On the Correlation Matrix of the Discrete Fourier Transform and the Fast Solution of Large Toeplitz Systems For Long-Memory Time Series

Willa Chen* Clifford M. Hurvich† Yi Lu ‡

July 28, 2004

Abstract

For long-memory time series, we show that the Toeplitz system $\Sigma_n(f)x = b$ can be solved in $O(n \log^{5/2} n)$ operations using a well-known version of the preconditioned conjugate gradient method, where $\Sigma_n(f)$ is the $n \times n$ covariance matrix, f is the spectral density and b is a known vector. Solutions of such systems are needed for optimal linear prediction and interpolation. We establish connections between this preconditioning method and the frequency domain analysis of time series. Indeed, the running time of the algorithm is determined by rate of increase of the condition number of the correlation matrix of the discrete Fourier transform vector, as the sample size tends to ∞. We derive an upper bound for this condition number. The bound is of interest in its own right, as it sheds some light on the widely-used but heuristic approximation that the standardized DFT coefficients are uncorrelated with equal variances. We present applications of the preconditioning methodology to the forecasting and smoothing of volatility in a long memory stochastic volatility model, and to the evaluation of the Gaussian likelihood function of a long-memory model.

1 Introduction

The solution of Toeplitz systems plays an important role in time series analysis, for example in linear forecasting and the evaluation of the Guassian likelihood function. Given a zero-mean weakly stationary time series $\{X_t\}_{t=-\infty}^{\infty}$ with spectral density f, the minimum mean squared error linear predictor based on $X = (X_0, \ldots, X_{n-1})'$ is obtained as the solution for x in a linear system of form $\Sigma_n(f)x = b$ involving the $n \times n$ Toeplitz covariance matrix $\Sigma_n(f) = cov(X)$, where b is known. We are interested in the fast solution of such systems when n is large, in the face of long-range dependence, so that we can efficiently forecast volatility and estimate long-memory models. Here, the autocovariances decline slowly, at a power law rate, and $\Sigma_n(f)$ is quite ill-conditioned. We will assume that the autocovariances are known. Although there is not always an analytical expression for these, they can be computed to any desired degree of

*Department of Statistics, Texas A & M University, College Station, Texas 77840, USA.
†New York University: 44 W. 4'th Street, New York NY 10012, USA.
‡New York University: 44 W. 4'th Street, New York NY 10012, USA.

accuracy for lags 0 to $n-1$ in $O(n \log n)$ operations using an algorithm of Bertelli and Caporin (2002) for all of the models in widespread use.

The Levinson Algorithm (Levinson 1946; see also Brockwell and Davis 1991, Percival and Walden 1993) yields the solution of an $n \times n$ Toeplitz system in $O(n^2)$ operations. By exploiting the Toeplitz structure, Levinson's Algorithm achieves a strong improvement over the $O(n^3)$ cost of solving general $n \times n$ linear systems. The algorithm also sheds light on time series analysis itself, revealing interesting and useful connections between autocorrelations, partial autocorrelations, the optimal linear prediction coefficients, and the variances of the one-step-ahead linear prediction errors.

For the analysis of time series in the frequency domain, the Fast Fourier Transform (FFT; Cooley and Tukey 1965; Gentleman and Sande 1966) yields the n discrete Fourier transform (DFT) coefficients of any n-dimensional vector in $O(n \log n)$ operations, as compared to the naive cost of $O(n^2)$. This allows for implementation of the FFT even for extremely large values of n, and accounts for the ubiquitous use of FFT algorithms in time series analysis and signal processing.

In recent years, it has been realized that the FFT can also be used for fast solution of Toeplitz systems, yielding algorithms that are far faster than Levinson's. Two main classes of FFT-based algorithms for solving large Toeplitz systems have developed: Superfast algorithms (see Ammar and Gragg 1988) and Preconditioned Conjugate Gradient (PCG) algorithms (see Axelsson and Barker 1984, Golub and Van Loan 1996 for the general algorithm, R. Chan and Ng 1996, Strang 1986 and T. Chan 1988 for the Toeplitz case). Unfortunately, in order to be numerically stable and accurate, the Superfast algorithms require that the Toeplitz matrix be well-conditioned (see Bunch 1985), which is not the case for long-memory series.

The PCG methods, which yield a numerical solution to the system given a desired degree of accuracy, require iterative approximations to the solution of an equivalent linear system $\widetilde{C}_n^{-1}(f)\Sigma_n(f)x = \widetilde{C}_n^{-1}(f)b$, where the matrix $\widetilde{C}_n(f)$, called a *preconditioner*, is defined in terms of the entries of the first row of $\Sigma_n(f)$, i.e., the autocovariances up to lag $n-1$, in such a way that each iteration can be carried out in $O(n \log n)$ operations using the FFT. It is known that the number of iterations required to attain convergence of the PCG algorithm is proportional to the square root of the condition number of the preconditioned matrix $\widetilde{C}_n^{-1}(f)\Sigma_n(f)$. Thus the growth rate of this condition number determines the asymptotic computational complexity of the PCG algorithm. This growth rate has been derived for a variety of situations and choices for the preconditioner (see, e.g., R. Chan 1989, R. Chan and Yeung 1992, R. Chan Yip and Ng 2000), but under conditions that rule out long memory. We will focus in this paper on a particular circulant preconditioner $\widetilde{C}_n(f)$ due to T. Chan (1988), defined by Equation (3) in the next section.

An interesting fact, apparently not previously noted, is that the preconditioned covariance matrix based on $\widetilde{C}_n(f)$ is equivalent, up to a similarity transform, to the correlation matrix of the DFT coefficients. This equivalence allows us to apply existing results from the long-memory time series literature on properties of covariances between DFTs to help us to derive sharp bounds for the condition number of the preconditioned matrix. Another interesting and useful connection is that the eigenvalues of $\widetilde{C}_n(f)$ are equal to the expected values of the periodogram ordinates at the Fourier frequencies.

We will show that for long-memory time series, the computational cost of the PCG algorithm based on $\widetilde{C}_n(f)$ is $O(n \log^{5/2} n)$. This rate follows from our Theorem 1 below, which states that under appropriate assumptions, the condition number of the preconditioned matrix is $O(\log^3 n)$. Our proof is based on

frequency domain time series techniques, exploiting the fact that the preconditioned matrix and the correlation matrix of the DFTs have the same condition number.

The derivation of the condition number of the correlation matrix of the DFTs is of interest in its own right, as it sheds some light on the widely-used but heuristic approximation that the standardized DFT coefficients are uncorrelated with equal variances. Our Lemma 2 is also of independent interest, as its proof establishes a lower bound on the smallest eigenvalue of the product of a Toeplitz matrix generated by a spectral density f and another Toeplitz matrix generated by $1/f$. Results on the trace of such a product in a long memory context have been studied by Dahlhaus (1989, Theorem 5.1). The approximation of this product by an identity matrix was used to justify in the long-memory case Whittle's (1953) approximation to the log likelihood for the parameters of Gaussian time series models.

In Section 2, we state our assumptions and establish notation. In Section 3, we present more details on PCG using $\widetilde{C}_n(f)$, and in Section 4 we point out its previously unexplored connections with DFTs. In Section 5, we present our results on the condition number of the correlation matrix of DFTs, along with a discussion to put these results into the context of existing results from the time series literature.

In Section 6, we present three applications of the PCG algorithm. The first two applications are for the long memory stochastic volatility (LMSV) model of Breidt, Crato and de Lima (1998) and Harvey (1998). For this model, there are no simple expressions for the infinite-order autoregressive $AR(\infty)$ coefficients, so one cannot obtain an approximation to the optimal linear predictor by truncating the $AR(\infty)$ predictor. Furthermore, for the purposes of diagnostic checking, one would like to be able to evaluate some proxy for the latent volatility process. This was suggested originally by Harvey (1998), who proposed to use the linear combination of log squared returns which minimizes the mean squared error. This minimization problem reduces to a Toeplitz system in the covariance matrix $\Sigma_n(f)$ of the log squared returns. Harvey (1998) mentioned that there are fast algorithms for solving such systems. He did not provide a reference, but presumably was referring to the class of Superfast algorithms, which as we have mentioned above do not actually provide a fast solution to this problem. However, as we show in this paper, the PCG algorithm does achieve this. An empirically successful application of the PCG algorithm for forecasting volatility with the LMSV model in large data sets was given by Deo, Hurvich and Lu (2004), though the computational complexity was not established there. Our search for a rigorously justifiable, computationally efficient solution to the problems of forecasting and smoothing in the LMSV model led eventually to the development of this paper.

The third application in Section 6 is to the evaluation of the Gaussian likelihood function of a long-memory model, focusing primarily on the Autoregressive Fractionally Integrated Moving Average $ARFIMA(p,d,q)$ case. Currently, one can use Levinson's algorithm to evaluate the likelihood function in $O(n^2)$ operations as advocated by Sowell (1992). We show that the quadratic form term in the likelihood can be evaluated in $O(n \log^{5/2} n)$ operations using the PCG algorithm together with a methodology called Toeplitz Embedding, described in Appendix C. We also provide an extremely accurate and easily computed approximation to the determinant term in the likelihood. We present simulations comparing the performance of the Maximum Likelihood method with that of Whittle's method and an approximate algorithm of Haslett and Raftery (1989) as implemented in Splus. We also investigate the effect on the maximum likelihood method of estimating the mean in the case where it is unknown.

2 Assumptions and Notation

We consider the zero-mean weakly stationary time series $\{X_t\}$ with spectral density $f(\omega)$, $\omega \in [-\pi, \pi]$. We impose the long-memory structure by the following assumption.

Assumption 1.

$$f(\omega) = |1 - e^{-i\omega}|^{-2d} f^*(\omega) \ , \ \omega \in [-\pi, \pi] \ ,$$

where $d \in (-1/2, 1/2)$, $f^*(\omega)$ *is positive on* $[-\pi, \pi]$ *and differentiable on* $[-\pi, \pi] \setminus \{0\}$ *and there exists* $c \in (0, \infty)$ *such that*

$$|f^{*\prime}(\omega)| \leq c|\omega|^{-1} \ , \ \omega \in [-\pi, \pi] \setminus \{0\} \ .$$

Assumption 1, also used by Moulines and Soulier (1999), is satisfied for all stationary invertible ARFIMA models with $d \in (-1/2, 1/2)$. The assumption is also satisfied in the signal plus noise case where f is the sum of the spectral density of a stationary invertible ARFIMA spectral density with $d \in (0, 1/2)$ and a white noise spectral density.

Given a partial realization $\{X_t\}_{t=0}^{n-1}$, define the discrete Fourier Transform and periodogram

$$J_{X,j} = \frac{1}{\sqrt{2\pi n}} \sum_{t=0}^{n-1} X_t e^{i\omega_j t} \ , \ I_{X,j} = |J_{X,j}|^2 \ ,$$

where $\omega_j = 2\pi j/n$ is the j'th Fourier frequency, $j = 0, \ldots, n-1$. Denote the autocovariance sequence of $\{X_t\}_{t=-\infty}^{\infty}$ by $\{c_r\}_{r=-\infty}^{\infty}$ where $c_r = E[X_t X_{t-r}]$. The expected value of the periodogram can be written (see Priestley 1981, pp. 395, 417-418) as

$$E[I_{X,j}] = \int_{-\pi}^{\pi} K_n(\omega_j - \omega) f(\omega) d\omega = \frac{1}{2\pi} \sum_{|r|<n} \frac{n - |r|}{n} c_r e^{ir\omega_j} \ , \tag{1}$$

for $j = 0, \ldots, n-1$, where

$$K_n(\mu) = \frac{\sin^2(n\mu/2)}{2\pi n \sin^2(\mu/2)}$$

is the Fejer kernel.

Let $\Sigma_n(f)$ denote the $n \times n$ Toeplitz covariance matrix of $(X_0, \ldots, X_{n-1})'$ with (j,k) element

$$\Sigma_n(f)_{j,k} = c_{j-k} = E(X_j X_k) = \int_{-\pi}^{\pi} e^{i(j-k)\omega} f(\omega) d\omega \ , \ j,k = 0, \ldots, n-1 \ . \tag{2}$$

For any positive definite Hermitian matrix A, denote the smallest and largest eigenvalues by $\lambda_{\min}(A)$ and $\lambda_{\max}(A)$, and denote the condition number by $\kappa(A) = \lambda_{\max}(A)/\lambda_{\min}(A)$.

3 Preconditioned Conjugate Gradient Algorithm and $\widetilde{C}_n(f)$

As we explain in Appendix A, $\kappa[\Sigma_n(f)] = O(n^{2|d|})$, so when $d \in (-1/2, 1/2) \setminus \{0\}$, $\Sigma_n(f)$ is not well-conditioned. As shown in Axelsson and Barker (1984), and Golub and Van Loan (1996), the number of

iterations required in the non-preconditioned Conjugate Gradient algorithm (CG) to reach a specified degree of relative reduction in the error is proportional to $\sqrt{\kappa[\Sigma_n(f)]}$ which in this case would be $O(n^{|d|})$. Furthermore, as explained in Appendix C, the cost of each iteration of CG in our situation is $O(n \log n)$. Thus, the cost of the conjugate gradient algorithm without preconditioning is $O(n^{1+|d|} \log n)$. This is faster than Levinson's algorithm, but can still be quite slow if $|d|$ is close to 1/2.

Preconditioning is a technique for improving the condition number of a matrix. Suppose that $C(f)$ is a symmetric, positive definite matrix that approximates $\Sigma_n(f)$, but is easier to invert. We can solve $\Sigma_n(f)x = b$ indirectly by solving the preconditioned system.

$$C^{-1}(f)\Sigma_n(f)x = C^{-1}(f)b.$$

The preconditioned conjugate gradient algorithm consists of applying the ordinary conjugate gradient algorithm to the preconditioned system. As explained in Appendix C, the number of iterations required for PCG is proportional to $\sqrt{\kappa[C^{-1}(f)\Sigma_n(f)]}$, which needs to be smaller than $\sqrt{\kappa[\Sigma_n(f)]}$ in order for PCG to be more computationally efficient than CG.

T. Chan's preconditioner $\widetilde{C}_n(f)$ is the $n \times n$ matrix with (j,k)th entry

$$\widetilde{C}_n(f)_{j,k} = \widetilde{c}_{j-k} = \frac{1}{2\pi}\left(\frac{n-|j-k|}{n}\right) c_{j-k} + \frac{1}{2\pi}\frac{|j-k|}{n} c_{n-|j-k|} \ , \tag{3}$$

for $j,k = 0,\ldots,n-1$. For notational convenience, we have divided here by a factor of 2π compared to T. Chan's original definition. Note that $\widetilde{c}_{j-k} = \widetilde{c}_{n-|j-k|}$, so $\widetilde{C}_n(f)$ is a circulant matrix. By the properties of circulant matrices (see, e.g., Brockwell and Davis 1991, Proposition 4.5.1, page 134) , $\widetilde{C}_n(f)$ has eigenvalues

$$\lambda_j[\widetilde{C}_n(f)] = \sum_{r=0}^{n-1} \widetilde{c}_r e^{ir\omega_j} \ , \tag{4}$$

and corresponding eigenvectors

$$v_j = \frac{1}{\sqrt{n}}\left(1, e^{i\omega_j}, e^{2i\omega_j}, \ldots, e^{(n-1)i\omega_j}\right)' \ ,$$

for $j = 0,\ldots,n-1$.

As we explain in Appendix C, if $\widetilde{C}_n(f)$ is used, the cost of each iteration of PCG is $O(n \log n)$ operations, so the total cost of the PCG algorithm is $O\left(n \log n \sqrt{\kappa[\widetilde{C}_n^{-1}(f)\Sigma_n(f)]}\right)$.

Next, we compare the number of iterations required to achieve a Σ_n norm (see Appendix C) of 10^{-20} for the error vector in the CG algorithm and the PCG algorithm (using $\widetilde{C}_n(f)$) for solving a Toeplitz system $\Sigma_n(f)x = b$, in a long-memory forecasting context. Table 1 describes the results for a long-memory ARFIMA(0,d,0) time series $\{X_t\}$ with spectral density

$$f(\omega) = \frac{\sigma_\eta^2}{2\pi}|2\sin(\omega/2)|^{-2d} \ , \qquad -\pi \le \omega \le \pi,$$

with $d = 0.37$, innovation variance $\sigma_\eta^2 = 0.27$ and autocovariances

$$c_k = \frac{\sigma_\eta^2 \Gamma(1-2d)\Gamma(k+d)}{\Gamma(d)\Gamma(1-d)\Gamma(k-d+1)} \ , \qquad k = 0,1,2,\ldots.$$

The minimum mean squared error one-step-ahead linear forecasting coefficients for X_n based on $X_{n-1}, \ldots, X_0$ are given by the solution to the Toeplitz system above, with $b = (c_1, \ldots, c_n)'$.

The numerical results in Table 1 confirm that preconditioning is beneficial.

Table 1: Number of Iterations for CG and PCG algorithms based on ARFIMA (0,d,0)

n	2^2	2^3	2^4	2^5	2^6	2^7	2^8	2^9	2^{10}	2^{11}	2^{12}	2^{13}	2^{14}	2^{15}
CG	4	9	14	18	25	32	42	54	67	88	112	143	186	238
PCG	4	6	7	7	7	8	9	8	9	9	10	11	10	11

4 Connections Between $\widetilde{C}_n(f)$ and Properties of DFTs

We observe first that the eigenvalues of $\widetilde{C}_n(f)$ are equal to the expected values of the periodogram ordinates, $\{E[I_{X,j}]\}_{j=0}^{n-1}$. Indeed, substituting the formula (3) for $\tilde{c}_r$ in (4) yields

$$\lambda_j[\widetilde{C}_n(f)] = \frac{1}{2\pi}\sum_{r=0}^{n-1}\left[\frac{n-r}{n}\,c_r + \frac{r}{n}\,c_{n-r}\right]e^{ir\omega_j} = \frac{1}{2\pi}\sum_{|r|<n}\frac{n-|r|}{n}\,c_r e^{ir\omega_j} = E[I_{X,j}] \tag{5}$$

by (1). Equation (5) suggests a link between the properties of T. Chan's circulant preconditioner $\widetilde{C}_n(f)$ and the frequency domain analysis of time series. We now expand this link by showing that the preconditioned matrix and the correlation matrix of the DFTs are similar matrices, and therefore have the same eigenvalues and condition number.

Define the unitary matrix $V_n = (v_0, \ldots, v_{n-1})$ and $\Lambda[\widetilde{C}_n(f)] = \text{diag}\left\{\lambda_0[\widetilde{C}_n(f)], \ldots, \lambda_{n-1}[\widetilde{C}_n(f)]\right\}$ so that

$$\widetilde{C}_n(f) = V_n\Lambda[\widetilde{C}_n(f)]V_n^*. \tag{6}$$

Denoting the similarity between two matrices A and B by $A \sim B$, we have

$$\begin{aligned}\widetilde{C}_n^{-1}(f)\Sigma_n(f) &\sim \widetilde{C}_n^{-1/2}(f)\Sigma_n(f)\widetilde{C}_n^{-1/2}(f) = V_n\Lambda^{-1/2}[\widetilde{C}_n(f)]V_n^*\Sigma_n(f)V_n\Lambda^{-1/2}[\widetilde{C}_n(f)]V_n^* \\ &\sim \Lambda^{-1/2}[\widetilde{C}_n(f)]V_n^*\Sigma_n(f)V_n\Lambda^{-1/2}[\widetilde{C}_n(f)] := M(f)\ . \end{aligned} \tag{7}$$

Let $D_n(\omega) = (2\pi n)^{-1/2}\sum_{j=0}^{n-1} e^{i\omega j}$ be the Dirichlet kernel. Then the (j,k)th entry of $M(f)$, $j,k = 0, \ldots, n-1$ is

$$\begin{aligned} m_{jk} &= \left\{\int_{-\pi}^{\pi} K_n(\omega_j - \omega)\, f(\omega)\, d\omega \int_{-\pi}^{\pi} K_n(\omega_k - \omega)\, f(\omega)\, d\omega\right\}^{-1/2} \int_{-\pi}^{\pi} D_n(\omega - \omega_j)\, D_n(\omega_k - \omega)\, f(\omega)\, d\omega \\ &= \{E[I_{X,j}]E[I_{X,k}]\}^{-1/2}\, E[\overline{J}_{X,j}J_{X,k}]. \end{aligned}$$

Hence the matrix $M(f)$ is the correlation matrix of the DFT vector, $(J_{X,0}, \ldots, J_{X,n-1})'$. Furthermore, from the discussion above, the condition number of $M(f)$ is the same as the condition number of the preconditioned matrix $\widetilde{C}_n^{-1}(f)\Sigma_n(f)$.

5 The Condition Number of the Correlation Matrix of DFTs

Results on pairwise correlations of DFTs from a long memory series have been obtained by Robinson (1995) in a local context, Moulines and Soulier (1999) in a global context. These bounds have been used to establish properties of various semiparametric estimators of the long memory parameter. Matrix properties of the full $n \times n$ correlation matrix are also of intrinsic interest. For principal submatrices of any fixed size, Moulines and Soulier (1999) have shown that as n increases the smallest eigenvalue is bounded away from zero, and derived an upper bound for the spectral radius of the difference between this submatrix and the identity matrix. However, the properties of the $n \times n$ correlation matrix as $n \to \infty$ remain an open question. We will derive bounds for the extreme eigenvalues of this matrix, which imply that the condition number is $O(\log^3 n)$.

An approximation often applied in a short-memory context is to treat the vector $J = (J_{X,0}, \ldots, J_{X,n-1})'$ as having covariance matrix $Cov(J) = E[JJ^*] = diag[f(\omega_0), \ldots, f(\omega_{n-1})]$, so that the correlation matrix of J is an $n \times n$ identity matrix. Thus, the standardized periodogram ordinates $I_{X,j}/f(\omega_j)$ for $j = 1, \ldots, n/2$ are often treated as independent exponential random variables, in keeping with Whittle's approximation to the Gaussian log likelihood. From (7), the approximation $Cov(J) = diag[f(\omega_0), \ldots, f(\omega_{n-1})]$ treats $\Sigma_n(f)$ as if it were the circulant matrix $V_n \Lambda[\widetilde{C}_n(f)] V_n^*$. Thus, Whittle's approximation may be viewed as a circulant approximation to the covariance matrix $\Sigma_n(f)$.

Even in short memory time series, none of the above approximations is exactly true except in the case of Gaussian white noise. Nevertheless, Brockwell and Davis (1991, Proposition 4.5.2, p. 136) show that the entries of $Cov(J) - diag[f(\omega_0), \ldots, f(\omega_{n-1})]$ converge uniformly to zero, assuming that the autocovariances are absolutely summable. Note, however, that this assumption rules out the long-memory case.

In long-memory time series, $diag[f(\omega_0), \ldots, f(\omega_{n-1})]$ is either undefined or has a zero entry, due to the nature of the spectral density at zero frequency. However, $M(f)$, the correlation matrix of J, remains well-defined and positive definite, and we analyze its extreme eigenvalues here.

We present an upper bound for $\lambda_{\max}[M(f)]$ in the following lemma.

Lemma 1. *Let $M(f)$ be given by (7) with $f(\omega)$ satisfying Assumption 1. Then*

$$\lambda_{\max}[M(f)] = O\left(\log^{\frac{3}{2}} n\right).$$

Our next lemma provides a lower bound for $\lambda_{\min}\left[\widetilde{C}_n^{-1}(f)\Sigma_n(f)\right]$.

Lemma 2. *Let $\Sigma_n(f)$ and $\widetilde{C}_n(f)$ be the matrices defined by (2) and (3) with f satisfying Assumption 1. Then*

$$\lambda_{\min}\left[\widetilde{C}_n^{-1}(f)\Sigma_n(f)\right] \geq C \log^{-\frac{3}{2}} n \ ,$$

where C is a positive constant.

Combining Lemmas 1, 2 and (7), we immediately obtain the following theorem.

Theorem 1. *Let $\Sigma_n(f)$ and $\widetilde{C}_n(f)$ be defined by (2) and (3) with f satisfying Assumption 1. Then the condition number of $\widetilde{C}_n^{-1}(f)\Sigma_n(f)$ is $\kappa\left[\widetilde{C}_n^{-1}(f)\Sigma_n(f)\right] = O\left(\log^3 n\right)$.*

6 Applications in Forecasting, Smoothing and Maximum Likelihood

6.1 Forecasting from a Long Memory Stochastic Volatility Model

An LMSV model for returns $\{r_t\}$ is given by

$$r_t = \sigma \exp\left(h_t/2\right)\varepsilon_t, \tag{8}$$

where $\sigma > 0$, the ε_t are *i.i.d.* with mean zero and variance 1, and $\{h_t\}$ is a stationary zero-mean Gaussian long-memory process independent of $\{\varepsilon_t\}$. Two popular choices for the distribution of ε_t are the standard normal distribution and a unit-variance normalized t-distribution with ν degrees of freedom. We will assume here that $\{h_t\}$ follows an Autoregressive Fractionally Integrated Moving Average ARFIMA(p, d, q) given by

$$\Phi\left(B\right)\left(1-B\right)^d h_t = \Theta\left(B\right)\eta_t,$$

where B denotes the backshift operator, the η_t are *i.i.d.* N(0,σ_η^2), $0 < d < 0.5$, $\Phi\left(B\right)$ and $\Theta\left(B\right)$ are polynomials of order p and q respectively with all roots outside the unit circle. It should be emphasized that all of our procedures can be easily extended to accommodate other long-memory model specifications for $\{h_t\}$, such as the Fractional Exponential (FEXP) model. See Hurvich (2002) for details on the FEXP model.

The spectral density of $\{\log r_t^2\}$ is $f(\omega) = f_1(\omega) + \mathrm{var}(\log \epsilon_t^2)/(2\pi)$, where $f_1(\omega)$ is the spectral density of $\{h_t\}$. The variance of $\{\log r_t^2\}$ is $\mathrm{var}(h_t) + \mathrm{var}(\log \epsilon_t^2)$, and the autocovariances of $\{\log r_t^2\}$ for all lags greater than zero are the same as the corresponding autocovariances of $\{h_t\}$.

Suppose that we are interested in forecasting $\{\log r_t^2\}$, which serves as a proxy for volatility. There is no simple formula for the $AR(\infty)$ coefficients of this series, so we cannot attempt to approximate the optimal linear forecast of $\log r_n^2$ based on mean-corrected values of $\log r_{n-1}^2, \ldots, \log r_0^2$ by truncating the $AR(\infty)$ predictor. Fortunately, as we have justified theoretically, the optimal linear forecasting coefficients for this model can be obtained efficiently by solving the Toeplitz system: $\Sigma_n x = b$ using the PCG algorithm. For example, the optimal one-step-ahead linear forecasting coefficients can be obtained by solving the system with $b = (c_1, \ldots, c_n)'$, the autocovariances of $\{h_t\}$ for lags 1 to n. Here, Σ_n is the covariance matrix of $\{\log r_t^2\}_{t=0}^{n-1}$. This methodology was used to forecast aggregates of future squared returns in Deo, Hurvich and Lu (2004).

In Table 2, we compare the number of iterations required to achieve a Σ_n norm of 10^{-20} for the error vector in the CG and PCG algorithms for solving the one-step prediction problem described above, assuming that $\{h_t\}$ is $ARFIMA(0, d, 0)$ with $d = 0.37$, innovation variance $\sigma_\eta^2 = 0.27$ and $var \log \epsilon_t^2 = 5\sigma_\eta^2$. The preconditioning is seen to speed the algorithm.

Table 2: Number of Iterations for CG and PCG Algorithms: LMSV Model

n	2^2	2^3	2^4	2^5	2^6	2^7	2^8	2^9	2^{10}	2^{11}	2^{12}	2^{13}	2^{14}	2^{15}
CG	4	7	8	10	13	15	18	23	27	34	43	54	66	87
PCG	4	5	5	6	6	7	7	8	8	8	9	9	9	10

6.2 Smoothing of Latent Process for LMSV model

As pointed out in Harvey (1998), when the latent process $\{h_t\}$ in Equation (8) is assumed to follow a finite-order autoregressive process, the minimum mean square linear estimator (MMSLE) of h_t can be computed easily with a state-space smoothing algorithm. However, such an approach is not easy to implement with an LMSV model. The $AR(\infty)$ representation of $\{h_t\}$ must be truncated and the truncation must be at large lags due to the long-memory nature of the process. Harvey (1998) proposed a direct approach for the exact smoothing which is feasible for high frequency data only with the PCG algorithm.

The MMSLE of h_t is given by

$$\hat{h} = \Sigma_h \Sigma_y^{-1}(y - \mu 1) \tag{9}$$

where $\mu = E[\log r_t^2]$, $y = (\log r_0^2, \log r_1^2, \ldots, \log r_{n-1}^2)'$, Σ_y is the autocovariance matrix of y, Σ_h is the autocovariance matrix of $(h_0, \ldots, h_{n-1})'$, and 1 is an $n \times 1$ vector of ones. Since $\{h_t\}$ is uncorrelated with $\{\log \epsilon_t^2\}$, and since $\{\log \epsilon_t^2\}$ has zero autocorrelations at all nonzero lags, $\Sigma_y = \Sigma_h + (\mathrm{var} \log \epsilon_t^2) I$, where I is an $n \times n$ identity matrix. The quantity μ can be estimated by the sample mean of $\log r_t^2$, though we ignore the distinction between these two quantities here.

In Equation (9), Σ_y and Σ_h are both $n \times n$ Toeplitz matrices. Efficient computation of $\hat{h}$ can be carried out in a two-step procedure. First, the PCG algorithm is applied to solve the Toeplitz system $\Sigma_y x = (y - \mu 1)$ for $x = \Sigma_y^{-1}(y - \mu 1)$. Then the circulant embedding technique (See Appendix C) is applied to evaluate $\Sigma_h x$.

We consider the log squared returns (seasonally adjusted) for S&P 500 index, recorded every 30 minutes, from Nov. 21, 1994 to Nov. 2, 1999, a total of 15000 observations. Figure 1 plots the log squared returns, and the smoothed latent volatility process $\{\hat{h}_t\}$ based on an LMSV-$ARFIMA(1, d, 0)$ model. The model was fitted to the log squared returns using Whittle's method.

The normalized log squared returns $\log r_t^2 - \hat{h}_t$ can be used for diagnostic checking and model selection. Fig 2 (a) and (b) present the sample autocorrelations of the raw and normalized log squared returns. Apparently, the $ARFIMA(1, d, 0)$ model for the latent process is adequate since most of the persistence in $\log r_t^2$ is captured by the model. The negative autocorrelations at the first few lags in Figure 2 (b) could be due to either mild misspecification of the model or to spurious autocorrelations introduced by the smoothing methodology. Simulations from this model (not shown here) exhibit the latter phenomenon. We will not pursue this issue further here, however.

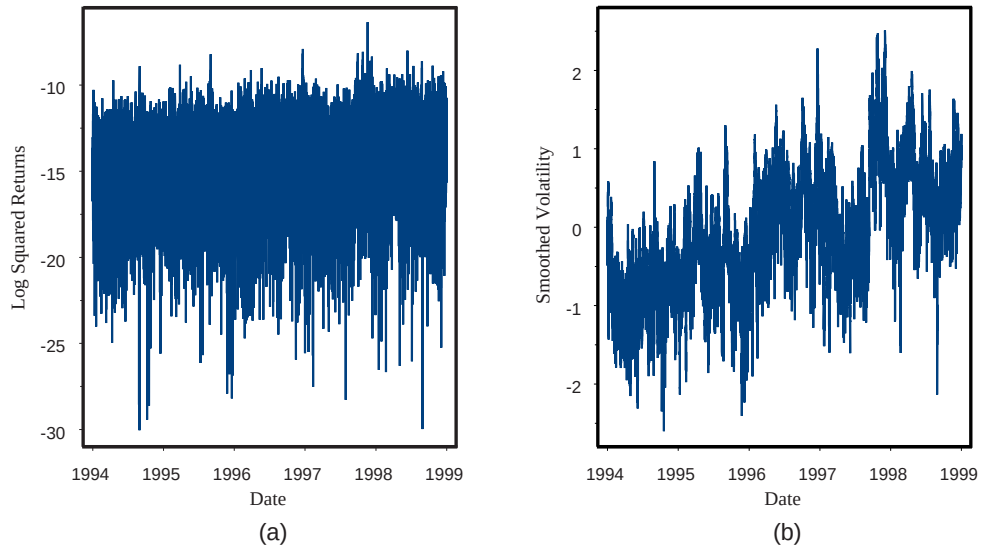

Figure 1: Time series plots of (a) Log Squared Returns, $\log r_t^2$ and (b)Smoothed Latent Volatility Process, $\hat{h}_t$. The plots are based on S&P 500 index, from Nov. 21, 1994 to Nov. 2, 1999, a total of 15000 observations.

6.3 Evaluation of the Gaussian Log Likelihood Function

Suppose we can assume a parametric model, indexed by a vector θ, for a zero-mean weakly stationary series $\{y_t\}$. Let $\Sigma_{n,\theta}$ denote the $n \times n$ covariance matrix for $(y_0, \ldots, y_{n-1})$ under the model θ. The Gaussian log likelihood function, multiplied by -2, is

$$-2 \log L(\theta) = n \log 2\pi + \log |\Sigma_{n,\theta}| + y' \Sigma_{n,\theta}^{-1} y \tag{10}$$

Exact Evaluation of this log likelihood for ARFIMA models was considered by Sowell (1992), who focused on the exact calculation of the autocovariances $c_{0,\theta}, \ldots, c_{n-1,\theta}$ under the model. But given these autocovariances, the calculation of the quadratic form $y' \Sigma_{n,\theta}^{-1} y$ using Levison's algorithm as advocated by Sowell (1992) would still require $O(n^2)$ operations, rendering the calculation of the maximum likelihood estimator infeasible in large sample sizes. Recently, Bertelli and Caporin (2002) proposed an FFT-based algorithm for numerically evaluating $c_{0,\theta}, \ldots, c_{n-1,\theta}$ for any ARFIMA model, to any desired degree of accuracy in $O(n \log n)$ operations. This methodology can also be employed for other model classes, such as the Fractional Exponential (FEXP) model (see Hurvich 2002). Furthermore, for either the ARFIMA or the FEXP model class, we can use the PCG method developed in this paper together with the circulant embedding described in Appendix C (see Equation (22)) to evaluate $y' \Sigma_{n,\theta}^{-1} y$ in $O(n \log^{5/2} n)$ operations.

Unfortunately, the determinant term $\log |\Sigma_{n,\theta}|$ cannot be calculated using PCG. While there do exist Superfast algorithms for evaluating this determinant in $O(n \log^2 n)$ operations (see Kravanja and Barel 2000), their stability in the long-memory case has not been explored. We will consider instead two approximations to this determinant term. The first approximation is from Böttcher and Silbermann (1999 Theorem 5.47, Page 177), and is given in Appendix D. The second approximation is the one used in Whittle's method, $n \log(2\pi) + \sum_{j=1}^{n-1} \log f_\theta(\omega_j)$, where f_θ is the spectral density under the model θ. In Table 3, we report the exact determinant together with the two approximations for several $ARFIMA(0, d, 0)$ and $ARFIMA(1, d, 0)$ models for various values of d with $n = 500$ and unit innovation variance. The $ARFIMA(1, d, 0)$ had $\Phi(B) = 1 - 0.35B$. It is seen that the approximation ($B\&S$) of Böttcher and Silbermann (1999) matches the exact value to several significant figures, while Whittle's approximation

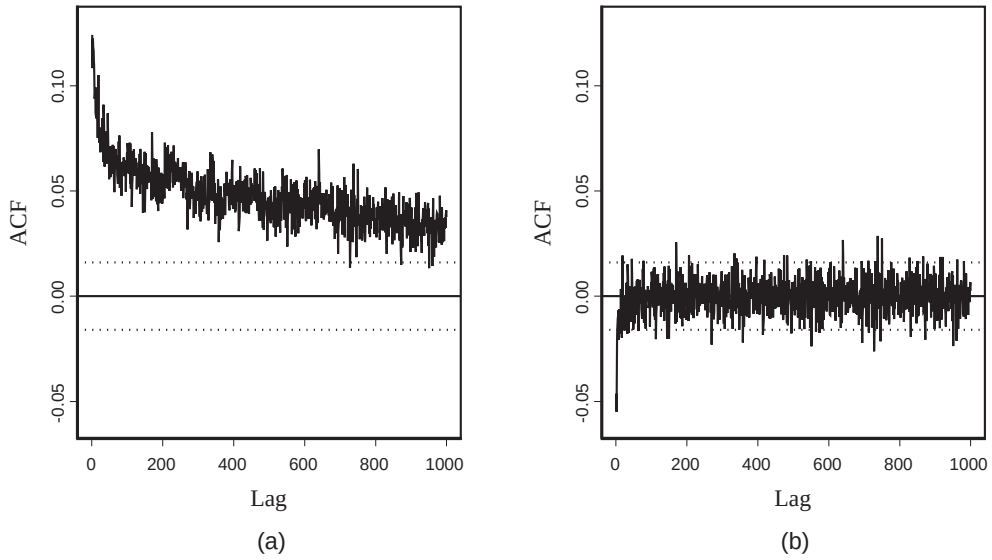

Figure 2: ACF of (a) Raw Log Squared Returns, $\log r_t^2$ and (b) Normalized Log Squared Returns, $\log r_t^2 - \hat{h}_t$.

is far less accurate.

Table 3: Log Determinant of Toeplitz Covariance Matirx for $ARFIMA$ Processes

Determinant	Model	$d = -0.45$	$d = -0.25$	$d = -0.05$	$d = 0.05$	$d = 0.25$	$d = 0.45$
Exact	$(0, d, 0)$	1.38147	0.44755	0.01909	0.01992	0.56576	2.64280
B&S		1.38129	0.44751	0.01909	0.01992	0.56579	2.64298
Whittle		5.59315	3.10730	0.62146	-0.62146	-3.10730	-5.59315
Exact	$(1, d, 0)$	1.12488	0.36297	0.10670	0.19368	0.91196	3.16162
B&S		1.12426	0.36280	0.10670	0.19368	0.91186	3.16136
Whittle		4.73158	2.24574	-0.24011	-1.48303	-3.96887	-6.45471

In Table 4 we report the mean squared errors of several estimators of d in the $ARFIMA(0, d, 0)$ model described above. The parameters to be estimated are the innovation variance, and the memory parameter d. We found similar results for the $ARFIMA(1, d, 0)$ model, which we do not report here to save space. For each of several choices of d we generated 1000 simulated replications, with sample sizes $n = 50, 500, 5000$. The first method was Maximum Likelihood (ML), based on minimizing (10), but replacing $\log|\Sigma_{n,\theta}|$ by the approximation of Böttcher and Silbermann. We note here that we also tried minimizing the exact log likelihood function (10) using the exact determinant in the $ARFIMA(0, d, 0)$ case, and the mean squared errors were identical to those obtained using Böttcher and Silbermann's, to several significant figures. The other two estimators considered in Table 4 are Whittle's estimator, i.e., the minimizer of

$$\sum_{j=1}^{n-1} [\log f_\theta(\omega_j) + I(\omega_j)/f_\theta(\omega_j)]$$

and a modified maximum likelihood (MML) estimator that minimizes

$$\sum_{j=1}^{n-1}[\log f_\theta(\omega_j)] + y'\Sigma_{n,\theta}^{-1}y \ .$$

Note that both of these objective functions can be evaluated in $O(n \log n)$ operations. In the MML objective function, the log determinant term from (10) is replaced by the approximation to it used in Whittle's method, but the second term is the same quadratic form that appears in the exact likelihood function, rather than Whittle's approximation to it, $\sum_{j=1}^{n-1} I(\omega_j)/f_\theta(\omega_j)$.

Table 4: MSE for Estimators of d for $ARFIMA(0, d, 0)$ Process When Mean is Known

Estimator	Sample Size	$d = -0.45$	$d = -0.25$	$d = -0.05$	$d = 0.05$	$d = 0.25$	$d = 0.45$
ML	50	0.006602	0.014849	0.016605	0.015882	0.013167	0.006792
Whittle		0.010869	0.018500	0.022051	0.022218	0.021602	0.013918
MML		0.011849	0.018520	0.019826	0.019632	0.017562	0.002942
ML	500	0.000926	0.001224	0.001196	0.001179	0.001129	0.000773
Whittle		0.001141	0.001282	0.001281	0.001284	0.001296	0.001057
MML		0.001011	0.001257	0.001244	0.001239	0.001231	0.000884
ML	5000	0.0001245	0.0001241	0.0001238	0.0001236	0.0001232	0.0001155
Whittle		0.0001342	0.0001251	0.0001256	0.0001260	0.0001274	0.0001342
MML		0.0001250	0.0001248	0.0001249	0.0001250	0.0001254	0.0001271

The results in Table 4 show that the ML estimator significantly outperformed the Whittle estimator, by more than 20% in some cases. The larger the absolute value of d and the smaller the sample size, the stronger the improvement. The MML estimator also outperformed the Whittle estimator.

In the above results, it was assumed that the mean of the time series is known to be zero. In practice, the mean will typically be unknown, and allowances must be made for this problem. Since the Whittle objective function omits zero frequency, it is invariant to the mean, and therefore needs no further adjustment. The ML objective function, however, is not invariant to the mean, and the traditional adjustment is to replace the quadratic form in (10) by $(x - \overline{x})'\Sigma_{n,\theta}(x - \overline{x})$ where $\overline{x}$ denotes the sample mean. It was found in Cheung and Diebold (1994) that the performance of the ML estimator with mean adjustment performs noticeably worse than the non-adjusted version. This is presumably caused in part by the slow convergence of the sample mean when $d \in (-1/2, 1/2) \setminus \{0\}$. For d in this range, $\text{var}(\overline{x}) \sim An^{2d-1}$, $A > 0$.

We re-ran the simulations described above, this time working with mean-adjusted data. Here, we adjusted the quadratic form as described above for both the ML and MML methods. We also included results on an approximate ML algorithm of Haslett and Raftery (1989) (H&R) as implemented in the Splus command arima.fracdiff, for which the program automatically removes the sample mean. The results are summarized in Table 5. We find, as did Cheung and Diebold (1994), that when the mean is treated as unknown and the sample size is small, the ML estimator does not outperform the Whittle estimator. However, we see that when $n = 5000$, ML does perform better than the Whittle estimator when $|d|$ is large. Furthermore, the MML estimator outperforms the Whittle estimator for all sample sizes when $|d|$ is large. The $H\&R$ estimator performs worse than the ML estimator in most cases.

Table 5: MSE for Estimators of d for $ARFIMA(0,d,0)$ Process When Mean is Unknown

Estimator	Sample Size	$d=-0.45$	$d=-0.25$	$d=-0.05$	$d=0.05$	$d=0.25$	$d=0.45$
ML	50	0.005184	0.017933	0.025730	0.026979	0.028055	0.030072
Whittle		0.010869	0.018500	0.022051	0.022218	0.021602	0.013918
MML		0.009463	0.018602	0.022845	0.023212	0.021955	0.011642
H&R		0.004918	0.020805	0.028117	0.028466	0.028306	0.030101
ML	500	0.000912	0.001327	0.001356	0.001360	0.001361	0.001297
Whittle		0.001141	0.001282	0.001281	0.001284	0.001296	0.001057
MML		0.000984	0.001291	0.001295	0.001292	0.001282	0.001027
H&R		0.001045	0.001386	0.001374	0.001366	0.001351	0.001276
ML	5000	0.0001243	0.0001261	0.0001267	0.0001269	0.0001271	0.0001244
Whittle		0.0001342	0.0001251	0.0001256	0.0001260	0.0001274	0.0001342
MML		0.0001248	0.0001257	0.0001261	0.0001261	0.0001263	0.0001262
H&R		0.0001539	0.0001250	0.0001266	0.0001271	0.0001297	0.0001326

7 Appendix A

Here, we explain why $\kappa[\Sigma_n(f)] = O(n^{2|d|})$, when f satisfies Assumption 1. For simplicity, we assume here that if $d \in (0, 1/2)$ the autocovariance sequence satisfies $c_r \sim Cr^{2d-1}$ as $r \to \infty$ where $C > 0$. This holds for all long-memory models in widespread use, though it is not equivalent to our Assumption 1. We refer to Robinson (1995) for further discussion on this point. By Brockwell and Davis (1991, Proposition 4.5.3, p. 137), $2\pi \inf_{\omega\in[-\pi,\pi]} f(\omega) \leq \lambda_{\min}[\Sigma_n(f)]$ and $\lambda_{\max}[\Sigma_n(f)] \leq 2\pi \sup_{\omega\in[-\pi,\pi]} f(\omega)$.

If $d \in (0, \frac{1}{2})$, Assumption 1 implies that f is bounded below by a positive constant so $\lambda_{\min}[\Sigma_n(f)]$ is bounded below by a positive constant, uniformly in n, and by Grenander & Szegö (1984), since $\Sigma_n(f)$ is Hermitian, $\lambda_{\max}[\Sigma_n(f)]$ is bounded by the maximum absolute row sum, which is in turn bounded by

$$2\left(|c_0| + |c_1| + \ldots + |c_{n-1}|\right) \sim C\sum_{r=0}^{n-1} r^{2d-1} \sim Cn^{2d} \ ,$$

so that $\kappa[\Sigma_n(f)] = O(n^{2|d|})$.

If $d \in (-\frac{1}{2}, 0)$, Assumption 1 implies that f is bounded above, so $\lambda_{\max}[\Sigma_n(f)]$ is bounded above, uniformly in n, and

$$\lambda_{\min}\left[\Sigma_n(f)\right] \geq \lambda_{\min}\left[\Sigma_n(f)\Sigma_n(f^{-1})\right] \lambda_{\min}\left[\Sigma_n^{-1}(f^{-1})\right] \geq 4\pi^2 \lambda_{\max}^{-1}\left[\Sigma_n(f^{-1})\right] \geq Cn^{-2|d|} \ ,$$

by the proof of Lemma 2 and the discussion above, so that once again $\kappa[\Sigma_n(f)] = O(n^{2|d|})$.

8 Appendix B: Proofs

Let $\widetilde{J}_k$ be the normalised DFT at frequency ω_k, $\widetilde{J}_k = J_{X,k}E^{-\frac{1}{2}}(I_{X,k})$ and $\widetilde{J} = \left(\widetilde{J}_0, \ldots, \widetilde{J}_{n-1}\right)$.

Proof of Lemma 1. Note that $M(f) = E\left(\widetilde{J}^* \widetilde{J}\right)$. First we write

$$M(f) = I_n + \Delta M,$$

where I_n is an $n \times n$ identity matrix. Since

$$\lambda_{\max}\left(M(f)\right) \leq 1 + \lambda_{\max}\left(\Delta M\right), \tag{11}$$

we will derive an upper bound for $\lambda_{\max}\left(\Delta M\right)$. We will use the following inequality:

$$\lambda_{\max}\left(\Delta M\right) \leq \max_k \left|\lambda_k\left(\Delta M\right)\right| = \lambda^{\frac{1}{2}}_{\max}\left(\Delta M \Delta M^*\right) \leq \operatorname{trace}^{\frac{1}{2}}\left(\Delta M \Delta M^*\right). \tag{12}$$

Let Δm_{jk} denote the (j,k)th entry of ΔM. Then

$$\operatorname{trace}\left(\Delta M \Delta M^*\right) = \sum_{j \neq k} \left|\Delta m_{jk}\right|^2 = \sum_{j \neq k} \left|E\left(\widetilde{J}^*_j \widetilde{J}_k\right)\right|^2$$

since $\Delta m_{jj} = 0$ for all j.

Let

$$\begin{aligned}
\phi_{jk} &= \Delta m_{jk} + \Delta m_{j,n-k} + \Delta m_{n-j,k} + \Delta m_{n-j,n-k} \\
&= E\left(\widetilde{J}^*_j \widetilde{J}_k\right) + E\left(\widetilde{J}^*_j \widetilde{J}^*_k\right) + E\left(\widetilde{J}^*_j \widetilde{J}_k\right) + E\left(\widetilde{J}_j \widetilde{J}_k\right),
\end{aligned}$$

$j, k = 0, \ldots, \left[\frac{n}{2}\right]$. Note that $E\left(J_0 J_{n/2}\right) = 0$ when n is even. By Lemma 9 (and its proof) of Moulines and Soulier (1999) and our Lemmas 4 and 5, we have

$$\begin{aligned}
\operatorname{trace}\left(\Delta M \Delta M^*\right) &= O\left[\sum_{k=1}^{[n/2]} \left(|\phi_{0k}| + |\phi_{n/2,k}| 1_{\{n \text{ is even}\}}\right) + \sum_{j=1}^{[n/2]} \sum_{k=1}^{j-1} |\phi_{jk}|\right] \\
&= O\left[\sum_{k=1}^{[n/2]} \left(k^{2|d|-2} + n^{-1}\right) + \sum_{j=1}^{[n/2]} \sum_{k=1}^{j-1} k^{-2|d|} j^{2|d|-2} \log^2 j\right] = O\left(\log^3 n\right).
\end{aligned}$$

By (11), (12) and the above equation, we have $\lambda_{\max}\left[M(f)\right] = O\left(\log^{\frac{3}{2}} n\right)$. □

Proof of Lemma 2. We write

$$\widetilde{C}^{-1}_n(f) \Sigma_n(f) = \left[\widetilde{C}_n\left(f^{-1}\right) \widetilde{C}_n(f)\right]^{-1} \left[\widetilde{C}_n\left(f^{-1}\right) \Sigma^{-1}_n\left(f^{-1}\right)\right] \left[\Sigma_n\left(f^{-1}\right) \Sigma_n(f)\right].$$

Since $\lambda_{\min}(ABC) \geq \lambda_{\min}(A) \lambda_{\min}(B) \lambda_{\min}(C)$, it suffices to show that

$$\lambda_{\min}\left[\widetilde{C}_n\left(f^{-1}\right) \widetilde{C}_n(f)\right]^{-1} \geq C, \tag{13}$$

$$\lambda_{\min}\left[\widetilde{C}_n\left(f^{-1}\right) \Sigma^{-1}_n\left(f^{-1}\right)\right] \geq C \log^{-\frac{3}{2}} n \tag{14}$$

and

$$\lambda_{\min}\left[\Sigma_n\left(f^{-1}\right) \Sigma_n(f)\right] \geq C. \tag{15}$$

Since both $\widetilde{C}_n(f)$ and $\widetilde{C}_n\left(f^{-1}\right)$ are circulant matrices (see Equation (6)),

$$\lambda_{\min}\left[\widetilde{C}_n\left(f^{-1}\right)\widetilde{C}_n(f)\right]^{-1}=\lambda_{\max}^{-1}\left[\widetilde{C}_n\left(f^{-1}\right)\widetilde{C}_n(f)\right]=\left\{\max_k\lambda_k\left[\widetilde{C}_n\left(f^{-1}\right)\right]\lambda_k\left[\widetilde{C}_n(f)\right]\right\}^{-1}. \quad (16)$$

Let $\{Y_t\}_{t=0}^{n-1}$ be from a zero-mean process $\{Y_t\}_{t=-\infty}^{\infty}$ with spectral density $f^{-1}(\omega)$. Also let

$$\eta_{n,k}=\begin{cases}\omega_k^{-2d}, & k\neq 0\\ n^{-2d}, & k=0\end{cases},$$

then

$$\begin{aligned}\max_k\lambda_k\left[\widetilde{C}_n\left(f^{-1}\right)\right]\lambda_k\left[\widetilde{C}_n(f)\right]&=\max_{0\le k\le[n/2]}E(I_{X,k})E(I_{Y,k})=\max_{0\le k\le[n/2]}\frac{E(I_{X,k})}{\eta_{n,k}}\frac{E(I_{Y,k})}{\eta_{n,k}^{-1}}\\ &\le\max_{0\le k\le[n/2]}\frac{E(I_{X,k})}{\eta_{n,k}}\max_{0\le k\le[n/2]}\frac{E(I_{Y,k})}{\eta_{n,k}^{-1}}\le C,\end{aligned}$$

by (5) and Lemma 3. Combining this with (16), we obtain (13).

By (7) and Lemma 2,

$$\lambda_{\min}\left[\widetilde{C}_n\left(f^{-1}\right)\Sigma_n^{-1}\left(f^{-1}\right)\right]=\lambda_{\max}^{-1}\left[\Sigma_n\left(f^{-1}\right)\widetilde{C}_n^{-1}\left(f^{-1}\right)\right]=\lambda_{\max}^{-1}M\left(f^{-1}\right)\ge C\log^{-3/2}n.$$

We have shown (14). We next prove (15). Let

$$u_n(\omega)=f^{\frac{1}{2}}(\omega)\left(1,e^{i\omega},\ldots,e^{i(n-1)\omega}\right)^*,\qquad v_n(\omega)=f^{-\frac{1}{2}}(\omega)\left(1,e^{i\omega},\ldots,e^{i(n-1)\omega}\right)^*,$$

then

$$\Sigma_n(f)=\int_{-\pi}^{\pi}u_n(\omega)u_n^*(\omega)\,d\omega,\qquad \Sigma_n\left(f^{-1}\right)=\int_{-\pi}^{\pi}v_n(\omega)v_n^*(\omega)\,d\omega.$$

Since $\int_{-\pi}^{\pi}u_n(\omega)v_n^*(\omega)\,d\omega=2\pi I_n$,

$$\lambda_{\min}\left[\Sigma_n\left(f^{-1}\right)\Sigma_n(f)\right]-4\pi^2\ge\lambda_{\min}\left[\Sigma_n\left(f^{-1}\right)\Sigma_n(f)-4\pi^2I_n\right]\ge 0,$$

by Lemma 6. □

Lemma 3. *There exist finite constants C_1 and C_2 such that,*

$$C_1\omega_k^{-2d}\le E(I_{X,k})\le C_2\omega_k^{-2d},$$

for $k=1,\ldots,\left[\frac{n}{2}\right]$ and

$$C_1\le n^{-2d}E(I_{X,0})\le C_2.$$

Proof. For $E(I_{X,0})$, the inequality follows from Theorem 2 of Deo and Hurvich (1998). For $k=1,\ldots,\left[\frac{n-1}{2}\right]$, the inequality follows from Lemma 6 of Moulines and Soulier (1999). When n is even, the lower bound for $E\left(I_{X,n/2}\right)$ follows from Lemma 8 of Moulines and Soulier (1999). We now derive an

upper bound for $E\left(I_{X,n/2}\right)$. Let δ be a positive constant, $0<\delta<\pi$,

$$\begin{aligned}
E\left(I_{X,n/2}\right) &= 2\int_0^{\pi} K_n(\omega) f(\pi-\omega)\, d\omega \\
&= 2\int_0^{\delta} K_n(\omega) f(\pi-\omega)\, d\omega + 2\int_{\delta}^{\pi} K_n(\omega) f(\pi-\omega)\, d\omega \\
&\leq 2 \max_{\omega\in(\pi-\delta,\pi)} f(\omega) \int_0^{\delta} K_n(\omega)\, d\omega + \frac{1}{2\pi n \sin^2(\delta)} \int_0^{\pi-\delta} f(\omega)\, d\omega \\
&\leq C,
\end{aligned}$$

since $\max_{\omega\in(\pi-\delta,\pi)} f(\omega)\leq C$, $\int_0^{\delta} K_n(\omega)\, d\omega < \int_0^{\pi} K_n(\omega)\, d\omega = 1/2$, $\max_{\omega\in(\delta,\pi)} K_n(\omega) = (4\pi n)^{-1}\sin^{-2}(\delta)$ (see p. 89 of Zygmund 1977) and $f(\omega)$ is integrable.□

Lemma 4. *For* $k=1,\ldots,\left[\frac{n-1}{2}\right]$,

$$E\left(\widetilde{J}_{X,0}\widetilde{J}_{X,k}\right) = \begin{cases} O\left(k^{d-1}\right), & d>0 \\ O\left(n^{2d}k^{d-1}\right), & d<0 \end{cases}.$$

Proof. By Lemma 3, we have

$$n^{-2d}E^{\frac{1}{2}}(I_{X,0})\, E^{\frac{1}{2}}(I_{X,k}) \geq Ck^{-d}. \tag{17}$$

Now

$$2\pi E(J_{X,0}J_{X,k}) = \frac{1}{n}\sum_{s=0}^{n-1}\sum_{t=0}^{n-1} E(X_sX_t)\, e^{i\omega_k s} = \frac{1}{n}\sum_{s=0}^{n-1}\sum_{t=0}^{n-1} c_{s-t}e^{i\omega_k s} = \frac{1}{n}c_0\sum_{s=0}^{n-1} e^{i\omega_k s} + \frac{1}{n}\sum_{s\neq t}^{n-1} c_{s-t}e^{i\omega_k s}.$$

The first term of RHS is zero. Letting $u=s-t$, the second term is

$$\frac{1}{n}\sum_{u=1}^{n-1} c_u\left(\sum_{v=0}^{n-1-u} e^{i\omega_k v} + e^{in\omega_k v}\sum_{v=1}^{n-1-u} e^{-i\omega_k v}\right) = O\left(\frac{1}{n}\sum_{u=1}^{n-1} u^{2d-1}\left|\omega_k\right|^{-1}\right) = \begin{cases} O\left(n^{2d}k^{-1}\right), & d>0 \\ O\left(k^{-1}\right), & d<0 \end{cases},$$

since $\sum_{v=0}^{u} e^{ixv} = O\left(x^{-1}\right)$ uniformly in u for $0<x<\pi$ by page 2 of Zygmund (1977). The lemma follows from (17) and the above equation.□

Lemma 5. *If* n *is even,* $E\left(\widetilde{J}_{X,n/2}\widetilde{J}_{X,k}\right) = O\left(n^{-1/2}\right), \quad k=1,\ldots,\left[\frac{n-1}{2}\right]$.

Proof. By Lemma 3, we have

$$E^{\frac{1}{2}}\left(I_{X,n/2}\right) E^{\frac{1}{2}}(I_{X,k}) \geq C\omega_k^{-d}. \tag{18}$$

Now

$$\left|D_n(\pi-\omega)\right| = \left|\frac{1}{(2\pi n)^{\frac{1}{2}}}\sum_{t=0}^{n-1} e^{i(\pi-\omega)t}\right| = \left|\frac{\left(1-e^{-i\omega}\right)}{(2\pi n)^{\frac{1}{2}}}\sum_{t=0}^{\frac{n}{2}-1} e^{-2i\omega t}\right| = \left|\frac{\left(1-e^{-i\omega}\right)}{(2\pi n)^{\frac{1}{2}}}\frac{\sin(n\omega/2)}{\sin\omega}\right| \leq Cn^{-\frac{1}{2}},$$

since $\left|1-e^{-i\omega}\right| \leq C\left|\omega\right|$ for $-\pi \leq \omega \leq \pi$. Hence

$$\begin{aligned}
\left|E\left(J_{X,n/2}J_{X,k}\right)\right| &= \left|\int_{-\pi}^{\pi} D_n\left(\pi-\omega\right) D\left(\omega_k-\omega\right) f\left(\omega\right) d\omega\right| \\
&\leq \int_{-\pi}^{\pi} \left|D_n\left(\pi-\omega\right)\right| \left|D\left(\omega_k-\omega\right)\right| f\left(\omega\right) d\omega \\
&\leq Cn^{-\frac{1}{2}} \int_{-\pi}^{\pi} \left|D_n\left(\omega_k-\omega\right)\right| f\left(\omega\right) d\omega \\
&\leq Cn^{-\frac{1}{2}} \left(\int_{-\pi}^{\pi} K_n\left(\omega_k-\omega\right) f\left(\omega\right) d\omega \int_{-\pi}^{\pi} f\left(\omega\right) d\omega\right)^{\frac{1}{2}} \\
&= Cn^{-\frac{1}{2}} c_0^{\frac{1}{2}} E^{\frac{1}{2}}\left(I_{X,k}\right) \\
&\leq Cn^{-\frac{1}{2}} \omega_k^{-d},
\end{aligned}$$

by the Cauchy Schwartz inequality and Lemma 3. The lemma follows from (18) and the above equation.□

The next lemma is a version of Cauchy-Schwartz inequality.

Lemma 6. *Let $\alpha\left(x\right)$ and $\beta\left(x\right)$ be two $n \times m$ matrices, $m \leq n$. Let*

$$U = \int \alpha\left(x\right)\alpha^*\left(x\right) dx, \quad V = \int \beta\left(x\right)\beta^*\left(x\right) dx, \quad G = \int \alpha\left(x\right)\beta^*\left(x\right) dx.$$

If U is positive definite, then

$$\lambda_{\min}\left(UV - GG^*\right) \geq 0 \ .$$

Proof. First note that

$$\begin{aligned}
\lambda_{\min}\left(V - U^{-1}GG^*\right) &= \lambda_{\min}\left(V - G^*U^{-1}G + G^*U^{-1}G - U^{-1}GG^*\right) \\
&\geq \lambda_{\min}\left(V - G^*U^{-1}G\right) + \lambda_{\min}\left(G^*U^{-1}G - U^{-1}GG^*\right) \ . \qquad (19)
\end{aligned}$$

Let A be an $n \times n$ matrix. Then

$$\int \left(A\alpha\left(x\right) + \beta\left(x\right)\right)\left(A\alpha\left(x\right) + \beta\left(x\right)\right)^* dx = AUA^* + AG + G^*A^* + V$$

is nonnegative definite. Plugging $A = -G^*U^{-1}$ into the above equation, we have the nonnegativity of $V - G^*U^{-1}G$, that is

$$\lambda_{\min}\left(V - G^*U^{-1}G\right) \geq 0 \ . \qquad (20)$$

Let a be the eigenvector corresponding to $\lambda_{\min}\left(U^{-1}GG^*\right)$. We have

$$\begin{aligned}
\lambda_{\min}\left(G^*U^{-1}G - U^{-1}GG^*\right) &\geq a^*\left(G^*U^{-1}G - U^{-1}GG^*\right)a \\
&\geq a^*G^*U^{-1}Ga - a^*U^{-1}GG^*a \\
&\geq \lambda_{\min}\left(G^*U^{-1}G\right) - \lambda_{\min}\left(U^{-1}GG^*\right) \\
&= 0 \ ,
\end{aligned}$$

since $G^*U^{-1}G$ and $U^{-1}GG^*$ have the same set of eigenvalues. Similarly, $\lambda_{\min}\left(U^{-1}GG^* - G^*U^{-1}\right) \geq 0$. Hence $\lambda_{\min}\left(G^*U^{-1}G - U^{-1}GG^*\right) = 0$. Combining this with (19) and (20), we obtain

$$\lambda_{\min}\left(V - U^{-1}GG^*\right) \geq 0 \ . \qquad (21)$$

Now

$$\lambda_{\min}\left(UV - GG^*\right) = \lambda_{\min}\left[U\left(V - U^{-1}GG^*\right)\right] \geq \lambda_{\min}(U)\lambda_{\min}\left(V - U^{-1}GG^*\right) \geq 0 \ ,$$

by the positive definiteness assumption on U and (20).□

9 Appendix C: Implementation of CG and PCG Algorithms

Here, we present some details on the CG and PCG algorithms. For a more extensive discussion, see Golub and Van Loan (1996) and Shewchuk (1994). Much of our description here applies for any $n \times n$ symmetric positive definite matrix Σ_n.

Suppose we wish to solve the system $\Sigma_n x = b$. The conjugate gradient method proceeds by generating successive approximations $x_{(i)}$ to the solution x, residuals $r_{(i)}$ corresponding to the $x_{(i)}$, and search directions $d_{(i)}$ used in updating the approximate solutions and residuals. The $x_{(i)}$ are updated at each iteration by a multiple α_i of the search direction vector $d_{(i)}$: $x_{(i+1)} = x_{(i)} + \alpha_i d_{(i)}$.

The **Conjugate Gradient Algorithm** is summarized by the following iterative procedure:

$$d_{(0)} = r_{(0)} = b - \Sigma_n x_{(0)} \ , \ \ x_{(0)} = (0, \ldots, 0)' \ ,$$

$$\alpha_{(i)} = \frac{{r_{(i)}}' r_{(i)}}{{d_{(i)}}' \Sigma_n d_{(i)}},$$

$$x_{(i+1)} = x_{(i)} + \alpha_{(i)} d_{(i)},$$

$$r_{(i+1)} = r_{(i)} - \alpha_{(i)} \Sigma_n d_{(i)},$$

$$\beta_{(i+1)} = \frac{{r_{(i+1)}}' r_{(i+1)}}{{r_{(i)}}' r_{(i)}},$$

$$d_{(i+1)} = r_{(i+1)} + \beta_{(i+1)} d_{(i)}.$$

For the Conjugate Gradient method, the error can be bounded in terms of the condition number $\kappa_n := \kappa(\Sigma_n)$. It can be shown (Golub & Van Loan 1996) that

$$||x_{(i)} - x||_{\Sigma_n} \leq 2\alpha^i ||x_{(0)} - x||_{\Sigma_n}$$

where $\alpha = \frac{\sqrt{\kappa_n}-1}{\sqrt{\kappa_n}+1}$ and $||y||_{\Sigma_n} = \sqrt{y'\Sigma_n y}$. The number of iterations j_ϵ required to reach a relative reduction of ϵ in the error is proportional to $\sqrt{\kappa_n}$ since

$$j_\epsilon \ = \ \frac{\log(\epsilon/2)}{\log(\alpha)} = \frac{\log(\epsilon/2)}{\log(1 - \frac{2}{\sqrt{\kappa_n}+1})} \sim \frac{\log(\epsilon/2)}{-\frac{2}{\sqrt{\kappa_n}+1}} \propto \sqrt{\kappa_n}$$

assuming $\kappa_n \to \infty$ as $n \to \infty$ and $\epsilon < 1$. Note that if κ_n remains bounded as n increases, then superlinear convergence (i.e. $O(1)$ iterations) is achieved.

The Preconditioned Conjugate Gradient algorithm essentially applies the conjugate gradient method to the preconditioned system $C^{-1}\Sigma_n x = C^{-1}b$, where C is symmetric and positive definite. The use of a preconditioner C will accelerate the convergence if $\kappa(C^{-1}\Sigma_n) < \kappa(\Sigma_n)$.

The PCG algorithm can be implemented as follows: we first decompose C as $C = EE'$ where E is positive definite. Then the system $\Sigma_n x = b$ can be transformed into the problem

$$(E^{-1}\Sigma_n E'^{-1})\hat{x} = E^{-1}b,$$

where $\hat{x} = E'x$. We solve first for $\hat{x}$ using the CG algorithm, replacing the Σ_n in the iterations described above with $E^{-1}\Sigma_n E'^{-1}$ and replacing b with $E^{-1}b$. Then $x = E'^{-1}\hat{x}$.

Suppose now that Σ_n is a positive definite, symmetric Toeplitz matrix, and that the preconditioner C is a circulant matrix $C = C_n(g) = V_n \Lambda_g V_n^*$ where g is a positive continuous function on $[0, 2\pi]$ symmetric around π, and $\Lambda_g = diag\{g(\omega_0), \ldots, g(\omega_{n-1})\}$. Note that $C_n^{\frac{1}{2}}(g) = V_n \Lambda_g^{\frac{1}{2}} V_n^*$ is also circulant. It is important to note that, given its eigenvalues $g(\omega_0), \ldots, g(\omega_{n-1})$, the circulant preconditioner $C_n(g)$ can be multiplied by a vector using the FFT in $O(n \log n)$ operations, since the m'th element of $C_n(g)y$ is

$$\begin{aligned}(C_n(g)y)_m &= (V_n \Lambda_g V_n^* y)_m \\ &= \frac{1}{n}\sum_{j=0}^{n-1} g(\omega_j)\exp(i\omega_m j)\sum_{k=0}^{n-1} y_k \exp(-i\omega_j k)\end{aligned}$$

for any $n \times 1$ vector y.

We are now ready to discuss the computational cost of implementing the PCG algorithm using T. Chan's circulant preconditioner $\widetilde{C}_n$ for solving Toeplitz systems. The eigenvalues of $\widetilde{C}_n$ can be computed using the FFT in $O(n \log n)$ operations in view of (4). For each iteration of the PCG algorithm, the most costly computations are to evaluate $\widetilde{C}_n^{-\frac{1}{2}} y$ or $\Sigma_n y$ for some vector y. Only $O(n \log n)$ operations are necessary for computing $\widetilde{C}_n^{-\frac{1}{2}} y$ using the FFT since $\widetilde{C}_n^{-\frac{1}{2}}$ is a circulant. To compute $\Sigma_n y$, we use a technique called circulant embedding, in which Σ_n is embedded into a $2n$-by-$2n$ circulant matrix, i.e.,

$$\begin{bmatrix} \Sigma_n & \times \\ \times & \Sigma_n \end{bmatrix} \begin{bmatrix} y \\ 0 \end{bmatrix} = \begin{bmatrix} \Sigma_n y \\ \dagger \end{bmatrix} . \tag{22}$$

For example, the first row of the $2n \times 2n$ circulant matrix (call it A) on the lefthand side of (22) is $\gamma = (c_0, c_1, \ldots, c_{n-1}, c_0, c_{n-1}, c_{n-2}, \ldots, c_1)$ where $(c_0, c_1, \ldots, c_{n-1})$ is the first row of Σ_n. The j'th eigenvalue of A is $\sum_{k=0}^{2n-1} \gamma_k \exp(i\omega_j k)$ where γ_k represents k'th element of γ. Therefore, the eigenvalues of A can be obtained using the FFT and the multiplication in the lefthand side of (22) can be carried out in $O(2n \log 2n)$ operations.

It follows that the cost per iteration in PCG is $O(n \log n)$ operations, and the total cost of the algorithm is $O\left[n \log n \sqrt{\kappa(\widetilde{C}_n^{-1}\Sigma_n)}\right]$.

10 Appendix D: Accurate Approximation For Determinant of Covariance Matrix

We specialize to our situation the result of Böttcher and Silbermann (1999 Theorem 5.47, Page 177), with the substitutions $\alpha_1 = -d$, $t_1 = 1$, $\beta_1 = 0$, $N = 1$ and $b = f^*$, noting a typographical error

in their Equation (5.80): Their $G(b)$ should be $G(b)^n$. Suppose that f satisfies Assumption 1, with $d \in (-1/2, 1/2)$. Define

$$\alpha_k = \frac{1}{2\pi} \int_{-\pi}^{\pi} \log f^*(\omega) \cos(k\omega) d\omega \ , \ \ k = 0, 1, 2, \ldots \ .$$

Then

$$|\Sigma_n(f)| \sim (2\pi)^n e^{n\alpha_0} n^{d^2} \tilde{E} \ , \tag{23}$$

as $n \to \infty$, where

$$\tilde{E} = \exp\left(\sum_{k=1}^{\infty} k\, \alpha_k^2 + 2d \sum_{k=1}^{\infty} \alpha_k \right) G^2(1-d)/G(1-2d) \tag{24}$$

and G is the Barnes G-Function (see Weisstein 2004).

For $ARFIMA$ models or $FEXP$ models, $\alpha_0 = \log[\sigma_\eta^2/(2\pi)]$ where σ_η^2 is the innovation variance. For the $ARFIMA(0, d, 0)$ model, $\alpha_k = 0$ for all $k > 0$. For the $ARFIMA(1, d, 0)$ model, f^* is the spectral density of an $AR(1)$ process, so the α_k decay exponentially fast to zero. For the particular case considered in this paper, we evaluated the α_k by numerical integration and found that the sums in (24) had effectively converged after the first 20 terms.

References

[1] Ammar G. S. and W.B. Gragg (1988), Superfast solution to real positive definite Toeplitz systems, *SIAM Journal of Matrix Analysis and Applications*, **9**, 61–76.

[2] Axelsson, O. and V. Barker (1984), *Finite Element Solution of Boundary Value Problems: Theory and Computation.* New York: Academic Press.

[3] Bertelli, S. and M. Caporin (2002). A note on calculating autocovariances of long-memory processes. *J. Time Ser. Anal.*, **23**, 503–508.

[4] Böttcher, A. and B. Silbermann (1999), *Introduction to Large Truncated Toeplitz Matrices*, New York: Springer-Verlag.

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

[6] Brockwell, P. J. and R. A. Davis (1991), *Time Series: Theory and Methods*, Second Ed. New York: Springer-Verlag.

[7] Bunch, J. R. (1985). Stability of Methods for Solving Toeplitz Systems of Equations, *SIAM J. Sci. Statist. Comput.*, **6**, 349–364.

[8] Chan, R. (1989), Circulant preconditioners for Hermitian Toeplitz systems, *SIAM J. Matrix Anal. Appl.*, **10**, 542–550.

[9] Chan, R. and M. Ng (1996), Conjugate Gradient Methods For Toeplitz Systems, *SIAM Review*, **38**, 427–482.

[10] Chan, R., A. M. Yip and M. Ng (2000), The Best Circulant Preconditioners For Hermitian Toeplitz Systems, *SIAM J. Numer. Anal.*, **38**, 876–896

[11] Chan, R. and M. Yeung (1992), Circulant preconditioners for Toeplitz matrices with positive continuous generating functions, *Math. Comp.*, **58**, 233–240.

[12] Chan, T. (1988), An optimal circulant preconditioner for Toeplitz systems, *SIAM J. Sci. Stat. Comput.*, **9**, 766–771.

[13] Cheung, Y.W. and F. Diebold (1994), On Maximum Likelihood Estimation of the Differencing Parameter of Fractionally-Integrated Noise With Unknown Mean, *J. Econometrics*, **62**, 301–316.

[14] Cooley, J. W. and Tukey, J. W. (1965), An Algorithm For the Machine Calculation of Complex Fourier Series, *Math. Comp.*, **19**, 297–301.

[15] Dahlhaus, R. (1989), Efficient Parameter Estimation for Self-Similar Processes, *Ann. Statist,* **17**, 1749–1766.

[16] Deo, R., and C. Hurvich (1998), Linear Trend with Fractionally Integrated Errors, *J. Time Ser. Anal.*, **19**, 379–397.

[17] Deo, R., C. Hurvich, Y. Lu (2003), Forecasting Realized Volatility using a Long Memory Stochastic Volatility Model: Estimation, Prediction and Seasonal Adjustment. Working Paper, Stern School of Business, New York University.

[18] Gentleman, W. M. and Sande, G. (1966) Fast Fourier Transforms - For Fun and Profit, *Proc. AFIPS*, **29**, 563–578.

[19] Golub, G. H. and Van Loan, C. F. (1996) Matrix Computations, 3'rd Ed. *Baltimore, MD: Johns Hopkins University Press.*

[20] Grenander, U. and Szegö, G. (1984), *Toeplitz Forms and Their Applications*, 2'nd Ed., New York: Chelsea.

[21] Harvey, A. (1998), Long Memory in Stochastic Volatility, in *Forecasting Volatility in Financial Markets*. J Knight and S Satchell (Eds.), 307–320. Oxford: Butterworth-Heineman.

[22] Haslett, J. and A.E. Raftery (1989) Space-Time Modelling With Long-Memory Dependence: Assessing Ireland's Wind Power Resource, *Applied Statistics*, **38**, 1–50.

[23] Hurvich, C.M. (2002) Multistep Forecasting of Long Memory Series Using Fractional Exponential Models, *International Journal of Forecasting*, **18**, 167–179.

[24] Kravanja, P. and M. V. Barel (2000) Coupled Vandermonde matrices and the superfast computation of Toeplitz determinants, *Numerical Algorithms*, **24**, 99–116.

[25] Levinson, N. (1946), The Wiener RMS (root mean square) error criterion in filter design and prediction, *J. Math. Phys.*, **25**, 261–178.

[26] Moulines, E. and P. Soulier (1999), Broadband Log-Periodogram Regression of Time Series With Long-Range Dependence, *Ann. Statist.*, **27**, 1415–1439.

[27] Percival, D. B. and A. T. Walden (1993), *Spectral Analysis for Physical Applications*, New York: Cambridge University Press.

[28] Priestley, M. B. (1981) *Spectral Analysis and Time Series*, New York: Academic Press.

[29] Robinson, P. M. (1995) Log-Periodogram Regression of Time Series with Long Range Dependence, *Ann. Statist.*, **23**, 1048–1072.

[30] Shewchuk, J. R. (1994), An introduction to the conjugate gradient method without agonizing pain, School of Computer Science, Carnegie Mellon University.

[31] Sowell, F. (1992), Maximum Likelihood Estimation of Stationary Univariate Fractionally Integrated Time Series Models, *J. Econometrics*, **53**, 165–188.

[32] Strang, G. (1986), A proposal for Toeplitz matrix calculations, Stud. Appl. Math., **74**, 171–176.

[33] Weisstein, E. W. (2004), "Barnes G-Function." From *MathWorld*–A Wolfram Web Resource. http://mathworld.wolfram.com/BarnesG–Function.html

[34] Whittle, P. (1953), Estimation and information in stationary time series, *Ark. Mat.*, **2**, 423–434.

[35] Zygmund, A. (1977) *Trigonometric Series*, Second Ed. New York: Cambridge University Press.