



HAL
open science

AdaVol: An Adaptive Recursive Volatility Prediction Method

Nicklas Werge, Olivier Wintenberger

► **To cite this version:**

Nicklas Werge, Olivier Wintenberger. AdaVol: An Adaptive Recursive Volatility Prediction Method. *Econometrics and Statistics*, 2022, 23, pp.2452-3062. 10.1016/j.ecosta.2021.01.004 . hal-02733439v3

HAL Id: hal-02733439

<https://hal.science/hal-02733439v3>

Submitted on 25 Jan 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

AdaVol: An Adaptive Recursive Volatility Prediction Method

Nicklas Werge^{a,*}, Olivier Wintenberger^a

^a*LPSM, Sorbonne Université, 4 place Jussieu, 75005 Paris, France*

Abstract

Quasi-Maximum Likelihood (QML) procedures are theoretically appealing and widely used for statistical inference. While there are extensive references on QML estimation in batch settings, it has attracted little attention in streaming settings until recently. An investigation of the convergence properties of the QML procedure in a general conditionally heteroscedastic time series model is conducted, and the classical batch optimization routines extended to the framework of streaming and large-scale problems. An adaptive recursive estimation routine for GARCH models named AdaVol is presented. The AdaVol procedure relies on stochastic approximations combined with the technique of Variance Targeting Estimation (VTE). This recursive method has computationally efficient properties, while VTE alleviates some convergence difficulties encountered by the usual QML estimation due to a lack of convexity. Empirical results demonstrate a favorable trade-off between AdaVol's stability and the ability to adapt to time-varying estimates for real-life data.

Keywords: volatility models, quasi-likelihood, recursive algorithm, GARCH, prediction method, stock index

1. Introduction

Time series analysis has attracted much attention in the last three decades. A central aspect of time series analysis is modeling heteroscedasticity of the conditional variance, e.g., volatility clustering in financial time series. Some well-known models incorporating this feature are the AutoRegressive Conditional Heteroscedasticity (ARCH) model and the Generalized ARCH (GARCH) model introduced by Engle (1982) and Bollerslev (1986), respectively. Many reasons can explain these models' success; they constitute a stationary time series model with a time-varying conditional variance, and secondly, they may model time series with heavier tails than the Gaussian ones, which often occurs in financial time series.

Quasi-Maximum Likelihood (QML) estimation is widely used for statistical inference in GARCH models due to their appealing theoretical nature and tolerance to overdispersion, which is often observed in empirical data. This paper studies the Quasi-Maximum Likelihood Estimator (QMLE) for the broader class of conditionally heteroscedastic time series models of multiplicative form given by

$$X_t = h_t(\theta_0)Z_t, \quad t \in \mathbb{Z}, \quad (1.1)$$

where θ_0 is the true underlying parameter vector, (Z_t) is a sequence of i.i.d. random variables with $\mathbb{E}[Z_0] = 0$ and $\mathbb{E}[Z_0^2] = 1$, and the (non-negative) volatility process $(h_t)_{t \in \mathbb{Z}}$ is defined as

$$h_t(\theta) = g_\theta(X_{t-1}, \dots, X_{t-p}, h_{t-1}(\theta), \dots, h_{t-q}(\theta)), \quad p, q \geq 0. \quad (1.2)$$

Suppose that the parameter set $\Theta \subset \mathbb{R}^d$ and $\{g_\theta | \theta \in \Theta\}$ denotes the (finite) parametric family of non-negative functions on $\mathbb{R}^p \times [0, \infty)^q$ satisfying certain regularity conditions. We also require that h_t is \mathcal{F}_{t-1} -measurable for all $t \in \mathbb{Z}$, where $\mathcal{F}_t = \sigma(Z_k : k \leq t)$ denotes the σ -field generated by the random variables $\{Z_k : k \leq t\}$.

The stability of model (1.1)-(1.2) is accomplished under the assumption that g_θ is a contraction. This condition is a random Lipschitz coefficient condition, where the Lipschitz coefficient has a negative logarithmic moment. The notion of contractivity is clarified in Straumann and Mikosch (2006) where they study QML inference of general conditionally heteroscedastic models with emphasis on the approximation (\widehat{h}_t) of the stochastic volatility (h_t) .

QML estimation of the parameters in the class of conditionally heteroscedastic time series models has been studied frequently in recent years, see e.g., Berkes et al. (2003), Francq and Zakoian (2004), Straumann and Mikosch (2006), and Wintenberger (2013). However, all these references consider iterative estimation, where one assembles a batch of data and afterward performs the statistical inference. Thus, one evaluates an objective function consisting of a sum of n loss terms. Each iteration would then have a cost of $O(nd)$, making the recursion cost $O(mnd)$, where m is the number of iterations. As the amount of data grows, these optimizers become prohibitively expensive and increasingly computationally inefficient. Moreover, iterative optimizers become unsuitable for streaming settings where we are modeling and predicting data as they arrive.

Many financial practices, such as banks, asset managers, and financial services institutes, find themselves estimating thousands of volatility models every day for risk and pricing purposes. In addition, the sampling of financial time series is increasingly at high frequency. Therefore, recursive procedures

*Corresponding author

Email addresses: nicklas.werge@upmc.fr (Nicklas Werge), olivier.wintenberger@upmc.fr (Olivier Wintenberger)

must undoubtedly be advantageous since one only processes observations once. In recursive QML estimation, we update the previous QML estimate with the new observations at time t in order to produce the QML estimate of the parameters at time t .

Thus, in modern statistical analysis, it is becoming increasingly common to work with streaming data where one observes only a group of observations at a time. Naturally, this has led to an expanded interest in time-scalable recursive estimation procedures with a cost of only $O(d)$ computations per recursion, e.g., see Bottou and Bousquet (2007). However, there has only been given a little amount of attention to recursive estimation in conditionally heteroscedastic time series models.

Dahlhaus and Subba Rao (2007) presented a recursive method for estimating the parameters of an ARCH process. Under sufficient assumptions on the underlying process, Aknouche and Guerbyenne (2006) showed consistency of their recursive least squares method for GARCH processes, and Kierkegaard et al. (2000) also developed a recursive estimation method for GARCH processes supported by empirical evidence. Convergence analysis of the recursive QML estimator for GARCH processes based on stochastic approximations with Markovian dynamics using a resetting mechanism has been previously presented (Gerencsér et al. (2010)). A self-weighted recursive estimation algorithm for GARCH models was proposed in Cipra and Hendrych (2018) with a robustification in Hendrych and Cipra (2018). However, none of the above references mention problems with convexity or address the obstacles that may occur when the true parameter θ_0 is close to the boundary of the parameter space.

The difficulty of estimating time-varying parameters of statistical models increases in the setting of streaming data. To sustain computational efficiency and be adaptive to changes in the estimates, one may decrease the number of observations in each iteration in the optimization procedure, which may decrease the stability of the statistical inference. We propose a natural adaptation of the QML method, relying on stochastic approximations combined with the Variance Targeting Estimation (VTE) technique, which we call AdaVol. This recursive method is time-scalable and memory-efficient, as it only requires the previous estimate to process new observations, and it only needs to treat the observations once. We present empirical evidence that AdaVol achieves a favorable trade-off between adaptability and stability.

The rest of the paper is organized as follows: Section 2 introduces the QML procedure for the general class of conditionally heteroscedastic time series models of multiplicative form and investigates the asymptotic properties of the Quasi-Likelihood (QL) function (Section 2.1). Next, in Section 2.2, we present the QML estimation of the GARCH parameters. In Section 3, we present our adaptive approach for recursively estimating GARCH parameters named AdaVol. We examine the AdaVol procedure on simulated and real-life observations in Section 4, and some concluding remarks are made in Section 5.

2. QML Estimation in Conditionally Heteroscedastic Time Series Models

The approximate QMLE $\widehat{\theta}_n^*$ is defined as

$$\widehat{\theta}_n^* \in \arg \min_{\theta \in \mathcal{K}} \widehat{L}_n(\theta), \quad (2.1)$$

where the parameter set \mathcal{K} is a suitable compact subset of the parameter space Θ . The QL function $L_n(\theta)$ and approximate QL function $\widehat{L}_n(\theta)$ are given by

$$L_n(\theta) = \sum_{t=1}^n l_t(\theta) \text{ and } \widehat{L}_n(\theta) = \sum_{t=1}^n \widehat{l}_t(\theta), \quad (2.2)$$

with QL losses, denoted $l_t(\theta)$ and $\widehat{l}_t(\theta)$, given as

$$l_t(\theta) = \frac{1}{2} \left(\frac{X_t^2}{h_t(\theta)} + \log h_t(\theta) \right) \text{ and } \widehat{l}_t(\theta) = \frac{1}{2} \left(\frac{X_t^2}{\widehat{h}_t(\theta)} + \log \widehat{h}_t(\theta) \right), \quad (2.3)$$

where (\widehat{h}_t) is an approximation of (h_t) defined recursively for $t \geq 1$ as in (1.2) with initialization $\widehat{h}_{-q+1} = \dots = \widehat{h}_0 = 0$ or any deterministic constant. From (Straumann, 2005, Proposition 5.2.12), we know the initialization error between (\widehat{h}_t) and the true (h_t) will vanish exponentially fast almost surely. Assuming Z_0 is standard normal distributed, we may note X_t is also Gaussian with variance h_t conditioned on \mathcal{F}_{t-1} . The QL function $L_n(\cdot)$ in (2.2) is derived under this Gaussian assumption.

The consistency and asymptotic properties of the QMLE $\widehat{\theta}_n^*$ combined with the robustness of the QL function for overdispersion make the method highly used in practice (e.g., see Patton (2006)). Under the assumptions in (Straumann and Mikosch, 2006, N.1, N.2, N.3 and N.4), the QMLE $\widehat{\theta}_n^*$ is strongly consistent and asymptotically normal, that is

$$\widehat{\theta}_n^* \xrightarrow{\text{a.s.}} \theta_0 \text{ and } \sqrt{n}(\widehat{\theta}_n^* - \theta_0) \rightarrow \mathcal{N}(0, V_0) \text{ as } n \rightarrow \infty, \quad (2.4)$$

with θ_0 as the true parameter vector and V_0 the asymptotic covariance matrix.

Unfortunately, these asymptotic properties in (2.4) come with a drawback on the QL loss; the consistency is achieved through careful domination of logarithmic moments. The concavity of logarithms makes the criterion insensitive to extreme values, but it also implies that the criterion itself behaves as a concave function. As most optimization algorithms are based on convex assumptions, this is striking.

In the next section, we show that the approximate Hessian $\widehat{H}_n(\theta) = n^{-1} \nabla_{\theta}^2 \widehat{L}_n(\theta)$ admits strictly positive eigenvalues for n sufficiently large dependent on the model specifications and the underlying data process. This means that for sufficiently large batch sizes of observations, the QMLE $\widehat{\theta}_n^*$ can be seen as the unique solution of a locally strongly convex optimization problem; the existence and uniqueness of $\widehat{\theta}_n^*$ ensure that usual iterative optimization routines can efficiently approximate it for n large enough.

2.1. Asymptotic Properties of the QL Function

To establish the asymptotic local convexity of the QL function of the model described by (1.1)-(1.2), we need the following assumptions: Assumption W1, W2, and W3, which naturally emerges from the arguments and properties Straumann and Mikosch (2006) made to ensure stability of the QL function and QMLE procedure. We will use two different matrix norms, namely, let $\|A\|_{op}$ denote the matrix operator norm of the matrix $A \in \mathbb{R}^{d \times d}$ with respect to the Euclidean norm, i.e., $\|A\|_{op} = \sup_{v \neq 0} |Av|/|v|$, and denote $\|A\|_{\mathcal{K}}$ the norm of the continuous matrix-valued function A on \mathcal{K} , i.e., $\|A\|_{\mathcal{K}} = \sup_{x \in \mathcal{K}} \|A(x)\|_{op}$, where \mathcal{K} is a compact set of \mathbb{R}^d .

Assumption W1. The model (1.1)-(1.2) with $\theta = \theta_0$ admits a unique stationary ergodic solution.

Assumption W2. Let $\mathcal{K} \subset \Theta$ be a compact set with true parameter vector $\theta_0 \in \mathcal{K}$ in the interior. The random functions fulfill certain conditions, such that $\mathbb{E}[\|l_0\|_{\mathcal{K}}] < \infty$, $\mathbb{E}[\|\nabla_{\theta}^2 l_0\|_{\mathcal{K}}] < \infty$, and furthermore have the following uniform convergences $\|n^{-1} \widehat{L}_n - L_n\|_{\mathcal{K}} \xrightarrow{a.s.} 0$ and $n^{-1} \|\nabla_{\theta}^2 \widehat{L}_n - \nabla_{\theta}^2 L_n\|_{\mathcal{K}} \xrightarrow{a.s.} 0$ as $n \rightarrow \infty$.

Assumption W3. The components of the vector $\nabla_{\theta} g_{\theta}(X_0, h_0)$ from (1.2) with $\theta = \theta_0$ are linearly independent random variables.

The following Theorem 2.1 is an extension of Ip et al. (2006), which established similar results for the likelihood function of GARCH models under the assumption that (X_t) is strictly stationary and strongly mixing with geometric rate, and (Z_t) is Gaussian. Solving the QML estimation problem in (2.1) for $\widehat{\theta}_n^*$ is known to be computationally heavy as one has to find the solution of a non-linear equation, namely (2.2). Nonetheless, Theorem 2.1 ensures the existence of an N such that we have a unique global QMLE $\widehat{\theta}_n^*$ for all $n \geq N$.

Theorem 2.1. *Under Assumption W1, W2, and W3, there exist positive constants $C, \delta > 0$, and a random positive integer $N \in \mathbb{N}$ such that*

$$g^T \widehat{H}_n(\theta) g > C g^T g, \quad \forall n \geq N, \quad a.s., \quad (2.5)$$

for all $\theta \in B(\theta_0, \delta)$ and $g \in \mathbb{R}^d \setminus \{0\}$.

The result above shows local strong convexity of the QL function \widehat{L}_n . The following corollary arises from the proof of Theorem 2.1:

Corollary 2.1. *Under Assumption W1, W2, and W3, the QMLE $\widehat{\theta}_n^*$ exists and is unique, that is*

$$\widehat{\theta}_n^* = \arg \min_{\theta \in \mathcal{K}} \widehat{L}_n(\theta).$$

Local strong convexity is crucial for guaranteeing the convergence of an optimization algorithm, although some methods go beyond this point (Ward et al. (2018)). Thus, Theorem 2.1 is an essential result for computing the QMLE $\widehat{\theta}_n^*$ parameters of the model in (1.1)-(1.2). Nevertheless, to guarantee the property in (2.5), we need a sufficiently large (and maybe unbounded)

random N , which depends on the true parameter vector θ_0 , the parameter estimates $(\widehat{\theta}_t^*)$, and the observations (X_t) . One often has a fixed size of observations in practice, so the iterative algorithm may not converge. To our experience, this phenomenon may occur when the true parameter vector θ_0 is close to the boundary of \mathcal{K} , or if the initial values $\widehat{\theta}_0^*$ are far away from the true parameters θ_0 .

2.2. QML Estimation of GARCH(p, q) Parameters

The general class of conditionally heteroscedastic time series models includes the very popular ARCH and GARCH models. For more than three decades, these models have attracted considerable amounts of attention in the literature since their introduction. A process (X_t) is called a GARCH(p, q) process with parameter vector $\theta = (\omega, \alpha_1, \dots, \alpha_p, \beta_1, \dots, \beta_q)^T$, if it satisfies

$$\begin{cases} X_t = \sigma_t Z_t, \\ \sigma_t^2 = \omega + \sum_{i=1}^p \alpha_i X_{t-i}^2 + \sum_{j=1}^q \beta_j \sigma_{t-j}^2, \end{cases} \quad (2.6)$$

where ω, α_i , and β_j for $1 \leq i \leq p$ and $1 \leq j \leq q$ are non-negative parameters ensuring the non-negativity of the conditional variance process (σ_t^2) . The innovations (Z_t) is a sequence of i.i.d. random variables with $\mathbb{E}[Z_0] = 0$ and $\mathbb{E}[Z_0^2] = 1$. Likewise, one can define an ARCH(p) process by setting $\beta_j = 0$ for $1 \leq j \leq q$ in (2.6). The GARCH(p, q) process (X_t) given in (2.6) has QL losses given by $\widehat{l}_t(\theta) = 2^{-1}(X_t^2/\widehat{\sigma}_t^2(\theta) + \log \widehat{\sigma}_t^2(\theta))$ with first-order derivative

$$\nabla_{\theta} \widehat{l}_t(\theta) = \nabla_{\theta} \widehat{\sigma}_t^2(\theta) \left(\frac{\widehat{\sigma}_t^2(\theta) - X_t^2}{2\widehat{\sigma}_t^4(\theta)} \right) \quad (2.7)$$

and second-order derivative

$$\nabla_{\theta}^2 \widehat{l}_t(\theta) = \nabla_{\theta} \widehat{\sigma}_t^2(\theta)^T \nabla_{\theta} \widehat{\sigma}_t^2(\theta) \left(\frac{2X_t^2 - \widehat{\sigma}_t^2(\theta)}{2\widehat{\sigma}_t^6(\theta)} \right) + \nabla_{\theta}^2 \widehat{\sigma}_t^2(\theta) \left(\frac{\widehat{\sigma}_t^2(\theta) - X_t^2}{2\widehat{\sigma}_t^4(\theta)} \right), \quad (2.8)$$

where $\nabla_{\theta} \widehat{\sigma}_t^2(\theta) = \vartheta_t(\theta) + \sum_{j=1}^q \beta_j \nabla_{\theta} \widehat{\sigma}_{t-j}^2(\theta)$ with $\vartheta_t(\theta) = (1, X_{t-1}^2, \dots, X_{t-p}^2, \widehat{\sigma}_{t-1}^2(\theta), \dots, \widehat{\sigma}_{t-q}^2(\theta))^T \in \mathbb{R}^{p+q+1}$ and Hessian $\widehat{H}_n(\theta) = n^{-1} \sum_{t=1}^n \nabla_{\theta}^2 \widehat{l}_t(\theta)$.

The equations (2.6) creates a complicated probabilistic structure that is not easily understood, although it looks relatively simple. The conditions ensuring the existence and uniqueness of a stationary solution to the equations (2.6) for GARCH(1, 1) was provided by Nelson (1990). Bougerol and Picard (1992) later showed it for the GARCH(p, q) model using that GARCH(p, q) can be embedded in a Iterated Random Lipschitz Map (IRLM). See Bougerol (1993) for a formal definition of IRLMs.

We can illustrate the IRLM method on the GARCH(1, 1) model with parameter vector $\theta = (\omega, \alpha_1, \beta_1)^T$. The IRLM for σ_t^2 is then given by $\sigma_t^2 = A_t \sigma_{t-1}^2 + B_t$ with $t \in \mathbb{Z}$, where $A_t = \alpha_1 Z_{t-1}^2 + \beta_1$ and $B_t = \omega$. Note $((A_t, B_t))$ constitutes an i.i.d. sequence. From the literature on IRLMs it is well known that the conditions $\mathbb{E}[\log |A_0|] < 0$ and $\mathbb{E}[\log^+ |B_0|] < \infty$ guarantee the existence and uniqueness of a strictly stationary solution of the IRLM $Y_t = A_t Y_{t-1} + B_t$ for $t \in \mathbb{Z}$ provided $((A_t, B_t))$ is a stationary ergodic sequence. Applying this to the GARCH(1, 1)

model, we get the known sufficient condition for the existence of a stationary solution, namely $\mathbb{E}[\log(\alpha_1 Z_0^2 + \beta_1)] < 0$. This also implies $\beta_1 < 1$ since $\log(\beta_1) \leq \mathbb{E}[\log(\alpha_1 Z_0^2 + \beta_1)] < 0$. Likewise, the ARCH(1) process ($\beta_1 = 0$) then requires $\mathbb{E}[\log(\alpha_1 Z_0^2)] < 0$, which is the same as $\alpha < 2e^\epsilon \approx 3.56$ with Z_0 being Gaussian. Thus, the stationary condition is much weaker than the second-order stationary condition in which we require $\alpha_1 + \beta_1 < 1$.

The statistical inference leads to further nontrivial problems since the exact distribution of (Z_t) remains unspecified, and so one usually determines the likelihoods under the hypothesis of standard Gaussian innovations. Moreover, the volatility (σ_t) is an unobserved quantity approximated by mimicking the recursion (2.6) with an initialization, for instance $X_{-p+1} = \dots = X_0 = 0$ and $\sigma_{-q+1}^2 = \dots = \sigma_0^2 = 0$. Berkes et al. (2003) showed under minimal assumptions that the QMLE is strongly consistent and asymptotically normal.

Furthermore, under Assumption W1-W3, we have asymptotic local strong convexity of the QL function in GARCH(p, q) models by Theorem 2.1. However, the number of observations needed to guarantee local strong convexity vary. This can easily be seen by looking at the simplest case, namely when (X_t) follows an ARCH(1) process with parameter vector $\theta = (\omega, \alpha_1)^T$. The volatility process $\sigma_t^2(\theta)$ is given as $\omega + \alpha_1 X_{t-1}^2$. The eigenvalues of $\nabla_{\theta}^2 l_t(\theta)$ are given by $\lambda_t = (\lambda_{t,1}, \lambda_{t,2}) = (0, \lambda_{t,2})$ with $\lambda_{t,2} = (1 + X_{t-1}^2)(2X_t^2 - \sigma_t^2(\theta))2^{-1}\sigma_t^{-6}(\theta)$. Thus, the non-negativity of $\lambda_{t,2}$ would ensure convexity at time t in our QML procedure. However, the probability of having convexity at each t is unlikely as $\mathbb{P}(\cap_{t=1}^n \nabla_{\theta}^2 l_t(\theta) \geq 0) = \mathbb{P}(\cap_{t=1}^n Z_t^2 \geq 1/2) = \mathbb{P}(Z_0^2 \geq 1/2)^n$ is approximately 0.52^n with i.i.d. Gaussian innovations (Z_t) , i.e., (Z_t^2) is χ^2 -distributed with 1 degree of freedom. On the other hand, increasing the number of observations used at each iteration would increase the probability of having local strong convexity.

3. Adaptive Recursive QML Estimation

Our recursive QML method relies on stochastic approximations introduced by Robbins and Monro (1951), which only requires the previous parameter estimate to update the parameter estimate using the new observation. We perform the first-order stochastic gradient method defined as

$$\widehat{\theta}_t = \widehat{\theta}_{t-1} - \eta_{t-1} \nabla_{\theta} \widehat{l}_t(\widehat{\theta}_{t-1}), \quad (3.1)$$

where $\eta_{t-1} > 0$ is the step-size at the $t - 1$ step, and $\nabla_{\theta} \widehat{l}_t(\widehat{\theta}_{t-1})$ is the gradient using the X_t observation and the QMLE estimate $\widehat{\theta}_{t-1}$. This method is computationally efficient as it only requires a cost of $\mathcal{O}(d)$ per recursion. Depending on the number of observations, we have a trade-off between the accuracy of the recursive QML estimates and the time it takes to perform a parameter update (Bottou and Bousquet (2007)).

According to Robbins and Monro (1951), we must schedule the step-size such that $\sum_{t=1}^{\infty} \eta_t = \infty$ and $\sum_{t=1}^{\infty} \eta_t^2 < \infty$, but these bounds do not make the choice of an appropriate step-size η_t easier in practice. A more suitable approach is an adaptive learning rate, which updates the step-size in (3.1) on the

fly pursuant to the gradient $\nabla_{\theta} \widehat{l}_t(\cdot)$. Thus, our choice of step-size η_t have less impact on performance, making convergence more robust and lower the demand for manually fine-tuning. Such an approach is often used in settings of streaming data as generic methods are preferred. Adaptive and separate learning rates for each parameter was proposed by Duchi et al. (2011) in their AdaGrad procedure. A different learning rate speeds up convergence in situations where the appropriate learning rates vary across parameters. Other well-known examples of adaptive learning rates could be AdaDelta by Zeiler (2012), RMSProp by Tieleman and Hinton (2012) and ADAM by Kingma and Ba (2015). As we may expect a lack of convexity, we select the AdaGrad algorithm since it has shown promising results in non-convex optimization (Ward et al. (2018)). The AdaGrad procedure is given by the updates

$$\widehat{\theta}_t = \widehat{\theta}_{t-1} - \frac{\eta}{\sqrt{\sum_{i=1}^t \nabla_{\theta} \widehat{l}_i(\widehat{\theta}_{i-1})^2 + \epsilon}} \nabla_{\theta} \widehat{l}_t(\widehat{\theta}_{t-1}), \quad (3.2)$$

where $\eta > 0$ is a constant learning rate and $\epsilon > 0$ a small number ensuring positivity. Good default values are $\eta = 0.1$ and $\epsilon = 10^{-8}$, see the AdaVol algorithm in Table 1. Note $\nabla_{\theta} \widehat{l}_i(\widehat{\theta}_{i-1})^2$ denotes the element-wise square $\nabla_{\theta} \widehat{l}_i(\widehat{\theta}_{i-1}) \odot \nabla_{\theta} \widehat{l}_i(\widehat{\theta}_{i-1})$.

As the QL loss is defined only for $\theta_n \in \mathcal{K}$, we will require that the recursive algorithm always takes values in \mathcal{K} . Zinkevich (2003) suggests we project our approximation $\widehat{\theta}_n$ onto \mathcal{K} , preventing large jumps and enforcing the convergence of our stochastic gradient method. By implementing this projection on (3.2), we have our method for updating estimates, namely

$$\widehat{\theta}_t = \text{P}_{\mathcal{K}} \left[\widehat{\theta}_{t-1} - \frac{\eta}{\sqrt{\sum_{i=1}^t \nabla_{\theta} \widehat{l}_i(\widehat{\theta}_{i-1})^2 + \epsilon}} \nabla_{\theta} \widehat{l}_t(\widehat{\theta}_{t-1}) \right]. \quad (3.3)$$

3.1. Adaptive Recursive QML Estimation for GARCH Models

The GARCH process (X_t) parameters can be numerically challenging to estimate in empirical applications. The numerical optimization algorithms can quickly fail or converge to irregular solutions (Zumbach (2000)). Therefore, examining the approximative QMLE $\widehat{\theta}_n^*$ must be made with a healthy amount of skepticism. A well-discussed problem for the GARCH(p, q) models is that the QMLE performs poorly for numerically small (but still positive) values of ω . The parameter ω is vital and often tricky to estimate. Stabilizing the estimation of ω would not only improve the ω estimate but also have a positive impact on the other model parameters.

One way to overcome small values of ω for the GARCH(p, q) model is by scaling (X_t) with some factor $\lambda > 0$ as we have homogeneity; let (X_t) follow a GARCH(p, q) process with parameter vector $\theta = (\omega, \alpha_1, \dots, \alpha_p, \beta_1, \dots, \beta_q)^T$ and innovations (Z_t) . Then for any $\lambda > 0$, the process $(\sqrt{\lambda} X_t)$ is a GARCH(p, q) process with parameter vector $\theta = (\lambda\omega, \alpha_1, \dots, \alpha_p, \beta_1, \dots, \beta_q)^T$ and identical innovations (Z_t) .

However, we wish to avoid this form of inference in our recursive algorithm as one then needs to come up with a scaling parameter that has to be estimated beforehand. Instead,

we circumvent this issue by introducing a concept called Variance Targeting Estimation (VTE) (Francq et al. (2011)). We apply VTE for estimating ω by use of γ^2 , which is the unconditional variance estimated by the sample variance (as seen in (3.4)). Thus we have a two-step estimator where we estimate the sample variance γ^2 recursively, and the remaining parameters $\theta = (\alpha_1, \dots, \alpha_p, \beta_1, \dots, \beta_q)^T$ are estimated by the QML method. Pseudo-code of the AdaVol algorithm is presented in Table 1. The reparametrization is obtained by defining

$$\omega = \gamma^2 \left(1 - \sum_{i=1}^p \alpha_i - \sum_{j=1}^q \beta_j \right). \quad (3.4)$$

The volatility process in the GARCH(p, q) process can then be rewritten as

$$(\sigma_t^2 - \gamma^2) = \sum_{i=1}^p \alpha_i (X_{t-i}^2 - \gamma^2) + \sum_{j=1}^q \beta_j (\sigma_{t-j}^2 - \gamma^2). \quad (3.5)$$

Similarly, one can define an ARCH(p) process by setting $\beta_j = 0$ for $1 \leq j \leq q$. The GARCH(p, q) process (X_t) in (3.5) has similar QL losses as before except $\nabla_{\theta} \widehat{\sigma}_t^2(\theta)$ in (2.7) and (2.8), where $\vartheta_t(\theta)$ is given as $(X_{t-1}^2 - \gamma^2, \dots, X_{t-p}^2 - \gamma^2, \widehat{\sigma}_{t-1}^2(\theta) - \gamma^2, \dots, \widehat{\sigma}_{t-q}^2(\theta) - \gamma^2)^T \in \mathbb{R}^{p+q}$ and the parameter space is defined by $\mathcal{K} = \{(\alpha_1, \dots, \alpha_p, \beta_1, \dots, \beta_q) \in \mathbb{R}_+^{p+q} \mid \sum_{i=1}^p \alpha_i + \sum_{j=1}^q \beta_j < 1\}$.

AdaVol: Adaptive recursive QML estimation for GARCH(p, q) models using the technique of VTE.

Data: $(X_t)_{t \geq 1}$ (observations)

input : θ_0 (initial parameter vector), $\eta = 0.1$, $\epsilon = 10^{-8}$

begin

initialize: $\widehat{\sigma}_1^2 = X_1^2$, $\widehat{\mu}_0 = 0$, $\widehat{\gamma}_0^2 = 0$, $\widehat{G}_0 = \epsilon$ and $t = 0$

while $\widehat{\theta}_t$ **not converged do**

$t = t + 1$

$\widehat{\mu}_t = t(t+1)^{-1} \widehat{\mu}_{t-1} + (t+1)^{-1} X_t$

$\widehat{\gamma}_t^2 = (t-1)t^{-1} \widehat{\gamma}_{t-1}^2 + t^{-1} (X_t - \widehat{\mu}_t)^2$

$\widehat{g}_t = \nabla_{\theta} \widehat{l}_t(\theta_{t-1})$

$\widehat{G}_t = \widehat{G}_{t-1} + \widehat{g}_t^2$

$\widehat{\theta}_t = \text{P}_{\mathcal{K}} \left[\widehat{\theta}_{t-1} - \eta \widehat{G}_t^{-1/2} \widehat{g}_t \right]$

$\widehat{\sigma}_{t+1}^2 = \widehat{\gamma}_t^2 + \sum_{i=1}^p \widehat{\alpha}_i^{(t)} (X_{t-i}^2 - \widehat{\gamma}_t^2) + \sum_{j=1}^q \widehat{\beta}_j^{(t)} (\widehat{\sigma}_{t-j}^2 - \widehat{\gamma}_t^2)$

end

end

Result: $\widehat{\theta}_t$ (resulting estimates), $\widehat{\sigma}_{t+1}^2$ (predicted volatility)

Table 1: Pseudo-code of the AdaVol algorithm.

The VTE is not a requirement for the recursive method, but it provides additional speed and numerical stability. Namely, the VTE ensures a consistent estimate of the long-run variance, even if the model is misspecified. Additionally, presuming γ is well estimated, we reduce the parameter space dimension and increase the speed of convergence of the recursive optimization

routines. Moreover, the geometry of the new set of optimization \mathcal{K} allows the projection step in (3.3) to be efficiently implemented following Duchi et al. (2008).

One should be aware that the VTE requires stronger assumptions for the existence of the variance and is likely to suffer from efficiency loss. Francq et al. (2011) also showed that the VTE would never be asymptotically more accurate than the QMLE. Another drawback of using the VTE is the need for a finite fourth moment of the process (X_t). Meaning, one would need $\alpha_1 < 0.57$ for an ARCH(1) model using standard Gaussian noise as $EX_t^4 < \infty$ if and only if $\alpha_1^2 + (EZ_0^4 - 1)\alpha_1^2 < 1$. For a GARCH(1, 1) model, we should have $(\alpha_1 + \beta_1)^2 + (EZ_0^4 - 1)\alpha_1^2 < 1$. These parameter bounds restrict the usefulness and range of applications for the VTE techniques. Fortunately, these constraints solely concern the batch setting.

4. Applications

In this section, we examine the AdaVol algorithm on simulated and real-life observations. Our implementation of AdaVol is provided in a repository at Werge (2019), and a relative speed comparison can be found in Appendix B. We compare our approach to the Iterative QMLE (IQMLE) approximation $\widehat{\theta}_n$, which is estimated at every two thousand increments using all observations up to this point, i.e., $(\widehat{\theta}_t)_{(k-2000)+1 \leq t \leq k}$ is estimated using $(X_t)_{1 \leq t \leq k}$ for $k = 2000, 4000, \dots, n$. In this way, we illuminate the large-scale learning trade-off of applying our recursive method instead of the iterative method, which is forward-looking with up to two thousand observations (Bottou and Bousquet (2007)). As suggested by Ip et al. (2006), we use the (bounded) *L-BFGS* algorithm to solve the nonlinear optimization problem in (2.1) for $\widehat{\theta}_n$ with initial guess $\widehat{\theta}_0 \in \mathcal{K}$. Our recursive QMLE approximation $\widehat{\theta}_n$ is produced by the AdaVol algorithm (described in Table 1). It takes our initial value $\widehat{\theta}_0 \in \mathcal{K}$, learning rate $\eta = 0.1$ and $\epsilon = 10^{-8}$ as input. At last, for a fair comparison, we always use the same initial guess for both methods, namely $\widehat{\theta}_0 = \widehat{\theta}_0 \in \mathcal{K}$.

It is possible to customize AdaVol by tuning the learning parameter η , e.g., by choosing the best performing learning rate evaluated on the first part of the observations. We use a fixed learning rate $\eta = 0.1$ across all applications (simulated and real-life observations) to avoid the learning rate's potential influence in our experiments. However, one should be aware of the versatility achieved with different learning rate choices. The choice of learning rates is cumbersome, as an excessive learning rate can cause the algorithm to deviate from the true parameter estimate. In contrast, a learning rate that is too small can lead to slow convergence. Nevertheless, a small learning rate may be preferred if one only wants to keep track of minor parameter estimation changes.

4.1. Simulations

All simulations are performed by the use of twenty thousand observations ($n = 20000$), and the simulated data (X_t) is always generated using Gaussian innovations with zero mean and unit variance. To avoid possible bias due to the choice of the true

parameter vector θ_0 and initial values $\widehat{\theta}_0, \widetilde{\theta}_0$, we conduct our experiments using random parameter vectors $\theta_0 \in \mathcal{K}$ and random initial guesses $\widehat{\theta}_0, \widetilde{\theta}_0 \in \mathcal{K}$. These parameter vectors are drawn randomly from our parameter space \mathcal{K} . The ω parameter is generated by taking a positive number from a uniform distribution, and then we multiply it with $10^{-\tau}$, where τ is some random positive integer up to eight. In this way, we cover a broad parameter domain while having parameter values close to the boundary. Similarly, the $(\alpha_i)_{1 \leq i \leq p}$ and $(\beta_j)_{1 \leq j \leq q}$ parameters is generated from a uniform distribution with the condition of having $\sum_{i=1}^p \alpha_i + \sum_{j=1}^q \beta_j < 1$. Note that the initial guesses $\widehat{\theta}_0$ and $\widetilde{\theta}_0$ are generated the same way. Thus, when we mention random parameters for the rest of the paper, we refer to this generation procedure.

4.1.1. ARCH Models

As discussed earlier, the iterative QMLE approximation $\widehat{\theta}_n$ performs poorly for numerically small $\omega > 0$ values, which are often encountered in financial time series. Before moving on to the case of small ω parameter values, we have in Figure 1 the trajectories of both QMLE approximations using an ARCH(1) process with true parameter vector and initial values given by

$$\theta_0 = \begin{pmatrix} \omega \\ \alpha_1 \end{pmatrix} = \begin{pmatrix} 2.0 \\ 0.6 \end{pmatrix} \text{ and } \widehat{\theta}_0 = \widetilde{\theta}_0 = \begin{pmatrix} 1.5 \\ 0.4 \end{pmatrix}. \quad (4.1)$$

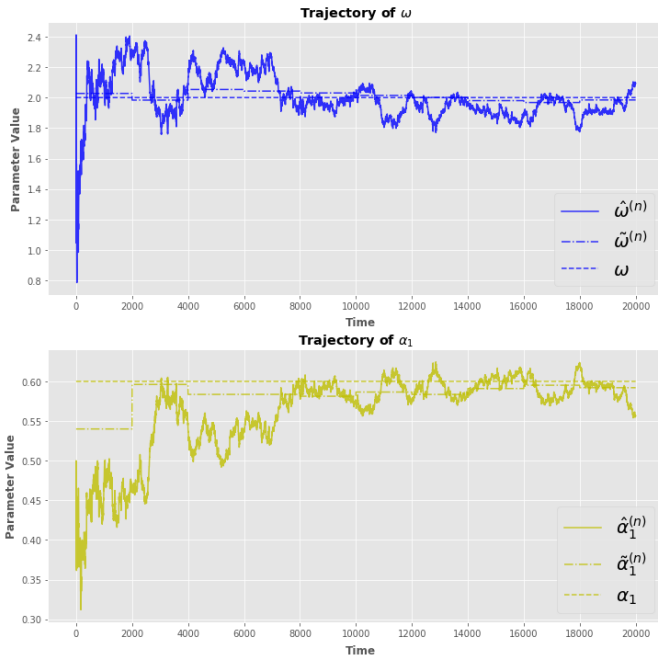


Figure 1: Trajectory of $\widehat{\theta}_n$ (solid line) and $\widetilde{\theta}_n$ (semi-dotted line) for an ARCH(1) process with true parameter vector (dotted line) and initial guess given in (4.1).

Figure 1 shows a very reasonable convergence of both estimators, $\widehat{\theta}_n = (\widehat{\omega}^{(n)}, \widehat{\alpha}_1^{(n)})^T$ and $\widetilde{\theta}_n = (\widetilde{\omega}^{(n)}, \widetilde{\alpha}_1^{(n)})^T$, when the true parameter $\omega = 2.0$. Not surprisingly, our method experiences some fluctuations initially, but as the learning rate decreases, the fluctuation likewise evaporates, and within the first few thousand observations, we hit the true parameter values.

Likewise, in Figure 2, we have the QMLE approximations' trajectories for an ARCH(1) process, but now with true parameter vector and initial guess given as

$$\theta_0 = \begin{pmatrix} 1 \cdot 10^{-8} \\ 0.6 \end{pmatrix} \text{ and } \widehat{\theta}_0 = \widetilde{\theta}_0 = \begin{pmatrix} 5 \cdot 10^{-8} \\ 0.4 \end{pmatrix}. \quad (4.2)$$

Figure 2 indicates a modest convergence of $\widehat{\theta}_n$ but shows slow convergence of $\widetilde{\alpha}_n$ towards the true α_1 parameter. In addition, $\widetilde{\alpha}_n$ seems biased concerning the initial value $\widetilde{\alpha}_0 = 0.4$ as it processes almost half of the observations before moving closer to the true $\alpha_1 = 0.6$.

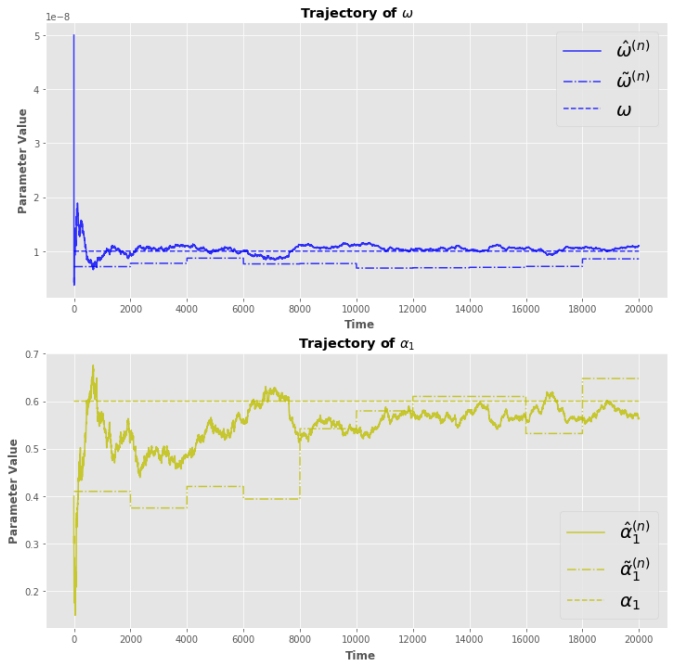


Figure 2: Trajectory of $\widehat{\theta}_n$ (solid line) and $\widetilde{\theta}_n$ (semi-dotted line) for an ARCH(1) process with true parameter vector (dotted line) and initial guess given in (4.2).

A way of demonstrating the variation of $\widehat{\theta}_n$ and $\widetilde{\theta}_n$ performance for small ω values is presented in Figure 3 and Figure 4, where we have the average trajectory of one hundred trajectories with their corresponding boxplots showing the distribution of these one hundred trajectories.

Here, in Figure 3, we can see that AdaVol converges to the true parameter values with low sensitivity to the choice of initial values. Moreover, this convergence occurs within the first few thousand observations. However, in Figure 4, we see the opposite in which $\widehat{\theta}_n$ has convergence issues; it is consistently underestimating the ω parameter. Furthermore, the α_1 parameter range does not appear to be decreasing over time, and the range seems larger than AdaVol's.

As we observe the true volatility process (σ_t) in this section, we can evaluate the predicted volatility processes' accuracy. We do this using the Mean Percentage Errors (MPE) given as

$$\widehat{\sigma}_{\text{MPE}} = \frac{1}{n} \sum_{t=1}^n \frac{\sigma_t - \widehat{\sigma}_t}{\sigma_t} \text{ and } \widetilde{\sigma}_{\text{MPE}} = \frac{1}{n} \sum_{t=1}^n \frac{\sigma_t - \widetilde{\sigma}_t}{\sigma_t}, \quad (4.3)$$

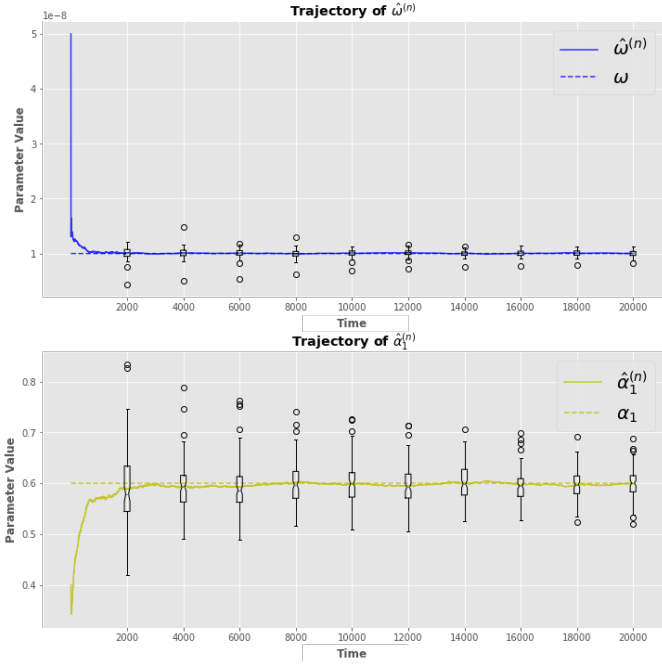


Figure 3: Average trajectory (solid line) of one hundred $\hat{\theta}_n$'s for an ARCH(1) process with true parameter vector (dotted line) and initial guess from (4.2). The boxplots shows the distribution of the one hundred trajectories.

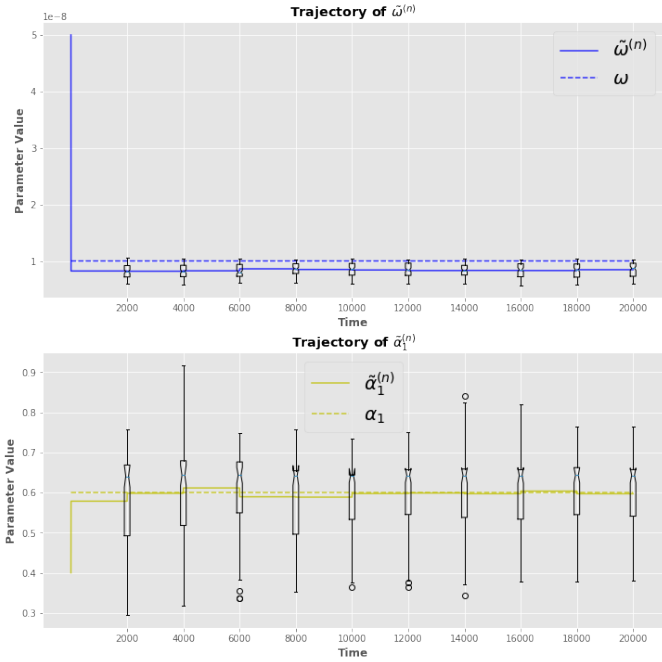


Figure 4: Average trajectory (solid line) of one hundred $\hat{\theta}_n$'s for an ARCH(1) process with true parameter vector (dotted line) and initial guess from (4.2). The boxplots shows the distribution of the one hundred trajectories.

and the Mean Absolute Percentage Errors (MAPE) given by

$$\hat{\sigma}_{\text{MAPE}} = \frac{1}{n} \sum_{t=1}^n \frac{|\sigma_t - \hat{\sigma}_t|}{\sigma_t} \text{ and } \tilde{\sigma}_{\text{MAPE}} = \frac{1}{n} \sum_{t=1}^n \frac{|\sigma_t - \tilde{\sigma}_t|}{\sigma_t}, \quad (4.4)$$

where $(\hat{\sigma}_t)$ is coming from AdaVol and $(\tilde{\sigma}_t)$ from the IQMLE

approximation. Note that $\tilde{\sigma}_t$'s estimation is the same as for the IQMLE approximation $\tilde{\theta}_t$, i.e., $(\tilde{\sigma}_t)_{(k-2000)+1 \leq t \leq k}$ is estimated using $(X_t)_{1 \leq t \leq k}$ for $k = 2000, 4000, \dots, n$.

In the rest of this section, we will use random parameters to generalize our studies, limiting the potential bias from having fixed parameters (See Section 4.1). Our routine is as follows: We draw a random true parameter vector $\theta_0 \in \mathcal{K}$ from which we generate our observations (X_t) . Based on these observations (X_t) , we calculate our estimates using (a random) $\hat{\theta}_0 = \tilde{\theta}_0 \in \mathcal{K}$. Then, we evaluate our estimates using an accuracy score, e.g., MPE and MAPE. Finally, we repeat all these steps the desired number of times. Boxplots of one hundred accuracy scores, MPE in (4.3) and MAPE in (4.4), can be found in Figure 5. In the top graph of Figure 5, one can observe the MPE (for both methods) is symmetric around zero, but $\tilde{\sigma}_{\text{MPE}}$ has a negative tail, meaning the iterative method may overestimate the volatility in some cases. Also, the spread of $\tilde{\sigma}_{\text{MPE}}$ is higher than the $\hat{\sigma}_{\text{MPE}}$, which is clearly seen by looking at $\hat{\sigma}_{\text{MAPE}}$ in the bottom graph of Figure 5.

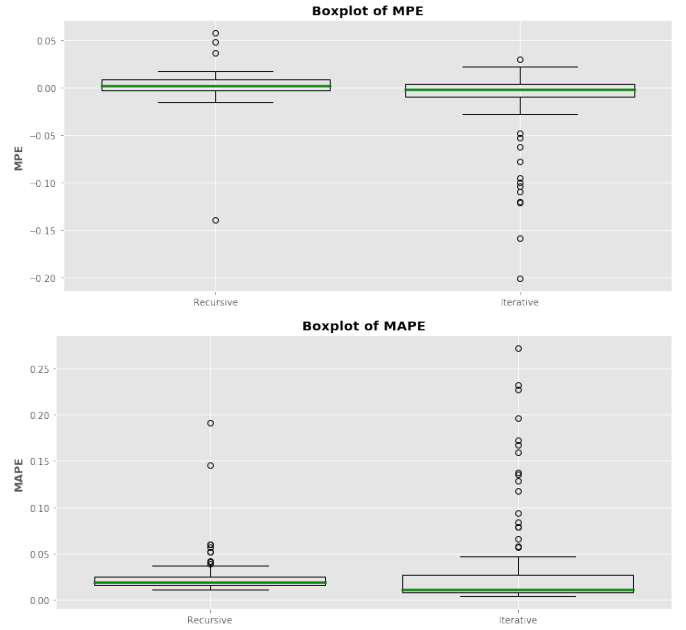


Figure 5: Boxplots of one hundred accuracy scores MPE (4.3) and MAPE (4.4) using an ARCH(1) process with random true parameter vector and initial guess in \mathcal{K} .

Another way of measuring the accuracy can be made by studying the conditional quantiles using the recursive $(\hat{\sigma}_t)$ and iterative $(\tilde{\sigma}_t)$ predicted volatility processes (Biau and Patra (2011)). Under the assumption of standard Gaussian innovations, X_t is Gaussian with zero mean and variance σ_t^2 . Thus, for any $\alpha \in (0, 1)$, the α -quantile of a Gaussian distribution $\mathcal{N}(0, \sigma_t^2)$ is $\sigma_t \Phi^{-1}(\alpha)$, where $\Phi^{-1}(\alpha)$ is the α -quantile of the standard Gaussian distribution. We use the so-called α -quantile loss function proposed by Koenker and Bassett (1978): The α -quantile loss function ρ_α using the volatility process σ_t is de-

defined as

$$\rho_\alpha(X_t, \sigma_t) = \begin{cases} \alpha(X_t - \Phi^{-1}(\alpha)\sigma_t), & \text{for } X_t > \Phi^{-1}(\alpha)\sigma_t, \\ (1 - \alpha)(\Phi^{-1}(\alpha)\sigma_t - X_t), & \text{for } X_t \leq \Phi^{-1}(\alpha)\sigma_t, \end{cases} \quad (4.5)$$

with tilting parameter $\alpha \in (0, 1)$. The idea behind the α -quantile loss function is to penalize quantiles of low probability more for overestimation than for underestimation (and reversely for high probability quantiles). We evaluate across the α -quantile scores ρ_α of (σ_t) by the (normalized) cumulative α -quantile scoring function QS_α :

$$QS_\alpha(X_n, \sigma_n) = \frac{1}{n} \sum_{t=1}^n \sum_{m=1}^M \rho_{\alpha_m}(X_t, \sigma_t), \quad (4.6)$$

with M as the number of quantiles $\alpha = \{\alpha_1, \dots, \alpha_M\}$. The lowest QS_α score indicates the best ability of volatility forecast. The findings of one hundred $QS_\alpha(X_n, \widehat{\sigma}_n)$ and $QS_\alpha(X_n, \widetilde{\sigma}_n)$ scores is presented in Figure 6, where we have used $\alpha = \{0.01, 0.02, \dots, 0.99\}$, a random true parameter vector and random initialization in \mathcal{K} . The QS_α scores in Figure 6 are indistinguishable. This indicates no loss of generality in using our recursive method even though our estimates are calculated only once, making them more adaptable over time. Surprisingly, the iterative method is not superior, even when forward-looking (with up to two thousand observations).

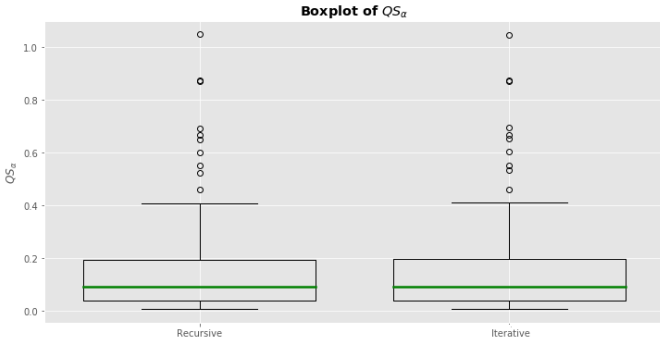


Figure 6: Boxplots of one hundred QS_α scores with $\alpha = \{0.01, 0.02, \dots, 0.99\}$ using an ARCH(1) model with random true parameter vector and initial value in \mathcal{K} .

4.1.2. GARCH Models

Figure 7 and 8 shows the trajectories of the parameter estimates $\widehat{\theta}_n = (\widehat{\omega}^{(n)}, \widehat{\alpha}_1^{(n)}, \widehat{\beta}_1^{(n)})^T$ and $\widetilde{\theta}_n = (\widetilde{\omega}^{(n)}, \widetilde{\alpha}_1^{(n)}, \widetilde{\beta}_1^{(n)})^T$ for a GARCH(1, 1) model with the true parameter vector and initial guess given by

$$\theta_0 = \begin{pmatrix} \omega \\ \alpha_1 \\ \beta_1 \end{pmatrix} = \begin{pmatrix} 1 \cdot 10^{-8} \\ 0.2 \\ 0.7 \end{pmatrix} \text{ and } \widetilde{\theta}_0 = \widetilde{\theta}_0 = \begin{pmatrix} 5 \cdot 10^{-8} \\ 0.1 \\ 0.8 \end{pmatrix}. \quad (4.7)$$

As for the ARCH(1) model, we observe a lower spread in the parameter trajectories coming from AdaVol $\widehat{\theta}_n$ than from the IQMLE approximation $\widetilde{\theta}_n$. Moreover, the iterative $\widetilde{\theta}_n$ is consistently overestimating the β_1 parameter (and underestimating the

α_1 parameter), indicating a bias relative to the initial value. It is worth mentioning that even if all initial values are in the stationary region, i.e., $\widetilde{\theta}_0 = \theta_0 \in \mathcal{K}$, we still have a proper amount of fluctuation in the parameter trajectories. As discussed before, this may partially be due to the volatility introduced by the gradient method and the flatness of the QL loss (Zumbach (2000)). Nevertheless, our recursive method possesses a remarkable convergence already after the first few thousand observations.

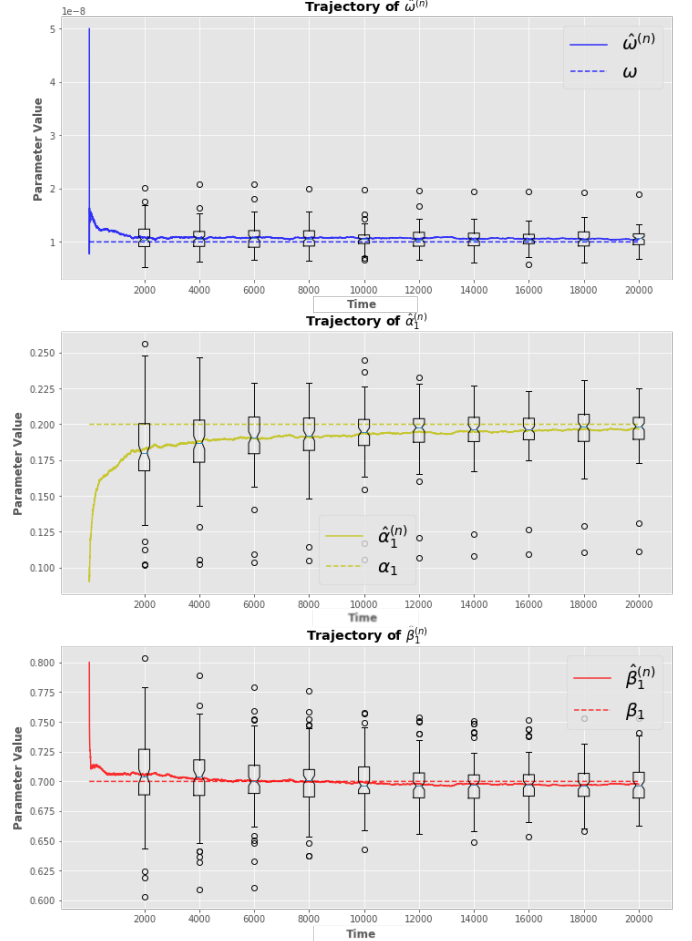


Figure 7: Average trajectory (solid line) of one hundred $\widehat{\theta}_n$'s for a GARCH(1, 1) process with true parameter vector (dotted line) and initial guess given in (4.7). The boxplots shows the distribution of the one hundred trajectories.

The accuracy scores, namely MPE from (4.3) and MAPE from (4.4), can be found in Figure 9 for the GARCH(1, 1) model using random true parameter vector and random initial values in \mathcal{K} . By comparing our methods using random initializations, we circumvent the possible bias from the initial guess, which we observed in Figure 8 for the iterative method. As in the ARCH(1) case, we obtain a lower spread for $\widehat{\sigma}_{\text{MPE}}$ than $\widetilde{\sigma}_{\text{MPE}}$. Nevertheless, one should still expect some probability of ending up with an irregular solution where the AdaVol algorithm fails to converge.

Figure 10 presents the results of one hundred QS_α scores with random true parameter vector and initial value in \mathcal{K} . Again, the QS_α scores are indistinguishable (even when the it-

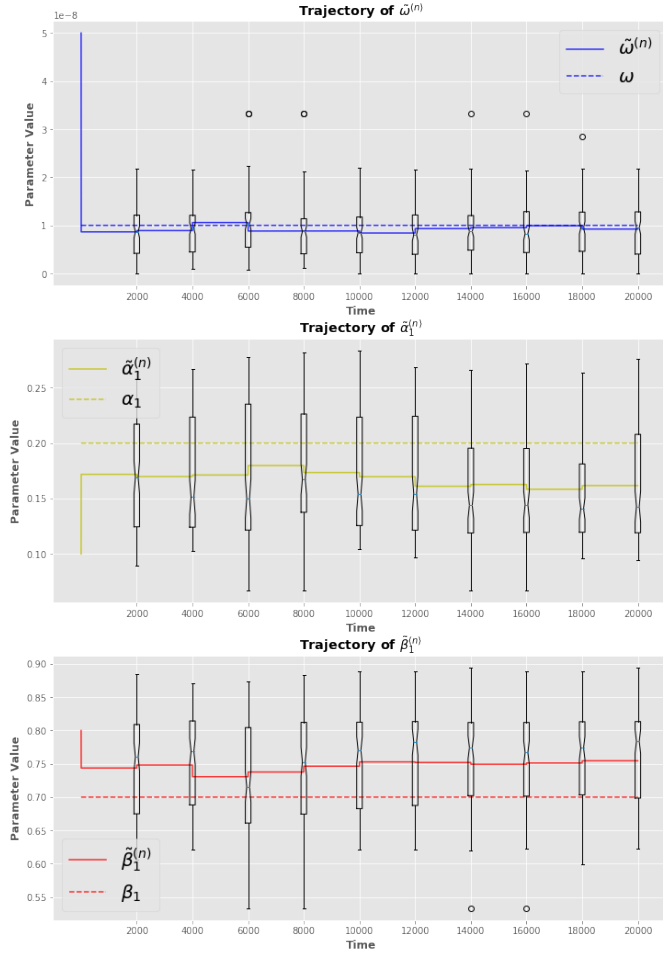


Figure 8: Average trajectory (solid line) of one hundred $\tilde{\theta}_n$'s for a GARCH(1, 1) process with true parameter vector (dotted line) and initial guess given in (4.7). The boxplots shows the distribution of the one hundred trajectories.

erative method is forward-looking).

4.2. Real-life Observations

We will now demonstrate AdaVol's abilities on real-life observations showing how our technique works in practice. Table 2 shows an overview of the used stock market indices. All empirical studies use the GARCH(1, 1) model, but higher-order parameters may yield a better fit for some stock market indices. As the observation period spans over a long time, it is unlikely that the log-return series is stationary. To exhibit AdaVol's ability to adapt to time-varying estimates, we begin by considering the S&P500 Index in Section 4.2.1. Afterward, in Section 4.2.2, we investigate the remaining six stock market indices presented in Table 2, namely the CAC, DAX, DJIA, NDAQ, NKY, and RUT index.

4.2.1. Application to the S&P500 Index

We apply our method on the S&P500 Index from January 1950 to September 2020, consisting of $n = 17672$ observations to test real-life data performance. We employ the GARCH(1, 1)

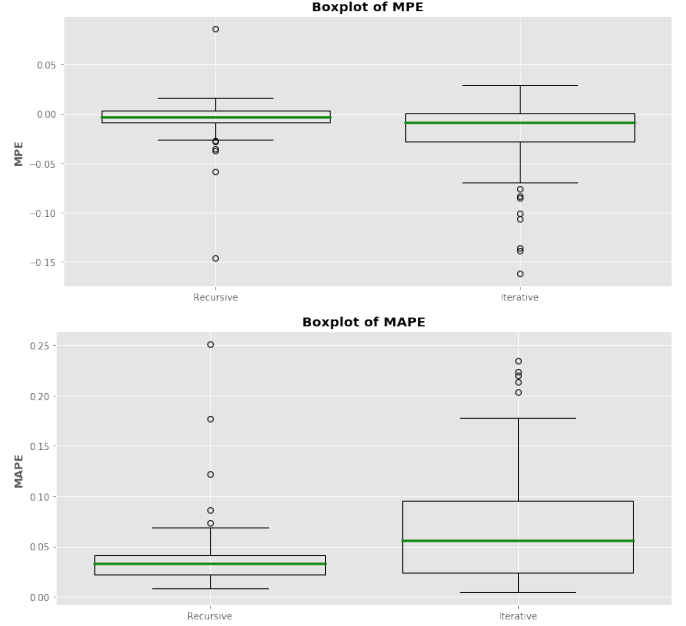


Figure 9: Boxplots of one hundred accuracy scores MPE (4.3) and MAPE (4.4) using a GARCH(1, 1) process with true parameter vector and random initial guess in \mathcal{K} .

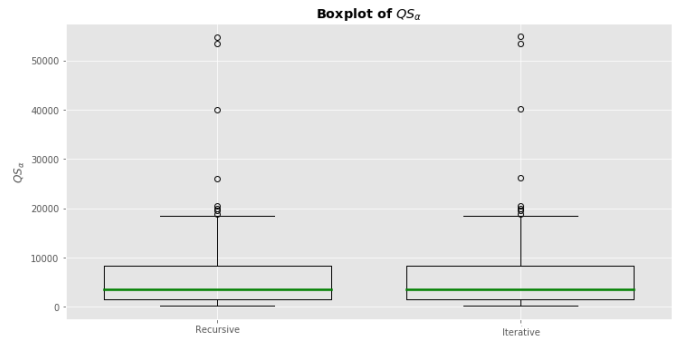


Figure 10: Boxplots of one hundred QS_α scores with $\alpha = \{0.01, 0.02, \dots, 0.99\}$ using the GARCH(1, 1) model with random true parameter vector and initial value in \mathcal{K} .

Stock Market Index		Period
CAC 40	(CAC)	March 1990 - Sep. 2020
DAX 30	(DAX)	Jan. 1988 - Sep. 2020
Dow Jones	(DJIA)	Feb. 1985 - Sep. 2020
NASDAQ Composite	(NDAQ)	Feb. 1971 - Sep. 2020
Nikkei 225	(NKY)	Jan. 1965 - Sep. 2020
Russell 2000	(RUT)	Nov. 1987 - Sep. 2020
Standard & Poor's 500	(S&P500)	Jan. 1950 - Sep. 2020

Table 2: Overview of considered stock market indices including their observation periods. The observations consist of daily log-returns which are defined as log differences of the closing prices of the index between two consecutive days.

model with initial values:

$$\widehat{\theta}_0 = \widetilde{\theta}_0 = \begin{pmatrix} 5 \cdot 10^{-5} \\ 0.05 \\ 0.9 \end{pmatrix}. \quad (4.8)$$

The QML trajectories can be seen in Figure 11. The produced AdaVol estimates $\widehat{\theta}_n = (\widehat{\omega}^{(n)}, \widehat{\alpha}_1^{(n)}, \widehat{\beta}_1^{(n)})^T$ experience some fluctuations initially, but as it vaporizes, it is clear that our estimates change over time. Most remarkable are the shifts our estimates make around some historical market crashes, e.g., Black Monday, the financial crisis, and COVID-19. The instant shift in our estimates is an appealing property for detecting structural breaks. It is noteworthy that the estimates of the IQMLE approximation $\widetilde{\theta}_n = (\widetilde{\omega}^{(n)}, \widetilde{\alpha}_1^{(n)}, \widetilde{\beta}_1^{(n)})^T$ are predominantly constant over time with minor changes except for some years between 1990 and 2000, where we detect a shift to lower $\widetilde{\beta}_1^{(n)}$ values and higher $\widetilde{\omega}^{(n)}$ values.

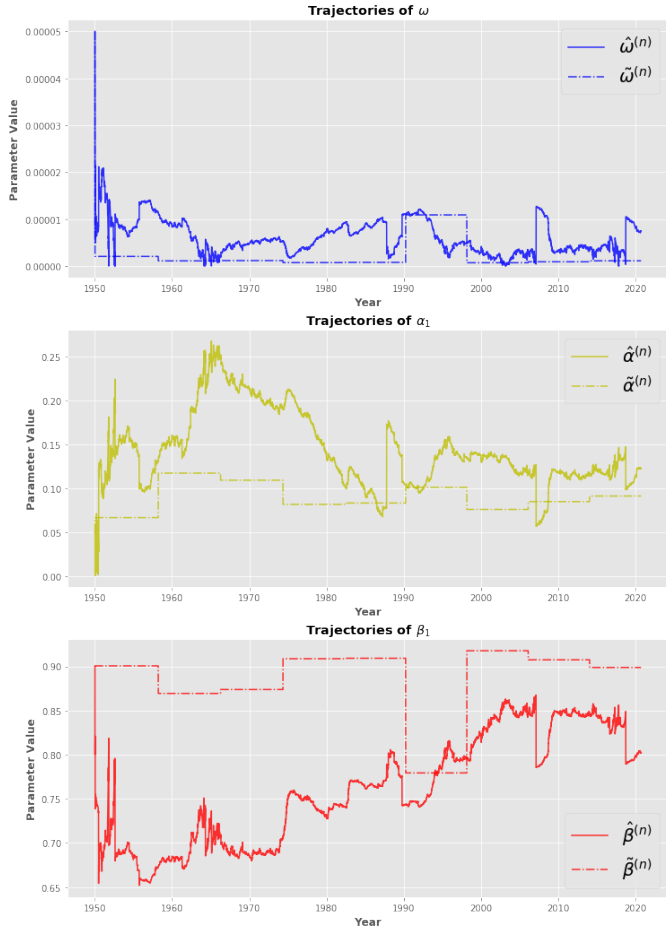


Figure 11: Trajectory of the recursive $\widehat{\theta}_n$ (solid line) and iterative $\widetilde{\theta}_n$ (semi-dotted line) QML estimate using a GARCH(1, 1) model on S&P500 Index log-returns from year 1950 to 2020. Both methods use initial value given in (4.8).

In Figure 12, we have the log-returns r_t of the S&P500 Index, and the confidence intervals $\bar{r} \pm 1.96\widehat{\sigma}_t$ and $\bar{r} \pm 1.96\widetilde{\sigma}_t$ using the recursive $\widehat{\sigma}_t$ and iterative $\widetilde{\sigma}_t$ predicted volatilities, where \bar{r} is the mean of the log-returns r_t . It seems that the recursive method $\widehat{\sigma}_t$ adapts more rapidly than the iterative one $\widetilde{\sigma}_t$ to changes in the S&P500 Index observations r_t . Especially in Figure 12, under the COVID-19 crisis, we encountered a period with a substantial volatility increase. Here, we observe $\widehat{\sigma}_t$'s ability to track changing volatilities better than $\widetilde{\sigma}_t$.

In the absence of the true (unobserved) variance process (σ_t^2),

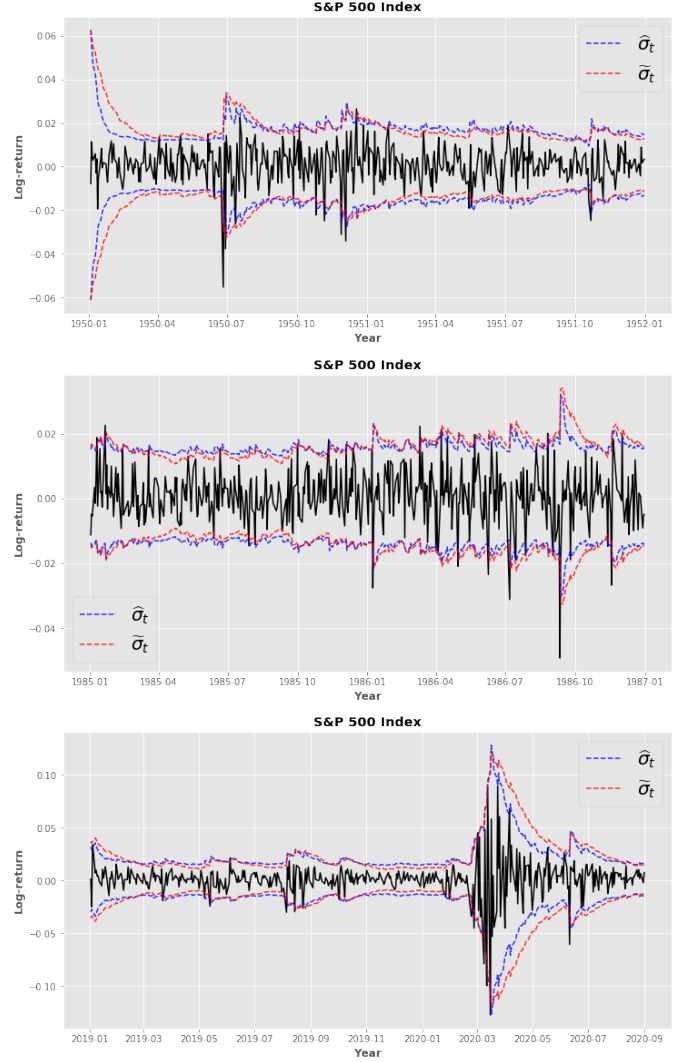


Figure 12: Log-returns r_t of S&P500 Index (solid lines) and confidence intervals $\bar{r} \pm 1.96\widehat{\sigma}_t$ and $\bar{r} \pm 1.96\widetilde{\sigma}_t$ (dotted lines) using the recursive $\widehat{\sigma}_t$ (blue) and iterative $\widetilde{\sigma}_t$ (red) predicted volatilities, where \bar{r} is the mean of the log-returns r_t . From top to bottom, we have Jan. 1950 to Jan. 1952, Jan. 1985 to Jan. 1987, and Jan. 2019 to Sep. 2020.

the efficiency of our recursive ($\widehat{\sigma}_t$) and the iterative ($\widetilde{\sigma}_t$) volatility can be appraised with the use of the squared log-returns (r_t^2). We use the Mean Absolute Errors (MAE) defined by

$$\widehat{\sigma}_{\text{MAE}}^2 = \frac{1}{n} \sum_{t=1}^n |r_t^2 - \widehat{\sigma}_t^2| \text{ and } \widetilde{\sigma}_{\text{MAE}}^2 = \frac{1}{n} \sum_{t=1}^n |r_t^2 - \widetilde{\sigma}_t^2|. \quad (4.9)$$

In Table 3, we consider the MAEs for the same periods used in Figure 12, including for the full dataset. The results in Table 3 confirm our conclusions about Figure 12; the AdaVol method tracks the volatility better than the iterative method.

Figure 13 contains the results of one hundred QS_α scores using the recursive ($\widehat{\sigma}_t$) and iterative ($\widetilde{\sigma}_t$) volatility process, respectively, with random initial values in \mathcal{K} . Remarkably, AdaVol outperforms the iterative method, although the latter uses future information, i.e., $(\widetilde{\sigma}_t)_{(k-2000)+1 \leq t \leq k}$ is estimated using $(r_t)_{1 \leq t \leq k}$ for $k = 2000, 4000, \dots, 16000, 17505$. This indicates

Period	$\widehat{\sigma}_{\text{MAE}}^2$	$\widetilde{\sigma}_{\text{MAE}}^2$
Jan. 1950 - Jan. 1952	8.2388	8.9049
Jan. 1985 - Jan. 1987	7.1214	7.4723
Jan. 2018 - Sep. 2020	26.9205	30.4775
Jan. 1950 - Sep. 2020	10.1861	10.6731

Table 3: MAEs (4.9) using log-returns r_t of S&P500 Index with the recursive $\widehat{\sigma}_t$ and iterative $\widetilde{\sigma}_t$ predicted volatilities. Both methods has initial value given in (4.8). The $\widehat{\sigma}_{\text{MAE}}^2$ and $\widetilde{\sigma}_{\text{MAE}}^2$ numbers are scaled by 10^{-5} .

that one could achieve better performance using the recursive method, even if it only predicts volatility using previous information.

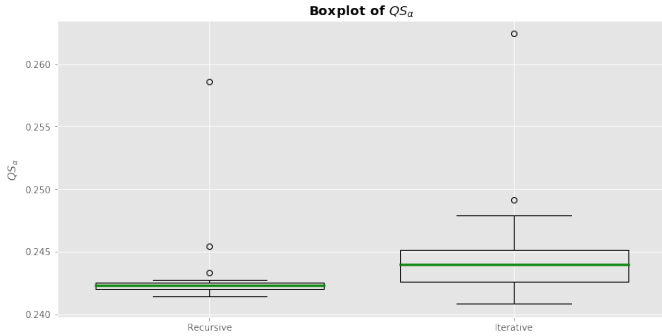


Figure 13: Boxplots of one hundred QS_α scores with use of the recursive $\widehat{\sigma}_t$ and iterative $\widetilde{\sigma}_t$ volatility process, respectively, for $\alpha = \{0.01, 0.02, \dots, 0.99\}$, using the GARCH(1, 1) model on the log-returns r_t of S&P500 Index with random initial value in \mathcal{K} .

4.2.2. Other Stock Market Indices

We now extend our analysis to the remaining stock market indices from Table 2, namely the CAC, DAX, DJIA, NDAQ, NKY, and RUT index. In Figure 14, we can observe AdaVol's ability to adapt to time-varying parameters seems to hold for several stock market indices. These figures show a clear benefit in recursive estimation as it increases adaptivity that may be advantageous under a financial crisis such as the COVID-19.

These conclusions are confirmed in Figure 15, where we have one hundred QS_α scores using the recursive ($\widehat{\sigma}_t$) and iterative ($\widetilde{\sigma}_t$) volatility process with random initial values in \mathcal{K} . As for the S&P500 Index (in Figure 13), our findings indicate that the recursive approach estimates the QS_α quantiles better than the iterative method, both on average and with a lower spread.

The assumption of having an underlying data generation process with constant "true" parameters may not hold in real-life examples. Thus, AdaVol seems to have an advantage compared to the iterative method, as it estimates the parameters step-by-step. In contrast, the iterative method always has to estimate the parameters using all observations over an extensive period of time.

5. Discussion

We proved asymptotic local convexity of the QL function in general conditionally heteroscedastic time series models of

multiplicative form. An interesting question arises: can one prove Theorem 2.1 for a bounded set of N observations? Expressed differently, can one find a N bounded, such that we have convergence/convexity of recursive algorithms, e.g., for the GARCH, EGARCH, and AGARCH models. To our knowledge, this has not been proved yet.

We proposed an adaptive approach to recursively estimate GARCH model parameters in a streaming setting using the VTE technique (AdaVol). AdaVol's design showed to produce resilient and adaptive estimates in our empirical investigations. The adaptation to time-varying parameters was a surprising advantage that appeared when we applied our method to real-life observations. As the assumption of having constant estimates seems not to be the case for the stock indices we analyze, then it is beneficial to have the ability to adapt. One could facilitate this ability more by incorporating a rolling volatility estimation of γ instead of using the sample volatility. Combining this with a different learning rate than AdaGrad, which enables continuous learning (e.g., ADAM by Kingma and Ba (2015)), could encourage adaptability.

The stability of using our recursive approach to solve the QML problem could be improved by using a mini-batch approach. A mini-batch approach will lower each incremental volatility as one uses more observations per recursion to update the QML estimate. Applying a mini-batch method does not require much more computational power than the stochastic gradient descent, only $\mathcal{O}(bd)$, where b is the number of observations used in each (mini-batch) recursion. Using more observations, we could achieve more consistency and smoothness in the estimation procedure's convergence while keeping favorable computational costs.

Furthermore, an accelerated convergence of our estimates could be obtained by recursion averaging, also called Polyak-Ruppert averaging, which is guaranteed under fairly relaxed conditions (Polyak and Juditsky (1992); Ruppert (1988)). This Polyak-Ruppert average estimate could be utilized solely or employed as a benchmark to detect structural breaks.

Acknowledgement

This work was supported by a grant from Région Ile de France.

Appendix A. Proofs

Proof of Theorem 2.1. To prove local strong convexity for the approximate QL function \underline{L}_n using the approximate QMLE $\widehat{\theta}_n$, we first list some bounds for the Hessians: under the regularity conditions on the derivatives of h_t , then using (2.3), we can write

$$\nabla_{\theta} l_t(\theta) = \frac{1}{2} \frac{\nabla_{\theta} h_t(\theta)}{h_t(\theta)} \left(1 - \frac{X_t^2}{h_t(\theta)} \right)$$

and

$$\nabla_{\theta}^2 l_t(\theta) = \frac{1}{2h_t^2(\theta)} \left(\nabla_{\theta} h_t(\theta)^T \nabla_{\theta} h_t(\theta) \left(\frac{2X_t^2}{h_t(\theta)} - 1 \right) + \nabla_{\theta}^2 h_t(\theta) (h_t(\theta) - X_t^2) \right),$$

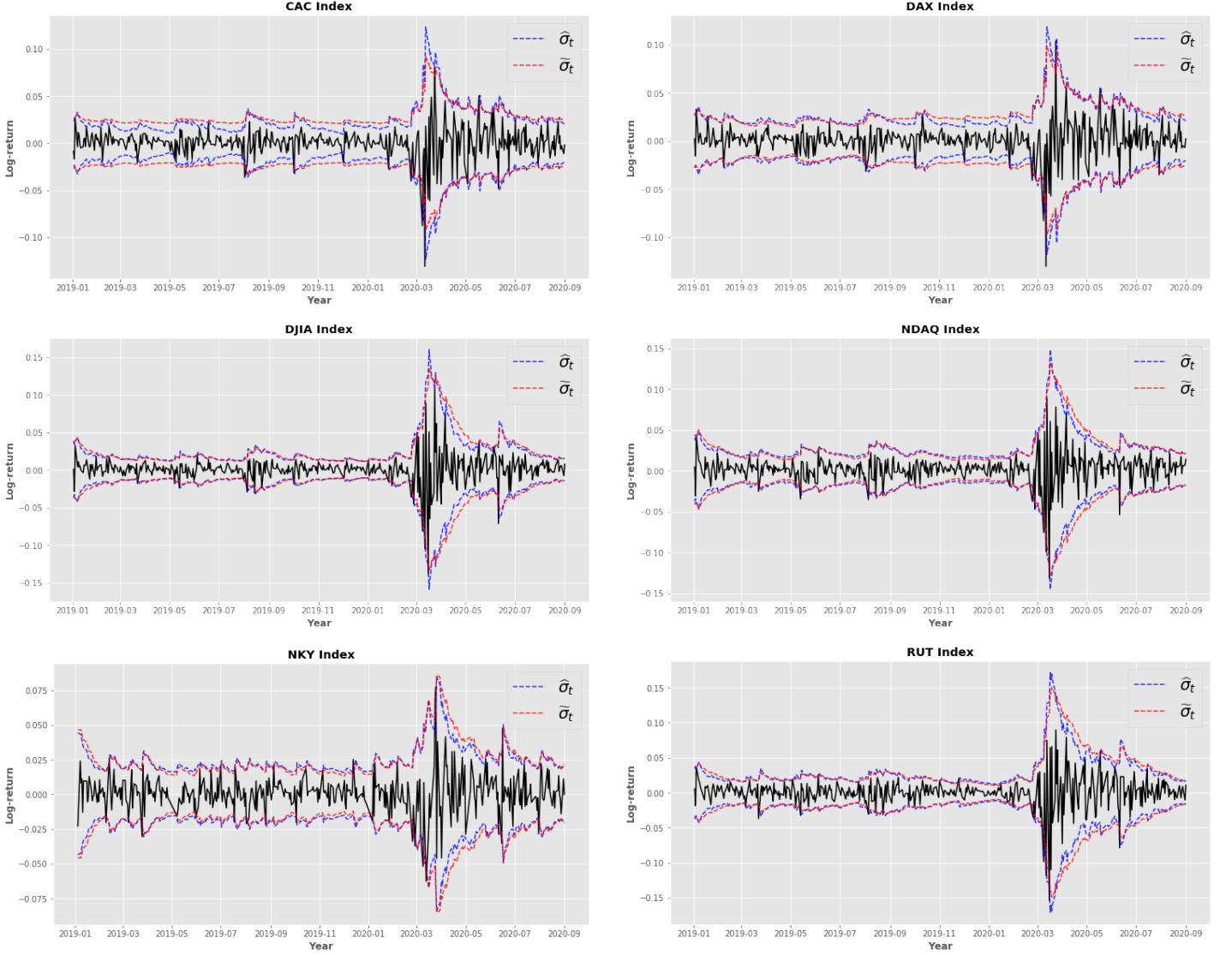


Figure 14: Log-returns r_t of the CAC (top-left), DAX (top-right), DJIA (mid-left), NDAQ (mid-right), NKY (bottom-left) and RUT (bottom-right) index (solid lines) and confidence intervals $\bar{r} \pm 1.96\hat{\sigma}_t$ and $\bar{r} \pm 1.96\bar{\sigma}_t$ (dotted lines) using the recursive $\hat{\sigma}_t$ (blue) and iterative $\bar{\sigma}_t$ (red) predicted volatilities, where \bar{r} is the mean of the log-returns r_t . The period is Jan. 2019 to Sep. 2020.

where the Hessian $H_n(\theta)$ is defined as $n^{-1}\nabla_{\theta}^2 L_n(\theta) = n^{-1}\sum_{t=1}^n \nabla_{\theta}^2 l_t(\theta)$. Similarly, for $\nabla_{\theta}\widehat{l}_t(\theta)$, $\nabla_{\theta}^2\widehat{l}_t(\theta)$, and $\widehat{H}_n(\theta)$, we replace $h_t(\theta)$, $\nabla_{\theta}h_t(\theta)$ and $\nabla_{\theta}^2h_t(\theta)$ by $\widehat{h}_t(\theta)$, $\nabla_{\theta}\widehat{h}_t(\theta)$ and $\nabla_{\theta}^2\widehat{h}_t(\theta)$, respectively. From Assumption W2, we know $n^{-1}\|\nabla_{\theta}^2\widehat{L}_n - \nabla_{\theta}^2L_n\|_{\mathcal{K}} \xrightarrow{\text{a.s.}} 0$ for $n \rightarrow \infty$. Hence, for some random N_1 large enough, there exists $\epsilon > 0$ such that $n^{-1}\|\nabla_{\theta}^2\widehat{L}_n - \nabla_{\theta}^2L_n\|_{\mathcal{K}} < \epsilon$ for all $n \geq N_1$ a.s. As a consequence, we get

$$\|\widehat{H}_n - H_n\|_{\mathcal{K}} < \epsilon, \quad \text{a.s.}, \quad (\text{A.1})$$

for all $n \geq N_1$. Similarly, applying the ergodic theorem on the integrable sequence (uniformly over \mathcal{K}) $(\nabla_{\theta}^2 l_t)$ of continuous functions over the compact set \mathcal{K} , we obtain $\|n^{-1}\sum_{t=1}^n \nabla_{\theta}^2 l_t - \mathbb{E}[\nabla_{\theta}^2 l_0]\|_{\mathcal{K}} \xrightarrow{\text{a.s.}} 0$ for $n \rightarrow \infty$. Then there exists N_2 such that

$$\|H_n - H_0\|_{\mathcal{K}} < \epsilon, \quad \text{a.s.}, \quad (\text{A.2})$$

for all $n \geq N_2$. Thus, by equation (A.1) and (A.2), we know there exists $N = \max(N_1, N_2)$ such that for all $n \geq N$, we have

$$\|\widehat{H}_n - H_0\|_{\mathcal{K}} \leq \|\widehat{H}_n - H_n\|_{\mathcal{K}} + \|H_n - H_0\|_{\mathcal{K}} < 2\epsilon, \quad \text{a.s.}$$

Especially, as $\|\widehat{H}_n - H_0\|_{\mathcal{K}}$ is defined as $\sup_{\theta \in \mathcal{K}} \|\widehat{H}_n(\theta) - H_0(\theta)\|_{op}$, then

$$\|\widehat{H}_n(\theta) - H_0(\theta)\|_{op} < 2\epsilon, \quad (\text{A.3})$$

for all $\theta \in \mathcal{K}$.

From (Straumann and Mikosch, 2006, Lemma 7.2), the asymptotic Hessian $H_0(\theta_0) = \mathbb{E}[\nabla_{\theta}^2 l_0(\theta_0)]$ is a symmetric positive definite matrix a.s. under Assumption W3. As $H_0(\theta)$ is the limit of the continuous matrix-valued function $H_n(\theta)$, it is itself a continuous matrix-valued function. Thus, the eigenvalue function $\lambda_0^i(\theta)$ for $1 \leq i \leq d$ of $H_0(\theta)$ is also continuous. The

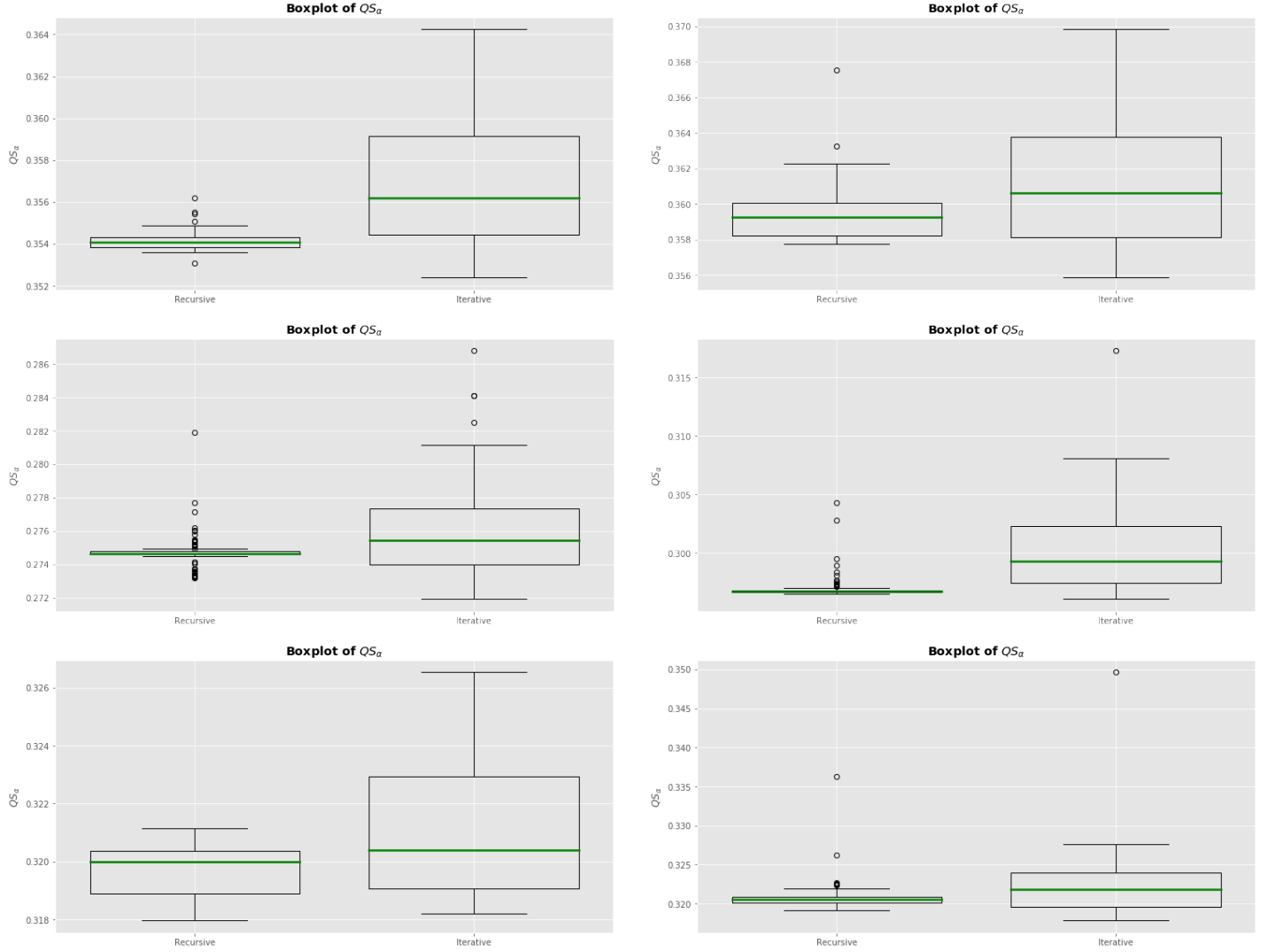


Figure 15: Boxplots of one hundred QS_α scores with the use of the recursive $\widehat{\sigma}_T$ and iterative $\bar{\sigma}_T$ volatility process, respectively, for $\alpha = \{0.01, 0.02, \dots, 0.99\}$, using the GARCH(1, 1) model on the log-returns r_t of the CAC (top-left), DAX (top-right), DJIA (mid-left), NDAQ (mid-right), NKY (bottom-left) and RUT (bottom-right) index with random initial values in \mathcal{K} .

eigenvalues $\lambda_0^i(\theta_0)$ are positive real numbers with the smallest one $\lambda_0^{\min}(\theta_0)$ denoted by

$$\lambda_0^{\min}(\theta_0) = \min_{1 \leq i \leq d} \lambda_0^i(\theta_0) > 0,$$

satisfying $g^T H_0(\theta_0)g \geq \lambda_0^{\min}(\theta_0)g^T g$ for all $g \in \mathbb{R}^d \setminus \{0\}$.

To shorten the notation, we write with no ambiguity $H_0(\theta_0) \geq \lambda_0^{\min}(\theta_0)I_d$ where I_d denotes the d -dimensional identity matrix. By continuity, $\lambda_0^{\min}(\theta)$ is positive on a neighborhood $B(\theta_0, \delta)$ such there exist $\epsilon > 0$ satisfying $\lambda_0^{\min}(\theta_0) - \epsilon > 0$, meaning

$$H_0(\theta) \geq (\lambda_0^{\min}(\theta_0) - \epsilon)I_d,$$

for $\theta \in B(\theta_0, \delta)$. Hence, for $\theta \in B(\theta_0, \delta)$ and $g \in \mathbb{R}^d \setminus \{0\}$, we

have

$$\begin{aligned} \frac{g^T \widehat{H}_n(\theta)}{g^T g} &= \frac{g^T H_0(\theta)g}{g^T g} + \frac{g^T (\widehat{H}_n(\theta) - H_0(\theta))g}{g^T g} \\ &\geq \lambda_{\min} - \epsilon - \frac{g^T \|\widehat{H}_n(\theta) - H_0(\theta)\|_{op} g}{g^T g} \\ &> \lambda_{\min} - 3\epsilon \\ &> C, \quad \text{a.s.}, \end{aligned}$$

using (A.3) for all $n \geq N$ by taking $0 < \epsilon < 6^{-1}\lambda_{\min}$ and letting $C = 2^{-1}\lambda_{\min}$. Then we have the desired inequality (2.5). \square

Proof of Corollary 2.1. The uniqueness of the QMLE $\widehat{\theta}_n^*$ follows from a Pfanzagl argument (Pfanzagl (1969)). By Theorem 2.1, we know there exists N such that

$$\inf_{\theta \in B(\theta_0, \delta_0)} g^T \widehat{H}_n(\theta)g > Cg^T g, \quad \text{a.s.},$$

for all $n \geq N$ where $B(\theta_0, \delta_0)$ denotes the open ball around θ_0 with radius $\delta_0 > 0$. For each element $\theta_i \in \mathcal{K}$, we make an open ball $B(\theta_i, \delta_i)$ for $\delta_i > 0$ such that the union of $B(\theta_i, \delta_i)$ for all i only contains θ_0 once, i.e., $\theta_0 \notin B(\theta_i, \delta_i)$ for $i \neq 0$. As \mathcal{K} is compact and contained in the union of all $B(\theta_i, \delta_i)$, then there is a finite covering of \mathcal{K} , i.e., $\mathcal{K} \subseteq \bigcup_{i=0}^k B(\theta_i, \delta_i)$. Let $\mathcal{K}' = \mathcal{K} \setminus B(\theta_0, \delta_0)$. As \mathcal{K}' is compact, the minimum of the continuous QL function $\mathbb{E}[l_0]$ exists. Moreover, as $\mathbb{E}[l_0]$ is a unique minimum at θ_0 under Assumption W1, we get

$$\inf_{\theta \in \mathcal{K}'} \mathbb{E}[l_0(\theta)] > \mathbb{E}[l_0(\theta_0)], \quad \text{a.s.}$$

From Assumption W2, we know that $\|n^{-1}\widehat{L}_n - L_0\|_{\mathcal{K}'} \xrightarrow{\text{a.s.}} 0$ as $n \rightarrow \infty$. Hence, we have

$$\inf_{\theta \in \mathcal{K}'} n^{-1}\widehat{L}_n(\theta) \xrightarrow{\text{a.s.}} \inf_{\theta \in \mathcal{K}'} L_0(\theta),$$

where $\inf_{\theta \in \mathcal{K}'} L_0(\theta) > \mathbb{E}[l_0(\theta_0)]$. Thus, the $B(\theta_0, \delta_0)$ gives us a unique global minimum of the QL function \widehat{L}_n , i.e.,

$$\inf_{\theta \in \mathcal{K}} n^{-1}\widehat{L}_n(\theta) \geq \mathbb{E}[l_0(\theta_0)], \quad \text{a.s.},$$

where equality only is attained when $\theta = \theta_0$. \square

Appendix B. Relative Speed Comparison

It is argued that the recursive procedure AdaVol is computationally advantageous as it only processes observations once. In order to illustrate this advantage, a relative computational speed comparison as in Sucarrat (2020) is presented. The code is not optimized; it is solely for illustration purposes. In the streaming data framework, the parameters are estimated recursively as described in Section 4. Meaning, for each t , the iterative estimate $\widehat{\theta}_t$ is estimated using the observations $(X_i)_{1 \leq i \leq t}$ and the previous iterative estimate $\widehat{\theta}_{t-1}$ as initialization.

An ARCH(1), GARCH(1, 1), and GARCH(2, 2) model is considered for the computational speed analysis. Table B.4 shows the relative speed comparison for these models with sample sizes $n = 1000$ and $n = 2000$. The overall conclusion is that the AdaVol procedure is faster than the iterative one, e.g., the iterative estimation of a GARCH(1, 1) model is about 205 times slower. Another important observation is the relative speed for different sample sizes n , namely, the larger the sample size n , the greater the relative speed gain is for the AdaVol procedure.

Model	n	AdaVol	arch
ARCH(1)	1000	1.00	163.64
	2000	1.00	190.12
GARCH(1, 1)	1000	1.00	204.89
	2000	1.00	233.86
GARCH(2, 2)	1000	1.00	322.33
	2000	1.00	328.50

Table B.4: Relative speed comparison between AdaVol (Werge (2019)) and arch version 4.15 (Sheppard (2020)). A value of 1.00 means the method is the fastest. A value of 163.64 means the estimation time of the method is 163.64 times larger than the fastest.

References

- Aknouche, A., Guerbyenne, H., 2006. Recursive estimation of garch models. *Communications in Statistics - Simulation and Computation* 35, 925–938.
- Berkes, I., Horváth, L., Kokoszka, P., 2003. GARCH processes: structure and estimation. *Bernoulli* 9(2), 201–227.
- Biau, G., Patra, B., 2011. Sequential quantile prediction of time series. *Information Theory, IEEE Transactions on* 57, 1664–1674.
- Bollerslev, T., 1986. Generalized autoregressive conditional heteroscedasticity. *Journal of Econometrics* 31(3), 307–327.
- Bottou, L., Bousquet, O., 2007. The tradeoffs of large scale learning. *Advances in Neural Information Processing Systems (NIPS)* 20, 161–168.
- Bougerol, P., 1993. Kalman filtering with random coefficients and contractions. *SIAM Journal on Control and Optimization* 31(4), 942–959.
- Bougerol, P., Picard, N., 1992. Stationarity of GARCH processes and of some nonnegative time series. *Journal of Econometrics* 52(1-2), 115–127.
- Cipra, T., Hendrych, R., 2018. Robust recursive estimation of garch models. *Kybernetika -Praha* 54, 1138–1155.
- Dahlhaus, R., Subba Rao, S., 2007. A recursive online algorithm for the estimation of time-varying arch parameters. *Bernoulli* 13, 389–422.
- Duchi, J., Hazan, E., Singer, Y., 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research* 12, 2121–2159.
- Duchi, J., Shalev-Shwartz, S., Singer, Y., Chandra, T., 2008. Efficient projections onto the l_1 -ball for learning in high dimensions. *Proceedings of the 25th International Conference on Machine Learning*, 272–279.
- Engle, R., 1982. Autoregressive conditional heteroscedasticity with estimates of the variance of the united kingdom inflation. *Econometrica* 50(4), 987–1008.
- Francq, C., Zakoian, J.M., 2004. Maximum likelihood estimation of pure garch and arma-garch processes. *Bernoulli* 10, 605–637.
- Francq, C., Zakoian, J.M., Horvath, L., 2011. Merits and drawbacks of variance targeting in garch models. *Journal of Financial Econometrics* 9, 619–656.
- Gerencsér, L., Orlovits, Z., Torma, B., 2010. Recursive estimation of garch processes, in: *The 19th International Symposium on Mathematical Theory of Networks and Systems (MTNS 2010)*, Budapest, Hungary, forthcoming, pp. 2415–2422.
- Hendrych, R., Cipra, T., 2018. Self-weighted recursive estimation of garch models. *Communications in Statistics - Simulation and Computation* 47, 315–328.
- Ip, W.C., Wong, H., Pan, J., Li, D., 2006. The asymptotic convexity of the negative likelihood function of garch models. *Computational Statistics & Data Analysis* 50, 311–331.
- Kierkegaard, J., Jensen, L., Madsen, H., 2000. Estimating garch models using recursive methods.
- Kingma, D., Ba, J., 2015. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*.
- Koenker, R.W., Bassett, G., 1978. Regression quantiles. *Econometrica* 46, 33–50.
- Nelson, D., 1990. Stationarity and persistence in the garch(1,1) model. *Econometric Theory* 6, 318–334.
- Patton, A., 2006. Volatility forecast comparison using imperfect volatility proxies. *Journal of Econometrics* 160, 246–256.
- Pfanzagl, J., 1969. On the measurability and consistency of minimum contrast estimates. *Metrika* 14, 249–272.
- Polyak, B., Juditsky, A., 1992. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization* 30, 838–855.
- Robbins, H., Monro, S., 1951. A stochastic approximation method. *Annals of Mathematical Statistics* 22, 400–407.
- Ruppert, D., 1988. Efficient estimations from a slowly convergent robbins-monro process, in: *Technical Report 781*, Cornell University Operations Research and Industrial Engineering.
- Sheppard, K., 2020. *bashtage/arch*: Release 4.15 (Version 4.15). Zenodo. doi:<https://doi.org/10.5281/zenodo.593254>.
- Straumann, D., 2005. Maximum Likelihood Estimation in Conditionally Heteroscedastic Time Series Models. chapter 5, pp. 85–140.
- Straumann, D., Mikosch, T., 2006. Quasi-maximum-likelihood estimation in conditionally heteroscedastic time series: A stochastic recurrence equations approach. *Annals of Statistics* 34(5), 2449–2495.
- Sucarrat, G., 2020. *garchx: Flexible and Robust GARCH-X Modelling*. MPRA Paper 100301. University Library of Munich, Germany. URL: <https://ideas.repec.org/p/pramprapa/100301.html>.

Tieleman, T., Hinton, G., 2012. Lecture 6.5-rmsprop, coursera: Neural networks for machine learning. University of Toronto, Technical Report.

Ward, R., Wu, X., Bottou, L., 2018. Adagrad stepsizes: Sharp convergence over nonconvex landscapes, from any initialization. [arXiv:1806.01811](https://arxiv.org/abs/1806.01811).

Werge, N., 2019. Adavol. GitHub repository URL: <https://github.com/nhwerge/AdaVol.git>.

Wintenberger, O., 2013. Continuous invertibility and stable qml estimation of the egarch(1,1) model. *Scandinavian Journal of Statistics* 40, 846–867.

Zeiler, M.D., 2012. Adadelta: An adaptive learning rate method. [arXiv:1212.5701](https://arxiv.org/abs/1212.5701).

Zinkevich, M., 2003. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the Twentieth International Conference on Machine Learning* 2, 928–936.

Zumbach, G., 2000. The pitfalls in fitting garch (1, 1) processes, in: *Advances in Quantitative Asset Management*. Springer, pp. 179–200.