MAXIMIZATION OF THE LIKELIHOOD FUNCTION IN FINANCIAL TIME SERIES MODELS

Josip Arnerić[1]

Zoran Babić[2]

Blanka Škrabić[3]

Abstract: *many financial time series such as stock returns or foreign exchange rates, observed on daily basis, have showed stylized facts. These facts include serially uncorrelated returns with zero mean, time-varying variance (heteroscedasticity), leptokurtic distribution of returns and volatility clustering. In empirical research we find that these characteristics can be parametrically described using GARCH(p,q) models (Generalized AutoRegrresive Conditional Heteroscedasticity models). In practice these models are used in forecasting market risk. However, parameter estimation in symmetric GARCH(1,1) model, assuming Gaussian distribution of returns is not that simple.*

Maximum likelihood estimation (MLE) is usually concerned in evaluating the parameters. Analytical solution of maximization of the likelihood function using first and second derivatives is too complex when the variance of innovations is not constant. Therefore, we present usefulness of quasi-Newton iteration procedure in parameter estimation of the conditional variance equation within BHHH algorithm. Namely, the advantage of BHHH algorithm in comparison to the other numerical optimization algorithms will be presented. To simplify optimization procedure algorithm uses the approximation of the matrix of second derivatives (Hessian). Within BHHH algorithm Hessian matrix is approximated according to information identity.

When assumption of normality is unrealistic the estimates are still consistent, but robust standard errors should be used. Solutions of the numerical optimization algorithms are sensitive to the initial values and convergence criteria. Optimization procedure will be illustrated by modeling daily returns of the most liquid stock in first quotation on Zagreb Stock Exchange. In final step, from the evaluated model, prognostic values of expected return and expected standard deviation are estimated. These prognostic values can be used to estimate alternative risk measures, such as Value at Risk (VaR) or Conditional Value at Risk (CVaR). Even so, from estimated GARCH(1,1) model we can reveal the intensity of volatility reaction on past information, and volatility persistence (time for shocks in volatility to die out).

KEY WORDS: LOG-LIKELIHOOD, GARCH MODEL, BHHH ALGORITHM, INFORMATION IDENTITY

1. INTRODUCTION

In econometric modeling parameters estimation is essential for measuring and quantifying different influences between observed variables. These estimations are also relevant for significance testing. For example, in linear regression model in matrix form:

$$Y = X\beta + \varepsilon, \tag{1}$$

the goal is to find such parameter vector β that minimizes the sum of squared residuals:

$$min\left(Y - X\beta\right)^T \left(Y - X\beta\right) = min\left(\varepsilon^T \varepsilon\right), \tag{2}$$

where ε is *n*-dimensional stochastic vector, i.e. vector of unknown deviations from the functional form of the regression model. Therefore the objective function in dependence of unknown parameters is defined as follows:

---

[1] Faculty of Economics, University of Split, Croatia

[2] Faculty of Economics, University of Split, Croatia

[3] Faculty of Economics, University of Split, Croatia

$$f(\beta) = Y^T Y - 2\beta X^T Y + \beta^T X^T X\beta \,. \tag{3}$$

The extremes of a function in equation (3) are simple characterized by its derivatives being equal to zero:

$$-2X^T Y + 2X^T X\beta = 0 \,. \tag{4}$$

Solving equation in (4) for β, k-dimensional vector of estimated parameters is given by:

$$\hat{\beta} = \left(X^T X\right)^{-1} X^T Y \,. \tag{5}$$

Parameters estimation in (5) is obtained my minimizing quadratic function in (3). These estimates are called ordinary least squares estimates (OLS).

Finding extremes for nonquadratic functions is not so easy. For this kind of problem Sir Isaac Newton proposed iterative solution:

- quadratic approximation of nonlinear function in neighborhood of some value x_k and find its extremes
- generate a new local approximation in neighborhood of previous minimum (maximum) and find new extremes

Local quadratic approximation of some nonlinear function $f(x)$ is given by Taylor expansion (according to Mean Value Theorem):

$$f(x) = f(x_k) + f'(x_k)(x - x_k) + \frac{1}{2}(x - x_k) f''(x_k)(x - x_k) \,. \tag{6}$$

Setting the derivatives of function in (6) to zero, and solving for x:

$$f'(x) = f'(x_k) + f''(x_k)(x - x_k) = 0 \,. \tag{7}$$

If the function is quadratic we arrive at the extremes in a single step, as in the case of ordinary least squares estimation. If the function is not quadratic, we must solve for the solution iteratively:

$$x = x_k - \left[f''(x_k)\right]^{-1} \cdot f'(x_k), \tag{8}$$

where $f''(x_k)$ is matrix of second derivatives evaluated at x_k, referred as Hessian matrix (later denoted as H_k), and $f'(x_k)$ is vector of the first derivatives evaluated at x_k, referred as gradient vector (later denoted as g_k). Namely, direction vector, denoted as $d_k = H_k^{-1} g_k$, is a vector describing a segment of a path from the starting point to the solution, where the inverse of the Hessian determines the angle of the direction and the gradient determines its size. However when the function is not well behaved then poorly approximation by the quadratic function results with inaccurate optimum.

2. MAXIMIZATION OF THE LIKELIHOOD FUNCTION

An alternative approach to estimate vector of parameters is to find vector β that maximizes likelihood function. Likelihood function, for linear regression model, is defined as joint probability distribution for observed $y_1, y_2, ..., y_n$.[4] According to the assumption that observations are normally distributed, likelihood function is denoted by:

$$L(\beta) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{\varepsilon_i}{\sigma}\right)^2} \,. \tag{9}$$

[4] Maximal likelihood estimation chooses coefficient estimates that maximize the likelihood of the sample data set being observed.

From above definition joint probability distribution is given as product of all normally distributed variables y_i. This relation is true by assumption that these variables are independent. For practical reasons function in (9) is transformed in a monotone increasing function, by taking it's natural logarithm:

$$ln\,L(\beta) = -\frac{n}{2} ln\left(2\pi\sigma^2\right) - \frac{1}{2\sigma^2}\sum_{i=1}^{n} \varepsilon_i^2 . \tag{10}$$

Speaking statistically, it is easy to take expectations and variance of sums, rather then products. Function defined in (10) is called log-likelihood function. By taking partial derivatives of above function with respect to parameters $\beta_0, \beta_1, ..., \beta_k$ and σ^2, and setting them equal to zero, results in the same estimation of vector β as in OLS case. Estimators given by maximization of log-likelihood function (MLE) are equivalent to OLS estimators if and only if *i.i.d.* assumption (independently and identically distributed variables) is introduced. Speaking statistically, assumption that variables y_i have normal distribution with expectation $X\hat{\beta}$ and constant variance is equivalent to the assumption that variables ε_i have standard normal distribution with zero mean and variance equal to unity; $\varepsilon \sim N(0, 1)$.

However, in financial time series models the assumption of constant variance is unrealistic. Financial time series do not have a constant mean and/or constant variance. Therefore, it is assumed that variance is time-varying (heteroscedasticity). The most common measure of volatility as dispersion in probability distribution is the standard deviation of a random variable. So, it is well-known that returns from financial instruments such as exchange rates, equity prices and interest rates measured over short time intervals, i.e. daily or weekly, are characterized by volatility clustering and high kurtosis.

Models which are used to account daily volatility are GARCH(p,q) models. According to the market efficiency hypothesis (*random walk hypothesis*), the returns are serially uncorrelated with a zero mean and hence unpredictable random variables, but autocorrelation of the squared returns suggests high dependency between them. This means that volatility changes over time and it is conditioned on its past information's. Therefore, assuming that σ_t is time-varying, log-likelihood function can be expressed as:

$$ln\,L(\beta) = -\frac{T}{2} ln(2 \cdot \pi) - \frac{1}{2}\sum_{t=1}^{T} ln\,\sigma_t^2 - \frac{1}{2}\sum_{t=1}^{T}\left(\frac{\varepsilon_t^2}{\sigma_t^2}\right) \tag{11}$$

By taking first derivatives of function (11), and after some rearrangement:

$$\frac{\partial ln\,L(\beta)}{\partial \beta} = \frac{1}{2}\sum_{t=1}^{T}\left[\frac{\varepsilon_t^2}{\sigma_t^2} - 1\right]\frac{1}{\sigma_t^2} \cdot \frac{\partial \sigma_t^2}{\partial \beta}, \tag{12}$$

and setting the system of equations in (12) equal to zero, becomes to complex to solve, i.e. it is difficult to solve it analytically. Therefore, a numerical approach is needed. Note that variance σ_t^2 is described through conditional variance equation according to GARCH(1,1) model, which will be discussed later. Thus, the interest of this paper is to estimate parameters of the GARCH(1,1) model by maximization of the log-likelihood function using numerical optimization procedure.

Before presenting numerical optimization procedure in maximization problem of log-likelihood function some properties of likelihood function will be introduced.

It can be noticed that $ln\,L(\beta)$ is always negative, since the likelihood is a probability between 0 and 1 and the *ln* of any number between 0 and 1 is negative. Numerically, the maximum can be found by "walking up" the likelihood function until no further increase can be found. The researcher specifies starting values β_s. Each iteration moves to a new value of the parameters at which $ln\,L(\beta)$ is

higher than at the previous step. If we denote the current value at iteration k by β_k, the question is: what is the best step we can take next, i.e. what is the best value for β_{k+1}?

To determine the best value of β_{k+1}, a second-order Taylor's approximation of $ln\,L(\beta_{k+1})$ around $ln\,L(\beta_k)$ is used:

$$ln\,L(\beta_{k+1}) = ln\,L(\beta_k) + (\beta_{k+1} - \beta_k)^T g_k + \frac{1}{2}(\beta_{k+1} - \beta_k)^T H_k (\beta_{k+1} - \beta_k). \tag{13}$$

Now we find the value of β_{k+1} that maximizes approximation in (13):

$$\begin{aligned} &\frac{\partial ln\,L(\beta_{k+1})}{\partial \beta_{k+1}} = g_k + H_k(\beta_{k+1} - \beta_k) = 0 \\ &H_k(\beta_{k+1} - \beta_k) = -g_k \\ &\beta_{k+1} - \beta_k = H_k^{-1} g_k \\ &\beta_{k+1} = \beta_k + \left(-H_k^{-1}\right) g_k \end{aligned} \quad . \tag{14}$$

The Newton procedure uses this formula. The step from the current value of β_k to the new value is $\left(-H_k^{-1}\right)g_k$, that is, the gradient vector multiplied by the negative of the inverse of the Hessian. The negative of this negative Hessian is positive and represents the degree of curvature. It means that, $-H_k$ is the positive curvature, assuming that log-likelihood function is globally concave. Therefore, each step is the slope of the log-likelihood function divided by its curvature. The curvature determines how large a step is made. If the curvature is great, meaning that the slope changes quickly, and the maximum is likely to be close, a small step is taken.

The scalar λ is introduced in the Newton iterative formula to assure that each step of the procedure provides an increase in $ln\,L(\beta)$. The adjustment is performed separately in each iteration:

$$\beta_{k+1} = \beta_k + \lambda_k \left(-H_k^{-1}\right) g_k \,. \tag{15}$$

The vector $\left(-H_k^{-1}\right)g_k$ is called the direction, denoted as d_k, and λ is called the step size. Classical modified Newton iterative procedure specified in (15) is often referred as Newton-Raphson algorithm when Hessian is determined analytically. Even so, calculation of the Hessian is usually computation-intensive, i.e. analytical Hessian is rarely available. Therefore, alternative to calculation of inverse Hessian matrix is it's approximation, which is related to the group of techniques known as quasi-Newton methods.

3. QUASI-NEWTON ITERATIVE PROCEDURES

The quasi-Newton methods that build up an approximation of the inverse Hessian are often regarded as the most sophisticated for solving unconstrained problems. Even so, taking expectation of the inverse Hessian is essential for variance and covariance estimates in econometric modeling.

Let

$$p_k = \beta_{k+1} - \beta_k \tag{16}$$

be the change in the parameters in the current iteration, and

$$q_k = g_{k+1} - g_k \tag{17}$$

be the change in the gradients. Then an estimate of the Hessian in the next iteration H_{k+1} would be the ratio of change in the gradient to the change in the parameters. This is called quasi-Newton condition.

There are many solutions to the quasi-Newton condition described above. Initial Hessian matrix is usually chosen as identity matrix which is updated by update formula. In classical modified quasi-Newton iterative procedure, assuming minimization problem:

$$\beta_{k+1} = \beta_k - \lambda_k \left(H_k^{-1}\right) g_k , \tag{18}$$

rank two updates are the most widely used. Earliest update formula for constructing the inverse Hessian was originally proposed by Davidon (1959) and later developed by Fletcher and Powell (1963). DFP update formula has nice property: for a quadratic objective function, it simultaneously generates the directions of the conjugate gradient method while constructing the inverse Hessian. DFP update formula for inverse Hessian is given by:

$$H_{k+1}^{-1} = H_k^{-1} + \frac{p_k p_k^T}{p_k^T q_k} - \frac{H_k^{-1} q_k q_k^T H_k^{-1}}{q_k^T H_k^{-1} q_k} . \tag{19}$$

According to Broyden-Fletcher-Goldfarb-Shanno (1970), update formula is given by:

$$H_{k+1}^{-1} = H_k^{-1} + \left(\frac{1 + q_k^T H_k q_k}{q_k^T p_k} \right) \frac{p_k p_k^T}{p_k^T q_k} - \frac{p_k q_k^T H_k + H_k q_k p_k^T}{q_k^T p_k} . \tag{20}$$

Weighted combinations of these formulas leads to a whole collection of updates:

$$H_k = (1-\alpha) H_k^{DFP} + \alpha \cdot H_k^{BFGS} \qquad for \quad 0 \le \alpha \le 1 , \tag{21}$$

according to parameter α.

Numerical experiments have shown that BFGS formula is superior over DFP formula. Hence BFGS is often preferred. Both methods have properties that guarantee relative fast convergence under standard requirements. i.e. the objective function is twice differentiable and Hessian matrix is positive (negative) definite, depending on convexity (concavity) of the objective function.

Log-likelihood function being concave means that its Hessian is negative definite at all values of β's. If H is negative definite, then H^{-1} is also negative definite. Suppose the log likelihood function has regions that are not concave. In these areas, the classical Newton procedure can fail to find an increase. If the function is convex at β_k, then the Newton procedure moves in the opposite direction to the slope of the log-likelihood function and $-H_k^{-1}$ is positive definite. Therefore, classical Newton procedure (i.e. Newton-Rapshon algorithm) has two main disadvantages. First, calculation of the Hessian is computation-intensive. So, procedures that avoid calculating the Hessian at each iteration can be much faster. Second, the procedure does not guarantee an increase in each step if the log-likelihood function is not globally concave. When $-H_k^{-1}$ is not positive definite, an increase is not guaranteed. Therefore, other approaches use approximations to the Hessian to overcome stated disadvantages. The methods differ in the form of the approximation, previously described.

4. INFORMATION IDENTITY AND OUTER PRODUCT OF GRADIENT

Berdnt, Hall, Hall and Hausman (1974) proposed using information identity in the numerical search for the maximum of the log-likelihood function. In particular, iterative procedure is defined as:

$$\begin{aligned} \beta_{k+1} &= \beta_k + \lambda_k \cdot d_k \\ d_k &= -H_k^{-1} g_k \\ -H_k &= \sum_{t=1}^{T} g_t g_t^T \\ g_k &= \sum_{t=1}^{T} g_t \end{aligned} \quad . \tag{22}$$

According to the relations in (22) information identity means that the negative of the expected Hessian at the true parameters is equal to the covariance matrix of the first derivatives. In other words, it means that negative Hessian can be approximated as outer product of gradient (OPG).

There are two advantages using BHHH algorithm in comparison to previously described quasi-Newton algorithms:

- toward information identity approximation, negative Hessian is faster to calculate
- approximated negative Hessian is necessarily positive definite, and therefore guaranteed to provide an increase in log-likelihood in each iteration, even in convex portions of the function

By the law of large numbers it can be shown:

$$\lim_{t\to\infty} Pr\left(\left|\hat{\beta} - \beta_p\right| > \varepsilon\right) = 0\,, \tag{23}$$

where β_p is vector of true parameter values (from population). Relation (23) describes asymptotic property of estimated vector $\hat{\beta}$. It means that $\hat{\beta}$ is consistent estimator, i.e. $\hat{\beta}$ converges to β_p in probability for every $\varepsilon > 0$. Also it can be shown, by the central limit theorem, that the asymptostic distribution of $\hat{\beta}$ is multivariate normal with mean vector β_p and variance matrix equal to inverse of negative expected Hessian:

$$Var\left(\hat{\beta}\right) = \left[-E\left(H\left(\beta_p\right)\right)\right]^{-1}. \tag{24}$$

If vector β_p is replaced with it's estimation, we gets:

$$Var\left(\hat{\beta}\right) = \left[-H\left(\hat{\beta}\right)\right]^{-1}. \tag{25}$$

Variance-covariance matrix in expression (25) can be calculated in other way:

$$Var\left(\hat{\beta}\right) \approx \left[Var\left(g\left(\beta_p\right)\right)\right]^{-1}. \tag{26}$$

By taking:

$$g\left(\beta_p\right) = g_1\left(\beta_p\right) + g_2\left(\beta_p\right) + \ldots + g_T\left(\beta_p\right), \tag{27}$$

the unbiased estimation of variance of gradient can be calculated as:

$$Var\left(g\left(\beta_p\right)\right) \approx \frac{T}{T-1}\sum_{t=1}^{T}\left(g_t\left(\beta_p\right)\right)\left(g_t\left(\beta_p\right)\right)^T. \tag{28}$$

Therefore, by replacing it's estimates it is valid:

$$Var\left(\hat{\beta}\right) = \frac{T}{T-1}\left(\sum_{t=1}^{T} g_t\left(\hat{\beta}\right) g_t\left(\hat{\beta}\right)^T\right)^{-1}. \tag{29}$$

Approximation of Hessian, according to BHHH algorithm is result in identity in relation (29). Identity in (29) is called information identity, which means that negative of the expected Hessian is equal to OPG. In other words, information identity means that variance-covariance matrix of first derivatives in (28) is equal to negative matrix of expected values of second derivatives, when $T \to \infty$.[5]

[5] When assumption of Gaussian white noise process of returns is not realistic, the estimates given my maximization of the likelihood function are called quasi-maximum likelihood estimates (QMLE), and the robust standard errors should be used. QMLE estimators are consistent and asymptotic normally.

After presentation of some properties of OPG estimators, numerical optimization procedure of BHHH algorithm could be summarized in following steps:

1. determine initial vector of parameters β_s, and convergence criteria $tol > 0$
2. at current iteration calculate a direction vector $d_k = \left[-H(\beta_k)\right]^{-1}$, while $-H(\beta_k)$ is calculated by the outer of the gradients [6]
3. calculate a new vector $\beta_{k+1} = \beta_k + \lambda d_k$, where λ is scalar. Start with $\lambda = 1$. If $f(\beta_k + d_k) > f(\beta_k)$ try with $\lambda = 2$. If $f(\beta_k + 2d_k) > f(\beta_k + d_k)$ try with $\lambda = 3$, etc. until lambda is found for which $f(\beta_k + \lambda d_k)$ is in maximum [7]
4. if convergence criteria is satisfied algorithm stops, if not repeat steps from 2 to 4

5. PARAMETER ESTIMATION IN SIMPLE GARCH(1,1) MODEL

The issue of modeling returns accouting for time-varying volatility has been widely analyzed in many financial econometrics literatures.[8] Since the introduction by Engle (1982) of the ARCH(p) (*Autoregressive Conditional Heteroscedasticity*) model and it's generalization, i.e. GARCH(1,1) model by Bollerslev (1986) a wide range of extensions and modifications have been developed. However, the interest of this paper is not presenting different volatility models, but investigating numerical optimization methods in parameter estimation of these models.

When modeling financial time series two equations are introduced: conditional expectation equation and conditional variance equation. The expected value of return series is calculated from the simple linear regression model usually taking constant as regressor. If there is significant autocorrelation in returns, best fitted ARMA(p,q) models are usually used, following Box-Jenkins procedure.

It has been shown that ARCH(p) process with infinite number of parameters is equivalent to generalized ARCH process, i.e. GARCH(p,q) process which is very well approximated by simple GARCH(1,1). As the time lag increases in an ARCH(p) model it becomes more difficult to estimate parameters. Besides it is recommended to use parsimonious model as GARCH(1,1) that is much easier to identify and estimate. Therefore, GARCH(1,1) model is used to parametrically describe conditional variance. Equations of conditional expectation and conditional variance of returns, according to simple GARCH(1,1) model, are given:

$$\begin{aligned} r_t &= \mu + \varepsilon_t; \quad \varepsilon_t = u_t \sqrt{\sigma_t^2} \\ \sigma_t^2 &= \alpha_0 + \alpha_1 \cdot \varepsilon_{t-1}^2 + \beta_1 \cdot \sigma_{t-1}^2 \end{aligned} . \tag{30}$$

In above relations Engle foundation is multiplicative structure of innovation process $\varepsilon_t = u_t \sqrt{\sigma_t^2}$, assuming $u_t \sim i.i.d.\ N(0, 1)$.

Before presentation of concrete application of described algorithms in maximization of log-likelihood function with respect to parameters μ, α_0, α_1 and β_1, time series of Pliva's stock prices and continuously compounded returns are shown on figure 1.

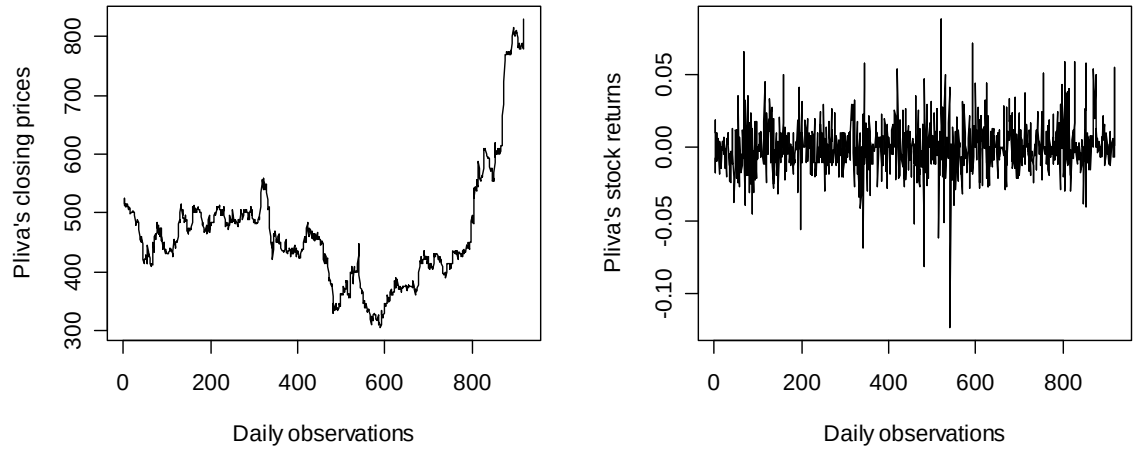
---

[6] If $-H(\beta_k)$ is not invertible, substitute $-H(\beta_k)$ by identity matrix. This is called "steepest ascent".

[7] If $f(\beta_k + \lambda d_k) \leq f(\beta_k + d_k)$ then lambda is reduced (procedure is backing up).

[8] Prices of observed stock are transformed into compound returns by taking logs, i.e. $r_t = ln(P_t) - ln(P_{t-1})$.

Figure 1: Pliva's closing prices and Pliva's stock returns from 1 September 2003 to 4 May 2007, daily observed on Zagreb Stock Exchange



In table 1. some results of iterative optimization procedure according DFP algorithm are presented within parameters estimation of GARCH(1,1) model using Pliva's stock returns on daily basis (including 919 trading days). Pliva stocks where chosen for modeling as the most liquid stock on Zagreb Stock Exchange (ZSE) according to trading volume and capitalization. It is worth to mention that Pliva is leading Pharmacy Company in Central and Eastern Europe, which is listed on London Stock Exchange from 1996.

Table 1: Iterative procedure according DFP algorithm in maximization of log-likelihood function in GARCH(1,1) model of Pliva stock returns [9]

```
Iteration 0:
------------------------------------------------------------------------------
Coefficient vector:
  eq1(cons):   ARCH(L1):  GARCH(L1): GARCH(cons):
y1   .0005299   .0174851  -2.12e-10    .000318
                                                 log likelihood =  2396.6145
Gradient vector (length =  427.143):
r1  -2.08e-09   427.1426  -5.45e-06   .5542549

Step (length =  427.143):
r1  -2.08e-09   427.1426  -5.45e-06   .5542549

b + step -> new b:
y1   .0005299   427.1426  -5.45e-06   .5545729
                                                 log likelihood = -657.72384
                                                      (initial step bad)
(1) Reducing step size (step length = 213.5715) -> new b:
y1   .0005299   213.5713  -2.73e-06   .2774454
                                                 log likelihood =   -339.656

(2) Reducing step size (step length = 106.7858) -> new b:
y1   .0005299   106.7857  -1.36e-06   .1388817
                                                 log likelihood = -22.018965
 ⋮
(14) Reducing step size (step length = .0260707) -> new b:
y1   .0005299   .0260707  -3.33e-10   .0003518
                                                 log likelihood =  2399.5426
```

[9] All empirical results are obtained using Stata 8.0 software package.

```
Iteration 1:
------------------------------------------------------------------------------
Coefficient vector:
  eq1(cons):   ARCH(L1):  GARCH(L1): GARCH(cons):
y1   .0005299   .0260707  -3.33e-10   .0003518
                                                 log likelihood =  2399.5426
Gradient vector (length = 180014.1):
r1  -272.9569   131.2554  -64.59978  -180013.8

Step (length = 427.1361):
r1   -.001905   427.1355  -.0004563  -.7020888

b + step -> new b:
y1  -.0013751   427.1616  -.0004563   -.701737
                                                 log likelihood =          .
                                                         (initial step bad)
(1) Reducing step size (step length = 213.5681) -> new b:
y1  -.0004226   213.5938  -.0002282  -.3506926
                                                 log likelihood =          .
 ⋮
(13) Reducing step size (step length = .0521406) -> new b:
y1   .0005297   .0782113  -5.60e-08   .0002661
                                                 log likelihood =  2413.1593
------------------------------------------------------------------------------
 ⋮
 ⋮
 ⋮
Iteration 16:
------------------------------------------------------------------------------
Coefficient vector:
  eq1(cons):   ARCH(L1):  GARCH(L1): GARCH(cons):
y1  .0001779  .2112905  .5785378  .0000614
                                                 log likelihood =  2425.0481
Gradient vector (length = 902.6982):
r1  -3.249362  -.1809387  -.2700238  -902.6923

Step (length = .0001644):
r1  -9.14e-07  -.0000634   .0001517  -6.91e-08

b + step -> new b:
y1   .000177  .2112272  .5786895  .0000613
                                                 log likelihood =  2425.0481
                                                        (initial step good)
(1) Stepping forward (step length = .0000206) -> new b:
y1  .0001769  .2112193  .5787085  .0000613
                                                 log likelihood =  2425.0481

(2) Stepping forward (step length = .0000411) -> new b:
y1  .0001769  .2112193  .5787085  .0000613
                                                 log likelihood =  2425.0481
                                                       (ignoring last step)
------------------------------------------------------------------------------
Iteration 17:
------------------------------------------------------------------------------
Coefficient vector:
  eq1(cons):   ARCH(L1):  GARCH(L1): GARCH(cons):
y1  .0001769  .2112193  .5787085  .0000613
                                                 log likelihood =  2425.0481
Gradient vector (length = 54.24553):
r1   .1479717  -.0106831  -.0170939  -54.24533
```

From table 1. it can be shown that procedure starts with calculating gradient vector with respect to initial parameter values given by ordinary least squares estimation. At each iteration the step size is reduced (stepping backward) or increased (stepping forward) in purpose to calculate "new b" for which the maximum of log-likelihood function is increased the most. Step size is reduced when the

initial step is "bad", and it is increased when the initial step is "good". Procedure stops at last iteration when a convergence criterion is satisfied and the last step is "ignored". In above example if a relative change in maximum likelihood between two successive iterations is less than given tolerance (0.001) then a convergence criterion is met.

Table 2: Summary of GARCH model estimation with significance testing (using OPG estimation of standard errors) and confidence intervals

```
Log likelihood =  2425.048

------------------------------------------------------------------------------
             |                 OPG
plivaret     |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
_cons        |   .0001769   .0005136     0.34   0.731    -.0008303    .0011828
-------------+----------------------------------------------------------------
arch         |
          L1 |   .2112265   .0323101     6.54   0.000      .148418    .2750712
garch        |
          L1 |   .5786619   .1154509     5.01   0.000      .349861    .8024201
cons         |   .0000613    .000031     1.98   0.048     1.10e-06    .0001228
```

It table 2. estimation of parameters (according to BHHH algorithm) are presented as well as standard errors of the estimates and significance tests. These standard errors are obtained from covariance matrix which is equal to the inverse of the information matrix, approximated as outer product of gradient with respect to estimated parameters at the last iteration. From above results GARCH(1,1) model can be write down:

$$\begin{aligned} r_t &= 0.0001769 \\ \sigma_t^2 &= 0.0000613 + 0.2112265 \cdot \varepsilon_{t-1}^2 + 0.5786619 \cdot \sigma_{t-1}^2 \end{aligned} . \tag{31}$$

Speaking statistically, all estimated parameters are significant at empirical *p*-value less then 5%, except constant term in conditional expectation equation of returns. Also sum of parameters $\alpha_1 + \beta_1$, according to equation (31), indicate that there is persistence volatility, i.e. conditional variance decays slowly, still far from long-memory model. Also parameter α_1 detects high intensity reaction of volatility on past information.

Because the sum of parameters $\alpha_1 + \beta_1$ is less then unity (0.7899) the stationary condition of estimated model is satisfied. If stationary condition is satisfied unconditional long-term variance can be calculated:

$$\overline{\sigma}^2 = \frac{\alpha_0}{1-(\alpha_1 + \beta_1)} = 0.0003 . \tag{32}$$

From first equation in (31) it can be expected the average return of Pliva stocks of 0.02% with average deviation of 1.73% in long term according to (32).

6. INSTEAD OF CONCLUSION

In theory the maximum of log-likelihood occurs when the gradient vector is zero. Namely, in practice the calculated gradient vector is never exactly zero, but can be very close. Therefore $g_k^T (-H_k)^{-1} g_k$ is often used to evaluate convergence. If inequality:

$$g_k^T (-H_k)^{-1} g_k < 0.0001 \tag{32}$$

is satisfied, the iterative process stops and the parameters at current iteration are considered as estimates. However, small changes in parameter values, with small increases in log-likelihood function, from one iteration to the next iteration could be evidence that convergence has been achieved. Even so, small changes in β_k and $lnL(\beta_k)$ accompanied by a gradient vector that is not close to zero indicate that we are not effective in finding the maximum.

To investigate if local maximum is the global optimum we should use different starting values and observe whether convergence occurs at the same parameter values. Empirical research has showed that initial vector of parameters as null-vector is not appropriate. Therefore, an ordinary least squares estimates (OLS) are taken into account as initial values.

In table 3.-5. summary results of parameters estimation according to BHHH, BFGS and DFP algorithms, with different convergence criteria are presented.

Table 3: Summary results of parameters estimation when a convergence criterion is satisfied if the relative change in maximum likelihood between two successive iterations is less than 0.001

Parameter	BHHH	BFGS	DFP
μ	0.0001755	0.0001772	0.0001769
α_0	0.0000614	0.0000617	0.0000613
α_1	0.2112189	0.2133791	0.2112193
β_1	0.5786991	0.5772437	0.5787085
Iter #	15	16	15
Log-likelihood	2425.048	2425.048	2425.048

Table 4: Summary results of parameters estimation when a convergence criterion is satisfied if the relative change in parameter values between two successive iterations is less than 0.0001

Parameter	BHHH	BFGS	DFP
μ	0.0001769	0.0001772	0.0001762
α_0	0.0000613	0.0000617	0.0000620
α_1	0.2112193	0.2133791	0.2117446
β_1	0.5787085	0.5772437	0.5761406
Iter #	15	16	18
Log-likelihood	2425.048	2425.048	2425.048

Table 5: Summary results of parameters estimation when a convergence criterion is satisfied if the gradient of log-likelihood function in current iteration is less than 0.0001

Parameter	BHHH	BFGS	DFP
μ	0.0001769	0.0001769	0.000177
α_0	0.0000613	0.0000613	0.0000613
α_1	0.2112265	0.2112292	0.2112522
β_1	0.5786619	0.5786426	0.5785453
Iter #	16	20	25
Log-likelihood	2425.048	2425.048	2425.048

First of all, from results in tables 3-5 it can be noticed that iterative optimization procedure does not have much influence on constant terms in both equations, respectively μ and α_0, while the changes of parameters α_1 and β_1 differs between algorithms. Also, these differences disappear when a convergence criterion based on gradient approximately close to zero is used, in comparison to the other convergence criteria. When this convergence criterion is used, it can be perceive, that more iterations are needed.

Namely, BHHH algorithm has approved to be faster according to number of iterations, with much stable parameter values according to different convergence criteria. Even so, convergence problem may arise, because the more parameters in the model are entered the "flatter" the log-likelihood function becomes, and therefore the more difficult it is to maximize.

REFERENCE LITERATURE

Alexander, C. (2001), *Market Models: A Guide to Financial Data Analysis*, John Wiley & Sons Ltd., New York.

Bazarra, M. S., Sherali H. D. and Shetty, C. M. (1993), *NonlinearProgramming - Theory and Algorithms* (second edition), John Wiley & Sons Inc., New York.

Berndt, E., Hall, B., Hall, R. and Hausman, J. (1974), "Estimation and Inference in Nonlinear Structural Models", Annals of Social Measurement*,* Vol. 3., p. 653-665.

Enders, W. (2004), *Applied Econometric Time Series* (second edition), John Wiley & Sons, New York.

Gould, W., Pitblado, J. and Sribney, W. (2006), *Maximum Likelihood Estimation with Stata* (third edition), College Station, StataCorp LP.

Neralić, L. (2003), *Uvod u Matematičko programiranje 1,* Element, Zagreb.

Petrić, J. and Zlobec, S. (1983), *Nelinearno programiranje,* Naučna knjiga, Beograd.

Posedel, P. (2005), "Properties and Estimation of GARCH(1,1) Model", Metodološki zvezki, Vol. 2, No. 2, p. 243-257.

Schoenberg R. (2001), "Optimization with the Quasi-Newton Method", Aptech Systems, Inc., Valley WA, working paper, p. 1-9.

Shanno, D. F. (1970), "Conditioning of quasi Newton methods for function minimization", Mathematics of Computation*,* No. 24., p. 145-160.