ROBUST ESTIMATION OF AUTOREGRESSIVE CONDITIONAL DURATION MODELS

ROBUST ESTIMATION OF AUTOREGRESSIVE CONDITIONAL DURATION MODELS

By

Rola El Sebai, Honours B.Sc.

A Thesis

Submitted to the School of Graduate Studies

in Partial Fulfilment of the Requirements

for the Degree

Master of Science

MASTER OF SCIENCE (2012) McMaster University

(Statistics) Hamilton, Ontario

TITLE: Robust Estimation of Autoregressive Conditional Duration Models

AUTHOR: Rola El Sebai

SUPERVISOR: Dr. Bei Chen, Dr. Narayanaswamy Balakrishnan

NUMBER OF PAGES: xi, 91

Acknowledgements

I would like to take this opportunity to thank my thesis supervisor, Dr. Bei Chen, for her encouragement throughout my graduate study period. She has inspired me to love research and at the same time, to be enthusiastic about my research interests. Because of her, I was able to gain a thorough understanding of time series analysis. Also, I want to thank Dr. Narayanaswamy Balakrishnan for providing me with ideas for my research. I appreciate his support and constant motivation. In addition, I would like to express my deepest admiration to Dr. Roman Viveros-Aguilera for he is the reason why I chose to pursue a thesis in time series analysis after taking a time series course with him in my fourth year of my undergraduate studies. I truly thank him for being such an inspiration to me. Moreover, special thanks are extended to the members of the defense committee. I am also indebted to Dr. Manfred Kolster, Dr. Stanley Alama and Dr. Miroslav Lovric. I thank them for their support and for providing me with reference letters. Finally and most importantly, I thank my parents and brothers for their unconditional faith in me and for all the love they give me.

Contents

Abstract

In this thesis, we apply the Ordinary Least Squares (OLS) and the Generalized Least Squares (GLS) methods for the estimation of Autoregressive Conditional Duration (ACD) models, as opposed to the typical approach of using the Quasi Maximum Likelihood Estimation (QMLE). The advantages of OLS and GLS as the underlying methods of estimation lie in their theoretical ease and computational convenience. The latter property is crucial for high frequency trading, where a transaction decision needs to be made within a minute. We show that both OLS and GLS estimates are asymptotically consistent and normally distributed. The normal approximation does not seem to be satisfactory in small samples. We also apply Residual Bootstrap to construct the confidence intervals based on the OLS and GLS estimates. The properties of the proposed methods are illustrated with intensive numerical simulations as well as by a case study on the IBM transaction data.

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Financial Time Series

"A time series is a sequence of observations taken sequentially in time" – Box et al. (2008)

Time series are widely encountered in economics, business, ecology, astronomy and medicine. One of the major applications of time series analysis is in finance. Financial time series analysis is an area of research that examines both the theoretical and practical valuation methods of assets (Tsay, 2005). For example, one can perform analysis on a financial time series to examine how asset price varies over time and to forecast future values.

Typically, a financial time series is non-stationary, which makes it difficult to study mathematically. A simple approach is to transform the original series into *returns*. Given $\{y_1, \dots, y_n\}$, the return of an asset is calculated as

$$r_t = log\left(\frac{y_t}{y_{t-1}}\right), t = 1, \dots, n.$$

Tsay (2005) mentioned that returns are stationary and uncorrelated over a period of time. Also, returns act as a "scale-free summary of the investment opportunity"; thus, investors often favour returns in order to examine price changes (Tsay, 2005). Some key stylized facts returns of assets exhibit are:

1. Returns are uncorrelated;
2. Squared returns are correlated;
3. Volatility clustering;
4. Heavy tail-endedness.

To illustrate the characteristics of returns, let us consider the Yen/US dollar daily exchange rate data set, with a total of 2175 observations that range from March 28, 1998 until July 28, 2006. Figure 1.1 is a plot of observations of the Yen/US dollar daily exchange rate.

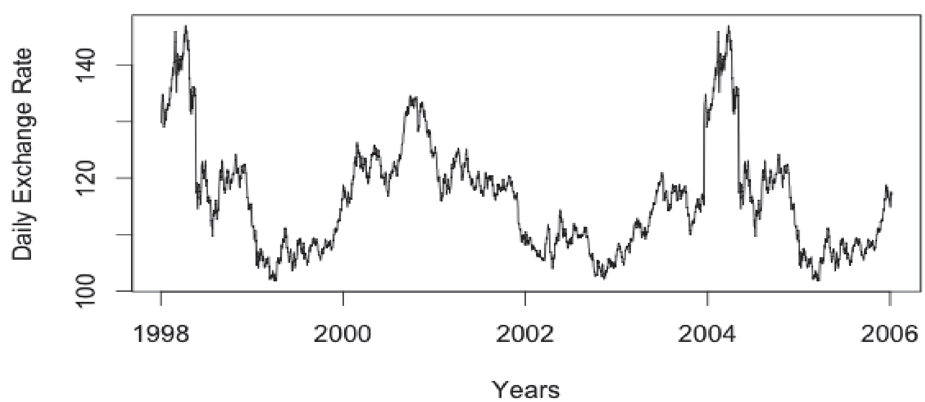


Figure 1.1: The Yen/US Dollar Daily Exchange Rate for March 28, 1998 until July 28, 2006.

Because the raw data clearly exhibits a non-stationary pattern, we transform the exchange rate into returns.

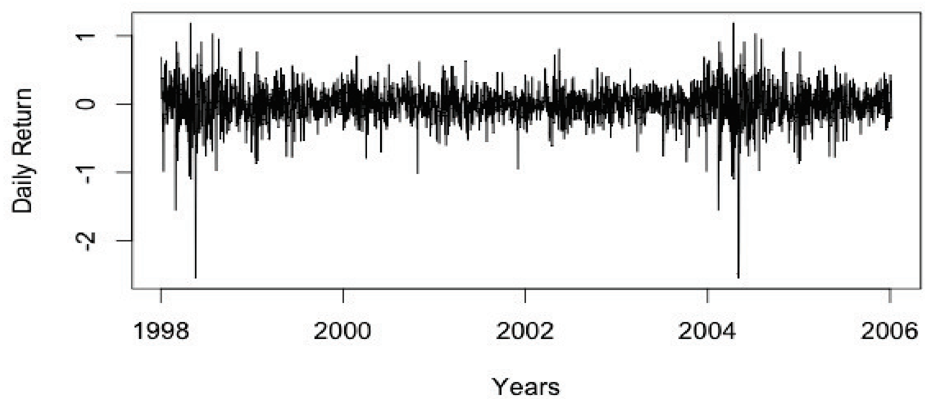


Figure 1.2: The Yen/US dollar Daily Return for March 28, 1998 until July 28, 2006.

The new series has a mean approximately equal to zero and is now stationary. Moreover, returns show a clear pattern of clustered volatilities. Also, as indicated below, an ACF plot of returns shows that they are uncorrelated, with lag 1 being significant, whereas an ACF plot of the squared returns shows that they are highly correlated. Squared returns usually follow a certain stochastic structure that is essential for forecasting.

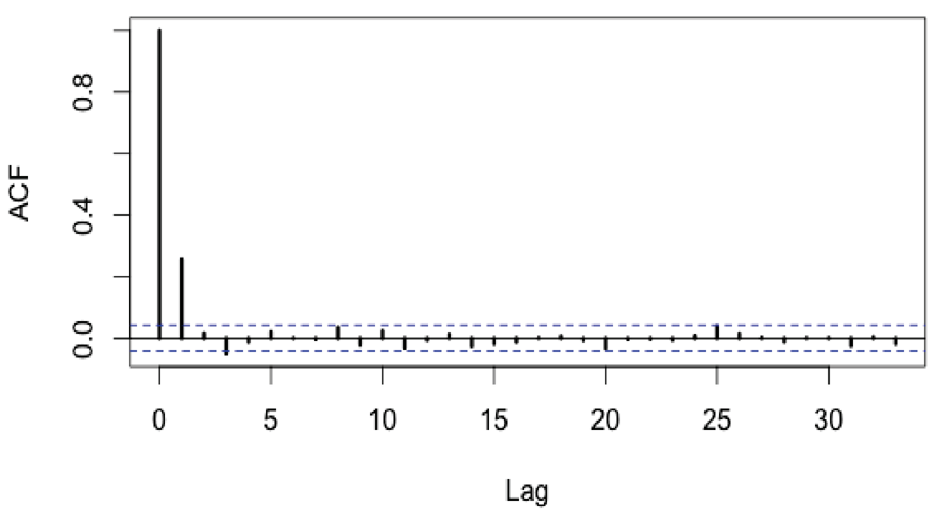


Figure 1.3: ACF Plot of the Yen/US Dollar Daily Return Process for March 28, 1998 until July 28, 2006.

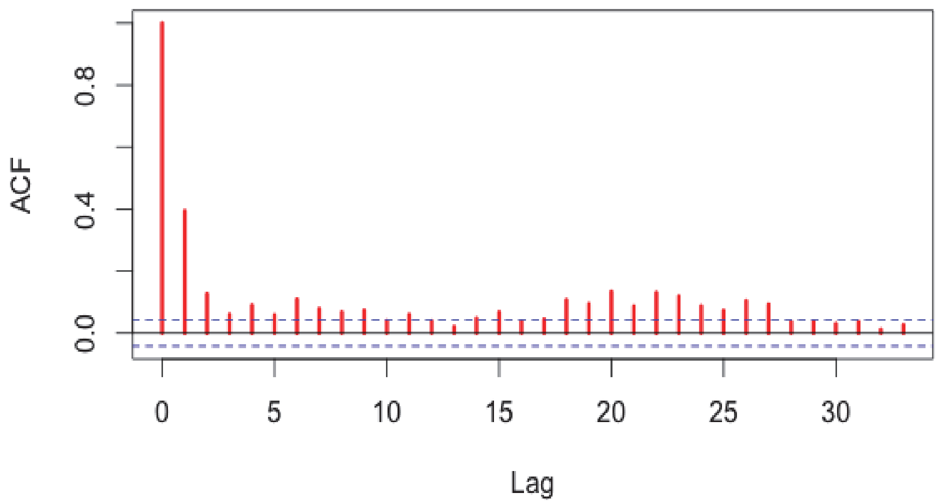


Figure 1.4: ACF Plot of the Yen/US Dollar Daily Squared Return Process for March 28, 1998 until July 28, 2006.

1.2 Volatility

Volatility is defined as a measurement of the variation of price over time that is utilized to give a description of the standard deviation of returns. It plays a key role in options trading, asset pricing and portfolio management due to the fact that it quantifies risk and hence is important in financial time series analysis. We next present an overview of the different perspectives to study volatility (Taylor, 2005).

1. **Realized Volatility**, also known as historical volatility, is an estimate of the return variation of some asset in the past. In other terms, it is the standard deviation of past returns, i.e.,

$$\sigma_t^R = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(r_{t-i}-\bar{r})^2},$$

where n is the number of trading periods and $\bar{r} = \frac{1}{n}\sum_{i=1}^{n} r_i$ represents the average of the returns. Realized volatility is a model-free volatility measure.

2. **Conditional Volatility** is the standard deviation of future returns which are conditional on past known information, i.e.,

$$\sigma_t = Var^{1/2}\,(r_t \mid r_{t-1}, \ldots, r_1).$$

   Typically, conditional volatility of financial returns does not remain constant over time, so-called *conditional heteroskedasticity*. Stochastic Volatility is used to model such conditional heteroskedasticity, e.g., the ARCH/GARCH model.

3. **Implied Volatility** is the market's evaluation of future volatility when option-pricing is based on some mathematical model of a financial market, i.e., Black-Scholes formula. Thus, it is a value implied by the market price that equals the volatility parameter when the equality of the price of an options market and the theoretical price holds true.

Modeling volatility is crucial in risk management as it provides a simple procedure of calculating the risk value of a financial position. Our main interest lies in stochastic models for modeling conditional volatility. In particular, there are two categories of stochastic volatility models: parameter-driven such as SV models, and observation-driven such as ARCH and GARCH models. Henceforth, we use the term "volatility" which stands for "conditional volatility".

1.2.1 SV Models

Stochastic volatility (SV) models, used for option pricing, are based on a specification of some stochastic process for volatility (Taylor, 2005). In other words, an innovation term to the conditional variance equation of $\{y_t\}_{t\in\mathbb{Z}}$ is proposed. There has been a fair amount of literature related to SV models. Clark (1973) investigated whether a relationship between price and volume affects the prediction errors in a market setting. Engle and Rosenberg (1994) proposed a new method of testing the structure of volatility based on a stochastic volatility process. Tauchen and Pitts (1983) examined the mixture of distribution hypothesis in order to model the joint distribution of asset returns and volumes that are conditional on some variable. In addition to having the ability of generalizing the results of an SV model to a multivariate case, Harvey and Shephard (1996) also proposed asymmetric SV models with leverage effect as a form of Euler approximation to the already used SV model. See Taylor (2005) for a detailed overview of SV models.

Suppose $\{y_t\}_{t\in\mathbb{Z}}$ follows a SV process

$$\begin{aligned} y_t &= \sigma_t \varepsilon_t, \\ (1-\alpha_1 B - \ldots - \alpha_m B^m)\log\left(\sigma_t^2\right) &= \alpha_0 + v_t, \end{aligned}$$

where α_0 is a constant, B is the backshift operator, $\varepsilon_t \sim N(0,1)$ and $v_t \sim N(0,\sigma_v^2)$. Note that $\{\varepsilon_t\}_{t\in\mathbb{Z}}$ and $\{v_t\}_{t\in\mathbb{Z}}$ are independent of each other and the roots of $1-\sum_{i=1}^{m}\alpha_i B^i$ lie outside the unit circle. SV models come up in the real world when pricing options in a setting where volatility constantly changes. Jacquier et al. (1994) performed a comparative study of the Quasi Maximum Likelihood Estimation (QMLE) and Monte Carlo methods of estimation for SV models. However, the process of estimating SV models is a relatively complicated procedure because there are two innovations $\{\varepsilon_t\}_{t\in\mathbb{Z}}$ and $\{v_t\}_{t\in\mathbb{Z}}$ for each of the y_t shocks.

1.2.2 ARCH and GARCH Models

One of the core techniques to model volatility is the Autoregressive Conditionally Heteroskedastic (ARCH) models which were introduced by Engle (1982). The GARCH (Generalized ARCH) models were later proposed by Bollerslev (1986), where he added a moving average term in order to take into account any available information on past volatilities. Engle and Bollerslev (1986) introduced the integrated GARCH (IGARCH) processes, which capture the long memory property of volatility. Engle and Ng (1993) derived the sign bias test, the negative sign bias test and the positive sign bias test of ARCH processes in order to test for asymmetry of the volatility of residuals. Conrad and Karanasos (2006) derived the impulse response function of the long memory GARCH (LMGARCH) processes, with the characteristic that their autocovariances decay slowly.

The essential idea of ARCH processes relates to the variance being conditional on past returns (Francq and Zakoian, 2010). The ARCH processes are immensely used in econometric and finance applications. They are particularly useful when the variance to be forecasted is known to change conditionally and could be predicted by previously forecasted errors (Engle, 1982). The key point of ARCH models is that the shock of an asset return, denoted by $\{y_t\}_{t\in\mathbb{Z}}$, "is serially uncorrelated but dependent, and the dependence of $\{y_t\}_{t\in\mathbb{Z}}$ can be described by a simple quadratic function of its lagged values" (Tsay, 2005). We start our discussion from a general class of GARCH (p,q) models and then consider ARCH (1) and GARCH (1,1) processes as examples.

Suppose that $\{y_t\}_{t\in\mathbb{Z}}$ follows a GARCH (p,q) process, with both p and $q \geq 0$,

$$y_t = \sigma_t \varepsilon_t, \tag{1.1}$$

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{p} \alpha_i y_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2 \tag{1.2}$$

for $\alpha_0,\ \alpha_i,\ \beta_j \geq 0$, $\varepsilon_t \sim i.i.d.(0,1)$ and $E\varepsilon_t^4 < \infty$. Also, $\{\sigma_t\}_{t\in\mathbb{Z}}$ is a stochastic process assumed to be independent of $\{\varepsilon_t\}_{t\in\mathbb{Z}}$. Assume that $\{y_t\}_{t\in\mathbb{Z}}$ is weakly stationary, i.e., for m = max (p,q),

$$\sum_{i=1}^{m} (\alpha_i + \beta_i) < 1 \tag{1.3}$$

is satisfied with $\alpha_i = 0$ for $i > p$ and $\beta_i = 0$ for $i > q$ (Tsay, 2005). When $q = 0$, $\{y_t\}_{t\in\mathbb{Z}}$ is referred to as an ARCH (p) process. From the structure of GARCH models, it is seen that large past shocks $\left\{y_{t-i}^2\right\}_{i=1}^{p}$ imply a large conditional variance $\left\{\sigma_t^2\right\}_{t\in\mathbb{Z}}$ for the innovation $\{\varepsilon_t\}_{t\in\mathbb{Z}}$. This consequently means that large shocks tend to be followed by large shocks.

The GARCH parameters are typically estimated by QMLE. In fact, the GARCH (p,q) process can be represented in a linear ARMA (p,q) form as

$$y_t^2 = \alpha_0 + \sum_{i=1}^{m} (\alpha_i + \beta_i) y_{t-i}^2 + v_t - \sum_{j=1}^{q} \beta_j v_{t-j}, \tag{1.4}$$

where $v_t = y_t^2 - \sigma_t^2$. Such ARMA representation allows us to apply the OLS and the GLS estimation for non-linear GARCH processes, which ultimately fasten the estimation procedure.

1.3 Ultra High Frequency (UHF) Data

GARCH processes themselves cannot directly model the intra-daily volatility for transaction data. Thus, in the context of high frequency trading, one is interested in additionally considering durations between transactions. Particularly, GARCH processes can be applied to intra-daily data after adjusting for the durations.

Ultra high frequency (UHF) data is a result of the advancements that technology has experienced and the development of computerized systems. Observations are recorded upon their arrivals based on some probability law together with characteristics such as prices, quotes, bid-ask spreads and volumes. The key characteristic that distinguishes UHF data is the fact that observations are spaced at irregular time intervals. High frequency indicates that prices are recorded more often than daily. Furthermore, the frequency of observations increases as more prices are recorded per day. The traditional research on time series employs regularly sampled data. For example, the most common frequency is one price every one or five minutes. In relation to finance, econometrics and time series analysis, UHF data has played a key role in providing a much richer understanding of market activity, both in the academic field and in a real trading setting (Glosten and Milgrom, 1985; Easley and O'Hara, 1992 and Copeland and Galai, 1983).

UHF data can be regarded as marked point processes, which exhibit the property of strong dependence. Engle and Russell (1998) based their research on introducing an autoregressive structure for such marked point processes. It is of importance to note that analyzing UHF data could potentially be the only method of observing temporal dependence between markets that have some "arbitrage inter-relationships" in common (Goodhart and O'Hara, 1997). In this thesis, we focus on one of the most widely used duration models, the standard Autoregressive Conditional Duration model (Engle and Russell, 1998), for modeling UHF data. Henceforth, we refer to standard ACD models as ACD models.

1.4 Autoregressive Conditional Duration (ACD) Models

Transactions data are often described by the time of a transaction and observed values, also known as marks of when transactions occur. When considering an IBM stock data set for example, the point of time can be referred to as the time of agreement for which a contract to trade some number of shares takes place. The times that elapse between events such as trades or price changes can be used to predict the times of future events and to explore the microstructure of markets. Trade duration is known to be the waiting time between two consecutive trades.

Let $\{t_0, t_1, \ldots, t_n\}$ be a sequence of arrival times with $0 \leq t_0 \leq t_1 \leq \ldots \leq t_n$. In order to study financial point processes, one can model the process of trade duration between two continuous points. Mathematically, trade duration, which is usually measured in seconds, can be represented by

$$X_i = t_i - t_{i-1}, \tag{1.5}$$

and is also referred to as the interval between two arrival times between events $i-1$ and i that occur at times t_{i-1} and t_i. Engle and Russell (1998) proposed a type of dependent Poisson process, known as the Autoregressive Conditional Duration (ACD) model with the objective of modeling time between events for heavily traded stocks. The ACD model provides a bridge between GARCH models and the UHF. The ACD (p,q) model uses a linear parameterization of ψ_t, also known as the expectation of the t^{th} duration

$$E\left(X_t \mid X_{t-1}, \ldots, X_1\right) = \psi_t(X_{t-1}, \ldots, X_1; \theta) \equiv \psi_t, \tag{1.6}$$

where ψ_t is dependent on p previous durations and q previous expected durations.

Suppose $\{X_t\}_{t\in\mathbb{Z}}$ follows an ACD (p,q) model

$$\begin{aligned} X_t &= \psi_t\varepsilon_t, && (1.7)\\ \psi_t &= \alpha_0 + \sum_{i=1}^{p}\alpha_i X_{t-i} + \sum_{i=1}^{q}\beta_i\psi_{t-i}, && (1.8)\end{aligned}$$

where $\alpha_0,\ \alpha_i,\ \beta_i \geq 0$, and ε_t are i.i.d. random variables with $E\varepsilon_t = 1$. A special case of ACD (p,q) processes is the ACD (p) process

$$\begin{aligned} X_t &= \psi_t\varepsilon_t, && (1.9)\\ \psi_t &= \alpha_0 + \sum_{i=1}^{p}\alpha_i X_{t-i}. && (1.10)\end{aligned}$$

As Engle and Russell (1998) and Engle (2000) indicate, the structure of ACD models is similar to that of GARCH processes, but rather than specifying a dynamic model on the conditional variance of the returns, a dynamic structure on the durations is introduced. In particular, the ACD (p) model is a counterpart of the ARCH (p) model in the duration model framework. Similar to GARCH models, letting $\eta_t = X_t - \psi_t$, the ACD (1,1) model can be represented as a linear ARMA (1,1) model

$$X_t = \alpha_0 + (\alpha_1 + \beta_1)X_{t-1} - \beta_1\eta_{t-1} + \eta_t, \quad (1.11)$$

where $t \in \mathbb{Z}$. Forecasts of waiting times can be computed directly from this representation using the conventional ARMA analytics. An inherited characteristic of ACD models from GARCH models is the capturing of duration clustering.

QMLE is a common method of parameter estimation for ACD (p,q) processes. However, QMLE is computationally expensive and may not be applicable in real time trading. In this thesis, we propose two alternative methods: OLS and GLS to estimate ACD processes. Ordinary Least Squares (OLS) is a method used to estimate the parameters of the ACD model and has the advantage of being numerically simple and computationally less expensive than QMLE. The proposed OLS method is of interest in practice because it provides initial estimators for the optimization procedure that is used on the QMLE method. However, OLS is not, in particular, an efficient estimator for ACD processes. We further propose to estimate the parameters of ACD models using Generalized Least Squares (GLS) to suppress this drawback. Our numerical experiments show that GLS outperforms OLS in finite samples, and performs equally well as QMLE while being significantly faster. Our theoretical studies show that both OLS and GLS estimates are asymptotically consistent and normally distributed. In addition, we apply the Residual Bootstrap method to construct confidence intervals based on the OLS and GLS estimates.

The remaining chapters of this thesis are organized as follows. The proposed OLS and GLS estimation methods of ACD models are investigated in Chapter 2. Chapter 3 introduces a Residual Bootstrap algorithm for the construction of the confidence intervals based on the OLS and GLS estimates. Chapter 4 presents a case study on IBM transaction data. Finally, we conclude the thesis with a summary and an outlook of future research in Chapter 5.

Chapter 2

Parameter Estimation of ACD Models

In this chapter, we propose to apply the Ordinary Least Squares (OLS) and the Generalized Least Squares (GLS) methods to estimate the parameters of ACD processes. We derive the theoretical results with the objective of proving that both OLS and GLS estimates are asymptotically consistent and normally distributed. In addition, the properties of the OLS and GLS estimates are further demonstrated by intensive Monte Carlo (MC) simulations.

The ACD model is a form of parameterization, which is defined on the basis of waiting times between two transactions. As introduced in Chapter 1, the ACD (p,q) model is defined as

$$\begin{aligned} X_t &= \psi_t \varepsilon_t, && (2.1) \\ \psi_t &= \alpha_0 + \sum_{i=1}^{p} \alpha_i X_{t-i} + \sum_{i=1}^{q} \beta_i \psi_{t-i}, && (2.2) \end{aligned}$$

where $\alpha_0,\ \alpha_i,\ \beta_i \geq 0$ and ε_t are i.i.d. random variables with $E\varepsilon_t = 1$. Let us define the probability density function of ε_t as $p\left(\epsilon, \phi\right)$ with a strictly positive domain and for parameters θ and ϕ as both being variation free. The baseline hazard can be defined as

$$\lambda_0 = \frac{p_0\left(\epsilon; \phi\right)}{S_0\left(\epsilon; \phi\right)}, \tag{2.3}$$

for $S_0(\epsilon;\phi) = \int_{\epsilon}^{\infty} p_0(u;\phi)\,du$ as the survivor function. The resulting intensity function of the ACD model is

$$\lambda\left(t \mid N(t), t_{i-1}, t_{i-2,}, \ldots, t_1, t_0\right) = \lambda_0\left(\frac{t - t_{N(t)-1}}{\psi_{N(t)}}\right)\frac{1}{\psi_{N(t)}}. \tag{2.4}$$

Notice that time change is dependent on ψ_t. Due to the various parameterizations of the conditional mean and distributions of ε_t, ACD models are known to be very flexible. Two common forms used for specifying (ϵ, ϕ) include the Exponential distribution and the Weibull distribution; the results are EACD (Exponential ACD) and WACD (Weibull ACD) models (Engle and Russell, 1998). Zhang et al. (2001) proposed ε_t to have a Gamma distribution; Grammig and Maurer (2000) based their analysis on the Burr distribution. For the case when the underlying distribution is Exponential, the baseline hazard is monotonic. When Weibull is the underlying distribution being considered, the baseline hazard follows an "inverted-U shape" (Engle and Russell, 1998).

2.1 Quasi Maximum Likelihood Estimation (QMLE)

The parameters of the ACD (p,q) model are often estimated by QMLE. Given a sample $\{X_1, \ldots, X_n\}$, in the case where ε_t follows an Exponential distribution, the log QML function is defined as

$$l(\theta) = -\sum_{t=1}^{n}\left(\frac{X_t}{\psi_t} + log\psi_t\right)$$

As noticed earlier, ACD processes have a form similar to that as GARCH processes. Results on QMLE for GARCH (1,1) processes can be immediately extended to EACD (1,1) (Lee and Hansen, 1994; Lumsdaine, 1996). According to Engle and Russell (1998), maximizing $l(\theta)$ of an ACD (p,q) process will result in both asymptotically consistent and normal estimates of θ along with a covariance matrix specified based on "robust standard errors" (Engle and Russell, 1998).

An important requirement for estimating ACD (p,q) processes using QMLE is for the validity of the conditional mean restrictions to hold true, i.e., the correct specification of the conditional mean function ψ_t needs to be satisfied. The Maximum Likelihood Estimation (MLE) procedure might be chosen over QMLE when the correct density function is specified, thus yielding a more efficient ML estimator. For a more general specification of the distribution of the error terms, the Weibull distribution can be chosen over the Exponential distribution. The resulting hazard function is either an increasing or a decreasing conditional intensity function. For instance, it is increasing when the shape parameter is greater than zero and is decreasing when the shape parameter is less than zero.

In summary, the distribution of an ACD model is specified directly conditional on past durations. However, for the case when the chosen model is misspecified, the estimator might be biased and inefficient. A disadvantage of QMLE relates to it being computationally expensive. In automated trading and market making settings where a decision is usually made within a minute or even seconds, a faster method of estimation is needed to account for such a limitation.

2.2 Ordinary Least Squares (OLS) Estimation

Based on the linear representation of ACD (p,q) processes, we propose to employ OLS to efficiently estimate the ACD parameters. First, let us consider a special case, an ACD (p) process

$$X_t = \psi_t \varepsilon_t, \tag{2.5}$$

$$\psi_t = \alpha_0 + \sum_{i=1}^{p} \alpha_i X_{t-i}, \tag{2.6}$$

Let $\theta_0 = (\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_p)'$ denote the vector of parameters of ACD (p). Let $X_t = 0$ and $\psi_t = 0$ for all $t \leq 0$ and let

$$Z_{t-1}' = (1,\, X_{t-1}, X_{t-2},\, \ldots,\, X_{t-p})\,. \tag{2.7}$$

Then, X_t satisfies

$$X_t = Z_{t-1}'\theta_0 + u_t, \tag{2.8}$$

where $u_t = X_t - \psi_t = (\varepsilon_t - 1)\,\psi_t$. Then, we obtain

$$Y = X\theta_0 + U, \tag{2.9}$$

where

$$X = \begin{pmatrix} 1 & X_0 & \dots & X_{1-p} \\ 1 & X_1 & \dots & X_{2-p} \\ \vdots & \vdots & \dots & \vdots \\ 1 & X_{n-1} & \dots & X_{n-p} \end{pmatrix} = \begin{pmatrix} Z_0' \\ Z_1' \\ \vdots \\ Z_{n-1}' \end{pmatrix} \tag{2.10}$$

Here, note that X is an $n \times (p+1)$ matrix. Moreover, both Y and U are $n \times 1$ vectors such that

$$Y = \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{pmatrix} \tag{2.11}$$

and

$$U = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix} \tag{2.12}$$

Assume that the inverse of the matrix $X^{'}X$ exists, the OLS estimator of θ_0 would satisfy:

$$\hat{\theta}_n = \arg\min_{\theta} \parallel Y - X\theta \parallel^2 = \left(X^{'}X\right)^{-1} X^{'}Y. \tag{2.13}$$

Next, we establish the asymptotic consistency and normality of the OLS estimator. We follow the framework of the proofs of Theorems 6.1 and 6.2 in Francq and Zakoian (2010). In order to establish asymptotic consistency of the parameter estimates, we assume

- **OLS assumption 1:** $\{X_t\}_{t\in\mathbb{Z}}$ is the non-anticipative strictly stationary solution of the ACD (p) model with $\alpha_0 > 0$.
- **OLS assumption 2:** $EX_t^2 < \infty$.
- **OLS assumption 3:** $P(\varepsilon_t = 1) \neq 1$. This enables the identification of the estimated parameters and to guarantee that the inverse of $X'X$ exists for sufficiently large n values and for $u_t \neq 0$.

Lemma 1. *(Consistency of the OLS Estimator of an ACD Model) Under the above presented assumptions, and for $\hat{\theta}_n$ being the sequence of estimators satisfying*

$$\hat{\theta}_n = \left(X^{'}X\right)^{-1} X^{'}Y, \tag{2.14}$$

then as $n \to \infty$,

$$\hat{\theta}_n \to \theta_0, \quad \hat{\psi}_n \to \psi_0, \ \ a.s. \tag{2.15}$$

The proof of Lemma 1 follows directly from Theorem 6.1 of Francq and Zakoian (2010).

In order to prove the asymptotic normality of the OLS estimator, we need to further make the following assumption

- **OLS assumption 4:** $EX_t^4 < \infty$,

and let **A** and **B** be $(p+1) \times (p+1)$ matrices defined as

$$\begin{aligned} \mathbf{A} &= E(Z_{t-1}Z_{t-1}') \\ \mathbf{B} &= E(\psi_t^2 Z_{t-1}Z_{t-1}'). \end{aligned}$$

Theorem 2. *(Asymptotic Normality of the OLS Estimator) Under the above stated OLS assumptions 1-4,*

$$\sqrt{n}(\hat{\theta}_n - \theta_0) \to N(0, (\kappa_n - 1)\,\mathbf{A}^{-1}\mathbf{B}\mathbf{A}^{-1}),$$

where $\kappa_n = E\varepsilon_t^2$.

Proof: Following the proof of Theorem 6.2 in Francq and Zakoian (2010), we consider the system of equations: $X_t = Z_{t-1}^{'}\theta_0 + u_t$.

Then the OLS estimator is

$$\begin{aligned}
\hat{\theta}_n &= \left(\frac{1}{n}\sum_{t=1}^{n} Z_{t-1}Z_{t-1}^{'}\right)^{-1}\left(\frac{1}{n}\sum_{t=1}^{n} Z_{t-1}X_t\right) \\
&= \left(\frac{1}{n}\sum_{t=1}^{n} Z_{t-1}Z_{t-1}^{'}\right)^{-1}\left(\frac{1}{n}\sum_{t=1}^{n} Z_{t-1}(Z_{t-1}^{'}\theta_0 + u_t)\right) \\
&= \left(\frac{1}{n}\sum_{t=1}^{n} Z_{t-1}Z_{t-1}^{'}\right)^{-1}\left(\frac{1}{n}\sum_{t=1}^{n} Z_{t-1}Z_{t-1}^{'}\theta_0\right) + \left(\frac{1}{n}\sum_{t=1}^{n} Z_{t-1}Z_{t-1}^{'}\right)^{-1}\left(\frac{1}{n}\sum_{t=1}^{n} Z_{t-1}u_t\right) \\
&= \left(\frac{1}{n}\sum_{t=1}^{n} Z_{t-1}Z_{t-1}^{'}\right)^{-1}\left(\frac{1}{n}\sum_{t=1}^{n} Z_{t-1}Z_{t-1}^{'}\right)\theta_0 + \left(\frac{1}{n}\sum_{t=1}^{n} Z_{t-1}Z_{t-1}^{'}\right)^{-1}\left(\frac{1}{n}\sum_{t=1}^{n} Z_{t-1}u_t\right) \\
&= I\theta_0 + \left(\frac{1}{n}\sum_{t=1}^{n} Z_{t-1}Z_{t-1}^{'}\right)^{-1}\left(\frac{1}{n}\sum_{t=1}^{n} Z_{t-1}u_t\right) \\
&= \theta_0 + \left(\frac{1}{n}\sum_{t=1}^{n} Z_{t-1}Z_{t-1}^{'}\right)^{-1}\left(\frac{1}{n}\sum_{t=1}^{n} Z_{t-1}u_t\right)
\end{aligned}$$

As a result,

$$\hat{\theta}_n - \theta_0 = \left(\frac{1}{n}\sum_{t=1}^{n} Z_{t-1}Z_{t-1}^{'}\right)^{-1}\left(\frac{1}{\sqrt{n}}\frac{1}{\sqrt{n}}\sum_{t=1}^{n} Z_{t-1}u_t\right)$$

and

$$\sqrt{n}(\hat{\theta}_n - \theta_0) \quad = \quad \left(\frac{1}{n} \sum_{t=1}^{n} Z_{t-1} Z_{t-1}^{'} \right)^{-1} \left(\frac{1}{\sqrt{n}} \sum_{t=1}^{n} Z_{t-1} u_t \right).$$

The variance of $\lambda^{'} Z_{t-1} u_t$ for $\lambda \in \mathbb{R}^{p+1}$ and $\lambda \neq 0$ is:

$$\begin{aligned}
Var(\lambda^{'} Z_{t-1} u_t) &= \lambda^{'} E\left(Z_{t-1} Z_{t-1}^{'} u_t^2 \right) \lambda \\
&= \lambda^{'} E\left(Z_{t-1} Z_{t-1}^{'} \left(\varepsilon_t - 1\right)^2 \psi_t^2 \right) \lambda \\
&= \lambda^{'} Z_{t-1} Z_{t-1}^{'} \psi_t^2 E\left(\varepsilon_t^2 - 2\varepsilon_t + 1 \right) \lambda \\
&= \left(\kappa_n - 1\right) \lambda^{\prime} \mathbf{B} \lambda
\end{aligned}$$

Therefore, it follows from the Central Limit Theorem that:

$$\frac{1}{\sqrt{n}} \sum_{t=1}^{n} \lambda^{'} Z_{t-1} u_t \rightarrow N(0, \left(\kappa_n - 1\right) \lambda^{\prime} \mathbf{B} \lambda). \tag{2.16}$$

The final result is then achieved by applying the Cramer-Wold device, which then implies:

$$\frac{1}{\sqrt{n}} \sum_{t=1}^{n} Z_{t-1} u_t \rightarrow N\left(0, \left(\kappa_n - 1\right) \mathbf{B}\right). \tag{2.17}$$

As a result,

$$\sqrt{n}(\hat{\theta}_n - \theta_0) \rightarrow N(0, \left(\kappa_n - 1\right) \mathbf{A}^{-1} \mathbf{B} \mathbf{A}^{-1}). \blacksquare \tag{2.18}$$

In practice, we may check OLS assumptions 1-4 by using Proposition 16 of Carrasco and Chen (2002) as re-stated below.

Proposition 3. *(Carrasco and Chen, 2002) For $s \geq 1$, let us assume that $\{\varepsilon_t\}_{t \in \mathbb{Z}}$ with domain*

$[0, \infty)$ *satisfies*

$$E \mid \alpha\varepsilon_t + \beta \mid^s < 1. \tag{2.19}$$

Result: $\{\psi_t\}_{t\in\mathbb{Z}}$ *is Markov ergodic. If the assumption that* $\{\psi_t\}_{t\in\mathbb{Z}}$ *is stationary is made, then* $\{\psi_t\}_{t\in\mathbb{Z}}$ *and* $\{X_t\}_{t\in\mathbb{Z}}$ *are strictly stationary. Also,* $E \mid \psi_t \mid^s < \infty$ *and* $E \mid X_t \mid^s < \infty$*. When* $s = 2$*, we have* $\{\psi_t\}_{t\in\mathbb{Z}}$ *as geometrically ergodic and* $E \mid \psi_t \mid^2 < \infty$*.*

Next, we present the OLS algorithm for the general ACD (p,q) process. A similar approach is proposed by Kavalieris et al. (2001) in an ARMA context.

Step 1: Fit an autoregression of order $\lfloor n^{1/2} \rfloor$ to the data. We then estimate the autoregressive (AR) parameters α_0, $\{\alpha_i\}_{i=1}^p$ and $\{\beta_i\}_{i=1}^q$ using OLS. Consequently, using the residuals obtained from the model, the estimate of ε is $\hat{\varepsilon}$.

Step 2: This stage is referred to as Innovations Substitution (IS) in Pukkila et al. (1990). Here, we regress $X_t - \hat{\varepsilon}_t$ on $X_{t-1}, \ldots, X_{t-p}, \hat{\varepsilon}_{t-1}, \ldots, \hat{\varepsilon}_{t-q}$, i.e.,

$$X_t - \hat{\varepsilon}_t = \alpha_0 + \alpha_1 X_{t-1} + \ldots + \alpha_p X_{t-p} + \beta_1 \hat{\varepsilon}_{t-1} + \ldots + \beta_q \hat{\varepsilon}_{t-1} \tag{2.20}$$

The parameter estimates $\{\hat{\alpha}_i\}_{i=1}^p$ and $\left\{\hat{\beta}_i\right\}_{i=1}^q$ are thus obtained using OLS.

Note: The theoretical justifications of the asymptotic consistency and normality of the OLS estimates for the parameters of the ACD (p,q) model are challenging and will be left for future research.

2.3 Generalized Least Squares (GLS) Estimation

The numerical procedure of estimating ACD (p,q) processes using OLS is simple, but the estimators are not efficient and the condition of $EX_t^4 < \infty$ has to hold. When considering a linear regression model, the GLS estimator is often more efficient than the OLS estimator. This occurs when the errors, which are conditional on the exogenous variables, are heteroskedastic. For the case of GLS, the moment condition required for the asymptotic normality of the latter estimator is $EX_t^2 < \infty$. Similar to the OLS setting, we start our discussion from ACD (p) processes.

First, define $\psi_t(\theta)$ and $\hat{\Omega}$ as

$$\psi_t(\theta) = \alpha_0 + \sum_{i=1}^{p} \alpha_i X_{t-i}, \tag{2.21}$$

$$\hat{\Omega} = diag\left(\hat{\psi}_1^{-2}\left(\hat{\theta}_n\right), \ldots, \hat{\psi}_n^{-2}\left(\hat{\theta}_n\right)\right). \tag{2.22}$$

Let $\kappa_n = E\varepsilon_t^2$. The GLS estimator is

$$\tilde{\theta}_n = \left(X'\hat{\Omega}X\right)^{-1} X'\hat{\Omega}Y. \tag{2.23}$$

Theorem 4. *(Asymptotic Properties of the GLS Estimator) For $\alpha_i > 0$ when $i = 1, ..., p$ and under OLS assumptions 1-3,*

$$\begin{aligned} \widetilde{\theta}_n &\rightarrow \theta_0, \quad a.s, && (2.24)\\ \sqrt{n}(\widetilde{\theta}_n - \theta_0) &\rightarrow N(0, (\kappa_n - 1)\boldsymbol{J}^{-1}) && (2.25) \end{aligned}$$

for $\mathbf{J} = E\left(\psi_t^{-2} Z_{t-1} Z_{t-1}'\right)$ being positive definite (follows Lemma 1.1 from Francq and Zakoian, 2010).

Proof: We follow the proof of Theorem 6.3 in Francq and Zakoian (2010).

$$\begin{aligned} \widetilde{\theta}_n &= \left(\frac{1}{n}\sum_{t=1}^{n}\psi_t^{-2}\left(\hat{\theta}_n\right) Z_{t-1}Z_{t-1}'\right)^{-1}\left(\frac{1}{n}\sum_{t=1}^{n}\psi_t^{-2}\left(\hat{\theta}_n\right) Z_{t-1}X_t\right)\\ &= \left(\frac{1}{n}\sum_{t=1}^{n}\psi_t^{-2}\left(\hat{\theta}_n\right) Z_{t-1}Z_{t-1}'\right)^{-1}\left(\frac{1}{n}\sum_{t=1}^{n}\psi_t^{-2}\left(\hat{\theta}_n\right) Z_{t-1}\left(Z_{t-1}'\theta_0 + u_t\right)\right)\\ &= \theta_0 + \left(\frac{1}{n}\sum_{t=1}^{n}\psi_t^{-2}\left(\hat{\theta}_n\right) Z_{t-1}Z_{t-1}'\right)^{-1}\left(\frac{1}{n}\sum_{t=1}^{n}\psi_t^{-2}\left(\hat{\theta}_n\right) Z_{t-1}u_t\right). \end{aligned}$$

For $\psi_t = \psi_t(\theta)$ and θ^* between $\hat{\theta}_n$ and θ_0, the Taylor series expansion around θ_0 gives

$$\psi_t^{-2}\left(\hat{\theta}_n\right) = \psi_t^{-2} - 2\psi_t^{-3}\left(\theta^*\right)\frac{\partial \psi_t}{\partial \theta'}\left(\theta^*\right)\left(\hat{\theta}_n - \theta_0\right).$$

In addition, since $\frac{\partial \psi_t}{\partial \theta}(\theta) = Z_{t-1}$ holds true for all θ, we have

$$\frac{1}{n}\sum_{t=1}^{n}\psi_t^{-2}\left(\hat{\theta}_n\right) Z_{t-1}Z_{t-1}' = \frac{1}{n}\sum_{t=1}^{n}\psi_t^{-2} Z_{t-1}Z_{t-1}' - \frac{2}{n}\sum_{t=1}^{n}\psi_t^{-3}\left(\theta^*\right) Z_{t-1}Z_{t-1}' \times Z_{t-1}'\left(\hat{\theta}_n - \theta_0\right).$$

Notice that due to the ergodicity theorem,

$$\frac{1}{n}\sum_{t=1}^{n}\psi_t^{-2} Z_{t-1}Z_{t-1}'$$

converges to $\mathbf{J} = E\left(\psi_t^{-2} Z_{t-1} Z'_{t-1}\right)$ a.s.

Also, when n is large enough,

$$\frac{2}{n}\sum_{t=1}^{n}\psi_t^{-3}\left(\theta^*\right) Z_{t-1} Z'_{t-1} \times Z'_{t-1}\left(\hat{\theta}_n - \theta_0\right)$$

converges to zero a.s. This is due to the OLS estimator being both consistent and bounded, i.e.,

$$\begin{aligned}\left\| \frac{1}{n}\sum_{t=1}^{n}\psi_t^{-3}\left(\theta^*\right) Z_{t-1} Z'_{t-1} \times Z'_{t-1}\left(\hat{\theta}_n - \theta_0\right)\right\| &\leq \left(\frac{1}{n}\sum_{t=1}^{n}\left\|\psi_t\left(\theta^*\right) Z_{t-1}\right\|^3\right)\left\|\left(\hat{\theta}_n - \theta_0\right)\right\| \\ &\leq K\left\|\hat{\theta}_n - \theta_0\right\|.\end{aligned}$$

Note that constant K is the result of $\hat{\theta}_n \to \theta_0$; hence it follows that

$$\psi_t^{-1}(\theta^*) X_{t-i} < \frac{1}{\theta_i^*}$$

for $i = 1, \ldots, p$ and finally $\| \psi_t^{-1}\left(\theta^*\right) Z_{t-1}\|$ is bounded. Thus, we have shown that

$$\left(\frac{1}{n}\sum_{t=1}^{n}\psi_t^{-2}\left(\hat{\theta}_n\right) Z_{t-1} Z'_{t-1}\right)^{-1} \to \mathbf{J}^{-1}$$

almost surely. Using the above arguments, the term $\frac{1}{n}\sum_{t=1}^{n}\psi_t^{-2}\left(\hat{\theta}_n\right) Z_{t-1} u_t$, which is equal to

$$\frac{1}{n}\sum_{t=1}^{n}\psi_t^{-2} Z_{t-1} u_t - \frac{2}{n}\sum_{t=1}^{n}\psi_t^{-3}\left(\theta^*\right) Z_{t-1} u_t \times Z'_{t-1}\left(\hat{\theta}_n - \theta_0\right)$$

converges to zero almost surely. Notice that the expectation of $\psi_t^{-2}\left(\hat{\theta}_n\right) Z_{t-1} u_t$ is zero and

$$\begin{aligned}\left\|\frac{2}{n}\sum_{t=1}^{n}\psi_t^{-3}\left(\theta^*\right) Z_{t-1} u_t \times Z'_{t-1}\left(\hat{\theta}_n - \theta_0\right)\right\| &= \left\|\frac{2}{n}\sum_{t=1}^{n}\psi_t^{-3}\left(\theta^*\right) Z_{t-1}\psi_t\left(\theta_0\right)\left(\varepsilon_t - 1\right) \times Z'_{t-1}\left(\hat{\theta}_n - \theta_0\right)\right\| \\ &\leq K\left(\frac{1}{n}\sum_{t=1}^{n}\mid \varepsilon_t - 1 \mid\right)\left\|\hat{\theta}_n - \theta_0\right\| \to 0.\end{aligned}$$

Therefore, for $R_n \to 0$ a.s,

$$\sqrt{n}\left(\tilde{\theta}-\theta_0\right) = \left(\mathbf{J}^{-1}+R_n\right)\left\{\frac{1}{\sqrt{n}}\sum_{t=1}^{n}\psi_t^{-2}Z_{t-1}u_t\right\} -\frac{2}{n}\left(\mathbf{J}^{-1}+R_n\right)\sum_{t=1}^{n}\psi_t^{-3}\left(\theta^*\right)Z_{t-1}u_t\times Z'_{t-1}\sqrt{n}\left(\hat{\theta}-\theta_0\right)$$

holds true. For θ^{**} between θ^* and θ_0, and using Taylor expansion around θ_0 would then imply that:

$$\psi_t^{-3}\left(\theta^*\right)=\psi_t^{-3}-3\psi_t^{-4}\left(\theta^{**}\right)Z'_{t-1}\left(\hat{\theta}_n-\theta_0\right).$$

As a result,

$$\begin{aligned}
&\sqrt{n}\left(\tilde{\theta}_n-\theta_0\right)\\
&=\left(\mathbf{J}^{-1}+R_n\right)\left\{\frac{1}{\sqrt{n}}\sum_{t=1}^{n}\psi_t^{-2}Z_{t-1}\left(\varepsilon_t-1\right)\right\}\\
&-\frac{2}{n}\left(\mathbf{J}^{-1}+R_n\right)\sum_{t=1}^{n}\psi_t^{-2}Z_{t-1}\left(\varepsilon_t-1\right)\times Z'_{t-1}\sqrt{n}\left(\hat{\theta}_n-\theta_0\right)\\
&+\frac{6}{n^{3/2}}\left(\mathbf{J}^{-1}+R_n\right)\sum_{t=1}^{n}\psi_t^{-4}\left(\theta^{**}\right)Z_{t-1}u_t\times\left\{Z'_{t-1}\sqrt{n}\left(\hat{\theta}_n-\theta_0\right)\right\}\times\left\{Z'_{t-1}\sqrt{n}\left(\theta^*-\theta_0\right)\right\}.\\
&:=\mathbf{A}+\mathbf{B}+\mathbf{C}
\end{aligned}$$

By applying the Central Limit Theorem to the ergodic and square integrable stationary martingale difference $\psi_t^{-1}Z_{t-1}\left(\varepsilon_t-1\right)$, it is seen that term $\mathbf{A}$ converges in distribution to a Gaussian vector with mean zero and a variance of:

$$\mathbf{J}^{-1}E\left\{\psi_t^{-2}\left(\varepsilon_t-1\right)^2Z_{t-1}Z'_{t-1}\right\}\mathbf{J}^{-1}=\left(\kappa_n-1\right)\mathbf{J}^{-1}.$$

As for term **B**, it converges to zero in probability by:

$$\frac{1}{n}\sum_{t=1}^{n}\psi_t^{-2}Z_{t-1}\left(\varepsilon_t-1\right)\times Z'_{t-1}\sqrt{n}\left(\hat{\theta}_n-\theta_0\right)$$
$$=\left\{\frac{1}{n}\sum_{t=1}^{n}\psi_t^{-2}Z_{t-1}\left(\varepsilon_t-1\right)\right\}\sqrt{n}\left(\hat{\omega}_n-\omega_0\right)+\sum_{j=1}^{q}\left\{\frac{1}{n}\sum_{t=1}^{n}\psi_t^{-2}Z_{t-1}\left(\varepsilon_t-1\right)X_{t-j}^2\right\}\sqrt{n}\left(\hat{\alpha}_{nj}-\alpha_{0j}\right).$$

Due to the ergodicity theorem, terms

$$\left\{\frac{1}{n}\sum_{t=1}^{n}\psi_t^{-2}Z_{t-1}\left(\varepsilon_t-1\right)\right\} \tag{2.26}$$

and

$$\left\{\frac{1}{n}\sum_{t=1}^{n}\psi_t^{-2}Z_{t-1}\left(\varepsilon_t-1\right)X_{t-j}^2\right\} \tag{2.27}$$

both converge to zero. Note that all three terms: $\sqrt{n}\left(\hat{\omega}_n-\omega_0\right)$, $\sqrt{n}\left(\hat{\alpha}_{nj}-\alpha_{0j}\right)$ and $(\mathbf{J}^{-1}+R_n)$ are bounded in probability. As a result, term **B** converges to zero in probability.

To take into account for the last term **C**, we can observe that:

$$\left\|C\right\|\leq\frac{K}{n^{1/2}}\left\|\mathbf{J}^{-1}+R_n\right\|\left\|\sqrt{n}\left(\hat{\theta}_n-\theta_0\right)\right\|^2\frac{1}{n}\sum_{t=1}^{n}\mid\varepsilon_t-1\mid\to 0. \tag{2.28}$$

So, the convergence in probability of term **C** holds true. Thus, $\sqrt{n}\left(\tilde{\theta}_n-\theta_0\right)$ is convergent. ■

Next, we discuss the GLS estimation procedure for the ACD (p,q) model. Steps 1 and 2 are identical to the method of estimation using OLS for ACD (p,q) processes presented previously. For the GLS estimates, we conduct an extra step in addition to the OLS procedure.

1. Step 3: From step 2 of estimating ACD (p,q) models using OLS, we perform a linear regression to re-estimate the parameters $\{\beta_i\}_{i=1}^q$ using GLS in order to achieve efficiency with the assumption that the errors are a result of the moving average (MA) of order q. Note that $\varepsilon_t-\hat{\varepsilon}_t$ are independent random variables.

Due to the restrictions of the parameters of ACD processes, i.e.: $\alpha_0 > 0$, $\{\alpha_i\}_{i=1}^{p} \geq 0$, and $\{\beta_i\}_{i=1}^{q} \geq 0$, if $\{\hat{\alpha}_i\}_{i=1}^{p} < 0$ and $\left\{\hat{\beta}_i\right\}_{i=1}^{q} < 0$, then we set $\{\hat{\alpha}_i\}_{i=1}^{p} = 0$ and $\left\{\hat{\beta}_i\right\}_{i=1}^{q} = 0$.

2.4 Numerical Results

In this section, we investigate the performance of our proposed OLS and GLS methods of estimation and we then compare them to QMLE for ACD (1), ACD (2) and ACD (1,1) processes using Monte Carlo (MC) simulations.

In particular, we examine the following 3 models:

Model 1: ACD (1)

$$X_t = \psi_t \varepsilon_t, \tag{2.29}$$

$$\psi_t = \alpha_0 + \alpha_1 X_{t-1}. \tag{2.30}$$

Model 2: ACD (2)

$$X_t = \psi_t \varepsilon_t, \tag{2.31}$$

$$\psi_t = \alpha_0 + \alpha_1 X_{t-1} + \alpha_2 X_{t-2}. \tag{2.32}$$

Model 3: ACD (1,1)

$$X_t = \psi_t \varepsilon_t, \tag{2.33}$$

$$\psi_t = \alpha_0 + \alpha_1 X_{t-1} + \beta_1 \psi_{t-1}, \tag{2.34}$$

for $\varepsilon_t \sim$ i.i.d. Exponential and Weibull. In all numerical examples, we use MC = 2000.

Recalling the probability density function of the Exponential distribution as:

$$f_X(x;\lambda) = \begin{cases} \lambda e^{-\lambda x}, & \text{for } x > 0 \\ 0, & \text{otherwise.} \end{cases}$$

In this study, the rate of the distribution that we used is $\lambda = 1$.

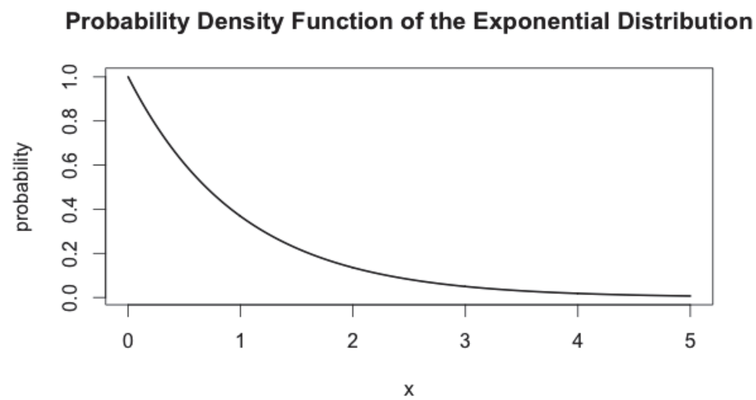


Figure 2.1: Probability Density Function of the Exponential Distribution with λ = 1.

Also, the probability density function of the Weibull distribution is:

$$f(x;\lambda,k) = \begin{cases} \frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1} e^{-\left(\frac{x}{\lambda}\right)^k}, & \text{for } x \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

where $k > 0$ is called the shape parameter of the distribution and $\lambda > 0$ is the scale parameter of the distribution.

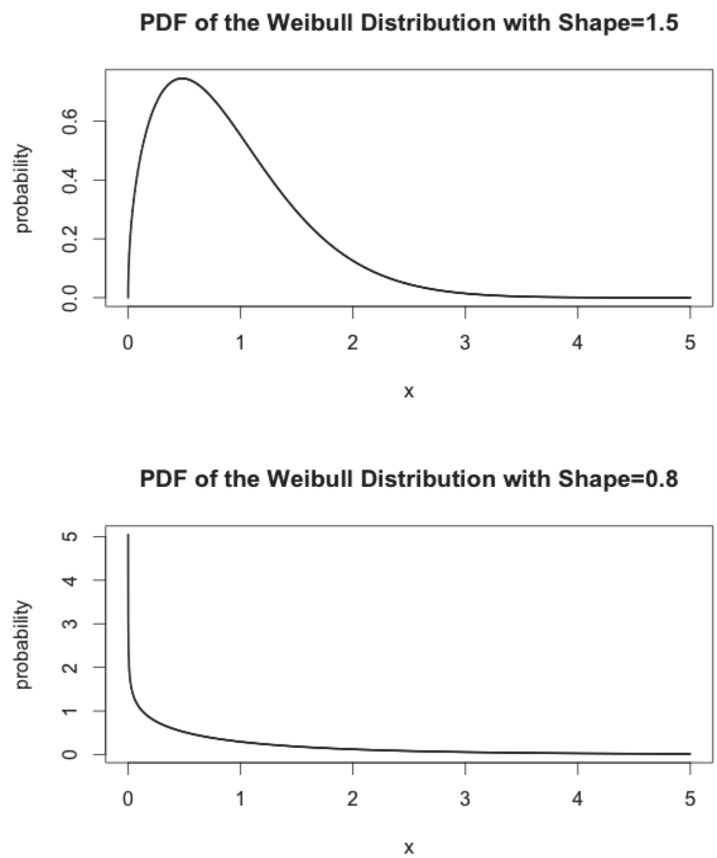


Figure 2.2: Probability Density Function of the Weibull Distribution with λ = 1, k = 1.5 and 0.8 respectively.

In the MC simulation study that we performed, we studied the estimates of the parameters of ACD processes; in particular, ACD (1), ACD (2) and ACD (1,1), based on the two proposed methods OLS and GLS and the already existing method of estimation QMLE. In particular, we examined the Exponential distribution and the Weibull distribution for sample size 500 and 5000 and 2000 MC replicates. Also, note that the nominal value of coverage is 95%. The analytics were performed using *Matlab* on a *MacBook* with a *2.4 GHz Intel Core 2 Duo Processor* and a *MacBook Pro* with a *2.4 GHz Intel Core i7*.

The following is a list of the parameters, distributions and related criteria used in this MC simulation study:

1. ACD (1), Exponential Distribution, $\lambda = 1$, N = 500 and 5000, α_0 = 1, α_1 = 0.4 and 0.1.

2. ACD (2), Exponential Distribution, $\lambda = 1$, N= 500 and 5000, α_0 = 1, (α_1, α_2) = (0.30, 0.15).

3. ACD (1), Exponential Distribution with Truncated Estimates, $\lambda = 1$, N = 500 and 5000, α_0 = 1, α_1 = 0.4 and 0.1.

4. ACD (2), Exponential Distribution with Truncated Estimates, $\lambda = 1$, N = 500 and 5000, α_0 = 1, (α_1, α_2) = (0.30, 0.15).

5. ACD (1), Weibull Distribution with Truncated Estimates, λ = 1, k = 0.8 and 1.5, N = 500 and 5000, α_0 = 1, α_1= 0.3.

6. ACD (1,1), Exponential Distribution, $\lambda = 1$, N = 500 and 5000 and α_0 = 0.2, α_1 = 0.4 and 0.1, β_1 = 0.3.

7. ACD (1,1), Exponential Distribution with Truncted Estimates, $\lambda = 1$, N = 500 and 5000 and α_0 = 0.2, α_1 = 0.4 and 0.1, β_1 = 0.3.

The Graphical Representations and Key Summary Statistics for the Estimates generated from the Monte Carlo Simulation

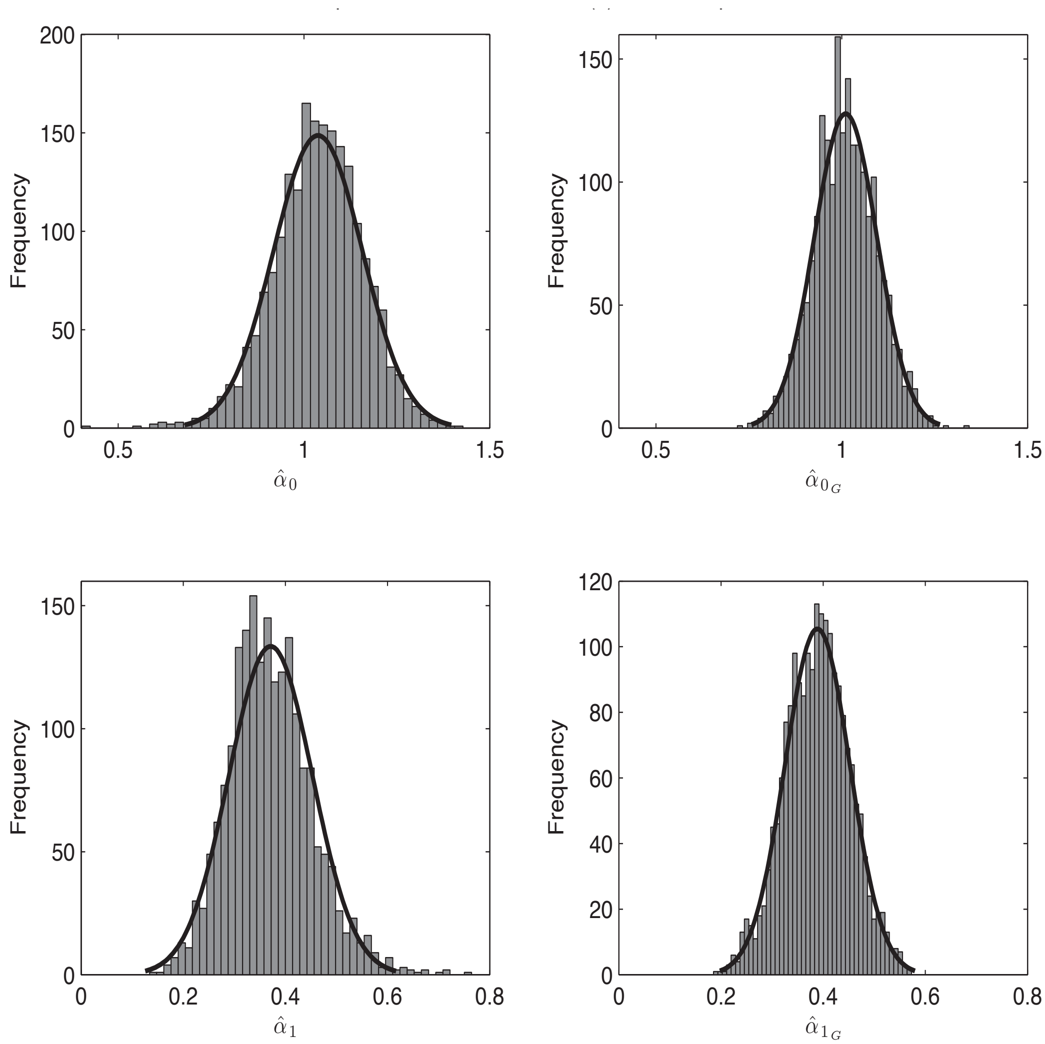


Figure 2.3: OLS and GLS Estimates of the Exponential Distribution for ACD (1) when $\lambda = 1$, N = 500 and for α_0 = 1, α_1 = 0.4.

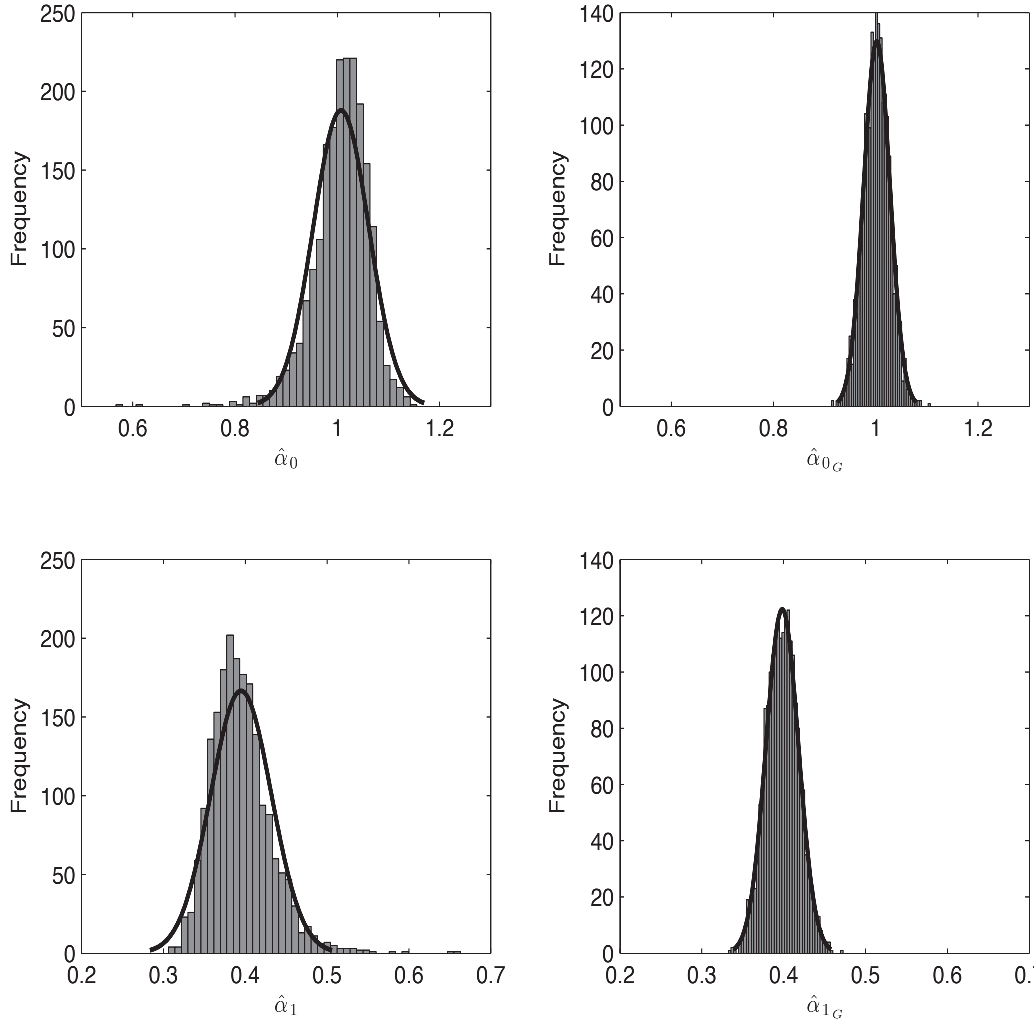


Figure 2.4: OLS and GLS Estimates of the Exponential Distribution for ACD (1) when $\lambda = 1$, N = 5000 and for α_0 = 1, α_1 = 0.4.

Figures 2.3 and 2.4 show the histograms of the estimates for the ACD (1) process for N = 500 and 5000 respectively and true values α_0 = 1 and α_1 = 0.4. It is apparent from the graphical representations that the estimates are consistent using both OLS and GLS because the mean is centered at the true value in both cases. However, it is noticable that the GLS estimates have a more apparent normal fit than the OLS estimates. Also, the spreads of the GLS estimates are significantly smaller than those of those OLS estimates.

We next show the parameter estimates based on QMLE method for the ACD (1) model with N = 500 and α_0 = 1, α_1 = 0.4.

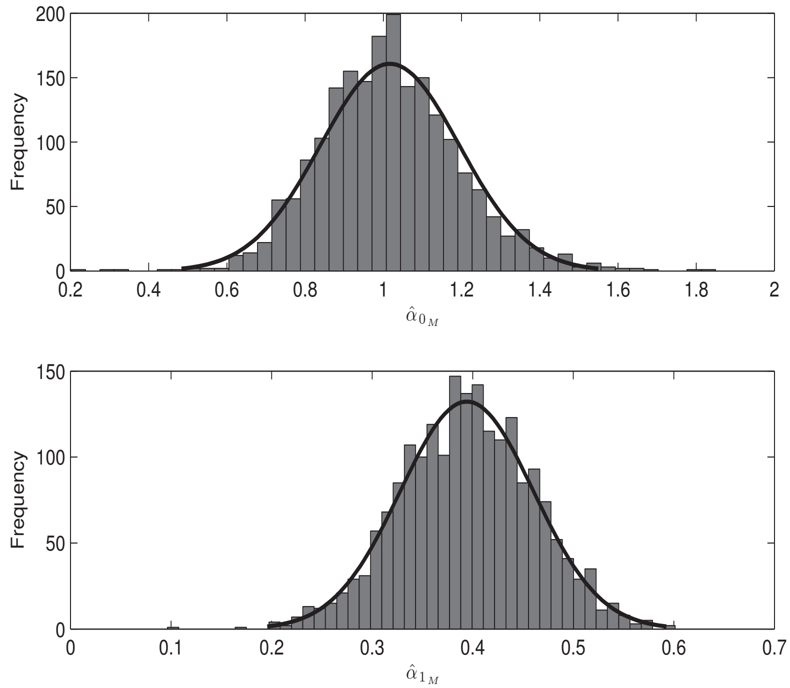


Figure 2.5: QMLE Estimates of the Exponential Distribution for ACD (1) when $\lambda = 1$, N = 500 and for $\alpha_1 = 0.4$.

Figures 2.3 and 2.5 show the histograms of the estimates of the ACD (1) process for N = 500 and true values α_0 = 1 and α_1 = 0.4. Comparing the OLS, GLS and QMLE estimates, it is apparent that the GLS estimates are similar to the OLS and QMLE estimates. However, in finite sample, QMLE outperforms GLS. The estimates of our proposed OLS method are still reliable and are considered to be consistent because it is the simplest of all three procedure.

Next, we show a few cases for comparing the estimates for ACD (2) and ACD (1,1) using OLS and GLS as the proposed methods of estimation.

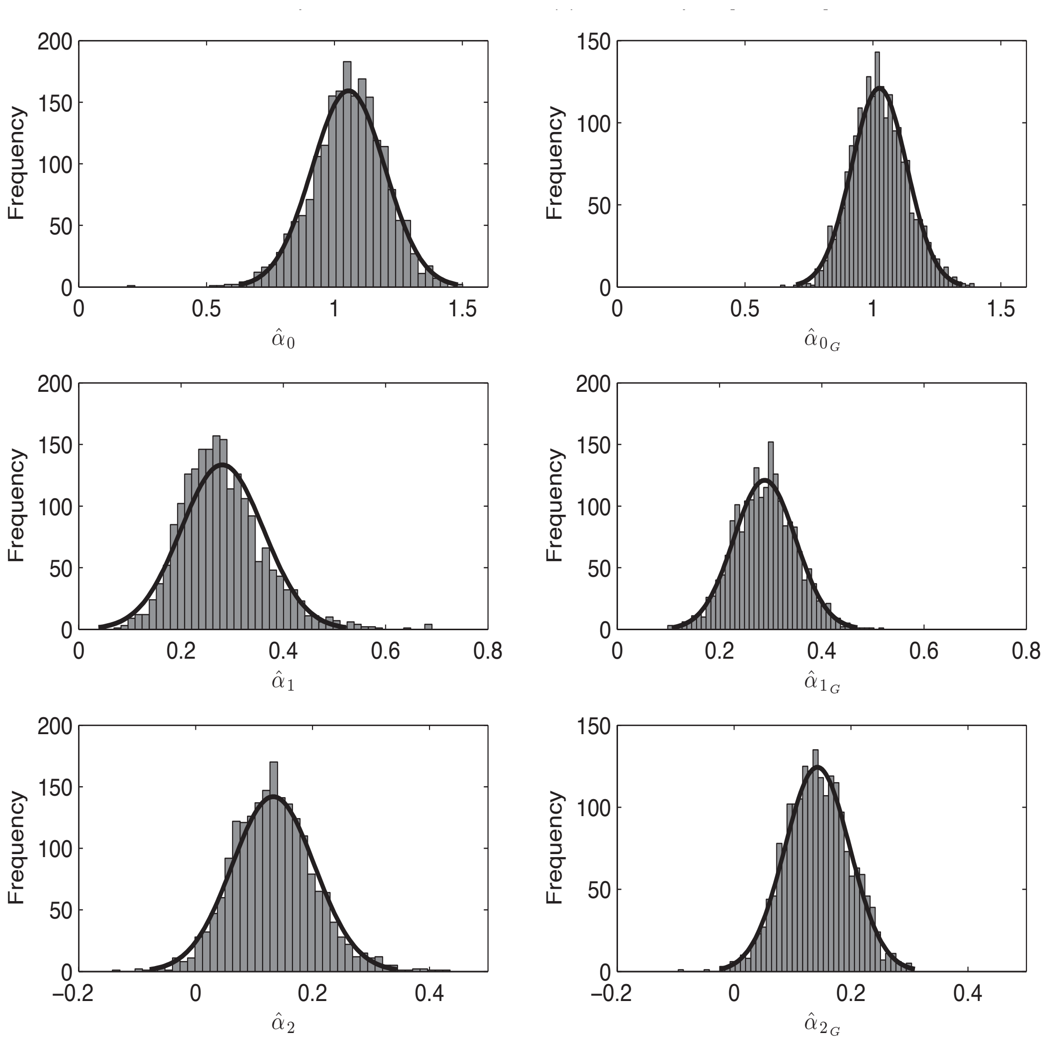


Figure 2.6: OLS and GLS Estimates of the Exponential Distribution for ACD (2) when $\lambda = 1$, N = 500 and α_0 = 1, (α_1, α_2) = (0.30, 0.15).

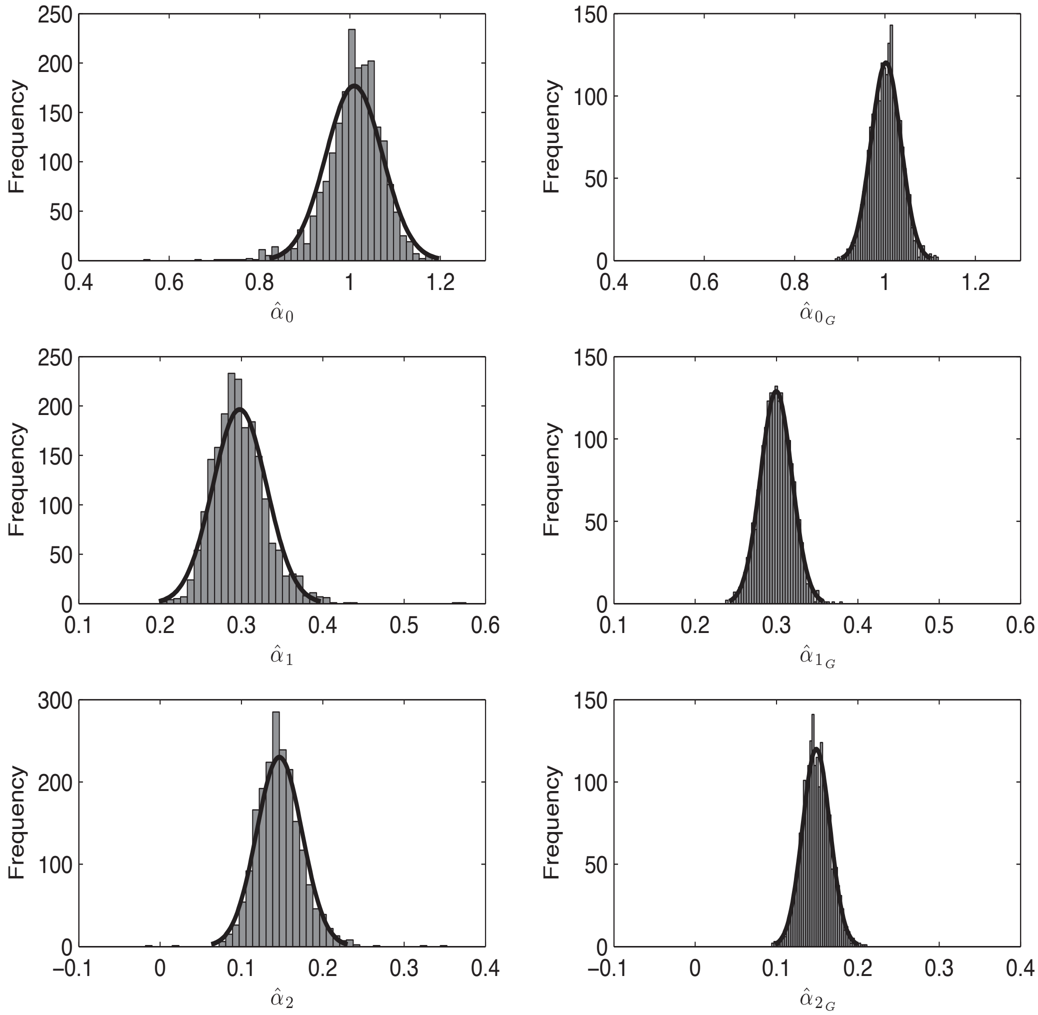


Figure 2.7: OLS and GLS Estimates of the Exponential Distribution for ACD (2) when $\lambda = 1$, N = 5000 and α_0 = 1, (α_1, α_2) = (0.30, 0.15).

Figures 2.6 and 2.7 show the OLS and GLS estimates of the Exponential Distribution for the ACD (2) model and N = 500 and 5000. The true values of the parameters are α_0 = 1, (α_1, α_2) = (0.30, 0.15). It is noticed that GLS outperforms OLS for both sample sizes. Also, the distribution of the GLS estimates seems to be more closer to normal than the distribution of the OLS estimates. All 12 histograms display the estimates $\hat{\alpha}_0$, $\hat{\alpha}_1$ and $\hat{\alpha}_2$ using OLS and GLS as being close to the true parameter values.

We next show the OLS and GLS estimates for ACD (1,1) when N = 5000 and α_0 = 0.2, α_1 = 0.1 and β_1 = 0.3. The parameter estimates appear to be consistent and are normally distributed.

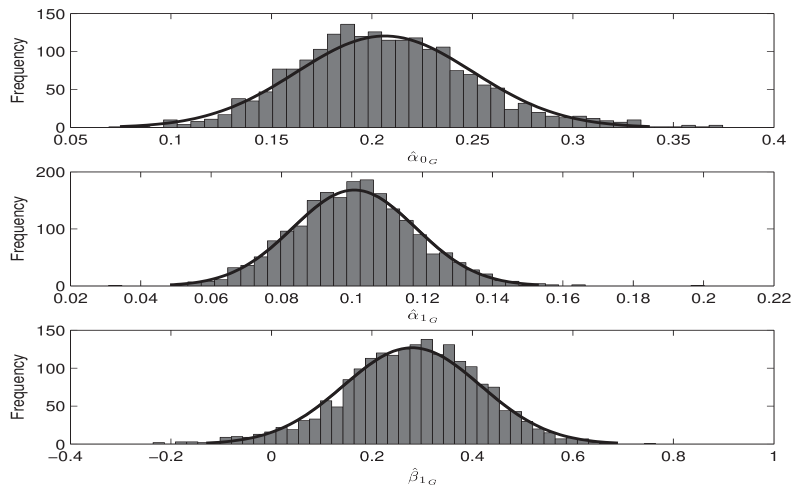


Figure 2.8: GLS Estimates of the Exponential Distribution with Truncated Estimates for ACD (1,1) when $\lambda = 1$, N = 5000 and for α_0 = 0.2, α_1 = 0.1 & β_1 = 0.3.

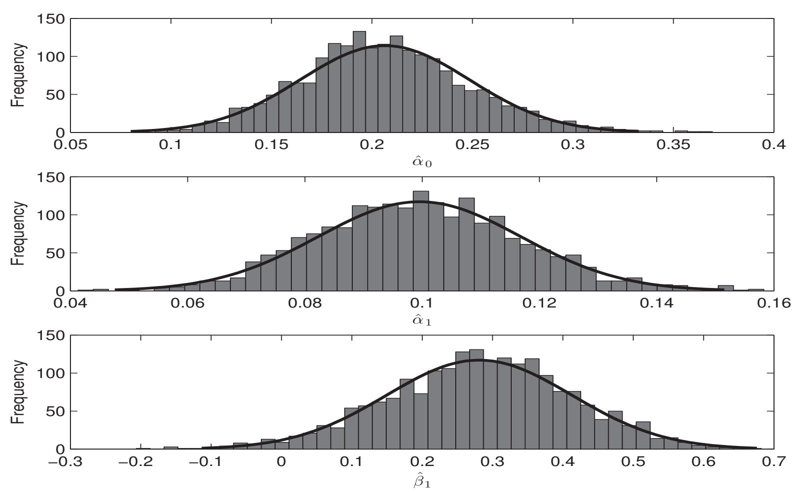


Figure 2.9: OLS Estimates of the Exponential Distribution with Truncated Estimates for ACD (1,1) when $\lambda = 1$, N = 5000 and for α_0 = 0.2, α_1 = 0.1 & β_1 = 0.3.

Table 2.1: The OLS and GLS Estimates of Various Monte Carlo Simulations for the Exponential Distribution with $\lambda = 1$ for ACD (1) and ACD (2) and α_0 = 1.

Simulation	**Mean**	**Variance**	**IQR**	**Coverage**
Exponential Distribution, ACD (1), N = 500, α_1 = 0.4				
$\hat{\alpha}_0$	1.0378	0.0143	0.1559	0.9410
$\hat{\alpha}_1$	0.3711	0.0067	0.1041	0.9410
$\hat{\alpha}_{0G}$	1.0105	0.0072	0.1141	0.9450
$\hat{\alpha}_{1G}$	0.3887	0.0041	0.0883	0.9430
Exponential Distribution, ACD (1), N = 500, α_1 = 0.1				
$\hat{\alpha}_0$	1.0032	0.0051	0.0924	0.9460
$\hat{\alpha}_1$	0.0950	0.0028	0.0686	0.9510
$\hat{\alpha}_{0G}$	1.0038	0.0049	0.0913	0.9450
$\hat{\alpha}_{1G}$	0.0945	0.0027	0.0696	0.9450
Exponential Distribution, ACD (1), N = 5000, α_1 = 0.4				
$\hat{\alpha}_0$	1.0072	0.0029	0.0641	0.9520
$\hat{\alpha}_1$	0.3948	0.0014	0.0445	0.9565
$\hat{\alpha}_{0G}$	1.0020	6.6420e-04	0.0349	0.9495
$\hat{\alpha}_{1G}$	0.3982	3.9458e-04	0.0274	0.9445
Exponential Distribution, ACD (1), N = 5000, α_1 = 0.1				
$\hat{\alpha}_0$	1.0011	4.9060e-04	0.0293	0.9465
$\hat{\alpha}_1$	0.0989	2.9003e-04	0.0229	0.9515
$\hat{\alpha}_{0G}$	1.0012	4.5374e-04	0.0284	0.9535
$\hat{\alpha}_{1G}$	0.0988	2.6137e-04	0.0220	0.9515
Exponential Distribution, ACD (2), N = 500, (α_1, α_2) = (0.30, 0.15)				
$\hat{\alpha}_0$	1.0550	0.0204	0.1827	0.9340
$\hat{\alpha}_1$	0.2806	0.0065	0.0991	0.9525
$\hat{\alpha}_2$	0.1328	0.0050	0.0926	0.9425
$\hat{\alpha}_{0G}$	1.0247	0.0118	0.1466	0.9425
$\hat{\alpha}_{1G}$	0.2883	0.0037	0.0823	0.9495
$\hat{\alpha}_{2G}$	0.1424	0.0031	0.0754	0.9480
Exponential Distribution, ACD (2), N = 5000, (α_1, α_2) = (0.30, 0.15)				
$\hat{\alpha}_0$	1.0095	0.0039	0.0704	0.9515
$\hat{\alpha}_1$	0.2978	0.0011	0.0409	0.9535
$\hat{\alpha}_2$	0.1466	7.8061e-04	0.0344	0.9510
$\hat{\alpha}_{0G}$	1.0022	0.0011	0.0451	0.9535
$\hat{\alpha}_{1G}$	0.2996	3.7281e-04	0.0265	0.9530
$\hat{\alpha}_{2G}$	0.1489	2.8718e-04	0.0227	0.9525

Table 2.2: The OLS and GLS Estimates of Various Monte Carlo Simulations for the Exponential Distribution with Truncated Estimates with $\lambda = 1$ for ACD (1) and ACD (2) and $\alpha_0 = 1$.

Simulation	**Mean**	**Variance**	**IQR**	**Coverage**
Exponential Distribution, ACD (1), N = 500, $\alpha_1 = 0.4$				
$\hat{\alpha}_0$	1.0378	0.0143	0.1559	0.9410
$\hat{\alpha}_1$	0.3711	0.0067	0.1041	0.9410
$\hat{\alpha}_{0_G}$	1.0105	0.0072	0.1141	0.9445
$\hat{\alpha}_{1_G}$	0.3887	0.0041	0.0883	0.9430
Exponential Distribution, ACD (1), N = 500, $\alpha_1 = 0.1$				
$\hat{\alpha}_0$	1.0049	0.0048	0.0926	0.9485
$\hat{\alpha}_1$	0.0946	0.0026	0.0712	0.9780
$\hat{\alpha}_{0_G}$	1.0063	0.0047	0.0945	0.9530
$\hat{\alpha}_{1_G}$	0.0940	0.0024	0.0679	0.9390
Exponential Distribution, ACD (1), N = 5000, $\alpha_1 = 0.4$				
$\hat{\alpha}_0$	1.0066	0.0028	0.0637	0.9535
$\hat{\alpha}_1$	0.3961	0.0013	0.0446	0.9550
$\hat{\alpha}_{0_G}$	1.0009	6.8511e-04	0.0357	0.9445
$\hat{\alpha}_{1_G}$	0.3996	3.9327e-04	0.0266	0.9480
Exponential Distribution, ACD (1), N = 5000, $\alpha_1 = 0.1$				
$\hat{\alpha}_0$	1.0001	4.9913e-04	0.0310	0.9560
$\hat{\alpha}_1$	0.1002	3.0203e-04	0.0232	0.9490
$\hat{\alpha}_{0_G}$	1.0004	4.6077e-04	0.0295	0.9555
$\hat{\alpha}_{1_G}$	0.1000	2.7207e-04	0.0224	0.9550
Exponential Distribution, ACD (2), N = 500, (α_1, α_2) = (0.30, 0.15)				
$\hat{\alpha}_0$	1.0544	0.0196	0.1790	0.9270
$\hat{\alpha}_1$	0.2806	0.0066	0.1040	0.9545
$\hat{\alpha}_2$	0.1321	0.0045	0.0853	0.9355
$\hat{\alpha}_{0_G}$	1.0213	0.0112	0.1412	0.9420
$\hat{\alpha}_{1_G}$	0.2894	0.0038	0.0860	0.9520
$\hat{\alpha}_{2_G}$	0.1421	0.0027	0.0703	0.9395
Exponential Distribution, ACD (2), N = 5000, (α_1, α_2) = (0.30, 0.15)				
$\hat{\alpha}_0$	1.0110	0.0039	0.0739	0.9520
$\hat{\alpha}_1$	0.2963	0.0011	0.0414	0.9545
$\hat{\alpha}_2$	0.1469	7.5499e-04	0.0336	0.9495
$\hat{\alpha}_{0_G}$	1.0034	0.0011	0.0470	0.9560
$\hat{\alpha}_{1_G}$	0.2983	3.8366e-04	0.0266	0.9465
$\hat{\alpha}_{2_G}$	0.1492	2.7685e-04	0.0232	0.9520

Table 2.3: The OLS and GLS Estimates of Various Monte Carlo Simulations for the Weibull Distribution with Truncated Estimates with Parameters [1,0.8] and [1,1.5] for ACD (1) and α_0 = 1.

Simulation	Mean	Variance	IQR	Coverage
Weibull Distribution [1,0.8], ACD (1), N = 500, α_1 = 0.3				
$\hat{\alpha}_0$	1.0312	0.0129	0.1482	0.9405
$\hat{\alpha}_1$	0.2717	0.0071	0.1076	0.9360
$\hat{\alpha}_{0_G}$	1.0099	0.0083	0.1192	0.9410
$\hat{\alpha}_{1_G}$	0.2872	0.0051	0.0983	0.9420
Weibull Distribution [1,0.8], ACD (1), N = 5000, α_1 = 0.3				
$\hat{\alpha}_0$	1.0058	0.0026	0.0592	0.9535
$\hat{\alpha}_1$	0.2951	0.0016	0.0469	0.9635
$\hat{\alpha}_{0_G}$	1.0021	7.5073e-04	0.0376	0.9520
$\hat{\alpha}_{1_G}$	0.2978	4.9201e-04	0.0301	0.9450
Weibull Distribution [1,1.5], ACD (1), N = 500, α_1 = 0.3				
$\hat{\alpha}_0$	1.0081	0.0056	0.1047	0.9495
$\hat{\alpha}_1$	0.2930	0.0030	0.0753	0.9500
$\hat{\alpha}_{0_G}$	1.0064	0.0046	0.0917	0.9525
$\hat{\alpha}_{1_G}$	0.2948	0.0025	0.0689	0.9510
Weibull Distribution [1,1.5], ACD (1), N = 5000, α_1 = 0.3				
$\hat{\alpha}_0$	1.0001	6.1678e-04	0.0335	0.9570
$\hat{\alpha}_1$	0.3000	3.4112e-04	0.0240	0.9485
$\hat{\alpha}_{0_G}$	1.0003	4.4929e-04	0.0287	0.9585
$\hat{\alpha}_{1_G}$	0.2999	2.4582e-04	0.0210	0.9510

Table 2.4: The OLS and GLS Estimates of Various Monte Carlo Simulations for the Exponential Distribution with $\lambda = 1$ for ACD (1,1) and $\alpha_0 = 0.2$.

Simulation	**Mean**	**Variance**	**IQR**	**Coverage**
Exponential Distribution, ACD (1,1), N = 500, $\alpha_1 = 0.4$, $\beta_1 = 0.3$				
$\hat{\alpha}_0$	0.2396	0.0056	0.0973	0.9155
$\hat{\alpha}_1$	0.3737	0.0089	0.1185	0.9555
$\hat{\beta}_1$	0.2621	0.0243	0.2089	0.9425
$\hat{\alpha}_{0G}$	0.2390	0.0062	0.0991	0.9245
$\hat{\alpha}_{1G}$	0.3753	0.0095	0.1166	0.9620
$\hat{\beta}_{1G}$	0.2603	0.0269	0.2141	0.9390
Exponential Distribution, ACD (1,1), N = 5000, $\alpha_1 = 0.4$, $\beta_1 = 0.3$				
$\hat{\alpha}_0$	0.2101	0.0011	0.0430	0.9470
$\hat{\alpha}_1$	0.3943	0.0026	0.0544	0.9625
$\hat{\beta}_1$	0.2898	0.0054	0.0878	0.9530
$\hat{\alpha}_{0G}$	0.2104	0.0011	0.0405	0.9395
$\hat{\alpha}_{1G}$	0.3911	0.0024	0.0532	0.9545
$\hat{\beta}_{1G}$	0.2920	0.0052	0.0870	0.9490
Exponential Distribution, ACD (1,1), N = 500, $\alpha_1 = 0.1$, $\beta_1 = 0.3$				
$\hat{\alpha}_0$	0.2483	0.0175	0.1738	0.9170
$\hat{\alpha}_1$	0.1000	0.0029	0.0746	0.9500
$\hat{\beta}_1$	0.1528	0.1617	0.5454	0.9180
$\hat{\alpha}_{0G}$	0.2553	0.0196	0.1774	0.9065
$\hat{\alpha}_{1G}$	0.0979	0.0029	0.0687	0.9495
$\hat{\beta}_{1G}$	0.1318	0.1777	0.5427	0.9085
Exponential Distribution, ACD (1,1), N = 5000, $\alpha_1 = 0.1$, $\beta_1 = 0.3$				
$\hat{\alpha}_0$	0.2064	0.0019	0.0579	0.9440
$\hat{\alpha}_1$	0.1007	3.0516e-04	0.0228	0.9490
$\hat{\beta}_1$	0.2800	0.0187	0.1784	0.9410
$\hat{\alpha}_{0G}$	0.2064	0.0018	0.0537	0.9460
$\hat{\alpha}_{1G}$	0.0995	3.0026e-04	0.0230	0.9495
$\hat{\beta}_{1G}$	0.2812	0.0173	0.1748	0.9475

Table 2.5: The OLS and GLS Estimates of Various Monte Carlo Simulations for the Exponential Distribution with Truncated Estimates with $\lambda = 1$ for ACD (1,1) and $\alpha_0 = 0.2$.

Simulation	**Mean**	**Variance**	**IQR**	**Coverage**
Exponential Distribution, ACD (1,1), N = 500, $\alpha_1 = 0.4$, $\beta_1 = 0.3$				
$\hat{\alpha}_0$	0.2431	0.0061	0.1028	0.9040
$\hat{\alpha}_1$	0.3655	0.0079	0.1121	0.9480
$\hat{\beta}_1$	0.2615	0.0226	0.2106	0.9110
$\hat{\alpha}_{0_G}$	0.2394	0.0060	0.1045	0.9160
$\hat{\alpha}_{1_G}$	0.3624	0.0081	0.1147	0.9410
$\hat{\beta}_{1_G}$	0.2713	0.0235	0.2342	0.9905
Exponential Distribution, ACD (1,1), N = 5000, $\alpha_1 = 0.4$, $\beta_1 = 0.3$				
$\hat{\alpha}_0$	0.2104	0.0011	0.0418	0.9410
$\hat{\alpha}_1$	0.3919	0.0022	0.0534	0.9600
$\hat{\beta}_1$	0.2915	0.0049	0.0855	0.9510
$\hat{\alpha}_{0_G}$	0.2099	0.0010	0.0405	0.9400
$\hat{\alpha}_{1_G}$	0.3934	0.0023	0.0548	0.9585
$\hat{\beta}_{1_G}$	0.2911	0.0049	0.0906	0.9480
Exponential Distribution, ACD (1,1), N = 500, $\alpha_1 = 0.1$, $\beta_1 = 0.3$				
$\hat{\alpha}_0$	0.2237	0.0085	0.1627	0.9945
$\hat{\alpha}_1$	0.0727	0.0036	0.1150	0.9890
$\hat{\beta}_1$	0.2541	0.0678	0.4580	0.9765
$\hat{\alpha}_{0_G}$	0.2227	0.0088	0.1669	0.9960
$\hat{\alpha}_{1_G}$	0.0707	0.0037	0.1164	0.9905
$\hat{\beta}_{1_G}$	0.2605	0.0689	0.4595	0.9770
Exponential Distribution, ACD (1,1), N = 5000, $\alpha_1 = 0.1$, $\beta_1 = 0.3$				
$\hat{\alpha}_0$	0.2064	0.0017	0.0553	0.9435
$\hat{\alpha}_1$	0.0986	3.5205e-04	0.0227	0.9605
$\hat{\beta}_1$	0.2820	0.0163	0.1744	0.9425
$\hat{\alpha}_{0_G}$	0.2051	0.0018	0.0547	0.9465
$\hat{\alpha}_{1_G}$	0.0984	3.6867e-04	0.0236	0.9615
$\hat{\beta}_{1_G}$	0.2864	0.0163	0.1693	0.9405

Table 2.6: Comparison of CPU Time for Various Monte Carlo Simulations for ACD Processes Using OLS, GLS and QMLE.

Simulation - using OLS	**CPU time (in seconds)**
Exponential Distribution, ACD (1), N = 500, α_1 = 0.4	0.0591
Exponential Distribution, ACD (1), N = 500, α_1 = 0.1	0.0582
Exponential Distribution, ACD (1), N = 5000, α_1 = 0.4	0.4994
Exponential Distribution, ACD (1), N = 5000, α_1 = 0.1	0.5052
Exponential Distribution*, ACD (1), N = 500, α_1 = 0.4	0.0608
Exponential Distribution*, ACD (1), N = 500, α_1 = 0.1	0.0721
Exponential Distribution*, ACD (1), N = 5000, α_1 = 0.4	0.5157
Exponential Distribution*, ACD (1), N = 5000, α_1 = 0.1	0.5025
Weibull Distribution* [1,0.8], ACD(1), N = 500, α_1 = 0.3	0.0762
Weibull Distribution* [1,0.8], ACD(1), N = 5000, α_1 = 0.3	0.5905
Weibull Distribution* [1,1.5], ACD(1), N = 500, α_1 = 0.3	0.0677
Weibull Distribution* [1,1.5], ACD(1), N = 5000, α_1 = 0.3	0.9760
Simulation - using GLS	**CPU time (in seconds)**
Exponential Distribution, ACD (1), N = 500, α_1 = 0.4	0.0901
Exponential Distribution, ACD (1), N = 500, α_1 = 0.1	0.0809
Exponential Distribution, ACD (1), N = 5000, α_1 = 0.4	0.8354
Exponential Distribution, ACD (1), N = 5000, α_1 = 0.1	0.8480
Exponential Distribution*, ACD (1), N = 500, α_1 = 0.4	0.0642
Exponential Distribution*, ACD (1), N = 500, α_1 = 0.1	0.0646
Exponential Distribution*, ACD (1), N = 5000, α_1 = 0.4	0.8277
Exponential Distribution*, ACD (1), N = 5000, α_1 = 0.1	0.8673
Weibull Distribution* [1,0.8], ACD (1), N = 500, α_1 = 0.3	0.0785
Weibull Distribution* [1,0.8], ACD (1), N = 5000, α_1 = 0.3	0.9140
Weibull Distribution* [1,1.5], ACD (1), N = 500, α_1 = 0.3	0.0884
Weibull Distribution* [1,1.5], ACD (1), N = 5000, α_1 = 0.3	1.0761
Simulation - using QMLE	**CPU time (in seconds)**
Exponential Distribution, ACD (1), N = 500, α_1 = 0.4	0.1399
Exponential Distribution, ACD (1), N = 5000, α_1 = 0.4	0.7981
Exponential Distribution, ACD (1), N = 500, α_1 = 0.1	0.1536
Exponential Distribution, ACD (1), N = 5000, α_1 = 0.1	1.0562
Weibull Distribution [1, 0.8], ACD (1), N = 500, α_1 = 0.1	0.2784
Weibull Distribution [1, 0.8], ACD (1), N = 5000, α_1 = 0.1	2.1074
Weibull Distribution [1, 1.5], ACD (1), N = 500, α_1 = 0.1	0.3492
Weibull Distribution [1, 1.5], ACD (1), N = 5000, α_1 = 0.1	2.3829

Note: * referes to the estimates being truncated. In all cases, $\alpha_0 = 1$.

2.4.1 Key Remarks

Our numerical findings and the comparative study of the CPU time for various MC simulations suggest that the estimation of ACD processes using both of our proposed methods OLS and GLS are satisfactory. In particular, we considered summary statistics including mean, variance, interquartile range (IQR) and coverage.

For example, from Table 2.1, for the case of an Exponential distribution for ACD (1) with parameters α_0 = 1, α_1 = 0.1 and N = 5000, the mean of $\hat{\alpha}_0$ by OLS is 1.0011 whereas the mean of $\hat{\alpha}_0$ by GLS is 1.0012. The mean of $\hat{\alpha}_1$ by OLS is 0.0989 whereas the mean of $\hat{\alpha}_1$ by GLS is 0.0988. The variances of the OLS estimates $\hat{\alpha}_0$ and $\hat{\alpha}_1$ are 4.9060e-04 and 2.9003e-04 respectively. The variances of the GLS estimates $\hat{\alpha}_0$ and $\hat{\alpha}_1$ are 4.5374e-04 and 2.6137e-04 respectively. Both means of the OLS and GLS estimates $\hat{\alpha}_0$ and $\hat{\alpha}_1$ are close to the true values of 1 and 0.1 respectively. Thus, both estimates are asymptotically consistent. However, the GLS estimates are less biased for finite samples because the mean of the GLS estimates is closer to the true value. The coverage of the OLS estimates for α_0 and α_1 are 0.9465 and 0.9515 and the coverage of the GLS estimates for α_0 and α_1 are 0.9535 and 0.9515. This suggests that GLS estimates are slightly more normally fit when compared to the OLS estimates at the 95% nominal level. Comparing the CPU time, we see that it takes 0.5052 seconds to run 1 MC replicate using OLS and 0.8480 seconds to run 1 MC replicate using GLS. Even though the OLS and GLS estimates are both asypmtotically consistent and normal, it is clear that OLS is much more efficient in terms of timing.

We consider another case from Table 2.2: Exponential distribution with truncated estimates for ACD (2) with parameters α_0 = 1 and (α_1, α_2) = (0.30, 0.15) and N = 500. The mean of $\hat{\alpha}_0$ by OLS is 1.0544 whereas the mean of $\hat{\alpha}_0$ by GLS is 1.0213. The mean of $\hat{\alpha}_1$ by OLS is 0.2806 whereas the mean of $\hat{\alpha}_1$ by GLS is 0.2894. The mean of $\hat{\alpha}_2$ by OLS is 0.1321 whereas the mean of $\hat{\alpha}_2$ by GLS is 0.1421. The variances of the OLS estimates $\hat{\alpha}_0$, $\hat{\alpha}_1$ and $\hat{\alpha}_2$ are 0.0196, 0.0066 and 0.0045 respectively. The variances of the GLS estimates $\hat{\alpha}_0$, $\hat{\alpha}_1$ and $\hat{\alpha}_2$ are 0.0112, 0.0038 and 0.0027 respectively. Evidently, the GLS and OLS estimates are asymptotically consistent. Also, as for the above case, the GLS estimates are less biased for finite samples because the mean of the GLS estimates is closer to the true value. The coverage of the OLS estimates for α_0, α_1 and α_2 are 0.9270, 0.9545 and 0.9355. The coverage of the GLS estimates for α_0, α_1 and α_2 are 0.9420, 0.9520 and 0.9395. So, the GLS estimates are significantly more normal than the OLS estimates at the 95% nominal level. Therefore, in this case, both OLS and GLS estimates are consistent and normally distributed.

Chapter 3

The Bootstrap Method

3.1 Introduction

Our goal is to construct confidence intervals based on the OLS and GLS estimates of ACD (p,q) processes. One possible approach is to utilize the asymptotic properties of the estimates, i.e., asymptotic normality. However, a drawback of this method is when sample size is small, the error due to the asymptotic distribution approximation is fairly large. Thus, an alternative approach should be applied to account for the limitation of not being able to analyze the accuracy of the estimators.

One of the core techniques which would serve as such a tool is the Bootstrap method. In principle, the idea is to find the empirical distribution of the estimator. Bootstrap is a non-parametric alternative to assess the reliability of the estimators. The Bootstrap method re-samples from an original sample made of i.i.d. observations with unknown probability distribution functions. The resulting samples generated, or the re-samples, are referred to as the bootstrap estimates. When the underlying data is strictly stationary, two bootstrap procedures could be used to perform the re-sampling.

The original boostrap procedure proposed by Efron (1979), Residual Bootstrap, regards the residuals of a regression model as the main focus. Efron's Residual Bootstrap procedure re-samples the residuals instead of the original observations of the data set. Because of the heteroskedasticity of the observations, there is some uncertainty about the performance of such a procedure. As a result, the fitted residuals will not behave as i.i.d. observations anymore; this in turn makes such a procedure not applicable to time series analysis. Instead, there will be some form of heteroskedasticity associated with the observations. Further research performed by Freedman (1981), Liu (1988) and Liu and Singh (1992) suggests that Efron's procedure is valid even if it was the case for data which are not independently distributed but just identically distributed.

The second procedure, called Block Bootstrap, is targeted for when less restrictions are present. The idea behind such a method is to divide the data into various blocks. The resulting blocks, which are also adjacent, are then re-sampled. This is particularly useful when there is an interest in capturing the dependence in "consecutive observations" (Koster, 1999). Carlstein (1986) suggested the concept of non-overlapping blocks. Later, Liu and Singh (1992) proposed the moving blocks procedure, which in turn uses overlapping blocks with the final objective of minimizing variability. For this study, due to the heavy tail characteristic of the time series data that we have, it is difficult to choose a block length in order to perform block bootstrapping. As a result, the Residual Bootstrap procedure will be used instead to perform the analysis.

3.2 Residual Bootstrap Confidence Intervals for ACD (p,q) Models

We will now use the insight gained from the given information in the previous section as a tool to help understand the Residual Bootstrap algorithm employed in this chapter. In the last chapter, we use OLS and GLS in order to estimate the parameters of three models: ACD (1), ACD (2) and ACD (1,1). However, one's interest might lie in constructing confidence intervals based on such parameter estimates. Thus, in this chapter, we construct confidence intervals based on the Residual Bootstrap method. The next section provides a description of the Residual Bootstrap method used to find the confidence intervals of $\hat{\alpha}_0$, $\{\hat{\alpha}_i\}_{i=1}^{p}$ and $\left\{\hat{\beta}_i\right\}_{i=1}^{q}$ for the ACD (p,q) model by using OLS/GLS.

3.2.1 Residual Bootstrap Procedure for ACD (p,q) Models Using OLS/GLS

In this section, we present the procedure for finding the confidence intervals of the estimates of the ACD (p,q) model using Residual Bootstrap.

1. Calculate $\hat{\alpha}_0$, $\{\hat{\alpha}_i\}_{i=1}^{p}$ and $\left\{\hat{\beta}_i\right\}_{i=1}^{q}$ of the ACD (p,q) model of $\{X_t\}_{t\in\mathbb{Z}}$ using OLS/GLS.

2. Estimate the residuals $\{\hat{\varepsilon}_t\}_{t\in\mathbb{Z}}$ based on the ACD (p,q) model.

3. Re-sample with replacement from $\{\hat{\varepsilon}_t\}_{t\in\mathbb{Z}}$. Therefore, the empirical distribution of $\{\hat{\varepsilon}_t\}_{t\in\mathbb{Z}}$ follows: $F_{\epsilon,n(x)} = \dfrac{1}{n}\displaystyle\sum_{t=1}^{n} \mathbf{1}_{\{x\geq\hat{\varepsilon}_t\}}$, for $\{x \geq \hat{\varepsilon}_t\}$ representing the number of observed residuals less than or equal to x. The resulting bootstrapped residuals are denoted by: $\{\varepsilon_t^*\}_{t\in\mathbb{Z}}$.

4. Construct the bootstrap process $\{X_t^*\}_{t\in\mathbb{Z}}$ by: $\hat{\psi}_t = \hat{\alpha}_0 + \displaystyle\sum_{i=1}^{p} \hat{\alpha}_i X_{t-i}^* + \sum_{i=1}^{q} \hat{\beta}_i \psi_{t-i}$ and

   $X_t^* = \psi_t \varepsilon_t^*$, for $\{\varepsilon_t^*\}_{t\in\mathbb{Z}}$ are i.i.d.

5. Re-estimate the ACD (p,q) coefficients: $\hat{\alpha}_0$, $\{\hat{\alpha}_i\}_{i=1}^{p}$ and $\left\{\hat{\beta}_i\right\}_{i=1}^{q}$ using OLS/GLS.

6. With *B* representing the number of bootstrap replicates, repeat steps 3-5 *B* times to obtain the bootstrap estimates: $\{\alpha_0^*\}$, $\{\alpha_i^*\}_{i=1}^{p}$ and $\{\beta_i^*\}_{i=1}^{q}$.

   Now we use the bootstrap distribution of α_0^*, $\{\alpha_i^*\}_{i=1}^{p}$ and $\{\beta_i^*\}_{i=1}^{q}$ produced by steps 3-6, i.e.: $\widehat{F}_{\alpha_0^*}$, $\widehat{F}_{\{\alpha_i^*\}_{i=1}^{p}}$ and $\widehat{F}_{\{\beta_i^*\}_{i=1}^{q}}$ to approximate the unknown distribution of α_0, $\{\alpha_i\}_{i=1}^{p}$ and $\{\beta_i\}_{i=1}^{q}$ respectively.

   The 100 (1 - α)% interval of α_0 is given by: $\left[M_n^*\left(\frac{\alpha}{2}\right), M_n^*(1-\frac{\alpha}{2})\right]$, where $M_n^*(1-\alpha)$ is the 100 (1 - α)% quantile of $\widehat{F}_{\alpha_0^*}$.

   The 100 (1 - α)% interval of $\{\alpha_i\}_{i=1}^{p}$ is given by: $\left[N_n^*\left(\frac{\alpha}{2}\right), N_n^*(1-\frac{\alpha}{2})\right]$, where $N_n^*(1-\alpha)$ is the 100 (1 - α)% quantile of $\widehat{F}_{\{\alpha_i^*\}_{i=1}^{p}}$.

   The 100 (1 - α)% interval of $\{\beta_i\}_{i=1}^{q}$ is given by: $\left[R_n^*\left(\frac{\alpha}{2}\right), R_n^*(1-\frac{\alpha}{2})\right]$, where $R_n^*(1-\alpha)$ is the 100 (1 - α)% quantile of $\widehat{F}_{\{\beta_i^*\}_{i=1}^{q}}$.

3.3 Monte Carlo Simulation Results of the Residual Bootstrapped Confidence Intervals

3.3.1 Results

By applying the Residual Bootstrap procedure for ACD (1), ACD (2) and ACD (1,1) models, we wish to find the confidence intervals for α_0 and α_1 for ACD (1), α_0, α_1 and α_2 for ACD (2) and α_0, α_1 and β_1 for ACD (1,1).

Recall from Chapter 2:

Model 1: ACD (1)

$$X_t = \psi_t \varepsilon_t, \tag{3.1}$$

$$\psi_t = \alpha_0 + \alpha_1 X_{t-1}. \tag{3.2}$$

Model 2: ACD (2)

$$X_t = \psi_t \varepsilon_t, \tag{3.3}$$

$$\psi_t = \alpha_0 + \alpha_1 X_{t-1} + \alpha_2 X_{t-2}. \tag{3.4}$$

Model 3: ACD (1,1)

$$X_t = \psi_t \varepsilon_t, \tag{3.5}$$

$$\psi_t = \alpha_0 + \alpha_1 X_{t-1} + \beta_1 \psi_{t-1}, \tag{3.6}$$

The residual bootstrap confidence intervals for ACD (1), ACD (2) and ACD (1,1) are estimated for various cases with MC = 1000 and B = 999. In all cases, α_0 = 1. Exponential distributions with truncated estimates are considered.

Table 3.1: Residual Bootstrap Confidence Intervals (RBCI) for ACD (1), ACD (2) and ACD (1,1).

Description	Method	RBCI α_0	RBCI α_1
ACD (1), α_1 = 0.4, N = 500	OLS	(0.8346,1.2847)	(0.2044,0.5264)
ACD (1), α_1 = 0.4, N = 500	GLS	(0.8683,1.1990)	(0.2474,0.4997)
ACD (1), α_1 = 0.1, N = 5000	OLS	(0.9247,1.0098)	(0.0839,0.1527)
ACD (1), α_1 = 0.1, N = 5000	GLS	(0.9298,1.0164)	(0.0766,0.1432)

Description	Method	RBCI α_0	RBCI α_1	RBCI α_2
ACD (2), (α_1, α_2) = (0.30, 0.15), N = 500	OLS	(0.6448,1.3718)	(0.2739,0.6880)	(0.0001,0.1607)
ACD (2), (α_1, α_2) = (0.30, 0.15), N = 500	GLS	(0.7715,1.2191)	(0.2186,0.4841)	(0.0306,0.2551)

Description	Method	RBCI α_0	RBCI α_1	RBCI β_1
ACD (1,1), α_1 = 0.4, β_1 = 0.3, N = 500	OLS	(1.0540,2.5836)	(0.1134,0.5604)	(0.0001,0.2937)
ACD (1,1), α_1 = 0.4, β_1 = 0.3, N = 500	GLS	(1.0263,2.5836)	(0.1075,0.5589)	(0.0001,0.3055)

As well, the coverage probabilities for the estimates of ACD (1), ACD (2) and ACD (1,1) are presented below. We note that the nominal value considered is 95%.

Table 3.2: Coverage of the OLS and GLS Estimates Using Residual Bootstrap for ACD (1)

Description	α_0^{GLS}	α_1^{GLS}	α_0^{OLS}	α_1^{OLS}
ACD (1), α_0 = 0.4, N = 500	0.9450	0.9220	0.9490	0.9140
ACD (1), α_0 = 0.4, N = 1000	0.9380	0.9310	0.9650	0.9220
ACD (1), α_0 = 0.4, N = 5000	0.9570	0.9480	0.9780	0.9240
ACD (1), α_0 = 0.1, N = 500	0.9520	0.9370	0.9490	0.9390
ACD (1), α_0 = 0.1, N = 1000	0.9420	0.9290	0.9470	0.9300
ACD (1), α_0 = 0.1, N = 5000	0.9530	0.9530	0.9500	0.9440

Table 3.3: Coverage of the OLS and GLS Estimates Using Residual Bootstrap for ACD (2)

Description	α_0^{GLS}	α_1^{GLS}	α_2^{GLS}	α_0^{OLS}	α_1^{OLS}	α_2^{OLS}
ACD (2), (α_1, α_2) = (0.30, 0.15), N = 500	0.9230	0.9220	0.9280	0.9140	0.9310	0.9200
ACD (2), (α_1, α_2) = (0.30, 0.15), N = 1000	0.9350	0.9430	0.9390	0.9550	0.9350	0.9330
ACD (2), (α_1, α_2) = (0.30, 0.15), N = 5000	0.9480	0.9489	0.9490	0.9670	0.9430	0.9470

Table 3.4: Coverage of the OLS and GLS Estimates Using Residual Bootstrap for ACD (1,1)

Description	α_0^{GLS}	α_1^{GLS}	β_1^{GLS}	α_0^{OLS}	α_1^{OLS}	β_1^{OLS}
ACD (1,1), α_1 = 0.4, β_1 = 0.3, N = 500	0.9050	0.9120	0.9490	0.8980	0.9240	0.9390
ACD (1,1), α_1 = 0.4, β_1 = 0.3, N = 500	0.9650	0.9520	0.9590	0.9530	0.9480	0.9530

3.3.2 Key Remarks

We first consider ACD (1) with true values α_0 = 1 and α_1 = 0.4, 1000 MC replicates, B = 999 and N = 500. In this case, the confidence interval generated using GLS estimation for α_0 is (0.8683,1.1990) and for α_1 is (0.2474,0.4997). The confidence interval generated using OLS estimation for α_0 is (0.8346,1.2847) and for α_1 is (0.2044,0.5264). Notice that the expectation of α_0, $E(\alpha_0)$, using OLS estimation is 1.0356 and using GLS estimation is 1.0105. The expectation of α_1, $E(\alpha_1)$, using OLS estimation is 0.3716 and using GLS estimation is 0.3878. In comparison to the true values of the parameters, it is concluded that $E(\alpha_0)$ of the OLS estimate is closer to the true value than $E(\alpha_0)$ of the GLS estimate. Also, $E(\alpha_1)$ of the GLS estimate is closer to the true value than $E(\alpha_1)$ of the OLS estimate. However, the difference is really negligible. Both methods of estimation are thus asymptotically consistent and efficient. The coverage probability of the estimates using the Residual Bootstrap procedure is 0.9450 for α_0 and 0.9220 for α_1 by using GLS as the method of estimation. Also, the coverage probability of the estimates using the Residual Bootstrap procedure is 0.9490 for α_0 and 0.9140 for α_1 by using OLS as the method of estimation.

We consider another case: ACD (1,1) with true values α_0 = 1, α_0 = 0.4, β_0 = 0.3 and 1000 MC, B = 999 and N = 500. In this case, the confidence interval generated using GLS estimation for α_0, α_1 and β_1 are (1.0263,2.5836), (0.1075,0.5589) and (0.0001,0.3055). The confidence intervals generated using OLS estimation for α_0, α_1 and β_1 are (1.0540,2.5836), (0.1134,0.5604) and (0.0001,0.2937). Also, the coverage probability of the estimates using the Residual Bootstrap procedure is 0.9050 for α_0, 0.9120 for α_1 and 0.9490 for β_1 by using GLS estimation. The coverage probability of the estimates using the Residual Bootstrap procedure is 0.8980 for α_0, 0.9240 for α_1 and 0.9390 for β_1 by using OLS estimation. In this case, the GLS estimates have a coverage closer to the nominal value of 95% than the OLS estimates. Therefore, the GLS method of estimation outperforms OLS even though it can be conluded that OLS is still consistent and efficent.

We discuss the results of finding the Residual Bootstrap confidence intervals when the underlying distribution is Weibull with truncated estimates. Here, the results are very similar to the case of Exponential distribution with truncated estimates but for certain scale parameters only. In other terms, it is noticed that as sample size increased to 5000, the coverage of the estimates decreased drastically and the results were not satisfactory at all. In particular, we need to theoretically justify the consistency of the moment condition of the Weibull distribution before attempting to find the Residual Bootstrap confidence intervals, and this requires more investigation in the future. At the moment, we are unable to finalize any conclusions about the efficiency of the estimates when a Weibull distribution with truncated estimates is considered.

Chapter 4

Case Study: IBM Stock

4.1 Description of the IBM Transaction Data Set

This section will be an application of the ACD model to IBM transaction data set. The IBM data set ranges from January 02, 2002 until February 28, 2002, with a total of 127309 observations. Detailed information such as the date of the transaction at hand, the time after midnight in seconds, the duration between the trades in seconds, the volume of the transactions, the total value of the trade (computed as a product of the shares and the price) and the average prices of the transactions were examined, with a minimum increment of 1 second. Duration was calculated as the difference in times between two consecutive trades. As mentioned in Chapter 1, trade duration is

$$X_i = t_i - t_{i-1}. \tag{4.1}$$

Let us take for example the first two trades that occurred on January 02, 2002. The time after midnight of the first trade (observation) is 35418 and the time after midnight of the second trade (observation) is 35421. Thus, trade duration between the first and second observation is 3 seconds.

IBM stock is traded on various US exchange markets that open at 9:30 a.m. EST time and close at 4:00 p.m. EST time.

Table 4.1: Summary Statistics for Duration Data of IBM Stock Set Between 10:00 a.m. and 4:00 p.m.

Sample Size	**Mean**	**Median**	**60% percentile**	**70% percentile**
127309	6.787	5.000	6.000	8.000

Table 4.1 depicts a right skewness pattern as the mean of the IBM duration data set lies in the 60% percentile - 70% percentile interval and is of a larger value than the median.

It is documented in time series and econometric literature that a daily pattern is present during a specified trading day (Engle, 2000; Pacurar, 2006; Engle and Russell, 2004). In particular, there appears to be certain trading periods which are considered to be relatively more or less active than other trading periods.

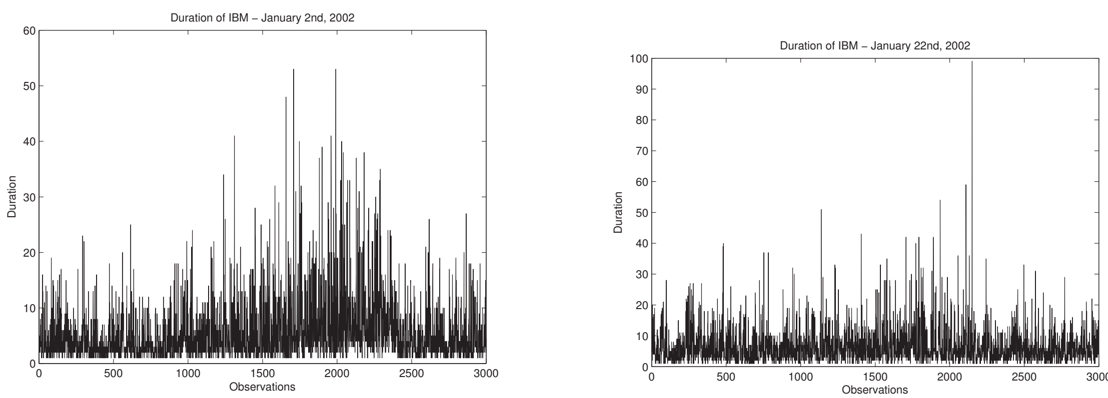
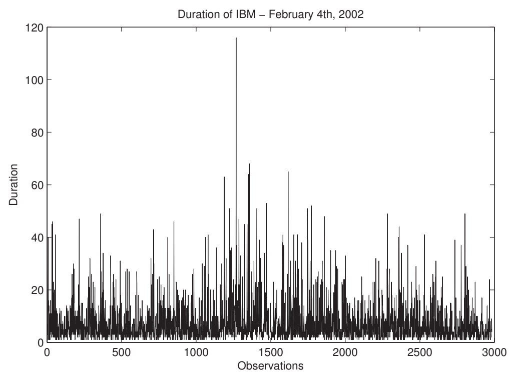


Figure 4.1: Duration Plots of IBM Observations for January 2, 2002 and January 22, 2002



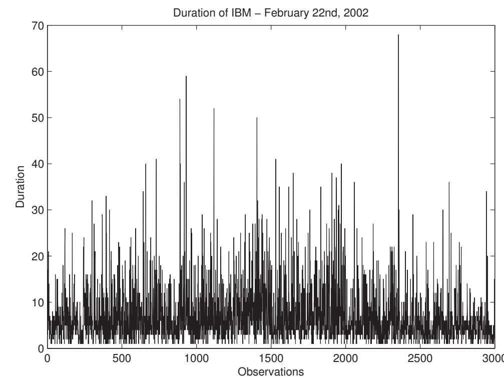


Figure 4.2: Duration Plots of IBM Observations for February 4, 2002 and February 22, 2002

Figures 4.1 and 4.2 are duration plots of observations for January 2, 2002, January 22, 2002, February 4, 2002 and February 22, 2002. There is a clear indication that IBM duration process has a diurnal pattern. In other words, durations are shorter at the opening of the trading day and closing hour of the trading day. They tend to be longer in the middle of the day (i.e., around noon time). Typically, there is a transaction every 35 seconds.

4.2 Removal of Diurnal Patterns

Due to the opening auctions of trading days, short durations are seen during the opening event. The latter auctions are events where the specialist sets a certain price in order to achieve the maximum number of transactions. The open transactions are all then recorded. Before the non-opening transactions begin, all open transactions must be complete. Because trading activity at the opening time might be delayed, the transactions would be recorded 34 seconds after 9:30 a.m. and might even extend until 9:45 a.m. (Engle and Russell, 1998). In order to account for the deterministic diurnality effect, transactions occuring between 9:30 a.m. and 9:50 a.m. of the trading day were deleted. Moreover, each trading day accounts for the 10 minutes prior to 10:00 a.m. by considering their conditional expected duration and setting it equal to the mean of the duration past those minutes. This also accounts for the effects carried from the closing of the previous trading day to the next trading day. For $\phi\left(t_{i-1};\theta_{\phi}\right)$ representing the daily seasonality effect which could be calculated using a cubic spline factor, the diurnally adjusted data can be represented by:

$$\tilde{X}_i = \frac{X_i}{\phi\left(t_{i-1};\theta_{\phi}\right).} \tag{4.2}$$

Also, the expected duration of the diurnally adjusted data can be formulated as:

$$E_{i-1}\left(X_i\right) = \phi\left(t_{i-1};\theta_{\phi}\right)\psi_i\left(\tilde{X}_i,\ldots,\tilde{X}_1;\theta_{\psi}\right). \tag{4.3}$$

As well, we note that nodes were set on each trading hour. The expectation is found by taking the average of the durations over each one-hour interval for all the trading days along with an extra 30-minute duration interval at the closing of the trading day. We then apply the cubic spline to smooth the time of the trading day function for each day as shown below in Figure 4.3. Figure 4.4 is an average of all cubic splines over January 02, 2002 until February 28, 2002. We also keep in mind that the constant factor for the cubic spline is recognized by equating the mean of the predicted diurnal factor and the observed sample mean.

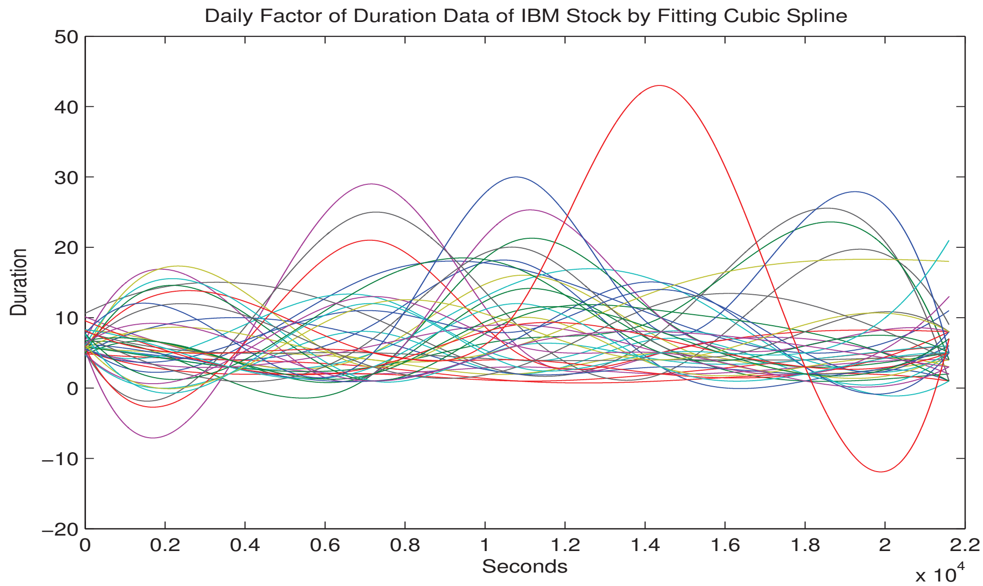


Figure 4.3: Daily Factor of Duration Data of IBM Stock by Fitting Cubic Spline for January 2, 2002 until February 28, 2002

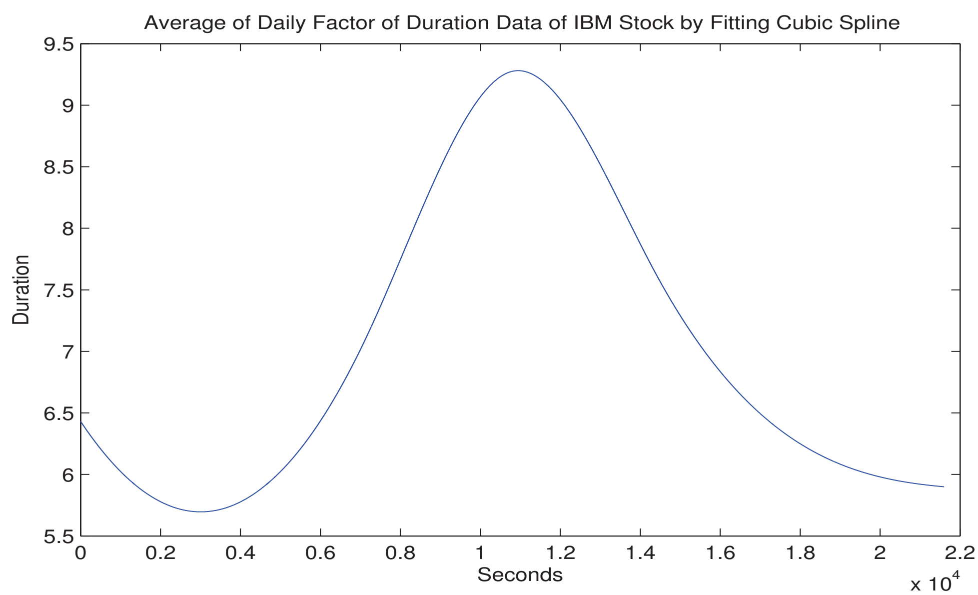


Figure 4.4: Average of Daily Factor of Duration Data of IBM Stock by Fitting Cubic Spline for January 2, 2002 until February 28, 2002

4.3 ACD (1,1) Model Estimation

As seen from Figure 4.4, it can be concluded that removing the diurnality effect results in a much smoother data set by excluding almost all the intra-daily pattern. For model estimation, we explore different model candidates such as: ACD (1), ACD (2), ACD (1,1) along with $\varepsilon_t \sim$ i.i.d. Exponential and Weibull. After testing various models, we came upon the conclusion that the ACD (1,1) model with ε_t i.i.d. Exponential appears to fit the IBM duration data set the best due to the fact that the ACF and PACF plots of ACD (1,1) both show significant lags.

Using OLS, and for $\hat{\alpha}_0$ = 0.2726, $\hat{\alpha}_1$ = 0.0551 and $\hat{\beta}_1$ = 0.9046, the ACD (1,1) model which appears to model the IBM data set is:

$$\begin{aligned} X_t &= \psi_t \varepsilon_t \\ \psi_t &= 0.2726 + 0.0551 X_{t-1} + 0.9046 \psi_{t-1}. \end{aligned}$$

Using GLS, and for $\hat{\alpha}_0$ = 0.2181, $\hat{\alpha}_1$ = 0.0483 and $\hat{\beta}_1$ = 0.9195, the ACD (1,1) model which appears to model the IBM data set is:

$$\begin{aligned} X_t &= \psi_t \varepsilon_t \\ \psi_t &= 0.2181 + 0.0483 X_{t-1} + 0.9195 \psi_{t-1}. \end{aligned}$$

For both OLS and GLS as the proposed methods of estimation, the summation of $\hat{\alpha}_1$ and $\hat{\beta}_1$ in the two cases is approximately 1. This implies that the model we suggested also captures the clustered duration feature well. The estimated residual bootstrap confidence intervals are shown below.

Table 4.2: Residual Bootstrap Confidence Intervals for ACD (1,1) for Duration Data of IBM Stock.

Method	RBCI α_0	RBCI α_1	RBCI β_1
OLS	(0.2349,0.3049)	(0.0508,0.0589)	(0.8974,0.9132)
GLS	(0.1931,0.2466)	(0.0446,0.0525)	(0.9121,0.9257)

Chapter 5

Concluding Remarks

Estimating the parameters of Autoregressive Conditional Duration (ACD) processes in past literature has been done by using the QMLE method. In this thesis, we use OLS and GLS in order to estimate ACD processes. We theoretically justify the asymptotic consistency and normality of ACD (p) processes. Furthermore, we then show that both OLS and GLS estimates are asymptotically consistent and normally distributed for ACD (p) processes through different MC simulations and using Residual Bootstrap procedure in order to construct confidence intervals based on the OLS and GLS estimates. In most of the cases examined, both the OLS and GLS estimates proved to be asymptotically consistent and appeared to have a normal fit. However, GLS did show better performance than OLS. The CPU time to complete 1 MC replicate using OLS requires half of the time in comparison to that for GLS. Also, in finite samples, it was shown that QMLE outperforms GLS estimation; however the time it takes for 1 MC replicate based on QMLE is greater than that based on GLS estimation and is significantly greater than the timing it takes for OLS estimation. Furthermore, using ACD (1,1), we then apply OLS and GLS estimation on the durations of the IBM transaction data set.

In summary, we were able to conclude that GLS is as efficient an estimator as QMLE for ACD (p) processes. Also, for small samples when considering innovation terms to be i.i.d. Exponential and i.i.d. Weibull, OLS and GLS outperform QMLE. It is of importance to note that both OLS and GLS are applicable methods of estimation in a real time trading setting due to their time efficiency.

5.0.1 Future Work

A theoretical justification of OLS and GLS for general ACD (p,q) processes is in our interest for future research. Also, we need to investigate the consistency of the moment conditions of the Weibull distribution when finding confidence intervals based on the Residual Bootstrap procedure. Applications to different distributions other than Exponential and Weibull could be carried out in order to assess if this incorporation is possible. In addition, our long term goal is to account for the duration component into GARCH processes, and propose a computationally fast method for intra-daily volatility modeling. We will further apply re-sampling and sub-sampling methods for ACD-GARCH models to forecast intra-daily volatility. This current work is a stepping stone in this direction.

Chapter 6

Appendices

6.1 Matlab Codes

6.1.1 QMLE - Exponential Distribution for ACD (1)

```
N=500;
dist='exp';
Coeff.w=1;
Coeff.q=0.1;
[simulDur]=ACD_Simul(N,Coeff,1,1,dist);
[specOut]=ACD_Fit(simulDur,dist,1,1,1);
MC=2000;
mcw=zeros(MC,1);
mcq=zeros(MC,1);
for i=1:MC
[simulDur]=ACD_Simul(N,Coeff,1,1,dist);
[specOut]=ACD_Fit(simulDur,dist,1,1,1);
mcw(i)=specOut.w;
```

```
mcq(i)=specOut.q;
end
```

6.1.2 QMLE - Weibull Distribution for ACD (1)

```
N=5000;
dist='weibull';
Coeff.w=1;
Coeff.q=0.1;
Coeff.y=0.8;
[simulDur]=ACD_Simul(N,Coeff,1,1,dist);
[specOut]=ACD_Fit(simulDur,dist,1,1,1);
MC=2000;
mcw=zeros(MC,1);
mcq=zeros(MC,1);
for i=1:MC
[simulDur]=ACD_Simul(N,Coeff,1,1,dist);
[specOut]=ACD_Fit(simulDur,dist,1,1,1);
mcw(i)=specOut.w;
mcq(i)=specOut.q;
end
```

6.1.3 OLS - Exponential Distribution for ACD (1)

```
N=500;
dist='exp';
Coeff.w=1;
Coeff.q=0.4;
Coeff.p=0;
```

```
MC=2000;
for i=1:MC
[simulDur]=ACD_Simul(N,Coeff,1,1,dist);
a1=armax(iddata(simulDur-mean(simulDur)), 'na', 1, 'nc', 1);
beta=-a1.c(2);
alpha=-a1.a(2)-beta;
omega=mean(simulDur)/(1-alpha-beta);
mbeta(i)=beta;
malpha(i)=alpha;
momega(i)=omega;
end
```

6.1.4 GLS - Exponential Distribution for ACD (1)

```
N=500;
dist='exp';
Coeff.w=1;
Coeff.q=0.4;
Coeff.p=0;
MC=2000;
p=1;
q=1;
alpha0_hat=zeros(MC,1);
alpha1_hat=zeros(MC,1);
alpha0_hat_GLS=zeros(MC,1);
alpha1_hat_GLS=zeros(MC,1);
for i=1:MC
[simulDur]=ACD_Simul(N,Coeff,1,1,dist);
```

```
y2=simulDur;
a1=ar(y2-mean(y2), q, 'ls');
alpha1_hat(i)=-a1.parameterVector(1);
alpha0_hat(i)=(1-alpha1_hat(i))*mean(y2);
sigma2_hat=zeros(N,1);
sigma2_hat(1)= alpha0_hat(i);
for j=2:N
sigma2_hat(j)= alpha0_hat(i) + alpha1_hat(i)* y2(j-1);
end
sigma4_hat= sigma2_hat.^(-2);
V=diag(sigma4_hat);
X=[ones(N,1), [0,y2(1:(N-1))']'];
alpha= inv(X'*V*X)*X'*V*y2;
alpha1_hat_GLS(i) = alpha(2);
alpha0_hat_GLS(i)= alpha(1);
end
```

6.1.5 GLS - Exponential Distribution for ACD (2)

```
N=5000;
dist='exp';
Coeff.w=1;
Coeff.q=[0.3 0.15];
Coeff.p=0;
MC=2000;
p=1;
q=size(Coeff.q,2);
alpha0_hat=zeros(MC,1);
```

```
alpha1_hat=zeros(MC,1);
alpha2_hat=zeros(MC,1);
alpha0_hat_GLS=zeros(MC,1);
alpha1_hat_GLS=zeros(MC,1);
alpha2_hat_GLS=zeros(MC,1);
for i=1:MC
[simulDur]=ACD_Simul(N,Coeff,2,1,dist);
y2=simulDur;
a1=ar(y2-mean(y2), q, 'ls');
alpha1_hat(i)=-a1.parameterVector(1);
alpha2_hat(i)=-a1.parameterVector(2);
alpha0_hat(i)=(1-alpha1_hat(i)-alpha2_hat(i))*mean(y2);
sigma2_hat=zeros(N,1);
sigma2_hat(1)= alpha0_hat(i);
sigma2_hat(2)= alpha0_hat(i);
for j=3:T
sigma2_hat(j)= alpha0_hat(i) + alpha1_hat(i)* y2(j-1) +
alpha2_hat(i)* y2(j-2);
end
sigma4_hat= sigma2_hat.^(-2);
V=diag(sigma4_hat);
X=[ones(N,1), [0,y2(1:(N-1))']', [0,0, y2(1:(N-2))']'];
alpha= inv(X'*V*X)*X'*V*y2;
alpha0_hat_GLS(i)= alpha(1);
alpha1_hat_GLS(i)= alpha(2);
alpha2_hat_GLS(i)= alpha(3);
```

```
end
subplot(3,2,1)
histfit(alpha0_hat)
subplot(3,2,2)
histfit(alpha1_hat)
subplot(3,2,3)
histfit(alpha0_hat_GLS)
subplot(3,2,4)
histfit(alpha1_hat_GLS)
subplot(3,2,5)
histfit(alpha2_hat)
subplot(3,2,6)
histfit(alpha2_hat_GLS)
```

6.1.6 GLS - Exponential Distribution (Truncated Estimates) for ACD (1)

```
N=5000;
dist='exp';
Coeff.w=1;
Coeff.q=0.4;
Coeff.p=0;
MC=2000;
p=1;
q=1;
alpha0_hat=zeros(MC,1);
alpha1_hat=zeros(MC,1);
alpha0_hat_GLS=zeros(MC,1);
alpha1_hat_GLS=zeros(MC,1);
```

```
for i=1:MC
[simulDur]=ACD_Simul(N,Coeff,1,1,dist);
y2=simulDur;
a1=ar(y2-mean(y2), q, 'ls');
if -a1.parameterVector(1)<0
alpha1_hat(i)= 0.0001;
else
alpha1_hat(i)=-a1.parameterVector(1);
end
alpha0_hat(i)=(1-alpha1_hat(i))*mean(y2);
sigma2_hat=zeros(N,1);
sigma2_hat(1)= alpha0_hat(i);
for j=2:N
sigma2_hat(j)= alpha0_hat(i) + alpha1_hat(i)* y2(j-1);
end
sigma4_hat= sigma2_hat.^(-2);
V=diag(sigma4_hat);
X=[ones(N,1), [0,y2(1:(N-1))']'];
alpha= inv(X'*V*X)*X'*V*y2;
if alpha(2) < 0
alpha1_hat_GLS(i) =0.0001;
else
alpha1_hat_GLS(i) = alpha(2);
end
alpha0_hat_GLS(i)= alpha(1);
end
```

6.1.7 GLS - Exponential Distribution (Truncated Estimates) for ACD (2)

```
N=5000;
dist='exp';
Coeff.w=1;
Coeff.q=[0.3  0.15];
Coeff.p=0;
MC=2000;
p=1;
q=size(Coeff.q,2);
alpha0_hat=zeros(MC,1);
alpha1_hat=zeros(MC,1);
alpha2_hat=zeros(MC,1);
alpha0_hat_GLS=zeros(MC,1);
alpha1_hat_GLS=zeros(MC,1);
alpha2_hat_GLS=zeros(MC,1);
for  i=1:MC
[simulDur]=ACD_Simul(N,Coeff,2,1,dist);
y2=simulDur;
a1=ar(y2-mean(y2),  q,  'ls');
if  -a1.parameterVector(1)<0
alpha1_hat(i)=  0.0001;
else
alpha1_hat(i)=-a1.parameterVector(1);
end
if  -a1.parameterVector(2)<0
alpha2_hat(i)=  0.0001;
```

```
else
alpha2_hat(i)=-a1.parameterVector(2);
end
alpha0_hat(i)=(1-alpha1_hat(i)-alpha2_hat(i))*mean(y2);
sigma2_hat=zeros(N,1);
sigma2_hat(1)= alpha0_hat(i);
sigma2_hat(2)= alpha0_hat(i);
for j=3:N
sigma2_hat(j)= alpha0_hat(i) + alpha1_hat(i)* y2(j-1) +
alpha2_hat(i)* y2(j-2);
end
sigma4_hat= sigma2_hat.^(-2);
V=diag(sigma4_hat);
X=[ones(N,1), [0,y2(1:(N-1))']', [0,0, y2(1:(N-2))']'];
alpha= inv(X'*V*X)*X'*V*y2;
alpha0_hat_GLS(i)= alpha(1);
if alpha(2) < 0
alpha1_hat_GLS(i)= 0.0001;
else
alpha1_hat_GLS(i)= alpha(2);
end
if alpha(3) < 0
alpha2_hat_GLS(i)= 0.0001;
else
alpha2_hat_GLS(i)= alpha(3);
end
```

```
end
```

6.1.8 GLS - Weibull Distribution for ACD (1)

```
N=5000;
dist='weibull';
Coeff.w=1;
Coeff.q=0.4;
Coeff.p=0;
Coeff.y=.8;
MC=2000;
p=1;
q=1;
alpha0_hat=zeros(MC,1);
alpha1_hat=zeros(MC,1);
alpha0_hat_GLS=zeros(MC,1);
alpha1_hat_GLS=zeros(MC,1);
for i=1:MC
[simulDur]=ACD_Simul(N,Coeff,1,1,dist);
y2=simulDur;
a1=ar(y2-mean(y2), q, 'ls');
alpha1_hat(i)=-a1.parameterVector(1);
alpha0_hat(i)=(1-alpha1_hat(i))*mean(y2);
sigma2_hat=zeros(N,1);
sigma2_hat(1)= alpha0_hat(i);
for j=2:N
sigma2_hat(j)= alpha0_hat(i) + alpha1_hat(i)* y2(j-1);
end
```

```
sigma4_hat= sigma2_hat.^(-2);
V=diag(sigma4_hat);
X=[ones(N,1), [0,y2(1:(N-1))']'];
alpha= inv(X'*V*X)*X'*V*y2;
alpha1_hat_GLS(i) = alpha(2);
alpha0_hat_GLS(i)= alpha(1);
end
```

6.1.9 GLS - Weibull Distribution for ACD (2)

```
N=5000;
dist='weibull';
Coeff.w=1;
Coeff.q=[0.3 0.15];
Coeff.p=0;
Coeff.y=.8;
MC=2000;
p=1;
q=size(Coeff.q,2);
alpha0_hat=zeros(MC,1);
alpha1_hat=zeros(MC,1);
alpha2_hat=zeros(MC,1);
alpha0_hat_GLS=zeros(MC,1);
alpha1_hat_GLS=zeros(MC,1);
alpha2_hat_GLS=zeros(MC,1);
for i=1:MC
[simulDur]=ACD_Simul(N,Coeff,2,1,dist);
y2=simulDur;
```

```
a1=ar(y2-mean(y2), q, 'ls');
alpha1_hat(i)=-a1.parameterVector(1);
alpha2_hat(i)=-a1.parameterVector(2);
alpha0_hat(i)=(1-alpha1_hat(i)-alpha2_hat(i))*mean(y2);
sigma2_hat=zeros(N,1);
sigma2_hat(1)= alpha0_hat(i);
sigma2_hat(2)= alpha0_hat(i);
for j=3:N
sigma2_hat(j)= alpha0_hat(i) + alpha1_hat(i)* y2(j-1) +
alpha2_hat(i)* y2(j-2);
end
sigma4_hat= sigma2_hat.^(-2);
V=diag(sigma4_hat);
X=[ones(N,1), [0,y2(1:(N-1))']', [0,0, y2(1:(N-2))']'];
alpha= inv(X'*V*X)*X'*V*y2;
alpha0_hat_GLS(i)= alpha(1);
alpha1_hat_GLS(i)= alpha(2);
alpha2_hat_GLS(i)= alpha(3);
end
```

6.1.10 GLS - Weibull Distribution (Truncated Estimates) for ACD (1)

```
N=5000;
dist='weibull';
Coeff.w=1;
Coeff.q=0.4;
Coeff.p=0;
Coeff.y=.8;
```

```
MC=2000;
p=1;
q=1;
alpha0_hat=zeros(MC,1);
alpha1_hat=zeros(MC,1);
alpha0_hat_GLS=zeros(MC,1);
alpha1_hat_GLS=zeros(MC,1);
for i=1:MC
[simulDur]=ACD_Simul(N,Coeff,1,1,dist);
y2=simulDur;
a1=ar(y2-mean(y2), q, 'ls');
if -a1.parameterVector(1)<0
alpha1_hat(i)= 0.0001;
else
alpha1_hat(i)=-a1.parameterVector(1);
end
alpha0_hat(i)=(1-alpha1_hat(i))*mean(y2);
sigma2_hat=zeros(N,1);
sigma2_hat(1)= alpha0_hat(i);
for j=2:N
sigma2_hat(j)= alpha0_hat(i) + alpha1_hat(i)* y2(j-1);
end
sigma4_hat= sigma2_hat.^(-2);
V=diag(sigma4_hat);
X=[ones(N,1), [0,y2(1:(N-1))']'];
alpha= inv(X'*V*X)*X'*V*y2;
```

```
if alpha(2) < 0
alpha1_hat_GLS(i) =0.0001;
else
alpha1_hat_GLS(i) = alpha(2);
end
alpha0_hat_GLS(i)= alpha(1);
end
```

6.1.11 GLS- Weibull Distribution (Truncated Estimates) for ACD (2)

```
N=500;
dist='weibull';
Coeff.w=1;
Coeff.q=[0.3 0.15];
Coeff.p=0;
Coeff.y=.8;
MC=2000;
p=1;
q=size(Coeff.q,2);
alpha0_hat=zeros(MC,1);
alpha1_hat=zeros(MC,1);
alpha2_hat=zeros(MC,1);
alpha0_hat_GLS=zeros(MC,1);
alpha1_hat_GLS=zeros(MC,1);
alpha2_hat_GLS=zeros(MC,1);
for i=1:MC
[simulDur]=ACD_Simul(N,Coeff,2,1,dist);
y2=simulDur;
```

```
a1=ar(y2-mean(y2), q, 'ls');
if -a1.parameterVector(1)<0
alpha1_hat(i)= 0.0001;
else
alpha1_hat(i)=-a1.parameterVector(1);
end
if -a1.parameterVector(2)<0
alpha2_hat(i)= 0.0001;
else
alpha2_hat(i)=-a1.parameterVector(2);
end
alpha0_hat(i)=(1-alpha1_hat(i)-alpha2_hat(i))*mean(y2);
sigma2_hat=zeros(N,1);
sigma2_hat(1)= alpha0_hat(i);
sigma2_hat(2)= alpha0_hat(i);
for j=3:N
sigma2_hat(j)= alpha0_hat(i) + alpha1_hat(i)* y2(j-1) +
alpha2_hat(i)* y2(j-2);
end
sigma4_hat= sigma2_hat.^(-2);
V=diag(sigma4_hat);
X=[ones(N,1), [0,y2(1:(N-1))']', [0,0, y2(1:(N-2))']'];
alpha= inv(X'*V*X)*X'*V*y2;
alpha0_hat_GLS(i)= alpha(1);
if alpha(2) < 0
alpha1_hat_GLS(i)= 0.0001;
```

```
else
alpha1_hat_GLS(i)= alpha(2);
end
if alpha(3) < 0
alpha2_hat_GLS(i)= 0.0001;
else
alpha2_hat_GLS(i)= alpha(3);
end
end
```

6.1.12 GLS - Exponential Distribution (Truncated Estimates) for ACD (1,1)

```
N=500;
dist='exp';
Coeff.w=0.2;
Coeff.q=0.4;
Coeff.p=0.3;
p=1;
q=1;
MC=2000;
for i=1:MC
[simulDur]=ACD_Simul(N,Coeff,1,1,dist);
y2=simulDur;
a1=armax(iddata(y2-mean(y2)),'na',1,'nc',1,'Maxiter=2',2);
if -a1.parameterVector(2) < 0
beta1_hat_GLS(i) = 0.0001
else
beta1_hat_GLS(i) = -a1.parameterVector(2)
```

```
end
if -a1.parameterVector(1) - beta1_hat_GLS(i) <0
alpha1_hat_GLS(i) = 0.0001
else
alpha1_hat_GLS(i) =-a1.parameterVector(1) - beta1_hat_GLS(i)
end
if (1-alpha1_hat_GLS(i)-beta1_hat_GLS(i))*mean(y2) <0
alpha0_hat_GLS(i)=0.0001
else
alpha0_hat_GLS(i)=(1-alpha1_hat_GLS(i)-beta1_hat_GLS(i))*mean(y2)
end
end
```

6.1.13 Procedure for Finding CPU time

```
dist='exp';
Coeff.w=1;
Coeff.q=0.4;
Coeff.p=0;
MC=1;
p=1;
q=1;
alpha0_hat=zeros(MC,1);
alpha1_hat=zeros(MC,1);
alpha0_hat_GLS=zeros(MC,1);
alpha1_hat_GLS=zeros(MC,1);
for i=1:MC
tic
```

```
[simulDur]=ACD_Simul(N,Coeff,1,1,dist);
y2=simulDur;
a1=ar(y2-mean(y2), q, 'ls');
alpha1_hat(i)=-a1.parameterVector(1);
alpha0_hat(i)=(1-alpha1_hat(i))*mean(y2);
sigma2_hat=zeros(N,1);
sigma2_hat(1)= alpha0_hat(i);
for j=2:N
sigma2_hat(j)= alpha0_hat(i) + alpha1_hat(i)* y2(j-1);
end
sigma4_hat= sigma2_hat.^(-2);
V=diag(sigma4_hat);
X=[ones(N,1), [0,y2(1:(N-1))']'];
alpha= inv(X'*V*X)*X'*V*y2;
alpha1_hat_GLS(i) = alpha(2);
alpha0_hat_GLS(i)= alpha(1);
t(i)=toc;
end
t(i)
```

6.1.14 Procedure for Finding the Residual Bootstrap Confidence Intervals - Exponential Distribution (Truncated Estimates) for ACD (1)

```
N=1000;
dist='exp';
Coeff0.w=1;
Coeff0.q=0.4;
Coeff0.p=0;
```

```
MC=2000;
p=1;
q=1;
count_alpha0_P=0;
count_alpha1_P=0;
count_alpha0_OLS_P=0;
count_alpha1_OLS_P=0;
for i=1:MC
[simulDur]=ACD_Simul(N,Coeff0,1,1,dist);
y2=simulDur;
a1=ar(y2-mean(y2), q, 'ls');
if -a1.parameterVector(1)<0
alpha1_hat_OLS(i)= 0.0001;
else
alpha1_hat_OLS(i)=-a1.parameterVector(1);
end
alpha0_hat_OLS(i)=(1-alpha1_hat_OLS(i))*mean(y2);
sigma2_hat_OLS=zeros(N,1);
sigma2_hat_OLS(1)= alpha0_hat_OLS(i);
for j=2:N
sigma2_hat_OLS(j)= alpha0_hat_OLS(i) + alpha1_hat_OLS(i)* y2(j-1);
end
sigma4_hat_OLS= sigma2_hat_OLS.^(-2);
V=diag(sigma4_hat_OLS);
X=[ones(N,1), [0,y2(1:(N-1))']'];
alpha= inv(X'*V*X)*X'*V*y2;
```

```
if alpha(2) < 0
alpha1_hat_GLS(i) = 0.0001;
else
alpha1_hat_GLS(i) = alpha(2);
end
alpha0_hat_GLS(i) = alpha(1);
sigma2_hat_GLS=zeros(N,1);
sigma2_hat_GLS(1)= alpha0_hat_GLS(i);
for j=2:N
sigma2_hat_GLS(j)= alpha0_hat_GLS(i) + alpha1_hat_GLS(i)* y2(j-1);
end
%estimate the residuals
r_OLS=zeros(1,N);
r_GLS=zeros(1,N);
for t=1:N
r_OLS(t)=y2(t)/sigma2_hat_OLS(t);
r_GLS(t)=y2(t)/sigma2_hat_GLS(t);
end
B=999;
alpha0_hat_OLS_bs=zeros(B, 1);
alpha1_hat_OLS_bs=zeros(B, 1);
alpha0_hat_GLS_bs=zeros(B, 1);
alpha1_hat_GLS_bs=zeros(B, 1);
for k=1:B
%bootstrap residuals
r_star_OLS=r_OLS(unidrnd(length(r_OLS),1,N));
```

```
r_star_GLS=r_GLS(unidrnd(length(r_GLS),1,N));
%construct the bootstrap process
phi_OLS=zeros(N,1);
y2_bs_OLS=zeros(N,1);
phi_OLS(1) = alpha0_hat_OLS(i);
y2_bs_OLS(1) = phi_OLS(1)*r_star_OLS(1);
phi_GLS=zeros(N,1);
y2_bs_GLS=zeros(N,1);
phi_GLS(1) = alpha0_hat_GLS(i);
y2_bs_GLS(1) = phi_GLS(1)*r_star_GLS(1);
for j = 2:N
phi_OLS(j) = alpha0_hat_OLS(i) + alpha1_hat_OLS(i)*y2_bs_OLS(j-1);
y2_bs_OLS(j) = phi_OLS(j)*r_star_OLS(j);
phi_GLS(j) = alpha0_hat_GLS(i) + alpha1_hat_GLS(i)*y2_bs_GLS(j-1);
y2_bs_GLS(j) = phi_GLS(j)*r_star_GLS(j);
end
a1_bs_OLS=ar(y2_bs_OLS-mean(y2_bs_OLS), q, 'ls');
a1_bs_GLS=ar(y2_bs_GLS-mean(y2_bs_GLS), q, 'ls');
if -a1_bs_OLS.parameterVector(1)<0
alpha1_hat_OLS_bs(k)= 0.0001;
else
alpha1_hat_OLS_bs(k)=-a1_bs_OLS.parameterVector(1);
end
alpha0_hat_OLS_bs(k)=(1-alpha1_hat_OLS_bs(k))*mean(y2_bs_OLS);
if -a1_bs_GLS.parameterVector(1)<0
alpha1_bs_temp= 0.0001;
```

```
else
alpha1_bs_temp=-a1_bs_GLS.parameterVector(1);
end
alpha0_bs_temp=(1-alpha1_bs_temp)*mean(y2_bs_GLS);
sigma2_hat_bs_temp=zeros(N,1);
sigma2_hat_bs_temp(1)= alpha0_bs_temp;
for j=2:N
sigma2_hat_bs_temp(j)= alpha0_bs_temp +
alpha1_bs_temp* y2_bs_GLS(j-1);
end
sigma4_hat_bs_temp= sigma2_hat_bs_temp.^(-2);
V_bs=diag(sigma4_hat_bs_temp);
X_bs=[ones(N,1), [0,y2_bs_GLS(1:(N-1))']'];
alpha_bs= inv(X_bs'*V_bs*X_bs)*X_bs'*V_bs*y2_bs_GLS;
if alpha_bs(2) < 0
alpha1_hat_GLS_bs(k) =0.0001;
else
alpha1_hat_GLS_bs(k) = alpha_bs(2);
end
alpha0_hat_GLS_bs(k)= alpha_bs(1);
end
s_GLS=sort(alpha0_hat_GLS_bs);
r_GLS=sort(alpha1_hat_GLS_bs);
s_OLS=sort(alpha0_hat_OLS_bs);
r_OLS=sort(alpha1_hat_OLS_bs);
[s_OLS(25) s_OLS(975)];
```

```
[s_GLS(25) s_GLS(975)];
[r_OLS(25) r_OLS(975)];
[r_GLS(25) r_GLS(975)];
if s_OLS(25)<Coeff0.w & s_OLS(975)>Coeff0.w
count_alpha0_OLS_P = count_alpha0_OLS_P +1;
end
if r_OLS(25)<Coeff0.q & r_OLS(975)>Coeff0.q
count_alpha1_OLS_P = count_alpha1_OLS_P +1;
end
if s_GLS(25)<Coeff0.w & s_GLS(975)>Coeff0.w
count_alpha0_P = count_alpha0_P +1;
end
if r_GLS(25)<Coeff0.q & r_GLS(975)>Coeff0.q
count_alpha1_P = count_alpha1_P +1;
end
```

6.1.15 Analysis of IBM Transaction Data Set

```
a = importdata('ibm_trades.txt','\t');
a.textdata(1) = [];
Time = a.data(:,1);
X = a.data(:,2);
sample_size_ibm = length(a.data(:,8))-sum(a.data(:,8));
median_ibm = median(X(a.data(:,8)==0));
pt60_ibm = prctile(X(a.data(:,8)==0),60);
mean_ibm = mean(X(a.data(:,8)==0));
pt70_ibm = prctile(X(a.data(:,8)==0),70);
a1=armax(iddata(X-mean(X)),'na',1,'nc',1,'Maxiter=2',2);
```

```
if -a1.parameterVector(2) < 0
beta1_hat = 0.0001
else
beta1_hat = -a1.parameterVector(2)
end
if -a1.parameterVector(1) - beta1_hat <0
alpha1_hat = 0.0001
else
alpha1_hat =-a1.parameterVector(1) - beta1_hat
end
if (1-alpha1_hat-beta1_hat)*mean(X) <0
alpha0_hat=0.0001
else
alpha0_hat=(1-alpha1_hat-beta1_hat)*mean(X)
end
T = length(X);
sigma2_hat=zeros(T,1);
sigma2_hat(1)= alpha0_hat;
for j=2:T
sigma2_hat(j)= alpha0_hat + alpha1_hat* X(j-1) +
beta1_hat*sigma2_hat(j-1);
end
r=zeros(1,T);
for t=1:T
r(t)=X(t)/sigma2_hat(t);
end
```

```
B=999;
alpha0_hat_bs=zeros(B, 1);
alpha1_hat_bs=zeros(B, 1);
beta1_hat_bs=zeros(B, 1);
for k=1:B
r_star=r(unidrnd(length(r),1,T));
phi=zeros(T,1);
X_bs=zeros(T,1);
phi(1) = alpha0_hat;
X_bs(1) = phi(1)*r_star(1);
for j = 2:T
phi(j) = alpha0_hat + alpha1_hat*X_bs(j-1) + beta1_hat*phi(j-1);
X_bs(j) = phi(j)*r_star(j);
end
a1_bs=armax(iddata(X_bs-mean(X_bs)),'na',1,'nc',1,'Maxiter=2',2);
if -a1_bs.parameterVector(2) < 0
beta1_hat_bs(k) = 0.0001;
else
beta1_hat_bs(k) = -a1_bs.parameterVector(2);
end
if -a1_bs.parameterVector(1) - beta1_hat_bs(k) <0
alpha1_hat_bs(k) = 0.0001;
else
alpha1_hat_bs(k) =-a1_bs.parameterVector(1) - beta1_hat_bs(k);
end
if (1-alpha1_hat_bs(k)-beta1_hat_bs(k))*mean(X_bs) <0
```

```
alpha0_hat_bs(k)=0.0001;
else
alpha0_hat_bs(k)=(1-alpha1_hat_bs(k)-beta1_hat_bs(k))*mean(X_bs);
end
end
s=sort(alpha0_hat_bs);
r=sort(alpha1_hat_bs);
v=sort(beta1_hat_bs);
CI_alpha0_hat_GLS = [s(25) s(975)]
CI_alpha1_hat_GLS = [r(25) r(975)]
CI_beta1_hat_GLS = [v(25) v(975)]
alpha0_hat_GLS = alpha0_hat;
alpha1_hat_GLS = alpha1_hat;
beta1_hat_GLS = beta1_hat;
a1=armax(iddata(X-mean(X)),'na',1,'nc',1,'Maxiter=1',1);
if -a1.parameterVector(2) < 0
beta1_hat = 0.0001
else
beta1_hat = -a1.parameterVector(2)
end
if -a1.parameterVector(1) - beta1_hat <0
alpha1_hat = 0.0001
else
alpha1_hat =-a1.parameterVector(1) - beta1_hat
end
if (1-alpha1_hat-beta1_hat)*mean(X) <0
```

```
alpha0_hat=0.0001
else
alpha0_hat=(1-alpha1_hat-beta1_hat)*mean(X)
end
T = length(X);
sigma2_hat=zeros(T,1);
sigma2_hat(1)= alpha0_hat;
for j=2:T
sigma2_hat(j)= alpha0_hat + alpha1_hat* X(j-1) +
beta1_hat*sigma2_hat(j-1);
end
r=zeros(1,T);
for t=1:T
r(t)=X(t)/sigma2_hat(t);
end
B=999;
alpha0_hat_bs=zeros(B, 1);
alpha1_hat_bs=zeros(B, 1);
beta1_hat_bs=zeros(B, 1);
for k=1:B
r_star=r(unidrnd(length(r),1,T));
phi=zeros(T,1);
X_bs=zeros(T,1);
phi(1) = alpha0_hat;
X_bs(1) = phi(1)*r_star(1);
for j = 2:T
```

```
phi(j) = alpha0_hat + alpha1_hat*X_bs(j-1) + beta1_hat*phi(j-1);
X_bs(j) = phi(j)*r_star(j);
end
a1_bs=armax(iddata(X_bs-mean(X_bs)),'na',1,'nc',1,'Maxiter=1',1);
if -a1_bs.parameterVector(2) < 0
beta1_hat_bs(k) = 0.0001;
else
beta1_hat_bs(k) = -a1_bs.parameterVector(2);
end
if -a1_bs.parameterVector(1) - beta1_hat_bs(k) <0
alpha1_hat_bs(k) = 0.0001;
else
alpha1_hat_bs(k) =-a1_bs.parameterVector(1) - beta1_hat_bs(k);
end
if (1-alpha1_hat_bs(k)-beta1_hat_bs(k))*mean(X_bs) <0
alpha0_hat_bs(k)=0.0001;
else
alpha0_hat_bs(k)=(1-alpha1_hat_bs(k)-beta1_hat_bs(k))*mean(X_bs);
end
end
s=sort(alpha0_hat_bs);
r=sort(alpha1_hat_bs);
v=sort(beta1_hat_bs);
CI_alpha0_hat_OLS = [s(25) s(975)]
CI_alpha1_hat_OLS = [r(25) r(975)]
CI_beta1_hat_OLS = [v(25) v(975)]
```

```
alpha0_hat_OLS = alpha0_hat;
alpha1_hat_OLS = alpha1_hat;
beta1_hat_OLS = beta1_hat;
```

References

(1) B. Abraham, N. Balakrishna and R. Sivakumar. Gamma stochastic volatility models. *Journal of Forecasting*, **25**:153-171, 2006.

(2) T. Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, **31**:307-327, 1986.

(3) T. Bollerslev, R.Y. Chou and K.F. Kroner. ARCH modeling in finance: a review of the theory and empirical evidence. *Journal of Econometrics*, **52**:5-59, 1992.

(4) G.E.P. Box, G.M. Jenkins and G.C. Reinsel. *Time Series Analysis: Forecasting and Control*. Wiley, Hoboken, New Jersey, 2008.

(5) E. Carlstein. The use of subseries values for estimating the variance of a general statistic from a stationary time series. *The Annals of Statistics*, **14**:1171-1179, 1986.

(6) M. Carrasco and X. Chen. Mixing and moment properties of various GARCH and stochastic volatility models. *Econometric Theory*, **18**:17-39, 2002.

(7) B. Chen, R.G. Yulia, N. Balakrishna and B. Abraham. Computationally efficient bootstrap prediction intervals for returns and volatilities in ARCH and GARCH processes. *Journal of Forecasting*, **30**:51-71, 2011.

(8) P.K. Clark. A subordinated stochastic process model with finite variance for speculative prices. *Econometrica*, **41**:135-155, 1973.

(9) C. Conrad and M. Karanasos. The impulse response function of the long memory GARCH process. *Economics Letters*, **90**:34-41, 2006.

(10) T. Copeland and D. Galai. Information effects on the bid-ask spread. *Journal of Finance*, **38**:1457-1469, 1983.

(11) V. Corradi and E.M. Iglesias. Bootstrap refinements for QML estimators of the GARCH (1,1) parameters. *Journal of Econometrics*, **144**:500-510, 2008.

(12) D. Easley and M. O'Hara. Time and the process of security price adjustment. *Journal of Finance*, **47**:576-605, 1992.

(13) B. Efron. Bootstrap methods: another look at the jackknife. *The Annals of Statistics*, **7**:1-26, 1979.

(14) R.F. Engle. Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation. *Econometrica*, **50**:987-1008, 1982.

(15) R.F. Engle. The econometrics of ultra-high-frequency data. *Econometrica*, **68**:1-22, 2000.

(16) R.F. Engle. GARCH 101: The use of ARCH/GARCH models in applied econometrics. *Journal of Economic Perspectives*, **15(4)**:157-168, 2001.

(17) R.F. Engle and T. Bollerslev. Modeling the persistence of conditional variances. *Econometric Reviews*, **5**:1-50, 1986.

(18) R.F. Engle and V.K. Ng. Measuring and testing the impact of news on volatility. *Journal of Finance*, **48(5)**:1749-1778, 1993.

(19) R.F. Engle and J. Rosenberg. Hedging options in a GARCH environment: testing the term structure of stochastic volatility models. *NBER*, **4958**, 1994.

(20) R.F. Engle and J.R. Russell. Autoregressive conditional duration: a new model for irregularly spaced transaction data. *Econometrica*, **66**:1127-1162, 1998.

(21) R.F. Engle and J.R. Russell. *Handbook of Financial Econometrics*. North Holland Press: Amsterdam, 2004.

(22) C. Francq and J-M. Zakoian. *GARCH Models - Structure, Statistical Inference and Financial Applications*. Wiley, Chichester, England, 2nd edition, 2010.

(23) D.A. Freedman. Bootstrapping regression models. *The Annals of Statistics*, **9**:1218-1228, 1981.

(24) L.R. Glosten and P.R. Milgrom. Bid, ask and transaction prices in a specialist market with heterogeneously informed traders. *Journal of Financial Economics*, **14**:71-100, 1985.

(25) C. Goodhart and M. O'Hara. High frequency data in financial markets: issues and applications. *Journal of Empirical Finance*, **4**:73-114, 1997.

(26) J. Grammig and K.O. Maurer. Non-monotonic hazard functions and the autoregressive conditional duration model. *The Econometrics Journal*, **3**:16-38, 2000.

(27) A.C. Harvey and N. Shephard. Estimation of an asymmetric stochastic volatility model for asset returns. *Journal of Business and Economic Statistics*, **14**:429-434, 1996.

(28) E. Jacquier, N.G. Polson, and P.E. Rossi. Bayesian analysis of stochastic volatility models. *Journal of Business and Economic Statistics*, **12**:371-389, 1994.

(29) L. Kavalieris, E. Hannan and M. Salau. Generalized Least Squares Estimation of ARMA Models. *Journal of Time Series Analysis*, **24(2)**:165-172, 2001.

(30) S. Kenichi. *Bootstrapping Stationary ARMA-GARCH Models*. Strauss GMBH. Morlenbach, Germany, 1st edition, 2010.

(31) F.H. Koster. *The Bootstrap Approach to Autoregressive Time Series Analysis*. Master's thesis, University of Pretoria, 1999.

(32) S.W. Lee and B.E. Hansen. Asymptotic theory for the GARCH (1,1) quasi maximum likelihood estimator. *Econometric Theory*, **10**:29-52, 1994.

(33) R.Y. Liu. Bootstrap procedures under some non-i.i.d. models. *The Annals of Statistics*, **16**:1696-1708, 1988.

(34) R.Y. Liu and K. Singh. Efficiency and robustness in resampling. *The Annals of Statistics*, **20**:370-384, 1992.

(35) R. L. Lumsdaine. Consistency and asymptotic normality of the quasi maximum likelihood estimator in IGARCH (1,1) and covariance stationary GARCH (1,1) models. *Econometrica*, **64**:575-596, 1996.

(36) A. Lunde. A conjugate gamma model for durations in transactions. Working Paper. Department of Economics, University of Aarhus. 1997.

(37) K. Mayorov. *Modelling Trade Durations with the Birnbaum-Saunders Autoregressive Model*. Master's thesis, McMaster University, 2011.

(38) M. Pacurar. *Three Essays on High Frequency Financial Data and Their Use for Risk Management*. PhD thesis, HEC Montreal, 2006.

(39) T. Pukkila, S. Koreisha and A. Kallinen. The identification of ARMA models. *Biometrika*, **77**:537-548, 1990.

(40) G.E. Tauchen and M. Pitts. The price variability-volume relationship on speculative markets. *Econometrica*, **51**:485-505, 1983.

(41) S.J. Taylor. *Asset Price Dynamics, Volatility, and Prediction*. Princeton University Press, Princeton, 2005.

(42) R.S. Tsay. *Analysis of Financial Time Series*. Wiley, Hoboken, New Jersey, 2nd edition, 2005.

(43) M.Y. Zhang, J.R. Russell and R.S. Tsay. A nonlinear autoregressive conditional duration model with applications to financial transaction data. *Journal of Econometrics*, **104**: 179-207, 2001.