Università degli Studi di Padova
Dipartimento di Scienze Statistiche

Dottorato in
Scienze Statistiche
Ciclo XIX

Bootstrap and approximation methods for long memory processes

Nedda Cecchinato

Direttore: Prof. S. Bordignon

Supervisore: Prof. S. Bordignon

Co-supervisori: Prof. L. Bisaglia, Prof. R. Wolff

31 Gennaio 2008

Eureka!
(Archimedes)
Chi non ha uno scopo non prova quasi mai diletto in nessuna operazione.
(Leopardi, Zibaldone, 268)

I would like to thank my mum that has always supported me.
I thank Professor Rodney Wolff who taught me so much: "Writing a thesis is like telling a story".
I thank Professor Silvano Bordignon and Professor Luisa Bisaglia for letting me going to Australia and for their precious help.
I thank Professor Dominique Guégan for the interesting discussion on the topic of Chapter 5.
I thank my friends in Brisbane: Phoebe, Irasema, Alison, Ximena, Pauline, Erika, Jason, Stefan and Ronny. A special mention to Bernard, Beth and Leo. I can not forget my Italian friends: Elena, Elisabetta, Erica, Sofia, Marco, Aldo and Teddy.
I thank everybody that let me feeling a special person.

Contents

List of Figures

List of Tables

Glossary

AR: Autoregressive process.
ARCH: Autoregressive Conditionally Heteroskedastic process.
ARFIMA: Autoregressive Fractionally Integrated Moving Average process.
ARFISMA: Autoregressive Fractionally Integrated Seasonal Moving Average process.
ARIMA: Autoregressive Integrated Moving Average process.
ARMA: Autoregressive Moving Average process.
FI: Fractionally Integrated process.
GARMA: Generalized or Gegenbauer ARMA process.
GARCH: Generalized Autoregressive Conditionally Heteroskedastic process.
GPH: Geweke Porter-Hudak semi-parametric estimator of the memory parameter d.
local Whittle: semi-parametric estimator of the memory parameter d.
MA: Moving Average process.
SCALM: Seasonal/Cyclical Asymmetric Long Memory.
Whittle: maximum likelihood estimator of ARFIMA processes.

Introduction

Italiano

I processi frazionari integrati ARFIMA(p, d, q), introdotti contemporaneamente nei lavori di Granger and Joyeux (1980) e di Hosking (1981), offrono uno strumento utile e flessibile per modellare la struttura di dipendenza di secondo ordine (funzioni di autocovarianza e di autocorrelazione) di una serie storica osservata. La letteratura è ricca di articoli su stima e identificazione del data generating process (d'ora in poi abbreviato con dgp): sulla stima parametrica del parametro di memoria lunga d e dei parametri della parte autoregressiva e a media mobile, ricordiamo i lavori di Yajima (1985), Fox and Taqqu (1986) e Dahlhaus (1988, 1989); sulla stima semi-parametrica del parametro di memoria lunga d sono da ricordare gli articoli di Hurst (1951), Geweke and Porter-Hudak (1983), Higuchi (1988), Robinson (1995a) e Hurvich et al. (1998). Nonostante la letteratura sia ricca di articoli, è ancora difficile definire quale sia il metodo migliore per l'identificazione e la stima dei parametri del dgp; a seconda della situazione ogni soluzione offre vantaggi e svantaggi. Gli stimatori parametrici (di Whittle e di massima verosimiglianza) sono asintoticamente Normali e sono più efficienti degli stimatori semi-parametrici ma, in caso di non corretta specificazione del modello, i risultati possono essere disastrosi fornendo stime altamente distorte. D'altra parte gli stimatori semi-parametrici (local Whittle, GPH, metodo dei ranghi riscalati solo per citarne alcuni) permettono di stimare il parametro di memoria lunga d indipendentemente dalla parte ARMA con lo svantaggio di un tasso di convergenza inferiore, $o(n^{-1/2})$ invece di $o(n^{-1})$ nel caso parametrico. Agiakloglou et al. (1993) hanno mostrato che la presenza di memoria corta introduce distorsione nello stimatore GPH (Geweke and Porter-Hudak, 1983), specialmente quando il parametro autoregressivo ϕ è vicino alla soglia di non-stazionarietà.

Una generalizzazione dei processi ARFIMA sono i processi di Gegenbauer, introdotti da Hosking (1981) e poi studiati da molti autori come Woodward et al. (1989), Woodward et al. (1998), Giraitis and Leipus (1995), Smallwood and Beaumont (2004), Sadek and Khotanzad (2004) e Caporale and Gil-Alana (2006). Stimatori parametrici e semi-parametrici sono stati sviluppati in letteratura. Uno dei maggiori problemi è la massimizzazione in spazio multidimensionale della funzione di verosimiglianza perché non esiste una soluzione in forma chiusa e le procedure numeriche esistenti sono computazionalmente lunghe. Esistono stimatori semi-parametrici che possono essere usati per ottenere dei buoni valori iniziali e per cercare di identificare l'ordine del dgp.

Nell'ambito delle serie storiche sono stati sviluppati negli ultimi anni vari metodi boot-

strap, si pensi ad esempio a metodi come il *model-based resampling*, il *block bootstrap* (Künsch, 1989), l'*autoregressive-aided periodogram* bootstrap (Kreiss and Paparoditis, 2003), il *local bootstrap* (Paparoditis and Politis, 1999), il SIEVE *bootstrap* (Kreiss, 1992), il *parametric bootstrap* (Andrews et al., 2006), il *kernel* bootstrap (Dahlhaus and Janas, 1996; Franke and Härdle, 1992) e il *phase scrambling* (Theiler et al., 1992). La replicazione di serie con metodi bootstrap apre la via ad interessanti applicazioni come lo sviluppo di test o l'approssimazione della distribuzione di tali test. Una vasta letteratura, inoltre, si occupa della costruzione di intervalli di confidenza basati sulle replicazioni bootstrap (Arteche and Orbe, 2005; Efron, 1979, 1982, 1987a,b; Hall, 1988, 1992b). Rimane ancora aperto il problema di replicare la struttura di dipendenza di una serie a memoria lunga di tipo ARFIMA(p, d, q).

In questa tesi sviluppiamo un nuovo metodo bootstrap per serie storiche, l'*ACF bootstrap*, basato su un risultato di Ramsey (1974), che genera serie surrogate a partire dalla funzione di autocorrelatione campionaria. La tesi è suddivisa in cinque capitoli: i primi due capitoli sono una rassegna della letteratura esistente, gli ultimi tre contengono i contributi innovativi della tesi.

Il primo capitolo descrive i processi a memoria lunga, le proprietà asintotiche delle funzioni di autocorrelazione e autocovarianza campionarie, i più comuni stimatori parametrici e semi-parametrici e, infine, processi di Gegenbauer. Il secondo capitolo è dedicato ad una rassegna dei principali metodi bootstrap per serie storiche.

Nella prima parte (Capitolo 3) introduciamo un nuovo metodo bootstrap per serie storiche. Applichiamo tale metodo per migliorare le performance di stimatori semi-parametrici del parametro di memoria lunga d in processi ARFIMA$(0, d, 0)$ in termini di minore errore standard, minore errore quadratico medio e migliore copertura degli intervalli di confidenza. L'assunzione di Gaussianità è alquanto restrittiva, tuttavia mostriamo, per mezzo di un'estesa simulazione Monte Carlo, che l'ACF bootstrap funziona anche in assenza di questa ipotesi. In particolare il metodo proposto sembra essere robusto rispetto a *code pesanti* e *asimmetria*. Altro campo di applicazione è la costruzione di intervalli di confidenza per il parametro di memoria d. Nel caso dello stimatore parametrico di Whittle gli intervalli basati sulla distribuzione bootstrap hanno una copertura più vicina a quella teorica se la serie è relativamente corta ($n = 128, 300$). Per gli stimatori semi-parametrici il bootstrap migliora la copertura degli intervalli per d quando il dgp è un processo ARFIMA$(1, d, 0)$.

La seconda parte della tesi (Capitolo 4) è dedicata allo studio del comportamento asintotico delle funzioni di autocorrelazione e di autocovarianza campionarie con lo scopo di dare un supporto teorico alla consistenza dell'ACF bootstrap nel replicare la memoria lunga.

Infine nel Capitolo 5, proponiamo un algoritmo per stimare non-parametricamente i processi di Gegenbauer con uno o due periodicità (uno o due picchi nella densità spettrale). L'ACF bootstrap è utile anche in questo contesto per fornire la stima della distribuzione del parametro di frequenza η. La sua distribuzione asintotica viene fornita per gli stimatori proposti da Chung (1996) e Sadek and Khotanzad (2004), ma è talmente complessa da essere quasi inutilizzabile. Lo scopo di questa parte del lavoro è proporre una procedura per indentificare persistenze stagionali e fornire dei buoni valori iniziali per massimizzare la funzione di verosimiglianza.

English

Fractionally integrated processes ARFIMA(p, d, q), introduced by Granger and Joyeux (1980) and Hosking (1981) independently, offer a useful tool to model the second order dependence structure (autocovariance and autocorrelation functions) of an observed time series. The literature is rich of paper on identification of the data generating process (dgp, from now on) and estimation of the parameters: Yajima (1985),Fox and Taqqu (1986) and Dahlhaus (1988, 1989) wrote papers on parametric estimate of the memory parameter d, whereas Hurst (1951), Geweke and Porter-Hudak (1983), Higuchi (1988), Robinson (1995a) and Hurvich et al. (1998) developed semi-parametric estimation methods. It is not possible to define the best method, according to the situation each method offers advantages and drawbacks. Parametric estimators are asymptotically Normal and they are the most efficient, however in the case of misspecification of the model the estimates might be dramatically biased. On the other hand, semi-parametric estimators offer the possibility of estimating the long memory parameter from the short memory part with the drawback of a slower convergence rate ($o(n^{-1/2})$ or less) than with parametric techniques ($o(n^{-1})$). Moreover, Agiakloglou et al. (1993) showed that the GPH (Geweke and Porter-Hudak, 1983) is biased in presence of ARMA parameter near the non-stationary area.

A generalisation of ARFIMA processes are the Gegenbauer processes, introduced by Hosking (1981) and then studied by Woodward et al. (1989), Woodward et al. (1998), Giraitis and Leipus (1995), Smallwood and Beaumont (2004), Sadek and Khotanzad (2004) and Caporale and Gil-Alana (2006). Also in this case parametric and semi-parametric technique are available in the literature. One the main problem is the maximization in a multidimensional space of the likelihood function because there is not a close form and the existing numerical procedures are quite burdensome. Semi-parametric procedures play and important role to compute good starting values to maximize the likelihood function and to identify the order of the dgp.

In the last years many bootstrap methods for time series have been developed, such as the *model-based resampling*, the *block bootstrap* (Künsch, 1989), the autoregressive-aided periodogram bootstrap (Kreiss and Paparoditis, 2003), the *local bootstrap* (Paparoditis and Politis, 1999), the SIEVE *bootstrap* (Kreiss, 1992), the *parametric bootstrap* (Andrews et al., 2006), the kernel bootstrap (Dahlhaus and Janas, 1996; Franke and Härdle, 1992) and the *phase scrambling* (Theiler et al., 1992). Bootstrap methods for time series have been widely used to build confidence intervals especially when asymptotic theory does not provide satisfactory results (Arteche and Orbe, 2005; Efron, 1979, 1982, 1987a,b; Hall, 1988, 1992b). The problem is still open when we want to replicate the dependence structure of a long memory process such as ARFIMA(p, d, q).

In this thesis we develop a new bootstrap method for time series, the *ACF bootstrap*, based on a result of Ramsey (1974), that generates the surrogate series from the observed autocorrelation function. The thesis is divided in five chapters: the first two chapters review some literature, the last three chapters are new contributions.

The first chapter reviews the literature on long memory processes, the properties of their sample autocorrelation and autocovariance functions, the most common parametric and semi-parametric estimators and, shortly, Gegenbauer processes. In the second chapter,

we introduce briefly some bootstrap methods for time series.

In Chapter 3 we introduce the new bootstrap method. We apply the ACF bootstrap to improve the performance of semi-parametric estimators for the memory parameter d for ARFIMA$(0, d, 0)$ processes in terms of smaller standard error, smaller mean squared error and better coverage for confidence intervals. Since the condition of Gaussianity of the observed process is very restrictive, we show, by means of Monte Carlo simulation, that the method is consistent even relaxing this hypothesis. In particular the method seems to be robust against *fat tails* and *asymmetry*. Another application is building confidence intervals for the memory parameter d. For the parametric Whittle estimator, the confidence intervals based on the bootstrap distribution have a closer coverage to the theoretical level if the time series is relatively short ($n = 128, 300$). For semi-parametric estimators, applying bootstrap improves coverage of confidence intervals for d when the dgp is a ARFIMA$(1, d, 0)$ process.

In Chapter 4 we study the asymptotic behaviour of sample autocovariance and autocorrelation functions of a long memory processes. This results are useful to give a theoretical support for the consistency of the method in replicating long memory.

Last, in Chapter 5, we propose an algorithm to estimate non-parametrically the parameters of a Gegenbauer process with one and two peaks in the spectral density. The bootstrap method will be useful to give an estimate of the distribution of the frequency parameter η. Its asymptotic distribution is given for the estimators proposed by Chung (1996) and Sadek and Khotanzad (2004) but it is very complicate and difficult to handle. The main aim is proposing a method to identify seasonal persistences and provide starting values for maximize a (penalized) likelihood function.

Chapter 1

Long memory processes

In this chapter we introduce long memory processes: this is a wide class of processes, however there are some main common features that can be found in all of them. Long memory was noticed in some processes because the decaying rate of the variance of the sample mean was proportional to $n^{-\alpha}$, with $\alpha \in (0,1)$, while the usual rate for a sample of i.i.d. observations or weakly correlated data is n^{-1}, where n is the series length. Another common feature in these series is that the autocovariance function decays to zero very slowly, as if very distant observations in the past still affect present behaviour of the series. Moreover, even though the series may look stationary ovk,erall there appears to be local trends and long periods with very large (or very small) observations without persistent cycles. The variance of the sample mean of correlated data depends also on the autocorrelations

$$\mathrm{var}(\bar{X}) = \frac{\sigma^2}{n^2} \sum_{i,j=1}^{n} \rho(i,j) = \sigma^2 \frac{(1+\delta_n(\rho))}{n},$$

where $\delta_n(\rho) = n^{-1} \sum_{i \neq j} \rho(i,j)$, with $\rho(i,j)$ =E$[(X_i - \mu)(X_j - \mu)]$ and μ =E$[X_i]$. This term is zero for non-correlated sequences. For weakly correlated data $\delta_n(\rho)$ is constant for each n. However, there exist processes where $\delta_n(\rho)$ increases with the sample size, affecting the decaying rate of the variance of the sample mean: in these cases the usual variance of the sample mean, i.e. the variance of the single observation divided by the sample size, is too small. These considerations lead to the belief that in some series the autocovariance function influences the sample variance so strongly as to change its decaying rate. On the other hand, confidence intervals for the sample mean $\bar{X}$ based on the Normal approximation

$$\bar{X} \pm z_{\alpha/2} s n^{-1/2}, \qquad s = \sqrt{\frac{1}{n} \sum_{j=1}^{n} (X_j - \bar{X})^2}$$

are too short. From a quantitative point of view it is also possible to notice a pole near the zero frequency in the spectral density of the process. Another consideration, from Baillie

(1996), is about the integration rate: these series are consistent neither with $I(0)$ process nor with $I(1)$ process. Their behaviour is somehow in between the two processes.

Fields like hydrology and climatology are rich with examples of long memory time series (Baillie, 1996). One of the most famous examples is the Nile minima: in Figure 1.1 there are the yearly minima water levels of the Nile River for the years $1871 - 1970$, measured at the Roda Gauge near Cairo (Beran, 1994, p. 22). Another example is the series of the tree ring data studied by Baillie (1996), shown in Figure 1.2: the series reports the annual tree ring measurement from Mount Campito from 3436BC to 1969AD for a total of 5405 observations. In Figure 1.3 there is the autocorrelation function of the tree ring data and it is possible to see that it decays to zero very slowly and also after more than 100 lags it is still significantly different from zero. Hurst (1951) analysed the persistence in hydrological and geophysical time series and to him it is due the Hurst estimator of the self-similarity parameter. This will be discussed shortly.

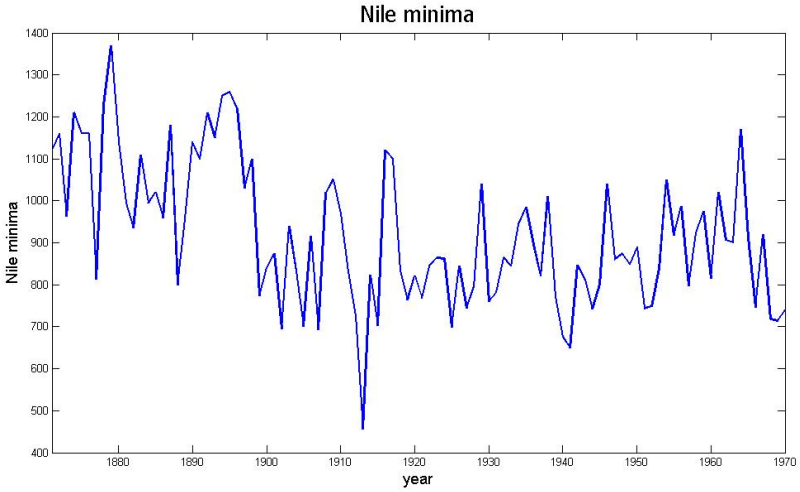


Figure 1.1: *Yearly minima water levels of the Nile River for the years* $1871 - 1970$*, measured at the Roda Gauge near Cairo (Beran, 1994, p. 22).*

More recently, measured traffic generated by modern high-speed communication networks exhibits long memory behaviour: Leland et al. (1994) showed that data collected on several Ethernet LAN's at the Bellcore Momstown Research and Engineering Center are statistically self-similar. The link between long memory and self-similarity will be clear later in this chapter in Section 1.1.

In the last twenty years a wide literature has been dedicated to the study of this type of memory structure, especially since the two papers of Granger and Joyeux (1980) and Hosking (1981) introduced, separately, the concept of fractional integration in time series analysis, by allowing the parameter d in an ARIMA(p, d, q) to assume non-integer real values. Before them fractional Brownian motion and fractional Gaussian noise had been introduced by Mandelbrot and van Ness (1968) (see below, this chapter).

As we just showed, fractional differenced processes have been widely used in diverse

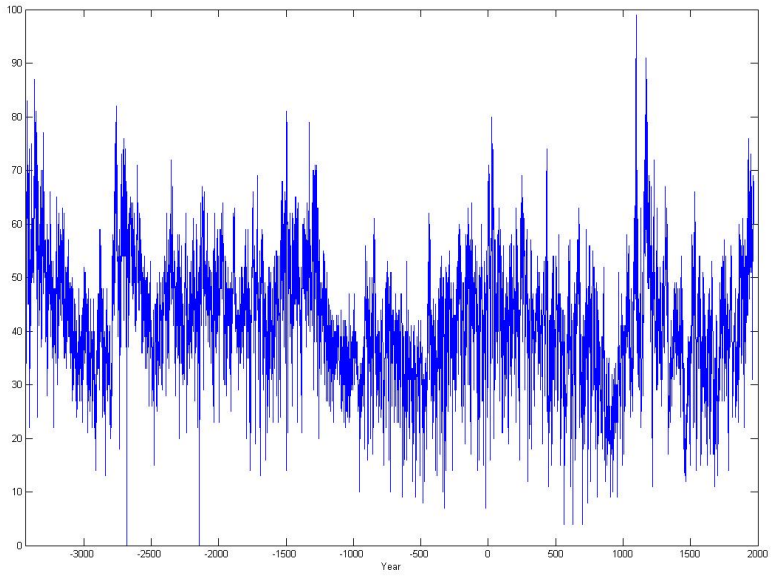


Figure 1.2: *Series of tree ring data measurement from Mount Campito between 3436BC and 1969AD for a total of 5405 observations.*

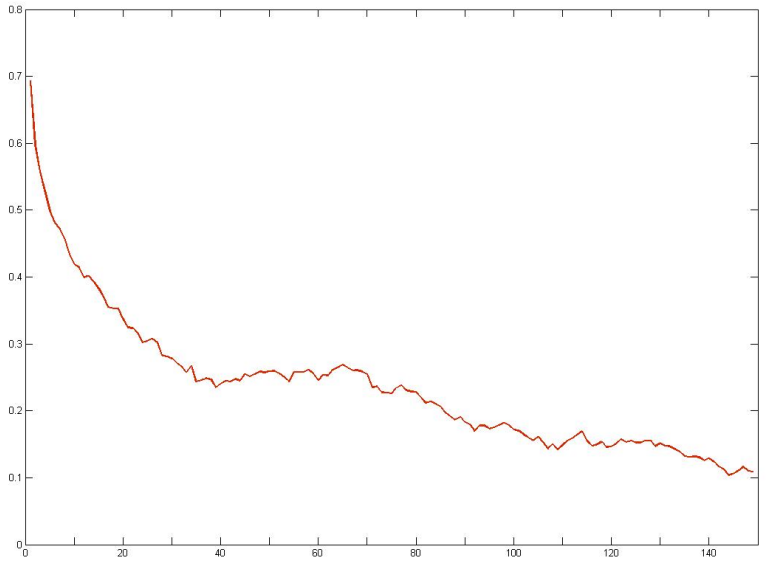


Figure 1.3: *Autocorrelation of the tree ring data, the series is given in Figure 1.2.*

fields, thus there are principally three different names for the measure of long memory in a process. The quantity $H \in (0,1)$ is called *Hurst exponent* or *self-similarity parameter* and is universally recognized by the literature on self-similar processes and fractional Brownian motion. The parameter $d \in (-0.5, 0.5)$ is the memory parameter and is due to the literature about integrated ARIMA(p,d,q) processes since when Granger and Joyeux (1980) and Hosking (1981) introduced the idea of considering a real non-integer value for d. Lastly $\alpha \in (0,2)$ indicates the convergence rate to zero of the variance of the sample mean of a long memory process. The relations between these parameter are given by

$$d = H - \tfrac{1}{2} \text{ and}$$
$$\alpha = 1 - 2d.$$

In particular, long memory is observed for $0 \leq d \leq 0.5$, i.e. $0.5 \leq H \leq 1$ and $0 \leq \alpha \leq 1$. We will introduce each of them in more detail throughout this chapter.

Now we are ready to give a mathematical definition of long memory processes (Beran, 1994).

Definition 1.1 *Let X_t be a stationary process for which the following holds. There exists a real number $\alpha \in (0,1)$ and a constant $c_\rho > 0$ such that*

$$\lim_{k \to \infty} \frac{\rho(k)}{c_\rho k^{-\alpha}} = 1$$

or alternatively,

$$\rho_k \sim c_\rho k^{-\alpha} \qquad \text{as } k \to \infty.$$

Then X_t is called a stationary process with long memory or long range dependence or strong dependence, or a stationary process with slowly decaying or long range correlations.

Definition 1.2 *Let X_t be a stationary process for which the following holds. There exists a real number $\alpha \in (0,1)$ and a constant $c_f > 0$ such that*

$$\lim_{\omega \to 0} \frac{f(\omega)}{c_f |\omega|^{\alpha - 1}} = 1$$

or alternatively,

$$f(\omega) \sim c_f |\omega|^{\alpha - 1} \qquad \text{as } \omega \to 0,$$

where $f(\omega)$ is the spectral density of the process X_t.

Then X_t is called a stationary process with long memory or long range dependence or strong dependence.

Since the convergence rate of the autocorrelation function implies that, for a long memory process, the sum is not finite, i.e. $\sum_{k=-\infty}^{\infty} \rho(k) = \infty$, some authors use this property as a definition for long memory.

It is possible to distinguish two principal long memory Gaussian processes. They have the same origin but one is defined in the continuous time and is called *fractional Gaussian noise*, whereas the second is its discrete version and is called *fractionally integrated noise*. We discuss them in Sections 1.1 and 1.2. In Section 1.3 we review literature concerning asymptotics on the sample autocovariance and autocorrelation functions: we will need these results to show the consistency of the ACF bootstrap for long memory processes (Chapter 3). Section 1.4 is dedicated to the problem of estimation and identification of long memory.

1.1 Fractional Gaussian noise

Fractional Gaussian noise is part of the very wide family of self-similar processes. For sake of completeness, we will give a brief introduction to the concept of self-similarity, also in view of a better understanding of long memory in general, and of estimation methods for the memory parameter in particular. In the following we will index the process $X(t)$ instead of X_t, because we want to distinguish between continuous and discrete processes.

1.1.1 Self-similar processes

A definition of self-similar process can be found in Samorodnitsky and Taqqu (1994), as follows.

Definition 1.3 *The real valued process $X(t)$[1], $t \in T$, is self-similar with index $H > 0$ (H-ss) if, for all $a > 0$, the finite-dimensional distributions of $X(at)$ are identical to the finite-dimensional distributions of $a^H X(t)$; i.e., if for any $k \geq 1$, $t_1, \ldots, t_k \in T$ and any $a > 0$,*

$$(X(at_1), X(at_2), \ldots, X(at_k)) \stackrel{d}{=} (a^H X(t_1), a^H X(t_2), \ldots, a^H X(t_k)), \tag{1.1}$$

where H is the self-similarity parameter (and plays a key role in the study of these processes).

Self-similarity is connected with the idea of fractals: these objects exhibit the same pattern, type of structure, on all scales. Nature is full of fascinating examples of self-similar processes: mountains, waves, leaves, fly's paths. Each branch of the fern leaf in Figure 1.4 repeats on a smaller scale throughout the whole leaf. The fractal originated by a Mandelbrot set[2] in Figure 1.5 shows the same pattern in each enlargement and it is virtually possible to keep enlarging it so that the pattern will repeat endlessly.

The botanist Robert Brown observed with a microscope the erratic path of pollen particles floating in water. Restricting the path to one dimension we obtain the simplest of

[1] In the context of this thesis, the notation X_t is used for a discrete process where t assumes integer values, $t = 0, \pm 1, \pm 2, \ldots$, whereas $X(t)$, $t \in T$, is a more general notation where T can be an interval of the real set $\mathbb{R}$, i.e. $X(t)$ is a continuous process.

[2] The Mandelbrot set is generated by a quadratic recurrence equation. Starting with a set of points $z_0 = C$ in the complex plane, $z_{n+1} = z_n^2 + C$ (for a definition of Mandelbrot set see, e.g., Devaney, 1999).

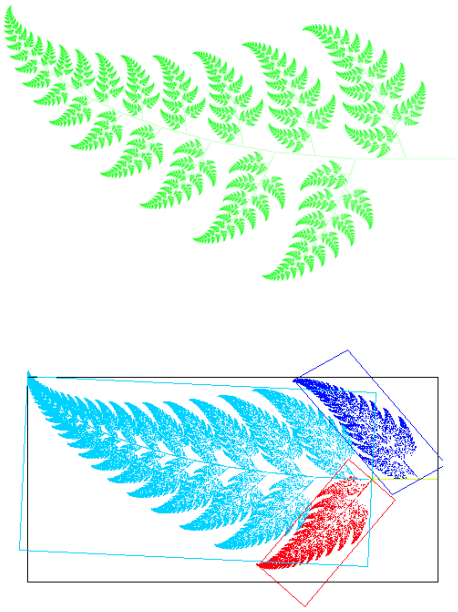
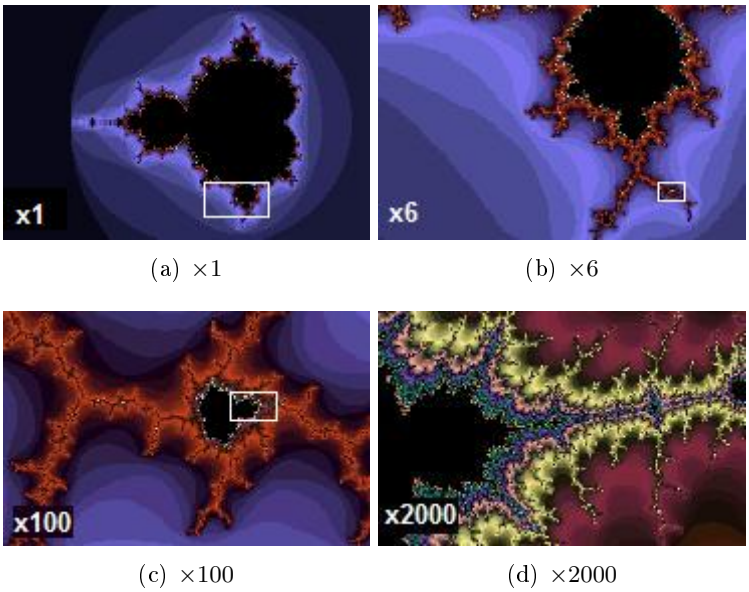


Figure 1.4: *Fern leaves are an example of nature self-similarity (source from webpage http://en.wikipedia.org/wiki/Fractal, 24^{th} September 2006).*



(a) $\times 1$ (b) $\times 6$

(c) $\times 100$ (d) $\times 2000$

Figure 1.5: *The Mandelbrot Set at four different enlargements (source from web pages http://en.wikipedia.org/wiki/Fractal and http://en.wikipedia.org/wiki/Self-similarity, 24^{th} September 2006).*

fractals, called the *Brownian motion* in honour of the botanist. This is the most important self-similar process and is at the heart of many different self-similar processes (Crilly et al., 1991). It plays a similar role to that of the Normal distribution in probability. The following definitions are respectively Beran (1994) (for an alternative definitio of the Brownian motion see, e.g. Rogers and Williams, 1979).

Definition 1.4 *Let $B(t)$ be a stochastic process with continuous sample paths and such that*

(i) *$B(t)$ is Gaussian,*

(ii) *$B(0) = 0$ almost surely,*

(iii) *$B(t)$ has independent increments, i.e. $\forall t, s \geq 0$ $B(t) - B(s)$ is independent of $B(u)$, $0 \leq u \leq \min(t, s)$,*

(iv) *$E[B(t) - B(s)] = 0$, and*

(v) *$\text{Var}[B(t) - B(s)] = \sigma^2 |t - s|$.*

Then $B(t)$ is called Brownian motion. *If $\sigma^2 = 1$, then we have the standardized Brownian motion.*

Generalizing Brownian motion we obtain *fractional Brownian motion* (fBm) with self-similarity parameter H.

Definition 1.5 *Let $a > 0$ be a positive scaling constant, and define a weight function w_H by*

$$\begin{aligned} w_H(t,u) &= 0 && \text{for } t \leq u, \\ w_H(t,u) &= (t-u)^{H\frac{1}{2}} && \text{for } 0 \leq u < t, \\ w_H(t,u) &= (t-u)^{H\frac{1}{2}} - (-u)^{H\frac{1}{2}} && \text{for } u < 0. \end{aligned}$$

Also, let $B(t)$ be a standardized Brownian motion. For $0 < H < 1$, let $B_H(t)$ be defined by the stochastic integral

$$B_H(t) = a \int w_H(t,u) dB(u), \tag{1.2}$$

where the convergence of the integral is to be understood in the L^2 norm with respect to the Lebesgue measure on the real numbers. Then $B_H(t)$ is called fractional Brownian motion *with self-similarity parameter H.*

Fractional Brownian motion has self-similarity parameter $0 < H < 1$: this interval guarantees stationary increments. For $H = 1/2$ we have Brownian motion.

In the following example we introduce the Rosenblatt process because it is the limiting distribution of the sample autocorrelation function of some long memory processes (refer to Section 1.3).

Example 1.6 The Rosenblatt process. *Another example of a self-similar process of order H with stationary long memory increments is the Rosenblatt process (??; Albin, 1998a,b; Leonenko and Anh, 2001; Tudor, 2006), that, we will see later, plays an important role in the so-called Non Central Limit Theorem. It is a special case of Hermite processes, a Wiener-Itô stochastic integral with respect to the Brownian motion $B(y)$,*

$$Z_H^k(t) = c(H,k) \int_{\mathbb{R}^k} \int_0^t \left(\prod_{j=1}^k (s-y_i)_+^{-\left(\frac{1}{2}+\frac{1-H}{k}\right)} \right) dsdB(y_1)\dots dB(y_k),$$

where $x_+ = \max(x,0)$ and $c(H,k)$ is a normalizing constant. In the bidimensional case, when $k=2$, we have the Rosenblatt process

$$Z_H(t) = c(H,2) \int_{\mathbb{R}^2} \int_0^t [(s-y_1)_+(s-y_2)_+]^{-\left(1-\frac{H}{2}\right)} dsdB(y_1)dB(y_2).$$

Unfortunately there is not a close form for the marginal distribution of this process, the Rosenblatt distribution, it is only possible to write the characteristic function

$$E\{e^{itY}\} = \exp\left\{ \sum_{k=2}^{\infty} \frac{(2it)^k}{2k} \int_{x\in[0,1]^k} |x_1 - x_k|^{2H} \prod_{j=2}^{k} |x_j - x_{j-1}|^{2H} \right\}.$$

1.1.2 From fractional Brownian motion to fractional Gaussian noise

Fractional Gaussian noise (fGn) is defined as the sequence of stationary increments of the fractional Brownian motion

$$Y_t = B_H(t+1) - B_H(t), \qquad t = 0, \pm 1, \dots$$

where $B_H(t)$ was given in (1.2). It can be also thought as the $(1/2-H)$ fractional derivative of continuous time white noise.

The following proposition provides the form of the autocovariance function and of the spectral density for a fractional Brownian motion; the proposition and its proof can be found in Samorodnitsky and Taqqu (1994).

Proposition 1.7 *The fractional Gaussian noise has autocovariance function*

$$\gamma_k = \frac{\sigma^2}{2}\left(|k+1|^{2H} - 2|k|^{2H} + |k-1|^{2H}\right), \qquad k = 0, \pm 1, \dots$$

and spectral density

$$f(\omega) = \frac{\sigma^2}{C^2(H)} |e^{i\omega} - 1|^2 \sum_{k=-\infty}^{\infty} \frac{1}{|\omega + 2\pi k|^{2H+1}}, \quad -\pi \le \omega \le \pi$$

where

$$C(H) = \left(\frac{\pi}{H\Gamma(2H)\sin H\pi} \right)^{1/2} \tag{1.3}$$

or, equivalently,

$$f(\omega) = \frac{\sigma^2 \int_0^\infty \cos x\omega (\sin^2(x/2)) x^{-2H-1} \mathrm{d}x}{\int_0^\infty \omega (\sin^2(x/2)) x^{-2H-1} \mathrm{d}x}, \quad -\pi \leq \omega \leq \pi.$$

The same authors proved that the autocovariance function of a fractional Gaussian noise decays hyperbolically and that its spectral density has a pole at the zero frequency: thus fractional Gaussian noise exhibits long memory.

Proposition 1.8 *Let X_t be fractional Gaussian noise. Then*

$$\gamma_k \sim \sigma_0^2 H(2H-1)k^{2H-2} \qquad \text{as } k \to \infty$$

for $H \neq 1/2$, and

$$f(\omega) \sim \frac{\sigma_0^2 |\omega|^{1-2H}}{C^2(H)} \qquad \text{as } \omega \to \infty$$

where $C(H)$ was defined in Formula (1.3).

We should say that, for $H = 1/2$, fractional Gaussian noise is white noise, and because it is Gaussian it is i.i.d..

1.2 Fractionally integrated processes

Until the works of Granger and Joyeux (1980) and Hosking (1981) linear time series were divided into stationary short memory and non-stationary (or integrated of order 1) time series. The former type of processes is the ARMA(p,q) and is a useful tool to model short memory behaviour. It is the stationary solution of the stochastic difference equation

$$\Phi(B)(X_t - \mu) = \Theta(B)\varepsilon_t$$

where μ is the mean of the process, B is the backward shift operator $BX_t = X_{t-1}$, ε_t is white noise $(0, \sigma^2)$, $\Phi(z) = 1 - \phi_1 z - \ldots - \phi_p z^p$ and $\Theta(z) = 1 + \theta_1 z + \ldots + \theta_q z^q$. For the process to be identifiable, $\Phi(\cdot)$ and $\Theta(\cdot)$ must have no common roots, and to be causal and invertible, the two polynomials must have all roots outside the unit circle $\Phi(z) \neq 0$ and $\Theta(z) \neq 0$ for $|z| \leq 1$ (for more details on ARMA processes, see, e.g. Brockwell and Davis, 1991; Chatfield, 1996; Wei, 1990). Without loss of generality we can assume from now on that X_t has $\mu = 0$ and ε has unit variance. The spectral density of an ARMA(p,q) process

is given by

$$f(\omega) = \frac{\sigma^2}{2\pi} \frac{|\Theta(e^{-i\omega})|}{|\Phi(e^{-i\omega})|}, \qquad -\pi \leq \omega \leq \pi,$$

where $\sigma^2 = 1$.

Box and Jenkins (1976) introduced non-stationary linear time series as a generalisation of ARMA processes, in the sense that Y_t is ARIMA(p, d, q), for a positive integer d, if after differencing Y_t a finite number of time (exactly d) we obtain an ARMA process X_t as defined in Formula (1.4):

$$\begin{aligned} \nabla^d X_t &= Y_t \\ \Phi(B)\nabla^d X_t &= \Theta(B)\varepsilon_t. \end{aligned} \tag{1.4}$$

where $\nabla X_t = (1-B)X_t$. Here, d is also allowed to take negative integer values and in this case we obtain an ARMA(p, q) by integrating the process exactly d times. ARIMA(p, d, q) processes are also called $I(d)$, i.e. integrated processes of order d. In this view ARMA(p, q) processes are a special case with $d = 0$ and they are $I(0)$. Another very common case is given by $d = 1$ when the series X_t is non-stationary but the differenced series, $Y_t = X_t - X_{t-1}$, is an ARMA process. Even though this class of processes is quite flexible because it can describe reasonably well the data generating process in many situations, some series are neither $I(0)$ nor $I(1)$: their behaviour seems to be in between these two cases. For this reason Granger and Joyeux (1980) and Hosking (1981) considered, independently, the possibility for d to assume any real number.

Fractional differenced processes possess many good qualities as highlighted by Hosking (1981): they are flexible enough to allow modelling of both short term and long term dependent behaviour and it is quite easy to generate a synthetic series from the model.

Of particular interest is the so-called *fractionally integrated noise* FI(d), i.e. ARFIMA$(0, d, 0)$ defined as the stationary solution of

$$\nabla^d X_t = \varepsilon_t.$$

The interesting range of values for d is $(-0.5, 0.5)$. It guarantees that the process is invertible ($d > -0.5$) and stationary ($d < 0.5$). When the parameter lies outside this interval we can integrate or differentiate the series an appropriate number of time until we fall in this range. We have long memory when d is positive, while for d negative it is said there is intermediate memory or antipersistency. For $d = 0$ the process X_t is a white noise.

It is possible to approximate a fractionally integrated noise with an ARMA process of large p, q order, but the approximation is quite poor: estimating many parameters leads to uncertain statistical inference and also their interpretation becomes complex and difficult. The theorem in Hosking (1981), stated below, provides a good summary of the properties of an ARFIMA$(0, d, 0)$ process. In the following we will use the Gamma function defined in

Brockwell and Davis (1991) for every real number as

$$\Gamma(x) = \begin{cases} \int_0^\infty t^{x-1}e^{-t}dt & x > 0, \\ \infty & x = 0, \\ x^{-1}\Gamma(1+x) & x < 0. \end{cases}$$

Theorem 1.9 *Let X_t be an FI(d) process.*

(a) When $d < 0.5$, X_t is stationary and has an infinite moving-average representation

$$X_t = \sum_{j=0}^{\infty} \psi_j \varepsilon_{t-j},$$

where

$$\psi_j = \frac{\Gamma(j+d)}{\Gamma(j+1)\Gamma(d)} = \prod_{0<k\leq j} \frac{k-1+d}{k} \sim \frac{j^{d-1}}{\Gamma(d)}. \tag{1.5}$$

(b) When $d > -0.5$, X_t is invertible and has infinite autoregressive representation

$$\sum_{j=0}^{\infty} \pi_j X_{t-j} = \varepsilon_t,$$

where

$$\pi_j = \frac{\Gamma(j-d)}{\Gamma(j+1)\Gamma(-d)} = \prod_{0<k\leq j} \frac{k-1-d}{k} \sim \frac{j^{-d-1}}{\Gamma(-d)}.$$

(c) The spectral density of X_t is $f(\omega) = (2\sin(\omega/2))^{-2d}/2\pi$ for $0 < \omega \leq \pi$ and $f(\omega) \sim \omega^{-2d}$ as $\omega \to 0$.

(d) The autocovariance function of X_t is

$$\gamma_k = E(X_t X_{t-k}) = \frac{\Gamma(k+d)\Gamma(1-2d)}{\Gamma(k+1-d)\Gamma(1-d)\Gamma(d)},$$

the autocorrelation function of X_t is

$$\rho_k = \frac{\gamma_k}{\gamma_0} = \frac{\Gamma(k+d)\Gamma(1-d)}{\Gamma(k+1-d)\Gamma(d)},$$

and

$$\rho_k \sim k^{2d-1}\frac{\Gamma(1-d)}{\Gamma(d)} \qquad \text{as } k \to \infty.$$

It can be shown that the asymptotic behaviours of the autocorrelation functions of ARFIMA(p,d,q) and FI(d) are the same, because the effect of the short memory parameter is negligible for distant lags.

The literature is full of estimators for these processes and we will introduce the most important later in this chapter. However it is interesting to highlight that parametric methods estimate all the parameters, $\vartheta = (\sigma^2, \phi_1, \ldots, \phi_p, d, \theta_1, \ldots, \theta_q)$, simultaneously, whereas semiparametric methods are usually concerned with estimating the long term behaviour. Both cases are of interest and either can outperform the other depending on the situation.

1.3 Asymptotics for the autocorrelation function

In this section we review the literature of asymptotic results for the distribution of the sample autocorrelation (and autocovariance) function of linear Gaussian processes. The importance of these results will be clear in Chapter 4, when we will study the asymptotic behaviour of sample autocovariance and autocorrelation functions of long memory processes.

A linear Gaussian process can be expressed in the form

$$X_t = \sum_{j=0}^{\infty} \psi_j \varepsilon_{t-j} \tag{1.6}$$

where ε_t is a Gaussian white noise and ψ_j is a sequence of constant, such that $\sum_{j=0}^{\infty} \psi_j^2 < \infty$ (Priestley, 1988). This class of processes includes stationary autoregressive moving average, fractional integrated and Gegenbauer processes.

The pioneering paper was written by Bartlett (1946), when he investigated the properties of the autocorrelation function of stationary linear processes. He started from the simple autoregressive model of order 1, $X_t = \rho X_{t-1} + \varepsilon_t$, and generalised first to an autoregressive process of any order and finally to linear processes as defined by Equation (1.6). The formulas, he gave in this paper, are very imprtant in the literature and are known as *Bartlett's formulas*

$$\operatorname{cov}(\hat{\gamma}_k, \hat{\gamma}_j) \simeq \frac{1}{n} \left\{ \sum_{i=-\infty}^{\infty} (\gamma_i \gamma_{i+k-j} + \gamma_i \gamma_{i+k+j}) + \kappa \gamma_k \gamma_j \right\} \tag{1.7}$$

$$\begin{aligned} \operatorname{cov}(\hat{\rho}_k, \hat{\rho}_j) \simeq \; & \frac{1}{n} \sum_{i=-\infty}^{\infty} (\rho_i \rho_{i+k-j} + \rho_i \rho_{i+k+j} + 2\rho_k \rho_j \rho_i^2 \\ & -2\rho_k \rho_i \rho_{i+j} - 2\rho_j \rho_i \rho_{i+k}) \end{aligned} \tag{1.8}$$

and the variance

$$\mathrm{var}(\hat{\gamma}_k) \simeq \frac{1}{n}\left\{\sum_{i=-\infty}^{\infty}(\gamma_i^2 + \gamma_{i+k}\gamma_{i-k}) + \kappa\gamma_k^2\right\} \tag{1.9}$$

$$\mathrm{var}(\hat{\rho}_k) \simeq \frac{1}{n}\left\{\sum_{i=-\infty}^{\infty}(\rho_i^2 + \rho_{i+k}\rho_{i-k}) - 4\rho_k\rho_i\rho_{i-k} + 2\rho_k^2\rho_i^2\right\} \tag{1.10}$$

of the sample autocovariance and autocorrelation functions, where κ is the excess of kurtosis[3] and $\kappa = 0$ if ε_t is Normal.

The following important result is due to Anderson and Walker (1964). They showed the asymptotic Normality of the estimator of the autocorrelation function for a class of linear stationary processes. They defined the estimator of the autocorrelation function as $r_k = \tilde{\gamma}_k/\tilde{\gamma}_0$, with

$$\tilde{\gamma}_k = \frac{1}{n-k}\sum_{t=1}^{n-k}(X_t - \mu)(X_{t+k} - \mu) \tag{1.11}$$

if the mean of the process μ is known, and otherwise $r_k^* = \tilde{\gamma}_k^*/\tilde{\gamma}_0^*$, with

$$\tilde{\gamma}_k^* = \frac{1}{n-k}\sum_{t=1}^{n-k}(X_t - \bar{X})(X_{t+k} - \bar{X})$$

where $\bar{X}$ is the sample mean. They showed that any set of sample autocorrelations $n^{1/2}(r_k - \rho_k)$ of finite s dimension is asymptotically s-variate Normal with zero mean and non-singular covariance function

$$W = (w_{kj}) = \mathrm{cov}(\hat{\rho}_k, \hat{\rho}_j), \tag{1.12}$$

where $\mathrm{cov}(\hat{\rho}_k, \hat{\rho}_j)$ is given in formula (1.8). The result we give in the following is valid also when we divide by n in formula (1.11).

Theorem 1.10 *Let X_t be a linear stochastic process, defined by*

$$(X_t - \mu) = \sum_{i=-\infty}^{\infty}\psi_i\varepsilon_{t-i}, \qquad t = 0, \pm1, \pm2, \ldots,$$

where $\sum_{i=-\infty}^{\infty}|\psi_i| < \infty$ and $\sum_{i=-\infty}^{\infty}|i|\psi_i^2 < \infty$ and ε_t is a set of independently and

[3]The excess of kurtosis is given by

$$\kappa_4 = \frac{k_4}{\mu_2^2} - 3,$$

where μ_2 and k_4 are the second and fourth order moments of X_t (Abramowitz and Stegun, 1972).

identically distributed random variables with $E(\varepsilon_t) = 0$ *and* $E(\varepsilon_t^2) = \sigma^2 < \infty$. *Let* $r_k = \tilde{\gamma}_k/\tilde{\gamma}_0$, $k = 1, 2, \cdots, n-1$. *Then the joint distribution of* $n^{1/2}(r_k - \rho_k)$, $1 \leq k \leq s \leq n$, *tends to an s-variate Normal distribution* $N(0, W)$ *when* $n \to \infty$, *where* W *is given in (1.12).*

Corollary 1.11 *Under the conditions of the theorem the joint distribution of* $n^{1/2}(r_k^* - \rho_k)$, $1 \leq k \leq s \leq n$, *tends to* $N(0, W)$ *when* $n \to \infty$ *where* W *is given in (1.12).*

The next contribution is due to Cavazos-Cadena (1994) that proved the same result of Anderson and Walker (1964) under slightly weaker conditions, i.e. squared summability of the infinite moving average representation coefficients ψ_i (see Formula (1.5)) and squared integrability of the spectrum.

Assumption 1.12 *The innovations* ε_t *are independent and identically distributed with zero mean and variance* 1.

Assumption 1.13 ψ_i *is non-null*($\psi_i \neq 0$), *for some* $i = 0, 1, \ldots$, *and*

$$\sum_{i=0}^{\infty} \psi_i^2 < \infty.$$

Assumption 1.14 *The spectral density is squared integrable*

$$\int_{-\pi}^{\pi} f(\omega)^2 d\omega < \infty.$$

In the following two theorems W_k, $k = 0, 1, \ldots$, is a sequence of i.i.d. random variables with a standard Normal distribution.

Theorem 1.15 *Suppose that Assumptions 1.12-1.14 hold true and that*

$$E(\varepsilon^4) < \infty. \tag{1.13}$$

Then, for every positive integer h

(i) *as* $n \to \infty$,

$$n^{1/2}(\hat{\gamma}_0 - \gamma_0, \ldots, \hat{\gamma}_h - \gamma_h) \to (Y_0, Y_1, \ldots, Y_h)$$

where

$$Y_j = (\gamma_j K) W_0 + \sum_{k=1}^{\infty} (\gamma_{k+j} + \gamma_{k-j}) W_k, \qquad j \geq 0,$$

and $K = (E(\varepsilon^4) - 1)^{1/2}$;

(ii) *as* $n \to \infty$,

$$n^{1/2}(\hat{\rho}_1 - \rho_1, \ldots, \hat{\rho}_h - \rho_h) \to (R_1, \ldots, R_h) \tag{1.14}$$

where

$$R_j = \sum_{k=1}^{\infty}(\rho_{k+j} + \rho_{k-j} - 2\rho_k\rho_j)W_k, \qquad j \geq 1. \tag{1.15}$$

In his second result, Cavazos-Cadena (1994) relaxes the condition given in formula (1.13) introducing a restriction on the sequence of ψ_i (see theorem below). This means that, under these restrictions, the sample autocorrelation and the sample autocovariance functions are asymptotically Normal even if the process is not Gaussian. In the following theorem two new quantities are introduced

$$\begin{array}{ll} \hat{v} = \sum_{t=1}^{n} \varepsilon_t^2/n, & \text{if} \qquad \sum_{t=1}^{n} \varepsilon_t^2 \neq 0, \\ \hat{v} = 1 & \text{otherwise} \end{array}$$

and

$$\bar{\gamma}_k = \frac{\hat{\gamma}_k}{\hat{v}}, \qquad k \in \mathbb{Z}.$$

Theorem 1.16 *Suppose that Assumptions 1.12-1.14 hold true and that*

$$m \sum_{|k|>m} \psi_k^2 \to 0 \qquad as \qquad m \to \infty.$$

Then, for every positive integer h

(i) *as* $n \to \infty$,

$$n^{1/2}(\bar{\gamma}_0 - \gamma_0, \ldots, \bar{\gamma}_h - \gamma_h) \to (\bar{Y}_0, \bar{Y}_1, \ldots, \bar{Y}_h)$$

where

$$\bar{Y}_j = \sum_{k=1}^{\infty}(\gamma_{k+j} + \gamma_{k-j})W_k, \qquad j \geq 0;$$

(ii) *as* $n \to \infty$,

$$n^{1/2}(\hat{\rho}_1 - \rho_1, \ldots, \hat{\rho}_h - \rho_h) \to (R_1, \ldots, R_h)$$

where R_j *is defined in Formula (1.15).*

Another small step is due to He (1996): he extended existing results and was able to show that also ARFIMA processes with $0 < d < 0.25$ have asymptotic Normality for sample autocorrelation function.

Assumption 1.17 *The sequence ψ_i satisfies, for some $\tau > 1/2$,*

$$m^\tau \sum_{|k|>m} \psi_k^2 \to 0 \qquad as \qquad m \to \infty.$$

Theorem 1.18 *Let X_t be defined by Equation (1.6) with spectral density $f(\omega)$ being squared integrable and Assumption 1.17 being valid for some $\tau > 1/2$. Suppose that ε_t is a strictly stationary martingale difference white noise with variance σ^2 and satisfies one of the following conditions:*

(i) *$E(\varepsilon_t^2 | \varepsilon_s; s < t) = \sigma^2$ a.s. for all t (no* GARCH *effects);*

(ii) *ε_t is ergodic, $E(\varepsilon_t^2 \varepsilon_s^2) = \sigma^4$ for all $s \neq t$, and $E(\varepsilon_t^2 \varepsilon_{t+j} \varepsilon_{t+k}) = 0$ for any $j, k \geq 0$, $j \neq k$.*

Then for any positive integer h the result (1.14) is true.

Eventually, Hosking (1996) gave the asymptotic distribution of the autocorrelation and autocovariance functions for ARFIMA processes as defined by Equation (1.4). He showed that the sample autocovariance and autocorrelation functions are Normal when $0 < d \leq 1/4$, even though with different convergence rates, while the limiting distribution for larger values of d, i.e. $1/4 < d < 1/2$, is a Rosenblatt distribution (see Example (1.6)) with cumulants given in the following theorems.

Theorem 1.19 *Let X_t satisfy Formula (1.6) and*

$$\begin{aligned} \gamma_k &\sim \lambda k^{2d-1}, & \lambda > 0, \quad 0 < d < 1/2, \quad k \to \infty \\ \psi_i &\sim \delta i^{d-1} & \delta > 0, \quad i \to \infty. \end{aligned}$$

(i) *If $1/4 < d < 1/2$ and $\varepsilon \sim N(0, \sigma^2)$ holds, let $C_k = n^{1-2d}(\hat{\gamma}_k - \gamma_k)$; then as $n \to \infty$, $C_k - C_l \xrightarrow{p} 0$ for $k \neq l$ and the common limiting distribution of the C_k has r-th cumulant*

$$\kappa_r = \lambda^r 2^{r-1} (r-1)! K_r, \tag{1.16}$$

where

$$K_1 = -\frac{2}{2d(1+2d)},$$

and

$$K_r = \int_0^1 \cdots \int_0^1 g(y_1, y_2) \ldots g(y_{r-1}, y_r) g(y_r, y_1) \mathrm{d}x_1 \ldots \mathrm{d}x_r, \tag{1.17}$$

for $r \geq 2$, with

$$\begin{aligned} g(x,y) = \; & |x-y|^{2d-1} - \{x^{2d} + (1-x)^{2d} + y^{2d} + (1-y)^{2d}\}/2d \\ & + 2/\{2d(1+2d)\}. \end{aligned}$$

(ii) *Under the same conditions of* (i), *let* $R_k = n^{1-2d}(\hat{\rho}_k - \rho_k)/(1-\rho_k)$; *then as* $n \to \infty$, $R_k - R_l \xrightarrow{p} 0$ *for* $k \neq l$ *and the common limiting distribution of the* R_k *has* r*th cumulant* $\gamma_0^{-r}\kappa_r$ *with* κ_r *defined by (1.16).*

(iii) *If* $d = 1/4$ *and* $\varepsilon \sim N(0, \sigma^2)$ *holds, let* $C_k = (n/\log n)^{1/2}(\hat{\gamma}_k - \gamma_k)$; *then as* $n \to \infty$, $C_k - C_l \xrightarrow{p} 0$ *for* $k \neq l$ *and the common limiting distribution of the* C_k *is* $\mathrm{N}(0, 4\lambda^2)$.

(iv) *Under the same conditions of* (iii), *let* $R_k = (n/\log n)^{1/2}(\hat{\rho}_k - \rho_k)/(1-\rho_k)$; *then as* $n \to \infty$, $R_k - R_l \xrightarrow{p} 0$ *for* $k \neq l$ *and the common limiting distribution of the* C_k *is* $\mathrm{N}(0, 4(\lambda/\gamma_0)^2)$.

(v) *If* $0 < d < 1/4$ *and* $E(\varepsilon^4) < \infty$ *holds, let* $C_k = n^{1/2}(\hat{\gamma}_k - \gamma_k)$; *then as* $n \to \infty$, *any finite subset of the* C_k *has a limiting distribution that is multivariate Normal with mean zero and covariances given by formula (1.7).*

(vi) *Under the same conditions of* (v), *let* $R_k = n^{1/2}(\hat{\rho}_k - \rho_k)$; *then as* $n \to \infty$, *any finite subset of the* R_k, $k \geq 1$, *has a limiting distribution that is multivariate Normal with mean zero and covariances given by formula (1.8).*

1.4 Estimation methods for long memory processes

There exists a wide literature about the problem of identifying and estimating an ARFIMA(p, d, q) process. The book of Beran (1994) gives a good overview on all these methods.

We will consider three of the most common estimators, the Whittle estimator and two semiparametric estimators, the local Whittle and the method of Geweke-Porter-Hudak (GPH in the following). For sake of completeness we will give a brief introduction to other less common estimators.

In this section we will indicate the true value of the parameter with a zero subscript, e.g. ϑ_0 or d_0, and its estimate with $\hat{\cdot}$, e.g. $\hat{\vartheta}$ or $\hat{d}$. Also, $f(\omega)$ is the spectral density and $I(\omega_j)$ is the periodogram

$$I(\omega_j) = \frac{1}{2\pi}\hat{\gamma}_0 + \frac{1}{\pi}\sum_{k=-(n-1)}^{(n-1)} \hat{\gamma}_k \cos \omega_j k,$$

at the Fourier frequencies $\omega_j = 2\pi j/n$ with $j = 1, \ldots, [(n-1)/2]$. The squared brackets $[\cdot]$ indicate the biggest integer less than or equal the number in it.

1.4.1 The exact Maximum Likelihood Estimator

If ε_t in Equation (1.4) is Gaussian, it is possible to write the joint distribution of $\mathrm{X}_n = (X_1, \ldots, X_t, \ldots, X_n)$ as

$$h(\mathbf{X}_n; \vartheta) = \frac{1}{(2\pi)^{n/2}} |T_n(\vartheta)|^{-1/2} \exp\left\{-\frac{1}{2}\mathbf{Z}' T_n^{-1}(\vartheta)\mathbf{Z}\right\},$$

where $\mathbf{Z} = (\mathbf{X}_n - \hat{\mu}\mathbf{1})$, $\vartheta = (\sigma^2, d, \phi_1, \ldots, \phi_p, \theta_1, \ldots, \theta_q)$ is the vector of parameters of dimension $m = p + q + 2$, $\hat{\mu}$ is a consistent estimate of the mean of the process (e.g., the sample mean $\bar{X}$), $\mathbf{1}$ is a column vector of ones and $T_n(\vartheta)$ is the Toeplitz matrix of generic element j, k

$$T_{n;j,k}(\vartheta) = \int_{-\pi}^{\pi} f(\omega;\vartheta)\exp\{i\omega(r-s)\}d\omega \qquad j,k = 1,2,\ldots,n,$$

with $f(\omega;\vartheta)$ the spectral density of the process X_t where we highlight the dependence on the parameter ϑ.

From the log-likelihood function

$$L_n(X;\vartheta) = \log h(X;\hat{\mu},\vartheta) = \\ -\frac{2}{n}\log 2\pi - \frac{1}{2}\log|T_n(\vartheta)| - \frac{1}{2}\mathbf{Z}'T_n^{-1}\mathbf{Z}$$

it is possible to derive the set of first partial derivatives that has to be solved to find the maximum likelihood estimator (MLE in the following).

Asymptotic normality of the MLE has been studied by Dahlhaus (1988, 1989) and Yajima (1985). The last one for fractional integrated processes ARFIMA$(0, d, 0)$ with Gaussian and non-Gaussian innovations.

The following assumptions are necessary for the results of consistency and asymptotic Normality of the maximum likelihood and the Whittle (introduced in the next section) estimators.

Assumption 1.20 *Let X_t, $t \in \mathbb{Z}$, be a stationary Gaussian sequence with mean μ and spectral density $f(x,\vartheta)$, $\vartheta \in \Theta \subset \mathbb{R}^m$, where μ and ϑ are unknown parameters. Let μ_0 and ϑ_0 be the true parameters of the process where ϑ_0 is in the interior of Θ which is assumed to be compact. If $\vartheta \neq \vartheta'$ the set $\{x : f(x,\vartheta) \neq f(x,\vartheta')\}$ is supposed to have positive Lebesgue measure, otherwise the model is not identifiable.*

Assumption 1.21 *Suppose $g(\vartheta) = \int_{-\pi}^{\pi}\log f(x,\vartheta)$ can be differentiated twice under the integral sign.*

Assumption 1.22 *Suppose $f(x,\vartheta)$ is continuous at all (x,ϑ), $x \neq 0$, $f^{-1}(x,\vartheta)$ is continuous at all (x,ϑ), and*

$$f(x,\vartheta) = O(|x|^{-\alpha(\vartheta)-\delta}) \quad as \quad x \to 0,$$

where $0 < \alpha(\vartheta) < 1$.

Assumption 1.23 *Suppose $\partial/\partial\vartheta_j f^{-1}(x,\vartheta)$, $\partial^2/\partial\vartheta_j\partial\vartheta_k f^{-1}(x,\vartheta)$ and $\partial^3/\partial\vartheta_j\partial\vartheta_k\partial\vartheta_l$ $f^{-1}(x,\vartheta)$ are continuous at all (x,ϑ)*

$$\frac{\partial}{\partial\vartheta_j}f^{-1}(x,\vartheta) = O(|x|^{\alpha(\vartheta)-\delta}) \quad as \quad x \to 0, \quad 1 \leq j \leq p,$$

$$\frac{\partial^2}{\partial\vartheta_j\vartheta_k}f^{-1}(x,\vartheta) = O(|x|^{\alpha(\vartheta)-\delta}) \quad as \quad x \to 0 \quad 1 \le j,k \le p$$

and

$$\frac{\partial^3}{\partial\vartheta_j\vartheta_k\vartheta_l}f^{-1}(x,\vartheta) = O(|x|^{\alpha(\vartheta)-\delta}) \quad as \quad x \to 0 \quad 1 \le j,k,l \le p.$$

Assumption 1.24 *Suppose $\partial/\partial x f(x,\vartheta)$ is continuous at all (x,θ), $x \neq 0$, and*

$$\frac{\partial}{\partial x}f(x,\vartheta) = O(|x|^{-\alpha(\vartheta)-1-\delta}) \quad as \quad x \to 0.$$

Assumption 1.25 *Suppose $\partial^2/\partial x\partial\vartheta_j f(x,\vartheta)$ is continuous at all (x,ϑ), $x \neq 0$, and*

$$\frac{\partial^2}{\partial x\partial\vartheta_j}f^{-1}(x,\theta) = O(|x|^{-\alpha(\vartheta)-1-\delta}) \quad as \quad x \to 0, \quad 1 \le j.$$

Assumption 1.26 *Suppose $\partial^3/\partial^2 x\partial\vartheta_j f(x,\theta)$ is continuous at all (x,ϑ), $x \neq 0$, and*

$$\frac{\partial^3}{\partial^2 x\partial\vartheta_j}f^{-1}(x,\theta) = O(|x|^{-\alpha(\vartheta)-2-\delta}) \quad as \quad x \to 0, \quad 1 \le j.$$

Assumption 1.27 *Suppose $\partial/\partial x f^{-1}(x,\vartheta)$ and $\partial^2/\partial^2 x f^{-1}(x,\vartheta)$ are continuous at all (x,ϑ), $x \neq 0$, and*

$$\left(\frac{\partial}{\partial x}\right)^k f^{-1}(x,\vartheta) = O(|x|^{\alpha(\vartheta)-k-\delta}) \qquad k = 0,1,2.$$

Assumption 1.28 *The above constant can be chosen independently of ϑ (not of δ).*

Assumption 1.29 *Assume α is continuous. Furthermore, there exists a constant C with*

$$|f(x,\vartheta) - f(x,\vartheta')| \le C|\vartheta - \vartheta'|f(x,\vartheta')$$

uniformly in all x and all ϑ,ϑ' with $\alpha(\vartheta) \le \alpha(\vartheta')$ where $|\cdot|$ denotes the Euclidean norm.

Theorem 1.30 *Suppose Assumptions 1.20, 1.22, 1.23 and 1.27-1.29 hold and $\hat{\mu}$ is a consistent estimate of μ_0. Then*

$$\hat{\vartheta} \xrightarrow{p} \vartheta_0.$$

Theorem 1.31 *Suppose Assumptions 1.20, 1.22, 1.23 and 1.27-1.29 hold and $\hat{\mu}$ is an $n^{(1-\alpha(\vartheta_0))/2}$-consistent estimate of μ_0. Then*

$$\sqrt{n}(\hat{\vartheta} - \vartheta_0) \xrightarrow{d} N(0, T_n^{-1}).$$

Even though this estimator is asymptotically Normal, it has the drawback of requiring a very large computational burden for maximizing a system of m equations and inverting the $n \times n$ Toeplitz matrix T_n.

1.4.2 The Whittle estimator

Fox and Taqqu (1986) introduced a maximum likelihood method based on the frequency domain, i.e. the MLE is found by maximizing

$$\frac{1}{\sigma}\exp\left\{-\frac{\mathbf{Z}T_n^{-1}\mathbf{Z}}{2n\sigma^2}\right\}.$$

They followed a suggestion of Whittle (1951), who proposed to use an approximation to invert the Toeplitz matrix $T_n(\vartheta)$. By Parseval's relation it is possible to show that a good approximation for T_n is given by the matrix $A_n(\vartheta)$ of generic element j,k

$$a_{j,k}(\vartheta) = \frac{1}{(2\pi)^2}\int_{-\pi}^{\pi}\exp\{i(j-k)x\}\left[f(x,\vartheta)\right]^{-1}dx.$$

This estimator extends the results of Hannan (1973), who applied Whittle's method to the estimation of the parameters of ARMA models. Fox and Taqqu's result, later generalized by Dahlhaus (1989) to the exact maximum likelihood estimator, is the basis of one of the most used methods for estimating the long (and short, if both are present) memory parameters in Gaussian time series. Giraitis and Surgailis (1990) generalized the result of Fox and Taqqu in order to prove the asymptotic normality of Whittle's estimator relaxing the Gaussianity assumption.

The exact maximum likelihood estimator has the drawback of requiring a large computational burden and it might also cause computational problems when calculating the autocovariances needed to evaluate the likelihood function (Sowell, 1992). These difficulties do not occur when using the Whittle estimator, which has the further advantage of not requiring the estimation of the mean of the series (generally unknown in practice). Besides, under some regularity assumptions (Dahlhaus, 1989; Fox and Taqqu, 1986) fulfilled by ARFIMA(p,d,q) processes, it is possible to prove that the Whittle estimator has the same asymptotic distribution as the exact maximum likelihood estimator and it converges to the true values of the parameters at the usual rate of $n^{-1/2}$. Eventually, for Gaussian processes the Whittle estimator is asymptotically efficient in the sense of Fisher.

If the Whittle approximation to the log-likelihood function is used, the parameter vector $\vartheta = (\sigma^2, d, \phi_1, \ldots, \phi_p, \theta_1, \ldots, \theta_q)$ is estimated by minimizing, with respect to ϑ, the estimated variance of the underlying white noise process:

$$\sigma^2(\vartheta) = \frac{1}{2\pi}\sum_{j=1}^{[(n-1)/2]}\frac{I(\omega_j)}{f(\omega_j,\vartheta)},$$

where $f(\omega_j,\vartheta)$ indicates the spectral density of the ARFIMA process at the Fourier frequency ω_j.

Theorem 1.32 *If $f(x,\vartheta)$ satisfies Assumptions 1.22 and 1.24, then with probability 1*

$$\lim_{N\to\infty}\hat{\vartheta}_N = \vartheta_0 \quad \textit{and} \quad \lim_{N\to\infty}\hat{\sigma}^2_N = \sigma^2_0.$$

Theorem 1.33 *If Assumptions 1.21-1.26 are satisfied, then the random vector* $\sqrt{n}(\bar{\vartheta}_n - \vartheta_0)$ *tends in distribution to a Normal random vector with mean* 0 *and covariance matrix* $4\pi W^{-1}(\vartheta_0)$*, with*

$$w_{jk}(\vartheta) = \int_{-\pi}^{\pi} f(x,\vartheta) \frac{\partial^2}{\partial\vartheta_j \partial\vartheta_k} f^{-1}(x,\vartheta) dx.$$

Example 1.34 *Let* X_t *be an* ARFIMA$(0,d,0)$ *process with* $d \in (-0.5, 0.5)$*, then the Whittle estimate for* d *can be found minimizing the function*

$$\sigma_n^2(d) = \frac{1}{n(2\pi)^2} \sum_{j,k=1}^{n} (X_j - \bar{X})(X_k - \bar{X}) \int_{-\pi}^{\pi} \frac{e^{i(j-k)x}}{f(x;d)} dx,$$

where $f(x,d)$ *is the spectral density of a FI(d) process. Moreover, according to Theorem (1.33)* d *is asymptotically normally distributed with variance*

$$\frac{4\pi}{\int_{-\pi}^{\pi} (\log(2 - 2\cos\omega))^2 d\omega} = \frac{6}{\pi^2} \approx 0.6079.$$

The drawback of this estimator is that it is necessary to assume the parametric form of the spectral density to be known *a priori*, i.e. specify the order of the ARMA polynomials, p and q, and decide if including the long memory behaviour. If the specified spectral density function is not the correct one (as it is often the case) the estimated parameters may be dramatically biased. On the other hand, it can be difficult to detect long memory together with the correct orders p and q of the short memory part. Usually, the short memory is dominated and confounded by the long term behaviour of the series.

1.4.3 The local Whittle estimator

The local Whittle estimator is a semi-parametric estimator of the memory parameter d developed by Robinson (1995a) following a suggestion of Künsch (1987). Robinson demonstrated that the local Whittle estimator is asymptotically more efficient than the GPH (later in this Chapter) in the stationary case, although it is not defined in closed form and numerical optimization methods are needed to calculate it.

The advantage of semi-parametric methods, like the local Whittle and the GPH (introduced in the next section), is that the specification of the model is not really necessary because the only information we need is the behaviour of the spectral density near the origin. Furthermore, the long memory parameter can be estimated separately from the short memory part.

It can be found minimizing the following objective function:

$$R(d) = \log\left[\frac{1}{m}\sum_{j=1}^{m} \omega_j^d I(\omega_j)\right] - d\frac{1}{m}\sum_{j=1}^{m} \log\omega_j^d, \tag{1.18}$$

where m is an integer less than $n/2$.

For the results on consistency and asymptotic normality of the local Whittle we need to state some assumptions.

Assumption 1.35 *As* $\omega \to 0+$,

$$f(\omega) \sim G_0 \omega^{-2d_0},$$

where $G_0 \in (0, \infty)$ *and* $d_0 \in (\Delta_1, \Delta_2)$, *with* $-0.5 < \Delta_1 < \Delta_2 < 0.5$.

Assumption 1.36 *In a neighbourhoood* $(0, \delta)$ *of the origin,* $f(\omega)$ *is differentiable and*

$$\frac{d}{d\omega} \log f(\omega) = O(\omega^{-1}) \quad \text{as } \omega \to 0+ .$$

Assumption 1.37 *We have*

$$x_t - E[x_0] = \sum_{j=-\infty}^{\infty} \alpha_j \varepsilon_{t-j}, \qquad \sum_{j=0}^{\infty} \alpha_j^2 < \infty,$$

where

$$E(\varepsilon_t | \mathcal{F}_{t-1}) = 0, \qquad E(\varepsilon_t^2 | \mathcal{F}_{t-1}) = 1, \ \text{a.s.}, \quad t = 0, \pm 1, \ldots,$$

in which $\mathcal{F}_t$ *is the* σ*-field of events generated by* ε_s, $s \leq t$, *and there exists a random variable* ε *such that* $E\varepsilon^2 < \infty$ *for all* $\eta > 0$ *and some* $K > 0$, $P(|\varepsilon_t| > \eta) \leq KP(|\varepsilon| > \eta)$.

Assumption 1.38 *As* $n \to \infty$,

$$\frac{1}{m} + \frac{m}{n} \to 0,$$

i.e. m *has to go to infinity but at a slower rate than* n.

Assumption 1.39 *For some* $\beta \in (0, 2]$,

$$f(\omega) \sim G_0 \omega^{2d_0}(1 + O(\omega^{\beta})) \qquad \text{as } \omega \to 0+,$$

where $G_0 \in (0, \infty)$ *and* $d_0 \in [\Delta_1, \Delta_2]$.

Assumption 1.40 *In a neighbourhood* $(0, \delta)$ *of the origin,* $\alpha(\omega)$, *the Fourier transform of* X_t, *is differentiable and*

$$\frac{d}{d\omega}\alpha(\omega) = O\left(\frac{|\alpha(\omega)|}{\omega}\right) \qquad \text{as } \omega \to 0+ .$$

Assumption 1.41 *Assumption 1.36 holds and also*

$$E(\varepsilon_t^3 | \mathcal{F}_{t-1}) = \mu_3, \quad \text{a.s.}, \quad E(\varepsilon_t^4) = \mu_4, \quad t = 0, \pm 1, \ldots,$$

for finite constant μ_3 *and* μ_4.

Assumption 1.42 *As* $n \to \infty$,

$$\frac{1}{m} + \frac{m^{1+2\beta}(\log m)^2}{n^{2\beta}} \to \infty.$$

Theorem 1.43 *Let Assumptions 1.35-1.38 hold. Then*

$$\hat{d} \xrightarrow{p} d_0 \qquad \text{as } n \to \infty.$$

Theorem 1.44 *Let Assumptions 1.39-1.42 hold. Then*

$$m^{1/2}(\hat{d} - d_0) \xrightarrow{d} N\left(0, \frac{1}{4}\right) \qquad \text{as } n \to \infty.$$

Notice that the convergence rate depends on $m^{1/2}$, the number of frequencies considered in the estimate. Thus, the local Whittle estimate is much less efficient than parametric estimates, like, for example, the Whittle, when they happen to be based on a correct model, but it is asymptotically more efficient than the GPH estimate.

1.4.4 The Geweke-Porter-Hudak estimator

The GPH was introduced by Geweke and Porter-Hudak (1983) and is one of the best known methods to estimate in a semi-parametric way the fractional parameters d of long range dependence behaviour.

This method was first introduced by Geweke and Porter-Hudak (1983) for the Gaussian case when d belongs to $(-1/2, 0)$ and then it was extended by Robinson (1995b).

Assume that the process X_t, $t = 1, 2, \ldots, n$, is an ARFIMA(p, d, q) model as defined in Equation (1.4), then we can observe that the spectral density of this model is proportional to $(4\sin^2(\omega/2))^{-d}$ near the origin, i.e.

$$f(\omega) \sim c_f(4\sin^2(\omega/2))^{-d}, \tag{1.19}$$

when ω tends to 0. Since the periodogram $I(\omega_j)$ is an asymptotically unbiased estimate of the spectral density, that is

$$\lim_{\omega \to 0} E[I(\omega_j)] = f(\omega_j),$$

it is possible to estimate d applying the least squares method to the equation

$$\log(I(\omega_j)) = \log\{\sigma^2 f_\varepsilon(0)2\pi\} - d\log\{4\sin^2(\omega_j/2)\} + u_j, \tag{1.20}$$

where u_j, $j = 1, 2, \ldots, [n/2]$ are i.i.d. error terms. Robinson (1995b) showed that it is possible to consider $-2\log\omega_j$ instead of $-\log(4\sin^2(\omega_j/2))$ because the results are equivalent at the first-order.

Equation (1.19) is an asymptotic relation that holds only in a neighbourhood of the origin; thus, if we use this relation from all periodogram ordinates ($-\pi < \omega < \pi$), the

estimator of d can be highly biased. Geweke and Porter-Hudak (1983) proposed to consider only the first $\sqrt{n}$ frequencies for the estimate since d is the memory parameter and influences mostly the lower frequencies. The higher frequencies are influenced by the short memory ARMA part.

Under some regularities conditions Hurvich et al. (1998) and Robinson (1995b) have shown the asymptotic Normality of the GPH estimator

$$\sqrt{m}(\hat{d} - d_0) \sim N\left(0, \frac{\pi^2}{24}\right),$$

where m is the number of frequency considered in the estimation and has to respect $m = o(n^{4/5})$ and $\log^2 n = o(m)$.

The main drawback of this estimator is its large standard deviation. Moreover Agiakloglou et al. (1993) showed that it is biased in presence of ARMA parameters near the non-stationary boundary.

An interesting advantage with respect to the Whittle is that both GPH and the local Whittle estimators can be easily applied without bothering about the ARMA part of the process.

1.4.5 The R/S estimator

The R/S estimator was widely used by the famous hydrologist Hurst (1951) in his investigations of the Nile River's flow (see Figure 1.1). He noticed the so-called *Joseph effect*, i.e. long periods of abundant water and long periods of low level. The problem was linked to the building of the ideal reservoir for the water. He could describe the ideal capacity with the *adjusted range* statistic $R(t,k)$ given by

$$\begin{aligned} R(t,k) &= \max_{0\le i\le k}\left[x_{t+i} - x_t - \frac{i}{k}(x_{t+i} - x_t)\right] \\ &\quad - \min_{0\le i\le k}\left[x_{t+i} - x_t - \frac{i}{k}(x_{t+i} - x_t)\right] \end{aligned}$$

that can be standardized by

$$S(t,k) = \sqrt{k^{-1}\sum_{i=t+1}^{t+k}(x_i - \bar{x}_{t,k})^2},$$

where $\bar{x}_{t,k} = k^{-1}\sum_{i=t+1}^{t+k} x_i$. The ratio is called the *rescaled adjusted range* or *R/S-statistic*

$$Q = Q(t,k) = R/S = \frac{R(t,k)}{S(t,k)}.$$

Plotting $\log Q(t,k)$ against k we should obtain a straight line. In the case of uncorrelated data the slope is around 1/2, whereas in the presence of long memory it assumes larger values, between 1/2 and 1. These empirical results are confirmed by two theorems of

Mandelbrot (1975). The first theorem is an asymptotic result for $k^{-\frac{1}{2}}Q(t,k)$ valid for short memory processes, whereas the second shows similar asymptotic properties for $k^{-H}Q(t,k)$ for processes converging to fractional Brownian motion.

Theorem 1.45 *Let X_t be such that X_t^2 is ergodic and $t^{-1/2}\sum_{s=1}^{t} X_s$ converges weakly to Brownian motion as $t \to \infty$. Then as $k \to \infty$,*

$$k^{-\frac{1}{2}}Q \xrightarrow{d} \xi$$

where ξ is a nondegenerate random variable.

Theorem 1.46 *Let X_t be such that X_t^2 is ergodic and $t^{-H}\sum_{s=1}^{t} X_s$ converges weakly to fractional Brownian motion as $t \to \infty$. Then as $k \to \infty$,*

$$k^{-H}Q \xrightarrow{d} \xi$$

where ξ is a nondegenerate random variable.

The R/S statistic has the nice property of being robust against heavy tails if the domain of attraction is a stable distribution with $0 < \alpha < 2$.

Definition 1.47 *A random variable Y is stable if, for every k, there are independent random variables $Y_1, \ldots, Y_k$ with the same law as Y and constants $a_k > 0, b_k$ such that*

$$Y_1 + \cdots + Y_k \stackrel{d}{=} a_k Y + b_k.$$

It can be proved that $A_k = k^{1/\alpha}$ with $0 < \alpha \leq 2$ and the number α is called the index of the stable law (for more details, see Revuz and Yor, 1994).

Theorem 1.48 *Let X_t be i.i.d. random variables with $E(X_t^2) = \infty$ and such that they are in the domain of attraction of stable distributions with index $0 < \alpha < 2$. Then the conclusion of Theorem 1.45 holds.*

Unfortunately there is not an asymptotic theory for Q, which makes it difficult, for example, to build confidence intervals. It is open the question of selecting the cut-off point for k and results may vary a lot for increasing values of it. Lastly, Bhattacharya et al. (1983) showed that this estimator lacks robustness against departures from stationarity: in particular it can give misleading results in the presence of a slowly decaying trend.

1.4.6 Variance-type estimator

This estimator was introduced by Taqqu et al. (1995) and then developed by Giraitis et al. (1999) and Teverovsky and Taqqu (1997),

$$\hat{d}_m = \frac{1}{2}\left(1 - \frac{\log S_m^2}{\log m}\right)$$

with

$$S_m^2 = \left[\frac{n}{m}\right]^{-1} \sum_{k=1}^{[n/m]} \left(X_k^{(m)} - \left[\frac{n}{m}\right]^{-1} \sum_{j=1}^{[n/m]} X_j^{(m)} \right)$$

where $X_k^{(m)}$ is the aggregated series of order m,

$$X_k^{(m)} = \frac{1}{m} \sum_{t=1}^{m} X_{t+(k-1)m}, \qquad k = 1, 2, \ldots .$$

Since the bias of the variance-type estimator is of order no less than $(\log n)^{-1}$, it can be useful only in very long series.

There exists a central limit theorem for the variance-type estimator if $0 < d < 0.25$. For larger values, i.e. $0.25 < d < 0.5$, the convergence rate depends on the unknown parameter and its limiting distribution is not Normal but depends on the Rosenblatt process (see Example 1.6), thus is relatively intractable.

1.4.7 Higuchi's method

Higuchi (1988) modified a method introduced by Burlaga and Klein (1986) where the authors try to calculate the fractal dimension. A curve is said to be fractal if $f(\omega) \sim \omega^{-D}$, where D is the fractal dimension and is related to the Hurst exponent H by the equation $D = H + 1$. In particular, he changed the way of calculating the length of a curve because the previous method by Burlaga and Klein (1986) gave very biased results for values of the parameter H approaching 1. Interestingly, its formula remembers the formula for the absolute variation of a generic real function.

Given a finite time series Y_t, $t = 1, 2, \ldots, n$, consider the series of partial sums $X_t = \sum_{i=1}^{t} Y_t$. For example if Y_t is a fractional Gaussian noise then X_t is fractional Brownian motion.

Let the length of the curve be defined as

$$L_m(k) = \frac{1}{k} \left\{ \left(\sum_{i=1}^{\left[\frac{n-m}{k}\right]} \left| X_{m+ik} - X_{m+(i-1)k} \right| \right) \frac{n-1}{\left[\frac{n-m}{k}\right] k} \right\}$$

where $(n-1)/([(n-m)/k]k)$ is a normalization factor, m and k are two integers indicating the initial time and the interval time, respectively. If X_t is a fractal curve then $\langle L(k) \rangle \sim k^{-H-1}$, with $\langle L(k) \rangle$ the average value over k sets of $L_m(k)$. Plotting $\langle L(k) \rangle$ and k on double logarithmic scale, we obtain a straight line and it is possible to obtain an estimate of H with the least squares method.

This estimator has two main drawbacks. On one hand, there is no result on its asymptotic distribution and properties; on the other, it can be useful only with quite long series because of its bias with small sample sizes.

1.5 Models with seasonal persistence

In this section we give a brief summary of models with seasonal persistencies, and then we describe in more detail Gegenbauer processes with k factors. In recent years many processes, SCALM, ARFISMA and GARMA processes, to model seasonal long memory have been developed together with different estimation methods.

Arteche and Robinson (2000) considered SCALM (Seasonal/Cyclical Asymmetric long memory) processes. This kind of process can model persistent seasonality with different increasing and decreasing rates, given by a peak in the spectrum which is not symmetric. SCALM processes are useful to model economic cycles, where growing periods are slower than periods of recession: this asymmetric feature can be captured by this model very realistically. They proposed to use a generalized version of GPH and local Whittle to estimate the parameters.

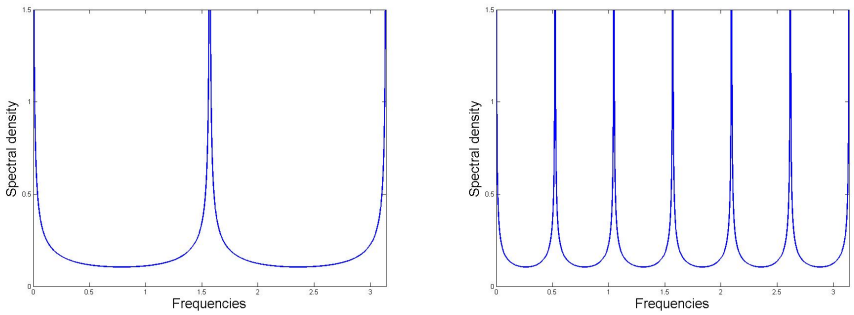
ARFISMA$(p,d,q)\times(P,D,Q)_S$ processes are a generalization of short memory seasonal behaviour given by

$$(1-B)^d(1-B^S)^D X_t = \varepsilon_t, \tag{1.21}$$

where in this instance $p=q=P=Q=0$, and they have been studied in many papers (for the complete model, see Reisen et al., 2006). They are not very flexible because they have singularities in the spectrum at each seasonality as shown in Figure 1.6. Palma and Chan (2005) studied the asymptotic properties of the exact maximum likelihood estimator and used ARFISMA to model internet traffic data. Reisen et al. (2006) proposed several methods to estimate the parameters: the regression method, i.e., a generalization of GPH (Geweke and Porter-Hudak, 1983); an adaptation of the semi-parametric estimator proposed by Arteche and Robinson (2000); and two parametric methods, the penalized (Whittle) and the exact maximum likelihood estimators.

1.5.1 Gegenbauer processes

A more flexible option is given by the k-factor Gegenbauer ARMA class that allows k peaks in the spectrum at any frequencies but not necessarily at regular intervals as in the ARFISMA. The story of GARMA processes starts with Hosking (1981), Woodward et al. (1989) and Giraitis and Leipus (1995). Other papers are Woodward et al. (1998), Smallwood and Beaumont (2004), Sadek and Khotanzad (2004) and Caporale and Gil-Alana (2006). These papers proposed to estimate the parameters of a Gegenbauer process using the maximum likelihood. However there are many problems to consider. Firstly, we will see that not all estimators have the same convergence rate and this can be very problematic. Secondly, maximizing the likelihood function is absolutely burdensome: it is necessary to consider a grid of values in a multidimensional space and it becomes onerous when there are more than two Gegenbauer frequencies. Lastly, the risk of misspecification is high since the short memory part is usually confounded with long memory behaviour and it can be recognized in the residuals after filtering the long memory behaviour. The first paper on these processes was written more than twenty years ago; however the existing



(a) ARFISMA$(0,0,0)(0,0.3,0)_4$ (b) ARFISMA$(0,0,0)(0,0.3,0)_{12}$

(c) ARFISMA$(0,0.4,0)(0,0.3,0)_4$ (d) ARFISMA$(0,0.4,0)(0,0.3,0)_{12}$

Figure 1.6: *Spectral density of a process with representation (1.21).*

literature is not very extensive and model identification and parameter estimation are still open problems.

The form of a k frequency Gegenbauer ARMA process is given by

$$\prod_{j=1}^{k}(1-2\eta_j B+B^2)^{d_j}(X_t-\mu_X)=\varepsilon_t \tag{1.22}$$

where μ_X is the mean of the process, $d=(d_1,d_2,\ldots,d_k)$ are the seasonal long memory parameters, $\eta=(\eta_1,\eta_2,\ldots,\eta_k)$ are the Gegenbauer frequencies and ε_t is an ARMA(p,q) process $\Phi(B)\varepsilon_t=\Theta(B)a_t$, with a_t Gaussian white noise. Each Gegenbauer polynomial has an infinite expansion given by

$$(1-2\eta_i z+z^2)^{-d_i}=\sum_{j=0}^{\infty}C_j^{(d_i)}(\eta_i)z^j,$$

for $|z|\leq 1$ and $|\eta_j|\leq 1$, where

$$C_j^{(d_i)}(\eta_i)=\sum_{k=0}^{[j/2]}\frac{(-1)^k(2\eta_i)^{j-2k}\Gamma(\lambda-k+j)}{k!(j-2k)!\Gamma(\lambda)},$$

and the coefficients $C_j^{(d_i)}$ can be easily computed via the recursion (Chung, 1996)

$$C_j^{(d_i)}(\eta_i)=2\eta_i\left(\frac{d_i-1}{j}+1\right)C_{j-1}^{(d_i)}(\eta_i)-\left(2\frac{d_i-1}{j}+1\right)C_{j-2}^{(d_i)}(\eta_i).$$

For the process to be stationary and invertible it is required that $|d_i|<0.5$ if $|\eta_i|<1$, and $|d_i|<0.25$ if $\eta_i=\pm 1$, apart from the conditions on the ARMA part (see Section 1.2).

The autocovariance function of a one-factor Gegenbauer process is given by

$$\gamma_k=\sigma^2\sum_{j=0}^{\infty}C_j^{(d)}(\eta)C_{j+k}^{(d)}(\eta).$$

Gray et al. (1994) showed that, in this simple case, the autocorrelation function has a slow decaying rate

$$\rho_k\sim k^{2d-1}\cos(2\pi k\omega_0),$$

where $\omega_0=(\cos^{-1}\eta)/2\pi$ is called the G-frequency.

In the case with more than one frequency the autocovariance function is much more complicated and there is not a closed form. However it is possible to compute a good approximation of γ_k by integrating the spectral density

$$\gamma_k=2\int_0^{0.5}f(\omega)\cos(2\pi\omega k)d\omega$$

where

$$f(\omega) = \sigma_a^2 \left| \frac{\Theta(e^{-i\omega})}{\Phi(e^{-i\omega})} \right| \prod_{j=1}^{k} [4(\cos\omega - \eta_j)^2]^{-d_j}, \tag{1.23}$$

where σ_a^2 is the variance of the white noise a_t.

There are few papers on estimating techniques and model identification for GARMA processes. Woodward et al. (1998) proposed an algorithm to maximize the log-likelihood function

$$\log L(X; \mu, \alpha) = \frac{1}{2} \log |\Sigma(\alpha)| - \frac{1}{2}(X - \mu\mathbf{1})^T \Sigma(\alpha)^{-1}(X - \mu\mathbf{1}), \tag{1.24}$$

where $\alpha = (\phi_1, \ldots, \phi_p, \theta_1, \ldots, \theta_q, d_1, \eta_1, \ldots, d_k, \eta_k, \sigma^2)$ is the vector of the parameters and $\Sigma(\alpha)$ is the covariance matrix. The algorithm is as follows

1. Determine the values of $\{d_1, \eta_1, \ldots, d_k, \eta_k\}$ to be considered in a grid search by positions and magnitudes of the peaks in the spectrum.

2. Backcast the time points $-1, -2, \ldots, -M+1$ using a high order AR model, where M is a sufficiently large number.

3. For each combination $\{d_1, \eta_1, \ldots, d_k, \eta_k\}$, carry out the transformation

$$W_t = \prod_{j=1}^{k} \left(\sum_{l=0}^{t+M-1} C_l^{(d_j)} B^l \right) X_t,$$

   to obtain the ARIMA process W_t.

4. Calculate the ARIMA-based likelihood value for W_t.

5. The combination $\{d_1, \eta_1, \ldots, d_k, \eta_k\}$ which is associated with the largest likelihood value is the approximate maximum likelihood estimator.

6. Calculate the AIC (Akaike, 1974) for W_t based on the obtained approximate maximum likelihood estimates.

7. To identify p and q the first five steps can be repeated with different values of p, q. The one associated with the minimum AIC value is selected as the model.

This algorithm uses three approximations: at step one estimating past values through a high order AR procees, at step three by truncating the sum to estimate the process W_t and at step six by using the likelihood function for W_t instead of X_t. Sadek and Khotanzad (2004) proposed a similar procedures without step 1.

Smallwood and Beaumont (2004) estimated the parameter by maximizing the conditional sum of squares (CSS)

$$\mathcal{L}(\alpha) = -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log(\sigma^2) - \frac{1}{2\sigma^2}\sum_{t=1}^{n}\varepsilon_t^2. \tag{1.25}$$

They highlight that this method has some nice properties compared to the other two:

- it can be easily extended with non-normal or GARCH residuals, under the assumption that the sequence ε_t is a martingale difference[4];
- they proved asymptotic normality for the estimates.

One of the main problems in estimating GARMA processes derives from the different convergence rates of the estimates. Even though $\hat{\delta} = (\hat{d}_1, \ldots, \hat{d}_k, \hat{\phi}', \hat{\theta}')$ and $\hat{\eta}^* = (\hat{\eta}_1, \ldots, \hat{\eta}_k)$ have independent asymptotic distributions, the elements of $\hat{\delta}$ are $O_p(n^{-1/2})$, whereas $\hat{\eta}_i$ is $O_p(n^{-1})$ if $|\hat{\eta}_i| < 1$ and $O_p(n^{-2})$ if $|\hat{\eta}_i| = 1$. The grid search might not produce consistent and efficient estimators. Smallwood and Beaumont (2004) proposed a grid search over η^* combined with a non-specified gradient method over δ. fine grid for η, good starting values.

All algorithms to maximize likelihood functions use a grid of values for d_j, η_j, $j = 1, \ldots, k$, however for increasing values of k the computational burden becomes enormous.

It is well known that parametric estimation methods are more efficient than non-parametric procedures; however in the case of GARMA processes it is almost impossible to identify the process *a priori*. As a first step it is better to use non-parametric methods to identify long run and short run behaviours separately: thus the non-parametric procedures are exploratory and in a second step some parametric procedure can be used to compute more efficient estimates. We propose an identification algorithm in a later chapter of this thesis.

[4]The sequence Y_t is a martingale difference if

$$E[Y_t | Z_0, \ldots, Z_{t-1}] = 0,$$

where Z_t is a martingale (Hall and Heyde, 1980).

Chapter 2

Bootstrap methods

Bootstrap methods were first introduced by Efron (1979) and they have developed quickly since then. They owe their popularity to the ease of use together with the advent of powerful calculators and nowadays they are useful in a wide range of problems, confidence intervals, hypothesis testing and distribution estimation are only some examples. Originally, they were born to easily compute measures of accuracy for a statistic of interest because we usually can only compute explicitly them for some quantities, such as the standard deviation for the sample mean. For a review of the bootstrap methodology, see Hinkley (1988). Monographs on the topic include Efron and Tibshirani (1993), Davison and Hinkley (1997), Shao and Tu (1995) and Lahiri (1992).

Bootstrap techniques for a sample $y_1, \ldots, y_n$ of i.i.d. data are very simple. Suppose that we are interested in a parameter θ such that $\hat{\theta} = T(y_1, \ldots, y_n)$, where $T(\cdot)$ is a statistic. The i.i.d. bootstrap resamples with replacement from $y_1, \ldots, y_n$ to obtain the bootstrap sample $y^*_{1,b}, \ldots, y^*_{n,b}$, $b = 1, \ldots, B$. By virtue of the *plug-in principle*, we can compute the bootstrap estimate $\hat{\theta}^*_b = T(y^*_{1,b}, \ldots, y^*_{n,b})$. The set of $\hat{\theta}^*_b$ is called the bootstrap distribution of $\hat{\theta}$ and can be used for the purposes mentioned above.

Unfortunately, things are not so easy with non-i.i.d. data and this technique destroys completely the dependence structure of any autocorrelated sequence of data. Other ways of resampling must be found to reproduce the dependence structure and to obtain a new series with dependence characteristics similar to the observed one. Li and Maddala (1996) discussed the difficulties found in the use of bootstrap for time series models, and gave some guidelines. Time series violate the i.i.d. assumption thus the observations are not exchangeable. They also highlight that the bootstrap leads to an improvement only with asymptotically pivotal statistics. More recently, Bühlmann (2002) reviewed and compared some bootstrap methods for time series illuminating some theoretical aspects of the procedure as well as their performance on finite-sample data. Politis (2003) showed that bootstrap methods allows to estimate consistently the variance of the sample autocorrelation function $\hat{\rho}_k$. The only explicit estimates available are Bartlett's formulas (Equations (1.7)(1.10)) but they are valid only for linear processes. For non-linear processes the variance of $\hat{\rho}_k$ is intractable involving infinite sums of fourth-order cumulants. Under stationarity assumptions, bootstrap leads to higher-order accuracy when estimating the

distribution of the sample mean.

Bootstrap methods for time series can be categorized into *time domain* and *frequency domain* methods. In the former group we find, for example, the parametric bootstrap, the block bootstrap (Künsch, 1989), the SIEVE bootstrap (Kreiss, 1992), the autoregressive-aided periodogram bootstrap (Kreiss and Paparoditis, 2003) and a local bootstrap in the time domain (Paparoditis and Politis, 2002). Examples of methods belonging to the latter group are the phase scrambling (Theiler et al., 1992), a local bootstrap in the frequency domain (Paparoditis and Politis, 1999) and the kernel bootstrap (Dahlhaus and Janas, 1996; Franke and Härdle, 1992).

In Sections 2.1-2.6 of this chapter we will give a brief introduction to the main methods developed: the *block bootstrap*, the *model-based resampling*, the *phase scrambling*, the SIEVE *bootstrap*, the *local bootstrap* and the *log-periodogram regression*. In spite of the great number of papers on bootstrap techniques for time series, the problem of replicating time series is still open since these techniques are not always satisfactory especially if the time series exhibits long range dependence.

Section 2.7 gives an overview of the use of bootstrap in building confidence intervals that usually provides better results and in some cases it also has a faster convergence rate. There is a wide literature on the topic: examples are the works of Efron (1982, 1987a,b,c); Hall (1988, 1992b); Li and Maddala (1996).

2.1 Resampling blocks

The block resampling or moving block bootstrap (MBB) is a generalisation of the resampling scheme for i.i.d. observations. The time series is divided in blocks $\mathcal{B}_1, \ldots, \mathcal{B}_k$ for some integer k, where the choice of the blocks will be discussed in the next paragraphs. The pseudo series is obtained by resampling the blocks with replacement $\mathcal{B}_1^*, \ldots, \mathcal{B}_{k'}^*$, for some integer k' such that the total length is n (or close to it). Distant observations are independent thus blocks can be treated as independent observations, as in the i.i.d. case, but the dependence structure within each block is preserved. This method can be used with stationary time series and is particularly appropriate when the series exhibits short range behaviour or the dependence structure is non-linear and the linear approximation is very poor (see the next section for more details).

The MBB is characterized by two main features: the choice of block length and whether blocks are overlapping or non-overlapping blocks. Many authors have proposed different procedures for dividing the blocks. The procedures proposed in the literature have different names corresponding to different techniques, however we highlight the common denominator that all methods sample blocks of consecutive observations.

Given an observed stationary series $X_1, \ldots, X_n$, Künsch (1989) developed a MBB with overlapping blocks $\mathcal{B}_1, \ldots, \mathcal{B}_{n-l}$ defined by

$$\mathcal{B}_j = (X_j, \ldots, X_{j+l-1}), \qquad j = 1, \ldots, n-l,$$

where l is the block length. The overlapping blocks have a side effect because the first and last $l-1$ observations are sampled less frequently than the rest. An alternative to

this inconvenience is given in Carlstein (1986), where the author proposed non-overlapping blocks $\mathcal{B}'_1, \ldots, \mathcal{B}'_{n-l}$, with

$$\mathcal{B}'_j = (X_{(j-1)l+1}, \ldots, X_{jl}), \qquad j = 1, \ldots, b,$$

where $lb \leq n$. Figure 2.1 shows the difference between overlapping and non-overlapping blocks.

Assuming that we are interested in describing a population through a parameter θ, the length of the block l affects its bootstrap distribution. Longer blocks reduce the variance of the estimator but increase the bias because there are fewer blocks to sample. Thus, choosing the block's length is a trade-off between variance and bias. Also the dependence plays an important role and we expect to need longer blocks to capture stronger dependence. Carlstein (1986) proposed to minimise the mean squared error of the statistic θ of interest. However, this is often not possible, especially when nothing is known about θ *a priori*. Künsch (1989) agrees with Carlstein that the choice of l is a very delicate issue and highlights the importance that the length $l \to \infty$ as $n \to \infty$ but at a slower rate, $l/n \to 0$.

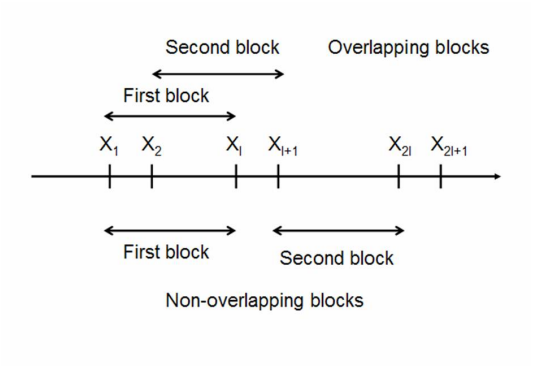


Figure 2.1: *Overlapping and non-overlapping blocks in the MBB.*

The MBB produces pseudo series that might not be stationary and exhibit artefacts where resampled blocks are linked together (Bühlmann, 2002). Moreover, the bootstrap estimate of the mean is biased, i.e. $E[\bar{X}^*|\{X_t\}] \neq \bar{X}$, where $\{X_t\}$ is the observed series. Politis and Romano (1994) described the stationary bootstrap where the length of each block is geometrically distributed with parameter p, thus the block's average length is $1/p$. To choose all the observations with the same probability they consider the data to be in a circle, i.e. X_1 follows X_n. This expedient avoids both the drawback of non-stationary pseudo series and the bias of mean and variance bootstrap estimates. From simulation results it appears that the choice of p is less problematic then the choice of l, in the sense that a wrong value has a smaller deleterious impact in the final result.

MBB is a very general method and is very easy to implement, however is more sensitive to the block length than the SIEVE bootstrap (refer to Section 2.4) to the model selection.

2.2 Parametric bootstrap

If we know that the data have been generated from a linear process, e.g. an ARMA(p, q), it is possible to estimate the residuals of the process. Under the assumption that the model is correctly specified, the residuals are independent and identically distributed: this is a perfect environment in which to apply i.i.d. bootstrap. The pseudo series is built recursively using the bootstrapped residuals and the estimated parameters. The main steps of the parametric bootstrap are as follows.

(a) Fit an appropriate model to the data, e.g. an AR(p), choosing the best model according to an appropriate criterion (AIC, BIC, etc.);

(b) estimate the residuals based on the model;

(c) bootstrap the residuals with i.i.d. bootstrap;

(d) use the estimates and the bootstrapped residuals to build up iteratively the pseudo-series.

Even though the procedure looks quite easy there are many different ways to perform it and special care is needed if data are non-stationary. We will give some details on the parametric bootstrap for stationary autoregressive and autoregressive moving average stationary processes. However it is possible to use a parametric bootstrap for unstable and explosive autoregressive processes (more details can be found in Lahiri, 1992).

The literature is full of works on the model based bootstrap. Swanepoel and van Wyk (1986) used bootstrap methods to build confidence bands for the spectral density. An interesting work is that of Bose (1988). He showed that the bootstrap distribution of the autoregressive parameters is more efficient than the normal approximation. Datta and Sriram (1997) approximated the least squares distribution of the parameter ϕ_1 of an AR(1) that is valid for any value of ϕ_1 even though bootstrap procedures usually fails for unstable processes, i.e. when $\phi_1 = \pm 1$.

The general AR(p) model is of the form

$$\Phi(B)(X_t - \mu) = \varepsilon_t$$

where $\Phi(B)$ is defined as in Formula (1.4), ε_t is a sequence of i.i.d. random variables with zero mean and variance σ^2. Without loss of generality we can assume that $\mu = 0$ and $\sigma^2 = 1$. Under these assumptions, the residuals can be estimated by

$$\hat{\varepsilon}_t = X_t - \hat{\phi}_1 X_{t-1} - \cdots - \hat{\phi}_p X_{t-p}, \qquad t = p+1, \cdots, n,$$

where $\hat{\phi}_i$ are consistent estimators, e.g. based on least squares. Lahiri (1992) suggests to centre the residuals

$$\tilde{\varepsilon}_t = \hat{\varepsilon}_t - \bar{\varepsilon}_t, \qquad t = p+1, \cdots, n,$$

where $\bar{\varepsilon}_t = \sum_{t=p+1}^{n} \hat{\varepsilon}_t/(n-p)$. Assuming that the centred residuals are i.i.d. it is possible to generate the bootstrap innovations ε_t^* by randomly sampling with replacement from $\tilde{\varepsilon}_{p+1}, \cdots, \tilde{\varepsilon}_n$. The bootstrap series is obtained recursively:

$$X_t^* = \hat{\phi}_1 X_{t-1}^* + \cdots + \hat{\phi}_p X_{t-p}^* + \varepsilon_t^*,$$

where it is common sense to consider a burning-in If the residuals are Normally distributed it is possible to generate the bootstrap errors from a Normal distribution with zero mean and variance estimated by the estimated residuals $\hat{\sigma}^2 = \hat{\text{var}}(\tilde{\varepsilon}_t)$.

Bootstrapping an ARMA process $\Phi(B)X_t = \Theta(B)\varepsilon_t$ is very similar to bootstrapping a pure autoregressive process. The formula to estimate the residuals becomes

$$\hat{\varepsilon}_t = \sum_{j=1}^{t} \tilde{\theta}_{j-1} \left(- \sum_{k=0}^{p} \hat{\phi}_k X_{t+1-j-k} \right)$$

where $\sum_{j=0}^{\infty} \tilde{\theta}_j z_j = (\hat{\theta}(z))^{-1}$, $\hat{\phi}_k$, $k = 1, \ldots, p$, and $\hat{\theta}_j$, $j = 1, \ldots, q$, are consistent estimators of the autoregressive and moving average parameters and $\hat{\phi}_0 = -1$.

This method is appropriate for a narrow class of processes, i.e. linear processes, and gives very good results when the model is correctly specified, otherwise the resampled series will not have the same properties as the observed process (Davison and Hinkley, 1997).

2.3 Phase scrambling

Theiler et al. (1992) introduced a bootstrap method for *surrogate data* based on the Fourier transform and we will base the discussion in this Section mainly on their results. This method is also known as *phase scrambling* (Davison and Hinkley, 1997) and the *Fourier bootstrap* (Braun and Kulperger, 1997). It has been used to assess non-linearity and non-stationarity of time series under the hypothesis of linearity (see also, Barnett, 2002; Theiler and Prichard, 1996). Under the null hypothesis of linearity of the process X_t, they generate a pseudo series that, on average, has the same second order structure of X_t, i.e., same spectrum and autocovariance function.

The discrete Fourier transform of a real times series is composed of a real part and an imaginary part,

$$H(\omega_j) = \sum_{t=1}^{n} X_t \exp\{-i\omega_j t\} = \sum_{t=1}^{n} X_t (\cos \omega_j t - i \sin \omega_j t).$$

The second order structure of the time series is completely captured by the periodogram $I(\omega_k) = |H(\omega_k)|^2 / n$ or equivalently by the modulus $|H(\omega_k)|$. The rest of the information carried by the series is in the phase:

$$\psi_j = \tan^{-1} \frac{\mathcal{I}H(\psi_j)}{\mathcal{R}H(\psi_j)} = \tan^{-1} \frac{\sum_{t=1}^{n} X_t \sin \omega_j t}{\sum_{t=1}^{n} X_t \cos \omega_j t},$$

where $\mathcal{I}$ is the imaginary part and $\mathcal{R}$ is the real part of the discrete Fourier transform. It is possible to show that the phases are independent and uniformly distributed in the interval $(0, 2\pi)$ under the assumption that the data come from a linear Gaussian process. This is the perfect environment to apply an i.i.d. bootstrap.

The *Unwindowed Fourier* transform (FT) algorithm was proposed by Theiler et al. (1992):

1. compute the discrete Fourier transform of X_t;
2. randomise the phase: $\psi_j^* \sim \mathcal{U}(0, 2\pi)$;
3. symmetrize the phase, $\psi(f) = -\psi(-f)$;
4. multiply each complex amplitude by $e^{i\psi}$;
5. apply the inverse Fourier transform to obtain the surrogate series X_t

$$X_t^* = \mathcal{R}\left(\frac{\sqrt{2}}{n}\sum_{j=1}^{n} I(\omega_k)\exp\{i\psi_j^* - 2\pi ijt/n\}\right).$$

The authors noted that this algorithm produces spurious low and high frequency effects. One solution is to multiply the series X_t by a function $w(t) = \sin(\pi t/n)$, which eliminates jump discontinuities by vanishing at the endpoints, $t = 0$ and $t = n$ (*Windowed Fourier* transform algorithm, WFT). However, this algorithm also introduces a spurious low frequency from the power spectrum of $w(t)$ itself, that has a peak at the zero frequency. The *Amplitude Adjusted Fourier* transform (AAFT) algorithm is suitable especially if the observed series X_t is a monotonic non-linear transformation of a linear Gaussian process, i.e. $X_t = g(Z_t)$ with $g(\cdot)$ a non-linear monotonic function and Z_t linear Gaussian process:

1. generate n random values Y_i from a standard Normal distribution;
2. order these values so that Y_t has the same rank as X_t;
3. apply the FT or WFT algorithm to Y_t, obtaining Y_t^*;
4. reorder X_t so that X_t^* has the same rank as Y_t^*.

Departure from Gaussianity can, in general, affect bootstrap methods, thus Davison and Hinkley (1997) proposed the *rescaled surrogate algorithm* for data with very asymmetric marginal distributions. As in the AAFT, they apply the FT to a more symmetric transformation of the data:

1. let $Y_t = \Phi^{-1}(r_t/(n+1))$ where r_t is the rank of X_t in the original series;
2. apply the standard algorithm to Y_t and obtain $Y_1^*, \ldots, Y_n^*$;
3. $X_t^* = X_{r_t'}$ where r_t' is the rank of Y_t^*.

A very interesting paper is that of Nur et al. (2001) where the authors applied the two different algorithms to assess the convergence of Markov Chain Monte Carlo algorithms. They also performed a wide Monte Carlo experiment and showed that the rescaled surrogate algorithm performs quite well with highly non-linear series even though the bootstrap series are more symmetric than the original one.

Braun and Kulperger (1997) showed the validity of the method for Gaussian sequences without the need of selecting a model for the data. Their contribution is useful for estimating variances and covariances but it does not work for the mean of the process since the series has to be centred before applying the algorithm. The most interesting result is a central limit theorem for the series itself. The surrogate version Y_t^* of the centred observed series $Y_t = X_t - \bar{X}$ converges to a Normal distribution with zero mean and same variance γ_0 as X_t. Moreover, a subset of observations $Y_{t_1}^*, \ldots, Y_{t_k}^*$, for any integer k, is asymptotically distributed as a $k-$variate Normal distribution with mean $\mathbf{0}$ and covariance matrix

$$\Sigma = \begin{bmatrix} \gamma_0 & \gamma_{12} & \cdots & \gamma_{1k} \\ \gamma_{12} & \gamma_0 & \cdots & \gamma_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{1k} & \gamma_{2k} & \cdots & \gamma_0 \end{bmatrix},$$

where

$$\gamma_0 = \lim_{n\to\infty} \frac{1}{n} \sum_{t=1}^{n} X_t^2$$

and

$$\gamma_i j = \lim_{n\to\infty} \frac{1}{n} \sum_{t=1}^{n} X_t X_{t+i-j}$$

Phase scrambling is an appealing bootstrap method with interesting applications, e.g., finding non-linearities in the data or testing for convergence of Monte Carlo Markov Chain. This bootstrap method has never been used to replicate long memory, however there is no evidence against or in favour.

2.4 Sieve bootstrap

All invertible linear processes have an AR(∞) representation,

$$\sum_{j=0}^{\infty} \pi_j X_{t-j} = \varepsilon_t, \quad \text{with } \pi_0 = 1.$$

Kreiss (1992) used this property to build a bootstrap method that can replicate linear processes. The idea is approximating the observed process X_t with an AR process and, based on this model, apply a procedure similar to the parametric bootstrap introduced in Section

(2.2). In a later paper, Bühlmann (1997) wrote that, by viewing such autoregressive approximations as a sieve for the underlying infinite-order process, the bootstrap procedure is model-free and may still be regarded as non-parametric. The SIEVE bootstrap is computationally simple and yields a (conditionally) stationary bootstrap sample that does not exhibit artefacts in the dependence structure. The variance of the mean exhibits a higher convergence rate if the dependence between distant observations decreases sufficiently fast. The method covers linear processes with representation given by

$$X_t - \mu_{X_t} = \sum_{j=0}^{\infty} \psi_j \varepsilon_{t-j}, \quad \psi_0 = 1,\, t \in \mathbb{Z},$$

where $\{\psi_j\}_{j=0}^{\infty}$ decays exponentially, $\sum_{j=0}^{\infty} \psi_j^2 < \infty$ and $\{\varepsilon_t\}_{t\in\mathbb{Z}}$ is an i.i.d. sequence with $E[\varepsilon_t] = 0$.

Given the sample $X_1, X_2, \ldots, X_n$, the algorithm follows some easy steps, as follows.

1. Fit an AR process, using the Yule-Walker equations as suggested by Bühlmann (1997) (for more detail, see Brockwell and Davis, 1991). The order of the model $p(n)$ must increase at the same rate with the sample size, $p(n) = o(n)$. The order p can be chosen in two different ways: the easiest method is selecting p on the basis of the AIC criterion. Otherwise, it is also possible to choose p such that the spectral density estimate of the residuals is constant, i.e. the residuals are white noise. Testing for white noise does not distinguish between independent and uncorrelated innovations.

2. Estimate the residuals of the best AR fitted model:

$$\hat{\varepsilon}_{t,n} = \sum_{j=0}^{p(n)} \hat{\phi}_{j,n}(X_{t-j} - \bar{X}), \quad \hat{\phi}_{0,n} = 1 \quad (t = p+1, \ldots, n),$$

   where $\bar{X}$ is the sample mean and $\hat{\phi}_{j,n}$ are the estimates of the autoregressive coefficients.

3. Centre the residuals

$$\tilde{\varepsilon}_{t,n} = \hat{\varepsilon}_{t,n} - \sum_{t=1}^{n} \frac{\hat{\varepsilon}_{t,n}}{n - p(n)}.$$

4. Resample the residuals with i.i.d. bootstrap $\hat{\varepsilon}_1^*, \ldots, \hat{\varepsilon}_n^*$.

5. At last, each bootstrap replicate can be calculated using the recursion:

$$\sum_{j=0}^{p(n)} \hat{\phi}_{j,n}(X_{t-j}^* - \bar{X}) = \hat{\varepsilon}_t^*$$

   where $\hat{\varepsilon}_t^*$ are the bootstrapped residuals.

Bickel and Bühlmann (1999) proposed to use a kernel density estimate for the distribution of the residuals

$$\hat{f}_{\hat{\varepsilon}} = (n-p)^{-1} \sum_{t=p+1}^{n} K\left(\frac{x - \hat{\epsilon}_{t,n}}{h}\right)$$

where $h = h(n)$ is a bandwidth with $h = h(n) \to 0$, $h(n)^{-1} = o(n)$ as $n \to \infty$. The bootstrapped residuals are resampled from

$$\varepsilon_t^* \quad \text{i.i.d.} \sim \hat{f}_{\hat{\varepsilon}}(x + \hat{\mu}_\varepsilon)$$

where $\hat{\mu}_\varepsilon = \int_{-\infty}^{\infty} x \hat{f}_{\hat{\varepsilon}}(x) \mathrm{d}x$.

In a later paper, Bühlmann (1998) applied the method in the presence of a trend, i.e. $X_t = s(t) + Y_t$. Assuming that the trend is deterministic, he estimated $s(t)$ using both least squares in a parametric model or with a non-parametric smoother. He applies the SIEVE bootstrap to $\hat{Y}_t = X_t - \hat{s}(t)$ obtaining $X_t^* = Y_t^* + \hat{s}(t)$.

In a very recent working paper Kapetanios and Psaradakis (2006) studied the properties of the SIEVE bootstrap for a class of linear processes with long range dependence. The authors established the first order asymptotic validity of the SIEVE bootstrap in the case of the sample mean and sample autocovariances. Their Monte Carlo experiment is not complete, as they considered small values of $d = 0.1, 0.2$ and they did not show any results for the autocorrelation function. The bootstrap estimates for the memory parameter d are more biased and with a larger variance than the Monte Carlo estimates. Recently, Franco and Reisen (2007) have applied the SIEVE method to bootstrap long memory processes and Poskitt (2007) has investigated the consequences of applying the SIEVE bootstrap under regularity conditions that are sufficiently general to encompass both fractionally integrated and non-invertible processes.

2.5 Local bootstrap

Paparoditis and Politis (1999) have proposed the non-parametric local bootstrap for weakly dependent stationary processes. It produces bootstrap versions of the periodogram $I(\omega_j)$ of the observed process X_t, so that it is useful when the aim is to make inference through the spectrum, e.g. confidence intervals for the memory parameter d in the case of long memory. The local bootstrap is consistent for ratio statistics, kernel estimators of the spectral density, Whittle estimators and parametric fits in the frequency domain.

For stationary time series with a bounded spectral density (see, Beran, 1994, page 77), it is well known that, given m distinct frequencies $0 < \omega_1 < \ldots < \omega_m < \pi$, the periodogram ordinates $f(\omega_1), \ldots, f(\omega_m)$ computed at these frequencies are asymptotically independent and exponentially distributed. On the other hand, if the spectral density is a smooth function of ω, the behaviour of the ordinates at any frequency ω_j is very similar to the frequencies in a small neighbourhood. The idea underlying the algorithm is very simple: the local bootstrap samples locally the frequencies of the periodogram. The neighbourhood of ω_j has to be such that it gets narrower and at the same time includes

an increasing number of frequencies as the sample size goes to infinity.

Given the time series $X_1, \cdots, X_n$, the local bootstrap algorithm that generates bootstrap replicates $I^*(\omega_j)$, $j = 0, 1, \cdots, [n/2]$, of the periodogram can then be described as follows.

1. Select a resampling width k_n where $k_n = k(n) \in \mathbb{N}$ and $k_n \leq [n/2]$. We want that $k_n \to \infty$ and $k_n = o(n)$ as $n \to \infty$.

2. Define i.i.d. discrete random variables $J_1, J_2, \cdots, J_{[n/2]}$ taking values in the set $\{-k_n, -k_n+1, \cdots, k_n\}$ with probability $p_{k_n,s}$, i.e., $P(J_i = s) = p_{k_n,s}$ for $s = 0, \pm 1, \cdots, \pm k_n$. The probabilities p_{k_n} have to be chosen such that $p_{k_n,s} = p_{k_n,-s}$, $\sum_{s=-k_n}^{k_n} p_{k_n,s} = 1$ and $\sum_{s=-k_n}^{k_n} p_{k_n,s}^2 \to 0$ as $k_n \to \infty$.

3. The bootstrap periodogram is defined by $I^*(\omega_j) = I_X(\omega_{J_j+j})$ for $j = 1, 2, \cdots, n/2$, $I^*(\omega_j) = I_X^*(-\omega_j)$ for $\omega_j \leq 0$ and for $\omega_j = 0$ we set $I^*(0) = 0$.

Paparoditis and Politis (1999) have shown the asymptotic consistency in the d_2 metric of the local bootstrap: this implies convergence in distribution and convergence of the first two moments (for more details, see Bickel and Freedman, 1981).

Silva et al. (2006) applied the local bootstrap to the estimation of the long memory parameter d and, by means of simulations, compare its performance with that of other bootstrap approaches. The authors established the efficacy of the local bootstrap in terms of low bias, short confidence intervals and low CPU times.

2.6 Log-periodogram regression bootstrap

This method has been introduced by Arteche and Orbe (2005) to improve the efficiency of the GPH estimator in presence of short memory behaviour. The applicability of the method is specific only to the GPH estimator of the memory parameter.

It assumes the residuals of the regression model given in Equation (1.20) to be independent and identically distributed. The three steps to obtain the bootstrap distribution of $\hat{d}$ are quite straightforward.

1. Calculate the least-squared estimates of $a = \log\{\sigma^2 f_\varepsilon(0)2\pi\}$ and d to estimate the residuals $\hat{u}_j = \log(I(\omega_j)) - \hat{a} + \hat{d}\log(4\sin^2(\omega_j/2))$;

2. resample B bootstrap samples from the residuals $\hat{u}_j$. Using the empirical distribution function of the residuals we obtain the corresponding bootstrap dependent variable $\log(I(\omega_{j,N})) = \hat{a} - \hat{d}\log(4\sin^2(\omega_j/2)) + \hat{u}_j^*$;

3. estimate d from the new models and compute its bootstrap distribution.

Even if the method is very specific, it gave nice results on building confidence intervals for d and we will compare its performance with the new bootstrap introduced in a later chapter of this thesis.

2.7 Bootstrap confidence intervals

One of the typical applications of bootstrap methods is the construction of confidence intervals. There are more reasons to use bootstrap for confidence intervals. When the estimator converges slowly to its asymptotic distribution, even with a large sample size the approximation is very rough. The normal approximation fails, especially when the distribution is very asymmetric. On the other hand, a small sample size gives poor results even when the convergence of the estimator is optimal.

Confidence intervals for the memory parameter are still an open problem and in the recent literature bootstrap methods were often used to solve it. Estimators of the long memory parameter are either very complicated to evaluate or they hardly converge to the normal distribution so that, sometimes, the traditional confidence intervals based on their asymptotic distribution give poor results. The problem is more complicated in the presence of short memory. Agiakloglou et al. (1993) proved that in this case the semi-parametric estimator GPH can be seriously biased and also the coverage of confidence intervals are badly influenced. On the other hand, misspecification is one of the worst drawbacks of parametric estimators based on the likelihood functions. Bootstrap methods help avoiding all these problems (for a review of bootstrap confidences intervals, refer to Efron, 1982; Li and Maddala, 1996).

In the following we introduce briefly the most common confidence intervals for the memory parameter d, describing their principal advantages and drawbacks.

1. *Asymptotic distribution of $\hat{d}$*: this interval is based on the asymptotic distribution of $\hat{d}$ and is symmetric by construction, given by

$$\mathrm{CI}_{se}(1-\alpha) = \hat{d} \pm z_{\alpha/2}\mathrm{se}(\hat{d}),$$

   where z_α is the 100α percentile of the standard normal distribution.

2. *Percentile confidence intervals*:

$$\mathrm{CI}_{pc}(1-\alpha) = \left(\hat{d}^*_{\alpha/2}, \hat{d}^*_{1-\alpha/2}\right),$$

   where d^*_α is the 100α percentile of the bootstrap distribution of $\hat{d}^*$. This interval can be asymmetric but it is equal-tailed.

3. *Percentile-t confidence intervals* (Hall, 1988, 1992a)

$$\mathrm{CI}_{pt}(1-\alpha) = \left(\hat{d} - se(\hat{d})\hat{t}_{1-\alpha/2}, \hat{d} - se(\hat{d})\hat{t}_{\alpha/2}\right),$$

   where t_α is the 100α percentile of $t = (\hat{d}^* - \hat{d})/\hat{se}(\hat{d}^*)$. Percentile-$t$ has been criticized because it produces bad results if the estimate of the variance is poor and because it is not invariant to transformations.

4. *Bias corrected confidence intervals* (BC, for more details, see Efron, 1982),

$$\mathrm{CI}_{BC}(1-\alpha) = \left(\hat{d}^*_{(M+1)(\hat{\alpha}/2)}; \hat{d}^*_{(M+1)(1-\hat{\alpha}/2)}\right)$$

where

$$\frac{\hat{\alpha}}{2} = \Phi(2k_0 + z_{\frac{\alpha}{2}}) \text{ and } 1 - \frac{\hat{\alpha}}{2} = \Phi(2k_0 + z_{1-\frac{\alpha}{2}})$$

with $k_0 = \Phi^{-1}(P(\hat{d}^* \geq \hat{d}))$ the bias-correction parameter.

5. *Accelerated bias corrected confidence intervals* (BC_α, for more details, see Efron, 1987a,b),

$$\mathrm{CI}_{BC_\alpha}(1-\alpha) = \left(\hat{d}^*_{(\tilde{\alpha}/2)}; \hat{d}^*_{(1-\tilde{\alpha}/2)}\right)$$

where

$$\frac{\tilde{\alpha}}{2} = \Phi\left(k_0 + \frac{k_0 + z_{\alpha/2}}{1 - s\left(k_0 + z_{\alpha/2}\right)}\right)$$

and

$$1 - \frac{\tilde{\alpha}}{2} = \Phi\left(k_0 + \frac{k_0 + z_{1-\alpha/2}}{1 - s\left(k_0 + z_{1-\alpha/2}\right)}\right)$$

6. *Bootstrap standard error confidence intervals*

$$\mathrm{CI}_{se^*}(1-\alpha) = \hat{d} \pm z_{\alpha/2}\mathrm{se}^*(\hat{d}^*).$$

The coverage errors for methods 1,2 and 4 is $O(n^{-1/2})$, for methods 3 and 5 $O(n^{-1})$.

Bootstrap methods are an excellent way of approximating the limiting distribution of an estimator. This is the case, for example, when the limiting distribution is unknown. However, bootstrap approximation can give better results under normality assumptions when the sample size is small and the distribution seems to be very asymmetric or the convergence rate is slow.

2.8 Conclusions

In the first two chapters of this thesis we described long memory processes and bootstrap methods for time series. We pointed out that identification and estimation of long memory is still an open problem even though the existing literature is very rich. Parametric estimators are very efficient but they require the specification of the model *a priori* including the short memory part and misspecification is always a danger. This is not an easy task because short and long memory confound each other when they are both present.

Semi-parametric methods can be used to estimate and test for long memory without specifying the short memory behaviour but they have very high standard deviation and their convergence rate is $n^{-1/2+\epsilon}$, for some $\epsilon > 0$ (usually $\epsilon = 1/4$, so that the rate is $n^{-1/4}$), whereas parametric estimates have convergence rate of $n^{-1/2}$.

We described a generalization of long memory processes developed to model seasonal persistences where there is a peak in the spectrum at some unknown frequency. We are interested in Gegenbauer processes with a peak at the zero frequency and one at an unknown frequency. Even more than with ARFIMA processes, the identification of these processes is very difficult. On the other hand, parametric estimators proposed until now imply a computational burden that increases enormously for each new frequency we want to estimate.

We introduced the delicate issue of bootstrapping correlated data. Many methods have been developed in the last twenty years: they usually give satisfactory results with short memory linear and non-linear processes. It is still quite difficult to replicate long memory.

This thesis will to develop a new bootstrap method. It is valid for all linear Gaussian stationary processes: this includes ARMA, seasonal ARMA, ARFIMA and GARMA processes. Thus, it is possible to use it to replicate not only short memory but also long memory and seasonal persistent behaviours. We are interested in particular in applying a bootstrap method to improve the performance of semi-parametric estimators of the memory parameter d.

In the next chapters we will try to answer the following problems.

1. Improve the performance of semi-parametric estimators for the memory parameter d for ARFIMA(p, d, q) and for pure long memory processes FI(d) in terms of smaller standard error, smaller mean squared error and better coverage error for confidence intervals.

2. Give a theoretical support for the consistency of the method in replicating long memory.

3. The conditions to use the bootstrap methods include Gaussianity of the process. This is a very restrictive assumption and we will show, by means of simulation, that the method is consistent even without Gaussianity. In particular the method is robust against *fat tails*, *outliers* and *asymmetry*.

4. Propose an algorithm to estimate non-parametrically the parameters of a Gegenbauer process with one or two peaks in the spectral density; the bootstrap method will be useful to give an estimate of the distribution of the frequency parameter η since its asymptotic distribution is unknown for the estimators we will consider. The main aim is proposing a method to identify seasonal persistences and provide starting values for maximize a (penalized) log-likelihood function.

Chapter 3

ACF bootstrap for long memory processes

In the first two chapters of this thesis, we introduced long memory processes and bootstrap methods for time series. Despite the existence of a wide literature on these topics, at the moment there are no satisfactory bootstrap methods to replicate long memory. On the other hand, short and long memory can confound each other when the parameters describing the short memory behaviour are near the boundary of non-stationarity. Parametric estimators are very efficient when the model is correctly specified but exhibit large biases otherwise, whereas semi-parametric and non-parametric estimators have large standard deviations and slow convergence rates.

In this chapter, we propose a new method based on the empirical autocovariance function and the Durbin-Levinson algorithm that seems to give satisfactory performance when improving the efficiency of semi-parametric estimators of the memory parameter d, especially with Gaussian long memory processes. Even though the method is equivalent to a Cholesky decomposition, its applicability is wider. Especially with long series, the Cholesky decomposition has to handle very large matrices and the most powerful calculators can have problems. On the contrary, the method we introduce is iterative and avoids the problem of large matrices.

The *ACF bootstrap*, as we call it, is based on a result of Ramsey (1974) (see below) and requires Gaussianity of the observed process X_t. This assumption is quite restrictive, however we will show that some deviations from Normality do not affect materially the method.

We apply this new method to improve the efficiency of two semi-parametric estimators of the long memory parameter d: the GPH estimator proposed by Geweke and Porter-Hudak (1983), and the local Whittle estimator proposed by Robinson (1995a). The parametric Whittle estimator (Fox and Taqqu, 1986) is used as benchmark (for more detail on these estimators, see Section 1.4).

We assess the validity of ACF bootstrap in three different scenarios:

- ACF bootstrap improves the performance of semi-parametric estimators of the memory parameter;

- we show, by mean of simulation, that the proposed method is robust against non-Gaussian innovations, asymmetry and fat tails, and
- bootstrap methods are widely used to build confidence intervals with a coverage level closer to the nominal level than confidence intervals obtained by applying asymptotic results.

In the first scenario we perform an extensive Monte Carlo experiment for different values of the memory parameter d. We compare ACF bootstrap performance with the performance of local and SIEVE bootstraps in terms of reduction of standard error and mean squared error of the estimates.

We conduct experiments on processes to test consistency of ACF bootstrap when the observed series is non-Normal, using Chi-squared innovations with one degree of freedom to test against skewness and Student t innovations (with four and six degrees of freedom) to test against fat tails.

In the last part of the chapter we aim to improve the coverage of confidence intervals for the memory parameter d in two different situations. Firstly, we consider the Whittle estimator. Even though the Whittle estimator is asymptotically Normal, if the assumption of correctly specified model is satisfied, then confidence intervals based on short series ($n = 128, 300$) have an actual coverage lower than the nominal coverage level. Secondly, we study the confounding effects when both long and short memory are present in the series. It has already been highlighted by Agiakloglou et al. (1993) that short memory introduces bias in the GPH estimates, and also the coverage of confidence intervals is affected. On this topic, Arteche and Orbe (2005) developed a bootstrap method and improved the performance of the GPH estimator but their solution is designed specifically for this estimator, and therefore does not apply to other estimators currently in use (see Section 2.6 for more detail).

The plan of the chapter is the following. In Section 3.1 we describe the new bootstrap method. Section 3.2 assesses the validity of the method for long memory Gaussian and non-Gaussian processes in an extensive Monte Carlo experiment. The problem of confidence intervals for the memory parameter d is developed in Section 3.3. We conclude and discuss future developments in Section 3.4.

3.1 ACF bootstrap

The new bootstrap method is based on a theorem of Ramsey (1974) that derives the distribution of X_t conditionally on the past values $X_0, \ldots, X_{t-1}$ of the process. Its distribution is Normal with mean and variance given in the theorem if the observed process is Gaussian itself. Without loss of generality we can assume that X_t is zero mean process.

THEOREM 3.1 *Let X_t be a Gaussian, wide-sense stationary*[1] *time series with mean μ and variance γ_0. Then the conditional distribution of X_t given $X_0, \cdots, X_{t-1}$ is Gaussian*

[1]The process X_t is said to be wide-sense stationary if first and second moments are independent of

with mean and variance given by

$$m_t = E(X_t|X_0, \cdots, X_{t-1}) = \sum_{j=1}^{t} \phi_{tj} X_{t-j},$$
$$v_t = Var(X_t|X_0, \cdots, X_{t-1}) = \gamma_0 \prod_{j=1}^{t} (1 - \phi_{jj}), \tag{3.1}$$

where ϕ_{jj} is the jth partial autocorrelation and ϕ_{tj} is the jth autoregressive coefficient in an autoregressive fit of order t.

The coefficients ϕ_{tj} and ϕ_{jj} can easily be obtained through the Durbin-Levinson recursion (see, e.g., Brockwell and Davis, 1991):

$$\phi_{tt} = N_t / D_t \tag{3.2}$$
$$\phi_{tj} = \phi_{t-1,j} - \phi_{tt}\phi_{t-1,t-j}, \quad j = 1, \cdots, n-1, \tag{3.3}$$

where

$$N_0 = 0$$
$$D_0 = 1$$
$$N_t = \rho_t - \sum_{j=1}^{t-1} \phi_{t-1,j}\rho_{t-j}$$
$$D_t = D_{t-1} - N_{t-1}^2 / D_{t-1}.$$

and ρ_t is the autocorrelation function of X_t at lag t.

The hypotheses of Theorem 3.1 admit all processes with an MA-infinite representation, e.g., stationary ARMA processes, ARFIMA processes with $|d| < 0.5$ and Gegenbauer processes with $|d_j| < 0.5$ if $|\eta_j| \neq 1$ and $|d_j| < 0.25$ if $|\eta_j| = 1$. This result has been used widely to generate some particular type of processes. Giraitis et al. (2001) showed the validity of the result for ARFIMA(p, d, q) processes. Woodward et al. (1998) first used it to generate a k-factor GARMA because generating through truncation of the MA-infinite representation does not have good convergence properties, even when considering 290 000 terms.

Instead of using a theoretical autocovariance function, the idea is to use the empirical autocorrelation function of an observed time series X_t to generate bootstrapped copies through the conditional mean and the conditional variance. From now on we will call this new procedure the *ACF bootstrap*. The steps to generate a bootstrap series are:

---

time, i.e. $\forall t = 0, \pm 1, \pm 2, \ldots$

$$E[X_t] = \mu,$$
$$E[(X_t - \mu)(X_{t+k} - \mu)] = \gamma_k.$$

1. compute the empirical autocorrelation function, $\hat{\rho}_k$, from the observed time series X_t;
2. perform the Durbin-Levinson recursion, given in Equations (3.2) and (3.3), for $\hat{\phi}_{tt}$ and $\hat{\phi}_{tj}$ based on the empirical autocorrelation function;
3. calculate v_t based on Equation (3.1) and m_t^* as follows

$$m_t^* = E(X_t^*|X_0^*, \cdots, X_{t-1}^*) = \sum_{j=1}^{t} \hat{\phi}_{tj} X_{t-j}^*,$$

   and thus m_t^* depends on the past values of the bootstrap series and the observed autocorrelation function of the original one; and
4. generate a starting value of X_0^* from an $N(0, v_0)$ distribution where v_0 is the sample variance of X_t;
5. generate the bootstrap replicate of X_t^* from $N(m_t^*, v_t)$; and
6. repeat step 4 and 5 until $t = n$, where n is the series length.

It is interesting to notice that the conditional means m_t^* depend on each bootstrap time series, whereas the conditional variances v_t are determined from the observed process and do not change for each bootstrap replication. The steps 3 to 6 have to be repeated for $b = 1, \ldots, B$, where B is the total number of bootstrap replicates. We omit the b subscript in the following when it is clear that we are referring to bootstrap replicates and the double subscript (e.g. $X_{t,b}^*$) would be redundant.

Under the assumption of Gaussianity, the ACF bootstrap has some nice properties concerning the second order structure, periodogram, autocovariance and autocorrelation functions.

First, we show that the autocovariance function γ_j^* of the bootstrap series is asymptotically unbiased, that is to say,

$$E[\gamma_k^*] = E[X_t^* X_{t-k}^*] = EE[\gamma_k^*|\{X_t\}] = EE[X_t^* X_{t-k}^*|\{X_t\}]$$

$$= E\left[\left(\frac{n-k}{n}\right)\hat{\gamma}_k\right] = \left(\frac{n-k}{n}\right) E[\hat{\gamma}_k] = \left(\frac{n-k}{n}\right)^2 \gamma_k.$$

Because the periodogram is a linear transformation of the autocovariance function, also the bootstrap periodogram is asymptotically unbiased:

$$E\left[I^*(\omega_j)\right] = EE\left[I^*(\omega_j)|\{X_t\}\right]$$

$$= E\left[\sum_{k=-(n-1)}^{n-1} \gamma_k^* \cos \omega k\right] = \sum_{k=-(n-1)}^{n-1} \left(\frac{n-k}{n}\right)^2 \gamma_k \cos \omega k.$$

Last, if we consider each bootstrap series X_t^* as a single observation, we can notice that they are independent and identically distributed conditionally on the observed series X_t. Besides, the i.i.d. property is preserved if we consider any transformation f of the data. This allows us to have an estimate of the distribution of $f(X_t)$ no matter how complicated the function f is, and it will be useful in the next Sections.

3.2 Monte Carlo results

In this section we conduct experiments with simulated data to assess the validity of the ACF bootstrap method with respect to the existing techniques in literature. In particular, we apply the proposed bootstrap method to long memory time series. We use the ACF, the SIEVE and the local bootstraps to replicate the observed series X_t, and GPH and local Whittle estimators to estimate the long memory parameter d.

As we said in the Introduction, we are interested in fractionally integrated processes, and especially in improving the performance of two semi-parametric estimators for the memory parameter d: the GPH and the local Whittle. Both of them estimate the parameter d through the periodogram of the observed series. We use the Whittle estimator as a benchmark since it is a parametric estimator used without the risk of misspecification, given the parametric assumptions in our simulation, thus it is the most efficient.

In the case of the GPH, it is straightforward to write the estimator in the following form, to highlight that this estimator is a weighted average of the first m periodogram ordinates:

$$\hat{d} = \frac{1}{mK} \sum_{j=1}^{m} p_j \log I(\omega_j), \tag{3.4}$$

where usually $m = \lfloor \sqrt{n} \rfloor$, $K = \text{Var}(\sin^2(\omega_j/2))$ and $p_j = \sin^2(\omega_j/2) - E[\sin^2(\omega_i/2)]$, for $j = 1, \ldots, n$. It is possible to use the bootstrap in two ways: estimate the memory parameter for each bootstrap replicate and then take the average value, namely

$$\begin{aligned} {}_1\hat{d}_{\text{GPH}} &= \frac{1}{B} \sum_{b=1}^{B} \hat{d}_{\text{GPH},b} \\ &= \frac{1}{mBK} \sum_{b=1}^{B} \sum_{j=1}^{m} p_j \log I_b^*(\omega_j) \\ &= \frac{1}{mK} \sum_{j=1}^{m} p_j \left(\log \prod_{b=1}^{B} I_b^*(\omega_j) \right)^{1/B}, \end{aligned} \tag{3.5}$$

or plug into Equation (3.4) the average value of the bootstrap periodograms obtaining

$${}_2\hat{d}_{\text{GPH}} = \frac{1}{mK} \sum_{j=1}^{m} p_j \log \frac{1}{B} \sum_{b=1}^{B} I_b^*(\omega_j). \tag{3.6}$$

The only difference between ${}_1\hat{d}_{\text{GPH}}$ and ${}_2\hat{d}_{\text{GPH}}$ is that the former estimate is the geometric mean of the bootstrap periodograms whereas the latter is the arithmetic mean. (This may have implications for their standard errors, and hence their relative efficiency.)

The same can be done when estimating d with the local Whittle, even though it is not possible to write it in a closed form because this estimator has to be found by maximization of the objective function $R(d)$ (see Equation (1.18)).

Even though we have two options when estimating the memory parameter we choose to use only the first (Equation (3.5)), i.e. we consider the average values of the estimates of each bootstrap series $\hat{d}_i^* = \sum_{b=1}^{B} \hat{d}_{b,i}^*/B$ with $i = GPH, lW$ for the following reasons. In some exploratory simulations we noticed that their standard deviations are approximately the same. Moreover, this solution is more informative because it allows to have the bootstrap distribution of the memory parameter that can be used for different purposes, e.g. to build confidence intervals in many different ways.

In the simulation study we generated series by $I(d)$ models for different values of the long memory parameter, $d = 0.1, 0.2, 0.3, 0.4, 0.45$, and increasing sample sizes, $n = 200, 500, 1000$.

The series are generated using the recursive Durbin-Levinson algorithm (see Brockwell and Davis, 1991). For each model we consider $S = 2000$ realizations and $B = 1000$ bootstrap replications. For each estimation method we calculate the Monte Carlo estimate, i.e.,

$$\hat{d} = \frac{1}{S}\sum_{j=1}^{S} \hat{d}_j,$$

where $\hat{d}_j$ is the estimated value for a single realization obtained with one of the estimators (Whittle, GPH, local Whittle) or the average of two thousand bootstrap estimates given by Equation (3.5) (plugging in GPH and local Whittle). To compare the performance of estimators and bootstrap methods, we compute standard errors

$$\hat{\text{se}}(\hat{d}) = \sqrt{\frac{1}{S-1}\sum_{j=1}^{S}(\hat{d}_j - \hat{d})^2}$$

and mean squared errors

$$\hat{\text{MSE}}(\hat{d}) = \text{Var}(\hat{d}) + \text{Bias}(\hat{d})^2 = \frac{1}{S-1}\sum_{j=1}^{S}(\hat{d}_j - \hat{d})^2 + (\hat{d} - d)^2$$

The results are presented in Tables 3.1-3.4 where also the Whittle estimator is included as a benchmark. The tables report results on $\hat{d}$ (in boldface), standard error of $\hat{d}$ (italic font) and MSE of $\hat{d}$ (normal font) for the three estimators treated and for the three bootstrap methods.

The Monte Carlo estimates are in accordance with known results (see, for example, Bisaglia and Guègan, 1998). As we expected, the Whittle estimator largely outperforms

all the others, since it is a parametric estimator in the best conditions, i.e., the estimates are based on the correctly specified parametric model.

Comparing the bootstrap methods, it is evident that the SIEVE bootstrap exhibits the worst performance. With regard to the other two methods, both give satisfactory results compared with the same estimators in the Monte Carlo simulations: the ACF is a slightly more biased but its standard deviation and the mean squared error are always smaller than using the local bootstrap.

Table 3.4 reports the gain, namely

$$\text{GAIN}\% = \frac{\hat{\text{se}}(\hat{d}_i) - \hat{\text{se}}(\hat{d}_i^*)}{\hat{\text{se}}(\hat{d}_i)} \times 100 \tag{3.7}$$

(where $i = \text{GPH}, \text{lW}$) calculated as a percentage, when using the ACF and local bootstraps with respect to the Monte Carlo estimates, for the GPH and local Whittle bootstrap estimator. The results confirm that the gain is always greater for the ACF bootstrap even if it decreases with increasing the series length. It is interesting that for the local bootstrap if d assumes large values, the gain is almost irrelevant for $n = 500$ and negative for $n = 1000$.

In conclusion to this section, the ACF bootstrap is promising for long memory Gaussian processes. It helps to improve the efficiency of some semi-parametric estimators (GPH and local Whittle) that usually have large standard deviations, and outperforms the other two bootstrap methods considered. The method is open to future developments:

- confidence intervals based on bootstrap approximation are largely used;
- developing a test for long memory based on the bootstrap sample distribution of the parameter d instead of using the asymptotic results; and
- it would be interesting to prove the consistency of ACF bootstrap with long memory Gaussian processes from a more theoretical point of view.

3.2.1 Non-Gaussian innovations

The assumption of Gaussianity is very restrictive and it would be interesting to see how much deviations from Normality affect the performance of the proposed bootstrap methods even though Gaussianity is one of the assumptions of Theorem 3.1. To this end we perform some simulations to compare the estimators when the observed process is non-Gaussian. We consider two different deviations from Normality. To test robustness against asymmetry, we generate long memory processes with Chi-squared innovations with one degree of freedom, giving skewness $\gamma_1 = 2\sqrt{2}$. To test robustness against fat tails, we use the Student t distributions with four and six degrees of freedom: the former does not have the fourth moment finite, the latter has excess of kurtosis $\gamma_2 = 3$.

In Tables 3.5-3.7 we report the results. All the estimates, Monte Carlo and bootstrap, are very similar to the results obtained with Gaussian innovations, in terms of both standard error and mean squared error. This suggests that ACF bootstrap can be useful also

relaxing the Gaussianity assumption. There is not the danger of obtaining bad results when there the suspect of non-Gaussian innovations and it is not necessary to correct or exclude extreme values. Also in this case the standard error and the mean squared error are smaller for the bootstrap estimates. These results are very important in view of applying the method to replicate the dependence structure of heteroskedastic data, such as white noise with GARCH effects or stochastic volatility processes.

3.3 Bootstrap confidence intervals for the memory parameter

We dedicate the last section of this chapter to study the performance of the ACF bootstrap to build confidence intervals for the memory parameter (for more details on bootstrap intervals, see Section 2.7). This is an interesting problem directly connected with the problem of testing the hypothesis of existence of long memory. According to the paper of Arteche and Orbe (2005), we build confidence intervals based on the percentile and the percentile-t methods because they turned out to be the best. We add confidence intervals based on Normal approximation with bootstrap standard error (for a review, see Section 2.7). We build confidence intervals in two different scenarios.

Firstly, we analyse the finite sample performance of the Whittle estimator in building confidence intervals for the memory parameter and test whether or not they can be improved by building confidence intervals based on bootstrap replications.

In the second scenario we examine confidence intervals for d based on semi-parametric estimators with the presence of the short memory part. As we already pointed out, it is useful to estimate and study long range behaviour separately from short range memory, because the two behaviours tend to confound each other and it can be difficult to distinguish between them. This is not possible with parametric estimators since we need to specify the whole model *a priori*. On the other hand, Agiakloglou et al. (1993) showed that the GPH estimator is influenced by the short memory part and its bias increases when the parameters of the short memory part approach non-stationarity boundary. We compare the ACF bootstrap with the local bootstrap. We do not use the SIEVE bootstrap since it gave quite poor results in the previous studies (see Section 3.2). In their paper Arteche and Orbe (2005) reduced the coverage error of confidence intervals for the memory parameter built with the GPH estimator. Their method is specifically designed for, and limited to, the GPH estimator. The methodology proposed in this thesis, the ACF bootstrap, has a more general applicability and can be used not only with the GPH estimator but also with other estimators of d, such as the local Whittle and the Whittle estimators.

3.3.1 Finite sample performance of ACF bootstrap

Even if the Whittle estimator is very efficient, in case of correct specification of the model, its performance in estimating long memory for small samples is not very good. Also the nominal level of confidence intervals for the memory parameter d is usually far from the actual level. Especially when detecting long range behaviour, it is necessary to have quite long series. We deem that bootstrap methods can improve the coverage level and

give satisfactory results with finite sample sizes. We compare Monte Carlo results of the Whittle estimator with the results given applying the ACF and local bootstraps.

Following the simulation plan of Arteche and Orbe (2005), we run simulations with $n = 128, 300, 1000$ and $d = 0, 0.2, 0.45, -0.45$. The results are given in Table 3.9. For small values of the parameter, $d = 0, 0.2$, the three confidence intervals based on ACF bootstrap are all better than the Monte Carlo intervals. The best results are given by the percentile method. For larger values of d in absolute value, it appears to be more difficult to have actual coverage close to the nominal, but the intervals obtained with the standard deviation and with percentile of ACF bootstrap outperform the asymptotic results especially for small sample sizes. Overall the best intervals are built with the standard deviation estimated using the ACF bootstrap, in particular in the most common coverage probabilities, 95 and 99%.

The confidence intervals built with the local bootstrap give very poor results. Only the percentile-t method gives reasonable results for $d = 0, 0.2$ but these intervals are very similar to the Monte Carlo and it is not worth using a bootstrap method if it does not lead to any improvement.

3.3.2 The influence of the short memory part

It is known that the semi-parametric estimators, which we introduced above, are biased in the presence of short memory behaviour. This affects also the coverage level of confidence intervals. In this work we aim to improve confidence intervals for the memory parameter when the data generating process is a simple ARFIMA$(1, d, 0)$, given by

$$(1 - B)^d(1 - \phi B)X_t = \varepsilon_t,$$

where $|\phi| < 1$ to assure stationarity. The problem is when ϕ gets close to unity: short memory and long memory confound each other and it is really difficult to distinguish the effects of the two parameters and consequently to build reliable confidence intervals for d. Figure 3.1 shows the spectral densities on the logarithmic scale of three processes: AR(1), ARFIMA$(1, d, 0)$ and ARFIMA$(0, d, 0)$, with $d = 0.4$ and $\phi = 0.8$. Their behaviours near the zero frequency are quite similar and it is logical to expect problems to discriminate if the peak is generated by a strictly positive d or by a value of ϕ approaching unity. Arteche and Orbe (2005) proposed a solution to the problem of the bias introduced by short memory behaviour in the estimation of d, bootstrapping the log-periodogram, however the method can be applied only to the GPH estimator.

Since in the first part of the chapter we showed that the ACF bootstrap can replicate long memory behaviour, it is rational to think that it could also improve the coverage level of confidence intervals.

The Monte Carlo experiments were run for all combinations of $n = 128, 300, 1000$, $d = 0, 0.2, 0.45, -0.45$ and $\phi = 0, 0.2, 0.4, 0.6, 0.8$. The number of bootstrap replicates was $B = 1000$ and each experiment was repeated $S = 2000$ times. The results are given in Tables 3.10-3.17.

The results are summarized in the following list:

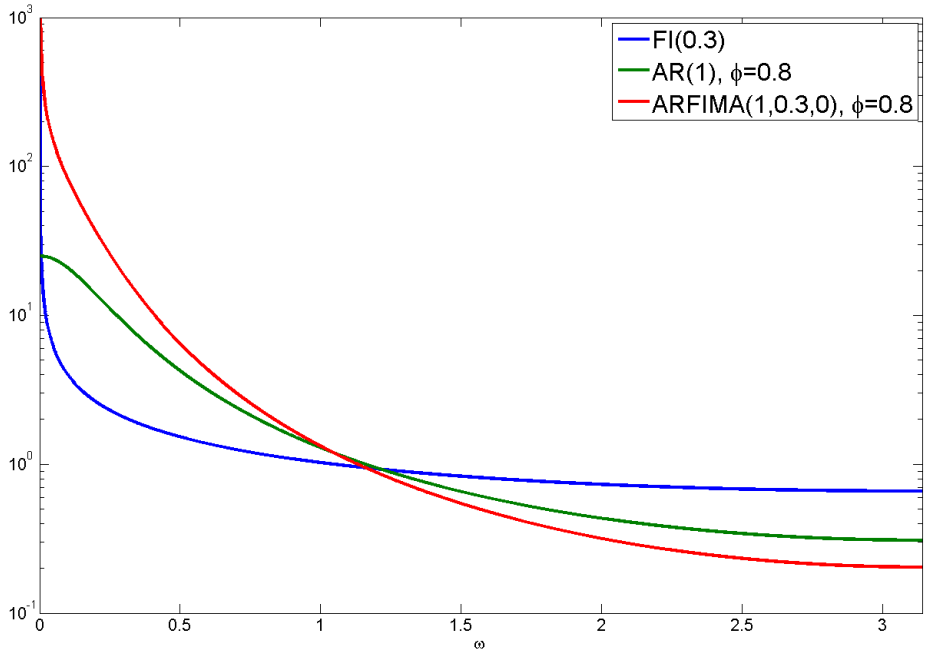


Figure 3.1: *Spectral densities of three different processes in logarithmic scale: FI(0.3) with* $f(\omega) = (2\sin(\omega/2))^{-2d}$ *(blue line),* AR*(1),* $\phi = 0.8$*, with* $f(\omega) = (1+\phi^2 - 2\phi\cos(\omega))^{-1}$ *(green line),* ARFIMA*(1,0.3,0),* $\psi = 0.8$*, with* $f(\omega) = (1+\phi^2 - 2\phi\cos(\omega))^{-1}(2\sin(\omega/2))^{-2d}$ *(red line).*

- the log-periodogram regression is still the best method to build confidence intervals through the GPH for the memory parameter but the ACF bootstrap is not much worse;
- the local bootstrap give results similar to the Monte Carlo, thus it is not useful for this problem; and
- the ACF bootstrap gives satisfactory results applied to the GPH estimator but not as good as the log-periodogram; however it is the best method to build confidence intervals when estimating d with the local Whittle.

3.4 Conclusions

In this chapter we presented a new bootstrap method for time series, ACF bootstrap. The Monte Carlo experiments showed that

- the ACF bootstrap is better than some existing bootstrap methods, local and SIEVE bootstrap. It outperformed both of them in terms of reduction of standard error and mean squared error of the estimates of d;
- the method is robust against deviations from Normality, like asymmetry (Chi-squared distribution with one degree of freedom) and extreme values (Student t distribution with four and six degrees of freedom);
- nominal coverage for confidence intervals for d based on the asymptotic distribution of the Whittle estimator are improved by using ACF bootstrap especially when the sample size is small, such as $n = 128$; also in this case the method we proposed outperformed the local bootstrap; and
- in the presence of short memory the ACF gave some improvements to the local Whittle, whereas the log-periodogram regression by Arteche and Orbe (2005) is still the best solution to build confidence interval with GPH estimator.

In conclusion, we can state that the ACF bootstrap is very promising in the variety of scenarios analysed and it is also open to future development.

This chapter is not exhaustive on the possible applications of ACF bootstrap. Some future developments can be

- building test statistic to detect long memory;
- building test statistic to distinguish long memory from structural break, or regime switching, models.

	Monte Carlo			ACF		Local B.		SIEVE	
d	Whittle	GPH	LW	GPH	LW	GPH	LW	GPH	LW
	0.0817	**0.0961**	**0.1232**	**0.0664**	**0.1337**	**0.0897**	**0.1322**	**0.0253**	**0.0831**
0.1	*0.0553*	*0.2318*	*0.1346*	*0.1714*	*0.1012*	*0.2175*	*0.1165*	*0.0706*	*0.0502*
	0.0034	0.0537	0.0186	0.0305	0.0114	0.0474	0.0146	0.0106	0.0028
	0.1802	**0.2074**	**0.2065**	**0.1672**	**0.1996**	**0.1946**	**0.2052**	**0.0789**	**0.1197**
0.2	*0.0613*	*0.2305*	*0.1620*	*0.1773*	*0.1249*	*0.2162*	*0.1417*	*0.1072*	*0.0842*
	0.0041	0.0532	0.0263	0.0325	0.0156	0.0468	0.0201	0.0262	0.0135
	0.2786	**0.3070**	**0.2916**	**0.2625**	**0.2700**	**0.2853**	**0.2825**	**0.1497**	**0.1742**
0.3	*0.0618*	*0.2270*	*0.1767*	*0.1752*	*0.1401*	*0.2148*	*0.1581*	*0.1382*	*0.1159*
	0.0043	0.0516	0.0313	0.0321	0.0205	0.0464	0.0253	0.0417	0.0293
	0.3817	**0.4107**	**0.3871**	**0.3584**	**0.3533**	**0.3846**	**0.3736**	**0.2481**	**0.2577**
0.4	*0.0629*	*0.2316*	*0.1890*	*0.1820*	*0.1564*	*0.2189*	*0.1736*	*0.1652*	*0.1497*
	0.0043	0.0538	0.0359	0.0348	0.0266	0.0481	0.0308	0.0504	0.0427
	0.4366	**0.4604**	**0.4365**	**0.4073**	**0.3955**	**0.4284**	**0.4181**	**0.3067**	**0.3110**
0.45	*0.0621*	*0.2306*	*0.1875*	*0.1784*	*0.1578*	*0.2214*	*0.1773*	*0.1752*	*0.1628*
	0.0040	0.0533	0.0353	0.0337	0.0279	0.0495	0.0325	0.0513	0.0458

Table 3.1: *Results of the estimators (Whittle, GPH and local Whittle, LW) analysed with parameter values* $d = 0.1, 0.2, 0.3, 0.4, 0.45$*, series length* $n = 200$*, bootstrap replications* $B = 1000$ *and replications* $S = 2000$*: average value (boldface), standard error (italic), mean squared error (normal font).*

	Monte Carlo			ACF		Local B.		SIEVE	
d	Whittle	GPH	LW	GPH	LW	GPH	LW	GPH	LW
	0.0912	**0.0976**	**0.1083**	**0.0733**	**0.1154**	**0.0946**	**0.1145**	**0.0187**	**0.0601**
0.1	*0.0361*	*0.1680*	*0.1075*	*0.1350*	*0.0847*	*0.1625*	*0.0964*	*0.0442*	*0.0334*
	0.0014	0.0282	0.0116	0.0189	0.0074	0.0264	0.0095	0.0086	0.0027
	0.1909	**0.2027**	**0.1914**	**0.1724**	**0.1844**	**0.1952**	**0.1903**	**0.0645**	**0.0926**
0.2	*0.0371*	*0.1677*	*0.1249*	*0.1335*	*0.1017*	*0.1611*	*0.1139*	*0.0788*	*0.0643*
	0.0015	0.0281	0.0157	0.0186	0.0106	0.0260	0.0131	0.0246	0.0157
	0.2912	**0.3067**	**0.2899**	**0.2728**	**0.2697**	**0.2944**	**0.2819**	**0.1535**	**0.1662**
0.3	*0.0375*	*0.1704*	*0.1366*	*0.1363*	*0.1182*	*0.1652*	*0.1289*	*0.1192*	*0.1096*
	0.0015	0.0291	0.0188	0.0193	0.0149	0.0273	0.0169	0.0357	0.0299
	0.3948	**0.4071**	**0.3899**	**0.3719**	**0.3610**	**0.3929**	**0.3800**	**0.2632**	**0.2685**
0.4	*0.0381*	*0.1713*	*0.1408*	*0.1364*	*0.1269*	*0.1649*	*0.1360*	*0.1476*	*0.1448*
	0.0015	0.0294	0.0199	0.0194	0.0176	0.0273	0.0189	0.0405	0.0383
	0.4440	**0.4613**	**0.4448**	**0.4255**	**0.4125**	**0.4418**	**0.4323**	**0.3276**	**0.3320**
0.45	*0.0386*	*0.1664*	*0.1397*	*0.1362*	*0.1303*	*0.1618*	*0.1384*	*0.1531*	*0.1535*
	0.0015	0.0278	0.0195	0.0192	0.0184	0.0262	0.0195	0.0384	0.0375

Table 3.2: *Results of the estimators (Whittle, GPH and local Whittle, LW) analysed with parameter values* $d = 0.1, 0.2, 0.3, 0.4, 0.45$*, series length* $n = 500$*, bootstrap replications* $B = 1000$ *and replications* $S = 2000$*: average value (boldface), standard error (italic), mean squared error (normal font).*

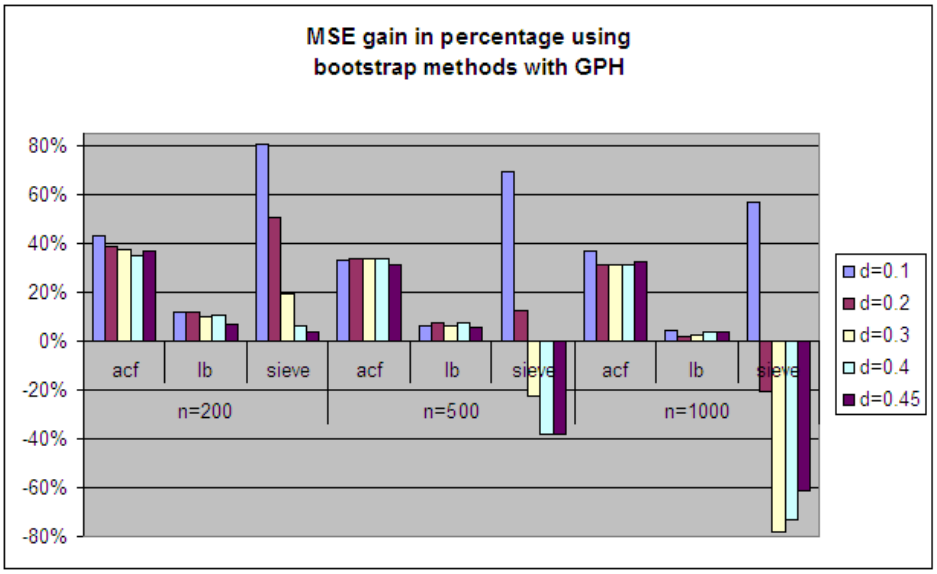
	Monte Carlo			ACF		Local B.		SIEVE	
d	Whittle	GPH	LW	GPH	LW	GPH	LW	GPH	LW
0.1	**0.0945**	**0.0981**	**0.1024**	**0.0807**	**0.1074**	**0.0950**	**0.1069**	**0.0153**	**0.0486**
	0.0256	*0.1393*	*0.0908*	*0.1089*	*0.0730*	*0.1361*	*0.0823*	*0.0354*	*0.0272*
	0.0007	0.0194	0.0083	0.0122	0.0054	0.0185	0.0068	0.0084	0.0034
0.2	**0.1954**	**0.1955**	**0.1907**	**0.1775**	**0.1812**	**0.1889**	**0.1873**	**0.0645**	**0.0844**
	0.0258	*0.1394*	*0.1078*	*0.1135*	*0.0919*	*0.1375*	*0.1004*	*0.0718*	*0.0623*
	0.0007	0.0194	0.0117	0.0134	0.0088	0.0190	0.0102	0.0235	0.0173
0.3	**0.2972**	**0.3059**	**0.2930**	**0.2813**	**0.2733**	**0.2970**	**0.2857**	**0.1574**	**0.1658**
	0.0258	*0.1353*	*0.1105*	*0.1105*	*0.1019*	*0.1338*	*0.1078*	*0.1110*	*0.1075*
	0.0007	0.0183	0.0123	0.0126	0.0111	0.0179	0.0118	0.0326	0.0296
0.4	**0.3984**	**0.4031**	**0.3925**	**0.3801**	**0.3697**	**0.3919**	**0.3841**	**0.2755**	**0.2826**
	0.0257	*0.1381*	*0.1120*	*0.1126*	*0.1062*	*0.1355*	*0.1115*	*0.1325*	*0.1363*
	0.0007	0.0191	0.0126	0.0131	0.0122	0.0184	0.0127	0.0330	0.0324
0.45	**0.4488**	**0.4657**	**0.4535**	**0.4383**	**0.4276**	**0.4522**	**0.4445**	**0.3489**	**0.3586**
	0.0264	*0.1389*	*0.1163*	*0.1145*	*0.1111*	*0.1370*	*0.1170*	*0.1461*	*0.1516*
	0.0007	0.0195	0.0135	0.0132	0.0128	0.0188	0.0137	0.0316	0.0313

Table 3.3: *Results of the estimators (Whittle, GPH and local Whittle, LW) analysed with parameter values* $d = 0.1, 0.2, 0.3, 0.4, 0.45$*, series length* $n = 1000$*, bootstrap replications* $B = 1000$ *and replications* $S = 2000$*: average value (boldface), standard error (italic), mean squared error (normal font).*

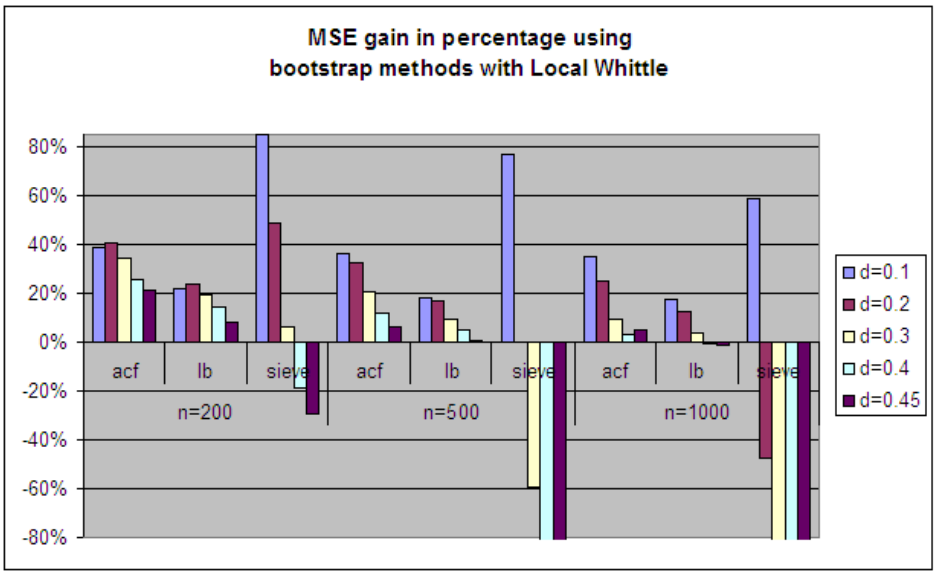
	$n = 200$		$n = 500$		$n = 1000$	
d	GPH	LW	GPH	LW	GPH	LW
0.1	*26.05*	*24.82*	*19.67*	*21.25*	*21.79*	*19.66*
	43.23	39.03	32.96	36.31	36.93	34.84
0.2	*23.11*	*22.91*	*20.38*	*18.60*	*18.56*	*14.75*
	38.92	40.66	33.92	32.50	31.13	24.86
0.3	*22.82*	*20.71*	*19.99*	*13.45*	*18.31*	*7.75*
	37.75	34.40	33.55	20.58	31.48	9.41
0.4	*21.43*	*17.28*	*20.37*	*9.91*	*18.49*	*5.25*
	35.17	25.83	34.01	11.61	31.51	3.35
0.45	*22.61*	*15.83*	*18.11*	*6.74*	*17.60*	*4.49*
	36.81	21.12	31.08	5.96	32.25	5.16

	$n = 200$		$n = 500$		$n = 1000$	
d	GPH	LW	GPH	LW	GPH	LW
0.1	*6.17*	*13.39*	*3.26*	*10.31*	*2.31*	*9.34*
	11.80	21.58	6.33	18.24	4.45	17.30
0.2	*6.21*	*12.53*	*3.90*	*8.84*	*1.34*	*6.88*
	12.06	23.50	7.59	16.69	2.12	12.55
0.3	*5.37*	*10.53*	*3.02*	*5.60*	*1.10*	*2.41*
	10.12	19.14	5.99	9.62	2.32	3.47
0.4	*5.49*	*8.17*	*3.69*	*3.40*	*1.94*	*0.45*
	10.42	14.12	7.23	5.14	3.55	-0.68
0.45	*3.98*	*5.43*	*2.77*	*0.94*	*1.41*	*-0.61*
	7.11	8.14	5.66	0.40	4.00	-1.36

Table 3.4: *Percentage of gain (see Equation (3.7))comparing the Monte Carlo results of estimators GPH and local Whittle (LW) with the bootstrap results in terms of standard deviation (italic) and mean squared error (normal font): in the first part of the table there is the gain using the ACF bootstrap, whereas in the second part there is the gain using the local bootstrap.*



(a) GPH



(b) local Whittle

Figure 3.2: *Percentage gain in term of MSE for (a) GPH and (b) local Whittle for the memory parameter* $d = 0.1, 0.2, 0.3, 0.4, 0.45$*, sample size* $n = 200, 500, 1000$ *and different bootstrap methods (ACF, local and* SIEVE *bootstrap.*

	Monte Carlo			ACF		Local B.	
d	Whittle	GPH	LW	GPH	LW	GPH	LW
	0.0816	**0.0978**	**0.1158**	**0.0630**	**0.1286**	**0.0906**	**0.1264**
0.1	*0.0558*	*0.2230*	*0.1303*	*0.1655*	*0.0977*	*0.2111*	*0.1146*
	0.0035	0.0498	0.0172	0.0288	0.0104	0.0446	0.0138
	0.1818	**0.1974**	**0.1940**	**0.1577**	**0.1897**	**0.1837**	**0.1949**
0.2	*0.0608*	*0.2294*	*0.1571*	*0.1705*	*0.1192*	*0.2145*	*0.1367*
	0.0040	0.0526	0.0247	0.0309	0.0143	0.0463	0.0187
	0.2804	**0.3051**	**0.2908**	**0.2607**	**0.2696**	**0.2859**	**0.2830**
0.3	*0.0606*	*0.2299*	*0.1760*	*0.1763*	*0.1413*	*0.2174*	*0.1592*
	0.0041	0.0529	0.0311	0.0326	0.0209	0.0475	0.0256
	0.3856	**0.4091**	**0.3896**	**0.3605**	**0.3549**	**0.3832**	**0.3745**
0.4	*0.0621*	*0.2274*	*0.1824*	*0.1772*	*0.1523*	*0.2123*	*0.1680*
	0.0041	0.0518	0.0334	0.0329	0.0252	0.0453	0.0289
	0.4360	**0.4628**	**0.4387**	**0.4088**	**0.3972**	**0.4319**	**0.4207**
0.45	*0.0627*	*0.2266*	*0.1861*	*0.1769*	*0.1568*	*0.2156*	*0.1745*
	0.0041	0.0515	0.0348	0.0330	0.0274	0.0468	0.0313
	0.0913	**0.0995**	**0.1096**	**0.0787**	**0.1170**	**0.0948**	**0.1155**
0.1	*0.0361*	*0.1753*	*0.1066*	*0.1324*	*0.0837*	*0.1699*	*0.0960*
	0.0014	0.0307	0.0115	0.0180	0.0073	0.0289	0.0095
	0.1911	**0.1997**	**0.1935**	**0.1743**	**0.1865**	**0.1921**	**0.1927**
0.2	*0.0361*	*0.1755*	*0.1279*	*0.1396*	*0.1058*	*0.1696*	*0.1177*
	0.0014	0.0308	0.0164	0.0201	0.0114	0.0288	0.0139
	0.2932	**0.3072**	**0.2940**	**0.2764**	**0.2727**	**0.2946**	**0.2852**
0.3	*0.0370*	*0.1678*	*0.1353*	*0.1352*	*0.1186*	*0.1618*	*0.1271*
	0.0014	0.0282	0.0183	0.0188	0.0148	0.0262	0.0164
	0.3949	**0.4092**	**0.3950**	**0.3770**	**0.3661**	**0.3926**	**0.3834**
0.4	*0.0367*	*0.1705*	*0.1399*	*0.1379*	*0.1276*	*0.1667*	*0.1365*
	0.0014	0.0292	0.0196	0.0195	0.0174	0.0278	0.0189
	0.4448	**0.4633**	**0.4447**	**0.4266**	**0.4128**	**0.4460**	**0.4335**
0.45	*0.0382*	*0.1675*	*0.1389*	*0.1348*	*0.1284*	*0.1621*	*0.1377*
	0.0015	0.0282	0.0193	0.0187	0.0179	0.0263	0.0192

Table 3.5: *Simulated series with Chi-squared* 1 *d.f.innovations: results of the estimators (Whittle, GPH and local Whittle, LW) analysed with memory parameter values* $d = 0.1, 0.2, 0.3, 0.4, 0.45$*, bootstrap replications* $B = 1000$ *and simulation replications* $S = 2000$*: average value (boldface), standard error (italic), mean squared error (normal font). In the upper part* $n = 200$*, whereas in the lower part* $n = 500$*.*

	Monte Carlo			ACF		Local B.	
d	Whittle	GPH	LW	GPH	LW	GPH	LW
	0.0822	**0.0995**	**0.1232**	**0.0686**	**0.1344**	**0.0915**	**0.1325**
0.1	*0.0554*	*0.2334*	*0.1328*	*0.1744*	*0.1005*	*0.2189*	*0.1168*
	0.0034	0.0545	0.0182	0.0314	0.0113	0.0480	0.0147
	0.1779	**0.2050**	**0.2014**	**0.1632**	**0.1968**	**0.1924**	**0.2026**
0.2	*0.0614*	*0.2304*	*0.1623*	*0.1791*	*0.1265*	*0.2174*	*0.1437*
	0.0043	0.0531	0.0263	0.0334	0.0160	0.0473	0.0206
	0.2796	**0.2936**	**0.2869**	**0.2538**	**0.2681**	**0.2771**	**0.2805**
0.3	*0.0625*	*0.2347*	*0.1816*	*0.1861*	*0.1462*	*0.2231*	*0.1641*
	0.0043	0.0551	0.0332	0.0368	0.0224	0.0503	0.0273
	0.3832	**0.4092**	**0.3890**	**0.3593**	**0.3544**	**0.3819**	**0.3729**
0.4	*0.0644*	*0.2310*	*0.1852*	*0.1788*	*0.1530*	*0.2157*	*0.1691*
	0.0044	0.0534	0.0344	0.0336	0.0255	0.0469	0.0293
	0.4365	**0.4650**	**0.4407**	**0.4094**	**0.3995**	**0.4368**	**0.4237**
0.45	*0.0623*	*0.2382*	*0.1941*	*0.1864*	*0.1629*	*0.2219*	*0.1804*
	0.0041	0.0570	0.0378	0.0364	0.0291	0.0494	0.0332
	0.0899	**0.1008**	**0.1076**	**0.0747**	**0.1150**	**0.0960**	**0.1140**
0.1	*0.0376*	*0.1698*	*0.1035*	*0.1316*	*0.0814*	*0.1645*	*0.0921*
	0.0015	0.0288	0.0108	0.0180	0.0068	0.0271	0.0087
	0.1907	**0.1980**	**0.1913**	**0.1710**	**0.1842**	**0.1898**	**0.1899**
0.2	*0.0374*	*0.1759*	*0.1259*	*0.1386*	*0.1027*	*0.1691*	*0.1137*
	0.0015	0.0309	0.0159	0.0201	0.0108	0.0287	0.0130
	0.2933	**0.3023**	**0.2902**	**0.2728**	**0.2701**	**0.2915**	**0.2830**
0.3	*0.0365*	*0.1720*	*0.1362*	*0.1359*	*0.1184*	*0.1667*	*0.1289*
	0.0014	0.0296	0.0186	0.0192	0.0149	0.0279	0.0169
	0.3950	**0.4104**	**0.3924**	**0.3767**	**0.3642**	**0.3948**	**0.3819**
0.4	*0.0372*	*0.1716*	*0.1413*	*0.1386*	*0.1282*	*0.1638*	*0.1371*
	0.0014	0.0295	0.0200	0.0198	0.0177	0.0268	0.0191
	0.4448	**0.4645**	**0.4474**	**0.4288**	**0.4154**	**0.4448**	**0.4344**
0.45	*0.0375*	*0.1678*	*0.1401*	*0.1366*	*0.1306*	*0.1623*	*0.1374*
	0.0014	0.0284	0.0196	0.0191	0.0183	0.0264	0.0191

Table 3.6: *Simulated series with Student t 4 d.f. innovations: results of the estimators (Whittle, GPH and local Whittle, LW) analysed with memory parameter values* $d = 0.1, 0.2, 0.3, 0.4, 0.45$*, bootstrap replications* $B = 1000$ *and simulation replications* $S = 2000$*: average value (boldface), standard error (italic), mean squared error (normal font). In the upper part* $n = 200$*, whereas in the lower part* $n = 500$*.*

	Monte Carlo			ACF		Local B.	
d	Whittle	GPH	LW	GPH	LW	GPH	LW
	0.0808	**0.0987**	**0.1227**	**0.0662**	**0.1330**	**0.0909**	**0.1317**
0.1	*0.0544*	*0.2303*	*0.1336*	*0.1735*	*0.1009*	*0.2167*	*0.1166*
	0.0033	0.0530	0.0184	0.0312	0.0113	0.0470	0.0146
	0.1793	**0.1930**	**0.1930**	**0.1551**	**0.1897**	**0.1820**	**0.1940**
0.2	*0.0626*	*0.2296*	*0.1602*	*0.1758*	*0.1234*	*0.2138*	*0.1407*
	0.0044	0.0528	0.0257	0.0329	0.0153	0.0461	0.0198
	0.2799	**0.3082**	**0.2902**	**0.2598**	**0.2685**	**0.2896**	**0.2825**
0.3	*0.0625*	*0.2254*	*0.1724*	*0.1754*	*0.1376*	*0.2116*	*0.1548*
	0.0043	0.0509	0.0298	0.0324	0.0199	0.0449	0.0243
	0.3834	**0.4084**	**0.3865**	**0.3580**	**0.3519**	**0.3845**	**0.3722**
0.4	*0.0622*	*0.2307*	*0.1871*	*0.1804*	*0.1556*	*0.2155*	*0.1725*
	0.0041	0.0533	0.0352	0.0343	0.0265	0.0467	0.0305
	0.4372	**0.4719**	**0.4462**	**0.4138**	**0.4034**	**0.4402**	**0.4275**
0.45	*0.0626*	*0.2349*	*0.1924*	*0.1834*	*0.1620*	*0.2214*	*0.1791*
	0.0041	0.0557	0.0370	0.0350	0.0284	0.0491	0.0326
	0.0908	**0.1006**	**0.1099**	**0.0778**	**0.1175**	**0.0961**	**0.1167**
0.1	*0.0369*	*0.1740*	*0.1075*	*0.1336*	*0.0848*	*0.1699*	*0.0970*
	0.0014	0.0303	0.0117	0.0183	0.0075	0.0289	0.0097
	0.1913	**0.2021**	**0.1938**	**0.1740**	**0.1858**	**0.1946**	**0.1927**
0.2	*0.0369*	*0.1700*	*0.1254*	*0.1332*	*0.1022*	*0.1647*	*0.1150*
	0.0014	0.0289	0.0158	0.0184	0.0106	0.0271	0.0133
	0.2918	**0.3029**	**0.2891**	**0.2740**	**0.2690**	**0.2910**	**0.2818**
0.3	*0.0363*	*0.1714*	*0.1353*	*0.1366*	*0.1174*	*0.1669*	*0.1288*
	0.0014	0.0294	0.0184	0.0193	0.0147	0.0279	0.0169
	0.3943	**0.4065**	**0.3909**	**0.3733**	**0.3626**	**0.3913**	**0.3798**
0.4	*0.0382*	*0.1731*	*0.1435*	*0.1407*	*0.1307*	*0.1656*	*0.1396*
	0.0015	0.0300	0.0207	0.0205	0.0185	0.0275	0.0199
	0.4455	**0.4635**	**0.4470**	**0.4288**	**0.4153**	**0.4442**	**0.4345**
0.45	*0.0388*	*0.1756*	*0.1432*	*0.1410*	*0.1320*	*0.1705*	*0.1415*
	0.0015	0.0310	0.0205	0.0203	0.0186	0.0291	0.0203

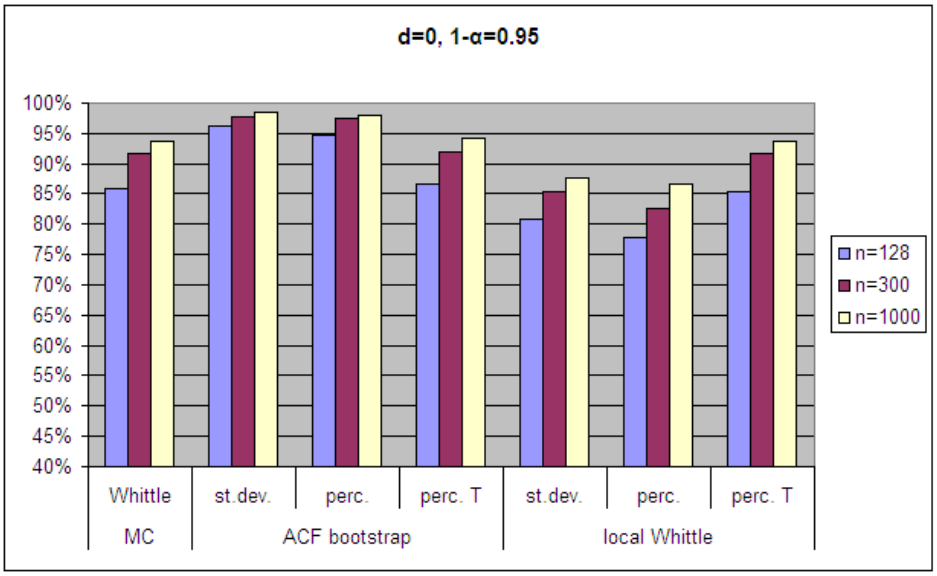
Table 3.7: *Simulated series with Student t 6 d.f. innovations: results of the estimators (Whittle, GPH and local Whittle, LW) analysed with memory parameter values $d = 0.1, 0.2, 0.3, 0.4, 0.45$, bootstrap replications $B = 1000$ and simulation replications $S = 2000$: average value (boldface), standard error (italic), mean squared error (normal font). In the upper part $n = 200$, whereas in the lower part $n = 500$.*

	$n = 200$				$n = 500$			
	ACF		Local B.		ACF		Local B.	
d	GPH	LW	GPH	LW	GPH	LW	GPH	LW
0.1	*25.80*	*25.03*	*5.36*	*12.07*	*24.48*	*21.50*	*3.11*	*9.98*
	42.20	39.88	10.27	19.77	41.49	36.36	6.05	17.52
0.2	*25.64*	*24.12*	*6.48*	*12.99*	*20.47*	*17.29*	*3.35*	*7.98*
	41.31	42.09	12.05	24.29	34.60	30.65	6.39	15.21
0.3	*23.30*	*19.72*	*5.42*	*9.53*	*19.40*	*12.35*	*3.54*	*6.05*
	38.27	32.75	10.21	17.44	33.17	19.26	7.03	10.72
0.4	*22.09*	*16.50*	*6.66*	*7.91*	*19.13*	*8.85*	*2.24*	*2.49*
	36.39	24.39	12.48	13.53	32.97	11.17	4.52	3.63
0.45	*21.94*	*15.76*	*4.86*	*6.24*	*19.52*	*7.52*	*3.23*	*0.86*
	35.97	21.26	9.13	9.95	33.70	7.42	6.89	0.43
0.1	*25.27*	*24.38*	*6.21*	*12.11*	*22.47*	*21.43*	*3.07*	*11.04*
	42.34	38.01	11.90	19.25	37.66	36.50	5.98	19.47
0.2	*22.25*	*22.07*	*5.65*	*11.46*	*21.18*	*18.43*	*3.88*	*9.66*
	37.04	39.23	10.91	21.59	35.16	32.21	7.28	18.13
0.3	*20.68*	*19.53*	*4.93*	*9.67*	*21.01*	*13.05*	*3.07*	*5.36*
	33.26	32.50	8.73	17.68	35.12	19.99	5.82	9.34
0.4	*22.60*	*17.40*	*6.60*	*8.73*	*19.18*	*9.24*	*4.54*	*2.95*
	37.09	25.98	12.29	14.85	33.08	11.46	9.12	4.44
0.45	*21.73*	*16.07*	*6.82*	*7.06*	*18.55*	*6.81*	*3.27*	*1.95*
	36.08	22.97	13.22	11.98	32.57	7.10	7.03	2.67
0.1	*24.67*	*24.51*	*5.91*	*12.72*	*23.19*	*21.13*	*2.32*	*9.78*
	41.10	38.66	11.31	20.48	39.37	35.68	4.53	16.90
0.2	*23.44*	*23.02*	*6.85*	*12.19*	*21.66*	*18.48*	*3.13*	*8.30*
	37.62	40.43	12.70	22.89	36.30	32.43	6.07	15.78
0.3	*22.18*	*20.17*	*6.12*	*10.21*	*20.31*	*13.25*	*2.62*	*4.78*
	36.34	33.15	11.78	18.62	34.21	20.04	4.94	8.12
0.4	*21.80*	*16.85*	*6.59*	*7.80*	*18.74*	*8.86*	*4.32*	*2.69*
	35.62	24.65	12.41	13.25	31.68	10.52	8.32	3.73
0.45	*21.92*	*15.80*	*5.74*	*6.91*	*19.71*	*7.77*	*2.90*	*1.18*
	37.20	23.28	11.74	12.01	34.46	9.11	6.17	1.22

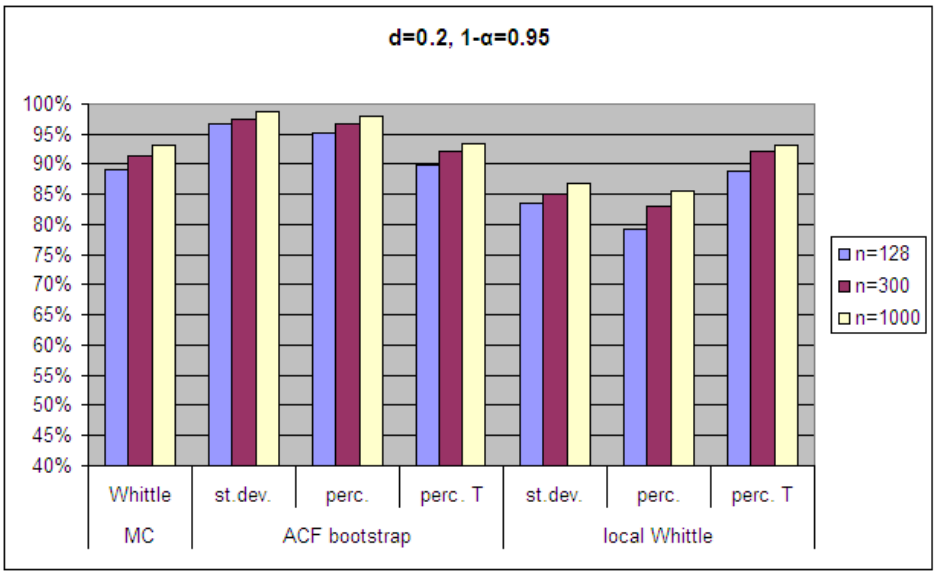
Table 3.8: *Percentage of gain comparing the Monte Carlo results of estimators GPH and local Whittle with the bootstrap results in terms of standard deviation (italic) and mean squared error (normal font) for fractional integrated non-Gaussian (Chi-squared with* 1 *d.f. and Student t with* 4 *and* 6 *d.f.) processes: first part series with Chi-squared* 1 *d.f. innovations, second part series with Student t* 4 *d.f. innovations, third part series with Student t* 6 *d.f. innovations.*

				ACF bootstrap			local bootstrap		
d	n	$1-\alpha$	MC	SD	P	PT	SD	P	PT
		90	78.60	91.85	89.80	79.75	73.50	70.60	78.60
	128	95	85.95	96.10	94.60	86.75	80.80	77.90	85.30
		99	94.35	98.90	98.60	94.95	89.50	86.80	93.45
		90	85.25	95.05	93.25	86.30	77.05	75.80	85.85
0	300	95	91.65	97.65	97.40	91.95	85.45	82.70	91.75
		99	97.15	99.85	99.60	97.55	93.75	92.40	96.80
		90	88.35	96.10	95.70	88.80	80.70	80.00	88.80
	1000	95	93.70	98.55	98.10	94.10	87.55	86.55	93.80
		99	98.55	99.60	99.60	98.80	95.40	94.65	98.55
		90	81.45	93.35	91.15	82.90	75.75	72.70	82.25
	128	95	89.05	96.60	95.15	89.95	83.45	79.25	88.75
		99	95.30	98.85	98.50	94.95	91.90	89.65	95.20
		90	84.90	94.80	92.90	85.70	77.80	76.05	85.50
0.2	300	95	91.40	97.55	96.75	92.05	85.05	83.00	92.15
		99	97.25	99.70	99.15	97.55	93.35	91.75	97.55
		90	87.15	95.40	94.65	87.90	79.45	79.00	87.75
	1000	95	93.05	98.65	97.85	93.40	86.95	85.55	93.20
		99	98.90	99.75	99.60	98.75	95.10	94.00	98.70
		90	85.75	94.05	91.35	63.25	73.50	75.40	64.20
	128	95	91.00	96.75	95.40	66.60	81.75	82.75	68.05
		99	95.95	99.20	98.65	70.40	88.85	90.60	72.25
		90	89.85	92.25	93.60	65.35	71.25	77.50	68.00
0.45	300	95	93.40	96.30	96.60	68.50	78.60	84.40	71.30
		99	97.65	99.15	99.20	71.60	88.35	93.35	74.65
		90	88.15	92.50	95.80	77.85	77.90	80.90	81.15
	1000	95	93.70	95.85	98.50	81.05	84.90	87.55	84.70
		99	98.95	98.45	99.65	83.30	93.25	95.35	86.75
		90	96.00	93.75	97.60	46.35	65.50	75.75	49.85
	128	95	97.70	96.70	99.30	48.35	72.15	82.00	51.50
		99	99.45	98.55	100.00	50.00	80.20	89.55	52.95
		90	94.85	92.70	97.95	59.60	68.40	79.60	63.20
-0.45	300	95	97.70	96.25	99.05	61.25	76.10	86.30	65.10
		99	99.30	98.95	99.85	63.35	84.35	93.40	67.20
		90	86.25	90.70	96.40	73.70	74.65	78.70	77.95
	1000	95	91.90	94.30	99.00	76.15	81.15	85.30	81.50
		99	99.35	98.40	100.00	79.30	89.70	93.90	84.25

Table 3.9: *Observed coverage (%) of the Whittle estimator relative to a confidence interval with $1-\alpha$ nominal level for different values of the parameter $d = 0, 0.2, 0.45, -0.45$ and different sample sizes $n = 128, 300, 1000$ computed using Monte Carlo (MC), bootstrap standard deviation (SD), percentile (P), percentile t (PT) methods and two different bootstrap procedures (ACF and local bootstrap).*

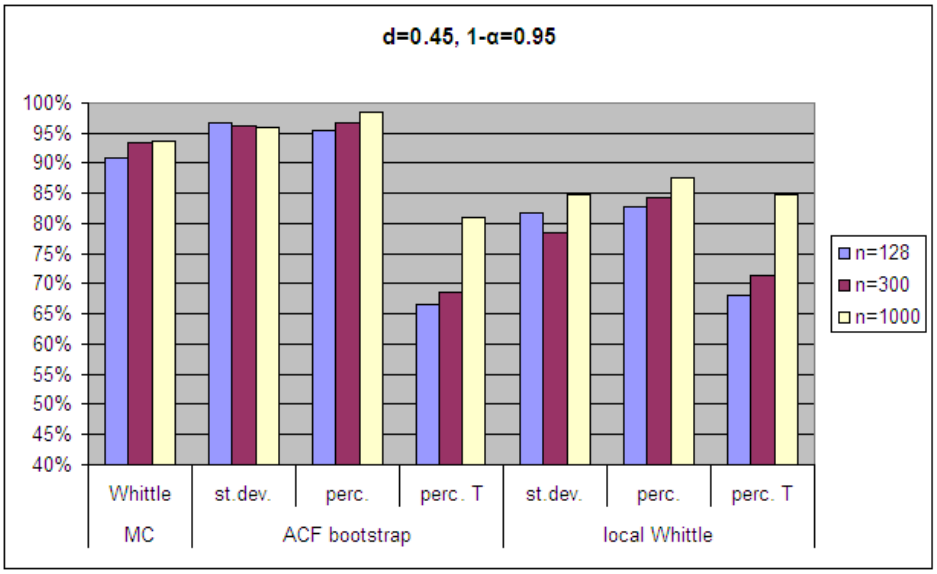


(a) $d = 0$

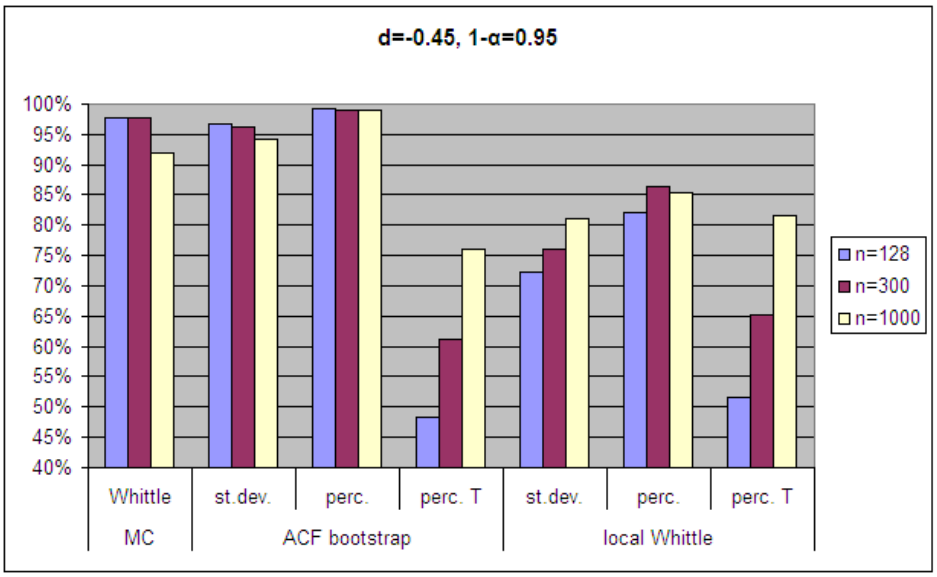


(b) $d = 0.2$

Figure 3.3: *Actual coverage of confidence intervals of Whittle estimator with confidence* $1 - \alpha = 0.95$*, memory parameter* $d = 0, 0.2$*, series length* $n = 128, 300, 1000$*, different bootstrap methods (ACF and local bootstrap) and different methods to build the interval (bootstrap standard error, percentile and percentile t).*



(a) $d = 0.45$



(b) $d = -0.45$

Figure 3.4: *Actual coverage of confidence intervals of Whittle estimator with confidence* $1 - \alpha = 0.95$*, memory parameter* $d = 0.45, -0.45$*, series length* $n = 128, 300, 1000$*, different bootstrap methods (ACF and local bootstrap) and different methods to build the interval (bootstrap standard error, percentile and percentile t).*

			ACF bootstrap			local bootstrap			log-periodogram		
ϕ		MC	SD	P	PT	SD	P	PT	SD	P	PT
	90	85.70	94.95	97.70	72.60	83.10	82.80	78.85	91.10	81.50	85.50
0	95	90.90	98.15	99.30	80.05	88.30	81.90	83.65	94.85	88.10	91.00
	99	96.40	99.85	100.00	91.05	94.30	88.75	90.10	98.50	94.40	95.95
	90	85.80	95.75	97.35	73.55	84.25	84.30	78.95	91.80	82.35	85.90
0.2	95	91.25	98.20	99.15	81.45	88.45	82.25	84.50	94.60	87.65	91.05
	99	96.35	99.75	99.90	91.60	94.15	89.20	89.50	97.95	93.90	96.05
	90	86.50	95.15	98.25	71.00	79.55	84.45	79.25	88.85	82.90	86.40
0.4	95	91.80	97.50	99.55	80.25	85.20	80.35	85.45	93.05	88.95	91.75
	99	97.25	99.60	99.95	92.05	92.65	87.40	91.70	97.85	95.45	96.95
	90	81.25	88.00	96.10	65.45	69.20	36.85	74.20	80.55	77.35	80.75
0.6	95	88.15	94.00	99.20	74.15	76.75	72.95	80.90	86.80	84.35	86.60
	99	95.60	98.70	100.00	88.10	86.00	81.75	88.85	94.65	93.30	94.60
	90	50.50	57.75	72.75	39.65	34.70	21.90	47.65	45.25	46.90	48.95
0.8	95	61.90	70.05	89.00	46.80	42.50	40.20	56.05	55.35	57.55	59.85
	99	80.15	89.70	99.40	64.45	56.15	51.75	69.15	74.25	72.90	76.30
	90	88.65	96.00	96.85	75.60	86.05	86.95	83.90	93.95	85.85	88.40
0	95	93.75	98.30	98.75	83.80	90.15	83.60	89.85	97.15	91.75	93.50
	99	97.90	99.50	99.90	93.75	95.25	90.05	94.70	99.25	96.90	97.85
	90	89.00	96.40	96.95	74.80	85.60	87.60	83.35	93.80	86.50	88.90
0.2	95	93.35	98.30	99.25	82.60	90.15	84.20	88.85	96.85	91.90	92.90
	99	97.75	99.75	99.90	93.40	95.50	91.90	94.05	99.10	96.95	97.55
	90	88.55	95.80	97.55	74.35	83.05	88.60	84.50	92.55	86.80	88.50
0.4	95	94.05	98.00	99.05	83.60	87.30	82.20	89.70	96.55	92.30	93.65
	99	98.00	99.65	100.00	94.40	93.60	90.15	95.25	99.10	96.95	97.90
	90	85.70	92.65	97.35	71.65	78.10	40.60	81.65	88.55	83.60	85.50
0.6	95	92.30	96.10	99.30	80.95	83.00	78.65	87.65	93.20	90.75	92.05
	99	98.20	99.25	100.00	91.55	90.25	87.10	94.40	98.20	97.15	97.45
	90	65.40	69.95	83.55	50.30	48.90	57.05	62.35	62.35	63.85	63.75
0.8	95	75.05	80.40	93.25	59.85	55.80	54.50	70.85	71.85	73.55	73.20
	99	89.20	93.25	99.60	76.15	68.55	64.55	83.50	86.70	86.90	86.50

Table 3.10: *Observed coverage (%) of the GPH estimator relative to a confidence interval with* $1-\alpha$ *nominal level (second column) for* $d = 0$*, different values of* $\phi = 0, 0.2, 0.4, 0.6, 0.8$ *computed using Monte Carlo (MC), bootstrap standard deviation (SD), percentile (P), percentile t (PT) methods and three different bootstrap procedures (ACF, local bootstrap and log-periodogram regression): upper part with sample size* $n = 128$*, lower part with sample size* $n = 300$*.*

ϕ		MC	ACF bootstrap SD	ACF bootstrap P	ACF bootstrap PT	local bootstrap SD	local bootstrap P	local bootstrap PT	log-periodogram SD	log-periodogram P	log-periodogram PT
	90	86.15	96.40	96.50	74.30	85.40	84.45	78.55	92.30	82.25	86.10
0	95	90.95	98.20	98.35	82.40	90.40	82.80	84.30	95.30	88.25	90.70
	99	96.80	99.80	99.95	92.65	95.70	89.75	90.15	98.30	94.35	96.15
	90	85.65	95.05	96.30	71.85	84.05	82.35	78.40	90.95	82.10	85.85
0.2	95	91.20	97.95	98.25	80.65	88.60	82.30	83.80	94.25	87.95	90.90
	99	95.90	99.50	99.85	91.10	94.15	89.05	89.85	97.40	93.80	95.35
	90	85.95	94.00	96.80	71.65	79.60	48.00	78.55	88.40	82.85	85.70
0.4	95	91.50	96.75	99.15	80.65	84.90	80.75	84.15	93.00	88.60	90.75
	99	97.05	99.45	99.95	91.95	92.75	88.40	90.80	97.40	94.75	96.10
	90	81.05	88.05	96.30	63.60	68.35	29.55	71.95	79.15	77.20	79.85
0.6	95	87.90	93.20	99.00	73.60	75.85	74.10	78.90	86.05	84.80	86.20
	99	95.55	98.20	99.90	87.45	85.80	82.35	87.90	94.55	92.85	94.65
	90	51.65	59.70	77.15	38.30	37.20	39.50	47.90	47.00	48.65	50.90
0.8	95	63.50	70.95	90.30	48.25	43.55	45.10	55.30	56.65	58.65	60.30
	99	79.85	89.05	99.45	64.15	58.20	54.85	66.90	74.00	73.90	76.95
	90	86.70	95.80	95.70	74.20	85.90	85.60	81.80	92.95	84.05	86.85
0	95	91.95	97.80	98.20	82.40	89.80	82.35	87.60	95.95	90.00	92.00
	99	97.55	99.60	99.60	92.80	94.95	90.15	93.70	98.60	96.40	97.20
	90	87.25	96.10	95.60	74.75	85.35	86.85	82.00	93.45	84.95	87.00
0.2	95	92.85	98.40	97.80	82.95	89.35	81.60	88.50	96.30	90.75	93.10
	99	97.45	99.80	99.70	93.55	95.40	90.15	94.50	99.05	96.15	97.35
	90	87.90	95.70	96.95	74.20	84.50	47.90	83.35	92.20	85.30	87.95
0.4	95	93.80	98.10	99.05	83.85	88.80	83.75	89.05	96.00	91.85	93.65
	99	98.75	99.90	99.90	94.50	94.85	90.60	95.60	99.30	98.20	98.55
	90	85.30	92.10	97.20	71.80	75.80	79.05	80.30	86.95	83.90	85.05
0.6	95	92.30	95.90	98.70	79.90	82.25	80.20	85.85	91.85	90.25	91.15
	99	98.00	99.35	99.95	91.00	90.25	87.90	93.35	97.80	96.45	97.15
	90	64.40	70.20	85.05	49.95	49.15	77.45	59.45	61.00	62.10	61.85
0.8	95	74.70	80.10	94.65	58.90	55.50	57.20	68.85	71.45	73.55	72.75
	99	89.95	94.25	99.60	75.85	70.50	68.55	81.70	87.00	87.90	87.15

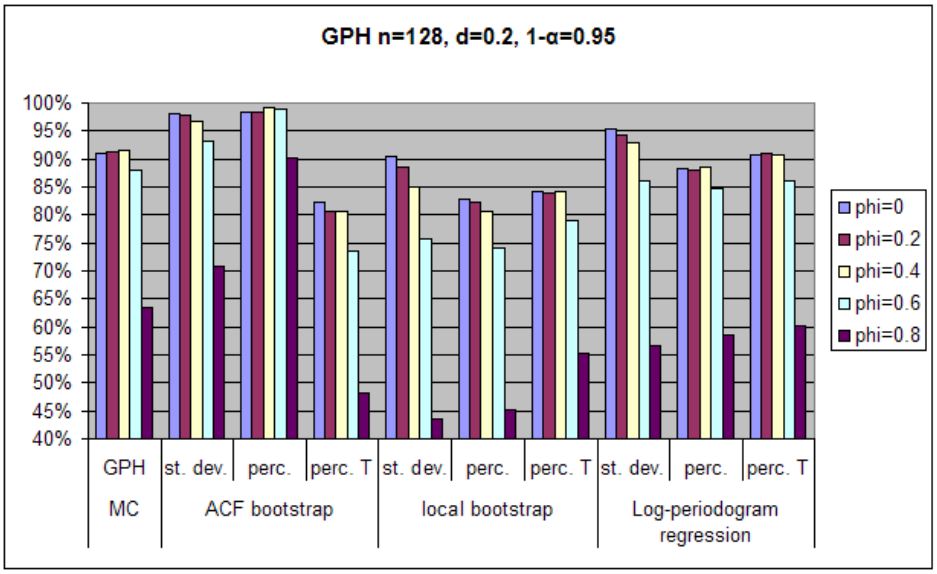
Table 3.11: *Observed coverage (%) of the GPH estimator relative to a confidence interval with* $1-\alpha$ *nominal level (second column) for* $d = 0.2$*, different values of* $\phi = 0, 0.2, 0.4, 0.6, 0.8$ *computed using Monte Carlo (MC), bootstrap standard deviation (SD), percentile (P), percentile t (PT) methods and three different bootstrap procedures (ACF, local bootstrap and log-periodogram regression): upper part with sample size* $n = 128$*, lower part with sample size* $n = 300$*.*

			ACF bootstrap			local bootstrap			log-periodogram		
ϕ		MC	SD	P	PT	SD	P	PT	SD	P	PT
	90	85.40	95.65	95.20	71.60	84.40	68.25	77.30	91.10	82.10	85.80
0	95	91.30	97.60	98.10	80.30	89.85	82.45	82.60	95.15	88.10	90.80
	99	96.75	99.80	99.70	92.15	95.35	89.55	90.00	98.15	94.30	96.05
	90	85.15	94.75	96.20	72.05	84.20	37.35	76.65	89.35	81.80	84.85
0.2	95	90.90	97.25	98.35	80.10	89.25	83.25	82.45	93.40	87.70	90.60
	99	96.95	99.40	99.75	91.00	95.05	90.00	89.35	97.90	94.85	96.65
	90	86.60	94.10	96.55	71.20	81.80	50.05	76.70	87.80	83.05	86.10
0.4	95	92.10	96.80	98.45	80.00	87.70	84.55	83.10	93.40	89.25	91.50
	99	96.65	99.55	99.80	90.65	94.45	90.45	90.30	97.40	94.60	96.20
	90	81.30	87.45	97.95	63.60	70.45	73.35	70.40	79.25	77.20	79.60
0.6	95	88.30	93.20	99.40	72.85	78.60	78.50	77.75	86.20	85.70	86.95
	99	95.80	98.10	100.00	85.30	88.30	86.35	86.45	94.80	93.15	94.80
	90	51.70	58.15	82.85	37.40	38.80	57.80	42.40	46.55	47.55	49.95
0.8	95	63.25	71.60	95.45	44.80	46.85	51.05	50.20	56.45	58.60	60.55
	99	81.40	89.95	100.00	61.85	63.45	63.10	65.15	75.10	75.05	77.20
	90	86.95	95.45	95.10	74.70	85.25	73.55	81.20	91.55	84.70	86.90
0	95	92.15	97.95	97.40	82.25	89.95	83.65	86.65	95.55	90.50	92.05
	99	97.20	99.70	99.45	92.55	95.20	90.70	93.00	98.55	96.05	97.40
	90	88.35	95.20	96.15	74.40	85.00	32.00	82.40	92.05	86.35	87.60
0.2	95	93.15	97.40	98.90	82.70	89.90	85.00	88.70	95.15	91.10	92.75
	99	98.20	99.60	99.85	92.95	95.05	91.55	94.35	98.70	97.35	97.90
	90	87.90	93.80	96.45	74.45	83.20	78.65	81.25	89.80	85.50	86.75
0.4	95	92.65	96.85	98.90	81.10	88.05	84.75	87.30	94.15	91.60	92.20
	99	97.70	99.30	99.70	91.95	94.45	91.55	93.50	98.10	96.80	97.05
	90	85.60	91.75	96.75	69.25	77.30	84.10	77.20	86.95	83.40	84.70
0.6	95	91.45	96.05	98.55	78.25	83.25	82.15	84.95	92.15	90.20	90.95
	99	97.30	99.45	99.80	90.00	92.20	90.10	92.75	97.30	96.50	96.80
	90	62.85	69.80	85.50	47.90	48.45	71.90	55.65	60.80	61.55	61.65
0.8	95	73.65	81.00	95.85	57.00	57.20	59.15	64.80	69.65	71.85	70.95
	99	87.70	93.95	99.85	72.60	72.10	71.50	78.40	85.25	86.30	85.30

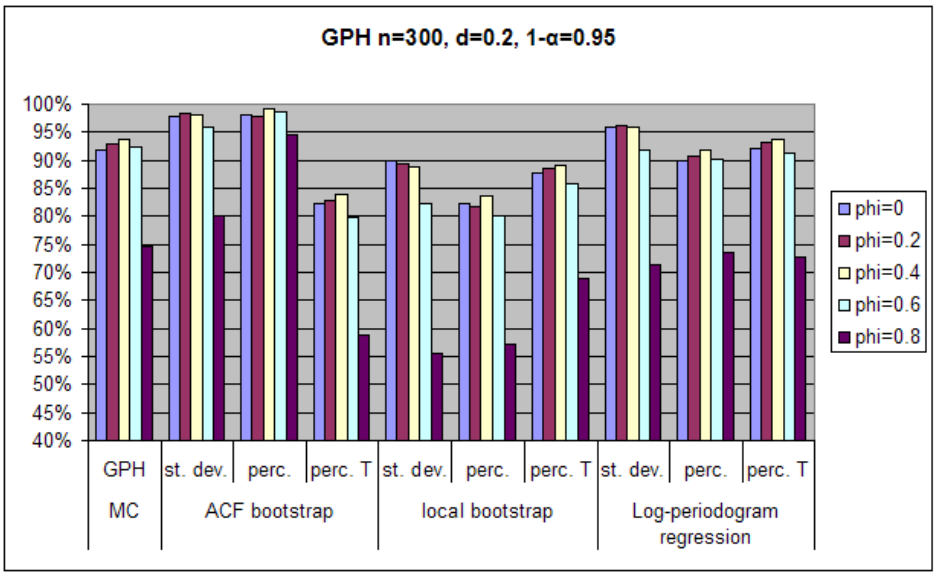
Table 3.12: *Observed coverage (%) of the GPH estimator relative to a confidence interval with* $1-\alpha$ *nominal level (second column) for* $d = 0.45$*, different values of* $\phi = 0, 0.2, 0.4, 0.6, 0.8$ *computed using Monte Carlo (MC), bootstrap standard deviation (SD), percentile (P), percentile t (PT) methods and three different bootstrap procedures (ACF, local bootstrap and log-periodogram regression): upper part with sample size* $n = 128$*, lower part with sample size* $n = 300$*.*

ϕ		MC	ACF bootstrap SD	ACF bootstrap P	ACF bootstrap PT	local bootstrap SD	local bootstrap P	local bootstrap PT	log-periodogram SD	log-periodogram P	log-periodogram PT
	90	86.40	95.40	99.30	68.55	81.75	60.95	79.80	90.50	83.20	86.40
0	95	91.40	98.00	99.85	78.70	86.95	80.05	85.55	94.30	88.95	91.25
	99	96.95	99.60	100.00	89.90	93.65	87.85	91.00	98.00	94.70	96.45
	90	84.95	94.30	98.75	68.95	80.45	81.85	79.45	89.35	81.95	85.10
0.2	95	90.55	97.40	99.85	78.20	85.75	78.95	85.55	93.25	87.40	90.25
	99	96.65	99.85	100.00	89.65	92.45	86.60	90.45	97.50	93.85	96.00
	90	85.50	91.35	98.10	69.95	77.05	84.80	80.40	85.90	81.40	84.85
0.4	95	91.10	95.50	99.65	78.10	82.95	76.15	85.95	91.55	87.95	90.15
	99	96.25	99.15	100.00	89.55	90.60	84.70	91.95	96.55	94.35	95.60
	90	79.50	86.55	94.50	63.75	65.15	78.45	76.55	77.00	75.30	79.00
0.6	95	87.25	92.65	98.60	73.35	72.10	66.90	83.15	84.90	83.60	86.15
	99	94.45	98.10	99.90	87.30	82.90	76.10	90.80	93.40	92.05	93.70
	90	47.60	55.30	63.85	39.85	34.25	41.25	50.65	42.95	44.55	46.40
0.8	95	60.15	68.40	82.80	48.30	41.55	35.75	59.60	52.20	54.70	56.95
	99	79.60	87.65	98.25	66.45	54.20	45.55	72.35	71.95	71.60	75.00
	90	87.30	94.55	98.40	72.55	81.90	72.70	84.05	91.10	85.30	87.00
0	95	92.45	97.55	99.85	79.40	86.40	81.20	89.60	94.75	91.00	92.15
	99	98.30	99.75	100.00	90.30	93.05	88.30	94.75	98.85	97.45	97.90
	90	87.75	94.30	98.45	70.35	79.55	87.00	83.65	91.10	85.05	87.50
0.2	95	93.00	97.00	99.90	79.15	85.50	79.45	89.30	94.65	91.40	92.50
	99	97.50	99.50	100.00	90.50	93.05	88.70	94.60	98.45	96.15	97.05
	90	85.80	93.35	97.95	71.25	79.20	85.95	82.65	89.05	82.85	85.25
0.4	95	92.10	96.85	99.55	79.85	84.75	78.40	88.60	93.50	90.00	91.55
	99	97.50	99.50	100.00	90.25	91.90	87.50	93.90	98.00	96.60	97.05
	90	83.95	90.85	96.15	69.65	73.10	83.80	82.20	85.20	81.80	82.85
0.6	95	90.90	95.55	99.00	79.00	78.80	73.95	87.90	91.05	89.55	90.10
	99	97.25	99.20	99.95	91.00	88.05	84.10	94.10	97.25	96.00	96.65
	90	61.80	68.50	75.60	53.30	46.35	59.80	64.95	58.95	60.15	60.40
0.8	95	73.40	79.05	89.40	62.50	54.85	48.95	73.65	69.40	71.85	70.95
	99	88.80	92.95	99.40	77.65	67.35	60.65	85.90	85.70	86.25	86.10

Table 3.13: *Observed coverage (%) of the GPH estimator relative to a confidence interval with $1-\alpha$ nominal level (second column) for $d = -0.45$, different values of $\phi = 0, 0.2, 0.4, 0.6, 0.8$ computed using Monte Carlo (MC), bootstrap standard deviation (SD), percentile (P), percentile t (PT) methods and three different bootstrap procedures (ACF, local bootstrap and log-periodogram regression): upper part with sample size $n = 128$, lower part with sample size $n = 300$.*



(a) $n = 128$



(b) $n = 300$

Figure 3.5: *Actual coverage of confidence intervals of GPH estimator with confidence* $1 - \alpha = 0.95$*, memory parameter* $d = 0, 0.2, 0.45, -0.45$*, autoregressive parameter* $\phi = 0, 0.2, 0.4, 0.6, 0.8$*, series length* $n = 128, 300, 1000$*, different bootstrap methods (ACF and local bootstrap) and different methods to build the interval (bootstrap standard error and percentile).*

			ACF bootstrap			local bootstrap		
ϕ		MC	SD	P	PT	SD	P	PT
	90	71.55	94.75	94.65	60.25	84.90	76.20	61.95
0	95	79.10	96.65	97.90	61.10	88.65	81.80	66.00
	99	89.05	98.75	99.70	61.20	93.55	88.45	68.40
	90	72.15	95.20	94.70	60.85	88.00	77.15	63.25
0.2	95	79.40	97.20	97.80	61.70	91.00	82.55	67.00
	99	90.20	98.90	99.60	61.70	94.75	89.45	70.15
	90	74.00	93.95	96.30	60.85	82.45	75.70	63.75
0.4	95	81.75	96.05	98.50	61.30	87.90	81.95	68.70
	99	90.65	98.65	99.75	61.45	92.90	88.80	71.90
	90	63.60	85.35	94.95	50.95	71.10	70.15	55.15
0.6	95	72.50	89.55	98.55	51.75	77.25	75.05	59.10
	99	85.25	95.15	99.85	51.80	85.20	82.10	61.95
	90	25.75	43.50	69.45	18.25	31.10	40.20	22.35
0.8	95	33.35	52.15	84.00	18.45	37.45	46.10	24.15
	99	49.20	65.95	97.85	18.50	48.00	54.30	25.25
	90	75.70	97.50	94.45	72.05	89.25	75.25	70.80
0	95	83.45	98.60	97.30	75.45	92.35	81.50	76.40
	99	93.65	99.60	99.45	76.15	96.55	89.05	81.85
	90	78.60	97.10	94.20	73.60	89.65	77.15	71.80
0.2	95	84.90	98.70	97.35	77.40	92.75	82.90	78.10
	99	93.45	99.55	99.55	78.45	96.70	91.00	83.40
	90	76.45	97.05	95.35	71.60	87.70	76.70	70.90
0.4	95	84.80	98.50	97.90	75.65	90.65	83.20	77.05
	99	93.95	99.65	99.40	76.30	95.55	89.95	83.35
	90	73.35	93.95	95.40	68.35	82.25	74.75	67.60
0.6	95	82.75	96.85	97.95	71.70	86.70	81.45	74.05
	99	92.70	98.95	99.70	72.90	93.40	88.65	80.35
	90	41.70	66.45	80.70	37.25	46.45	49.75	39.25
0.8	95	51.30	74.75	91.00	40.05	53.80	57.05	45.70
	99	69.50	87.30	98.65	40.70	67.70	68.60	52.15

Table 3.14: *Observed coverage (%) of the local Whittle estimator relative to a confidence interval with* $1-\alpha$ *nominal level for* $d = 0$*, different values of* $\phi = 0, 0.2, 0.4, 0.6, 0.8$ *computed using Monte Carlo (MC), bootstrap standard deviation (SD), percentile (P), percentile t (PT) methods and two different bootstrap procedures (ACF and local bootstrap): upper part with sample size* $n = 128$*, lower part with sample size* $n = 300$*.*

			ACF bootstrap			local bootstrap		
ϕ		MC	SD	P	PT	SD	P	PT
	90	72.05	94.25	93.65	60.00	84.95	75.75	59.85
0	95	80.20	97.55	96.90	60.55	88.25	81.90	63.05
	99	93.75	99.50	99.55	60.60	92.40	89.45	65.60
	90	72.20	92.40	94.20	56.60	84.50	76.00	58.30
0.2	95	79.45	95.60	96.85	57.30	87.45	82.35	60.35
	99	93.90	99.00	99.45	57.30	90.95	89.00	63.40
	90	72.15	89.75	95.15	53.20	80.40	76.25	56.80
0.4	95	80.05	95.30	97.90	53.55	84.00	82.15	58.55
	99	95.90	98.55	99.70	53.55	88.70	88.95	60.55
	90	63.15	79.40	94.25	42.15	66.55	69.95	46.65
0.6	95	72.30	88.65	97.80	42.40	71.45	75.80	47.85
	99	97.60	96.05	99.80	42.45	77.90	83.25	49.15
	90	26.55	36.30	72.35	12.50	28.80	42.95	14.75
0.8	95	33.30	49.90	86.10	12.55	33.75	48.65	14.95
	99	99.80	70.50	98.05	12.55	42.50	57.15	15.05
	90	76.75	94.65	92.80	67.55	87.80	76.40	69.35
0	95	84.20	96.40	96.60	70.40	91.20	82.75	73.10
	99	95.45	98.45	99.15	71.85	94.50	90.45	76.40
	90	76.25	94.40	92.55	68.25	88.80	74.95	70.60
0.2	95	83.45	96.35	96.45	71.95	91.00	80.05	73.45
	99	95.10	98.80	99.05	73.20	94.10	88.45	77.40
	90	77.05	93.05	95.05	65.70	85.55	76.65	68.75
0.4	95	84.60	95.10	97.70	68.80	88.60	83.75	71.60
	99	96.90	98.60	99.65	69.75	93.15	91.00	74.65
	90	72.85	86.80	95.45	58.25	76.70	74.90	62.70
0.6	95	81.00	90.75	98.05	60.85	81.00	82.10	65.55
	99	98.10	96.95	99.65	61.50	87.40	89.80	67.45
	90	41.75	53.20	83.00	26.20	41.55	51.30	31.40
0.8	95	50.90	61.30	92.00	26.65	47.40	60.35	32.20
	99	99.70	78.55	99.05	26.90	57.55	72.40	32.75

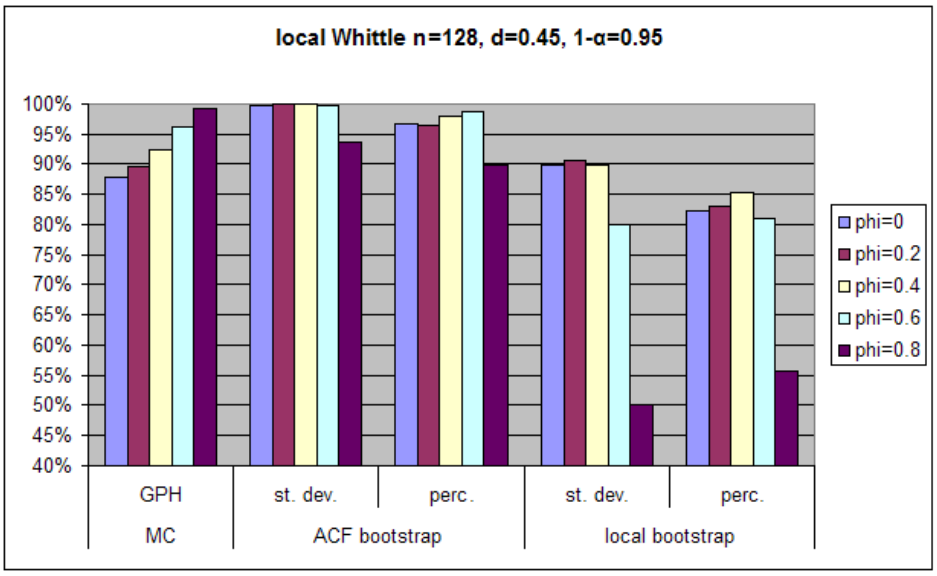
Table 3.15: *Observed coverage (%) of the local Whittle estimator relative to a confidence interval with* $1-\alpha$ *nominal level for* $d = 0.2$*, different values of* $\phi = 0, 0.2, 0.4, 0.6, 0.8$ *computed using Monte Carlo (MC), bootstrap standard deviation (SD), percentile (P), percentile t (PT) methods and two different bootstrap procedures (ACF and local bootstrap): upper part with sample size* $n = 128$*, lower part with sample size* $n = 300$*.*

			ACF bootstrap			local bootstrap		
ϕ		MC	SD	P	PT	SD	P	PT
	90	82.75	99.80	92.50	40.85	87.55	76.85	38.80
0	95	87.80	99.85	96.65	44.00	89.90	82.25	41.60
	99	93.00	100.00	99.35	45.20	92.95	88.45	44.25
	90	86.35	99.80	94.20	41.00	89.25	77.25	39.30
0.2	95	89.55	99.90	96.50	44.25	90.65	83.00	41.60
	99	94.50	100.00	99.00	44.90	92.70	90.50	43.75
	90	89.40	99.85	95.35	38.40	87.85	80.10	38.30
0.4	95	92.35	99.90	97.90	40.25	89.75	85.40	39.95
	99	95.95	100.00	99.40	41.45	92.15	91.75	41.55
	90	94.75	99.65	95.90	27.90	77.05	75.00	27.25
0.6	95	96.30	99.75	98.80	29.35	80.10	81.00	28.35
	99	98.35	99.85	99.90	30.00	82.65	89.20	29.75
	90	99.00	91.00	78.30	6.55	47.20	48.75	6.65
0.8	95	99.25	93.70	89.85	6.70	50.10	55.70	6.70
	99	99.55	97.40	99.15	6.80	53.75	66.70	7.20
	90	86.90	99.55	92.90	46.55	87.70	77.60	46.35
0	95	90.95	99.90	96.80	49.95	90.15	84.35	49.40
	99	95.80	99.95	99.05	53.10	92.70	91.00	52.55
	90	88.60	99.40	94.55	43.30	88.85	78.85	43.15
0.2	95	91.65	99.65	97.50	46.30	90.60	85.20	45.55
	99	96.45	99.80	99.40	50.10	92.85	92.30	48.90
	90	90.15	99.25	95.05	42.90	86.95	79.90	42.65
0.4	95	94.10	99.50	98.35	45.85	89.40	86.10	44.75
	99	97.70	99.75	99.60	47.95	91.90	92.50	47.75
	90	93.35	99.10	94.90	35.05	81.35	77.85	34.90
0.6	95	95.25	99.70	98.20	36.15	84.15	83.75	36.55
	99	97.70	99.90	99.90	37.95	87.60	91.40	37.80
	90	98.90	88.00	82.35	10.60	50.50	53.50	11.10
0.8	95	99.30	91.50	91.85	11.00	54.30	62.75	11.40
	99	99.80	95.50	99.00	11.40	59.60	75.05	11.90

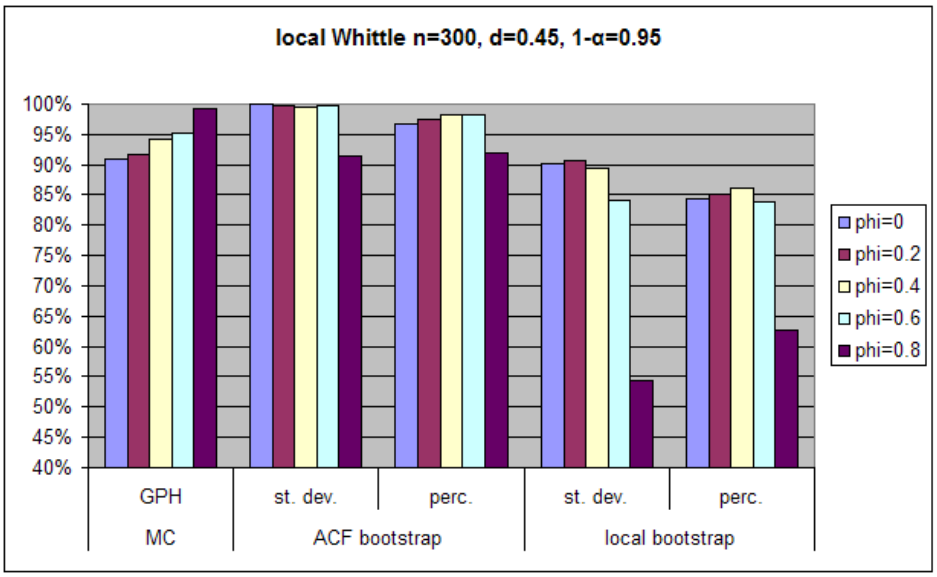
Table 3.16: *Observed coverage (%) of the local Whittle estimator relative to a confidence interval with* $1-\alpha$ *nominal level for* $d = 0.45$*, different values of* $\phi = 0, 0.2, 0.4, 0.6, 0.8$ *computed using Monte Carlo (MC), bootstrap standard deviation (SD), percentile (P), percentile t (PT) methods and two different bootstrap procedures (ACF and local bootstrap): upper part with sample size* $n = 128$*, lower part with sample size* $n = 300$*.*

			ACF bootstrap			local bootstrap		
ϕ		MC	SD	P	PT	SD	P	PT
	90	86.95	95.55	99.00	42.70	79.05	75.95	43.65
0	95	91.05	97.95	99.80	46.80	84.70	81.70	47.00
	99	96.00	99.60	99.95	50.15	89.95	87.60	50.50
	90	84.85	94.75	98.30	42.70	83.50	76.05	43.05
0.2	95	88.85	97.85	99.50	46.50	88.10	80.35	47.05
	99	95.70	99.75	100.00	51.60	93.25	86.75	50.80
	90	80.30	92.20	97.25	45.65	80.75	73.45	46.10
0.4	95	85.20	96.35	99.15	49.65	86.55	78.85	50.20
	99	92.85	99.15	100.00	54.70	93.00	85.05	55.15
	90	67.45	85.05	93.75	47.20	67.70	64.60	48.20
0.6	95	73.85	93.05	97.35	53.00	75.95	69.95	54.65
	99	85.55	98.35	99.75	60.20	85.65	76.65	61.55
	90	23.55	47.65	61.65	23.95	29.30	32.30	24.85
0.8	95	30.65	62.85	78.70	29.45	38.05	38.95	31.35
	99	45.55	84.30	96.05	37.10	55.20	47.00	41.75
	90	88.40	95.80	97.85	49.00	79.50	77.75	50.45
0	95	92.00	98.55	99.80	52.45	84.90	82.60	53.80
	99	96.85	99.95	100.00	56.85	91.35	89.20	57.90
	90	87.40	95.70	97.90	48.15	84.40	75.10	48.35
0.2	95	92.00	98.60	99.55	51.70	89.20	81.45	52.30
	99	96.80	99.80	100.00	56.65	94.50	88.45	56.60
	90	84.70	93.90	97.35	51.05	81.95	74.00	50.45
0.4	95	89.60	97.90	99.15	55.40	87.35	80.30	54.65
	99	95.85	99.70	100.00	60.65	92.95	86.85	60.40
	90	78.05	92.30	95.50	50.80	76.60	71.85	51.75
0.6	95	84.50	96.55	98.30	56.35	83.20	77.70	56.85
	99	93.80	99.50	99.90	64.80	90.45	85.00	64.70
	90	39.90	66.00	73.70	35.40	43.70	43.80	37.45
0.8	95	48.45	79.50	85.45	44.35	53.30	51.60	44.90
	99	67.20	94.45	97.55	60.85	68.90	62.65	60.55

Table 3.17: *Observed coverage (%) of the local Whittle estimator relative to a confidence interval with* $1-\alpha$ *nominal level for* $d=-0.45$*, different values of* $\phi=0, 0.2, 0.4, 0.6, 0.8$ *computed using Monte Carlo (MC), bootstrap standard deviation (SD), percentile (P), percentile t (PT) methods and two different bootstrap procedures (ACF and local bootstrap): upper part with sample size* $n=128$*, lower part with sample size* $n=300$*.*



(a) $n = 128$



(b) $n = 300$

Figure 3.6: *Actual coverage of confidence intervals of local Whittle estimator with confidence* $1 - \alpha = 0.95$*, memory parameter* $d = 0, 0.2, 0.45, -0.45$*, autoregressive parameter* $\phi = 0, 0.2, 0.4, 0.6, 0.8$*, series length* $n = 128, 300, 1000$*, different bootstrap methods (ACF and local bootstrap) and different methods to build the interval (bootstrap standard error, percentile and percentile t).*

Chapter 4

Edgeworth expansion for the sample autocorrelation function

In the previous chapter we developed a new bootstrap method for long memory time series. The ACF bootstrap (as it is called) is based on a result of Ramsey (1974) and generates a surrogate series X_t^* through the sample autocorrelation function $\hat{\rho}_k$, using the Durbin-Levinson algorithm. The validity of the method for fractionally integrated noise FI(d) (see Section 1.2) was supported by an extensive simulation experiment. However it would be interesting to assess formally the validity of ACF bootstrap by showing that the surrogate autocovariance $\hat{\gamma}_k^*$ function and autocorrelation $\hat{\rho}_k^*$ function, i.e., the second order functions of $\{X_t^*\}$, converge in some appropriate mode to the sample autocovariance $\hat{\gamma}_k$ function and autocorrelation $\hat{\rho}_k$ function, respectively, and then also to the theoretical values γ_k and ρ_k.

Throughout this chapter, we distinguish between *weak long memory* (weakly dependent process) with $0 \leq d \leq 0.25$ and *strong long memory* (strongly dependent process) with $0.25 < d < 0.5$, because the value of the memory parameter determines the asymptotic behaviours of sample autocovariance and autocorrelation functions (refer to Section 1.3).

The sample autocovariance and autocorrelation functions of weakly dependent processes are asymptotically Normally distributed, whereas their limiting distribution for strongly dependent processes, the Rosenblatt distribution, is quite complicated and difficult to handle (refer to Section 1.3 for a detailed description of the asymptotics of the sample autocovariance and autocorrelation functions).

We wish to study the normalised quantities $C_k = n^{1-2d}(\hat{\gamma}_k - \gamma_k)$ and $R_k = n^{1-2d}(\hat{\rho}_k - \rho_k)/(1 - \rho_k)$ for strong long memory processes with the help of Edgeworth expansions (Section 4.1). The aim of the study is to understand how the sample autocovariance and autocorrelation functions behave asymptotically for increasing values of d, and to say something about the convergence rate.

In the second part of this chapter we investigate the convergence of the bootstrap autocorrelation function for long memory processes. We compared the sample and the bootstrap autocorrelation functions in terms of standard deviation and bias.

The chapter is organised as follows. In Section 4.1, we briefly introduce Edgeworth

and Cornish-Fisher expansions and we study, by means of simulation, the properties of sample autocovariance and autocorrelation functions of fractionally integrated processes. In Section (4.2) we show that the ACF bootstrap can replicate the second order dependence structure of any long memory processes ($0 < d < 0.5$)) with Gaussian and non-Gaussian innovations. We conclude the chapter and propose some future development in Section 4.3.

4.1 Edgeworth and Cornish-Fisher expansions

Asymptotic Normality is a common and desirable property of estimators. However this is not always the case and Edgeworth expansions can be a useful tools to investigate or correct asymptotic distributions. Not all estimators satisfy assumptions for a central limit theorem or sometimes convergence is so slow that the Normal approximation turns out to be very poor. In this chapter we investigate if it is possible to correct with Edgeworth expansions the asymptotic distribution of the sample autocovariance and autocorrelation functions of fractionally integrated processes with $0.25 < d < 0.5$. We aim to find how many terms of the expansion really influence the convergence and how far from Normality sample autocovariance and autocorrelation functions of strong long memory processes lie.

The idea of an Edgeworth expansion is to approximate the distribution of a statistic by the Normal distribution plus some correction terms, which depend on the cumulants of the statistic of interest. The general formula for an Edgeworth expansion to approximate distributions of estimates $\hat{\theta}$ of unknown quantities θ_0 is given by

$$P\left\{n^{1/2}\left(\frac{\hat{\theta}-\theta_0}{\sigma}\right) \leq x\right\} = \Phi(x) + n^{-1/2}p_1(x)\phi(x) + \ldots + n^{-j/2}p_j(x)\phi(x) + \ldots,$$

where n is the sample size, σ the standard deviation of $\hat{\theta}$, $\Phi(\cdot)$ and $\phi(\cdot)$ are the standard Normal distribution and density function, respectively, $p_j(\cdot)$ is a polynomial depending on cumulants up to order $3j-1$ and is an odd or even function according to whether j is even or odd, respectively (Hall, 1992b).

In our case we want to study the normalised quantities $C_k = n^{1-2d}(\hat{\gamma}_k - \gamma_k)$ and $R_k = n^{1-2d}(\hat{\rho}_k - \rho_k)/(1-\rho_k)$ because we know that they are asymptotically distributed as a Rosenblatt distribution (see Section 1.3). We aim to see how far from Normality these two quantities depart and if the first two terms of Edgeworth expansions correct the non-Normality adequately for the following reasons. Higher order correction terms become very unstable because of all the cumulants we need to estimate and, in any case, Hall (1992b) (pg. 94) already warned that the results based on high order correction are unattainable. In the literature there are not many papers on this topic, besides the work of Hosking (1996). The Rosenblatt distribution is quite complicated and if the sample distributions are not very far from Normality then it could be easier to prove that the ACF bootstrap is a consistent method for long memory Gaussian processes.

From the paper of Hosking (1996), we know that the normalisation constant is n^{1-2d}

and we approximate the asymptotic distributions by Edgeworth expansion as

$$P(C_k^* \leq z) = \Phi(z) + \sum_{j=1}^{\infty} \left(n^{1-2d}\right)^{-j} P_j(z)\phi(z),$$
$$P(R_k^* \leq z) = \Phi(z) + \sum_{j=1}^{\infty} \left(n^{1-2d}\right)^{-j} p_j(z)\phi(z),$$

where $C_k^* = C_k/\sqrt{\text{var}(C_k)}$, $R_k^* = R_k/\sqrt{\text{var}(R_k)}$ and P_j and p_j are defined in Section 1.3. In the simulation study we do not divide C_k and R_k by their standard deviations: only when asymptotic Normality is assessed the variance is given by the well known Bartlett's formulas (see Section 1.3). However, the scale in not important because we compare the Monte Carlo distribution with the Normal distribution and the Normal distribution corrected with Edgeworth expansion in terms of quantiles, thus we do not need to divide by the standard error. For our purpose it is more useful to have an approximation of the quantiles of the distribution to have a graphical insight through Q-Q plots. They can be calculated through Cornish-Fisher expansions, the inverse formula of Edgeworth expansions, given by

$$C_{k,\alpha}^* = z_\alpha + \sum_{j=1}^{\infty} \left(n^{2d-1}\right)^{j} Q_j(z_\alpha), \tag{4.1}$$
$$R_{k,\alpha}^* = z_\alpha + \sum_{j=1}^{\infty} \left(n^{2d-1}\right)^{j} q_j(z_\alpha), \tag{4.2}$$

where $C_{k,\alpha}^*$, $R_{k,\alpha}^*$ and z_α are the α-level quantiles of C_k^*, R_k^* and of the Normal distribution respectively, and Q_j and q_j are defined in Section 1.3.

We compare the distributions of C_k and R_k with three different levels of approximations based on formulas (4.1) and (4.2). The first is the Normal approximation where we shall not consider any correction. In the second and third cases, respectively, we consider up to the first and up to the second terms:

$$\begin{aligned}
C_{k,\alpha}^* &\doteq_1 C_{k,\alpha} = z_\alpha + \left(n^{2d-1}\right) Q_1(z_\alpha), \\
R_{k,\alpha}^* &\doteq_1 R_{k,\alpha} = z_\alpha + \left(n^{2d-1}\right) q_1(z_\alpha), \qquad (4.3) \\
C_{k,\alpha}^* &\doteq_2 C_{k,\alpha} = z_\alpha + \left(n^{2d-1}\right) Q_1(z_\alpha) + \left(n^{2d-1}\right)^2 Q_2(z_\alpha), \\
R_{k,\alpha}^* &\doteq_2 R_{k,\alpha} = z_\alpha + \left(n^{2d-1}\right) q_1(z_\alpha) + \left(n^{2d-1}\right)^2 q_2(z_\alpha). \qquad (4.4)
\end{aligned}$$

The coefficients Q_1 and Q_2 are given by

$$\begin{aligned}
Q_1(x) &= -P_1(x) \\
Q_2(x) &= P_1(x)P_1'(x) - \frac{1}{2}xP_1(x)^2 - P_2(x),
\end{aligned}$$

where P_1 and P_2 are the Edgeworth expansion's coefficients with

$$P_1(x) = -\frac{1}{6}k_3(x^2-1)$$
$$P_2(x) = -x\left\{\frac{1}{24}k_4(x^2-3) + \frac{1}{72}k_3^2(x^4-10x^2+15)\right\}$$

and k_3 and k_4 are the third and fourth order cumulants

$$k_3(\hat{\gamma}_k) = E\{(\hat{\gamma}_k - E[\hat{\gamma}_k])^3\} \tag{4.5}$$
$$k_4(\hat{\gamma}_k) = E\{(\hat{\gamma}_k - E[\hat{\gamma}_k])^4\} - 3\mathrm{Var}\{\hat{\gamma}_k\}^2. \tag{4.6}$$

The same relationship is valid between q_1, q_2 and p_1, p_2 with the third and fourth cumulants of $\hat{\rho}_k$.

For more details about *Edgeworth* and *Cornish-Fisher* expansions refer to Hall (1992b) and Barndorff-Nielsen and Cox (1989).

4.1.1 Numerical Experiments

In this section we present the results of a wide numerical experiment. The aim of the experiment is to compare the distribution of the normalised sample autocovariance function, C_k $(k = 0, 1, 2, 5, 10)$, and sample autocorrelation function, R_k $(k = 1, 2, 5, 10)$, of a strong long memory process with three different theoretical distributions: the Normal distribution and two corrected distributions, calculated using equations (4.3) and (4.4).

The Monte Carlo experiment consists of generating $S = 20000$ series of length $n = 300, 1000$ and 2000 for different values of the memory parameter $d = 0.26, 0.27, \ldots, 0.49$. We use simulated data for two purposes. Firstly, we calculate the normalised sample autocovariance and autocorrelation functions, C_k and R_k, for each series, obtaining estimates of their distributions. Secondly, we thus obtain reliable estimates of the third and fourth cumulants of C_k and R_k, based on formulas (4.5) and (4.6). The theoretical third and fourth cumulants are too burdensome to evaluate numerically, being the solutions of the multiple integral of equation (1.17). As we already explained in Section 4.1, C_k and R_k were not normalised in the simulations by their standard errors, because we are interested in comparing the quantiles of the distributions, so the scale is not important.

We preliminarily explore the data through Q-Q plots of the standard Normal distribution, of ${}_1C_k$ and of ${}_2C_k$ versus C_k. In the case of the autocovariance function, see Figures (4.1)-(4.3), the data are clearly non-Normal, and non-Normality is more pronounced for larger values of the memory parameter. On the other hand, it seems that for longer series the distribution of the sample autocovariance function gets closer to the Normal distribution. When we analyse the effects of the Cornish-Fisher corrections, we find that the first correction has a visible impact: the Normal distribution with one correction term is closer to the sample distribution. The second correction has almost no impact, suggesting that only the first correction is significant: we do not draw the second order correction in the graphs because it would superimpose the first order correction. For longer series, when the distribution of the sample autocovariance function is closer to normality, the contribution

of Cornish-Fisher is smaller. Similar Q-Q plots, given in Figures (4.4)-(4.6), of the autocorrelation function seem noticeably different. The sample autocorrelation function has a distribution quite close to Normality, and a slight deviation can be observed for larger values of the parameter, i.e., $d = 0.45, 0.49$. The contributions of both the first and the second correction terms seemed to be irrelevant from a graphical point of view. It must be underlined that, for $d = 0.49$, the convergence of the autocorrelation function is so slow that the correction is not well estimated and the first order correction is behaving in a unusual way. Another indication of the slow convergence rate of the sample autocovariance and autocorrelation functions can be noticed in the ordinates of the figures: it should be centred around zero, however when the value of the memory parameter d increases, the bias becomes huge for both the sample autocovariance and the autocorrelation functions. This issue was already investigated by Newbold and Agiakloglou (1993).

An interesting insight of the sample distribution is given by Figure 4.7: for increasing values of d (abscissa) and different sample sizes, there is the estimate of the sample kurtosis (part a) and of the sample skewness (part b) for the autocovariance and autocorrelation functions. The two quantities are positively correlated with the memory parameter d in the case of the autocovariance function meaning that the sample distribution becomes more skewed and heavy tailed, whereas they remain approximately constant and with values closed to the Normal distribution in the case of the sample autocorrelation function.

As a second step we investigate if the improvements pointed out with the graphical analysis are statistically significant. We studied three standard linear regression models, where the dependent variable Y is the quantile of the sample autocovariance (autocorrelation) function while the covariate changes:

$$\begin{aligned} &\text{Model 1:} \quad Y = \alpha_0 + \alpha_1 X_1 + \varepsilon_1, \\ &\text{Model 2:} \quad Y = \beta_0 + \beta_1 X_2 + \varepsilon_2, \\ &\text{Model 3:} \quad Y = \tau_0 + \tau_1 X_3 + \varepsilon_3, \end{aligned} \tag{4.7}$$

where X_1 is the quantile of the Normal distribution, whereas X_2 and X_3 are the quantiles of the Normal distribution corrected with one and two terms, respectively, of the Cornish-Fisher expansion. As expected the R squared values are all close to unity with $R^2(Y, X_1) < R^2(Y, X_2) < R^2(Y, X_3)$. We cannot use the F test to compare the three models because they are not nested. An alternative is offered by the Cox test. This test statistic is Normally distributed if the errors ε_i, $i = 1, 2, 3$, are Gaussian. Even though this is not the case, in Appendix A we introduce briefly the Cox test and we show, by means of simulation, that Normality of the test is not violated. We refer to the Cox test with $\hat{S}_{k,1}$ and $\hat{s}_{k,1}$ when we compare the first two models, $\hat{S}_{k,2}$ and $\hat{s}_{k,2}$ for the second and the third, where k indicates the lag, capital letter for sample autocovariance function and lower case letter for sample autocorrelation.

In Tables 4.1-4.3 we report the observed results of the test for the sample autocovaraince function: for each value of the long memory parameter, in the first column there is the value of the test $\hat{S}_{k,1}$ ($\hat{s}_{k,1}$) when we compare the Normal approximation with the first order approximation (first and second model of Equations (4.7)), while the second

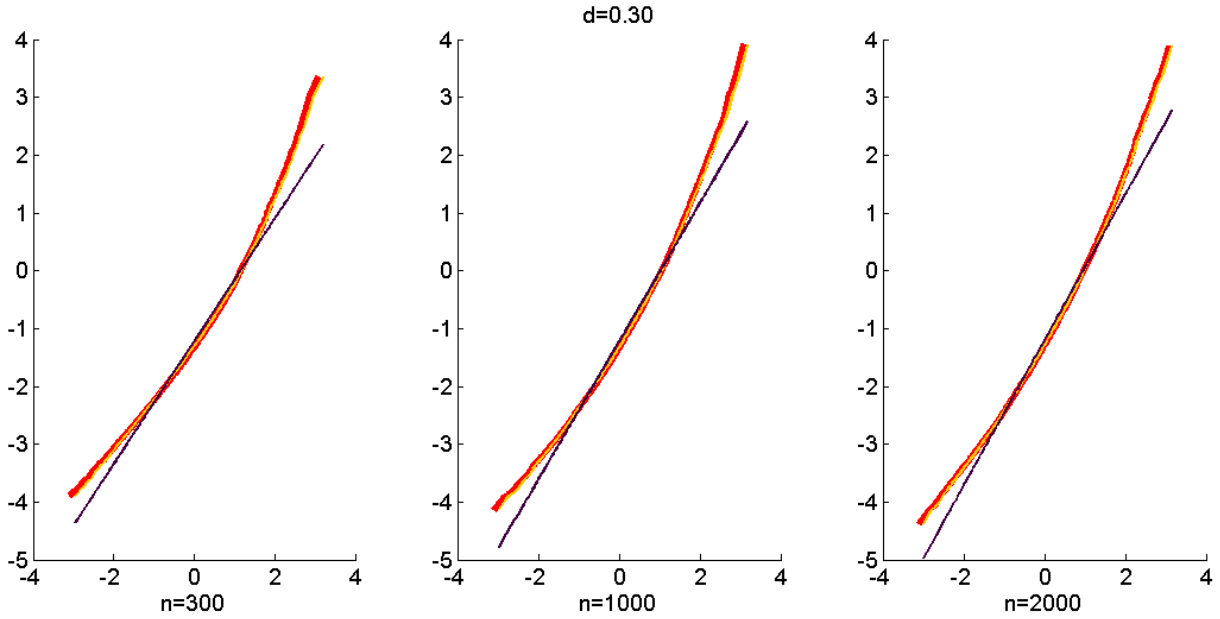


Figure 4.1: *Q-Q plots of the sample autocovariance function for the memory parameter* $d = 0.30$ *and for different sample sizes,* $n = 300, 1000, 2000$*; the red line is the Normal approximation, the yellow line is the first order approximation of the Cornish-Fisher expansion (see equation (4.1)) and the black line is the benchmark.*

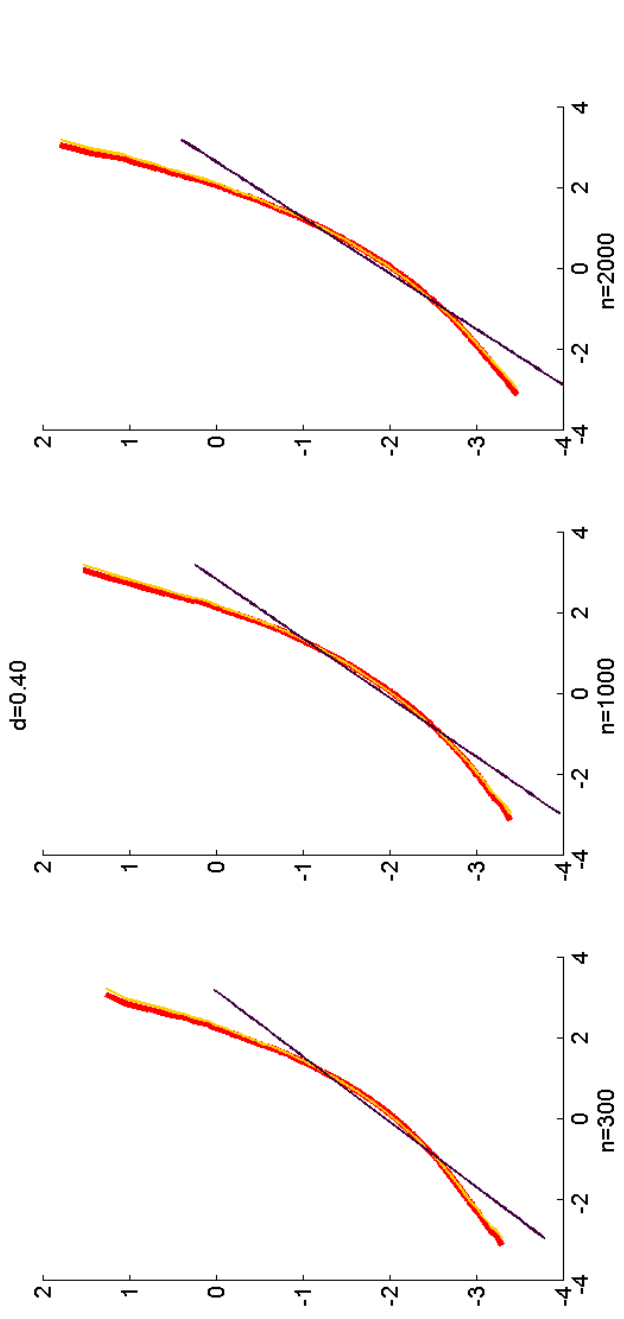


Figure 4.2: *Q-Q plots of the sample autocovariance function for the memory parameter $d = 0.40$ and for different sample sizes, $n = 300, 1000, 2000$; the red line is the Normal approximation, the yellow line is the first order approximation of the Cornish-Fisher expansion (see equation (4.1)) and the black line is the benchmark.*

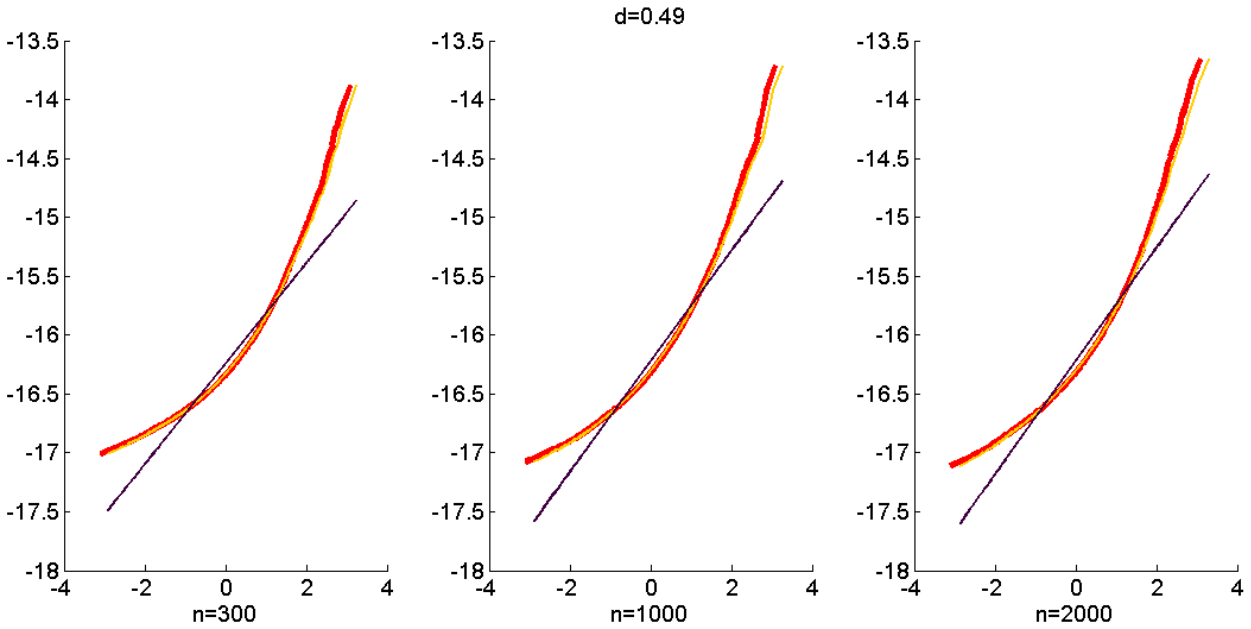


Figure 4.3: *Q-Q plots of the sample autocovariance function for the memory parameter* $d = 0.49$ *and for different sample sizes,* $n = 300, 1000, 2000$*; the red line is the Normal approximation, the yellow line is the first order approximation of the Cornish-Fisher expansion (see equation (4.1)) and the black line is the benchmark.*

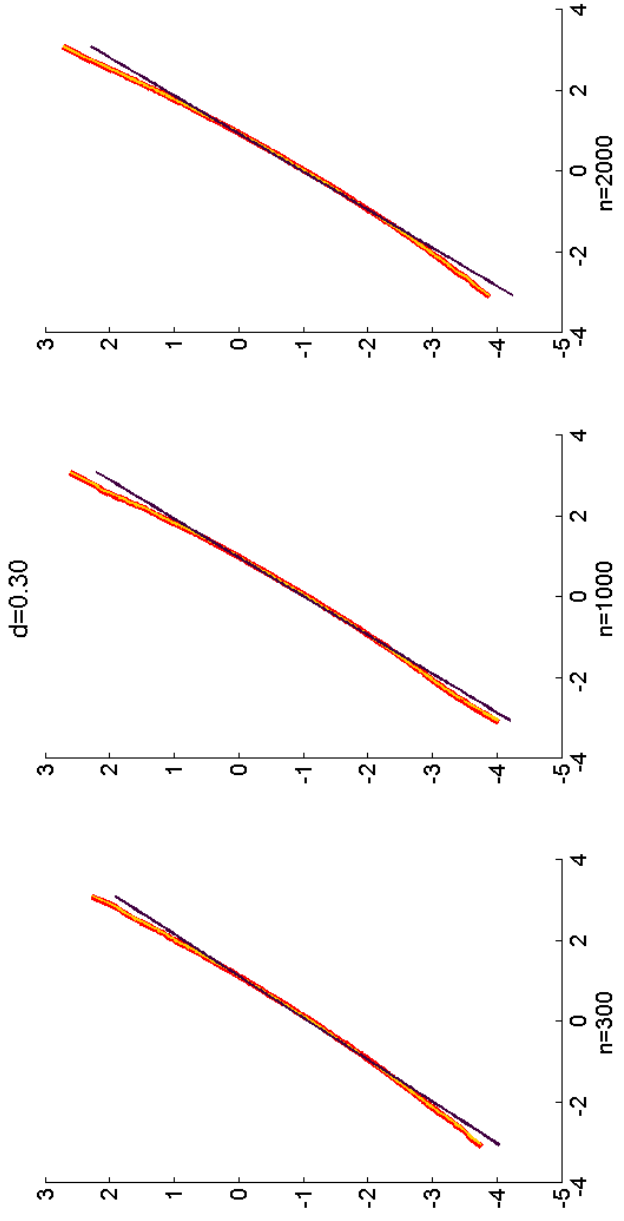


Figure 4.4: *Q-Q plots of the sample autocorrelation function for the memory parameter $d = 0.30$ and for different sample sizes, $n = 300, 1000, 2000$; the red line is the Normal approximation, the yellow line is the first order approximation of the Cornish-Fisher expansion (see equation (4.2)) and the black line is the benchmark.*

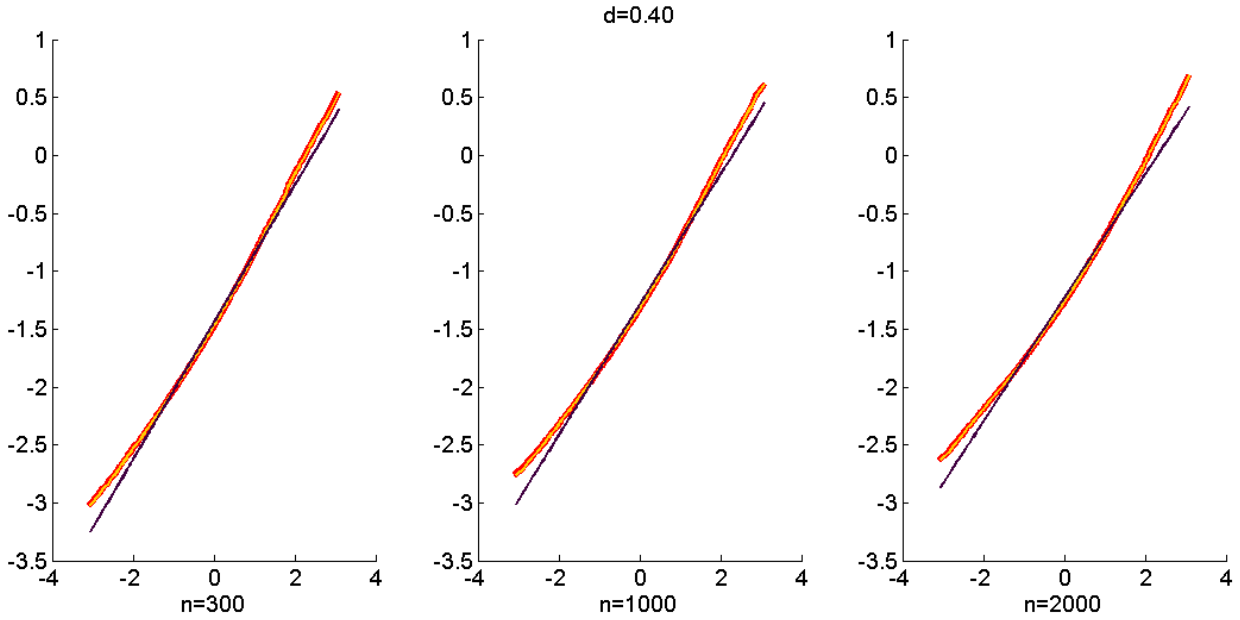


Figure 4.5: *Q-Q plots of the sample autocorrelation function for the memory parameter* $d = 0.40$ *and for different sample sizes,* $n = 300, 1000, 2000$*; the red line is the Normal approximation, the yellow line is the first order approximation of the Cornish-Fisher expansion (see equation (4.2)) and the black line is the benchmark.*

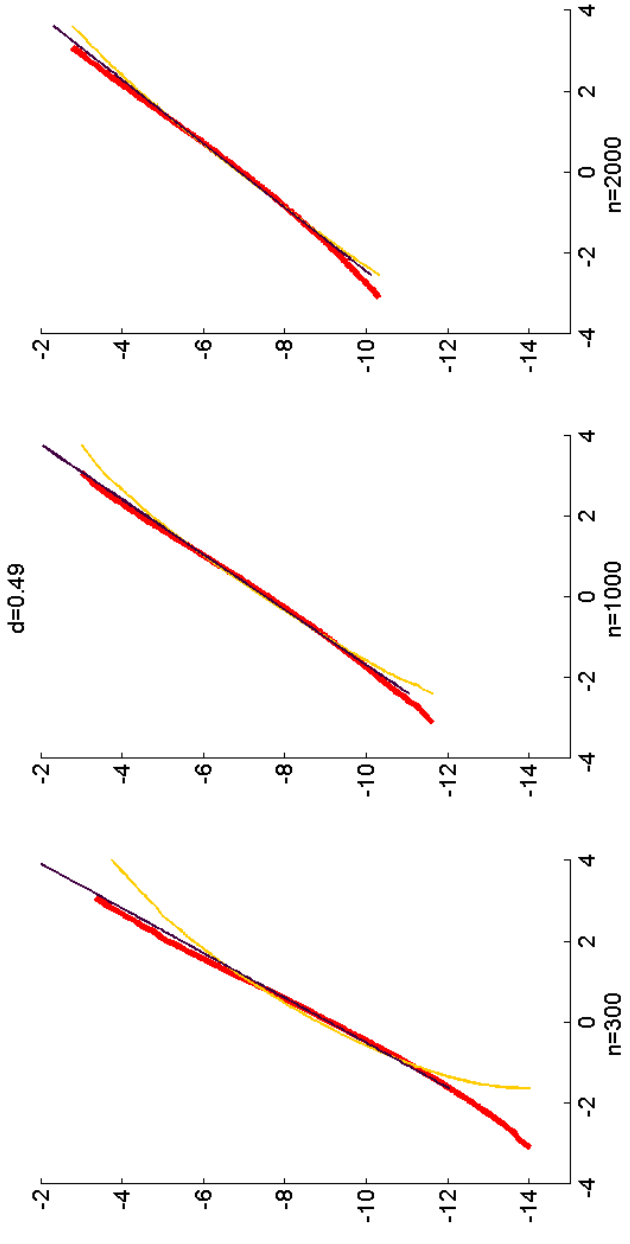
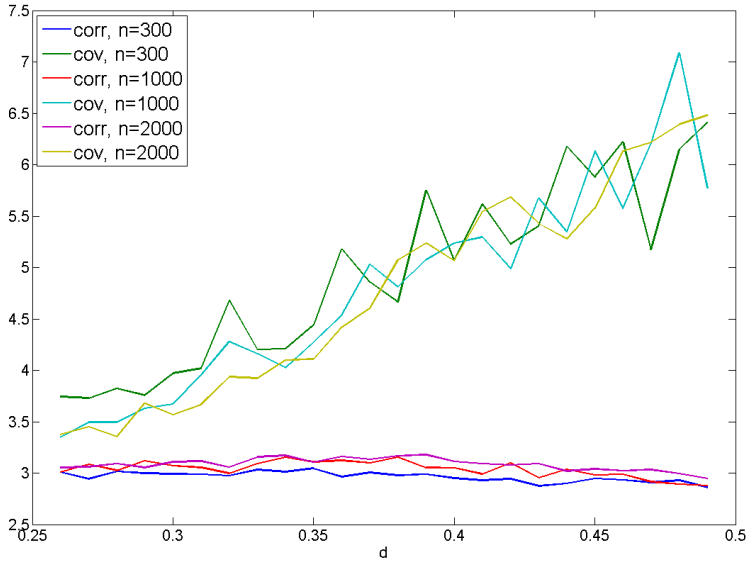
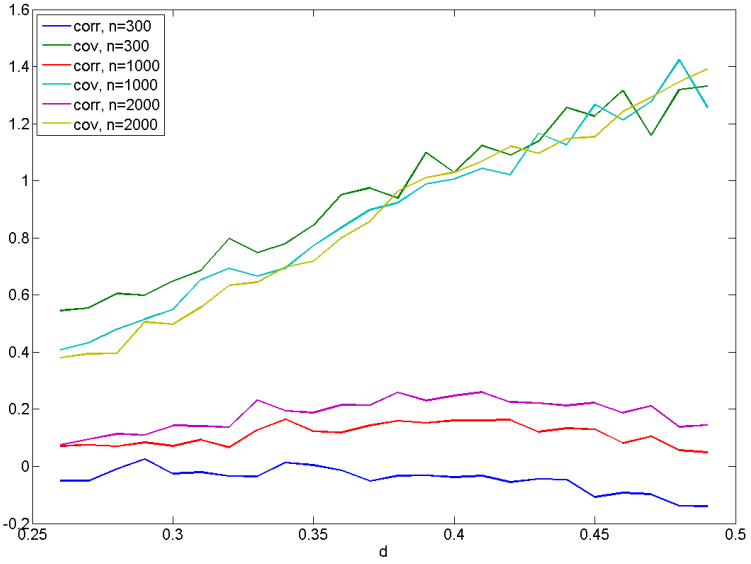


Figure 4.6: *Q-Q plots of the sample autocorrelation function for the memory parameter $d = 0.49$ and for different sample sizes, $n = 300, 1000, 2000$; the red line is the Normal approximation, the yellow line is the first order approximation of the Cornish-Fisher expansion (see equation (4.2) and the black line is the benchmark.*



(a) kurtosis



(b) skewness

Figure 4.7: *Sample kurtosis* $k_4(x) = E\{(x - E[x])^4\} - 3Var\{x\}^2$ *and sample skewness* $k_3(x) = E\{(x - E[x])^3\}$ *for increasing values of* $d = 0.26, 0.27, \ldots, 0.49$ *of the sample autocorrelation and autocovariance function for different series length* $n = 300, 1000, 2000$.

column reports the values of the test $\hat{S}_{k,2}$ ($\hat{s}_{k,2}$) when we compare the first and the second order approximations (second and third model of Equations (4.7)). These values have to be compared with the 5% quantile of the Normal distribution, i.e., ± 1.96. For the autocovariance function the first order correction term is significant, as we already noticed in the Q-Q plots. The second order term is not significant for values of d smaller than 0.42. However, the correction of the second term becomes less significant, when increasing the sample size, suggesting that, asymptotically, the second term is negligible for any value of the memory parameter d.

In Tables 4.4-4.6 we report the observed results of the test for the sample autocorrelation function. The results for the sample autocorrelation function are controversial. The second order correction is not as significant as for the autocovariance function. From the Q-Q plot it seemed that the sample autocorrelation function is quite close to Normality; however the observed values of the Cox test indicate that the first order correction is very significant.

We conclude this section by noting that, from the Q-Q plots, the sample autocorrelation function seems closer to Normality than the sample autocovariance function; however the first order correction is significant in both cases. This is not a surprise since the sample distributions are very asymmetric and the first correction depends on moments up to the third order and corrects asymmetry, whereas the second order correction depends on the moments up to the fourth order and correct tails. As regards the convergence rate of the sample autocovariance function, it really depends on the value of d: the larger it is the slower the convergence. Increasing the sample size diminishes the importance of the correction.

4.2 Comparing sample and bootstrap autocorrelation functions

The previous section gave a detailed insight into the asymptotic behaviour of the sample distribution of the second order dependence structure of a long memory time series. Now that we can compare the sample autocovariance and autocorrelation functions with their bootstrap estimates, we now wish to show that the ACF bootstrap is consistent for long memory processes because it can replicate their second order structure. This statement is supported by the observed average values and standard deviations of $\hat{\gamma}_k$, $\hat{\gamma}_k^*$, $\hat{\rho}_k$ and $\hat{\rho}_k^*$ and by comparing their distributions.

We run a wide simulation experiment and compare the sample autocorrelation function with the bootstrap autocorrelation function. For sake of completeness we consider the range of values of $d = 0.1, 0.15, 0.2, \ldots, 0.45$ and increasing series lengths $n = 100, 200, \ldots, 2000$. We repeat the same experiment with innovations distributed as Chi-squared with one degree of freedom, and Student t with four and six degrees of freedom, to support our belief that ACF bootstrap can replicate the second order structure of a process X_t even when the innovations are non-Gaussian.

In Figures 4.8 and 4.9 we show the empirical densities of the sample and bootstrap variance and autocovariance function. From the densities of the variance, it is interesting to notice that the distribution is quite skewed and for large values of d it is also quite biased.

Increasing the sample size from 1000 to 2000 the density moves to the right significantly. Also for the autocorrelation at lag one, there is bias for $d = 0.4, 0.45$. In both cases the bootstrap densities seem to follow the Monte Carlo pattern, but there are fewer extreme values and the distributions are a bit more concentrated.

The results for all sample sizes considered are given in Tables 4.10 and 4.11. We report in the first column the difference between the Monte Carlo sample autocorrelation function and the bootstrap estimate, $\hat{\rho}_k - \hat{\rho}_k^*$. The bootstrap variance*** is biased downward. The bias is positively correlated with the strength of the long memory and negatively correlated with the sample size. However, for sample sizes larger than 1000 the bias is always less than 2%. In terms of standard deviation, using the bootstrap technique there is on average an improvement and its order is positively correlated with the value of the memory parameter.

4.3 Conclusions

In this chapter we investigated deeply the asymptotic behaviour of the sample autocovariance and autocorrelation functions of a strong long memory Gaussian process, i.e., $0.25 < d < 0.5$. We had some nice results, as follows.

The sample autocovariance function is not Normal, however we have a significant improvement correcting the Normal distribution with the first term of the Cornish-Fisher expansion. Deviation from Normality gets smaller when increasing the sample size, whereas it is more evident for stronger long memory processes; also the second order correction term has a more important contribution.

The sample autocorrelation function is closer to Normality and small deviation from Normality can be detected for very large values of the memory parameter, i.e., $d > 0.4$. In this case the Cornish-Fisher corrections seem not to give any contribution from a graphical point of view (Figures (4.4)-(4.6), even though a significant contribution is detected by the Cox test.

By means of simulation, we showed that the ACF bootstrap can replicate the second order structure of a long memory Gaussian process. The method gave satisfactory results also with non-Gaussian processes, indicating that at least the second order structure is preserved, and asymmetry or extreme values in the innovation distribution do not affect the performance dramatically.

This work is open to future interesting developments:

- it is a good starting point to try to prove theoretically the consistency of ACF bootstrap, and
- given the good results on non-Gaussian time series, it is probably possible to prove that the ACF bootstrap replicates second order structure of linear processes no matter what innovations drive the data generating process.

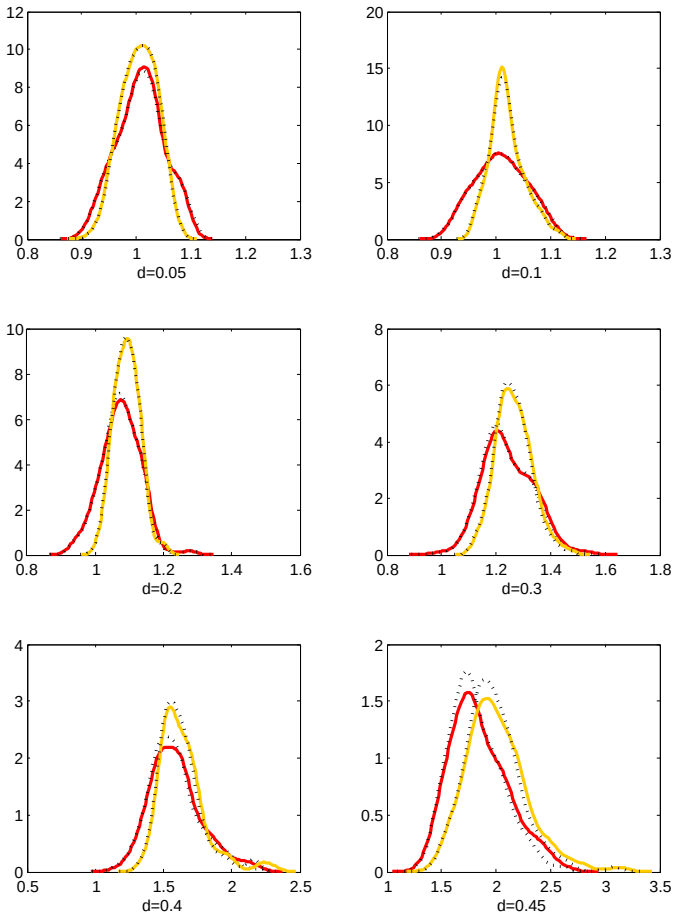


Figure 4.8: *Plots of the density of the sample and the bootstrap variance for different values of* $d = 0.05, 0.1, 0.2, 0.3, 0.4, 0.45$ *and different series length. The red line is for the sample autocorrelation function with* $n = 1000$*, the yellow line is for* $n = 2000$*. The dotted lines are the bootstrap autocorrelation densities.*

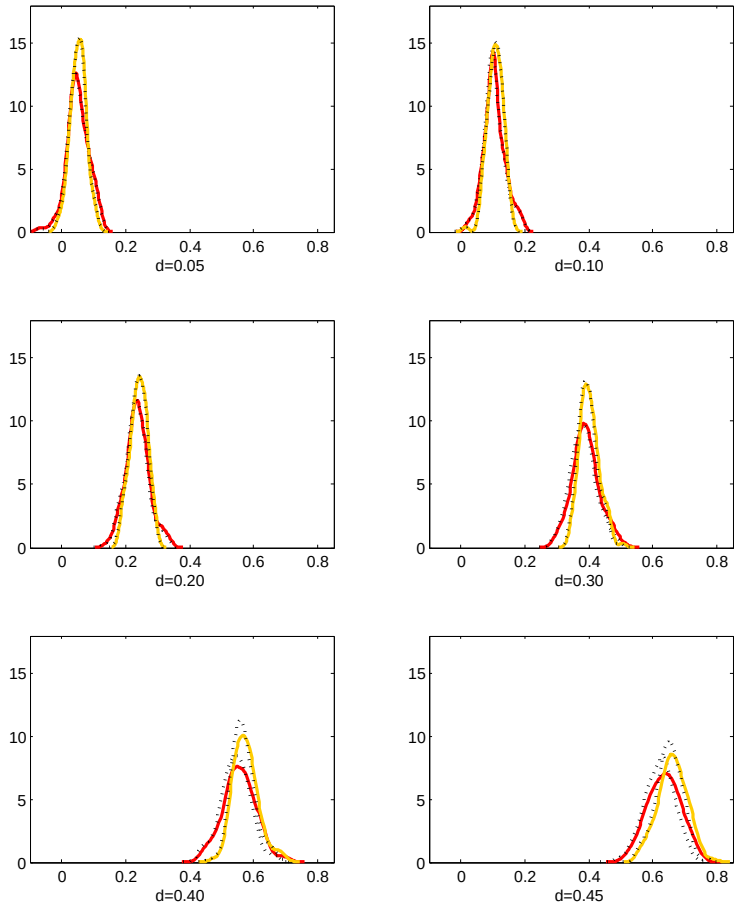


Figure 4.9: *Plots of the density of the sample autocorrelation function and the bootstrap autocorrelation function at lag $k = 1$ for different values of $d = 0.05, 0.1, 0.2, 0.3, 0.4, 0.45$ and different series length. The red line is for the sample autocorrelation function with $n = 1000$, the yellow line is for $n = 2000$. The dotted lines are the bootstrap autocorrelation densities.*

lag	$d = 0.26$		$d = 0.27$		$d = 0.28$		$d = 0.29$	
0	-35.04	*0.40*	-32.02	*0.27*	-30.56	*0.20*	-30.13	*0.13*
1	-29.96	*0.20*	-29.64	*0.14*	-29.56	*0.07*	-29.31	*0.03*
2	-29.44	*0.17*	-29.24	*0.10*	-29.33	*-0.01*	-29.34	*0.04*
5	-28.34	*0.03*	-28.60	*-0.08*	-28.71	*-0.17*	-28.93	*-0.05*
10	-27.24	*-0.19*	-28.06	*-0.21*	-27.81	*-0.38*	-28.07	*-0.27*
lag	$d = 0.30$		$d = 0.31$		$d = 0.32$		$d = 0.33$	
0	-29.82	*-0.03*	-29.22	*-0.14*	-29.28	*-0.38*	-29.17	*-0.29*
1	-29.47	*-0.07*	-29.17	*-0.12*	-28.78	*-0.47*	-29.33	*-0.24*
2	-29.27	*-0.12*	-29.01	*-0.21*	-28.72	*-0.54*	-29.26	*-0.31*
5	-28.79	*-0.24*	-28.95	*-0.27*	-28.69	*-0.70*	-29.06	*-0.43*
10	-28.26	*-0.35*	-28.06	*-0.54*	-28.28	*-0.79*	-28.64	*-0.56*
lag	$d = 0.34$		$d = 0.35$		$d = 0.36$		$d = 0.37$	
0	-29.30	*-0.21*	-29.08	*-0.40*	-28.58	*-0.89*	-29.01	*-0.63*
1	-29.32	*-0.25*	-29.16	*-0.39*	-28.57	*-0.93*	-29.14	*-0.63*
2	-29.29	*-0.32*	-29.21	*-0.41*	-28.64	*-0.94*	-29.14	*-0.68*
5	-29.18	*-0.37*	-28.97	*-0.57*	-28.68	*-0.99*	-28.95	*-0.86*
10	-29.03	*-0.50*	-28.64	*-0.72*	-28.50	*-1.15*	-28.81	*-1.01*
lag	$d = 0.38$		$d = 0.39$		$d = 0.40$		$d = 0.41$	
0	-29.20	*-0.64*	-28.61	*-1.36*	-28.96	*-1.09*	-28.91	*-1.58*
1	-29.33	*-0.61*	-28.60	*-1.44*	-29.09	*-1.04*	-29.13	*-1.38*
2	-29.33	*-0.67*	-28.78	*-1.40*	-29.22	*-0.98*	-29.25	*-1.33*
5	-29.28	*-0.79*	-28.47	*-1.64*	-29.15	*-1.14*	-29.20	*-1.48*
10	-29.04	*-1.03*	-28.39	*-1.70*	-28.95	*-1.35*	-29.05	*-1.78*
lag	$d = 0.42$		$d = 0.43$		$d = 0.44$		$d = 0.45$	
0	-29.03	*-1.30*	-28.98	*-1.54*	-28.88	*-2.19*	-29.09	*-2.11*
1	-29.19	*-1.25*	-29.13	*-1.48*	-29.02	*-2.12*	-29.27	*-1.91*
2	-29.29	*-1.23*	-29.19	*-1.46*	-29.06	*-2.08*	-29.30	*-1.94*
5	-29.32	*-1.32*	-29.19	*-1.56*	-29.12	*-2.11*	-29.42	*-1.90*
10	-29.17	*-1.56*	-29.31	*-1.57*	-29.11	*-2.14*	-29.44	*-1.98*
lag	$d = 0.46$		$d = 0.47$		$d = 0.48$		$d = 0.49$	
0	-28.91	*-2.62*	-29.50	*-1.57*	-29.17	*-2.74*	-29.22	*-3.11*
1	-29.07	*-2.45*	-29.58	*-1.52*	-29.25	*-2.62*	-29.33	*-2.93*
2	-29.14	*-2.36*	-29.69	*-1.45*	-29.34	*-2.52*	-29.34	*-2.93*
5	-29.25	*-2.33*	-29.73	*-1.53*	-29.43	*-2.57*	-29.46	*-2.82*
10	-29.12	*-2.65*	-29.69	*-1.74*	-29.43	*-2.72*	-29.57	*-2.81*

Table 4.1: *Cox test to compare the Monte Carlo distribution of the sample autocovariance function of a long memory process (equation (1.4)) for different values of the memory parameter d at different lags, $k = 0, 1, 2, 5, 10$, for $n = 300$.*

lag	$d = 0.26$		$d = 0.27$		$d = 0.28$		$d = 0.29$	
0	-29.96	*0.18*	-29.45	*0.09*	-29.44	*0.10*	-29.14	*0.05*
1	-29.50	*0.12*	-29.32	*0.07*	-29.33	*0.08*	-29.26	*0.04*
2	-29.38	*0.12*	-29.37	*0.07*	-29.20	*0.06*	-29.33	*0.02*
5	-29.30	*0.10*	-28.58	*0.02*	-29.36	*0.05*	-28.86	*0.00*
10	-29.06	*0.09*	-28.26	*-0.05*	-28.99	*0.04*	-28.73	*-0.03*
lag	$d = 0.30$		$d = 0.31$		$d = 0.32$		$d = 0.33$	
0	-29.10	*0.02*	-28.62	*-0.06*	-28.74	*-0.19*	-29.12	*-0.21*
1	-29.22	*0.01*	-28.96	*-0.04*	-29.01	*-0.18*	-29.04	*-0.24*
2	-29.40	*0.01*	-28.66	*-0.09*	-29.12	*-0.16*	-29.01	*-0.24*
5	-29.40	*-0.02*	-28.87	*-0.07*	-28.92	*-0.18*	-29.28	*-0.21*
10	-29.23	*-0.05*	-28.77	*-0.10*	-29.23	*-0.19*	-29.27	*-0.21*
lag	$d = 0.34$		$d = 0.35$		$d = 0.36$		$d = 0.37$	
0	-29.29	*-0.15*	-28.92	*-0.28*	-29.04	*-0.41*	-29.12	*-0.65*
1	-29.40	*-0.16*	-29.19	*-0.26*	-29.14	*-0.37*	-29.10	*-0.67*
2	-29.53	*-0.12*	-29.30	*-0.24*	-29.23	*-0.36*	-29.22	*-0.66*
5	-29.46	*-0.15*	-29.17	*-0.30*	-29.53	*-0.31*	-29.44	*-0.57*
10	-29.43	*-0.21*	-29.24	*-0.31*	-29.12	*-0.48*	-29.35	*-0.60*
lag	$d = 0.38$		$d = 0.39$		$d = 0.40$		$d = 0.41$	
0	-29.29	*-0.52*	-28.86	*-0.80*	-28.51	*-1.12*	-29.09	*-1.05*
1	-29.36	*-0.52*	-28.89	*-0.82*	-28.73	*-1.05*	-29.21	*-1.01*
2	-29.37	*-0.53*	-28.99	*-0.80*	-28.81	*-1.02*	-29.29	*-0.96*
5	-29.36	*-0.59*	-29.18	*-0.73*	-28.98	*-0.98*	-29.27	*-1.04*
10	-29.47	*-0.58*	-29.31	*-0.69*	-29.01	*-0.99*	-29.33	*-1.04*
lag	$d = 0.42$		$d = 0.43$		$d = 0.44$		$d = 0.45$	
0	-29.23	*-0.98*	-28.97	*-1.52*	-29.34	*-1.28*	-29.14	*-1.94*
1	-29.20	*-1.01*	-29.06	*-1.46*	-29.38	*-1.29*	-29.22	*-1.89*
2	-29.25	*-1.01*	-29.09	*-1.46*	-29.41	*-1.27*	-29.19	*-1.96*
5	-29.36	*-1.02*	-29.25	*-1.40*	-29.52	*-1.20*	-29.32	*-1.85*
10	-29.48	*-0.98*	-29.39	*-1.34*	-29.66	*-1.13*	-29.39	*-1.85*
lag	$d = 0.46$		$d = 0.47$		$d = 0.48$		$d = 0.49$	
0	-29.16	*-1.85*	-29.07	*-2.57*	-28.86	*-3.52*	-29.45	*-2.30*
1	-29.19	*-1.83*	-29.12	*-2.53*	-28.87	*-3.50*	-29.47	*-2.30*
2	-29.21	*-1.84*	-29.10	*-2.56*	-28.92	*-3.43*	-29.48	*-2.28*
5	-29.26	*-1.83*	-29.09	*-2.62*	-29.03	*-3.28*	-29.51	*-2.27*
10	-29.30	*-1.85*	-29.22	*-2.52*	-29.08	*-3.19*	-29.57	*-2.27*

Table 4.2: *Cox test to compare the Monte Carlo distribution of the sample autocovariance function of a long memory process (equation (1.4)) for different values of the memory parameter d at different lags, $k = 0, 1, 2, 5, 10$, for $n = 1000$.*

lag	$d = 0.26$		$d = 0.27$		$d = 0.28$		$d = 0.29$	
0	-29.13	*0.12*	-28.88	*0.07*	-28.65	*0.07*	-28.67	*0.04*
1	-29.07	*0.10*	-28.93	*0.06*	-29.29	*0.06*	-28.81	*0.02*
2	-29.20	*0.10*	-28.61	*0.07*	-29.24	*0.05*	-28.87	*0.02*
5	-29.06	*0.09*	-28.70	*0.04*	-28.91	*0.06*	-28.96	*0.01*
10	-28.46	*0.07*	-29.46	*0.04*	-29.59	*0.04*	-28.95	*-0.01*
lag	$d = 0.30$		$d = 0.31$		$d = 0.32$		$d = 0.33$	
0	-28.76	*0.03*	-28.96	*0.00*	-29.09	*-0.07*	-28.75	*-0.12*
1	-29.01	*0.02*	-29.20	*0.00*	-29.15	*-0.07*	-29.09	*-0.10*
2	-29.19	*0.02*	-29.07	*-0.02*	-29.13	*-0.07*	-29.11	*-0.10*
5	-29.02	*0.01*	-29.05	*-0.03*	-29.23	*-0.08*	-28.93	*-0.13*
10	-29.44	*0.00*	-28.99	*-0.05*	-29.47	*-0.07*	-29.14	*-0.12*
lag	$d = 0.34$		$d = 0.35$		$d = 0.36$		$d = 0.37$	
0	-28.90	*-0.20*	-29.05	*-0.20*	-28.88	*-0.34*	-29.18	*-0.38*
1	-29.29	*-0.15*	-29.15	*-0.20*	-29.10	*-0.31*	-29.13	*-0.40*
2	-29.20	*-0.17*	-29.15	*-0.20*	-29.20	*-0.31*	-29.25	*-0.40*
5	-29.23	*-0.19*	-29.08	*-0.24*	-29.13	*-0.35*	-29.31	*-0.38*
10	-29.15	*-0.21*	-29.16	*-0.24*	-29.30	*-0.31*	-29.35	*-0.41*
lag	$d = 0.38$		$d = 0.39$		$d = 0.40$		$d = 0.41$	
0	-28.77	*-0.65*	-28.71	*-0.78*	-29.00	*-0.75*	-29.00	*-1.06*
1	-28.76	*-0.66*	-28.74	*-0.78*	-29.04	*-0.73*	-29.01	*-1.07*
2	-28.96	*-0.61*	-28.83	*-0.75*	-29.08	*-0.74*	-29.01	*-1.06*
5	-29.08	*-0.57*	-28.83	*-0.77*	-29.18	*-0.72*	-29.04	*-1.09*
10	-29.14	*-0.58*	-28.81	*-0.82*	-29.24	*-0.72*	-29.08	*-1.08*
lag	$d = 0.42$		$d = 0.43$		$d = 0.44$		$d = 0.45$	
0	-28.95	*-1.26*	-29.22	*-1.23*	-29.18	*-1.22*	-29.14	*-1.61*
1	-29.01	*-1.23*	-29.25	*-1.20*	-29.22	*-1.21*	-29.19	*-1.59*
2	-29.05	*-1.21*	-29.32	*-1.16*	-29.24	*-1.21*	-29.18	*-1.60*
5	-29.12	*-1.18*	-29.38	*-1.12*	-29.27	*-1.21*	-29.15	*-1.66*
10	-29.15	*-1.18*	-29.50	*-1.07*	-29.38	*-1.15*	-29.24	*-1.60*
lag	$d = 0.46$		$d = 0.47$		$d = 0.48$		$d = 0.49$	
0	-29.31	*-2.00*	-29.05	*-2.42*	-29.08	*-2.82*	-28.98	*-3.25*
1	-29.36	*-1.94*	-29.08	*-2.40*	-29.12	*-2.77*	-29.02	*-3.19*
2	-29.39	*-1.93*	-29.13	*-2.35*	-29.12	*-2.78*	-29.04	*-3.18*
5	-29.47	*-1.81*	-29.13	*-2.36*	-29.13	*-2.77*	-29.05	*-3.17*
10	-29.55	*-1.76*	-29.24	*-2.27*	-29.25	*-2.64*	-29.09	*-3.16*

Table 4.3: *Cox test to compare the Monte Carlo distribution of the sample autocovariance function of a long memory process (equation (1.4)) for different values of the memory parameter d at different lags, $k = 0, 1, 2, 5, 10$, for $n = 2000$.*

lag	$d = 0.26$		$d = 0.27$		$d = 0.28$		$d = 0.29$	
1	-21.09	*0.00*	-24.99	*-0.57*	8.99	*-0.66*	-17.71	*0.01*
2	-23.06	*0.00*	-23.70	*0.00*	-28.28	*-0.13*	-24.23	*-0.14*
5	-28.76	*0.01*	-27.65	*-0.02*	-29.28	*0.00*	-29.00	*0.00*
10	-28.12	*-0.04*	-28.32	*-0.08*	-29.14	*-0.03*	-29.32	*-0.03*
lag	$d = 0.30$		$d = 0.31$		$d = 0.32$		$d = 0.33$	
1	-16.44	*0.06*	-13.31	*-0.06*	-18.64	*-0.20*	-12.49	*-0.68*
2	-25.35	*0.02*	-26.97	*-0.05*	-22.93	*-0.03*	-26.27	*-0.14*
5	-29.96	*-0.01*	-29.70	*-0.01*	-29.49	*-0.01*	-29.49	*0.00*
10	-29.80	*-0.01*	-28.98	*-0.05*	-29.71	*0.00*	-29.84	*-0.02*
lag	$d = 0.34$		$d = 0.35$		$d = 0.36$		$d = 0.37$	
1	-4.07	*-0.20*	17.92	*-2.70*	-18.16	*-0.97*	-24.68	*0.01*
2	-25.08	*-0.12*	-27.46	*-0.04*	-19.18	*-0.40*	-25.61	*-1.34*
5	-29.48	*-0.01*	-29.89	*0.00*	-29.99	*0.00*	-29.00	*0.00*
10	-29.89	*-0.01*	-29.62	*-0.11*	-29.64	*-0.02*	-30.24	*0.00*
lag	$d = 0.38$		$d = 0.39$		$d = 0.40$		$d = 0.41$	
1	-20.85	*-0.07*	-14.84	*0.16*	-23.18	*-0.93*	-22.79	*-3.36*
2	-26.02	*-0.10*	-25.04	*-0.53*	-22.33	*-0.99*	-25.50	*-2.30*
5	-29.84	*-0.06*	-29.98	*-0.23*	-29.63	*-0.22*	-29.62	*-0.52*
10	-30.20	*-0.01*	-30.26	*-0.09*	-30.27	*-0.03*	-30.30	*-0.05*
lag	$d = 0.42$		$d = 0.43$		$d = 0.44$		$d = 0.45$	
1	-26.14	*-1.22*	-23.66	*-6.61*	-25.28	*-5.75*	-27.62	*-1.16*
2	-21.96	*-4.64*	-20.27	*-9.41*	-25.41	*-12.63*	-22.62	*-8.78*
5	-29.25	*-1.02*	-27.15	*-2.16*	-26.81	*-1.72*	-25.25	*-5.09*
10	-29.93	*-0.15*	-29.29	*-0.47*	-29.45	*-0.45*	-28.68	*-0.90*
lag	$d = 0.46$		$d = 0.47$		$d = 0.48$		$d = 0.49$	
1	-26.65	*-2.29*	-30.40	*-9.32*	-39.23	*-22.16*	129.91	*-27.57*
2	-17.17	*-9.83*	-15.77	*-19.81*	-19.60	*-55.67*	103.68	*-12.60*
5	-23.65	*-6.64*	-20.49	*-15.24*	-12.41	*-214.04*	71.80	*-8.23*
10	-28.15	*-1.51*	-27.62	*-4.45*	-49.55	*-98.48*	87.08	*-21.60*

Table 4.4: *Cox test to compare the Monte Carlo distribution of the sample autocorrelation function of a long memory process (equation (1.4)) for different values of the memory parameter d at different lags, $k = 1, 2, 5, 10$, for $n = 300$.*

lag	$d = 0.26$		$d = 0.27$		$d = 0.28$		$d = 0.29$	
1	-25.85	*0.02*	-21.33	*-0.29*	-24.42	*-0.02*	-20.74	*-0.55*
2	-26.41	*-0.02*	-24.59	*-0.15*	-28.40	*0.01*	-26.22	*-0.05*
5	-29.08	*0.01*	-28.42	*-0.04*	-29.50	*0.01*	-29.14	*-0.03*
10	-29.38	*0.01*	-28.15	*-0.05*	-29.40	*-0.02*	-28.82	*-0.07*
lag	$d = 0.30$		$d = 0.31$		$d = 0.32$		$d = 0.33$	
1	-19.86	*-0.36*	-24.51	*-0.11*	-26.24	*-0.01*	-26.44	*-0.12*
2	-24.18	*-0.26*	-27.46	*-0.02*	-27.84	*-0.07*	-28.62	*-0.02*
5	-28.21	*-0.05*	-29.00	*-0.01*	-29.03	*-0.03*	-28.88	*-0.07*
10	-28.99	*-0.04*	-29.27	*-0.04*	-29.85	*-0.01*	-29.26	*-0.05*
lag	$d = 0.34$		$d = 0.35$		$d = 0.36$		$d = 0.37$	
1	-25.70	*-0.27*	-24.23	*-0.27*	-22.50	*-0.36*	-25.35	*-0.23*
2	-28.27	*-0.13*	-26.05	*-0.15*	-23.82	*-0.26*	-27.14	*-0.12*
5	-29.36	*-0.07*	-28.84	*-0.05*	-28.91	*-0.02*	-28.78	*-0.07*
10	-29.15	*-0.12*	-30.13	*-0.01*	-29.83	*-0.01*	-29.77	*-0.03*
lag	$d = 0.38$		$d = 0.39$		$d = 0.40$		$d = 0.41$	
1	-24.65	*-0.45*	-27.11	*-0.06*	-27.90	*-0.06*	-28.83	*-0.03*
2	-27.51	*-0.14*	-29.19	*-0.03*	-29.07	*-0.01*	-29.70	*-0.02*
5	-29.15	*-0.06*	-30.11	*0.00*	-29.62	*-0.01*	-30.18	*0.00*
10	-29.58	*-0.06*	-29.79	*-0.04*	-30.25	*0.00*	-30.41	*0.01*
lag	$d = 0.42$		$d = 0.43$		$d = 0.44$		$d = 0.45$	
1	-26.78	*-0.26*	-29.56	*-0.17*	-27.11	*-0.10*	-28.86	*-0.04*
2	-28.43	*-0.10*	-29.25	*-0.06*	-29.34	*0.00*	-29.87	*-0.18*
5	-29.63	*-0.04*	-30.10	*-0.03*	-30.21	*0.00*	-29.83	*-0.10*
10	-30.04	*-0.03*	-30.34	*-0.02*	-30.20	*0.00*	-30.24	*-0.14*
lag	$d = 0.46$		$d = 0.47$		$d = 0.48$		$d = 0.49$	
1	-26.04	*-0.01*	-27.18	*-1.71*	-22.52	*-12.70*	28.63	*-10.73*
2	-27.97	*-0.51*	-27.27	*-2.58*	-27.51	*-13.88*	35.08	*-9.36*
5	-29.40	*-0.64*	-28.27	*-1.59*	-32.02	*-6.48*	33.53	*-8.76*
10	-29.67	*-0.42*	-28.94	*-1.01*	-32.25	*-6.77*	27.72	*-7.08*

Table 4.5: *Cox test to compare the Monte Carlo distribution of the sample autocorrelation function of a long memory process (equation (1.4)) for different values of the memory parameter d at different lags, $k = 1, 2, 5, 10$, for $n = 1000$.*

lag	$d = 0.26$		$d = 0.27$		$d = 0.28$		$d = 0.29$	
1	-23.13	*-0.06*	-27.27	*-0.02*	-26.47	*-0.07*	-26.53	*-0.02*
2	-27.52	*-0.01*	-25.52	*-0.09*	-26.29	*-0.09*	-26.98	*-0.08*
5	-29.01	*0.00*	-26.22	*-0.08*	-27.71	*-0.03*	-28.25	*-0.04*
10	-29.12	*0.01*	-29.00	*0.00*	-26.90	*-0.12*	-28.74	*-0.02*
lag	$d = 0.30$		$d = 0.31$		$d = 0.32$		$d = 0.33$	
1	-25.87	*-0.10*	-25.08	*-0.12*	-27.12	*-0.03*	-28.50	*-0.07*
2	-28.32	*-0.01*	-27.21	*-0.07*	-28.43	*-0.01*	-29.22	*-0.02*
5	-29.83	*0.00*	-28.14	*-0.04*	-29.08	*-0.01*	-28.88	*-0.07*
10	-29.73	*-0.01*	-28.93	*-0.03*	-29.68	*0.00*	-28.69	*-0.11*
lag	$d = 0.34$		$d = 0.35$		$d = 0.36$		$d = 0.37$	
1	-26.53	*-0.15*	-26.94	*-0.07*	-27.02	*-0.14*	-27.93	*-0.06*
2	-27.91	*-0.12*	-28.70	*-0.03*	-27.78	*-0.10*	-29.01	*-0.03*
5	-29.43	*-0.05*	-28.90	*-0.04*	-28.67	*-0.06*	-28.62	*-0.09*
10	-29.69	*-0.04*	-29.40	*-0.03*	-29.11	*-0.05*	-29.45	*-0.07*
lag	$d = 0.38$		$d = 0.39$		$d = 0.40$		$d = 0.41$	
1	-28.15	*-0.11*	-26.37	*-0.20*	-28.44	*-0.05*	-29.43	*-0.02*
2	-28.50	*-0.10*	-28.08	*-0.09*	-28.42	*-0.10*	-29.50	*-0.02*
5	-29.11	*-0.10*	-28.80	*-0.05*	-29.33	*-0.04*	-29.58	*-0.01*
10	-29.29	*-0.08*	-29.22	*-0.04*	-29.73	*-0.03*	-29.97	*0.00*
lag	$d = 0.42$		$d = 0.43$		$d = 0.44$		$d = 0.45$	
1	-27.94	*-0.04*	-28.27	*-0.05*	-29.59	*0.00*	-30.03	*0.01*
2	-28.77	*-0.04*	-29.05	*-0.02*	-30.06	*0.00*	-30.05	*0.02*
5	-28.85	*-0.07*	-29.86	*0.01*	-30.03	*0.00*	-30.36	*0.02*
10	-29.72	*-0.01*	-29.75	*-0.05*	-30.41	*0.01*	-30.43	*0.03*
lag	$d = 0.46$		$d = 0.47$		$d = 0.48$		$d = 0.49$	
1	-29.58	*0.00*	-29.06	*-0.01*	-29.44	*-0.03*	-20.20	*-5.23*
2	-29.99	*0.00*	-29.19	*0.00*	-30.10	*-0.30*	-25.85	*-7.65*
5	-29.89	*0.00*	-29.50	*-0.05*	-30.10	*-0.29*	-28.31	*-9.86*
10	-30.29	*-0.03*	-29.76	*-0.04*	-30.22	*-0.49*	-35.97	*-15.91*

Table 4.6: *Cox test to compare the Monte Carlo distribution of the sample autocorrelation function of a long memory process (equation (1.4)) for different values of the memory parameter d at different lags, $k = 1, 2, 5, 10$, for $n = 1000$.*

	$d = 0.05$				$d = 0.10$				$d = 0.15$			
n	$\hat{\gamma}_0$	$\hat{\gamma}_0^*$	$\hat{se}(\hat{\gamma}_0)$	$\hat{se}(\hat{\gamma}_0^*)$	$\hat{\gamma}_0$	$\hat{\gamma}_0^*$	$\hat{se}(\hat{\gamma}_0)$	$\hat{se}(\hat{\gamma}_0^*)$	$\hat{\gamma}_0$	$\hat{\gamma}_0^*$	$\hat{se}(\hat{\gamma}_0)$	$\hat{se}(\hat{\gamma}_0^*)$
100	**1.0204**	**1.0225**	0.1151	0.1150	**1.0046**	**1.0078**	0.1522	0.1523	**1.0303**	**1.0324**	0.1414	0.1409
200	**0.9989**	**1.0006**	0.0768	0.0771	**1.0250**	**1.0273**	0.1006	0.1009	**1.0207**	**1.0224**	0.0960	0.0963
300	**1.0116**	**1.0126**	0.0670	0.0674	**1.0104**	**1.0115**	0.0892	0.0890	**1.0196**	**1.0203**	0.0882	0.0883
400	**1.0079**	**1.0090**	0.0669	0.0681	**1.0064**	**1.0074**	0.0739	0.0740	**1.0411**	**1.0410**	0.0695	0.0692
500	**0.9936**	**0.9942**	0.0501	0.0504	**1.0205**	**1.0209**	0.0681	0.0671	**1.0500**	**1.0498**	0.0804	0.0798
600	**0.9944**	**0.9957**	0.0508	0.0517	**1.0063**	**1.0073**	0.0642	0.0637	**1.0287**	**1.0278**	0.0672	0.0673
700	**1.0095**	**1.0102**	0.0471	0.0474	**1.0143**	**1.0153**	0.0596	0.0594	**1.0307**	**1.0307**	0.0559	0.0561
800	**1.0024**	**1.0030**	0.0505	0.0508	**1.0115**	**1.0113**	0.0546	0.0542	**1.0459**	**1.0453**	0.0537	0.0538
900	**1.0078**	**1.0084**	0.0425	0.0432	**1.0141**	**1.0142**	0.0491	0.0491	**1.0366**	**1.0361**	0.0579	0.0576
1000	**1.0028**	**1.0032**	0.0366	0.0364	**1.0092**	**1.0096**	0.0460	0.0463	**1.0447**	**1.0442**	0.0514	0.0516
1100	**1.0032**	**1.0035**	0.0390	0.0390	**1.0294**	**1.0295**	0.0460	0.0457	**1.0469**	**1.0465**	0.0495	0.0489
1200	**1.0048**	**1.0053**	0.0366	0.0366	**1.0248**	**1.0248**	0.0397	0.0396	**1.0484**	**1.0478**	0.0436	0.0437
1300	**1.0053**	**1.0053**	0.0326	0.0327	**1.0106**	**1.0112**	0.0418	0.0423	**1.0456**	**1.0448**	0.0428	0.0424
1400	**1.0099**	**1.0101**	0.0286	0.0288	**1.0135**	**1.0134**	0.0392	0.0391	**1.0408**	**1.0404**	0.0474	0.0472
1500	**1.0012**	**1.0014**	0.0337	0.0335	**1.0150**	**1.0144**	0.0386	0.0385	**1.0438**	**1.0438**	0.0429	0.0425
1600	**0.9990**	**0.9992**	0.0381	0.0385	**1.0223**	**1.0221**	0.0392	0.0392	**1.0383**	**1.0382**	0.0356	0.0358
1700	**1.0041**	**1.0043**	0.0333	0.0335	**1.0197**	**1.0193**	0.0359	0.0361	**1.0388**	**1.0377**	0.0370	0.0372
1800	**1.0041**	**1.0047**	0.0300	0.0300	**1.0141**	**1.0141**	0.0352	0.0355	**1.0459**	**1.0457**	0.0398	0.0391
1900	**1.0053**	**1.0053**	0.0314	0.0312	**1.0204**	**1.0200**	0.0367	0.0366	**1.0463**	**1.0457**	0.0348	0.0347
2000	**1.0017**	**1.0020**	0.0348	0.0345	**1.0197**	**1.0198**	0.0319	0.0318	**1.0414**	**1.0413**	0.0374	0.0375

Table 4.7: *Comparison between sample and bootstrap variances, of fractionally Gaussian integrated noise for* $d = 0.05, 0.10, 0.15$ *and increasing series length.*

	$d = 0.20$				$d = 0.25$				$d = 0.30$			
n	$\hat{\gamma}_0$	$\hat{\gamma}_0^*$	$\hat{se}(\hat{\gamma}_0)$	$\hat{se}(\hat{\gamma}_0^*)$	$\hat{\gamma}_0$	$\hat{\gamma}_0^*$	$\hat{se}(\hat{\gamma}_0)$	$\hat{se}(\hat{\gamma}_0^*)$	$\hat{\gamma}_0$	$\hat{\gamma}_0^*$	$\hat{se}(\hat{\gamma}_0)$	$\hat{se}(\hat{\gamma}_0^*)$
100	**1.0413**	**1.0424**	0.1624	0.1619	**1.0766**	**1.0757**	0.1589	0.1541	**1.1182**	**1.1134**	0.2066	0.1986
200	**1.0439**	**1.0431**	0.1041	0.1025	**1.1052**	**1.1011**	0.1389	0.1362	**1.1801**	**1.1710**	0.1714	0.1626
300	**1.0648**	**1.0620**	0.1128	0.1112	**1.1258**	**1.1225**	0.1224	0.1204	**1.2029**	**1.1950**	0.1370	0.1301
400	**1.0732**	**1.0707**	0.0840	0.0821	**1.1193**	**1.1172**	0.0962	0.0958	**1.2046**	**1.1975**	0.1088	0.1054
500	**1.0718**	**1.0700**	0.0804	0.0801	**1.1341**	**1.1288**	0.0972	0.0934	**1.1987**	**1.1903**	0.1280	0.1214
600	**1.0829**	**1.0801**	0.0806	0.0788	**1.1327**	**1.1286**	0.0789	0.0762	**1.2123**	**1.2069**	0.1050	0.1025
700	**1.0710**	**1.0691**	0.0609	0.0605	**1.1502**	**1.1471**	0.0727	0.0720	**1.2261**	**1.2192**	0.0904	0.0863
800	**1.0821**	**1.0807**	0.0656	0.0648	**1.1439**	**1.1403**	0.0834	0.0835	**1.2326**	**1.2265**	0.1018	0.0997
900	**1.0770**	**1.0757**	0.0540	0.0535	**1.1491**	**1.1451**	0.0740	0.0726	**1.2518**	**1.2446**	0.1097	0.1031
1000	**1.0745**	**1.0733**	0.0565	0.0559	**1.1477**	**1.1451**	0.0659	0.0642	**1.2446**	**1.2381**	0.0902	0.0868
1100	**1.0785**	**1.0768**	0.0579	0.0569	**1.1468**	**1.1433**	0.0664	0.0653	**1.2308**	**1.2264**	0.0760	0.0752
1200	**1.0916**	**1.0900**	0.0482	0.0483	**1.1448**	**1.1411**	0.0597	0.0573	**1.2510**	**1.2456**	0.0749	0.0702
1300	**1.0912**	**1.0898**	0.0523	0.0521	**1.1502**	**1.1474**	0.0589	0.0581	**1.2397**	**1.2344**	0.0893	0.0861
1400	**1.0851**	**1.0837**	0.0487	0.0481	**1.1462**	**1.1443**	0.0531	0.0524	**1.2419**	**1.2379**	0.0757	0.0746
1500	**1.0837**	**1.0828**	0.0490	0.0488	**1.1524**	**1.1497**	0.0610	0.0602	**1.2490**	**1.2450**	0.0711	0.0677
1600	**1.0903**	**1.0886**	0.0421	0.0415	**1.1530**	**1.1502**	0.0521	0.0512	**1.2388**	**1.2348**	0.0587	0.0560
1700	**1.0868**	**1.0852**	0.0418	0.0419	**1.1551**	**1.1526**	0.0491	0.0478	**1.2591**	**1.2534**	0.0704	0.0678
1800	**1.0902**	**1.0888**	0.0375	0.0379	**1.1627**	**1.1604**	0.0502	0.0498	**1.2661**	**1.2614**	0.0769	0.0747
1900	**1.0977**	**1.0968**	0.0381	0.0381	**1.1579**	**1.1551**	0.0485	0.0483	**1.2609**	**1.2565**	0.0734	0.0709
2000	**1.0927**	**1.0916**	0.0372	0.0368	**1.1534**	**1.1505**	0.0507	0.0492	**1.2655**	**1.2602**	0.0644	0.0622

Table 4.8: *Comparison between sample and bootstrap variances, of fractionally Gaussian integrated noise for* $d = 0.20, 0.25, 0.30$ *and increasing series length.*

	$d = 0.35$				$d = 0.40$				$d = 0.45$			
n	$\hat{\gamma}_0$	$\hat{\gamma}_0^*$	$\hat{se}(\hat{\gamma}_0)$	$\hat{se}(\hat{\gamma}_0^*)$	$\hat{\gamma}_0$	$\hat{\gamma}_0^*$	$\hat{se}(\hat{\gamma}_0)$	$\hat{se}(\hat{\gamma}_0^*)$	$\hat{\gamma}_0$	$\hat{\gamma}_0^*$	$\hat{se}(\hat{\gamma}_0)$	$\hat{se}(\hat{\gamma}_0^*)$
100	**1.2065**	**1.1928**	0.2436	0.2305	**1.2456**	**1.2310**	0.2834	0.2703	**1.5009**	**1.4642**	0.3574	0.3278
200	**1.2522**	**1.2375**	0.1922	0.1822	**1.4147**	**1.3910**	0.2629	0.2434	**1.5934**	**1.5519**	0.3061	0.2776
300	**1.2956**	**1.2840**	0.1942	0.1802	**1.4611**	**1.4406**	0.2374	0.2252	**1.6809**	**1.6422**	0.3515	0.3227
400	**1.3163**	**1.3048**	0.1673	0.1600	**1.5402**	**1.5203**	0.2633	0.2483	**1.7376**	**1.7030**	0.3579	0.3292
500	**1.3552**	**1.3447**	0.1740	0.1644	**1.5215**	**1.4976**	0.2351	0.2125	**1.7439**	**1.7113**	0.3415	0.3085
600	**1.3764**	**1.3623**	0.1417	0.1345	**1.5596**	**1.5347**	0.2460	0.2272	**1.8080**	**1.7724**	0.3205	0.2911
700	**1.3614**	**1.3492**	0.1337	0.1248	**1.5571**	**1.5382**	0.2112	0.1970	**1.7961**	**1.7608**	0.3277	0.2987
800	**1.3614**	**1.3506**	0.1464	0.1367	**1.5573**	**1.5397**	0.1812	0.1717	**1.8552**	**1.8234**	0.2544	0.2365
900	**1.3636**	**1.3544**	0.1267	0.1212	**1.5625**	**1.5463**	0.1603	0.1497	**1.8849**	**1.8438**	0.3291	0.2864
1000	**1.3890**	**1.3801**	0.1299	0.1256	**1.5909**	**1.5718**	0.1907	0.1775	**1.8489**	**1.8167**	0.2562	0.2337
1100	**1.3898**	**1.3791**	0.1325	0.1246	**1.5837**	**1.5659**	0.1670	0.1569	**1.9098**	**1.8777**	0.2946	0.2705
1200	**1.4002**	**1.3887**	0.1244	0.1199	**1.5951**	**1.5791**	0.1623	0.1531	**1.8647**	**1.8370**	0.2924	0.2674
1300	**1.3991**	**1.3882**	0.1144	0.1054	**1.6048**	**1.5888**	0.1546	0.1452	**1.9579**	**1.9279**	0.3020	0.2786
1400	**1.4056**	**1.3953**	0.1065	0.0997	**1.6116**	**1.5902**	0.1978	0.1858	**1.9419**	**1.9102**	0.2485	0.2302
1500	**1.3863**	**1.3752**	0.1156	0.1085	**1.5969**	**1.5793**	0.1624	0.1522	**1.9801**	**1.9459**	0.3115	0.2776
1600	**1.3956**	**1.3886**	0.0981	0.0941	**1.6223**	**1.6027**	0.1928	0.1748	**1.9852**	**1.9508**	0.2783	0.2536
1700	**1.4183**	**1.4088**	0.1156	0.1097	**1.6368**	**1.6189**	0.1642	0.1518	**1.9746**	**1.9388**	0.2917	0.2543
1800	**1.4141**	**1.4037**	0.1161	0.1082	**1.6470**	**1.6299**	0.1720	0.1613	**2.0033**	**1.9699**	0.3096	0.2885
1900	**1.4028**	**1.3949**	0.1046	0.1007	**1.6257**	**1.6114**	0.1849	0.1723	**1.9956**	**1.9638**	0.2538	0.2296
2000	**1.4035**	**1.3953**	0.0972	0.0925	**1.6384**	**1.6233**	0.1689	0.1554	**2.0084**	**1.9783**	0.2784	0.2568

Table 4.9: *Comparison between sample and bootstrap variances, of fractionally Gaussian integrated noise for $d = 0.35, 0.40, 0.45$ and increasing series length.*

	$d = 0.05$				$d = 0.10$				$d = 0.15$			
n	$\hat{\rho}_1$	$\hat{\rho}_1^*$	$\hat{se}(\hat{\rho}_1)$	$\hat{se}(\hat{\rho}_1^*)$	$\hat{\rho}_1$	$\hat{\rho}_1^*$	$\hat{se}(\hat{\rho}_1)$	$\hat{se}(\hat{\rho}_1^*)$	$\hat{\rho}_1$	$\hat{\rho}_1^*$	$\hat{se}(\hat{\rho}_1)$	$\hat{se}(\hat{\rho}_1^*)$
100	**0.0225**	**0.0192**	0.0657	0.0630	**0.0355**	**0.0266**	0.1113	0.1029	**0.0635**	**0.0517**	0.1052	0.0981
200	**0.0160**	**0.0140**	0.0586	0.0572	**0.0571**	**0.0525**	0.0743	0.0716	**0.0728**	**0.0664**	0.0812	0.0781
300	**0.0211**	**0.0192**	0.0551	0.0539	**0.0572**	**0.0537**	0.0581	0.0573	**0.0937**	**0.0885**	0.0676	0.0653
400	**0.0151**	**0.0140**	0.0447	0.0439	**0.0564**	**0.0542**	0.0544	0.0535	**0.0899**	**0.0855**	0.0576	0.0559
500	**0.0273**	**0.0261**	0.0413	0.0411	**0.0622**	**0.0594**	0.0462	0.0451	**0.1000**	**0.0967**	0.0471	0.0463
600	**0.0263**	**0.0253**	0.0366	0.0366	**0.0610**	**0.0588**	0.0440	0.0439	**0.1060**	**0.1027**	0.0462	0.0454
700	**0.0227**	**0.0218**	0.0332	0.0330	**0.0524**	**0.0509**	0.0431	0.0428	**0.0978**	**0.0953**	0.0386	0.0380
800	**0.0251**	**0.0246**	0.0353	0.0350	**0.0556**	**0.0541**	0.0351	0.0349	**0.1000**	**0.0969**	0.0397	0.0395
900	**0.0232**	**0.0226**	0.0330	0.0329	**0.0564**	**0.0552**	0.0347	0.0345	**0.1117**	**0.1093**	0.0377	0.0372
1000	**0.0207**	**0.0201**	0.0284	0.0281	**0.0589**	**0.0573**	0.0327	0.0327	**0.1054**	**0.1032**	0.0364	0.0358
1100	**0.0225**	**0.0222**	0.0312	0.0309	**0.0619**	**0.0604**	0.0353	0.0351	**0.1055**	**0.1031**	0.0352	0.0349
1200	**0.0224**	**0.0220**	0.0286	0.0283	**0.0616**	**0.0603**	0.0307	0.0307	**0.1099**	**0.1077**	0.0332	0.0326
1300	**0.0249**	**0.0245**	0.0273	0.0273	**0.0590**	**0.0582**	0.0273	0.0273	**0.1024**	**0.1005**	0.0277	0.0271
1400	**0.0240**	**0.0234**	0.0246	0.0247	**0.0623**	**0.0613**	0.0276	0.0273	**0.0970**	**0.0955**	0.0349	0.0344
1500	**0.0271**	**0.0267**	0.0263	0.0264	**0.0613**	**0.0602**	0.0278	0.0277	**0.1015**	**0.0999**	0.0308	0.0306
1600	**0.0298**	**0.0295**	0.0213	0.0213	**0.0612**	**0.0600**	0.0284	0.0284	**0.1020**	**0.1007**	0.0279	0.0277
1700	**0.0241**	**0.0236**	0.0241	0.0239	**0.0649**	**0.0641**	0.0249	0.0251	**0.1034**	**0.1017**	0.0272	0.0269
1800	**0.0270**	**0.0266**	0.0224	0.0225	**0.0639**	**0.0633**	0.0238	0.0236	**0.1076**	**0.1062**	0.0276	0.0272
1900	**0.0242**	**0.0241**	0.0251	0.0251	**0.0611**	**0.0602**	0.0242	0.0239	**0.1080**	**0.1069**	0.0271	0.0266
2000	**0.0270**	**0.0266**	0.0240	0.0241	**0.0599**	**0.0594**	0.0239	0.0238	**0.1067**	**0.1052**	0.0254	0.0255

Table 4.10: *Comparison between sample and bootstrap autocorrelation functions at lag one, of fractionally Gaussian integrated noise for* $d = 0.05, 0.10, 0.15$ *and increasing series length.*

	$d = 0.20$				$d = 0.25$				$d = 0.30$			
n	$\hat{\rho}_1$	$\hat{\rho}_1^*$	$\hat{se}(\hat{\rho}_1)$	$\hat{se}(\hat{\rho}_1^*)$	$\hat{\rho}_1$	$\hat{\rho}_1^*$	$\hat{se}(\hat{\rho}_1)$	$\hat{se}(\hat{\rho}_1^*)$	$\hat{\rho}_1$	$\hat{\rho}_1^*$	$\hat{se}(\hat{\rho}_1)$	$\hat{se}(\hat{\rho}_1^*)$
100	**0.0874**	**0.0729**	0.1044	0.0965	**0.1508**	**0.1288**	0.1177	0.1056	**0.3245**	**0.2972**	0.1084	0.1002
200	**0.1132**	**0.1035**	0.0743	0.0700	**0.1665**	**0.1524**	0.0832	0.0757	**0.3517**	**0.3315**	0.0796	0.0735
300	**0.1449**	**0.1356**	0.0699	0.0654	**0.1851**	**0.1739**	0.0763	0.0712	**0.3632**	**0.3479**	0.0712	0.0671
400	**0.1471**	**0.1401**	0.0515	0.0489	**0.1869**	**0.1789**	0.0672	0.0649	**0.3710**	**0.3586**	0.0620	0.0589
500	**0.1440**	**0.1380**	0.0563	0.0544	**0.2092**	**0.1995**	0.0566	0.0529	**0.3687**	**0.3567**	0.0604	0.0572
600	**0.1444**	**0.1384**	0.0460	0.0440	**0.2020**	**0.1944**	0.0495	0.0463	**0.3793**	**0.3697**	0.0491	0.0466
700	**0.1439**	**0.1395**	0.0440	0.0427	**0.2039**	**0.1976**	0.0434	0.0422	**0.3823**	**0.3728**	0.0446	0.0421
800	**0.1554**	**0.1511**	0.0457	0.0443	**0.2149**	**0.2085**	0.0495	0.0484	**0.3897**	**0.3814**	0.0487	0.0467
900	**0.1542**	**0.1505**	0.0406	0.0396	**0.2141**	**0.2076**	0.0439	0.0421	**0.3896**	**0.3813**	0.0483	0.0449
1000	**0.1568**	**0.1534**	0.0393	0.0388	**0.2087**	**0.2034**	0.0434	0.0414	**0.3903**	**0.3827**	0.0424	0.0405
1100	**0.1521**	**0.1485**	0.0402	0.0394	**0.2178**	**0.2123**	0.0413	0.0394	**0.3860**	**0.3799**	0.0375	0.0370
1200	**0.1606**	**0.1569**	0.0340	0.0334	**0.2137**	**0.2092**	0.0379	0.0365	**0.3902**	**0.3841**	0.0402	0.0384
1300	**0.1593**	**0.1564**	0.0335	0.0331	**0.2153**	**0.2103**	0.0446	0.0433	**0.3909**	**0.3846**	0.0430	0.0412
1400	**0.1499**	**0.1467**	0.0296	0.0289	**0.2172**	**0.2136**	0.0322	0.0313	**0.3960**	**0.3905**	0.0366	0.0354
1500	**0.1550**	**0.1523**	0.0293	0.0288	**0.2214**	**0.2173**	0.0374	0.0359	**0.3948**	**0.3894**	0.0358	0.0342
1600	**0.1612**	**0.1587**	0.0308	0.0302	**0.2209**	**0.2167**	0.0353	0.0341	**0.3921**	**0.3875**	0.0302	0.0289
1700	**0.1501**	**0.1475**	0.0298	0.0291	**0.2185**	**0.2148**	0.0357	0.0348	**0.3950**	**0.3894**	0.0340	0.0328
1800	**0.1627**	**0.1602**	0.0290	0.0282	**0.2207**	**0.2171**	0.0316	0.0307	**0.4050**	**0.3999**	0.0402	0.0388
1900	**0.1598**	**0.1575**	0.0281	0.0277	**0.2240**	**0.2201**	0.0313	0.0306	**0.4040**	**0.3990**	0.0349	0.0336
2000	**0.1568**	**0.1548**	0.0253	0.0249	**0.2242**	**0.2206**	0.0319	0.0307	**0.4021**	**0.3971**	0.0306	0.0294

Table 4.11: *Comparison between sample and bootstrap autocorrelation functions at lag one, of fractionally Gaussian integrated noise for* $d = 0.20, 0.25, 0.30$ *and increasing series length.*

	$d = 0.35$				$d = 0.40$				$d = 0.45$			
n	$\hat{\rho}_1$	$\hat{\rho}_1^*$	$\hat{se}(\hat{\rho}_1)$	$\hat{se}(\hat{\rho}_1^*)$	$\hat{\rho}_1$	$\hat{\rho}_1^*$	$\hat{se}(\hat{\rho}_1)$	$\hat{se}(\hat{\rho}_1^*)$	$\hat{\rho}_1$	$\hat{\rho}_1^*$	$\hat{se}(\hat{\rho}_1)$	$\hat{se}(\hat{\rho}_1^*)$
100	**0.3727**	**0.3383**	0.1183	0.1078	**0.4228**	**0.3857**	0.1099	0.0985	**0.5306**	**0.4822**	0.1072	0.0964
200	**0.4128**	**0.3898**	0.0886	0.0814	**0.4880**	**0.4586**	0.0924	0.0861	**0.5653**	**0.5295**	0.0853	0.0786
300	**0.4344**	**0.4167**	0.0824	0.0757	**0.5080**	**0.4856**	0.0759	0.0705	**0.5912**	**0.5637**	0.0750	0.0656
400	**0.4417**	**0.4265**	0.0706	0.0669	**0.5343**	**0.5150**	0.0774	0.0717	**0.5975**	**0.5748**	0.0783	0.0724
500	**0.4566**	**0.4433**	0.0621	0.0581	**0.5374**	**0.5186**	0.0623	0.0571	**0.6075**	**0.5870**	0.0722	0.0649
600	**0.4619**	**0.4479**	0.0578	0.0539	**0.5476**	**0.5299**	0.0626	0.0566	**0.6193**	**0.6000**	0.0615	0.0553
700	**0.4654**	**0.4529**	0.0493	0.0457	**0.5497**	**0.5349**	0.0612	0.0565	**0.6228**	**0.6051**	0.0638	0.0575
800	**0.4702**	**0.4591**	0.0515	0.0480	**0.5480**	**0.5342**	0.0551	0.0520	**0.6339**	**0.6168**	0.0490	0.0453
900	**0.4627**	**0.4530**	0.0455	0.0428	**0.5532**	**0.5407**	0.0462	0.0427	**0.6359**	**0.6191**	0.0566	0.0499
1000	**0.4757**	**0.4667**	0.0448	0.0423	**0.5560**	**0.5433**	0.0511	0.0467	**0.6353**	**0.6202**	0.0498	0.0449
1100	**0.4792**	**0.4697**	0.0448	0.0413	**0.5578**	**0.5461**	0.0497	0.0466	**0.6441**	**0.6292**	0.0522	0.0472
1200	**0.4825**	**0.4732**	0.0458	0.0433	**0.5625**	**0.5515**	0.0445	0.0418	**0.6367**	**0.6236**	0.0530	0.0488
1300	**0.4792**	**0.4700**	0.0426	0.0391	**0.5629**	**0.5523**	0.0444	0.0413	**0.6535**	**0.6401**	0.0499	0.0455
1400	**0.4838**	**0.4754**	0.0418	0.0388	**0.5665**	**0.5544**	0.0498	0.0464	**0.6517**	**0.6381**	0.0444	0.0404
1500	**0.4837**	**0.4753**	0.0406	0.0379	**0.5657**	**0.5555**	0.0409	0.0375	**0.6559**	**0.6425**	0.0493	0.0431
1600	**0.4820**	**0.4753**	0.0408	0.0390	**0.5705**	**0.5599**	0.0453	0.0408	**0.6595**	**0.6463**	0.0437	0.0390
1700	**0.4903**	**0.4827**	0.0414	0.0390	**0.5746**	**0.5647**	0.0402	0.0364	**0.6575**	**0.6448**	0.0444	0.0391
1800	**0.4884**	**0.4810**	0.0430	0.0395	**0.5773**	**0.5676**	0.0458	0.0423	**0.6611**	**0.6484**	0.0475	0.0428
1900	**0.4838**	**0.4774**	0.0368	0.0344	**0.5687**	**0.5600**	0.0458	0.0419	**0.6615**	**0.6494**	0.0426	0.0384
2000	**0.4873**	**0.4810**	0.0354	0.0335	**0.5748**	**0.5662**	0.0395	0.0359	**0.6615**	**0.6497**	0.0448	0.0401

Table 4.12: *Comparison between sample and bootstrap autocorrelation functions at lag one, of fractionally Gaussian integrated noise for* $d = 0.35, 0.40, 0.45$ *and increasing series length.*

Chapter 5

Semi-parametric estimators for Garma processes

5.1 Introduction

In Chapter 1 of this thesis we highlighted that identification and estimation of seasonal persistences are still open problems. The existing parametric estimators are quite efficient, even though there are not many asymptotic results (Sadek and Khotanzad, 2004; Smallwood and Beaumont, 2004; Woodward et al., 1998). On the other side, the main problem is the identification of the data generating process, in particular the number and the position of the peaks in the spectral density and the short memory behaviour. Without a good specification of the model the maximum likelihood estimates might be very biased. Semi-parametric techniques can be useful in an exploratory stage, providing a better understanding of the data generating process.

Among the processes developed to model seasonal persistences, we choose one of the most flexible, the Gegenbauer process introduced in Section 1.5 with representation given in Equation (1.22) and spectral density (1.23). For the sake of simplicity, we limit the study to Gegenbauer processes with $k \leq 2$ and with ε_t Gaussian white noise. An example is given in Figure 5.1 with $k = 2$, where one peak is at the zero frequency, $\eta_1 = 1$, and the other at $\omega_2 = \arccos \eta_2$, $0 \leq \omega_2 \leq \pi$[1].

In this study we propose an algorithm to identify seasonal persistences and long memory behaviour, using semi-parametric techniques. We also apply the bootstrap technique introduced in Chapter 3 to improve the performance of the semi-parametric estimators and to have the bootstrap distributions of the estimates. The algorithm and the bootstrap method can be used not only to identify the model, but also to provide good starting values to maximize the likelihood function (see Equations (1.24) and (1.25)).

Even if the Gegenbauer frequency is usually known, we want to see how the estimation of the frequency influences the other estimates. In the case η_i are known, the existing semi-parametric estimation procedures (the generalised GPH and the generalised local Whittle

[1] We indicate with $0 < \omega < \pi$ the frequency of the spectral density and with $-1 < \eta < 1$ the Gegenbauer frequency. The two parameters are linked by the relationship $\omega = \arccos\eta$.

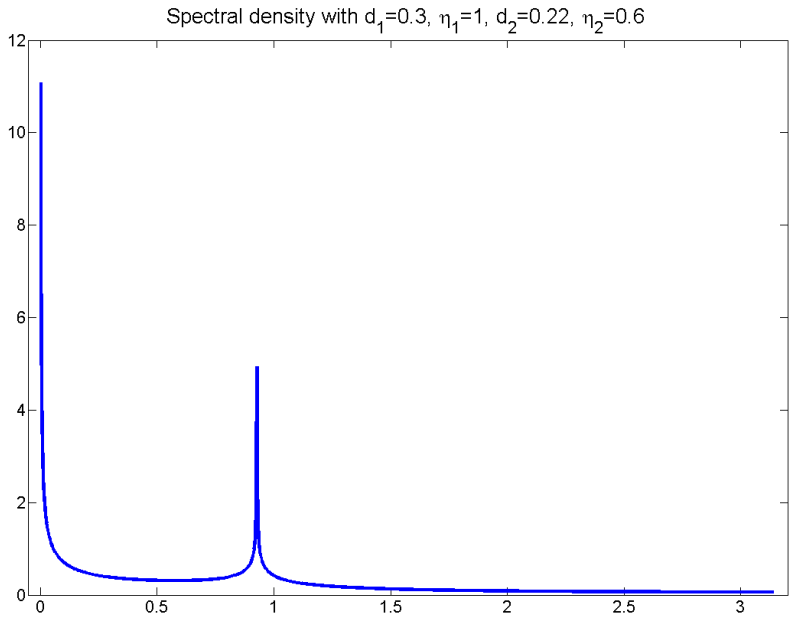


Figure 5.1: *Spectral density of a process with representation (1.22) with* $k = 2$, $d_1 = 0.3$, $d_2 = 0.22$ *and* $\eta_1 = 1$, $\eta_2 = 0.6$. *It is possible to see the two poles, one at the zero frequency,* $\omega_1 = \arccos 1 = 0$, *and the other at* $\omega_2 = \arccos 0.6 = 0.9273$.

proposed by Arteche and Robinson, 2000) are consistent and asymptotically Normal. We will observe that, when we have to estimate the frequency, also the estimates of the memory parameters are affected.

The chapter is organised as follows. In Section 5.2 we introduce the estimation procedures. In Section 5.3 we describe the Monte Carlo experiments, then we provide and comment the results. In Section 5.4 we conclude the chapter and provide future developments.

5.2 Estimation procedures

To estimate the memory parameter d, we use generalized versions of semi-parametric estimators, the GPH and the local Whittle. These estimators were generalized by Arteche and Robinson (2000) in the asymmetric case (see Section 1.5) but we simplify them to the symmetric case.

The case studied by Arteche and Robinson (2000) assumed that the position of the peak, η, was known, whereas we want to study the case when we do not know its position. The simultaneous estimation of the two parameters (d_i, η_i) is an interesting issue. The Gegenbauer frequency η is easier to locate if the long memory parameter d is larger, whereas we need to know η (or at least to have a good estimate) to estimate d. Also in the parametric estimation procedures, special care is needed because the convergence rates of the two parameters are different (Smallwood and Beaumont, 2004).

However, we notice that we are allowed to estimate the spectrum at any frequency even though the asymptotic properties of the periodogram at the Fourier frequencies ($\omega_j = 2\pi j/n$) are more interesting:

- pg.405 Priestley (1988): $\text{cov}\{I(\omega_1), I(\omega_2)\} = O(1/n^2)$ if X_t is normal and $|\omega_1 \pm \omega_2| >> 2\pi/n$;
- under the conditions $\omega_j \pm \omega_{j'} \neq 2\pi l$, $(l \in Z)$, theorem 3.7 of Beran (1994) provides the asymptotic result

$$[I(\omega_1), \ldots, I(\omega_m)] \xrightarrow{d} [f(\omega_1)\xi_1, \ldots, f(\omega_m)\xi_m],$$

  where $\xi_1, \ldots, \xi_m$ are independent exponential random variables with mean 1;
- proposition 10.3.2 of Brockwell and Davis (1991) shows that asymptotic independence is valid for any group of frequencies $0 < \omega_1 < \ldots < \omega_m < \pi$ without any assumption on m.

These properties allow us to generalize GPH and local Whittle at any frequency. The generalized GPH estimator is given by

$$\hat{d}_i = \frac{\sum_{j=1}^{m} v_j \log I(\omega_i + \omega_j)}{\sum_{j=1}^{m} v_j^2},$$

where $v_j = \log|j| - \sum_{l=1}^{m} \log l/m$ (Arteche and Robinson, 2000) and ω_i the Fourier frequency corresponding to the peak in the spectral density. From now on we use the simpler notation $\omega_j = 2\pi j/n$ for the Fourier frequencies. The generalized local Whittle is found by minimizing the function

$$R(d_j) = \log\left(\frac{1}{m}\sum_{j=1}^{m} \omega_j^{2d_i} I(\lambda_i + \omega_j)\right) - \frac{2d_i}{m}\sum_{j=1}^{m} \log \omega_j.$$

In the asymmetric case, Arteche and Robinson (2000) considered both sides of the peak to build a test for asymmetry. For simplicity we consider only the right side of the symmetric peak. If η is known, both estimators are asymptotically Normal.

The main issue is finding a good procedure to identify the frequency with the peak. We consider the following two solutions.

- A very intuitive procedure is selecting through a graphical analysis an interval where there could be a peak, then compute the periodogram for a very fine grid of values of η and choose the frequency η corresponding to the highest periodogram estimates.
- Since the periodogram is not a consistent estimator of the spectral density, as an alternative we choose the frequency with the same procedure of the previous point after smoothing the periodogram. The issue of smoothing the periodogram opens a vast discussion about the choice of the best window and the optimal width, however as an exploratory work we do not go into details and use the Parzen window[2].

In the case with two peaks we develop two different procedures. Both estimators are semi-parametric and it is not necessary to specify the model, thus we can estimate the couple (d_1, η_1) independently from the couple (d_2, η_2). It is our belief that the closer the Gegenbauer frequencies are, the more difficult it is to distinguish the existence of two peaks and we already mentioned that the short memory part (see Section 3.2) introduces bias to semi-parametric estimators. For these reasons, we expect the estimates to be more biased and to influence each other. To take this dependence into account, we develop the following algorithm, that estimates the couple (d_j, η_j), $j = 1, 2$, after filtering from the other couple:

1. estimate (d_1, η_1) with the procedure above;
2. filter the series $Y_t = (1 - 2\hat{\eta}_1 B + B^2)^{\hat{d}_1} X_t$;
3. estimate (d_2, η_2) on the filtered series Y_t;

[2]The Parzen window is defined as follows

$$\begin{aligned} w(j) &= 1 - 6\left(\tfrac{2j}{n}\right)^2\left(1 - \tfrac{2|j|}{n}\right), && 0 \le |j| \le \tfrac{n}{4} \\ w(j) &= \left(1 - \tfrac{2|j|}{n}\right)^3, && 0 \le |j| \le \tfrac{n}{4} \end{aligned}$$

where n is the series length. The smoothed periodogram is calculated after multiplying the sample autocovariance function by the wights $w(j)$.

4. filter the series $Z_t = (1 - 2\hat{\eta}_2 B + B^2)^{\hat{d}_2} X_t$;

5. estimate (d_1, η_1) on the filtered series Z_t;

6. back to step 2 until some convergence criterion is satisfied.

Since we have only a finite series, in step two and four we need to approximate the filter. We think that it is not adequate to use the AR-infinite representation because we are forced to truncate the series and it is known that for a good approximation a large number of terms is required (Woodward et al., 1998). An option is given by the theorem of Ramsey (1974) given in Chapter 3. We are allowed to use this result because we assume Normality and the values of the parameters (d, η) lie in the stationary subset. The theorem provides the distribution of X_t conditionally on the past values. We can use the difference between the conditional mean and the observed value as an estimate of the filtered series. Let the first value of the filtered series be $\hat{Y}_1 = X_1$ (the procedure is the same for estimating Z_t).

$$\hat{Y}_t = X_t - \hat{m}_t, \qquad t = 2, \ldots, n,$$

where

$$\hat{m}_t = E[X_t | X_0, \cdots, X_{t-1}] = \sum_{j=1}^{t} \hat{\phi}_{tj} X_{t-j}$$

and $\hat{\phi}_{tj}$ are based on the estimated parameters $(\hat{d}_i, \hat{\eta}_i)$, $i = 1, 2$ (for more detail, refer to Chapter 1). After some preliminary results we noticed that few cycles of the iterative procedure are enough to reach a convergence and after few cycles (between five and ten) the results do not change. The algorithm does not affect too much the CPU time: in the following we report the average and the standard deviation of CPU time in seconds for different series length using the algorithm to estimate a two-factor Gegenbauer process with number of cycles $j = 10$:

n	mean	st.dev.
300	5.3540	0.0078
500	8.9764	0.0094
1000	19.0530	0.0128.

5.3 Results

We use the ACF bootstrap to replicate the series and compare with Monte Carlo results to see if the bootstrap brings any improvement to the estimating technique. First, we estimate the parameters in each surrogate series and then we consider the average of the estimates. As second option, we estimate the periodogram in each surrogate series and then we use the average value of the periodograms to estimate the parameters (see Section 3.2, Equations (3.5) and (3.6) for more details).

5.3.1 One-factor Gegenbauer process

First we consider the case of one frequency, we simulate Gegenbauer processes with $d = 0.2, 0.4$ and $\eta = 0.6, 0.7, 0.8, 0.9$. The results are given in Tables 5.2-5.5. For the estimation of one-frequency Gegenbauer process we summarize the results in the following. Increasing the sample size standard error and mean squared error slowly decrease and also the estimates are less biased. However, the convergence appears to be quite slow. The position of the peak influences the estimates, the closer η is to one, the larger the standard error and the mean squared error are: this happens because the periodogram is estimated over more values of the sample autocovariance function. The first bootstrap method (B1) is more biased than the Monte Carlo estimates but its standard error and its mean squared error are much smaller. The second bootstrap method (B2) has the same standard error as the first method but its estimates are very closed to the true value of the parameter especially for $d = 0.2$ but also for $d = 0.4$ this method gives on average better estimates. The choice of the Gegenbauer frequency η affects the estimates of the memory parameter d: it appears that it is better to estimate the frequency smoothing the periodogram in terms of less biased estimates, but the standard error and the mean squared error are both smaller without smoothing the periodogram. The local Whittle gives on average more biased estimates but with a smaller standard error than the GPH. The second bootstrap estimate is better using the local Whittle without smoothing the periodogram, whereas the first bootstrap method is better using the local Whittle and smoothing the periodogram. Overall we can say that the best combination of techniques is estimate with the local Whittle, smooth the periodogram and apply the first bootstrap method.

As regard the estimation of the frequency the results are given in Table 5.6. The bias is negligible and does not depend on the technique used. In terms of standard deviation it is definitely clear that the first bootstrap method outperforms the other two estimates.

5.3.2 Two-factor Gegenbauer process

The simulations of the previous paragraph confirm the validity of the semi-parametric estimators (GPH and local Whittle) and show the advantage of applying the ACF bootstrap. Now we extend the results to two-factor Gegenbauer processes.

It is quite common in the data to find long memory and a seasonal persistent component, i.e. to observe a spectrum like in Figure 5.1 with a peak at zero frequency and a peak at an unknown frequency. We run a wide experiment on this special case of the process solution of Equation (1.22) with $k = 2$. We set different values of the memory parameter $d_1 = 0.2, 0.4$, $d_2 = 0.2$, $\eta_1 = 1$ and $\eta_2 = 0.6, 0.7, 0.8, 0.9$. The results are given in Tables 5.7-5.10. Increasing the series length the estimates converge, they are less biased and with smaller standard deviation; however the convergence appears to be quite slow. As we expected, the long memory parameters influence each other, when $d_1 = 0.4$ the estimates of d_2 are always larger than when $d_1 = 0.2$, other conditions being equal. On the other side with η approaching one both estimates become less accurate. The Monte Carlo estimates are quite good, in the sense that their bias is small, even if the standard deviation is quite large. The algorithm (MC2) improves the estimates of d_1 in terms of

smaller bias and smaller standard error whereas d_2 is very biased; we must highlight that the algorithm starts estimating first d_1 and this could influence the estimates. It would be interesting to investigate if the estimates are very different starting the algorithm from d_2. Applying the ACF bootstrap helps to decrease the standard errors of the estimates. In the case of d_2 also the bias is reduced especially with the second technique (B2), whereas for d_1 the estimates based on bootstrap are not reliable. Between the GPH and the local Whittle, the latter has a smaller standard error. In terms of bias we observe approximately the same for both of d_2, whereas in the case of d_1 the estimates are so biased that the technique cannot be considered reliable.

Another little step in the generalization brings us to consider a two-factor Gegenbauer process with both peaks away from the zero frequency, i.e. $\eta_i \neq 1$, $i = 1, 2$. We choose quite closed peaks because this is the problematic case when the peaks most influence each other: the four models are given in Table 5.1. We run simulations with series length $n = 300, 500$.

	d	η
1^{st} model	(0.40 0.2)	(0.9 0.7)
2^{nd} model	(0.40 0.2)	(0.9 0.8)
3^{rd} model	(0.45 0.2)	(0.8 0.6)
4^{th} model	(0.45 0.2)	(0.8 0.7)

Table 5.1: *Values of the parameters $\eta = (\eta_1, \eta_2)$ and $d = (d_1, d_2)$ considered in the simulations.*

The results, given in Tables 5.12-5.15, are interesting. The bias is in most cases quite large suggesting the need of longer series to identify the peaks and estimate accurately the parameters. Also in this case estimating the Gegenbauer frequencies after smoothing the periodogram improves the memory parameter estimates in terms of smaller bias and standard error.

As in the previous experiment, the standard error increases when the Gegenbauer frequencies get closer. On the other hand, the bias is huge if we estimate the parameters independently. Applying the algorithm it is possible to correct the estimates removing the bias almost completely.

It is very interesting to notice that the algorithm (MC2) seems to work better when $\eta_i \neq 1$, $i = 1, 2$, i.e. there is not a pole at the zero frequency. Both estimates, d_1 and d_2, are much less biased and their standard errors are smaller.

Using the ACF bootstrap does not improve the estimates of the memory parameter, the Monte Carlo estimates still outperform the bootstrap. However, the bootstrap technique can be used for the distribution estimations of the frequency parameter. We suggest to use the bootstrap estimate of the frequency and in a second step the semi-parametric estimators, generalized GPH and local Whittle, for the memory parameters d_i, $i = 1, 2$.

5.4 Conclusions

In this chapter we showed the performance of two semi-parametric estimators, the generalized GPH and local Whittle, for Gegenbauer processes when we do not know the position of the frequency η. It was clear that the long memory parameters influence each other especially when the two Gegenbauer frequencies are close to each other. Thus, special care is required in these cases. The recursive algorithm given in Section 5.2 gave some good results especially when there is not a pole at the zero frequency.

Overall, we think that a combination of the Monte Carlo estimates, the bootstrap method and the recursive algorithm proposed in this chapter is the best solution to avoid bias but also to decrease the standard errors of the estimates. In particular, the algorithm gives good results if $\eta_i \neq 1$, whereas it is better to estimate the long memory parameter associated to the zero frequency ($\eta = 1$) with a semi-parametric estimator applying a bootstrap technique to decrease its standard error (see Chapter 3).

It is quite common to find in high frequency data more than seasonalities, i.e. more than two peaks in the spectral density. Thus, it would be interesting to extend the results of this chapter to $k > 2$ in a future work.

We smoothed the periodogram to estimate the Gegenbauer frequencies η and in most cases we had better results than with the raw periodogram. However, we considered only one type of window and we do not know how much the choice of Parzen window influences the results. Thus, an open problem is investigating the influence of the choice of the smoothing window on the estimates of the Gegenbauer frequencies η but also on the long memory parameters d.

	$d = 0.2$, GPH estimator											
	$\eta = 0.6$			$\eta = 0.7$			$\eta = 0.8$			$\eta = 0.9$		
n	MC	B1	B2	MC	B1	B2	MC	B1	B2	MC	B1	B2
	0.1513	**0.1398**	**0.1892**	**0.1569**	**0.1430**	**0.1936**	**0.1522**	**0.1386**	**0.1914**	**0.1399**	**0.1338**	**0.1883**
300	*0.1817*	*0.1169*	*0.1240*	*0.1792*	*0.1197*	*0.1260*	*0.1812*	*0.1200*	*0.1299*	*0.1748*	*0.1077*	*0.1186*
	0.0339	0.0145	0.0154	0.0328	0.0151	0.0159	0.0337	0.0153	0.0169	0.0318	0.0126	0.0141
	0.1459	**0.1385**	**0.1869**	**0.1501**	**0.1439**	**0.1935**	**0.1523**	**0.1418**	**0.1917**	**0.1496**	**0.1387**	**0.1901**
500	*0.1539*	*0.1020*	*0.1084*	*0.1611*	*0.1031*	*0.1114*	*0.1545*	*0.1038*	*0.1136*	*0.1541*	*0.0939*	*0.1069*
	0.0246	0.0112	0.0118	0.0268	0.0113	0.0124	0.0246	0.0115	0.0129	0.0245	0.0095	0.0114
	0.1634	**0.1471**	**0.1911**	**0.1524**	**0.1455**	**0.1896**	**0.1615**	**0.1509**	**0.1972**	**0.1590**	**0.1465**	**0.1924**
1000	*0.1340*	*0.0871*	*0.0973*	*0.1359*	*0.0834*	*0.0967*	*0.1309*	*0.0819*	*0.0937*	*0.1283*	*0.0741*	*0.0888*
	0.0183	0.0081	0.0095	0.0191	0.0075	0.0094	0.0175	0.0071	0.0088	0.0169	0.0059	0.0079
	0.1871	**0.1774**	**0.2196**	**0.1897**	**0.1778**	**0.2192**	**0.1805**	**0.1712**	**0.2149**	**0.1880**	**0.1747**	**0.2192**
300	*0.1945*	*0.1228*	*0.1288*	*0.1912*	*0.1187*	*0.1263*	*0.1881*	*0.1154*	*0.1234*	*0.1743*	*0.1039*	*0.1142*
	0.0379	0.0152	0.0167	0.0366	0.0142	0.0160	0.0355	0.0135	0.0153	0.0304	0.0109	0.0131
	0.1907	**0.1744**	**0.2120**	**0.1842**	**0.1726**	**0.2098**	**0.1819**	**0.1755**	**0.2139**	**0.1763**	**0.1703**	**0.2094**
500	*0.1641*	*0.1057*	*0.1147*	*0.1609*	*0.1062*	*0.1143*	*0.1625*	*0.0994*	*0.1107*	*0.1601*	*0.0903*	*0.1046*
	0.0270	0.0113	0.0132	0.0260	0.0114	0.0131	0.0265	0.0100	0.0123	0.0258	0.0083	0.0110
	0.1813	**0.1778**	**0.2105**	**0.1842**	**0.1790**	**0.2093**	**0.1880**	**0.1778**	**0.2105**	**0.1827**	**0.1766**	**0.2095**
1000	*0.1370*	*0.0871*	*0.0973*	*0.1325*	*0.0805*	*0.0900*	*0.1318*	*0.0774*	*0.0900*	*0.1344*	*0.0742*	*0.0904*
	0.0189	0.0077	0.0095	0.0176	0.0066	0.0081	0.0174	0.0061	0.0081	0.0182	0.0056	0.0082

Table 5.2: *GPH estimates of the memory parameter $d = 0.2$ for one frequency Gegenbauer process $(1 - 2\eta B + B^2)^d X_t = \varepsilon_t$, for different values of $\eta = 0.6, 0.7, 0.8, 0.9$ with MC Monte Carlo, B1 and B2 bootstrap estimates, where n is the sample size, 300 bootstrap replications, 1000 repetitions of the experiment, average value in boldface, standard error in italic and mean squared error in normal font: in the top part of the table the Gegenbauer frequency is estimated from the raw periodogram, whereas in the bottom part the Gegenbauer frequency is estimated after smoothing periodogram.*

	$d = 0.2$, local Whittle estimator											
	$\eta = 0.6$			$\eta = 0.7$			$\eta = 0.8$			$\eta = 0.9$		
n	MC	B1	B2	MC	B1	B2	MC	B1	B2	MC	B1	B2
	0.1394	**0.1205**	**0.1833**	**0.1453**	**0.1241**	**0.1893**	**0.1473**	**0.1252**	**0.1936**	**0.1409**	**0.1195**	**0.1947**
300	*0.1593*	*0.1295*	*0.1297*	*0.1635*	*0.1355*	*0.1331*	*0.1712*	*0.1408*	*0.1428*	*0.1650*	*0.1288*	*0.1327*
	0.0266	0.0185	0.0169	0.0277	0.0200	0.0178	0.0303	0.0214	0.0204	0.0284	0.0183	0.0176
	0.1372	**0.1183**	**0.1792**	**0.1474**	**0.1260**	**0.1889**	**0.1472**	**0.1247**	**0.1893**	**0.1475**	**0.1210**	**0.1912**
500	*0.1360*	*0.1136*	*0.1131*	*0.1448*	*0.1169*	*0.1185*	*0.1422*	*0.1159*	*0.1194*	*0.1401*	*0.1068*	*0.1148*
	0.0196	0.0145	0.0129	0.0218	0.0150	0.0141	0.0210	0.0148	0.0143	0.0204	0.0128	0.0132
	0.1526	**0.1276**	**0.1832**	**0.1493**	**0.1259**	**0.1828**	**0.1607**	**0.1336**	**0.1943**	**0.1578**	**0.1295**	**0.1917**
1000	*0.1142*	*0.0926*	*0.0990*	*0.1184*	*0.0905*	*0.0993*	*0.1153*	*0.0886*	*0.0975*	*0.1116*	*0.0817*	*0.0943*
	0.0136	0.0096	0.0099	0.0146	0.0092	0.0099	0.0137	0.0087	0.0095	0.0128	0.0075	0.0089
	0.1770	**0.1536**	**0.2212**	**0.1795**	**0.1561**	**0.2234**	**0.1718**	**0.1481**	**0.2204**	**0.1852**	**0.1568**	**0.2324**
300	*0.1743*	*0.1348*	*0.1301*	*0.1732*	*0.1330*	*0.1289*	*0.1670*	*0.1272*	*0.1272*	*0.1664*	*0.1209*	*0.1228*
	0.0306	0.0187	0.0171	0.0301	0.0182	0.0168	0.0282	0.0169	0.0163	0.0278	0.0151	0.0153
	0.1742	**0.1493**	**0.2089**	**0.1772**	**0.1505**	**0.2113**	**0.1725**	**0.1536**	**0.2176**	**0.1747**	**0.1503**	**0.2178**
500	*0.1458*	*0.1139*	*0.1149*	*0.1450*	*0.1151*	*0.1151*	*0.1473*	*0.1088*	*0.1127*	*0.1423*	*0.1001*	*0.1070*
	0.0214	0.0136	0.0132	0.0212	0.0138	0.0133	0.0219	0.0123	0.0128	0.0204	0.0105	0.0115
	0.1735	**0.1548**	**0.2083**	**0.1776**	**0.1572**	**0.2074**	**0.1833**	**0.1569**	**0.2105**	**0.1823**	**0.1586**	**0.2146**
1000	*0.1199*	*0.0910*	*0.0946*	*0.1150*	*0.0851*	*0.0880*	*0.1139*	*0.0814*	*0.0888*	*0.1184*	*0.0799*	*0.0910*
	0.0146	0.0087	0.0090	0.0133	0.0076	0.0078	0.0130	0.0069	0.0079	0.0141	0.0067	0.0083

Table 5.3: *Local Whittle estimates of the memory parameter* $d = 0.2$ *for one frequency Gegenbauer process* $(1 - 2\eta B + B^2)^d X_t = \varepsilon_t$, *for different values of* $\eta = 0.6, 0.7, 0.8, 0.9$ *with MC Monte Carlo, B1 and B2 bootstrap estimates, where* n *is the sample size,* 300 *bootstrap replications,* 1000 *repetitions of the experiment, average value in boldface, standard error in italic and mean squared error in normal font: in the top part of the table the Gegenbauer frequency is estimated from the raw periodogram, whereas in the bottom part the Gegenbauer frequency is estimated after smoothing periodogram.*

	$d = 0.4$, GPH estimator											
	$\eta = 0.6$			$\eta = 0.7$			$\eta = 0.8$			$\eta = 0.9$		
n	MC	B1	B2	MC	B1	B2	MC	B1	B2	MC	B1	B2
	0.3563	**0.3720**	**0.4405**	**0.3520**	**0.3645**	**0.4314**	**0.3597**	**0.3644**	**0.4331**	**0.3547**	**0.3598**	**0.4294**
300	*0.1989*	*0.1253*	*0.1331*	*0.1948*	*0.1205*	*0.1299*	*0.1834*	*0.1171*	*0.1247*	*0.1749*	*0.1092*	*0.1203*
	0.0403	0.0159	0.0182	0.0388	0.0148	0.0171	0.0342	0.0140	0.0158	0.0313	0.0123	0.0147
	0.3595	**0.3713**	**0.4329**	**0.3715**	**0.3730**	**0.4332**	**0.3655**	**0.3686**	**0.4313**	**0.3538**	**0.3606**	**0.4209**
500	*0.1623*	*0.1067*	*0.1147*	*0.1674*	*0.1056*	*0.1151*	*0.1615*	*0.1023*	*0.1104*	*0.1577*	*0.1018*	*0.1127*
	0.0269	0.0116	0.0134	0.0283	0.0113	0.0135	0.0265	0.0107	0.0124	0.0255	0.0107	0.0128
	0.3708	**0.3742**	**0.4265**	**0.3765**	**0.3760**	**0.4290**	**0.3605**	**0.3652**	**0.4162**	**0.3752**	**0.3705**	**0.4231**
1000	*0.1349*	*0.0880*	*0.0964*	*0.1406*	*0.0897*	*0.0998*	*0.1434*	*0.0927*	*0.1038*	*0.1321*	*0.0873*	*0.0967*
	0.0184	0.0079	0.0094	0.0199	0.0081	0.0101	0.0210	0.0088	0.0108	0.0176	0.0078	0.0095
	0.4021	**0.4117**	**0.4887**	**0.3952**	**0.4057**	**0.4806**	**0.3879**	**0.3985**	**0.4709**	**0.3850**	**0.3971**	**0.4714**
300	*0.1927*	*0.1191*	*0.1345*	*0.1925*	*0.1188*	*0.1333*	*0.1903*	*0.1138*	*0.1287*	*0.1850*	*0.1093*	*0.1290*
	0.0371	0.0142	0.0203	0.0371	0.0141	0.0196	0.0363	0.0130	0.0179	0.0343	0.0120	0.0180
	0.4036	**0.4067**	**0.4635**	**0.3939**	**0.4027**	**0.4615**	**0.4016**	**0.4069**	**0.4650**	**0.3966**	**0.3997**	**0.4580**
500	*0.1679*	*0.1047*	*0.1174*	*0.1744*	*0.1072*	*0.1199*	*0.1690*	*0.1040*	*0.1183*	*0.1597*	*0.0968*	*0.1128*
	0.0282	0.0110	0.0148	0.0304	0.0115	0.0153	0.0286	0.0108	0.0150	0.0255	0.0094	0.0135
	0.4012	**0.4058**	**0.4505**	**0.4002**	**0.4068**	**0.4529**	**0.3935**	**0.3981**	**0.4446**	**0.4015**	**0.4037**	**0.4506**
1000	*0.1412*	*0.0849*	*0.0977*	*0.1405*	*0.0845*	*0.0971*	*0.1375*	*0.0844*	*0.0966*	*0.1404*	*0.0859*	*0.0984*
	0.0199	0.0072	0.0101	0.0197	0.0072	0.0100	0.0189	0.0071	0.0097	0.0197	0.0074	0.0102

Table 5.4: *GPH estimates of the memory parameter $d = 0.4$ for one frequency Gegenbauer process $(1 - 2\eta B + B^2)^d X_t = \varepsilon_t$, for different values of $\eta = 0.6, 0.7, 0.8, 0.9$ with MC Monte Carlo, B1 and B2 bootstrap estimates, where n is the sample size, 300 bootstrap replications, 1000 repetitions of the experiment, average value in boldface, standard error in italic and mean squared error in normal font: in the top part of the table the Gegenbauer frequency is estimated from the raw periodogram, whereas in the bottom part the Gegenbauer frequency is estimated after smoothing periodogram.*

	$d = 0.4$, local Whittle estimator											
	$\eta = 0.6$			$\eta = 0.7$			$\eta = 0.8$			$\eta = 0.9$		
n	MC	B1	B2	MC	B1	B2	MC	B1	B2	MC	B1	B2
	0.3634	**0.3668**	**0.4453**	**0.3607**	**0.3611**	**0.4400**	**0.3691**	**0.3669**	**0.4482**	**0.3796**	**0.3743**	**0.4618**
300	*0.1744*	*0.1384*	*0.1373*	*0.1742*	*0.1342*	*0.1376*	*0.1739*	*0.1336*	*0.1343*	*0.1713*	*0.1294*	*0.1351*
	0.0309	0.0195	0.0194	0.0309	0.0184	0.0194	0.0306	0.0181	0.0187	0.0295	0.0169	0.0193
	0.3628	**0.3620**	**0.4332**	**0.3715**	**0.3673**	**0.4363**	**0.3746**	**0.3685**	**0.4425**	**0.3742**	**0.3675**	**0.4420**
500	*0.1470*	*0.1167*	*0.1199*	*0.1490*	*0.1155*	*0.1194*	*0.1451*	*0.1147*	*0.1175*	*0.1450*	*0.1141*	*0.1208*
	0.0220	0.0140	0.0146	0.0224	0.0136	0.0146	0.0212	0.0134	0.0142	0.0212	0.0133	0.0150
	0.3706	**0.3629**	**0.4225**	**0.3799**	**0.3688**	**0.4304**	**0.3738**	**0.3613**	**0.4219**	**0.3893**	**0.3744**	**0.4371**
1000	*0.1126*	*0.0929*	*0.0976*	*0.1207*	*0.0959*	*0.1020*	*0.1242*	*0.0995*	*0.1067*	*0.1198*	*0.0951*	*0.1023*
	0.0129	0.0089	0.0096	0.0147	0.0094	0.0106	0.0156	0.0102	0.0115	0.0144	0.0092	0.0107
	0.4006	**0.4012**	**0.5006**	**0.4037**	**0.3998**	**0.4994**	**0.4035**	**0.3987**	**0.4980**	**0.4141**	**0.4121**	**0.5148**
300	*0.1750*	*0.1314*	*0.1364*	*0.1768*	*0.1319*	*0.1377*	*0.1823*	*0.1302*	*0.1351*	*0.1802*	*0.1276*	*0.1390*
	0.0306	0.0173	0.0215	0.0313	0.0174	0.0218	0.0333	0.0170	0.0209	0.0325	0.0163	0.0232
	0.3968	**0.3924**	**0.4698**	**0.3955**	**0.3936**	**0.4751**	**0.4066**	**0.4034**	**0.4841**	**0.4175**	**0.4069**	**0.4889**
500	*0.1497*	*0.1116*	*0.1167*	*0.1554*	*0.1154*	*0.1197*	*0.1607*	*0.1151*	*0.1218*	*0.1543*	*0.1096*	*0.1182*
	0.0224	0.0125	0.0148	0.0242	0.0133	0.0157	0.0258	0.0132	0.0166	0.0239	0.0120	0.0159
	0.4020	**0.3921**	**0.4552**	**0.4022**	**0.3958**	**0.4607**	**0.4023**	**0.3917**	**0.4553**	**0.4149**	**0.4058**	**0.4710**
1000	*0.1244*	*0.0892*	*0.0962*	*0.1278*	*0.0895*	*0.0971*	*0.1242*	*0.0902*	*0.0973*	*0.1283*	*0.0919*	*0.1006*
	0.0155	0.0080	0.0099	0.0163	0.0080	0.0101	0.0154	0.0081	0.0101	0.0165	0.0085	0.0112

Table 5.5: *Local Whittle estimates of the memory parameter* $d = 0.4$ *for one frequency Gegenbauer process* $(1 - 2\eta B + B^2)^d X_t = \varepsilon_t$, *for different values of* $\eta = 0.6, 0.7, 0.8, 0.9$ *with MC Monte Carlo, B1 and B2 bootstrap estimates, where* n *is the sample size,* 300 *bootstrap replications,* 1000 *repetitions of the experiment, average value in boldface, standard error in italic and mean squared error in normal font: in the top part of the table the Gegenbauer frequency is estimated from the raw periodogram, whereas in the bottom part the Gegenbauer frequency is estimated after smoothing periodogram.*

	$\eta = 0.6$ GPH			$\eta = 0.7$ local Whittle			$\eta = 0.8$ GPH			$\eta = 0.9$ local Whittle		
n	MC	B1	B2	MC	B1	B2	MC	B1	B2	MC	B1	B2
	0.6004	**0.6004**	**0.6007**	**0.7000**	**0.6999**	**0.6997**	**0.8005**	**0.8004**	**0.8005**	**0.9002**	**0.9005**	**0.9003**
300	*0.0113*	*0.0055*	*0.0116*	*0.0111*	*0.0055*	*0.0115*	*0.0107*	*0.0052*	*0.0111*	*0.0093*	*0.0048*	*0.0096*
	0.0001	0.0000	0.0001	0.0001	0.0000	0.0001	0.0001	0.0000	0.0001	0.0001	0.0000	0.0001
	0.6005	**0.6003**	**0.6005**	**0.7000**	**0.7000**	**0.6999**	**0.8004**	**0.8003**	**0.8003**	**0.9008**	**0.9005**	**0.9007**
500	*0.0101*	*0.0051*	*0.0104*	*0.0096*	*0.0047*	*0.0097*	*0.0093*	*0.0046*	*0.0095*	*0.0089*	*0.0044*	*0.0087*
	0.0001	0.0000	0.0001	0.0001	0.0000	0.0001	0.0001	0.0000	0.0001	0.0001	0.0000	0.0001
	0.6000	**0.6001**	**0.5999**	**0.7004**	**0.7002**	**0.7003**	**0.8002**	**0.8002**	**0.8005**	**0.9005**	**0.9004**	**0.9006**
1000	*0.0082*	*0.0042*	*0.0084*	*0.0084*	*0.0042*	*0.0085*	*0.0080*	*0.0038*	*0.0080*	*0.0068*	*0.0033*	*0.0068*
	0.0001	0.0000	0.0001	0.0001	0.0000	0.0001	0.0001	0.0000	0.0001	0.0000	0.0000	0.0000
	0.5998	**0.5999**	**0.6000**	**0.7006**	**0.7004**	**0.7004**	**0.8000**	**0.8001**	**0.8000**	**0.9000**	**0.9002**	**0.9001**
300	*0.0078*	*0.0042*	*0.0079*	*0.0077*	*0.0040*	*0.0079*	*0.0067*	*0.0039*	*0.0069*	*0.0061*	*0.0036*	*0.0063*
	0.0001	0.0000	0.0001	0.0001	0.0000	0.0001	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.6001	**0.6002**	**0.6002**	**0.7001**	**0.7001**	**0.7000**	**0.8001**	**0.8001**	**0.8002**	**0.9001**	**0.9002**	**0.9001**
500	*0.0062*	*0.0036*	*0.0064*	*0.0061*	*0.0036*	*0.0063*	*0.0054*	*0.0032*	*0.0055*	*0.0045*	*0.0027*	*0.0045*
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.6000	**0.6001**	**0.6001**	**0.7000**	**0.7001**	**0.7000**	**0.8001**	**0.8001**	**0.8001**	**0.9001**	**0.9001**	**0.9000**
1000	*0.0043*	*0.0027*	*0.0043*	*0.0041*	*0.0023*	*0.0041*	*0.0036*	*0.0022*	*0.0036*	*0.0032*	*0.0018*	*0.0030*
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Table 5.6: *Estimation of the frequency parameter η for one frequency Gegenbauer process $(1 - 2\eta B + B^2)^d X_t = \varepsilon_t$, for different values of $\eta = 0.6, 0.7, 0.8, 0.9$ and different estimation methods, GPH and local Whittle with MC Monte Carlo, B1 and B2 bootstrap estimates, where n is the sample size, 300 bootstrap replications, 1000 repetitions of the experiment, average value in boldface, standard error in italic and mean squared error in normal font: in the top part of the table the Gegenbauer frequency is estimated from the raw periodogram, whereas in the bottom part the Gegenbauer frequency is estimated after smoothing periodogram.*

		$d_0 = 0.2, 0.4$, GPH estimator							
		smoothed periodogram				periodogram			
n	η	MC1	MC2	B1	B2	MC1	MC2	B1	B2
		0.1750	**0.1937**	**0.1405**	**0.1183**	**0.1725**	**0.1859**	**0.1393**	**0.1214**
	0.8	*0.2021*	*0.1971*	*0.0066*	*0.1276*	*0.2065*	*0.1992*	*0.0067*	*0.1317*
300		0.0415	0.0389	0.0036	0.0230	0.0434	0.0399	0.0037	0.0235
		0.1505	**0.1955**	**0.1108**	**0.0184**	**0.1491**	**0.1839**	**0.1099**	**0.0155**
	0.9	*0.1962*	*0.1846*	*0.0066*	*0.1300*	*0.2021*	*0.1862*	*0.0064*	*0.1304*
		0.0410	0.0341	0.0080	0.0499	0.0434	0.0349	0.0082	0.0510
		0.1953	**0.2059**	**0.1653**	**0.1532**	**0.1937**	**0.2023**	**0.1649**	**0.1545**
	0.8	*0.1658*	*0.1631*	*0.0057*	*0.1054*	*0.1713*	*0.1689*	*0.0058*	*0.1127*
500		0.0275	0.0266	0.0012	0.0133	0.0294	0.0285	0.0013	0.0148
		0.1732	**0.1967**	**0.1461**	**0.1123**	**0.1710**	**0.1887**	**0.1455**	**0.1151**
	0.9	*0.1678*	*0.1604*	*0.0056*	*0.1096*	*0.1736*	*0.1676*	*0.0054*	*0.1109*
		0.0289	0.0257	0.0029	0.0197	0.0310	0.0282	0.0030	0.0195
		0.3910	**0.4082**	**0.3460**	**0.3257**	**0.3913**	**0.4018**	**0.3460**	**0.3276**
	0.8	*0.2119*	*0.2032*	*0.0065*	*0.1350*	*0.2097*	*0.2046*	*0.0066*	*0.1387*
300		0.0450	0.0414	0.0030	0.0237	0.0441	0.0419	0.0030	0.0245
		0.3474	**0.3820**	**0.3045**	**0.2133**	**0.3612**	**0.3844**	**0.3156**	**0.2177**
	0.9	*0.2059*	*0.1921*	*0.0063*	*0.1369*	*0.2030*	*0.1904*	*0.0066*	*0.1351*
		0.0452	0.0372	0.0092	0.0536	0.0427	0.0365	0.0072	0.0515
		0.4037	**0.4128**	**0.3702**	**0.3669**	**0.4007**	**0.4074**	**0.3643**	**0.3606**
	0.8	*0.1694*	*0.1669*	*0.0057*	*0.1116*	*0.1645*	*0.1627*	*0.0054*	*0.1115*
500		0.0287	0.0280	0.0009	0.0136	0.0271	0.0265	0.0013	0.0140
		0.3814	**0.3983**	**0.3462**	**0.3157**	**0.3811**	**0.3935**	**0.3475**	**0.3184**
	0.9	*0.1687*	*0.1624*	*0.0052*	*0.1116*	*0.1692*	*0.1626*	*0.0053*	*0.1155*
		0.0288	0.0264	0.0029	0.0196	0.0290	0.0265	0.0028	0.0200

Table 5.7: *GPH estimates of the memory parameter d_0 for two-factor Gegenbauer process $(1-B)^{d_0}(1-2\eta B+B^2)^{d_1}X_t = \varepsilon_t$, for different values of $\eta = 0.8, 0.9$ and different Monte Carlo procedures (MC1, MC2), B1 and B2 bootstrap estimates, where n is the sample size,* 300 *bootstrap replications,* 1000 *repetitions of the experiment, average value (boldface), standard error (italic), mean squared error (normal font): in the top part of the table the Gegenbauer frequency is estimated from the raw periodogram, whereas in the bottom part the Gegenbauer frequency is estimated after smoothing periodogram.*

		$d_0 = 0.2, 0.4$, local Whittle estimator							
		smoothed periodogram				periodogram			
n	η	MC1	MC2	B1	B2	MC1	MC2	B1	B2
		0.1526	**0.1712**	**0.1118**	**0.1143**	**0.1523**	**0.1660**	**0.1117**	**0.1158**
	0.8	*0.1656*	*0.1612*	*0.0058*	*0.1246*	*0.1668*	*0.1620*	*0.0056*	*0.1282*
300		0.0297	0.0268	0.0078	0.0229	0.0301	0.0274	0.0078	0.0235
		0.1248	**0.1712**	**0.0829**	**0.0022**	**0.1243**	**0.1611**	**0.0834**	**-0.0001**
	0.9	*0.1684*	*0.1544*	*0.0060*	*0.1306*	*0.1688*	*0.1544*	*0.0055*	*0.1298*
		0.0340	0.0247	0.0137	0.0562	0.0342	0.0253	0.0136	0.0569
		0.1746	**0.1852**	**0.1429**	**0.1491**	**0.1743**	**0.1825**	**0.1416**	**0.1509**
	0.8	*0.1390*	*0.1366*	*0.0051*	*0.1020*	*0.1383*	*0.1363*	*0.0050*	*0.1107*
500		0.0200	0.0189	0.0033	0.0130	0.0198	0.0189	0.0034	0.0147
		0.1550	**0.1786**	**0.1236**	**0.1075**	**0.1571**	**0.1754**	**0.1242**	**0.1113**
	0.9	*0.1411*	*0.1350*	*0.0050*	*0.1071*	*0.1416*	*0.1364*	*0.0049*	*0.1068*
		0.0219	0.0187	0.0059	0.0200	0.0219	0.0192	0.0058	0.0193
		0.3462	**0.3601**	**0.2890**	**0.3162**	**0.3478**	**0.3570**	**0.2904**	**0.3191**
	0.8	*0.1440*	*0.1364*	*0.0051*	*0.1236*	*0.1484*	*0.1413*	*0.0051*	*0.1252*
300		0.0236	0.0202	0.0124	0.0223	0.0247	0.0218	0.0120	0.0222
		0.3138	**0.3453**	**0.2565**	**0.1983**	**0.3250**	**0.3452**	**0.2661**	**0.2018**
	0.9	*0.1567*	*0.1401*	*0.0052*	*0.1356*	*0.1501*	*0.1363*	*0.0054*	*0.1329*
		0.0320	0.0226	0.0206	0.0591	0.0281	0.0216	0.0180	0.0569
		0.3705	**0.3777**	**0.3229**	**0.3571**	**0.3690**	**0.3747**	**0.3197**	**0.3545**
	0.8	*0.1177*	*0.1148*	*0.0045*	*0.0991*	*0.1164*	*0.1137*	*0.0042*	*0.1020*
500		0.0147	0.0137	0.0060	0.0117	0.0145	0.0136	0.0065	0.0125
		0.3521	**0.3680**	**0.3047**	**0.3101**	**0.3521**	**0.3641**	**0.3050**	**0.3112**
	0.9	*0.1256*	*0.1185*	*0.0044*	*0.1058*	*0.1269*	*0.1194*	*0.0043*	*0.1068*
		0.0181	0.0151	0.0091	0.0193	0.0184	0.0155	0.0090	0.0193

Table 5.8: *Local Whittle estimates of the memory parameter d_0 for two-factor Gegenbauer process $(1-B)^{d_0}(1-2\eta B+B^2)^{d_1}X_t = \varepsilon_t$, for different values of $\eta = 0.8, 0.9$ and different Monte Carlo procedures (MC1, MC2), B1 and B2 bootstrap estimates, where n is the sample size,* 300 *bootstrap replications,* 1000 *repetitions of the experiment, average value (boldface), standard error (italic), mean squared error (normal font): in the top part of the table the Gegenbauer frequency is estimated from the raw periodogram, whereas in the bottom part the Gegenbauer frequency is estimated after smoothing periodogram.*

		$d_1 = 0.2$, GPH estimator							
		smoothed periodogram				periodogram			
n	η	MC1	MC2	B1	B2	MC1	MC2	B1	B2
		0.2069	**0.1519**	**0.1961**	**0.2107**	**0.1659**	**0.1115**	**0.1565**	**0.2109**
	0.8	*0.1859*	*0.1857*	*0.0059*	*0.1270*	*0.1795*	*0.1826*	*0.0059*	*0.1246*
300		0.0346	0.0368	0.0000	0.0163	0.0334	0.0412	0.0019	0.0156
		0.2202	**0.1473**	**0.2087**	**0.2246**	**0.1826**	**0.1171**	**0.1677**	**0.2236**
	0.9	*0.1804*	*0.1747*	*0.0059*	*0.1223*	*0.1761*	*0.1748*	*0.0059*	*0.1160*
		0.0329	0.0333	0.0001	0.0156	0.0313	0.0374	0.0011	0.0140
		0.2064	**0.1596**	**0.1962**	**0.2145**	**0.1743**	**0.1280**	**0.1632**	**0.2146**
	0.8	*0.1637*	*0.1652*	*0.0056*	*0.1106*	*0.1616*	*0.1652*	*0.0056*	*0.1115*
500		0.0268	0.0289	0.0000	0.0125	0.0268	0.0325	0.0014	0.0126
		0.2188	**0.1575**	**0.2048**	**0.2205**	**0.1810**	**0.1237**	**0.1699**	**0.2204**
	0.9	*0.1555*	*0.1562*	*0.0054*	*0.1101*	*0.1567*	*0.1571*	*0.0054*	*0.1068*
		0.0245	0.0262	0.0001	0.0125	0.0249	0.0305	0.0009	0.0118
		0.2448	**0.1334**	**0.2236**	**0.2465**	**0.1975**	**0.0833**	**0.1843**	**0.2471**
	0.8	*0.1759*	*0.1839*	*0.0060*	*0.1270*	*0.1855*	*0.1856*	*0.0059*	*0.1299*
300		0.0330	0.0382	0.0006	0.0183	0.0344	0.0481	0.0003	0.0191
		0.2593	**0.1153**	**0.2442**	**0.2693**	**0.2183**	**0.0774**	**0.2037**	**0.2668**
	0.9	*0.1716*	*0.1661*	*0.0059*	*0.1203*	*0.1724*	*0.1678*	*0.0057*	*0.1212*
		0.0330	0.0348	0.0020	0.0193	0.0301	0.0432	0.0000	0.0191
		0.2260	**0.1344**	**0.2152**	**0.2356**	**0.1934**	**0.1036**	**0.1818**	**0.2382**
	0.8	*0.1534*	*0.1569*	*0.0050*	*0.1072*	*0.1589*	*0.1571*	*0.0052*	*0.1142*
500		0.0242	0.0289	0.0003	0.0128	0.0253	0.0340	0.0004	0.0145
		0.2352	**0.1173**	**0.2325**	**0.2514**	**0.2031**	**0.0855**	**0.1952**	**0.2498**
	0.9	*0.1599*	*0.1600*	*0.0054*	*0.1091*	*0.1542*	*0.1596*	*0.0052*	*0.1057*
		0.0268	0.0325	0.0011	0.0145	0.0238	0.0386	0.0001	0.0136

Table 5.9: *GPH estimates of the memory parameter d_0 for two-factor Gegenbauer process $(1-B)^{d_0}(1-2\eta B+B^2)^{d_1}X_t = \varepsilon_t$, for different values of $\eta = 0.8, 0.9$ and different Monte Carlo procedures (MC1, MC2), B1 and B2 bootstrap estimates, where n is the sample size,* 300 *bootstrap replications,* 1000 *repetitions of the experiment, average value (boldface), standard error (italic), mean squared error (normal font): in the top part of the table the Gegenbauer frequency is estimated from the raw periodogram, whereas in the bottom part the Gegenbauer frequency is estimated after smoothing periodogram.*

		$d_1 = 0.2$, local Whittle estimator							
		smoothed periodogram				periodogram			
n	η	MC1	MC2	B1	B2	MC1	MC2	B1	B2
		0.2006	**0.1485**	**0.1766**	**0.2109**	**0.1623**	**0.1098**	**0.1428**	**0.2127**
	0.8	*0.1698*	*0.1708*	*0.0060*	*0.1382*	*0.1718*	*0.1712*	*0.0062*	*0.1345*
300		0.0288	0.0318	0.0006	0.0192	0.0309	0.0375	0.0033	0.0183
		0.2227	**0.1508**	**0.1981**	**0.2351**	**0.1848**	**0.1167**	**0.1583**	**0.2335**
	0.9	*0.1692*	*0.1625*	*0.0062*	*0.1362*	*0.1644*	*0.1616*	*0.0062*	*0.1294*
		0.0292	0.0288	0.0000	0.0198	0.0273	0.0331	0.0018	0.0179
		0.2001	**0.1545**	**0.1774**	**0.2127**	**0.1681**	**0.1232**	**0.1458**	**0.2112**
	0.8	*0.1462*	*0.1475*	*0.0054*	*0.1174*	*0.1387*	*0.1411*	*0.0054*	*0.1173*
500		0.0214	0.0238	0.0005	0.0139	0.0203	0.0258	0.0030	0.0139
		0.2183	**0.1572**	**0.1892**	**0.2232**	**0.1846**	**0.1239**	**0.1566**	**0.2243**
	0.9	*0.1451*	*0.1463*	*0.0057*	*0.1200*	*0.1424*	*0.1413*	*0.0056*	*0.1168*
		0.0214	0.0232	0.0001	0.0149	0.0205	0.0258	0.0019	0.0142
		0.2380	**0.1305**	**0.2097**	**0.2495**	**0.2001**	**0.0863**	**0.1751**	**0.2514**
	0.8	*0.1643*	*0.1660*	*0.0063*	*0.1373*	*0.1739*	*0.1764*	*0.0063*	*0.1422*
300		0.0284	0.0324	0.0001	0.0213	0.0302	0.0440	0.0007	0.0229
		0.2697	**0.1238**	**0.2410**	**0.2848**	**0.2247**	**0.0778**	**0.2015**	**0.2811**
	0.9	*0.1696*	*0.1626*	*0.0061*	*0.1350*	*0.1691*	*0.1631*	*0.0064*	*0.1360*
		0.0336	0.0322	0.0017	0.0254	0.0292	0.0415	0.0000	0.0251
		0.2181	**0.1281**	**0.1981**	**0.2341**	**0.1911**	**0.1005**	**0.1688**	**0.2370**
	0.8	*0.1398*	*0.1384*	*0.0053*	*0.1159*	*0.1453*	*0.1446*	*0.0055*	*0.1207*
500		0.0199	0.0243	0.0000	0.0146	0.0212	0.0308	0.0010	0.0159
		0.2469	**0.1241**	**0.2225**	**0.2590**	**0.2130**	**0.0894**	**0.1872**	**0.2576**
	0.9	*0.1461*	*0.1439*	*0.0054*	*0.1207*	*0.1389*	*0.1447*	*0.0052*	*0.1157*
		0.0235	0.0265	0.0005	0.0180	0.0195	0.0332	0.0002	0.0167

Table 5.10: *Local Whittle estimates of the memory parameter d_0 for two-factor Gegenbauer process $(1-B)^{d_0}(1-2\eta B+B^2)^{d_1}X_t = \varepsilon_t$, for different values of $\eta = 0.8, 0.9$ and different Monte Carlo procedures (MC1, MC2), B1 and B2 bootstrap estimates, where n is the sample size,* 300 *bootstrap replications,* 1000 *repetitions of the experiment, average value (boldface), standard error (italic), mean squared error (normal font): in the top part of the table the Gegenbauer frequency is estimated from the raw periodogram, whereas in the bottom part the Gegenbauer frequency is estimated after smoothing periodogram.*

η	MC1	gph	lw	B1	B2
0.8	**0.8003**	**0.7999**	**0.8000**	**0.8002**	**0.8005**
	0.0095	*0.0096*	*0.0096*	*0.0003*	*0.0096*
0.9	**0.9004**	**0.8998**	**0.8998**	**0.9006**	**0.9002**
	0.0086	*0.0087*	*0.0087*	*0.0003*	*0.0088*
0.8	**0.8007**	**0.8001**	**0.8002**	**0.8006**	**0.8008**
	0.0094	*0.0095*	*0.0095*	*0.0003*	*0.0096*
0.9	**0.9009**	**0.8998**	**0.8998**	**0.9011**	**0.9012**
	0.0086	*0.0084*	*0.0085*	*0.0003*	*0.0085*
0.8	**0.8008**	**0.8002**	**0.8004**	**0.8005**	**0.8011**
	0.0114	*0.0113*	*0.0113*	*0.0004*	*0.0092*
0.9	**0.9010**	**0.9000**	**0.9001**	**0.9010**	**0.9012**
	0.0104	*0.0105*	*0.0105*	*0.0003*	*0.0086*
0.8	**0.8000**	**0.7993**	**0.7993**	**0.8005**	**0.8011**
	0.0111	*0.0110*	*0.0111*	*0.0004*	*0.0096*
0.9	**0.9009**	**0.8992**	**0.8993**	**0.9011**	**0.9008**
	0.0103	*0.0102*	*0.0102*	*0.0003*	*0.0086*

η	MC1	gph	lw	B1	B2
0.8	**0.8009**	**0.8004**	**0.8004**	**0.8004**	**0.8004**
	0.0128	*0.0128*	*0.0128*	*0.0004*	*0.0112*
0.9	**0.8998**	**0.8991**	**0.8993**	**0.9006**	**0.9002**
	0.0118	*0.0117*	*0.0118*	*0.0004*	*0.0102*
0.8	**0.8013**	**0.8006**	**0.8006**	**0.8011**	**0.8014**
	0.0127	*0.0127*	*0.0127*	*0.0004*	*0.0111*
0.9	**0.9012**	**0.8997**	**0.9000**	**0.9014**	**0.9016**
	0.0113	*0.0115*	*0.0115*	*0.0004*	*0.0097*
0.8	**0.8002**	**0.7997**	**0.7997**	**0.8004**	**0.8002**
	0.0108	*0.0108*	*0.0108*	*0.0003*	*0.0113*
0.9	**0.9005**	**0.8998**	**0.8999**	**0.9007**	**0.9005**
	0.0098	*0.0099*	*0.0099*	*0.0003*	*0.0101*
0.8	**0.8006**	**0.7998**	**0.7999**	**0.8005**	**0.8010**
	0.0101	*0.0101*	*0.0101*	*0.0003*	*0.0104*
0.9	**0.9013**	**0.8995**	**0.8998**	**0.9011**	**0.9014**
	0.0095	*0.0095*	*0.0095*	*0.0003*	*0.0098*

Table 5.11: *Estimates of the Gegenbauer frequency η for two-factor Gegenbauer process $(1-B)^{d_1}(1-2\eta B+B^2)^{d_2}X_t=\varepsilon_t$, for different values of $\eta=0.8, 0.9$, different Monte Carlo procedures (MC1, MC2-gph, MC2-lw), B1 and B2 bootstrap estimates, where n is the sample size,* 300 *bootstrap replications,* 1000 *repetitions of the experiment, average value (boldface), standard error (italic): in the top part of the table the Gegenbauer frequency is estimated from the raw periodogram, whereas in the bottom part the Gegenbauer frequency is estimated after smoothing periodogram.*

		$d_1 = 0.4, 0.45$, GPH estimates							
		smoothed periodogram				periodogram			
n	(d, η)	MC1	MC2	B1	B2	MC1	MC2	B1	B2
300	(0.4,0.2,0.9,0.7)	**0.2810**	**0.3777**	**0.2309**	**0.2539**	**0.1736**	**0.3287**	**0.1907**	**0.2529**
		0.2181	*0.1880*	*0.0065*	*0.1280*	*0.1857*	*0.1835*	*0.0063*	*0.1323*
		0.0617	0.0358	0.0286	0.0377	0.0858	0.0388	0.0439	0.0391
	(0.4,0.2,0.9,0.8)	**0.4370**	**0.3812**	**0.3527**	**0.3826**	**0.3069**	**0.3342**	**0.3120**	**0.3803**
		0.2360	*0.1875*	*0.0057*	*0.1257*	*0.1806*	*0.1723*	*0.0060*	*0.1186*
		0.0571	0.0355	0.0023	0.0161	0.0413	0.0340	0.0078	0.0144
	(0.45,0.2,0.8,0.6)	**0.3405**	**0.4202**	**0.2873**	**0.3182**	**0.2171**	**0.3718**	**0.2492**	**0.3200**
		0.2336	*0.1739*	*0.0067*	*0.1318*	*0.1946*	*0.1721*	*0.0058*	*0.1350*
		0.0666	0.0311	0.0265	0.0347	0.0921	0.0357	0.0403	0.0351
	(0.45,0.2,0.8,0.7)	**0.5648**	**0.4435**	**0.4728**	**0.5064**	**0.4402**	**0.4030**	**0.4448**	**0.5147**
		0.2345	*0.2071*	*0.0062*	*0.1225*	*0.1866*	*0.2208*	*0.0064*	*0.1256*
		0.0681	0.0429	0.0006	0.0182	0.0349	0.0510	0.0001	0.0200
500	(0.4,0.2,0.9,0.7)	**0.3836**	**0.3951**	**0.3263**	**0.3581**	**0.2892**	**0.3503**	**0.2894**	**0.3526**
		0.2023	*0.1628*	*0.0057*	*0.1094*	*0.1614*	*0.1602*	*0.0057*	*0.1126*
		0.0412	0.0265	0.0055	0.0137	0.0383	0.0281	0.0123	0.0149
	(0.4,0.2,0.9,0.8)	**0.3169**	**0.3742**	**0.2716**	**0.2972**	**0.2306**	**0.3403**	**0.2364**	**0.2977**
		0.1993	*0.1555*	*0.0051*	*0.1103*	*0.1550*	*0.1388*	*0.0052*	*0.1120*
		0.0467	0.0248	0.0165	0.0227	0.0527	0.0228	0.0268	0.0230
	(0.45,0.2,0.8,0.6)	**0.3591**	**0.4369**	**0.3134**	**0.3395**	**0.2566**	**0.3936**	**0.2719**	**0.3366**
		0.2071	*0.1754*	*0.0060*	*0.1242*	*0.1741*	*0.1723*	*0.0056*	*0.1217*
		0.0511	0.0309	0.0187	0.0277	0.0677	0.0329	0.0317	0.0277
	(0.45,0.2,0.8,0.7)	**0.4415**	**0.4389**	**0.3914**	**0.4170**	**0.3509**	**0.4033**	**0.3633**	**0.4256**
		0.2035	*0.1598*	*0.0059*	*0.1151*	*0.1557*	*0.1463*	*0.0051*	*0.1159*
		0.0415	0.0257	0.0035	0.0143	0.0341	0.0236	0.0075	0.0140

Table 5.12: *GPH estimates of the memory parameter d_1 for two-factor Gegenbauer process $\prod_{j=1}^{2}(1 - 2\eta_j B + B^2)^{d_j} X_t = \varepsilon_t$, for different values of $d_1 = 0.4, 0.45$, $d_2 = 0.2, 0.3$, $\eta_1 = 0.8, 0.9$ and $\eta_2 = 0.6, 0.7, 0.8$, different Monte Carlo procedures (MC1, MC2), B1 and B2 bootstrap estimates, where n is the sample size,* 300 *bootstrap replications,* 1000 *repetitions of the experiment: average value (boldface), standard error (italic), mean squared error (normal font).*

		$d_1 = 0.4, 0.45$, local Whittle estimates							
n	(d, η)	MC1	MC2	B1	B2	MC1	MC2	B1	B2
300	(0.4,0.2,0.9,0.7)	**0.2992**	**0.3875**	**0.2157**	**0.2592**	**0.1602**	**0.3336**	**0.1734**	**0.2580**
		0.2147	*0.1787*	*0.0065*	*0.1414*	*0.1777*	*0.1720*	*0.0064*	*0.1454*
		0.0563	0.0321	0.0340	0.0398	0.0891	0.0340	0.0514	0.0413
	(0.4,0.2,0.9,0.8)	**0.4352**	**0.3965**	**0.3350**	**0.3852**	**0.2944**	**0.3483**	**0.2916**	**0.3827**
		0.2332	*0.1710*	*0.0062*	*0.1438*	*0.1733*	*0.1566*	*0.0064*	*0.1376*
		0.0556	0.0292	0.0043	0.0209	0.0412	0.0272	0.0118	0.0192
	(0.45,0.2,0.8,0.6)	**0.3551**	**0.4189**	**0.2587**	**0.3157**	**0.1979**	**0.3781**	**0.2202**	**0.3162**
		0.2165	*0.1653*	*0.0065*	*0.1410*	*0.1777*	*0.1581*	*0.0057*	*0.1439*
		0.0559	0.0283	0.0367	0.0379	0.0951	0.0302	0.0528	0.0386
	(0.45,0.2,0.8,0.7)	**0.5712**	**0.4756**	**0.4828**	**0.5280**	**0.4647**	**0.4398**	**0.4605**	**0.5369**
		0.2260	*0.2201*	*0.0063*	*0.1370*	*0.1771*	*0.2237*	*0.0066*	*0.1405*
		0.0658	0.0491	0.0011	0.0249	0.0316	0.0501	0.0002	0.0273
500	(0.4,0.2,0.9,0.7)	**0.3933**	**0.4092**	**0.3236**	**0.3728**	**0.2981**	**0.3639**	**0.2869**	**0.3681**
		0.1860	*0.1453*	*0.0055*	*0.1160*	*0.1461*	*0.1422*	*0.0057*	*0.1203*
		0.0346	0.0212	0.0059	0.0142	0.0317	0.0215	0.0128	0.0155
	(0.4,0.2,0.9,0.8)	**0.3086**	**0.3820**	**0.2378**	**0.2869**	**0.2059**	**0.3441**	**0.2042**	**0.2899**
		0.1808	*0.1397*	*0.0054*	*0.1206*	*0.1500*	*0.1298*	*0.0056*	*0.1235*
		0.0410	0.0198	0.0263	0.0273	0.0602	0.0200	0.0384	0.0274
	(0.45,0.2,0.8,0.6)	**0.3600**	**0.4399**	**0.2934**	**0.3353**	**0.2467**	**0.3977**	**0.2533**	**0.3340**
		0.1839	*0.1651*	*0.0059*	*0.1305*	*0.1566*	*0.1593*	*0.0054*	*0.1257*
		0.0419	0.0273	0.0246	0.0302	0.0658	0.0281	0.0387	0.0293
	(0.45,0.2,0.8,0.7)	**0.4240**	**0.4372**	**0.3571**	**0.4000**	**0.3319**	**0.4092**	**0.3331**	**0.4129**
		0.1871	*0.1425*	*0.0057*	*0.1242*	*0.1458*	*0.1282*	*0.0053*	*0.1255*
		0.0357	0.0205	0.0087	0.0179	0.0352	0.0181	0.0137	0.0171

Table 5.13: *Local Whittle estimates of the memory parameter d_1 for two-factor Gegenbauer process $\prod_{j=1}^{2}(1-2\eta_j B+B^2)^{d_j}X_t=\varepsilon_t$, for different values of $d_1=0.4, 0.45$, $d_2=0.2, 0.3$, $\eta_1=0.8, 0.9$ and $\eta_2=0.6, 0.7, 0.8$, different Monte Carlo procedures (MC1, MC2), B1 and B2 bootstrap estimates, where n is the sample size,* 300 *bootstrap replications,* 1000 *repetitions of the experiment: average value (boldface), standard error (italic), mean squared error (normal font).*

		$d_2 = 0.2, 0.3$, GPH estimates							
		smoothed periodogram				periodogram			
n	(d, η)	MC1	MC2	B1	B2	MC1	MC2	B1	B2
300	(0.4,0.2,0.9,0.7)	**0.3311**	**0.1998**	**0.2881**	**0.3066**	**0.2667**	**0.1717**	**0.2490**	**0.3053**
		0.2275	*0.1639*	*0.0060*	*0.1193*	*0.1874*	*0.1666*	*0.0060*	*0.1242*
		0.0690	0.0269	0.0078	0.0256	0.0396	0.0285	0.0024	0.0265
	(0.4,0.2,0.9,0.8)	**0.3800**	**0.1904**	**0.3296**	**0.3493**	**0.3054**	**0.1638**	**0.2944**	**0.3505**
		0.2225	*0.1997*	*0.0060*	*0.1170*	*0.1802*	*0.2039*	*0.0060*	*0.1183*
		0.0819	0.0400	0.0168	0.0360	0.0436	0.0429	0.0090	0.0367
	(0.45,0.2,0.8,0.6)	**0.3439**	**0.1842**	**0.2992**	**0.3184**	**0.2901**	**0.1752**	**0.2729**	**0.3265**
		0.2345	*0.1732*	*0.0064*	*0.1229*	*0.1832*	*0.1703*	*0.0064*	*0.1203*
		0.0757	0.0302	0.0099	0.0291	0.0417	0.0296	0.0054	0.0305
	(0.45,0.2,0.8,0.7)	**0.4242**	**0.1890**	**0.3683**	**0.3880**	**0.3563**	**0.1705**	**0.3381**	**0.3931**
		0.2350	*0.2227*	*0.0063*	*0.1125*	*0.1852*	*0.2288*	*0.0063*	*0.1158*
		0.1055	0.0497	0.0284	0.0480	0.0587	0.0532	0.0191	0.0507
500	(0.4,0.2,0.9,0.7)	**0.2983**	**0.1764**	**0.2647**	**0.2791**	**0.2332**	**0.1513**	**0.2304**	**0.2809**
		0.2042	*0.1603*	*0.0053*	*0.1073*	*0.1619*	*0.1543*	*0.0055*	*0.1070*
		0.0514	0.0263	0.0042	0.0178	0.0273	0.0262	0.0010	0.0180
	(0.4,0.2,0.9,0.8)	**0.3703**	**0.1950**	**0.3125**	**0.3298**	**0.2910**	**0.1722**	**0.2727**	**0.3258**
		0.2119	*0.1614*	*0.0057*	*0.1132*	*0.1569*	*0.1531*	*0.0052*	*0.1089*
		0.0739	0.0261	0.0127	0.0297	0.0329	0.0242	0.0053	0.0277
	(0.45,0.2,0.8,0.6)	**0.3280**	**0.1866**	**0.2851**	**0.3019**	**0.2710**	**0.1718**	**0.2582**	**0.3084**
		0.2170	*0.1503*	*0.0055*	*0.1124*	*0.1582*	*0.1401*	*0.0053*	*0.1109*
		0.0635	0.0228	0.0073	0.0230	0.0301	0.0204	0.0034	0.0241
	(0.45,0.2,0.8,0.7)	**0.4015**	**0.1997**	**0.3502**	**0.3692**	**0.3342**	**0.1741**	**0.3157**	**0.3655**
		0.1979	*0.1654*	*0.0055*	*0.1061*	*0.1603*	*0.1639*	*0.0054*	*0.1067*
		0.0798	0.0273	0.0226	0.0399	0.0437	0.0275	0.0134	0.0388

Table 5.14: *GPH estimates of the memory parameter d_2 for two-factor Gegenbauer process $\prod_{j=1}^{2}(1 - 2\eta_j B + B^2)^{d_j} X_t = \varepsilon_t$, for different values of $d_1 = 0.4, 0.45$, $d_2 = 0.2, 0.3$, $\eta_1 = 0.8, 0.9$ and $\eta_2 = 0.6, 0.7, 0.8$, different Monte Carlo procedures (MC1, MC2), B1 and B2 bootstrap estimates, where n is the sample size,* 300 *bootstrap replications,* 1000 *repetitions of the experiment: average value (boldface), standard error (italic), mean squared error (normal font).*

		$d_2 = 0.2, 0.3$, local Whittle estimates							
		smoothed periodogram				periodogram			
n	(d, η)	MC1	MC2	B1	B2	MC1	MC2	B1	B2
300	(0.4,0.2,0.9,0.7)	**0.3334**	**0.1853**	**0.2828**	**0.3139**	**0.2741**	**0.1651**	**0.2495**	**0.3129**
		0.2228	*0.1380*	*0.0060*	*0.1306*	*0.1738*	*0.1437*	*0.0061*	*0.1358*
		0.0674	0.0193	0.0069	0.0300	0.0357	0.0219	0.0025	0.0312
	(0.4,0.2,0.9,0.8)	**0.3855**	**0.1773**	**0.3327**	**0.3662**	**0.3232**	**0.1586**	**0.3001**	**0.3655**
		0.2204	*0.1837*	*0.0063*	*0.1322*	*0.1712*	*0.1825*	*0.0061*	*0.1321*
		0.0830	0.0343	0.0177	0.0451	0.0445	0.0350	0.0101	0.0448
	(0.45,0.2,0.8,0.6)	**0.3404**	**0.1759**	**0.2935**	**0.3245**	**0.3004**	**0.1705**	**0.2759**	**0.3352**
		0.2273	*0.1514*	*0.0064*	*0.1329*	*0.1689*	*0.1452*	*0.0067*	*0.1304*
		0.0714	0.0235	0.0088	0.0332	0.0386	0.0219	0.0058	0.0353
	(0.45,0.2,0.8,0.7)	**0.4316**	**0.1639**	**0.3709**	**0.4037**	**0.3712**	**0.1459**	**0.3472**	**0.4076**
		0.2217	*0.2235*	*0.0061*	*0.1244*	*0.1728*	*0.2241*	*0.0066*	*0.1293*
		0.1028	0.0513	0.0292	0.0570	0.0592	0.0531	0.0217	0.0598
500	(0.4,0.2,0.9,0.7)	**0.2890**	**0.1629**	**0.2521**	**0.2793**	**0.2417**	**0.1458**	**0.2229**	**0.2821**
		0.1860	*0.1386*	*0.0057*	*0.1165*	*0.1440*	*0.1331*	*0.0055*	*0.1149*
		0.0425	0.0206	0.0027	0.0199	0.0225	0.0206	0.0006	0.0199
	(0.4,0.2,0.9,0.8)	**0.3656**	**0.1885**	**0.3075**	**0.3398**	**0.2998**	**0.1658**	**0.2718**	**0.3351**
		0.1990	*0.1415*	*0.0057*	*0.1237*	*0.1473*	*0.1379*	*0.0054*	*0.1224*
		0.0670	0.0201	0.0116	0.0348	0.0317	0.0202	0.0052	0.0332
	(0.45,0.2,0.8,0.6)	**0.3200**	**0.1731**	**0.2730**	**0.3019**	**0.2770**	**0.1659**	**0.2535**	**0.3100**
		0.1998	*0.1301*	*0.0055*	*0.1199*	*0.1486*	*0.1240*	*0.0054*	*0.1192*
		0.0543	0.0176	0.0054	0.0248	0.0280	0.0165	0.0029	0.0263
	(0.45,0.2,0.8,0.7)	**0.3946**	**0.1952**	**0.3463**	**0.3778**	**0.3397**	**0.1673**	**0.3156**	**0.3723**
		0.1858	*0.1434*	*0.0056*	*0.1159*	*0.1457*	*0.1485*	*0.0055*	*0.1180*
		0.0724	0.0206	0.0214	0.0451	0.0408	0.0231	0.0134	0.0436

Table 5.15: *Local Whittle estimates of the memory parameter d_2 for two-factor Gegenbauer process $\prod_{j=1}^{2}(1-2\eta_j B+B^2)^{d_j}X_t=\varepsilon_t$, for different values of $d_1 = 0.4, 0.45$, $d_2 = 0.2, 0.3$, $\eta_1 = 0.8, 0.9$ and $\eta_2 = 0.6, 0.7, 0.8$, different Monte Carlo procedures (MC1, MC2), B1 and B2 bootstrap estimates, where n is the sample size,* 300 *bootstrap replications,* 1000 *repetitions of the experiment: average value (boldface), standard error (italic), mean squared error (normal font).*

η	MC1	gph	lw	B1	B2
0.6	**0.6016**	**0.6005**	**0.6005**	**0.6011**	**0.6013**
	0.0117	*0.0118*	*0.0118*	*0.0004*	*0.0102*
0.7	**0.7007**	**0.7000**	**0.6999**	**0.7009**	**0.7006**
	0.0118	*0.0118*	*0.0118*	*0.0004*	*0.0101*
0.7	**0.7029**	**0.7010**	**0.7011**	**0.7021**	**0.7024**
	0.0116	*0.0116*	*0.0117*	*0.0004*	*0.0100*
0.8	**0.8021**	**0.8009**	**0.8008**	**0.8019**	**0.8021**
	0.0111	*0.0111*	*0.0110*	*0.0004*	*0.0093*
0.6	**0.6010**	**0.6006**	**0.6006**	**0.6008**	**0.6009**
	0.0101	*0.0102*	*0.0102*	*0.0003*	*0.0105*
0.7	**0.7007**	**0.7001**	**0.7001**	**0.7006**	**0.7007**
	0.0095	*0.0094*	*0.0094*	*0.0003*	*0.0097*
0.7	**0.7019**	**0.7001**	**0.7001**	**0.7015**	**0.7019**
	0.0099	*0.0099*	*0.0099*	*0.0003*	*0.0100*
0.8	**0.8023**	**0.8010**	**0.8009**	**0.8017**	**0.8024**
	0.0094	*0.0095*	*0.0095*	*0.0003*	*0.0096*

η	MC1	gph	lw	B1	B2
0.6	**0.6020**	**0.6007**	**0.6007**	**0.6014**	**0.6015**
	0.0137	*0.0138*	*0.0138*	*0.0005*	*0.0117*
0.7	**0.7003**	**0.6996**	**0.6997**	**0.7007**	**0.7000**
	0.0135	*0.0135*	*0.0135*	*0.0004*	*0.0119*
0.7	**0.7027**	**0.6997**	**0.6996**	**0.7024**	**0.7025**
	0.0136	*0.0136*	*0.0135*	*0.0004*	*0.0117*
0.8	**0.8029**	**0.8014**	**0.8011**	**0.8021**	**0.8027**
	0.0126	*0.0129*	*0.0129*	*0.0004*	*0.0110*
0.6	**0.6009**	**0.5999**	**0.6000**	**0.6008**	**0.6007**
	0.0114	*0.0112*	*0.0112*	*0.0004*	*0.0117*
0.7	**0.7008**	**0.7001**	**0.7001**	**0.7008**	**0.7009**
	0.0112	*0.0113*	*0.0113*	*0.0003*	*0.0116*
0.7	**0.7028**	**0.7011**	**0.7009**	**0.7024**	**0.7037**
	0.0112	*0.0113*	*0.0113*	*0.0004*	*0.0116*
0.8	**0.8018**	**0.8004**	**0.8003**	**0.8017**	**0.8021**
	0.0103	*0.0103*	*0.0103*	*0.0003*	*0.0108*

Table 5.16: *Estimates of the Gegenbauer frequency η_1 for two-factor Gegenbauer process $\prod_{j=1}^{2}(1-2\eta_j B+B^2)^{d_j}X_t=\varepsilon_t$, for different values of $d_1=0.4, 0.45$, $d_2=0.2, 0.3$, $\eta_1=0.8, 0.9$ and $\eta_2=0.6, 0.7, 0.8$, different Monte Carlo procedures (MC1, MC2-gph, MC2-lw), B1 and B2 bootstrap estimates, where n is the sample size,* 300 *bootstrap replications,* 1000 *repetitions of the experiment, average value (boldface), standard error (italic): in the top part of the table the Gegenbauer frequency is estimated from the raw periodogram, whereas in the bottom part the Gegenbauer frequency is estimated after smoothing periodogram.*

η	MC1	gph	lw	B1	B2
0.8	**0.7997**	**0.7998**	**0.7998**	**0.7999**	**0.7999**
	0.0068	*0.0068*	*0.0068*	*0.0003*	*0.0050*
0.9	**0.9000**	**0.9001**	**0.9001**	**0.8999**	**0.9001**
	0.0061	*0.0060*	*0.0060*	*0.0002*	*0.0045*
0.8	**0.7996**	**0.7999**	**0.7997**	**0.7997**	**0.7996**
	0.0068	*0.0068*	*0.0068*	*0.0003*	*0.0051*
0.9	**0.8999**	**0.9001**	**0.9001**	**0.8998**	**0.8999**
	0.0065	*0.0065*	*0.0064*	*0.0003*	*0.0048*
0.8	**0.8000**	**0.8000**	**0.8000**	**0.7999**	**0.7999**
	0.0046	*0.0046*	*0.0046*	*0.0002*	*0.0046*
0.9	**0.9001**	**0.9001**	**0.9002**	**0.9001**	**0.9001**
	0.0046	*0.0046*	*0.0046*	*0.0002*	*0.0045*
0.8	**0.8000**	**0.8001**	**0.8001**	**0.7999**	**0.7999**
	0.0046	*0.0046*	*0.0046*	*0.0002*	*0.0046*
0.9	**0.9001**	**0.9002**	**0.9002**	**0.9000**	**0.9001**
	0.0046	*0.0046*	*0.0046*	*0.0002*	*0.0047*

η	MC1	gph	lw	B1	B2
0.8	**0.8004**	**0.8005**	**0.8005**	**0.7999**	**0.7999**
	0.0081	*0.0081*	*0.0081*	*0.0003*	*0.0061*
0.9	**0.9001**	**0.9002**	**0.9002**	**0.9001**	**0.8999**
	0.0081	*0.0081*	*0.0080*	*0.0003*	*0.0064*
0.8	**0.8000**	**0.8005**	**0.8004**	**0.7999**	**0.7999**
	0.0082	*0.0082*	*0.0082*	*0.0003*	*0.0065*
0.9	**0.8995**	**0.9000**	**0.8999**	**0.8997**	**0.8998**
	0.0077	*0.0076*	*0.0076*	*0.0003*	*0.0057*
0.8	**0.7997**	**0.7999**	**0.7999**	**0.7997**	**0.7997**
	0.0064	*0.0063*	*0.0063*	*0.0003*	*0.0063*
0.9	**0.8999**	**0.9001**	**0.9001**	**0.9002**	**0.8999**
	0.0061	*0.0061*	*0.0061*	*0.0003*	*0.0062*
0.8	**0.7996**	**0.8000**	**0.7999**	**0.7997**	**0.7995**
	0.0064	*0.0064*	*0.0064*	*0.0003*	*0.0065*
0.9	**0.8999**	**0.9001**	**0.9001**	**0.9001**	**0.8999**
	0.0059	*0.0060*	*0.0060*	*0.0002*	*0.0060*

Table 5.17: *Estimates of the Gegenbauer frequency η_2 for two-factor Gegenbauer process $\prod_{j=1}^{2}(1-2\eta_j B+B^2)^{d_j}X_t=\varepsilon_t$, for different values of $d_1=0.4, 0.45$, $d_2=0.2, 0.3$, $\eta_1=0.8, 0.9$ and $\eta_2=0.6, 0.7, 0.8$, different Monte Carlo procedures (MC1, MC2-gph, MC2-lw), B1 and B2 bootstrap estimates, where n is the sample size,* 300 *bootstrap replications,* 1000 *repetitions of the experiment, average value (boldface), standard error (italic): in the top part of the table the Gegenbauer frequency is estimated from the raw periodogram, whereas in the bottom part the Gegenbauer frequency is estimated after smoothing periodogram.*

Appendix A

Cox test for two different linear models

In this section we shall give a brief introduction to the Cox test for two different linear models (for more details, refer to Plasmans, 2006). This test can be used to compare two non-nested linear models

$$\begin{cases} Y = \alpha_1 X_1 + \varepsilon_1 \\ Y = \alpha_2 X_2 + \varepsilon_2, \end{cases} \tag{A.1}$$

where ε_1 and ε_2 are Gaussian errors. The test statistic is

$$\hat{s}_0 = \frac{n}{2} \log \frac{Y' M_1 Y}{Y' M_0 Y + \hat{\alpha}_1' X_1' M_1 X_1 \hat{\alpha}_1}$$

where $\hat{\alpha}_1$ is the maximum likelihood estimator of α_1, n is the sample size and

$$M_0 = I_n - X_1 (X_1' X_1)^{-1} X_1'$$
$$M_1 = I_n - X_2 (X_2' X_2)^{-1} X_2'$$

with I_n the identity matrix of order n. Under the normality assumption the test is normally distributed with asymptotic variance given by Pesaran (1974)

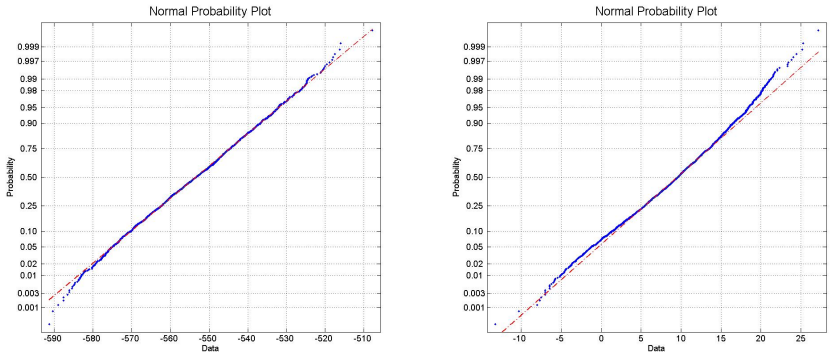
$$\hat{\sigma}^2(\hat{s}_0) = \frac{\hat{\sigma}_0^2}{\hat{\tilde{\sigma}}_{10}^4} \hat{\alpha}_1' X_1' M_1' M_0 M_1 X_1 \hat{\alpha}_1$$

where $\hat{\sigma}_0^2 = Y' M_0 Y / n$ and $\hat{\tilde{\sigma}}_{10}^2 = \hat{\sigma}_0^2 + \hat{\alpha}_1' X_1' M_1 X_1 \hat{\alpha}_1 / n$. If $\hat{s}_0$ is significantly larger than zero (smaller than zero), the first model in (A.1) outperforms the second (the second outperforms the first). If $\hat{s}_0$ is not significantly different from zero the two models have the same goodness of fit.

The Cox test is valid under the hypotheses of normality and independence of the errors. In our case neither of these assumptions is valid. We performed a bootstrap investigation on the Cox test to check how far from normality is its distribution if we relax normality

and independence hypotheses.

For $d = 0.35$ and for R_1 (autocovariance at lag 1) we estimate the distribution of the Cox tests $t_{R_1,1}$ and $t_{R_1,2}$ through a bootstrap resampling of $[Y_i, X_{1,i}, X_{2,i}, X_{3,i}]$. Both distributions appear normal from the Q-Q plots of Figure (A.1), thus we can consider valid the asymptotic normality of the test.



(a) (b)

Figure A.1: *Q-Q plots of the bootstrap distribution of the Cox test for* $d = 0.35$ *and lag* $= 1$*: (a) test between the first and the second linear models given in equation (4.7), (b) test between the second and the third linear models given in equation (4.7).*

Bibliography

Abramowitz, M. and I. Stegun (Eds.) (1972). *Handbook Of Mathematical Functions With Formulas, Graphs, And Mathematical Tables*. New York: Dover.

Agiakloglou, C., P. Newbold, and M. Wohar (1993). Bias in an estimator of the fractional difference parameter. *Journal of Time Series Analysis 14*(3), 235–246.

Akaike, H. (1974). A new look to the statistical model identification. *IEEE Trans. Autom. Control 19*, 716–723.

Albin, J. (1998a). A note on rosenblatt distributions. *Statistics & Probability Letters 40*(1), 83–91.

Albin, J. (1998b). On extremal theory for self-similarity processes. *The Annals of Probability 26*(2), 743–793.

Anderson, T. and A. Walker (1964). On the asymptotic distribution of the autocorrelations of a sample from a linear stochastic process. *Annals of Mathematical Statistics 35*(3), 1296–1303.

Andrews, D., O. Lieberman, and V. Marmer (2006). Higher-order improvements of the parametric bootstrap for long-memory Gaussian processes. *Journal of Econometrics 133*, 633–702.

Arteche, J. and J. Orbe (2005). Bootstrapping the log-periodogram regression. *Economics Letters 86*, 79–85.

Arteche, J. and P. Robinson (2000). Semiparametric inference in seasonal and cyclical long memory processes. *Journal of Time Series Analysis 21*(1), 1–25.

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

Barndorff-Nielsen, O. and D. Cox (1989). *Asymptotic techniques for use in statistics*. Chapman & Hall.

Barnett, A. (2002). *On the use of bispectrum to detect and model non-linearity*. Ph. D. thesis, School of Physical Science, The University of Queensland.

Bartlett, M. S. (1946). On the theoretical specification and sampling properties of autocorrelated time-series. *Supplement of the Journal of the Royal Statistical Society 8*(1), 27–41.

Beran, J. (1994). *Statistics for Long-Memory Processes.* Chapmann & Hall.

Bhattacharya, R., V. Gupta, and W. Waymire (1983). The Hurst effect under trends. *Journal of Applied Probability 20*, 649–662.

Bickel, P. and P. Bühlmann (1999). A new mixing notion and functional central limit theorems for a SIEVE bootstrap in time series. *Bernoulli 5*(3), 413–446.

Bickel, P. and D. Freedman (1981). Some asymptotic theory for the bootstrap. *The Annals of Statistics 9*(6), 1196–1217.

Bisaglia, L. and D. Guègan (1998). A comparison of techniques of estimation in long-memory process. *Computational statistics and data analysis 27*, 61–81.

Bose, A. (1988). Edgeworth correction by bootstrap in autoregressions. *The Annals of Statistics 16*(4), 1709–1722.

Box, G. and G. Jenkins (1976). *Time Series Analysis Forecasting and Control.* 2nd edition. San Francisco: Holden-Day.

Braun, W. and R. Kulperger (1997). Properties of a Fourier bootstrap method for time series. *Communications in Statistics Theory and Method 26*(6), 1329–1336.

Brockwell, P. and R. Davis (1991). *Time series: theory and methods.* Springer-Verlag.

Bühlmann, P. (1997). Sieve bootstrap for time series. *Bernoulli 3*, 123–148.

Bühlmann, P. (1998). Sieve bootstrap for smoothing in nonstationary time series. *The Annals of Statistics 26*(1), 48–82.

Bühlmann, P. (2002). Bootstrap for time series. *Statistical science 17*(1), 52–72.

Burlaga, L. and L. Klein (1986). Fractal structure of the interplanetary magnetic field. *Journal of Geophysical Research 91*, 347–350.

Caporale, G. and L. Gil-Alana (2006). Long memory at the long-run and the seasonal monthly frequencies in the US money stock. *Applied Economics Letters 13*, 965–968.

Carlstein, E. (1986). The use of subseries values for estimating the variance of a general statistic from a stationary sequence. *The Annals of Statistics 14*(3), 1171–1179.

Cavazos-Cadena, R. (1994). The asymptotic distribution of sample autocorrelations for a class of linear filters. *Journal of Multivariate Analysis 48*, 249–274.

Chatfield, C. (1996). *The analysis of time series: an introduction.* Chapman & Hall.

Chung, C.-F. (1996). Estimating a generalized long memory process. *Journal of Econometrics 73*(1), 237–259.

Crilly, A., R. Earnshaw, and H. Jones (Eds.) (1991). *Fractals and Chaos*. Springer-Verlag.

Dahlhaus, R. (1988). Small sample effects in time series analysis: a new asymptotic theory and a new estimate. *The Annals of Statistics 16*(2), 808–841.

Dahlhaus, R. (1989). Efficient parameter estimation for self-similar processes. *Annals of Statistics 17*, 1749–1766.

Dahlhaus, R. and D. Janas (1996). A frequency domain bootstrap for ratio statistics in time series analysis. *The Annals of Statistics 24*(5), 1934–1963.

Datta, S. and T. Sriram (1997). A modified bootstrap for autoregression without stationarity. *Journal of Statistical Planning and Inference 59*, 19–30.

Davison, A. and D. Hinkley (1997). *Bootstrap methods and their application*. Cambridge university press.

Devaney, R. (1999). The mandelbrot set, the farey tree, and the fibonacci sequence. *The American Mathematical Monthly 106*(4), 289–302.

Efron, B. (1979). Bootstrap methods: another look at the jackknife. *The Annals of Statistics 7*, 1–26.

Efron, B. (1982). *The jackknife, the bootstrap and other resampling plans*. CBMS-NSF Regional conference series in applied mathematics.

Efron, B. (1987a). Better bootstrap confidence intervals. *Journal of the American Statistical Association 82*(397), 171–185.

Efron, B. (1987b). Better bootstrap confidence intervals: rejoinder. *Journal of the American Statistical Association 82*(397), 198–200.

Efron, B. (1987c). Empirical confidence intervals based on bootstrap samples: comment. *Journal of the American Statistical Association 82*(399), 754.

Efron, B. and R. J. Tibshirani (1993). *An Introduction to the Bootstrap*. Chapman & Hall.

Fox, R. and M. Taqqu (1986). Large-sample properties of parameter estimates for strongly dependent stationary gaussian time series. *The Annals of Statistics 14*(2), 517–532.

Franco, G. and V. Reisen (2007). Bootstrap approaches and confidence intervals for stationary and non-stationary long-range dependence. *Physica A 375*(2), 546–562.

Franke, J. and W. Härdle (1992). On bootstrapping kernel spectral estimates. *The Annals of Statistics 20*(1), 121–145.

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

Giraitis, L., P. Kokoszka, and R. Leipus (2001). Testing for long memory in the presence of a general trend. *Journal of Applied Probability 38*(4), 1033–1054.

Giraitis, L. and R. Leipus (1995). A generalized fractionally differencing approach in long-memory modeling. *Lithuanian Mathematical Journal 35*(1), 53–65.

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

Giraitis, L. and D. Surgailis (1990). A central limit theorem for quadratic forms in strongly dependent linear variables and its application to asymptotical normality of Whittle's estimate. *Probability Theory and Related Fields 86*, 87–104.

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

Gray, H., N. Zhang, and W. Woodward (1994). Correction on 'On generalized fractional processes'. *Journal of Time Series Analysis 15*, 561–562.

Hall, P. (1988). Theoretical comparison of bootstrap confidence interval. *The Annals of Statistics 16*(3), 927–953.

Hall, P. (1992a). *The Bootstrap and Edgeworth Expansion*. Springer-Verlag.

Hall, P. (1992b). Effect of bias estimation on coverage accuracy of bootstrap confidence intervals for a probability density. *The Annals of Statistics 20*(2), 675–694.

Hall, P. and C. Heyde (1980). *Martingale limit theory and its application*. New York: Academic Press.

Hannan, E. (1973). The asymptotic theory of linear time series models. *Journal of Applied Probability 10*, 130–145.

He, S. (1996). A note on the asymptotic normality of sample autocorrelations for a linear stationarity sequence. *Journal of Multivariate Analysis 58*, 182–188.

Higuchi, T. (1988). Approach to an irregular time series on the basis of the fractal theory. *Physica D 31*, 277–283.

Hinkley, D. (1988). Bootstrap methods (with discussion). *Journal of the Royal Statistical Society, B 50*, 321–337.

Hosking, J. (1981). Fractional differencing. *Biometrika 68*(1), 165–176.

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

Hurst, H. (1951). Long-term storage capacity of reservoirs. *Transactions of the American Society of Civil Engineers 116*, 770–799.

Hurvich, C., R. Deo, and J. Brodsky (1998). The mena 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*(1), 19–46.

Kapetanios, G. and Z. Psaradakis (2006). Sieve bootstrap for strongly dependent stationary processes. *Working Paper No. 552, Queen Mary, University of London*.

Kreiss, J. (1992). *Bootstrap procedures for AR(∞)-process*. Springer: Heidelberg.

Kreiss, J.-P. and E. Paparoditis (2003). Autoregressive-aided periodogram bootstrap for time series. *Institute of Mathematical Statistics 31*(6), 1923–1955.

Künsch, H. (1989). The jackknife and the bootstrap for general stationary observations. *The Annals of Statistics 17*(3), 1217–1241.

Künsch, H. R. (1987). Statistical aspects of self-similar processes. *Proceedings of the First World Congress of the Bernoulli Society 1*, 67–74.

Lahiri, S. (1992). *Resampling Methods for Dependent Data*. Springer.

Leland, W., M. Taqqu, W. Willinger, and D. Wilson (1994). On the self-similar nature of Ethernet traffic (extended version). *IEEE/ACM Trans. Network 2*(1), 1–15.

Leonenko, N. and V. Anh (2001). Rate of convergence to the Rosenblatt distribution fro additive functionals of stochastic processes with long-range dependence. *Journal of Applied Mathematics and Stochastic Analysis 14*(1), 27–46.

Li, H. and G. Maddala (1996). Bootstrapping time series models. *Econometric reviews 15*(2), 115–158.

Mandelbrot, B. (1975). Limit theorems of the self-normalized range for weakly and strongly dependent processes. *Z. Wahr. verw. Geb. 31*, 271–285.

Mandelbrot, B. and J. van Ness (1968). Fractional brownian motion, fractional noise and application. *SIAM Review 10*, 422–437.

Newbold, P. and C. Agiakloglou (1993). Bias in the sample autocorrelations of fractional noise. *Biometrika 80*(3), 698–702.

Nur, D., R. Wolff, and K. Mergensen (2001). Phase randomisation: numerical study of higher order cumulants behaviour. *Computational Statistics & Data Analysis 37*, 487–513.

Palma, W. and N. Chan (2005). Efficient estimation of seasonal long-range-dependent processes. *Journal of Time Series Analysis 26*(6), 863–892.

Paparoditis, E. and D. Politis (1999). The local bootstrap for periodogram statistics. *Journal of Time Series Analysis 20*(2), 193–222.

Paparoditis, E. and D. Politis (2002). The local bootstrap for Markov processes. *Journal of Statistical Planning and Inference 108*, 301–328.

Pesaran, M. (1974). On the general problem of model selection. *Review of Economic Studies 41*, 153–171.

Plasmans, J. (2006). *Modern linear and nonlinear econometrics*. Springer.

Politis, D. (2003). The impact of bootstrap methods on time series analysis. *Statistical science 18*(2), 219–230.

Politis, D. and J. Romano (1994). The stationary bootstrap. *JASA 89*(428), 1303–1313.

Poskitt, D. (2007). Properties of the Sieve bootstrap for fractionally integrated and non-invertible processes. *Journal of Time Series Analysis Accepted*.

Priestley, M. (1988). *Spectral Analysis and Time Series*. Harcourt Brace & Company, Publisher.

Ramsey, F. (1974). Characterization of the partial autocorrelation function. *Annals of Statistics 2*, 1296–1301.

Reisen, V., A. Rodrigues, and W. Palma (2006). Estimating seasonal long-memory processes: a Monte Carlo study. *Journal of Statistical Computation and Simulation 76*(4), 305–316.

Revuz, D. and M. Yor (1994). *Continuous Martingales and Brownian Motion*. Springer-Verlag, Second Edition.

Robinson, P. (1995a). Gaussian semiparametric estimation of long range dependence. *The Annals of Statistics 23*(5), 1630–1661.

Robinson, P. (1995b). Log-periodogram regression of time series with long range dependence. *The Annals of Statistics 23*(3), 1048–1072.

Rogers, L. and D. Williams (1979). *Diffusions, Markov processes, and martingales. Volume One: Foundations*. John Wiley & Sons.

Sadek, N. and A. Khotanzad (2004). K-Factor Gegenbauer ARMA process for network traffic simulation. *Proceedings of the Ninth International Symposium on Computers and Communications 2004 2*, 963–968.

Samorodnitsky, G. and M. Taqqu (1994). *Stable non-Gaussian random processes*. Chapman & Hall.

Shao, J. and D. Tu (1995). *The Jackknife and Bootstrap*. Springer.

Silva, E., G. Franco, V. Reisen, and F. Cruz (2006). Local bootstrap approaches for fractional differential parameter estimation in ARFIMA models. *Computational statistics & data analysis 51*(2), 1002–1011.

Smallwood, A. and P. Beaumont (2004). Multiple frequency long memory. Discussion paper, Department of Economics, University of Oklahoma.

Sowell, F. (1992). Maximum likelihood estimation of stationary univariate fractionally integrated time series models. *Journal of Econometrics 53*, 165–188.

Swanepoel, J. and J. van Wyk (1986). The bootstrap applied to power spectral density function estimation. *Biometrika 73*(1), 135–141.

Taqqu, M., V. Teverovsky, and W. Willinger (1995). Estimators for long-range dependence: an empirical study. *Fractals 3*, 785–798.

Teverovsky, V. and M. Taqqu (1997). Testing for long-range dependence in the presence of shifting means or a slowly declining trend, using a variance-type estimator. *Journal of Time Series Analysis 18*(3), 279–304.

Theiler, J., S. Eubank, A. Longtin, B. Galdrikian, and J. Farmer (1992). Testing for nonlinearity in time series: the method of surrogate data. *Physica D 58*, 77–94.

Theiler, J. and D. Prichard (1996). Constrained-realization Monte-Carlo for hypothesis testing. *Physica D 58*, 221–235.

Tudor, C. (2006). Analysis of the Rosenblatt process. *Working paper, www.archives-ouvertes.fr*.

Wei, W. (1990). *Time series analysis: univariate and multivariate methods.* Addison-Wesley Publishing Company.

Whittle, P. (1951). *Hypothesis testing in time series analysis.* Hafner, New York.

Woodward, W., Q. Cheng, and H. Gray (1998). A k-factor GARMA long memory model. *Journal of time series analysis 19*(4), 485–504.

Woodward, W., H. Gray, and N. Zhang (1989). On generalized fractional process. *Journal of time series analysis 10*, 233–257.

Yajima, Y. (1985). On estimation of long-memory time series models. *Australian Journal of Statistics 27*, 303–320.