Semiparametric smoothing estimators for long-memory processes with added noise

Nuno Crato

Department of Mathematics, ISEG – Instituto Superior de Economia e Gestão, Technical University of Lisbon, R. do Quelhas 6, 1200–781 Lisbon, Portugal

Bonnie K. Ray*

Department of Mathematical Sciences, New Jersey Institute of Technology, Newark, NJ 07102

and

Department of Mathematical Sciences, IBM T.J. Watson Research Center, P.O. Box 218, Yorktown Hts., NY 10598

Abstract: The development of Long Memory Stochastic Volatility (LMSV) models has increased the interest in the estimation of persistent processes observed with added noise. This paper investigates the performance of semi-parametric methods for estimating the long memory-parameter in the long-range dependence plus noise case and demonstrates improvements obtained by preliminary smoothing and aggregation of the series.

Keywords: ARFIMA, LMSV model, long memory, R/S analysis, volatility.

---

* Correspondence author: Bonnie K. Ray, Department of Mathematical Sciences, New Jersey Institute of Technology, Newark, NJ 07102.

E-mail: borayx@m.njit.edu

Fax: (973) 596 5591.

Semiparametric smoothing estimators for long-memory processes with added noise

Abstract: The development of Long Memory Stochastic Volatility (LMSV) models has increased the interest in the estimation of persistent processes observed with added noise. This paper investigates the performance of semi-parametric methods for estimating the long-memory parameter in the long-range dependence plus noise case and demonstrates improvements obtained by preliminary smoothing or aggregation of the series. *Keywords:* ARFIMA, LMSV model, long memory, R/S analysis, volatility.

1 Introduction

In time series analysis, the extraction of signal from noisy observations can be a difficult task. When the signal has long memory, which is difficult to estimate even in the absence of added noise, the estimation of the signal parameters is even more difficult, particularly when the signal-to-noise ratio is small. However, various types of empirical processes have been found to have this structure, namely those encountered in the analysis of financial volatility.

The most widely used class of long-memory models is the Autoregressive Fractionally Integrated Moving Average (ARFIMA) model, which has the form

$$(1 - \phi_1 B + \ldots - \phi_p B^p)(1 - B)^d y_t = (1 + \theta_1 B + \ldots + \theta_q B^q)\eta_t, \tag{1}$$

where B denotes the backward difference operator, *i.e.* $By_t = y_{t-1}$. In this difference equation, y_t is the observed process and η_t is a zero-mean uncorrelated process, *i.e.*, a white noise. The real coefficients ϕ_i and θ_i are such that the AR and MA polynomials have no common roots and all the roots are outside the unit circle. The *fractional integration* parameter d can be any real number. Researchers are typically interested in series having $d \in (0, 1/2)$, for which the process is stationary, yet has autocorrelation decaying to zero very slowly. The ARFIMA model was introduced by Granger and Joyeaux (1980) and Hosking (1981). See Beran (1994) for additional details.

In this paper, we are concerned with the estimation of the integration parameter d when the process y_t is not observed directly. Instead, we assume the analyst observes the process x_t such that

$$x_t = \mu + y_t + \epsilon_t, \tag{2}$$

where μ denotes the mean of the process and ϵ_t is a white noise, independent of y_t. The signal-to-noise ratio is the quantity $\sigma_y^2/\sigma_\epsilon^2$. In order to consider the no-added-noise case in the general setting of equation (2), we will discuss the noise-to-signal ratio

$$ns = \sigma_\epsilon^2/\sigma_y^2 \tag{3}$$

in the remainder of the paper.

The long-memory model with added noise (2) appears naturally in the analysis of financial volatility processes. As shown in the works of Ding, Granger, and Engle (1993), de Lima and Crato (1993), Crato and de Lima (1994), Baillie, Bollerslev, and Mikkelsen (1996), Breidt, Crato, and de Lima (1998), and others, financial returns on the major markets have two striking characteristics. On the one hand, the returns process $r_t = \log p_t - \log p_{t-1}$, constructed on the asset prices p_t, has no significant autocorrelation. On the other hand, the volatility process, computed by squaring the returns or by other methods, is not only autocorrelated but even persistent, *i.e.*, has slowly decaying autocorrelations that are significantly positive for a large number of lags. The Long-Memory Stochastic Volatility (LMSV) model has been used to characterize empirical financial volatility processes.

The LMSV model is defined by

$$r_t = \sigma_t \xi_t, \quad \sigma_t = \sigma \exp\{y_t/2\}, \tag{4}$$

where r_t represents the returns, possibly obtained after some filtering transformation, y_t is an ARFIMA(p, d, q) process independent of ξ_t, and ξ_t is independent and identically distributed (*i.i.d.*) with mean zero and variance one.

In this model, y_t is a sequence of zero-mean uncorrelated variables. The variance of this process, however, displays a dependence determined by y_t. The variance structure is simple to analyze after applying a logarithmic transformation to the squared returns process, giving the following stationary process

$$\begin{aligned} x_t = \log y_t^2 &= \log \sigma^2 + E[\log \xi_t^2] + y_t + \left(\log \xi_t^2 - E[\log \xi_t^2]\right) && (5) \\ &= \mu + y_t + \epsilon_t, && (6) \end{aligned}$$

which is a special form of the ARFIMA model plus noise of equation (2). For details, see Breidt, Crato, and de Lima (1998). The LMSV model has found application in the U.S. and other stock markets (see, *e.g.*, Péerez and Ruiz, 2001) as well as other financial markets (Crato and Ray, 2000).

In practice, the estimated noise to signal ratio ns is very high, causing difficulties in estimating the parameters of the signal. Additionally, it is often difficult both to estimate all the model parameters and to reliably distinguish between competing ARFIMA models having different AR and MA orders. For these reasons, researchers have focused on obtaining reliable estimators of the memory parameter d, which characterizes the long-run properties of the process. The resulting estimate of d can be used either to difference the original series and obtain the remaining ARMA parameters, or simply to assess the long-run properties of the process.

Various semi-parametric estimators of d have been suggested in the literature for the case of no added noise. Theoretical properties have been derived for only one of these estimators, the spectral regression estimator, when the process consists of a long memory signal with added noise. For that estimator, a higher noise to signal ratio was found to significantly negatively bias the resulting estimate (Deo and Hurvich, 2001). Although the work of Robinson and Henry (1999) touches on the issue of long-range dependence in volatility, they focus only on estimation of long memory in the *levels* of a series, showing that estimation of d in the level of a series undisturbed by added noise is unaffected by heteroscedasticity in the error process, even of the long memory type.

In this paper, we investigate the performance of several semi-parametric estimation methods in the long-range dependence plus noise case when preliminary smoothing or aggregation of the series is used to reduce the noise. Taqqu and Teverovsky (1998) have studied the effects of aggregation on some semi-parametric estimators, but without considering added noise. Aggregation reduces the series' length. Fortunately, a common characteristic of financial data is that the time series have high frequency. Thus, we usually have series with many data points, making it possible to aggregate the original series without reducing the sample size to unreasonable levels. We use simulation to assess the performance of these estimators. Although smoothing or aggregation can reduce the noise to signal ratio, it cannot eliminate it completely. Hence, theoretical properties of the estimators in this situation are currently unknown and the general theory suggests itself to be quite difficult.

The rest of the paper is organized as follows. In Section 2, we provide a rationale for the suggested smoothing devices. In Section 3, we describe the estimation methods. In Section 4, we give results of a simulation study to assess the overall performance of the proposed methods. Section 4 concludes.

2 Smoothing and Aggregating

We consider two ways of smoothing a long-memory time series with added noise of the type considered in (2): moving-average filtering and aggregation.

Moving average filter M. By applying a moving average filtering of order k we replace the observed series $x_t,\ t = 1, 2, \ldots, n$ by

$$x'_t = (x_t + x_{t+1} + \ldots + x_{t+k-1})\,/k, \quad \text{for } t = 1, 2, \ldots, n-k+1. \tag{7}$$

The resulting process is a moving average of order $k-1$ of the long-memory signal y_t with an added MA($k-1$) noise independent of y_t.

$$x'_t = \mu + \mathcal{M}(B)y_t + \mathcal{M}(B)\varepsilon_t, \quad \text{with } \mathcal{M}(B) = 1 + B + \ldots + B^{k-1} \tag{8}$$

The long-run properties of the process do not change. In fact, $\{x'_t\}$ is a fractionally integrated process of the same order d. The moving average smoother $\mathcal{M}$ introduces autocorrelation in the added noise, but this autocorrelation vanishes after lag $k-1$. The noise to signal ratio decreases, potentially improving the estimation of d.

Aggregation filter $\mathcal{A}$. Aggregation is another way of filtering the observations, giving the new process

$$\begin{aligned} x'_1 &= x_1 + x_2 + \ldots + x_k, \\ x'_{k+1} &= x_{k+1} + x_{k+2} + \ldots + x_{2k}, \\ &\ldots \\ x'_{[n/k]} &= x_{(k-1)[n/k]} + x_{(k-1)[n/k]+1} + \ldots + x_{k[n/k]} \end{aligned} \tag{9}$$

where $[\cdot]$ represents the greatest integer function.

The resulting process is a new long-memory signal

$$y'_\tau = \sum_{t=k(\tau-1)+1}^{k\tau} y_t = \mathcal{A}(B)y_t \tag{10}$$

with an added noise independent of y'_t:

$$x'_\tau = \mu + \mathcal{A}(B) y_t + \mathcal{A}(B)\varepsilon_t, \quad \text{with } \mathcal{A}(B) = 1 + B + \ldots + B^{k-1}. \tag{11}$$

As the variables x'_τ have no overlapping noise variables ξ_t, the added component $\mathcal{A}(B)\xi_t$ is a white noise, introducing by itself no changes in the autocorrelation structure of the filtered observations. As shown in Teles, Wei, and Crato (1999), if y_t is an ARFIMA(p, d, q), then the process $\mathcal{A}(B) y_t$ is an ARFIMA(p, d, ∞). This infinite-order moving average generates a short-memory factor that does not change the fractional integration order. Chambers (1998) has examined empirically the order of fractional integration for some macroeconomic time series and their aggregates.

For long-memory processes having $d > 0$, aggregation may increase the variance of the signal relative to changes in the noise variance. In fact,

$$\text{Var}[y'_\tau] = k\text{Var}[y_t] + 2\sum_{j=1}^{k-1} \text{Corr}[y_t, y_{t-j}], \tag{12}$$

which increases the variance of y_t by more than a factor of k when the second term is positive. This is the case for a persistent fractional noise, *i.e.*, for an ARFIMA($0, d, 0$), since all of its autocorrelation values are positive. It is also the case for a general ARFIMA(p, d, q) case, provided a sufficiently large k is chosen. In contrast, the variance of the aggregated noise is simply

$$\text{Var}[\xi'_\tau] = k\text{Var}[\xi_t]. \tag{13}$$

As far as the estimation of d alone is concerned, aggregation has the further potential advantage of reducing the short-term autocorrelation in the signal. However, the infinite-order moving average introduced in the aggregated signal may disturb the estimation of d. We investigate these issues in Section 4.

3 Estimation procedures

We focus on three commonly used methods for estimating d for the smoothed series. The original series appears as a particular case of these smoothed series, corresponding to the $k = 1$ situation. Below, we briefly explain the procedures used.

Spectral regression—GPH. The periodogram-based method suggested by Geweke and Porter-Hudak (1983), hereafter GPH, is one of the most widely used methods for estimating the memory parameter d in the absence of added noise. It is based on the form of the spectral density function, $f(\lambda)$, of a long-memory process, which can be written as:

$$f(\lambda) = |1 - e^{-i\lambda}|^{-2d} f_U(\lambda), \quad \text{for } -\pi < \lambda \leq \pi, \tag{14}$$

where f_U is a slowly varying function, finite and bounded above from zero at the origin.

For a given time series, Geweke and Porter-Hudak (1983) suggested regressing the log of the estimated spectrum of the series on the log of the frequency argument λ. They also suggested that the regression be performed using a set of Fourier frequencies close to zero, where the slope of the log spectrum relative to the frequency is mostly dependent on the long-memory parameter d. They argued that their regression estimator could capture the long-memory characteristic of the process without being "contaminated" in the estimation by short-memory correlation in the time series evident at higher frequencies.

Use of this method requires choice of a truncation parameter m to determine the number of Fourier frequencies used. Based on simulations, Geweke and Porter-Hudak suggested the use of $m = [n^u]$ with $u = .5$, where n is the number of observations and $[\cdot]$ is the greatest integer function. Robinson (1995a) argued that a consistent estimator for d could be obtained if an additional low-order truncation $l > 1$ was introduced, thus using the Fourier frequencies $j = l + 1, \ldots, m$. More recent results by Hurvich, Deo, and Brodsky (1998) find that this lower truncation is not necessary. They conclude that setting $l = 1$, *i.e.*, no lower truncation and $[n^{.8}]$ for the upper truncation is a more optimal choice. The distribution of the estimated d parameter can be shown to be Gaussian under certain conditions.

The spectral density of the long-memory plus noise model adds the spectrum of the noise to equation (14). If the noise is white, this additional term is simply $\sigma_\varepsilon^2/(2\pi)$. Failure to account for the added noise component biases the resulting estimate of d. Deo and Hurvich (2001) derived precisely the mean squared error of the GPH estimator for the LMSV model. They showed that the estimate of d is negatively biased, with bias increasing as m increases and as ns increases.

In the case of an MA-filtered ARFIMA(p, d, q) model with added noise, the spectral density function is

$$f(\lambda) = \frac{\sigma_\eta^2 |\mathcal{M}(e^{-i\lambda})\theta(e^{-i\lambda})|^2}{2\pi\, \left|1 - e^{-i\lambda}\right|^{2d} |\phi(e^{-i\lambda})|^2} + \frac{\sigma_\varepsilon^2 |\mathcal{M}(e^{-i\lambda})|^2}{2\pi}, \quad -\pi < \lambda \leq \pi, \tag{15}$$

where $\mathcal{M}(e^{-i\lambda}) = \sum_{j=0}^{k} e^{-i2\lambda}$.

In the case of an aggregated ARFIMA(p, d, q) model with added noise, the spectral density function is

$$f(\lambda) = \frac{\sigma_\eta^2 |\theta_\infty(e^{-i\lambda})|^2}{2\pi\, \left|1 - e^{-i\lambda}\right|^{2d} |\phi(e^{-i\lambda})|^2} + \frac{k\sigma_\varepsilon^2}{2\pi}, \quad -\pi < \lambda \leq \pi, \tag{16}$$

where $\theta_\infty(\cdot)$ represents the new MA(∞) polynomial introduced in the signal process. Its expression has a complicated analytical form. It suffices to say that in a sufficiently small neighborhood of zero, it is dominated by the term in d (Teles, Wei, and Crato 1999).

Truncated Whittle—TW. Robinson (1995b) introduced the use of a Gaussian semi-parametric method for estimating d. The method follows from the fact that the negative log-likelihood function of a Gaussian ARFIMA(p, d, q) process, denoted by L, can be approximated (up to an additive constant) using a method due to Whittle (1953) as

$$L = \frac{1}{\pi}\left[\sum_{j=1}^{n^*} \frac{I_n(\lambda_j)}{f(\lambda_j)} \frac{2\pi}{n} + \sum_{j=1}^{n^*} \log f(\lambda_j) \frac{2\pi}{n}\right], \tag{17}$$

where n^* denotes the integer part of $(n-1)/2$ and $f(\lambda_j)$ is the spectrum of the ARFIMA process at Fourier frequency λ_j. From (14), we have

$$L \approx \frac{1}{m}\left[\sum_{j=1}^{m} \frac{I_n(\lambda_j)}{c\lambda_j^{-2d}} + \sum_{j=1}^{m} \log c\lambda_j^{-2d}\right] \tag{18}$$

in a neighborhood of zero frequency determined by m. An estimate of d is obtained by minimizing L. Robinson and Henry (1999) have shown that the limiting distribution of this estimator is not affected by the presence of heteroskedasticity. As with the GPH estimator, failure to account for the added noise term in the spectrum of x_t causes bias in the resulting estimate.

Variance Ratio—VR. One of the key properties of a long-memory process is that the variance of the sample mean converges to zero at rate $O(n^{2d-1})$, in contrast to the rate

$O(n^{-1})$ for short-range dependent processes, such as stationary ARMA. This property motivates a semi-parametric estimate of d obtained by the following procedure: 1) compute the sample average for each of m_k subseries of length k of the original series, for a range of k values $2 \le m_0 \le k \le m_1 \le [n/2]$. 2) compute the sample standard deviation of the m_k sample averages for each k. 3) regress the logarithm of the sample variance for each k on the logarithm of k, obtaining regression coefficient $\hat{\theta}$. Then $\hat{d}$ corresponds to $(\hat{\theta}+1)/2$. Giraitis, Robinson, and Surgailis (1999) derive the asymptotic properties of this Variance-Ratio (VR) type estimator for long-range dependent processes undisturbed by added noise. Taqqu and Teverovsky (1998) report results of an empirical study showing that this estimator is well-behaved, again for processes undisturbed by added noise. The range of k determined by m_0 and m_1 is chosen to balance the trade-off between bias and variance in the resulting estimate.

In the long-range signal plus white noise case, we have

$$\mathrm{Var}(\bar{x}_k) \approx c_1 k^{2d-1} + c_2 k^{-1}$$

when k is large, where $\bar{x}_k$ denotes the sample average of a subseries of x_t of length k. The variance of the sample average due to the added noise component decreases at a faster rate than the component of variance due to the long-memory signal. Thus it is reasonable to believe that the VR method may provide adequate estimates of d even in the LM signal plus added noise case. Since the VR method implicitly depends on aggregation of the process for estimation, we investigate its performance only for the original, unfiltered series. The VR estimator is closely related to wavelet methods for analyzing long-memory signals (Abry, Veitch, and Flandrin, 1998). We investigate the size of this bias and the improvements due to smoothing and aggregation in the next section.

4 Simulation study

We conducted a simulation study to assess the performance of the three estimation procedures on smoothed or aggregated long memory signals disturbed by noise. We first generated Gaussian ARFIMA$(0, d, 0)$ processes of length n using the method described by Hosking (1984). Included values of d were $d = 0.2, 0.35, 0.45$. These values give processes which are stationary and long-range dependent. To each series, a Gaussian white noise was added to give a specified noise-to-signal ratio, ns. For sample sizes

$n = 1000$ and 5000, we considered four different noise to signal ratios: $ns = 0, 2, 5, 10$. The value $ns = 0$ corresponds to an ARFIMA process with no added noise. Deo and Hurvich (2001) show that negative bias of the GPH estimate in a LMSV model increases with ns. We generated 1000 series for each value of d, ns, and n.

To each series, we applied an MA$(k-1)$ filter with $k = 1, 10, 20, 30$. The value $k = 1$ corresponds to no filtering. We also aggregated each series using block lengths $k = 1, 2, 5, 10$. To each smoothed (MA filtered or aggregated) series, we applied the two spectral-based estimation methods: GPH and TW. The GPH and TW methods were applied using truncation values $m = n^{0.4}, m = n^{0.5}, m = n^{0.6}, m = [(n-1)/2]$. For fixed ns, Deo and Hurvich (2001) show that negative bias of the GPH estimate increases with m for a LMSV model. The VR method was applied only to the unfiltered series. Since the VR method implicitly incorporates aggregation, it does make sense to apply this method to series which have already been smoothed.

Tables I and II give results for the GPH method applied to the MA filtered and aggregated data, respectively, in the case $d = 0.35$, while Tables III and IV show corresponding results for the TW method. Values of d in the range 0.30 to 0.40 have been reported for many types of financial volatility series. See, for example, Ray and Tsay (2000). Results for other values of d show similar patterns. Complete results can be obtained from the authors upon request.

Spectral Regression—GPH

The first vertical block of Table I gives results when there is no added noise ($ns = 0$). The case $k = 1$ corresponds to no smoothing, *i.e.*, the results are those typically found when the GPH method is applied to standard ARFIMA$(0, d, 0)$ processes. Specifically, the estimates are quite variable, with variance decreasing as m increases. For the same no-added noise case ($ns = 0$), we notice an increasing positive bias as k increases. This is naturally due to the introduction of an MA component, as discussed in Section 2. A smaller truncation value, m, is needed to mitigate the effect of the MA component on the estimate of d. These results indicate that a tradeoff exists between the order of MA filtering and the selected truncation point in the estimation procedure: the first introduces a positive bias while the second introduces large variability.

As the added noise becomes prominent, a serious negative bias in the estimates is apparent, which increases with m. This result is expected from the theoretical results of Deo and Hurvich (2001), but the current simulations show its magnitude. By simple visual inspection of the first horizontal block of the table ($k = 1$), we notice that this bias increases dramatically as the noise-to-ratio parameter ns increases. It is worthwhile to consider the case $ns = 10$, which corresponds to the order of magnitude of currently assumed added noise in many financial time series. For this value of d, there is no hope of reliably estimating the memory parameter using GPH without some kind of smoothing unless n is extremely large. As the MA filtering parameter k increases, the estimates for d increase, but there are still large negative biases when $m < [n/2]$.

————INSERT TABLE I HERE————

Table II shows the estimates for the aggregated data. The first horizontal block of the table ($k = 1$) is essentially identical to the first block of the previous one. The differences only show the variability of the GPH estimator for ARFIMA$(0, d, 0)$ processes.

Concentrating on the first vertical block of the table ($ns = 0$), we notice again a positive bias due to the positive correlation introduced by aggregation. However, this bias is much smaller than the one obtained using MA filtering. When the added noise is predominant, a small truncation point again produces less negatively biased estimates. Even the largest level of aggregation ($k = 10$ in our study) does not offset this negative bias due to the added noise component. This is in contrast to the MA filtering case, in which using even a modest truncation value could result in tremendous upward biases. Even with the reduction of degrees of freedom due to the aggregation of the series, the variance of the estimates is not much larger than that obtained using MA filtering. Thus the aggregation method would appear to have some advantages over the moving average smoothing method. It appears promising for very long financial series, on the order of $n \geq 10,000$.

————INSERT TABLE II HERE————

Truncated Whittle—TW

Tables III and IV show the results of the spectral approximate maximum likelihood method due to Whittle for the smoothed and aggregated series. As expected given its *theoretical* asymptotic variance, we see a sizeable decrease in variability over the GPH method, at the expense of a greater computational complexity.

For the MA filtered case (Table III), we note that the effect of this type of smoothing is again the introduction of a positive bias. This is clearly seen in the vertical block corresponding to the simple fractional noise case ($ns = 0$). In general, we also notice a reduction in the bias with the increase of the sample size n. When smoothing and noise are introduced, there is again a combined effect of positive bias due to smoothing and negative bias due to added noise. The optimal results are obtained for a moderate truncation value, say $m = [n^{0.5}]$ or $m = [n^{0.6}]$. This means that more spectral information should be used when applying Whittle's method relative to the GPH procedure. The order of truncation is dependent on the sample size. A smaller n requires a smaller truncation parameter.

————INSERT TABLES III, IV HERE————

For the aggregation case (Table IV), the biases are almost uniformly negative except when $ns = 0$. However, this suggests that a positive estimate of d from the truncated Whittle method provides strong evidence of persistence in the data. Testing for long memory with truncated Whittle estimates can reliably point to a positive d parameter when positive estimates of $\hat{d}$ are obtained, even if the estimated d values are smaller than the actual d value. As expected, there is an increase in variance due to the reduction of degrees of freedom introduced by the aggregation.

Variance Ratio—VR

Table V presents the results for estimates obtained using the Variance Ratio method. We used subseries of length k, with $k = 50, 51, 52, \ldots, [n/10]$. A negative bias is seen for all values of d in our study, even when $ns = 0$. This again suggests that a positive estimate of d for a particular dataset can be taken as an indication of persistence. The variability of the estimates corresponds roughly to that obtained for aggregated data of order $k = 5$ when $m = [n{\cdot}6]$. Comparing the bias size to that obtained using for the MA filtered or aggregated data, we see that the negative bias does not increase as dramatically as the noise-to-signal ratio ns increases. This again suggests that for long series having strong persistence, the VR method can be employed for testing the existence of long memory.

————INSERT TABLE V HERE————

5 Discussion

Standard methods for estimating d in a long memory process, when applied to a long-memory signal with added noise, suffer from biases due to failure to account for the added noise variance, which appears in the spectrum and, correspondingly, the variance of the x_t process. The simulation results presented here for the $ns = 0$ case show the magnitude of these biases, which have not been previously studied for the TW or VR method. These results show that MA filtering or aggregation can be used to reduce the bias, although introducing a smoothing order k that must be selected by the user. Aggregation also serves to increase the variance of the resulting estimates, as the sample size is reduced by a factor $1/k$. As for the standard ARFIMA model, the TW estimation method, although still exhibiting negatives biases, is preferred over the GPH method due to its smaller variability. Estimation based on the aggregated data or using the VR method can be recommended for assessing the existence of long memory in long financial time series, even though the estimated d values are expected to be smaller than the actual d. A method for optimally selecting both k, the order of aggregation, and m, the number of Fourier frequencies, along the lines of Hurvich and Deo (2001), is a matter for future research.

Other methods for estimating d using the form of the sample autocorrelation function (ACF) of an ARFIMA(p,d,q) process $\gamma(k)$ when k is large have been investigated by Delgado and Robinson (1994) and Hall, Koul, and Turlach (1997), among others. Since the ACF of an ARFIMA disturbed by white noise is exactly the same as that of the ARFIMA(p,d,q) signal x_t at lags $k > 0$, it is reasonable to consider such methods for this case. However, Hosking (1996) shows that the sample ACF of an ARFIMA(p,d,q) series is grossly underestimated in finite samples when the mean of the process is unknown and $d \geq 0.25$. The results of Hall, Koul, and Turlach (1997) show that estimators of d based on the sample ACF do not have a normal limiting distribution when $d \geq 0.25$. Thus a semiparametric estimator of d based on the sample ACF of x_t for the ARFIMA signal plus white noise case is not expected to perform well over the range of d values of interest. In fact, Wright (1999) investigated a generalized-method-of-moments (GMM) estimator for d in the LMSV case, which is a generalization of the method of Delgado and Robinson (1994), and found this to be the case.

The performance of modified semiparametric estimation methods for d in the long-memory signal plus noise case, which explicit account for the added noise variance σ_u^2, remains a matter for future research. Breidt, Crato, and de Lima (1998) and Pérez and Ruiz (2001) have used the Whittle approximation to the log-likelihood of an LMSV process for fully parametric estimation of all process parameters, with good results when the noise-to-signal ratio is not too large.

References

Abry, P., Veitch, D., and Flandrin, P. (1998). Long-range dependence: revisiting aggregation with wavelets. *Journal of Time Series Analysis*, 19, 253–266.

Baillie, R. T., Bollerslev, T., and Mikkelsen, H. O. (1996): Fractionally integrated generalized autoregressive conditional heteroskedasticity, *Journal of Econometrics*, 74, 3–30.

Beran, J. (1994), *Statistics for Long Memory Processes*, New York: Chapman & Hall.

Breidt, F. J., Crato, N., and de Lima, P. J. F. (1998): The detection and estimation of long memory in stochastic volatility, *Journal of Econometrics*, 83, 325–348.

Chambers, M. J. (1998). Long memory and aggregation in macroeconomic time series. Symposium on Forecasting and Empirical Methods in Macroeconomics and Finance. *International Economic Review*, 39, 1053–1072.

Crato, N. and de Lima, P. J. F. (1994): Long-range dependence in the conditional variance of stock returns, *Economics Letters* **45** 3, 281–285.

Crato, N. and Ray, B.K. (2000). Memory in returns and volatilities of futures' contracts, *The Journal of Futures Markets*, 20, 525–544.

De Lima, P. and Crato, N. (1993). Long-memory in stock returns and volatilities, American Statistical Association, Proceedings of the Business and Economic Statistics Section.

Delgado, M. and Robinson, P. (1994). New methods for the analysis of long-memory time series: applications to Spanish inflation, *Journal of Forecasting*, 13, 97–107.

Deo, R. S. and Hurvich C. M. (2001): On the log periodogram regression estimator of the memory parameter in long memory stochastic volatility models, *Econometric Theory*, forthcoming.

Ding, Z., Granger, C., and Engle, R. F. (1993). A long memory property of stock market returns and a new model, *Journal of Empirical Finance*, 1, 83–106.

Geweke, J. and Porter-Hudak, S. (1983). The estimation and application of long memory time series models, *Journal of Time Series Analysis* **4**, 4, 221-238.

Giraitis, L., Robinson, P. M., and Surgailis, D. (1999). Variance-type estimation of long memory. *Stochastic Processes and their Applications*, 80, 1–24.

Granger, C. W. J. and Joyeux, R. (1980). An introduction to long-memory time series models and fractional differencing, *Journal of Time Series Analysis* **1**, 15–29.

Hall, P., Koul, H., and Turlach, B. A. (1997). Note on convergence rates of semiparametric estimators of dependence index. *Annals of Statistics*, 25, 1725–1739.

Hosking, J. R. M. (1996). Asymptotic distributions of the sample mean, autocovariances, and autocorrelations of long-memory time series. *Journal of Econometrics*, 73, 261–284.

Hosking, J. R. M. (1981). Fractional differencing, *Biometrika* **68**, 1, 165-176.

Hosking, J.R.M. (1984). Modeling persistence in hydrological time series using fractional differencing. *Water Resources Research* **20**, 1898-1908.

Hurvich, C. M. and Deo, R. S. (1999). Plug-in selection of the number of frequencies in regression estimates of the memory parameter of a long-memory time series. *Journal of Time Series Analysis*, 20, 331–341.

Hurvich, C. M., Deo, R. S., and Brodsky, J. (1998). The mean squared error of Geweke and Porter-Hudak's estimator of the memory parameter of a long memory time series, *Journal of Time Series Analysis*, 19: 19–46.

Pérez, A. and Ruiz, E. (2001). Finite sample properties of a QML estimator of stochastic volatility models with long memory, *Economics Letters* **70**, 157–164.

Ray, B.K. and Tsay, R.S. (2000). Size effects on common long-range dependence in stock volatilities, *Journal of Business & Economic Statistics* **18**, 254–262.

Robinson, P. M. (1995a). Log-periodogram regression of time series with long range dependence, *Annals of Statistics* **23**, 1048–1072.

Robinson, P. M. (1995b). Gaussian semiparametric estimation of long range dependence. *Annals of Statistics* **23**, 1630–1661.

Robinson, P.M. and Henry, M. (1999). Long and short memory conditional heteroskedasticity in estimating the memory parameter on levels, *Econometric Theory* **15**, 299–336.

Teles, P., Wei, W.W.S., and Crato, N. (1999). The use of aggregate time series in testing for long memory, *Bulletin of the International Statistical Institute, 52nd Session*, 1999, 341–342.

Taqqu, M.S. and Teverovsky, V. (1998). On estimating the intensity of long-range dependence in finite and infinite variance time series, *A Practical Guide to Heavy Tails*, Birkhäuser, Boston, 177–217.

Wright, J. (1999). A new estimator of the fractionally integrated stochastic volatility model, *Economics Letters* **63**, 295–303.

Table I: GPH Estimates with MA Filtered Data — $d = 0.35$

		$ns = 0$		$ns = 2$		$ns = 5$		$ns = 10$	
k	$m \quad \backslash n$	1000	5000	1000	5000	1000	5000	1000	5000
1	$m = [n^{0.4}]$	0.363	0.320	0.297	0.286	0.230	0.260	0.179	0.222
		(0.211)	(0.171)	(0.213)	(0.172)	(0.222)	(0.169)	(0.219)	(0.174)
1	$m = [n^{0.5}]$	0.358	0.334	0.268	0.293	0.199	0.249	0.141	0.201
		(0.135)	(0.102)	(0.136)	(0.104)	(0.139)	(0.100)	(0.138)	(0.101)
1	$m = [n^{0.6}]$	0.355	0.345	0.237	0.277	0.164	0.221	0.114	0.169
		(0.091)	(0.064)	(0.087)	(0.063)	(0.090)	(0.062)	(0.091)	(0.061)
1	$m = [n/2]$	0.355	0.372	0.149	0.177	0.087	0.110	0.055	0.070
		(0.034)	(0.017)	(0.030)	(0.015)	(0.032)	(0.016)	(0.032)	(0.016)
10	$m = [n^{0.4}]$	0.372	0.316	0.304	0.292	0.239	0.260	0.175	0.220
		(0.222)	(0.172)	(0.220)	(0.170)	(0.218)	(0.175)	(0.216)	(0.167)
10	$m = [n^{0.5}]$	0.401	0.339	0.309	0.299	0.241	0.251	0.180	0.205
		(0.138)	(0.105)	(0.139)	(0.101)	(0.134)	(0.103)	(0.139)	(0.102)
10	$m = [n^{0.6}]$	0.530	0.361	0.414	0.295	0.344	0.236	0.285	0.183
		(0.092)	(0.065)	(0.091)	(0.063)	(0.087)	(0.062)	(0.092)	(0.062)
10	$m = [n/2]$	1.209	1.270	1.029	1.090	0.973	1.025	0.940	0.988
		(0.043)	(0.025)	(0.034)	(0.018)	(0.033)	(0.018)	(0.034)	(0.018)
20	$m = [n^{0.4}]$	0.407	0.320	0.343	0.293	0.278	0.263	0.213	0.232
		(0.228)	(0.174)	(0.220)	(0.167)	(0.221)	(0.173)	(0.218)	(0.167)
20	$m = [n^{0.5}]$	0.543	0.348	0.460	0.306	0.392	0.261	0.334	0.218
		(0.142)	(0.105)	(0.136)	(0.103)	(0.133)	(0.101)	(0.136)	(0.100)
20	$m = [n^{0.6}]$	1.123	0.413	1.043	0.345	0.980	0.287	0.934	0.238
		(0.111)	(0.063)	(0.104)	(0.062)	(0.100)	(0.061)	(0.103)	(0.062)
20	$m = [n/2]$	1.247	1.331	1.097	1.160	1.043	1.098	1.012	1.061
		(0.055)	(0.030)	(0.037)	(0.020)	(0.035)	(0.018)	(0.035)	(0.018)
30	$m = [n^{0.4}]$	0.473	0.323	0.405	0.295	0.342	0.264	0.288	0.232
		(0.221)	(0.174)	(0.221)	(0.173)	(0.217)	(0.168)	(0.219)	(0.170)
30	$m = [n^{0.5}]$	0.880	0.367	0.795	0.322	0.731	0.276	0.683	0.232
		(0.139)	(0.104)	(0.141)	(0.106)	(0.135)	(0.100)	(0.136)	(0.101)
30	$m = [n^{0.6}]$	1.069	0.506	0.970	0.442	0.904	0.382	0.856	0.331
		(0.097)	(0.066)	(0.096)	(0.062)	(0.091)	(0.062)	(0.093)	(0.062)
30	$m = [n/2]$	1.251	1.344	1.119	1.181	1.068	1.120	1.040	1.084
		(0.064)	(0.035)	(0.039)	(0.020)	(0.036)	(0.019)	(0.035)	(0.018)

The order q of the MA filter is $q = k - 1$, since k is the number of observations used for each filtered data point. Values below the estimates for d in parentheses are the standard errors for the estimates obtained by the simulations.

Table II: GPH Estimates with Aggregated Data — $d = 0.35$

		$ns = 0$		$ns = 2$		$ns = 5$		$ns = 10$	
k	$m \quad \backslash n$	1000	5000	1000	5000	1000	5000	1000	5000
1	$m = [n^{0.4}]$	0.348	0.319	0.280	0.296	0.236	0.252	0.159	0.219
		(0.219)	(0.169)	(0.218)	(0.170)	(0.216)	(0.172)	(0.227)	(0.168)
1	$m = [n^{0.5}]$	0.356	0.338	0.267	0.294	0.198	0.243	0.139	0.199
		(0.141)	(0.106)	(0.134)	(0.103)	(0.134)	(0.104)	(0.139)	(0.101)
1	$m = [n^{0.6}]$	0.353	0.345	0.238	0.277	0.164	0.217	0.112	0.164
		(0.092)	(0.064)	(0.086)	(0.063)	(0.091)	(0.063)	(0.089)	(0.060)
1	$m = [n/2]$	0.355	0.372	0.150	0.177	0.088	0.110	0.056	0.070
		(0.033)	(0.017)	(0.029)	(0.015)	(0.033)	(0.015)	(0.033)	(0.016)
2	$m = [n^{0.4}]$	0.360	0.313	0.298	0.279	0.228	0.255	0.182	0.203
		(0.251)	(0.197)	(0.249)	(0.214)	(0.254)	(0.203)	(0.258)	(0.200)
2	$m = [n^{0.5}]$	0.362	0.331	0.279	0.296	0.209	0.254	0.153	0.203
		(0.164)	(0.121)	(0.165)	(0.126)	(0.167)	(0.122)	(0.169)	(0.125)
2	$m = [n^{0.6}]$	0.356	0.343	0.262	0.285	0.187	0.235	0.133	0.182
		(0.115)	(0.077)	(0.114)	(0.076)	(0.115)	(0.079)	(0.116)	(0.080)
2	$m = [n/2]$	0.394	0.393	0.196	0.219	0.122	0.145	0.079	0.096
		(0.049)	(0.024)	(0.043)	(0.022)	(0.046)	(0.022)	(0.044)	(0.023)
5	$m = [n^{0.4}]$	0.317	0.359	0.281	0.326	0.221	0.305	0.182	0.261
		(0.378)	(0.218)	(0.352)	(0.223)	(0.370)	(0.220)	(0.363)	(0.212)
5	$m = [n^{0.5}]$	0.338	0.361	0.272	0.317	0.220	0.279	0.160	0.231
		(0.259)	(0.138)	(0.243)	(0.141)	(0.249)	(0.132)	(0.245)	(0.134)
5	$m = [n^{0.6}]$	0.348	0.357	0.269	0.299	0.208	0.248	0.146	0.199
		(0.174)	(0.094)	(0.177)	(0.093)	(0.175)	(0.089)	(0.178)	(0.087)
5	$m = [n/2]$	0.418	0.415	0.264	0.252	0.182	0.173	0.120	0.117
		(0.081)	(0.033)	(0.080)	(0.030)	(0.082)	(0.030)	(0.082)	(0.031)
10	$m = [n^{0.4}]$	0.307	0.349	0.266	0.340	0.209	0.318	0.190	0.268
		(0.449)	(0.255)	(0.451)	(0.248)	(0.458)	(0.255)	(0.443)	(0.250)
10	$m = [n^{0.5}]$	0.322	0.350	0.269	0.328	0.217	0.289	0.188	0.249
		(0.318)	(0.170)	(0.307)	(0.167)	(0.308)	(0.167)	(0.317)	(0.171)
10	$m = [n^{0.6}]$	0.340	0.355	0.273	0.314	0.225	0.270	0.171	0.216
		(0.246)	(0.118)	(0.230)	(0.117)	(0.234)	(0.114)	(0.244)	(0.114)
10	$m = [n/2]$	0.428	0.422	0.302	0.295	0.222	0.215	0.156	0.155
		(0.126)	(0.048)	(0.123)	(0.044)	(0.121)	(0.045)	(0.126)	(0.045)

The order of aggregation is k. Values below the estimates for d in parentheses are the standard errors for the estimates obtained by the simulations.

Table III: Truncated Whittle Estimates with MA Filtered Data— $d = 0.35$

		$ns = 0$		$ns = 2$		$ns = 5$		$ns = 10$	
k	$m \quad \backslash n$	1000	5000	1000	5000	1000	5000	1000	5000
1	$m = [n^{0.4}]$	0.336	0.362	0.279	0.327	0.218	0.273	0.166	0.216
		(0.168)	(0.137)	(0.176)	(0.146)	(0.173)	(0.147)	(0.167)	(0.141)
1	$m = [n^{0.5}]$	0.371	0.382	0.248	0.310	0.174	0.238	0.113	0.175
		(0.130)	(0.104)	(0.142)	(0.099)	(0.134)	(0.084)	(0.117)	(0.089)
1	$m = [n^{0.6}]$	0.380	0.391	0.214	0.266	0.130	0.200	0.074	0.128
		(0.104)	(0.087)	(0.093)	(0.042)	(0.098)	(0.049)	(0.087)	(0.073)
1	$m = [n/2]$	0.316	0.316	0.101	0.136	0.008	0.001	0.001	0.001
		(0.027)	(0.010)	(0.063)	(0.017)	(0.028)	(0.000)	(0.000)	(0.000)
10	$m = [n^{0.4}]$	0.349	0.364	0.290	0.329	0.229	0.274	0.178	0.219
		(0.163)	(0.136)	(0.176)	(0.147)	(0.175)	(0.147)	(0.168)	(0.142)
10	$m = [n^{0.5}]$	0.416	0.395	0.305	0.321	0.227	0.248	0.165	0.186
		(0.115)	(0.103)	(0.141)	(0.100)	(0.142)	(0.085)	(0.131)	(0.086)
10	$m = [n^{0.6}]$	0.497	0.459	0.450	0.317	0.371	0.245	0.307	0.189
		(0.019)	(0.069)	(0.084)	(0.059)	(0.106)	(0.043)	(0.099)	(0.050)
10	$m = [n/2]$	0.499	0.499	0.499	0.499	0.499	0.499	0.499	0.499
		(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)
20	$m = [n^{0.4}]$	0.380	0.370	0.324	0.334	0.266	0.281	0.214	0.224
		(0.153)	(0.134)	(0.171)	(0.146)	(0.178)	(0.148)	(0.173)	(0.143)
20	$m = [n^{0.5}]$	0.489	0.426	0.453	0.357	0.403	0.279	0.356	0.215
		(0.041)	(0.095)	(0.091)	(0.104)	(0.120)	(0.089)	(0.131)	(0.084)
20	$m = [n^{0.6}]$	0.499	0.499	0.499	0.497	0.499	0.463	0.499	0.371
		(0.000)	(0.000)	(0.000)	(0.016)	(0.000)	(0.067)	(0.000)	(0.080)
20	$m = [n/2]$	0.499	0.499	0.499	0.499	0.498	0.499	0.494	0.499
		(0.000)	(0.000)	(0.000)	(0.000)	(0.015)	(0.000)	(0.029)	(0.000)
30	$m = [n^{0.4}]$	0.422	0.381	0.380	0.345	0.329	0.292	0.283	0.234
		(0.125)	(0.130)	(0.155)	(0.145)	(0.172)	(0.148)	(0.172)	(0.145)
30	$m = [n^{0.5}]$	0.499	0.466	0.499	0.416	0.497	0.339	0.494	0.270
		(0.000)	(0.070)	(0.007)	(0.096)	(0.018)	(0.098)	(0.031)	(0.085)
30	$m = [n^{0.6}]$	0.499	0.499	0.499	0.499	0.499	0.499	0.499	0.499
		(0.000)	(0.000)	(0.000)	(0.000)	(0.006)	(0.000)	(0.000)	(0.000)
30	$m = [n/2]$	0.499	0.499	0.490	0.499	0.461	0.499	0.416	0.499
		(0.000)	(0.000)	(0.037)	(0.000)	(0.072)	(0.000)	(0.092)	(0.000)

The order q of the MA filter is $q = k - 1$, since $k$is the number of observations used for each filtered data point. Values below the estimates for d in parentheses are the standard errors for the estimates obtained by the simulations.

Table IV: Truncated Whittle Estimates with Aggregated Data — $d = 0.35$

		$ns = 0$		$ns = 2$		$ns = 5$		$ns = 10$	
k	$m \quad \backslash \, n$	1000	5000	1000	5000	1000	5000	1000	5000
1	$m = [n^{0.4}]$	0.336	0.362	0.279	0.327	0.218	0.273	0.166	0.216
		(0.168)	(0.137)	(0.176)	(0.146)	(0.173)	(0.147)	(0.167)	(0.141)
1	$m = [n^{0.5}]$	0.371	0.382	0.248	0.310	0.174	0.238	0.113	0.175
		(0.130)	(0.104)	(0.142)	(0.099)	(0.134)	(0.084)	(0.117	(0.089)
1	$m = [n^{0.6}]$	0.380	0.391	0.214	0.266	0.130	0.200	0.074	0.128
		(0.104)	(0.087)	(0.093)	(0.042)	(0.098)	(0.049)	(0.087	(0.073)
1	$m = [n/2]$	0.316	0.316	0.101	0.136	0.008	0.001	0.001	0.001
		(0.027)	(0.010)	(0.063)	(0.017)	(0.028)	(0.000)	(0.000)	(0.000)
2	$m = [n^{0.4}]$	0.328	0.352	0.280	0.314	0.229	0.278	0.183	0.231
		(0.179)	(0.152)	(0.190)	(0.162)	(0.184)	(0.161)	(0.179)	(0.159)
2	$m = [n^{0.5}]$	0.356	0.379	0.264	0.321	0.190	0.255	0.138	0.187
		(0.145)	(0.114)	(0.160)	(0.114)	(0.150)	(0.110)	(0.139)	(0.108)
2	$m = [n^{0.6}]$	0.373	0.386	0.239	0.285	0.155	0.217	0.099	0.151
		(0.113)	(0.092)	(0.122)	(0.068)	(0.115)	(0.059)	(0.104)	(0.077)
2	$m = [n/2]$	0.387	0.374	0.157	0.176	0.051	0.056	0.007	0.002
		(0.079)	(0.059)	(0.050)	(0.013)	(0.066)	(0.061)	(0.027)	(0.011)
5	$m = [n^{0.4}]$	0.308	0.330	0.282	0.313	0.239	0.286	0.206	0.241
		(0.191)	(0.168)	(0.201)	(0.185)	(0.194)	(0.180)	(0.195)	(0.174)
5	$m = [n^{0.5}]$	0.336	0.367	0.282	0.331	0.220	0.274	0.173	0.214
		(0.171)	(0.133)	(0.177)	(0.145)	(0.177)	(0.146)	(0.172)	(0.138)
5	$m = [n^{0.6}]$	0.362	0.383	0.265	0.315	0.190	0.244	0.134	0.179
		(0.137)	(0.106)	(0.157)	(0.106)	(0.145)	(0.094)	(0.134)	(0.095)
5	$m = [n/2]$	0.423	0.440	0.205	0.227	0.112	0.151	0.049	0.045
		(0.091)	(0.073)	(0.069)	(0.021)	(0.086)	(0.037)	(0.071)	(0.062)
10	$m = [n^{0.4}]$	0.282	0.316	0.276	0.309	0.246	0.289	0.209	0.240
		(0.206)	(0.181)	(0.207)	(0.188)	(0.205)	(0.189)	(0.203)	(0.184)
10	$m = [n^{0.5}]$	0.323	0.353	0.281	0.315	0.237	0.279	0.203	0.232
		(0.183)	(0.152)	(0.198)	(0.162)	(0.187)	(0.161)	(0.187)	(0.159)
10	$m = [n^{0.6}]$	0.345	0.378	0.282	0.331	0.219	0.264	0.166	0.199
		(0.165)	(0.121)	(0.176)	(0.126)	(0.175)	(0.123)	(0.163)	(0.114)
10	$m = [n/2]$	0.414	0.447	0.245	0.263	0.154	0.191	0.091	0.107
		(0.101)	(0.074)	(0.110)	(0.031)	(0.109)	(0.039)	(0.095)	(0.070)

The order of aggregation is k. Values below the estimates for d in parentheses are the standard errors for the estimates obtained by the simulations.

Table V: Variance Ratio Estimates

	$ns = 0$		$ns = 2$		$ns = 5$		$ns = 10$	
	1000	5000	1000	5000	1000	5000	1000	5000
$d = 0.00$	−0.075	−0.014	−0.006	−0.033	−0.047	−0.038	−0.104	−0.026
	(0.194)	(0.088)	(0.168)	(0.094)	(0.194)	(0.113)	(0.213)	(0.111)
$d = 0.20$	0.180	0.171	0.061	0.099	0.017	0.074	0.046	0.065
	(0.123)	(0.076)	(0.152)	(0.126)	(0.192)	(0.092)	(0.223)	(0.103)
$d = 0.30$	0.207	0.232	0.150	0.254	0.138	0.198	0.099	0.156
	(0.155)	(0.086)	(0.160)	(0.083)	(0.148)	(0.104)	(0.230)	(0.093)
$d = 0.45$	0.266	0.301	0.253	0.298	0.244	0.278	0.180	0.250
	(0.156)	(0.097)	(0.162)	(0.092)	(0.124)	(0.092)	(0.165)	(0.069)

Values below the estimates for d in parentheses are the standard errors for the estimates obtained by the simulations.