ERROR ANALYSIS OF INCOMING AND OUTGOING SCHEMES FOR THE TRIGONOMETRIC MOMENT PROBLEM

A. Bultheel
K.U.LEUVEN
Afd. Toegepaste Wiskunde
en Programmatie
Celestijnenlaan 200 A
B-3030 Heverlee
BELGIUM

Abstract

The solution of the trigonometric moment problem involves the computation of a (0/n) Laurent-Padé approximant for a positive real function on the complex unit circle. The incoming scheme is equivalent with the recursion for Szegö's orthogonal polynomials, while the outgoing scheme is equivalent to the Schur recursion for contractions of the unit disc. The numerical stability of both algorithms is proved under certain conditions via a backward error analysis.

1. Introduction

The trigonometric moment problem (TMP) is one of the classical moment problems [1-3]. It consists of asking whether a prescribed sequence $\ldots t_{-2}, t_{-1}, t_0 = 1, t_1, t_2, \ldots$ with $t_k + t_{-k}$ real can be represented as

$$t_k = \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{-jk\theta} \, d\mu(\theta) \qquad , \; k = 0, \pm 1, \pm 2, \ldots$$

with $d\mu \geq 0$, a positive measure of the unit circle.
A constructive way to study the problem is to define the linear functional $L(z^{-k}) = t_k$ and the polynomials orthogonal with respect to the inner product $(f,g) = L(f(z)\overline{g(1/\bar{z})})$. If $d\mu$ exists, we may also represent this product as

$$(f,g) = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(e^{j\theta}) \overline{g(e^{j\theta})} \, d\mu(\theta) \; .$$

The corresponding orthonormal system $\phi_0, \phi_1, \ldots$ is the set of Szegö polynomials [4]. Note that ϕ_n depends only upon t_k, $|k| \leq n$, and is related to them in a Padé sense. Namely : if $w(z)$ is the weight function, representing the absolutely continuous part of $d\mu$, i.e. $d\mu = w(e^{j\theta})d\theta$, then $w_n(z) = [\phi_n(z)\overline{\phi_n(1/\bar{z})}]^{-1}$ is a [0/n] Laurent-Padé approximant for $w(z)$ [5]. For the TMP, one continues to study the convergence of $w_n(z)$ (or $\phi_n(z)$) to establish the existence and/or uniqueness of $d\mu$.
We shall not dig deeper into the analysis than strictly necessary. For our purpose we consider the problem as an approximation problem. We suppose $w(e^{j\theta})$ exists and

we study numerical aspects of algorithms constructing the successive approximants w_n (i.e. ϕ_n) of w for n = 0,1,2,... . As we mentioned, this is a special case of a Laurent-Padé approximation problem [8], but it is also related to Padé-type approximation [6], 2 point Padé approximation, T-fractions [7] least squares approximation [2] etc. . One of the classical applications of the TMP is in the theory of linear prediction of stationary stochastic processes [2,9]. In practical situations, the algorithms we are going to consider compute the parameters of a digital filter which generates the linear predictor. (i.e. ϕ_n) [10,11]. It is from this point of view that we shall look at the algorithms. We place ourselves in the simplest situation and formulate the algorithms in that case. Suppose that $d\mu = wd\theta$ exists and that $t_0 = 1$ and $t_k = t_{-k}$ is real for all k, then $\phi_0 = 1$ and the Szegö polynomials have real coefficients. Denote by $q_n(z) = \phi_n(z).r_n^{1/2}$ the monic version of the orthonormal Szegö polynomial $\phi_n(z)$. I.e.

$$q_n(z) = \varrho_{0n} + \varrho_{1n}\, z + \ldots + \varrho_{nn}\, z^n = \varrho_n^T[1\ z \ldots z^n]^T$$

with $\varrho_{nn} = 1$ and $\varrho_n^T = [\varrho_{0n} \ldots \varrho_{nn}]$. We define the reciprocal of a real polynomial of degree n as $\hat{q}_n(z) = z^n\, q(1/z)$ and for the corresponding coefficient vector $\hat{\varrho}_n^T = [\varrho_{nn} \ldots \varrho_{0n}]$.

The first algorithm which we call *incoming algorithm* is the recursion scheme for the Szegö polynomials $q_n(z)$:

$$\varrho_0 = r_0 = 1$$

for i = 0,1,2,...,n-1

$$p_i = [t_1 \ldots t_{i+1}]\varrho_i \quad ; \quad \rho_{i+1} = p_i\, r_i^{-1}$$

$$\varrho_{i+1}^T = [0 \quad \varrho_i^T] - [\hat{\varrho}_i^T \quad 0]\rho_{i+1}$$

$$r_{i+1} = r_i(1-\rho_{i+1}^2) \ .$$

This may also be written as

$$\begin{bmatrix} q_{i+1}(z) \\ \hat{q}_{i+1}(z) \end{bmatrix} = \begin{bmatrix} 1 & -\rho_{i+1} \\ -\rho_{i+1} & 1 \end{bmatrix} \begin{bmatrix} z & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} q_i(z) \\ \hat{q}_i(z) \end{bmatrix} = \prod_{j=1}^{i+1} \begin{bmatrix} z & -\rho_j \\ -z\rho_j & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} .$$

Given the ρ_i parameters, (i.e. *reflection coefficients*), the latter recursion can be seen as a method to evaluate $q_{i+1}(z)$. It can be implemented efficiently as a ladder filter [10,11] such that it becomes a scheme competitive with the classical Horner scheme to evaluate $q_{i+1}(z)$. There are some other advantages such as low sensitivity etc. so that we shall use the incoming scheme *not* as an algorithm to evaluate the coefficients ϱ_i but as an algorithm to compute the set of reflection coefficients ρ_i.

Another algorithm which computes the reflection coefficients is the algorithm of Schur [12], also given as one of the applications of the TMP in [2]. We call it the

outgoing algorithm. If we set $a_0(z) = \sum_0^\infty t_k z^k$ and $b_0(z) = \sum_1^\infty t_k z^k$ while recursively

$$\begin{bmatrix} a_{i+1}(z) \\ b_{i+1}(z) \end{bmatrix} = \begin{bmatrix} 1 & -\rho_{i+1} \\ -\rho_{i+1} & 1 \end{bmatrix} \begin{bmatrix} z & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} a_i(z) \\ b_i(z) \end{bmatrix},$$

then $a_i(z) = r_i z^i + O(z^{i+1})$ and $b_i(z) = p_i z^{i+1} + O(z^{i+2})$.
Of course, in practical situations one cannot use an infinite sum. To formulate the finite version we need cutting operators to delete the first or last element of a vector. They are

$$\Delta = \left[\begin{array}{c|c} 0 & 1 \\ \vdots & \ddots \\ 0 & \end{array}\right] \quad \text{and} \quad \nabla = \left[\begin{array}{c|c} 1 & \\ \ddots & \vdots \\ & 0 \end{array}\right].$$

e_0^T is a unit vector $e_0^T = [1\ 0\ 0\ \ldots]$. The algorithm is :

$$r_0 = 1 \qquad\qquad p_0 = t_1$$
$$A_0^T = [t_0 \ \ldots \ t_n] \qquad\qquad B_0^T = [t_1 \ \ldots \ t_{n+1}]$$

$$\text{for } i = 0,1,\ldots,n-1$$
$$\rho_{i+1} = p_i \, r_i^{-1}$$

$$A_{i+1} = \nabla(A_i - B_i \, \rho_{i+1}) \qquad\qquad B_{i+1} = \Delta(B_i - A_i \, \rho_{i+1})$$
$$r_{i+1} = e_0^T A_{i+1} \qquad\qquad p_{i+1} = e_0^T B_{i+1}$$

We call the first algorithm "incoming" because in every iteration, new information (t_{i+1}) comes in and this is accumulated in Q_i, a vector of increasing length. The second algorithm starts with all the data at the beginning (A_0,B_0) and extracts in every step some information (ρ_i) so that something of the data can be left (cutting operations). The A_i,B_i are outgoing vectors (of decreasing length).

The outgoing algorithm uses two vectors that need updating, but no evaluation of an inner product to find p_i. The incoming algorithm needs only one vector Q_i but n inner products of increasing length to evaluate all p_i. These inner products involve the summation of about n(n+1)/2 products. In the outgoing algorithm we have in the first step n inner products ahead. One product/sum operation is executed for each of these products. In the second step only (n-1) inner products remain and again one product/sum operation is executed for each of them, etc. . This results in the updating of a decreasing length vector. Although the computations involved are not exactly the same after the first step, for the incoming and outgoing algorithms, the main difference is something like reordering the computations.

A similar duality exists between the Gauss and Crout methods for triangular factorization of a matrix or the Gram-Schmidt and modified Gram-Schmidt method for orthogonalisation.

There exists a matrix interpretation for the incoming and outgoing algorithm. If T_n is the symmetric Toeplitz matrix based on the parameters $\{t_0,t_1,\ldots,t_n\}$, U_n a unit upper triangular with columns Q_i and L_n a lower triangular with the A_i as columns (thus with r_i as i-th diagonal element) we have

$$T_n U_n = L_n \quad \text{and} \quad L_n = U_n^{-T} D_n \quad \text{with } D_n = \text{diag}(r_0,\ldots,r_n).$$

For

$$\underline{T}_n = \begin{bmatrix} t_1 & \cdots & t_{n+1} \\ t_0 & \ddots & \vdots \\ \vdots & \ddots & t_1 \\ t_{n-1} & \cdots & t_0 \end{bmatrix}, \quad \underline{L}_n = \begin{bmatrix} \star & & 0 \\ & \ddots & \\ 1 & \cdots & 1 \end{bmatrix} \quad \text{and} \quad \underline{U}_n = \begin{bmatrix} \cdots & \boxed{B_i} & \cdots \\ & \ddots & \\ 0 & & \end{bmatrix}$$

we have

$$\underline{T}_n \underline{L}_n = \underline{U}_n .$$

The algorithms given above are special forms of $O(n^2)$ factorization algorithms for general Toeplitz matrices (see [13] and the references therein). These algorithms are known to be unstable in general situations [14,15]. However the special structure of the problem does not only give symmetric forms of the algorithms but the existence of $w(e^{j\theta})$ also seems to have important implications for the numerical stability. Indeed this existence implies the $T_n \geq 0$ for $n = 0,1,\ldots$ or equivalently all $|\rho_i| \leq 1$, and thus r_i is decreasing but nonnegative. The latter is also equivalent with all the zeros of $q_n(z)$ being inside the closed unit disc.

The previous algorithms thus compute the Cholesky factors of T_n or T_n^{-1}. The Cholesky algorithm is known to be stable, without pivoting [16]. In practical applications incoming and outgoing algorithms have been used frequently and no stability problems have been reported. However, a theoretical justification by an error analysis in the sense of Wilkinson [16] has not been given, except for [17]. In this paper we try to answer the question: If we want to find the set of parameters $\{\rho_1,\rho_2,\ldots,\rho_n\}$, given $\{t_0 = 1,t_1,\ldots,t_n\}$ is there a numerical preference for one of both, the incoming or outgoing algorithm. This paper is related to the one of Graves-Morris and of L. Wuytack in these proceedings.

In the next section we introduce the notions of stability and conditioning and a criticism on the error analysis performed in [17]. Section 3 and 4 give a backward error analysis for the incoming and outgoing scheme.

2. Conditioning, Stability, Error Analysis

In the next sections we shall do a backward error analysis in the sense of

Wilkinson [16] and conclude from it that the incoming and outgoing algorithms are stable [18], provided the condition number remains bounded. We recall briefly these definitions and introduce some notations.

For our problem, the input data are vectors $t_n = \{t_0 = 1, t_1, \ldots, t_n\}$ such that the symmetric Toeplitz matrices $T_0 = 1, T_1, \ldots, T_n$ that are constructed from these parameters are all positive semi-definite. The result is a vector of reflection coefficients $p_n = \{\rho_1, \rho_2, \ldots, \rho_n\}$ with $|\rho_j| \leq 1$. The *condition number* is a measure for the sensitivity of p_n w.r.t. small perturbations of t_n. Note that our condition number is not the "spectral sensitivity" of the reflection coefficients [11, p. 234] which is the sensitivity of $w_n(e^{j\theta})$ w.r.t. changes in p_n. Neither is it the classical condition number of T_n, because this measures the sensitivity of Q_n or A_n w.r.t. changes (possibly non Toeplitz) in T_n. The latter condition number is $\kappa(T_i) = \max_j \lambda_j(T_i) / \min_j \lambda_j(T_i)$, where $\lambda_j(T_i), j = 0,1,\ldots,i$, are the eigenvalues of T_i. It becomes infinite if T_i is singular. Although in that situation, both algorithms break down, a solution theoretically exists. I.e. if T_i is singular, then $r_i = 0$, $A_i = B_i = 0$, $|\rho_i| = 1$ and $\rho_j = 0$ for $j > i$. The approximation is exact : $w_i = w$. Thus for T_i close to a singular matrix, the solution is not even close to non existence, which usually is the same as ill conditioning. To make the algorithms reliable we should introduce a test to check whether $r_i = 0$ and eventually stop the algorithm. In practical programs we shall replace it by the test $r_i < r^i$ for some $r < 1$ (see further). We do not propose a condition number for our problem and leave its definition as an open problem, but we remember from the foregoing that it is certainly not $\kappa(T_n)$. $\kappa(T_n)$ is only an upperbound for the condition of $T_n \rightarrow Q_n$.

Using the definitions of de Jong [18], it is sufficient to prove forward or backward stability to establish the numerical stability of the method. However, the forward error usually contains the condition number as a factor. Since we have no plausible condition number at this moment, we prefer to do a backward error analysis because this yields more information about the stability of the algorithm. For our algorithms, backward stability means the following. Suppose $p_n = f(t_n)$ is the exact relation between p_n and t_n. If the numerical implementation of f is $\bar{f}$, giving $\bar{p}_n = \bar{f}(t_n)$ and if $\bar{p}_n = f(\bar{t}_n)$ then $\bar{f}$ is called backward stable if $E_n = \bar{t}_n - t_n$ can be made as small as we like by increasing the machine precision ε, and this for all possible t_n in the set of input data. It is clear from this that we cannot allow input vectors t_n such that r_i becomes arbitrary close to 0, because then $\kappa(T_i)$ is arbitrary large, causing uncontrolable errors in Q_i or A_i. This justifies why we shall only prove stability for a restricted set of input data. I.e. the set of t_n such that $\kappa(T_n)$ remains bounded. This means that we may suppose that $|\rho_i| \leq \rho < 1$ or $r_i = \prod_{j=1}^{i} (1-\rho_j^2) \geq r^i > 0$ (with $r = 1-\rho^2$) for all $i = 0,1,\ldots,n-1$ and in the last step either $|\rho_n| = 1$ (and $r_n = 0$) or $|\rho_n| \leq \rho$. r should be "reasonably" large compared with n and ε. In this case we may replace the stop criterion $r_i = 0$ by $r_i < r^i$.

This is reasonable because r_i has the meaning of a least squares error [2] and thus we stop if the approximation is good enough. Our criterion means that w_n stays reasonably far away from w, and if it converges, it may not be slow in the last steps. The following property gives an asymptotic bound for $\kappa(T_n)$:

property 2.1

The spectral condition number $\kappa(T_n)$ is non decreasing with n and if $\lim_{n\to\infty} w_n \neq w$ then $\kappa(T_n)$ is bounded by ess $\sup_\theta w(e^{j\theta})$/ess $\inf_\theta w(e^{j\theta})$. [2, p. 64].

An error analysis for the Durbin algorithm (this is almost the same as our incoming algorithm) was done in [17]. The outcome was that the algorithm is comparable with the Cholesky method which is known to be stable [16]. However, the error analisys which was done there has some drawbacks. 1° The statement is based on an error estimate of Q_n (forward error). 2° The result was derived by a computation of the residual of the linear system of Yule-Walker equations. This is a kind of backward error analysis, but it does not respect the Toeplitz structure of the problem (the t_k in the right hand side have backward errors but the same t_k in the left hand side have not). 3° The condition number $\kappa(T_n)$ is not the correct one for our purposes. 4° We are interested in the computation of p_n and not of Q_n. This justifies the error analysis we are going to describe in the next sections.

3. Error analysis for the incoming algorithm

As described in the previous section we compute $\bar{p}_n = \bar{f}(t_n)$ or componentwise : $\bar{p}_i = \bar{f}_i(t_i)$, and we are looking for $\bar{t}_n = t_n + E_n$ such that $\bar{p}_n = f(\bar{t}_n)$. The analysis is recursive. Suppose we know $E_i = [\varepsilon_0, \varepsilon_1, \ldots, \varepsilon_i]^T$ then we try to find ε_{i+1} such that $\bar{\rho}_{i+1} = f_{i+1}(\bar{t}_i, t_{i+1}+\varepsilon_{i+1})$.

Define $u_i = \nabla Q_i$ and let fl denote the floating point result obtained with the rules of Wilkinson [16] and let ε be the machine precision.
We have for step i (fl_2 [16] simplifies the analysis a bit)

$$\bar{\rho}_{i+1} = fl\left(\frac{fl_2(t_{i+1} + [t_1 \ldots t_i]\bar{u}_i)}{fl(\bar{r}_{i-1}.fl(1 - fl(\bar{\rho}_i^2)))}\right)$$

$$\bar{u}_i = \nabla\bar{Q}_i = fl([0\ \bar{u}_{i-1}^T]^T - fl([1\ \bar{u}_{i-1}^T]^T\bar{\rho}_i)).$$

The definition of ε_{i+1} is given via the equality

$$\bar{\rho}_{i+1} = \frac{t_{i+1} + \varepsilon_{i+1} + [\bar{t}_1 \ldots \bar{t}_i]u_i'}{r_{i-1}'(1 - \bar{\rho}_i^2)}.$$

The quantities with a prime represent the exact values that would have been obtained if $\bar{t}_n$ was used as the input. Note $\bar{\rho}_i = \rho_i'$ by definition of $\bar{t}_n$.

Using the rules of floating point computations, we can bring the first form into the second and thus find that

$$\varepsilon_{i+1} = \varepsilon^{(1)} + \varepsilon^{(2)} + \varepsilon^{(3)},$$

$\varepsilon^{(1)}$ originates from the error in the denominator : $|\varepsilon^{(1)}| \leqslant 3i\ \bar{r}_{i-1}\ \varepsilon$

$\varepsilon^{(2)}$ comes from the inner product and division : $|\varepsilon^{(2)}| \leqslant 2\ \bar{r}_{i+1}\ \varepsilon$

$\varepsilon^{(3)}$ accumulates the errors made in previous operations :

$\varepsilon^{(3)} = [\bar{t}_1 \ \ldots\ \bar{t}_i](\bar{u}_i - u_i') - [\varepsilon_1 \ \ldots\ \varepsilon_i]\bar{u}_i$, thus with $y_j = \bar{u}_i - u_i'$:

$|\varepsilon^{(3)}| < \|y_i\|_1 + \|E_i\|_\infty \cdot \|\bar{u}_i\|_1$.

It was shown in [17] that $\|\bar{u}_i\|_1 \leqslant \prod_1^i (1+|\bar{\rho}_i|) - 1 < 2^i - 1$. Bounding the first term in $\varepsilon^{(3)}$ involves a kind of forward error analysis for the evaluation of u_i, given p_{i-1}. To find a bound for $\|y_i\|_1$ we do an error analysis on the fl-expression for $\bar{u}_i$, given above and write $\bar{u}_i$ as

$$\bar{u}_i = [0\,(u_{i-1}')^T]^T - \bar{\rho}_i[1\,(\hat{u}_{i-1}')^T]^T + y_i = u_i' + y_i .$$

We find (only first order terms)

$$\|y_i\|_1 \leqslant \varepsilon\|u_i'\|_1 + \|y_{i-1}\|_1(1-\bar{\rho}_i) + \bar{\rho}_i\varepsilon\|[1\ \bar{u}_{i-1}^T]^T\|_1 .$$

Thus, if $\|y_i\|_1 < \eta_i\ \varepsilon$, then $\eta_i < 2(2^i + \eta_{i-1})$ or $\eta_i < i\ 2^{i+1}$. All this results in a bound for ε_{i+1} viz.

$$|\varepsilon_{i+1}| < (i\ 2^{i+1} + 3i + 2)\varepsilon + \|E_i\|_\infty(2^i - 1) = G(i)\varepsilon.$$

For finite n, G(i), i = 1,2,...,n-1 is bounded and thus, according to its definition, we may conclude that the incoming algorithm is stable.

Although the estimate derived for $|\varepsilon_{i+1}|$ may not be sharp, the analysis is instructive because it shows that 1° The factor 2^{i+1} mainly comes from the growth of the vector u_i. If all $\rho_j \approx -1$, then this bound is closely followed. This rapid growth is associated with the ill-conditioning of the problem $t_i \to u_i$. 2° The evaluation of the inner product in fl_2 is recommended because possibly large numbers $(\bar{u}_i)$ are used to find a small number $(\bar{p}_{i+1})$. Note also that a scaling of u_i is of no use because this factor would also appear as a multiplier of t_{i+1} in the numerator of ρ_{i+1}.

4. Error analysis for the outgoing algorithm

The situation for the outgoing algorithm is inverted. Here the components of A_i and B_i will be decreasing in magnitude, together with $r_i^{1/2}$. We have the following lemma's (see [15]) :

Lemma 4.1 : Let $T = L\,L^T$ be the lower-upper and $T = U\,U^T$ the upper-lower Cholesky factorization of a Toëplitz matrix $T>0$, then $|L_{ji}| \leqslant L_{ii}$, $\forall j \geqslant i$, $i = 0,1,\ldots$ and $|U_{ij}| \leqslant U_{jj}$, $\forall i \leqslant j$, $j = 0,1,\ldots$.

Lemma 4.2 : With the functions $a_i(z)$ and $b_i(z)$ as defined in section 1, we have that $c_i(z) = b_i(z)/a_i(z)$ is a Schur function (i.e. analytic and contractive on the unit disc). Its Taylor series coefficients are bounded by 1 in magnitude.

The first lemma and the matrix interpretation of the outgoing scheme gives bounds for the elements in the vectors A_i. If $A_i^T = [a_{0i} \ \dots \ a_{n-i,i}]$ then $|a_{ij}| < r_i^{1/2}$. If we set $B_i^T = [b_{0i} \ \dots \ b_{n-i,i}]$ and find these coefficients from $b_i(z) = c_i(z)a_i(z)$ we may bound them by $|b_{ij}| \leq (j+1)r_i^{1/2}$.

If we write out the computations to obtain $\bar{\rho}_{i+1}$ we obtain

$$\bar{\rho}_{i+1} = \mathrm{fl}\left(\frac{b_{i,0} - \mathrm{fl}\left(\sum_{j=1}^{i} \mathrm{fl}(\bar{\rho}_{i-j+1}\ \bar{a}_{j,i-j})\right)}{\mathrm{fl}(\bar{a}_{0,i-1} - \mathrm{fl}(\bar{b}_{0,i-1}\ \bar{\rho}_i))}\right) .$$

ε_{i+1} is defined by the equality

$$\bar{\rho}_{i+1} = \frac{(b_{i,0}+\varepsilon_{i+1}) - \sum_{j=1}^{i} \bar{\rho}_{i-j+1}\ a'_{j,i-j}}{a'_{0,i-1} - b'_{0,i-1}\ \bar{\rho}_i} ,$$

where again primed quantities are exact values obtained from the input vector $\bar{t}_n$. Now ε_{i+1} has the following composition

$$\varepsilon_{i+1} = \varepsilon^{(1)} + \varepsilon^{(2)} + \varepsilon^{(3)} + \varepsilon^{(4)} .$$

$\varepsilon^{(1)}$ is the error from the division and the evaluation of the denominator :

$$|\varepsilon^{(1)}| < 3|\bar{\rho}_{i+1}|\varepsilon .$$

$\varepsilon^{(2)}$ comes from the evaluation of the numerator :

$$|\varepsilon^{(2)}| < |t_{i+1}|i\varepsilon + |\bar{\rho}_1\ a_{i,0}|(i+1)\varepsilon + |\bar{\rho}_2\ \bar{a}_{i-1,1}|i\varepsilon + \dots + |\bar{\rho}_i\ \bar{a}_{1,i-1}|2\varepsilon$$
$$< (i+5)i\varepsilon/2.$$

$\varepsilon^{(3)} = \sum_{j=1}^{i} \bar{\rho}_{i+1-j}(a'_{j,i-j} - \bar{a}_{j,i-j})$ and $\varepsilon^{(4)} = (\bar{a}_{0,i-1} - a'_{0,i-1}) - \bar{\rho}_i(\bar{b}_{0,i-1} - b'_{0,i-1})$.

Bounding $\varepsilon^{(3)}$ and $\varepsilon^{(4)}$ requires again a forward analysis for the computation of A_j and B_j. From its definition :

$$\bar{A}_j = \nabla\mathrm{fl}(\bar{A}_{j-1} - \mathrm{fl}(\bar{B}_{j-1}\ \bar{\rho}_j)) \stackrel{\mathrm{def}}{=} \nabla(A'_j + y_j) \quad \text{and}$$

$$\bar{B}_j = \Delta\mathrm{fl}(\bar{B}_{j-1} - \mathrm{fl}(\bar{A}_{j-1}\ \bar{\rho}_j)) \stackrel{\mathrm{def}}{=} \Delta(B'_j + z_j) .$$

An error analysis gives, up to higher order terms :

$$\|y_j\|_\infty < \|y_{j-1}\|_\infty + \|z_{j-1}\|_\infty + \varepsilon\|B_{j-1}\|_\infty + \varepsilon\|A'_j\|_\infty$$
$$< \|y_{j-1}\|_\infty + \|z_{j-1}\|_\infty + \varepsilon(i+2).$$

Because of the symmetry we have the same bound for $\|z_j\|_\infty$.

Thus if $\|y_j\|_\infty$ and $\|z_j\|_\infty < \eta_j\ \varepsilon$, then $\eta_j < 2\eta_{j-1} + (i+2)$, or $\eta_j < (2^j-1)(i+2)$. This gives for $\varepsilon^{(3)}$ and $\varepsilon^{(4)}$

$$|\varepsilon^{(3)}| \leq \sum_{j=1}^{i} \|y_{i-j}\|_\infty \leq (2^i-i-1)(i+2)\varepsilon \quad \text{and}$$

$$|\varepsilon^{(4)}| \leq 2\,(2^{i-1}-1)(i+2)\varepsilon\,.$$

This adds up to the following bound for ε_{i+1} :

$$|\varepsilon_{i+1}| \leq \frac{\varepsilon}{2}(i+2)\{2^{i+2}-i-3\} = H(i)\varepsilon\,.$$

Again H(i) can be bounded for finite i and thus is the outgoing algorithm backward stable.

As in the analysis of the incoming algorithm, the main term in the error comes from a forward error for computing (A_n,B_n) from t_n. No fl_2-trick can be done now but the bound for ε_{i+1} does not contain $\|E_i\|_\infty$.

5. Conclusion

In the previous sections we have done a backward error analysis of both the incoming and outgoing algorithm to compute the reflection coefficients from the moments of a positive definite measure. In both cases we had to introduce a kind of forward error which caused the main term in the backward error to be of the type $i\,2^i\,\varepsilon$. However for the incoming algorithm, the bound for ε_{i+1} depended on $\|E_i\|_\infty$ so that potentially this bound can grow faster than the corresponding bound in the outgoing scheme. The forward error contains the condition of the problems associated with it, i.e. evaluation of Q_i or (A_i,B_i) from t_i. That is why we could only prove stability on input data for which $\kappa(T_n)$ remains bounded. $\kappa(T_n)$ is bounded because we required $|\rho_i| \leq \rho < 1$. We said that ρ should be "reasonable" in view of n and ε. We used this fact in the analysis because we supposed that the perturbations were such that the bounds that were valid for the exact values remain valid for the perturbed values. This is exactly what we mean by reasonable.

The fact that the upper bound for $|\varepsilon_{i+1}|$ is potentially larger in the case of the incoming algorithm than in the case of the outgoing one is no proof that the outgoing algorithm is better than the incoming one. However, because the components of Q_i are growing and simultaneously have to produce the numbers p_i that are decreasing we are not in a numerically favorable situation. In the outgoing algorithm, all numbers are decreasing together with $r_i^{1/2}$. Thus intuitively and partly because of the analysis, we should recommend the outgoing algorithm above the incoming one. Numerical experiments have not justified this conjecture, unless $\kappa(T_n)$ was small.

Finally, notice that we have used certain bounds in our analysis that were only true if all $T_j > 0$, $j = 0,1,\ldots$. Thus a generalization of this analysis to arbitrary Toeplitz matrices (the incoming and outgoing algorithms for the Padé problem) is more than just an asymmetric version of the foregoing. The definition of a suitable condition number for our problem and for the general Padé problem, and also the deter-

mination of situations where recursive methods are stable in a general Padé context is still an open problem.

References

[1] N.I. Akhiezer, The classical moment problem, Oliver and Boyd, Edinburgh, London, 1965.

[2] U. Grenander, G. Szegö, Toeplitz forms and their applications, University of California Press, Berkeley, 1958.

[3] H.J. Landau, The classical moment problem : Hilbertian proofs, Journ. of Funct. Anal. 38, (1980), pp. 255-272.

[4] G. Szegö, Orthogonal polynomials, A.M.S. Colloquium publ. XXIII, AMS Providence, Rhode Island, 1939.

[5] W.B. Gragg, Laurent-, Fourier- and Chebyshev-Padé tables, in E.B. Saff, R.S. Varga (eds.), Padé and rational approximation, Theory and applications, Academic Press, New York, 1977, pp. 61-72.

[6] C. Brezinski, Padé type approximation and general orthogonal polynomials, Birkhauser Verlag, Basel, 1980.

[7] W.J. Thorn, Two-point Padé tables, T-fractions and sequences of Schur, in E.B. Saff, R.S. Varga (op. cit.), pp. 215-226.

[8] A. Bultheel, P. Dewilde, On the relation between Padé approximation algorithms and Levinson/Schur recursive methods, in Proc. Conf. Signal Processing, EUSIPCO-80, M. Kunt, F. de Coulon (eds.), North-Holland, 1980, pp. 517-523.

[9] N. Wiener, P. Masani, The prediction theory of multivariate stochastic processes, Acta Math. 98, (1957), pp. 111-150 and 99, (1958), pp. 93-139.

[10] V. Cappellini, A.G. Constantinides, P. Emiliani, Digital filters and their applications, Ac. Press, New York, 1978.

[11] J.D. Markel, A.H. Gray Jr., Linear prediction of speech, Springer-Verlag, Berlin, 1976.

[12] J. Schur, Ueber Polenzreihen die im Innern des Einheitskreises beschränkt sind, Z. Reine Angew. Math. 147, (1917), pp. 205-232 and 148 (1918, pp. 122-145.

[13] A. Bultheel, Recursive algorithms for the matrix Padé problem, Math. of Comp. 35 (151), 1980, pp. 875-892.

[14] L.S. de Jong, Numerical aspects of the recursive realization algorithm, SIAM J. Cont. and Opt. 16, (1978), pp. 646-659.

[15] A. Bultheel, Towards an error analysis of fast Toeplitz factorization, Rept. TW 44, K.U.Leuven, Afd. Toeg. Wisk. & Progr., May 1979.

[16] J.H. Wilkinson, The algebraic eigenvalue problem, Claredon Press, Oxford, 1965.

[17] G. Cybenko, Error analysis of Durbin's algorithm, to appear in SIAM J. for Scient. and Stat. Comp.

[18] L.S. de Jong, Towards a formal definition of numerical stability, Numer. Math. 28, (1977), pp. 211-219.