

Computational Aspects of Maximum Likelihood Estimation of Autoregressive Fractionally Integrated Moving Average models

Jurgen A. Doornik*

Nuffield College, University of Oxford, Oxford OX1 1NF, UK

Marius Ooms

*Department of Economics, Free University of Amsterdam
1081 HV Amsterdam, The Netherlands*

November 29, 2001

Abstract

We discuss computational aspects of likelihood-based estimation of univariate ARFIMA(p, d, q) models. We show how efficient computation and simulation is feasible, even for large samples. We also discuss the implementation of analytical bias corrections.

Keywords: Long memory, Bias, Modified profile likelihood, Restricted maximum likelihood estimator, Time-series regression model likelihood.

JEL classification: C22, C63

1 Introduction

The fractionally integrated ARMA model, denoted ARFIMA(p, d, q), has become increasingly popular to describe time series that exhibit long memory. In many cases, it provides a more parsimonious description of economic time-series data than the ARMA model.

It is important that exact maximum likelihood estimation of the Arfima model is readily available, if only to serve as a benchmark for other estimators. This problem has essentially been solved by Hosking (1981), and Sowell (1987). Unfortunately, there were some problems that remained unresolved. Initial implementations did not take the structure of the variance matrix into account and were not suitable for extensions with regression parameters. Consequently, they suffered from numerical instability, and could only be used on small data sets. This led to frequent remarks in the literature that it is very difficult to estimate Arfima models by exact maximum likelihood, a misconception that persists to date. For example, Bollerslev and Jubinski (1999, p. 12) say: ‘even for simple low-order fractionally integrated models, exact maximum likelihood estimation is extremely time consuming and would not be practical for the sample sizes (8000) and number of individual stocks (100) analyzed here’. The Arfima package by Doornik and Ooms (1999) showed that exact MLE is possible for long time series. Ooms and Doornik (1999) build on this by implementing a parametric bootstrap for Arfima models of UK and US inflation.

*Correspondence to: jurgen.doornik@nuffield.ox.ac.uk

The improvements that were made include:

- concentration of the log-likelihood with respect to scale and regression parameters;
- partial removal of the restriction that the autoregressive roots must be non-zero;
- exploiting the Toeplitz structure of the variance-covariance matrix;
- efficient implementation of the modified profile likelihood method and related bias reduction methods for ML estimators.

The organization of this paper is as follows. Section 2 reviews the Arfima model, and compares several methods to compute the autocovariances. In §3 we implement some improvements to the expressions for the autocorrelation functions, and consider issues of numerical stability. We discuss the evaluation of the log-likelihood in §4, where we also pay attention to analytical bias corrections. Data generation for Monte Carlo experiments is the topic of §5. Section 6 concludes.

2 Maximum likelihood estimation of the Arfima model

The Gaussian ARFIMA(p, d, q) model is written as

$$\Phi(L)(1-L)^d(y_t - \mu) = \Theta(L)\varepsilon_t \quad \varepsilon_t \sim \text{NID}[0, \sigma_\varepsilon^2]. \quad (1)$$

Where d is the fractional integration parameter, $\Phi(L) = 1 - \phi_1 L \dots - \phi_p L^p$ specifies the AR lag-polynomial, and $\Theta(L) = 1 + \theta_1 L \dots + \theta_q L^q$ the MA polynomial.

The autocovariance function of a stationary ARMA process with mean μ ,

$$\gamma_i = \text{E}[(y_t - \mu)(y_{t-i} - \mu)],$$

defines the variance matrix of the joint distribution of $y = (y_1, \dots, y_T)'$:

$$\text{V}[y] = \begin{pmatrix} \gamma_0 & \gamma_1 & \cdots & \gamma_{T-1} \\ \gamma_1 & \gamma_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \gamma_1 \\ \gamma_{T-1} & \cdots & \gamma_1 & \gamma_0 \end{pmatrix} = \Sigma, \quad (2)$$

which is a symmetric Toeplitz matrix, denoted by $\mathcal{T}[\gamma_0, \dots, \gamma_{T-1}]$. Under normality:

$$y \sim \text{N}_T(\mu, \Sigma), \quad (3)$$

and, combined with a procedure to compute the autocovariances in (2), the log-likelihood is (writing $z = y - \mu$):

$$\log L(d, \phi, \theta, \beta, \sigma_\varepsilon^2) = -\frac{T}{2} \log(2\pi) - \frac{1}{2} \log |\Sigma| - \frac{1}{2} z' \Sigma^{-1} z. \quad (4)$$

Additional regression parameters in μ are denoted by β , but can be ignored initially.

The autocorrelation function, γ_i/γ_0 , of a stationary ARMA process is discussed in many textbooks. Here, we frequently work with the autocovariances scaled by the error variance, $r_i = \gamma_i/\sigma_\varepsilon^2$:

$$R = \mathcal{T}[\gamma_0/\sigma_\varepsilon^2, \dots, \gamma_{T-1}/\sigma_\varepsilon^2].$$

There are two issues that need to be resolved to allow maximum likelihood estimation: evaluation of the autocovariance function to construct R , and handling this matrix which is of dimension $T \times T$. Regarding the former, the main focus here is on the method developed by Sowell (1992). We compare this to alternatives in the next section, prior to a discussion of the implementation details in §3.

2.1 Computing the autocovariances

Various techniques are available to evaluate the autocovariance function of a stationary ARFIMA process.

A conceptually very simple procedure is to compute the autocovariances from the MA representation. Because the lag polynomial on $z_t = y_t - \mu$ is invertible for $-1 < d < 0.5$, its MA representation is:

$$z_t = \Phi(L)^{-1}(1 - L)^{-d}\Theta(L)\varepsilon_t = \sum_{j=0}^{\infty} \psi_j^z \varepsilon_t,$$

with $\psi_0 = 1$. Based on this:

$$\gamma_k = \sum_{j=0}^{\infty} \psi_j^z \psi_{j+|k|}^z \sigma_\varepsilon^2. \quad (5)$$

Chan and Palma (1998) used this approach. The drawback is that, because ψ_j^z declines hyperbolically, many terms are needed for an accurate approximation.

Hosking (1981) expressed γ_k as a function of the autocovariances of the ARFIMA(0, d , 0) process $w_t = \{\Theta(L)\}^{-1}\Phi(L)z_t$ and the ARMA(p , q) process $u_t = (1 - L)^d z_t$:

$$\gamma_k = \sum_{j=-\infty}^{\infty} \gamma_j^u \gamma_{k-j}^w. \quad (6)$$

Both γ_j^w and γ_j^u are easily calculated using simple recursions, see e.g. Brockwell and Davis (1993, §3.3 and §13.2). Fast decay in the γ_j^u will help convergence, but when $\Phi(z)$ has roots close to the unit circle, it remains necessary to use many terms for accurate computation of γ_k . For pure ARFIMA(0, d , q) processes, only $2q + 1$ terms in (6) are required, which is preferable to using the MA representation (5).

Ravishanker and Ray (1997) use the MA coefficients ψ_i^u of $u_t = (1 - L)^d z_t$ instead:

$$\gamma_k = \sum_{j=0}^{\infty} \sum_{i=0}^{\infty} \psi_i^u \gamma_{k+i-j}^w \psi_j^u \sigma_\varepsilon^2, \quad (7)$$

which alleviates the slow convergence of (5) caused by a large d .

A seemingly simple alternative is to numerically integrate over the spectrum, see e.g. Harvey (1993, p. 229):

$$\gamma_k = \int_{-\pi}^{\pi} f_z(\omega) e^{i\omega k} d\omega, \quad (8)$$

where the spectrum of the ARFIMA-process, $f_z(\omega)$, is easily computed. However, numerical integration for each k does rapidly get prohibitively slow.

More specific methods improve speed and accuracy of the computation of γ_k : Hosking (1981) provided a closed form alternative to (6) for the ARFIMA(1, d , 0) case. Sowell (1992) succeeded in extending and numerically implementing the results of Hosking for the ARFIMA(p , d , q) case, using the assumption of the unicity of the roots ρ_j^{-1} , $j = 1, \dots, p$, of the AR polynomial $\Phi(z) = 0$. Each γ_k requires the computation of at least p hypergeometric function values, which is a slowly converging series for $|\rho_i| \rightarrow 1$. Sowell (1992) achieved a major speed-up of the algorithm by evaluating the hypergeometric functions recursively, see §3.

Table 1 compares the methods for an ARFIMA(1, $d = 0.45$, 1) model with $\phi = 0.8$ and $\theta = -0.5$. This is an example where the autocorrelations exhibit slow decay. First, we look at the accuracy of the

Table 1: Comparison of ACF computation methods, $d = 0.45, \phi = 0.8, \theta = -0.5$

	MA (5)	Hosking (6)	Integration (8)	γ_k^S , Sowell
γ_{31}/γ_0^S	0.12543 (64 terms) 0.17978 (128) 0.49223 (320 000)	0.74707 (64) 0.74771 (128)	0.74771	0.74771
timing	> 1000	15	250	1

Listed are the values of the scaled 31st autocovariance of methods (5), (6), (8), and Sowell's method. The numbers in parentheses are the number of terms used in each approximation. The final row lists the computational time relative to Sowell's method.

approximations. All methods involve infinite sums, which must be terminated at some point. The first row in the table lists the 31th autocovariance, standardized with respect to the variance from Sowell's method: γ_{31}/γ_0^S . The number in parentheses for methods (5) and (6) gives the number of terms used in the summation. The numerical integration for (8) was implemented using QuadPack function QAGS, see Piessens, de Donker-Kapenga, Überhuber, and Kahaner (1983); QAGS required more than 200 function evaluations to attain the reported precision. The reported timings are for computing 1024 autocovariances, and are relative to Sowell's method.

Table 1 shows that neither the MA representation nor numerical integration are of practical use. The MA representation requires an infeasibly large number of terms to attain any precision when there is even moderately slow decay of the MA terms. The method is also slow, despite using fast Fourier transforms to compute ψ_j^z and to evaluate the sum. Numerical integration is also too slow unless only a few autocovariances are required.

Sowell's method, on the other hand, is both fast and accurate. For Hosking's method we implemented the convolution using the fast Fourier transform, which made it a reasonably fast procedure. Its simplicity may occasionally be an argument for using it, e.g. in more complex models.

3 Autocovariance function

3.1 Introduction

An algorithm for the computation of the autocovariances of the ARFIMA process (1) is derived in Sowell (1992):

$$\gamma_i = \sigma_\varepsilon^2 \sum_{k=-q}^q \sum_{j=1}^p \psi_k \zeta_j C(d, p + k - i, \rho_j), \quad (9)$$

where ρ_1, \dots, ρ_p are the (possibly complex) roots of the AR polynomial, and

$$\psi_k = \sum_{s=|k|}^q \theta_s \theta_{s-|k|}, \quad \zeta_j^{-1} = \rho_j \left[\prod_{i=1}^p (1 - \rho_i \rho_j) \prod_{\substack{m=1 \\ m \neq j}}^p (\rho_j - \rho_m) \right], \quad (10)$$

where $\theta_0 = 1$. C is defined as:¹

$$C(d, h, \rho) = \frac{\Gamma(1-2d)}{[\Gamma(1-d)]^2} \frac{(d)_h}{(1-d)_h} \times [\rho^{2p} F(d+h, 1; 1-d+h; \rho) + F(d-h, 1; 1-d-h; \rho) - 1]. \quad (11)$$

Here Γ is the gamma function, ρ_j are the roots of the AR polynomial (assumed to be distinct), and $F(a, 1; c; \rho)$ is the hypergeometric function, see e.g. Abramowitz and Stegun (1970, Ch. 15):

$$F(a, b; c; \rho) = \sum_{i=0}^{\infty} \frac{(a)_i (b)_i \rho^i}{(c)_i i!},$$

where we use Pochhammer's symbol:

$$(a)_i = a(a+1)(a+2)\cdots(a+i-1), \quad (a)_0 = 1.$$

So $(1)_i$ equals $i!$. Computation of $F(a, 1; c; \rho)$ can be done recursively, as noted in Appendix 3 of Sowell (1992):

$$F(a, 1; c; \rho) = \frac{c-1}{\rho(a-1)} [F(a-1, 1; c-1; \rho) - 1]. \quad (12)$$

In the absence of AR parameters (9) reduces to:

$$\gamma_i = \sigma_\varepsilon^2 \sum_{k=-q}^q \psi_k \frac{\Gamma(1-2d)}{[\Gamma(1-d)]^2} \frac{(d)_{k-i}}{(1-d)_{k-i}}.$$

The ratio

$$\frac{(d)_h}{(1-d)_h} \text{ for } h = p - q - T + 1, \dots, 0, \dots, p + q,$$

can be computed using a forward recursion for $h > 0$:

$$(d)_h = (d+h-1)(d)_{h-1}, \quad h > 0,$$

and a backward recursion otherwise:

$$(d)_h = \frac{(d)_{h-1}}{(d-h)}, \quad h < 0.$$

3.2 Refinements

Sowell (1992) gives several tricks for recursively computing various quantities needed in (9). This is further refined in this section.

The autocorrelations are accumulated in a loop over j to avoid a second storage dimension:

$$\gamma_i = \sigma_\varepsilon^2 \sum_{j=1}^p \zeta_j \left[\sum_{k=-q}^q \psi_k C(d, p+k-i, \rho_j) \right], \quad i = 0, \dots, T-1,$$

¹Note the typo in the equation below (8) in Sowell (1992, p.173): $\Gamma(d+s-l)$ in the numerator should read $\Gamma(d-s+l)$.

Table 2: Leading terms of hypergeometric sequence, $F(d+h, 1; 1-d+h; \rho)$, $h = 2-T, \dots, 17-T$, from backward and forward recursions, with $d = 0.4$, $\rho = -0.1$, $p = 1$, $q = 0$

term	backward	forward	term	backward	forward
1	0.90892	0.90892	9	0.90891	173.55
2	0.90892	0.90890	10	0.90890	-1721.6
3	0.90892	0.90909	11	0.90890	17187.
4	0.90892	0.90717	12	0.90890	-1.7146×10^5
5	0.90891	0.92633	13	0.90890	1.7106×10^6
6	0.90891	0.73512	14	0.90890	-1.7066×10^7
7	0.90891	2.6430	15	0.90889	1.7025×10^8
8	0.90891	-16.394	16	0.90889	-1.6983×10^9

which just reorders the summations in (9).

For each j , we need a sequence of

$$C(d, p - q - T + 1, \rho_j) \cdots C(d, p + q, \rho_j).$$

Because both hypergeometric functions in (11) largely overlap, this is a sequence of functions

$$F(d + h, 1; 1 - d + h; \rho_j), \text{ for } h = p - q - T + 1, \dots, 0, \dots, -(p - q - T + 1).$$

Computation of this sequence can be done with a forward recursion using (12), or with the backward recursion:

$$F(a - 1, 1; c - 1; \rho) = \frac{a - 1}{c - 1} \rho F(a, 1; c; \rho) + 1, \quad (13)$$

starting from the final term:

$$F(d - p + q + T - 1, 1; -d - p + q + T; \rho_j).$$

It is crucial that the recurrence is computed backward, the forward recursion (12) will start to grow exponentially at some stage (see the discussion in Press, Flannery, Teukolsky, and Vetterling, 1993, §5.5). For an example, see Table 2, which gives the first terms (i.e. the terms computed last in the backward recursion, but first in the forward recursion) from the two recursions. The forward recursion becomes meaningless very rapidly.

Returning to (10), the expression for ζ_j involves division by ρ_j , which can create problems when ρ_j gets close to zero. In addition the expression is numerically unstable for a single root close to zero. Equations (9) and (10) can be rewritten slightly, by moving the factor ρ_j^{-1} from ζ_j to C . Both problems are then solved by writing:

$$G(a; c; \rho) = \rho^{-1} \{F(a, 1; c; \rho) - 1\} = \sum_{i=0}^{\infty} \frac{(a)_{i+1}}{(c)_{i+1}} \rho^i.$$

The backward recursion formula for G is:

$$G(a - 1; c - 1; \rho) = \frac{a - 1}{c - 1} [1 + \rho G(a; c; \rho)],$$

Table 3: Comparison of first 5 autocovariances for ARFIMA(2, d , 2) process, with respectively: $\phi_2 = 0$, ϕ_2 omitted, and $\phi_2 = 0$ with singularity removed.

0.3, -0.5, 0 using F	0.3, -0.5 using F	0.3, -0.5, 0 using G
1.2774	1.2726	1.2726
-0.17380	-0.27486	-0.27486
-0.72536	-0.34655	-0.34655
0.59408	-0.045409	-0.045409
-0.018257	0.13155	0.13155

and $\rho^{-1}[\rho^{2p}F(\cdot) + F(\cdot) - 1]$ from (11) becomes:

$$[\rho^{2p}G(d+h; 1-d+h; \rho) + \rho^{2p-1} + G(d-h; 1-d-h; \rho)].$$

Therefore the singularity caused by a single root at zero as present in the expressions of Sowell (1992) turns out to be unnecessary.

To illustrate the improvement, we choose a model with $d = -0.3$, AR parameters 0.3, -0.5, 0 and MA parameters -0.4, 0.3. This model has AR roots: $(0.15 + 0.691i)$, $(0.15 - 0.691i)$, (10^{-17}) . Rounding errors cause the last root to be close but not identical to zero. We may compare this to a model which omits the third AR parameter. Table 3 lists the first 5 autocovariances, clearly showing the increased numerical stability of this reformulation.

4 Likelihood evaluation in estimation

4.1 Concentrating the log-likelihood

Concentrating σ_ϵ^2 out of the log-likelihood saves one parameter for estimation. Starting by writing $\Sigma = R\sigma_\epsilon^2$ in (4):

$$\log L(d, \phi, \theta, \beta, \sigma_\epsilon^2) \propto -\frac{1}{2} \log |R| - \frac{T}{2} \log \sigma_\epsilon^2 - \frac{1}{2\sigma_\epsilon^2} z' R^{-1} z.$$

Differentiating with respect to σ_ϵ^2 , and solving yields

$$\hat{\sigma}_\epsilon^2 = T^{-1} z' R^{-1} z, \quad (14)$$

with concentrated likelihood (CLF):

$$\ell_c(d, \phi, \theta, \beta) = -\frac{T}{2} \log(2\pi) - \frac{T}{2} - \frac{1}{2} \log |R| - \frac{T}{2} \log [T^{-1} z' R^{-1} z].$$

When $\mu = X\beta$ it also beneficial to concentrate β out of the likelihood. The resulting normal profile log-likelihood function becomes:

$$\ell_P(d, \phi, \theta) = -\frac{T}{2} (1 + \log 2\pi) - \frac{1}{2} \log |R| - \frac{T}{2} \log [T^{-1} \hat{z}' R^{-1} \hat{z}], \quad (15)$$

where

$$\hat{z} = y - X\hat{\beta}, \quad \hat{\beta} = (X'R^{-1}X)^{-1} X'R^{-1}y. \quad (16)$$

Finally, the function to be used in the maximization procedure is:

$$-\frac{1}{2} \{T^{-1} \log |R| + \log \sigma_\epsilon^2\}, \quad (17)$$

which can be maximized using numerical derivatives. The formulation of the autocovariances in terms of the ρ_i facilitates an implementation that impose stationarity of the AR polynomial.

4.2 Computing the determinant term and generalized residuals

The second issue to be solved is the evaluation of (17) (and (18) below), involving the inverse of R and computation of its determinant. R is the $T \times T$ Toeplitz matrix of the scaled autocovariances. There are various ways of approaching this problem:

- Naive Choleski decomposition

Using the standard decomposition requires storing the $T \times T$ matrix $R = \mathcal{T} [r_0, \dots, r_{T-1}]$, and Choleski decomposition (lower diagonal, so strictly a matrix with $\frac{1}{2}T(T+1)$ elements, but stored in a $T \times T$ matrix in an econometric programming language like Ox, Doornik, 2001). For large T , this method becomes very memory intensive and slow: the decomposition is of order T^3 .

- Efficient Choleski decomposition

A more efficient algorithm derives the Choleski decomposition directly from r_0, \dots, r_{T-1} , avoiding storage of R . This can be combined with a version of Levinson's algorithm (discussed next) to compute the Choleski decomposition in computations of order T^2 . However, storing the $\frac{1}{2}T(T+1)$ Choleski factor remains prohibitive for large T . This method is used in Sowell (1992) and Smith Jr, Sowell, and Zin (1997).

- Levinson algorithm

The Levinson algorithm solves $\mathcal{T}x = y$ directly, involving an operation count of order T^2 and avoiding storage of order T^2 . The algorithm is described in Golub and Van Loan (1989, §4.7.3).² Evaluation of $z'R^{-1}z$ is achieved in two steps. First rewrite $R^{-1}z = x$ as $z = Rx$ and solve this using the Levinson algorithm to obtain \hat{x} . Then $\hat{\sigma}_\epsilon^2 = T^{-1}z'\hat{x}$, and, with the determinant as a byproduct, the likelihood can be evaluated. Thus, storage of the $\frac{1}{2}T(T+1)$ Choleski factor is avoided. This method was used in earlier versions (up to 0.77) of our Arfima package.

- Durbin algorithm

This method (see Golub and Van Loan, 1989, §4.7.2) amounts to computing the Choleski decomposition of the inverted Toeplitz matrix.³ Durbin's method solves

$$\mathcal{T} [r_0, \dots, r_{T-1}] = LDL' = PP', \quad e = D^{-1/2}L^{-1}z = P^{-1}z.$$

again with an operation count of order T^2 . So we can write:

$$z'R^{-1}z = e'e.$$

²This is implemented in Ox as the `solvetoepplitz` function, which optionally also returns the determinant.

³This is implemented in Ox as the `pacf` function, which optionally also returns the determinant.

Table 4: Timings in seconds for alternative solution methods to evaluate the log-likelihood for ARFIMA(1, d , 1) with $d = 0.45$, $\phi = 0.8$, $\theta = -0.5$.

Sample size	1000	2000	4000	8000	16000
Efficient Choleski	0.03	0.15	0.60	38.70	failed
Levinson algorithm	0.02	0.07	0.25	1.02	12.01
Durbin algorithm	0.01	0.04	0.17	0.77	8.37
Autocovariances	0.0005	0.0011	0.0026	0.0057	0.0131

By applying the factorization as it is computed, storage of the $\frac{1}{2}T(T+1)$ matrix is avoided. This method leads to a more elegant expression of the log-likelihood (in addition to being marginally faster), and is currently used in our Arfima package, Doornik and Ooms (1999).

Durbin's method gives residuals that can be used for testing, and is also useful to generate forecasts.

Table 4 gives timings for one ARFIMA(1, d , 1) likelihood evaluation, when the autocorrelations have already been computed. The timings are for a 256MB Pentium III/700Mhz notebook using Ox 3.01 running under Windows 2000.⁴ At 8 000 observations, the $\frac{1}{2}T(T+1)$ matrix required by the efficient Choleski method takes 288MB to store. This pushes the program into virtual memory (i.e. using hard disk space as memory), making it run much slower. At 16 000 the matrix requires more than 1GB, and does not fit into memory anymore.

Golub and Van Loan (1989, §4.7) gives the floating point operation count of the Levinson algorithm as $4T^2$, and the Durbin algorithm as $2T^2$. The relative timings in Table 4 are somewhat lower than a factor of two. The final row of the table gives the time required for the evaluation of the autocovariances, using the improvements suggested in this note. The time grows linearly with sample size, but, more importantly, is completely negligible at these samples sizes. Therefore, ARFIMA estimation is comparable to ARMA estimation in terms of computational demands (except for the lack of analytical derivatives).

For completeness, we note that, when an ARMA(p , q) model is estimated, there are two additional methods available:

- Banded Choleski

In the ARMA case, the Toeplitz matrix can be transformed to a banded Choleski matrix. This method has been proposed by Ansley (1979).

- Kalman filter

The Kalman filter is implemented in SSFPack, see Koopman, Shephard, and Doornik (1999).

⁴When we started implementation we only had a 32MB Pentium 90Mhz at our disposal, which of course was considerably slower. The lower amount of memory resulted in failure at lower sample size:

Sample size	1000	2000	3000	10000
Efficient Choleski	0.42	2.03	102.15	failed
Levinson algorithm	0.40	1.83	4.03	48.15

4.3 Bias corrected ML estimation

Higher order asymptotic theory has delivered corrections for the bias in the estimation of d , ϕ and θ caused by the estimation of regression parameters, including the constant term. This source of bias has the largest impact on estimates of d , but is also relevant for ϕ . The bias correction leads to lower mean squared errors for the estimators, since this type of bias correction does not increase the variance. See Cheang and Reinsel (2000) for recent analytical results on this topic. Bias corrected estimation of d , ϕ and θ also lead to more accurate inference for the regression parameters and more precise forecast intervals.

The objective functions for the resulting bias corrected estimators can be written as modified likelihood functions in a number of different approaches. We implemented the modified profile likelihood, MPL. The modified profile log-likelihood, ℓ_M , for the regression model with stationary ARFIMA-errors and $\mu = X\beta$ is written as:

$$\ell_M(d, \phi, \theta) = -\frac{T}{2}(1 + \log 2\pi) - \left(\frac{1}{2} - \frac{1}{T}\right) \log |R| - \frac{T-k-2}{2} \log [T^{-1}\hat{z}'R^{-1}\hat{z}] - \frac{1}{2} \log |X'R^{-1}X|, \quad (18)$$

see An and Bloomfield (1993), who applied the idea of Cox and Reid (1987) to remove the first order bias of the EML estimator due to the presence of unknown nuisance parameters of the regressors. Ooms and Doornik (1999) showed the effectiveness of the bias correction for the estimation of d in ARFIMA models for US and UK inflation rates.

The ‘bias correction term’ $|X'R^{-1}X|$ also appears in the objective functions for the marginal likelihood of Tunnicliffe Wilson (1989), and in restricted maximum likelihood, REML, of Harville (1974), see Laskar and King (1998) for a comparison of the different formulas. Note that the term is especially large when X contains polynomial trends (of high order). Evaluation of $|X'R^{-1}X|$ for MPL proceeds along the lines suggested above for $\hat{z}'R^{-1}\hat{z}$.

The residual variance estimator can also be bias corrected using modified likelihood functions. When $p = q = d = 0$, this leads to the familiar degrees of freedom correction due to the estimation of regression parameters.

$$\hat{\sigma}_\epsilon^2 = \frac{1}{T-k} \hat{z}'R^{-1}\hat{z}. \quad (19)$$

Other analytical methods of bias correction lead to degrees of freedom corrections due to the estimation of ϕ and d . Cheang and Reinsel (2000) derived the degrees of freedom correction due to estimating AR-parameters in approximate ML estimation of regression models with AR disturbances. Lieberman (2001) analysed the modified score estimator of Firth (1993) for (d, σ_ϵ^2) in an ARFIMA(0, d , 0) model and showed that first order unbiased estimation of σ_ϵ^2 requires a degrees of freedom correction for the estimation of the parameter d .

4.4 Invertibility of MA polynomial

The same likelihood pertains when the roots of the MA polynomial are inverted. Since the likelihood of a non-invertible MA can be evaluated without problems, estimation is not affected. In a Monte Carlo experiment, however, it is essential that non-invertibility is taken into account. Take an MA(1), with $\theta = 0.5$. Since $\theta = 2$ yields the same likelihood, it is thinkable that half the experiments yield $\hat{\theta} \approx 0.5$ and the other half $\hat{\theta} \approx 2$, resulting in poor average estimates from the Monte Carlo experiment.

The following table illustrates the issue ($T = 100$, $M = 100$). The first set of results removes the non-invertible MA (required in 19 cases), the second leaves the MA roots unchanged:

coefficients	DGP	mean $\hat{\theta}$	std.dev	mean bias
with MA inversion				
θ_1	0.9	0.89157	0.075471	-0.0084326
θ_2	0.81	0.81967	0.11363	0.0096664
without inversion				
θ_1	0.9	0.93546	0.26420	0.035462
θ_2	0.81	0.86154	0.13834	0.051540

4.5 Alternative computations and approximations of the likelihood

Two alternatives for the computation of the exact log-likelihood have been suggested, which attempt to avoid the computation of the T autocovariances for each (d, ϕ, θ) . Chan and Palma (1998) use a prediction error decomposition of the likelihood, which is easily calculated using Kalman Filter recursions. Unfortunately, the required state dimension equals the sample size T and the computation of the covariance matrix of the initial state still requires the computation of Σ . Both with regard to storage requirements and number of computations this is not an attractive method for exact maximum likelihood estimation.

Ravishanker and Ray (1997) employ a factorization of the likelihood earlier suggested by Pai and Ravishanker (1996). Their method only requires computation of the first p and the last $2p + q - 1$ autocovariances and is easily extended to multivariate models. They achieve this by introducing $p + q$ extra parameters which represent p pre-sample values of z_t and q pre-sample values of ε_t .

Beran (1995) investigated the non-linear least squares method, which is also applicable for non-stationary ARFIMA-processes with $d > 0.5$. The approximate log likelihood is:

$$\log L_A(d, \phi, \theta, \beta) = c - \frac{1}{2} \log \frac{1}{T - k} \sum_{t=2}^T \tilde{\varepsilon}_t^2. \quad (20)$$

The $\tilde{\varepsilon}_t$ are the one-step-prediction errors from the AR(∞) representation of z_t :

$$z_t = \sum_{j=1}^{\infty} \pi_j z_{t-j} + \varepsilon_t,$$

where pre-sample values are set to zero in forecasting. Beran proved asymptotic efficiency and normality of the resulting estimators for (d, ϕ, θ) . Since nonlinear least squares estimation is computationally simpler than maximum likelihood, and since it does not require d to be in the stationarity region, it is an attractive method to obtain starting values for ML estimation. Beveridge and Oickle (1993) and Chung and Baillie (1993) presented Monte Carlo evidence which suggest it to be a good estimator for ARFIMA(0, d , 0) models with unknown mean.

4.6 Multivariate extensions

The issues in the discussion of ML estimation for univariate ARFIMA processes extend to the multivariate case. Sowell showed how to evaluate the autocovariances of bivariate ARFIMA process, see Sowell (1987) and Sowell (1989). In the bivariate case the likelihood contains a scaled covariance matrix R of dimension $2T$. Evaluation of the inverse and determinant of this matrix corresponding to R

in (4) then involves the application of a multivariate Durbin-Levinson Algorithm, see e.g. Brockwell and Davis (1993, §11.4). Dueker and Startz (1998) implemented the likelihood function of Sowell (1987) in a bivariate ARFIMA(2, d ,1) model for 121 observations. Without the improvements in the likelihood evaluation suggested above, estimation of their model was still rather time consuming, so as to make interactive model selection practically impossible ⁵.

5 ARFIMA data generation

Finite sample improvements of estimators and inferences can also be achieved by simulation methods like the parametric bootstrap. Simulation methods require efficient generation of pseudo-samples. The problem in data generation for the ARFIMA(p, d, q) process is analogue to that set out in §4.2.

For small samples one can use the naive Choleski method as in the likelihood evaluation. Let r be the standardized autocovariances of the specified process, and $\mathcal{T}[r] = PP'$, then

$$y = \sigma_\varepsilon P\varepsilon + \mu,$$

where ε are drawings from the standard normal distribution. For small T , this is convenient, because P only needs to be computed once. Once the Choleski decomposition has been computed, generating data is only of order T^2 .

For larger samples, a modified version of Durbin's algorithm is used to apply the inverted filter:

$$\mathcal{T}[r_0, \dots, r_{T-1}] = PP', \quad z = Pe.$$

This algorithm is of order T^2 , but perhaps somewhat slower than the naive method for small T . However, it allows for simulation with a large number of observations.

6 Conclusion

Maximum Likelihood estimation of ARFIMA models with explanatory variables is often considered prohibitively slow. We attend key factors in the estimation process, as to make ML estimation of ARFIMA no more problematic than ML estimation of ARMA models. These factors are: efficient computation of autocovariance functions, careful evaluation of associated hypergeometric functions, concentrating out regression and scale parameters, and application of the Durbin algorithm to take advantage of the Toeplitz structure of the covariance matrix. The Durbin algorithm is also key in the efficient simulation of ARFIMA processed and in the evaluation of extra terms in bias corrected ML estimation.

Acknowledgements

Financial support from the UK Economic and Social Research Council (grant R000237500) is gratefully acknowledged by JAD.

⁵Dueker and Startz (1998) reproduced the autocovariance formulas by Sowell, but forgot to define $B(z) = \text{Adj}[\Phi(z)]\Theta(z)$

References

- Abramowitz, M. and I. A. Stegun (1970). *Handbook of Mathematical Functions*. New York: Dover Publications Inc.
- An, S. and P. Bloomfield (1993). Cox and Reid's modification in regression models with correlated errors. Discussion paper, Department of Statistics, North Carolina State University, Raleigh, NC 27695-8203, U.S.A.
- Ansley, C. F. (1979). An algorithm for the exact likelihood of a mixed autoregressive-moving average process. *Biometrika* 66, 59–65.
- Beran, J. (1995). Maximum likelihood estimation of the differencing parameter for invertible short and long memory autoregressive integrated moving average models. *Journal of the Royal Statistical Society* 57, 659–672.
- Beveridge, S. and C. Oickle (1993). Estimating fractionally integrated time series models. *Economics Letters* 43, 137–142.
- Bollerslev, T. and D. Jubinski (1999). Equity trading volume and volatility: Latent information arrivals and common long-run dependencies. *Journal of Business & Economic Statistics* 17, 9–21.
- Brockwell, P. J. and R. A. Davis (1993). *Time Series: Theory and Methods (2nd ed.)*. USA: Springer-Verlag, New-York.
- Chan, N. H. and W. Palma (1998). State space modeling of long-memory processes. *Annals of Statistics* 26, 719–740.
- Cheang, W. K. and G. C. Reinsel (2000). Bias reduction of autoregressive estimates in time series regression model through restricted maximum likelihood. *Journal of the American Statistical Association* 95, 1173–1184.
- Chung, C.-F. and R. T. Baillie (1993). Small sample bias in conditional sum-of-squares estimators of fractionally integrated ARMA models. *Empirical Economics* 18, 791–806.
- Cox, D. R. and N. Reid (1987). Parameter orthogonality and approximate conditional inference (with discussion). *Journal of the Royal Statistical Society Series B* 49, 1–39.
- Doornik, J. A. (2001). *Object-Oriented Matrix Programming using Ox* (4th ed.). London: Timberlake Consultants Press.
- Doornik, J. A. and M. Ooms (1999). A package for estimating, forecasting and simulating ARFIMA models: Arfima package 1.0 for Ox. Discussion paper, Nuffield College, Oxford.
- Dueker, M. and R. Startz (1998). Maximum-likelihood estimation of fractional cointegration with an application to U.S. and Canadian bond rates. *Review of Economics and Statistics* 80, 420–426.
- Firth, D. (1993). Bias reduction of maximum likelihood estimates. *Biometrika* 80, 27–38.
- Golub, G. H. and C. F. Van Loan (1989). *Matrix Computations*. Baltimore: The Johns Hopkins University Press.
- Harvey, A. C. (1993). *Time Series Models, second edition*. New York, London: Harvest Wheatsheaf.
- Harville, D. A. (1974). Bayesian inference for covariance components using only error contrasts. *Biometrika* 61, 383–385.

- Hosking, J. R. M. (1981). Fractional differencing. *Biometrika* 68, 165–176.
- Koopman, S. J., N. Shephard, and J. A. Doornik (1999). Statistical algorithms for models in state space using SsfPack 2.2 (with discussion). *Econometrics Journal* 2, 107–160.
- Laskar, M. and M. King (1998). Estimation and testing of regression disturbances based on modified likelihood functions. *Journal of Statistical Planning and Inference* 71, 75–92.
- Lieberman, O. (2001). Penalised maximum likelihood estimation for fractional Gaussian processes. *Biometrika* 88, 888–894.
- Ooms, M. and J. A. Doornik (1999). Inference and forecasting for fractional autoregressive integrated moving average models, with an application to US and UK inflation. Discussion Paper EI 9947/A, Econometric Institute, Erasmus University Rotterdam.
- Pai, J. S. and N. Ravishanker (1996). Exact likelihood function forms for an ARFIMA process. In *Data Analysis and Information Systems, Statistical and Conceptual Approaches. Proceedings of the 19th Annual Conference of the Gesellschaft für Klassifikation e.V., University of Basel, March 8-10, 1995.*, pp. 323–331. Springer Verlag, Berlin.
- Piessens, R., E. de Donker-Kapenga, C. W. Überhuber, and D. K. Kahaner (1983). *QUADPACK, A Subroutine Package for Automatic Integration*. Heidelberg: Springer-Verlag.
- Press, W. H., B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling (1993). *Numerical Recipes in C* (2nd ed.). New York: Cambridge University Press.
- Ravishanker, N. and B. K. Ray (1997). Bayesian analysis of vector ARFIMA processes. *Australian Journal of Statistics* 39, 295–312.
- Smith Jr, A. A., F. Sowell, and S. E. Zin (1997). Fractional integration with drift: Estimation in small samples. *Empirical Economics* 22, 103–116.
- Sowell, F. (1987). Maximum likelihood estimation of fractionally integrated time series models. Discussion Paper 87-07, Department of Economics, Duke University.
- Sowell, F. (1989). A decomposition of block toeplitz matrices with applications to vector time series. Discussion paper, GSIA, Carnegie Mellon University.
- Sowell, F. (1992). Maximum likelihood estimation of stationary univariate fractionally integrated time series models. *Journal of Econometrics* 53, 165–188.
- Tunnicliffe Wilson, G. (1989). On the use of marginal likelihood in time series model estimation. *Journal of the Royal Statistical Society, Series B* 51, 15–27.