strongly diagonally dominant is based on empirical evidences, but the empirical studies are made on credit transition matrices published by rating agencies. In practice, banks compute specific credit transition matrices that consider only their own operations. This could lead to CTMs that are far from being strongly diagonally dominant: one main reason could be that the number of rating classes is higher than eight (that is usually considered by the most famous credit agencies) and this leads to a higher number of migrations, which means a weakening of the diagonal. In order to test our algorithms in such situations, we provide results of their application on a realistic $21 \times 21$ credit transition matrix provided to Prometeia by an Italian bank. The presence of such a big number of classes makes the diagonal weaker, in particular the mean of the diagonal elements is 0.4.

*Remark* 3.2.1. QOM, QOG and Clip need a real matrix as input. However, this matrix does not admit a real 12-th root and for this reason we compute the principal branch complex root and then take the real part. In this way, a first error is made yet in the computation of the exact root and in this example

$$\|(\mathrm{Re}(A^{\frac{1}{12}}))^{12} - A\| = 0.0074.$$

**Accuracy results** The results are condensed into Table 3.3. Again, the generators approach performs worse than the exact root approach. The incredible result is in the improvement we got applying the nonlinear optimization methods: Clip, QOM and QOG return matrices that are essentially unusable due to the error, that goes as high as 0.26 on a single element. In our financial application, this means an error of more than 26% on the probability of transition from one rating class to another. On the other hand, BAM has a maximum error on the single element of 0.026. In our financial application, this means an error of 2.6% on the probability of migration from one rating class to another. This error is high, but it may be acceptable if it is located in a non-crucial part of the CTM (this type of analysis depends strictly on the application).

**Efficiency** The biggest concern about the usage of nonlinear optimization techniques is efficiency. In this section, we will not try to compare nonlinear optimization methods with projection methods, since the latter are obviously extremely efficient even in higher dimensions; rather we will try to understand

| Measure | **Clip** | **QOM** | **QOG** | **BAM** | **BAG** |
|---|---|---|---|---|---|
| **Frob. norm** | 0.7200 | 0.6273 | 0.2733 | 0.0867 | 0.1017 |
| **Max err.** | 0.2228 | 0.2598 | 0.0853 | 0.0262 | 0.0319 |
| **1-norm** | 8.1240 | 5.9522 | 2.8434 | 0.9745 | 1.1395 |

Table 3.3: Errors in Frobenius norm for the 21x21 matrix

what is the behavior of BAM and BAG when we decrease the tolerance.

Please notice that the goal of the study is to give a general idea of the decrease in efficiency that we get when performing the nonlinear optimization with a smaller tolerance; we do not provide any time performance since to do that the experiments should be repeated thousands of times and with a bigger dataset of matrices.

**Efficiency results** We provide the results in the form of graphs (Figure 3.2 and 3.3), with $-\log_{10} tol$ on the $x$ axis and error or number of iterations on the $y$ axis. We can see that a decrease in tolerance can imply a small improvement in accuracy but a big decrease in efficiency, and for this reason looking for an optimal tolerance is necessary. The meaning of optimality is strictly dependent on the goal of the application. In the example of CTMs, if we need to compute the monthly matrix for a single scenario we can ask for a very low tolerance, but if we need to compute the monthly matrix for thousands of different macroeconomic scenarios we have to settle for a lower tolerance.

*Remark* 3.2.2. Experiments with different values of $p$ were carried on. The index of the root has a determinant influence on the performances: setting a high value for $p$ makes the generators approach much more competitive. In fact, the errors tend to coincide and $BAG$ becomes more efficient than BAM. This behavior could probably be justified looking at the power series representation of the exponential: with a lower $p$ BAM is more efficient on the single iteration, but with a high value of $p$ we have that the numerical approximation of the exponential requires less computation than $X^p$.

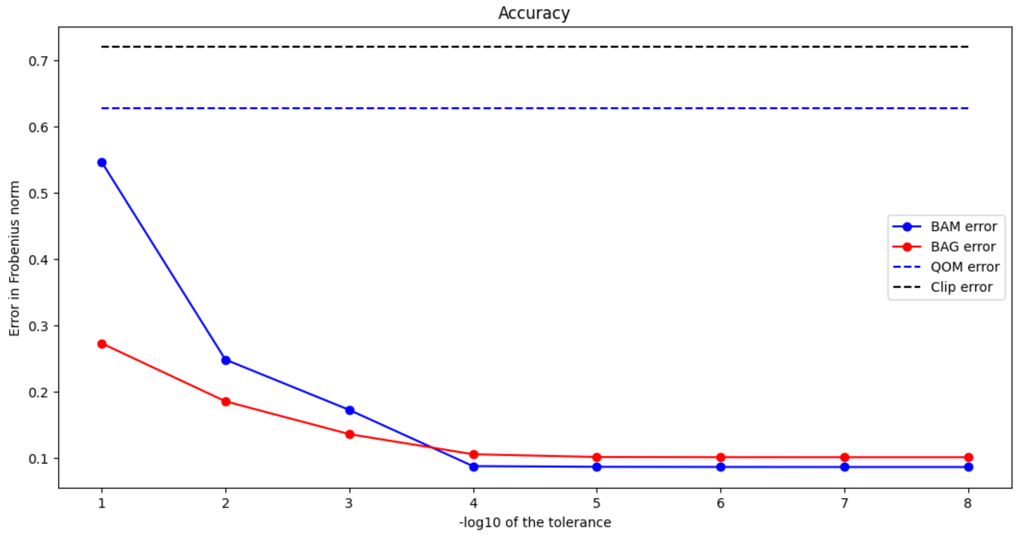

Figure 3.2: Evolution of the error as the tolerance decreases

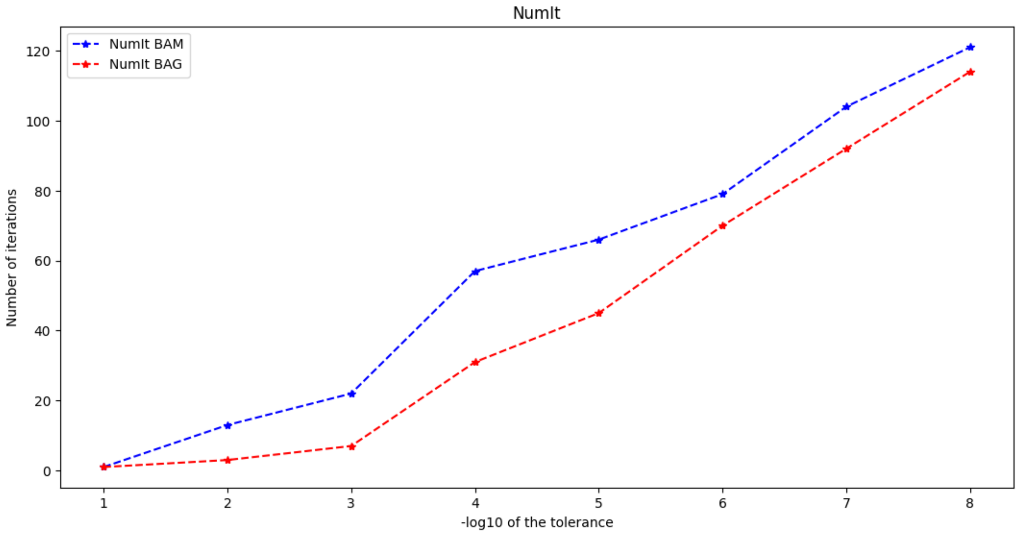

Figure 3.3: Evolution of the number of iterations as the tolerance decreases

# Chapter 4

# Conditioning quarterly CTMs to a macroeconomic scenario

In this chapter we will see two possible techniques to condition a quarterly CTM to a macroeconomic scenario; the quarterly CTM will be obtained by applying some of the techniques seen in the previous chapter. For what concerns the credit risk model, our main references were [27] and [12].

## 4.1 Scenario-dependent credit transition matrix

**The average transition matrix** First of all, we will assume that an average annual credit transition matrix is available. An average CTM represents the estimated transition probabilities in case of a "neutral" macroeconomic scenario; it is usually obtained by applying the *cohort method* to previous years' data.

The cohort method estimates the probability of transition from credit class $i$ to class $j$, $a_{ij}$, with the following formula:

$$a_{ij} = \frac{N_{ij}}{N_i},$$

where $N_{ij}$ represents the amount of debtors migrated from credit class $i$ to $j$ and $N_i$ is the amount of debtors in $i$ at the beginning of the period.

Note that, in general, it can happen that $\sum_{j=1}^{n} N_{ij} \neq N_i$ because some of the debtors might not be rated at the end of the year. In this case, the matrix is not stochastic and needs to be adapted in order to be considered as a transition matrix for a Markov chain model. This adjustment is usually made normalizing the matrix row by row, so that each row-sum is equal to one and the matrix becomes stochastic.

**Creditworthiness of a debtor** Let us start considering only two possible outcomes, default and performing. We name PD the probability of default unconditional on the current rating class. Consider a set of debtors $\mathcal{D}$ and let $\mathcal{K}$ be a partition of $\mathcal{D}$: a set $k \in \mathcal{K}$ is called *cluster* and represents a set of debtors sharing some features; examples of clusters are risk segments, such as private, small business and corporate. This distinction is needed since changes in different macroeconomic variables may influence, for example, small business segment more than corporate segment, or vice versa.

Now consider a random vector $X(t) \in \mathbb{R}^m$ that represents the value of $m$ chosen macroeconomic factors, such as GDP or unemployment rate. This vector is assumed to be normally distributed with mean 0 and covariance matrix $\Omega$. Since the economic effect on the PD is not the same for every cluster, we define the *systematic component of risk* $z_k(t)$ as a function of $X(t)$:

$$z_k(t) := \beta_k^\top X(t) + \eta_k(t),$$

where $\beta_k$ is a vector of cluster-specific weights and $\eta_k(t)$ is a cluster-specific deterministic scalar.

The systematic component of risk is not sufficient since it differentiates only between clusters and for this reason we add an *idiosyncratic component of risk* $\epsilon_d(t)$, $d \in \mathcal{D}$, that includes all the debtor-specific risk and is assumed to be independent to $z_k(t)$. We assume this component to be distributed as a standard normal.

Finally, we define the *obligor-specific creditworthiness* $y_d(t)$ as

$$y_d(t) := \alpha_d z_k(t) + \sqrt{1 - \alpha_d^2}\, \epsilon_d(t),$$

where $\alpha_d$ is the correlation of the single debtor $d \in k$, $k \in \mathcal{K}$, to the systematic component. Notice that $y_d(t)$ has a normal distribution.

**Scenario-dependent PD** Following the approach by [27] we define an obligor-specific threshold $c_d$ such that

$$PD_d(t) = \text{Prob}(y_d(t) \leq c_d),$$

where $PD_d(t)$ is the probability of default in time $t$ of debtor $d \in \mathcal{D}$ (that is, the probability that debtor $d$ is in default at time $t$). To find such thresholds we observe that the default occurs when

$$\epsilon_d(t) \leq \frac{c_d - \alpha_d z_k(t)}{\sqrt{1-\alpha_d^2}}$$

and since we supposed that the idiosyncratic risk factor is distributed as a standard normal we have that

$$PD_d(t) = \text{Prob}\left(\epsilon_d(t) \leq \frac{c_d - \alpha_d z_k(t)}{\sqrt{1-\alpha_d^2}}\right) = \Phi\left(\frac{c_d - \alpha_d z_k(t)}{\sqrt{1-\alpha_d^2}}\right),$$

where $\Phi$ is the cumulative distribution function (CDF) of the standard normal. Applying the quantile function we find that

$$\Phi^{-1}(PD_d(t)) = \frac{c_d - \alpha_d z_k(t)}{\sqrt{1-\alpha_d^2}}$$

and then the explicit formula for $c_d$

$$c_d = \sqrt{1-\alpha_d^2}\,\Phi^{-1}(PD_d(t)) + \alpha_d z_k(t).$$

This formula is extremely useful since $c_d$ is constant over time. For this reason, given $0 \leq t_1 < t_2$, we can obtain $c_d$ as a function of $PD_d(t_1)$ and $z_k(t_1)$ and we can use this expression to compute forward probabilities of default:

$$\begin{aligned} PD_d(t_2) &= \Phi\left(\frac{\sqrt{1-\alpha_d^2}\,\Phi^{-1}(PD_d(t_1)) + \alpha_d z_k(t_1) - \alpha_d z_k(t_2)}{\sqrt{1-\alpha_d^2}}\right) \\ &= \Phi\left(\Phi^{-1}(PD_d(t_1)) - \frac{\alpha_d}{\sqrt{1-\alpha_d^2}}(\Delta z_k(t_1,t_2))\right), \end{aligned}$$

where $\Delta z_k(t_1, t_2) = z_k(t_2) - z_k(t_1)$ represents the variation of systematic risk in the interval $[t_1, t_2]$. Now suppose that at time $t_1$ we want to estimate the probability of default at time $t_2$. Conditioning to the information at time $t_1$, the only random variable left is $z_k(t_2)$. In practice, we can substitute this quantity with an approximation $\tilde{z}_k(t_2)$; as an approximation, we will use the systematic risk factor that is obtained replacing $X(t_2)$ with qualitative forecasts of the macroeconomic factors at time $t_2$. Moreover, we suppose that the correlation $\alpha_d$ does not depend on the debtor but only on the cluster,

$$\alpha_d = \alpha_k \, \forall d \in k,$$

so that we can also define the cluster-dependent *credit index*

$$\Delta m_k(t_2|t_1) := -\frac{\alpha_k}{\sqrt{1-\alpha_k^2}}(\Delta \tilde{z}_k(t_2|t_1)),$$

where $\Delta \tilde{z}_k(t_2|t_1) := \tilde{z}_k(t_2) - z_k(t_1)$. Then, for every cluster $k \in K$ and for every debtor $d \in k$, we have that the probability of default at time $t_2$ conditioning to the information at time $t_1$ is

$$PD_d(t_2|t_1) = \Phi(\Phi^{-1}(PD_d(t_1)) + \Delta m_k(t_2|t_1)),$$

$\forall d \in k$, $\forall k \in \mathcal{K}$, $\forall 0 \leq t_1 < t_2$.

**Scenario-dependent CTM** In the previous paragraph we focused on the PD, but now we want to include the scenario in the full credit transition matrix, that is, we want to compute scenario-dependent migration probabilities.

Given a CTM at time $t_1$, $A(t_1)$, we want to estimate the CTM at time $t_2 > t_1$ conditional to the information at time $t_1$, $A(t_2|t_1)$. This framework works well for any choice of $t_2$ and $t_1$, but in our application it will always be $t_1 = t_2 - \frac{\#months}{12}$ since we are usually interested in credit rating migrations that occur in a fixed number of months.

To begin with, let us simplify the notation assuming that all the debtors in the same cluster share the same CTM. Furthermore, we remove the cluster-dependence in $\alpha$, $z$ and $\Delta m$ (we will not have to consider it again since in our application we will consider one cluster with a CTM that is the same for all debtors in the cluster). Notice that all the computation will be conditional on the starting rating class, that we will call $i$. Mathematically speaking, we are updating $A$ row by row, so we want to write an expression for $a_{ij}(t_2|t_1)$ depending on $a_{ij}(t_1)$ and the credit index $\Delta m(t_2|t_1)$, for every $j = 1, ..., n$. We define $n-1$ thresholds $\{c_{i,0}, ..., c_{i,n-2}\}$ such that:

- $a_{i,n}(t) = \text{Prob}(y_d(t) \leq c_{i,0})$;
- $a_{i,j}(t) = \text{Prob}(c_{i,m-1} \leq y_d(t) \leq c_{i,m})$ for $m = 1, ...n-2$, $j = n - m$;
- $a_{i,1}(t) = \text{Prob}(y_d(t) \geq c_{i,n-2})$;

where $y_d(t)$ is the creditworthiness of debtor $d \in \mathcal{D}$ at time $t$.
Even if $y_d(t)$ is not a standard gaussian, we can rescale it so that the mean is zero and the variance is one; then, we can provide a graphical representation of the rescaled thresholds dividing the area under the standard gaussian in bins. For example, Figure 4.1 represents the thresholds for the third row of Table 1.1, while Figure 4.2 represents the same but for the first row. In this case not all the migrations are possible and for this reason some colors are missing: this is because the area under the curve delimited by the thresholds is zero.

For what concern the absorbing default DEF3 we have that it is verified if

$$\epsilon_d(t) \leq \frac{c_{i,0} - \alpha z(t)}{\sqrt{1-\alpha^2}}$$

and so

$$a_{i,n}(t) = \Phi\left(\frac{c_{i,0} - \alpha z(t)}{\sqrt{1-\alpha^2}}\right).$$

From this equality we find an explicit expression for $c_{i,0}$ depending on $t$ and so, following the same reasoning as before

$$a_{in}(t_2|t_1) = \Phi(\Phi^{-1}(a_{in}(t_1)) + \Delta m(t_2|t_1)).$$

For what concerns BO1, we find the scenario-dependent probability of migration imposing that the row-sum is 1, that is

$$a_{i1}(t_2|t_1) = 1 - \sum_{j=2}^{n} a_{ij}(t_2|t_1).$$

For all the other states we need to compute the probability that $y_d(t)$ is in the area delimited by the respective thresholds. Writing it in terms of $\epsilon_d(t)$ for $t > 0$, $d \in \mathcal{D}$, $m = 1, ..., n-2$ and $j = n - m$, we find

$$a_{ij}(t) = \text{Prob}\left(\frac{c_{i,m-1} - \alpha z(t)}{\sqrt{1-\alpha^2}} \leq \epsilon_d(t) \leq \frac{c_{i,m} - \alpha z(t)}{\sqrt{1-\alpha^2}}\right),$$

that can be seen as the intersection of two events (with implicit dependence on $d$ and $t$)

$$E_1 = \left(\frac{c_{i,m-1} - \alpha z(t)}{\sqrt{1-\alpha^2}} \leq \epsilon_d(t)\right) \qquad E_2 = \left(\epsilon_d(t) \leq \frac{c_{i,m} - \alpha z(t)}{\sqrt{1-\alpha^2}}\right),$$

whose probabilities are

$$P(E_1) = 1 - \Phi\left(\frac{c_{i,m-1} - \alpha z(t)}{\sqrt{1-\alpha^2}}\right) \qquad P(E_2) = \Phi\left(\frac{c_{i,m} - \alpha z(t)}{\sqrt{1-\alpha^2}}\right).$$

Then, observing that $P(E_1 \cup E_2) = 1$ we have that

$$\begin{aligned} a_{ij}(t) &= P(E_1 \cap E_2) = P(E_1) + P(E_2) - P(E_1 \cup E_2) \\ &= \Phi\left(\frac{c_{i,m} - \alpha z(t)}{\sqrt{1-\alpha^2}}\right) - \Phi\left(\frac{c_{i,m-1} - \alpha z(t)}{\sqrt{1-\alpha^2}}\right). \end{aligned}$$

Now we prove by induction that for all $m = 0, ..., n-2$ we have that

$$\Phi\left(\frac{c_{i,m} - \alpha z(t)}{\sqrt{1-\alpha^2}}\right) = \sum_{\gamma=0}^{m} a_{i,n-\gamma}(t).$$

The statement is trivial for $m = 0$ since it is exactly the explicit formula for the probability of absorbing default. Supposing that the statement holds for $m - 1$ we have that

$$\begin{aligned} a_{ij}(t) &= \Phi\left(\frac{c_{i,m} - \alpha z(t)}{\sqrt{1-\alpha^2}}\right) - \Phi\left(\frac{c_{i,m-1} - \alpha z(t)}{\sqrt{1-\alpha^2}}\right) \\ &= \Phi\left(\frac{c_{i,m} - \alpha z(t)}{\sqrt{1-\alpha^2}}\right) - \sum_{\gamma=0}^{m-1} a_{i,n-\gamma}(t), \end{aligned}$$

from which the statement holds since $j = n - m$.

From the equality that we have just proven, we find that

$$c_{i,m} = \sqrt{1-\alpha^2}\,\Phi^{-1}\left(\sum_{\gamma=0}^{m} a_{i,n-\gamma}(t_1)\right) + \alpha z(t_1).$$

Finally, we have that

$$\begin{aligned} a_{ij}(t_2) &= \Phi\left(\frac{c_{i,m} - \alpha z(t_2)}{\sqrt{1-\alpha^2}}\right) - \Phi\left(\frac{c_{i,m-1} - \alpha z(t_2)}{\sqrt{1-\alpha^2}}\right) \\ &= \Phi\left(\frac{\sqrt{1-\alpha^2}\,\Phi^{-1}(\sum_{\gamma=1}^{m-1} a_{i,n-\gamma}(t_1) + \alpha z(t_1) - \alpha z(t_2)}{\sqrt{1-\alpha^2}}\right) - \sum_{\gamma=1}^{m-1} a_{i,n-\gamma}(t_2) \\ &= \Phi\left(\Phi^{-1}\left(\sum_{\gamma=1}^{m} a_{i,n-\gamma}(t_1)\right) - \frac{\alpha}{\sqrt{1-\alpha^2}}\Delta z(t_1,t_2)\right) - \sum_{\gamma=1}^{m-1} a_{i,n-\gamma}(t_2). \end{aligned}$$

Conditioning to the information at time $t_1$ and using the same definition of credit index given for the PD we find that

$$a_{ij}(t_2|t_1) = \Phi\left(\Phi^{-1}\left(\sum_{k=1}^{j} a_{ik}(t_1)\right) + \Delta m(t_2|t_1)\right) - \sum_{k=1}^{j-1} a_{ik}(t_2|t_1),$$

for all $j = 2, ..., n-1$, where $a_{ij}(t_2|t_1)$ is the annual probability of migration from class $i$ to class $j$ at time $t_2$ conditioning to the information at time $t_1$.

*Remark* 4.1.1. As we saw in Figure 4.1 and 4.2, rescaling the normal distribution of the credit worthiness, the distribution of each row of a CTM can be represented as a subdivision of the area under the standard gaussian. If we look at the formulas we found before, upgrading a CTM is equivalent to shifting all the thresholds by the factor $\Delta m$ (Figure 4.3): for this reason, we refer to the scenario-dependent upgrade of a CTM as *shift*.

*Remark* 4.1.2. A positive credit index corresponds to a worsening of the macroeconomic scenario, in the sense that the number of defaults in the cluster increases and the probability of migration to a lower class is higher. Vice versa, a negative credit index (Figure 4.3) corresponds to an improvement of the macroeconomic scenario.

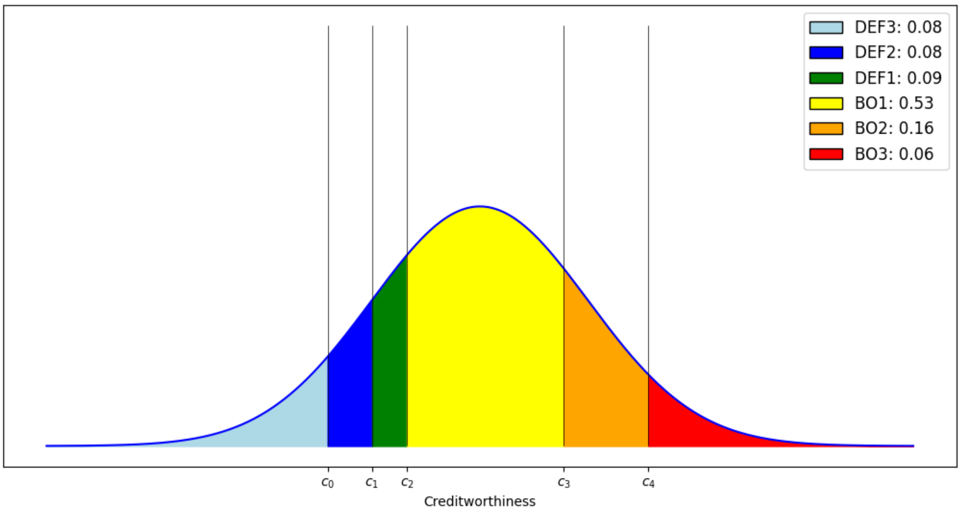

Figure 4.1: pdf of the probability of migration from BO3

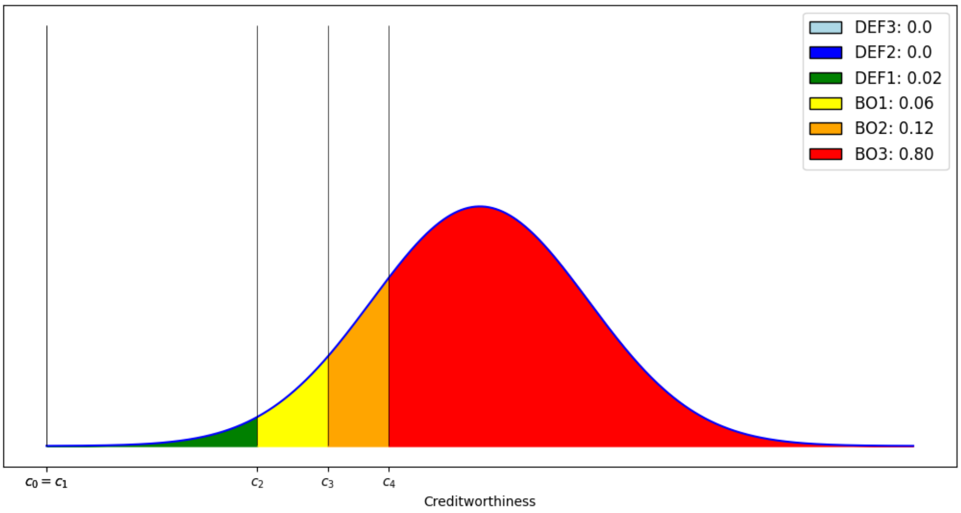

Figure 4.2: pdf of the probability of migration from BO1

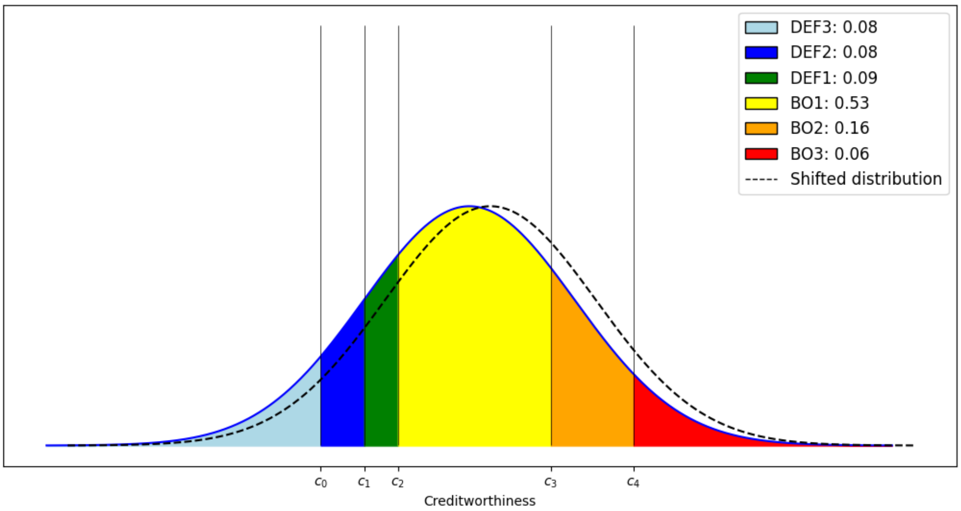

Figure 4.3: shifted distribution of the probability of migration from BO3 with $\Delta m = -0.2$

## 4.2 Including the macroeconomic scenario in quarterly CTMs

In this subsection, we present through an example two possible strategies for obtaining quarterly credit transition matrices that include information about the macroeconomic scenario. Inconsistencies in the computation could be due to the fact that for the sake of illustration all the numbers in the example are rounded to the fourth decimal place, but the computation was done considering all the digits.

**The data** We list the macroeconomic factors considered and provide the sources we used:

- Italian GDP at market value, with historical data from [16] and forecasts from [24];

- Italian GDP with reference year 2015, where historical data are from [16] and the forecasts are from [24];
- Italian unemployment rate, with historical data and forecasts that can be found in [23];
- Euribor 3-month rate, with historical data from [31] and forecasts from [24],

where historical refers to all data prior to 2022 and forecasts means macroeconomic forecasts for years 2023 and 2024. We will focus on one cluster, small business, for which we suppose that the annual credit transition matrix at the end of 2022 is the one in Table 4.1 and we will consider a quarterly time step. Our goal is to estimate the quarterly CTM at 2023Q1, 2023Q2, 2023Q3, 2023Q4, 2024Q1, 2024Q2, 2024Q3, 2024Q4 where Q stands for quarter.

| | BO1 | BO2 | BO3 | DEF1 | DEF2 | DEF3 |
|---|---|---|---|---|---|---|
| BO1 | 0.86 | 0.11 | 0.02 | 0.01 | 0 | 0 |
| BO2 | 0.22 | 0.64 | 0.11 | 0.02 | 0.01 | 0 |
| BO3 | 0.05 | 0.19 | 0.6 | 0.1 | 0.03 | 0.03 |
| DEF1 | 0.02 | 0.05 | 0.15 | 0.53 | 0.15 | 0.1 |
| DEF2 | 0 | 0.03 | 0.05 | 0.08 | 0.71 | 0.13 |
| DEF3 | 0 | 0 | 0 | 0 | 0 | 1 |

Table 4.1: CTM at 2022Q4 for small business

**Credit index** To find the annual credit index at each time step we use a PD satellite model implemented by Prometeia. A PD satellite model is a statistical model that links macroeconomic factor forecasts with the future evolution of the probability of default.

Assuming that the PD at the end of 2022 is 0.0103 we find that the forecasts for the future PDs are:

| **2023Q1** | **2023Q2** | **2023Q3** | **2023Q4** |
|---|---|---|---|
| 0.0125 | 0.0137 | 0.0148 | 0.0160 |

| **2024Q1** | **2024Q2** | **2024Q3** | **2024Q4** |
|---|---|---|---|
| 0.0168 | 0.0187 | 0.0189 | 0.0190 |

Using this data we can find an annual credit index for each time step by applying the following:

$$\Delta m(t_2|t_1) = \Phi^{-1}(PD(t_2)) - \Phi^{-1}(PD(t_1)).$$

**First strategy: homogeneous method** Let's start by applying the above formula to find annual credit indexes for the annual probability of default

$$\Delta m(2023Q4|2022Q4) = 0.1693,$$

$$\Delta m(2024Q4|2023Q4) = 0.0706.$$

We can therefore use these credit indexes to obtain the two CTMs at time 2023Q4 and 2024Q4 (Tables 4.2 and 4.3). We propose to compute the quarterly CTMs by applying the BAM method to find the best stochastic approximation of the fourth root of both matrices; we call this method homogeneous since we have four identical quarterly matrices per year (Tables 4.4 and 4.5).

| | BO1 | BO2 | BO3 | DEF1 | DEF2 | DEF3 |
|---|---|---|---|---|---|---|
| BO1 | 0.8188 | 0.1377 | 0.0280 | 0.0155 | 0 | 0 |
| BO2 | 0.1732 | 0.6456 | 0.1377 | 0.0280 | 0.0155 | 0 |
| BO3 | 0.0348 | 0.1558 | 0.6047 | 0.1217 | 0.0395 | 0.0435 |
| DEF1 | 0.0131 | 0.0369 | 0.1232 | 0.5201 | 0.1737 | 0.1330 |
| DEF2 | 0 | 0.0202 | 0.0375 | 0.0646 | 0.7085 | 0.1693 |
| DEF3 | 0 | 0 | 0 | 0 | 0 | 1 |

Table 4.2: CTM at 2023Q4 for small business

**Second strategy: non-homogeneous method** We can also find quarterly scenario-dependent CTMs by updating the best stochastic approximation of the fourth root of the annual CTM with an appropriate credit index.

| | BO1 | BO2 | BO3 | DEF1 | DEF2 | DEF3 |
|---|---|---|---|---|---|---|
| BO1 | 0.7997 | 0.1499 | 0.0319 | 0.0185 | 0 | 0 |
| BO2 | 0.1557 | 0.6439 | 0.1499 | 0.0319 | 0.0185 | 0 |
| BO3 | 0.0297 | 0.1423 | 0.6027 | 0.1310 | 0.0439 | 0.0504 |
| DEF1 | 0.0109 | 0.0322 | 0.1126 | 0.5123 | 0.1831 | 0.1488 |
| DEF2 | 0 | 0.0170 | 0.0330 | 0.0585 | 0.7038 | 0.1877 |
| DEF3 | 0 | 0 | 0 | 0 | 0 | 1 |

Table 4.3: CTM at 2024Q4 for small business

| | BO1 | BO2 | BO3 | DEF1 | DEF2 | DEF3 |
|---|---|---|---|---|---|---|
| BO1 | 0.9818 | 0.0153 | 0.0015 | 0.0014 | 0 | 0 |
| BO2 | 0.0193 | 0.9602 | 0.0176 | 0.0018 | 0.0012 | 0 |
| BO3 | 0.0018 | 0.0200 | 0.9551 | 0.0172 | 0.0027 | 0.0032 |
| DEF1 | 0.0010 | 0.0028 | 0.0170 | 0.9439 | 0.0228 | 0.0125 |
| DEF2 | 0 | 0.0018 | 0.0037 | 0.0081 | 0.9706 | 0.0158 |
| DEF3 | 0 | 0 | 0 | 0 | 0 | 1.0000 |

Table 4.4: Homogeneous quarterly CTM in 2023 for small business

| | BO1 | BO2 | BO3 | DEF1 | DEF2 | DEF3 |
|---|---|---|---|---|---|---|
| BO1 | 0.9798 | 0.0169 | 0.0017 | 0.0017 | 0 | 0 |
| BO2 | 0.0175 | 0.9599 | 0.0192 | 0.0020 | 0.0014 | 0 |
| BO3 | 0.0016 | 0.0183 | 0.9549 | 0.0187 | 0.0029 | 0.0036 |
| DEF1 | 0.0009 | 0.0025 | 0.0156 | 0.9428 | 0.0243 | 0.0139 |
| DEF2 | 0 | 0.0016 | 0.0033 | 0.0074 | 0.9701 | 0.0177 |
| DEF3 | 0 | 0 | 0 | 0 | 0 | 1 |

Table 4.5: Homogeneous quarterly CTM in 2024 for small business

Since we need a quarterly credit index for each quarter, we introduce the formula

$$PD_q = 1 - (1 - PD_y)^{\frac{1}{4}},$$

which is used in applications to approximate homogeneously the quarterly probability of default $PD_q$ starting from the annual probability of default $PD_y$. We find

| **2022Q4** | **2023Q1** | **2023Q2** | **2023Q3** | **2023Q4** |
|---|---|---|---|---|
| 0.0026 | 0.0031 | 0.0034 | 0.0037 | 0.0040 |

| **2024Q1** | **2024Q2** | **2024Q3** | **2024Q4** |
|---|---|---|---|
| 0.0042 | 0.0047 | 0.0048 | 0.0048 |

from which the quarterly credit indexes per quarter

| **22Q4-23Q1** | **23Q1-23Q2** | **23Q2-23Q3** | **23Q3-23Q4** |
|---|---|---|---|
| 0.0628 | 0.0301 | 0.0261 | 0.0259 |

| **23Q4-24Q1** | **24Q1-24Q2** | **24Q2-24Q3** | **24Q3-24Q4** |
|---|---|---|---|
| 0.0169 | 0.0381 | 0.0032 | 0.0016 |

These credit indexes can be used to shift the quarterly CTMs starting from the one at time 2022Q4, which is computed with the BAM approach. In this way, a different matrix is obtained for each of the eight quarters (e.g. 2023Q1 and 2023Q2 in Tables 4.6 and 4.7).

| | BO1 | BO2 | BO3 | DEF1 | DEF2 | DEF3 |
|---|---|---|---|---|---|---|
| BO1 | 0.9835 | 0.0139 | 0.0014 | 0.0011 | 0 | 0 |
| BO2 | 0.0208 | 0.9602 | 0.0164 | 0.0016 | 0.0009 | 0 |
| BO3 | 0.0019 | 0.0214 | 0.9552 | 0.0162 | 0.0025 | 0.0028 |
| DEF1 | 0.0012 | 0.0030 | 0.0179 | 0.9445 | 0.0222 | 0.0112 |
| DEF2 | 0 | 0.0021 | 0.0040 | 0.0087 | 0.9710 | 0.0141 |
| DEF3 | 0 | 0 | 0 | 0 | 0 | 1 |

Table 4.6: Quarterly CTM at 2023Q1 for small business

**Comparison** Making a comparison is not an easy task since we do not have an error measure. We therefore report only the differences in the one

| | BO1 | BO2 | BO3 | DEF1 | DEF2 | DEF3 |
|---|---|---|---|---|---|---|
| BO1 | 0.9822 | 0.0150 | 0.0016 | 0.0012 | 0 | 0 |
| BO2 | 0.0194 | 0.9602 | 0.0176 | 0.0018 | 0.0011 | 0 |
| BO3 | 0.0017 | 0.0199 | 0.9552 | 0.0173 | 0.0028 | 0.0031 |
| DEF1 | 0.0011 | 0.0027 | 0.0168 | 0.9438 | 0.0236 | 0.0121 |
| DEF2 | 0 | 0.0019 | 0.0037 | 0.0081 | 0.9710 | 0.0152 |
| DEF3 | 0 | 0 | 0 | 0 | 0 | 1 |

Table 4.7: Quarterly CTM at 2023Q2 for small business

and two year cumulative matrices, which are obtained multiplying all the matrices of each quarter in the period considered.

After one year the Frobenius norm of the difference between the cumulative matrices obtained with the two strategies is $7 \times 10^{-3}$ and the maximum difference on a single element is 0.4%. As we could expect, the difference gets bigger after two years, in particular the Frobenius norm is $3 \times 10^{-2}$ and the maximum difference on a single element is 1.5%.

In conclusion, the two strategies are not equivalent and we look forward to exploring the difference in the future to better understand whether the non-homogeneous approach is more convenient or not.

*Remark* 4.2.1. An alternative could be to perturb the macroeconomic forecasts with a stochastic noise and apply a Monte Carlo approach for the estimation of the CTMs. In this case we observe that, while with the non-homogeneous method one can compute the root before the simulation, following the homogeneous strategy the computation of the root has to be done twice for every different scenario. For this reason, it would probably be better to compute the stochastic approximation of the root using the QOM method that is more efficient.

*Remark* 4.2.2. It is possible to obtain quarterly scenario-dependent CTMs with a non-homogeneous approach and annual credit indexes: indeed, one can compute A(2023Q1|2022Q4) as the best stochastic approximation of the fourth root of A(2023Q4|2022Q4). This approach was not implemented.

# Conclusions and future perspectives

The computation of short-term credit transition matrices is a problem for which many competitive solutions are already available. We have seen that, given a CTM $A \in \mathbb{S}^{n\times n}$, the problem reduces to the computation of a stochastic matrix $X \in \mathbb{S}^{n\times n}$ that well approximates $A^{\frac{1}{p}}$. With the strong diagonal dominance assumption QOM and QOG methods perform very well, with a preference for the first, especially for smaller values of $p$. Nonlinear optimization methods stop after few iterations and provide an improvement that is often negligible. However, bank-sourced credit transition matrices do not always meet the strong diagonal dominance condition; in the example provided in Section 3.2, nonlinear optimization gives a huge accuracy improvement but proves to be non-efficient. In the future, it could be interesting to extend the empirical studies on CTMs to bank-sourced matrices and understand whether they are usually strongly diagonally dominant. In case they are not, it would be useful to build a consistent dataset of bank-sourced matrices to conduct a deeper study.

For what concerns the scenario-dependence, we have seen that the classical technique for including the macroeconomic scenario-dependence in an annual CTM can be adapted to evolve short-time credit transition matrices. In future studies, it would be interesting to perform a retrospective validation of the presented strategies and make a deeper comparison between them.

# Appendix A

# Review of Markov chains

**Definition A.0.1** (Discrete-time Markov chain)**.** A discrete-time stochastic process $\{X_n\}_{n\geq 0}$ valued in a discrete space $E$ is a *Markov chain* if the following property holds:

$$P(X_{n+1} = j \mid X_n = i_n, X_{n-1} = i_{n-1}, \dots, X_0 = i_0) = P(X_{n+1} = j \mid X_n = i_n),$$

for every $n \geq 0$ and for every choice of $i_0, \dots, i_n, j \in E$ such that the conditional probability is well defined.
In particolar, the elements of $E$ are called *states* of the chain and $E$ is named *state space.*

**Definition A.0.2** (HMC)**.** A discrete-time Markov chain $\{X_n\}_{n\geq 0}$ is said to be *time-homogeneous* (HMC) if, for all $i, j \in E$, $P(X_{n+1} = j \mid X_n = i)$ does not depend on $n$.

**Definition A.0.3** (Transition matrix)**.** Let $X$ be a discrete-time HMC. We define the transition matrix $P$ as the matrix $P = (p_{ij})_{i,j\in E}$ such that

$$p_{ij} := P(X_{n+1} = j \mid X_n = i).$$

Since $p_{ij}$ represents the probability of an event, we observe that a transition matrix is always stochastic.

**Definition A.0.4.** Consider a HMC $\{X_n\}_{n\geq 0}$, with transition matrix $P$. We say that $X_0$ is the *initial state*; moreover, its distribution

$$v(i) := P(X_0 = i) \ \forall i \in E$$

is named *initial distribution.*

*Remark* A.0.1. The distribution of a HMC at any time step is determined by its initial distribution and its transition matrix. Indeed,

$$P(X_0 = i_0, X_1 = i_1, \ldots, X_k = i_k) =$$

$$P(X_0 = i_0)P(X_1 = i_1 \mid X_0 = i_0) \cdots P(X_k = i_k \mid X_{k-1} = i_{k-1}, \ldots, X_0 = i_0),$$

from which we have that for the Markov property

$$P(X_0 = i_0, X_1 = i_1, \ldots, X_k = i_k) = v(i_0)p_{i_0 i_1} \cdots p_{i_{k-1} i_k}.$$

**Definition A.0.5** (Absorbing state)**.** Let $X$ be a discrete-time HMC and consider a state $i \in E$. We say that $i$ is absorbing for $X$ if $p_{ii} = 1$, or, equivalently, $p_{ij} = 0$ for all $j \in E$, $j \neq i$.

**Definition A.0.6** (Accessible state)**.** Let $X$ be a discrete-time HMC and consider two states $i, j \in E$. We say that $j$ is accessible from $i$ for $X$ if there exist $k, n \in \mathbb{N}$ such that

$$P\left(X_{n+k} = j \mid X_n = i\right) > 0.$$

**Definition A.0.7** (Continuous-time Markov chain)**.** Consider a continuous-time stochastic process $\{X\}_{t\geq 0}$ defined on a probability space with filtration $\{\mathcal{F}\}_{t\geq 0}$ and valued in a discrete space $E$. We say that $X$ is a Markov chain if, for all $i \in E$ and for all $0 \leq t_1 < t_2$,

$$P\left(X_{t_2} = i \mid X_{t_1}\right) = P\left(X_{t_2} = i \mid \mathcal{F}_{t_1}\right).$$

**Definition A.0.8** (Continuous-time HMC)**.** Consider a continuous-time Markov chain $\{X\}_{t\geq 0}$. We say that $X$ is time-homogeneous if, for all $0 \leq t_1 < t_2$,

$$P\left(X_{t_2} \mid X_{t_1}\right) = P\left(X_{t_2 - t_1} \mid X_0\right).$$

**Definition A.0.9** (Transition semigroup)**.** Consider a continuous-time HMC $\{X\}_{t\geq 0}$ and a transition semigroup $P = (p_{ij})_{i,j\in E}$. We say that $P$ is the transition semigroup associated with $X$ if, for all $t \geq 0$,

$$p_{ij}(t) = P(X_t = j \mid X_0 = i).$$

**Definition A.0.10** (Absorbing state)**.** Let $X$ be a continuous-time HMC with transition semigroup $P$ and consider a state $i \in E$. We say that $i$ is absorbing for $X$ if there exists a $\bar{t}$ such that $p_{ii}(t) = 1$ for every $t > \bar{t}$.

**Definition A.0.11** (Accessible state)**.** Let $X$ be a continuous-time HMC with transition semigroup $P$ and consider two states $i, j \in E$. We say that $i$ is accessible from $j$ for $X$ if there exists a $\bar{t} > 0$ such that $p_{ij}(\bar{t}) > 0$.

# Bibliography

[1] Banca d'Italia Circolare n.272, 2006.

[2] Banca d'Italia Glossario, *https://www.bancaditalia.it/footer/glossario/index.html*

[3] Banca d'Italia Non-performing loans (NPLs) in Italy's banking system. *https://www.bancaditalia.it/media/views/2017/npl/index.html*

[4] Belink, B., Suchower, S., and Forest L. R. A one-parameter representation of credit risk and transition matrices. *CreditMetrics Monitor*, Third Quarter 1998.

[5] Boggs, P. T. and Tolle, J. W. Sequential Quadratic Programming. *Acta Numerica*, 1996.

[6] Boyd, S. and Vandenberghe, L. Convex Optimization. *Cambridge University Press*, 2004.

[7] Brémaud, P. Markov Chains. *Springer*, 2013.

[8] Coleman, J.S. Introduction to Mathematical Sociology. *Free press*, 1964.

[9] Davies, E. Embeddable Markov Matrices. *Electronic Journal of Probability*, 2010.

[10] European Banking Authority Capital Requirements Regulation, part threee, title II, chapter 3, Section 6, Sub-Section 2, Article 178. *EBA*, 1996.

[11] Gallier, J. Logarithms and Square Roots of Real Matrices Existence, Uniqueness, and Applications in Medical Imaging. *Department of Computer and Information Science University of Pennsylvania Philadelphia*, 2019.

[12] Gordy, M. B. A Comparative Anatomy of Credit Risk Models. *Board of Governors of the Federal Reserve System*, 1998.

[13] Higham, N.J. Function of matrices, Theory and computation. *Siam*, 2008.

[14] Hughes, M. and Werner, R. Choosing Markovian Credit Migration Matrices by Nonlinear Optimization. *MDPI*, 2016.

[15] Israel, R., Rosenthal, J. and Wei, J. Finding generators for Markov chains via empirical transition matrices, with application to credit raitings. *Mathematical Finance, 11(2): 245-265*, 2001.

[16] ISTAT https://www.istat.it

[17] Kamm, K. A Unified Model for XVA, including Interest Rates and Rating. *Alma Mater Studiorum – Università di Bologna*, 2023.

[18] Kraft, D. Projection onto the probability simplex: A Software Package for Sequential Quadratic Programming. *Institute for Flight Mechanics, Koln, Germany*, 1988.

[19] Kreinin, A. and Sidelnikova, M. Regularization algorithms for transition matrices. *Algo Research Quarterly, 4(1/2):23-40*, 2001.

[20] Lin, L. Roots of stochastic matrices and fractional matrix powers. *University of Manchester*, 2011.

[21] Ludwig, T. Markov Chain based PD Term Structure Modelling in an IFRS 9 Framework. *Universität Innsbruck*, 2019.

[22] Norris, J.R. Markov chains. *Cambridge University press*, 1997.

[23] OECD https://data.oecd.org/

[24] Prometeia Prometeia Brief - Italy in the Global economy April 2023. *Prometeia*, 2023.

[25] Quarteroni, A., Sacco, R., Saleri, F. and Gervasio, P. Matematica Numerica. *Springer*, 2014.

[26] Singer, B. and Spilerman, S. The representation of social processes by Markov models. *American journal of sociology*, 1976.

[27] Toffano, C., Nisi, F. and Maurri, L. IFRS 9, Stress Testing, ICAAP: a comprehensive framework for PD calculation. *Tech. rep. Prometeia*, 2017.

[28] Wei, J.Z. A multi-factor, credit migration model for sovereign and corporate debts. *Journal of International Money and Finance*, 2003.

[29] Wang, W. and Carreira-Perpiñán, M.Á. Projection onto the probability simplex: An efficient algorithm with a simple proof, and an application. *University of California*, 2013.

[30] Wright, S.J. and Nocedal, J. Numerical Optimization. *Springer*, 2006.

[31] https://sdw.ecb.europa.eu