A NUMERICAL ALGORITHM FOR STABLE 2D AUTOREGRESSIVE FILTER DESIGN

Hugo J. Woerdeman, Jeffrey S. Geronimo, and Glaysar Castro

Department of Mathematics, The College of William and Mary, P.O. Box 8795, Williamsburg, VA 23187-8795

School of Mathematics, Georgia Institute of Technology, Atlanta, GA 30332-0160

School of Mathematics, Universidad Central de Venezuela, Caracas, Venezuela

Abstract. Based on previous theoretical results we present in this paper a global estimation scheme for solving the stable 2D autoregressive filter problem. The different algorithms are based on the traditional Newton method and on the log barrier method that is employed in semi-definite programming. The Newton method is the faster one but the barrier method ensures that the iterates stay in the cone of positive semidefinites. In addition, a numerical test for the existence of a stable factorization of a two-variable squared magnitude response function is presented.

Corresponding author:

Hugo J. Woerdeman
hugo@math.wm.edu

Mailing address until July 1, 2003:
ESAT-SISTA
Katholieke Universiteit Leuven
Kasteelpark Arenberg 10

1991 *Mathematics Subject Classification.* 42C05, 15A48, 42B05, 47A57, 47A20, 60G25, 60G10.

Key words and phrases. Autoregressive filter, AR process, bivariate stationary stochastic processes, structured matrix completions, two-variable Toeplitz matrix, two-variable polynomials, stability.

B-3001 Leuven-Heverlee
Belgium

voice: +32-16.32.17.98
Fax: +32-16.32.19.70

Mailing address after August 1, 2002:
Department of Mathematics
P. O. Box 8795
The College of William and Mary
Williamsburg, VA 23187-8795

voice: +1-757-221-2022
fax: +1-757-221-2988
hugo@math.wm.edu

§1 Introduction

In one dimension the stable autoregressive model has had much success in time series analysis with applications to prediction theory and analysis of speech. It is well known that given complex numbers $\overline{c_{-k}} = c_k$, $k = 0, \ldots, n$, a necessary and sufficient condition for the existence of a stable autoregressive process with autocorrelation coefficients c_k is that the Toeplitz matrix $(c_{k-l})_{k,l=0}^{n}$ be positive definite. Moreover, the filter coefficients may be found via the celebrated Yule-Walker equation.

A closely related problem is that of filter design. A common approach to the frequency domain design of one dimensional recursive digital filters is to find the squared magnitude response of an ARMA filter that best approximates the desired response. Spectral factorization is then performed to find the unique minimum-phase filter that upon taking its magnitude square gives back the original response. Such a filter is, by construction, stable. Note that this factorization is usually accomplished by transforming the squared

magnitude function into a polynomial then finding its roots. This technique uses the fundamental theorem of algebra which says that every polynomial may be factored over the complex numbers into a product of linear factors. In more than one variable there is no fundamental theorem of algebra.

In this paper we address the stable autoregressive filter representation problem for bivariate processes and the two variable spectral factorization problem for the squared magnitude response function. For the stable autoregressive filter representation problem we are given real (or complex) numbers $c_{k,l}$ where the indices (k,l) range in a finite set Λ_+, i.e. $\Lambda_+ = \{0 \leq k \leq n, 0 \leq l \leq m\}$. The objective is to construct a stable autoregressive filter that yields a bivariate stationary process with autocorrelation coefficients $c_{k,l}$. In a recent paper [15] two of the authors gave necessary and sufficient conditions on the bi-Toeplitz matrix $\Gamma = (c_{k-r,l-s})_{(k,l),(r,s)\in\Lambda_+}$ so that such a filter exists. Unlike in the one dimensional case the positive definiteness of the above matrix is not sufficient to guarantee the existence of a stable filter. Also unlike the one dimensional case there are entries in Γ that are not just the complex conjugate of the coefficients $c_{l,k},\ (l,k) \in \Lambda_+$. In this paper we present global estimation schemes that allows one to construct the missing entries in the matrix Γ if they exist, and subsequently the stable filter associated with this process.

For the spectral factorization problem instead of using the cepstrum we will use the impulse response function (Fourier coefficients) associated with one over the magnitude squared function. With this impulse response function we construct Γ given above and if it satisfies certain conditions given below we construct a stable spectral factor for the two variable magnitude squared function.

Given that the amount of computation needed for finding suitable filters for two variable problems can be prohibitive there has been recent effort to develop fast algorithms to compute least squares solutions [18]. Other algorithms exploit structures inherent in the system to help speed up computations [6]. These algorithms however are not guaranteed to give stable filters that give the prescribed correlation coefficients.

The layout of the paper is as follows. In section 2 we review the theoretical results regarding the stable autoregressive filter representation problem for bivariate processes and the two variable spectral factorization problem for the squared magnitude response function. In sections 3 and 4 we present two different techniques for solving the stable autoregressive filter representation problem numerically. In the case of real data, the numerical schemes discussed in section 3 and 4 have been implemented in Matlab, and in section 5 we present the outcome of a series of numerical experiments. In section 6 we apply the theoretical results in Section 1 regarding the factorization of the squared magnitude response function and show numerical data. The algorithm checks whether the squared magnitude function has a stable spectral factor and if so we compute this factor. This is accomplished by examining the impulse response function (Fourier coefficients) associated with one over the squared magnitude function. In section 7, we discuss briefly the case of complex data. Finally, in section 8 we draw conclusions.

§2 Theoretical Results

Let $X = \{x_{k,l} : (k,l) \in \mathbb{Z}^2\}$ be a second-order zero-mean stationary stochastic process of two discrete variables. In particular, $E(x_{k,l}) = 0$ and $E(x_{k,l}\bar{x}_{p,q}) = E(x_{k+t,l+s}\bar{x}_{p+t,q+s})$ for all $k,l,p,q,t,s \in \mathbb{Z}$. Let $\Lambda_+ = \{0,\dots,n\} \times \{0,\dots,m\}$. We recall the definition (see, e.g.,

[5], [17]) of autoregressive (AR) processes. A second-order zero-mean stationary stochastic process X is said to be AR(Λ_+) if there exists complex numbers $a_{kl}, (k,l) \in \Lambda_+ \setminus \{(0,0)\}$, so that for every t and s,

$$x_{t,s} + \sum_{\substack{k,l\in\Lambda_+ \\ (k,l)\neq(0,0)}} a_{kl} x_{t-k,s-l} = e_{t,s}, \qquad (t,s) \in \mathbb{Z}^2, \tag{2.1}$$

where $\{e_{k,l}\ ; (k,l) \in \mathbb{Z}^2\}$ is a white noise zero mean process with variance σ^2. Let $H = \{(n,m) : n > 0 \text{ or } (n = 0 \text{ and } m > 0)\}$ be the standard halfspace in $\mathbb{Z}^2$. The AR(Λ_+) process is said to be *causal* if there is a solution to equations (2.1) of the form

$$x_{t,s} = \sum_{k,l\in H\cup\{(0,0)\}} \phi_{k,l} e_{t-k,s-l}, (t,s) \in \mathbb{Z}^2, \tag{2.2}$$

with $\sum_{k,l\in H\cup\{(0,0)\}} |\phi_{k,l}| < \infty$. It is not difficult to see that the $AR(\Lambda_+)$ process X is causal if and only if its associated pseudopolynomial

$$p(z,w) = 1 + \sum_{\substack{k,l\in\Lambda_+ \\ (k,l)\neq(0,0)}} \overline{a}_{kl} z^k w^l$$

is *stable*, i.e., $p(z,w) \neq 0$ $(z,w) \in (\{0\}\times\overline{\mathbb{D}})\cup(\overline{\mathbb{D}}\times\mathbb{T})$ (see [14]), where $\mathbb{D} = \{z \in \mathbb{C} : |z| < 1\}$, $\mathbb{T} = \{z \in \mathbb{C} : |z| = 1\}$, and $\overline{\mathbb{D}} = \mathbb{D} \cup \mathbb{T}$. In [15] it was proven that the stability of p is equivalent to the statement that $p(z,w) \neq 0$, $(z,w) \in \overline{\mathbb{D}} \times \overline{\mathbb{D}}$. Using this equivalence it follows that a causal AR(Λ_+) process is in fact *quarterplane causal*, which by definition means that there is a solution to equations (2.1) of the form

$$x_{t,s} = \sum_{\substack{k,l\geq 0 \\ (k,l)\neq(0,0)}} \phi_{k,l} e_{t-k,s-l}, (t,s) \in \mathbb{Z}^2, \tag{2.3}$$

Thus we see that it is important to develop criteria for when a polynomial of two variables is stable. Conditions for stability have been extensively investigated [23], [10], [12], [13] and more recently in [2]. In [15] a set of three one dimensional tests were developed that characterize whether or not a two variable polynomial is stable.

The *bivariate autoregressive filter design problem* is the following. Given are autocorrelation elements

$$c_{k,l} = E(x_{k,l}\bar{x}_{0,0}), \qquad (k,l) \in \Lambda_+.$$

What conditions must the autocorrelation coefficients satisfy in order that these are the autocorrelation coefficients of a causal AR(Λ_+) process? And in that case, how does one compute the filter coefficients $a_{k,l}$, $(k,l) \in \Lambda_+ \setminus \{(0,0)\}$ and σ^2?

The following characterization for the existence of a causal solution for an AR(Λ_+) process was obtained in [15].

Theorem 2.1. [15] *Let $\Lambda_+ = \{0,\dots,n\} \times \{0,\dots,m\}$, and let $c_{k,l}$, $(k,l) \in \Lambda_+$, be given complex numbers. There exists a causal AR(Λ_+) process with the given autocorrelation elements if and only if there exist complex numbers $c_{k,l}$, $(k,l) \in \{1,\dots,n\} \times \{-m,\dots,1\}$, so that the $(n+1)(m+1) \times (n+1)(m+1)$ doubly indexed Toeplitz matrix*

$$\Gamma = \begin{bmatrix} C_0 & \cdots & C_{-n} \\ \vdots & \ddots & \vdots \\ C_n & \cdots & C_0 \end{bmatrix},$$

where

$$C_j = \begin{bmatrix} c_{j0} & \cdots & c_{j,-m} \\ \vdots & \ddots & \vdots \\ c_{jm} & \cdots & c_{j0} \end{bmatrix}, \qquad j = -n,\dots,n,$$

and $c_{-k,-l} = \overline{c_{k,l}}$, has the following two properties:

(1) Γ *is positive definite;*

(2) *the inverse of the* $(nm+n+m)\times(nm+n+m)$ *matrix* $\widetilde{\Gamma}$ *obtained from* Γ *by removing its first scalar row and first scalar column, satisfies*

$$(\widetilde{\Gamma}^{-1})_{qr}=0, q\in\{m+1,2(m+1),\dots,n(m+1)\}, r\in\{1,\dots,m\}. \tag{2.4}$$

In this case one finds the vector

$$\frac{1}{\sigma^2}[a_{nm}\cdots a_{n0}\ \cdots\ a_{0m}\cdots a_{01}\ 1]$$

as the last row of the inverse of Γ.

The above result, as was shown in [15], has an important consequence regarding the question of stable factorization of a two-variable trigonometric polynomial $f(z,w)=\sum_{k=-n}^{n}\sum_{l=-m}^{m}f_{kl}z^kw^l$. For one-variable trigonometric polynomials $f(z)=\sum_{i=-n}^{n}f_iz^i$ it is the classical Riesz-Fejer lemma that states that as soon as $f(z)\geq 0$, $|z|=1$, one may factor f as $f(z)=p(z)\bar{p}(1/z)$ where $p(z)=\sum_{i=0}^{n}p_iz^i$ is a polynomial of degree n and $\bar{p}(z)=\sum_{i=0}^{n}\bar{p}_iz^i$. In fact, one may choose $p(z)$ to have all its roots outside the open unit disk. In case $f(z)>0$ then $p(z)$ may be chosen to have all its roots outside the closed unit disk (i.e., p is stable), and one speaks of a stable factorization. It is easy to see that mere positivity on the bi-circle does not suffice for a two-variable trigonometric polynomial to have a stable factorization. The following result, which is a corollary of Theorem 2.1 (see [15] for details), gives necessary and sufficient conditions when a stable factorization does exist.

Theorem 2.2. [15] *Suppose that* $f(z,w)=\sum_{k=-n}^{n}\sum_{l=-m}^{m}f_{kl}z^kw^l$ *is positive for* $|z|=|w|=1$. *Then there exists a polynomial* $p(z,w)=\sum_{k=0}^{n}\sum_{l=0}^{m}p_{kl}z^kw^l$ *with* $p(z,w)\neq 0$ *for*

$|z|, |w| \leq 1$, and $f(z,w) = |p(z,w)|^2$ if and only if the matrix Γ as in Theorem 2.1 built from the Fourier coefficients $c_{k,l} := \widehat{\frac{1}{f}}\,(k,l)$ of the reciprocal of f, satisfies condition (2) of Theorem 2.1. In that case, the polynomial may be found via $p_{kl} = \frac{b_{kl}}{b_{00}}$, where the vector

$$[b_{00}\ b_{01}\ \cdots\ b_{0m}\ \cdots\ \cdots\ b_{n0}\ b_{n1}\ \cdots\ b_{nm}]^T$$

is the first column of the inverse of Γ. The polynomial p is unique up to multiplication with a complex number of modulus 1.

In Section 6 we will apply this theorem to numerical examples.

§3 Newton Method

Since a solution to the 2D autoregressive filter problem is characterized by zeros in an inverse, one may use the Newton method to find such a solution. In order to implement this we use the observation that the (3,1) block entry of the inverse of $(A_{ij})_{i,j=1}^3$ equals zero if and only if $A_{13} - A_{12}A_{22}^{-1}A_{23} = 0$, provided the appropriate inverses exist. We shall use the following notations. If $A = (a_{ij})_{i\in I, j\in J}$ $I, J \subset \mathbb{Z}^2$, then for $K \subseteq I$ and $L \subseteq J$ we denote $A(K,L) = (a_{ij})_{i\in K, j\in L}$. For instance, in this notation, $\Gamma = (c_{i-j})_{i,j\in\Lambda_+}$ and $\widetilde{\Gamma} = \Gamma(\Lambda_+ \setminus \{(0,0)\}, \Lambda_+ \setminus \{(0,0)\})$. In addition $\mathbb{C}^K$ denotes the vector space of vectors $(a_k)_{k\in K}$, where $a_k \in \mathbb{C}$.

We now introduce

$$S_+ = \{1, \ldots, n\} \times \{-m, \ldots, -1\},$$

$$x = (c_{kl})_{kl\in S_+} \in \mathbb{C}^{S_+}, F(x) : \mathbb{C}^{S_+} \to \mathbb{C}^{n\times m},$$

where

$$F(x) = \Gamma(v_1, v_3) - \Gamma(v_1, v_2)\Gamma(v_2, v_2)^{-1}\Gamma(v_2, v_3),$$

$$v_1 = \{1, \dots, n\} \times \{0\}, v_2 = \{1, \dots, n\} \times \{1, \dots, m\},$$

$$v_3 = \{0\} \times \{1, \dots, m\},$$

and $\Gamma = \Gamma(x)$ is as in Theorem 2.1 viewed as a function of x. As an example, note that when $n = m = 2$, then

$$F(x_{1,-2}, x_{1,-1}, x_{2,-1}, x_{2,-2}) = \begin{pmatrix} x_{1,-1} & x_{1,-2} \\ x_{2,-1} & x_{2,-1} \end{pmatrix} - \begin{pmatrix} c_{0,-1} & c_{0,-2} & c_{-1,-1} & c_{-1,-2} \\ x_{1,-1} & x_{1,-2} & c_{0,-1} & c_{0,-2} \end{pmatrix} \times$$
$$\times \begin{pmatrix} c_{0,0} & c_{0,-1} & c_{-1,0} & c_{-1,0} \\ c_{0,1} & c_{0,0} & \overline{x_{1,-1}} & c_{-1,0} \\ c_{1,0} & x_{1,-1} & c_{0,0} & c_{0,-1} \\ c_{1,1} & c_{1,0} & c_{0,1} & c_{0,0} \end{pmatrix}^{-1} \begin{pmatrix} c_{1,0} & x_{1,-1} \\ c_{1,1} & c_{1,0} \\ c_{2,0} & x_{2,-1} \\ c_{2,1} & c_{2,0} \end{pmatrix},$$

where as usual we have used the lexicographical ordering.

Notice that condition (2) in Theorem 2.1 is equivalent to $F(x) = 0$ (provided the appropriate inverse exists). Consequently, we may perform the Newton method to find an x_* so that $F(x_*) = 0$, and if it so happens that $\Gamma(x_*) > 0$, then we have found a solution to the 2D autoregressive filter problem. We have implemented the search for a solution to $F(x) = 0$ by simply using the routine **fsolve** in Matlab. We chose the default, which corresponds to a large-scale algorithm. This algorithm is a subspace trust region method and is based on the interior-reflective Newton method described in [8], [9].

§4. The Barrier Method

A possible downside of the Newton method as described in the previous section is that there is no guarantee that the solution is in the cone of positive definite matrices. An alternative way to deal with this is to use the log barrier method, in which the function $\log \det$ is used as a barrier to prevent the iterates, starting from a positive definite matrix, from leaving the cone of positive definite matrices. For exact details, see [21] or [4].

Since the characterization in Theorem 2.1 (2) involves zeros in the inverse of the positive definite matrix $\widetilde{\Gamma}$ (= the matrix Γ without the first row and column), the problem is reminiscent of finding an analytic center of a linear matrix inequality $A(x) = A(x_1, \dots, x_p) := A_0 + \sum_{i=1}^{p} x_i A_i > 0$, say. Here A_i, $i = 1, \dots, p$, are real symmetric matrices. Typically the analytic center $A(x_*)$, is characterized by the optimality conditions $\mathrm{trace} A(x_*)^{-1} A_i = 0$ and is found by maximizing the function $\log\det A(x)$ over the set $\{x \ : \ A(x) > 0\}$ (see Section 2.4.1 in [4]). Unfortunately for the case under consideration it is necessary to modify the above optimality conditions. Indeed, the optimality conditions are $\mathrm{trace} A(x_*)^{-1} P(A_i) = 0$, $i = 1, \dots, p$, where $P(A_i)$ is obtained from A_i by making some entries of A_i zero, in a manner to be explained below.

In order to conform to the setting in [4] we will write the matrices as affine matrix functions. In particular, let $x = (x_1, \dots, x_{nm}) = (c_{k,l})_{(k,l)\in\{1,\dots,n\}\times\{-m,\dots,-1\}} \in \mathbb{C}^{nm}$ denote the unknowns in one's favorite ordering. With the x given above write $\widetilde{\Gamma}(x) = A_0 + \sum_{i=1}^{nm} x_i A_i$, where $A_0 = \widetilde{\Gamma}(0)$ and $A_i = \frac{\partial}{\partial x_i}\widetilde{\Gamma}(x)$. For instance, when $n = 2$, and $m = 1$, we find that $x = (x_1, x_2)$ where $x_1 = c_{1,-1}$, and $x_2 = c_{2,-1}$. In this case using a lexicographic ordering (our favorite choice), we have that

$$A_0 = \begin{bmatrix} c_{0,0} & 0 & c_{1,0} & 0 & c_{2,0} \\ 0 & c_{0,0} & c_{0,1} & c_{1,0} & c_{1,1} \\ c_{1,0} & c_{0,1} & c_{0,0} & 0 & c_{1,0} \\ 0 & c_{1,0} & 0 & c_{0,0} & c_{0,1} \\ c_{2,0} & c_{1,1} & c_{1,0} & c_{0,1} & c_{0,0} \end{bmatrix}, A_1 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, A_2 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Thus the rows here are indexed by $\Lambda_+ \setminus \{(0,0)\} = \{(0,1),(1,0),(1,1),(2,0),(2,1)\}$. The matrices $P(A_i)$ that appear in the optimality conditions are obtained from A_i by leaving the entries $((k_1,l_1),(k_2,l_2)) \in S := [(\{1,\dots,n\}\times\{0\})\times(\{0\}\times\{1,\dots,m\})] \cup [(\{0\}\times\{1,\dots,m\})\times(\{1,\dots,n\}\times\{0\})]$ alone, and making the others equal to 0. Thus in the

above example

$$P(A_1) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, P(A_2) = A_2.$$

Note that S corresponds exactly to the locations in $\widetilde{\Gamma}^{-1}$ that we want to equal 0. We let the vector y denote these entries that lie in the lower triangular part, or in other words,

$$y = \widetilde{\Gamma}^{-1}(\{1, \ldots, n\} \times \{0\}, \{0\} \times \{1, \ldots, m\}) =: (y_{ij})_{i=1,j=-m}^{n \quad -1}.$$

We order y as a vector in $\mathbb{C}^{nm}$ with the same ordering that was used for x. In the example above we would have that $y = (y_{1,-1}, y_{2,-1})$ where $y_{1,-1}$ and $y_{2,-1}$ are the $((1,0),(0,1))$ and the $((2,0),(0,1))$ entries of $\widetilde{\Gamma}^{-1}$. As the rows and columns of $\widetilde{\Gamma}^{-1}$ are indexed by $\Lambda_+ \setminus \{(0,0)\}$ and we have ordered that set lexicographically, these correspond to the (2,1) and the (4,1) entries of $\widetilde{\Gamma}^{-1}$ when one would just number the rows and columns 1,2,3,4,5.

We now describe our **algorithm**. We begin with determining an x such that $\Gamma(x) > 0$ (we used SDPPACK [1] for this step). Next we perform the following while loop,

while $\|y\| >$tol, **do**

$$H_{ij} = \text{trace}(\widetilde{\Gamma}^{-1} A_i \widetilde{\Gamma}^{-1} A_j), H = (H_{ij})_{i,j=1}^{nm},$$

$$v = H^{-1} y, \delta = \sqrt{v^T y},$$

$$\alpha = 1 \textbf{ if } \delta < 1/4 \textbf{ and } \frac{1}{1+\delta} \textbf{ if } \delta \geq 1/4,$$

$$x := x + \alpha v, \widetilde{\Gamma} = A_0 + \sum_{i=1}^{nm} x_i A_i,$$

$$y := \widetilde{\Gamma}^{-1}(\{1, \ldots, n\} \times \{0\}, \{0\} \times \{1, \ldots, m\}).$$

end{while}.

It follows from [21] (see also [3]) that the above procedure takes positive matrices to positive matrices. To see this write $A(z) = A(x) + \sum(z_i - x_i)A_i$. If $A(x) > 0$ write $\tilde{A}(z) = A(x)^{-1/2}A(z)A(x)^{-1/2}$ and $\tilde{A}_i = A(x)^{-1/2}A_iA(x)^{-1/2}$. Put $z = x + \alpha v$. Note that $(z-x)^T H^{-1}(z-x) = \alpha^2 v^T H^{-1} v = \alpha^2\delta^2 < 1$. But then

$$\begin{aligned} 1 >& (z-x)^T H^{-1}(z-x) \\ &= \sum_{i,j}(z_i - x_i)\text{trace}(A(x)^{-1}A_iA(x)^{-1}A_j)(z_j - x_j) \\ &= ||\tilde{A}(z) - I||_F^2 > ||\tilde{A}(z) - I||^2, \end{aligned}$$

where $\|\cdot\|_F$ stands for the Frobenius norm. This implies that $\tilde{A}(z) > 0$, and hence $A(z) > 0$.

Several experiments have been run with sizes up to $n = m = 5$. Experimental data are presented in Section 5.

§5 Experimental results

In order to compare the Newton method and the barrier method we apply both methods to collections $\{c_{k,l} \ : \ (k,l) \in \Lambda_+\}$. We generate such data according to the rule $c_{kl} = 1.8^{-k-l}(2\mathbf{rand} - 1), (k,l) \in \Lambda_+ \setminus \{(0,0)\}$, $c_{00} = 0.9 + |2\mathbf{rand} - 1|$. Here **rand** is Matlab's random generator, which produces a number between 0 and 1. When an algorithm (Newton or barrier) is successful a number is computed quantifying the goodness of the fit. This measure of the goodness of the fit is computed by associating with the AR-filter with

coefficients $a_{k,l}$ and σ^2, a polynomial

$$p(z,w) = \frac{1}{\sigma} \sum_{k,l \in \Lambda_+} a_{k,l} z^k w^l,$$

where we set $a_{0,0} = 1$. Subsequently the Fourier coefficients $\widehat{f}(k,l)$ of $f = \frac{1}{|p|^2}$ are approximated. The approximations for $\widehat{f}(k,l)$ are obtained by first performing a 2D-fft on $(p_{kl})_{k,l=0}^{N-1}$ (where p_{kl} denotes the (k,l)th coefficient of p; in particular, $p_{kl} = 0$, $(k,l) \notin \Lambda_+$). This computes $p(e^{\frac{2\pi ki}{N}}, e^{\frac{2\pi li}{N}})$, $k,l = 0, \ldots, N-1$. Next, $f(e^{\frac{2\pi ki}{N}}, e^{\frac{2\pi li}{N}})$, $k,l = 0, \ldots, N-1$, are computed, and a 2D-ifft is applied to those values, yielding approximations for $\widehat{f}(k,l)$, $k,l = 0, \ldots, N-1$. Unless specified otherwise, we took $N = 64$. Since the Fourier coefficients $\widehat{f}(k,l)$ are supposed to coincide with the given data c_{kl}, the *goodness of the fit* may now be quantified by

$$\sqrt{\sum_{(k,l) \in \Lambda_+} |\widehat{f}(k,l) - c_{kl}|^2} \ .$$

The computed goodness of fit quantities may be found in the columns "Newton fit" and "barrier fit", for the Newton and the barrier algorithm, respectively.

We now present our results.

Data set	n	m	Newton fit	Newton #iter	distance between solutions	barrier fit	barrier #iter
A	2	2	$8.3274e-15$	26	$7.3371e-11$	$7.7233e-11$	154
B	2	2	$1.3159e-11$	26	$6.9607e-11$	$6.0795e-11$	203
C	2	3	$1.6257e-10$	52	$1.0445e-10$	$2.0332e-10$	377
D	2	3	$2.5014e-13$	34	$1.2866e-10$	$1.2473e-10$	221
E	2	4	$3.2353e-14$	42	$3.5122e-11$	$4.222e-11$	298
F	2	4	$3.5427e-14$	42	$9.0617e-11$	$9.1178e-11$	317
G	2	5	$1.9767e-14$	50	$5.6561e-11$	$6.5615e-11$	352
H	2	5	$3.1553e-13$	51	$1.1928e-10$	$1.1189e-10$	442
I	3	2	$4.1458e-15$	43	$4.8987e-11$	$4.3441e-11$	249
J	3	2	$6.904e-15$	43	$9.8957e-11$	$9.0654e-11$	237
K	3	3	$4.6912e-15$	58	$9.827e-11$	$8.2036e-11$	565
L	3	3	$1.2545e-05$	59	$7.4691e-11$	$1.2545e-05$	424
M	3	4	$8.7797e-07$	74	$6.1487e-11$	$8.7796e-07$	735
N	3	4	$5.6098e-13$	74	$1.0232e-10$	$9.7676e-11$	413
O	3	5	$2.4368e-13$	88	$5.1368e-11$	$5.3247e-11$	569
P	3	5	$3.5533e-13$	70	$6.3883e-11$	$6.5833e-11$	526
Q	4	2	$7.9025e-14$	42	$5.992e-11$	$5.5675e-11$	328
R	4	2	$4.8735e-15$	42	$2.8133e-11$	$3.3248e-11$	271
S	4	3	$2.2233e-13$	73	$8.6311e-11$	$9.3553e-11$	487
T	4	3	$1.3002e-14$	74	$9.8415e-11$	$1.1244e-10$	486
U	4	4	$7.0978e-12$	74	$1.0448e-10$	$1.0244e-10$	704
V	4	4	$7.5379e-13$	74	$6.5584e-11$	$6.632e-11$	622
W	4	5	$4.2006e-12$	90	$7.1178e-11$	$7.2428e-11$	829
X	4	5	0.15885	208	not computed	not computed	1000
Y	5	2	$3.0851e-11$	50	$1.6934e-10$	$1.5591e-10$	393
Z	5	2	$7.5054e-15$	63	$8.0711e-11$	$8.814e-11$	412
AA	5	3	0.034452	107	not computed	not computed	1000
BB	5	3	0.18486	584	not computed	not computed	1000
CC	5	4	$1.009e-14$	114	not computed	not computed	1000
DD	5	4	4.7418	1392	not computed	not computed	0
EE	5	5	16.56	457	not computed	not computed	0
FF	5	5	$5.5743e-15$	138	not computed	not computed	1000

Notice that in the majority of the experiments (A-W,Y,Z) both methods reach the same solution, and the Newton method does it in far fewer iterations. Let us look at the other

experiments in further detail.

In experiment X the barrier method does not converge, and is stopped after 1000 iterations. The Newton method does converge to a matrix that satisfies condition (2) in Theorem 2.1, but the matrix is not positive definite (-1.24014 is an eigenvalue). The latter explains the fact that the Fourier coefficients fail to match (notice the goodness of fit of 0.15885). We suspect that in this case no solution exists, though we do not have a proof for it.

In experiment AA the Newton method converges to a matrix that satisfies conditions (1) and (2) in Theorem 2.1, however the goodness of fit of 0.034452 suggests that the Fourier coefficients do not match. Since this would violate Theorem 2.1 we recomputed the Fourier coefficients using grid size $N = 512$, and found a goodness of fit of 5.2901e-8. Thus the Fourier coefficients match, but a better approximation to compute them was necessary.

In experiment BB the Newton method stops after 584 iterations but fails to give a matrix satisfying (2) in Theorem 2.1. Retrying it with another initial condition, the Newton method fails to stop before the maximal number of iterations (1500). The barrier method also fails to converge within the allotted number of iterations (1000). We suspect that in this case no solution exists, though we do not have a proof for it.

In experiments CC and FF the Newton method converges to an answer, while the barrier does not reach the required tolerance of 1e-10 within the allotted 1000 iterations. In experiments DD and EE the semidefinite programming package fails to give a positive definite completion $\Gamma(x)$, and one may assume therefore that none exists. Consequently,

condition (1) in Theorem 2.1 cannot be met and therefore the barrier method is not initiated (0 iterations are recorded). Since in the Newton method we do not check for the existence of a positive definite completion (although we clearly could) it does yield answers, but they have no meaning (and, for instance, fail to match the Fourier coefficients).

§6 Two-variable Riesz-Fejer factorization

As explained in the first paragraph of Section 5, it is not hard to compute numerically the Fourier coefficients of the reciprocal of a trigonometric polynomial. Thus, given a two-variable trigonometric polynomial $f(z,w)$, we may easily check numerically the condition in Theorem 2.2 to see if f allow a stable factorization. Let us illustrate this on an example.

Example 6.1. Let $f(z,w)=\sum_{i=-2}^{2}\sum_{j=-2}^{2} z^i w^j(\sum_{r=0}^{2-|i|}\sum_{s=0}^{2-|j|} 2^{-2(r+s)-|i|-|j|})$. Computing the Fourier coefficients of the reciprocal of f (using Matlab; truncating the Fourier series at index 64), we get:

$$c_{0,0}=1.6125, c_{0,1}=c_{1,0}=-0.6450, c_{0,2}=c_{2,0}=-0.0806, c_{1,-2}=0.0322,$$

$$c_{1,-1}=0.2580, c_{1,1}=0.2580, c_{1,2}=c_{2,1}=0.0322, c_{2,-2}=0.0040,$$

$$c_{2,-1}=0.0322, c_{2,2}=0.0040,$$

where only the first four decimal digits show. In order to check (5.2) (where $n=m=2, q=0$) we compute

$$\begin{pmatrix} c_{0,0} & c_{0,-1} & c_{-1,1} & c_{-1,0} & c_{-1,-1} & c_{-2,1} & c_{-2,0} & c_{-2,-1} \\ c_{0,1} & c_{0,0} & c_{-1,2} & c_{-1,1} & c_{-1,0} & c_{-2,2} & c_{-2,1} & c_{-2,0} \\ c_{1,-1} & c_{1,-2} & c_{0,0} & c_{0,-1} & c_{0,-2} & c_{-1,0} & c_{-1,-1} & c_{-1,-2} \\ c_{1,0} & c_{1,-1} & c_{0,1} & c_{0,0} & c_{0,-1} & c_{-1,1} & c_{-1,0} & c_{-1,-1} \\ c_{1,1} & c_{1,0} & c_{0,2} & c_{0,1} & c_{0,0} & c_{-1,2} & c_{-1,1} & c_{-1,0} \\ c_{2,-1} & c_{2,-2} & c_{1,0} & c_{1,-1} & c_{1,-2} & c_{0,0} & c_{0,-1} & c_{0,-2} \\ c_{2,0} & c_{2,-1} & c_{1,1} & c_{1,0} & c_{1,-1} & c_{0,1} & c_{0,0} & c_{0,-1} \\ c_{2,1} & c_{2,0} & c_{1,2} & c_{1,1} & c_{1,0} & c_{0,2} & c_{0,1} & c_{0,0} \end{pmatrix}^{-1} =$$

$$\begin{pmatrix} 0.9375 & 0.3750 & 0.0000 & 0.4688 & 0.1875 & 0.0000 & 0.2344 & 0.0938 \\ 0.3750 & 0.9375 & 0.0000 & 0.1875 & 0.4688 & 0.0000 & 0.0938 & 0.2344 \\ 0.0000 & 0.0000 & 0.9375 & 0.4688 & 0.2344 & 0.3750 & 0.1875 & 0.0938 \\ 0.4688 & 0.1875 & 0.4688 & 1.3477 & 0.5625 & 0.1875 & 0.5625 & 0.2344 \\ 0.1875 & 0.4688 & 0.2344 & 0.5625 & 1.1719 & 0.0938 & 0.2344 & 0.4922 \\ 0.0000 & 0.0000 & 0.3750 & 0.1875 & 0.0938 & 0.9375 & 0.4688 & 0.2344 \\ 0.2344 & 0.0938 & 0.1875 & 0.5625 & 0.2344 & 0.4688 & 1.1719 & 0.4922 \\ 0.0938 & 0.2344 & 0.0938 & 0.2344 & 0.4922 & 0.2344 & 0.4922 & 0.9961 \end{pmatrix},$$

which has zeroes in the required positions. When we compute $p(z, w)$ we find indeed an answer close to the exact solution $p(z, w) = \sum_{k,l=0}^{2} 2^{-k-l} z^k w^l$.

Of course, in many cases the test will fail, and thus a stable factorization does not exist.

Example 6.2. Let $f(z, w) = 1 + .3w + .4z + .15zw + .07z/w + .3/w + .4/z + .15/(zw) + .07w/z$. Then we obtain that

$$\begin{pmatrix} c_{0,0} & c_{-1,1} & c_{-1,0} \\ c_{1,-1} & c_{0,0} & c_{0,-1} \\ c_{1,0} & c_{0,1} & c_{0,0} \end{pmatrix}^{-1} = \begin{pmatrix} 2.4422 & 0.7890 & -1.3683 \\ 0.7890 & 2.4422 & -1.0196 \\ -1.3683 & -1.0196 & 2.4422 \end{pmatrix}^{-1},$$

which has a nonzero (2,1) entry (namely -0.0653). Thus $f(z, w)$ does not have a stable factorization.

Of course, in Sections 3 and 4 we have developed algorithms that alter the coefficients $c_{-i,j}$, $i = 1, \ldots, n$, $j = 1, \ldots, m$, so that the corresponding inverse does have the required zeros. It turns out though that this does not yield an answer close to the original trigonometric polynomial. For instance, in Example 6.2 this would correspond to altering the value of $c_{-1,1} = \overline{c_{1,-1}}$ to equal 0.5712. Then, indeed, we do get the required zero in the inverse. The first column of the inverse of Γ equals now

$$[0.7387\ 0.3229\ 0.4229\ 0.2025]^T.$$

So, by Theorem 2.2 we find that the corresponding polynomial is given by

$$p(z, w) = 0.8595 + 0.3757w + 0.4920z + 0.2355zw.$$

The corresponding trigonometric polynomial is

$$1.1774+0.4387/w+0.5113/z+0.2024/zw+.04387w+0.1848w/z+0.5113z+0.1848z/w+0.2024zw.$$

Clearly, this is not a desired factorable approximation of the original trigonometric polynomial. It is the study of ongoing research how to construct more reasonable factorable approximations.

§7 The complex case

Though we only implemented the algorithms when the given coefficients are real, one may without much additional effort also do the complex case (that is, allowing $c_{k,l} \in \mathbb{C}$). In that case, one may use the following simple observation to adjust the algorithms.

Lemma 7.1. *The complex matrix $A = B + iC$, with $B, C \in \mathbb{R}^{q\times q}$, is positive definite (invertible) if and only if the block matrix $\begin{bmatrix} B & C \\ -C & B \end{bmatrix}$ is positive definite (invertible). Moreover, if $A^{-1} = D + iE$ with $D, E \in \mathbb{R}^{q\times q}$, then*

$$\begin{bmatrix} B & C \\ -C & B \end{bmatrix}^{-1} = \begin{bmatrix} D & E \\ -E & D \end{bmatrix}. \tag{7.1}$$

Proof. Let W^* denote the complex conjugate transpose of W and W^T the transpose of W. First note that $A = A^*$ if and only if $B = B^T$ and $C = -C^T$. Moreover, if $C = -C^T$, we get that $y^TCy = (y^TCy)^T = -y^TCy$ and thus $y^TCy = 0$ for all $y \in \mathbb{R}^q$. Let $x \in \mathbb{C}^q$ and write $x = y + iz$ with $y, z \in \mathbb{R}^q$. Then one easily checks that when $A = A^*$ (or equivalently, $B = B^T$ and $C = -C^T$) we have that

$$x^*Ax = \begin{bmatrix} y^T & z^T \end{bmatrix} \begin{bmatrix} B & C \\ -C & B \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix}.$$

But then the equivalence of the positive definiteness of A and $\begin{bmatrix} B & C \\ -C & B \end{bmatrix}$ follows.

Next, if A is invertible and $A^{-1} = D + iE$ with $D, E \in \mathbb{R}^{q\times q}$, it is straightforward to check that (7.1) holds. Conversely, if $\begin{bmatrix} B & C \\ -C & B \end{bmatrix}$ has an inverse $(K_{ij})_{i,j=1,2}$, then

$$\begin{bmatrix} K_{22} & -K_{21} \\ -K_{12} & K_{11} \end{bmatrix} = \begin{bmatrix} 0 & -I \\ I & 0 \end{bmatrix} \begin{bmatrix} B & C \\ -C & B \end{bmatrix}^{-1} \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix} = \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix}$$

and thus $(K_{ij})_{i,j=1,2}$ has to be of the form (7.1). Let now $Z = D + iE$ and one easily checks that $AZ = ZA = I$. □

§8 Conclusions

In this paper we have developed effective numerical algorithms for solving the 2D autoregressive filter problem based on the characterizations developed in [15]. These characterizations involve a positive definite matrix with a submatrix whose inverse contains a zero block. Algorithms based on the Newton method and the log barrier method were implemented using Matlab. An array of experiments have shown that the Newton method is much faster than the log barrier method, even though in the Newton method there is no guarantee that the final solution lies in the cone of positive definites. Another advantage of the Newton method is that (at least so far) we were not required to compute an initial value that lies in the cone of positive definites. Finally, we have decsribed and implemented a test to check for stable factorability of positive two-variable trigonometric polynomials.

Acknowledgment. We would like to thank Professors Russell Mersereau and Mark Smith for numerous useful discussions on the topic. Finally, we thank the referee for pointing us to references [2], [6], [18] and [24].

References

[1] Farid Alizadeh, Jean-Pierre A. Haeberly, Madhu V. Nayakkankuppam and Michael L. Overton, SDPpack Version 0.8 Beta for Matlab 4.2: Semidefinite Programs, Courant Institute, March 28, 1997 (http://www.cs.nyu.edu/overton/sdppack/sdppack.html).

[2] M. Benedir, "Sufficient conditions for the stability of multidimensional recursive digital filters," Proceedings ICASSP, Vol. 4, 1991, 2885–2888.

[3] S. Boyd, L. El Ghaoui, "Method of Centers for Minimizing Generalized Eigenvalues," Linear Algebra Appl., Vol. 188, 1993, 63–111.

[4] S. Boyd, L. El Ghaoui, E. Feron, V. Balakrishnan, "Linear Matrix Inequalities in System and Control Theory," in: SIAM Studies in Applied Mathematics 15, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1994.

[5] P. J. Brockwell and R. A. Davis, Time Series: Theory and Methods, Springer-Verlag, New York 1987.

[6] V. Buzenac-Setterini and M. Najim, "OLRIV: A new fast adaptive algorithm for rectangular-block Toeplitz systems," IEEE Transactions on Signal Processing, Vol. 48, 2000, 2519–2534.

[7] G. Castro, "Coefficient de Réflexion Généralisés. Extension de Covariance Multidimensionelles et Autres Applications," Ph.D. Thesis, Université de Paris-Sud Centre d'Orsay, 1997.

[8] T. F. Coleman and Y. Li, "On the Convergence of Reflective Newton Methods for Large-Scale Nonlinear Minimization Subject to Bounds," Mathematical Programming, Vol. 67, 1994, 189–224.

[9] T. F. Coleman and Y. Li, "On the Convergence of Reflective Newton Methods for Large-Scale Nonlinear Minimization Subject to Bounds," SIAM J. Optimization, Vol.

6, 1996, 418–445.

[10] R. A. DeCarlo, J. Murray, and R. Sacks, "Multivariate Nyquist theory," Int. J. Contr., Vol. 25, 1977, 657–675.

[11] E. J. Delp, R. L. Kashyap and O. R. Mitchell, "Image Data Compression Using Autoregressive Time Series Models," Pattern Recognition, Vol. 11, 1979, 313–323.

[12] P. Delsarte, Y. Genin, and Y. Kamp, "A simple proof of Rudin's Multivariable Stability Theorem," IEEE Trans ASSP, Vol. 28, 1980, 701–705.

[13] D. E. Dudgeon and R. M. Merserau, Multidimensional Digital Signal Processing, Prentice-Hall, Englewood Cliffs, 1984.

[14] M. P. Ekstrom and J. W Woods, "Two-dimensional spectral factorization with applications in recursive digital filtering," IEEE Trans. Acoustics, Speech and Signal Processing, Vol. 24, 1976, 115–128.

[15] J. S. Geronimo and H. J. Woerdeman, "Two-variable positive extensions and Riesz-Fejer factorization for positive trigonometric polynomials," submitted. Available at http://www.math.wm.edu/~hugo/publ.html.

[16] U. Grenander and G. Szegö, Toeplitz Forms and Their Applications, University of California Press, Berkeley, CA, 1958.

[17] X. Guyon, Random fields on a Network Modeling, Statistics and Applications, Springer Verlag, New York, NY, 1991.

[18] X. Liu and M. Najim, "A two-dimensional fast lattice recursive least squares algorithm," IEEE Trans. Signal Proc., Vol. 44, No. 10, 1996, 2557–2567.

[19] F. Liu and R. W. Picard, "Periodicity, Directionality, and Randomness: Wold Features for Image Modeling and Retrieval," IEEE Trans. Pattern Anal. Machine Intell. (7), Vol. 18, July 1996, 722–733.

[20] J. Mao and A.K. Jain, "Texture Classification and Segmentation Using Multireso-

lution Simultaneous Autoregressive Models," Pattern Recognition (2), Vol. 25, 1992, 173–188.

[21] Yu. Nesterov and A. Nemirovskii, "Interior-point polynomial algorithms in convex programming," in: SIAM Studies in Applied Mathematics, 13, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1994.

[22] R.M. Picket, "Visual Analysis of Texture in the Detection and Recognition of Objects," in: Picture Processing and Psychpictoris, B.C. Lipkin and A. Rosenfeld, Eds., New York: Academic, 1970, 289–308.

[23] M. G. Strinzis, "Tests of stability of multidimensional filters," IEEE Trans. Circuits and Systems, Vol. 24, no. 8, 1977, 432–437.

[24] H. Youlal, M. Janati-I, and M. Najim, "Two dimensional joint process lattice LMS for adaptive restoration of images," IEEE Trans. Image Processing, Vol. 1, no. 3, 1992, 366–378.