Technische Universität München

ZENTRUM MATHEMATIK

Estimation of COGARCH Models with implementation in R

Masterarbeit

von

Marlit Granzer

Themenstellerin:	Prof. Dr. Claudia Klüppelberg
Betreuer:	Dr. Stephan Haug
Abgabetermin:	23. April 2013

Hiermit erkläre ich, Marlit Granzer, dass ich die Masterarbeit selbstständig angefertigt und nur die angegebenen Quellen verwendet habe.

Garching, den 23. April 2013

Acknowledgments

First of all I would like to thank Prof. Dr. Claudia Klüppelberg for the interesting topic of this thesis. Special thanks go to my supervisor Dr. Stephan Haug. I would like to thank him for the great valuable support, motivation and advise via many talks, emails and fast answers! Also I would like to thank my friends. Especially, for the company and motivation during all the hours we spent in the library together. Furthermore I am very thankful for the patience and all the support of my boyfriend Jonas during the last months. Additionally I am grateful that my family supported me during my years of studying.

Abstract

In this thesis we consider continuous time GARCH models. Klüppelberg et al. (2004) suggested a new approach to obtain a continuous time GARCH model based on a discrete time GARCH model. We give an insight into some extensions of the continuous time GARCH model and their properties. These models are the COGARCH(p,q) model, the exponential COGARCH(p,q) model and asymmetric COGARCH models like the continuous time GJR GARCH$(1,1)$ model. Furthermore, we introduce estimation methods for the determination for the parameters of these models. A moment estimator method developed by Haug et al. (2007) and the pseudo-maximum likelihood method of Maller et al. (2008) are discussed. We extended this method for the case of a COGARCH$(1,1)$ process driven by a compound Poisson process with random time points. Both the simulation methods of the different continuous time GARCH models and the estimation methods of their parameters are implemented in R and summarized in the package *cogarch*. We offer a detailed explanation of this R package and provide several examples for their application in this thesis.

Contents

Chapter 1

Introduction

In the field of financial time series the volatility of the price process is of great interest. Due to future developments of the stock market, for example. In financial time series the phenomenon of volatility clustering can be observed, i.e. extreme returns are followed by other extreme returns. Over years, volatility was modeled under the assumption of constant volatility. The development of discrete time models for changing volatility by Engle (1982) and Bollerslev (1986) was a milestone in the analysis of financial time series. Engle (1982) developed the famous ARCH (autoregressive conditionally heteroscedastic) model and Bollerslev (1986) introduced the generalization of this model - the GARCH (generalized ARCH) model. Models with generalized autoregressive conditional heteroscedasticity provide the volatility by the previous values of the process. The GARCH model captures the main characteristics of financial data, the "stylized facts".

Nowadays, nearly all data available can be recorded. Considering for example trading transactions, every single transaction is recorded. This growing amount of available data is called "high-frequency-data", as it is not recorded in fixed time intervals only. Taking the data only at fixed time intervals neglects some of the information which is available. In order to analyse and model this huge amount of data, which can be unequally spaced in time, an extension from discrete time models to continuous time models is necessary. Several attempts have been made to define a continuous time GARCH model that covers the properties of the discrete time GARCH model.

A first approach to create a continuous time GARCH model goes back to Nelson (1990). He tried to extend the discrete-time model by making diffusion approximations. In his paper, he tried to bridge the gap between continuous time nonlinear stochastic differential systems and their relation with ARCH stochastic difference equation systems. Discrete time GARCH models have only one source of uncertainty (randomness), whereas stochastic volatility models like the model of Nelson (1990) have two sources of uncertainty. Furthermore, the continuous time stochastic volatility model of Barndorff-Nielsen and Shepard (2001) models the volatility with an Ornstein-Uhlenbeck process driven by a Lévy process. Modelling jumps with this model is possible but it contains two independent stochastic processes.
The new approach of Klüppelberg et al. (2004) includes only one source of uncertainty. The idea for the construction of their continuous time GARCH$(1,1)$ model (COGA-

RCH model) is to preserve the structure and the main characteristics of a discrete time GARCH model. The COGARCH$(1,1)$ model is obtained with a discrete time GARCH model and depends on a driving Lévy process. Brockwell et al. (2006) generalized this model for $1 \leq p \leq q$.

Another typical characteristic of financial time data is the so-called Leverage effect. This means that volatility tends to increase after neagtive shocks and to decrease after positive ones. Like in the discrete time case, the COGARCH model cannot model this phenomenon. Therefore, some extensions have been developed. Based on the discrete time EGARCH model proposed by Nelson (1990), the exponential COGARCH model has been introduced by Haug (2006). Furthermore, Ding et al. (1993) defined an asymmetric continuous time GARCH model - the APARCH model.

Regarding the discrete time GARCH models, several packages for the analysis of those models are provided by the R Core Team (2012). In this thesis, a new package, called *cogarch* package, which can be used for the analysis of continuous GARCH models, is introduced.

Aim and Structure of this thesis

The aim of this thesis is to give an overview of the different continuous time GARCH models and to explain and illustrate the implementation of the R package *cogarch*. In the beginning we give an introduction on the theory of Lévy processes in Chapter 2, which we will utilize in this thesis. Especially we will focus on the compound Poisson and the Variance Gamma process. The introduction of the continuous time process developed by Klüppelberg et al. (2004) follows in Chapter 3. Using their approach, it is possible to develop the continuous time GARCH model (COGARCH) based on the discrete time GARCH model, developed by Bollerslev (1986). We summarize some properties like stationarity and moments of this model. Furthermore, we consider two examples of COGARCH$(1,1)$ processes - the first process driven by a compound Poisson process and the second one driven by a Variance Gamma process. In Chapter 3.1.4, we give an insight in the extension of the COGARCH model to $p, q > 1$ introduced by Brockwell et al. (2006). In Chapter 3.2, two different estimation methods are proposed. First, the moment estimation method developed by Haug et al. (2007) is considered. The proposed estimation algorithm is basis of the corresponding function implemented in R. We then give a detailed description of the pseudo-maximum-likelihood (PML) method, which can be used for irregularly spaced time series. This method is based on an approximation of the continuous time GARCH process by an embedded sequence of discrete time GARCH series, which converge in probability to the continuous time model in the Skorokhod metric. In Chapter 3.2.3, we extend the PML method to a "modified" PML method. This modified method can be applied to random time points, e.g. the random jumptimes of a compound Poisson process. We conclude the chapter about estimation methods with a simulation study in Chapter 3.3. In Chapter 4, we give an overview of the exponential COGARCH (ECOGARCH) model, developed by Haug (2006).We provide some examples in the end of this chapter for further insights. Following, we summarize the approach for an asymmetric continuous time GARCH model in Chapter 5. Lee (2010) developed such an asymmetric model - the

continuous time APARCH model, which is based on a discrete time APARCH model. The PML method for the (symmetric) continuous time GARCH model can be adjusted to the (asymmetric) continuous GJR GARCH model as shown by Mayr (2013). Again, the chapter is concluded by a small simulation study, see Chapter 5.2.3. In Chapter 6 we explain the implementation of our R package *cogarch*, see also the Appendix for the documentation of each function. We explain the different simulation and estimation functions. Furthermore, we give a small example how to use each of these functions. In the last Chapter of this thesis, we use the asymmetric PML method to fit a GJR GARCH$(1,1)$ model to the increments of windspeed data.

Chapter 2

Lévy Processes

For the construction of a continuous time version of the discrete time GARCH process we will replace the innovations in the discrete time GARCH model with the increments of a Lévy process. Therefore we start with a short introduction on some facts on Lévy processes as a basis for this thesis.
The following chapter is based on Applebaum (2009). More details, examples and the corresponding proofs can be found there. For further information on Lévy processes Sato (1999) and Kyprianou (2006) can be recommended.

Definition 2.1 *Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space. A real-valued stochastic process $L := (L_t)_{t\geq 0}$ is a **Lévy process** if*

(L1) $L_0 = 0$ *a.s.,*

(L2) *L has independent increments: $L_t - L_s$ is independent of $\{L_u : u \leq s\}$ for $0 \leq s \leq t < \infty$,*

(L3) *L has stationary increments: $L_t - L_s$ is equal in distribution to L_{t-s} for $0 \leq s \leq t < \infty$,*

(L4) *L is stochastically continuous:* $\lim_{t\to s} \mathbb{P}(|L_t - L_s| > \epsilon) = 0.$

Some important examples of Lévy processes are the Brownian motion, the Poisson process and the compound Poisson process, which we describe in the following examples. First we give an example of a standard Brownian motion.

Example 2.2 *(Standard Brownian motion) (Applebaum (2009, Example 1.3.8))*
A standard Brownian motion in $\mathbb{R}$ is a Lévy process $B = (B_t)_{t\geq 0}$ for which,

(i) $B_t \sim \mathcal{N}(0, t)$ *for every $t \geq 0$ and*

(ii) *B has continuous paths.*

A sample path of a standard Brownian motion can be seen in Figure 2.1.

The *compound Poisson processes* uses a Poisson processes as a basis. A Possion process is defined as follows.

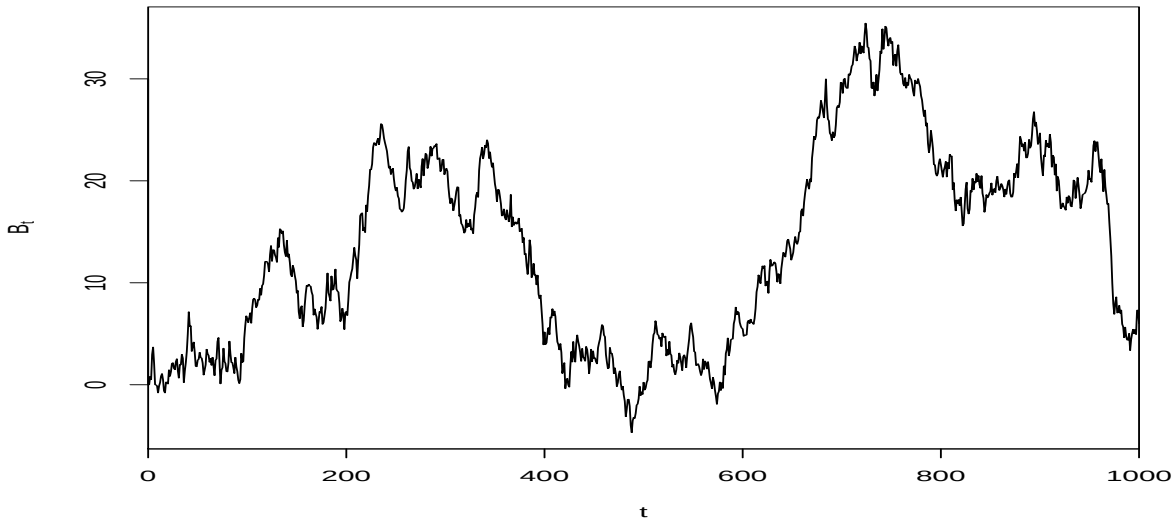


Figure 2.1: Standard Brownian Motion

Example 2.3 *(Poisson process) (Applebaum (2009, Example 1.3.9))*
*A **Possion process** of intensity $\lambda > 0$ is a **Lévy process** N taking values in $\mathbb{N}_0$ wherein each $N_t \sim \pi(\lambda t))$, with $\pi(\cdot)$ the Posisson distribution, so that we have*

$$\mathbb{P}(N_t = n) = \frac{(\lambda t)^n}{n!} e^{-\lambda t},$$

for each $n \in \mathbb{N}_0$.
The sample paths of N are clearly piecewise constant on finite intervals with jump discontinuities of size 1 at each of the random times $(T_n, n \in \mathbb{N})$, where $T_0 := 0$ and $T_n := inf\{t \geq 0; N_t = n\}$. A sample path of a Poisson process with rate $\lambda = 1$ is displayed in Figure 2.2.

Now we are ready to continue with further details on the compound Poisson process.

Example 2.4 *(Compound Poisson process) (Applebaum (2009, Example 1.3.10))*
*We have a Poisson process N for $t \geq 0$ and parameter $\lambda > 0$, cf. Example 2.3. If this process is independent of an i.i.d. (independent and identically distributed) sequence of random variables $(Y_i)_{i \in \mathbb{N}}$, then a **compound Poisson process** L is defined as*

$$L_t = \sum_{i=1}^{N_t} Y_i, \qquad t \geq 0.$$

The compound Poisson process has jumps with random size instead of the constant jumps of size 1 of a Poisson process.

In Figure 2.3 we simulated a sample path of a compound Poisson process with $\lambda = 2$ and normally distributed jumpsizes with $\mu = 0$ and $\sigma = 1$. In Chapter 6 we describe

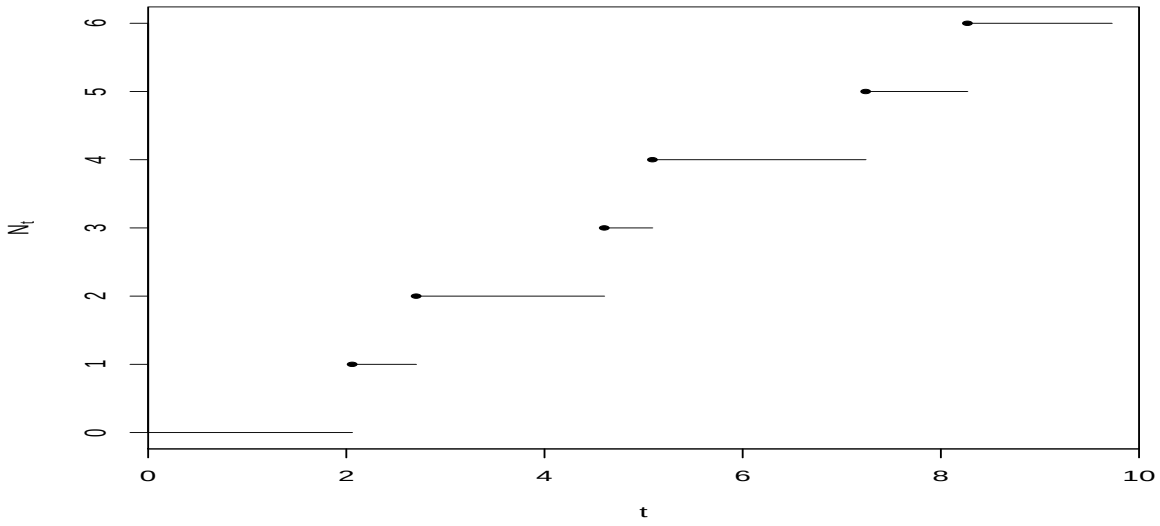


Figure 2.2: Poisson Process with intensity $\lambda = 1$

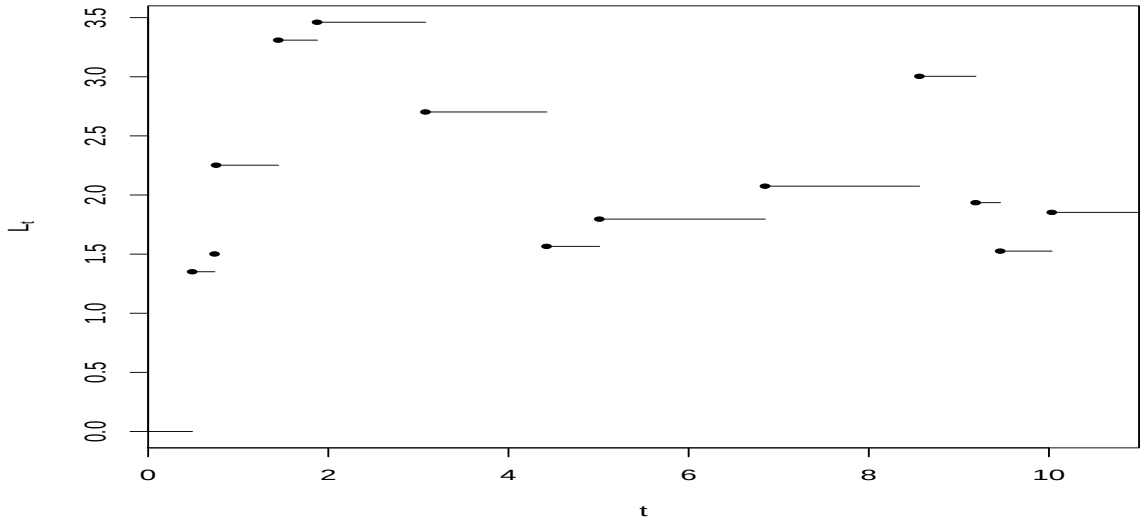


Figure 2.3: Compound Poisson process with $\lambda = 2$ and normally distributed jumpsizes with $\mu = 0$ and $\sigma = 1$ on the time interval $[0, 10]$.

how such a simulation can be implemented in R. In order to simulate the process on a random grid, we assume that the time intervals of the jumps are i.i.d. exponentially distributed random variables with a specified rate.

Another example we are going to use is the Variance Gamma process, see also Chapter 6.

Example 2.5 *(Variance Gamma process)*
Due to Madan et al. (1998, Section 2) a Variance Gamma process is obtained by evaluating

Brownian motion with drift at a random time given by a Gamma process. Let a Brownian motion with drift θ and variance σ be defined as

$$b(t, \theta, \sigma) = \theta t + \sigma B(t),$$

where $B(t)$ is a standard Brownian motion.
The time change of the Brownian motion is done with respect to a Gamma process $(H_t)_{t\geq 0}$ with parameters $a, b > 0$, such that each of the i.i.d increments is Gamma distributed with density

$$f_{H_t}(x) = \frac{b^{at}}{\Gamma(at)} x^{at-1} e^{-bx},$$

for $x \geq 0$, where $\Gamma(\cdot)$ denotes the Gamma function. Then the Variance Gamma process $V := (V_t)_{t\geq 0}$ with parameters $\sigma > 0$, $\tau > 0$ and $\theta \in \mathbb{R}$ can be obtained by

$$V_t := \theta H_t + \sigma B_{H_t}, \tag{2.1}$$

where $(B_t)_{t\geq 0}$ is a standard Brownian motion and $(H_t)_{t\geq 0}$ a Gamma process with parameters $a = 1/\tau$ and $b = 1/\tau$.
The characteristic function of V for $t \geq 0$ is then given by

$$\mathbb{E}(e^{iuV_t}) = \left(1 - iu\theta\tau + \frac{1}{2}\sigma^2\tau u^2\right)^{-t/\tau}, \tag{2.2}$$

for $u \in \mathbb{R}$. In Madan et al. (1998, Equation (13)) the Lévy measure ν_V has been defined as

$$\nu_V(dx) = \begin{cases} \frac{\mu_n^2}{\tau_n} \frac{\exp(-\frac{\mu_n}{\tau_n}|x|)}{|x|} dx, & \text{for } x < 0, \\ \frac{\mu_p^2}{\tau_p} \frac{\exp(-\frac{\mu_p}{\tau_p}|x|)}{|x|} dx, & \text{for } x > 0, \end{cases} \tag{2.3}$$

with

$$\begin{aligned} \mu_p &= \frac{1}{2}\sqrt{\theta^2 + \frac{2\sigma^2}{\tau}} + \frac{\theta}{2}, \\ \mu_n &= \frac{1}{2}\sqrt{\theta^2 + \frac{2\sigma^2}{\tau}} - \frac{\theta}{2}, \\ \tau_p &= \left(\frac{1}{2}\sqrt{\theta^2 + \frac{2\sigma^2}{\tau}} + \frac{\theta}{2}\right)^2 \tau, \\ \tau_n &= \left(\frac{1}{2}\sqrt{\theta^2 + \frac{2\sigma^2}{\tau}} - \frac{\theta}{2}\right)^2 \tau. \end{aligned}$$

Inserting μ_p, μ_n, τ_p and τ_n in (2.3) gives for $x < 0$

$$\nu_V(x) = \frac{1}{\tau|x|} \exp\left(-\frac{1}{\left(\frac{1}{2}\sqrt{\theta^2 + \frac{2\sigma^2}{\tau}} - \frac{\theta}{2}\right)\tau}|x|\right) dx$$

$$= \underbrace{\frac{1}{\tau}}_{=:C} \frac{1}{|x|} \exp\left(x \underbrace{\left(\sqrt{\frac{1}{4}\theta^2\tau^2 + \frac{1}{2}\sigma^2\tau} - \frac{\theta\tau}{2} \right)^{-1}}_{=:G} \right) dx,$$

and for $x > 0$,

$$\nu_V(x) = \frac{1}{\tau x} \exp\left(- \frac{1}{\left(\frac{1}{2}\sqrt{\theta^2 + \frac{2\sigma^2}{\tau}} + \frac{\theta}{2} \right)\tau} x \right) dx$$

$$= \underbrace{\frac{1}{\tau}}_{=:C} \frac{1}{x} \exp\left(-x \underbrace{\left(\sqrt{\frac{1}{4}\theta^2\tau^2 + \frac{1}{2}\sigma^2\tau} + \frac{\theta\tau}{2} \right)^{-1}}_{=:M} \right) dx.$$

Thus, we can express the Lévy measure ν_V *as*

$$\nu_V(dx) = \begin{cases} C\exp(Gx)|x|^{-1}dx, & \text{for } x < 0, \\ C\exp(-Mx)x^{-1}dx, & \text{for } x > 0, \end{cases} \tag{2.4}$$

with

$$C = 1/\tau, \quad G = \left(\sqrt{\frac{1}{4}\theta^2\tau^2 + \frac{1}{2}\sigma^2\tau} - \frac{1}{2}\theta\tau \right)^{-1} \text{ and } M = \left(\sqrt{\frac{1}{4}\theta^2\tau^2 + \frac{1}{2}\sigma^2\tau} + \frac{1}{2}\theta\tau \right)^{-1},$$

cf. Haug (2006, Example 1.1.7).
Furthermore the mean and the variance of V *at time* $t \geq 0$ *are given by*

$$\mathbb{E}(V_t) = \theta t \quad \text{and} \quad \mathbb{V}ar = (\theta^2\tau + \sigma^2)t. \tag{2.5}$$

In Chapter 6 we are going to describe how to simulate a sample path of a Variance Gamma process in more detail. For an example of a simulated sample path of a Variance Gamma process with parameters $\sigma = 0.3$, $\theta = -0.03$ and $\tau = 0.5$ see Figure 2.4.

Through the 1960s and 1970s Lévy processes have been referred to as *processes with stationary independent increments*. Through a complete characterisation of infinitely divisible distributions, Lévy characterised the general class of processes with stationary independent increments in 1934. Later, Khintchine (1937) and Itô (1942) dealt with Lévy's original proof and provided further simplification[1]. We continue with further details about the link between Lévy processes and infinitely divisible distributions, and their definition.
Applebaum (2009) defines infinite divisibility in Chapter 1.2.2 as follows.

[1] Kyprianou (2006, Chapter 1.1)

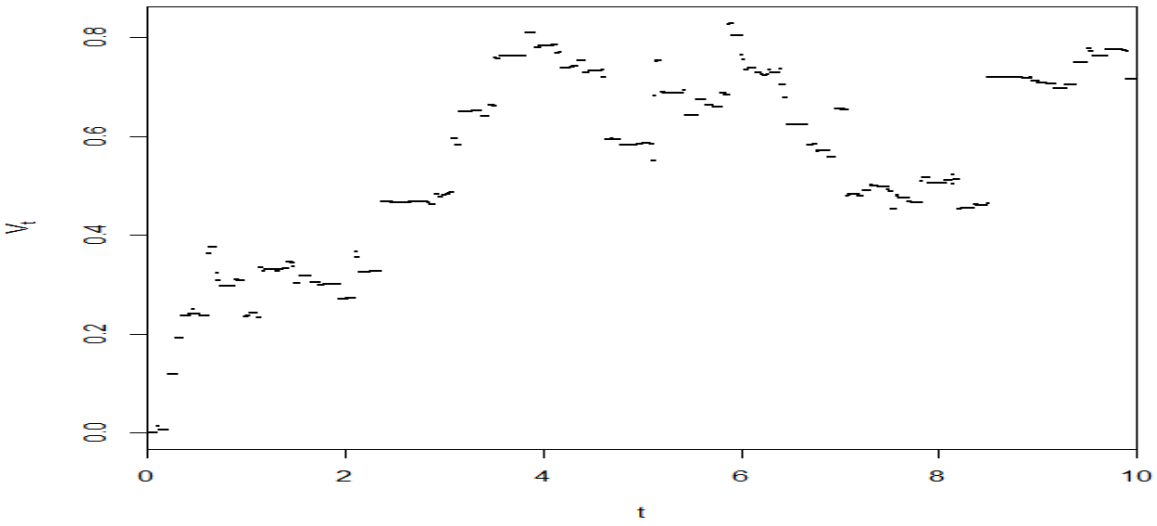


Figure 2.4: Variance Gamma process with $\sigma = 0.3$, $\theta = -0.03$, $\tau = 0.5$.

Definition 2.6 *Let X be a real-valued random variable. We say that X has an **infinitely divisible distribution** if for each $n \in \mathbb{N}$ there exist i.i.d. random variables $Y_1^{(n)}, \ldots, Y_n^{(n)}$ such that*

$$X \stackrel{d}{=} Y_1^{(n)} + \cdots + Y_n^{(n)}, \tag{2.6}$$

where $\stackrel{d}{=}$ denotes equality in distribution.

In addition we recall the definition of a characteristic function, as we will use it below.

Definition 2.7 *(Applebaum (2009, Ch. 1.1.6))*
*Let X be a random variable defined on $(\Omega, \mathcal{F}, \mathbb{P})$ and taking values in $\mathbb{R}$ with the probability law μ_X. Its **characteristic function** φ_X is defined by*

$$\varphi_X(u) = \mathbb{E}(e^{iuX}) = \int_\Omega e^{iuX(\omega)} \mathbb{P}(d\omega) = \int_{\mathbb{R}} e^{iux} \mu_X(dx), \tag{2.7}$$

for each $u \in \mathbb{R}$.

According to Proposition 1.2.6 in Applebaum (2009) the following two statements are equivalent:

- X is infinitely divisible, i.e. has an infinitely divisible distribution;
- φ_X has a nth root that is itself the characteristic function of a random variable, for each $n \in \mathbb{N}$. That is $\varphi_X(u) = (\varphi_n(u))^n$ where φ_n is the characteristic function of a random variable.

In Applebaum (2009, Chapter 1.2.3) you can find some examples of infinitely divisible distributions: e.g. the Normal distribution, the Poisson distribution or the compound Poisson distribution.

Example 2.8 *(Poisson random variables) (Applebaum (2009, Example 1.2.9))*
Let us consider a random variable X taking values in $n \in \mathbb{N}_0$ with

$$\mathbb{P}(X = n) = \frac{\lambda^n}{n!} e^{-\lambda}.$$

In this case we have $\mathbb{E}(X) = Var(X) = \lambda$. The calculation of the characteristic function φ_X yields

$$\varphi_X(u) = \mathbb{E}(e^{iuX}) = \sum_{n=0}^{\infty} e^{iun} \frac{\lambda^n}{n!} e^{-\lambda} = e^{-\lambda} \sum_{n=0}^{\infty} \frac{(\lambda e^{iu})^n}{n!} = e^{\lambda(e^{iu}-1)} = (e^{\frac{\lambda}{n}(e^{iu}-1)})^n. \quad (2.8)$$

Which, considering Definition 2.6, implies that X is infinitely divisible with each $Y_k^{(n)} \sim \pi(\frac{\lambda}{n})$, for $1 \leq k \leq n$, $n \in \mathbb{N}$ and $\pi(\cdot)$ the Poisson distribution.

Considering a Lévy process L, we can re-write it as

$$L_t = L_t + L_{\frac{t}{n}} - L_{\frac{t}{n}} + L_{\frac{2t}{n}} \mp \cdots - L_{\frac{(n-1)t}{n}} = L_{\frac{t}{n}} + (L_{\frac{2t}{n}} - L_{\frac{t}{n}}) + \cdots + (L_t - L_{\frac{(n-1)t}{n}}). \quad (2.9)$$

As L_t has stationary independent increments, it follows that the summands on the right hand side of Equation 2.9 are i.i.d. random variables. Therefore we can conclude that L_t has an infinitely divisible distribution for each $t \geq 0$. So any Lévy process can be associated with an infinitely divisible distribution.

But given an infinitely divisible distribution, we would like to know if it is possible to construct a Lévy process L, such that L_1 has that distribution[2]. We will see that by the *Lévy-Khintchine formula for Lévy processes*, or also called the *Lévy-Khintchine representation*, it is possible to give a characterisation of every infinitely divisible distribution through its characteristic function[3]. According to Applebaum (2009, Overview and Theorem 1.3.3) any Lévy process has a specific form of its characteristic function, as it has stationary independent increments – the Lévy-Khintchine representation[4].

Theorem 2.9 ***(Lévy-Khintchine)***
*If L is a Lévy process, then for all $t \geq 0$, $u \in \mathbb{R}$ the **Lévy-Khintchine formula** is given by*

$$\mathbb{E}(e^{iuL_t}) = e^{t\psi_L(u)} \quad (2.10)$$

where

$$\psi_L(u) = iu\gamma_L - \frac{1}{2}\sigma_L^2 u^2 + \int_{\mathbb{R}} [e^{iux} - 1 - iux\mathbb{1}_{[-1,1]}(x)]\nu_L(dx), \quad (2.11)$$

with $\gamma_L \in \mathbb{R}$, $\sigma_L > 0$ and ν_L a measure on $\mathbb{R}$ satisfying

$$\nu_L(\{0\}) = 0 \quad \text{and} \quad \int_{\mathbb{R}} (|x|^2 \wedge 1)\nu_L(dx) < \infty. \quad (2.12)$$

[2] Kyprianou (2006, Chapter 1.1) and Applebaum (2009, Proposition 1.3.1)
[3] Sato (1999, Theorem 8.1)
[4] Applebaum (2009, Theorem 1.2.14 and Equation (1.19))

We call ν_L the **Lévy measure** of L. Furthermore the representation of the Lévy-Khintchine formula in Theorem 2.9 by γ_L, σ_L and ν_L is unique. This triplet $(\gamma_L, \sigma_L, \nu_L)$ is called the **characteristic triplet** of L. The map ψ_L (2.11) is known as the **Lévy symbol** or **Lévy exponent**.

Remark 2.10 *According to Applebaum (2009, Chapter 1.2.4) it follows from Equation* (2.12) *that for a Lévy measure ν_L we have $\nu_L((-\epsilon, \epsilon)^c) < \infty$ for all $\epsilon > 0$. In this case we have a **finite activity** Lévy process, e.g a compound Poisson process, see Example 2.4. On the other hand it is possible to have $\nu_L(\mathbb{R}) = \infty$, which implies a Lévy process with **infinite activity**. For example a Variance Gamma process has infinity activity, i.e. it has an infinte number of jumps in any time interval, see Example 2.5.*

Later we will denote the jumps of a Lévy process by $\Delta L_t := L_t - L_{t-}$, where L_{t-} is the left limit of the sample path of L at time $t > 0$, that is defined as below.

Definition 2.11 *(Applebaum (2009, Chapter 2.9 Appendix))*
For $0 \leq s \leq t$ we define the left limit L_{t-} as

$$L_{t-} := \lim_{s \uparrow t} L_s \tag{2.13}$$

Additionally we are going to use the abbreviations "*càdlàg*" and "*càglàd*", which are defined as follows.

Definition 2.12 *(càdlàg/ càglàd)*
*The term "càdlàg" stands for the French "continue à droite limité a gauche". A stochastic process $(X_t)_{t \geq 0}$ is called **càdlàg** if it has right continuous sample paths and limits on the left almost surely.*
*Analogously "càglàd" is the short form of "continue à gauche limité a droite", i.e. $(X_t)_{t \geq 0}$ is **càglàd** if it has left continuous sample paths and limits on the right almost surely.*

According to Applebaum (2009, Theorem 2.1.8) every Lévy process has a càdlàg modification that is itself a Lévy process.

In order to give the definition of the Lévy-Itô decomposition we need to introduce some notations.[5] Regarding the jump process $\Delta L := (\Delta L_t)_{t \geq 0}$ of the Lévy process L, we can define a random measure counting the jumps of L of a specific size. For $0 \leq t < \infty$ and $A \in \mathcal{B}(\mathbb{R})$ define

$$N_L(t, A) = \#\{0 \leq s < t; \Delta L_s \in A\} = \sum_{0 \leq s < t} \mathbb{1}_{\{\Delta L_s \in A\}}.$$

Due to Applebaum (2009, Chapter 2.3.1) the measure N_L is a Poisson random measure, which implies that:

(i) For each $t > 0$, $\omega \in \Omega$, $N_L(t, \cdot)(\omega)$ is a counting measure on $\mathcal{B}(\mathbb{R} \backslash \{0\})$.

[5]The following can be found in Haug (2006, Preliminaries) and is based on Applebaum (2009, Chapter 2.3.1 and 2.3.2).

(ii) For each A bounded away from 0 $(N_L(t,A), t \geq 0)$ is a Poisson process with intensity $\mathbb{E}(N_L(1,A))$, where $A \in \mathcal{B}(\mathbb{R})$ is said to be bounded away from 0 if $0 \notin \overline{A}$.

The intensity of the Poisson process $(N_L(t,A), t \geq 0)$, $A \in \mathcal{B}(\mathbb{R}\backslash\{0\})$, describes the expected number, per unit time, of jumps with size belonging to A, and defines a measure on $\mathcal{B}(\mathbb{R}\backslash\{0\})$ which is equal to ν_L, i.e.

$$\nu_L(A) = \mathbb{E}(N_L(1,A)).$$

This measure ν_L is called Lévy measure and is the same as in Theorem 2.9. Furthermore define for each $t \geq 0$ and A bounded away from 0 the *compensated Poisson random measure* by

$$\tilde{N}_L(t,A) = N_L(t,A) - t\nu_L(A). \tag{2.14}$$

In (Applebaum (2009, Theorem 2.4.16)) the following Definition for the Lévy-Itô decomposition of the sample path of L into continuous and jump part can be found. For this, the concept of integration with respect to a Poisson random measure is necessary. For details see Chapter 2.3.2 of Applebaum (2009).

Definition 2.13 *(The Lévy-Itô decomposition)*
If L is a Lévy process, then there exists $\gamma_L \in \mathbb{R}$, a Brownian motion B with variance σ_L^2 and an independent Poisson random measure N_L on $\mathbb{R}_+ \times \mathbb{R}\backslash\{0\}$ sucht that, for each $t \geq 0$

$$L_t = \gamma_L t + B_t + \int_{|x|<1} x\tilde{N}_L(t,dx) + \int_{|x|\geq 1} xN_L(t,dx), \tag{2.15}$$

where $\gamma_L = \mathbb{E}\left(L_1 - \int_{|x|\geq 1} xN_L(1,dx)\right)$.

Chapter 3

Continuous time GARCH Processes

3.1 COGARCH Process

In order to model stochastic volatility there have been different approaches to find a suitable model. For example, Barndorff-Nielsen and Shepard (2001) developed a stochastic volatility model, where the volatility process $(\sigma_t^2)_{t\geq 0}$ is described by an Ornstein-Uhlenbeck (OU) type process, which is driven by a Lévy process $(L_t)_{t\geq 0}$. For further information see also Barndorff-Nielsen and Shepard (2002).
In this thesis we will focus on a completely new approach: the continuous time GARCH$(1,1)$ ("COGARCH") model. This model has been developed by Klüppelberg et al. (2004) as an extension of the discrete time GARCH model. The following will be based on their paper, including the proofs necessary for the following. The generalized case of a continuous time GARCH model can be found in Brockwell et al. (2006), and see Chadraa (2010) for details.

We are going to introduce the continuous time GARCH type models below. First we will have a look at the discrete time GARCH$(1,1)$ model as we will see that it can be extended to the continuous time GARCH$(1,1)$ model. The discrete time GARCH(1,1) process developed by Bollerslev (1986) was defined as

$$Y_n = \epsilon_n \sigma_n \tag{3.1}$$

$$\sigma_n^2 = \beta + \lambda Y_{n-1}^2 + \delta \sigma_{n-1}^2,\ n \in \mathbb{N}, \tag{3.2}$$

with the parameters $\beta > 0$, $\lambda \geq 0$, $\delta \geq 0$ and where $(\epsilon_n)_{n\in\mathbb{N}}$ is an i.i.d. innovation sequence. The abbreviation GARCH stands for "Generalised Autoregressive Conditional Heteroscedasticity". These models are able to incorporate "feedback" between an observation and its volatility. Autoregressive indicates that past observations and the past volatilities have an impact on the present value of the volatility and therefore on the present observation, which is referred to as the "autoregressive aspect" of the recursion formula (3.1). Furthermore conditional heteroscedasticity means that we have a time-varying, non-constant, conditional volatility. The setting of non-constant volatility is more realistic. In financial time series we can observe that more extreme observations lead to large fluctuations in the volatility. In the general ARCH setting, the conditional volatility not only depends on the past observations but also on the

past conditional volatilities. Moreover Equation (3.1) is specifying the "mean level" process (the observed data) and Equation (3.2) models the conditional volatility process, which is time dependent and randomly fluctuating.[6] The GARCH$(1,1)$ model captures the main characteristics of financial data, the "stylized facts", see Remark 3.1.

Remark 3.1 *(Stylized Facts)(McNeil et al. (2005, Chapter 4.1.1))*
Considering data of financial time series such as log-returns on indexes or exchange rates, a collection of empirical observations, known as ***stylized facts****, can be noticed. A version of these facts is as follows.*

- *Return series are not i.i.d. although they show little serial correlation.*
- *Series of absolute or squared returns show profound serial correlation.*
- *Conditional expected returns are close to zero.*
- *Volatility appears to vary over time.*
- *Return series are leptokurtic (i.e. more narrow in the center, but longer and heavier tails than the normal distribution) or heavy-tailed.*
- *Extreme returns appear in clusters (volatility clustering).*

With the GARCH$(1,1)$ model it is possible to analyse financial time series data which has equidistant time data, e.g. one data record per day or every 5 minutes. But on account of the fast development of higher and higher memory capacities of computers, it has been possible to record more and more data during the last years. Considering for example trading transactions, every single transaction is recorded nowadays. We refer to this growing amount of data as "high-frequency data". This amount of data is not recorded in fixed time intervals only. Taking this data only on fixed time intervals neglects some of the information which is available. In order to analyse and model this huge amount of data, which can be irregularly spaced in time, an extension from discrete time models to continuous time models is necessary.
A first approach to create a continuous time GARCH model goes back to Nelson (1990). He tried to extend the discrete-time model by making diffusion approximations. In his paper, he tried to bridge the gap on the relation between continuous time nonlinear stochastic differential systems and the ARCH stochastic difference equation systems. He realised his approach by developing conditions under which ARCH stochastic difference equation systems converge in distribution to Itô processes as the length of the discrete-time intervals between the observations goes to zero.[7] His limiting diffusion model is:

$$dY_t = \sigma_t dB_t^{(1)},\ t \geq 0,$$

where σ_t, the volatility process, satisfies

$$d\sigma_t^2 = (\beta - \eta\sigma_t^2)dt + \phi\sigma_t^2 dB_t^{(2)},$$

[6] cf. Klüppelberg et al. (2011, p.3)
[7] cf. Nelson (1990, p.8).

where $B^{(1)}$ and $B^{(2)}$ are independent Brownian motions, and $\beta > 0$, $\eta \geq 0$ and $\phi \geq 0$ are constants.[8] Discrete time GARCH models have only one source of uncertainty (randomness), whereas stochastic volatility models like the model of Nelson (1990), described above, have two sources of uncertainty due to the two independent Brownian motions. Further information on approaches to create continuous time GARCH models can be found in the paper of Drost and Werker (1996).
The problem of such diffusion limits to discrete time GARCH models is that many of the features of the discrete time GARCH models (see Remark 3.1) are lost. Fasen et al. (2006) showed in their paper that classical stochastic volatility models driven by Brownian motion can model heavy tails, but obviously they are not able to model volatility jumps. As empirical volatility has upwards jumps, it seems to make sense to choose a model driven by a Lévy process, due to its ability of modelling such jumps. [9]

The approach of Klüppelberg et al. (2004) includes only one source of uncertainty (two of the same kind). The increments of a Lévy process replace the innovations of a discrete time GARCH model. Additionally it contains the autoregressive property which can be found in the discrete time case.

As mentioned above in this chapter, the idea of Klüppelberg et al. (2004) for the construction of a continuous time GARCH model is to preserve the structure and the main characteristics of a discrete time GARCH model, e.g. the "stylized facts", see Remark 3.1. Furthermore it is possible to obtain information about the stationarity, the moments and properties of the tails of this model.

As a motivation to obtain a continuous time GARCH model based on a discrete time GARCH model, we now have a closer look at the volatility term of the discrete time GARCH model. Thus we continue with a modification of the volatility term of the discrete time GARCH model. From equation (3.2) we obtain recursively

$$\begin{aligned}\sigma_n^2 &= \beta + \lambda Y_{n-1}^2 + \delta\sigma_{n-1}^2 \\ &= \beta + (\delta + \lambda\epsilon_{n-1}^2)\sigma_{n-1}^2 \\ &= \beta \sum_{i=0}^{n-1} \prod_{j=i+1}^{n-1} (\delta + \lambda\epsilon_j^2) + \sigma_0^2 \prod_{j=0}^{n-1} (\delta + \lambda\epsilon_j^2), \end{aligned} \tag{3.3}$$

(cf. Klüppelberg et al. (2004, Eq.(2.4) and Eq.(2.5)) and Zapp (2004, p.11)). Then by writing the sums of (3.3) as an integral, we can re-write equation (3.3) as

$$\begin{aligned}\sigma_n^2 &= \beta \int_0^n \exp\left(\sum_{j=\lfloor s\rfloor+1}^{n-1} \log(\delta + \lambda\epsilon_j^2)\right) ds + \sigma_0^2 \exp\left(\sum_{j=0}^{n-1} \log(\delta + \lambda\epsilon_j^2)\right) \\ &= \exp\left(\sum_{j=0}^{n-1} \log(\delta + \lambda\epsilon_j^2)\right) \left[\sigma_0^2 + \beta \int_0^n \exp\left(-\sum_{j=0}^{\lfloor s\rfloor} \log(\delta + \lambda\epsilon_j^2)\right) ds\right] \\ &= \exp\left(\sum_{j=0}^{n-1} \log(\delta(1 + \frac{\lambda}{\delta}\epsilon_j^2))\right) \left[\sigma_0^2 + \beta \int_0^n \exp\left(-\sum_{j=0}^{\lfloor s\rfloor} \log(\delta(1 + \frac{\lambda}{\delta}\epsilon_j^2))\right) ds\right] \end{aligned}$$

[8] cf. Klüppelberg et al. (2011, p.4,5).
[9] cf. the abstract of Fasen et al. (2006).

$$
\begin{aligned}
&= \exp\left(n \underbrace{\log(\delta)}_{=-\eta} + \sum_{j=0}^{n-1} \log(1 + \underbrace{\frac{\lambda}{\delta}}_{=\varphi} \epsilon_j^2) \right) \\
&\qquad \times \left[\sigma_0^2 + \beta \int_0^n \exp\left(\underbrace{-\log(\delta)}_{=\eta} (\lfloor s \rfloor + 1) - \sum_{j=0}^{\lfloor s \rfloor} \log(1 + \underbrace{\frac{\lambda}{\delta}}_{=\varphi} \epsilon_j^2) \right) ds \right] \\
&= \exp\left(-n\eta + \sum_{j=0}^{n-1} \log(1 + \varphi \epsilon_j^2) \right) \\
&\qquad \times \left[\sigma_0^2 + \beta \int_0^n \exp\left(\eta(\lfloor s \rfloor + 1) - \sum_{j=0}^{\lfloor s \rfloor} \log(1 + \varphi \epsilon_j^2) \right) ds \right] \qquad (3.4)
\end{aligned}
$$

This representation gives the opportunity to replace the innovations ϵ_j by the jumps ΔL_t as defined in Chapter 2.

Regarding the equation (3.4) we define the càdlàg process $(X_t)_{t \geq 0}$ as

$$X_t = \eta t - \sum_{0 < s \leq t} \log(1 + \varphi(\Delta L_s)^2), \; t \geq 0, \qquad (3.5)$$

with $\eta, \varphi > 0$.[10] And moreover let the left-continuous volatility process be defined by

$$\sigma_{t-}^2 = \left(\beta \int_0^t e^{X_s} ds + \sigma_0^2 \right) e^{-X_{t-}}, \; t \geq 0, \qquad (3.6)$$

assuming that $\beta > 0$ and $\sigma_0 < \infty$ almost surely (a.s.) and independent of $(L_t)_{t \geq 0}$. Inserting X_t in (3.6) we can see the analogy to (3.4):

$$
\begin{aligned}
\sigma_{t-}^2 = &\left(\beta \int_0^t \exp\left(\eta s - \sum_{0 < u \leq s} \log(1 + \varphi(\Delta L_u)^2) \right) ds + \sigma_0^2 \right) \\
&\times \exp\left(-\eta t + \sum_{0 < u \leq t-} \log(1 + \varphi(\Delta L_u)^2) \right).
\end{aligned}
$$

As stated in Klüppelberg et al. (2011, p.6)[11] the *integrated* **continuous time GARCH process** $(G_t)_{t \geq 0}$ can then be defined as the SDE

$$dG_t = \sigma_{t-} dL_t, \; G_0 = 0, \qquad (3.7)$$

where the volatility process $(\sigma_t^2)_{t \geq 0}$ satisfies

$$d\sigma_t^2 = (\beta - \eta \sigma_{t-}^2) dt + \varphi \sigma_{t-}^2 d[L, L]_t^d. \qquad (3.8)$$

[10] Compared to Equation (3.3) of (Klüppelberg et al. (2004, p.7)) we used the following reparameterization: $\eta = -\log \delta$ and $\varphi = \lambda / \delta$ with $0 < \delta < 1$ and $\lambda \geq 0$.

[11] Originally the definition is given in (Klüppelberg et al. (2004, p.7)). But here we use σ_{t-} instead of σ_t.

With the left limit σ_{t-}^2 we denote the volatility at time t without the jump at time t. And σ_t^2 is the volatility including the jump at time t accordingly. Furthermore $[L,L]_t^d = \sum_{0<s<t}(\Delta L_s)^2$, $t \geq 0$, is the discrete part of the quadratic variation process[12] $([L,L]_t)_{t\geq 0}$ of the Lévy process L. The process $(G_t)_{t\geq 0}$ is càdlàg and jumps at the same time as L does. The size of its jump at time t is denoted by $\Delta G_t = \sigma_{t-}\Delta L_t$, $t \geq 0$.

Regarding the special structure of $(X_t)_{t\geq 0}$ we give Proposition 3.2, from which we see that $(X_t)_{t\geq 0}$ is a Lévy process according to its definition in (3.5), and has no positive jumps.

Proposition 3.2 *(Klüppelberg et al. (2004, Proposition 3.1))*
The process $(X_t)_{t\geq 0}$ is a spectrally negative Lévy process of bounded variation with drift $\gamma_{X,0} = -\log\delta = \eta$, Gaussian component $\sigma_X^2 = 0$, and Lévy measure ν_X given by

$$\nu_X([0,\infty)) = 0 \quad \textit{and} \quad \nu_X = ((-\infty,-x]) = \nu_L(\{y \in \mathbb{R} : |y| \geq \sqrt{(e^x-1)\delta/\varphi}\}), \quad x > 0.$$

Furthermore, according to Klüppelberg et al. (2011, p.7), returns over time intervals of fixed length $r > 0$ are denoted by

$$G_t^{(r)} := G_t - G_{t-r} = \int_{(t-r,t]} \sigma_{s-} dL_s, \qquad t \geq r, \tag{3.9}$$

so that $(G_{ri}^{(r)})_{i\in\mathbb{N}}$ describes an equidistant sequence of non-overlapping returns. Calculating the corresponding quantity for the volatility yields

$$\begin{aligned}\sigma_{ri}^{2(r)} &:= \sigma_{ri}^2 - \sigma_{r(i-1)}^2 = \int_{(r(i-1),ri]} ((\beta - \eta\sigma_s^2)ds + \varphi\sigma_{s-}^2 d[L,L]_s^d) \\ &= \beta r - \eta\int_{(r(i-1),ri]} \sigma_s^2 ds + \varphi\int_{(r(i-1),ri]} \sigma_{s-}^2 d[L,L]_s^d \end{aligned} \tag{3.10}$$

It is worth noting that the stochastic process

$$\int_{(0,t]} \sigma_{s-}^2 d[L,L]_s^d = \sum_{0<s\leq t} \sigma_{s-}^2 (\Delta L_s)^2, \qquad t \geq 0, \tag{3.11}$$

is the discrete part of the quadratic variation $[G,G]_t$ of G, which satisfies

$$[G,G]_t = \int_0^t \sigma_{s-}^2 d[L,L]_s.$$

Thus $\int_{(r(i-1),ri]} \sigma_{s-}^2 d[L,L]_s^d$ in (3.10) corresponds to the jump part of the quadratic variation of G accumulated during $(r(i-1),ri]$.

Now we are going to have a look at some stationarity properties of the volatility process $(\sigma_t^2)_{t\geq 0}$. As reference see Theorem 3.1, Theorem 3.2 and Corollary 3.1 of Klüppelberg et al. (2004).

[12] For a definition of quadratic variation, see for example Protter (2005, p.277).

Theorem 3.3 *(Klüppelberg et al. (2004, Theorem 3.1))*
Suppose

$$\int_{\mathbb{R}} log(1+\frac{\lambda}{\delta}y^2)\nu_L(dy) < -\log\delta \tag{3.12}$$

(which, since $\delta > 0$, incorporates the requirement that the integral be finite). Then $\sigma_t^2 \xrightarrow{D} \sigma_\infty^2$, as $t \to \infty$, for a finite random variable σ_∞ satisfying

$$\sigma_\infty^2 \stackrel{D}{=} \beta \int_0^\infty e^{-X_t} dt.$$

Conversely, if (3.12) *does not hold, then $\sigma_t^2 \xrightarrow{P} \infty$ as $t \to \infty$.*[13]

Moreover, the next Theorem states that $(\sigma_t^2)_{t\geq 0}$ is Markovian and further that, if the process is started at $\sigma_0^2 \stackrel{D}{=} \sigma_\infty^2$, it is strictly stationary.

Theorem 3.4 *(Klüppelberg et al. (2004, Theorem 3.2))*
The volatility process $(\sigma_t^2)_{t\geq 0}$, as given in (3.6)*, is a time homogeneous Markov process. Moreover, if the limit variable σ_∞^2 in Theorem 3.3 exists and $\sigma_0^2 \stackrel{D}{=} \sigma_\infty^2$, independent of $(L_t)_{t\geq 0}$, then $(\sigma_t^2)_{t\geq 0}$ is strictly stationary.*

For the process $G_t = \int_0^t \sigma_{s-} dL_s$, $t \geq 0$, note that for any $0 \leq y < t$,

$$G_t = G_y + \int_{y+}^t \sigma_{s-} dL_s, \qquad t \geq 0.$$

Here $(\sigma_s)_{y<s\leq t}$ depends on the past until time y only through σ_y, and the integrator is independent of this past. From Theorem 3.4 we thus obtain:

Corollary 3.5 *(Klüppelberg et al. (2004, Corollary 3.1))*
The bivariate process $(\sigma_t, G_t)_{t\geq 0}$ is Markovian. If $(\sigma_t^2)_{t\geq 0}$ is the stationary version of the process with $\sigma_0^2 \stackrel{D}{=} \sigma_\infty^2$, then $(G_t)_{t\geq 0}$ is a process with stationary increments, i.e. the increment process $(G_t^{(r)})_{t\geq 0}$, see (3.9)*, is stationary for each fixed $r > 0$.*

3.1.1 Second order properties of the volatility process

According to Klüppelberg et al. (2004, Section 4) it is possible to derive moments and the autocorrelation function of the stochastic volatility process $(\sigma_t^2)_{t\geq 0}$. We will need the Laplace transform which is defined as $\mathbb{E}(e^{-sX_t}) = e^{t\Psi(s)}$.

Lemma 3.6 *(Klüppelberg et al. (2004, Lemma 4.1))*
Keep $c > 0$ throughout.

[13] Here "$\xrightarrow{D}$" means "convergence in distribution" and "$\xrightarrow{P}$" means "convergence in probability". Analogously "$\stackrel{D}{=}$" denotes "equal in distribution".

(a) *Let $\lambda > 0$. Then the Laplace transform $\mathbb{E}(e^{-cX_t})$ of X_t at c is finite for some $t > 0$, or, equivalently, for all $t > 0$, if and only if $\mathbb{E}(L_1^{2c}) < \infty$.*

(b) *When $\mathbb{E}(e^{-cX_1}) < \infty$, define $\Psi(c) = \Psi_X(c) = \log \mathbb{E}(e^{-cX_1})$. Then $|\Psi(c)| < \infty$, $\mathbb{E}(e^{-cX_t}) = e^{t\Psi(c)}$, and*

$$\Psi(c) = c \log \delta + \int_{\mathbb{R}} ((1 + (\lambda/\delta)y^2)^c - 1)\nu_L(dy). \tag{3.13}$$

(c) *If $\mathbb{E}(L_1^2) < \infty$ and $\Psi(1) < 0$, then (3.12) holds, and σ_t^2 converges in distribution to a finite random variable.*

(d) *If $\Psi(c) < 0$ for some $c > 0$, then $\Psi(d) < 0$ for all $0 < d < c$.*

We will use the Laplace exponent $\Psi(c)$ in (3.13) denoted with the parameters $\eta := -\log \delta$ and $\varphi := \lambda/\delta$ as follows.

$$\Psi(c) = -\eta c + \int_{\mathbb{R}} ((1 + \varphi y^2)^c - 1)\nu_L(dy), \qquad c > 0. \tag{3.14}$$

From Lemma 3.6 (a) we know that $\mathbb{E}(e^{-cX_t}) < \infty$ for all $t > 0$, if and only if $\mathbb{E}(L_1^{2c}) < \infty$, which is equivalent to a finite integral in (3.14). Assuming $\mathbb{E}(L_1^{2c}) < \infty$ and $\Psi(c) < 0$ for some $c > 0$, Klüppelberg et al. (2006) showed in Section 4.1, Lemma 1 (d) that then there exists a stationary version of the volatility process. In the next Proposition Klüppelberg et al. (2004) calculated the first two moments and the autocovariance function of $(\sigma_t^2)_{t\geq 0}$ in terms of the Laplace exponent Ψ. Furthermore they showed that the autocovariance function decreases exponentially fast with the lag.

Proposition 3.7 *(Klüppelberg et al. (2004, Proposition 4.1))*
Let $\lambda > 0$, $t > 0$, $h \geq 0$.

(a) *$\mathbb{E}(\sigma_t^2) < \infty$ if and only if $\mathbb{E}(L_1^2) < \infty$ and $\mathbb{E}(\sigma_0^2) < \infty$. If this is, then*

$$\mathbb{E}(\sigma_t^2) = \frac{\beta}{-\Psi(1)} + \left(\mathbb{E}(\sigma_0^2) + \frac{\beta}{\Psi(1)}\right) e^{t\Psi(1)}, \tag{3.15}$$

where for $\Psi(1) = 0$ the righthand side has to be interpreted as its limit as $\Psi(1) \to 0$, i.e. $\mathbb{E}(\sigma_t^2) = \beta t + \mathbb{E}(\sigma_0^2)$.

(b) *$\mathbb{E}(\sigma_t^4) < \infty$ if and only if $\mathbb{E}(L_1^4) < \infty$ and $\mathbb{E}(\sigma_0^4) < \infty$. In that case, the following formulae hold (with a suitable interpretation as a limit if some of the denominators are zero):*

$$\mathbb{E}(\sigma_t^4) = \frac{2\beta^2}{\Psi(1)\Psi(2)} + \frac{2\beta^2}{\Psi(2) - \Psi(1)}\left(\frac{e^{t\Psi(2)}}{\Psi(2)} - \frac{e^{t\Psi(1)}}{\Psi(1)}\right) \tag{3.16}$$

$$+ 2\beta\mathbb{E}(\sigma_0^2)\left(\frac{e^{t\Psi(2)}}{\Psi(2)} - \frac{e^{t\Psi(1)}}{\Psi(1)}\right) + \mathbb{E}(\sigma_0^4)e^{t\Psi(2)}, \tag{3.17}$$

$$\mathbb{C}ov(\sigma_t^2, \sigma_{t+h}^2) = \mathbb{V}ar(\sigma_t^2)e^{h\Psi(1)}. \tag{3.18}$$

In the following we consider the stationary version of a volatility process.
The k-th moment of σ_∞^2 is finite if and only if $\mathbb{E}(L_1^{2k}) < \infty$ and $\Psi(k) < 0$, $k \in \mathbb{N}$, see Klüppelberg et al. (2004), Proposition 4.2. Therefore it is possible to obtain the mean and second moment of σ_∞^2. The first two moments of the stationary process (with $\sigma_0^2 \stackrel{D}{=} \sigma_\infty^2$), then are

$$\mathbb{E}(\sigma_\infty^2) = \frac{\beta}{-\Psi(1)}, \tag{3.19}$$

$$\mathbb{E}(\sigma_\infty^4) = \frac{2\beta^2}{\Psi(1)\Psi(2)}, \tag{3.20}$$

provided $\mathbb{E}(L_1^{2k}) < \infty$ and $\Psi(k) < 0$, with $k = 1$ for (3.19) and $k = 2$ for (3.20), respectively.
And the autocovariance function of the stationary volatility process satisfies

$$\mathbb{C}\mathrm{ov}(\sigma_t^2, \sigma_{t+h}^2) = \beta^2 \left(\frac{2}{\Psi(1)\Psi(2)} - \frac{1}{\Psi^2(1)} \right) e^{h\Psi(1)}, \quad t, h \geq 0,$$

provided $\mathbb{E}(L_1^4) < \infty$ and $\Psi(2) < \infty$, cf. Klüppelberg et al. (2004, Corollary 4.1) and Klüppelberg et al. (2006, Section 4.1, Theorem 3). Due to the stationarity of the volatility process, we have for $\mathbb{E}(L_1^4) < \infty$ and $\Psi(2) < 0$

$$\mathbb{E}(\sigma_t^2) = \frac{\beta}{|\Psi(1)|} \quad \text{and} \quad \mathbb{E}(\sigma_t^4) = \frac{2\beta^2}{|\Psi(1)\Psi(2)|},$$
$$\mathbb{C}\mathrm{ov}(\sigma_t^2, \sigma_{t+h}^2) = \beta^2 \left(\frac{2}{\Psi(1)\Psi(2)} - \frac{1}{\Psi^2(1)} \right) e^{-h|\Psi(1)|} = \mathbb{V}\mathrm{ar}(\sigma_t^2) e^{-h|\Psi(1)|}, \quad t, h \geq 0.$$

Econometric literature suggests that volatility is quite persistent, which would imply that $e^{-|\Psi(1)|}$ is close to 1; i.e. $\Psi(1) < 0$ near 0. This should be kept in mind, when estimating the model parameters, see Chapter 3.2.[14]

3.1.2 Second order properties of the continuous GARCH process

In the following we consider second order properties of the increments of $(G_t)_{t\geq 0}$. Recall the notation of $(G_t^{(r)})_{t\geq 0}$, which is denoted as in (3.9),

$$G_t^{(r)} = G_t - G_{t-r}, \quad t \geq r > 0.$$

We still assume to have a stationary version of the volatility process. By Corollary 3.5 this implies strict stationarity of $(G_t^{(r)})_{t\geq 0}$. In Proposition 3.8 we introduce the moments of this process, which are independent of t by stationarity.

Proposition 3.8 *(Haug et al. (2007, Proposition 2.1))*
Suppose that the Lévy process $(L_t)_{t\geq 0}$ has finite variance (i.e. $\mathbb{E}(L_1^2) < \infty$) and zero mean

[14] cf. Haug et al. (2007, Remark 2.3).

(i.e. $\mathbb{E}(L_1) = 0$), and that $\Psi(1) < 0$. Let $(\sigma_t^2)_{t\geq 0}$ be the stationary volatility process, so that $(G_t)_{t\geq 0}$ has stationary increments. Then $\mathbb{E}(G_t^2) < \infty$ for all $t \geq 0$, and for every $t, h \geq r > 0$ it holds

$$\mathbb{E}(G_t^{(r)}) = 0, \qquad \mathbb{E}(G_t^{(r)})^2 = \frac{\beta r}{|\Psi(1)|}\mathbb{E}(L_1^2), \qquad \mathbb{C}ov(G_t^{(r)}, G_{t+h}^{(r)}) = 0. \tag{3.21}$$

If further $\mathbb{E}(L_1^4) < \infty$ and $\Psi(2) < 0$, then $\mathbb{E}(G_t^4) < \infty$ for all $t \geq 0$ and, if additionally the Lévy measure ν_L of L is such that $\int_{\mathbb{R}} x^3 \nu_L(dx) = 0$, then it holds for every $t, h \geq r > 0$

$$\begin{aligned}\mathbb{E}(G_t^{(r)})^4 = {} & 6\mathbb{E}(L_1^2)\frac{\beta^2}{\Psi(1)^2}(2\eta\varphi^{-1} + 2\sigma_L^2 - \mathbb{E}(L_1^2))\left(\frac{2}{|\Psi(2)|} - \frac{1}{|\Psi(1)|}\right)\left(r - \frac{1 - e^{-r|\Psi(1)|}}{|\Psi(1)|}\right) \\ & + \frac{2\beta^2}{\varphi^2}\left(\frac{2}{|\Psi(2)|} - \frac{1}{|\Psi(1)|}\right)r + 3\frac{\beta^2}{\Psi(1)^2}(\mathbb{E}(L_1^2))^2 r^2 \end{aligned} \tag{3.22}$$

and

$$\begin{aligned}\mathbb{C}ov((G_t^{(r)}))^2, (G_{t+h}^{(r)})^2) = {} & \mathbb{E}(L_1^2)\frac{\beta^2}{|\Psi(1)|^3}(2\eta\varphi^{-1} + 2\sigma_L^2 - \mathbb{E}(L_1^2))\left(\frac{2}{|\Psi(2)|} - \frac{1}{|\Psi(1)|}\right) \\ & \times \left(1 - e^{-r|\Psi(1)|}\right)\left(e^{r|\Psi(1)|} - 1\right)e^{-h|\Psi(1)|} > 0. \end{aligned} \tag{3.23}$$

For a proof of this proposition, we refer to Appendix A of Haug et al. (2007). Moreover Proposition 3.8 tells us that the returns are uncorrelated, while the squared returns are correlated. The autocorrelation function of the squared returns decays exponentially.[15]

3.1.3 Examples of COGARCH$(1,1)$ processes

In this chapter we have a closer look on two examples of COGARCH$(1,1)$ processes. The first one will be a COGARCH$(1,1)$ process driven by a compound Poisson process. The other one will be a COGARCH$(1,1)$ process driven by a Variance Gamma process. Thus, we will consider Lévy processes with finite and infinite activity, see Remark 2.10. The following is based on Chapter 2.4 of Haug (2006).

Compound Poisson COGARCH$(1,1)$

A compound Process L has been introduced in Example 2.4 and is of the form

$$L_t = \sum_{i=1}^{N_t} Y_i, \qquad t \geq 0,$$

where $N = (N_t)_{t\geq 0}$ is a Poisson process with jump rate $\lambda > 0$, and $(Y_i)_{i\in\mathbb{N}}$ are i.i.d. random variables, independent of N. There is no Brownian motion component, thus

[15] cf. Klüppelberg et al. (2006, Section 4.2).

$\sigma_L^2 = 0$. We introduce a random variable Y with the same distribution function as Y_i, denoted by F_Y. The Lèvy measure of L is $\nu_L(dx) = \lambda F_Y(dx)$. Using this information, we can calculate the Laplace exponent from (3.14). We get

$$\Psi(c) = -\eta s + \lambda \int_{\mathbb{R}} ((1+\varphi y^2)^c - 1) F_Y(dy),$$

and thus,

$$\begin{aligned}
\Psi(1) &= -\eta + \lambda \int_{\mathbb{R}} ((1+\varphi y^2)^1 - 1) F_Y(dy) \\
&= -\eta + \lambda\varphi \int_{\mathbb{R}} y^2 F_Y(dy) \\
&= -\eta + \varphi\lambda \mathbb{E}(Y^2) \\
\Psi(2) &= -2\eta + \lambda \int_{\mathbb{R}} ((1+\varphi y^2)^2 - 1) F_Y(dy) \\
&= -2\eta + \lambda \int_{\mathbb{R}} ((1+2\varphi y^2 + \varphi^2 y^4) - 1) F_Y(dy) \\
&= -2\eta + 2\lambda\varphi \int_{\mathbb{R}} y^2 F_Y(dy) + \lambda\varphi^2 \int_{\mathbb{R}} y^4 F_Y(dy) \\
&= -2\eta + 2\lambda\varphi \mathbb{E}(Y^2) + \lambda\varphi^2 \mathbb{E}(Y^4).
\end{aligned}$$

If we assume $\mathbb{E}(L_1) = 0$ and $\mathbb{V}\mathrm{ar}(L_1) = \mathbb{E}(L_1^2) = 1$, we must have $\mathbb{E}(Y^2) = 1/\lambda$. Inserting this in $\Psi(1)$ and $\Psi(2)$ yields

$$\Psi(1) = -\eta + \varphi, \tag{3.24}$$

$$\Psi(2) = 2(\varphi - \eta) + \varphi^2 \mathbb{E}(Y^4)/\mathbb{E}(Y^2). \tag{3.25}$$

In Chapter 3.2 we will need the Laplace exponents $\Psi(1)$ and $\Psi(2)$ in order to check if the chosen parameters fulfill some conditions, see conditions (H1) to (H5) in Chapter 3.2.

In Figure 3.1 we plotted simulated sample paths for the time interval $[0, 1000]$ of the compound Poisson COGARCH$(1,1)$ process G, the log-return data $G^{(1)}$, the volatility process σ^2 and the driving Lévy process L with jump rate $\lambda = 1$ and $\mathcal{N}(0,1)$ distributed jumps. We used the parameters $\beta = 0.04$, $\eta = 0.053$, $\varphi = 0.038$ and the rate $\lambda = 1$. It can be observed that the sample path of G and L only differ by their jump sizes. Furthermore the property of volatility clustering, which can be observed in real data (see Remark 3.1), can be discovered here.

Moreover it is possible to estimate the jump rate λ from the discretised data $G_i^{(1)}$. With $z(n)$ we denote the number of intervals, where G does not change, i.e. $z(n) = \sum_{i=1}^{n} \mathbb{1}_{\{G_i^{(1)}=0\}}$. This implies that a fine enough observation grid is required.

Proposition 3.9 *(Haug (2006, Proposition 2.4.2))*
Let $(L_t)_{t\geq 0}$ be a compound Poisson process with continuous jump distribution F_Y and jump rate $\lambda > 0$. Then

$$\hat{\lambda}_n := -\log\left(\frac{z(n)}{n}\right) \overset{a.s}{\to} \lambda, \quad n \to \infty,$$

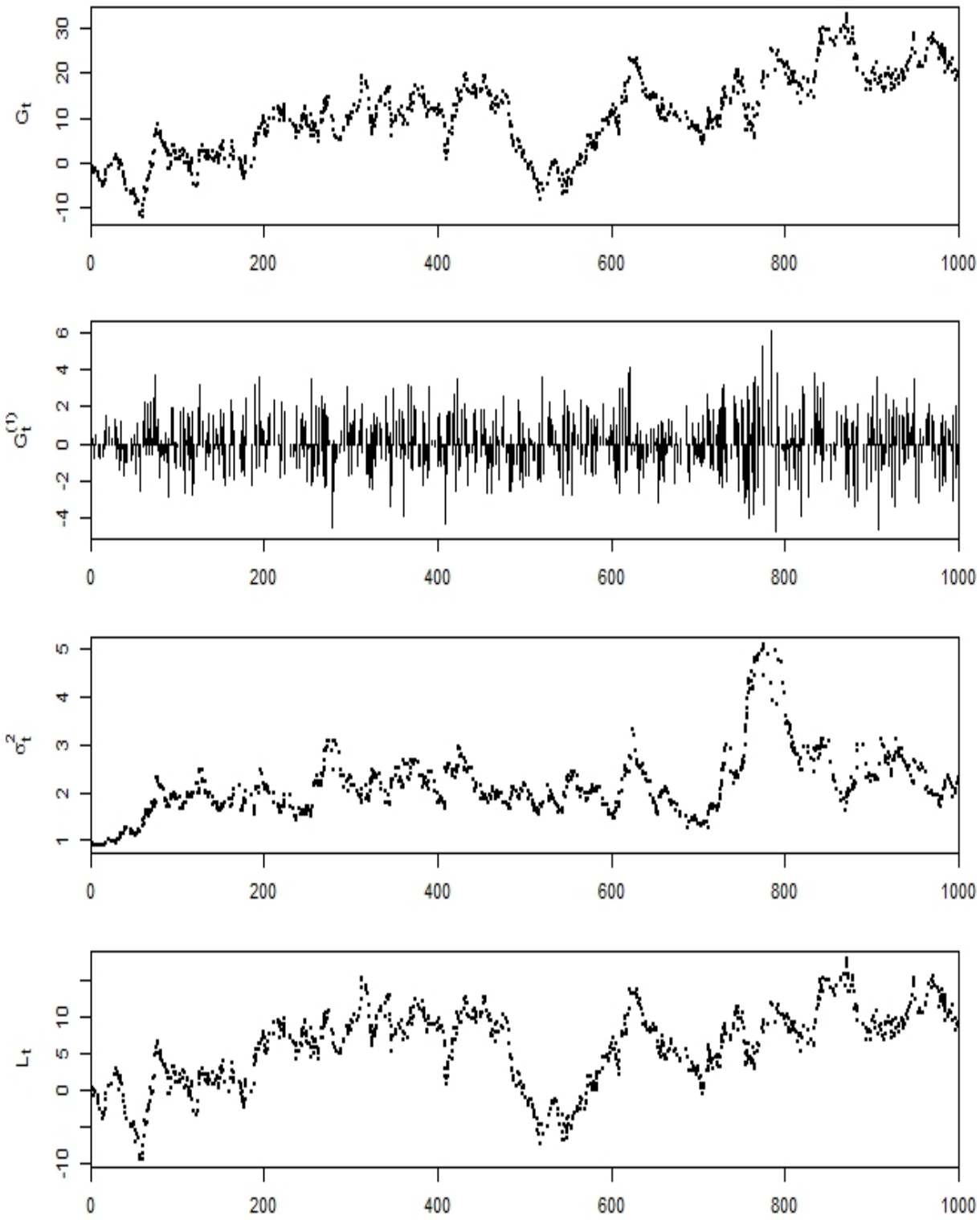


Figure 3.1: Simulation of a compound Poisson driven COGARCH$(1,1)$ process (G_t) with parameters $\beta = 0.04$, $\eta = 0.053$, $\varphi = 0.038$, rate$= 1$ and standard normally distributed jumpsizes (first), log-return process $(G_t^{(1)})$(second), volatility process (σ_t^2) (third) and driving compound Poisson process $(L_t)_{0\leq t\leq 1000}$ with jump rate 1 and $\mathcal{N}(0,1)$-distributed jumps (last).

and

$$\sqrt{n}(\hat{\lambda}_n - \lambda) \xrightarrow{d} \mathcal{N}(0, e^{\lambda}(1 - e^{-\lambda})), \quad n \to \infty.$$

Variance Gamma COGARCH$(1,1)$

The Variance Gamma process has been introduced in Example 2.5. Again the Lévy process has no Brownian component, i.e. $\sigma_L^2 = 0$. Furthermore we assume $\mathbb{E}(L_1) = 0$ and $\mathbb{V}\mathrm{ar}(L_1) = 1$. Thus, with the mean and the variance of the Variance Gamma process, given in (2.5), we can conclude that we have $\theta = 0$ and $\sigma = 1$. Moreover it follows from (2.2) that the characteristic function at time $t \geq 0$ is given by

$$\mathbb{E}(e^{iuL_t}) = \left(1 + \frac{u^2}{2C}\right)^{-tC}.$$

And the Lévy measure ν_L, defined in (2.4), is then of the form

$$\nu_L(dx) = \frac{C}{|x|} \exp\left(-(2C)^{1/2}|x|\right) dx, \quad x \neq 0. \tag{3.26}$$

The Laplace exponent can be calculated by inserting (3.26) in (3.14),

$$\Psi(1) = -\eta + \varphi \underbrace{\int_{\mathbb{R}} x^2 \frac{C}{|x|} e^{-\sqrt{2C}|x|} dx}_{=1} = -\eta + \varphi \tag{3.27}$$

and

$$\begin{aligned}
\Psi(2) &= -2\eta + \int_{\mathbb{R}} ((1 + \varphi x^2)^2 - 1) \frac{C}{|x|} e^{-\sqrt{2C}|x|} dx && (3.28)\\
&= -2\eta + \int_{\mathbb{R}} (2\varphi x^2 + \varphi^2 x^4) \frac{C}{|x|} e^{-\sqrt{2C}|x|} dx && (3.29)\\
&= -2\eta + 2\varphi \underbrace{\int_{\mathbb{R}} x^2 \frac{C}{|x|} e^{-\sqrt{2C}|x|} dx}_{=1} + \varphi^2 \int_{\mathbb{R}} x^4 \frac{C}{|x|} e^{-\sqrt{2C}|x|} dx. && (3.30)
\end{aligned}$$

As the last integral of (3.30) is

$$\begin{aligned}
\int_{\mathbb{R}} x^4 \frac{C}{|x|} e^{-\sqrt{2C}|x|} dx &= C \int_{\mathbb{R}} x^3 e^{-\sqrt{2C}|x|} dx \\
&= 2C \int_0^{\infty} x^3 e^{-\sqrt{2C}|x|} dx = 2C \frac{3!}{\sqrt{2C}^4} = \frac{3}{C},
\end{aligned}$$

we have

$$\Psi(2) = -2\eta + 2\varphi + 3\varphi^2 C^{-1}. \tag{3.31}$$

In Figure 3.2 we plotted simulated sample paths for the time interval $[0, 1000]$ of the Variance Gamma COGARCH$(1,1)$ process G, the log-return process $G^{(1)}$, the volatility process σ^2 and the driving Lévy process L.

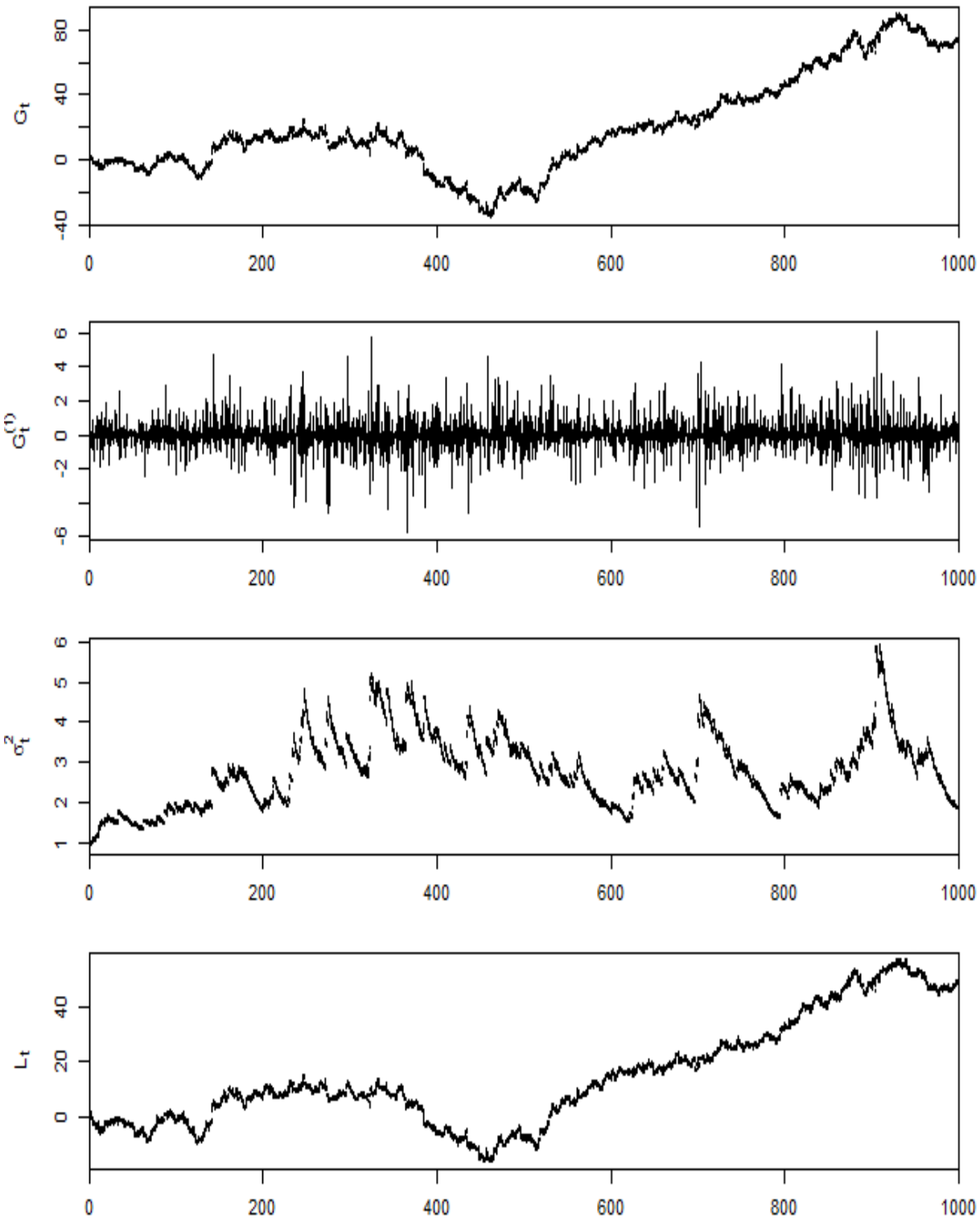


Figure 3.2: Simulation of a Variance Gamma driven COGARCH$(1,1)$ process (G_t) with parameters $\beta = 0.04$, $\eta = 0.053$, $\varphi = 0.038$ (first), log-return process $(G_t^{(1)})$(second), volatility process (σ_t^2) (third) and driving Variance Gamma process $(L_t)_{0\leq t\leq 1000}$ with parameters $\theta = 0$, $\sigma = 1$ and $\tau = 1$ (last).

3.1.4 COGARCH(p,q) Process

We implemented the simulation of a COGARCH process not only for $p = q = 1$ but also for the more general case where $q \geq p \geq 1$. Therefore we give an overview of the extension to COGARCH(p,q) by Brockwell et al. (2006).

Definition 3.10 *(Chadraa (2010, Definition 2.1))*
If p and q are integers such that $q \geq p \geq 1$, $\alpha_0 > 0$, $\alpha_1, \ldots, \alpha_p \in \mathbb{R}$, $\beta_1, \ldots, \beta_q \in \mathbb{R}$, $\alpha_p \neq 0$, $\beta_q \neq 0$ and $\alpha_{p+1} = \cdots = \alpha_q = 0$, we define the $(q \times q)$ matrix B and the vectors $\boldsymbol{a}$ and $\boldsymbol{e}$ by

$$B = \begin{bmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & 1 \\ -\beta_q & -\beta_{q-1} & -\beta_{q-2} & \ldots & -\beta_1 \end{bmatrix},$$

$$\boldsymbol{a} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{q-1} \\ \alpha_q \end{bmatrix}, \boldsymbol{e} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix},$$

with $B := -\beta_1$ if $q = 1$. Then if $L = (L_t)_{t\geq 0}$ is a Lévy process with nontrivial Lévy measure, we define the (left-continuous) volatility process $V = (V_t)_{t\geq 0}$ with parameters B, $\boldsymbol{a}$, α_0 and driving Lévy process L by

$$V_t = \alpha_0 + \boldsymbol{a}'\boldsymbol{Y}_{t-}, \quad , t > 0, \quad V_0 = \alpha_0 + \boldsymbol{a}'\boldsymbol{Y}_0,$$

where the state process $\boldsymbol{Y} = (\boldsymbol{Y}_t)_{t\geq 0}$ is the unique càdlàg solution of the stochastic differential equation

$$d\boldsymbol{Y}_t = B\boldsymbol{Y}_{t-}dt + \boldsymbol{e}(\alpha_0 + \boldsymbol{a}'\boldsymbol{Y}_{t-})d[L,L]^d_t, \tag{3.32}$$

with initial value $\boldsymbol{Y}_0$, independent of the driving Lévy process $(L_t)_{t\geq 0}$. If the process $(V_t)_{t\geq 0}$ is strictly stationary and nonnegative almost surely, we say that $G = (G_t)_{t\geq 0}$ given by

$$dG_t = \sqrt{V_t}dL_t, \quad , t > 0, \quad G_0 = 0,$$

is a COGARCH(p,q) process with parameters B, $\boldsymbol{a}$,α_0 and driving Lévy process L.

From standard theorems on stochastic differential equations (e.g. Protter (2005, Chapter V, Theorem 7)) it follows that there is in fact a unique solution of (3.32) for any starting random vector $\boldsymbol{Y}_0$. The stochastic integrals are interpreted with respect to the filtration $(\mathcal{F}_t)_{t\geq 0}$, which is defined to be the smallest right-continuous filtration such that $\mathcal{F}_0$ contains all the P-null sets of $\mathcal{F}$; $(L_t)_{t\geq 0}$ is adapted and $\boldsymbol{Y}_0$ is $\mathcal{F}_0$-measurable.

The volatility process $(V_t)_{t\geq 0}$ specified in Definiton 3.10 is strictly stationary under the conditions given in Theorem 3.11. Assume that the matrix B can be diagonalized.

Since the only eigenvectors corresponding to the eigenvalue λ_i are constant multiples of $[1, \lambda_i, \lambda_i^2, \ldots, \lambda_i^{q-1}]'$, this is equivalent to the assumption that the eigenvalues of B are distinct. Let S be a matrix such that $S^{-1}BS$ is a diagonal matrix, for example,

$$S = \begin{bmatrix} 1 & \cdots & 1 \\ \lambda_1 & \cdots & \lambda_q \\ \vdots & \cdots & \vdots \\ \lambda_1^{q-1} & \cdots & \lambda_q^{q-1} \end{bmatrix}$$

For this particular choice, $S^{-1}BS = diag(\lambda_1, \ldots, \lambda_q)$, where $diag(\cdot)$ denotes a diagonal matrix with the eigenvalues on its diagonal.

Theorem 3.11 *(Brockwell et al. (2006, Theorem 3.1))*
Let $(\boldsymbol{Y}_t)_{t\geq 0}$ be the state process of the COGARCH(p,q) process with parameters B, $\boldsymbol{a}$, and α_0. Suppose that all the eigenvalues of B are distinct. Let L be a Lévy process with nontrivial Lévy measure ν_L and suppose there is some $r \in [1, \infty)$ such that

$$\int_{\mathbb{R}} \log(1 + ||S^{-1}\boldsymbol{e}\boldsymbol{a}'S||_r y^2) d\nu_L(y) < -\lambda \tag{3.33}$$

for some matrix S such that $S^{-1}BS$ is diagonal. Then $\boldsymbol{Y}_t$ converges in distribution to a finite random variable $\boldsymbol{Y}_\infty$, as $t \to \infty$. It follows that if $\boldsymbol{Y}_0 \stackrel{d}{=} \boldsymbol{Y}_\infty$, then $(\boldsymbol{Y}_t)_{t\geq 0}$ and $(V_t)_{t\geq 0}$ are strictly stationary.

We need some conditions which ensure that V is nonnegative to ensure that G is well-defined. In the following Theorem 3.12 it will be shown that if $\boldsymbol{a}'e^{Bt}\boldsymbol{e} \geq 0$ for all $t \geq 0$ and $\boldsymbol{Y}_0$ is such that V is strictly stationary, then V is nonnegative with probability 1.

Theorem 3.12 *(Brockwell et al. (2006, Theorem 5.1))*

(a) Let $(\boldsymbol{Y}_t)_{t\geq 0}$ be the state vector of a COGARCH(p,q) volatility process $(V_t)_{t\geq 0}$ with parameters B, $\boldsymbol{a}$, and $\alpha_0 > 0$. Let $\gamma \geq -\alpha_0$ be a real constant. Suppose that the following two conditions hold:

$$\boldsymbol{a}'e^{Bt}\boldsymbol{e} \geq 0 \qquad \forall t \geq 0, \tag{3.34}$$

$$\boldsymbol{a}'e^{Bt}\boldsymbol{Y}_0 \geq \gamma \qquad a.s.\ \forall t \geq 0. \tag{3.35}$$

Then for any driving Lévy process, with probability 1,

$$V_t \geq \alpha_0 + \gamma \geq 0 \qquad t \geq 0.$$

Conversely, if either (3.35) *fails or* (3.35) *holds with $\gamma > -\alpha_0$ and* (3.34) *fails, then there exists a driving compound Poisson process L and $t_0 \geq 0$ such that $P(V_{t_0} < 0) > 0$.*

(b) Suppose that all eigenvalues of B are distinct and that (3.33) *and* (3.34) *both hold. Then with probability* 1 *the stationary COGARCH(p,q) volatility process $(V_t)_{t\geq 0}$ satisfies*

$$V_t \geq \alpha_0 > 0 \qquad \forall t \geq 0.$$

For the stationary COGARCH volatility process or for the process with $\mathbf{Y}_0 = \mathbf{0}$*, the condition* (3.34) *alone is sufficient for almost sure nonnegativity.*

As an example, we simulate a COGARCH$(1,3)$ process driven by a compound Poisson process with jump rate 1 and normally distributed jumps with mean 0 and variance 0.74. The chosen coefficients are $\alpha_0 = \alpha_1 = 1$, $\beta_1 = 1.2$, $\beta_2 = 0.48 + \pi^2$ and $\beta_3 = 0.064 + 0.4\pi^2$. We check the eigenvalues of B, they should all be negative, which is satified for: -0.4, $-0.4 + \pi i$ and $-0.4 - \pi i$. In Brockwell et al. (2006, Section 7) they chose the same parameters as we did and checked if the conditions for a nonnegative volatility process are satisfied. For detailed information on those conditions we therefore refer to their paper, especially Proposition 4.1 and Theorem 5.2. In Figure 3.3 we simulated a sample path of G with the parameters specified above *(top)*. The plot in the center of Figure 3.3 shows the differenced process $G_t^{(1)}$ and in the bottom plot the volatility process σ^2 for $t = 1:1000$ is shown.

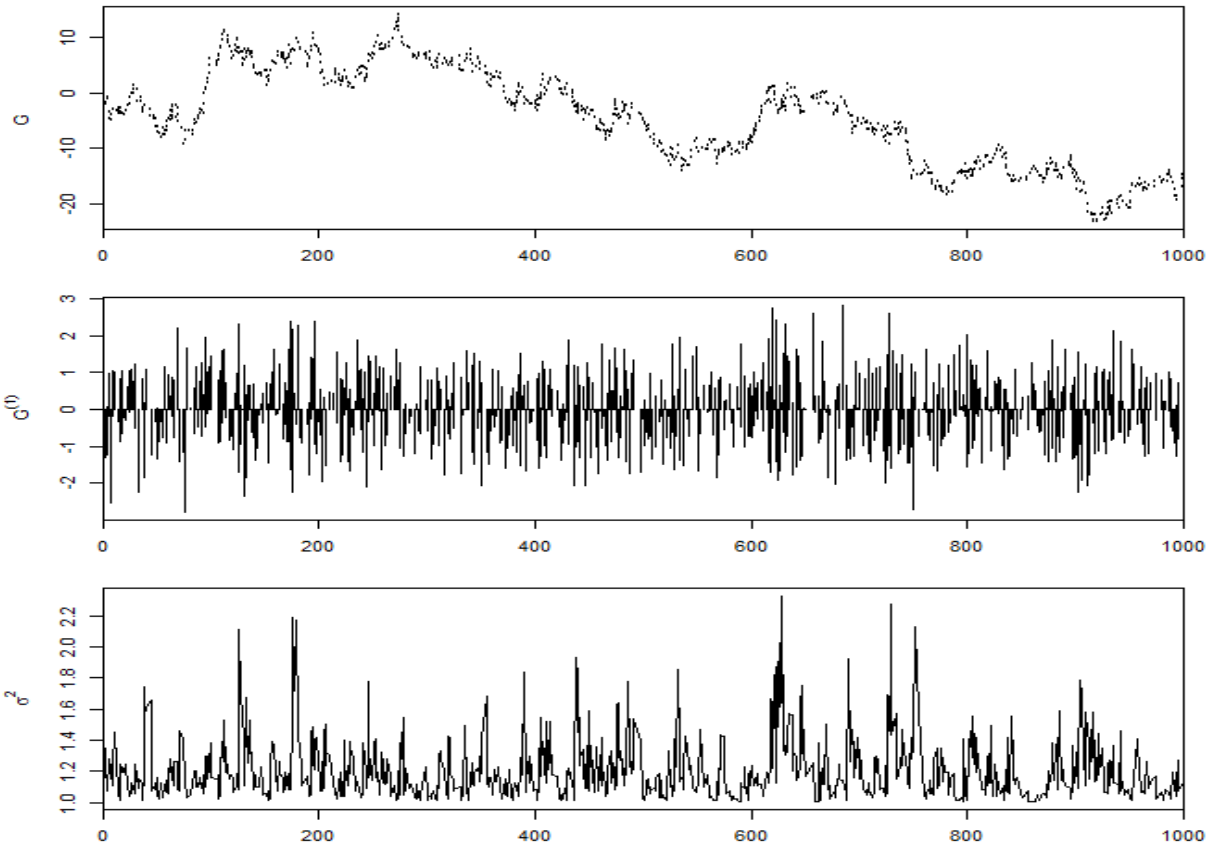


Figure 3.3: Simulated compound Poisson driven COGARCH$(1,3)$ with jump rate 1, normally distributed jumps with mean 0 and variance 0.74, and parameters $\alpha_0 = \alpha_1 = 1$, $\beta_1 = 1.2$, $\beta_2 = 0.48 + \pi^2$ and $\beta_3 = 0.064 + 0.4\pi^2$. Top: process G_t, center: increments $G_t^{(1)}$ and bottom: volatility process $(V_t = \sigma_t^2)$.

3.2 Estimators

The aim of this chapter is the estimation of the model parameters (β, η, φ). We introduce two possible estimation methods. The first is a method of moments estimation. To apply this method we assume to have a sample of equally spaced data. In order to get estimates for unequally spaced data, we introduce a second method which uses a pseudo maximum likelihood (PML) estimation. Throughout this chapter we assume that the driving Lévy process has no Gaussian part, i.e. that $\sigma_L^2 = 0$. This chapter is based on the papers of Haug et al. (2007), Klüppelberg et al. (2011) and Maller et al. (2008).

3.2.1 Moment Estimators

We assume that we have a sample of equally spaced returns. In order to get data equally spaced in time, we need to discretise the continuous time GARCH onto a discrete grid over a finite time interval. The data should be given as described in (3.9).
The model parameters (β, η, φ) can then be estimated by matching the empirical autocorrelations and moments to their theoretical counterparts given in Proposition 3.8. The next result shows that the parameters are identifiable by this estimation procedure for driving Lévy processes L as in Proposition 3.8. We assume throughout that $\mathbb{E}(L_1) = 0$ and $\mathbb{E}(L_1^2) = 1$. For simplicity we set $r = 1$.

Theorem 3.13 *(Haug et al. (2007, Theorem 3.1))*
Suppose $(L_t)_{t\geq 0}$ is a Lévy process such that $\mathbb{E}(L_1) = 0$, $\mathbb{V}ar(L_1) = 1$, the variance σ_L^2 of the Brownian motion component of L is known with $\sigma_L^2 = 0$, $\mathbb{E}(L_1^4) < \infty$ and $\int_{\mathbb{R}} x^3 \nu_L(dx) = 0$. Assume also that $\Psi(2) < 0$, and denote by $(G_i^{(1)})_{i\in\mathbb{N}}$ the stationary increment process of the COGARCH$(1,1)$ process with parameters $\beta, \eta, \varphi > 0$. Let $\mu, \gamma(0), k, p > 0$ be constants such that

$$\begin{aligned} \mathbb{E}((G_i^{(1)})^2) &= \mu, \\ \mathbb{V}ar((G_i^{(1)})^2) &= \gamma(0), \\ \rho(h) = \mathbb{C}orr((G_i^{(1)})^2, (G_{i+h}^{(1)})^2) &= ke^{-hp}, h \in \mathbb{N}. \end{aligned}$$

Define

$$M_1 := \gamma(0) - 2\mu^2 - 6\frac{1 - p - e^{-p}}{(1 - e^p)(1 - e^{-p})} k\gamma(0), \tag{3.36}$$

$$M_2 := \frac{2k\gamma(0)p}{M_1(e^p - 1)(1 - e^{-p})}. \tag{3.37}$$

Then $M_1, M_2 > 0$, and the parameters β, η, φ are uniquely determined by $\mu, \gamma(0), k$ and p, and are given by the formulas

$$\beta = p\mu, \tag{3.38}$$

$$\varphi = p\sqrt{1 + M_2} - p, \tag{3.39}$$

$$\eta = p\sqrt{1+M_2}(1-\sigma_L^2) + p\sigma_L^2 = p + \varphi(1-\sigma_L^2). \tag{3.40}$$

The proof of this Theorem can be found in Haug et al. (2007).

We conclude from (3.38)-(3.40) that our model parameter vector (β, η, φ) is a continuous function of the first two moments μ, $\gamma(0)$ and the parameters of the autocorrelation function p and k. Hence, by continuity, consistency of the moment estimators will immediately imply consistency of the corresponding plug-in estimates for (β, η, φ).

Estimation Algorithm

As a next step we present an estimation algorithm. In Chapter 6 we are going to use this algorithm in order to implement a program for estimating the model parameters with the R-Code 6.10. Furthermore we are going to conduct a simluation study in 3.3. The parameters are estimated under following assumptions:

(H1) We have in time equally spaced observations G_i, $i = 0, \ldots, n$, of the integrated COGARCH as defined and parameterised in (3.7) and (3.8), assumed to be in its stationary regime. This yields return data $G_i^{(1)} = G_i - G_{i-1}$, $i = 1, \ldots, n$.

(H2) $\mathbb{E}(L_1) = 0$ and $\mathbb{E}(L_1^2) = 1$, i.e. $(\sigma_t^2)_{t\geq 0}$ can be interpreted as the volatility.

(H3) The driving Lévy process has no Gaussian part, i.e. $\sigma_L^2 = 0$.

(H4) $\int_{\mathbb{R}} x^3 \nu_L(dx) = 0$, $\mathbb{E}(L_1^4) < \infty$ and $\Psi(2) < 0$. This assumptions ensure that the moment $\mathbb{E}(\sigma_t^4)$ of the stationary volatility process exists. (We know that the moment $\mathbb{E}(\sigma_t^{2k})$ of the stationary volatility process exists if and only if $\mathbb{E}(L_1^{2k}) < \infty$ and $\Psi(k) < \infty$, see Proposition 3.7.)

We define the parameter vectors $\theta := (p, k)$ and $\vartheta := (\beta, \eta, \varphi)$, where k and p are as in Theorem 3.13, and proceed as follows.

(1) Calculate the moment estimator $\hat{\mu}_n$ of μ as

$$\hat{\mu}_n := \frac{1}{n}\sum_{i=1}^{n} (G_i^{(1)})^2,$$

and for fixed $d \geq 2$ the empirical autocovariances $\hat{\gamma}_n := (\hat{\gamma}_n(0), \hat{\gamma}_n(1), \ldots, \hat{\gamma}_n(d))^\top$ as

$$\hat{\gamma}_n(h) := \frac{1}{n}\sum_{i=1}^{n-h} ((G_{i+h}^{(1)})^2 - \hat{\mu}_n)((G_i^{(1)})^2 - \hat{\mu}_n), \qquad h = 0, \ldots, d.$$

(2) Compute the empirical autocorrelations $\hat{\rho}_n := (\hat{\gamma}_n(1)/\hat{\gamma}_n(0), \ldots, \hat{\gamma}_n(d)/\hat{\gamma}_n(0))^\top$.

(3) For fixed $d \geq 2$ define the mapping $H : \mathbb{R}_+^{d+2} \to \mathbb{R}$ by

$$H(\hat{\rho}_n, \theta) := \sum_{h=1}^{d} (\log(\hat{\rho}_n(h)) - \log(k) + ph)^2.$$

Compute the least square estimator

$$\hat{\theta}_n := \operatorname{argmin}_{\theta \in \mathbb{R}_+^2} H(\hat{\rho}_n, \theta).$$

(4) Define the mapping $J : \mathbb{R}_+^4 \to [0, \infty)^3$ by

$$J(\mu, \gamma(0), \theta) := \begin{cases} (p\mu, p\sqrt{1 + M_2} - p, p\sqrt{1 + M_2}), & \text{if } p, M_2 > 0, \\ (0, 0, 0), & \text{otherwise}, \end{cases}$$

where M_2 is defined as in Theorem 3.13. Finally compute the estimator

$$\hat{\vartheta}_n = J(\hat{\mu}_n, \hat{\gamma}_n(0), \hat{\theta}_n).$$

As stated in Haug (2006), Remark 2.2.1, we know that under the chosen conditions, $\rho(h) > 0$ for all $h \in \mathbb{N}$. Also, M_1 and M_2 are strictly positive. This does not imply that the corresponding empirical estimates are strictly positive. From Theorem 3.16 it will follow that the above estimators are strongly consistent. Therefore, all sufficiently large sample paths will have strictly positive empirical estimates and all parameter estimates are well-defined.
Furthermore, for a stationary model the parameter p has to be strictly positive. Computing the unrestricted minimum of $H(\hat{\boldsymbol{\rho}}_n, \boldsymbol{\theta})$ leads to

$$\hat{p}_n^* := -\frac{\sum_{h=1}^{d} \left(\log(\hat{\rho}_n(h)) - \overline{\log(\hat{\rho}_n)}\right) \left(h - \frac{d+1}{2}\right)}{\sum_{h=1}^{d} \left(h - \frac{d+1}{2}\right)^2} \tag{3.41}$$

$$\hat{k}_n := \exp\left\{\overline{\log(\hat{\rho}_n)} + \frac{d+1}{2}\hat{p}_n^*\right\}, \tag{3.42}$$

with $\overline{\log(\hat{\rho}_n)} := \frac{1}{d}\sum_{h=1}^{d} \log(\hat{\rho}_n(h))$ and $\hat{p}_n^*$ may be negative. Therefore we define the estimator of p as

$$\hat{p}_n := \max\{\hat{p}_n^*, 0\} \tag{3.43}$$

and take $\hat{p}_n = 0$ as an indication that the data is non-stationary. Defining the mapping $S : \mathbb{R}_+^{d+1} \to \mathbb{R}_+^2$ by the equations and noting that $\hat{\rho}_n(h) = \hat{\gamma}_n(h)/\hat{\gamma}_n(0)$ presents the least squares estimator $\hat{\boldsymbol{\theta}}_n := (\hat{k}_n, \hat{p}_n)$ as a function of $\hat{\boldsymbol{\gamma}}_n$:

$$\hat{\boldsymbol{\theta}}_n = S(\hat{\boldsymbol{\gamma}}_n). \tag{3.44}$$

Asymptotic Properties

To conclude this chapter about moment estimators, we review some asymptotic properties of the moment estimators. This part will be a summary of Haug (2006), Chapter 2.3.

The following Corollary will imply strong consistency of the estimator $\hat{\boldsymbol{\vartheta}}_n$, as stated in (3.45) below.

Corollary 3.14 *(Haug (2006, Corollary 2.3.6))*
Suppose that $(L_t)_{t\geq 0}$ is such that $\mathbb{E}(L_1^4) < \infty$ and the parameters of the COGARCH$(1,1)$ process satisfy $\Psi(2) < 0$. Let $\sigma^2 := (\sigma_t^2)_{t\geq 0}$ be the strictly stationary volatility process given as solution to (3.8)*. Then we obtain for $n \to \infty$*

$$\hat{\mu}_n \overset{a.s}{=} \mu, \quad \hat{\gamma}_n \overset{a.s}{=} \gamma.$$

In order to obtain asymptotic normality of the empirical estimates we give Proposition 3.15. This Proposition includes an additional assumption which requires for G a finite moment of higher order than the eighth, which is the case if $\mathbb{E}(L_1^{8+\delta}) < \infty$ and the $(4+\delta)$-moment of the volatility is finite, i.e. $\Psi(4+\delta) < 0$.

Proposition 3.15 *(Haug (2006, Proposition 2.3.7))*
Suppose that $(L_t)_{t\geq 0}$ is such that $\mathbb{E}(L_1^4) < \infty$ and the parameters of the COGARCH$(1,1)$ process satisfy $\Psi(2) < 0$. Let $\sigma^2 := (\sigma_t^2)_{t\geq 0}$ be the strictly stationary volatility process given as solution to (3.8)*. Assume further*

(H5) There exists a positive constant $\delta > 0$ such that $\mathbb{E}\left(G_1^{8+\delta}\right) < \infty$.

Then as $n \to \infty$,

$$\sqrt{n}\left(\begin{bmatrix}\hat{\mu}_n \\ \hat{\gamma}_n\end{bmatrix} - \begin{bmatrix}\mu \\ \gamma\end{bmatrix}\right) \xrightarrow{d} \mathcal{N}_{d+2}(\mathbf{0}, \boldsymbol{\Sigma}),$$

where the covariance $\boldsymbol{\Sigma}$ has components

$$\begin{aligned}\Sigma_{k+2,l+2} =& \mathbb{C}ov((G_1^{(1)})^2(G_{1+k}^{(1)})^2, (G_1^{(1)})^2(G_{1+l}^{(1)})^2) \\ &+ 2\sum_{j=1}^{\infty} \mathbb{C}ov((G_1^{(1)})^2(G_{1+k}^{(1)})^2, (G_{1+j}^{(1)})^2(G_{1+l+j}^{(1)})^2)\end{aligned}$$

for $k, l = 0, \ldots, d$,

$$\Sigma_{1,k+2} = \mathbb{C}ov((G_1^{(1)})^2, (G_1^{(1)})^2(G_{1+k}^{(1)})^2) + 2\sum_{j=1}^{\infty} \mathbb{C}ov((G_1^{(1)})^2, (G_{1+j}^{(1)})^2(G_{1+k+j}^{(1)})^2)$$

for $k = 0, \ldots, d$ and $\Sigma_{1,1} = \gamma(0) + 2\sum_{h=1}^{\infty} k_\gamma e^{-ph}$, with $k_\gamma := \mathbb{C}ov((G_i^{(1)})^2, (G_{i+1}^{(1)})^2)e^p$.

By Theorem 3.16 the strong consistency of the estimator $\hat{\boldsymbol{\vartheta}}_n$ follows. Furthermore it gives asymptotic normality of the parameter estimates. The true parameter vector and the corresponding moments are indicated by $\boldsymbol{\vartheta}_0$, μ_0 and $\boldsymbol{\gamma}_0$ respectively. And $\mathbb{P}_{\boldsymbol{\vartheta}_0}$ denotes the probability measure with respect to the parameter vector $\boldsymbol{\vartheta}_0$.

Theorem 3.16 *(Haug (2006, Theorem 2.3.9))*
Suppose that $(L_t)_{t\geq 0}$ is such that $\mathbb{E}(L_1^4) < \infty$ and the parameters of the COGARCH$(1,1)$ process satisfy $\Psi(2) < 0$. Let $\sigma^2 := (\sigma_t^2)_{t\geq 0}$ be the strictly stationary volatility process given as solution to (3.8)*. Assume that (H1)-(H4) are satisfied. For $S(\boldsymbol{\gamma})$ as in* (3.44)*, define the mapping $Q : \mathbb{R}^{d+2} \to \mathbb{R}^3$ by $(\mu, \boldsymbol{\gamma}^\top) \mapsto Q((\mu, \boldsymbol{\gamma}^\top)) := J(\mu, \gamma(0), S(\boldsymbol{\gamma}))$. Then as $n \to \infty$,*

$$\hat{\boldsymbol{\vartheta}}_n \overset{a.s.}{\to} \boldsymbol{\vartheta}_0. \tag{3.45}$$

Assume additionally (H5). Then, under $\mathbb{P}_{\boldsymbol{\vartheta}_0}$, as $n \to \infty$,

$$\sqrt{n}(\hat{\boldsymbol{\vartheta}}_n - \boldsymbol{\vartheta}_0) \overset{d}{\to} \partial_{(\mu,\boldsymbol{\gamma})} Q((\mu_0, \boldsymbol{\gamma}_0)) \mathcal{N}_{d+2}(\mathbf{0}, \boldsymbol{\Sigma}), \tag{3.46}$$

where $\boldsymbol{\Sigma}$ is as in Proposition 3.15 and with $(\mu, \boldsymbol{\gamma}) \in \mathbb{R}^{d+2}$ the derivative is denoted by

$$\begin{aligned}\partial_{(\mu,\boldsymbol{\gamma})} Q((\mu_0, \boldsymbol{\gamma}_0)) &= (\partial_1 Q((\mu_0, \boldsymbol{\gamma}_0)), \partial_2 Q((\mu_0, \boldsymbol{\gamma}_0)), \ldots, \partial_{d+2} Q((\mu_0, \boldsymbol{\gamma}_0))) \\ &= \left(\partial_\mu Q((\mu_0, \boldsymbol{\gamma}_0)), \partial_{\gamma(0)} Q((\mu_0, \boldsymbol{\gamma}_0)), \ldots, \partial_{\gamma(d)} Q((\mu_0, \boldsymbol{\gamma}_0))\right).\end{aligned}$$

Estimation of the volatility σ_t^2

Given the estimated parameters $\hat{\boldsymbol{\vartheta}}_n$ the volatility process can be calculated recursively. For $r = 1$ we have with (3.10),

$$\sigma_i^2 = \sigma_{i-1}^2 + \beta - \eta \int_{(i-1,i]} \sigma_s^2 ds + \varphi \sum_{i-1<s\leq i} \sigma_s^2 (\Delta L_s)^2, \quad i \in \mathbb{N}. \tag{3.47}$$

Since σ_s is latent variable and ΔL_s is usually not observable, we have to approximate the integral and the sum on the right hand side of (3.47). We use a simple Euler approximation for the integral

$$\int_{(i-1,i]} \sigma_s^2 ds \approx \sigma_{i-1}^2, \quad i \in \mathbb{N}. \tag{3.48}$$

As we observe G only at integer times, we approximate

$$\sum_{i-1<s\leq i} \sigma_s^2 (\Delta L_s)^2 \approx (G_i - G_{i-1})^2 = (G_i^{(1)})^2, \quad i \in \mathbb{N}. \tag{3.49}$$

With the approximations (3.48) and (3.49), the volatility process $(\sigma_t^2)_{t\geq 0}$ can be calculated recursively by

$$\hat{\sigma_i}^2 = \hat{\sigma}_{i-1}^2 + \hat{\beta} - \hat{\eta}\hat{\sigma}_{i-1}^2 + \hat{\varphi}(G_i^{(1)})^2 \tag{3.50}$$

$$= \hat{\beta} + (1 - \hat{\eta})\hat{\sigma}_{i-1}^2 + \hat{\varphi}(G_i^{(1)})^2, \quad i \in \mathbb{N}. \tag{3.51}$$

We require that $0 < \eta < 1$ because $\hat{\sigma}_i$ defines the conditional variance of a discrete time GARCH$(1,1)$ model, see (3.2).

3.2.2 Pseudo-Maximum Likelihood Method

In this chapter, we focus on a pseudo-maximum likelihood (PML) method. With this method, it should be possible to consider not only equally spaced (in time) data, but also in time unequally spaced data. This approach has been developed by Maller et al. (2008) in their paper "GARCH modelling in continuous time for irregularly spaced time series data". They show how to fit the continuous time GARCH model to irregularly spaced time series data using discrete-time GARCH methodology, by approximating the COGARCH with an embedded sequence of discrete time GARCH series. In Chapter 3.2.2 we introduce a method of fitting this model to data which is irregularly spaced in time. Therefore we need to discretize the continuous time volatility process as we describe in Chapter 3.2.2. For the discrete approximation of the continuous time GARCH a "first jump" approximation, described in Chapter 3.2.2, is used.

The "First Jump" Approximation for a Lévy Process

In the following part, we present the results stated in Section 2. and 3. of Maller and Szimayer (2007)[16].

The following setup is assumed. Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a completed probability space on which a real-valued Lévy process $L := (L_t)_{t\geq 0}$, $L_0 = 0$, with càdlàg paths is defined. Let $\mathbb{F}^L = (\mathcal{F}_t)_{t\geq 0}$ be the right-continuous filtration generated by L, and assume $\mathcal{F}_0$ contains all $\mathbb{P}$-nullsets, and that $\mathcal{F}_\infty = \mathcal{F}$. Furthermore $\sigma_L^2 = 0$, i.e. no Brownian component is present and L is therefore a pure jump process. The Lévy process L is characterized by its Lévy triplet $(\gamma_L, 0, \nu_L)$, where $\gamma_L \in \mathbb{R}$ and $\nu_L(\cdot)$ is the Lévy measure, see (2.12), and γ_L is a constant obtained under the standard truncation function $x \mapsto \mathbb{1}_{[-1,1]}(x)$. The jumps of L are denoted as in Chapter 2: $\Delta L_t = L_t - L_{t-}$, for $t \geq 0$ (with $L_{0-} = 0$). Let

$$\overline{\nu}_L(x) = \nu_L((x, \infty)) + \nu_L((-\infty, -x]), \quad x > 0, \tag{3.52}$$

denote the tail of $\nu_L(\cdot)$. L is to be approximated on a finite time interval $[0, T]$, $0 < T < \infty$, partitioned into N_n not necessarily equally spaced time intervals. Let $(N_n)_{n\in\mathbb{N}}$ be an increasing sequence of integers diverging to infinity as $n \to \infty$. For each $n \in \mathbb{N}$, form a deterministic partition $0 = t_0(n) < t_1(n) < \cdots < t_{N_n}(n) = T$ of $[0, T]$. Then, two approximating processes to L are constructed in Maller and Szimayer (2007).

The first approximation, $\overline{L}_t(n)$ for $n \in \mathbb{N}$ is formed by taking the first jump, if one occurs, of L_t in each time subinterval $(t_{j-1}(n), t_j(n)]$, $j = 1, 2, \ldots, N_n$, where the jump sizes are bounded away from 0, then discretizing ("binning") these jumps to get an approximating process which takes only a finite number of values on a finite state space. The state space does not include 0, as we must avoide the possible singularity in ν_L at 0. If no jumps occur in a subinterval, $\overline{L}_t(n)$ remains constant in that subinterval.

A second approximating process, $L_t(n)$, $n \in \mathbb{N}$, is then taken as the discrete skeleton of $\overline{L}_t(n)$ on the time grid $(t_j(n))_{j=0,1,\ldots,N_n}$. This means $L_t(n)$ is obtained from $\overline{L}_t(n)$ by delaying its jump in an interval to the next following point on the grid, see (3.55).

[16]The summarized results can be found in Klüppelberg et al. (2011, Section 5.2).

The time and space intervals are allowed to shrink and the state space to expand at appropriate rates, so as to get convergence of $\overline{L}_t(n)$ and $L_t(n)$ to L_t, as $n \to \infty$, in various modes.

Take two sequences of real numbers $(m_n)_{n\in\mathbb{N}}$ and $(M_n)_{n\in\mathbb{N}}$, satisfying $1 > m_n \downarrow 0$ and $1 < M_n \uparrow \infty$, as $n \to \infty$. The first approximating process, $\overline{L}_t(n)$, takes discrete values in the set

$$J(n) = [-M_n, -m_n) \cup (m_n, M_n],\ n \in \mathbb{N}.$$

To construct it, denote the time of the first jump with magnitude in $(m_n, M_n]$ in interval j by,

$$\tau_j(n) := \inf\{t : t_{j-1}(n) < t \leq t_j(n); \Delta L_t \in J(n)\}, \quad \text{for } 1 \leq j \leq N_n,$$

where the infimum over the empty set is defined as ∞. Then decompose L as

$$L_t = \gamma_{Ln} t + L_t^{(1)}(n) + L_t^{(2)}(n) + L_t^{(3)}(n), \quad \text{for } 0 \leq t \leq T, \tag{3.53}$$

where for all $n \geq 1$ and $0 \leq t \leq T$:

$$\begin{aligned}
L_t^{(1)}(n) &= \text{a.s}\lim_{\epsilon\downarrow 0}\left(\sum_{0<s\leq t} \Delta L_s \mathbb{1}_{\{\epsilon<|\Delta L_s|\leq m_n\}} - t\int_{\epsilon<|x|\leq m_n} x\nu_L(dx)\right),\\
L_t^{(2)}(n) &= \sum_{0<s\leq t} \Delta L_s \mathbb{1}_{\{M_n<|\Delta L_s|\}},\\
L_t^{(3)}(n) &= \sum_{0<s\leq t} \Delta L_s \mathbb{1}_{\{m_n<|\Delta L_s|\leq M_n\}},
\end{aligned}$$

and

$$\gamma_{Ln} = \gamma_L - \int_{m_n<|x|\leq 1} x\nu_L(dx).$$

Equation (3.53) is a variant of the Lévy-Itô decomposition, cf. Definition 2.13, in which, for each n, $L_t^{(1)}(n)$ is a compensated "small jump" martingale, and $L_t^{(2)}(n)$ and $L_t^{(3)}(n)$ might be thought of as "large jumps" and "medium jumps", respectively.

With no assumptions on L, Maller and Szimayer (2007) show that, for $j = 1, 2$, $\lim_{n\to\infty} \sup_{o\leq t\leq T} L_t^{(j)}(n) = 0$ a.s. and $L_t^{(3)}(n)$ can be further decomposed as follows:

$$L_t^{(3)}(n) = L_t^{(3,1)}(n) + L_t^{(3,2)}(n),$$

where

$$L_t^{(3,2)}(n) = \sum_{j=1}^{N_n} \mathbb{1}_{\{\tau_j(n)\leq t\}} \Delta L^{(3)}_{\tau_j(n)}.$$

Thus $L_t^{(3,2)}(n)$ is the sum of the sizes of the first jump of L_t in each subinterval whose magnitude is in $(m_n, M_n]$, where such jumps occur, while $L_t^{(3,1)}(n)$ collects, over all subintervals, the sizes of those jumps with magnitudes in $(m_n, M_n]$ (except for the first jump), provided at least two such jumps occur in a subinterval.

Since we allow for the possibility that L has "infinite activity", that is, that $\nu_L(\mathbb{R}\backslash\{0\}) = \infty$, see Remark 2.10, we need a restriction on how fast m_n may tend to the

possible singularity of ν_L at 0, by comparison with the speed at which the time mesh shrinks. With appropriate assumptions, $\lim_{n\to\infty} \sup_{0\le t\le T} |L_t^{(3,1)}(n)| = 0$ in probability, in $\mathcal{L}^1$, or, alternatively, in almost sure sense. This leaves $L_t^{(3.2)}(n)$ as the predominant component, asymptotically, of L, and the penultimate step is to approximate it by a process $\overline{L}(n)$ that lives on a finite state space. So we discretize the state space $J(n)$ with a grid of mesh size $\Delta(n) > 0$, where $\Delta(n) \searrow 0$ as $n \to \infty$, and set

$$\overline{L}_t(n) = \gamma_{Ln} + \sum_{j=1}^{N_n} \mathbb{1}_{\{\tau_j(n)\le t\}} \left\lfloor \frac{\Delta L^{(3)}_{\tau_j(n)}}{\Delta(n)} \right\rfloor \Delta(n). \tag{3.54}$$

(By $\lfloor x \rfloor$ the integer part of $x \in \mathbb{R}$ is denoted.) Again under certain conditions, the difference between $L^{(3,2)}(n)$ and $\overline{L}(n)$ disappears, asymptotically, in the $\mathcal{L}_1$ or almost sure sense, uniformly in $0 \le t \le T$. Thus $\overline{L}(n)$ approximates L, in the sense that the distance between them as measured by the supremum metric tends to 0 in $\mathcal{L}_1$ or almost surely, in our setup.

The second approximation, $L(n)$, is obtained by evaluating $\overline{L}(n)$ on the same discrete time grid as we have used so far. Thus $L(n)$ is the piecewise constant process defined by

$$L_t(n) = \overline{L}_{t_{j-1}(n)}(n), \text{ when } t_{j-1}(n) \le t \le t_j(n),\ j = 1, 2, \ldots, N_n, \tag{3.55}$$

and with $L_T(n) = \overline{L}_T(n)$. Because the original jumps are displaced in time in $L(n)$, we no longer expect convergence to L in the supremum metric. Instead, we obtain $\lim_{n\to\infty} \rho(L(n), L) = 0$, where $\rho(\cdot,\cdot)$ denotes the Skorokhod J_1 distance [17] in $\mathbb{D}[0,T]$, see Theorem 3.18. The processes $L(n)$ approximate L, pointwise, in probability, but not uniformly in $0 \le t \le T$. However, the convergence in probability in the Skorokhod topology suffices for certain applications that we discuss later.

All in all, the discrete approximation scheme can be conceptualized as follows. With $\Delta L^{(3)}_{\tau_j(n)}$ the sizes of the first jump of of L_t in each interval j, $j = 1, 2, \ldots$, whose size is in $(m_n, M_n]$, are denoted. By $k = k(j, n, \Delta(n))$ the (integer) number of times that Δn divides the jump $\Delta L^{(3)}_{\tau_j(n)}$ is defined. If there exists a jump $\Delta L^{(3)}_{\tau_j(n)}$ with magnitude in $(m_n, M_n]$ in interval j, take the first jump and replace it by $k\Delta(n)$, where $\Delta(n)$ denotes the mesh size of the grid. If there is no such jump, take $k = 0$. In order to get (3.54) add up over all such jumps up to time t (and add in $\gamma_{Ln}t$) to get $\overline{L}_t(n)$). The paths of $\overline{L}_t(n)$ are step functions which jump at most once per subinterval. Evaluating the paths at the times of the discrete time grid (taking the skeleton of $\overline{L}_t(n)$) consequently yields a càdlàg process $L_t(n)$ with discrete and finite state space which is constant between grid points. It might seem surprising at first that taking only the first jump in each subinterval of a discretization of the time axis provides sufficient information, in a

[17] The Skorokhod J_1 distance between two processes X_t, Y_t in $\mathbb{D}[0,T]$ is defined by $\rho(X,Y) = \inf_{\lambda\in\Lambda} \left\{ \sup_{0\le t\le T} \left| X_t - Y_{\lambda(t)} \right| + \sup_{0\le t\le T} |\lambda(t) - t| \right\}$, where Λ is the set of strictly increasing continuous functions λ on $[0,T]$, with $\lambda(0) = 0$ and $\lambda(T) = T$.

sense, to approximate even an infinite activity Lévy process. However, this does occur if we let the mesh sizes tend to zero fast enough, as specified by (3.56) below.

Now we state the theorems from Maller and Szimayer (2007, Section 3), which provide the convergence of $\overline{L}_t(n)$ and $L_t(n)$ to L_t. In the following, let $\mathbb{F}^L$, $\mathbb{F}^{\overline{L}(n)}$ and $\mathbb{F}^{L(n)}$ be natural filtrations generated by $(L_t)_{t\geq 0}$, $(\overline{L}_t(n))_{t\geq 0}$ and $(L_t(n))_{t\geq 0}$, respectively. Our construction clearly gives inclusion of the filtrations, that is, for each $n \geq 1$

$$\mathbb{F}^{L(n)} \subseteq \mathbb{F}^{\overline{L}(n)} \subseteq \mathbb{F}^{L}.$$

Recall from (3.52) that $\overline{\nu}_L$ denotes the tail of the Lévy measure of L_t. Let

$$\Delta t(n) := \max_{1\leq j\leq n} (t_j(n) - t_{j-1}(n)).$$

The main result for $\overline{L}_t(n)$ is the following Theorem.

Theorem 3.17 *(Maller and Szimayer (2007, Theorem 3.1(a)))*
Suppose

$$\lim_{n\to\infty} \Delta t(n)\overline{\nu}_L^2(m_n) = 0 \text{ and } \lim_{n\to\infty} \Delta(n)\overline{\nu}_L(m_n). \tag{3.56}$$

Then

$$\sup_{0\leq t\leq T} |\overline{L}_t(n) - L_t| \xrightarrow{P} 0, \text{ as } n \to \infty.$$

Next we consider the second approximating process, $L_t(n)$, as defined in (3.55). With a view to applications, we need the following property. The processess $(L_t(n))_{n\in\mathbb{N}}$ are said to satisfy *Aldous' criterion for tightness* if:

$$\forall \epsilon > 0 : \lim_{\delta\searrow 0} \limsup_{n\to\infty} \sup_{\sigma,\tau\in\mathcal{S}_{0,T}(n),\sigma\leq\tau\leq\sigma+\delta} \mathbb{P}(|L_\tau(n) - L_\sigma(n)| \geq \epsilon) = 0,$$

where $\mathcal{S}_{t,T}(n)$ is the set of $\mathbb{F}^{L(n)}$-stopping times taking values in $[0, T]$, for $0 \leq t \leq T$. Let $\mathbb{D}[0, T]$ be the space of càdlàg real-valued functions on $[0, T]$ and $\rho(\cdot, \cdot)$ the Skorokhod J_1 distance between two processes in $\mathbb{D}[0, T]$. The main result regarding $L_t(n)$ is:

Theorem 3.18 *Assume that Condition* (3.56) *of Theorem 3.17 holds. Then:*

- $\rho(L(n), L) \xrightarrow{P} 0$ *as* $n \to \infty$;
- *the sequence* $(L_t(n))_{n\in\mathbb{N}}$ *satisfies Aldous' criterion for tightness.*

A Discrete Approximation to the COGARCH

The following about the discrete approximation to the COGARCH can be found in Maller et al. (2008, Section 2.1) and in Klüppelberg et al. (2011, Section 5.3). They demonstrate how to approximate a COGARCH pair $(G(t), \sigma(t))_{t\geq 0}$ with an embedded sequence of discrete time GARCH pairs, $(G_n(t), \sigma_n(t))_{t\geq 0}$, using the first jump

approximation developed in the section before. After appropriate rescaling to match the discrete and continuous time parameter sets G_n will be shown to converge to G in a quite strong sense, as the approximating grid grows finer. It is assumed that L has finite variance and mean 0.

Fix $T > 0$, and take deterministic sequences $(N_n)_{n\geq 1}$ with $\lim_{n\to\infty} N_n = \infty$ and $0 = t_0(n) < t_1(n) < \cdots < t_{N_n}(n) = T$, and, for each $n = 1, 2, \ldots$, divide $[0, T]$ into N_n subintervals of length $\Delta t_i(n) := t_i(n) - t_{i-1}(n)$ for $i = 1, 2, \ldots, N_n$. Assume $\Delta t(n) := \max_{i=1,\ldots,N_n} \Delta t_i(n) \to 0$ as $n \to \infty$ and define for each $n = 1, 2, \ldots$, a discrete-time process $(G_{i,n})_{i=1,\ldots,N_n}$ satisfying

$$G_{i,n} = G_{i-1,n} + \sigma_{i-1,n}\sqrt{\Delta t_i(n)}\epsilon_{i,n}, \qquad i = 1, 2, \ldots, N_n, \tag{3.57}$$

where $G_{0,n} = G(0) = 0$ and the variance $\sigma^2_{i,n}$ follows the recursion

$$\sigma^2_{i,n} = \beta\Delta t_i(n) + (1 + \varphi\Delta t_i(n)\epsilon^2_{i,n})e^{-\eta\Delta t_i(n)}\sigma^2_{i-1,n}, \qquad i = 1, 2, \ldots, N_n. \tag{3.58}$$

Here, the innovations $(\epsilon_{i,n})_{i=1,\ldots,N_n}$, $n \in \mathbb{N}$, are constructed using a "first jump" approximation to the Lévy process as follows. Since we assume a finite variance for L, we only need a single sequence $1 \geq m_n \downarrow 0$ bounding the jumps of L away from 0. We assume it satisfies $\lim_{n\to\infty} \Delta t(n)\overline{\nu}^2_L(m_n) = 0$, where $\overline{\nu}_L(x) = \int_{|y|>x} \nu_L(dy)$ is the tail of ν_L. Such a sequence always exists, as $\lim_{x\downarrow 0} x^2\overline{\nu}_L(x) = 0$ for any Lévy measure. Fix $n \geq 1$ and define stopping times $\tau_{i,n}$ by

$$\tau_{i,n} = \inf\{t \in [t_{i-1}(n), t_i(n)) : |\Delta L_t| \geq m_n\}, \quad i = 1, \ldots, N_n. \tag{3.59}$$

Thus, $\tau_{i,n}$ is the time of the first jump of L in the ith interval where the jump magnitude exceeds m_n, if such a jump occurs.

By the strong Markov property, $\left(\mathbb{1}_{\{\tau_{i,n}<\infty\}}\Delta L(\tau_{i,n})\right)_{i=1,\ldots,N_n}$ is, for each $n = 1, 2, \ldots$, a sequence of independent random variables, with distribution specified by:

$$\frac{\nu(dx)\mathbb{1}_{\{|x|>m_n\}}}{\overline{\nu}(m_n)}\left(1 - e^{-\Delta t_i(n)\overline{\nu}(m_n)}\right), \quad x \in \mathbb{R}\backslash\{0\},\ i = 1, 2, \ldots, N_n,$$

and with mass $e^{-\Delta t_i(n)\overline{\nu}(m_n)}$ at 0. These random variables have finite mean $\kappa_i(n)$, and variance, $\xi_i(n)$, say, since $\mathbb{E}(L_1^2)$ is finite. The innovations series $(\epsilon_{i,n})_{i=1,\ldots,N_n}$ required for (3.57) is now defined by

$$\epsilon_{i,n} = \frac{\mathbb{1}_{\{\tau_{i,n}<\infty\}}\Delta L(\tau_{i,n}) - \kappa_i(n)}{\xi_i(n)}, \quad i = 1, 2, \ldots, N_n.$$

For each $n \in \mathbb{N}$, the $\epsilon_{i,n}$ are independent random variables with $\mathbb{E}(\epsilon_{1,n}) = 0$ and $\mathbb{V}\mathrm{ar}(\epsilon_{1,n}) = 1$. Finally, in (3.58), we take $\sigma^2_{0,n} = \sigma^2_0$, independent of the $\epsilon_{i,n}$.

Remark 3.19 *Equations* (3.57) *and* (3.58) *specify a GARCH*$(1,1)$*-type recursion in the following sense. In the ordinary discrete time GARCH*$(1,1)$ *series, the volatility sequence satisfies* (3.2)*, i.e.*

$$\sigma^2_i = \beta + (1 + (\lambda/\delta)\epsilon^2_{i-1})\delta\sigma^2_{i-1}. \tag{3.60}$$

When the time grid is equally spaced, so $\Delta t_i(n) = \Delta t(n)$, $i = 1, 2, \ldots, N_n$, (3.58) *is equivalent to* (3.60)*, after rescaling by* $\Delta t_i(n)$ *and a reparametrisation from* (β, φ, η) *to* (β, λ, δ)*. Equation* (3.57) *becomes a rescaled GARCH equation for the differenced sequence* $G_{i,n} - G_{i-1,n}$*. More generally, with an unequally spaced time grid, if the series are scaled as in* (3.57) *and* (3.58)*, convergence to the COGARCH is obtained as follows.*

Embed the discrete time-time processes $G_{\cdot,n}$ and $\sigma^2_{\cdot,n}$ into continuous time versions G_n and σ^2_n defined by

$$G_n(t) := G_{i,n} \text{ and } \sigma^2_n(t) := \sigma^2_{i,n}, \text{ when } t \in [t_{i-1}(n), t_i(n)), 0 \geq t \geq T, \tag{3.61}$$

with $G_n(0) = 0$. The processes G_n and σ_n are in $\mathbb{D}[0, T]$.

Theorem 3.20 *(Maller et al. (2008, Theorem 2.1))*
In the above setup, the Skorokhod distance between the processes (G, σ^2) *defined by* (3.7) *and* (3.8)*, and the discretized, piecewise constant processes* $(G_n, \sigma^2_n)_{n\geq 1}$ *defined by* (3.61)*, converge in probability to* 0 *as* $n \to \infty$*, that is*

$$\rho((G_n, \sigma^2_n), (G, \sigma^2)) \xrightarrow{\mathbb{P}} 0, \qquad \text{as } n \to \infty. \tag{3.62}$$

Consequently, we also have convergence in distribution in $\mathbb{D}[0, T] \times \mathbb{D}[0, T] : (G_n, \sigma^2_n) \xrightarrow{D} (G, \sigma^2)$ *as* $n \to \infty$.

Estimation Algorithm

The following deals with the application of the discrete approximation to the continuous time GARCH process in order to find a method of fitting the model to unequally spaced time series data. Therefore, Maller et al. (2008) use the methodology developed for the dicrete-time GARCH. Klüppelberg et al. (2011) summarized the approach in an **estimation algorithm**, which is as follows (cf. Klüppelberg et al. (2011, Section 5.4.1)). The parameters are estimated under the following assumptions:

(H1) Suppose given observations G_{t_i}, $0 = t_0 < t_1 < \cdots < t_N = T$, on the integrated COGARCH as defined and parameterised in (3.7) and (3.8), assumed to be in its stationary regime.

(H2) The (t_i) are assumed fixed (non-random) time points.

(H3) $\mathbb{E}(L_1) = 0$ and $\mathbb{E}(L_1^2) = 1$; i.e. σ^2 can be interpreted as the volatility.

(H4) The driving Lévy process has no Gaussian part, i.e. $\sigma_L^2 = 0$.

Then we proceed as follows.

(1) Let $Y_i = G_{t_i} - G_{t_{i-1}}$ denote the observed increments and put $\Delta t_i := t_i - t_{i-1}$. Then from (3.7) we can write

$$Y_i = \int_{t_{i-1}}^{t_i} \sigma_{s-} dL_s, \tag{3.63}$$

compare (3.9) for $r = 1$.

(2) We can use a pseudo-maximum likelihood (PML) method to estimate the parameters (β, η, φ) from the observed $Y_1, Y_2, \ldots, Y_N$. The pseudo-likelihood function can be derived as follows. Because $(\sigma_t)_{t\geq 0}$ is Markovian (see Theorem 3.4), Y_i is conditionally independent of $Y_{i-1}, Y_{i-2}, \ldots$, given $\mathcal{F}_{t_{i-1}}$. We have $\mathbb{E}(Y_i|\mathcal{F}_{t_{i-1}}) = \mathbb{E}(Y_i) = \mathbb{E}(G_{t_i} - G_{t_{i-1}}) = 0$ for the conditional expectation of Y_i, and for the conditional variance,

$$\rho_i^2 := \mathbb{E}(Y_i^2|\mathcal{F}_{t_{i-1}}) = \left(\sigma_{t_{i-1}}^2 - \frac{\beta}{\eta - \varphi}\right)\left(\frac{e^{(\varphi-\eta)\Delta t_i} - 1}{\varphi - \eta}\right) + \frac{\beta \Delta t_i}{\eta - \varphi}. \tag{3.64}$$

For the calculation of $\mathbb{E}(Y_i^2|\mathcal{F}_{t_{i-1}})$ we can proceed as follows.[18] In the proof of Lemma 4.1, Klüppelberg et al. (2004) showed that

$$\mathbb{E}(\sigma_{t+h}^2|\mathcal{F}_t) = (\sigma_t^2 - \mathbb{E}(\sigma_0^2))e^{h\Psi(1)} + \mathbb{E}(\sigma_h^2). \tag{3.65}$$

Furthermore in the proof of Proposition 5.1 they calculated that

$$\begin{aligned}
\mathbb{E}((G_h^{(r)})^2|\mathcal{F}_r) &:= \mathbb{E}_r((G_h^{(r)})^2) \\
&= \mathbb{E}(L_1^2) \int_h^{h+r} \mathbb{E}_r(\sigma_s^2) ds \\
&= (\sigma_r^2 - \mathbb{E}(\sigma_0^2))\mathbb{E}(L_1^2) \int_0^r e^{-s\Psi(1)} ds e^{h\Psi(1)} + \mathbb{E}(\sigma_0^2)\mathbb{E}(L_1^2) r.
\end{aligned} \tag{3.66}$$

Using (3.66) we obtain

$$\begin{aligned}
\mathbb{E}(Y_i^2|\mathcal{F}_{t_{i-1}}) &= \mathbb{E}((G_{t_i} - G_{t_{i-1}})^2|\mathcal{F}_{t_{i-1}}) \\
&= \mathbb{E}((G_{t_i}^{(\Delta t_i)})^2|\mathcal{F}_{t_{i-1}}) \\
&= \mathbb{E}(L_1^2) \int_{t_{i-1}}^{t_i} \mathbb{E}(\sigma_s^2|\mathcal{F}_{t_{i-1}}) ds
\end{aligned} \tag{3.67}$$

For $\mathbb{E}(\sigma_s^2|\mathcal{F}_{t_{i-1}})$ we get

$$\mathbb{E}(\sigma_s^2|\mathcal{F}_{t_{i-1}}) \overset{(3.65)}{=} (\sigma_{t_{i-1}}^2 - \mathbb{E}(\sigma_0^2))e^{\Psi(1)(s-t_{i-1})} + \mathbb{E}(\sigma_{s-t_{i-1}}^2). \tag{3.68}$$

Furthermore we have

$$\int_{t_{i-1}}^{t_i} \mathbb{E}(\sigma_{s-t_{i-1}}^2) ds = \int_{t_{i-1}}^{t_i} \left(\frac{\beta}{-\Psi(1)} + \left(\mathbb{E}(\sigma_0^2) + \frac{\beta}{\Psi(1)}\right)e^{(s-t_{i-1})\Psi(1)}\right) ds$$

[18] We give the calculation of $\mathbb{E}(Y_i^2|\mathcal{F}_{t_{i-1}})$ very detailed as (3.64) slightly differs from Equation (3.2) in Maller et al. (2008).

$$= \Delta t_i \frac{\beta}{-\Psi(1)} + \big(\mathbb{E}(\sigma_0^2) + \frac{\beta}{\Psi(1)}\big) \int_{t_{i-1}}^{t_i} e^{(s-t_{i-1})\Psi(1)} ds$$
$$= \Delta t_i \frac{\beta}{-\Psi(1)} + \big(\mathbb{E}(\sigma_0^2) + \frac{\beta}{\Psi(1)}\big) \left(\frac{e^{\Delta t_i \Psi(1)} - 1}{\Psi(1)} \right). \tag{3.69}$$

By inserting (3.68) in (3.67) and using (3.69) we have

$$\begin{aligned}
&\mathbb{E}(L_1^2) \int_{t_{i-1}}^{t_i} \mathbb{E}(\sigma_s^2 | \mathcal{F}_{t_{i-1}}) ds \\
&= \mathbb{E}(L_1^2)(\sigma_{t_{i-1}}^2 - \mathbb{E}(\sigma_0^2)) \int_{t_{i-1}}^{t_i} e^{(s-t_{i-1})\Psi(1)} ds + \mathbb{E}(L_1^2) \int_{t_{i-1}}^{t_i} \mathbb{E}(\sigma_{s-t_{i-1}}^2) ds \\
&= \mathbb{E}(L_1^2)(\sigma_{t_{i-1}}^2 - \mathbb{E}(\sigma_0^2)) \left(\frac{e^{(t_i - t_{i-1})\Psi(1)} - 1}{\Psi(1)} \right) + \mathbb{E}(L_1^2) \int_{t_{i-1}}^{t_i} \mathbb{E}(\sigma_{s-t_{i-1}}^2) ds \\
&= \mathbb{E}(L_1^2)(\sigma_{t_{i-1}}^2 - \mathbb{E}(\sigma_0^2)) \left(\frac{e^{\Delta t_i \Psi(1)} - 1}{\Psi(1)} \right) \\
&\qquad + \mathbb{E}(L_1^2)(\Delta t_i \frac{\beta}{-\Psi(1)} + (\mathbb{E}(\sigma_0^2) + \frac{\beta}{\Psi(1)}) \left(\frac{e^{\Delta t_i \Psi(1)} - 1}{\Psi(1)} \right)) \\
&= \mathbb{E}(L_1^2) \Delta t_i \frac{\beta}{-\Psi(1)} + \mathbb{E}(L_1^2) \left(\frac{e^{\Delta t_i \Psi(1)} - 1}{\Psi(1)} \right) (\sigma_{t_{i-1}}^2 - \mathbb{E}(\sigma_0^2) + \mathbb{E}(\sigma_0^2) + \frac{\beta}{\Psi(1)}) \\
&= \mathbb{E}(L_1^2) \Delta t_i \frac{\beta}{-\Psi(1)} + \mathbb{E}(L_1^2) \left(\frac{e^{\Delta t_i \Psi(1)} - 1}{\Psi(1)} \right) (\sigma_{t_{i-1}}^2 - \frac{\beta}{-\Psi(1)})
\end{aligned} \tag{3.70}$$

We know by (3.14) that $\Psi(1) = -\eta 1 + \int_{\mathbb{R}} ((1 + \varphi x^2)^1 - 1) \nu_L(dx) = -\eta + \varphi \int_{\mathbb{R}} x^2 \nu_L(dx) = -\eta + \varphi$, as we assumed $\mathbb{E}(L_1^2) = \int_{\mathbb{R}} x^2 \nu_L(dx) = 1$. Then we put this in (3.69) in order to obtain (3.64) as follows,

$$\begin{aligned}
\mathbb{E}(Y_i^2 | \mathcal{F}_{t_{i-1}}) &= \mathbb{E}(L_1^2) \Delta t_i \frac{\beta}{-\Psi(1)} + \mathbb{E}(L_1^2) \left(\frac{e^{\Delta t_i \Psi(1)} - 1}{\Psi(1)} \right) (\sigma_{t_{i-1}}^2 - \frac{\beta}{-\Psi(1)}) \\
&= \mathbb{E}(L_1^2) \Delta t_i \frac{\beta}{\eta - \varphi} + \mathbb{E}(L_1^2) \left(\frac{e^{\Delta t_i (\varphi - \eta)} - 1}{\varphi - \eta} \right) (\sigma_{t_{i-1}}^2 - \frac{\beta}{\eta - \varphi}) \\
&= \Delta t_i \frac{\beta}{\eta - \varphi} + \left(\frac{e^{\Delta t_i (\varphi - \eta)} - 1}{\varphi - \eta} \right) (\sigma_{t_{i-1}}^2 - \frac{\beta}{\eta - \varphi})
\end{aligned} \tag{3.71}$$

To ensure stationarity, we take $\mathbb{E}(\sigma_0^2) = \beta/(\eta - \varphi)$, with $\eta > \varphi$.

(3) Applying the PML method, we assume that the Y_i are conditionally $N(0, \rho_i^2)$,[19] and use recursive conditioning to write a pseudo-log-likelihood function for the observations $Y_1, Y_2, \ldots, Y_N$ as

$$\mathcal{L}_N = \mathcal{L}_N(\beta, \eta, \varphi) = -\frac{1}{2} \sum_{i=1}^{N} \left(\frac{Y_i^2}{\rho_i^2} \right) - \frac{1}{2} \sum_{i=1}^{N} \log(\rho_i^2) - \frac{N}{2} \log(2\pi). \tag{3.72}$$

[19] Here the Gaussian lig-likelihood is used as a contrast function. See Andersen (2009, p. 849) for details.

The pseudo-log-likelihood function (3.72) can be calculated as follows. As the Y_i are conditionally $\mathcal{N}(0, \rho_i^2)$, we have

$$f(y_i) = \frac{1}{\sqrt{2\pi\rho_i^2}} \exp\Big(-\frac{1}{2}\frac{y_i^2}{\rho_i^2}\Big). \tag{3.73}$$

Thus, we have the likelihood function

$$L_N(\theta) := L_N(\beta, \eta, \varphi) = \prod_{i=1}^{N} \Big(\frac{1}{\sqrt{2\pi\rho_i^2}} \exp\Big(-\frac{1}{2}\frac{Y_i^2}{\rho_i^2}\Big)\Big). \tag{3.74}$$

Finally we get (3.72) by

$$\begin{aligned}
\mathcal{L}_N &= \mathcal{L}_N(\beta, \eta, \varphi) = \log(L_N) \\
&= \log\Bigg(\prod_{i=1}^{N} \Big(\frac{1}{\sqrt{2\pi\rho_i^2}} \exp\Big(-\frac{1}{2}\frac{Y_i^2}{\rho_i^2}\Big)\Big)\Bigg) \\
&= \sum_{i=1}^{N} \Big(-\frac{1}{2}\log(2\pi\rho_i^2) - \frac{1}{2}\frac{Y_i^2}{\rho_i^2}\Big) \\
&= -\frac{1}{2}\sum_{i=1}^{N}\frac{Y_i^2}{\rho_i^2} - \frac{1}{2}\sum_{i=1}^{N}\log(\rho_i^2) - \frac{N}{2}\log(2\pi).
\end{aligned}$$

(4) We must substitute an approximation for ρ_i^2 in (3.72), hence, we need such for $\sigma_{t_{i-1}}$ in (3.64). From (3.57), we have $G_{i,n} - G_{i-1,n} = \sigma_{i-1,n}\sqrt{\Delta t_i(n)}\epsilon_{i,n}$. We use this in order to see how we can write (3.58) in the present notation, we get

$$\begin{aligned}
\sigma_{i,n}^2 &= \beta\Delta t_i(n) + (1 + \varphi\Delta t_i(n)\epsilon_{i,n}^2)\exp(-\eta\Delta t_i(n))\sigma_{i-1,n} \\
&= \beta\Delta t_i(n) + \exp(-\eta\Delta t_i(n))\sigma_{i-1,n} + \varphi\exp(-\eta\Delta t_i(n))\underbrace{\Delta t_i(n)\epsilon_{i,n}^2\sigma_{i-1,n}}_{\overset{(3.57)}{=}(G_{i,n}-G_{i-1,n})^2}
\end{aligned} \tag{3.75}$$

Now we discretise the continuous time volatility process just as was done in Theorem 3.20. As $Y_i = G_{t_i} - G_{t_{i-1}}$, (3.58) reads, in the present notation,

$$\sigma_i^2 = \beta\Delta t_i + e^{-\eta\Delta t_i}\sigma_{i-1}^2 + \varphi e^{-\eta\Delta t_i}Y_i^2. \tag{3.76}$$

(5) Finally, note that (3.76) is a GARCH-type recursion, so, after substituting σ_{i-1}^2 for $\sigma_{t_{i-1}}$ in (3.64), and the resulting modified ρ_i^2 in (3.72), we can think of (3.72) as the pseudo-log-likelihood function for fitting a GARCH model to the unequally spaced series.

This algorithm is implemented in Chapter 6.2.2. Taking $\beta/(\eta - \varphi)$ as a starting value for σ_0^2, we can maximize the function $\mathcal{L}_N$, see (3.72), to get pseudo-maximum likelihood estimates (PMLEs) of (β, η, φ). In the implemented PML function the first step is to calculate σ_i^2 in (3.76) recursively. Then the ρ_i^2 can be calculated by (3.79) and plugged into (3.72).

3.2.3 Modified PML

In Chapter 3.2.2, we assumed that the time points t_i, $i = 0, \dots, N$, are fixed, non-random time points. For random time points this approach did not work. Therefore, we modify this assumption to construct a "modified" PML method in this chapter. Suppose we have observations G_{t_i}, $0 = t_0 < t_1 < \cdots < t_N = T$, on the integrated COGARCH as defined and parameterized in (3.7) and (3.8), assumed to be in its stationary regime. Furthermore, we assume that the driving Lévy process of the COGARCH process is a compound Poisson process with the rate λ. We then have an exponential distribution for the Δt_i, $i = 0, \dots, N$, since they will be the interarrival times of the driving compound Poisson process of the COGARCH$(1,1)$ process. This means we do not have non-random time points like before, but we can simulate these random time points. Therefore, we assume to know the information contained in $\mathcal{F}_{t_{i-1}} \cup (\Delta t_i)$.

In Chapter 3.2.2, we calculated $\mathbb{E}(Y_i^2|\mathcal{F}_{t_{i-1}})$, where $\mathcal{F}_{t_{i-1}}$ includes the information we have up to time t_{i-1}. This means we do not know what Δt_i looks like. Now, we additionally calculate $\mathbb{E}(Y_i^2|\mathcal{F}_{t_{i-1}}) = \mathbb{E}(\mathbb{E}(Y_i^2|\mathcal{F}_{t_{i-1}} \cup (\Delta t_i))|\mathcal{F}_{t_{i-1}})$ by using (3.71).

$$\begin{aligned}
&\mathbb{E}(\mathbb{E}(Y_i^2|\mathcal{F}_{t_{i-1}} \cup (\Delta t_i))|\mathcal{F}_{t_{i-1}}) = \\
&\quad = \mathbb{E}\left(\frac{\beta}{\eta-\varphi}\Delta t_i + \left(\sigma^2_{t_{i-1}} - \frac{\beta}{\eta-\varphi}\right)\left(\frac{e^{(\varphi-\eta)\Delta t_i}-1}{\varphi-\eta}\right)\middle|\mathcal{F}_{t_{i-1}}\right) \\
&\quad = \frac{\beta}{\eta-\varphi}\mathbb{E}\left(\Delta t_i|\mathcal{F}_{t_{i-1}}\right) + \left(\sigma^2_{t_{i-1}} - \frac{\beta}{\eta-\varphi}\right)\mathbb{E}\left(\frac{e^{(\varphi-\eta)\Delta t_i}-1}{\varphi-\eta}\middle|\mathcal{F}_{t_{i-1}}\right) \\
&\quad = \frac{\beta}{\eta-\varphi}\mathbb{E}\left(\Delta t_i\right) + \left(\sigma^2_{t_{i-1}} - \frac{\beta}{\eta-\varphi}\right)\frac{\mathbb{E}\left(e^{(\varphi-\eta)\Delta t_i}\right)-1}{\varphi-\eta} \qquad (3.77)
\end{aligned}$$

We assumed that the driving Lévy process is a compound Poisson process with the rate λ, we know that $\Delta t_1, \dots, \Delta t_n \overset{i.i.d}{\sim} \text{Exp}(\lambda)$. Therefore we have $\mathbb{E}(\Delta t_i) = 1/\lambda$ and we can calculate $\mathbb{E}\left(e^{(\varphi-\eta)\Delta t_i}\right)$. We use the moment generating function, which is defined by

$$M_X(t) := \mathbb{E}(e^{tX}).$$

For $X \sim \text{Exp}(\lambda)$, $M_X(t) = \lambda/(\lambda - t)$ follows. Thus we have

$$\mathbb{E}\left(e^{(\varphi-\eta)\Delta t_i}\right) = \frac{\lambda}{\lambda-(\varphi-\eta)} = \frac{\lambda}{\lambda+\eta-\varphi}, \qquad (3.78)$$

and therefore (3.77) turns into

$$\begin{aligned}
&\mathbb{E}(\mathbb{E}(Y_i^2|\mathcal{F}_{t_{i-1}} \cup (\Delta t_i))|\mathcal{F}_{t_{i-1}}) = \\
&\quad = \frac{\beta}{\lambda(\eta-\varphi)} + \left(\sigma^2_{t_{i-1}} - \frac{\beta}{\eta-\varphi}\right)\frac{1}{\varphi-\eta}\left(\frac{\lambda}{\lambda+\eta-\varphi}-1\right). \qquad (3.79)
\end{aligned}$$

Recall from above that $\Delta t_1, \dots, \Delta t_n \overset{i.i.d}{\sim} \text{Exp}(\lambda)$. Therefore we can use the maximum likelihood estimate for the rate λ. Thus it can be estimated by

$$\hat{\lambda} = \frac{1}{\frac{1}{n}\sum_{i=1}^n \Delta t_i}. \qquad (3.80)$$

In the implementation of the PML method, see Chapter 6.2.2, it will be the first step to estimate $\hat{\lambda}$. Moreover we will use (3.79) for the calculation of ρ_i^2.

In Chapter 3.3 where we conduct a simulation study, we also applied the "modified" PML method. The results of the outcome of this estimation can be found in Table 3.9.

3.3 Simulation study

In order to illustrate the theory we introduced before, we now conduct a simulation study. The following simulations and estimations are based on the examples introduced in Chapter 3.1.3. The COGARCH processes are either driven by a compound Poisson or a Variance Gamma Lévy process. We will use two different approaches to estimate the parameters of the models: the moment estimation method (cf. Chapter 3.2.1) and the pseudo-maximum likelihood (PML) method (cf. Chapter 3.2.2).

In order to get an overview of the examples and their results in the following, we give a short overview in Table 3.1.

Compound Poisson COGARCH$(1,1)$ process

In Chapter 3.1.3, some properties of the compound Poisson COGARCH$(1,1)$ process have been reviewed, e.g. the Laplace exponents $\Psi(1)$ in (3.24) and $\Psi(2)$ in (3.25). In order to apply the estimation methods, some assumptions have to be fulfilled. See Chapter 3.2.1 (H1)-(H5) and Chapter 3.2.2 (H1)-(H4).
For the moment estimation method it is required that $\mathbb{E}(L_1) = 0$ and $\mathbb{V}\mathrm{ar}(L_1) = \mathbb{E}(L_1^2) = 1$. Therefore, in (3.25) we must have $\mathbb{E}(Y^2) = 1/\lambda$, where Y denotes the jumpsizes. Moreover, the condition $\int_{\mathbb{R}} x^3 \nu_L(dx) = 0$ yields $\mathbb{E}(Y^3) = 0$. The condition $\Psi(2) < 0$ leads to $\varphi^2 < 2(\eta - \varphi)\mathbb{E}(Y^2)/\mathbb{E}(Y^4)$. Condition (H5) requires for G a finite moment of higher order than the eighth, which is the case if $\mathbb{E}(L_1^{8+\delta}) < \infty$, $\delta > 0$, and the $(4+\delta)$-moment of the volatility is finite, i.e. $\Psi(4+\delta) < 0$. The $(8+\delta)$-moment of L will be finite if $\mathbb{E}(Y^{8+\delta}) < \infty$. The volatility will have a finite fourth moment if

$$\Psi(4) = 4(\varphi - \eta) + 6\lambda\varphi^2\mathbb{E}(Y^4) + 4\lambda\varphi^3\mathbb{E}(Y^6) + \lambda\varphi^4\mathbb{E}(Y^8)$$

is negative.[20]

We are going to simulate a COGARCH$(1,1)$ process driven by a compound Poisson Lévy process. We then have a look at the parameters estimated by Algorithm 3.2.1. To apply this Algorithm, in time equally spaced observations are necessary. The function `cogarch_sim` used for the simulation of a COGARCH process, gives as an output the values (G_{t_i}) at the random jumptimes $t_i \geq 0$ for $i = 1, \ldots, n$, $n \in \mathbb{N}$. As these times are not equally spaced, it is possible to apply the function `prevTick`. This gives the values of the COGARCH process at a specified time-grid by previous tick interpolation. For more details on the approach of the implemented programs see Chapter 6.
In the simulations of a COGARCH$(1,1)$ process driven by a compound Poisson Lévy

[20] cf. Haug (2006, Chapter 2.4.1).

No.	Parameters	Lévy Process	n	Method	Type	Table
1	$\beta = 1$ $\eta = 0.06$ $\varphi = 0.0425$	cp	5000	Moment Est. (equally spaced)	Estimated values	3.2
2	$\beta = 1$ $\eta = 0.06$ $\varphi = 0.0425$	cp	20000	Moment Est. (equally spaced)	Estimated values	3.3
					Residuals	3.4
3	$\beta = 1$ $\eta = 0.06$ $\varphi = 0.0425$	cp	5000	PML (equally spaced)	Estimated values	3.5
					Residuals	3.6
4	$\beta = 1$ $\eta = 0.06$ $\varphi = 0.0425$	cp	20000	PML (unequally spaced)	Estimated values	3.7
5	$\beta = 1$ $\eta = 0.06$ $\varphi = 0.0425$	cp	5000	PML (unequally spaced)	Estimated values	3.8
6	$\beta = 0.05$ $\eta = 0.06$ $\varphi = 0.04$	cp	5000	”modified” PML (unequally spaced)	Estimated values	3.9
	$\lambda = 1$				Estimated values	3.10
7	$\beta = 0.04$ $\eta = 0.053$ $\varphi = 0.038$	vg	1000	Moment Est. (equally spaced)	Estimated values	3.11
8	$\beta = 0.04$ $\eta = 0.053$ $\varphi = 0.038$	vg	1000	PML (equally spaced)	Estimated values	3.12
9	$\beta = 0.07$ $\eta = 0.06$ $\varphi = 0.04$	vg	1000	Moment Est. (equally spaced)	Estimated values	3.13
10	$\beta = 0.07$ $\eta = 0.06$ $\varphi = 0.04$	vg	1000	PML (equally spaced)	Estimated values	3.14

Table 3.1: Overview of Examples

process, the parameters are first chosen as $\beta = 1$, $\eta = 0.06$ and $\varphi = 0.0425$. So we get $\Psi(1) = -0.0175$ and $\Psi(2) = -0.02958$. In the assumptions of Proposition 3.8 we demand that $\Psi(1) < 0$ and $\Psi(2) < 0$ should be fulfilled, which is the case for the parameters chosen. Moreover we choose the jumps to be standard normally distributed and $\lambda = 1$. In this case all the assumptions stated in Chapter 3.2.1 are satisfied.

We simulate 900 samples of $n = 5000$ equidistant observations $G_i^{(1)}$, $i = 1, \ldots, n$. Table 3.2 summarizes $\hat{\beta}$, $\hat{\eta}$ and $\hat{\varphi}$ the mean, the bias, the relative bias, the mean squared error (MSE), the mean absolute error (MAE), the median and the bias of the median for the estimated parameters. Compared to the true parameters $\beta = 1$, $\eta = 0.06$ and

n=5000	$\hat{\beta}$	$\hat{\eta}$	$\hat{\varphi}$
Mean	0.91513	0.05193	0.03599
Bias	-0.08487	-0.00807	-0.00651
Rel. Bias	-0.08487	-0.13457	-0.15323
MSE	0.14928	0.00041	0.00023
MAE	0.30852	0.01657	0.01247
Median	0.88923	0.05007	0.03419
Bias Median	-0.11077	-0.00993	-0.00831
MSE Median	0.07176	0.00022	0.00013
MAE Median	0.26788	0.01477	0.01130

Table 3.2: Estimated mean, bias, relative bias, MSE, MAE, median, bias of the median, MSE of the median and MAE of the median for $\hat{\beta}$, $\hat{\eta}$ and $\hat{\varphi}$ of 900 samples estimated with the **Moment Estimator Method**. The true values are $\beta = 1$, $\eta = 0.06$ and $\varphi = 0.0425$.

$\varphi = 0.0425$ the mean values for the estimated parameters do not differ very much. Both the MSE and the MAE values are the highest for the estimator $\hat{\beta}$. We included the median values in order to see if the mean parameter estimates are influenced by some outliers, see the boxplots in Figure 3.5. The Figure on the lefthand side confirms that the estimator $\hat{\beta}$ varies the most, i.e. has some huge outliers. Comparing the true values for β, η and ϕ (marked by a dotted horizontal line) to the median of the estimated values $\hat{\beta}$, $\hat{\eta}$ and $\hat{\phi}$, we observe that they are quite close.

Furthermore, we simulated 900 samples of $n = 20000$ equidistant observations $G_i^{(1)}$, $i = 1, \ldots, n$. The outcomes can be found in Table 3.3. Estimating the parameters for a longer time horizon leads to better estimation results, see for example the median values for the estimators in Table 3.3.

In the following we would like to conduct a residual analysis. Therefore, we need the estimated values of $\hat{\sigma}_t^2$. In Chapter 3.2.1, it has been described how the volatility process $\hat{\sigma}_t^2$ can be estimated. We take the parameters $\hat{\beta}$, $\hat{\eta}$ and $\hat{\varphi}$ as estimated for our example above. We estimate them by both methods, the moment estimation and the PML method. Then, in order to calculate the volatility process recursively, we plug in those estimators and the squared returns of the process in (3.51). As starting value σ_0^2

n=20000	$\hat{\beta}$	$\hat{\eta}$	$\hat{\varphi}$
Mean	0.92295	0.05450	0.03831
Bias	-0.07705	-0.00550	-0.00419
Rel. Bias	-0.07705	-0.09170	-0.09858
MSE	0.04296	0.00016	0.00010
MAE	0.16669	0.01032	0.00818
Median	0.92020	0.05319	0.03703
Bias Median	-0.07980	-0.00681	-0.00547
MSE Median	0.02145	0.00009	0.00006
MAE Median	0.14645	0.00942	0.00749

Table 3.3: Estimated mean, bias, relative bias, MSE, MAE, median, bias of the median, MSE of the median and MAE of the median for $\hat{\beta}$, $\hat{\eta}$ and $\hat{\varphi}$ of 900 samples estimated with the **Moment Estimator Method**. The true values are $\beta = 1$, $\eta = 0.06$ and $\varphi = 0.0425$.

we choose [21] $\sigma_0^2 = \hat{\beta}/\hat{\eta}$ In Figure 3.4 the sample paths of σ_t^2 (black) and of $\hat{\sigma}_t^2$ (blue) of our example can be found. The plot on the top of this figure corresponds to the parameters estimated by the moment estimation method and the plot on the bottom to the parameters estimated by the PML method. By a residual analysis[22] we investi-

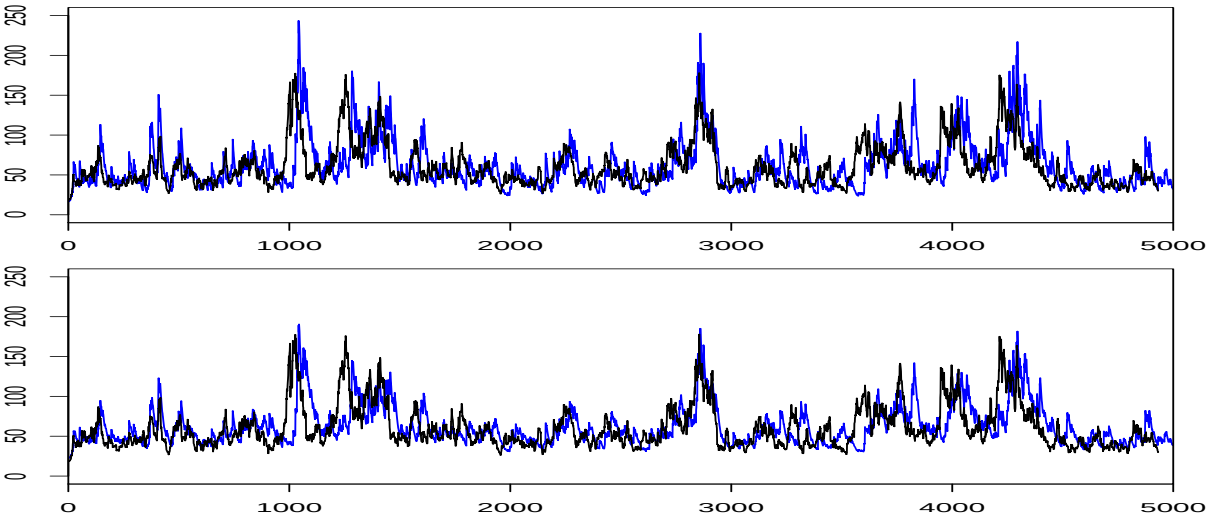


Figure 3.4: Sample paths of σ_t^2 (black) and $\hat{\sigma}_t^2$ (blue) of one simulation, where the true values were $\beta = 1$, $\eta = 0.06$ and $\varphi = 0.0425$ for a moment estimation (top) and a PML estimation (bottom).

gate the goodness of fit of our estimation method. The estimates residuals are given by $G_i^{(1)}/\hat{\sigma}_{i-1}$ for $i = 1, \ldots, n$. In Table 3.4 the mean, MSE, MAE and the correspond-

[21] We choose σ_0^2 like this because in Definition 3.10 the volatility process was defined as $V_0 = \alpha_0 + \boldsymbol{a}'\boldsymbol{Y}_0$. And for $\boldsymbol{Y}_0 = 0$, $V_0 = \alpha_0 = \beta/\eta$ follows.

[22] This analysis is conducted analogously to Haug et al. (2007, Section 4.1).

n=20000	mean$(G_n^{(1)}/\hat{\sigma}_{n-1})$	std$(G_n^{(1)}/\hat{\sigma}_{n-1})$	skewness$(G_n^{(1)}/\hat{\sigma}_{n-1})$
Mean	-0.00020 (0.00024)	1.01027 (0.00019)	-0.00238 (0.00178)
MSE	0.00005 ($3{\cdot}10^{-6}$)	0.00014 ($6{\cdot}10^{-6}$)	0.00286 (0.00014)
MAE	0.00570 (0.00015)	0.01027 (0.00019)	0.04227 (0.00109)

Table 3.4: Estimated mean, MSE and MAE for the mean, standard deviation and skewness of the **residuals** with corresponding estimated standard deviations in brackets.

ing standard deviations for the mean, the standard deviation and the skewness of the residuals $G_i^{(1)}/\hat{\sigma}_{i-1}$ based on 900 simulations can be found. Since we assumed a symmetric jump distribution with zero mean, the residuals should be symmetric around zero and their mean should be close to zero. Furthermore we expect the standard deviation to be close to one. The skewness should also be close to zero. Therefore the outcomes indicate a reasonable fit.

Analogously we estimate the parameters with the PML Method. In Table 3.5 the outcomes of the estimation for those 900 samples of $n = 5000$ equally spaced timepoints can be found. Like the mean of the parameters estimated by the moment estimation method before, the mean of the parameters obtained by the PML approach is not too different from the true parameter values. Table 3.6 summarizes the outcomes of

n=5000	$\hat{\beta}$	$\hat{\eta}$	$\hat{\varphi}$
Mean	0.99036	0.04816	0.03098
Bias	-0.00964	-0.01184	-0.0115
Rel. Bias	-0.00964	-0.19731	-0.27097
MSE	0.08902	0.00023	0.00017
MAE	0.23388	0.01330	0.01182
Median	0.94971	0.04743	0.03071
Bias Median	-0.05029	-0.01257	-0.01179
MSE Median	0.04025	0.00017	0.00014
MAE Median	0.20063	0.01289	0.01180

Table 3.5: Estimated mean, relative bias and MSE for $\hat{\beta}$, $\hat{\eta}$ and $\hat{\varphi}$ of 900 samples with the **PML Method**. The true values are $\beta = 1$, $\eta = 0.06$ and $\varphi = 0.0425$.

the analysis of the residuals. Additionally having a look at Figure 3.5 which shows the boxplots of both examples studied, we see that in this case there is no big difference in the outcomes of the two estimation approaches. In the to row of Figure 3.5 the boxplots for $\hat{\beta}$ are displayed. In the bottom row the boxplots of $\hat{\eta}$ *(left in each plot)* and $\hat{\varphi}$ *(right in each plot)* can bee seen. On the lefthand side the parameters estimated via the moment estimation method are shown. On the righthand side we applied the PML method. By a small horizontal dotted line the true values for β, η and φ are indicated. In both cases this line is very close to the median (fat horizontal line in the middle of the box) which can be found in Table 3.2 and 3.5 respectively.

Furthermore we applied the PML method to unequally spaced data. Therefore

n=5000	mean$(G_n^{(1)}/\hat{\sigma}_{n-1})$	std$(G_n^{(1)}/\hat{\sigma}_{n-1})$	skewness$(G_n^{(1)}/\hat{\sigma}_{n-1})$
Mean	-0.00012 (0.00048)	0.99911 (0.00010)	-0.00022 (0.00341)
MSE	0.00021 (0.00001)	0.00001 ($5 \cdot 10^{-7}$)	0.01045 (0.00055)
MAE	0.01141 (0.00029)	0.00240 (0.00006)	0.08081 (0.00209)

Table 3.6: Estimated mean, MSE and MAE for the mean, standard deviation and skewness of the **residuals** with corresponding estimated standard deviations in brackets.

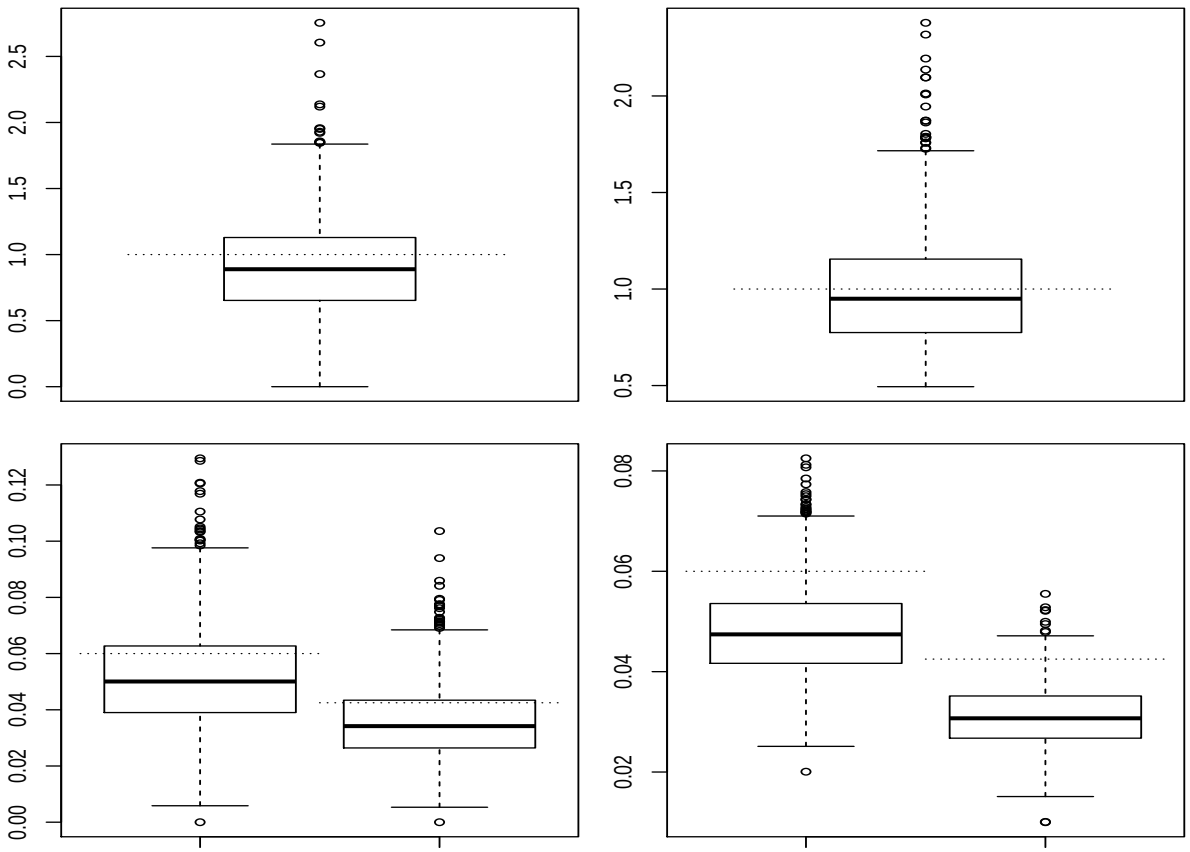


Figure 3.5: Boxplots of the estimated parameters $\hat{\beta}$, $\hat{\eta}$ and $\hat{\varphi}$ of 900 samples estimated with the Moment Estimator Method (*left*) and with the PML Method (*right*). The true values are $\beta = 1$, $\eta = 0.06$ and $\varphi = 0.0425$.

n=5000	$\hat{\beta}$	$\hat{\eta}$	$\hat{\varphi}$
Mean	0.93023	0.04605	0.03013
Bias	-0.06977	-0.01395	-0.01237
Rel. Bias	-0.06977	-0.23242	-0.29104
MSE	0.03083	0.00022	0.00017
MAE	0.14193	0.01398	0.01237
Median	0.91911	0.04589	0.03013
Bias Median	-0.08089	-0.01411	-0.01237
MSE Median	0.01572	0.00020	0.00015
MAE Median	0.12538	0.01411	0.01237

Table 3.7: Estimated mean, relative bias and MSE for $\hat{\beta}$, $\hat{\eta}$ and $\hat{\varphi}$ of 900 samples with the **PML Method** for data observed at irregularly spaced time points. The true values are $\beta = 1$, $\eta = 0.06$ and $\varphi = 0.0425$.

we simulated 900 samples over the time interval $[0, 5000]$ and the same parameters as above: $\beta = 1$, $\eta = 0.06$ and $\varphi = 0.0425$. To get the values of the corresponding COGARCH$(1,1)$ process at unequally spaced time points, we will use the function `prevTick`. As a first step we simulate a COGARCH$(1,1)$ process by the function `cogarch_sim`, then we specify a time grid with unequally spaced times where we would like to observe the process for. In our example, we specified uniform distributed times with values 0.5, 1 and 1.5 which occur with the probabilities 0.3, 0.4 and 0.3. With the `prevTick` function we get the values for these times by previous tick interpolation. We then apply the PML estimation method to obtain estimated parameter values. The outcome of this estimation is displayed in Table 3.7. Compared to the PML estimation we conducted before, see Table 3.5, we get estimation results of similar quality. This shows that the PML method can be applied to data observed at irregularly spaced time points as well. Furthermore, we estimate the parameters with the PML method for the same setting as above, but for a larger period of time, $t \in [0, 20000]$. In Table 3.8 we see that the results of the estimators become slightly better.

In Chapter 3.2.3 we computed $\mathbb{E}(\mathbb{E}(Y_i^2|\mathcal{F}_{t_{i-1}} \cup (\Delta t_i))|\mathcal{F}_{t_{i-1}})$ for a compound Poisson driven COGARCH process. We considered this quantity because if $\mathcal{F}_{t_{i-1}}$ is given, Δt_i is not known. Therefore, we modified the PML estimation function and estimated the parameters. The input parameters are chosen slightly different as before: $\beta = 0.05$, $\eta = 0.06$ and $\varphi = 0.04$. The outcome is summarized in Table 3.9. Moreover $\hat{\lambda}$ has been estimated for each of the samples. In Table 3.10 the outcomes are shown. The mean and the median of the estimates are very close to the input parameter for λ. The bias and MSE values are very small. Therefore, we can conclude to have a good estimator for λ as expected. Having a look at Figure 3.6. we can observe in the boxplots of the estimated parameters that the median (and the mean) of all three parameters lie below the true parameter values. Comparing these boxplots to the boxplots of our examples before, it can be observed that the variation, i.e. the size of the "box", is bigger for all parameters. Before, $\hat{\beta}$ had the largest variation whereas in this example $\hat{\eta}$ and $\hat{\varphi}$ vary more than before.

n=20000	$\hat{\beta}$	$\hat{\eta}$	$\hat{\varphi}$
Mean	0.93120	0.04613	0.03018
Bias	-0.06880	-0.01388	-0.01232
Rel. Bias	-0.06880	-0.23125	-0.28990
MSE	0.03182	0.00022	0.00016
MAE	0.14573	0.01390	0.01232
Median	0.91852	0.04572	0.03013
Bias Median	-0.08148	-0.01428	-0.01237
MSE Median	0.01667	0.00020	0.00015
MAE Median	0.12913	0.01428	0.01237

Table 3.8: Estimated mean, relative bias and MSE for $\hat{\beta}$, $\hat{\eta}$ and $\hat{\varphi}$ of 900 samples with the **PML Method** for data observed at irregularly spaced time points. The true values are $\beta = 1$, $\eta = 0.06$ and $\varphi = 0.0425$.

n=5000	$\hat{\beta}$	$\hat{\eta}$	$\hat{\varphi}$
Mean	0.03991	0.05320	0.03643
Bias	-0.01009	-0.00680	-0.00357
Rel. Bias	-0.20181	-0.11331	-0.08935
MSE	0.00017	0.00009	0.00004
MAE	0.01124	0.00805	0.00491
Median	0.03948	0.05289	0.03635
Bias Median	-0.01052	-0.00711	-0.00365
MSE Median	0.00012	0.00006	0.00002
MAE Median	0.01075	0.00747	0.00428

Table 3.9: Estimated mean, relative bias and MSE for $\hat{\beta}$, $\hat{\eta}$ and $\hat{\varphi}$ of 900 samples with the **"modified PML Method"** for in time **unequally** spaced data (random jumptimes). The true values are $\beta = 0.05$, $\eta = 0.06$ and $\varphi = 0.04$.

Variance Gamma COGARCH$(1,1)$ process

Analogoue to the approach we followed in the compound Poisson case, we proceed in the Variance Gamma COGARCH process. In Tables 3.11 and 3.12 the estimation results for 900 simulations can be found. In both cases the time period has been chosen as t in $[0, 1000]$. The input parameters have been $\beta = 0.04$, $\eta = 0.053$ and $\varphi = 0.038$. For this example we chose the input parameter of β smaller than before, to see if we get better estimation results.

Again we estimated the model parameters by the moment estimator method at first and then by the PML method. The values for $\hat{\beta}$, $\hat{\eta}$ and $\hat{\varphi}$ estimated by the moment estimation method, are close to the true parameter values, see Table 3.11. Having a look at the outcomes of the PML estimation in Table 3.12 the mean values do not look promising. But due to the median values it is obvious that the mean has been influenced by huge outliers. Additionally, this can be observed in the boxplots in Figure 3.7.

Furthermore we generated sample paths of a COGARCH$(1,1)$ process with differ-

n=5000	$\hat{\lambda}$
Mean	0.99991
Bias	-0.00009
Rel. Bias	-0.00009
MSE	0.00018
MAE	0.01054
Median	0.99997
Bias Median	-0.00003
MSE Median	0.00018
MAE Median	0.00869

Table 3.10: Estimated mean, bias, relative bias, MSE, MAE, median, bias median, MSE median, MAE median for $\hat{\lambda}$ and corresponding standard deviations in brackets. The true value is $\lambda = 1$.

n=1000	$\hat{\beta}$	$\hat{\eta}$	$\hat{\varphi}$
Mean	0.04535	0.04477	0.02633
Bias	0.00535	-0.00823	-0.01167
Rel. Bias	0.13386	-0.15524	-0.30702
MSE	0.00301	0.00233	0.00091
MAE	0.04163	0.03879	0.02432
Median	0.02741	0.03670	0.02350
Bias Median	-0.01259	-0.01630	-0.01450
MSE Median	0.00160	0.00159	0.00049
MAE Median	0.04000	0.03992	0.02207

Table 3.11: Estimated mean, relative bias and MSE for $\hat{\beta}$, $\hat{\eta}$ and $\hat{\varphi}$ of 900 samples with the **Moment Estimator Method**. The true values are $\beta = 0.04$, $\eta = 0.053$ and $\varphi = 0.038$.

ent input parameters $\beta = 0.07$, $\eta = 0.06$ and $\varphi = 0.04$ and estimated the parameters of them. We receive the results displayed in Tables 3.13 (moment estimation method) and 3.14 (PML method). In this example the median values of the estimates is closer to the input parameters with the PML estimation method.

n=1000	$\hat{\beta}$	$\hat{\eta}$	$\hat{\varphi}$
Mean	0.14185	0.1291	0.06835
Bias	0.10185	0.07610	0.03035
Rel. Bias	2.54625	1.43582	0.79879
MSE	0.18269	0.22265	0.13558
MAE	0.10783	0.09188	0.05154
Median	0.05786	0.05232	0.02864
Bias Median	0.01786	-0.00068	-0.00936
MSE Median	0.00036	0.00034	0.00022
MAE Median	0.01903	0.01834	0.01493

Table 3.12: Estimated mean, relative bias and MSE for $\hat{\beta}$, $\hat{\eta}$ and $\hat{\varphi}$ of 900 samples with the **PML Method** for in time equally spaced data. The true values are $\beta = 0.04$, $\eta = 0.053$ and $\varphi = 0.038$.

n=1000	$\hat{\beta}$	$\hat{\eta}$	$\hat{\varphi}$
Mean	0.05945	0.04680	0.02944
Bias	-0.01055	-0.01320	-0.01056
Rel. Bias	-0.15076	-0.22004	-0.26392
MSE	0.00526	0.00291	0.00132
MAE	0.05724	0.04194	0.02642
Median	0.03616	0.03704	0.02461
Bias Median	-0.03384	-0.02296	-0.01539
MSE Median	0.00361	0.001560	0.00055
MAE Median	0.06007	0.03949	0.02340

Table 3.13: Estimated mean, relative bias and MSE for $\hat{\beta}$, $\hat{\eta}$ and $\hat{\varphi}$ of 900 samples with the **Moment Estimation Method**. The true values are $\beta = 0.07$, $\eta = 0.06$ and $\varphi = 0.04$.

n=1000	$\hat{\beta}$	$\hat{\eta}$	$\hat{\varphi}$
Mean	0.23024	0.14917	0.03018
Bias	0.16024	0.08917	-0.00982
Rel. Bias	2.28919	1.48620	-0.24552
MSE	0.45940	0.25534	0.00011
MAE	0.17454	0.10809	0.00983
Median	0.10237	0.06015	0.03013
Bias Median	0.03237	0.00015	-0.00987
MSE Median	0.00138	0.00048	0.00010
MAE Median	0.03718	0.02197	0.00987

Table 3.14: Estimated mean, relative bias and MSE for $\hat{\beta}$, $\hat{\eta}$ and $\hat{\varphi}$ of 900 samples with the **PML Method**. The true values are $\beta = 0.07$, $\eta = 0.06$ and $\varphi = 0.04$.

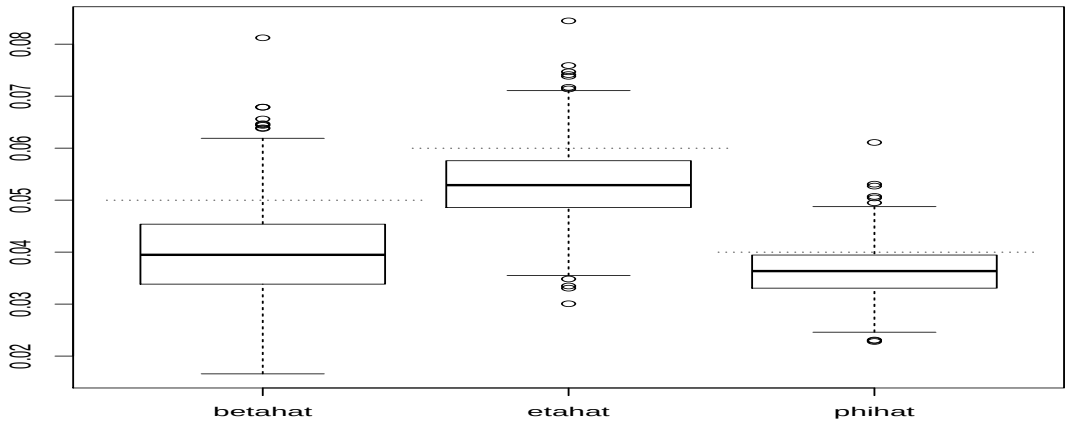


Figure 3.6: Boxplots of the estimated parameters $\hat{\beta}$, $\hat{\eta}$ and $\hat{\varphi}$ of 900 samples estimated with the modified PML Method. The true values are $\beta = 0.05$, $\eta = 0.06$ and $\varphi = 0.04$.

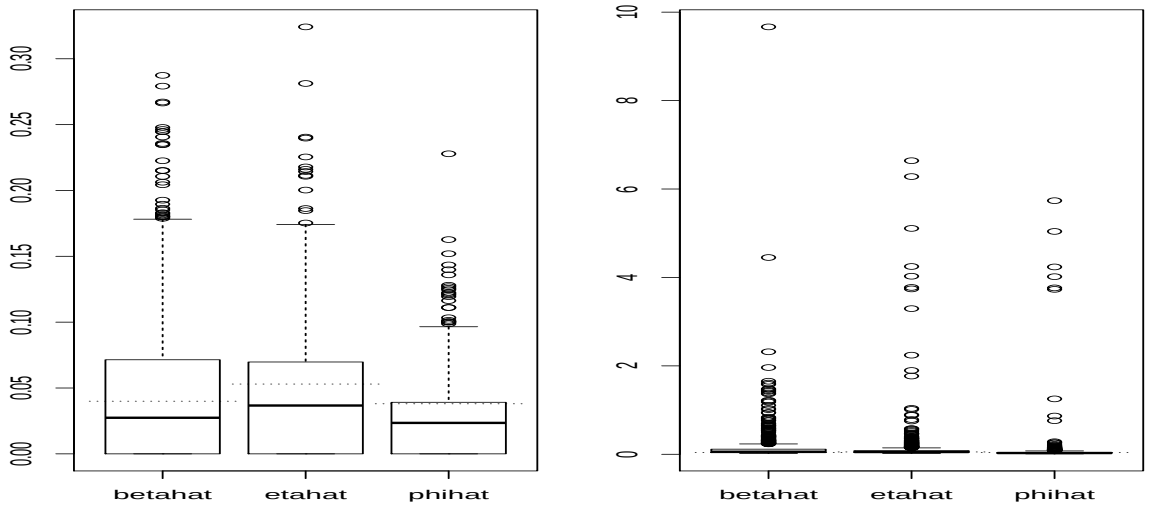


Figure 3.7: Boxplots of the estimated parameters $\hat{\beta}$, $\hat{\eta}$ and $\hat{\varphi}$ of 900 samples estimated with the Moment Estimator Method (*left*) and with the PML Method (*right*). The true values are $\beta = 0.04$, $\eta = 0.053$ and $\varphi = 0.038$.

Chapter 4

Exponential COGARCH Processes

When observing financial data, e.g stock returns, some main characteristics can be found, see Remark 3.1. One of these is that stock returns seem to be negatively correlated with changes in the volatility, i.e. that volatility tends to increase after neagtive shocks and to decrease after positive ones. This effect is called **leverage effect** and can not be modeled by a GARCH type process without further extensions.[23] In order to model this effect (the leverage effect), Nelson (1991) defined the exponential GARCH (EGARCH) process.
This whole chapter is based on the Paper of Haug and Czado (2007).

4.1 Discrete Time EGARCH Process

Definition 4.1 *(Haug and Czado (2007, p.2,3))*
The process $(X_n)_{n\in\mathbb{Z}}$ of the form $X_n = \sigma_n\epsilon_n$, $n \in \mathbb{Z}$, where $(\epsilon_n)_{n\in\mathbb{Z}}$ is an i.i.d. sequence with $\mathbb{E}(\epsilon_1) = 0$ and $\mathbb{V}ar(\epsilon_1) = 1$, is called an EGARCH process, if the volatility process $(\sigma_n)_{n\in\mathbb{Z}}$ satisfies

$$\log(\sigma_n^2) = \mu + \sum_{k=1}^{\infty} \beta_k f(\epsilon_{n-k}),$$

where $f : \mathbb{R} \to \mathbb{R}$ is some measurable real valued deterministic function, $\mu \in \mathbb{R}$ and $(\beta_k)_{k\in\mathbb{N}}$ are real coefficients such that $\mathbb{E}(|f(\epsilon_n)|) < \infty$, $\mathbb{V}ar(f(\epsilon_n)) < \infty$ and $\sum_{k=1}^{\infty} |\beta_k| < \infty$.

Furthermore, Nelson (1991) suggested a finite parameter model by modeling the log-volatility as an ARMA$(q, p-1)$ process instead of an infinite moving average process. This leads to the EGARCH(p, q) model, which is defined as follows.

Definition 4.2 *(Haug and Czado (2007, p.3))*
Let $p, q \in \mathbb{N}$, $\mu, \alpha_1, \ldots, \alpha_q\beta_1, \ldots, \beta_p \in \mathbb{R}$, suppose $\alpha_q \neq 0$, $\beta_p \neq 0$ and that the autoregressive polynomial $\phi(z) := 1 - \alpha_1 z - \cdots - \alpha_q z^q$ and the moving average polynomial $\psi(z) := \beta_1 + \beta_2 z + \cdots + \beta_p z^{p-1}$ have no common zeros and that $\phi \neq 0$ on $\{z \in \mathbb{C} | |z| \leq 1\}$.

[23] cf. Haug and Czado (2007), Introduction

Let $(\epsilon_n)_{n\in\mathbb{Z}}$ be an i.i.d. sequence with $\mathbb{E}(\epsilon_1) = 0$ and $\mathbb{V}ar(\epsilon_1) = 1$, and let $f(\cdot)$ be such that $\mathbb{E}(|f(\epsilon_n)|) < \infty$ and $\mathbb{V}ar(f(\epsilon_n)) < \infty$. Then $(X_n)_{n\in\mathbb{Z}}$, where $X_n = \sigma_n\epsilon_n$ and

$$\log(\sigma_n^2) = \mu + \sum_{k=1}^{p} \beta_k f(\epsilon_{n-k}) + \sum_{k=1}^{q} \alpha_k \log(\sigma_{n-k}^2) \tag{4.1}$$

*is called an **EGARCH**(p, q) process.*

Due to Nelson (1991), $f(\epsilon_n)$ must be a function of the magnitude and the sign of ϵ_n, in order to achieve the asymmetric relation between the stock returns and the volatility. Therefore he proposed the following function

$$f(\epsilon_n) := \theta\epsilon_n + \gamma[|\epsilon_n| - \mathbb{E}(|\epsilon_n|)], \tag{4.2}$$

with real coefficients θ and γ. We see that $f(\epsilon_n)$ is piecewise linear in ϵ_n and has slope $\theta + \gamma$ for positive shocks ϵ_n and $\theta - \gamma$ for negative ones. Therefore $f(\epsilon_n)$ allows the volatility process $(\sigma_n^2)_{n\in\mathbb{Z}}$ to respond asymmetrically to positive and negative jumps in the stock price.

4.2 Exponential COGARCH Process

Analogously to the idea of Klüppelberg et al. (2004) where the continuous time GARCH process has been constructed based on the discrete time GARCH process, Haug and Czado (2007) developed a continuous time analogue of the discrete time EGARCH(p, q) process. As in Klüppelberg et al. (2004), we will replace the noise variables ϵ_n by the increments of a Lévy process $L = (L_t)_{t\geq 0}$.

We consider a Lévy process L defined on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ with jumps $\Delta L_t = L_t - L_{t-}$, zero mean and finite variance. Recall from (2.14) the definition of the compensated random measure associate to the Poisson random measure: $\tilde{N}_L(t, dx) = N_L(x, dx) - t\nu_L(dx)$, $t \geq 0$.

For the continuous time model the driving noise process will be constructed similar to the discrete time case. For a Lévy process L with $\mathbb{E}(L_1) = 0$ and $\mathbb{E}(L_1^2) < \infty$ the driving process M of the log-volatility process is defined by

$$M_t := \int_{\mathbb{R}\setminus\{0\}} |h(x)|\tilde{N}_L(t, dx), \quad t \geq 0, \tag{4.3}$$

with $h(x) := \theta x + \gamma|x|$ and $\theta, \gamma \in \mathbb{R}$.

Remark 4.3 *(Haug and Czado (2007, Remark 1))*
(i) The process M defined by (4.3) *is by construction a process with independent and stationary increments and by Theorem 4.3.4 in Applebaum (2009) well defined if*

$$\int_{\mathbb{R}} |h(x)|^2 \nu_L(dx) < \infty. \tag{4.4}$$

Condition (4.4) *is satisfied since ν_L is a Lévy measure and L has finite variance. By Equation (2.9) of Applebaum (2009) the characteristic triplet of M is $(\gamma_M, 0, \nu_M)$, where $\nu_M := \nu_L \circ h^{-1}$*

is the Lévy measure of M and $\gamma_M := -\int_{|x|>1} x\nu_M(dx)$. *The precise form of* ν_M *depends on the sign and size of* θ *and* γ *and is given in the following formulas:*

$$\nu_M((-\infty,-x]) = \begin{cases} \nu_L([-\frac{x}{\theta+\gamma},\infty)) + \nu_L((-\infty,-\frac{x}{\theta-\gamma}]), & -\gamma > \theta > \gamma \\ \nu_L((-\infty,-\frac{x}{\theta-\gamma}]), & -\theta < \gamma < \theta \\ \nu_L([-\frac{x}{\theta+\gamma},\infty)) & -\theta > \gamma > \theta \\ 0 & -\gamma < \theta < \gamma \end{cases}$$

and

$$\nu_M([x,\infty)) = \begin{cases} \nu_L([\frac{x}{\theta+\gamma},\infty)) + \nu_L((-\infty,\frac{x}{\theta-\gamma}]), & -\gamma < \theta < \gamma \\ \nu_L((-\infty,\frac{x}{\theta-\gamma}]), & -\theta > \gamma > \theta \\ \nu_L([\frac{x}{\theta+\gamma},\infty)) & -\theta < \gamma < \theta \\ 0 & -\gamma > \theta > \gamma \end{cases}$$

for $x > 0$. *One recognizes that M is a spectrally negative*[24] *Lévy process for* $\gamma < \theta < -\gamma$, *i.e. M has only negative jumps, and a spectrally positive Lévy process for* $-\gamma < \theta < \gamma$.
(ii) In case the jump part of L is of finite variation, M is a Lévy process of finite variation with Lévy-Itô decomposition

$$M_t := \sum_{0<s\le t} [\theta\Delta L_s + \gamma|\Delta L_s|] - Ct, \quad t > 0,$$

where $C := \gamma\int_{\mathbb{R}} |x|\nu_L(dx)$.

In the following definition the exponential COGARCH(p,q) process is stated by specifying the log-volatility process as a continuous time ARMA$(q,p-1)$ process, denoted by CARMA(p,q) process[25].

Definition 4.4 *(Haug and Czado (2007, Definition 1))*
Let L be a zero mean Lévy process with Lévy measure ν_L *such that* $\int_{|x|\ge 1} x^2\nu_L(dx)$. *Then we define the exponential COGARCH*(p,q) *process G, shortly ECOGARCH*(p,q), *as the stochastic process satisfying*

$$dG_t := \sigma_{t-}dL_t,\ t > 0,\ G_0 = 0, \tag{4.5}$$

where the log-volatility process $\log(\sigma^2) = (\log(\sigma_t^2))_{t\ge 0}$ *is a CARMA*$(q,p-1)$ *process,* $1 \le p \le q$, *with mean* $\mu \in \mathbb{R}$ *and state space representation*

$$\log(\sigma_t^2) := \mu + b^\top X_t, \quad t > 0, \quad \log(\sigma_0^2) = \mu + b^\top X_0 \tag{4.6}$$

$$dX_t := AX_t dt + 1_q dM_t, \quad t > 0 \tag{4.7}$$

[24] A spectrally negative Lévy process X is a process that contains only negative jumps, i.e. $\nu_X((-\infty,0]) = 0$ and a spectrally positive process is a process that contains only positive jumps, i.e. $\nu_X([0,\infty)) = 0$.

[25] For details on CARMA processes see e.g.Brockwell and Marquardt (2005).

where $X_0 \in \mathbb{R}^q$ is independent of the driving Lévy process M. The $q \times q$ matrix A and the vectors $b \in \mathbb{R}^q$ and $1_q \in \mathbb{R}^q$ are defined by

$$A = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -a_q & -a_{q-1} & -a_{q-2} & \dots & -a_1 \end{bmatrix}, b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{q-1} \\ b_q \end{bmatrix}, 1_q = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \tag{4.8}$$

with coefficients $a_1, \dots, a_q, b_1, \dots, b_p \in \mathbb{R}$, where $a_q \neq 0$, $b_p \neq 0$ and $b_{p+1} = \dots = b_q = 0$.

As defined in Chapter 3, we refer to G and $G^{(r)}$ as the (log-)price process and (log-) return process, respectively. Also σ^2 and $\log(\sigma^2)$ will be the notation for the volatility process and log-volatility process, respectively.

Remark 4.5 *(Haug and Czado (2007, Remark 2)) The solution of the continuous time state space model* (4.6) *and* (4.7) *has the representation*

$$\log(\sigma_t^2) = \mu + \boldsymbol{b}^\top e^{At} \boldsymbol{X}_0 + \int_0^t \boldsymbol{b}^\top e^{A(t-u)} 1_q dM_u, \quad t > 0.$$

The next Proposition gives conditions for the strict stationarity of $\log(\sigma^2)$ and σ^2.

Proposition 4.6 *(Haug and Czado (2007, Proposition 3.1))*
Let σ^2 and G be as in Definition 4.4, with θ and γ not both equal to zero. If the eigenvalues of A all have negative real parts and X_0 has the same distribution as $\int_0^\infty e^{Au} 1_q dM_u$, then $\log(\sigma^2)$ and σ^2 are strictly stationary.

Corollary 4.7 *(Haug and Czado (2007, Corollary 1))*
If σ^2 is strictly stationary, then G has strictly stationary increments.

The following remark summarizes some properties of the volatility process. For the simulation of an ECOGARCH(p,q) process we have to check if the parameters need to satisfy any constraints in order to get a positive volatility process. Furthermore, it is of interest in which case the volatility process contains jumps.

Remark 4.8 *(Haug and Czado (2007, Remark 3))*
(i) If $q \geq p+1$ the log-volatility process is $(q-p-1)$ times differentiable, which follows from the state space representation of $\log(\sigma^2)$, and hence the volatility process has continuous sample paths. In particular the volatility will only contain jumps for $p = q$.
(ii) The volatility of the ECOGARCH(p,q) process is positive by definition. Therefore, the parameters do not need to satisfy any constraints to assure positivity of the volatility. This is not the case for the COGARCH(p,q) model, see Chapter 3.1.4. For higher order COGARCH(p,q) processes these conditions become quite difficult to check (see Theorem 5.1 in Brockwell et al. (2006)).

4.3 Examples of ECOGARCH(p,q) processes

Our first example is an ECOGARCH$(1,1)$ process[26]. The process is driven by a Lévy process L with Lévy process symbol

$$\Psi_L(u) = -\frac{u^2}{2} + \int_{\mathbb{R}} (e^{iux} - 1)\lambda \Phi_{0,1/\lambda}(dx),$$

where $\Phi_{0,1/\lambda}(\cdot)$ is the distribution function of a normal distribution with mean 0 and variance $1/\lambda$. This means that L is the sum of a standard Brownian motion W and the compound Poisson process $J_t = \sum_{k=1}^{N_t} Z_k$, $t \geq 0$, where $(N_t)_{t\in\mathbb{R}}$ is an independent Poisson process with intensity $\lambda > 0$ and jump times $(T_k)_{k\in\mathbb{Z}}$. The Poisson process N is also independent from the i.i.d. sequence of jump sizes $(Z_k)_{k\in\mathbb{Z}}$, with $Z_1 \sim \mathcal{N}(0, 1/\lambda)$. The Lévy process M is then given by

$$M_t = \sum_{k=1}^{N_t} [\theta Z_k + \gamma |Z_k|] - Ct, \quad t > 0,$$

with

$$\begin{aligned}
C &= \gamma \int \mathbb{R}|x|\lambda \Phi_{0,1/\lambda}(dx) \\
&= \gamma \int \mathbb{R}|x|\lambda \frac{1}{\sqrt{2\pi\frac{1}{\lambda}}} \exp\left\{-\frac{\lambda x^2}{2}\right\} dx \\
&= 2\gamma \int_0^\infty x \frac{\lambda}{\sqrt{2\pi\frac{1}{\lambda}}} \exp\left\{-\frac{\lambda x^2}{2}\right\} dx \\
&= \gamma \frac{2}{\sqrt{2\pi\frac{1}{\lambda}}} \underbrace{\int_0^\infty x\lambda \exp\left\{-\frac{\lambda x^2}{2}\right\} dx}_{\left[-\exp\{-\frac{\lambda x^2}{2}\}\right]_0^\infty = 1} \\
&= \sqrt{\frac{2\lambda}{\pi}}\gamma.
\end{aligned}$$

M_{-t}, $t \geq 0$ is defined analogously. If we just consider the case where $\theta < -\gamma < 0$, then the Lévy measure ν_M of M is defined by

$$\nu_M((-\infty, -x]) = \lambda \Phi_{0,1/\lambda}\left([-\frac{x}{\theta+\gamma}, \infty)\right), \quad x > 0,$$

on the negative half real line and by

$$\nu_M([x, \infty)) = \lambda \Phi_{0,1/\lambda}((-\infty, \frac{x}{\theta-\gamma}]), \quad x > 0,$$

[26]This example is based on Example 1. of Haug and Czado (2007).

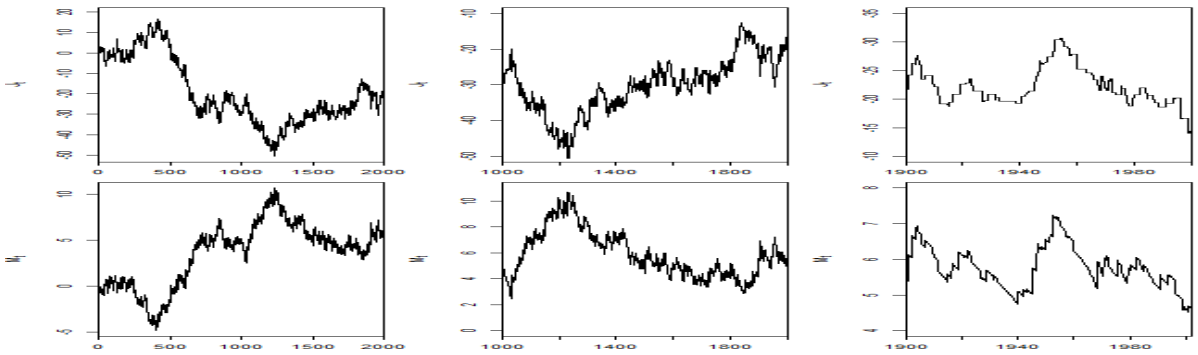


Figure 4.1: Simulated sample paths of J (*upper row*) and M (*lower row*), with parameters $\theta = -0.2$, $\gamma = 0.1$, $\lambda = 1$ and standard normally distributed jumps over three different time scales.

on the positive half real line.

Figure 4.1 displayes the sample paths of the processes J and M over three different time scales. The upper row of Figure 4.1 is a compound Poisson process with standard normally distributed jumps and intensity $\lambda = 1$. The lower row shows the process M with parameters $\theta = -0.2$ and $\gamma = 0.1$. It can be observed that if J jumps upwards, then M jumps down and the other way round. If J is constant, then M moves down due to the drift.

With Remark 4.5 the log-volatility process is of the form

$$\begin{aligned}\log(\sigma_t^2) &= \mu + b_1 e^{-a_1 t} X_0 + \int_0^t b_1 e^{-a_1(t-s)} dM_s \\ &= \mu + b_1 e^{-a_1 t} X_0 + \sum_{k=1}^{N_t} b_1 e^{-a_1(t-T_k)} [\theta Z_k + \gamma |Z_k|] - C\frac{b_1}{a_1}(1 - e^{-a_1 t}),\end{aligned}$$

for $t > 0$, and the log-price process is given by

$$G_t = \int_0^t \sigma_{s-} dW_s \sum_{k=1}^{N_t} \sigma_{T_k -} Z_k, \quad t > 0, \quad G_0 = 0,$$

with jump times T_k, $k \in \mathbb{N}$.

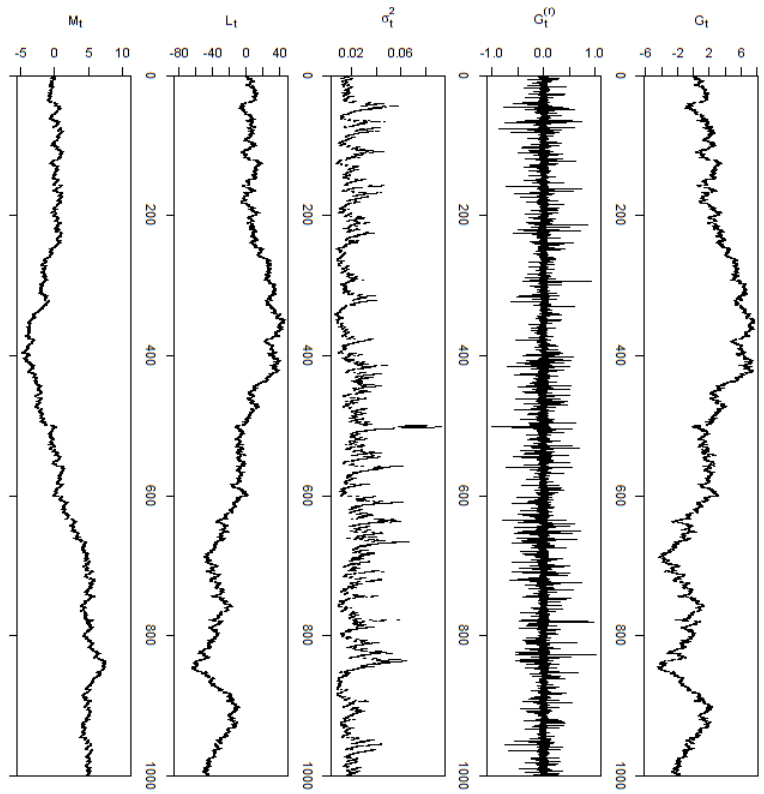
For the simulation of a sample path of the log-price process G and the log-volatility process $\log(\sigma^2)$ over the time interval $[0, T]$ we will follow the steps of the Algorithm implemented in Chapter 6.3. The results of such a simulation are shown in Figure 4.2(a). Therefore, we use the simulation of an ECOGARCH$(1,1)$ process driven by a compound Poisson Lévy process with parameters $a_1 = -0.1$, $b_1 = 1$, $\mu = -4$, $\theta = -0.2$, $\gamma = 0.1$, $\lambda = 1$ and standard normally distributed jumps from the example above (cf. Figure 4.1). Comparing the path of log-price process $G = (G_t)_{t \geq 0}$ with the paths of the driving Lévy process $L = (L_t)_{t \geq 0}$ we can see that G jumps at the same time as L does. Furthermore comparing L and $M = (M_t)_{t \geq 0}$ we can observe the asymmetry, i.e. if L jumps up, then M jumps down and the other way round. From the paths of the returns $G_t^{(r)}$ and the volatility process σ_t^2, we can see that the volatility increases after large

negative returns and decreases after a large positive return. This effect is known as the leverage effect which has been introduced at the beginning of Chapter 4. Moreover, we simulated a sample path of an ECOGARCH process with higher order of p and q. As an example a sample path of an ECOGARCH$(2, 2)$ process driven by a compound Poisson Lévy process with parameters $a_1 = -0.1$, $b_1 = 1$, $\mu = -4$, $\theta = -0.2$, $\gamma = 0.1$, $\lambda = 1$ and standard normally distributed jumps is shown in Figure 4.2(b).

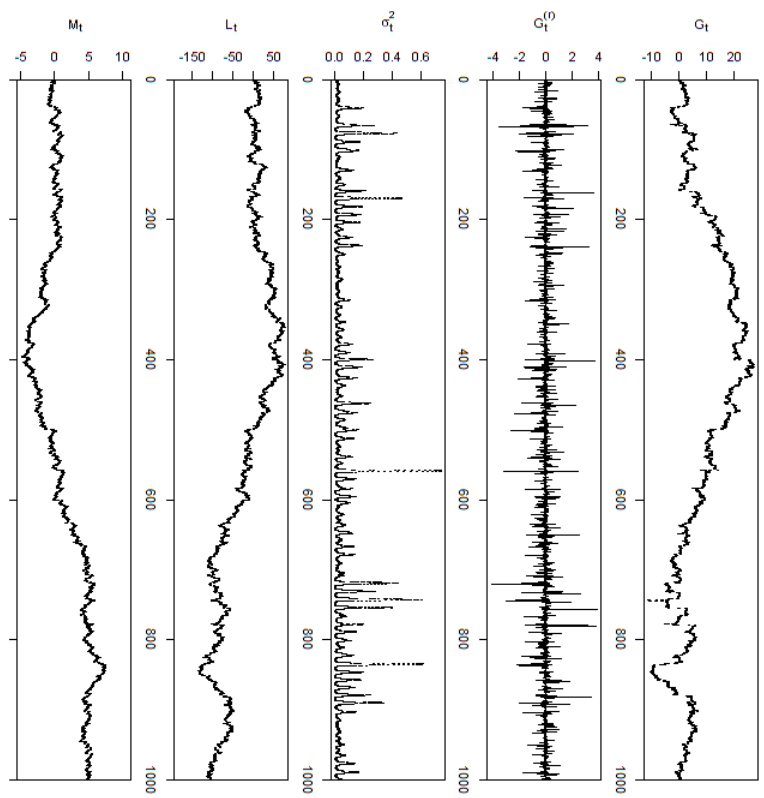
The two examples before assumed p and q of the same order. In order to see what is happening for $p < q$, we simulated an ECOGARCH process with $p = 2$ and $q = 3$, which can be found in Figure 4.3.

Due to Remark 4.8(i) the volatility will only contain jumps for $p = q$. In order to see this property, we have a closer look at the volatility processes of two ECOGARCH processes, one with $p = q$ and the other one with $q > p$. In Figure 4.4 the volatility processes of an ECOGARCH$(1, 1)$ process *(left)* and of an ECOGARCH$(2, 3)$ process *(right)* can be observed for two different time scales. We examine the volatility processes for jumps. Especially in the bottom row of Figure 4.4, it is obvious that the volatility of the ECOGARCH process for $p = q = 1$ *(left)* contains jumps, but for $p = 2$ and $q = 3$ *(right)* it does not.

For the simulation of an ECOGARCH(p, q) process we can use the function `ecogarch_sim`. First, we have to choose some parameters. They should be chosen in a way that we get a strictly stationary volatility process. Therefore we are going to check that the eigenvalues of A, which includes the chosen parameters $a_1, \ldots, a_q$, all have negative real parts. Due to Proposition 4.6 the volatility is then strictly stationary.



(a) Simulation of a compound Poisson driven ECOGARCH(1,1) process with parameters $a_1 = -0.1$, $b_1 = 1$, $\mu = -4$, $\theta = -0.2$, $\gamma = 0.1$, $\lambda = 1$ and standard normally distributed jumps.



(b) Simulation of a compound Poisson driven ECOGARCH(2,2) process with parameters $a_1 = -0.1$, $a_2 = -0.3$, $b_1 = b_2 = 1$, $\mu = -4$, $\theta = -0.2$, $\gamma = 0.1$, $\lambda = 1$ and standard normally distributed jumps.

Figure 4.2: Simulations of an ECOGARCH$(1,1)$ process (*left*) and an ECOGARCH$(2,2)$ process (*right*).

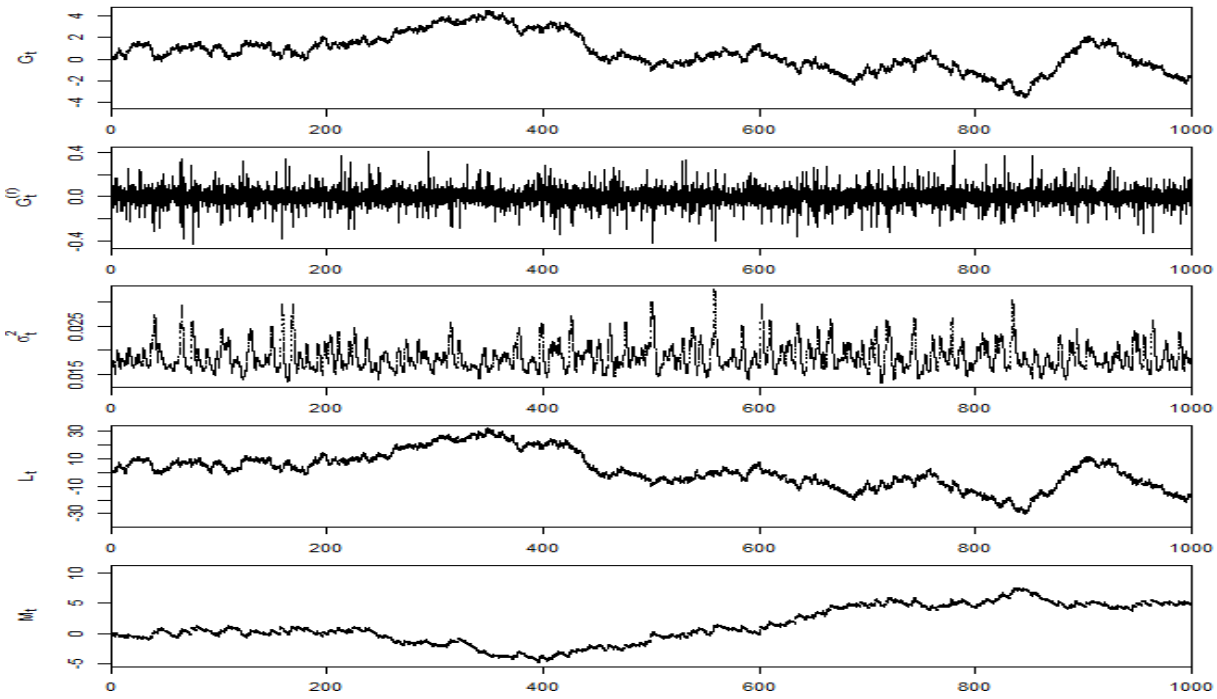


Figure 4.3: Simulation of a compound Poisson driven ECOGARCH$(2, 3)$ process with parameters $a_1 = -3$, $a_2 = -2$, $a_3 = -1$, $b_1 = b_2 = 1$, $\mu = -4$, $\theta = -0.2$, $\gamma = 0.1$, $\lambda = 1$ and standard normally distributed jumps.

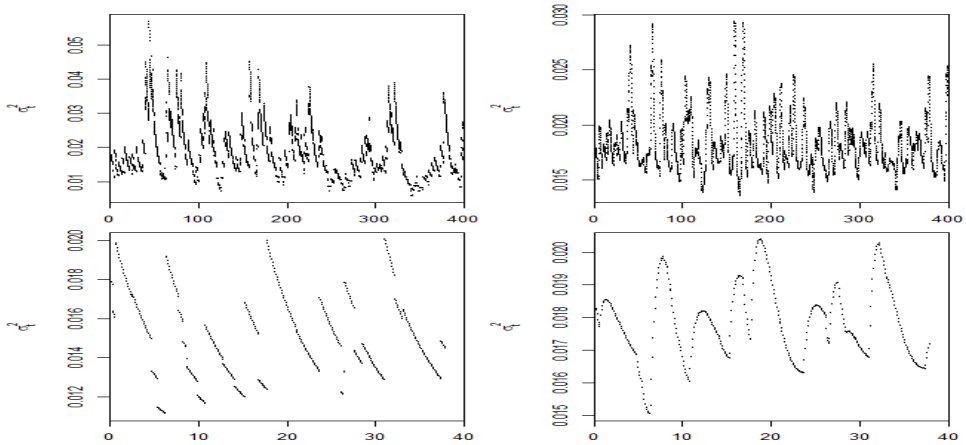


Figure 4.4: Comparison of the volatility processes of an ECOGARCH$(1, 1)$ *(left)* and an ECOGARCH$(2, 3)$ *(right)* process for different time scales. The parameters for the ECOGARCH$(1, 1)$ are $a_1 = -0.1$, $b_1 = 1$ and for the ECOGARCH$(2, 3)$ $a_1 = -3$, $a_2 = -2$, $a_3 = -1$, $b_1 = b_2 = 1$ and for both cases $\mu = -4$, $\theta = -0.2$, $\gamma = 0.1$, $\lambda = 1$.

Chapter 5

Asymmetric COGARCH Processes

In this chapter, some asymmetric continous time extensions of the GARCH model will be considered. We focus especially on the (asymmetric) GJR-COGARCH$(1,1)$ model. This chapter is based on the master's thesis of Mayr (2013). We refer to this thesis for more details and the proofs of the following.

5.1 APARCH Process

5.1.1 Discrete Time APARCH

The (symmetric) GARCH model cannot model the so-called leverage effect which can be found in return data. Due to Nelson (1991), the leverage effect is the phenomenon that a negative shock increases the future volatility more than a positive one of the same size. In order to take this effect into account several discrete time models have been introduced. Nelson (1991) developed for example the exponential GARCH model (EGARCH), see Chapter 4. Furthermore the Threshold GARCH model was introcduced by Rabemananjara and Zakoian (1993) and Zakoian (1994), and Glosten et al. (1993) suggested the so-called GJR-Modell. In Ding et al. (1993) they introduced the Asymmetric Power ARCH (APARCH) model. This model includes asymmetric models like the GJR and the Threshold GARCH model but also the symmetric GARCH model. [27] It is defined as follows.

Definition 5.1 *(?, Section 6)*
Let $(\varepsilon_n)_{n\in\mathbb{N}}$ be a sequence of i.i.d. random variables such that $\mathbb{E}(\varepsilon_n) = 0$ and $\mathbb{V}ar(\varepsilon_n) = 1$. The process $(Y_n)_{n\in\mathbb{N}}$ is called Asymmetric Power ARCH(p,q), (APARCH(p,q)) if it is satisfying an equation of the following form.

$$Y_n = \varepsilon_n \sigma_n \tag{5.1}$$

$$\sigma_n^\delta = \theta + \sum_{i=1}^{q} \alpha_i h(Y_{n-i}) + \sum_{j=1}^{p} \beta_j \sigma_{n-j}^\delta, \tag{5.2}$$

with $h(x) = (|x| - \gamma x)^\delta$, $\theta > 0$, $\delta > 0$, $\alpha_i \geq 0$, $\beta_i \geq 0$ and $|\gamma_i| < 1$.

[27] cf. Chapter 4 in Mayr (2013).

Remark 5.2 *(Mayr (2013, Remark 4.2))*
The function $h(x) = (|x| - \gamma x)^\delta$ from Definition 5.1 is strictly positive for all $x \in \mathbb{R}\backslash\{0\}$ and $\delta > 0$, as due to the condition $|\gamma| < 1$ it holds $|x| > \gamma x$.

The APARCH model, defined in (5.1) and (5.2) with $h(x) = (|x| - \gamma x)^\delta$ includes the following models.

- **GARCH** model: For $\delta = 2$, $\gamma_i = 0$ for $i = 1, \dots, q$ and $h(x) = x^2$ we obtain the discrete time GARCH model.
- **Threshold GARCH** model: By inserting $\delta = 1$ in $h(x)$ it follows $h(x) = (|x| - \gamma x)$ and
$$\sigma_n = \theta + \sum_{i=1}^{q} \alpha_i^+ Y_{n-i} \mathbb{1}_{\{Y_{n-i}>0\}} - \sum_{i=1}^{q} \gamma_i^- Y_{n-i} \mathbb{1}_{\{Y_{n-i}<0\}} + \sum_{j=1}^{p} \beta_j \sigma_{n-j},$$
with $\alpha_i^+ = \alpha_i(1-\gamma_i)$ and $\gamma_i^- = \alpha_i(1+\gamma_i)$, see Rabemananjara and Zakoian (1993) and Zakoian (1994).
- **GJR** model: For $\delta = 2$ we have $h(x) = (|x| - \gamma x)^2$ and get the GJR model from Glosten et al. (1993) which is named after its authors Glosten, Jagannathan und Runkle.
For $0 \le \gamma_i < 1$ we obtain the GJR model,
$$\sigma_n^2 = \theta + \sum_{i=1}^{q} \alpha_i^* Y_{n-i}^2 + \sum_{j=1}^{p} \beta_j \sigma_{n-j}^2 + \sum_{i=1}^{q} \gamma_i^* \mathbb{1}_{\{Y_{n-i}<0\}} Y_{n-i}^2,$$
with $\alpha_i^* = \alpha_i(1-\gamma_i)^2$ and $\gamma_i^* = 4\alpha_i\gamma_i$.
For $-1 < \gamma_i < 0$ the GJR model is given by
$$\sigma_n^2 = \theta + \sum_{i=1}^{q} \alpha_i^* Y_{n-i}^2 + \sum_{j=1}^{p} \beta_j \sigma_{n-j}^2 + \sum_{i=1}^{q} \gamma_i^* \mathbb{1}_{\{Y_{n-i}>0\}} Y_{n-i}^2,$$
with $\alpha_i^* = \alpha_i(1+\gamma_i)^2$ and $\gamma_i^* = -4\alpha_i\gamma_i$.

5.1.2 Continuous Time APARCH

As we have seen in Chapter 3.1, it is possible to construct a continuous time GARCH model from a discrete time GARCH model. The same approach can be used to get a continuous time APARCH model based on a discrete time APARCH model. This has been shown in Lee (2010) and Lee (2012), where also some properties of this continuous time version can be found. Based on these papers Mayr (2013) states the APARCH$(1,1)$ model with its moments, stationarity and mixing properties in Chapter 4.2.

Inserting $p = q = 1$ in the discrete time APARCH(p,q) equations (5.1) and (5.2) leads to the discrete time APARCH$(1,1)$ process,

$$Y_n = \varepsilon_n \sigma_n, \quad \sigma_n^\delta = \theta + \alpha \left(|Y_{n-1}| - \gamma Y_{n-1}\right)^\delta + \beta \sigma_{n-1}^\delta, \tag{5.3}$$

where $h(x) = (|x| - \gamma x)^\delta$, $\theta > 0$, $\delta > 0$, $\alpha \geq 0$, $\beta \geq 0$ and $|\gamma| < 1$.

The idea is to proceed as in the GARCH case. That is to replace the innovations of the discrete time APARCH through the increments of a Lévy process. Therefore, the volatility process should relay only on the innovations:

$$\begin{aligned}
\sigma_n^\delta &= \theta + \alpha \left(|Y_{n-1}| - \gamma Y_{n-1}\right)^\delta + \beta \sigma_{n-1}^\delta \\
&= \theta + \alpha \left(|\varepsilon_{n-1}\sigma_{n-1}| - \gamma \varepsilon_{n-1}\sigma_{n-1}\right)^\delta + \beta \sigma_{n-1}^\delta \\
&= \theta + \alpha \left(|\varepsilon_{n-1}| - \gamma \varepsilon_{n-1}\right)^\delta \sigma_{n-1}^\delta + \beta \sigma_{n-1}^\delta \\
&= \theta + \left(\alpha h(\varepsilon_{n-1}) + \beta\right) \sigma_{n-1}^\delta. \qquad (5.4)
\end{aligned}$$

Through iteration of (5.4) it follows

$$\begin{aligned}
\sigma_n^\delta &= \theta + \left(\beta + \alpha h(\varepsilon_{n-1})\right) \sigma_{n-1}^\delta \\
&= \theta + \left(\beta + \alpha h(\varepsilon_{n-1})\right) \left(\theta + \left(\beta + \alpha h(\varepsilon_{n-2})\right) \sigma_{n-2}^\delta\right) \\
&= \cdots \\
&= \theta \sum_{i=0}^{n-1} \prod_{j=i+1}^{n-1} \left(\beta + \alpha h(\varepsilon_j)\right) + \sigma_0^\delta \prod_{j=0}^{n-1} \left(\beta + \alpha h(\varepsilon_j)\right) \\
&= \theta \int_0^n \exp\left(\sum_{j=\lfloor s \rfloor + 1}^{n-1} \log\left(\beta + \alpha h(\varepsilon_j)\right) \right) ds + \sigma_0^\delta \exp\left(\sum_{j=0}^{n-1} \log\left(\beta + \alpha h(\varepsilon_j)\right) \right) \\
&= \theta \int_0^n \exp\left(\sum_{j=\lfloor s \rfloor + 1}^{n-1} \left(\log \beta + \log\left(1 + \frac{\alpha}{\beta} h(\varepsilon_j)\right)\right)\right) ds \\
&\quad + \sigma_0^\delta \exp\left(\sum_{j=0}^{n-1} \left(\log \beta + \log\left(1 + \frac{\alpha}{\beta} h(\varepsilon_j)\right)\right)\right) \\
&= \theta \int_0^n \exp\left((n - \lfloor s \rfloor) \log \beta + \sum_{j=\lfloor s \rfloor + 1}^{n-1} \left(\log\left(1 + \frac{\alpha}{\beta} h(\varepsilon_j)\right)\right)\right) ds \\
&\quad + \sigma_0^\delta \exp\left(\sum_{j=0}^{n-1} \left(\log \beta + \log\left(1 + \frac{\alpha}{\beta} h(\varepsilon_j)\right)\right)\right) \qquad (5.5)
\end{aligned}$$

Then, in order to obtain a continuous time version like for the COGARCH process in Chapter 3.1, the innovations ε_j are replaced by the increments of the Lévy process $L = (L_t)_{t \geq 0}$ with Lévy measure $\nu_L \neq 0$. Regarding (5.5), a càdlàg process is defined as follows

$$X_t = -t \log \beta - \sum_{0 < s \leq t} \log\left(1 + \left(\frac{\alpha}{\beta}\right) h(\Delta L_s)\right), \quad t \geq 0, \qquad (5.6)$$

where $\theta > 0$, $\alpha > 0$, $0 < \beta < 1$, $|\gamma| < 1$, $\delta > 0$ and $h(x) = (|x| - \gamma x)^\delta$. With σ_0^δ a finite positive random variable independent of $(L_t)_{t \geq 0}$, define the left-continuous volatility

process analogously to (5.5) by

$$\sigma_t^\delta = \left(\theta \int_0^t e^{X_s} ds + \sigma_0^\delta\right) e^{-X_{t-}}, \quad t \geq 0. \tag{5.7}$$

Define the integrated continuous time APARCH$(1,1)$ process $G = (G_t)_{t\geq 0}$ as the càdlàg process satisfying

$$dG_t = \sigma_{t-} dL_t, \quad t \geq 0, \quad G_0 = 0. \tag{5.8}$$

The process G jumps at the same time as L does, with jump size $\Delta G_t = \sigma_t \Delta L_t$, $t \geq 0$.

Continuous Time GJR GARCH Process

It is possible to obtain a continuous time version of the discrete time GJR GARCH model with the method of Klüppelberg et al. (2004). In Chapter 6.2, Mayr (2013) showed the following. Recall from Definition 5.1 the discrete time APARCH$(1,1)$ process. For $\delta = 2$ it includes the GJR GARCH model, cf. Ding et al. (1993). This can be shown by setting the parameters $p = q = 1$, $\delta = 2$, $\theta = a > 0$, $\alpha = b > 0$, $\beta = c \in (0,1)$ and $\gamma = d \in (0,1)$. Equation (5.2) implies

$$\begin{aligned}
\sigma_i^2 &= a + b\left(|Y_{i-1}| - dY_{i-1}\right)^2 + c\sigma_{i-1}^2 \\
&= a + b\left(|\varepsilon_{i-1}\sigma_{i-1}| - d\varepsilon_{i-1}\sigma_{i-1}\right)^2 + c\sigma_{i-1}^2 \\
&= a + b\varepsilon_{i-1}^2\sigma_{i-1}^2 - 2bd|\varepsilon_{i-1}\sigma_{i-1}|\varepsilon_{i-1}\sigma_{i-1} + bd^2\varepsilon_{i-1}^2\sigma_{i-1}^2 + c\sigma_{i-1}^2 \\
&= a + b(1-d)^2\varepsilon_{i-1}^2\sigma_{i-1}^2\mathbb{1}_{\{\varepsilon_{i-1}\geq 0\}} + b(1+d)^2\varepsilon_{i-1}^2\sigma_{i-1}^2\mathbb{1}_{\{\varepsilon_{i-1}<0\}} + c\sigma_{i-1}^2.
\end{aligned}$$

For $b^* = b(1-d)^2 > 0$ and $d^* = b(1+d)^2 > 0$ the continuous time GJR GARCH process follows.

$$\sigma_i^2 = a + b^*\varepsilon_{i-1}^2\sigma_{i-1}^2\mathbb{1}_{\{\varepsilon_{i-1}\geq 0\}} + d^*\varepsilon_{i-1}^2\sigma_{i-1}^2\mathbb{1}_{\{\varepsilon_{i-1}<0\}} + c\sigma_{i-1}^2. \tag{5.9}$$

With (5.9) a continuous time version of the volatility process can be obtained, based on the approach of Klüppelberg et al. (2004).

$$\begin{aligned}
\sigma_i^2 &= a + b^*\varepsilon_{i-1}^2\sigma_{i-1}^2\mathbb{1}_{\{\varepsilon_{i-1}\geq 0\}} + d^*\varepsilon_{i-1}^2\sigma_{i-1}^2\mathbb{1}_{\{\varepsilon_{i-1}<0\}} + c\sigma_{i-1}^2 \\
&= a + \left(c + \left(b^*\mathbb{1}_{\{\varepsilon_{i-1}\geq 0\}} + d^*\mathbb{1}_{\{\varepsilon_{i-1}<0\}}\right)\varepsilon_{i-1}^2\right)\sigma_{i-1}^2 \\
&= a + \left(c + \left(b^*\mathbb{1}_{\{\varepsilon_{i-1}\geq 0\}} + d^*\mathbb{1}_{\{\varepsilon_{i-1}<0\}}\right)\varepsilon_{i-1}^2\right) \\
&\qquad \times \left(a + \left(c + \left(b^*\mathbb{1}_{\{\varepsilon_{i-2}\geq 0\}} + d^*\mathbb{1}_{\{\varepsilon_{i-2}<0\}}\right)\varepsilon_{i-2}^2\right)\right)\sigma_{i-2}^2 \\
&= a + a\left[\left(c + \left(b^*\mathbb{1}_{\{\varepsilon_{i-1}\geq 0\}} + d^*\mathbb{1}_{\{\varepsilon_{i-1}<0\}}\right)\varepsilon_{i-1}^2\right) + \left(c + \left(b^*\mathbb{1}_{\{\varepsilon_{i-1}\geq 0\}} + \right.\right.\right. \\
&\qquad \left.\left.\left. + d^*\mathbb{1}_{\{\varepsilon_{i-1}<0\}}\right)\varepsilon_{i-1}^2\right)\left(c + \left(b^*\mathbb{1}_{\{\varepsilon_{i-2}\geq 0\}} + d^*\mathbb{1}_{\{\varepsilon_{i-2}<0\}}\right)\varepsilon_{i-2}^2\right)\right]\sigma_{i-2}^2 \\
&= \dots
\end{aligned}$$

$$= a \sum_{j=0}^{i-1} \prod_{k=j+1}^{i-1} \left(c + \left(b^* \mathbb{1}_{\{\varepsilon_k \geq 0\}} + d^* \mathbb{1}_{\{\varepsilon_k < 0\}} \right) \varepsilon_k^2 \right) \tag{5.10}$$

$$+ \sigma_0^2 \prod_{k=0}^{i-1} \left(c + \left(b^* \mathbb{1}_{\{\varepsilon_k \geq 0\}} + d^* \mathbb{1}_{\{\varepsilon_k < 0\}} \right) \varepsilon_k^2 \right)$$

$$= a \int_0^i \exp \left(\sum_{k=\lfloor s \rfloor + 1}^{i-1} \log \left(c + \left(b^* \mathbb{1}_{\{\varepsilon_k \geq 0\}} + d^* \mathbb{1}_{\{\varepsilon_k < 0\}} \right) \varepsilon_k^2 \right) \right) ds$$

$$+ \sigma_0^2 \exp \left(\sum_{k=0}^{i-1} \log \left(c + \left(b^* \mathbb{1}_{\{\varepsilon_k \geq 0\}} + d^* \mathbb{1}_{\{\varepsilon_k < 0\}} \right) \varepsilon_k^2 \right) \right) \tag{5.11}$$

$$= a \int_0^i \exp \left(\sum_{k=\lfloor s \rfloor + 1}^{i-1} \left(\log c + \log \left(1 + \frac{b^* \mathbb{1}_{\{\varepsilon_k \geq 0\}} + d^* \mathbb{1}_{\{\varepsilon < 0\}}}{c} \varepsilon_k^2 \right) \right) \right) ds$$

$$+ \sigma_0^2 \exp \left(\sum_{k=0}^{i-1} \left(\log c + \log \left(1 + \frac{b^* \mathbb{1}_{\{\varepsilon_k \geq 0\}} + d^* \mathbb{1}_{\{\varepsilon < 0\}}}{c} \varepsilon_k^2 \right) \right) \right)$$

$$= a \int_0^i \exp \left((i - \lfloor s \rfloor) \log c + \sum_{k=\lfloor s \rfloor + 1}^{i-1} \left(\log \left(1 + \frac{b^* \mathbb{1}_{\{\varepsilon_k \geq 0\}} + d^* \mathbb{1}_{\{\varepsilon < 0\}}}{c} \varepsilon_k^2 \right) \right) \right) ds$$

$$+ \sigma_0^2 \exp \left(\sum_{k=0}^{i-1} \left(\log c + \log \left(1 + \frac{b^* \mathbb{1}_{\{\varepsilon_k \geq 0\}} + d^* \mathbb{1}_{\{\varepsilon < 0\}}}{c} \varepsilon_k^2 \right) \right) \right) \tag{5.12}$$

This is well-defined as in (5.11) it holds that $c + \left(b^* \mathbb{1}_{\{\varepsilon_k \geq 0\}} + d^* \mathbb{1}_{\{\varepsilon_k < 0\}} \right) \varepsilon_k^2 > 0$. The process $X = (X(t))_{t \geq 0}$ is defined by

$$X_t = -t \log c - \sum_{0 \leq s \leq t} \log \left(1 + \frac{\left(b^* \mathbb{1}_{\{\Delta L(s) \geq 0\}} + d^* \mathbb{1}_{\{\Delta L(s) < 0\}} \right)}{c} (\Delta L_s)^2 \right).$$

Choosing the parameters $c = e^{-\eta}$, $b = \varphi e^{-\eta}$, $\varphi > 0$ and $d = \gamma \in (0,1)$ it follows that $b^* = b(1-d)^2 = \varphi e^{-\eta}(1-\gamma)^2$ and $d^* = b(1+d)^2 = \varphi e^{-\eta}(1+\gamma)^2$. We can write

$$X_t = \eta t - \sum_{0 \leq s \leq t} \log \left(1 + \left[(1-\gamma)^2 \mathbb{1}_{\{\Delta L_s \geq 0\}} + (1+\gamma)^2 \mathbb{1}_{\{\Delta L_s < 0\}} \right] \varphi (\Delta L_s)^2 \right). \tag{5.13}$$

With (5.11) and $a = \beta$ the volatility process $\sigma^2 = \left(\sigma_t^2 \right)_{t \geq 0}$ can now be defined as

$$\sigma_t^2 = \left(\theta \int_0^t e^{X_s} ds + \sigma_0^2 \right) e^{-X_t}, \quad t \geq 0, \tag{5.14}$$

The integrated continuous time GJR GARCH process $G = (G_t)_{t \geq 0}$ is defined by

$$G_t = \int_0^t \sigma_{s-} dL_s, \quad t \geq 0. \tag{5.15}$$

We will use the continuous time GJR GARCH$(1,1)$ process for the simulations and estimations in the following.

It is possible to obtain a stochastic differential equation for the continuous time APARCH process.

Theorem 5.3 *The process* $\left(\sigma_t^\delta\right)_{t\geq 0}$ *satisfies the stochastic differential equation*

$$d\sigma_{t+}^\delta = \theta dt + \sigma_t^\delta e^{X_{t-}} d\left(e^{-X_t}\right), \quad t > 0, \tag{5.16}$$

and we have

$$\sigma_t^\delta = \sigma_0^\delta + \theta t + \log\beta \int_0^t \sigma_s^\delta ds + \frac{\alpha}{\beta} \sum_{0<s\leq t} \sigma_s^\delta h(\Delta L_s), \quad t \geq 0. \tag{5.17}$$

For $\delta = 2$ we obtain the continuous time GJR GARCH$(1,1)$ process. Equation (5.17) can then be written as

$$\sigma_t^2 = \sigma_0^2 + \theta t + \log\beta \int_0^t \sigma_s^2 ds + \frac{\alpha}{\beta} \sum_{0<s\leq t} \sigma_s^2 h(\Delta L_s), \tag{5.18}$$

with $h(x) = (|x| - \gamma x)^2$. For $h(\Delta L_s)$ we obtain

$$\begin{aligned}
h(\Delta L_s) &= (|\Delta L_s| - \gamma \Delta L_s)^2 \\
&= \Delta L_s^2 - 2\gamma |\Delta L_s| \Delta L_s + \gamma^2 \Delta L_s^2 \\
&= (1+\gamma^2)\Delta L_s^2 - \gamma |\Delta L_s| \Delta L_s \\
&= (1+\gamma^2)\Delta L_s^2 - 2\gamma \Delta L_s^2 \mathbb{1}_{\{\Delta L_s \geq 0\}} + 2\gamma \Delta L_s^2 \mathbb{1}_{\{\Delta L_s < 0\}} \\
&= (1+\gamma^2-2\gamma)\Delta L_s^2 \mathbb{1}_{\{\Delta L_s \geq 0\}} + (1+\gamma^2+2\gamma)\Delta L_s^2 \mathbb{1}_{\{\Delta L_s < 0\}} \\
&= (1-\gamma)^2 \Delta L_s^2 \mathbb{1}_{\{\Delta L_s \geq 0\}} + (1+\gamma)^2 \Delta L_s^2 \mathbb{1}_{\{\Delta L_s < 0\}}
\end{aligned} \tag{5.19}$$

By inserting (5.19) in (5.18) and reparameterize (5.18) with $\theta = \beta$, $\log(\beta) = -\eta$ and $\alpha/\beta = \varphi$ we obatin

$$\begin{aligned}
\sigma_t^2 = \sigma_0^2 + \beta t - \eta \int_0^t \sigma_s^2 ds \\
+ \sum_{0<s\leq t} \underbrace{\varphi \left[(1-\gamma)^2 \mathbb{1}_{\{\Delta L_s \geq 0\}} + (1+\gamma)^2 \mathbb{1}_{\{\Delta L_s < 0\}}\right]}_{=:\varphi^*} \sigma_s^2 \Delta L_s^2
\end{aligned} \tag{5.20}$$

Futhermore, we give some properties of the continuous time APARCH model. Based on Klüppelberg et al. (2004, Theorem 3.1), Mayr (2013) stated a stability condition for the volatility process $\left(\sigma_t^\delta\right)_{t\geq 0}$ of the continuous time APARCH process and defines σ_∞^δ.

Theorem 5.4 *(Mayr (2013, Theorem 4.6))*
Suppose

$$\int_{\mathbb{R}} \log\left(1 + \frac{\alpha}{\beta} h(y)\right) \nu_L(dy) < -\log\beta. \tag{5.21}$$

Then $\sigma_t^\delta \xrightarrow{D} \sigma_\infty^\delta$ *as* $t \to \infty$*, for a finite random variable* σ_∞^δ *satisfying*

$$\sigma_\infty^\delta \stackrel{D}{=} \theta \int_0^\infty e^{-X_t} dt. \tag{5.22}$$

Conversely, if (5.21) *does not hold, then* $\sigma_t^\delta \xrightarrow{P} \infty$ *as* $t \to \infty$.

Next, based on (Klüppelberg et al., 2004, Theorem 3.2) and Lee (2012) the following Theorem is proved in Mayr (2013, Theorem 4.7).

Theorem 5.5 *The processes* $\left(\sigma_t^\delta\right)_{t\geq 0}$ *and* $\left(\sigma_t^\delta, G_t\right)_{t\geq 0}$ *are time homogeneous Markov processes. If* $\sigma_0^\delta \stackrel{D}{=} \sigma_\infty^\delta$ *and* σ_0^δ *are independent of* $(L_t)_{t\geq 0}$*, then* σ_t^δ *is strictly stationary and* $(G_t)_{t\geq 0}$ *is a process with stationary increments.*

From (5.7) it follows that the moments of the volatility process $(\sigma_t^\delta)_{t\geq 0}$ correspond to certain exponential moments of $(X_t)_{t\geq 0}$. Lemma 5.6 specifies this relationship.

Lemma 5.6 *(Mayr (2013, Lemma 4.8))*
Keep $c > 0$ *throughout.*

(a) Let $\alpha > 0$*. Then the Laplace transform* $\mathbb{E}\left(e^{-cX_t}\right)$ *of* X_t *at* c *is finite for some* $t > 0$*, or, equivalently, for all* $t > 0$*, if and only if* $\mathbb{E}\left(|L_1|^{\delta c}\right) < \infty$.

(b) When $\mathbb{E}\left(e^{-cX_1}\right) < \infty$*, define* $\Psi(c)$ *as* $\Psi(c) = \Psi_X(c) = \log \mathbb{E}\left(e^{-cX_1}\right)$*. Then* $|\Psi(c)| < \infty$*,* $\mathbb{E}\left(e^{-cX_t}\right) = e^{t\Psi(c)}$*, and*

$$\Psi(c) = c\log(\beta) + \int_{\mathbb{R}} \left(\left(1 + \frac{\alpha}{\beta} h(y)\right)^c - 1\right) \nu_L(dy), \tag{5.23}$$

where $\Psi(c)$ *is the Laplace exponent of the Laplace transform* $\mathbb{E}\left(e^{-cX_t}\right) = e^{t\Psi(c)}$ *of the auxiliary process* X*.*

(c) If $\mathbb{E}\left(|L_1|^\delta\right) < \infty$ *and* $\Psi(1) < 0$*, then*

$$\int_{\mathbb{R}} \log\left(1 + \frac{\alpha}{\beta} h(y)\right) \nu_L(dy) < -\log\beta, \tag{5.24}$$

holds, and $\sigma_t^\delta \xrightarrow{D} \sigma_\infty^\delta$ *if* $t \to \infty$*, for a finite random variable* σ_∞^δ*, with* $\sigma_\infty^\delta \stackrel{D}{=} \theta \int_0^\infty e^{-X_t} dt$.

(d) If $\Psi(c) < 0$ *for some* $c > 0$*, then* $\Psi(d) < 0$ *for all* $0 < d < c$.

Furthermore, it is possible to state the first two moments and the autocovariance function of the volatility process $\left(\sigma_t^\delta\right)_{t\geq 0}$ in terms of Ψ.

Proposition 5.7 *Let* $\alpha > 0$*,* $t > 0$*,* $h \geq 0$.

(i) $\mathbb{E}\left(\sigma_t^\delta\right) < \infty$ *if and only if* $\mathbb{E}\left(|L_1|^\delta\right) < \infty$ *and* $\mathbb{E}\left(\sigma_0^\delta\right) < \infty$. *If this is the case, then*

$$\mathbb{E}\left(\sigma_t^\delta\right) = \frac{\theta}{-\Psi(1)} + \left(\mathbb{E}\left(\sigma_0^\delta\right) + \frac{\theta}{\Psi(1)}\right) e^{t\Psi(1)}, \tag{5.25}$$

where for $\Psi(1) = 0$ *the righthand side has to be interpreted as its limit as* $\Psi(1) \to 0$, *i.e.* $\mathbb{E}\left(\sigma_t^\delta\right) = \theta t + \mathbb{E}\left(\sigma_0^\delta\right)$.

(ii) $\mathbb{E}\left(\sigma_t^{2\delta}\right) < \infty$ *if and only if* $\mathbb{E}\left(|L_1|^{2\delta}\right) < \infty$ *and* $\mathbb{E}\left(\sigma_0^{2\delta}\right) < \infty$. *In that case, the following formulae hold*

$$\begin{aligned}\mathbb{E}\left(\sigma_t^{2\delta}\right) = {} & \frac{2\theta^2}{\Psi(1)\Psi(2)} + \frac{2\theta^2}{\Psi(2)-\Psi(1)}\left(\frac{e^{t\Psi(2)}}{\Psi(2)} - \frac{e^{t\Psi(1)}}{\Psi(1)}\right) \\ & + 2\theta\mathbb{E}\left(\sigma_0^\delta\right)\left(\frac{e^{t\Psi(2)} - e^{t\Psi(1)}}{\Psi(2)-\Psi(1)}\right) + \mathbb{E}\left(\sigma_0^{2\delta}\right) e^{t\Psi(2)}, \end{aligned} \tag{5.26}$$

$$\mathbb{C}ov\left(\sigma_t^\delta, \sigma_{t+h}^\delta\right) = \mathbb{V}ar\left(\sigma_t^\delta\right) e^{h\Psi(1)}. \tag{5.27}$$

In the following the moments of the continuous time APARCH model should be considered. It is not known how to calculate the moments for every $\delta > 0$. Therefore, only models with $\delta = 2$ will be considered in the following. The integrated APARCH process was defined to satisfy $dG_t = \sigma_t dL_t$ for $t > 0$, i.e G jumps at the same time L does and has jumps of size $\Delta G_t = \sigma_t \Delta L_t$. This definition implies that for any fixed timepoint t all moments of ΔG_t are zero. But it makes sense to calculate moments for the increments of G in arbitrary time intervals. The increments of G are denoted by $G_t^{(r)}$ and for $r > 0$

$$G_t^{(r)} := G_{t+r} - G_t = \int_{t+}^{t+r} \sigma_s ds, \quad t \geq 0.$$

In Theorem 5.8 the moments of the increments $G_t^{(r)}$ are calculated.

Theorem 5.8 *(Mayr (2013, Theorem 4.14))*
Suppose $(L_t)_{t\geq 0}$ *is a quadratic pure jump process (i.e.* $\sigma_L^2 = 0$*), with* $\mathbb{E}\left(L_1^2\right) < \infty$ *and* $\mathbb{E}\left(L_1\right) = 0$. *For the Laplace exponent from* (5.23) *it holds that* $\Psi(1) < 0$. *Let* $\left(\sigma_t^2\right)_{t\geq 0}$ *be the stationary volatility process from* (5.17) *with* $\sigma_\infty^2 \stackrel{D}{=} \sigma_0^2$. *Then for any* $t \geq 0$ *and* $h \geq r > 0$,

$$\mathbb{E}\left(G_t^{(r)}\right) = 0, \tag{5.28}$$

$$\mathbb{E}\left(\left(G_t^{(r)}\right)^2\right) = \frac{\theta r}{-\Psi(1)}\mathbb{E}\left(L_1^2\right), \tag{5.29}$$

$$\mathbb{C}ov\left(G_t^{(r)}, G_{t+h}^{(r)}\right) = 0. \tag{5.30}$$

Assume further that $\mathbb{E}\left(L_1^4\right) < \infty$ *and* $\Psi(2) < 0$. *Then*

$$\mathbb{C}ov\left(\left(G_t^{(r)}\right)^2, \left(G_{t+h}^{(r)}\right)^2\right) = \left(\frac{e^{-r\Psi(1)} - 1}{-\Psi(1)}\right)\mathbb{E}\left(L_1^2\right)\mathbb{C}ov\left(G_r^2, \sigma_r^2\right) e^{h\Psi(1)}. \tag{5.31}$$

Where

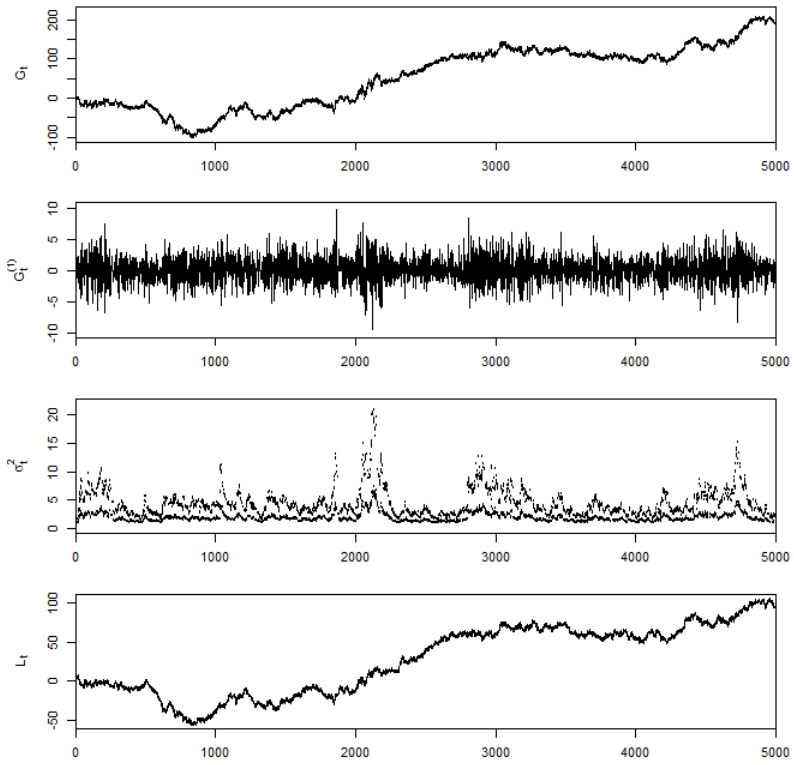
$$\mathbb{C}ov\left(G_t^2, \sigma_t^2\right) = \left(\frac{\beta}{\alpha}\right) \text{var}\left(\sigma_0^2\right) \left(1 - e^{t\Psi(1)} - \log\beta\left(\frac{1 - e^{t\Psi(1)}}{-\Psi(1)}\right)\right). \tag{5.32}$$

For the COGARCH$(1,1)$ *process and the GJR GARCH*$(1,1)$ *process the righthand side of* (5.32) *is strictly positive, if* $\alpha > 0$, $\mathbb{E}\left(|L_1|^8\right) < \infty$, $\Psi(4) < 0$, $\int_{[-1,1]} |x| \nu_L(dx) < 0$ *and* $\int_{\mathbb{R}} x^3 \nu_L(dx) = 0$.

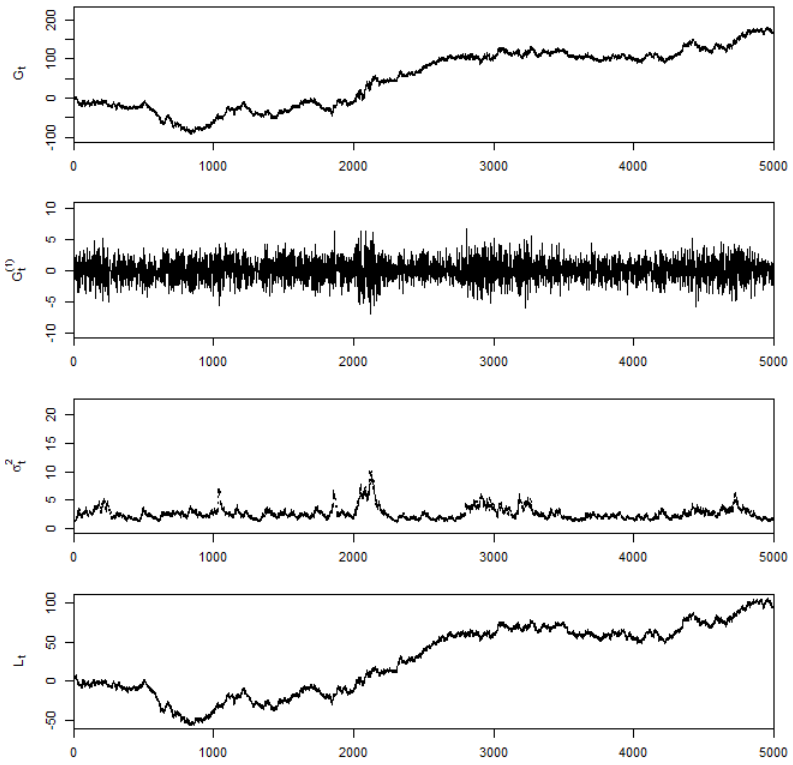
In the first plot of Figure 5.1(a), a simulated sample path of a continuous time GJR GARCH$(1,1)$ process, $(G_t)_{t>0}$, driven by a compound Poisson process with $\lambda = 1$ and standard normally distributed jumps is shown. The parameters were chosen as $\beta = 0.04$, $\eta = 0.053$, $\varphi = 0.038$ and $\gamma = 0.3$. Furthermore, the returns $(G_t^{(1)})_{t>0}$ of this process are displayed in the second plot from the top. In the third plot from the top, the volatility process $(\sigma_t^2)_{t>0}$ can be observed. Finally, the sample paths of a (compound Poisson) Lévy process are plotted in the figure on the bottom. As a comparison we will simulate a symmetric COGARCH$(1,1)$ process driven by the same Lévy process as the continuous time GJR GARCH$(1,1)$ process in Figure 5.1(a).The corresponding sample paths can be seen in Figure 5.1(b). By the choice of the parameter γ the asymmetry of the process is influenced. If a large value (between 0 and 1) is chosen for γ, the negative jumps of the driving Lévy process are weighted more then positive jumps of the same size. For $\gamma = 0$ positive and negative jumps are weighted equally and we then get symmetric COGARCH model. In Figures 5.1(a) and 5.1(b) the influence of γ on the asymmetry of the process can be observed.

The plot of the volatility process in Figure 5.1(a) seems to contain two paths. Therefore, we have a look at a smaller range of time in Figure 5.2. In the plot on the top we can observe the volatility process of the symmetric COGARCH$(1,1)$ process (blue line) of our example before. The volatility process of the continuous time GJR GARCH$(1,1)$ process is displayed by the black dots. In the bottom plot the jumpsizes of the underlying Lévy process are shown. For a positive jump we see that at the next point in time the volatility process of the asymmetric model is below the volatility process of the symmetric model. And for a negative jump it lies above the volatility process of the symmetric model.

In order to illustrate the effect of different chosen values for the parameter γ, we simulated the continuous time GJR GARCH$(1,1)$ model for $\gamma = \{0, 0.2, 0.3, 0.4\}$. Figure 5.3(a) shows the sample paths of the continuous time GJR GARCH$(1,1)$ process $(G_t)_{t\geq 0}$ for different γ. We can observe the different weighting according to positive or negative jumps in the plots. In Figure 5.3(b) the returns $(G_t^{(1)})_{t\geq 0}$ are displayed. There we can observe bigger volatility clustering, the larger we choose the value for γ. The volatility processes for $\gamma = \{0, 0.2, 0.3, 0.4\}$ can be found in Figure 5.4. It is obvious that positive and negative jumps are weighted differently. A negative jump has more impact on the volatility process than a positive one. This leads to the effect that in the plot it seem to be two paths again. See Figure 5.2 for an explanation.



(a) Continuous time (asymmetric) GJR GARCH$(1,1)$ process driven by a compound Poisson process with $\lambda = 1$ and standard normally distributed jumps. The parameters were chosen as $\beta = 0.04$, $\eta = 0.053$, $\varphi = 0.038$ and $\gamma = 0.3$.



(b) Symmetric COGARCH$(1,1)$ process driven by a compound Poisson process with $\lambda = 1$ and standard normally distributed jumps. The parameters were chosen as $\beta = 0.04$, $\eta = 0.053$, $\varphi = 0.038$ and $\gamma = 0.3$.

Figure 5.1: Comparison of an asymmetric *(left)* and a symmetric *(right)* COGARCH$(1,1)$ process.

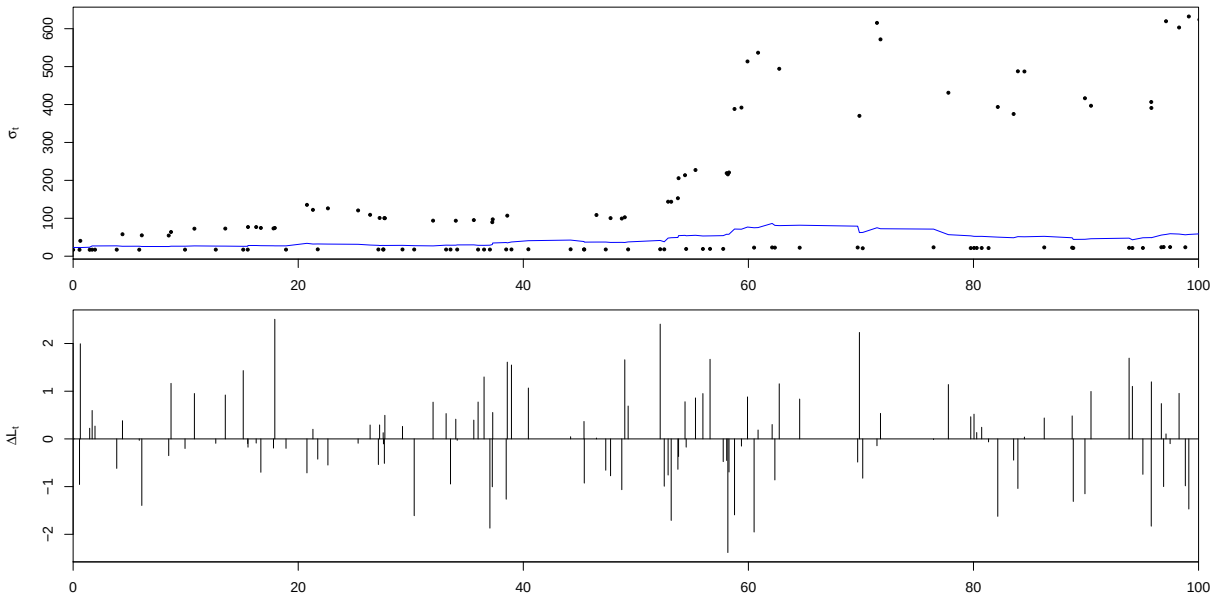


Figure 5.2: Comparison of the volatility process σ_t^2 of a symmetric COGARCH$(1,1)$ process driven by a compound Poisson process with standard normally distributed jumps and $\lambda = 1$ *(blue)* and the volatility process of a continuous time GJR GARCH$(1,1)$ process *(black dots)* with parameters $\beta = 1$, $\eta = 0.06$, $\varphi = 0.0425$ and $\gamma = 0.8$ **(top)**. And the jumpsizes ΔL_t **(bottom)**.

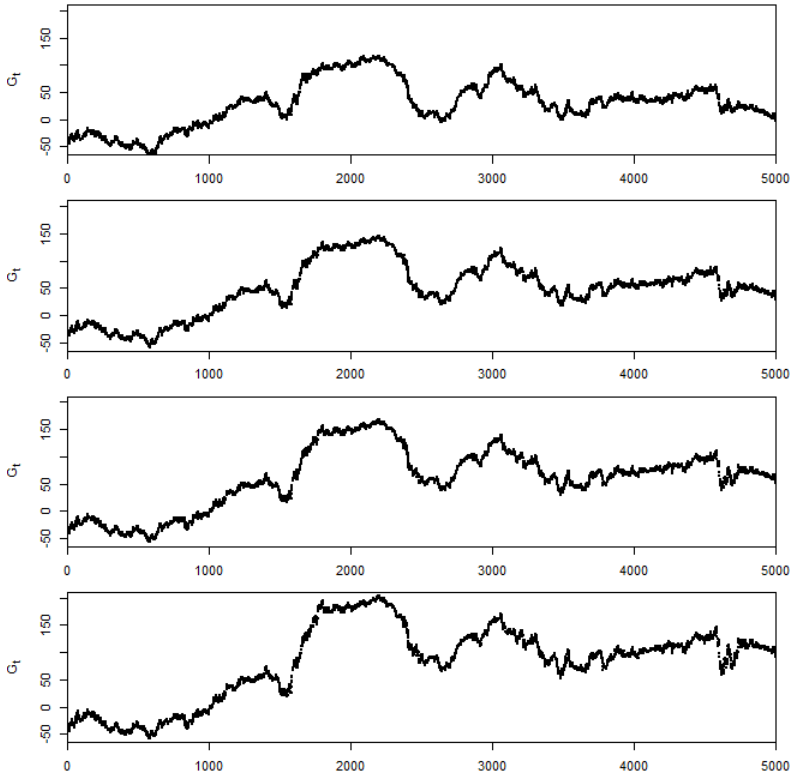
5.2 Estimation method for the GJR COGARCH

In this chapter, we first approximate the continuous time GJR GARCH$(1,1)$ process by a discrete time model. Then we use this approximation to get a discretization of the continuous time volatility process. Finally we state an estimation method which can be apply to unequally space asymmetric data. The finite sample properties of this estimator are analysed in a small simulation study.

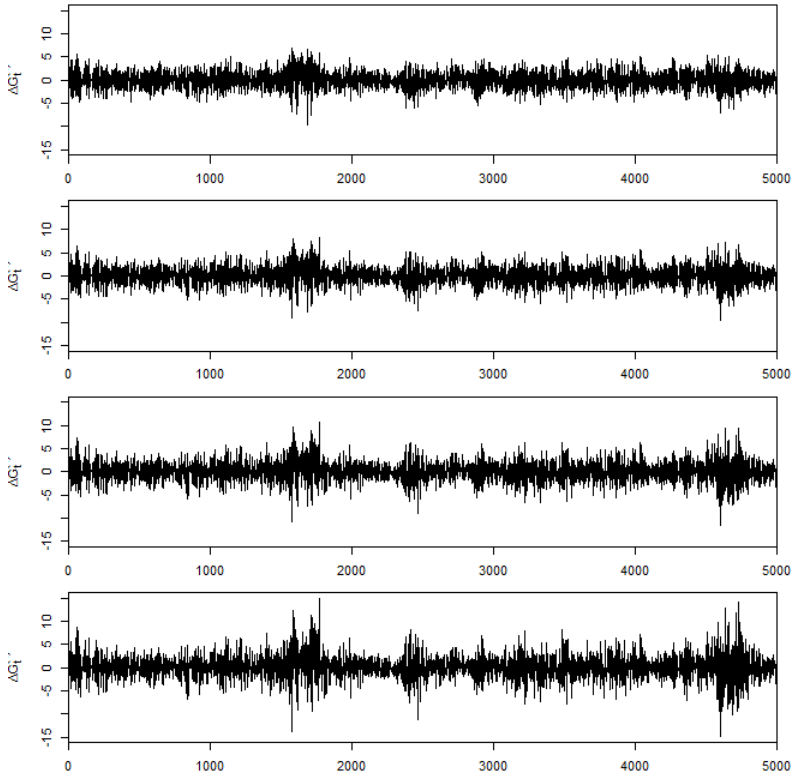
5.2.1 Approximating the continuous time GJR GARCH process

Mayr (2013) showed in Chapters 6 and 7 that the approach introduced in the paper of Maller et al. (2008) can be used for the continuous time GJR GARCH$(1,1)$ model. This chapter is based on these chapters. The aim of this chapter is to define - based on a continuous time GJR GARCH$(1,1)$ process G - discrete time GJR GARCH$(1,1)$ processes $G_n = (G_n(t))_{t\geq 0}$, $n = 1, 2, \ldots$, which approximate the continuous time GJR GARCH$(1,1)$ process G. After appropriate rescaling to match the discrete and continuous time parameters, it will be shown that the discrete time process $(G_n)_n$ converges in probability to the continuous time process G in the Skorokhod metric.

We can then continue to approximate the continuous time GJR GARCH$(1,1)$ process, as described before for the COGARCH$(1,1)$ process in Chapter 3.2.2 based on Maller et al. (2008, Section 2.1).
Fix $T > 0$, and take deterministic sequences $(N_n)_{n\geq 1}$ with $\lim_{n\to\infty} N_n = \infty$ and $0 =$



(a) Continuous time GJR GARCH$(1,1)$ process driven by a compound Poisson process with $\lambda = 1$ and standard normally distributed jumps. The parameters were chosen as $\beta = 0.04$, $\eta = 0.053$, $\varphi = 0.038$ and $\gamma = \{0, 0.2, 0.3, 0.4\}$ *((top)-(bottom))*.



(b) Returns of a continuous time GJR GARCH$(1,1)$ process driven by a compound Poisson process with $\lambda = 1$ and standard normally distributed jumps. The parameters were chosen as $\beta = 0.04$, $\eta = 0.053$, $\varphi = 0.038$ and $\gamma = \{0, 0.2, 0.3, 0.4\}$ *((top)-(bottom))*.

Figure 5.3: Simulation of a continuous time GJR GARCH$(1,1)$ process driven by a compound Poisson process for different values of γ. On the *left* the process G_t and on the *right* the returns $G_t^{(1)} = G_{t-1} - G_t$ of this process are shown.

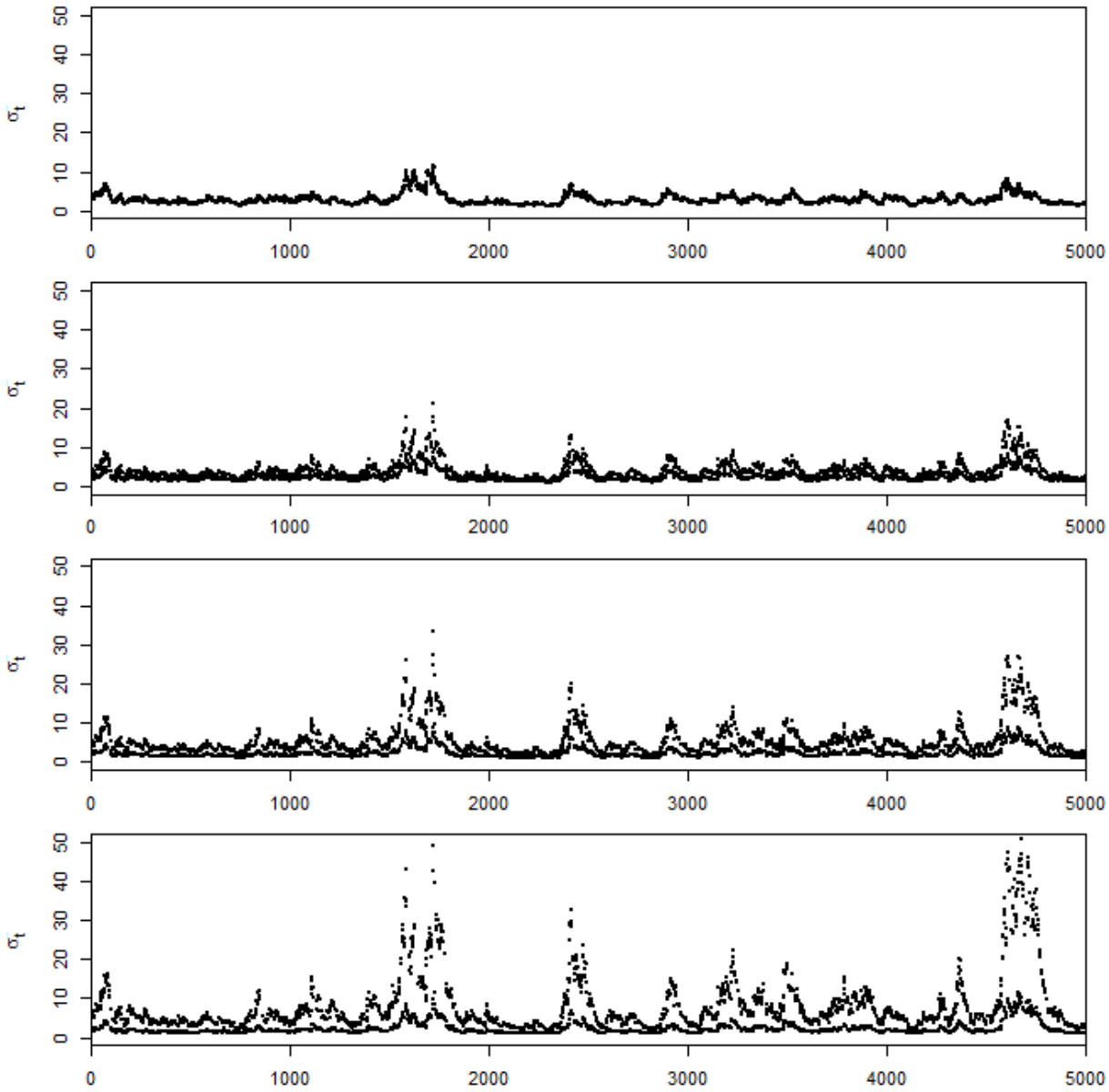


Figure 5.4: Volatility processes of a continuous time GJR GARCH$(1,1)$ process driven by a compound Poisson process with $\lambda = 1$ and standard normally distributed jumps. The parameters were chosen as $\beta = 0.04$, $\eta = 0.053$, $\varphi = 0.038$ and $\gamma = \{0, 0.2, 0.3, 0.4\}$ *((top)-(bottom))*.

$t_0(n) < t_1(n) < \cdots < t_{N_n}(n) = T$, and, for each $n = 1, 2, \ldots$, divide $[0, T]$ into N_n subintervals of length $\Delta t_i(n) := t_i(n) - t_{i-1}(n)$ for $i = 1, 2, \ldots, N_n$. Assume $\Delta t(n) := \max_{i=1,\ldots,N_n} \Delta t_i(n) \to 0$ as $n \to \infty$ and define for each $n = 1, 2, \ldots$, a discrete-time process $(G_{i,n})_{i=1,\ldots,N_n}$ satisfying

$$G_{i,n} = G_{i-1,n} + \sigma_{i-1,n}\sqrt{\Delta t_i(n)}\epsilon_{i,n}, \quad i = 1, 2, \ldots, N_n, \tag{5.33}$$

where $G_{0,n} = G_0 = 0$. The volatility $\sigma^2_{i,n}$ can be described through the following recursion

$$\begin{aligned}\sigma^2_{i,n} =& \theta\Delta t_i(n) + e^{-\eta\Delta t_i(n)}\sigma^2_{i-1,n} \\ &\times \left(1 + [(1-\gamma)^2 \mathbb{1}_{\{\epsilon_{i-1,n}\geq 0\}} + (1+\gamma)^2 \mathbb{1}_{\{\epsilon_{i-1,n}<0\}}]\varphi\Delta t_i \epsilon^2_{i-1,n}\right).\end{aligned} \tag{5.34}$$

Equations (5.33) and (5.34) describe a GJR GARCH$(1,1)$ similar recursion. If we choose for all time subintervals the same length, i.e. $\Delta t_i(n) = \Delta t(n), i = 1, 2, \ldots, N_n$, then, after rescaling with $\Delta t(n)$ and appropriate reparametrization, (5.34) is equivalent to (5.9). For the reparametrization of the parameters from $(\theta, \eta, \varphi, \gamma)$ to (a, b^*, c, d^*) we choose $a = \theta$, $b^* = b(1-d)^2 = \varphi e^{-\eta}(1-\gamma)^2$, $c = e^{-\eta}$ and $d^* = b(1+d)^2 = \varphi e^{-\eta}(1+\gamma)^2$ as before.

The innovations $(\epsilon_{i,n})_{i=1,\ldots,N_n}$, $i \in \mathbb{N}$ are constructed using a "first jump" approximation of the Lévy process, as introduced for the COGARCH case in Chapter 3.2.2. In the stopping time defined in (3.59) only the value of the jump size $|\Delta L_t|$ is considered. For the continuous time GJR GARCH$(1,1)$ model it is of impact if the driving Lévy process jumps up or down. Due to the asymmetry of the model there is a difference in weighting positive and negative jumps. Therefore we have to observe additionally if the jump is positive or negative.

Analogously to the COGARCH case, the discrete time processes $G_{\cdot,n}$ and $\sigma^2_{\cdot,n}$ are embedded into the continuous time versions G_n and σ^2_n defined by

$$G_n(t) := G_{i,n} \quad \text{and} \quad \sigma^2_n(t) := \sigma^2_{i,n}, \quad \text{when } t \in [t_{i-1}(n), t_i(n)),\ 0 \leq t \leq T, \tag{5.35}$$

with $G_n(0) = 0$. In Theorem 5.9 the convergence of the discrete time processes to the continuous time processes is specified.

Theorem 5.9 *(Mayr (2013, Theorem 6.6))*
Let the processes (G, σ^2) be defined as in (5.7) *and* (5.8)*. Moreover $(G_n, \sigma^2_n)_{n\geq 1}$ defined as in* (5.35)*. Then the Skorokhod distance between the processes and the discretized, piecewise constant processes converges in probability to* 0 *as $n \to \infty$, i.e*

$$\rho((G_n, \sigma^2_n), (G, \sigma^2)) \xrightarrow{\mathbb{P}} 0 \quad \text{as } n \to \infty. \tag{5.36}$$

Consequently, we also have convergence in distribution in $\mathbb{D}[0,T] \times \mathbb{D}[0,T]$.

5.2.2 Pseudo-Maximum Likelihood Method

In Section 3 of Maller et al. (2008), they suggested a method of fitting a continuous time GARCH process to unequally spaced time data. We introduced this method (PML method) in Chapter 3.2.2. It is possible to use this approach with modified equations (3.64), (3.74) and (3.76) for a continuous time GJR GARCH$(1,1)$ model as we will show in the following.[28]

Suppose having observations G_{t_i} at the timepoints $0 = t_0 < t_1 < \cdots < t_N = T$ on the integrated continuous time GJR GARCH. Furthermore, it is assumed that the integrated continuous time GJR GARCH process is in its stationary regime and the timepoints $\{t_i\}$ are fixed (non-random). Let $Y_i = G_{t_i} - G_{t_{i-1}}$ denote the observed returns and let $\Delta t_i = t_i - t_{i-1}$. Then

$$Y_i = \int_{t_{i-1}}^{t_i} \sigma_{s-} dL_s,$$

where L is a Lévy process with $\mathbb{E}(L_1) = 0$ and $\mathbb{E}(L_1^2) = 1$ assumed.

We would like to use the PML method to estimate the parameters $(\beta, \eta, \varphi, \gamma)$ from the observed $Y_1, Y_2, \ldots, Y_N$. The return Y_i is conditionally independent from $Y_{i-1}, Y_{i-2}, \ldots$, given $\mathcal{F}_{t_{i-1}}$, as $(\sigma_t^2)_{t\geq 0}$ is a Markov process, cf. Theorem 5.5. Like before it holds for the conditional expectation $\mathbb{E}(Y_t|\mathcal{F}_{t_{i-1}}) = 0$ and for the conditional variance we have, as in (3.70)

$$\rho_i^2 = \mathbb{E}(Y_i^2|\mathcal{F}_{t_{i-1}}) = \mathbb{E}(L_1^2)\Delta t_i \frac{\beta}{-\Psi(1)} + \mathbb{E}(L_1^2)\left(\frac{e^{\Delta t_i \Psi(1)} - 1}{\Psi(1)}\right)\left(\sigma_{t_{i-1}}^2 - \frac{\beta}{-\Psi(1)}\right). \tag{5.37}$$

Consequently, we only have to calculate $\Psi(1)$ for the continuous time GJR GARCH process with $h(x) = (|x| - \gamma x)^2$. With Lemma 5.6 and the reparametrization $\log(\beta) = -\eta$ and $\frac{\alpha}{\beta} = \varphi$, the Laplace exponent for the continuous time GJR GARCH model can be calculated by

$$\Psi(c) = -\eta c + \int_{\mathbb{R}} ((1 + \varphi(|y| - \gamma y)^2)^c - 1)\nu_L(dy).$$

In order to calculate $\Psi(1)$ for the continuous time GJR GARCH model we restrict ourselves to symmetric jump distributions for the driving Lévy process. Assuming $\int_{\mathbb{R}} y^2 \nu_L(dy) = \mathbb{E}(L_1^2) = 1$ and a symmetric jump distribution we get

$$\begin{aligned}
\Psi(1) &= -\eta + \int_{\mathbb{R}} ((1 + \varphi(|y| - \gamma y)^2) - 1)\nu_L(dy) \\
&= -\eta + \varphi \int_{\mathbb{R}} (|y| - \gamma y)^2 \nu_L(dy) \\
&= -\eta + \varphi \int_{\mathbb{R}} (|y|^2 - 2\gamma |y| y + \gamma^2 y^2)\nu_L(dy) \\
&= -\eta + \varphi \left((1 + \gamma^2) \int_{\mathbb{R}} y^2 \nu_L(dy) - 2\gamma \int_{\mathbb{R}} |y| y \nu_L(dy) \right)
\end{aligned}$$

[28] This has been developed based on the theory of Mayr (2013).

$$= -\eta + \varphi(1+\gamma^2). \tag{5.38}$$

Inserting (5.38) in (5.37) we get

$$\begin{aligned}\rho_i^2 &= \mathbb{E}(Y_i^2|\mathcal{F}_{t_{i-1}}) \\ &= \Delta t_i \frac{\beta}{-\Psi(1)} + \left(\frac{e^{\Delta t_i \Psi(1)}-1}{\Psi(1)}\right)\left(\sigma_{t_{i-1}}^2 - \frac{\beta}{-\Psi(1)}\right) \\ &= \Delta t_i \frac{\beta}{\eta-\varphi(1+\gamma^2)} + \left(\frac{e^{\Delta t_i(-\eta+\varphi(1+\gamma^2))}-1}{-\eta+\varphi(1+\gamma^2)}\right)\left(\sigma_{t_{i-1}}^2 - \frac{\beta}{\eta-\varphi(1+\gamma^2)}\right)\end{aligned} \tag{5.39}$$

Applying the PML method, we assume the Y_i are conditionally $\mathcal{N}(0,\rho_i^2)$. Analogously to (3.72) we obtain the following pseudo-log-likelihood function for $Y_1, Y_2, \ldots, Y_N$

$$\mathcal{L}_N = \mathcal{L}_N(\beta,\eta,\varphi) = -\frac{1}{2}\sum_{i=1}^{N}\left(\frac{Y_i^2}{\rho_i^2}\right) - \frac{1}{2}\sum_{i=1}^{N}\log(\rho_i^2) - \frac{N}{2}\log(2\pi), \tag{5.40}$$

with ρ_i^2 from (5.39). In order to implement this method, we will need a calculable quantity for ρ_i^2. Therefore we discretize the continuous time volatility process as it was done in Theorem 5.9. Thus, (5.34) is in the present notation, with $\beta = \theta$, of the following form,

$$\begin{aligned}\sigma_i^2 &= \beta\Delta t_i + \left(1 + [(1-\gamma)^2 \mathbb{1}_{\{\epsilon_{i-1}\geq 0\}} + (1+\gamma)^2 \mathbb{1}_{\{\epsilon_{i-1}<0\}}]\varphi\Delta t_i \epsilon_{i-1}^2\right) e^{-\eta\Delta t_i}\sigma_{i-1}^2 \\ &= \beta\Delta t_i + e^{-\eta\Delta t_i}\sigma_{i-1}^2 + [(1-\gamma)^2 \mathbb{1}_{\{\epsilon_{i-1}\geq 0\}} + (1+\gamma)^2 \mathbb{1}_{\{\epsilon_{i-1}<0\}}]\varphi e^{-\eta\Delta t_i}\Delta t_i \epsilon_{i-1}^2 \sigma_{i-1}^2 \\ &= \beta\Delta t_i + e^{-\eta\Delta t_i}\sigma_{i-1}^2 + [(1-\gamma)^2 \mathbb{1}_{\{\epsilon_{i-1}\geq 0\}} + (1+\gamma)^2 \mathbb{1}_{\{\epsilon_{i-1}<0\}}]\varphi e^{-\eta\Delta t_i} Y_i^2.\end{aligned} \tag{5.41}$$

5.2.3 Simulation Study

We apply the estimation approach introduced above to simulated samples of a continuous time GJR GARCH$(1,1)$ process. For $n = 5000$, we simulated 900 samples with input parameters $\beta = 0.04$, $\eta = 0.053$, $\varphi = 0.038$ and $\gamma = 0.3$. Then, we apply the moment estimation method in order to get starting values for β, η and φ. As it is not possible to apply the moment estimation method to estimate all the parameters for the GJR GARCH model, we chose $\gamma = 0.3$. We can use those parameters as starting values for the PML method introduced above and implemented in Chapter 6.4. In Table 5.1 the results are shown. The means of the estimators $\hat{\beta}$, $\hat{\eta}$ and $\hat{\varphi}$ are close to the input parameters and therefore the estimation method worked quite well. The mean of the estimators $\hat{\gamma}$ does not look so good. The input parameter has been $\gamma = 0.3$, but the mean of the estimated values $\hat{\gamma}$ is 0.16645 and it is not influenced by large outliers as the median is 0.15601. If we choose $gamma = 0.7$ for example, the estimated values for $\hat{\gamma}$ are equally small as in our example with $\gamma = 0.3$.

n=5000	$\hat{\beta}$	$\hat{\eta}$	$\hat{\varphi}$	$\hat{\gamma}$
Mean	0.04739	0.04457	0.02980	0.16645
Bias	0.00739	-0.00843	-0.00821	-0.13355
Rel. Bias	0.18472	-0.15908	-0.21595	-0.44517
MSE	0.00036	0.00017	0.00012	0.03112
MAE	0.01208	0.01102	0.00912	0.15356
Median	0.04258	0.04318	0.02939	0.15601
Bias Median	0.00258	-0.00982	-0.00861	-0.14399
MSE Median	$6 \cdot 10^{-5}$	0.00011	$7 \cdot 10^{-5}$	0.02222
MAE Median	0.00777	0.01035	0.00872	0.15356

Table 5.1: Estimated mean, bias, relative bias, MSE, MAE, median, bias of the median, MSE of the median and MAE of the median for $\hat{\beta}$, $\hat{\eta}$ and $\hat{\varphi}$ of 900 samples with the **PML Method** for in time **unequally spaced asymmetric** data. The true values are $\beta = 0.04$, $\eta = 0.053$, $\varphi = 0.038$ and $\gamma = 0.3$.

Chapter 6

Implementation in R

In Chapter 3 we introduced continuous time GARCH models and tried to estimate their parameters. Furthermore, we simulated sample paths of the ECOGARCH(p,q) process in Chapter 4. In Chapter 5 we had a look at the simulation and estimation of an asymmetric continuous time model: the continuous time GJR GARCH$(1,1)$ model. Based on the theory, we are going to present some useful implementations of these models using the program R (R Core Team (2012)). In the following, we state some functions written in R-code and explain the corresponding approach. All of these functions are summarized in the R-package *cogarch*. For the documentation of this package see Appendix A.2.

6.1 Simulation of a continuous time GARCH process

We start with the simulation of a COGARCH$(1,1)$ process. The simulation of such a process can be conducted with the function `cogarch_sim`. The COGARCH process can be based on different Lévy processes, so it is possible to choose either a compound Poisson process or a Variance Gamma Process. These processes can be simulated with the functions `compoundPoisson` or `vargamma`. Moreover, we simulated a COGARCH(p,q) process with a compound Poisson process as driving Lévy process in Chapter 3.1.4. For the simulation of a COGARCH process with $p,q \geq 1$ the function `cogarch_pq_sim` can be used, see Chapter 6.1.4.

6.1.1 Simulation of a compound Poisson process

We already introduced the compound Poisson process in Example 2.4. In Figure 2.3 a simulated sample path of a compound Poisson processes with $\lambda = 2$ and standard normally distributed jumpsizes is shown. This has been generated by the function `compoundPoisson`, which can be found in R-Code A.1. the following arguments should be specified as input: the considered points of a time grid are stated by the vector t. Additionally the *rate* which corresponds to the intensity λ and the *distribution* of the jumpsizes with its *mean* and *variance* can be chosen. For example, *distribution="normal"*, *mean*$= 0$ and *var*$= 1$ (i.e. $\mathcal{N}(\mu = 0, \sigma = 1)$) can be specified as input parameters. Ad-

ditionally the *Bernoulli* and the *Uniform* distribution with different mean and variance are implemented in this function. The function starts with a simulation of the random jumptimes t_i, $i = 1, \ldots, N$. From these random jumptimes we get the random time intervals, $\Delta t = t_i - t_{i-1}$, which are i.i.d. exponentially distributed random variables with the selected *rate*. As an output we get a matrix including the "jumptimes", the "values of ΔL_t" and the "values of L_t".
R-Code 6.1 shows how to use this function in order to simulate and then plot a compound Poisson process.

```
#input parameters
t<-0:10
rate<-1
distribution<-"normal" #jumpsizes are normally distributed
mean<-0
var<-1

#simulation
set.seed(123) #choose any random seed
output<-compoundPoisson(0:10,rate,distribution,mean,var)
jumptimes<-output[,1]
no<-length(jumptimes)
max<-ceiling(jumptimes[no])+1

#plot jumpsizes delta_Lt and Lt
plot(stepfun(c(jumptimes),c(0,output[,2]),right=FALSE),verticals=FALSE,pch=20,xlab="t",
     ylab=expression(Delta*L[t]),xlim=c(0,max),xaxs="i")#plot jumpsizes delta_Lt
plot(stepfun(c(jumptimes),c(0,output[,3]),right=FALSE),verticals=FALSE,pch=20,xlab="t",
     ylab=expression(L[t]),xlim=c(0,max),xaxs="i") #plot Lt
```

R-Code 6.1: Example compound Poisson Process

6.1.2 Simulation of a Variance Gamma process

A further example of a Lévy process is the Variance Gamma (VG) process. We already introduced this kind of process in Example 2.5.
Madan et al. (1998) state that a Variance Gamma process can be obtained by evaluating a Brownian motion with drift at a random time given by a Gamma process. The approach we use for simulating such a process is taken from Schoutens (2003). The idea is to sample a standard Brownian Motion and a Gamma process in order to get a sample path of a VG process. First we simulate the sample path of a Gamma process $(H_t)_{t\geq 0}$, such that each H_t follows a $\Gamma(at, b)$ law, with density

$$f_{H_t}(x) = \frac{b^{at}}{\Gamma(at)} x^{at-1} e^{-bx}, \tag{6.1}$$

where $\Gamma(\cdot)$ denotes the Gamma function. The process is simulated over a time grid $\{n\Delta t, n = 0, 1, \ldots\}$. By generating $\Gamma(a\Delta t, b)$ distributed random variables g_n, $n \geq 0$, we can approximate the Gamma process by

$$H_0 := 0, \qquad H_{n\Delta t} = H_{(n-1)\Delta t} + g_n. \tag{6.2}$$

Finally we can get the VG process $(V_t)_{t\geq 0}$ with parameters $\sigma > 0$, $\tau > 0$ and $\theta \in \mathbb{R}$ by

$$V_t := \theta H_t + \sigma B_{G_t}, \tag{6.3}$$

where $(B_t)_{t\geq 0}$ is a standard Brownian motion with drift given by a Gamma process $(H_t)_{t\geq 0}$ with parameters $a = 1/\tau$ and $b = 1/\tau$.

An example of a simulated VG process with $\sigma = 0.3$, $\theta = -0.03$, $\tau = 0.5$ and grid size $\Delta t = 0.01$ on the time interval $[0, 10]$ is shown in Figure 2.4.

In order to simulate this kind of process the function `vargamma`[29] of R-Code A.2 can be used. Our example can be generated by R-Code 6.2 which uses this function.

```
sigma <- 0.3
theta <- -0.03
nu <- 0.5
t <- seq(0,1,0.1)
gs <- 0.00001

set.seed(123)
output <- vargamma(sigma,nu,theta,t,gs)
plot(output[,1],output[,2],type="p",cex=0.01,xlab="t",ylab=expression(V[t]),xlim=c(0,10),xaxs="i")
```

R-Code 6.2: Example Variance Gamma Process

6.1.3 Simulation of a COGARCH(1,1) process

As mentioned before the Lévy process used for the simulation of a COGARCH$(1,1)$ process can be selected. It can be either a compound Poisson process or a Variance Gamma process. Consequently, it is possible to perform this simulation (see R-Code 6.3) based on R-Code A.1 or A.2. To understand the approach of the COGARCH$(1,1)$ simulation we should have a look at the concept of the Euler scheme, also called the Euler-Maruyama scheme.

Euler scheme

The Euler scheme[30] is an extension of the Euler method for solving ordinary differential equations to solve stochastic differential equations (SDEs) numerically. Its idea is the following. Let the Itô process $\{X_t, 0 \leq t \leq T\}$ be a given solution of the stochastic differential equation

$$dX_t = b(t, X_t)dt + \sigma(t, X_t)dW_t,$$

with initial deterministic value $X_{t_0} = X_0$ and the discretization $\Pi_N = \Pi_N([0, T])$ of the interval $[0, T]$, $0 = t_0 < t_1 \cdots < t_N = T$. We set $Y_i := Y_{t_i}$ and $W_i := W_{t_i}$ for a Wiener process. The **Euler scheme** of X is a continuous stochastic process Y satisfying the iterative scheme

$$Y_{i+1} = Y_i + b(t_i, Y_i)(\Delta t_i) + \sigma(t_i, Y_i)\Delta W_i, \tag{6.4}$$

for $i = 0, 1, \ldots, N-1$, with $Y_0 = X_0$. By Δt_i we denote the time increments $t_{i+1} - t_i$. Analogously, we have the $\mathcal{N}(0, \Delta t_i)$-distributed increments of the Wiener process W on the time interval $[t_i, t_{i+1}]$, denoted by $\Delta W_i = W_{i+1} - W_i$. In order to simulate the process Y, it can be seen from Equation (6.4) that a simulation of the increments of the

[29] Pay attention to the fact that in the function `vargamma` the parameter τ is denoted by the input parameter *nu*.

[30] see Kloeden and Platen (2011, Chapter 10.2), Iacus (2008, Chapter 2) and Protter and Talay (1997) for details and examples.

Wiener process is sufficent. The Euler scheme is strongly convergent of order $\gamma = 0.5$ and weakly convergent of order $\beta = 1$. [31]

In Brockwell et al. (2006), Theorem 2.2, it has been demonstrated that if $p = q = 1$, the solution of the COGARCH(p, q) SDEs coincides with the solution of the COGARCH$(1, 1)$ SDEs of Klüppelberg et al. (2004).[32] Thus, we can use this result, adjusted with our parameters, to apply the Euler scheme.
For the case $p = q = 1$ we find in the proof of Theorem 2.2 in Brockwell et al. (2006) the following equations:

$$dY_t = -\eta Y_{t-} dt + V_t d[L, L]_t^{(d)} \quad \text{and} \quad V_t = \frac{\beta}{\eta} + \varphi Y_{t-}, \tag{6.5}$$

and it follows that

$$dV_t = \varphi dY_t = -\varphi\eta \frac{V_t - \beta/\eta}{\varphi} dt + \varphi V_t d[L, L]_t^{(d)}.$$

Hence we have

$$V_t = \frac{\beta}{\eta}\eta t - \eta \int_0^t V_s ds + \varphi \sum_{0<s\leq t} V_s \Delta L_s^2 + V_0. \tag{6.6}$$

Equation (6.6) is also satisfied for the volatility process $(\sigma_t^2)_{t\geq 0}$ of Equation (3.6), as we have

$$\begin{aligned} \sigma_t^2 &= \frac{\beta}{\eta}\eta t - \eta \int_0^t \sigma_s^2 ds + \varphi \sum_{0<s\leq t} \sigma_s^2 \Delta L_s^2 + \sigma_0^2 \\ &= \beta t - \eta \int_0^t \sigma_s^2 ds + \varphi \sum_{0<s\leq t} \sigma_s^2 \Delta L_s^2 + \sigma_0^2. \end{aligned} \tag{6.7}$$

Compare Equation (6.7) with Equation (3.8) of Proposition 3.2 in Klüppelberg et al. (2004), where $\eta = -\log(\delta)$ and $\varphi = \frac{\lambda}{\delta}$. Now we are ready to apply the Euler scheme to Equation (6.5). We have $b(t, Y_t) = -\eta Y_{t-}$ and $\sigma(t, Y_t) = V_t$ with $V_t = \frac{\beta}{\eta} + \varphi Y_{t-}$. With Equation (6.4) we can conclude

$$Y_{i+1} = Y_i - \eta Y_{i-} \Delta t_i + V_i \Delta L_i^2 \quad \text{and} \quad V_i = \frac{\beta}{\eta} + \varphi Y_{i-}, \tag{6.8}$$

with $V_0 = \beta/\eta$ and $Y_0 = 0$. In the corresponding R-Code 6.3, we first simulate the driving Lévy process for a small interval of time. As mentioned before, it is possible

[31] Cf. Iacus (2008, p.61 and 62): A time-discretized approximation Y_δ of a continuous time process Y, with δ the maximum time increment of the discretization, is said to be of general **strong** order of convergence γ to Y if for any fixed time horizon T it holds true that $\mathbb{E}|Y_\delta(T) - Y(T)| \leq C\delta^\gamma$, $\forall \delta < \delta_0$, with $\delta_0 > 0$ and C constant not depending on δ. Moreover, Y_δ is said to converge **weakly** of order β to Y if for any fixed time horizon T and any $2(\beta + 1)$ continuous differentiable function g of polynomial growth, it holds true that $|\mathbb{E}g(Y(T)) - \mathbb{E}g(Y_\delta(T))| \leq C\delta^\beta$, $\forall \delta < \delta_0$, with $\delta_0 > 0$ and C a constant not depending on δ.

[32] Also compare Chapter 3.1 in this master's thesis.

to choose either a compound Poisson process (via Lp="cp") or a Variance Gamma process (via Lp="vg"). In this step the random jumptimes of the chosen Lévy process are also simulated. As a next step we initialize the Y_{i+1} by `Yi`, the volatility V_i by `voli` and ΔG_t by `g`. Afterwards, we conduct a simulation of the driving Lévy process and its random jumptimes for the whole time interval chosen by the input parameter `t`. Then we calculate `vol`, where `vol[1]<-voli`, `Y`, where `Y[1]<-Yi` and `delta_G`, where `delta_G[1]<-g`. Finally we obtain the desired COGARCH(1,1) process G_t by using the cumulated sum of the ΔG_t.

```
cogarch_sim<-function(t=0:10,beta=1,eta=0.05,phi=0.03,Lp="cp",rate=1,distribution="normal",mean=0,var=1,sigma=1,nu=0.5,theta
    =1,gs=0.01)
{# Simulation of a COGARCH(1,1) process driven by a compound Poisson or a Variance Gamma Lévy process
  #Input: t - fixed time grid
  #       beta,eta,phi - Input parameters for the cogarch process, have to be positive
  #       Lp - driving Lévy process: either "cp"=compound Poisson or ""vg"= Variance Gamma
  #       rate - intensity of the compound Poisson Levy process
  #       distribution - jump size distribution, e.g "normal"
  #       mean,var - mean and variance of specified jump distribution
  #       sigma, nu, theta - parameters for Variance Gamma process (for E(L_1)=0 and E(L_1^2)=1 choose theta=0 and sigma=1)
  #       gs - time grid size
  #Output: G - COGARCH(1,1) process
  #        vol - volatility process
  #        Lt - driving Lévy process
  #        delta_Lt - jumps of Lévy process
  #        randomjumptimes - random jumptimes

  #use the Euler-Maruyama scheme in order to get a numerical solution for a SDE
  #See the Book: Peter E. Kloeden: "Numerical solution of stochastic differential equations"
  t<-sort(t)
  n_t<-length(t)

  #find starting values
  if(Lp=="cp"){
    index<-1
  }else if(Lp=="vg"){
    index<-2
  }else{
    stop("No valid Levy process specified! Select 'cp' or 'vg'.")
  }
  switch(index,
{output_cp<-compoundPoisson(0:10,rate,distribution,mean,var) #calls the function compoundPoisson with output c(randomjumptimes
    ,randomjumpsizes,Lt)
 rt<-output_cp[,1] #randomjumptimes
 #delta_rt<-rt[2:N]-rt[1:(N-1)] #randomjumpintervals, delta_ti
 delta_rt<-output_cp[,4]#randomjumpintervals, delta_ti
 delta_Lt<-output_cp[,2] #randomjumpsizes, delta_Lti
 Lt<-output_cp[,3] #L_t
},
{output_vg<-vargamma(0:10,sigma,nu,theta,gs) #calls the function varGamma with output timesequence and V
 Lt<-output_vg[,2]
 rt<-output_vg[,1]
 nv<-length(Lt)
 nt<-length(rt)

 #increments of the process
 delta_Lt<-Lt[2:nv]-Lt[1:(nv-1)]
 delta_rt<-rt[2:nt]-rt[1:(nt-1)]
}
  )
  n<-length(delta_Lt)

  delta_Lt_0<-c(0,delta_Lt[1:(n-1)]) #used for sigma_t-, i.e. the sigma_t without the jump at time t!

#start value volatility, for state process set Y=0 in the beginning and therefore for the observation process: V = beta/eta +
    phi*Y = beta/eta
  voli<-beta/eta

#state space process
  Yi<-0
  g<-0

  for (k in 1:n){
    Yi<-Yi-eta*Yi*delta_rt[k]+voli*delta_Lt_0[k]^2  # delta_Lt hat null im ersten Vektoreintrag, daher erhalte sigma_t-
    voli<-beta/eta+phi*Yi
    g<-g+sqrt(voli)*delta_Lt[k]
  }

#-----------------------------------------------
#obtain values via Euler approximation

switch(index,
{output_cp_new<-compoundPoisson(t,rate,distribution,mean,var) #calls function compoundPoisson

 rt_new<-output_cp_new[,1] #randomjumptimes
 delta_rt_new<-output_cp_new[,4]
```

```
 delta_Lt_new<-output_cp_new[,2] #randomjumpsizes, delta_Lti
 Lt_new<-output_cp_new[,3] #L_t
},
{output_vg_new<-vargamma(t,sigma,nu,theta,gs) #calls the function varGamma with output timesequence and V

 Lt_new<-output_vg_new[,2]
 rt_new<-output_vg_new[,1]
 nvn<-length(Lt_new)
 ntn<-length(rt_new)

 #increments of the process
 delta_Lt_new<-Lt_new[2:nvn]-Lt_new[1:(nvn-1)]
 delta_rt_new<-rt_new[2:ntn]-rt_new[1:(ntn-1)]
}
  )

  nn<-length(delta_Lt_new)
  #add again 0 as first entry of delta_Lt_new in order to get sigma_t- !
  delta_Lt_new_0<-c(0,delta_Lt_new[1:(nn-1)]) #length nn

 #volatility
  vol<-vector(length=(nn))
  vol[1]<-voli

 #state space
  Y<-vector(length=(nn))
  Y[1]<-Yi

  for (k in 1:nn){

    Y[k+1]<-Y[k]-eta*Y[k]*delta_rt_new[k]+vol[k]*delta_Lt_new_0[k]^2
    vol[k+1]<-beta/eta+phi*Y[k]
  }

  delta_G<-vector(length=length(vol)) #length(vol)=nn+1 mit nn=length(delta_Lt_new)
  delta_G[1]<-g
  s<-sqrt(vol[1:(length(vol)-1)]) #length=nn
  delta_G[2:length(vol)]<-s*delta_Lt_new[1:nn] #length=nn
  G<-cumsum(delta_G) #length=nn+1

 switch(index,
{output<-cbind(G,vol,c(0,Lt_new),c(0,delta_Lt_new),c(0,rt_new))
 colnames(output)<-c("G","vol","Lt","delta_Lt","randomjumptimes")
},
{output<-cbind(G,vol,Lt_new,c(0,delta_Lt_new),rt_new)
 colnames(output)<-c("G","vol","Lt","delta_Lt","randomjumptimes")
}
  )
  return(output)
}
```

R-Code 6.3: Simulation of a COGARCH$(1,1)$ process

As the function `cogarch_sim` needs several input parameters, we give an example for better understanding how to use this function. In R-Code 6.4 we provide an example of the simulation of a COGARCH$(1,1)$ process driven by a compound Poisson process with $\lambda = 1$ and standard normally distributed jumps. The parameters are $\beta = 0.04$, $\eta = 0.053$ and $\varphi = 0.038$. Moreover, this example can be used to reproduce the plots shown in Figure 3.1.

```
#input parameters
t<-0:100
beta<-0.04
eta<-0.053
phi<-0.038

Lp="cp" #compound Poisson process, Lp="cp"
rate<-1
distribution<-"normal" #jumpsizes are normally distributed
mean<-0
var<-1

#simulation
set.seed(100) #chosen any random number
output <- cogarch_sim(t,beta,eta,phi,Lp="cp",rate,distribution,mean,var) #compound Poisson

G<-output[,1]
vol<-output[,2]
Lt<-output[,3]
delta_Lt<-output[,4]
rt<-output[,5]

#plots
#par(mfrow=c(4,1))
plot(rt,Lt,type="l",xlim=c(0,10),xaxs="i",xlab="t",ylab=expression(L[t]))
```

```
plot(rt,G,type="l",xlim=c(0,10), xaxs="i",xlab="t",ylab=expression(G[t]))
plot(rt,vol,type="l",xlim=c(0,10), xaxs="i",xlab="t",ylab=expression(sigma[t]))
plot(rt,delta_Lt,type="l",xlim=c(0,10),xaxs="i",xlab="t",ylab=expression(paste(Delta,L[t],sep="")))
```

R-Code 6.4: Simulation of a COGARCH$(1,1)$ process with a compound Poisson Lévy process

In R-Code 6.5 a simulation of a COGARCH$(1,1)$ process driven by a Variance Gamma process with parameters $\sigma = 0.3$, $\theta = -0.03$ and $\tau = 0.3$ can be found. The gridsize is chosen to be 0.001.

```
#input parameters
t<-0:100
beta<-0.04
eta<-0.053
phi<-0.038

Lp="vg"   #variance gamma process, Lp="vg"
sigma<- 0.3
theta<- -0.03
nu<-0.3
gs<-0.001

#simulation
set.seed(100) #chosen any random number
output_c<-cogarch_sim(t,beta,eta,phi,Lp="vg",sigma,nu,theta,gs) #variance gamma

#plots analogously to the Example of the Simulation of a COGARCH(1,1) process with a compound Poisson Lévy process
```

R-Code 6.5: Simulation of a COGARCH$(1,1)$ process with a Variance Gamma Lévy process

6.1.4 Simulation of a COGARCH(p,q) process

For the simulation of a COGARCH(p,q) process, introduced in Chapter 3.1.4, we modified the function `cogarch_sim` for $p,q > 1$. First we tried to apply the Euler scheme analogously to Chapter 6.1.3. As this approach was not successful we now use the following equation to obtain Y_{i+1},

$$Y_{i+1} = e^{B\Delta t_i} Y_i + eV_i \Delta L_i^2.$$

And the volatility process is calculated by

$$V_{i+1} = \alpha_0 + AY_i.$$

In R-code 6.6 the function `cogarch_pq_sim` is shown.

```
cogarch_pq_sim<-function(t=0:10,a0=0.04,a=0.038,b=-0.053,Lp="cp",rate=1,distribution="normal",mean=0,var=1){
#Simulation of a COGARCH(p,q) process

#Input:  t - time grid
#        a0 - start value
#        a=[alpha_1,alpha_2,...,alpha_{p-1},alpha_p], where alpha_1=phi
#        b=[-beta_1,-beta_2,...,-beta_{q-1},-beta_q], where beta_1=eta and if q=1: b=[-beta_1]
#        Lp - driving Lévy process: "cp"=compound Poisson process
#        rate - intensity of the compound Poisson process
#        distribution - jump size distribution, e.g "normal"
#        mean,var - mean and variance of specified jump distribution
#Output: G - COGARCH(p,q) process
#        vol - volatility process
#        Lt - driving Lévy process
#        delta_Lt - jumps of Lévy process
#        randomjumptimes - random jumptimes
#        eig - eigenvalues of matrix B

  t<-sort(t)
  n_t<-length(t)
```

```
  #initialize matrix A and vectors b,e
  q=length(b)
  B=matrix(0,q,q)
  if(q>1){
    for (i in 1:(q-1)){
      B[i,i+1]<-1
    }}
  B[q,]<-b[q:1]
  eig<-eigen(B)$value
  reig<-Re(eig)
  for(i in 1:q){
  if (reig[i]>0){
    stop("Positive eigenvalues! Please specify different values for b!")
  }}

  e<-rep(0,length=q)
  e[q]<-1

  A<-rep(0,length=q)
  p<-length(a)
  A[1:p]<-a
  #-----------------------------------

  output_cp<-compoundPoisson(0:10,rate,distribution,mean,var) #calls the function compoundPoisson with output c(
      randomjumptimes,randomjumpsizes,Lt)
  rt<-output_cp[,1] #randomjumptimes
  delta_rt<-output_cp[,4]#randomjumpintervals, delta_ti
  delta_Lt<-output_cp[,2] #randomjumpsizes, delta_Lti
  Lt<-output_cp[,3] #L_t
  n<-length(delta_Lt)

  delta_Lt_0<-c(0,delta_Lt[1:(n-1)]) #used for sigma_t-, i.e. the sigma_t without the jump at time t!

  #---------------------------------------


  yi<-matrix(0,q,n)
  g<-rep(0,length=n)

  voli<-a0
  yi[,1]<-rep(1,length=q)

  for (k in 1:n){
    yi[,1]=as.numeric(expm(B*delta_rt[k])%*%yi[,1])+voli*delta_Lt_0[k]^2*e
    voli<-(a0+t(A)%*%yi[,1])
    g[1]<-g[1]+sqrt(voli)*delta_Lt[k]
    }

  #---------------------
  output_cp_new<-compoundPoisson(t,rate,distribution,mean,var) #calls function compoundPoisson

  rt_new<-output_cp_new[,1] #randomjumptimes
  #NN<-length(rt_new)
  #delta_rt_new<-rt_new[2:NN]-rt_new[1:(NN-1)] #randomjumpintervals, delta_ti
  delta_rt_new<-output_cp_new[,4]
  delta_Lt_new<-output_cp_new[,2] #randomjumpsizes, delta_Lti
  Lt_new<-output_cp_new[,3] #L_t

  nn<-length(delta_Lt_new)
  #add again 0 as first entry of delta_Lt_new in order to get sigma_t- !
  delta_Lt_new_0<-c(0,delta_Lt_new[1:(nn-1)]) #length nn

#volatility
  vol<-vector(length=(nn+1))
  vol[1]<-voli

  y<-matrix(0,q,(nn+1))
  y[,1]<-yi[,1]

  for (k in 1:nn){
    y[,k+1]=as.numeric(expm(B*delta_rt_new[k])%*%y[,k])+vol[k]*delta_Lt_new_0[k]^2*e
    vol[k+1]<-(a0+t(A)%*%y[,k])
  }

  delta_G<-vector(length=length(vol))  #length(vol)=nn+1 mit nn=length(delta_Lt_new)
  delta_G[1]<-g[1]
  s<-sqrt(vol[1:(length(vol)-1)]) #length=nn
  delta_G[2:length(vol)]<-s*delta_Lt_new[1:nn] #length=nn
  G<-cumsum(delta_G) #length=nn+1

  return(list(G,vol,Lt_new,delta_Lt_new,rt_new,eig))
}
```

R-Code 6.6: Simulation of a COGARCH(p,q) process

The Example in R-Code 6.7 describes the simulation of a COGARCH$(1,3)$ process.

```
t=0:500
a0=1
a=c(1)
```

```
b=c(-1.2,-0.48-pi^2,-0.064-0.4*pi^2)
Lp="cp"
rate=1
distribution="normal"
mean=0
var=1
output<-cogarch_pq_sim(t,a0,a,b,Lp,rate,distribution,mean,var)
G<-unlist(output[[1]])
vol<-unlist(output[[2]])
Lt<-unlist(output[[3]])
delta_Lt<-unlist(output[[4]])
rt<-unlist(output[[5]])
eig<-unlist(output[[6]])

#plots
par(mfrow=c(4,1))
plot(c(0,rt),G,cex=0.1,xaxs="i",xlim=c(0,500),xlab="t")
plot(rt,diff(G),type="h",xaxs="i",xlim=c(0,500),ylab=expression(G^{(1)}),xlab="t")
plot(c(0,rt),vol,cex=0.1,xaxs="i",xlim=c(0,500),ylab=expression(sigma^2),xlab="t")
plot(rt,Lt,cex=0.1,xaxs="i",xlim=c(0,500),xlab="t")
```

R-Code 6.7: Simulation of a COGARCH(p,q) process with a compound Poisson Lévy process

6.2 Estimation of the COGARCH$(1,1)$ process

In 3.2 we gave an overview on estimation methods which can be used for estimating the parameters of the COGARCH$(1,1)$ model. In this chapter, we will explain how this could be done with R functions.

6.2.1 Method of Moment Estimators

It is of great interest to estimate the parameters of the COGARCH model. For equally spaced data we can use the method of moments estimation. The background of this method has been given in Chapter 3.2.1.

To conduct an estimation with the method of moment estimators we need the input process to be equally spaced in time. The output of a simulation of the COGARCH$(1,1)$ process with the function `cogarch_sim` is not equally spaced in time. We get the random jumptimes and the values of the process G at these random jumptimes. In order to specify a time grid which is equally spaced and get the values of G at these equally spaced times, we introduce the function `prevTick` in R-Code 6.8. This function transforms the time series `X`, e.g. the process G, at the irregular time points `it` into a homogeneous time series y at the specified time points `t`. This is done with previous tick interpolation. That means in order to get the value for the timepoint t_i, $i = 1,\dots,N$, of the specified time grid, we take the value of the process `X` at the irregular time point `it` which is smaller than or equal to the time point t_i.

```
prevTick <-
function(t,it,X){
  #transforms the time series X, observed at the irregular time points it,
  #into a homogeneous time series y at regularly time points t
  #uses previous tick interpolation
  #-> i.e it takes the value of the maximal irregular time point<= regular time point t

  lt<-length(t)
  y<-rep(0,lt)
  lit<-length(it)

  for (i in 1:lt){
    m<-which(it<=t[i])
    max<-max(m)
    y[i]<-X[max]
  }
```

```
return(y)
}
```

R-Code 6.8: PrevTick function

In R-Code 6.9 we give an example for the usage of the function `prevTick`. We specified an equally spaced time grid in the interval $[0, 100]$. It is also possible to transform the process to any specified time grid, e.g. the time intervals between the t_i could be uniformly distributed (use the following command to generate these time points `t<-c(0,cumsum(sample(c(.5,1,1.5),100,replace=TRUE,prob=c(.3,.4,.3))))`).

```
#simulate compoundPoissonprocessCOGARCH
t<-0:100 #equally spaced time grid
beta<-0.04
eta<-0.053
phi<-0.038
Lp="cp" #compound Poisson process, Lp="cp"
rate<-1
distribution<-"normal" #jumpsizes are normally distributed
mean<-0
var<-1
output<-cogarch.sim(t,beta,eta,phi,Lp="cp",rate,distribution,mean,var)

G<-output[,1]
vol<-output[,2]
Lt<-output[,3]
delta.Lt<-output[,4]
rt<-output[,5]

#use function prevTick to get data on the specified time grid t
it<-rt #random jumptimes
X<-G
pt<-prevTick(t,it,X) output: process G at specified time grid t
```

R-Code 6.9: Simulation of a COGARCH$(1,1)$ process with a compound Poisson process and transformation of the process to an equally spaced time grid with the function prevTick.

In the following, we give the R-Code 6.10 which estimates the parameters β, η and φ of a COGARCH process according to Theorem 3.13 and the estimation algorithm given in the end of Chapter 3.2.1. The function `est_cogarch_ret` needs returns $(G_t^{(1)})_{t\geq 0}$ (equally spaced in time) as an input. The number of lags is chosen to be $d \approx \sqrt{n}$, where n is the number or returns, see page 11 in Haug et al. (2007). The number of lags d is taken into account by the robust linear regression we perfom to estimate the parameters of the model. Moreover, the resulting estimates are less sensitive to outliers in the data.[33]

```
est.cogarch.ret <-
function(r){
#Estimation of parameters with Moment Estimation Method

#r<-(G[2:n]-G[1:(n-1)])
#input: r =EQUALLY spaced Cogarch11 process returns with Lévy process: compound Poisson or variance gamma
#       #d=lags ,choose d=sqrt(n)
#output: betahat,etahat,phihat,p,k

n<-length(r)
d<-floor(sqrt(n)) #number of lags, see p.11 of paper 'Estimating the COGARCH(1,1) model-a first go'
s=1
#define output vectors
m1<-0
m2<-0
p<-0
k<-0
betahat<-0
etahat<-0
phihat<-0
y<-rep(0,(d+1)) #vector for acf

#calculate the squared returns
```

[33] cf. Haug et al. (2007, p.12).

```
r<-r^2

nr<-length(r)

#moments
mu<-mean(r)  #mean
gamma<-mean(r^2) #variance

#m1<-mu #first moments
#m2<-gamma #second moments

#calculate autocovariances and autocorrelations
muhat<-mu*rep(1,nr) #row vector
g0<-sum((r-muhat)*(r-muhat))/nr

for (h in 0:d){
  u<-nr-h
  y[h+1]<-sum((r[(1+h):nr]-mu*rep(1,u))*(r[1:u]-mu*rep(1,u)))
}
y<-y/(nr*g0) #empirical autocorrelation
acfr<-y[2:(d+1)]  #analogously to Matlab program

#regression least square method to obtain k and p
n_acfr_pos<-which(acfr>0) #rows with positive acfr entry
z<-log(acfr[n_acfr_pos])
X<-(-n_acfr_pos)
#par<-lm(z~X)  #model with intercept=log(k), X=p
require(MASS)
par<-rlm(z~X)#robust regression

par1<-as.numeric(par$coefficients[1])
par2<-as.numeric(par$coefficients[2])
k<-exp(par1)
p<-par2
#################
  if(p>0){
    betahat<-p*mu

    M1<-gamma-2*mu^2-6*((1-p-exp(-p))/((1-exp(p))*(1-exp(-p))))*k*gamma
    M2<-(2*k*gamma*p)/(M1*(exp(p)-1)*(1-exp(-p)))

      if(M2>0){
        phihat<-p*sqrt(1+M2)-p
        etahat<-p*sqrt(1+M2)
      }else{
        c<-rep(0,length(s))
        c<-1
         #stop("M2<=0!")
      }
  }else{
    cp<-rep(0,length(s))
    cp<-1
    #stop("p<=0!")
  }

#}#end of "for m=1:s"

est<-matrix(0,nrow=5,ncol=1)
est<-rbind(betahat,etahat,phihat,p,k) #estimated values
return(est)
}
```

R-Code 6.10: Estimation of the parameters of a COGARCH$(1,1)$ process

The function `est_cogarch_ret` can be used as demonstrated in R-Code 6.11. The first step is the simulation of a COGARCH$(1,1)$ process driven by a compound Poisson process with standard normally distributed jumps and $\lambda = 1$. We then obtain data on an equally spaced time grid with the function `prevTick` and calculate the returns of this data. Finally we estimate the parameters of the COGARCH$(1,1)$ process. The estimated parameters $\hat{\beta}$, $\hat{\eta}$ and $\hat{\varphi}$ should be close to the original input parameters $\beta = 0.04$, $\eta = 0.05$ and $\varphi = 0.03$. For examples and their estimation results see Chapter 3.3.

```
#parameters
beta<-0.04
eta<-0.05
phi<-0.03

t<-0:1000
Lp="cp" #compound Poisson process, Lp="cp"
rate<-1
distribution<-"normal" #jumpsizes are normally distributed
mean<-0
var<-1
#simulated cogarch process
```

```
output<-cogarch.sim(t,beta,eta,phi,Lp="cp",rate,distribution,mean,var) #compound Poisson
G<-output[,1]
vol<-output[,2]
Lt<-output[,3]
delta.Lt<-output[,4]
rt<-output[,5]

#get equally spaced data
it<-rt
X<-G
pt<-prevTick(t,it,X)
G<-pt
n<-length(G)
r<-(G[2:n]-G[1:(n-1)])
nr<-length(r)
r<-(G[2:length(G)]-G[1:(length(G)-1)])

#estimate parameters
estimates<-est.cogarch.ret(r)
betahat<-estimates[1,]
etahat<-estimates[2,]
phihat<-estimates[3,]
p<-estimates[4,]
k<-estimates[5,]

ev<-c(betahat,etahat,phihat) #estimated values
ov<-c(beta,eta,phi) #original values
```

R-Code 6.11: Estimation via the Moment Estimation Method.

Furthermore the package *cogarch* contains the estimation function `est_cogarch`, see R-Code A.5. Analogously to the function `est_cogarch_ret`, R-Code 6.10, this function estimates the parameters with the moment estimation method. The difference is, that this function needs a matrix containing $N > 1$ in time equally spaced COGARCH$(1,1)$ processes as an input. This matrix can be generated using the functions `Cogarch_cp_Npaths` for the simulation of N COGARCH$(1,1)$ processes driven by a compound Poisson process (see R-Code A.3) or `Cogarch_vg_Npaths` for N COGARCH$(1,1)$ processes driven by a Variance Gamma process (see R-Code A.4).

6.2.2 Pseudo-Maximum Likelihood Method

Another method for estimating the model parameters has been introduced in Chapter 3.2.2. The Pseudo-Maximum Likelihood (PML) method can be applied to data which is equally or unequally spaced in time. In Chapter 3.2.2 we introduced the estimation algorithm developed by Maller et al. (2008) and summarized in Klüppelberg et al. (2011, Section 5.4.1). We have to ensure that the assumptions (H1)-(H4) are fulfilled. Especially, we have to transform any simulated COGARCH process to fixed (non-random) time points via the function `prevTick` before we can apply the PML method. To use this method for random time points we introduced a modified PML method in Chapter 3.2.3. The modified PML method can be applied for observations of a COGARCH process driven by a compound Poisson process. The function `pml_cp`, see R-Code 6.14, is explained below.

"Normal" PML

Now we have a look at the "normal" PML function, see R-Code 6.12. The output of this function is the negative of the pseudo-log-likelihood function specified in (3.72). The input parameter of this function is a vector x containing some starting values β, η and φ for the maximization of the pseudo-log-likelihood function. The recursion in (3.76) takes $\sigma_0^2 = \beta/(\eta - \varphi)$ with $\eta > \varphi$ as its starting value to ensure stationarity. We

insert the values from σ_i^2 in (3.64) to obtain the values of ρ_i^2 and finally the pseudo-log-likelihood function (3.72).

```
pmk<-function(x){
  #Pseudo-Maximum Likelihood function
  #Input: x - vector with parameters beta, eta, phi (can be estimated by Moment Estimation Method first)
  #Output: LL - Pseudo-Maximum Likelihood function

  beta= x[1]
  eta=x[2]
  phi=x[3]

  #start value:  sigma0=sigma0^2
  sigma0<-beta/(eta-phi)
  sigma<-rep(0,length=Nt)
  rho<-rep(0,length=Nt)
  L=rep(0,Nt)

  rho[1]<-(beta*delta_t[1])/(eta-phi)
  sigma[1]<-beta*delta_t[1]+exp(-eta*delta_t[1])*(beta/(eta-phi))+phi*exp(-eta*delta_t[1])*r[1]^2

  for (i in 2:(Nt)){
    #sigma^2 here defined as sigma
    sigma[i]<-beta*delta_t[i]+exp(-eta*delta_t[i])*sigma[i-1]+phi*exp(-eta*delta_t[i])*r[i]^2
  }

  rho[101:(Nt)]<-(sigma[100:(Nt-1)]-beta/(eta-phi))*((exp(-(eta-phi)*delta_t[101:(Nt)])-1)/(phi-eta))+(beta*delta_t[101:(Nt)])
      /(eta-phi)

  L<-0.5*(r[101:(Nt)]^2/rho[101:(Nt)])+0.5*log(rho[101:(Nt)])

  LL = sum(L)

  LL<-LL+(Nt/2)*log(2*pi) #add last term of likelihood function

  return(LL)
}
```

R-Code 6.12: Estimation of the parameters of a continuous time GARCH(1,1) process with the PML Method

In R-Code 6.13 we illustrate the approach to get the PML estimates of (β, η, φ). The times are assumed to be irregularly spaced. To obtain the simulated COGARCH(1,1) process driven by a compound Poisson process at the specified time points we use the function `prevTick` again. The `pml` function needs Δt_i, $i = 1, \ldots, N$ and the returns $G_t^{(1)}$ as an input. Furthermore, we specify the constraints of our parameters via the matrix A and the vector d. These are β, η and $\varphi > 0$ and $\eta > \varphi$. The output of the `pml` function is the negative log-likelihood function and therefore we minimize this function with `constrOptim` to get the estimated values of β, η and φ.

```
t<-c(0,cumsum(sample(c(.5,1,1.5),1000,replace=TRUE,prob=c(.3,.4,.3))))
beta=1
eta=0.06
phi=0.0425

Lp="cp" #compound Poisson process, Lp="cp"
rate<-1
distribution<-"normal" #jumpsizes are normally distributed
mean<-0
var<-1
#simulated cogarch process
output<-cogarch_sim(t,beta,eta,phi,Lp,rate,distribution,mean,var,sigma,nu,theta,gs)
G<-output[,1]
rt<-output[,5]

#get data on the time grid specified by t
it<-rt
X<-G
pt<-prevTick(t,it,X)
G<-pt
n<-length(G)
r<-(G[2:n]-G[1:(n-1)]) #returns
nr<-length(r)

#PML method
par<-c(beta,eta,phi) #starting values for the optimization

delta_t<-t[2:length(t)]-t[1:(length(t)-1)]
Nt<-length(delta_t)
```

```
A<-matrix(c(1,0,0,0,1,0,0,0,1,0,1,-1),ncol=3,byrow=T)
d<-c(0.000001,0.01,0.01,0.01)
m<- constrOptim(par,pml, NULL , ui = A, ci = d)
estimates_pml<-m$par
betahat<-estimates_pml[1]
etahat<-estimates_pml[2]
phihat<-estimates_pml[3]

estimates<-c(betahat,etahat,phihat)
```

R-Code 6.13: Estimation via the PML Method.

"Modified" PML

In Chapter 3.2.3 we introduced a PML approach for simulated (random) time points. For the case of a compound Poisson process as driving Lévy process we calculated a "modified" conditional variance in (3.79). Furthermore, we showed how we can obtain $\hat{\lambda}$ due to the i.i.d exponentially distributed Δt_i, $i = 1, \ldots, n$ with rate λ. The function `pml_cp` includes the estimation of $\hat{\lambda}$ and the modified ρ_i^2, see R-Code 6.14.

```
pml_cp<-function(x){
#Pseudo-Maximum Likelihood function for the compound Poisson case
#Input: x - vector with parameters beta, eta, phi (can be estimated by Moment Estimation Method first)
#Output: LL - Pseudo-Maximum Likelihood function

#estimate lambda
#delta_t_i are the time changes, which are exponentially distributed with parameter lambda
  sum_delta_t<-sum(delta_t[1:length(delta_t)])
  lambdahat<-length(delta_t)/sum_delta_t

#pml
    beta= x[1]
    eta=x[2]
    phi=x[3]

#start value:  sigma0=sigma0^2
    sigma0<-beta/(eta-phi)
    sigma<-rep(0,length=Nt)
    rho<-rep(0,length=Nt)
    L=rep(0,Nt)

    rho[1]<-(beta*lambdahat)/(eta-phi)
    sigma[1]<-beta*delta_t[1]+exp(-eta*delta_t[1])*(beta/(eta-phi))+phi*exp(-eta*delta_t[1])*r[1]^2

    for (i in 2:(Nt)){
      #sigma^2 here defined as sigma
      sigma[i]<-beta*delta_t[i]+exp(-eta*delta_t[i])*sigma[i-1]+phi*exp(-eta*delta_t[i])*r[i]^2
    }

    rho[101:(Nt)]<-(sigma[100:(Nt-1)]-beta/(eta-phi))*(((lambdahat/(lambdahat+eta-phi))-1)/(phi-eta))+(beta*lambdahat)/(eta-
        phi)

    L<-0.5*(r[101:(Nt)]^2/rho[101:(Nt)])+0.5*log(rho[101:(Nt)])

    LL = sum(L)

    LL<-LL+(Nt/2)*log(2*pi) #add last term of likelihood function
    return(LL)
  }
```

R-Code 6.14: Estimation of the parameters of a continuous time GARCH$(1,1)$ process with the modified PML Method

We implemented the function `pml_cp` for a compound Poisson process as driving Lévy process for a COGARCH$(1,1)$ process. In R-Code 6.15 we give an example on how this function can be applied. We simulate a COGARCH$(1,1)$ process driven by a compound Poisson process with standard normally distributed jumps and rate $\lambda = 1$, like in R-Code 6.13. We then use the function `prevTick` to get the values of the COGARCH process at the specified grid. Then we calculate the MLE of λ to obtain $\hat{\lambda}$. This step is also implemented in the beginning of the function `pml_cp`. We inserted the additional

calculation of $\hat{\lambda}$ as the output of the estimation procedure is only a vector containing the estimated parameters $\hat{\beta}$, $\hat{\eta}$ and $\hat{\varphi}$.

```
t<-c(0,cumsum(sample(c(.5,1,1.5),1000,replace=TRUE,prob=c(.3,.4,.3))))
beta=1
eta=0.06
phi=0.0425

Lp="cp" #compound Poisson process, Lp="cp"
rate<-1
distribution<-"normal" #jumpsizes are normally distributed
mean<-0
var<-1
#simulated cogarch process
output<-cogarch_sim(t,beta,eta,phi,Lp,rate,distribution,mean,var,sigma,nu,theta,gs)
G<-output[,1]
rt<-output[,5]

#get data on the time grid specified by t
it<-rt
X<-G
pt<-prevTick(t,it,X)
G<-pt
n<-length(G)
r<-(G[2:n]-G[1:(n-1)]) #returns
nr<-length(r)

#PML method
par<-c(beta,eta,phi) #starting values for the optimization

delta_t<-t[2:length(t)]-t[1:(length(t)-1)]
Nt<-length(delta_t)

#calculate lambdahat
sum_delta_t<-sum(delta_t[1:length(delta_t)])
lambdahat<-length(delta_t)/sum_delta_t

A<-matrix(c(1,0,0,0,1,0,0,0,1,0,1,-1),ncol=3,byrow=T)
d<-c(0.000001,0.01,0.01,0.01)
m<- constrOptim(par,pml_cp, NULL , ui = A, ci = d)
estimates_pml<-m$par
betahat<-estimates_pml[1]
etahat<-estimates_pml[2]
phihat<-estimates_pml[3]

estimates<-c(betahat,etahat,phihat,lambdahat)
```

R-Code 6.15: Estimation via the modified PML Method.

6.3 Simulation of an ECOGARCH(p, q) process

For the simulation of an ECOGARCH process we again use an Euler approximation (see 6.1.3) for the approximation of the state space process X_t. The ECOGARCH(p, q) process is defined as in Chapter 4. The simulation of a sample path of the log-price process G and the log-volatility process $\log(\sigma^2)$ over a time interval $[0, T]$ is done in the following steps.

(1) Choose observation times $0 = t_0 < t_1 < \cdots < t_n \leq T$, possibly random.

(2) Simulate the jump times (T_k), $k = 1, \ldots, n_T$, with $n_T := \max\{k \in \mathbb{N} : T_k \leq T\}$, of the compound Poisson process[34] J.

(3) Approximate the state process (4.7) of the log-volatility by a stochastic Euler scheme.

[34]Here we only consider the case where the ECOGARCH process is driven by a compound Poisson process.

(4) Compute an approximation $\hat{G}$ via the recursion

$$\hat{G}_{t_i} = \hat{G}_{t_{i-1}} + \sigma_{t_{i-1}-}\tilde{W}_i + \sum_{k=N_{t_{i-1}}+1}^{N_{t_i}} \sqrt{\exp\{\mu + \boldsymbol{b}^\top \hat{\boldsymbol{X}}_{T_k-}\}} Z_k, \tag{6.9}$$

where $\tilde{W}_i \sim \mathcal{N}(0, t_i - t_{i-1})$ and $\hat{\boldsymbol{X}}_{T_k-}$ is the Euler approximation without the jump ΔM_{T_k}.

This steps can be found in the implementation of the function `ecogarch_sim` in R-Code 6.16.

```
ecogarch_sim<-function(t=0:10,a=-0.1,b=1,mu=-1,theta=-0.2,gamma=0.1,rate=1,mean=0,var=1){
  #Simulation Ecogarch(p,q) process driven by a compound Poisson process

  #Input: t - fixed time grid
  #       a,b - alpha and beta values corresponding to p and q, respectively
  #             a has to be a negative vector, i.e a=c(-a_q,-a_{q-1},...,-a_1)
  #       mu,theta,gamma - parameters for M
  #       rate - intensity of the compound Poisson Levy process
  #       mean,var - mean and variance of specified jump distribution
  #Output: List including: G - logprice process on a specified grid (prevTick applied)
  #                        GexpTime -logprice process on times including random jump times
  #                        logsigmaexpTime - log(sigma) process on random jumptimes
  #                        Ylimleft - log(sigma) process on times including random jump times without jump
  #                        Y - log(sigma) process on times including random jump times with jump
  #                        Tylimleft - ordered times including random jump times
  #                        Brownian - Brownian motion
  #                        Mt - Lévy process M_t
  #                        Lt - compound Poisson Lévy process with specified rate, normally distributed jumps with mean and
      var as specified
  #                        TM - first column: times including random jump times, second column: M_t
  #                        sigmaJump - volatility process, i.e exp(Y)


  #initialize matrix A and vectors b,e
  q=length(a)
  A=matrix(0,q,q)
  if(q>1){
    for (i in 1:(q-1)){
      A[i,i+1]<-1
    }}
  A[q,]<-a[q:1]

  #test if the eigenvalues are negative
  eig<-eigen(A)$values
  reig<-Re(eig)
  for(i in 1:q){
    if (reig[i]>0){
      stop("Positive eigenvalues! Please specify different values for a!")
    }}

  e<-rep(0,length=q)
  e[q]<-1

  B<-rep(0,length=q)
  p<-length(b)
  B[1:p]<-b
  #----------------------------------------

  #compound Poisson
  #generate randomjumptimes---------------------------------
  t<-sort(t) #sort vector t
  nt<-length(t)  #number of timepoints
  I<-t[2:nt]-t[1:(nt-1)]  #timeintervals
  NI<-length(I) #vector with lengths of timeintervals

  #heuristic for number of jump times needed to cover t
  nmax <- ceiling(t[nt] *2*rate)
  maxtime <- t[nt] #maximal entry of time, i.e. last timepoint
  randomjumpintervals <- rexp(nmax, rate = rate)
  randomjumptimes<-cumsum(randomjumpintervals)

  validtimes=TRUE
  k=1

  while(validtimes)
  {if(randomjumptimes[length(randomjumptimes)]>maxtime){
    if(randomjumptimes[k]< maxtime){
      randomjumptimes[k]<-randomjumptimes[k]
      k<-k+1
    }else{
      randomjumptimes<-randomjumptimes[1:k]
```

```
    n<-length(randomjumptimes)
    validtimes=F
  }
}else{
  #if not enough intervals to cover all t, extend the vector with exponentially distributed rvs
  randomjumpintervals2<-rexp(nmax, rate=rate)
  randomjumptimes<-c(randomjumptimes,cumsum(randomjumpintervals2))
}
}
#generate jumps----------
std<-sqrt(var)
Levy<- rnorm(n,mean,std) #normally distributed with mean and sigma=sqrt(var)

#generate equidistant times
tequi<-seq(0,maxtime,by=0.1)
#add randomjumptimes to vector
tall<-c(tequi,randomjumptimes)
#no jumps at tequi
zerojump<-rep(0,length=length(tequi))
#add zerojump and M to Mall
Lall<-c(zerojump,Levy)
#generate matrix with columns time and M, then sort after time column
TM<-cbind(tall,Lall)
TM<-TM[order(TM[,1]),]
#Levy process Lt
Lt<-cumsum(TM[,2])

#calculate the constant C (for compound Poisson)
C=gamma*sqrt(2*rate/pi)
#calculate Lévy process Mt
M<-theta*Levy+gamma*abs(Levy) #berechne die Mt zu den randomjumptimes für die wir Levy generiert haben

#add zerojump and M to Mall
Mall<-c(zerojump,M)
#generate matrix with columns time and M, then sort after time column
TM<-cbind(tall,Mall)
TM<-TM[order(TM[,1]),]

#indices of the jumptimes in TM
isprung<-which(TM[,2] !=0)
N<-length(TM[,1])  #length time

#calculate Mt
TI<-c(TM[1,1],(TM[2:N,1]-TM[1:(N-1),1])) #timeintervals
TM[,2]<-TM[,2]-C*TI
Drift<-C*TI

Mt<-cumsum(TM[,2])

#plot(TM[,1],Mt,type="l")
#plot(Lt,type="l")

#-----------------------------------------------------
#calculate Xt with Euler approximation
X<-matrix(0,q,N+1)
Xlimleft<-matrix(0,q,N+1)

indexsprung=1
for(j in 1:N){
  if(j==isprung[indexsprung]){
    X[,j+1]<-X[,j]+TI[j]*(A%*%X[,j])+TM[j,2]*e
    Xlimleft[,j+1]<-X[,j]+TI[j]*(A%*%X[,j])+Drift[j]*e
    indexsprung=min(indexsprung+1,length(isprung))
  }else{
    X[,j+1]<-X[,j]+TI[j]*(A%*%X[,j])+TM[j,2]*e
    Xlimleft[,j+1]<-X[,j]+TI[j]*(A%*%X[,j])+TM[j,2]*e
  }
}

#----------------------------------
#calculate Yt=log(sigma_t^2)
Y<-matrix(0,1,N)
Ylimleft<-matrix(0,1,N)

if(q==1){
  Y<-B*X[,(2:(N+1))]+mu
  Ylimleft<-B*Xlimleft[,(2:(N+1))]+mu
}else{
  Y<-B%*%X[,(2:(N+1))]+mu
  Ylimleft<-B%*%Xlimleft[,(2:(N+1))]+mu}

Tylimleft<-TM[,1]

#volatility with jump
sigmaJump<-exp(Y)

#logvolatility
logsigmaexpTime<-Ylimleft[isprung]
#logsigma<-prevTick(t,c(0,Tylimleft),c(mu,Y))

#generate Levynoise Brownian+jump
Levynoise<-rnorm(N,mean,sqrt(TI))
```

```
  Brownian<-cumsum(Levynoise)

  Levynoise[isprung]<-Levynoise[isprung]+Levy

  Lt<-cumsum(Levynoise)
  #logprice process
  GexpTime<-cumsum(sqrt(exp(Ylimleft))*Levynoise)

  G=prevTick(t,c(0,TM[,1]),c(0,GexpTime)) #function prevTick puts G on a specified time grid t


  return(list(G,GexpTime,logsigmaexpTime,Ylimleft,Y,Tylimleft,Brownian,Mt,Lt,TM,sigmaJump))

}
```

R-Code 6.16: Simulation of an ECOGARCH(p,q) process

An example of the application of the function `ecogarch_sim` for the simulation of an ECOGARCH(p,q) process can be found in R-Code 6.17

```
t=0:100
a=c(-0.1,-0.3)
b=c(1,1)
mu=-4
theta=-0.2
gamma=0.1
rate=1
mean=0
var=1

output<-ecogarch_sim(t,a,b,mu,theta,gamma,rate,mean,var)
G<-unlist(output[[1]])
GexpTime<-unlist(output[[2]])
logsigmaexpTime<-unlist(output[[3]])
Ylimleft<-unlist(output[[4]])
Y<-unlist(output[[5]])
Tylimleft<-unlist(output[[6]])
Brownian<-unlist(output[[7]])
Mt<-unlist(output[[8]])
Lt<-unlist(output[[9]])
TM<-unlist(output[[10]])
sigmaJump<-unlist(output[[11]])

#plots
par(mfrow=c(5,1),mar=c(1.8,4.5,1,1))
plot(TM[,1],Lt,cex=0.01,ylab=expression(L[t]),xaxs="i",xlim=c(0,100))
plot(TM[,1],Mt,cex=0.01,ylab=expression(M[t]),xaxs="i",xlim=c(0,100))
plot(TM[,1],GexpTime,cex=0.01,ylab=expression(G[t]),xaxs="i",xlim=c(0,100))
plot(TM[,1],c(0,(GexpTime[2:length(GexpTime)]-GexpTime[1:(length(GexpTime)-1)])),
type="l",ylab=expression(G[t]^(r)),xaxs="i",xlim=c(0,100))
plot(TM[,1],sigmaJump,cex=0.01,ylab=expression(sigma[t]^2),xaxs="i",xlim=c(0,100))
```

R-Code 6.17: Simulation of an ECOGARCH(p,q) process driven by a compound Poisson process.

6.4 Simulation and Estimation of a GJR COGARCH$(1,1)$ process

This chapter is based on the theory introduced in Chapter 5. We implemented the function `cogarch_sim_assym`, see R-Code 6.18, for the simulation of a continuous time GJR GARCH$(1,1)$ model. Furthermore, we extended the PML method for the estimation of the parameters β, η, φ and γ of an asymmetric continuous time model, explicitly the continuous time GJR GARCH$(1,1)$ model. The estimation can be done with the function `pml_asym`, see R-Code 6.20.

6.4.1 Simulation of a continuous time GJR GARCH(1,1) process

For the simulation of the continuous time GJR GARCH(1,1) model we modified the simulation function for the symmetric COGARCH(1,1) process.
In Chapter 6.1.3 we showed how we can use the Euler scheme to get an approximation of the volatility process of the symmetric COGARCH(1,1) model. Analoguously to (6.5), (6.6) and (6.7) we can get equations which can be approximated with the Euler scheme.

For the equations

$$dY_t = -\eta Y_{t-}dt + V_t d[L,L]_t^{(d)} \quad \text{and} \quad V_t = \frac{\beta}{\eta} + \varphi^* Y_{t-}, \tag{6.10}$$

with $\varphi^* = \varphi\left((1-\gamma)^2 \mathbb{1}_{\{\Delta L_s > 0\}} + (1+\gamma)^2 \mathbb{1}_{\{\Delta L_s < 0\}}\right)$, it follows that

$$\begin{aligned} dV_t = \varphi^* Y_t &= -\varphi^* \eta \frac{V_t - \beta/\eta}{\varphi^*} dt + \varphi^* V_t d[L,L]_t^{(d)} \\ &= -\eta V_t dt + \beta dt + \varphi^* V_t d[L,L]_t^{(d)}. \end{aligned}$$

Hence we have

$$V_t = \beta t - \eta \int_0^t V_s ds + \varphi^* \sum_{0<s\leq t} V_s \Delta L_s + V_0. \tag{6.11}$$

Equation (6.11) is also satisfied for the volatility process $(\sigma_t^2)_{t\geq 0}$ of Equation (5.20), as we have

$$\sigma_t^2 = \beta t - \eta \int_0^t \sigma_s^2 ds + \varphi^* \sum_{0<s\leq t} \sigma_s^2 \Delta L_s + \sigma_0^2. \tag{6.12}$$

If we compare this equation to (6.7) we can observe that they coincide except of the parameters φ and φ^* respectively. Therefore, the Euler scheme can be applied to (6.10) analogously as it has been applied to Equation (6.5). We get

$$Y_{i+1} = Y_i - \eta Y_{i-}\Delta t_i + V_i \Delta L_i^2 \quad \text{and} \quad V_i = \frac{\beta}{\eta} + \varphi^* Y_{i-}, \tag{6.13}$$

with $V_0 = \beta/\eta$ and $Y_0 = 0$. In the corresponding R-Code 6.18 of the function `cogarch_sim_assym` we first simulate the driving Lévy process for a small amount of time points. We choose via the input arguments either a compound Poisson or a Variance Gamma process. We then initialize the Y_{i+1} by `Yi`, the volatility V_i by `voli` and ΔG_t by `g`. Afterwards we conduct a simulation of the driving Lévy process and its random jumptimes for the whole time interval chosen by the input parameter `t`. We then calculate `vol`, where `vol[1]<-voli`, `Y`, where `Y[1]<-Yi` and `delta_G`, where `delta_G[1]<-g`. Finally we obtain the desired continuous time GJR GARCH(1,1) process G_t by using the cumulated sum of the ΔG_t. The function `cogarch_sim_assym`, see R-Code 6.18, coincides with the function `cogarch_sim`, see R-Code 6.3, except of the parameters φ and φ^* respectively.

```
cogarch_sim_assym<-function(t=0:10,beta=1,eta=0.05,phi=0.03,gamma=0.4,Lp="cp",rate=1,distribution="normal",mean=0,var=1,sigma
    =1,nu=0.5,theta=1,gs=0.01)
{
  t<-sort(t)
  n_t<-length(t)

#find starting values
  if(Lp=="cp"){
    index<-1
  }else if(Lp=="vg"){
    index<-2
  }else{
    stop("No valid Levy process specified! Select 'cp' or 'vg'.")
  }
  switch(index,
{output_cp<-compoundPoisson(0:10,rate,distribution,mean,var) #calls the function compoundPoisson with output c(randomjumptimes
    ,randomjumpsizes,Lt)
 rt<-output_cp[,1] #randomjumptimes
 #delta_rt<-rt[2:N]-rt[1:(N-1)] #randomjumpintervals, delta_ti
 delta_rt<-output_cp[,4]#randomjumpintervals, delta_ti
 delta_Lt<-output_cp[,2] #randomjumpsizes, delta_Lti
 Lt<-output_cp[,3] #L_t
},
{output_vg<-vargamma(0:10,sigma,nu,theta,gs) #calls the function varGamma with output timesequence and V
 Lt<-output_vg[,2]
 rt<-output_vg[,1]
 nv<-length(Lt)
 nt<-length(rt)

 #increments of the process
 delta_Lt<-Lt[2:nv]-Lt[1:(nv-1)]
 delta_rt<-rt[2:nt]-rt[1:(nt-1)]
}
  )

  n<-length(delta_Lt)

  delta_Lt_0<-c(0,delta_Lt[1:(n-1)]) #used for sigma_t-, i.e. the sigma_t without the jump at time t!

#asymmetric COGARCH  gamma definieren
  term<-rep(0,length=length(delta_Lt_0))
  for(i in 1:length(delta_Lt_0)){
  if(delta_Lt_0[i] < 0){
    term[i]<-phi*(1+gamma)^2
  }else{
    term[i]<-phi*(1-gamma)^2
  }}
 phi_neu<-rep(0,length=length(delta_Lt_0))
 phi_neu<-term
########

#start value volatility, for state process set Y=0 in the beginning and therefore for the observation process: V = beta/eta +
    phi*Y = beta/eta
  voli<-beta/eta

  #state space process
  Yi<-0
  g<-0

  for (k in 1:n){

    Yi<-Yi-eta*Yi*delta_rt[k]+voli*delta_Lt_0[k]^2  # delta_Lt hat null im ersten Vektoreintrag, daher erhalte sigma_t-
    voli<-beta/eta+phi_neu[k]*Yi
    g<-g+sqrt(voli)*delta_Lt[k]
  }

#-----------------------------------------------
#obtain values via Euler approximation

  switch(index,
{output_cp_new<-compoundPoisson(t,rate,distribution,mean,var) #calls function compoundPoisson

 rt_new<-output_cp_new[,1] #randomjumptimes
 #NN<-length(rt_new)
 #delta_rt_new<-rt_new[2:NN]-rt_new[1:(NN-1)] #randomjumpintervals, delta_ti
 delta_rt_new<-output_cp_new[,4]
 delta_Lt_new<-output_cp_new[,2] #randomjumpsizes, delta_Lti
 Lt_new<-output_cp_new[,3] #L_t
},
{output_vg_new<-vargamma(t,sigma,nu,theta,gs) #calls the function varGamma with output timesequence and V

 Lt_new<-output_vg_new[,2]
 rt_new<-output_vg_new[,1]
 nvn<-length(Lt_new)
 ntn<-length(rt_new)

 #increments of the process
 delta_Lt_new<-Lt_new[2:nvn]-Lt_new[1:(nvn-1)]
 delta_rt_new<-rt_new[2:ntn]-rt_new[1:(ntn-1)]

}
  )
```

```
  nn<-length(delta_Lt_new)
  #add again 0 as first entry of delta_Lt_new in order to get sigma_t- !
  delta_Lt_new_0<-c(0,delta_Lt_new[1:(nn-1)]) #length nn

#asymmetric COGARCH  gamma definieren!
  term<-rep(0,length=length(delta_Lt_new_0))
  for(i in 1:length(delta_Lt_new_0)){
    if(delta_Lt_new_0[i] < 0){
      term[i]<-phi*(1+gamma)^2
    }else{
      term[i]<-phi*(1-gamma)^2
    }}
  phi_neu<-rep(0,length=length(delta_Lt_new_0))
  phi_neu<-term

  #volatility
  vol<-vector(length=length(nn))
  vol[1]<-voli

  #state space
  Y<-vector(length=length(nn))
  Y[1]<-Yi

  for (k in 1:nn){

    Y[k+1]<-Y[k]-eta*Y[k]*delta_rt_new[k]+vol[k]*delta_Lt_new_0[k]^2
    vol[k+1]<-beta/eta+phi_neu[k]*Y[k]
  }

  delta_G<-vector(length=length(vol))  #length(vol)=nn+1 mit nn=length(delta_Lt_new)
  delta_G[1]<-g
  s<-sqrt(vol[1:(length(vol)-1)]) #length=nn
  delta_G[2:length(vol)]<-s*delta_Lt_new[1:nn] #length=nn
  G<-cumsum(delta_G) #length=nn+1

  switch(index,
{output<-cbind(G,vol,c(0,Lt_new),c(0,delta_Lt_new),c(0,rt_new))
 colnames(output)<-c("G","vol","Lt","delta_Lt","randomjumptimes")
},
{output<-cbind(G,vol,Lt_new,c(0,delta_Lt_new),rt_new)
 colnames(output)<-c("G","vol","Lt","delta_Lt","randomjumptimes")
}
  )
  return(output)
}
```

R-Code 6.18: Simulation of a continuous time GJR GARCH(1,1) process

A simulation of a sample path of a continuous time GJR GARCH(1,1) process driven by a Variance Gamma process can be conducted as in R-Code 6.19.

```
t<-0:500
beta=0.08
eta=0.06
phi=0.04
gamma=0.6

Lp="vg" #Variance Gamma process
sigma<- 1
theta<- 0
nu<-1
gs<-0.01

output<-cogarch_sim_assym(t,beta,eta,phi,gamma,Lp,rate,distribution,mean,var,sigma,nu,theta,gs)
G<-output[,1]
vol<-output[,2]
Lt<-output[,3]
delta_Lt<-output[,4]
rt<-output[,5]

#plots
par(mfrow=c(4,1))
plot(rt,Lt,cex=0.1,xlim=c(0,500),xaxs="i",xlab="t",ylab=expression(L[t]))
plot(rt,G,cex=0.1,xlim=c(0,500), xaxs="i",xlab="t",ylab=expression(G[t]))
plot(rt,vol,cex=0.1,xlim=c(0,500), xaxs="i",xlab="t",ylab=expression(sigma[t]))
plot(rt,delta_Lt,type="h",xlim=c(0,500),xaxs="i",xlab="t",ylab=expression(paste(Delta,L[t],sep="")))
```

R-Code 6.19: Simulation of a continuous time GJR GARCH(1,1) process.

6.4.2 Estimation of a continuous time GJR GARCH$(1,1)$ process

In Chapter 5.2 we introduced a pseudo-maximum-likelihood method for the estimation of the parameters of a continuous time GJR GARCH$(1,1)$ process. We adjusted the PML approach in Chapter 5.2.2 in order to apply it to the continuous time GJR GARCH$(1,1)$ model. In R-Code 6.20 we provide the function `pml_asym`. If we compare this function to the "basic" PML function `pml` for COGARCH$(1,1)$ models, we see that only the parameter φ is adjusted. For the "asymmetric PML approach" we have to modifiy the calculation of ρ_i^2 as provided in 5.39 and of σ_i^2 as stated in 5.41. Therefore, we define in our function a new parameter φ^* and replace the parameter φ in the function `pml` with φ^*.

```
pml_asym<-function(x){
  #Pseudo-Maximum-Likelihood function for the continuous time GJR GARCH(1,1) model
  #Input: x - vector with beta, eta, phi and gamma (beta, eta and phi can be estimated first with the Miment Estimation Method
       )
  #Output: LL - Pseudo-Maximum Likelihood function

  beta= x[1]
  eta=x[2]
  phi=x[3]
  gamma=x[4]

  #----------------------------
  #assym
  #check if return is positive or negative
  term<-rep(0,length=length(r))
  for(i in 1:length(r)){
    if(r[i] < 0){
      term[i]<-phi*(1+gamma)^2
    }else{
      term[i]<-phi*(1-gamma)^2
    }}
  phi_neu<-rep(0,length=length(r))
  phi_neu<-term

  psi<--eta+phi*(1+gamma^2)
  #-----------------------------------

  #start value:  sigma0=sigma0^2
  sigma0<-beta/(-psi)
  sigma<-rep(0,length=Nt)
  rho<-rep(0,length=Nt)
  L=rep(0,Nt)

  rho[1]<-(beta*delta_t[1])/(-psi)
  sigma[1]<-beta*delta_t[1]+exp(-eta*delta_t[1])*(beta/(eta-phi))+phi_neu[1]*exp(-eta*delta_t[1])*r[1]^2

  for (i in 2:(Nt)){
    #sigma^2 here defined as sigma
    sigma[i]<-beta*delta_t[i]+exp(-eta*delta_t[i])*sigma[i-1]+phi_neu[i]*exp(-eta*delta_t[i])*r[i]^2
  }

  rho[101:(Nt)]<-(sigma[100:(Nt-1)]-beta/(-psi))*((exp(psi*delta_t[101:(Nt)])-1)/(psi))+(beta*delta_t[101:(Nt)])/(-psi)

  L<-0.5*(r[101:(Nt)]^2/rho[101:(Nt)])+0.5*log(rho[101:(Nt)])

  LL = sum(L)

  LL<-LL+(Nt/2)*log(2*pi) #add last term of likelihood function
  return(LL)
}
```

R-Code 6.20: Estimation of the parameters of a continuous time GJR GARCH$(1,1)$ process with the PML Method

The function `pml_asym` can be used analogously to the pseudo-maximum-likelihood function described in 6.2.2. In R-Code 6.21 we give an example how to estimate the parameters of a continuous time GJR GARCH$(1,1)$ model with this function.

```
t<-0:1000
beta=1
eta=0.06
phi=0.0425
gamma=0.4
Lp="cp" #compound Poisson process, Lp="cp"
rate<-1
```

```
distribution<-"normal" #jumpsizes are normally distributed
mean<-0
var<-1

output_c<-cogarch_sim_assym(t,beta,eta,phi,gamma,Lp,rate,distribution,mean,var,sigma,nu,theta,gs)
G<-output_c[,1]
vol<-output_c[,2]
Lt<-output_c[,3]
delta_Lt<-output_c[,4]
rt<-output_c[,5]
N<-length(G)
r<-G[2:N]-G[1:(N-1)]
nr<-length(r)
delta_t<-rt[2:length(rt)]-rt[1:(length(rt)-1)] #for unequally spaced data
Nt<-length(delta_t)

par<-c(beta,eta,phi,gamma)
A<-matrix(c(1,0,0,0,0,1,0,0,0,0,1,0,0,1,-1,0,0,0,0,1,0,0,0,-1),ncol=4,byrow=TRUE)
d<-c(0.000001,0.01,0.01,0.001,0.01,-1)
m<- constrOptim(par, pml_asym, NULL , ui = A, ci = d)
estimates<-m$par
```

R-Code 6.21: Estimationof the parameters of a continuous time GJR GARCH(1,1) process.

Chapter 7

Application to Data

In this chapter we consider turbulent windspeed data from the Brookhaven National Laboratory. We will refer to this data as "Brookhaven data" in the following. We will fit a continuous time GJR GARCH$(1,1)$ model to this data set and apply the asymmetric PML method introduced in Chapter 5.2.2 for the parameter estimation.

7.1 Data

The information about the data set is taken from Ferrazzano (2010). The data set contains windspeed data from the Brookhaven National Laboratory, located in Upton, New York on Long Island. It is a high frequency data set, recorded by Drhuva (2000). The total Brookhaven data set consists of $n = 20 \cdot 10^6$ measurements sampled at 5000Hz, i.e. $5 \cdot 10^3$ points per second.

Ferrazzano and Fuchs (2013) proposed a method to estimate the increments $\Delta_k^n X(\omega)$ (of the intermittency process $X_0(\omega), \ldots, X_{\Delta n}(\omega)$ on a equidistant grid) from an observed sample $V_0(\omega), \ldots, V_{n\Delta}(\omega)$ of the velocity $V = (V_t)_{t \in \mathbb{R}}$.[35] This method - which, in principle, can be seen as applying an auto-regressive filter - has been employed to the Brookhaven data set. These estimates are treated then as if they were observed true increments of the intermittency process. In the following, we refer to these increments as "(intermittency) data".[36]

We will only consider the first $1 \cdot 10^6$ data points of the intermittency increments. The series then covers a time interval of approximately 3 minutes (200 seconds). In Figure 7.1 this data set can be seen. Furthermore we consider different time segments of this data set. We will refer to the data set containing the first $1 \cdot 10^6$ data points as Example 1, to the data set containing the data points in $[1, 10000]$ as Example 2, to the data set containing the data points in $[50000, 60000]$ as Example 3, to the data set containing the data points in $[500000, 510000]$ as Example 4 and to the data set containing the data points in $[1, 100000]$ as Example 5.

[35] The velocity increments display a distinctive clustering, this phenomenon is called intermittency.
[36] Cf. Ueltzhöfer (2013, Chapter 6.3).

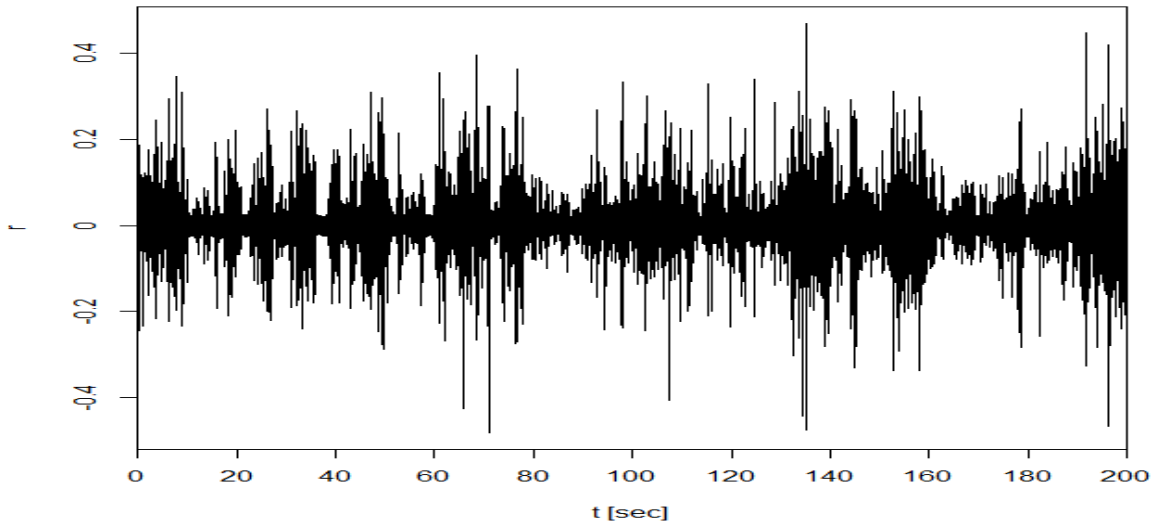


Figure 7.1: Increments of the Brookhaven data where the time is given in seconds and there are 5000 data points per second.

7.2 Parameter Estimation

We apply the asymmetric PML estimation method from Chapter 5.2.2 to our data sets of the different time segements (referred to as Examples 1 - 5).

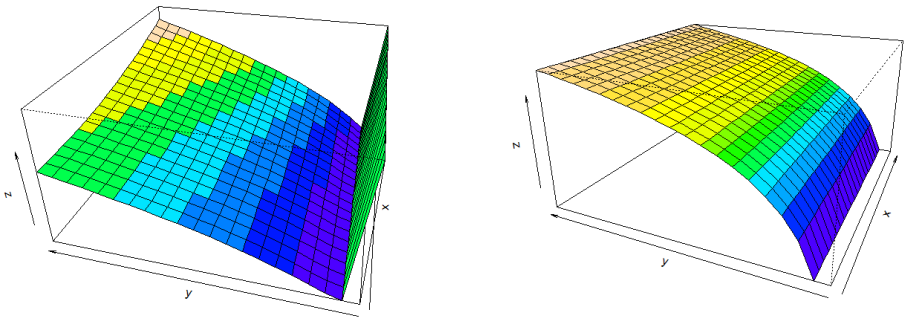
We start with Example 1 (containing the data points $[1, 1000000]$). In order to estimate the parameters $\hat{\beta}$, $\hat{\eta}$, $\hat{\varphi}$ and $\hat{\gamma}$ we need some starting values for these parameters. As the Brookhaven data is equally spaced in time, we can apply the moment estimation method to get starting values for $\hat{\beta}$, $\hat{\eta}$ and $\hat{\varphi}$. For $\hat{\gamma}$ we choose 0.3 as a starting value. Moreover, we had a look at Example 4^* which corresponds to Example 4 but uses $\hat{\gamma} = 0.8$. The results of the estimation are displayed in the first row of Table 7.1. We can observe that the value of $\hat{\beta}$ is very small. The lower boundary for it was set

Example	$\hat{\beta}$	$\hat{\eta}$	$\hat{\varphi}$	$\hat{\gamma}$
1	$1.3 \cdot 10^{-6}$	0.05999	0.04999	0.01000
2	$1.8 \cdot 10^{-6}$	0.07007	0.06007	0.02506
3	$3.0 \cdot 10^{-7}$	0.02249	0.01824	0.12117
4	$3.8 \cdot 10^{-6}$	0.10884	0.09809	0.00002
4^*	$3.7 \cdot 10^{-6}$	0.15995	0.15985	0.03636
5	$4.2 \cdot 10^{-7}$	0.02977	0.02681	0.00010

Table 7.1: Estimates of the different data sets

to $1 \cdot 10^{-7}$. The value of $\hat{\beta}$ obviously tends to this lower boundary. For $\hat{\gamma}$ we have a similar case, i.e. the value is equal to its lower boundary which was set to 0.01. For the Example 4 and 5 this boundary was even lower ($1 \cdot 10^{-5}$) and again the estimated value

tends to this boundary. For Example 2 we estimated the starting parameters with the moment estimation method. The obtained values are $\hat{\beta} = 0.000005$, $\hat{\eta} = 0.072199$ and $\hat{\varphi} = 0.060064$. We would like to see what the pseudo-log-likelihood function looks like if we fix the values of $\hat{\beta}$ and $\hat{\eta}$ and vary the values of $\hat{\varphi}$ and $\hat{\gamma}$ between 0 and 1. The result is shown in Figure 7.2(a). In this figure x denotes $\hat{\gamma}$, y denotes $\hat{\varphi}$ and z stands for the log-likelihood value. The smallest value of the pseudo-log-likelihood function is obtained when both $\hat{\varphi}$ and $\hat{\gamma}$ are very small. In Figure 7.2(b) we fixed $\hat{\varphi} = 0.060064$ and $\hat{\eta} = 0.072199$ and varied $\hat{\beta}$ and $\hat{\gamma}$ between 0 and 1. Again x denotes $\hat{\gamma}$ and z stands for the log-likelihood value and y denotes $\hat{\beta}$.



(a) fixed $\hat{\beta} = 0.000005$ and $\hat{\eta} = 0.072199$ and variable $\hat{\varphi}$ and $\hat{\gamma}$ between 0 and 1.

(b) fixed $\hat{\varphi} = 0.060064$ and $\hat{\eta} = 0.072199$ and variable $\hat{\beta}$ and $\hat{\gamma}$ between 0 and 1.

Figure 7.2: Pseudo-log-likelihood function of Example 2.

Furthermore we calculated $\hat{\sigma}^2$ recursively. For Example 1 the values of $\hat{\sigma}^2$ can be seen in Figure 7.3. Using these estimated values of the volatility process the residuals

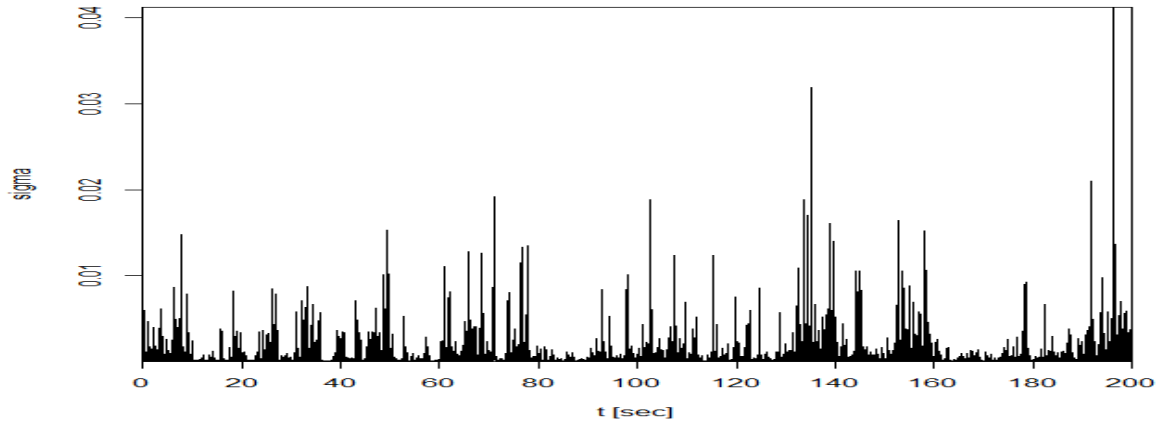


Figure 7.3: $\hat{\sigma}^2$ of Brookhaven data

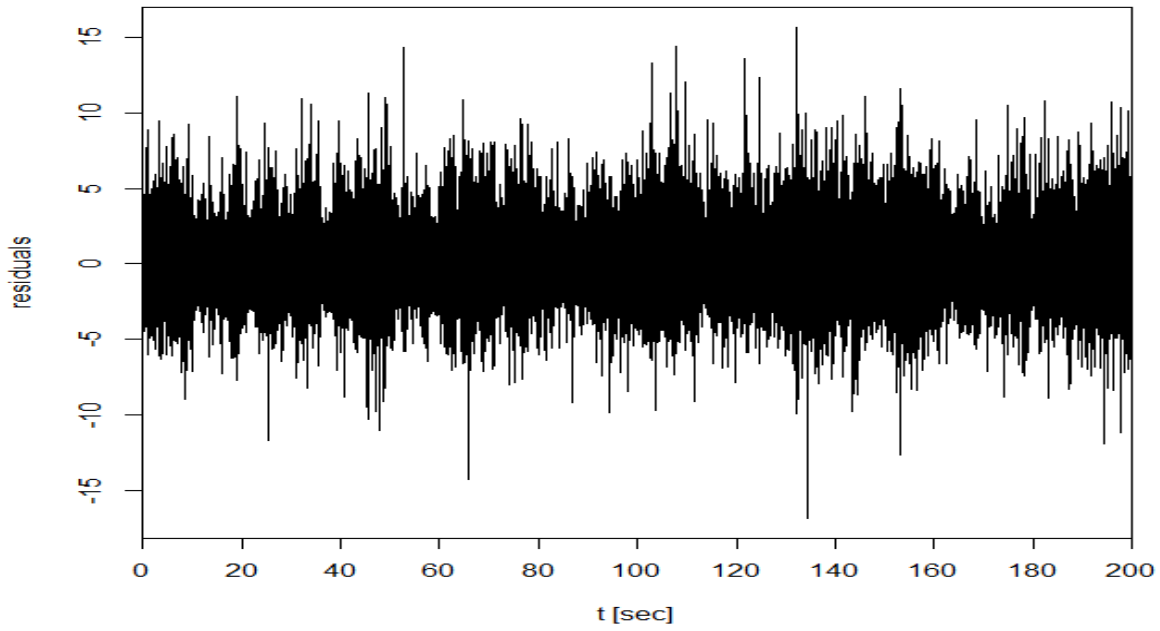


Figure 7.4: Residuals of Brookhaven data

can be calculated. For Example 1 these are shown in Figure 7.4. The mean, standard deviation and the skewness of the residuals of all our examples can be found in Table 7.2. Again Example 4* corresponds to Example 4, but $\hat{\gamma} = 0.8$ was chosen as a starting value. The mean values of the residuals show for all examples are close to 0 according to our model. The standard deviations of the residuals are close to 1 as expected for our model. The skewness values of the residuals indicate that the residuals are significantly skewed.

Example	mean$(G_i^{(1)}/\hat{\sigma}_{i-1})$	std$(G_i^{(1)}/\hat{\sigma}_{i-1})$	skewness$(G_i^{(1)}/\hat{\sigma}_{i-1})$
1	-0.00079	1.04947	0.31473
2	0.00103	1.03068	0.28618
3	0.02577	1.01133	0.10719
4	-0.00521	1.01682	0.34661
4*	-0.00078	1.01115	0.27702
5	0.00647	1.02874	0.24817

Table 7.2: Residuals of the different data sets

In Figure 7.3 the empirical autocorrelation functions (ACF) of the squared returns (top) and of the squared residuals (bottom) for Example 1, i.e. $1 \cdot 10^{-6}$ data points, are plotted. We can observe significant correlations of the squared residuals. Additionally we plotted the empirical autocorrelation functions for the other examples (2-5) in Figure 7.6. Again the top plot of each example displays the ACFs for the squared re-

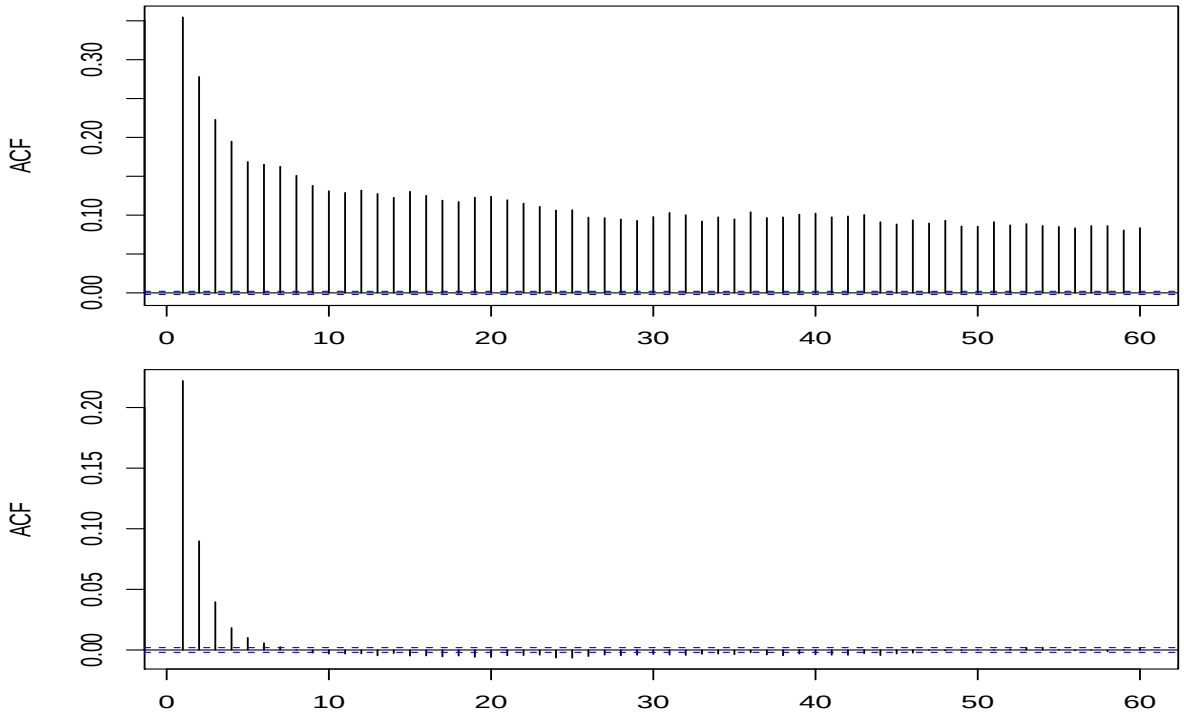
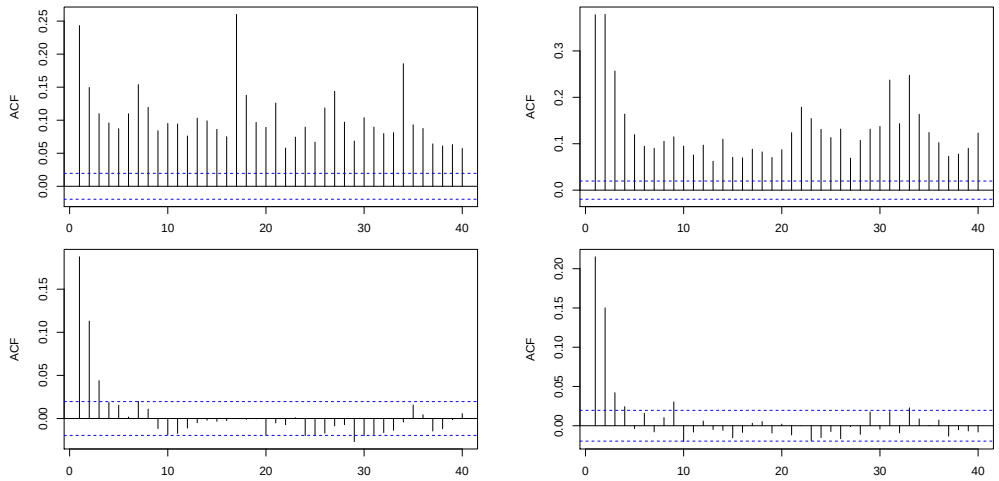


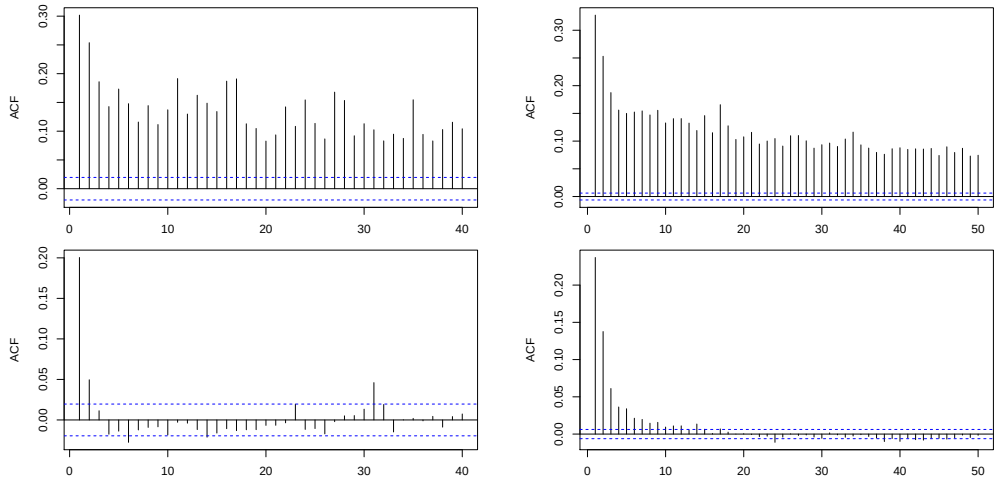
Figure 7.5: ACF of squared returns *(top)* and squared residuals *(bottom)* of 1.Mio data points

turns and the bottom plot for the squared residuals, which are significantly correlated again. We can conclude that this model does not perfectly fit to our data. In Chapter 5.2.3 we estimated the parameters of 5000 sample paths of the continuous time GJR GARCH$(1,1)$. The estimated values for $\hat{\gamma}$ already differed from the expected value of 0.3. Therefore it should be considered if it is possible to obtain better results by adding some constraints to the otimization procedure or finding an approach to get a better starting value for $\hat{\gamma}$.



(a) Example 2

(b) Example 3



(c) Example 4

(d) Example 5

Figure 7.6: ACFs of squared returns *(top of each plot)* and squared residuals *(bottom of each plot)* of different data sets.

Appendix A

A.1 Some R-Codes

R-Code A.1: Compound Poisson process

```
compoundPoisson <-
function(t=0:10,rate=1,distribution="normal",mean=0,var=1){
#jumps arrive accordingly to a poisson process
#size of jumps is also random, with a specified probability distribution,
# sprung size independent of jump time and iid distributed with a specified distribution
#jumpintervals are random: deltaT_n=T_n-T_(n-1) iid exponentialrvs

#Input: t- time grid
#       rate - intensity of compound Poisson process
#       distribution - jump size distribution, e.g. normal, uniform, bernoulli
#       mean, var  - mean and variance of the specified distribution
#Output: matrix with the jumptimes, values of delta_Lt, Lt and length of jumptimes

#########################

t<-sort(t) #sort vector t
nt<-length(t)  #number of timepoints
I<-t[2:nt]-t[1:(nt-1)]  #timeintervals
Nk<-length(I) #vector with lengths of timeintervals

#heuristic for number of jump times needed to cover t
nmax <- ceiling(t[nt] *2*rate)
maxtime <- t[nt] #maximal entry of time, i.e. last timepoint

#set up the matrix which will contain the sigmas for t
sigma.t.process.data <- matrix(nrow = nt, ncol = 2)
sigma.t.process.data[, 1] <- t #The t are in the first column

#Create the random jump times.
#Assume that deltaT_n=T_n-T_(n-1) iid exponentialrvs and N_t=max(n>=1:T_n<=t)
#Generate  nmax (defined as nmax <- ceiling(t[n] * 2*rate) random jump times distributed as exp(rate)
 randomjumpintervals <- rexp(nmax, rate = rate)     #deltaT_n are exponentially distributed (generate more intervals then
      necessary!)

#as long as it holds that: sum of the jumpintervals <= maxtime+1, write them into the vector
 validtimes=T
 k<-1

while(validtimes)
 {randomjumptimes <- cumsum(randomjumpintervals)
  if(randomjumptimes[nmax]>maxtime){
    if(randomjumptimes[k]< maxtime){
      randomjumpintervals[k]<-randomjumpintervals[k]
      randomjumptimes[k]<-randomjumptimes[k]
      k<-k+1
    }else{
      randomjumpintervals<-randomjumpintervals[1:(k-1)]
      randomjumptimes<-randomjumptimes[1:k]
      n<-length(randomjumptimes)
      validtimes=F
    }
  }else{
    #if not enough intervals to cover all t, extend the vector with exponentially distributed rvs
    randomjumpintervals<-c(randomjumpintervals,rexp(nmax,rate=rate))
  }
}

#Calculate the process Lt
if (distribution== "normal"){
   std<-sqrt(var)
   nrv<- rnorm(n,mean,std) #normally distributed with mean and sigma=sqrt(var)
   }else{
```

```
    if(distribution=="bernoulli"){
      p <- mean #has to be <=1
      nrv <- rbinom(n,1,p) #Bernoulli distributed
    }else{
      if(distribution=="uniform"){
        parameter1<-mean-sqrt(3*var)
        parameter2<-mean+sqrt(3*var)
       nrv <-  runif(n, parameter1, parameter2) #uniform distributed
      }else{
        if(distribution=="binomial"){
          parameter1<-round((mean^2)/(mean-var)) #has to be in |N!
          parameter2<-(mean-var)/mean #has to be <=1
          nrv <-  rbinom(n, parameter1, parameter2) #binomial distributed
        }else{
          stop("No valid distribution for the jumps specified!")
        }
      }
    }
  }

#summation of the jumpsizes and write them in Lt[i]
Lt<- cumsum(nrv)

#output matrix with the jumptimes, values of delta_Lt and values of Lt
output<-matrix(c(randomjumptimes,nrv,Lt,c(randomjumpintervals,0)),n)
#randomjumpsizes=delta_Lt, T=randomjumpintervals incl. 0 (for the right dimensions of outputmatrix)
colnames(output)<-c("randomjumptimes","delta_Lt","Lt","T")
return(output)
}
```

R-Code A.2: Variance Gamma Process

```
vargamma <-
function(t=0:10,sigma=1,nu=0.05,theta=0.5,gs=0.01){

#simulation of a variance gamma process V with parameters sigma>0, nu>0 and theta
#TG: time change is done wrt a Gamma process which is defined as TG
#W_TG: standard Brownian Motion
#
# V=theta*TG+sigma*W_TG
#Input: parameters sigma, nu, theta, t, gs=gridsize
#Output: variance gamma process V and ts (sequence of small time grid)

################

n<-length(t)
ts<-seq(0,t[n],gs) #generate timepoints on a very small grid
nt<-length(ts)
delta_ts<-ts[2:nt]-ts[1:(nt-1)] #time intervals
nd<-length(delta_ts)

g<-rgamma(nd,shape=(1/nu)*delta_ts,scale=nu) #randomvariables which are gammadistributed with shapeparameter=(1/nu)*delta_t
      and scaleparameter=nu
TG<-vector(length=length(g)+1)
TG[1]<-0
TG[2:(nd+1)]<-cumsum(g) #time changes are done wrt to gamma process

w<-rnorm(nd,0,sqrt(g)) #normal randomvariables with mean=0 and sigma=sqrt(g)
#w<-rnorm(nd,0,g)   #g=TG[idelta_t]-TG[(i-1)delta_t]
W_TG<-vector(length=length(w)+1)  #vector for standard brownian motion
W_TG[1]<-0
W_TG[2:(nd+1)]<-cumsum(w)

V<-theta*TG+sigma*W_TG  #variancegamma process


return(cbind(ts,V))  #return small timeintervalls and the variance gamma process

}
```

R-Code A.3: Simulate N sample paths of a COGARCH$(1,1)$ process driven by a compound Poisson process

```
Cogarch_cp_Npaths<-function(t=0:10,N=5,beta=1,eta=0.05,phi=0.03,Lp="cp",rate=1,distribution="normal",mean=0,var=1){
  #simulate N path of a cogarch11 process with compound Poisson levy process
  #for fixed time grid (with prev tick program)

  #Input: t - fixed time grid
  #       N - number of samples to be generated
  #       beta, eta, phi - parameters for cogarch process
  #       Lp - specify driving Lévy process: "cp"=compound Poisson
  #       rate - intensity lambda of cp
  #       distribution - distribution of jumps for cp : "normal"
  #       mean, var - mean and variance of specified jump distribution
  #Output: G - cogarch11 process
  #        rt - jumptimes
```

```
  G<-matrix(0,nrow=length(t),ncol=N)
  rt<-list()

  i<-0
  while(i<N){
    output<-cogarch_sim(t,beta,eta,phi,Lp="cp",rate,distribution,mean,var) #output:random jump times
    it<-output[,5] #irregular jump times of cogarch process
    X<-output[,1] #simulated Cogarch process

    pt<-prevTick(t,it,X) #uses function prevtick to get process at specified times t
    G[,i+1]<-pt
    rt[[length(rt)+1]]<-it
    i<-i+1
  }

  return(list(G=G,rt=rt))
}
```

R-Code A.4: Simulate N sample paths of a COGARCH$(1,1)$ process driven by a Variance Gamma process

```
Cogarch_vg_Npaths<-function(t=0:10,N=5,beta=1,eta=0.05,phi=0.03,Lp="vg",sigma=1,nu=0.3,theta=1,gs=0.01){
  #simulate N path of a cogarch11 with variance gamma levy process
  #for fixed time grid (with prev tick program)

  #Input: t - fixed time grid
  #       N - number of samples to be generated
  #       beta, eta, phi - parameters for cogarch process
  #       Lp - specify driving Lévy process: "vg"=Variance Gamma
  #       sigma, nu, theta - parameters for Variance Gamma process (for E(L_1)=0 and E(L_1^2)=1 choose theta=0 and sigma=1)
  #       gs - gridsize
  #Output: G - cogarch11 process
  #        rt - jumptimes


  G<-matrix(0,nrow=length(t),ncol=N)
  rt<-list()

  i<-0
  while(i<N){
    output<-cogarch_sim(t,beta,eta,phi,Lp="vg",sigma,nu,theta,gs) #output:random jump times!
    #wenn compoundPoisson with exponentially distributed jumptimeintervals: output immer unterschiedlich lang, je nachdem
        wieviele sprünge!!
    it<-output[,5] #irregular jump times of cogarch process
    X<-output[,1] #simulated Cogarch process

    pt<-prevTick(t,it,X) #uses function prevtick

    G[,i+1]<-pt
    rt[[length(rt)+1]]<-it
    i<-i+1
  }

  return(list(G=G,rt=rt))
}
```

R-Code A.5: Estimation of the parameters of a COGARCH$(1,1)$ process with $N > 1$ sample paths as an input

```
est_cogarch <-
function(G){
#Estimation of parameters with moment estimation method

#input: G =EQUALLY spaced Cogarch11 process with Lévy process: compound Poisson or variance gamma
#         with N>1
#       #d=lags ,choose d=sqrt(n)
#output: betahat,etahat,phihat,p,k

  #G has several columns
n<-dim(G)[1]
s<-dim(G)[2]
d<-floor(sqrt(n)) #number of lags, see p.11 of paper 'Estimating the COGARCH(1,1) model-a first go'

#define output vectors
#value for every of the N simulated cogarch processes in a new column
m1<-matrix(0,nrow=1,ncol=s)
m2<-matrix(0,nrow=1,ncol=s)
p<-matrix(0,nrow=1,ncol=s)
k<-matrix(0,nrow=1,ncol=s)
betahat<-matrix(0,nrow=1,ncol=s)
etahat<-matrix(0,nrow=1,ncol=s)
phihat<-matrix(0,nrow=1,ncol=s)
y<-rep(0,(d+1)) #vector for acf
```

```
#calculate the returns
for (m in 1:s){
r<-(G[2:n,m]-G[1:(n-1),m])^2

nr<-length(r)

#moments
mu<-mean(r)  #mean
gamma<-mean(r^2) #variance

m1[m]<-mu #first moments
m2[m]<-gamma #second moments

#####calculate autocovariances and autocorrelations
muhat<-mu*rep(1,nr) #row vector
g0<-sum((r-muhat)*(r-muhat))/nr

for (h in 0:d){
  u<-nr-h
  y[h+1]<-sum((r[(1+h):nr]-mu*rep(1,u))*(r[1:u]-mu*rep(1,u)))
}
y<-y/(nr*g0) #empirical autocorrelation
acfr<-y[2:(d+1)]  #analogously to Matlab program

#################

####regression least square method to obtain k and p
n_acfr_pos<-which(acfr>0) #rows with positive acfr entry
z<-log(acfr[n_acfr_pos])
#X<-cbind(matrix(1,nrow=length(n_acfr_pos),ncol=1),-n_acfr_pos)
X<-(-n_acfr_pos)
#par<-lm(z~X)  #model with intercept=log(k), X=p#
require(MASS)
par<-rlm(z~X)#robust regression
par1<-as.numeric(par$coefficients[1])
par2<-as.numeric(par$coefficients[2])
k[m]<-exp(par1)
p[m]<-par2
#################
  if(p[m]>0){
    betahat[m]<-p[m]*mu

  M1<-gamma-2*mu^2-6*((1-p[m]-exp(-p[m]))/((1-exp(p[m]))*(1-exp(-p[m]))))*k[m]*gamma
  M2<-(2*k[m]*gamma*p[m])/(M1*(exp(p[m])-1)*(1-exp(-p[m])))

   if(M2>0){
    phihat[m]<-p[m]*sqrt(1+M2)-p[m]
    etahat[m]<-p[m]*sqrt(1+M2)
    } else{
      cm<-rep(0,length(s))
      cm[m]<-1
      #stop("M2<=0!")
    }
  }else{
    cp<-rep(0,length(s))
    cp[m]<-1
    #stop("p<=0!")
  }

}#end of "for m=1:s"

est<-matrix(0,nrow=5,ncol=s)
est<-rbind(betahat,etahat,phihat,p,k) #estimated values
return(est)
}
```

A.2 Documentation for the R-Package

Package 'cogarch'

April 18, 2013

Type Package

Title Simulation and Estimation of COGARCH processes

Version 0.1.3

Date 2013-04-18

Author Marlit Granzer

Maintainer Marlit Granzer <marlit_g@hotmail.de>

Description This package provides functions for the context of continuous time GARCH processes. It can be used to simulate COGARCH(p,q) processes with different background Levy processes (compound Poisson and Variance Gamma). There are functions to generate N equally spaced sample paths of a COGARCH(1,1) process. Furthermore it is possible to simulate ECOGARCH(p,q) process or an asymmetric continuous time GJR GARCH(1,1) process. Moreover the package provides functions which can be used to estimated the model parameters. Forin time equally spaced data the moment estimation method can be used. Additionally the estimation can be conducted via a Pseudo-Maximum Likelihood (PML) estimation, which works also for data which is unequally spaced in time.
There is one pml function which can be applied under the assumption to have fixed (non-random) timepoints.
Without this assumption, i.e. we assume to have random timepoints like all the jumptimes, it is possible to apply the function pml_cp for the compound Poisson case.
For more details on the theory and descriptions of the functions, see the master's thesis, Estimation of COGARCH models and implementation in R, of Marlit Granzer.

License GPL (>=2)

Depends stats,utils,graphics,MASS,Matrix

R topics documented:

`cogarch-package` *Continuous Time GARCH Processes*

Description

This package provides functions for the context of continuous time GARCH processes. It can be used to simulate COGARCH(p,q) processes with different background Levy processes (compound Poisson and Variance Gamma). There are functions to generate N equally spaced sample paths of a COGARCH(1,1) process. Furthermore it is possible to simulate ECOGARCH(p,q) process or an asymmetric continuous time GJR GARCH(1,1) process. Moreover the package provides functions which can be used to estimated the model parameters. Forin time equally spaced data the moment estimation method can be used. Additionally the estimation can be conducted via a Pseudo-Maximum Likelihood (PML) estimation, which works also for data which is unequally spaced in time. There is one pml function which can be applied under the assumption to have fixed (non-random) timepoints. Without this assumption, i.e. we assume to have random timepoints like all the jumptimes, it is possible to apply the function pml_cp for the compound Poisson case. For more details on the theory and descriptions of the functions, see the master's thesis, "Estimation of COGARCH models and implementation in R"", 2013, of Marlit Granzer.

Details

Package:	cogarch
Type:	Package
Version:	0.1.3
Date:	2013-04-18
License:	GPL-2

Author(s)

Marlit Granzer

References

Marlit Granzer, Master's thesis "Estimation of COGARCH models and implementation in R", 2013, Technische Universitaet Muenchen.

`Cogarch_cp_Npaths` *N sample paths of a compound Poisson COGARCH(1,1) process*

Description

Simulates N sample paths of a COGARCH(1,1) process with a compound Poisson process as background Levy process. Furthermore the function "prevTick" is used, i.e. the data is transfered to a fixed time grid.

Usage

```
Cogarch_cp_Npaths(t=0:10,N=5,beta=1,eta=0.05,phi=0.03,Lp="cp",rate=1,
                  distribution="normal",mean=0,var=1)
```

Arguments

`beta`	model parameter
`eta`	model parameter
`phi`	model parameter
`N`	number of simulated sample paths
`t`	time
`Lp`	the background Levy process: "cp"=compound Poisson
`rate`	necessary if Lp="cp", rate for the exponentially distributed time intervals
`distribution`	necessary if Lp="cp", distribution of the jumpsizes
`mean`	necessary if Lp="cp", mean of the distribution of the jumpsizes
`var`	necessary if Lp="cp", variance of the distribution of the jumpsizes

Details

Simulates N sample paths of a COGARCH(1,1) process with a compound Poisson process as background Levy process . See "cogarch_sim" with Lp="cp" for further information. Furthermore the function "prevTick"" is used, i.e. the data is transfered to a fixed time grid.

Value

`G`	list with one of the N equally spaced sample paths in each of the N columns
`rt`	random jump times

Author(s)

Marlit Granzer

References

Marlit Granzer, Master's thesis "Estimation of COGARCH models and implementation in R", 2013, Technische Universitaet Muenchen.

Examples

```
N=3
t<-0:500
beta=0.05
eta=0.06
phi=0.04

Lp="cp" #compound Poisson process, Lp="cp"
rate<-1
distribution<-"normal" #jumpsizes are normally distributed
mean<-0
var<-1

output_cp<-Cogarch_cp_Npaths(t,N,beta,eta,phi,Lp,rate,distribution,mean,var)
G<-output_cp$G
rt<-output_cp$rt

#estimate parameters for the cogarch processes: Method of Moment Estimation
estimates_mm<-est_cogarch(G)#needs equally spaced data as an input
betahat<-estimates_mm[1,]
etahat<-estimates_mm[2,]
phihat<-estimates_mm[3,]
p<-estimates_mm[4,]
k<-estimates_mm[5,]

#mean
mb<-mean(betahat)
me<-mean(etahat)
mp<-mean(phihat)
ev_mm<-c(mb,me,mp) #estimated values
ov_mm<-c(beta,eta,phi) #original values
```

---

cogarch_pq_sim *Simulation of a COGARCH(p,q) process*

Description

Generates a sample path of a COGARCH(p,q) (p<=q) process with a compound Poisson process as underlying background Levy process.

Usage

```
cogarch_pq_sim(t=0:10,a0=0.04,a=0.038,b=-0.053,Lp="cp",rate=1,
               distribution="normal",mean=0,var=1)
```

Arguments

t	time
a0	corresponds to alpha_0, in order to get a starting value for the volatility process
a	vector of alpha_p, coefficients of model

b	vector of beta_q, coefficients of model
Lp	choose Levy process: "cp"=compound Poisson
rate	intensity of the cp Levy process
distribution	distribution of the jumpsizes of the compound Poisson process, e.g. "normal"
mean	mean of the jump distribution
var	variance of the jump distribution

Details

Generates a sample path of a COGARCH(p,q) process with a compound Poisson process as underlying background Levy process.

Value

Output: List with the following entries:

G	COGARCH(p,q) process
vol	Volatility of the process (sigma)
c(0,Lt_new)	driving Levy process
c(0,delta_Lt_new)	jumps sizes of Levy process
c(0,rt_new)	random jumptimes
eig	eigenvalues of matrix B

Author(s)

Marlit

References

Master's thesis

Examples

```
t=0:500

a0=1
a=c(1)
b=c(-1.2,-0.48-pi^2,-0.064-0.4*pi^2)
Lp="cp"
rate=1
distribution="normal"
mean=0
var=1

output<-cogarch_pq_sim(t,a0,a,b,Lp,rate,distribution,mean,var)
G<-unlist(output[[1]])
vol<-unlist(output[[2]])
Lt<-unlist(output[[3]])
delta_Lt<-unlist(output[[4]])
rt<-unlist(output[[5]])
eig<-unlist(output[[6]])
```

```
#plots
par(mfrow=c(4,1))
plot(c(0,rt),G,cex=0.1,xaxs="i",xlim=c(0,500),xlab="t")
plot(rt,diff(G),type="h",xaxs="i",xlim=c(0,500),ylab=expression(G^{(1)}),xlab="t")
plot(c(0,rt),vol,cex=0.1,xaxs="i",xlim=c(0,500),ylab=expression(sigma^2),xlab="t")
plot(rt,Lt,cex=0.1,xaxs="i",xlim=c(0,500),xlab="t")
```

`cogarch_sim` *Simulation of a COGARCH(1,1) process*

Description

Generates a sample path of a COGARCH(1,1) process with either a compound Poisson process or a variance gamma process as underlying background Levy process.

Usage

```
cogarch_sim(t=0:10,beta=1,eta=0.05,phi=0.03,Lp="cp",rate=1,
            distribution="normal",mean=0,var=1,sigma=1,nu=0.5,theta=1,gs=0.01)
```

Arguments

`t`	time
`beta`	model parameter
`eta`	model parameter
`phi`	model parameter
`Lp`	choose Levy process: "cp"=compound Poisson or "vg"=variance gamma
`rate`	intensity of the cp Levy process
`distribution`	distribution of the jumpsizes of the compound Poisson process, e.g. "normal"
`mean`	mean of the jump distribution
`var`	variance of the jump distribution
`sigma`	parameter for the variance gamma process
`nu`	parameter for the variance gamma process
`theta`	parameter for the variance gamma process
`gs`	time grid size

Details

Generates a sample path of a COGARCH(1,1) process with either a compound Poisson process or a variance gamma process as underlying background Levy process.

Value

Output: Matrix with the following columns:

G	COGARCH(1,1) process
vol	Volatility of the process (sigma)
c(0,Lt_new)	Levy process
c(0,delta_Lt_new)	jumps sizes of Levy process
c(0,rt_new)	random jumptimes

Author(s)

Marlit Granzer

References

Marlit Granzer, Master's thesis "Estimation of COGARCH models and implementation in R", 2013, Technische Universitaet Muenchen.

Examples

```
# t<-0:100
# beta<-0.04
# eta<-0.053
# phi<-0.038
#
# #Compound Poisson driven COGARCH(1,1) process
# Lp="cp" #compound Poisson process, Lp="cp"
# rate<-1
# distribution<-"normal" #jumpsizes are normally distributed
# mean<-0
# var<-1
#
# set.seed(100)
# output<-cogarch_sim(t,beta,eta,phi,Lp="cp",rate,distribution,mean,var)
# G<-output[,1]
# vol<-output[,2]
# Lt<-output[,3]
# delta_Lt<-output[,4]
# rt<-output[,5]
#
# #plot
# par(mfrow=c(4,1))
# plot(rt,G,xlim=c(0,100),cex=0.1, xaxs="i",xlab="t",ylab=expression(G[t]))
# plot(rt,c(0,G[2:length(G)]-G[1:(length(G)-1)]),type="l",xlim=c(0,100), xaxs="i",
#       xlab="t",ylab=expression(G[t]^(1)))
# plot(rt,vol,xlim=c(0,100), xaxs="i",cex=0.1,xlab="t",ylab=expression(sigma[t]^2))
# plot(rt,Lt,xlim=c(0,100),xaxs="i",cex=0.1,xlab="t",ylab=expression(L[t]))
#
# #-----------------------------------------------
# t<-0:100
# beta<-0.04
# eta<-0.053
# phi<-0.038
#
```

```
# #Variance-Gamma driven COGARCH(1,1) process
# Lp="vg"   #variance-gamma process, Lp="vg"
# sigma<- 1
# theta<- 0
# nu<-1
# gs<-0.01
#
# set.seed(100)
# output<-cogarch_sim(t,beta,eta,phi,Lp="vg",sigma,nu,theta,gs)
# G<-output[,1]
# vol<-output[,2]
# Lt<-output[,3]
# delta_Lt<-output[,4]
# rt<-output[,5]
#
# #plot
# par(mfrow=c(4,1))
# plot(rt,G,xlim=c(0,100),cex=0.1, xaxs="i",xlab="t",ylab=expression(G[t]))
# plot(rt,c(0,G[2:length(G)]-G[1:(length(G)-1)]),type="l",xlim=c(0,100), xaxs="i",
#       xlab="t",ylab=expression(G[t]^(1)))
# plot(rt,vol,xlim=c(0,100), xaxs="i",cex=0.1,xlab="t",ylab=expression(sigma[t]^2))
# plot(rt,Lt,xlim=c(0,100),xaxs="i",cex=0.1,xlab="t",ylab=expression(L[t]))
```

`cogarch_sim_assym` *Simulation of Assymetric continuous time GJR GARCH(1,1) process*

Description

Generates a sample path of an assymetric continuous time GJR GARCH(1,1) process with either a compound Poisson process or a variance gamma process as underlying background Levy process.

Usage

```
cogarch_sim_assym(t=0:10,beta=1,eta=0.05,phi=0.03,gamma=0.4,Lp="cp",rate=1,
                  distribution="normal",mean=0,var=1,sigma=1,nu=0.5,theta=1,gs=0.01)
```

Arguments

`t`	time
`beta`	model parameter
`eta`	model parameter
`phi`	model parameter
`gamma`	model parameter
`Lp`	Levy process: "cp"=compound Poisson or "vg"-variance gamma
`rate`	parameter for compound Poisson process
`distribution`	distribution of the jumpsizes of the compound Poisson process
`mean`	mean of the specified distribution
`var`	var of the specified distribution
`sigma`	parameter for the variance gamma process
`nu`	parameter for the variance gamma process
`theta`	parameter for the variance gamma process
`gs`	gridsize

Details

Generates a sample path of an asymmetric continuous time GJR GARCH(1,1) process with either a compound Poisson process or a Varince-Gamma process as underlying background Levy process.

Value

Output: Matrix with the folowing columns:

`G`	continuous time GJR GARCH(1,1) process
`vol`	volatility of the process (sigma)
`c(0,Lt_new)`	Levy process
`c(0,delta_Lt_new)`	jumps sizes of Levy process
`c(0,rt_new)`	random jumptimes

Author(s)

Marlit Granzer

References

Marlit Granzer, Master's thesis "Estimation of COGARCH models and implementation in R", 2013, Technische Universitaet Muenchen.

Examples

```
t<-0:100
beta=1
eta=0.06
phi=0.04
gamma=0.3

Lp="cp" #compound Poisson process,
rate<-1
distribution<-"normal" #jumpsizes are normally distributed
mean<-0
var<-1

#simulation
output<-cogarch_sim_assym(t,beta,eta,phi,gamma,Lp,rate,distribution,mean,var,sigma,nu,theta,gs)
G<-output[,1]
vol<-output[,2]
Lt<-output[,3]
delta_Lt<-output[,4]
rt<-output[,5]

#plots
par(mfrow=c(4,1))
plot(rt,Lt,type="l",xlim=c(0,100),xaxs="i",xlab="t",ylab=expression(L[t]))
plot(rt,G,type="l",xlim=c(0,100), xaxs="i",xlab="t",ylab=expression(G[t]))
plot(rt,vol,type="l",xlim=c(0,100), xaxs="i",xlab="t",ylab=expression(sigma[t]))
plot(rt,delta_Lt,type="l",xlim=c(0,100),xaxs="i",xlab="t",ylab=expression(paste(Delta,L[t],sep="")))
```

Cogarch_vg_Npaths *N sample paths of a variance gamma COGARCH(1,1) process*

Description

Simulates N sample paths of a COGARCH(1,1) process with a variance gamma process as background Levy process. Furthermore the function "prevTick"" is used, i.e. the data is transfered to a fixed time grid.

Usage

```
Cogarch_vg_Npaths(t=0:10,N=5,beta=1,eta=0.05,phi=0.03,Lp="vg",
                  sigma=1,nu=0.3,theta=1,gs=0.01)
```

Arguments

beta	model parameter
eta	model parameter
phi	model parameter
N	number of simulated sample paths
t	time
Lp	the background Levy process: "vg"=variance gamma
sigma	parameter used to calculate the variance gamma process V<-theta*TG+sigma*W_TG
nu	parameter used for the gamma distribution
theta	parameter used to calculate the variance gamma process V<-theta*TG+sigma*W_TG
gs	grid size

Details

Simulates N sample paths of a COGARCH(1,1) process with a variance gamma process as background Levy process . Furthermore the function "prevTick" is used, i.e. the data is transfered to a fixed time grid.

Value

G	list with one of the N equally spaced sample paths in each of the N columns
rt	random jump times

Author(s)

Marlit Granzer

References

Marlit Granzer, Master's thesis "Estimation of COGARCH models and implementation in R", 2013, Technische Universitaet Muenchen.

Examples

```
N=3
t<-0:500
beta=0.05
eta=0.06
phi=0.04

Lp="vg" #variance-gamma process
sigma=1
nu=0.3
theta=0
gs=0.01

output_vg<-Cogarch_vg_Npaths(t,N,beta,eta,phi,Lp="vg",sigma,nu,theta,gs)
G<-output_vg$G
rt<-output_vg$rt

#estimate parameters for the cogarch processes: Method of Moment Estimation
estimates_mm<-est_cogarch(G)#needs equally spaced data as an input
betahat<-estimates_mm[1,]
etahat<-estimates_mm[2,]
phihat<-estimates_mm[3,]
p<-estimates_mm[4,]
k<-estimates_mm[5,]

#mean
mb<-mean(betahat)
me<-mean(etahat)
mp<-mean(phihat)
ev_mm<-c(mb,me,mp) #estimated values
ov_mm<-c(beta,eta,phi) #original values
```

`compoundPoisson` *Simulation of a compound Poisson Levy process*

Description

This function gives the possibility to simulate compound Poisson processes with different parameters. It is possible to specify the jump distribution, e.g. "normally", "uniform",... and the mean and the variance of the distribution. The jumpintervals are exponentially distributed with the specified rate (intensity).

Usage

```
compoundPoisson(t, rate, distribution, mean, var)
```

Arguments

`t`	time vector, e.g. t<-0:100
`rate`	rate of exponentially distributed jumptimeintervals, correspondents to λ
`distribution`	distribution for the jumpsizes, e.g "normal", "uniform"
`mean`	mean of the specified distribution
`var`	variance of the specified distribution

Details

more details can be found in the Master's thesis "Estimation of COGARCH models and implementation in R", 2013, Technische Universitaet Muenchen, Marlit Granzer.

Value

`randomjumptimes`
: random times where the process jumps

`nrv`
: size of the jumps, i.e. delta_Lt

`Lt`
: compound Poisson Levy process Lt

`randomjumpintervals`
: randomjumpintervals which are iid exponentially distributed with chosen rate

Author(s)

Marlit Granzer

References

Marlit Granzer, Master's thesis "Estimation of COGARCH models and implementation in R", 2013, Technische Universitaet Muenchen.

Examples

```
delta=0.95 #parameter
sigma0squared = 20 #parameter
rate=1
distribution<-"normal"
mean<-0 #mu
var<-1 #variance

output<-compoundPoisson(0:10,1,distribution,mean,var)

jumptimes<-output[,1]
no<-length(jumptimes)
max<-ceiling(jumptimes[no])+1

plot(stepfun(c(jumptimes),c(0,output[,2]),right=FALSE),verticals=FALSE,pch=20,xlab="t",
    ylab=expression(L[t]),xlim=c(0,max),xaxs="i",main="") #plot jumpsizes delta_Lt
plot(stepfun(c(jumptimes),c(0,output[,3]),right=FALSE),verticals=FALSE,pch=20,xlab="t",
    ylab=expression(L[t]),xlim=c(0,max),xaxs="i",main="") #plot Lt
```

---

ecogarch_sim *Simulation of ECOGARCH(p,q) process*

Description

Generates a sample path of an ECOGARCH(p,q) process with a compound Poisson process as underlying background Levy process.

Usage

```
ecogarch_sim(t=0:10,a=-0.1,b=1,mu=-1,theta=-0.2,gamma=0.1,rate=1,mean=0,var=1)
```

Arguments

t	time
a	alpha coefficients
b	beta coefficients
mu	parameter for process M
theta	parameter for process M
gamma	parameter for process M
rate	parameter for compound Poisson process
mean	mean of the normal distribution
var	var of the normal distribution

Details

Generates a sample path of an ECOGARCH(p,q) process with a compound Poisson process as underlying background Levy process.

Value

Output: List with the folowing columns:

G	ECOGARCH(p,q) process, logprice process on a specified grid (prevTick applied)
GexpTime	logprice process on times including random jump times
logsigmaexpTime	log(sigma) process on random jumptimes
Ylimleft	log(sigma) process on times including random jump times without jump
Y	log(sigma) process on times including random jump times with jump
Tylimleft	ordered times including random jump times
Brownian	Brownian motion with specified mean and sqrt(timeintervals) as variance
Mt	Levy process Mt
Lt	compound Poisson Levy process with specified rate, normally distributed jumps with mean and variance as specified
TM	first column: times including random jump times, second column: Mt
sigmaJump	volatility process, i.e exp(Y)

Author(s)

Marlit Granzer

References

Master's thesis

Examples

```
t=0:100
a=c(-0.1,-0.3)
b=c(1,1)
mu=-4
theta=-0.2
gamma=0.1
rate=1
mean=0
var=1

output<-ecogarch_sim(t,a,b,mu,theta,gamma,rate,mean,var)
G<-unlist(output[[1]])
GexpTime<-unlist(output[[2]])
logsigmaexpTime<-unlist(output[[3]])
Ylimleft<-unlist(output[[4]])
Y<-unlist(output[[5]])
Tylimleft<-unlist(output[[6]])
Brownian<-unlist(output[[7]])
Mt<-unlist(output[[8]])
Lt<-unlist(output[[9]])
TM<-unlist(output[[10]])
sigmaJump<-unlist(output[[11]])

#plots
par(mfrow=c(5,1),mar=c(1.8,4.5,1,1))
plot(TM[,1],Lt,cex=0.01,ylab=expression(L[t]),xaxs="i",xlim=c(0,100))
plot(TM[,1],Mt,cex=0.01,ylab=expression(M[t]),xaxs="i",xlim=c(0,100))
plot(TM[,1],GexpTime,cex=0.01,ylab=expression(G[t]),xaxs="i",xlim=c(0,100))
plot(TM[,1],c(0,(GexpTime[2:length(GexpTime)]-GexpTime[1:(length(GexpTime)-1)])),
      type="l",ylab=expression(G[t]^(r)),xaxs="i",xlim=c(0,100))
plot(TM[,1],sigmaJump,cex=0.01,ylab=expression(sigma[t]^2),xaxs="i",xlim=c(0,100))
```

---

est_cogarch *Method of Moment Estimation*

Description

Function estimates the parameters beta, eta and phi of a continuous time GARCH(1,1) process

Usage

```
est_cogarch(G)
```

Arguments

G	Matrix with N sample paths of a COGARCH(1,1) process, see functions "Cogarch_cp_Npaths" or "Cogarch_vg_Npaths"

Details

Function estimates the parameters beta, eta and phi of a COGARCH(1,1) process, The input data G has to be EQUALLY spaced!

Value

Returns a matrix with the following rows:

betahat	estimates for beta
etahat	estimates for eta
phihat	estimates for phi
p	positive constants
k	positive constants

Author(s)

Marlit Granzer

References

Marlit Granzer, Master's thesis "Estimation of COGARCH models and implementation in R", 2013, Technische Universitaet Muenchen.

Examples

```
#parameters
beta<-0.04
eta<-0.053
phi<-0.038

t<-0:1000
N<-3
Lp="cp" #compound Poisson process, Lp="cp"
rate<-1
distribution<-"normal" #jumpsizes are normally distributed
mean<-0
var<-1

#simulation
output<-Cogarch_cp_Npaths(t,N,beta,eta,phi,Lp="cp",rate,distribution,mean,var) #compound poisson
#output G is already equally spaced as Cogarch_cp_Npaths includes the function prevTick()
G<-output$G

#estimate parameters for the cogarch processes
estimates<-est_cogarch(G)#needs equally spaced data as an input
betahat<-estimates[1,]
etahat<-estimates[2,]
phihat<-estimates[3,]
p<-estimates[4,]
k<-estimates[5,]

mb<-mean(betahat)
me<-mean(etahat)
mp<-mean(phihat)
ev<-c(mb,me,mp) #estimated values
ov<-c(beta,eta,phi) #original values
```

`est_cogarch_ret` *Method of moment estimation*

Description

Function estimates the parameters beta, eta and phi of a continuous time GARCH(1,1) process.

Usage

```
est_cogarch_ret(r)
```

Arguments

`r`	Returns of a COGARCH(1,1) process (equally spaced).

Details

Function estimates the parameters beta, eta and phi of a COGARCH(1,1) process, with the moment estimation method. The input data r has to be EQUALLY spaced! (If not already equally spaced you can use the "prevTick"" function to adjust it!)

Value

Returns a matrix with the following rows:

`betahat`	estimates for beta
`etahat`	estimates for eta
`phihat`	estimates for phi
`p`	positive constants
`k`	positive constants

Author(s)

Marlit Granzer

References

Marlit Granzer, Master's thesis "Estimation of COGARCH models and implementation in R", 2013, Technische Universitaet Muenchen.

Examples

```
#parameters
beta<-0.04
eta<-0.053
phi<-0.038

t<-0:1000
Lp="cp" #compound Poisson process, Lp="cp"
rate<-1
distribution<-"normal" #jumpsizes are normally distributed
mean<-0
```

```
var<-1
#simulated cogarch process
output_c<-cogarch_sim(t,beta,eta,phi,Lp="cp",rate,distribution,mean,var) #compound Poisson
G<-output_c[,1]
vol<-output_c[,2]
Lt<-output_c[,3]
delta_Lt<-output_c[,4]
rt<-output_c[,5]

#get equally spaced data
it<-rt
X<-G
pt<-prevTick(t,it,X)
G<-pt
n<-length(G)
r<-(G[2:n]-G[1:(n-1)])
nr<-length(r)
r<-(G[2:length(G)]-G[1:(length(G)-1)])

#estimate parameters
estimates<-est_cogarch_ret(r)
betahat<-estimates[1,]
etahat<-estimates[2,]
phihat<-estimates[3,]
p<-estimates[4,]
k<-estimates[5,]

ev<-c(betahat,etahat,phihat) #estimated values
ov<-c(beta,eta,phi) #original values
```

---

pml *Estimation with a pseudo-maximum likelihood method (PML)*

Description

Function estimates the parameters beta, eta and phi of a continuous time GARCH(1,1) process with equally or unequally spaced data. The t_i are assumed fixed (non-random) timepoints.

Usage

```
pml(x)
```

Arguments

x	input parameters beta, eta and phi.

Details

Function estimates the parameters beta, eta and phi of a COGARCH(1,1) process, The input data r can be unequally spaced! Pay attention to the assumption of fixed (non-random) timepoints. It the timepoints are random use function "pml_cp" for a compound Poisson process. Returns a Loglikelihood function.

Value

LL loglikelihood function

Author(s)

Marlit Granzer

References

Marlit Granzer, Master's thesis "Estimation of COGARCH models and implementation in R", 2013, Technische Universitaet Muenchen.

Examples

```
N=3
t<-0:500
beta=0.08
eta=0.06
phi=0.04

Lp="cp" #compound Poisson process, Lp="cp"
rate<-1
distribution<-"normal" #jumpsizes are normally distributed
mean<-0
var<-1

 output<-cogarch_sim(t,beta,eta,phi,Lp,rate,
                     distribution,mean,var,sigma,nu,theta,gs)
  G<-output[,1]
  vol<-output[,2]
  Lt<-output[,3]
  delta_Lt<-output[,4]
  rt<-output[,5]

#PML
  par<-c(beta,eta,phi)

  delta_t<-rt[2:length(rt)]-rt[1:(length(rt)-1)]
  Nt<-length(delta_t)
  n<-length(G)
  r<-(G[2:n]-G[1:(n-1)])
  nr<-length(r)

  A<-matrix(c(1,0,0,0,1,0,0,0,1,0,1,-1),ncol=3,byrow=TRUE)
  d<-c(0.000001,0.01,0.01,0.001)
  m<- constrOptim(par,pml, NULL , ui = A, ci = d)
  estimates_pml<-m$par
  betahat<-estimates_pml[1]
  etahat<-estimates_pml[2]
  phihat<-estimates_pml[3]

  estimates<-c(betahat,etahat,phihat)
```

pml_asym *Estimation with PML for asymmetric continuous time GJR GARCH(1,1) model*

Description

Function estimates the parameters beta, eta and phi of an asymmetric continuous time GJR GARCH(1,1) process with equally or unequally spaced data.

Usage

```
pml_asym(x)
```

Arguments

x	input parameters beta, eta, phi and gamma.

Details

Function estimates the parameters beta, eta and phi of an asymmetric continuous time GJR GARCH(1,1) process. The returns (input data) r can be unequally spaced! The output of this function is a log-likelihood function.

Value

LL	loglikelihood function

Author(s)

Marlit Granzer

References

Marlit Granzer, Master's thesis "Estimation of COGARCH models and implementation in R", 2013, Technische Universitaet Muenchen.

Examples

```
t<-0:1000
beta=1
eta=0.06
phi=0.0425
gamma=0.4
Lp="cp" #compound Poisson process, Lp="cp"
rate<-1
distribution<-"normal" #jumpsizes are normally distributed
mean<-0
var<-1
output_c<-cogarch_sim_assym(t,beta,eta,phi,gamma,Lp,rate,distribution,mean,var,sigma,nu,theta,gs)
G<-output_c[,1]
vol<-output_c[,2]
Lt<-output_c[,3]
delta_Lt<-output_c[,4]
```

```
rt<-output_c[,5]

N<-length(G)
r<-G[2:N]-G[1:(N-1)]
nr<-length(r)
delta_t<-rt[2:length(rt)]-rt[1:(length(rt)-1)] #for unequally spaced data
Nt<-length(delta_t)

par<-c(beta,eta,phi,gamma)
A<-matrix(c(1,0,0,0,0,1,0,0,0,0,1,0,0,1,-1,0,0,0,0,1,0,0,0,-1),ncol=4,byrow=TRUE)
d<-c(0.000001,0.01,0.01,0.001,0.01,-1)
m<- constrOptim(par, pml_asym, NULL , ui = A, ci = d)
estimates<-m$par
```

---

pml_cp *Estimation with PML for compound Poisson case*

Description

Function estimates the parameters beta, eta and phi of a COGARCH(1,1) process driven by a compound Poisson process with unequally spaced data, assumed all t_i are random but known.

Usage

```
pml_cp(x)
```

Arguments

x	input parameters beta, eta and phi.

Details

Function estimates the parameters beta, eta and phi of COGARCH(1,1) process driven by a compound Poisson process. The input data r can be unequally spaced! It is assumed that all t_i are random but known. The output of this function is a loglikelihood function.

Value

LL	loglikelihood function

Author(s)

Marlit Granzer

References

Marlit Granzer, Master's thesis "Estimation of COGARCH models and implementation in R", 2013, Technische Universitaet Muenchen.

Examples

```
beta=0.05
eta=0.06
phi=0.04

Lp="cp" #compound Poisson process, Lp="cp"
rate<-1
distribution<-"normal" #jumpsizes are normally distributed
mean<-0
var<-1
t=0:5000

output<-cogarch_sim(t,beta,eta,phi,Lp,rate,distribution,mean,var,sigma,nu,theta,gs)
G<-output[,1]
rt<-output[,5]
n<-length(G)
r<-(G[2:n]-G[1:(n-1)])
nr<-length(r)
delta_t<-rt[2:length(rt)]-rt[1:(length(rt)-1)]
Nt<-length(delta_t)

#calculate lambdahat
sum_delta_t<-sum(delta_t[1:length(delta_t)])
lambdahat<-length(delta_t)/sum_delta_t

par<-c(beta,eta,phi)
A<-matrix(c(1,0,0,0,1,0,0,0,1,0,1,-1),ncol=3,byrow=TRUE)
d<-c(0.000001,0.001,0.001,0.0001)
m<- constrOptim(par,pml_cp, NULL , ui = A, ci = d)
estimates_pml<-m$par
betahat<-estimates_pml[1]
etahat<-estimates_pml[2]
phihat<-estimates_pml[3]

#estimated values
estimates<-c(betahat,etahat,phihat,lambdahat)
```

`prevTick` *Previous Tick Interpolation*

Description

Transforms the time series X, observed at the irregular time points it, into a homogeneous time series Y at regularly time points t. uses previous tick interpolation -> i.e it takes the value of the maximal irregular time point <= regular time point t

Usage

```
prevTick(t, it, X)
```

Arguments

t	equally spaced time points
it	iregular spaced time points
X	time series observed at the irregular time points

Details

Transforms the time series X, observed at the irregular time points it, into a homogeneous time series Y at regularly time points t. uses previous tick interpolation -> i.e it takes the value of the maximal irregular time point <= regular time point t

Value

y	equally spaced data

Author(s)

Marlit Granzer

References

Marlit Granzer, Master's thesis "Estimation of COGARCH models and implementation in R", 2013, Technische Universitaet Muenchen.

Examples

```
t<-0:100 #fixed time grid

beta<-0.04
eta<-0.053
phi<-0.038
Lp="cp" #compound Poisson process
rate<-1
distribution<-"normal" #jumpsizes are normally distributed
mean<-0
var<-1
output<-cogarch_sim(t,beta,eta,phi,Lp="cp",rate,distribution,mean,var)

G<-output[,1]
vol<-output[,2]
Lt<-output[,3]
delta_Lt<-output[,4]
rt<-output[,5]

#use prevTick
it<-rt #random jumptimes
X<-G

pt<-prevTick(t,it,X)

G<-pt #process at times of the specified time grid t
n<-length(G)
r<-(G[2:n]-G[1:(n-1)]) #returns
nr<-length(r)
```

`sigmahat` *Estimation of the volatility process*

Description

Calculates sigmahat recursively.

Usage

```
sigmahat(r,betahat,etahat,phihat,gammahat,sym)
```

Arguments

`r`	returns of the COGARCH process
`betahat`	estimates for beta
`etahat`	estimates for eta
`phihat`	estimates for phi
`gammahat`	estimates for gamma
`sym`	TRUE for the symmetric COGARCH and FALSE for the asymmetric continuous time GJR GARCH

Details

This function calculates recursively the estimate of the volatility process. It is possible to choose if the estimated values of betahat, etahat and phihat come from a symmetric COGARCH (1,1) process or from an asymmetric GJR GARCH(1,1) process (additionally gammahat) by choosing either sym=TRUE or sym=FALSE.

Value

Returns

`sigma`	sigmahat of a continuous time GARCH(1,1) process either symmetric or asymmetric (GJR GARCH(1,1) process)

Author(s)

Marlit Granzer

References

Marlit Granzer, Master's thesis "Estimation of COGARCH models and implementation in R", 2013, Technische Universitaet Muenchen.

Examples

```
t<-0:1000
beta<-0.04
eta<-0.053
phi<-0.038
Lp="cp" #compound Poisson process
rate<-1
distribution<-"normal" #jumpsizes are normally distributed
mean<-0
var<-1

output_c<-cogarch_sim(t,beta,eta,phi,Lp="cp",rate,distribution,mean,var) #compound Poisson
G<-output_c[,1]
vol<-output_c[,2]
Lt<-output_c[,3]
delta_Lt<-output_c[,4]
rt<-output_c[,5]

r_diff<-diff(G)

x<-est_cogarch_ret(r_diff)
betahat<-x[1]
etahat<-x[2]
phihat<-x[3]

r<-r_diff
s<-sigmahat(r,betahat,etahat,phihat)

plot(rt,vol,xlim=c(0,1000),ylim=c(0,16), xaxs="i",yaxs="i",cex=0.1,
     xlab="t",ylab=expression(sigma[t]^2))
lines(s,col=4)
```

---

vargamma *Simulation of a Variance-Gamma Levy process*

Description

Simulation of a Variance-Gamma Levy process

Usage

```
vargamma(t=0:10,sigma=1,nu=0.05,theta=0.5,gs=0.01)
```

Arguments

sigma	parameter used to calculate the variance gamma process V<-theta*TG+sigma*W_TG
nu	parameter used for the gamma distribution
theta	parameter used to calculate the variance gamma process V<-theta*TG+sigma*W_TG
t	time vector, e.g. t<-0:100
gs	specifies size of the grid

Value

`ts`	timeintervals
`V`	Variance Gamma process

Author(s)

Marlit Granzer

References

Marlit Granzer, Master's thesis "Estimation of COGARCH models and implementation in R", 2013, Technische Universitaet Muenchen.

Examples

```
t<-0:10
sigma=1
nu=1
theta=0
gs=0.01

output<-vargamma(t,sigma,nu,theta,gs)
  t<-output[,1]
  V<-output[,2]

#plot
plot(t,V,cex=0.1,xlab="t",ylab=expression(V[t]),xlim=c(0,10),xaxs="i")
```

Index

List of R-Codes

List of Figures

List of Tables

Bibliography

Andersen, T. *Handbook of financial time series*. Mathematics and Statistics. Springer-Verlag Berlin Heidelberg, 2009.

Applebaum, D. *Lévy Processes and Stochastic Calculus*. Cambridge studies in advanced mathematics ; 116. Cambridge University Press, Cambridge, United Kingdom, 2nd edition, 2009.

Barndorff-Nielsen, O. and Shepard, N. Non-Gaussian Ornstein-Uhlenbeck-based models and some of their uses in financial economics. *Journal of the Royal Statistical Society*, 63(2):167–241, 2001.

Barndorff-Nielsen, O. and Shepard, N. Econometric analysis of realized volatility and its use in estimating stochastic volatility models. *Journal of the Royal Statistical Society*, 64(2):253–280, 2002.

Bollerslev, T. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31(3):307–327, 1986.

Brockwell, P. and Marquardt, T. Lévy process-driven and fractionally integrated ARMA process with continuous time parameter. *Statistica Sinica*, 15:477–494, 2005.

Brockwell, P., Chadraa, E., and Lindner, A. Continuous-Time GARCH Processes. *The annals of applied probability*, 16(2):790–826, 2006.

Chadraa, E. *Statistical modeling with COGARCH(p,q) processes* . Dissertation, Colorado State University, 2010.

Ding, Z., Granger, C., and Engle, R. A long memory property of stock returns and a new model. *Journal of Empirical Finance*, 1:83–106, 1993.

Drhuva, B. *An experimental study of high Reynolds number turbulence in the atmosphere*. Dissertation, Yale University, 2000.

Drost, F. C. and Werker, B. J. M. Closing the GARCH gap: continuous time GARCH modelling. *Journal of Econometrics*, 74(1):31–57, 1996.

Engle, R. Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. *Econometrica*, 50:987–1007, 1982.

Fasen, V., Klüppelberg, C., and Lindner, A. *Extremal behavior of stochastic volatility models*. Stochastic Finance. Springer, 2006.

Ferrazzano, V. Windspeed recording process and related issues, 2010.

Ferrazzano, V. and Fuchs, F. Noise recovery for LÃ©vy-driven CARMA processes and high-frequency behaviour of approximating Riemann sums. *Electronic Journal of Statistics*, 7:533–561, 2013.

Glosten, L., Jagannathan, R., and Runkle, D. On the relation between expected return on stocks. *Journal of Finance*, 48:1779–1801, 1993.

Haug, S. *Exponential COGARCH and other continuous time models with applications to high frequency data.* Dissertation, Technische Universität München, 2006.

Haug, S. and Czado, C. An exponential continuous time GARCH process. *Journal of Applied Probability*, 44(4):960–976, 2007.

Haug, S., Klüppelberg, C., Lindner, A., and Zapp, M. Estimating the cogarch(1,1) model - a first go. *The Econometrics Journal*, 10(2):320–341, 2007.

Iacus, S. M. *Simulation and Inference for Stochastic Differential Equations*. Springer Series in Statistics. Springer, New York, 2008.

Kloeden, P. and Platen, E. *Numerical Solution of Stochastic Differential Equations*. Stochastic Modelling and Applied Probability, Vol. 23. Springer, Berlin, 2011.

Klüppelberg, C., Lindner, A., and Maller, R. A continous time GARCH process driven by a Lévy process: stationarity and second order behaviour. *Journal of Applied Probability*, 41(3):601–622, 2004.

Klüppelberg, C., Lindner, A., and Maller, R. *Continuous time volatility modelling: COGARCH versus Ornstein-Uhlenbeck models*. From Stochastic Calculus to Mathematical Finance. The Shiryaev Festschrift. Springer, 2006.

Klüppelberg, C., Maller, R., and Szimayer, A. The COGARCH: A Review, with News on Option Pricing and Statistical Inference. In *Surveys in Stochastic Processes. Proc. of the 33rd SPA Conference in Berlin*, pages 29–58. EMS Series of Congress Reports, EMS Publishing House, 2011.

Kyprianou, A. *Introductory Lectures on Fluctuations of Lévy Processes with Applications*. Springer, Berlin, Heidelberg, 2006.

Lee, O. A continuous time asymmetric power GARCH process driven by a Lévy process. *Journal of the Korean Data and Information Science Society*, 21(6):1311–1317, 2010.

Lee, O. V-uniform ergodicity of a continuous time asymmetric power GARCH(1,1) model. *Statistics and Probability Letters*, 82:812–817, 2012.

Madan, D., Carr, P., and Chang, E. The Variance Gamma Process and Option Pricing. *European Finance Review*, 2:79–105, 1998.

Maller, R. and Szimayer, A. Finite approximation schemes for Lévy processes, and their application to optimal stopping problems. *Stochastic Processes and their Applications*, 117:1422–1447, 2007.

Maller, R., Müller, G., and Szimayer, A. GARCH modelling in continuous time for irregularly spaced time series data. *Bernoulli*, 14(2):519–542, 2008.

Mayr, K. *Der asymmetrische COGARCH: seine Definition, Approximation und Schätzung*. Masterarbeit, Technische Universität München, 2013.

McNeil, A. J., Frey, R., and Embrechts, P. *Quantitative Risk Management: Concepts, Techniques, and Tools*. Princeton Series in Finance. Princeton University Press, 2005.

Müller, G., Durand, R., Maller, R., and Klüppelberg, C. Analysis of stock market volatility by continuous-time garch models. In *Stock Market Volatility*, pages 31–50. Chapman Hall/Taylor and Francis, London, 2009.

Nelson, D. B. Arch models as diffusion approximations. *Journal of Econometrics*, 45(1-2): 7–38, 1990.

Nelson, D. B. Conditional heteroskedasticity in asset returns: a new approach. *Econometrica*, 59(2):347–370, 1991.

Protter, P. *Stochastic Integration and Differential Equations*. Number 21 in Stochastic Modelling and Applied Probability. Springer, 2nd edition, 2005.

Protter, P. and Talay, D. The Euler scheme for Lévy driven stochastic differential equations. *The Annals of Probability*, 25(1):393–423, 1997.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2012. URL `http://www.R-project.org/`. ISBN 3-900051-07-0.

Rabemananjara, R. and Zakoian, J. Threshold Arch Models and asymmetries in volatilityThreshold Arch Models and asymmetries in volatility. *Journal of Applied Econometrics*, 8(1):31–49, 1993.

Sato, K. *Lévy Processes and Infinitely Divisible Distributions*. Cambridge studies in advanced mathematics ; 68. Cambridge University Press, Cambridge, United Kingdom, 1999.

Schlosser, J. *Wissenschaftliche Arbeiten schreiben mit LaTeX*. Mitp Verlag, Bonn, 2006.

Schoutens, W. *Lévy Processes in Finance: Pricing Financial Derivatives*. John Wiley & Sohns Ltd, Chichester, 2003.

Ueltzhöfer, F. *On the estimation of jumps of continuous-time stochastic processes*. Dissertation, Technische Universität München, 2013.

Zakoian, J. Threshold heteroskedastic models. *Journal of Economic Dynamics and Control*, 18(5):931–955, 1994.

Zapp, M. *Schätzverfahren in einem neuen COGARCH-Modell*. Diplomarbeit, Technische Universität München, 2004.