Dipartimento di Statistica
"Giuseppe Parenti"

WORKING PAPER 2002/03

Analytic Hessian Matrices and the Computation of FIGARCH Estimates

Marco J. Lombardi,
Giampiero M. Gallo

Università degli Studi
di Firenze

Econometrics

Dipartimento di Statistica "G. Parenti" – Viale Morgagni 59 – 50134 Firenze - www.ds.unifi.it

Analytic Hessian Matrices and the Computation of FIGARCH Estimates

Marco J. Lombardi Giampiero M. Gallo[*]

February 11, 2002

Abstract

Long memory in conditional variance is one of the empirical features of most financial time series. One class of models that was suggested to capture this behavior refers to the so-called Fractionally Integrated GARCH processes (Baillie, Bollerslev and Mikkelsen 1996) in which the ideas of fractional integration originally introduced by Granger (1980) and Hosking (1981) for processes for the mean are applied to a GARCH framework. In this paper we derive analytic expressions for the second-order derivatives of the log-likelihood function of FIGARCH processes with a view to the advantages that can be gained in computational speed and estimation accuracy. The comparison is computationally intensive given the typical sample size of the time series involved and the way the likelihood function is built. An illustration is provided on exchange rate and stock index data.

Keywords: Long Memory, Volatility Modelling, FIGARCH Processes.

JEL Classification : C63, C51, C22.

1 Introduction

The statistical analysis of financial time series provides evidence of various stylized facts, among which volatility clustering has received considerable attention. Many models have been added throughout the years to the Autoregressive Conditional Heteroskedasticity (ARCH) family, following the seminal paper by Engle (1982),

[*] Corresponding author. Email: `gallog@ds.unifi.it`. A preliminary version of this paper was presented at the conference S.Co. 2001 in Bressanone. We would like to thank Silvano Bordignon for his insightful discussion and Luisa Bisaglia and Giorgio Calzolari for providing useful comments. We also thank Tim Bollerslev for providing the data on the DEM/USD exchange rate used in Baillie, Bollerslev and Mikkelsen (1996).

which capture the short-run dependency of the conditional variances. Among the many empirical regularities which volatility models try to capture, one is that the decay exhibited by estimated conditional variances seems to be decreasing hyperbolically rather than exponentially. Another way of expressing this feature is that the process possesses long memory properties in the conditional variance. The advantage of modelling long memory applied to volatility processes is that the forecasting properties of the model so derived better suit the needs of medium-to-long term predictions which is crucial in derivative pricing models.

One class of models that was suggested in this direction is the so-called Fractionally Integrated GARCH (FIGARCH) process in which the ideas of fractional integration suggested by Granger (1980) and Hosking (1981) for processes for the mean are applied to a GARCH framework. Fractional integration serves the purpose of extending ARIMA processes to a more general class, ARFIMA, giving a continuum of possibilities between the polar cases of unit roots processes and of integrated processes of order 0. The order of integration in such a case becomes a real parameter d^* assuming values between 0 and 1 which can be estimated in the time or in the frequency domain.

In the FIGARCH case (Baillie et al. 1996), the extension captures the same idea, with some characterizations which we will summarize in what follows, and exploits the fact that estimated GARCH models often border with the case in which the conditional variance is an integrated process. Applications have shown that these long memory models fit well the data and research is still being undertaken as of which statistical properties these models possess.

In this paper we derive analytical expressions for the Hessian matrix for these models and exploit it for numerical optimization and to derive estimates of the variance – covariance matrix. Our goal is twofold: on the one hand we want to examine the properties of the Newton – Raphson algorithm when compared with other numerical algorithms; on the other we want to analyze the impact that a specific choice of an estimation method has for the variance – covariance matrix of the parameter estimators, as far as computation time and values obtained are concerned.

The comparison undertaken is computationally intensive given the typical large sample size of the financial time series involved and the way the likelihood function is built. Some savings in computation time can be achieved when analytic derivatives are employed, providing also a benchmark against which numerical approaches can be gauged. In this sense, our contribution extends the work by Fiorentini, Calzolari and Panattoni (1996) who have shown how different methods can lead to quite different results in the standard GARCH case.

The structure of the paper is as follows: in order to establish notation we summarize the main facts about FIGARCH in section 2. Two different simulation exercises (sections 3 and 4) show that our suggestion (references to the notation for alternative estimators of the variance–covariance matrices adopted here and the formulas for the appropriate derivatives to build the analytic Hessian are contained in an appendix) provides a considerable gain in computation time and gives some insights about the performance of some numerical estimators. Finally (section 5), we compare parameter estimator standard errors from financial time series. Concluding remarks follow.

2 Long memory and Fractionally Integrated GARCH

The concept of long memory[1] was introduced in time series analysis by Granger (1980) and Hosking (1981). It is well known (e.g., Hamilton (1994) – p. 50) that, for standard ARMA processes, the autocorrelation function decreases exponentially. By contrast, one way of arguing that a process possesses long memory suggested in the literature is that its autocorrelation function decreases "slowly". The most simple way to obtain long memory is to incorporate in the standard ARMA (p^*, q^*) formulation the *fractional difference operator* $(1-L)^{d^*}$, with $0 \leq d^* \leq 1$, where L denotes the lag operator. The resulting process is known as ARFIMA(p^*, d^*, q^*), that is AutoRegressive Fractionally Integrated Moving Average:

$$\eta(L)(1-L)^{d^*}(y_t - \mu) = \zeta(L)\epsilon_t.$$

with $\eta(L)$ and $\zeta(L)$ polynomials in the difference operator L of order p^*, respectively, q^*. It may be shown (Hosking 1981) that the autocorrelation function of ARFIMA processes decreases hyperbolically for $0 < d^* < 1$, that is sensibly slower than the standard ARMA case. The fractional difference operator may be expanded in a McLaurin series to produce an infinite polynomial in L:

$$(1-L)^{d^*} = \sum_{k=0}^{\infty} \frac{\Gamma(k-d^*)}{\Gamma(k+1)\Gamma(-d^*)} L^k = \sum_{k=0}^{\infty} \varphi_k(d^*) L^k, \tag{1}$$

which is more suitable for estimation purposes, since the φ_k coefficients have a known form ($\Gamma(\cdot)$ denotes the gamma function) which depends on d^*. It is obvious that, in order to obtain a manageable expression to be used for estimation, (1) must be truncated for some sufficiently high number M.

One of the most interesting results (Hosking 1981) is that ARFIMA processes with $d^* < 1/2$ are both long memory and covariance stationary processes; this result makes them suitable for the modelling of several economic variables. Several

[1] See, for a complete presentation, the monographic work by Beran (1994).

applications have been proposed up to now[2]; most of them are focused on the estimation of long memory in inflation, unemployment or interest rate time series.

A natural question to ask is whether long memory can characterize conditional volatility processes: (Ding, Granger and Engle 1993) suggest that in financial time series the speed of decay of squared residuals autocorrelations is slower than exponential. This stylized fact clashes with the standard GARCH models (Bollerslev (1986)) which do not reproduce such a feature. Bollerslev (1988) showed that the squared residuals autocorrelation function in a GARCH (1, 1) decreases exponentially, and, as such, the sum of the absolute values of autocorrelations converges; in order to exhibit long memory the same sum should diverge. Furthermore, Ding and Granger (1996) have shown that the autocorrelation function decreases exponentially also in the Integrated GARCH (Engle and Bollerslev 1986) case[3].

The Fractionally Integrated GARCH model was first introduced by Baillie et al. (1996) to account for long memory in conditional volatility. From the standard GARCH (*p*, *q*) formulation:

$$h_t = \omega + \alpha(L)\epsilon_{t-1}^2 + \beta(L)h_{t-1},$$

that may be written as an ARMA model for the squared residuals ϵ_t^2, namely,

$$[1-\alpha(L)-\beta(L)]\epsilon_t^2 \equiv \psi(L)\epsilon_t^2 = \omega + [1-\beta(L)]w_t, \tag{2}$$

where $w_t = \epsilon_t^2 - h_t$, the FIGARCH(p, d, q) model is simply obtained by inserting the operator $(1-L)^d$ in (2):

$$\psi(L)(1-L)^d\epsilon_t^2 = \omega + [1-\beta(L)]w_t, \tag{3}$$

where, again, use is made of the expansion (1) and of a suitable truncation at the estimation stage[4]. Another discrepancy with ARFIMA processes is that FIGARCH processes are not covariance-stationary, not even when when $0 < d < 1/2$. On

[2] Cf. the survey by Baillie (1996).

[3] This shows the fallacy of the parallel with I(1) processes for which the theoretical autocorrelations are equal to 1 at all lags. The issue is quite counterintuitive and involves being clear about whether persistence is a property of the correlogram or whether it impacts on the profile of the forecast function (cf. Ding and Granger (1996), for more details).

[4] Note that, as a result, $\psi(L)$ is not necessarily equal to $[1-\alpha(L)-\beta(L)]$: this has an impact, for example, on the interpretation of where a dynamic forecast of conditional variance with a FIGARCH converges. Also, in spite of the similarity, this formulation is not equivalent to specifying an ARFIMA model for the squared residuals ϵ_t^2: in this case, the fractional difference operator is not applied to ω. To overcome this notational incongruity, Chung (1999) proposed an alternative parameterization for the FIGARCH model (3): $\psi(L)(1-L)^d(\epsilon_t^2-\sigma^2) = [1-\beta(L)]w_t$ which we will not adopt here. This means that extension of the results for the ARFIMA to the FIGARCH case does not carry automatically.

the other hand, some results seemingly suggest that the process is indeed strongly stationary for $0 \leq d \leq 1$.[5]

Calculating conditional variances for FIGARCH models involves several computations and “long” loops and thus may be very time-consuming. Since the numerical evaluation of derivatives involves the computation of the log-likelihood function in several points, it is clear that calculating an Hessian matrices (or, for that matter, gradients) may require quite a lot of time, even on fast machines. Furthermore, numerical derivatives may be, by definition, quite inaccurate, especially if the log-likelihood function is flat over portions of its domain. On the other hand, there are clear advantages in considering analytic closed-form expressions for the second-order derivatives: the analytic Hessian matrix can be used at estimation stage to potentially improve upon precision and speed of convergence to the maximum likelihood estimates and can be used to derive expressions for the estimators' variance – covariance matrices (standard and robust, for example).

In the next section we will address the first set of issues by resorting to a comparison among optimization algorithms by simulation, leaving to section 4 the task of performing a similar comparison on variance – covariance matrix estimation procedures.

3 Alternative optimization algorithms

The most popular methods of optimization make use of exact or approximated Hessian matrices. The Newton – Raphson method, for example, uses the exact Hessian matrix. However, in the absence of analytic expressions for the second-order derivatives of the log-likelihood function, this method turns out to be too much time-consuming to be used in practice. Among the methods proposed in order to avoid the computation of the Hessian matrix, the suggestion by Berndt, Hall, Hall and Hausman (1974) is very popular. Given the gradient of the log-likelihood function for each observation, $\hat{\mathbf{g}}_t = \frac{\partial \ln \ell(\hat{\vartheta}, y_t)}{\partial \hat{\vartheta}}$, the Hessian matrix is approximated by:

$$\mathbf{P}(\vartheta) = \sum_{t=1}^{T} \hat{\mathbf{g}}_t \hat{\mathbf{g}}_t'; \tag{4}$$

Another common approach, introduced by Davidon – Fletcher – Powell and

[5] This result, presented in Baillie et al. (1996) is quite counterintuitive and still debated. A more formal proof of the stationarity is provided in Caporin (2001); an alternative approach to the concept of long memory for volatility processes that sheds some light on the question is presented in Davidson (2001).

improved by Broyden – Fletcher – Goldfarb – Shanno[6] is the so-called BFGS method, in which we approximate the inverse of the Hessian matrix with the following updating formula:

$$\Delta\mathbf{B}_{m+1}(\vartheta) = \frac{\mathbf{B}_m(\vartheta)\Delta\hat{\mathbf{g}}_{m+1}(\vartheta)\Delta\hat{\mathbf{g}}'_{m+1}(\vartheta)\mathbf{B}_m(\vartheta)}{\Delta\hat{\mathbf{g}}'_{m+1}(\vartheta)\mathbf{B}_m(\vartheta)\Delta\hat{\mathbf{g}}_{m+1}(\vartheta)} - \frac{\Delta\hat{\vartheta}_{m+1}\Delta\hat{\vartheta}'_{m+1}}{\Delta\hat{\mathbf{g}}'_{m+1}(\vartheta)\Delta\hat{\vartheta}_{m+1}}, \quad (5)$$

where $\Delta\hat{\mathbf{g}}_{m+1}(\vartheta) = [\hat{\mathbf{g}}_{m+1}(\vartheta) - \hat{\mathbf{g}}_m(\vartheta)]$ and $\Delta\hat{\vartheta}_{m+1} = (\hat{\vartheta}_{m+1} - \hat{\vartheta}_m)$. This formula is updated at each iteration m of the algorithm.

The first part of the simulation study we carried out consists in simulating 1000 FIGARCH processes:

$$\begin{aligned} y_t &= \mu + \epsilon_t \\ \epsilon_t | I_{t-1} &\sim N(0, h_t) \end{aligned} \quad (6)$$

where h_t is determined according to (3). For each replication the DGP generated 3000 observations with parameters $\mu = 0.0$, $\omega = 0.01$ $\psi = 0.2$, $\beta = 0.7$, $d = 0.5$; the variance of the noise term used in the simulation was set to 1. The processes were then estimated according to four different optimization methods: Newton-Raphson, BFGS with analytic gradient, BFGS with numeric gradient and BHHH. We have tried two different choices of starting values: in the first case, we use the true parameter values of the DGP, in the second one, we use $\mu = 0.005$, $\omega = 0.005$ $\psi = 0.3$, $\beta = 0.6$, $d = 0.3$. Results are presented in Tables 1 and 2.

Besides the obvious fact that the numeric-gradient BFGS takes quite a lot of time to converge, we may note that both BFGS algorithms take approximately twice the number of iterations required by the Newton – Raphson and the BHHH. Furthermore, whilst the Newton – Raphson seems to be quicker in terms of the number of iterations, at least for what concerns the “true” starting values, it is slower in terms of seconds required to converge than the BHHH. This result is also evident from Figure 1, where we have plotted the estimated distribution functions of the number of iterations to convergence.

What is really annoying is the fact that the Newton – Raphson algorithm seems to fall in weak convergence too often, especially when using alternative starting values[7]. The algorithm seems to perform correctly in the first two steps, but it often ends up wandering around the maximum. On the other hand, from Table 2, we may observe that working with the Newton – Raphson method pays quite well in terms of RMSE. This remains true also when we use the alternative starting values, provided we discard weak convergences.

[6] These are Quasi-Newton methods: see Thisted (1988) for complete references.

[7] We have also tried a smaller number of replications with completely “wrong” starting values and the results are even worse: on 100 replications, we got about 70% of weak convergences.

Table 1: Simulation results for 1000 replications. The headings "True values" and "Alternative values" denote the different choice of starting values. The column "NR" is for the Newton-Raphson method, "ABFGS" and "NBFGS" denote, respectively, the analytic-gradient and the numeric-gradient BFGS method. The first rows report the mean of the estimated coefficients across the replications, the row "Iter" reports the average number of iterations required for convergence, the row "Time" reports the average time (in seconds) to convergence on an AMD Athlon processor at 1.1 GHz and finally the row "Weak" reports the number of weak convergences experienced.

	NR	*ABFGS*	*NBFGS*	*BHHH*
		True values		
μ	0.00021	0.00021	0.00021	0.00022
ω	0.01118	0.01145	0.01146	0.01145
ψ	0.18741	0.18923	0.18939	0.18904
β	0.69834	0.69532	0.69524	0.69542
d	0.50969	0.50481	0.50452	0.50510
Iter	5.09	10.10	10.06	5.91
Time	1.96	2.46	30.26	1.78
Weak	51	4	15	1
		Alternative values		
μ	0.00350	0.00022	-0.00029	0.00022
ω	0.01847	0.01147	0.01150	0.01145
ψ	0.19657	0.18938	0.19005	0.18905
β	0.67201	0.69505	0.69191	0.69536
d	0.47478	0.50441	0.49987	0.50504
Iter	7.47	15.87	15.77	7.20
Time	2.85	3.34	45.21	2.10
Weak	121	7	39	1

Figure 1: Empirical cumulated distribution functions of the number of iterations and the required time (in seconds) for convergence for the optimization methods considered. The solid lines denote the Newton-Raphson method (NR), the dashed lines are for the analytic-gradient BFGS (ABFGS) and the dotted lines are for the BHHH.

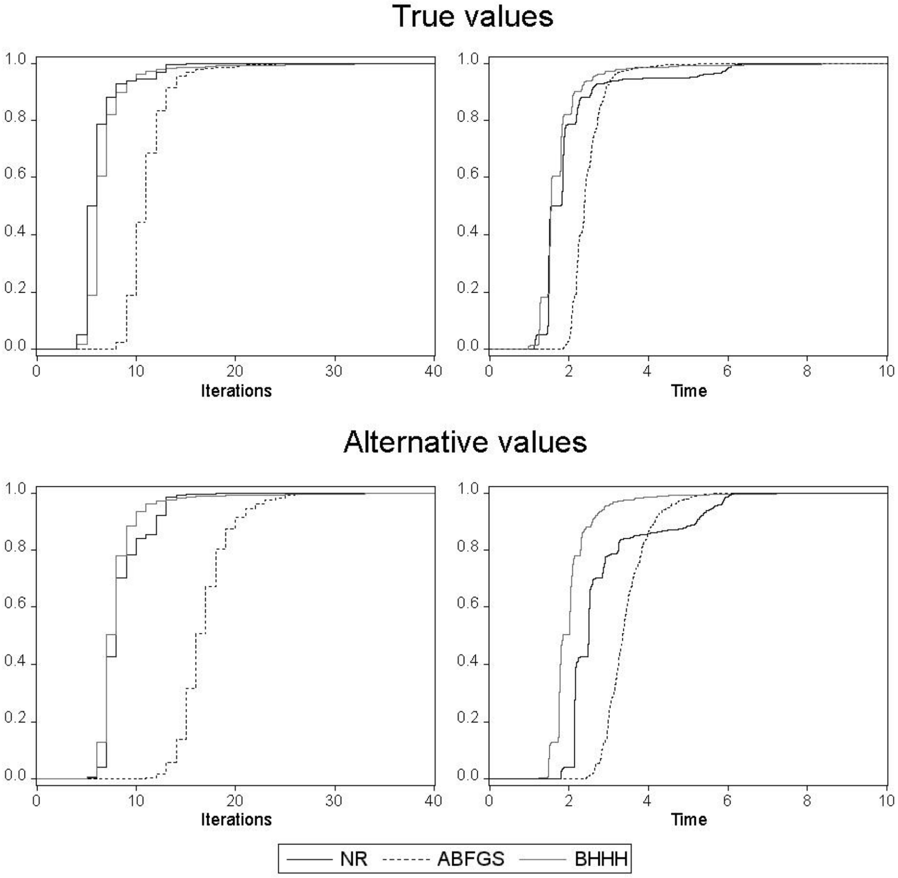


Table 2: Simulation results for 1000 replications. The rows report the root mean squared error of the estimated coefficients across the replications. The column “NR” is for the Newton – Raphson method, the column “NRS” denote the Newton – Raphson with weak convergences excluded, “ABFGS” and “NBFGS” denote, respectively, the analytic-gradient and the numeric-gradient BFGS method.

Parameters	**NR**	**NRS**	**ABFGS**	**NBFGS**	**BHHH**
			True values		
μ	0.01145	–	0.01162	0.01162	0.01162
ω	0.00574	–	0.00616	0.00616	0.00616
ψ	0.05031	–	0.05348	0.05307	0.05405
β	0.06454	–	0.06839	0.06840	0.06867
d	0.09352	–	0.10072	0.10029	0.10191
			Alternative values		
μ	0.01661	0.01164	0.01162	0.01241	0.01162
ω	0.02311	0.00625	0.00616	0.00624	0.00616
ψ	0.05803	0.05331	0.05307	0.05325	0.05402
β	0.09796	0.06933	0.06832	0.07137	0.06873
d	0.13461	0.10198	0.10024	0.10468	0.10193

This suggests, as previously noted by Fiorentini et al. (1996) for the standard GARCH case, the use of a mixed algorithm: we could perform the fisrt three steps with the BHHH to close on the maximum and then switch to the Newton – Raphson to obtain the best performance in terms of RMSE. The results of this approach are encouraging; we report them in Table 3.

Table 3: Simulation results for 1000 replications – Mixed algorithm – Alternative starting values. The column "Estimates" reports the mean of the estimated coefficients across the replications, while the column "RMSE" reports the square root of the mean squared error. The "MIX" columns reports the results for the mixed algorithm presented earlier, while the columns "NR" and "BHHH" report, respectively, the results previously obtained for the Newton – Raphson and the BHHH algorithms.

	MIX		**NR**		**BHHH**	
	Estimates	*RMSE*	*Estimates*	*RMSE*	*Estimates*	*RMSE*
μ	0.00035	0.01187	0.00350	0.01661	0.00022	0.01162
ω	0.01166	0.00640	0.01847	0.02311	0.01145	0.00616
ψ	0.19022	0.05304	0.19657	0.05803	0.18905	0.05402
β	0.69402	0.06812	0.67201	0.09796	0.69536	0.06873
d	0.50253	0.10021	0.47478	0.13461	0.50504	0.10193
Iter	5.92		7.47		7.20	
Time	1.92		2.85		2.10	
Weak	14		121		1	

We may note that the performance of the mixed algorithm is clearly superior, both in terms of time and iterations and in terms of RMSE. Examining more closely the RMSE column, we may observe that the mixed algorithm performs better especially for what concerns the parameters of the variance equation (the main focus of the experiment).

4 Alternative estimators of variance – covariance matrices

Let us now move to the estimation of the parameters variances. In order to avoid the calculation of the expected value in the information matrix one may use the Hessian matrix evaluated at the maximum likelihood estimated parameter vector

$\hat{\vartheta}$, namely

$$\mathbf{H}(\hat{\vartheta}) = -\frac{\partial^2 \ln \ell(\hat{\vartheta})}{\partial\hat{\vartheta}\partial\hat{\vartheta}'}.$$

The approach suggested by Berndt et al. (1974) applies also to this inferential framework: the information matrix may be estimated by inverting (4).

The last estimator we consider (White 1982) is known as *Quasi-Maximum Likelihood* estimator (QML):

$$\mathbf{Q}(\vartheta) = \mathbf{H}^{-1}(\vartheta)\mathbf{P}(\vartheta)\mathbf{H}^{-1}(\vartheta),$$

which is robust over a variety of model misspecification errors.

If the log-likelihood function is maximized using Hessian-based algorithms, the information matrix can be obtained as a by-product of the optimization procedure. This is surely the case when the optimization algorithm uses the exact Hessian (Newton – Raphson method) or an approximation that improves as one draws nearer to the optimum (BHHH method). On the other hand, as pointed out in Thisted (1988), if quasi-Newton methods[8] such as Davidon – Fletcher – Powell (DFP) or Broyden – Fletcher – Goldfarb – Shanno (BFGS), are employed, the estimated Hessian matrix is not guaranteed to converge to its exact value when approaching the optimum. Using those methods to obtain the estimated information matrix may thus be misleading.

The second simulation study we carried out is intended to assess whether different methods for the estimation of the information matrix may lead to discrepancies in the standard errors of the parameters. We used the same simulation design as in the previous section; each replication was estimated using the mixed BHHH/NR method with an analytic gradient, and then the standard errors of the parameters were calculated with the various methods mentioned earlier. In Figure 2, we report the empirical distributions of the estimated parameters across replications.
The average computation time required by the analytic method was 0.18 seconds, as opposed to a time of 15.82 seconds required by the computation of the numeric Hessian matrices, which shows obvious computational advantages.

The simulation results, presented in Table 4, report the mean and standard deviation of the estimated parameter distribution across replications, and the standard errors associated with the estimated parameters, according to the various methods.

First, note that the standard errors for μ are in line with one another, hinting at the possibility that the exact information matrix is indeed block diagonal also in the FIGARCH case (as in Bollerslev, 1986, for the standard GARCH case). For what concerns the parameters of the conditional variance equation, the methods considered seem to perform slightly differently. Note that the results obtained from

[8] See Thisted (1988) for complete references.

Figure 2: Kernel densities for the estimates of the parameters. Bandwidths are reported in each graph as b; arrows point at the "true" parameter values of the DGP.

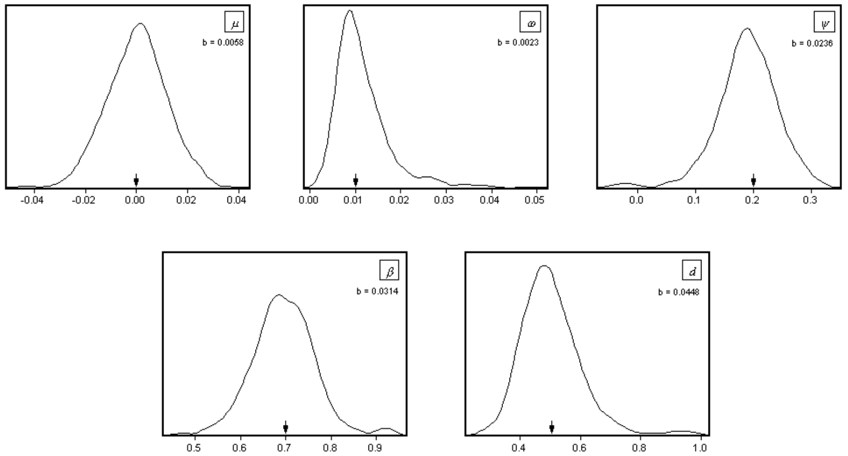


Table 4: Simulation results for 1000 replications. In the section "Parameters Distribution" we report the mean and the standard deviation of the distribution of estimated parameters across replications. Under the heading "Standard Errors", we report the square root of the average value of the parameter variances computed according to each method.

	μ	ω	ψ	β	d
Parameters Distribution					
Mean	0.000222	0.011450	0.189043	0.695417	0.505097
Standard deviation	0.011619	0.005916	0.052924	0.068513	0.101779
Standard Errors					
Analytic Hessian	0.012042	0.005477	0.046925	0.060366	0.081240
Numeric Hessian	0.012083	0.005568	0.046271	0.060819	0.080517
Outer product	0.012083	0.005477	0.045673	0.058983	0.077240
QML	0.012083	0.005657	0.053451	0.066030	0.095373

the analytic Hessian and its numeric counterpart are very close, but for the latter we have to observe that in several cases the estimation ended up with no results[9]. In what concerns the Outer-product estimator, the estimated variances seem to be systematically smaller than those obtain by Hessian matrices (both analytic and numeric). For the simulation design adopted here, this would lead to a systematic underestimation of the standard error by numeric Hessian or the Outer Product, possibly leading to a tendency to over-rejection in a testing framework. As far as QML is concerned, we have to note that this method performs much worse than the others, leading the corresponding test statistic to a tendency to under-reject for a given level of significance, at least for the simulation design adopted. To check whether this discrepancy is caused by the small sample size used, we have replicated the experiment using series of 10,000 and 30,000 instead of 3,000 observations; the results are reported in Table 5 (further simulations based on 50,000 observations led to similar results).

The results obtained seem to confirm our guess: the order of the discrepancy gets definitely smaller. For example, if we consider the d parameter, the discrepancy between the analytic Hessian and the QML is 0.014133 (that is, the robust standard errors are 17.39% higher) for the series of 3000 observations, but drops to 0.001047 (2.49% higher) for 10000 observations and to 0.000109 (0.59% higher) for 30000 observations. This result could be caused by the fact that, for estimation purposes, we use, as suggested by Baillie et al. (1996), a set of 1000 pre-sample values equal to the unconditional variance of the process; the presence of those "fake" observations could lead to discrepancies in the results. One possible way to test this conjecture is to use a different estimation approach – for example using every available observation, as suggested in Teyssière (1996), or "discarding" observations from the original series, or the one proposed in Chung (1999). A related issue is addressed by Teyssière (1996), who presents simulation results suggesting that the magnitude of the truncation order M may affect the estimates of the ω parameter.

It may be also instructive to examine the distribution of the estimated variances across methods as done in Figure 3, where we have plotted the sample distribution of the variances of the parameters calculated according to the following methods: analytic Hessian, numeric Hessian and Outer Product[10]. There are no clear advantages of one method over the other (apart from the computation time involved): the variance estimators have very similar distributions with the exception of the variance of $\hat{\psi}$, for which the Hessians are slightly more precise, and of $\hat{\beta}$, for which

[9] In 19 cases the differentiation algorithm did not converge, while in 18 cases the Hessian matrix reported was singular. Over the 1000 replication the ratio of failed numeric Hessian is thus 3.7%.

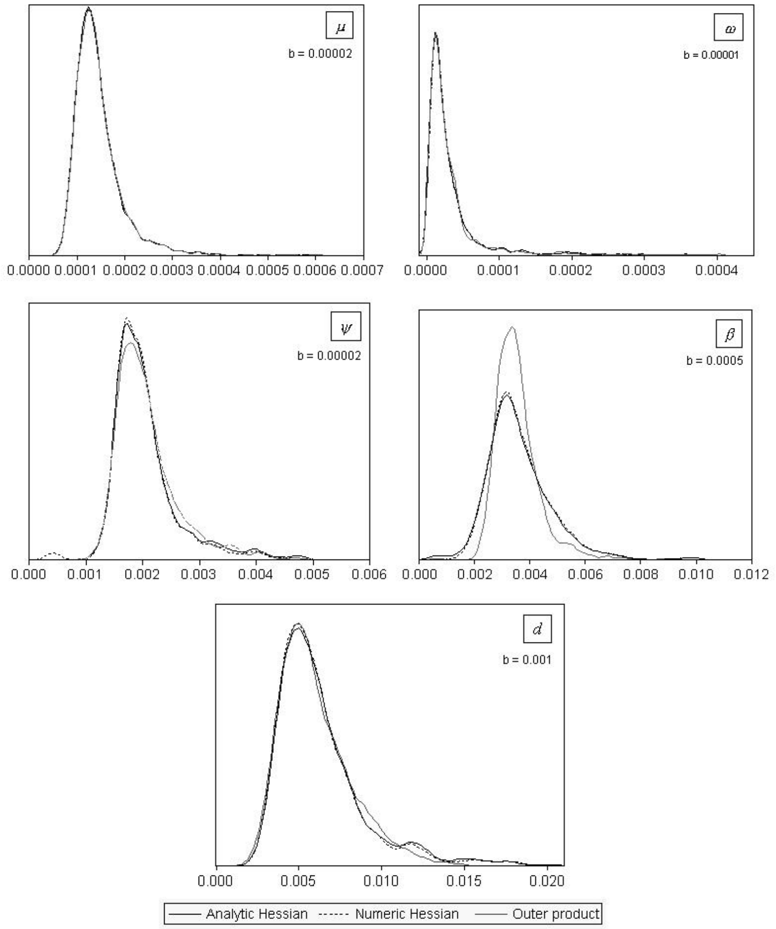
[10] We have not included QML in the graphical comparison given the different order of magnitude of estimated variances which would have made the results for the other three methods look the same.

Table 5: Simulation results for 1000 replications – series of 10,000 and 30,000 observations. In the section "Parameters" we report the mean and the standard deviation of the distribution of estimated parameters across replications. Under the heading "Standard Errors", we report the square root of the average value of the parameter variances computed according to each method.

	μ	ω	ψ	β	d
			10,000 observations		
Parameters					
Mean	-0.000131	0.010166	0.197914	0.700269	0.502970
Std. deviation	0.006325	0.002449	0.024839	0.032680	0.045967
Standard Errors					
Analytic Hessian	0.006633	0.002449	0.023302	0.031289	0.041976
Outer product	0.006633	0.002449	0.023238	0.030952	0.041328
QML	0.006633	0.002449	0.023473	0.031984	0.043023
			30,000 observations		
Parameters					
Mean	0.000033	0.010002	0.199456	0.700410	0.500950
Std. deviation	0.002828	0.001000	0.010050	0.013601	0.018166
Standard Errors					
Analytic Hessian	0.003000	0.001000	0.010099	0.013675	0.018221
Outer product	0.003000	0.001000	0.010099	0.013638	0.018166
QML	0.003000	0.001000	0.010099	0.013711	0.018330

BHHH has a less dispersed distribution.

Figure 3: Kernel densities for the estimated variances of the parameters. Bandwidths are reported in each graph as B. The solid lines denote the analytic Hessian, the dashed lines the numeric Hessian and the dotted lines are for the Outer Product.



5 An Application on Financial Series

With these results in hand, we now turn to illustrating how the procedure performs on real world data. We chose to replicate the results obtained by Baillie et al. (1996) using their same data on the Deutsche Mark – US dollar exchange rate and estimating[11] a FIGARCH (1, *d*, 1) model on the daily returns of the DEM/USD spot exchange rate from March 13, 1979 to December 30, 1992, for a total of 3454 observations, using the Newton-Raphson optimization method. The results are presented in Table 6, in order to show the wide difference between "standard" and robust standard errors.

Table 6: Estimation of a FIGARCH (1, *d*, 1) model for the daily returns of the spot exchange rate of the Deutsche Mark versus the U.S. dollar, from March 13, 1979 to December 30, 1992 (3454 observations). The "ML" columns report the maximum likelihood estimated standard errors and p-values, while the "QML" columns report the same quantities computed by resorting to quasi-maximum likelihood.

		ML		**QML**	
	Estimate	*Standard Error*	*P-Value*	*Standard Error*	*P-Value*
μ	-0.004068	0.011105	0.714155	0.011415	0.721587
ω	0.018158	0.003994	0.000001	0.005633	0.001279
ψ	0.091543	0.048470	0.059022	0.076918	0.234073
β	0.679127	0.069966	0.000000	0.104912	0.000000
d	0.658186	0.102210	0.000000	0.163107	0.000056

Let us see another example, where we consider a longer series from the Standard and Poor 500 index from January 3, 1950 to June 29, 2001 (13045 observations). The estimation results with a comparison between analytic and numeric Hessians are presented in Table 7. We may observe that the relative difference between the two standard errors does not change appreciably leaving open the question whether the difference is due to poorer performance of QML or considerable impact of misspecified errors.

6 Concluding remarks

In this paper we have derived closed-form formulas for the second-order analytic derivatives of the log-likelihood function for the FIGARCH (*p*, *d*, *q*) model and we have employed them in a comparison exercise among the main estimation methods. Somewhat surprisingly, the Newton Raphson algorithm performs well only when used in conjunction with the BHHH, since for the FIGARCH model it shows a tendency to end up in weak convergence. Overall, the savings in computational time that can be gained from analytical expressions are quite remarkable. As far as

[11] The estimation has been carried out with a C routine available, in executable version, at the URL *http://www.ds.unifi.it/˜mjl/bin/Figarch.exe*.

Table 7: Estimation of a FIGARCH (1, *d*, 1) model for the daily returns of the Standard & Poor 500 index, from January 3, 1950 to June 29, 2001 (13045 observations). The "ML" columns report the maximum likelihood estimated standard errors and p-values, while the "QML" columns report the same quantities computed by resorting to quasi-maximum likelihood.

		ML		**QML**	
	Estimate	*Standard Error*	*P-Value*	*Standard Error*	*P-Value*
μ	0.050458	0.006053	0.000000	0.007211	0.000000
ω	0.018460	0.002767	0.000000	0.005375	0.000000
ψ	0.366847	0.025776	0.000000	0.044326	0.000000
β	0.669021	0.029659	0.000000	0.052520	0.000000
d	0.444385	0.033014	0.000000	0.063681	0.000000

the variance – covariance matrix estimation is concerned, our results show that the quality of the estimates is improved upon when the analytical Hessian is employed; robust standard errors appear to be systematically higher than their standard counterpart. When normal errors are employed, the simulation results show that the difference between standard errors decreases only for very large sample sizes.

References

Baillie, R. (1996). Long memory processes and fractional integration in econometrics, *Journal of Econometrics* **73**, 5–59.

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

Beran, J. (1994). *Statistics for long memory processes*, Chapman and Hall, New York.

Berndt, E., Hall, B., Hall, R. and Hausman, J. (1974). Modelling the persistence of conditional variances, *Annals of Economic and Social Measurement* **3**, 653–665.

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

Bollerslev, T. (1988). On the correlation structure for the GARCH process, *Journal of Time Series Analysis* **9**, 121–131.

Caporin, M. (2001). Estimation and identification of FIGARCH, Università “Ca’ Foscari” di Venezia, Venezia, I.

Chung, C.-F. (1999). Estimating the fractionally integrated GARCH model, National Taiwan University, Taipei, TW.

Davidson, J. (2001). Moment and memory properties of linear conditional heteroscedasticity models, Cardiff University, Cardiff, UK.

Ding, Z.-X. and Granger, C. (1996). Modelling volatility persistence of speculative returns: a new approach, *Journal of Econometrics* **73**, 185–215.

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

Engle, R. (1982). Autoregressive conditional heteroskedasticity with estimates of the variance of U.K. inflation, *Econometrica* **50**, 987–1008.

Engle, R. and Bollerslev, T. (1986). Modelling the persistence of conditional variances, *Econometric Reviews* **5**, 1–50.

Fiorentini, G., Calzolari, G. and Panattoni, L. (1996). Analytic derivatives and the computation of GARCH estimates, *Journal of Applied Econometrics* **11**, 399–417.

Granger, C. (1980). Long-memory relationships and the aggregation of dynamic models, *Journal of Econometrics* **14**, 228–238.

Hamilton, J. (1994). *Time series analysis*, Princeton University Press, Princeton.

Hosking, J. (1981). Fractional differencing, *Biometrika* **68**, 165–176.

Lombardi, M. (2001). *Forecasting volatility of financial assets: Long memory processes and fractionally integrated GARCH*, Master’s thesis, Università di Firenze. In italian.

Teyssière, G. (1996). Double long memory financial time series, *QMW working paper 348*, Univeristy of London.

Thisted, R. (1988). *Elements of Statistical Computing*, Chapman and Hall, New York.

White, H. (1982). Maximum likelihood estimation of misspecified models, *Econometrica* **50**, 1–25.

Appendix. Analytic Derivatives for FIGARCH processes

In a likelihood-based estimation framework, once a suitable estimation procedure is adopted to derive the value of the estimates, inference needs require the appropriate computation of the estimators' variance – covariance matrix (that is the inverse of the information matrix $\mathbf{I}(\vartheta) = \mathrm{E}\left[-\partial^2 \ln \ell(\vartheta)/\partial\vartheta\partial\vartheta'\right]$.

In a GARCH-type framework, let us consider a model with a simple constant term μ in the mean equation and a generic conditional variance h_t. The relevant parameter vector $\vartheta = (\mu, \vartheta_v')'$, where ϑ_v is a vector containing the parameters in the variance equation. The log-likelihood function is defined as:

$$\ln \ell(\vartheta) = -\frac{T}{2}\ln(2\pi) - \sum_{t=1}^{T}\frac{\ln h_t}{2} - \sum_{t=1}^{T}\frac{\epsilon_t}{2h_t}. \tag{7}$$

Differentiating with respect to parameters ϑ yields:

$$\frac{\partial \ln \ell(\vartheta)}{\partial\vartheta} = \frac{1}{2}\sum_{t=1}^{T}\frac{1}{h_t}\left[\frac{\partial h_t}{\partial\vartheta}\frac{\epsilon_t^2 - h_t}{h_t} - 2\frac{\partial\epsilon_t}{\partial\vartheta}\epsilon_t\right],$$

that is

$$\frac{\partial \ln \ell(\vartheta)}{\partial\mu} = \frac{1}{2}\sum_{t=1}^{T}\frac{1}{h_t}\left[2\epsilon_t + \frac{\partial h_t}{\partial\mu}\frac{\epsilon_t^2 - h_t}{h_t}\right] \tag{8}$$

and

$$\frac{\partial \ln \ell(\vartheta)}{\partial\vartheta_v} = \frac{1}{2}\sum_{t=1}^{T}\frac{\partial h_t}{\partial\vartheta_v}\frac{\epsilon_t^2 - h_t}{h_t^2}. \tag{9}$$

Analytic formulas for $\frac{\partial h_t}{\partial\vartheta_v}$ for the FIGARCH model have been derived by Chung (1999), in its alternative parameterization already mentioned. Exploiting some of these results, Lombardi (2001) found the analytic expressions for the standard FIGARCH (*p*, *d*, *q*) case:

$$\frac{\partial h_t}{\partial\mu} = 2\left[\frac{\psi(L)(1-L)^d}{1-\beta(L)} - 1\right]\epsilon_t,$$

$$\frac{\partial h_t}{\partial\omega} = \frac{1}{1-\beta(L)},$$

$$\frac{\partial h_t}{\partial\psi_i} = \frac{1}{1-\beta(L)}\sum_{k=0}^{M}\varphi_k\epsilon_{t-k-i}^2, \quad i = 1, 2, ..., p,$$

$$\frac{\partial h_t}{\partial \beta_j} = \frac{h_{t-j} - \epsilon_{t-j}^2}{1-\beta(L)}, \quad j = 1, 2, ..., q,$$

$$\frac{\partial h_t}{\partial d} = -\frac{\psi(L)}{1-\beta(L)} \sum_{k=1}^{M} \frac{\mathrm{d}\varphi_k}{\mathrm{d}d} \epsilon_{t-k}^2;$$

As far as the gamma function in expression 1 is concerned, let us first note that its exact derivative is given by the Digamma function (sometimes indicated as $\psi(\cdot)$ function), defined as:

$$\psi(d) = \frac{\mathrm{d}\Gamma(d)}{\mathrm{d}d} = \sum_{k=1}^{\infty} \left[\frac{1}{k} - \frac{1}{k+d-1}\right] - \gamma,$$

where γ denotes the Euler-Mascheroni constant. In order to obtain the derivative of φ_k with respect to d, we need to compute the derivative of the ratio of two different gamma functions

$$\frac{\mathrm{d}\varphi_k}{\mathrm{d}d} = \frac{1}{\Gamma(k+1)} \frac{\partial}{\partial d} \left[\frac{\Gamma(k-d)}{\Gamma(d)}\right];$$

using digamma functions can thus be quite unpractical and time-consuming. A more manageable expression, proposed in Chung (1999), is the approximation:

$$\frac{\mathrm{d}\varphi_k}{\mathrm{d}d} = -\varphi_k \sum_{j=1}^{k} \frac{1}{k-j-d}.$$

Second-order analytic derivatives for GARCH models have been introduced by Fiorentini et al. (1996). Referring to the model with the only μ in the mean equation we have:

$$\begin{aligned}\frac{\partial^2 \ln \ell(\vartheta)}{\partial \mu^2} &= \sum_{t=1}^{T} \left[\frac{1}{2} \frac{\partial^2 h_t}{\partial \mu^2} \frac{\epsilon_t^2 - h_t}{h_t^2} - \frac{1}{h_t} - 2\frac{\partial h_t}{\partial \mu} \frac{\epsilon_t}{h_t^2}\right] + \\ &-\frac{1}{2} \sum_{t=1}^{T} \left[\frac{\partial h_t}{\partial \mu} \frac{\partial h_t}{\partial \mu} \frac{\epsilon_t^2 - h_t}{h_t^3} + \frac{\partial h_t}{\partial \mu} \frac{\partial h_t}{\partial \mu} \frac{\epsilon_t^2}{h_t^3}\right], \qquad (10)\end{aligned}$$

$$\frac{\partial^2 \ln \ell(\vartheta)}{\partial \vartheta_v \partial \vartheta_v'} = \frac{1}{2} \sum_{t=1}^{T} \left[\frac{\partial^2 h_t}{\partial \vartheta_v \partial \vartheta_v'} \frac{\epsilon_t^2 - h_t}{h_t^2} - \frac{\partial h_t}{\partial \vartheta_v} \frac{\partial h_t}{\partial \vartheta_v'} \frac{\epsilon_t^2 - h_t}{h_t^3} - \frac{\partial h_t}{\partial \vartheta_v} \frac{\partial h_t}{\partial \vartheta_v'} \frac{\epsilon_t^2}{h_t^3}\right], \qquad (11)$$

$$\begin{aligned}\frac{\partial^2 \ln \ell(\vartheta)}{\partial \mu \partial \vartheta_v'} &= -\sum_{t=1}^{T} \left[\frac{\partial h_t}{\partial \vartheta_v'} \frac{\epsilon_t}{h_t^2} + \frac{1}{2} \frac{\partial h_t}{\partial \mu} \frac{\partial h_t}{\partial \vartheta_v'} \frac{\epsilon_t^2 - h_t}{h_t^3}\right] + \\ &+\frac{1}{2} \sum_{t=1}^{T} \left[\frac{\partial^2 h_t}{\partial \mu \partial \vartheta_v'} \frac{\epsilon_t^2 - h_t}{h_t^2} - \frac{\partial h_t}{\partial \mu} \frac{\partial h_t}{\partial \vartheta_v'} \frac{\epsilon_t^2}{h_t^3}\right]. \qquad (12)\end{aligned}$$

Extending these results to the FIGARCH case requires the analytic expressions for the $\frac{\partial^2 h_t}{\partial\vartheta\partial\vartheta'}$'s. Let us consider a general FIGARCH (*p*, *d*, *q*) model with the only constant term μ in the mean equation; we may show that:

$$\frac{\mathrm{d}^2\varphi_k}{\mathrm{d}d^2} = \varphi_k \left[\sum_{j=1}^{k} \frac{1}{k-j-d}\right]^2 - \varphi_k \sum_{j=1}^{k} \frac{1}{(k-j-d)^2};\,[12]$$

$$\frac{\partial^2 h_t}{\partial\mu^2} = 2 - 2\frac{\psi(1)}{1-\beta(L)}\sum_{k=0}^{M}\varphi_k,$$

$$\frac{\partial^2 h_t}{\partial\mu\partial\psi_i} = -\frac{2}{1-\beta(L)}\sum_{k=0}^{M}\varphi_k\epsilon_{t-k-i}, \quad i = 1, 2, ..., p,$$

$$\frac{\partial^2 h_t}{\partial\mu\partial\beta_j} = \frac{2\epsilon_{t-j} + \frac{\partial h_{t-j}}{\partial\mu}}{1-\beta(L)}, \quad j = 1, 2, ..., q,$$

$$\frac{\partial^2 h_t}{\partial\mu\partial d} = 2\frac{\psi(L)}{1-\beta(L)}\sum_{k=1}^{M}\frac{\mathrm{d}\varphi_k}{\mathrm{d}d}\epsilon_{t-k},$$

$$\frac{\partial^2 h_t}{\partial\mu\partial\omega} = \frac{\partial^2 h_t}{\partial\omega^2} = \frac{\partial^2 h_t}{\partial\omega\partial\psi} = \frac{\partial^2 h_t}{\partial\omega\partial d} = 0,$$

$$\frac{\partial^2 h_t}{\partial\psi_{i_1}\partial\psi_{i_2}} = 0, \quad i_1 = 1, 2, ..., p, \quad i_2 = 1, 2, ..., p,$$

$$\frac{\partial^2 h_t}{\partial\omega\partial\beta_j} = \frac{\partial h_{t-j}}{\partial\omega} + \beta_j\frac{\partial^2 h_{t-j}}{\partial\omega\partial\beta_j}, \quad j = 1, 2, ..., q,$$

$$\frac{\partial^2 h_t}{\partial\psi_i\partial\beta_j} = \frac{\partial h_{t-j}}{\partial\psi_i} + \beta_j\frac{\partial^2 h_{t-j}}{\partial\psi_i\partial\beta_j}, \quad i = 1, 2, ..., p, \quad j = 1, 2, ..., q,$$

$$\frac{\partial^2 h_t}{\partial\psi_i\partial d} = \frac{L}{1-\beta(L)}\sum_{k=1}^{M}\varphi_k\epsilon^2_{t-k}, \quad i_1 = 1, 2, ..., p,$$

$$\frac{\partial^2 h_t}{\partial\beta_{j_1}\partial\beta_{j_2}} = 2\frac{h_{t-j_1-j_2} - \epsilon^2_{t-j_1-j_2}}{(1-\beta(L))^2}, \quad j_1 = 1, 2, ..., q, \quad j_2 = 1, 2, ..., q,$$

$$\frac{\partial^2 h_t}{\partial\beta_j\partial d} = \frac{\partial h_{t-j}}{\partial d} + \beta\frac{\partial^2 h_{t-j}}{\partial d\partial\beta_j}, \quad j = 1, 2, ..., q,$$

$$\frac{\partial^2 h_t}{\partial d^2} = \frac{\psi(L)}{1-\beta(L)}\sum_{k=1}^{M}\frac{\mathrm{d}^2\varphi_k}{\mathrm{d}d^2}\epsilon^2_{t-k}.$$

[12] Once again, we are considering a simple approximation that allows us to avoid the computation of the polygamma function.