

Bayesian Estimation of QGARCH Models

Congcong Sun

School of Economics and Law, University of Science and Technology Liaoning

Yuming Dong*

School of Economics and Law, University of Science and Technology Liaoning

*E-mail: haomiao816@sohu.com

Abstract—In this paper both Bayesian and Maximum Likelihood (ML) estimation methods are applied to QGARCH model. The Bayesian estimation is based on Metropolis-Hastings (MH) algorithm with an adaptive method. Simulation studies indicate that the method is very efficient for generating uncorrelated Monte Carlo data as the autocorrelation test and some other hypothesis tests demonstrate. All programs are coded in C++, R and SAS. For illustration purpose, the estimation methods are applied to ten years data of Nikkei 225 from 2005-2014.

Index Terms—GARCH, QGARCH; Bayesian estimation; Metropolis-Hasting (MH) algorithm; Markov Chain Monte Carlo method

I. INTRODUCTION

In statistics, signal processing, econometrics and mathematical finance, a time series is a sequence of data points, measured typically at successive time instants spaced at uniform time intervals. Time series models can have many forms and represent different stochastic processes. Among these time series models, Box-Jenkins (1970) models are widely used. Two useful types of Box-Jenkins models are the autoregressive (AR) and the moving average (MA) models. [1] The Box-Jenkins models require both stationary and invertibility. A time series is stationary if the statistical properties such as the mean, the variance of the time series are essentially constant over time. This implies that $E[y_t] = \mu$, $Var(y_t) = \sigma^2$ for all t . Invertibility means that a time series y_t can be written as a function of past observation i.e. y_{t-1}, y_{t-2}, \dots . In practice, before using the Box-Jenkins models, we may use sample autocorrelation function (SAC) and the sample partial autocorrelation function (SPAC) to verify these two assumptions. If a time series is non-stationary, we may transform it into a stationary time series by using some form of differencing transformations (First difference, Second difference). [2] Once a tentative Box-Jenkins model is chosen based on the behavior of SAC and SPAC for y_t , we may check the adequacy of the model by analyzing the residuals obtained by fitting the model. There are many ways to test all model assumption and its adequacy, Box-Pierce statistic or Ljung-Box statistic is used widely in practice. [3]

Although the Box-Jenkins models win big success in application, they have limitations when fitting the times

series which shows volatility clustering property. [4] The volatility clustering property is the tendency for large (small) swings in value of a series to be followed by large (small) swings of random direction. It captures the idea that some markets represent periods of notably high or low volatility. [5] There is a degree of autocorrelation in the riskiness of financial returns. Financial analysts, looking at plots of daily returns such as in Figure 1, notice that the amplitude of the returns varies over time and describe this as “volatility clustering.”

For this reason, Engle (1982) relaxes the constant variance assumption of classic Box-Jenkins models. [6] Engle (1982) introduced a new class of stochastic process called autoregressive conditional heteroscedastic (ARCH). The AR comes from the fact that these models are autoregressive models in squared returns. The conditional comes from the fact that in these models, the next period's volatility is conditioned on information of the current period. Heteroscedasticity means non constant volatility. And a more general class of processes, GARCH (Generalized Autoregressive Conditional Heteroskedastic), is introduced by Bollerslev (1986), allowing for a much more flexible lag structure. The extension of the ARCH process to the GARCH process bears much resemblance to the extension of the standard time series AR process to the general ARMA process and, as argued below, permits a more parsimonious description in many situations. [7] There are many extensions of the GARCH model to include additional properties of the volatility, such as EGARCH (Exponential GARCH) by Nelson (1991), [8] FGARCH (Factor GARCH) by Nerlove (1989), IGARCH (Integrated GARCH) by Nelson (1990) and so on. Here is the formula of a simple GARCH (1,1) model, which is the simplest but often very useful.

$$\begin{cases} y_t = \sigma_t \varepsilon_t, \\ \sigma_t^2 = \omega + \alpha_1 y_{t-1}^2 + \beta_1 \sigma_{t-1}^2, \quad \omega > 0, \quad \alpha_1 \geq 0, \quad \beta_1 \geq 0. \end{cases} \quad (1)$$

It is true (as will be seen in Fig. 2 that the volatility response is higher after negative news (returns), which is known as the leverage effect, first observed by Black (1972). In order to cope with this fact, some models which introduce asymmetry into the volatility response function are proposed. In our paper, we focus on the QGARCH (Quadratic GARCH) model

developed by Sentana (1995), which adds an additional term proportional to a return to the volatility response function. [9]

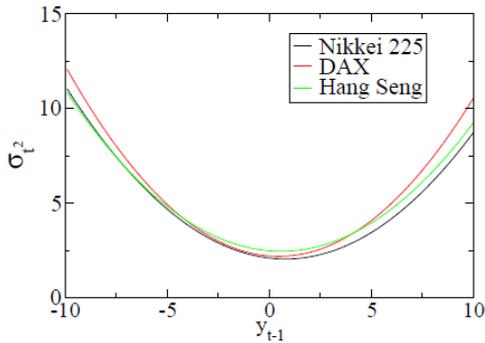


Figure 1. News impact curves of the Nikkei 225, DAX and Hang Seng indexes

Source: Tetsuya Takaishi, “Bayesian inference with an adaptive proposal density for GARCH models”, Computational Finance (q-fin.CP), J. Phys.: Conf. Ser. 221 (2010) 012011

II. QGARCH MODELS

The volatility response function of GARCH model is symmetric under positive or negative observations y_t . However, in the market, some asset return is shown the leverage effect, i.e. the volatility is higher after negative observations than after positive ones. [10] In this study we use the QGARCH (1, 1) model given by

$$\begin{cases} y_t = \sigma_t \varepsilon_t, \\ \sigma_t^2 = \omega + \gamma y_{t-1} + \alpha y_{t-1}^2 + \beta \sigma_{t-1}^2, \quad \omega > 0, \gamma < 0, \alpha \geq 0, \beta \geq 0. \end{cases} \quad (2)$$

where the additional term γy_{t-1} introduces the asymmetry into the model. And a proof of the wide-sense stationary condition of QGARCH model is shown that $E(y_t) = 0$,

$$\text{and the unconditional variance } \text{var}(y_t) = \frac{\omega}{1 - \alpha_1 - \beta_1}.$$

For Bayesian estimation, recall that the likelihood function $L(y|\theta) = P(\text{data} | \text{hypothesis})$ is a function of y_t ($t=1,2,\dots,n$) given that θ is known. We can use Bayes theorem to write $P(\text{hypothesis}|\text{data})$ as follows:

$$P(\text{Hypothesis}|\text{data}) = \frac{P(\text{data}|\text{Hypothesis})P(\text{Hypothesis})}{P(\text{data})}, \quad (3)$$

$P(\text{Hypothesis})$ is called the "Prior Probability" because it reflects prior beliefs about the probability of a given hypothesis. In our study, we use $\pi(\theta)$ to denote the prior probability. In Bayesian analysis, the choice for $\pi(\theta)$ depends on the behavior of the characteristic that the random variable θ represents. $\pi(\theta)$ represents uncertainty about θ before the data is taken into account, the choice for $\pi(\theta)$ is often a subjective assessment. Here

we assume θ follows uniform distribution before the data is concerned, therefore the prior probability density function $\pi(\theta)$ is constant. $P(\text{Hypothesis}|\text{data})$ is usually called the posterior probability, meaning that it reflects the belief about the hypothesis after collecting the data. In our study, we use $\pi(\theta|y)$ to denote the posterior probability.

The posterior distribution replaces the likelihood function as an expression that incorporates all information. If we want to estimate θ and parallel the development of the Maximum Likelihood estimator of θ , we could take as our estimator of θ which maximizes the posterior distribution, that is, estimate θ with the mode of the posterior distribution. The posterior distribution is a distribution function; so we could just as well estimate θ with the median or mean of the posterior distribution. We will use the mean of the posterior distribution as our estimate of θ , and in general we could estimate $\tau(\theta)$ given $Y_1 = y_1, \dots, Y_n = y_n$; that is, take $E[\tau(\theta)|Y_1, \dots, Y_n]$ as our estimate of $\tau(\theta)$. It is a well-known fact that under the squared error loss function, mean of the posterior distribution is the optimal estimator for θ . Since in our case $\tau(\theta) = \theta$, the expectation can be expressed as:

$$E[\tau(\theta)|Y_1, \dots, Y_n] = \int \theta \pi(\theta|y) d\theta = \frac{\int \theta L(y|\theta) \pi(\theta) d\theta}{\int L(y|\theta) \pi(\theta) d\theta} \quad (4)$$

where θ represents vector of parameters and the sign of integration represents multiple integrals.

For a specific simple case, the expected value of parameter can be calculated through the integral directly. But, it seems to be cumbersome when the posterior function is complicated, just like in our case. So, instead of evaluating integral in formula (2.5.3), we evaluate it using MCMC technique described in Section 3.

III. MCMC STIMULATION WITH MH ALGORITHM

Markov chain Monte Carlo (MCMC) methods are a class of algorithms for sampling from probability distributions based on constructing a Markov chain that has the desired distribution as its stationary distribution. The state of the chain after a large number of steps is then used as a sample from the desired distribution. Markov chain Monte Carlo (MCMC) algorithms such as the Metropolis-Hastings (MH) algorithm and the Gibbs sampler have become extremely popular in statistics, as a way of approximately sampling from complicated probability distributions in high dimensions. The development of MCMC algorithms has transformed Bayesian inference by allowing practitioners to sample from complicated posterior distributions. For Equation 2.5.3,

we see that $E[\tau(\theta)|Y_1, \dots, Y_n]$ is the ‘‘posterior mean’’ of $\tau(\theta)$. Let $\phi(\tau) = E[\tau(\theta)|Y_1, \dots, Y_n]$, in Bayesian inference we are interested in estimating $\phi(\tau)$ to obtain Bayes estimate.

MCMC algorithm is to construct a Markov Chain that has the desired distribution as its stationary distribution by using Monte Carlo simulation method. An MCMC algorithm requires a probability distribution $P(\cdot)$ on a state space Ω , a Markov chain on Ω which can be easily executed on a computer, $P(\cdot)$ has to be stationary. $P(\cdot)$ ($i \in \Omega$) is also called the long run proportion of time the process will spend in state i . If $P(\cdot)$ is stationary, the key notion is that the space Ω must be reversible.

Definition. A Markov chain on a state space Ω is reversible with respect to a probability distribution $P(\cdot)$ on Ω if $P(i)P(i, j) = P(j)P(j, i)$, $i, j \in \Omega$, $P(i, j)$ is the transition probability from state i to state j .

The reversibility condition implies that the rate of transition from i to j is the same as the rate of transition from j to i . The reversibility condition implies stationary. Because $P(i) = P(i) \int_{j \in \Omega} P(i, j) = \int_{j \in \Omega} P(i)P(i, j) = \int_{j \in \Omega} P(j)P(j, i)$, which proves stationary. Thus, when constructing an MCMC algorithm, it suffices to create a Markov chain which is reversible with respect to $P(\cdot)$. The simplest way to do so is to use Metropolis-Hastings (MH) algorithm.

Let $\pi_u(\theta) = L(y|\theta)\pi(\theta)$, as a result the posterior density $P(\cdot)$ can be written as

$$P(\theta | \theta \in A) = \frac{\int_A \pi_u(x) dx}{\int_{\Omega} \pi_u(x) dx} \tag{5}$$

where A is subset of state space Ω .

Let $Q(x, \cdot)$ be a Markov chain on p -dimensional state space χ , where x is specific value of X , the Metropolis Hastings algorithm proceeds as follows:

- 1) The target distribution has p.d.f. $P(\theta)$ on χ .
- 2) Suppose $x^{(1)}, x^{(2)}, \dots, x^{(t-1)}$ are generated by using MH algorithm, starting from initial state $x^{(0)}$.
- 3) A proposal distribution with p.d.f. $g(x|x')$ is defined for all $x, x' \in \chi$. For general MH methods, the proposal distribution can be any conditional distribution.
- 4) At step $t-1$, the current state is $x^{(t-1)}$. A candidate state x^* is generated from the current proposal distribution, $x^* \sim g(x^*|x^{(t-1)})$.
- 5) The decision on whether or not to accept x^* as the next state is based on a Bernoulli trial with probability of success $\alpha_t = \alpha(x^{(t-1)}, x^*)$ defined as follow:

$$\alpha_t = \min\left[1, \frac{P(x^*)g(x^{(t-1)}|x^*)}{P(x^{(t-1)})g(x^*|x^{(t-1)})}\right] \tag{6}$$

- 6) If the outcome is success we accept the proposed state and let $x^t = x^*$; if it is not a success, we reject the proposed state and let $x^t = x^{(t-1)}$. More precisely,

$$x^{(t)} = \begin{cases} x^*, & \text{with probability } \alpha_t, \\ x^{(t-1)}, & \text{otherwise} \end{cases} \tag{7}$$

- 7) Replace $t-1$ by t and repeat until we get an acceptable proposed state.

The Markov chain constructed by MH algorithm is reversible, which means $P(x|x')P(x') = P(x'|x)P(x)$. So $P(\cdot)$ is stationary distribution. And this stationary distribution is same as posterior distribution. How should we choose the proposal distribution of $g(x|x')$? There are many different classes of ways of choosing the proposal density, such as symmetric Metropolis Algorithm or independence sampler. The details of how to implement the MH algorithm on GARCH model are provided. In order to facilitate understanding, the simulation can be summarized in following steps:

- (1) First we set an initial value $\theta_0 = (\omega_0, \gamma_0, \alpha_0, \beta_0)$ and $i=1$.
- (2) Then we generate a new value θ_i from a certain proposal density $g(\theta_i|\theta_{i-1})$. In our case, we choose multivariate Student’s t -distribution as our proposal density. And it does not depend on the previous value, i.e. $g(\theta_i|\theta_{i-1}) = g(\theta_i)$. In other word, it can be calculated by the density function of multivariate Student’s t -distribution as follows:

$$g(\theta) = \left[1 + \frac{(\theta - M)' \Sigma^{-1} (\theta - M)}{v}\right]^{-(v+p)/2} \times \frac{\Gamma((v+p)/2) / \Gamma(v/2)}{\text{Det} \Sigma^{1/2} (v\pi)^{p/2}} \tag{8}$$

Where M is vector of expected value of θ , and Σ is covariance matrix of θ .

- (3) We calculate the acceptance probability as
$$\alpha = \min\left[1, \frac{P(\theta_i) g(\theta_{i-1}|\theta_i)}{P(\theta_{i-1}) g(\theta_i|\theta_{i-1})}\right] \tag{9}$$

In our case, $g(\theta_i|\theta_{i-1}) = g(\theta_i)$, above formula can be rewritten as

$$\alpha = \min\left[1, \frac{P(\theta_i) g(\theta_{i-1})}{P(\theta_{i-1}) g(\theta_i)}\right] \tag{10}$$

where $P(\theta_i) = \pi_u(\theta_i) = L(y|\theta_i)\pi(\theta_i)$, $L(y|\theta_i)$ is likelihood function of θ_i . Since we assume

$\pi(\theta_i)$ is uniform distributed which has flat p.d.f., the acceptance probability becomes

$$\alpha = \min \left[1, \frac{L(y|\theta_i) g(\theta_{i-1})}{L(y|\theta_{i-1}) g(\theta_i)} \right]. \tag{11}$$

- (4) Generate $u \sim \text{Uniform}(0,1)$.
- (5) If

$$\begin{cases} u < \alpha, & \text{accept } \theta_i \\ u \geq \alpha, & \text{reject } \theta_i, \text{ then } \theta_i = \theta_{i-1} \end{cases} \tag{12}$$

- (6) Go back to (2) with an increment of $i=i+1$.

The efficiency of the MH algorithm depends on how we choose the proposal density and initial value of parameters. The posterior density of GARCH parameters often shows a Gaussian-like shape. Thus one may choose a density similar to a Gaussian distribution as the proposal density. Also, to cover the fat tails of the posterior density, some researchers attempt to use a multivariate Student's t-distribution instead of Gaussian distribution as the proposal density.

The (p-dimensional) multivariate Student's t-distribution is given by

$$g(\theta) = \left[1 + \frac{(\theta - M)' \Sigma^{-1} (\theta - M)}{\nu} \right]^{-(\nu+p)/2} \times \frac{\Gamma((\nu+p)/2) / \Gamma(\nu/2)}{\text{Det} \Sigma^{1/2} (\nu \pi)^{p/2}} \tag{13}$$

where θ is the vector of parameters and M is expected value of θ , $M_i = E[\theta_i]$. Σ is the covariance matrix of θ . ν is degree of freedom. When $\nu \rightarrow \infty$, the Student's t-distribution converges to the Gaussian distribution. At small ν , Student's t-distribution has a fat-tail.

The idea of adaptive method is to update vector M and covariance matrix Σ adaptively during the process. The steps are as follows:

- (1) First we make a short run by any MCMC method and sample some data on θ .
- (2) Then we estimate mean vector M and covariance matrix Σ from those data.
- (3) By using MH simulation with the proposal density, we accumulate data until we get say 1000 data point.
- (4) Then we recalculate new M and Σ from accumulated data and substitute the previous ones.

In this way, we adaptively change the shape of multivariate Student's t-distribution to fit the posterior density more accurately and fast. An empirical example is given in Section 4.

IV. EMPIRICAL STUDIES

In this Section, we make empirical study based on Nikkei 225 index, which shows the leverage effect and volatility clustering property. The sample period is from 4th January 2005 to 30th December 2014. Let the dependent variable be daily return, denoted by r_t . $r_t = 100(\ln(p_t / p_{t-1}) - \mu)$

where p_t is daily closing settlement price of Nikkei

225, $i = 1, 2, \dots, 2771$. μ is average of $\ln(p_i / p_{i-1})$. Figure 4 shows the time series of the return of the Nikkei 225 index calculated.

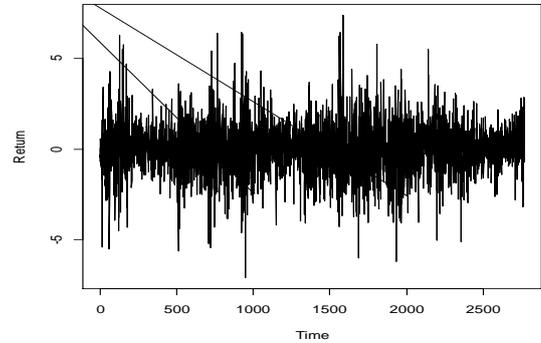


Figure 2. Time series of the return of the Nikkei 225 index

4.1 Bayesian Inference by adaptive MCMC method

When dealing with the Nikkei index, we use the same strategy which can be summarized as follows:

First, use the input data on r_t ($t=1,2,\dots,2771$),

initial value of $\theta_0 = (\omega_0, \gamma_0, \alpha_0, \beta_0)$ and MCMC to make short run and generate 6000 samples from the posterior distribution. Considering the simplification of the problem and reducing the execution time of computer, I choose MH algorithm with symmetric Metropolis Algorithm. So, $g(x|x') = g(x'|x)$, and the acceptance probability simplifies to

$$\alpha(x, y) = \min \left[1, \frac{\pi_u(y)}{\pi_u(x)} \right] \tag{14}$$

Second, each generated sample is of the form of $\theta = (\omega, \gamma, \alpha, \beta)$. Discard first 5000 samples as burn-in process and retain the last 1000 samples. Then, estimate M and Σ , by calculating the mean vector and covariance matrix of these 1000 samples.

Third, having the initial M and Σ , we can use adaptive MCMC method now. Our strategy is to update Σ and M once we accumulate 100 samples based on newly updated M and Σ . In this adaptive MCMC method, we use MH algorithm with multivariate Student's t-distribution as the proposal distribution. We still choose R in this step, because there is on existing package in R which generates multivariate Student's t-distribution.

Last, keep updating M and Σ until we accumulate 5000 acceptable samples, then we can find Bayes estimate = $(1/N) \hat{\theta} = (1/5000) \sum_{k=1}^{5000} \theta_k$

Following the above steps, we finally find the parameter estimates of QGARCH model. The summary of result is in Table 1. And from Figure 5, we can see these parameters converge to the final

value quickly.

TABLE I.
RESULTS OF QGARCH PARAMETERS. SD STANDS FOR STANDARD
DEVIATION

	ω	γ	α	β
Nikkei 225	0.0634563 14	-0.1176120 16	0.0735166 86	0.9000200 55
SD	0.0200451 65	0.0173420 19	0.0105352 09	0.0163187 41

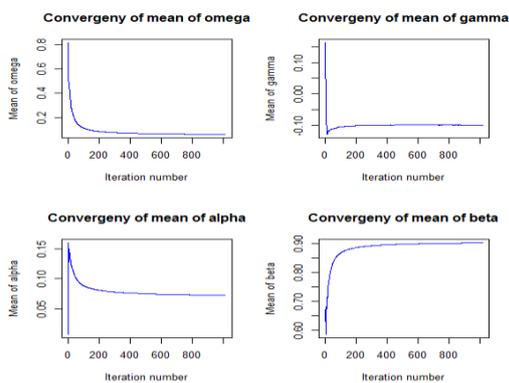


Figure 3. The convergence of the parameters of QGARCH

A Diagnostic test

By using the estimator of parameters we get from Table 1, we could construct a simulated variance sequence of time series r_t , and then we could construct residual ϵ_t . We are using autocorrelation test and Ljung box test with 15 lagged autocorrelation to test these constructed residuals.

First, autocorrelation function test

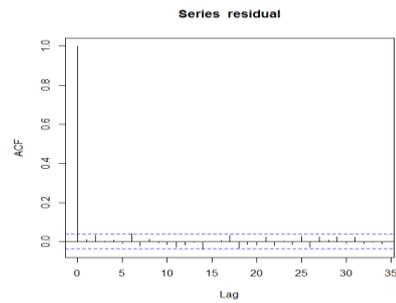


Figure 4. Autocorrelation function test

In statistics, the autocorrelation function (ACF) test is used to identify the degree of autocorrelation in a time series. It measures the correlation between the current and lagged observations of the time series of stock returns. Since the autocorrelation values of all lags are within the range, we could conclude that there is no autocorrelation in residual. This is consistent with the assumption of QGARCH model for ϵ_t .

Second, Ljung Box test

Given an estimate of the parameters for a model, it is desirable to determine if the model adequately explains the variance process. A common approach is to divide each y_t by the estimated standard deviation σ_t for that time point, and square these standardized residuals or we call it constructed ϵ_t^2 . Finally perform a Ljung-Box test on ϵ_t^2 . If the statistic is large or p-value is small, then there is evidence that the model is inadequate. Wong and Li (1995) studied the rank Ljung-Box test in this setting. Often a “Ljung box test” with 15 lagged autocorrelations is used for QGARCH model.

```
> Box.test(residual^2, lag = 15, type = "Ljung-Box")

Box-Ljung test

data: residual^2
X-squared = 13.3932, df = 15, p-value = 0.572
```

Due to the large p-value, we can conclude that the QGARCH model is good fit of the time series data.

V. CONCLUSIONS

In this paper, we introduce background of GARCH and QGARCH models at the beginning. They have been proved to explain the financial time series data successfully. Then, we have performed Maximum Likelihood estimation and the Bayesian inference of the QGARCH model by the MCMC algorithm. The MCMC algorithm was implemented by the MH method with the

adaptive proposal density. The adaptive proposal density is assumed to be the Student’s t-distribution and the distribution parameters are determined by the data sampled by the MCMC simulation. The distribution parameters are updated during the MCMC simulation adaptively to match the posterior density of the model parameters. We have applied our method for Nikkei 225. The diagnosis test results show that our method is very efficient for generating uncorrelated Monte Carlo data.

ACKNOWLEDGEMENTS

The authors really appreciate the support for research work offered by Liaoning College Students Innovation and Entrepreneurship Training Program. Program Number: **201610146004**

REFETENCES

- [1] Gareth O. Roberts and Jeffrey S. Rosenthal (2004), "General state space Markov chains and MCMC algorithms", Vol. 1 (2004) 20-71, ISSN: 1549-5787
- [2] Duarte Queiros S M and Tsallis C (2005), "Bridging the ARCH model for finance and nonextensive entropy". *Europhys. Lett.* 69 893
- [3] Bollerslev, T. (1986), "Generalized Autoregressive Conditional Heteroskedasticity," *Journal of Econometrics*, 31, 307-327.
- [4] Robert. F. Engle. (2001), "The Use of ARCH/GARCH Model in Applied Econometrics". *Journal of Economic Perspectives—Volume 15, Number 4—Fall 2001—Pages 157–168*
- [5] Tetsuya Takaishi, (2010), "Bayesian inference with an adaptive proposal density for GARCH models", *Computational Finance (q-fin.CP), J. Phys.: Conf. Ser.* 221 012011
- [6] Engle, R.F. (1982), "Autoregressive Conditional Heteroskedasticity with Estimates of the Variance of U.K. Inflation," *Econometrica*, 50, 987-1008.
- [7] Duarte Queiros S M and Tsallis C (2005), "On the connection between financial processes with stochastic volatility and nonextensive statistical mechanics". *Eur. Phys. J. B* 48 139
- [8] Nelson D B (1991), "Conditional Heteroskedasticity in Asset Returns: A New Approach", *Econometrica* 59, 347
- [9] Sentana E (1995), "Quadratic ARCH models", *Review of Economic Studies* 62 639
- [10] W.K. Hastings (1970), "Monte Carlo sampling methods using Markov chains and their applications". *Biometrika* 57, 97-109.