ON THE RELATION BETWEEN PADE APPROXIMATION ALGORITHMS AND LEVINSON/SCHUR RECURSIVE METHODS.

by A. BULTHEEL

Katholieke Universiteit Leuven
Leuven - Belgium

P. DEWILDE

Delft University of Technology
Delft - The Netherlands

Report TW 45, February 1980

Applied Mathematics and Programming Division
KATHOLIEKE UNIVERSITEIT LEUVEN (BELGIUM)

ABSTRACT

The approximation of a transfer function from a related Taylor or Laurent series (spectral information) is treated by means of recursive computations of elements in a Padé table.
A systematic elimination scheme for setting up the recursive computations is shown and specialized to the computation of a row or a pair of rows in the Padé table. The algorithm appears to have an "incoming" and "outgoing" version. It is shown, via the theory of Laurent-Padé approximation, how the "incoming" version is a generalization of the Levinson algorithm, and the "outgoing" version is a generalization of the Schur algorithm, both well known in Linear Prediction Theory.

I. Introduction

Padé approximation has been used to find approximating realizable transfer functions in a variety of situations [1-5]. Its main advantage is its simplicity of computation (one has only to compute the solution of a simple system of linear equations with a special structure). Its main drawback is the lack of guaranteed stability so that a separate Lehmer-Schur test is required to test the result and no guarantee of convergence is available. Often, one is willing to restrict the filter to be of autoregressive type, in which case the Szegö-Levinson algorithm [8,7,6] has become a standard procedure. The stability here is guaranteed and controlled by means of the magnitude of the reflection coefficients.

In the paper we show how a Levinson-Schur type recursion is available in the general Padé-situation, and how it can be specialized to the Levinson case by reinterpreting the algorithm for Laurent-Padé approximations.

In par. 2 we discuss the Levinson (incoming) algorithm as applied to Padé row computations. In par. 3 we deduce the corresponding Schur (outgoing) version of the algorithm. In par. 4 we show how the algorithms apply to the Laurent-Padé approximation problem, and in par. 5 we show how the Levinson-Schur algorithm is a special case of the theory developed. Finaly in par. 6 we give the physical interpretation for the recursions and indicate how the ARMA filtering problem can be treated.

Most of the individual results of the paper may be found scattered in the literature. This paper is an attempt to bring them together in a systematic way,and to classify the relationship between two important algorithms in signal processing. Numerical examples are also included.

2. Row Padé Recursions : a Review

In this section we quickly review a systematic procedure for producing recursive Padé approximations, and specialize to the case which will prove to be a generalization of the Levinson scheme.

Let be given a formal power series :

$$f(z) = f_0 + f_1 z + f_2 z^2 + \ldots$$

Define $[m/n] \triangleq P^{(m/n)}(z)/Q^{(m/n)}(z)$ with $P^{(m/n)}(z)$ and $Q^{(m/n)}(z)$ relative prime polynomials of degree m and n respectively. [m/n] is called a Padé approximant for f if their formal series expansions in z agree through as many terms as possible (at least m+n+1). So we must have

$$f(z)Q^{(m/n)}(z) - P^{(m/n)}(z) = r_0^{(m/n)} z^{m+n+1} + r_1^{(m/n)} z^{m+n+2} + \ldots$$

Let

$$P^{(m/n)}(z) \triangleq p_0^{(m/n)} + p_1^{(m/n)} z + \ldots p_m^{(m/n)} z^m$$

$$Q^{(m/n)}(z) \triangleq q_0^{(m/n)} + q_1^{(m/n)} z + \ldots + q_n^{(m/n)} z^n$$

then the following system of Padé equations is obtained

$$\sum_{j=0}^{n} f_{i-j}\, q_j^{(m/n)} = \begin{cases} p_i^{(m/n)} & \text{for } i = 0, \ldots, m \\ 0 & \text{for } i = m+1, \ldots, m+n \\ r_{i-(m+n+1)}^{(m/n)} & \text{for } i = m+n+1, \ldots \end{cases} \qquad (2.1)$$

Define $f_k = 0$ for $k < 0$.

Let T(k,i,j) denote a Toeplitz matrix based on the coefficients $\{f_k\}$ as follows :

$$T(k,i,j) \triangleq \begin{bmatrix} f_k & f_{k-1} & \cdots & f_{k-i} \\ f_{k+1} & & & \\ \vdots & & & \\ f_{k+j} & & \cdots & f_{k+j-i} \end{bmatrix} \qquad (2.2)$$

and set :

$$\mathbb{Q}^{(m/n)} = [q_0^{(m/n)} \ \ldots \ q_n^{(m/n)}]^T$$
$$\mathbb{P}^{(m/n)} = [p_0^{(m/n)} \ \ldots \ p_n^{(m/n)}]^T$$
$$\mathbb{R}^{(m/n)} = [r_0^{(m/n)} \ r_1^{(m/n)} \ \ldots \ ]^T$$

The system (2.1) becomes :

$$\begin{bmatrix} T(0,n,m) \\ T(m+1,n,n-1) \\ T(m+n+1,n,\infty) \end{bmatrix} \mathbb{Q}^{(m/n)} = \begin{bmatrix} \mathbb{P}^{(m/n)} \\ 0 \\ \mathbb{R}^{(m/n)} \end{bmatrix} \qquad (2.3)$$

Note that in (2.3) there is still a constant multiplier (normalization) by which both sides may be multiplied. We shall use two normalizations : $q_n^{(m/n)} = 1$ or $q_0^{(m/n)} = 1$. In the latter case $\mathbb{Q}^{(m/n)}$ is determined by :

$$T(m+1,n-1,n-1)[q_1^{(m/n} \ \ldots \ q_n^{(m/n)}]^T = -[f_{m-n+1} \ f_{m-n+2} \ \ldots \ f_m]^T \qquad (2.4)$$

In any event, by adding a row and a column to (2.4) the following equivalent forms to (2.4) are obtained :

$$\begin{cases} T(m+1,n,n)\mathbb{Q}^{(m/n)} = [0 \ \ldots \ 0 \ r_0^{(m,n)}]^T & (2.5) \\ T(m,n,n)\mathbb{Q}^{(m/n)} = [p_n^{(m,n)} \ 0 \ \ldots \ 0]^T & (2.6) \end{cases}$$

If all Padé approximants for f exist, then they can be ordered in a so-called Padé-table with the element [m/n] as its $(m,n)^{th}$ entry [18].
There is a systematic way to construct recursive solutions for (2.5) and (2.6), see e.g. [14,15]. We specialize to the computation of a row of the Padé-table and illustrate the technique for this special case which will appear later to yield the Levinson-Schur algorithm.

Fix m and m+1, selecting rows m and m+1 in the Padé-table. Introduce the following abbreviations :

$$T_n \triangleq T(m+1,n,n),\ \underline{T}_n \triangleq T(m,n,n),\ \bar{T}_n \triangleq T(m+2,n,n),$$
$$\mathbb{Q}_n \triangleq \mathbb{Q}^{(m/n)} \quad \text{with } \underline{\text{monic normalization}} \quad q_n^{(m/n)} = 1$$
$$\bar{\mathbb{Q}}_n \triangleq \mathbb{Q}^{(m+1/n)} \quad \text{with } \underline{\text{comonic normalization}} \quad q_0^{(m/n)} = 1$$
$$\mathbb{P}_n \triangleq \mathbb{P}^{(m/n)},\ p_n \triangleq p_m^{(m/n)},\ \bar{\mathbb{P}}_n \triangleq \mathbb{P}^{(m+1/n)},\ \bar{p}_n \triangleq p_{m+1}^{(m+1/n)},$$
$$\mathbb{R}_n \triangleq \mathbb{R}^{(m/n)},\ r_n \triangleq r_0^{(m/n)},\ r'_n \triangleq r_1^{(m/n)}$$
$$\bar{\mathbb{R}}_n \triangleq \mathbb{R}^{(m+1/n)},\ \bar{r}_n \triangleq r_0^{(m+1/n)},\ r'_n \triangleq r_1^{(m+1/n)}$$

Note that upper bar normalization is different from the no bar normalization for reasons which will soon become apparent.

We first attempt to compute $\mathbb{Q}_{n+1}$ from $\mathbb{Q}_n$ and $\mathbb{Q}_{n-1}$. Returning to (2.3) it is easy to see that :

$$T_{n+1}\left[\begin{array}{c|c|c} 0 & 0 & \mathbb{Q}_n \\ & \mathbb{Q}_{n-1} & \\ \mathbb{Q}_n & 0 & 0 \end{array}\right] = \left[\begin{array}{c|c|c} p_n & p_{n-1} & 0 \\ 0 & 0 & \vdots \\ \vdots & \vdots & 0 \\ 0 & r_{n-1} & r_n \\ r_n & r'_{n-1} & r'_n \end{array}\right],$$

and since $\mathbb{Q}_{n+1}$ is defined by

$$T_{n+1}\,\mathbb{Q}_{n+1} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ r_{n+1} \end{bmatrix},$$

we see that a three terms recursion for $\mathbb{Q}_{n+1}$ must exist by elimination, keeping in mind that $\mathbb{Q}_n$ is monic :

$$\mathbb{Q}_{i+1} = \begin{bmatrix} 0 \\ \\ \mathbb{Q}_i \end{bmatrix} - \left(\begin{bmatrix} 0 \\ \mathbb{Q}_{i-1} \\ 0 \end{bmatrix} - \begin{bmatrix} \mathbb{Q}_i \\ \\ 0 \end{bmatrix} \frac{r_{i-1}}{r_i} \right) \frac{p_i}{p_{i-1}} \qquad (2.7)$$

$$(i = 0,1,\ldots)$$

with initial conditions :

$$\mathbb{Q}_{-1} = [\phi] \text{ , the empty vector}$$

$$p_{-1} = 1,\ r_{-1} = -1,\ Q_0 = 1,\ p_0 = f_m,\ r_0 = f_{m+1}\ .$$

In this "incoming" scheme, the p_i and r_i needed in the recursion are given by :

$$\begin{bmatrix} p_i \\ \\ r_i \end{bmatrix} = \begin{bmatrix} f_m & \cdots & f_{m-i} \\ & & \\ f_{m+i+1} & f_{m+i} & \cdots & f_{m+1} \end{bmatrix} \mathbb{Q}_i\ .$$

This scheme is the Trench-Zohar [10-11] algorithm for the triangular factorization of a Toeplitz Matrix.
It follows immediately from (2.3) that an upper-lower triangular factorization is obtained

$$T_n \begin{bmatrix} 1 & \star & \cdots & \begin{bmatrix} \mathbb{Q}_i \end{bmatrix} & \star \\ & 1 & \ddots & & \\ & & \ddots & & \\ 0 & & & & 1 \end{bmatrix} = \begin{bmatrix} r_0 & & & 0 \\ \star & \ddots & & \\ \vdots & & \begin{bmatrix} \mathbb{Z}_i \end{bmatrix} & \\ \star & & & r_n \end{bmatrix} \qquad (2.8)$$

where $\mathbb{Q}_i$ is already defined and $\mathbb{Z}_i$ is the "outgoing" vector defined by (2.8). See next paragraph for the significance of the outgoing scheme. As the length of $\mathbb{Q}_i$ increases, the length of $\mathbb{Z}_i$ decreases and it will be shown that knowledge of the $\mathbb{Z}_i$ is sufficient for the determination of the recursion coefficients. Another factorization is :

$$T_n \begin{bmatrix} \star & & 0 \\ & \begin{bmatrix} \mathbb{Q}_i \end{bmatrix} & \\ 1 & & 1 \end{bmatrix} = \begin{bmatrix} p_n & \star & \begin{bmatrix} \mathbb{S}_i \end{bmatrix} & \star \\ & \ddots & & \\ 0 & & & p_0 \end{bmatrix} \qquad (2.9)$$

Replacing m by m+1 and upper-baring all quantities, a similar recursion is available for $\bar{\mathbb{Q}}_i$:

$$\bar{T}_n \begin{bmatrix} 1 & \cdots & \begin{bmatrix} \bar{\mathbb{Q}}_i \end{bmatrix} & \cdots & 1 \\ & \ddots & & & \vdots \\ & & \ddots & & \\ 0 & & & \ddots & \\ & & & & \star \end{bmatrix} = \begin{bmatrix} \bar{r}_o & & & 0 \\ & \ddots & & \\ & & \ddots & \\ \star & & \begin{bmatrix} \bar{\mathbb{Z}}_i \end{bmatrix} & \\ & & & \bar{r}_n \end{bmatrix} \tag{2.10}$$

and

$$T_n \begin{bmatrix} 1 & & & 0 \\ & \ddots & & \\ & & \begin{bmatrix} \bar{\mathbb{Q}}_i \end{bmatrix} & \\ & & & \ddots \\ \star & & & 1 \end{bmatrix} = \begin{bmatrix} \bar{p}_n & & \begin{bmatrix} \bar{\mathbb{S}}_i \end{bmatrix} & \star \\ & \ddots & & \\ & & \ddots & \\ 0 & & & \bar{p}_0 \end{bmatrix} \tag{2.11}$$

The change in normalization assures here a lower triangular form with ones on the main diagonal. This is the reason for the different normalization of the upper bar quantities.

Instead of looking for a three terms recursion with elements based on one row, one can look for recursions in a sawtooth fashion whereby the sequence of computation goes as follows :

$$\underbrace{\left.\begin{matrix} \mathbb{Q}_n \\ \\ \bar{\mathbb{Q}}_n \end{matrix}\right\} \Rightarrow \mathbb{Q}_{n+1}}_{\text{STEP 1}}, \quad \underbrace{\left.\begin{matrix} \mathbb{Q}_{n+1} \\ \\ \bar{\mathbb{Q}}_n \end{matrix}\right\} \Rightarrow \bar{\mathbb{Q}}_{n+1}}_{\text{STEP 2}}, \quad \underbrace{\left.\begin{matrix} \mathbb{Q}_{n+1} \\ \\ \bar{\mathbb{Q}}_{n+1} \end{matrix}\right\} \Rightarrow \mathbb{Q}_{n+2}}_{\text{STEP1}}, \text{ etc...}$$

In step 1 we seek to express $\mathbb{Q}_{n+1}$ as a function of $\mathbb{Q}_n$ and $\bar{\mathbb{Q}}_n$ which is obtained by remarking :

$$T_{n+1}\begin{bmatrix} 0 & \bar{\mathbb{Q}}_n \\ \mathbb{Q}_n & 0 \end{bmatrix} = \begin{bmatrix} p_n & \bar{p}_n \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ r_n & \bar{r}_n \end{bmatrix}$$

and in step 2 $\bar{\mathbb{Q}}_{n+1}$ should be computed from $\mathbb{Q}_{n+1}$ and $\bar{\mathbb{Q}}_n$:

$$\bar{T}_{n+1}\begin{bmatrix} \mathbb{Q}_{n+1} & \bar{\mathbb{Q}}_n \\ & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ r_{n+1} & \bar{r}_{n+1} \\ r'_{n+1} & \bar{r}'_{n+1} \end{bmatrix}$$

elimination gives the recursive scheme :

$$\mathbb{Q}_0 = 1 \ , \ \bar{\mathbb{Q}}_0 = 1$$

$$\mathbb{Q}_{i+1} = \begin{bmatrix} 0 \\ \mathbb{Q}_i \end{bmatrix} - \begin{bmatrix} \bar{\mathbb{Q}}_i \\ 0 \end{bmatrix} \frac{p_i}{\bar{p}_i} \ , \ r_{i+1} = r_i - \frac{\bar{r}_i \ \phi_i}{\bar{p}_i}$$

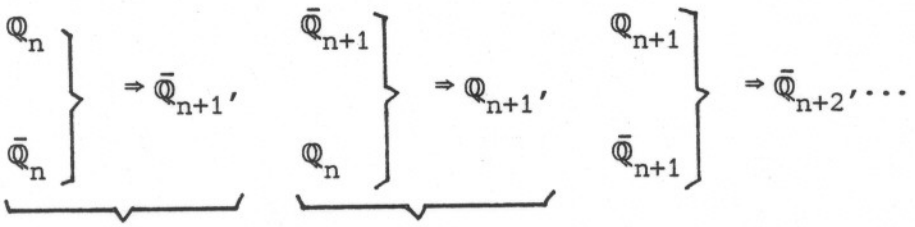
$$\bar{\mathbb{Q}}_{i+1} = \left(\mathbb{Q}_{i+1} - \begin{bmatrix} \bar{\mathbb{Q}}_i \\ 0 \end{bmatrix} \frac{r_{i+1}}{r_i} \right) \nu_i \ , \ \nu_i = \left(q_0^{(m/i+1)} - \frac{r_{i+1}}{r_i} \right)^{-1} \qquad (2.12)$$

$$\bar{p}_{i+1} = - \frac{\bar{p}_i \ r_{i+1}}{\bar{r}_i} \nu_i$$

$$p_i = [f_m \ \dots \ f_{m-i}] \ \mathbb{Q}_i, \ \bar{r}_i = [f_{m+i+2}, \dots, f_{m+2}] \ \bar{\mathbb{Q}}_i$$

where ν_i was to be introduced to preserve the proper normalization on $\bar{\mathbb{Q}}_i$.

A dual sawtooth yields :



and the recursion :

$$\bar{\mathbb{Q}}_0 = 1 \ , \ \mathbb{Q}_0 = 1$$

$$\bar{\mathbb{Q}}_{i+1} = \left(\begin{bmatrix} \bar{\mathbb{Q}}_i \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ \mathbb{Q}_i \end{bmatrix} \frac{\bar{r}_i}{r_i} \right) \ , \ \bar{p}_{i+1} = \bar{p}_i - \frac{p_i \bar{r}_i}{r_i}$$

$$\mathbb{Q}_{i+1} = \left(\bar{\mathbb{Q}}_{i+1} - \begin{bmatrix} 0 \\ \mathbb{Q}_i \end{bmatrix} \frac{\bar{p}_{i+1}}{p_i} \right) \lambda_i \ ,$$

$$\lambda_i = \left(\bar{q}_{i+1}^{(m+1/i+1)} - \frac{\bar{p}_{i+1}}{p_i} \right)^{-1} \qquad (2.13)$$

$$r_{i+1} = - \frac{r_i \bar{p}_{i+1}}{p_i} \lambda_i$$

with p_i and $\bar{r}_i$ as in (2.12) and λ_i such that a comonic normalisation for $\bar{\mathbb{Q}}_i$ is obtained.

A mixed version of (2.12) and (2.13) gives the analog of the Levinson-scheme for Padé-approximations :

$$\mathcal{Q}_0 = \bar{\mathcal{Q}}_0 = 1$$

$$\mathcal{Q}_{i+1} = \begin{bmatrix} 0 \\ \\ \mathcal{Q}_i \end{bmatrix} + \begin{bmatrix} \bar{\mathcal{Q}}_i \\ \\ 0 \end{bmatrix} \alpha_i \ , \ \alpha_i = -\frac{p_i}{\bar{p}_i} \ , \ r_{i+1} = r_i(1-\alpha_i\beta_i)$$

$$\bar{\mathcal{Q}}_{i+1} = \begin{bmatrix} \bar{\mathcal{Q}}_i \\ \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \\ \mathcal{Q}_i \end{bmatrix} \beta_i, \ \beta_i = -\frac{\bar{r}_i}{r_i} \ , \ \bar{p}_{i+1} = \bar{p}_i(1-\alpha_i\beta_i) \qquad (2.14)$$

with p_i and $\bar{r}_i$ as in (2.12). Writing out polynomials one has :

$$\begin{bmatrix} Q_{i+1}(z) \\ \\ \bar{Q}_{i+1}(z) \end{bmatrix} = \begin{bmatrix} 1 & \alpha_i \\ \\ \beta_i & 1 \end{bmatrix} \begin{bmatrix} z & 0 \\ \\ 0 & 1 \end{bmatrix} \begin{bmatrix} Q_i(z) \\ \\ \bar{Q}_i(z) \end{bmatrix} \qquad (2.15)$$

$$= \theta_i \cdots \theta_0 \begin{bmatrix} 1 \\ \\ 1 \end{bmatrix} = \Theta_i(z) \begin{bmatrix} 1 \\ \\ 1 \end{bmatrix}$$

where

$$\theta_i(z) = \begin{bmatrix} 1 & \alpha_i \\ \\ \beta_i & 1 \end{bmatrix} \begin{bmatrix} z & 0 \\ \\ 0 & 1 \end{bmatrix}$$

Notice that $r_i = \bar{p}_i$, $i = 0,1,2,\ldots$ so that (2.14) maybe slightly simplified.

In the preceeding algorithms it was tacitly understood that whenever a division has to be made no zero denominator occurs. In the case the Padé-table is normal, this does not produce any difficulties. In case the Padé-table is not normal, then an extension of the given algorithm is possible [14] but is not discussed here.
Matrix extensions for the above algorithms have been obtained in [30-32].

3. Outgoing (Schur-type) recursions

The recursions derived so far have the disadvantage that the essential recursion coefficients α_i and β_i or r_i, p_i are computed in a non-recursive fashion. The expressions given for them may lead to large numerical errors. A method which does produce the coefficients in a recursive way is obtained by writing out recursions for the $\mathbb{Z}_i$ and $\mathbb{S}_i$ vectors previously defined. The $\mathbb{Z}_i$ and $\mathbb{S}_i$ will appear to contain all information needed, at step i, to continue the algorithm. Once the recursion-coefficients are known, then the Padé-approximant may be computed using schemes from the previous section. However, in practice, one shall usually implement these recursions in a filter which then will be of ladder-type.

In order not to handle infinite vectors we fix temporarily the maximal order of approximation n. We select further a fixed row in the Padé-table m, and use the notations of the preceeding section. With increasing i, the vectors $\mathbb{Z}_i$ and $\mathbb{S}_i$ decrease in length, so that "cutting" operators are needed to express the recursions. Let

$$\lrcorner \triangleq \left[\begin{array}{cc|c} 1 & 0 & \\ & \ddots & 0 \\ 0 & 1 & \end{array}\right], \qquad \urcorner \triangleq \left[\begin{array}{c|cc} & 1 & 0 \\ 0 & & \ddots \\ & 0 & 1 \end{array}\right]$$

then (2.7) generates the following recursions on $\mathbb{Z}_i$ and $\mathbb{S}_i$:

(3.1)

$$\mathbb{Z}_{-1}^T = [-1 \underbrace{0 \ \ldots \ 0}_{n+1}] \qquad \mathbb{S}_{-1}^T = [\underbrace{0 \ \ldots \ 0}_{n+1} 1]$$

$$\mathbb{Z}_0^T = [f_{m+1} \ \ldots \ f_{m+n+1}] \qquad \mathbb{S}_0^T = [f_{m-n} \ \ldots \ f_m]$$

$$\mathbb{Z}_{i+1} = \lrcorner\mathbb{Z}_i + \urcorner(\lrcorner\mathbb{Z}_{i-1} + \mathbb{Z}_i u_i) v_i \qquad \mathbb{S}_{i+1} = \urcorner(\lrcorner\mathbb{S}_{i-1} + \mathbb{S}_i u_i) v_i + \lrcorner\mathbb{S}_i$$

$$u_i = -\, r_{i-1}/r_i \qquad v_i = -p_i/p_{i-1}$$

$$r_i = [10 \ \ldots \ 0]\mathbb{Z}_i \qquad p_i = [0 \ \ldots \ 01]\mathbb{S}_i$$

The important point with these "outgoing" recursions is that the coefficients u_i and v_i are given in the $\mathbb{Z}_i$, $\mathbb{Z}_{i-1}$, respectively $\mathbb{S}_i$ and $\mathbb{S}_{i-1}$ so

that one does not need to compute inner products on the original sequence $\{f_k\}$. The $\mathbb{Z}_i$ contain all the information needed about the sequence for further computation.

The proof of the outgoing recursions is given in [14,15]. It follows easily from (2.7-2.9). The coefficients u_i and v_i appear in the following continued fractions :

$$f(z) = P^{(m/0)}(z) + z^m \sum_{i=0}^{\infty} \frac{z\, v_i|}{|z+u_i v_i} \tag{3.2}$$

which has the Padé-approximant

$$P^{(m/j)}(z)/Q^{(m/j)}(z)$$

for its jth convergent (with monic normalization for the denominator).

The incoming recursion (2.5) is nothing but a recursive scheme to compute the convergents in the continued fraction expansion, while the outgoing recursion (3.1) gives recursively the coefficients in the continued fraction expansions.

Consider next the outgoing recursion for (2.14), which will be shown to be a generalization of the Schur-algorithm :

(3.3)

$$r_0 = f_{m+1} = \bar{p}_0$$

$\bar{r}_0 = f_{m+2}$	$p_0 = f_m$
$\mathbb{Z}_0^T = [f_{m+1} \ldots f_{m+n+1}]$	$\mathbb{S}_0^T = [f_{m-n} \ldots f_m]$
$\bar{\mathbb{Z}}_0^T = [f_{m+2} \ldots f_{m+n+2}]$	$\bar{\mathbb{S}}_0^T = [f_{m+1-n} \ldots f_{m+1}]$
$\beta_i = -\bar{r}_i/r_i$	$\alpha_i = -p_i/\bar{p}_i$
$\bar{\mathbb{Z}}_{i+1} = \neg(\bar{\mathbb{Z}}_i + \mathbb{Z}_i\beta_i)$	$\mathbb{S}_{i+1} = \lrcorner(\mathbb{S}_i + \bar{\mathbb{S}}_i\alpha_i)$
$\mathbb{Z}_{i+1} = \lrcorner(\mathbb{Z}_i + \bar{\mathbb{Z}}_i\alpha_i)$	$\bar{\mathbb{S}}_{i+1} = \neg(\bar{\mathbb{S}}_i + \mathbb{S}_i\beta_i)$

$$r_{i+1} = [10 \ldots 0]\mathbb{Z}_{i+1} = [0 \ldots 01]\bar{\mathbb{S}}_{i+1} = \bar{p}_{i+1}$$

$\bar{r}_{i+1} = [10 \ldots 0]\bar{\mathbb{Z}}_{i+1}$	$p_{i+1} = [0 \ldots 01]\mathbb{S}_{i+1}$

The outgoing recursion (3.3) can be expressed in the following compact form : take $n = \infty$ and define

$$Z_i(z) = z^{m+i+1}[1 \quad z \quad z^2 \quad \ldots]\mathbb{Z}_i \quad \text{and}$$

$$\bar{Z}_i(z) = z^{m+i+2}[1 \quad z \quad z^2 \quad \ldots]\bar{\mathbb{Z}}_i$$

then

$$\begin{bmatrix} Z_{i+1}(z) \\ \bar{Z}_{i+1}(z) \end{bmatrix} = \begin{bmatrix} 1 & \alpha_i \\ \beta_i & 1 \end{bmatrix} \begin{bmatrix} z & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} Z_i(z) \\ \bar{Z}_i(z) \end{bmatrix} = \theta_i(z) \begin{bmatrix} Z_i(z) \\ \bar{Z}_i(z) \end{bmatrix} = \Theta_i(z) \begin{bmatrix} Z_0(z) \\ \bar{Z}_0(z) \end{bmatrix}$$

where $\Theta_i(z)$ is defined in (2.15). A similar expression exists for the $\mathbb{S}_i$. The residuals $Z_i(z)$ are related to the continued fraction (3.2) as follows :

$$f(z) = P^{(m/0)}(z) + \frac{z^{m+1}v_0}{|z+u_0v_0} + \ldots + \frac{z\,v_n}{|z+u_nv_n} - \frac{Z_{n+1}(z)}{|Z_n(z)}$$

Example 1

We take $f(z) = (z+1.1)^{-1}(z-3.3)^{-1}(z^2+1.4^2)^{-1} = \sum_{k=0}^{\infty} f_k z^k$ with f_k given in table 1

k	f_k
0	-0.14055209
1	0.08518308
2	-0.01863545
3	0.03476062
4	-0.06278763
5	0.04762904
6	-0.02749619
7	0.02978531

table 1

The righthand-side triangular factors of (2.8-2.11) contain the vectors $\mathbb{Z}_i$, $\bar{\mathbb{Z}}_i$, $\mathbb{S}_i$ and $\bar{\mathbb{S}}_i$ produced by algorithm (3.3). For m = 0 and n = 4 they are given in table 2.

i	0	1	2	3	4
$\mathbb{Z}_i$	0.08518304				
	-0.01863545	0.05443460			
	0.03476062	0.03871958	0.1851584		
	-0.06278764	-0.06883898	-0.1963802	-0.06388731	
	0.04762904	0.01580028	0.07555752	0.03781958	0.0

table 2.a

i	0	1	2	3	4
$\bar{\mathbb{Z}}_i$	-0.01863545				
	0.03476062	0.03068376			
	-0.06278764	-0.05518308	-0.07700858		
	0.04762904	0.03389303	0.07269627	-0.008979495	
	-0.02749619	-0.01707642	-0.02598274	0.005442118	0.0

table 2.b

i	4	3	2	1	0
$\mathbb{S}_i$	1.0	0.0	0.0	0.0	0.0
		-0.4545455	0.0	0.0	0.0
			-0.5988024	0.0	0.0
				-0.2319109	0.0
					-0.1405521

table 2.c

i	4	3	2	1	0
$\bar{\mathbb{S}}_i$	0.0	-0.1405521 -0.0638873	0.0 -0.1405521 0.1851584	0.0 0.0 -0.1405521 0.0544346	0.0 0.0 0.0 -0.1405521 0.0851831

table 2.d

The r_i, $\bar{r}_i$, p_i and $\bar{p}_i$ are found on the diagonals of the matrices. Note that $r_i = \bar{p}_i$. Remark also that the $\mathbb{S}_i$ matrix is diagonal because it contains the numerators of row 0 in the Padé table i.e. constants. The $\bar{\mathbb{S}}_i$ matrix contains the numerators of row 1, and therefore it is bidiagonal. The recursion coefficients are given in table 3.

i	α_i	β_i
0	1.6500000	0.21876933
1	4.2603593	-0.56368121
2	3.2340000	0.41590641
3	-7.1148000	-0.14055209

table 3

They can be found from the diagonal elements in table 2 according to their definitions

$$\alpha_i = -p_i/\bar{p}_i \qquad \text{and} \qquad \beta_i = -\bar{r}_i/r_i$$

These coefficients can be used to give denominators $\mathbb{Q}_i$ of row 0 and denominators $\bar{\mathbb{Q}}_i$ of row 1 according to the recursion (2.14). The results are found in table 4.

i	4	3	2	1	0
$\mathbb{Q}_i$	1.0	1.65	4.2603593	3.23400000	-7.1148
		1.0	2.5820359	1.96000000	-4.3120
			1.0	0.75909091	-1.6700
				1.0	-2.2000
					1.0000

table 4.a

i	4	3	2	1	0
$\bar{\mathbb{Q}}_i$	1.0				
	0.60606060	1.0			
	0.23472199	1.06060606	1.0		
	0.309214595	0.51020408	-0.71130469	1.0	
	-0.14055209	0.41590641	-0.56368121	0.218769327	1.0

table 4.b

4. Padé recursions in the Laurent-Padé problem

In order to obtain the connection between Padé recursions and the Levinson-Schur recursions, we introduce at this stage the Laurent-Padé problem and show that it may be solved with the previous algorithms.

Let be given

$$f(z) = \sum_{k=-\infty}^{\infty} f_k z^k$$

and define

$$f^{\pm}(z) = f_0 + 2 \sum_{k=1}^{\infty} f_{\pm k} z^k$$

so that

$$f(z) = \frac{1}{2}[f^{+}(z) + f^{-}(1/z)].$$

We call, for nonnegative integers r and s, $\sum_{-r}^{s} \phi_j z^j$ a Laurent polynomial of degree max(r,s). An [m/n] Laurent-Padé approximant to f is defined to be the following object :

$$[m/n] \triangleq \frac{p^{(m/n)}(z)}{q^{(m/n)}(z)} \triangleq \frac{\frac{1}{2}(p^{+}(z)q^{-}(\frac{1}{z})+p^{-}(\frac{1}{z})q^{+}(z))}{q^{+}(z)q^{-}(\frac{1}{z})}$$

where :

(i) $q^{\pm}(z)$ have degree n;

(ii) $p^{\pm}(z)$ have degree at most max(m,n);

(iii) $r^{\pm}(z) = \frac{p^{\pm}(z)}{q^{\pm}(z)}$ is a Padé approximation to $f^{\pm}(z)$ of order (m+n+1);

(iv) $p^{(m/n)}(z)$ has degree at most m.

Here again a proper normalisation may be imposed on $q^{\pm}(z)$.

Notice that, as $m \geqslant n$, the Laurent-Padé problem reduces to just two regular Padé problems. The more interesting case here is when $m < n$. Then the Padé-part of the problem gives only m+n+1 conditions for 2n+1 coefficients. An extra set of conditions is obtained from the degree condition on the pseudo-polynomial $p^{(m/n)}(z)$.

The Levinson estimation problem will appear to be a Laurent-Padé approximation problem with m = 0. In that case, and in that case only, supplementary stability properties will be fulfilled. For an [m/n]

Laurent-Padé approximant we have that the Laurent series for [m/n] coincides with the given Laurent series in the coefficients of z^k for $|k| < m+n+1$. We say that the Laurent order of the residual $f(z)-[m/n]$ is $m+n+1$.

It is shown in [20] that the conditions to be imposed on the coefficients are :

$$p_i^{\pm} = 2 \sum_0^{i\,\prime} f_{\pm(i-j)} q_j^{\pm} \qquad 0 \leqslant i \leqslant \ell = \max(m,n) \tag{4.1}$$

$$\sum_0^n f_{\pm(i-j)} q_j^{\pm} = 0 \qquad m < i \leqslant m+n$$

where the prime on the summation indicates that f_0 is to be replaced by $f_0/2$. Furthermore $p_i^{\pm} = 0$ for $i<0$ and $i>\ell$ and $q_j^{\pm} = 0$ for $j<0$ and $j>n$.

The equations result now in two Toeplitz systems, one for q^+ and one for q^- so that the algorithms discussed in section 2 and 3 may be used as given on the series $f^{\pm}(z)$, the only difference being that two Padé-problems instead of one is to be solved.

In the case $m = 0$, (4.1) specializes to

$$p_i^+ = 2 \sum_0^i f_{i-j} q_j^+ \qquad 0 \leqslant i \leqslant n \tag{4.2a}$$

$$\sum_0^n f_{(i-j)} q_j^+ = 0 \qquad 0 < i \leqslant n \tag{4.2b}$$

which is nothing but the set of Levinson equations as anounced.

Further applications of the scheme are [19] :

1) <u>Fourier-Padé approximation</u>

Here one wishes to approximate

$$S(\omega) = a_0 + 2 \sum_i^{\infty} (a_k \cos k\omega + b_k \sin k\omega)$$

Setting $z = e^{j\omega}$, $f_0 = a_0$ and $f_{\mp k}$ $a_k \pm jb_k$ $(k \geqslant 1)$, one has

$$S(\omega) = f(z) = \sum_{-\infty}^{\infty} f_k z^k$$

so that the Laurent-Padé approximation for this series is in fact a Fourier-Padé approximation.
When the $\{f_k\}_{-\infty}^{\infty}$ form a nonnegative sequence, i.e. when the T(0,n,n) are positive semidefinite for all n, then there is a direct connection with the trigonometric moment problem [12,19].

2) The Chebyshev-Padé approximation

Here $b_k = 0$ for $k \geqslant 1$ in the above series, so that $S(\omega)$ reduces to a Chebyshev series in x when $x = \cos \omega$. $f_k = f_{-k}$ is real. This topic has been treated in [19-21].
For a relation with two-point Padé approximation see [29].

For these applications one may use the symmetry in the Laurent-coefficients f_k to reduce the amount of computation. Only one Padé-recursion has to be executed, the other being either identical or complex-conjugate. In the latter case (Fourier-Padé approximation) one has :
(upper star denotes complex conjugate)

$$\begin{aligned} q^+(z) &= (q^-(z^\star))^\star \\ p^+(z) &= (p^-(z^\star))^\star \end{aligned} \tag{4.3}$$

up to some normalization, which can be deduced immediately from (4.1).
Let us denote by the understar the parahemitian conjugate defined by :

$$f_\star(z) = [f(\frac{1}{z^\star})]^\star$$

so that $q^-(1/z) = q_\star^+(z)$, $p^-(1/z) = p_\star^+(z)$ and $r^-(z) = r_\star^+(z)$.
Hence, all quantities may be expressed in terms of p^+ and q^+. The [m/n] Fourier-Padé approximant becomes

$$[m/n] = \frac{\frac{1}{2}[p^+(z)q_\star^+(z) + p_\star^+(z)q^+(z)]}{q^+(z)q_\star^+(z)} \tag{4.4}$$

Remark that in [m/n] of (4.4) the poles appear in pairs like $(\alpha_i, 1/\alpha_i^\star)$. The zeros have a similar property i.e. if β_i is a zero, then $1/\beta_i^\star$ is also a zero, but if $|\beta_i| = 1$, then $1/\beta_i^\star = \beta_i$, so that a factorization in minimal phase-maximal phase stable factors is impossible unless all unimodular zeros are of even multiplicity. Only in that case is (4.4) a positive real function on the unit circle.

Example 2

Take the low-pass filter of [4], i.e. $F(z^{-1}) = f_0 + 2\sum_{k=1}^{\infty} f_k z^{-k}$ with

$f_0 = (\omega_s+\omega_p)/2\pi$, $f_k = \frac{1}{(\omega_s-\omega_p)k^2\pi}[\cos(k\omega_p)-\cos(k\omega_s)]$

where $\omega_s = 0.5201\pi$ and $\omega_p = 0.4299\pi$.

The Laurent-Padé table has been computed for $0 \leq m, n \leq 6$.

Zeros not on the unit circle appear only for $m \geq n+2$.

But all entries for $m \geq 1$ have simple zeros about equally spaced on the left half of the unit circle. This was to be expected since the approximation oscillates around the exact value in the stopband, as illustrated in appendix 1.

We remark also that $q^+(z)$ as defined above has zeros which are usually stable. Only the (2,1), (4,1), (4,3), (6,1), (6,4) and (6,6) entries have an unstable pole from $q^+(z)$, but in those cases they are approximately cancelled by a corresponding zero.

5. The Levinson and Schur algorithms

Suppose first that a function $S(z) = \sum_{k=0}^{\infty} s_k z^{-k}$ is given, which is the transfer function of a stable filter, i.e. all the poles of $S(z)$ are inside the unit disc. A Padé approximation of $S(z)$ will often result in an unstable approximant $S_a(z)$.
This is illustrated in the following example.

Example 3

Let $S(z) = [(z^{-1}+1.1)(z^{-1}-3.3)(z^{-2}+1.96)]^{-1}$
with poles given in table 5

i	pole i of S(z)
1	-1/1.1 ≃ -0.90909
2	1/3.3 ≃ 0.30303
3	j/1.4 ≃ 0.71429j
4	-j/1.4 ≃ -0.71429j

table 5

The power series expansion $S(z) = \sum_{k=0}^{\infty} s_k z^{-k}$ has coefficients given in table 6.

k	s_k
0	-0.1406e+00
1	-0.8518e-01
2	-0.1864e-01
3	0.3476e-01
4	-0.6279e-01
5	0.4763e-01

table 6

The (1,2) entry of the Padé table is

$$[1/2] = \frac{p_0+p_1 z^{-1}}{q_0+q_1 z^{-1}+q_2 z^{-2}} \quad ,$$

the coefficients of which are resumed in table 7, together with zero a_0 and the poles b_0 and b_1.

i	p_i	q_i	a_i	b_i
0	0.24934677	-1.7740524	1.317	1.186
1	-0.32848078	1.2618918	-	0.475
2	-	1.0	-	-

table 7

Clearly b_0 is an unstable pole.

To avoid such stability problems suppose next that some other information than $S(z)$ is available, e.g. suppose as it is done in the autocorrelation method that the power spectrums $r(\omega) = |S(e^{j\omega})|^2$ of $S(z)$ is known. Then we can write $r(\omega) = C_0 + 2\sum_{k=1}^{\infty} C_k \cos k\omega = \frac{1}{2}[F(z)+F_\star(z)] = 2\,\mathrm{Re}\,F(z) = S(z)S_\star(z)$ with $F(z) = C_0 + 2\sum_{k=1}^{\infty} C_k z^{-k}$ and $z = e^{j\omega}$.

$F(z)$ is a positive real function, i.e. $\mathrm{Re}\,F(e^{j\omega}) \geqslant 0$. We suppose also that the correlation coefficients C_k are real.

An attempt to approximate $S(z)$ by a rational function $S_a(z)$, such that $|S(z)| \approx |S_a(z)|$ could be as follows. First compute a Fourier-Padé approximation of $r(\omega) \overset{\Delta}{=} f(z) = \sum_{-\infty}^{\infty} C_{|k|} z^{-k}$ and then find its spectral factor which is a candidate for $S_a(z)$. Unfortunately, as it is illustrated by example 2, such a factorization is not always possible.

However if we take an approximant from row zero in the Fourier-Padé table, then (4.1) reduces to (4.2) and we shall reduce for this case the set of equations (4.2) to a form identical to the one considered in par. 2. We choose q^+ to be comonic and hence, because of (4.3), q^- to be monic. To (4.2b) we add the row equation for i = 0, namely

$$\sum_{0}^{n} f_{-j}\, q_j^+ = R_n \qquad (5.1)$$

(4.2b) and (5.1) now generate the set :

$$\begin{bmatrix} f_0 & f_{-1} & \cdots & f_{-n} \\ \hline & & & \\ & & & \\ & & & \\ & T(1,n-1,n) & & \end{bmatrix} \begin{bmatrix} 1 \\ q_1^+ \\ \vdots \\ q_n^+ \end{bmatrix} = T(0,n,n) \begin{bmatrix} 1 \\ q_1^+ \\ \vdots \\ q_n^+ \end{bmatrix} = \begin{bmatrix} R_n \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{5.2}$$

which is a well conditioned, positive semidefinite set of equations. The outgoing resursion for solving (5.2) may now be adopted with $m = -1$, and starting values $r_0 = f_0 = \bar{p}_0$, where $\bar{p}_0$ has been renamed here and is equal to R_0. One obtains $\rho_i \triangleq \alpha_i = \beta_i$ known as reflection coefficients and the recursion becomes the so called "Schur algorithm", see [15,25]. The matrices θ_i of (2.15) and (3.4) are given by :

$$\theta_i(z) = \begin{bmatrix} z & \rho_i \\ & \\ z\rho_i & 1 \end{bmatrix}, \quad |\rho_i| \leq 1 \tag{5.3}$$

The physical significance of these quantities will be discussed in the next section.

In the incoming algorithm $\bar{\mathbb{Q}}_i$ becomes $\mathbb{Q}_i$ in reverse order. It generates the Szegö polynomial of order i [8].

This approximation of $S(z)$ is of autoregressive type and it is optimal in a weighted least squares sense [6].

Looking at the factorization interpretations (2.8) - (2.11) we see that the Szegö-Levinson algorithm gives triangular factors of the inverse of a symmetric Toeplitz matrix and of a shifted one (left hand side factors) and the Schur algorithm gives likewise the triangular factor of the Toeplitz matrix itself and of a shifted version thereof (right hand side factors). In this perspective it is not difficult to understand that the recursive techniques for Padé approximation will be numerically unstable in a general situation because a linear system of equations is solved without a pivoting strategy.

However, for a symmetric positive definite matrix, the Cholesky method of factorization is known to be stable without pivoting [27] and a similar property seems to be shared by the Schur and Levinson algorithms [16],[28]. When the algorithms run into numerical difficulties, it will be due to the ill conditioning of the problem. There are however some indications that the Schur algorithm is to be preferred [16].

6. Discussion

For the Schur-Levinson case discussed in the last paragraph, there is a nice scattering interpretation. The recursion $\Theta_i(z) = \theta_i(z) \ldots \theta_0(z)$ defined in (2.15) with $\rho_i = \alpha_i = \beta_i$ may be viewed as a discrete analog of a wave-propagating system (ladderfilter). The waves with their propagation direction are defined in fig. 1

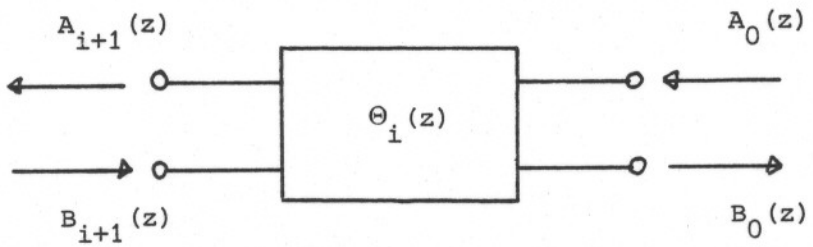


fig. 1

$$\begin{bmatrix} A_{i+1}(z) \\ B_{i+1}(z) \end{bmatrix} = \Theta_i(z) \begin{bmatrix} A_0(z) \\ B_0(z) \end{bmatrix}$$

$\Theta_i(z)$ is a genuine chain scattering matrix for lossless propagation because it satisfies the necessary and sufficient conditions for that situation. These will not be discussed here and we refer to [25] for further exploration of this topic. However, the recursions obtained for the incoming and outgoing quantities have the following wave interpretation : take as boundary conditions at the output $A_0(z) = B_0(z) = 1$, then the waves are $A_{i+1}(z) = Q_{i+1}(z)$ and $B_{i+1}(z) = \bar{Q}_{i+1}(z)$. This is the incoming recursion. On the other hand, take as boundary conditions $A_0(z) = Z_0(z)$ and $B_0(z) = \bar{Z}_0(z)$, then at level i one obtains the waves $A_{i+1}(z) = Z_{i+1}(z)$ and $B_{i+1}(z) = \bar{Z}_{i+1}(z)$. At initial time i = 0 the "waves" $Z_0(z)$ and $\bar{Z}_0(z)$ are known and given. The outgoing recursion hence solves an inverse scattering problem whereby the reflection coefficient ρ_i is recursively computed together with the outgoing waves at that stage. This allows for the computation of the quantities of the next stage. Once the outgoing inverse problem has been solved, then introducing the boundary conditions $A_0(z) = B_0(z) = 1$ produces the incoming quantities. However, in practical filtering applications, it is not necessary to compute the incoming quantities. The wave-filter or ladder filter may be used directly as prediction or modelling filter.

In another direction one may wonder whether the stability properties of the Schur-Levinson algorithm may be extended to the ARMA case. This is indeed the case and has been treated in other papers, see e.g. [33]; [15], [25]. However one leaves the Padé scheme in that case to replace it with a Nevanlinne -Pick type interpolation problem. However, some recent experiments have shown that the ARMA Padé case still may be solved. This will be the content of another paper.

References

1. F.Brophy - A.C.Salazar,"Considerations of the Padé approximant technique in the synthesis of recursive digital filters", IEEE Trans. Audio Electroacoustics, vol. AU-21,pp. 500-505,1973.

2. ____ ,"Recursive digital filter synthesis in the time domain", IEEE Trans. Acoust.,Speech, Signal Processing, vol. ASSP-22,pp.45-55,1974.

3. C.K.Chui, P.W.Smith,L.Y.Su,"A minimization problem related to Padé synthesis of recursive digital filters" in Padé and rational approximation theory and applications, E.B.Saff and R.S.Varga (eds.), New York, Acad. Press,1977, pp.247-256.

4. C.K.Chui, A.K. Chan,"A two-sided rational approximation method for recursive digital filtering", IEEE Trans. Acoust.,Speech,Signal Proc., vol ASSP-27, pp.141-145,1979.

5. R.Hastings-James,S.Mehra,"Extensions of Padé approximant technique for the design of recursive digital filters",IEEE Trans. Acoust.,Speech,Signal Proc., vol. ASSP-25,pp.501-509,1977.

6. J.Makhoul,"Linear prediction: a tutorial review", Proc. IEEE, vol.63, pp. 561-580,1975.

7. N.Levinson,"The Wiener RMS (root mean square) error criterion in filter design and prediction", J. Mathematical Phys., vol. 25,pp. 261-278,1947.

8. G.Szegõ,"Orthogonal Polynomials", Amer. Math. Soc. Colloquium publ. XXIII, AMS, Providence, Rhode Island, 1939.

9. J.L.Walsh,"Interpolation and approximation by rational functions in the complex domain", Amer. Math. Soc. Colloquium publ. XX (rev.),AMS,Providence, Rhode Island, 1960.

10. W.F.Trench,"An algorithm for the inversion of finite Toeplitz matrices", SIAM J. Appl. Math.,vol.12,pp. 515-522, 1964.

11. S.Zohar,"Toeplitz matrix inversion: the algorithm of W.F.Trench",J.Assoc. Comput. Mach., vol.16,pp. 592-601,1969.

12. N.I.Akhiezer,"The classical moment problem", Oliver and Boyd, Edinburgh-London, 1965.

13. A.Bultheel,"Recursive algorithms for the Padé table: two approaches" in Padé approximation and its applications, L.Wuytack (ed.), Lecture notes in mathematics, Springer Verlag, Berlin, 1979,pp. 211-230.

14. ____, "Fast factorization algorithms and Padé approximation: a survey", submitted for publication.

15. ____, "Rekursieve rationale benaderingen", Ph.D. thesis, Katholieke Universiteit Leuven, Belgium, sept. 1979 (in Dutch).

16. ____, "Towards an error analysis of fast Toeplitz factorization", Katholieke Universiteit Leuven, Applied math. and programming division, rept. TW44, may 1979.

17. A.Bultheel and P. Dewilde,"Orthogonal functions related to the Nevanlinna-Pick problem" in Proc. MTNS conference, Delft, July 1979, P.Dewilde (ed.), Western Periodicals Comp., North Hollywood, 1979,pp.207-211.

18. J.Gilewicz,"Approximants de Padé", Lecture notes in mathematics 667, Springer Verlag, Berlin, 1978.

19. W.B.Gragg,"Laurent, Fourier and Chebyshev-Padé tables" in Padé and rational approximation theory and applications, E.B.Saff and R.S.Varga (eds.), Acad. Press, New York,1977,pp.61-72.

20. W.B.Gragg,G.D.Johnson,"The Laurent-Padé table" in Information Processing 74, Proc. IFIP congress, North Holland, Amsterdam,1974,pp. 632-637.

21. J.S.R.Chisholm, A.Common,"Padé-Chebyshev approximants" in Padé approximation and its applications, L.Wuytack (ed.), Lecture notes in mathematics 765, Springer Verlag, Berlin,1979,pp. 1-19.

22. J.Schur,"Ueber Potenzreihen die im Innern des Einheitskreises beschränkt sind", Z. Reine Angew. Math.,vol.147,pp.205-232,1917 and vol. 148,pp.122-145,1918.

23. R.Nouta,"Studies in Wave digital filter theory and design", Ph.D. thesis, Univ. of Technology, Delft, The Netherlands, Dept. E.E., 1979.

24. E. Deprettere, P.Dewilde,"Generalized orthogonal filters for stochastic prediction and modelling" in Digital Signal Processing, Academic Press, New York, 1979.

25. P.Dewilde, A.Vieira, T.Kailath,"On a generalized Szegő-Levinson realization algorithm for optimal linear predictors based on a network synthesis approach", IEEE Trans.on Circuits and Systems, vol. CAS-25, 1978,pp. 663-675.

26. Ph. Delsarte, Y.Genin,Y.Kamp,"The Nevanlinna-Pick problem for matrix valued functions", SIAM J. Appl. Math.,vol 36,1979,pp.47-61.

27. J.H. Wilkinson,"The algebraic eigenvalue problem", Claredon Press, Oxford, 1965.

28. G. Cybenko,"Round-off error propagation in Durbin's, Levinson's and Trench's algorithms", presented at Internatl. conf. on Acoust.,Speech and Signal Proc., IEEE, Washington DC, april 1979.

29. W.J. Thron,"Two-point Padé tables, T-fractions and sequences of Schur" in Padé and rational approximation theory and applications, E.B. Saff and R.S.Varga (eds.), Academic Press, New York, 1977,pp. 215-226.

30. H.Akaike,"Block Toeplitz matrix inversion", SIAM J. Appl. Math., vol.24, pp. 234-241, 1973.

31. J.RIssanen,"Algorithms for triangular decomposition of block Hankel and Toeplitz matrices, application to factoring positive matrix polynomials" Math. of Computation, vol 27,pp. 147-154, 1973.

32. A.Bultheel,"Recursive algorithms for the matrix-Padé problem", to appear mathematics of computation, 1980.

33. P. Dewilde, H. Dym: "Schur recursions, error formulas and convergence of rational estimators for stationary stochastic sequences", Delft University of Technology, Rept. 92, 1979.

Appendix 1

In this appendix we represent graphically the power spectra of the Laurent-Padé approximations [m/n] to the lowpass filter considered in [4].
The power spectrum is given by

$$H(\omega) = f_0 + 2 \sum_{k=1}^{\infty} f_k \cos k\omega$$

with $f_0 = (\omega_s + \omega_p)/2\pi$

and $f_k = [\cos(k\omega_p) - \cos(k\omega_s)]/((\omega_s-\omega_p)k^2\pi)$,

$\omega_s = 0.5201\pi$, $\omega_p = 0.4299\pi$

(remark that in [4] the factor 2 before the summation has been forgotten and that two digits of ω_p and ω_s have been switched).

The Laurent-Padé approximations [m/n] for

$$f(z) = \sum_{-\infty}^{\infty} f_k z^k$$

are computed for $0 \leq m,n \leq 6$ and the resulting approximants $H_L(\omega)$ are plotted for $0 \leq \omega \leq \pi$ in the next figures. They are grouped row by row, each row on one page.
The scale in y-direction is different for each function, therefore we have crossed the points (0,0.5) and (0,1) on the y-axis to indicate the scale used.
Remark that row zero, which gives the Levinson approximants, yields positive real approximations for each value of n but the approximation is not good in the passband.
The zeros on the unit circle are found as points where the approximation intersects the x-axis in the stopband. For higher values of n, the approximation in the stopband is quite good, but the approximation is not positive real (there are intersections with the x-axis, invisible at the scale used).
It is clear that if we add a very small constant to the given power spectrum, then the approximation will probably be positive real and deviate only very little from the Laurent-Padé approximation of the exact spectrum.
Remark however that adding a constant to the given power series does not result in adding the same constant to the Laurent-Padé approximant.

Fig. A.0

lowpass filter
Laurent-Padé approximants
for row 0.

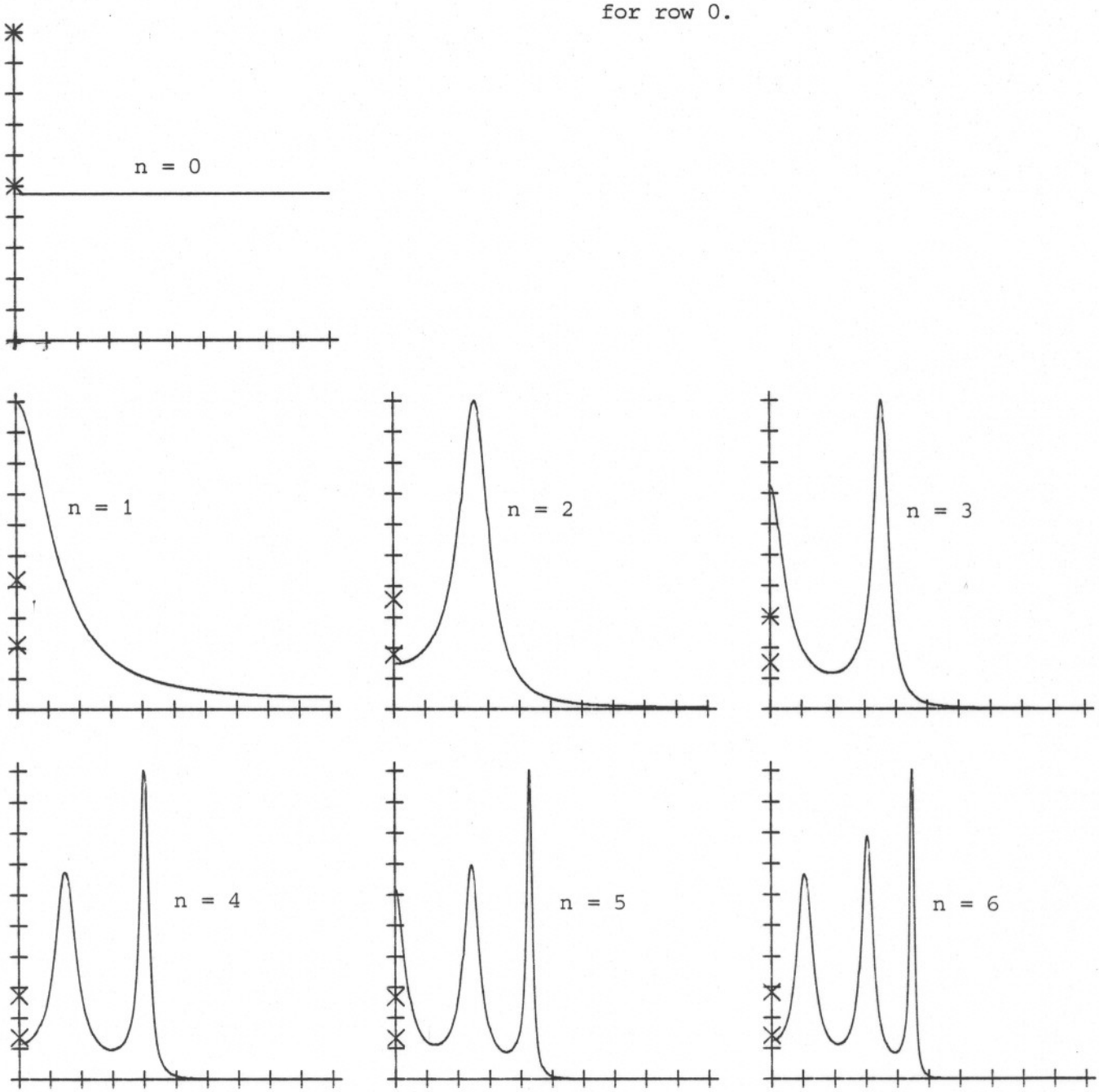


Fig. A.1

lowpass filter
Laurent-Padé approximants
for row 1.

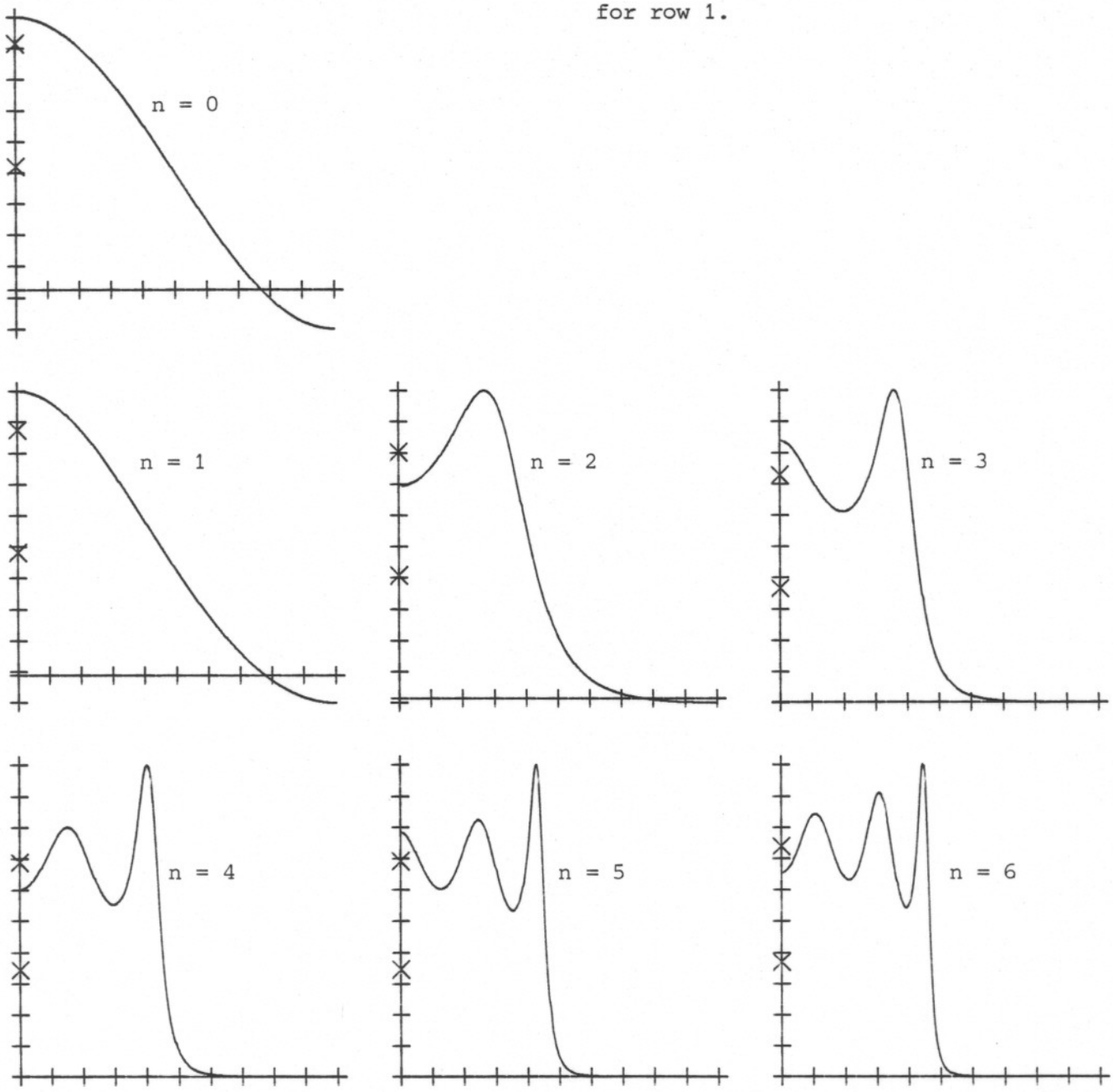


Fig. A.2

lowpass filter
Laurent-Padé approximants
for row 2.

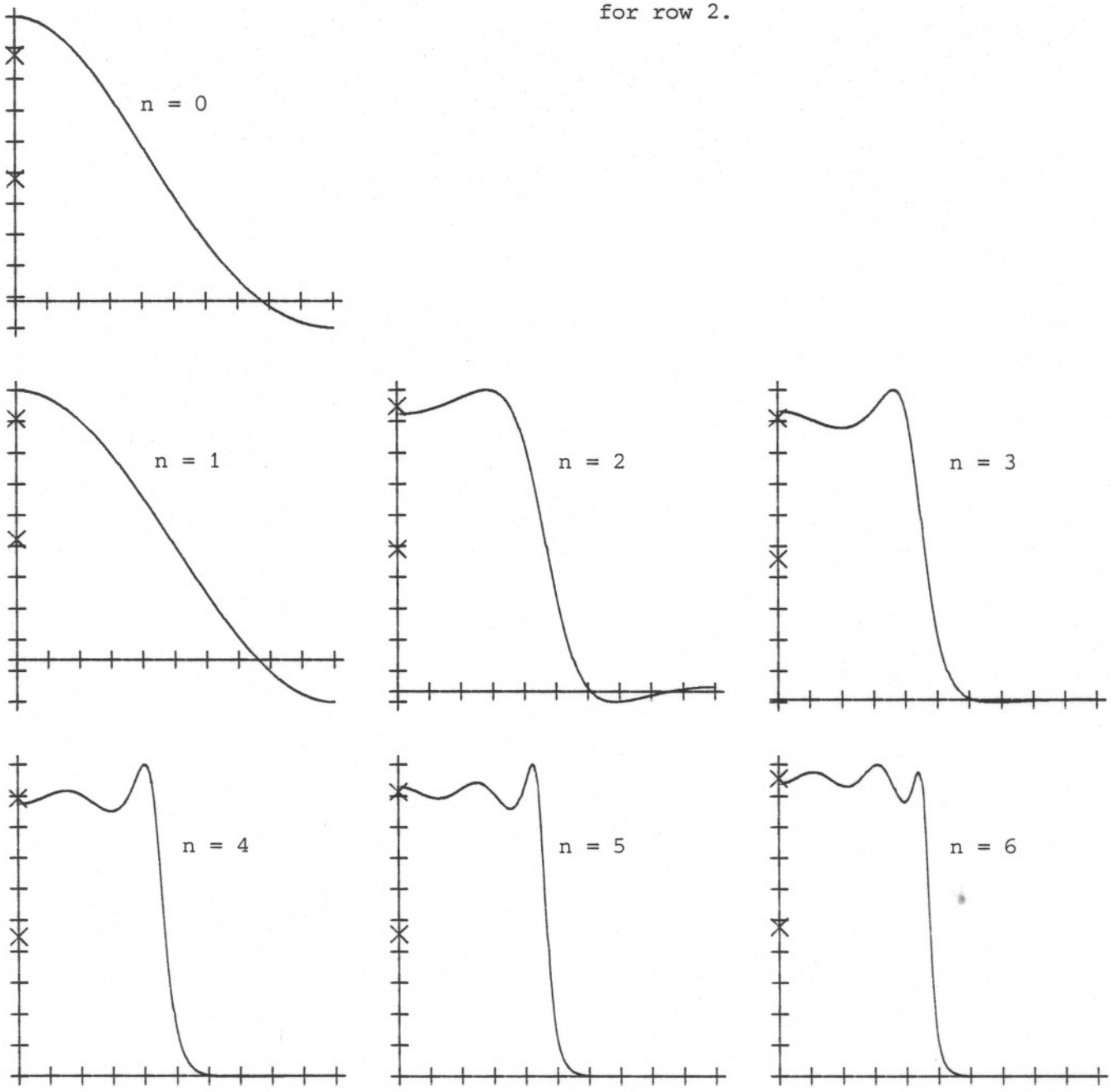


Fig. A.3

lowpass filter
Laurent-Padé approximants
for row 3.

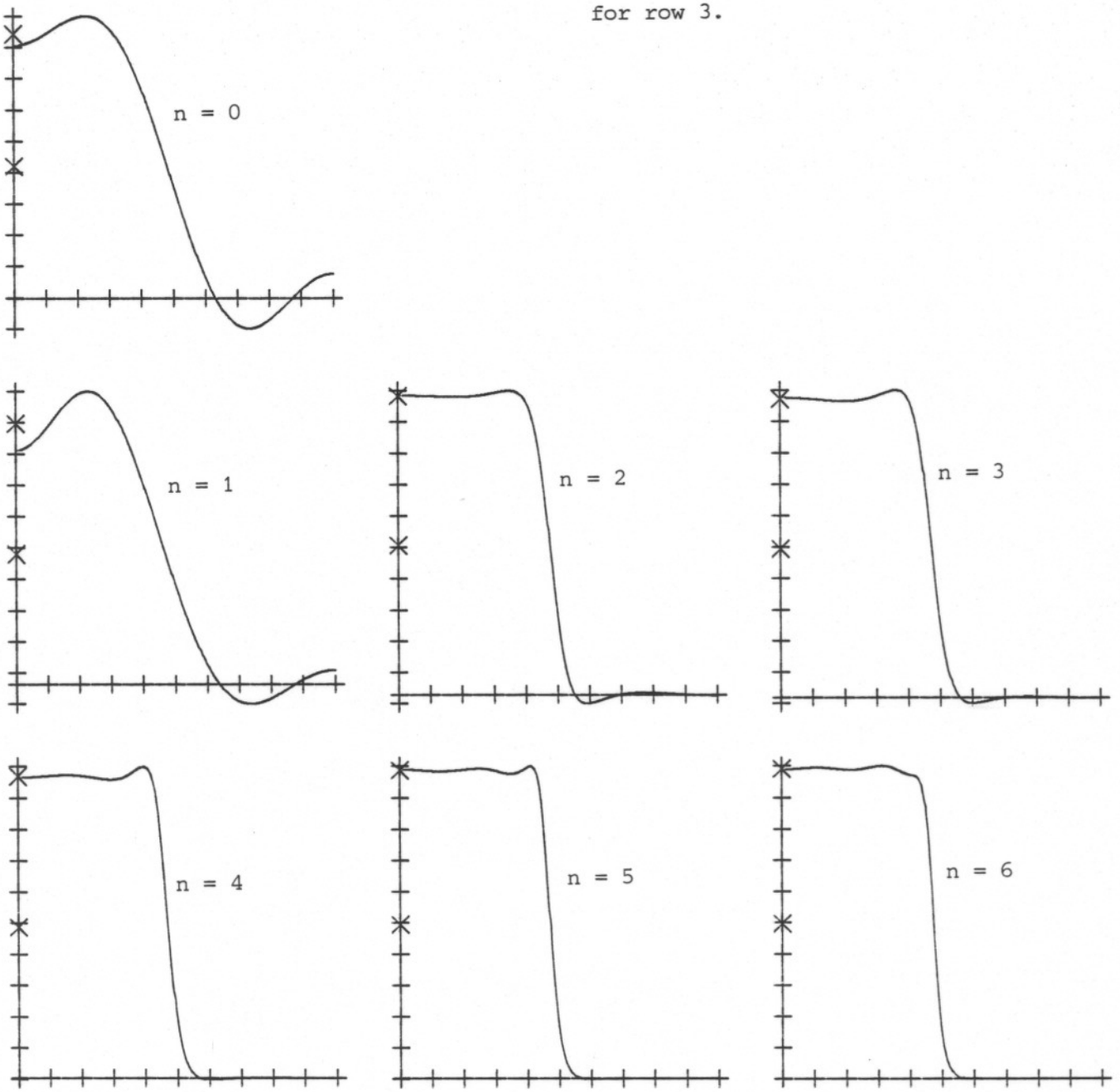


Fig. A.4

lowpass filter
Laurent-Padé approximants
for row 4.

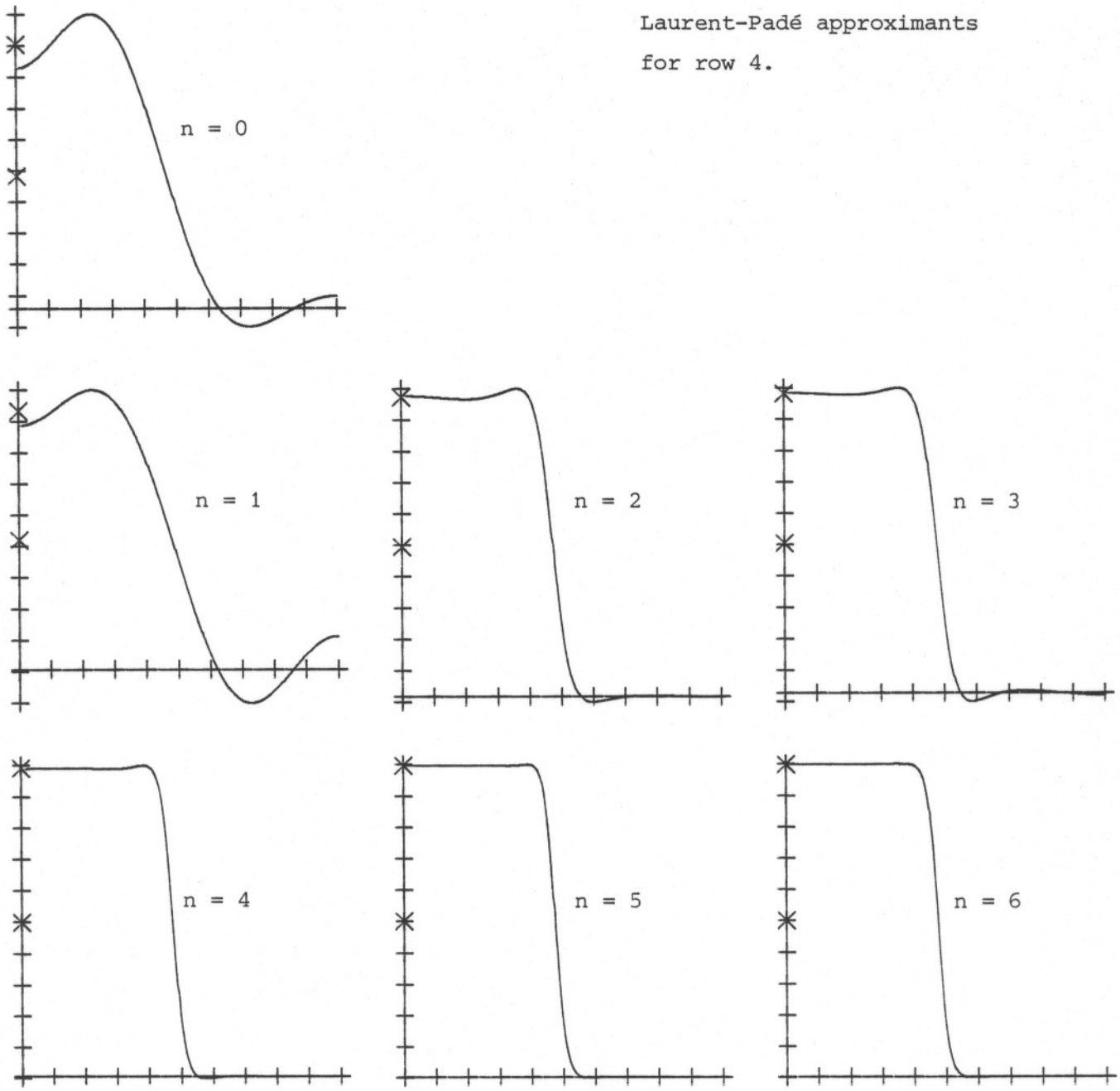


Fig. A.5

lowpass filter
Laurent-Padé approximants
for row 5.

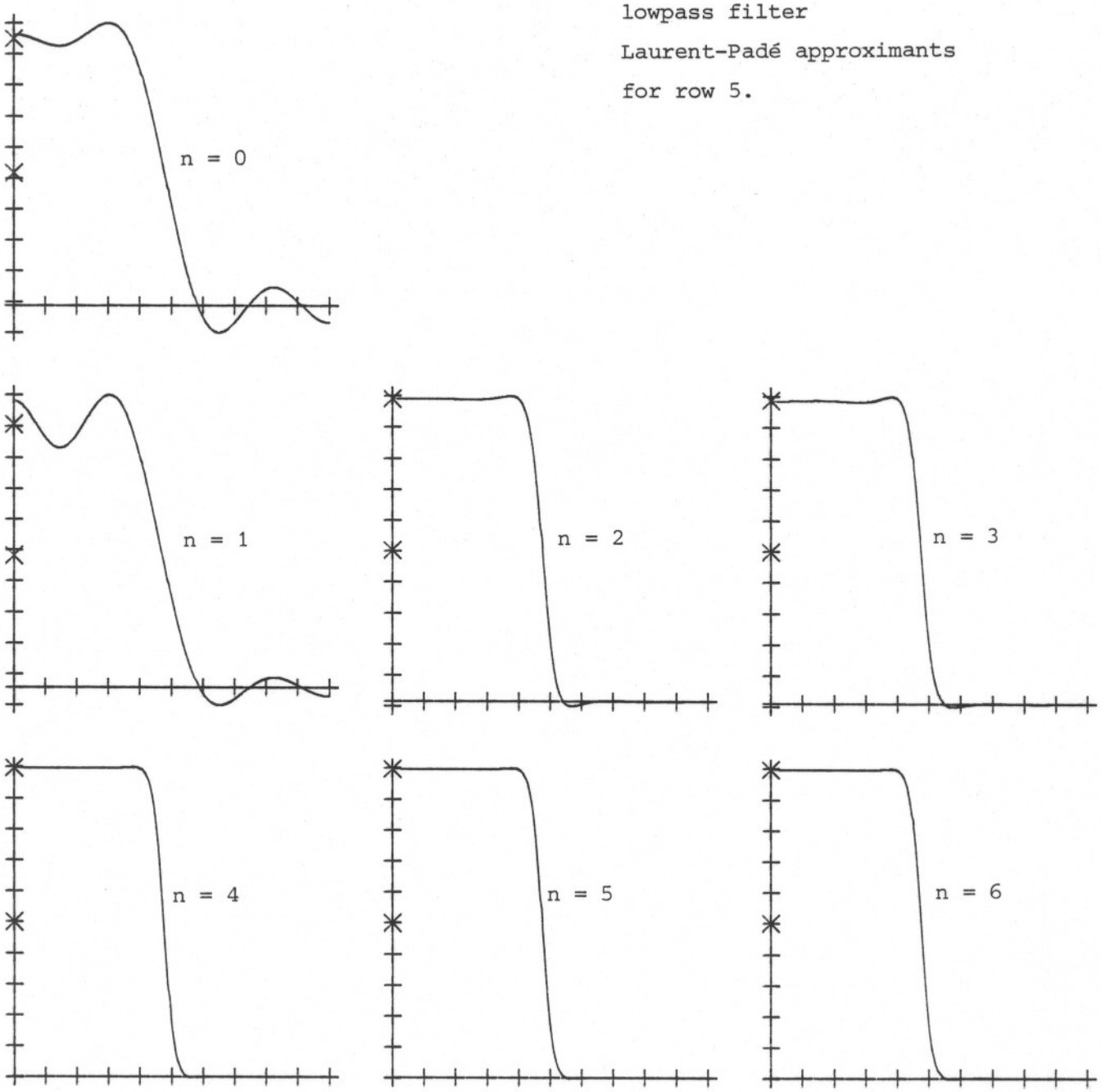


Fig. A.6

lowpass filter
Laurent-Padé approximants
for row 6.

