Comments on determining the number of zeros of a complex polynomial in a half-plane

Peeters, R.L.M.

1991

Faculteit der Economische Wetenschappen en Econometrie

Serie Research Memoranda

Comments on Determining the Number of Zeros of a Complex Polynomial in a Half-Plane *

R.L.M. Peeters

Research Memorandum 1991-31
April 1991

vrije Universiteit *amsterdam*

Comments on determining the number of zeros of a complex polynomial in a half-plane *

R.L.M. Peeters †

Abstract

We comment on recently proposed algorithms for determining the number of zeros of a complex polynomial in a half-plane, such as Agashe's method (1985) and Benidir and Picinbono's ERT (1991). Following an exposition of Talbot (1960) we construct an easier device, which we call "Talbot's Table" (TT), to replace the old Routh's Table (RT). Moreover, it is shown that the old RT is capable of answering stability questions even when it breaks down.

1 Introduction

In the last decade a number of articles appeared on the topic of determining the number of zeros of a (complex or real) polynomial in a half-plane. The motivation for this kind of research is two-fold : on the one hand there is the interest from a theoretical point of view, on the other hand we have a direct application of importance, namely the stability of a polynomial or matrix — a core topic in systems theory. For this application it is of interest also whether one is able to deduce the number of zeros of a polynomial that are on the imaginary axis (and their multiplicities), thus providing the engineer with a tool to distinguish between what is called (marginal) stability and asymptotic stability.

One century ago it was Routh [13], [14] who presented a method for calculating the number of zeros of a real polynomial in a half-plane. This method however contained some deficiencies in the sense that it was not generally applicable to any real polynomial but only to a restricted class. Two kinds of degeneracy could occur, of which one was solved quite easily (introducing a derivative operation), whereas the other however, turned out to be of a more fundamental nature. Many different strategies for removing this second singularity, among which the rather popular ϵ-method, have been proposed (see e.g. [4], [8], [14] and references given in [2], [3]), but all of them lacked the desired propety of general applicability. The same can be said about alternative treatments of the subject as initiated by Hurwitz [11] and Frobenius [5], [6]. Then, in 1985, Agashe [1] presented an algorithm that can deal with the most general

*Submitted in a comprised form to IEEE Transactions on Automatic Control.

†Address: Free University, Economics and Econometrics Department, De Boelelaan 1105, 1081 HV, Amsterdam, The Netherlands. E-mail: ralf@sara.nl.

case. Admittedly, his schemes are not as easy to apply as was Routh's Table (RT), but the matter seemed to have been settled. Surprisingly, work on the ϵ-method continued. Recently, in 1990 and 1991, Benidir and Picinbono [2], [3] came up with another method, which they called the Extended Routh's Table (ERT).
A striking aspect of the stream of literature of the last fifteen years however, is the fact that all the articles mentioned so far *do not refer to yet another basic treatment of the subject, presented in 1960 by Talbot* [15]. He gave a generally applicable algorithm that yields all desired information. The relation between the location of zeros of a polynomial and continued fraction expansions has been recognized already for a long time ([12], [16]). The latter subject being of importance in realization theory, we can find references where the results of Talbot are apparently well-known, see for instance Gragg and Lindquist [9] and Fuhrmann and Krishnaprasad [7]. Talbot's treatment, like the ones in [2] and [3], does not rely on Sturm's theorem but only on Euclid's division algorithm for polynomials. His proofs are of a very elegant and remarkably short nature. It is interesting to note that whereas the validity of the ERT was proven using properties of the Cauchy-index for rational functions, the validity of the TT is based on complex analysis only (with a key role played by Cauchy's principle of the argument), a version of Sturm's theorem being proven *en passant*. Furthermore, it is rather straightforward to show that Agashe's algorithm is essentialy identical to Talbot's, thus contradicting the use of the word "new" in the title of Agashe's paper.
In this note we shall construct a table, referred to as Talbot's Table (TT), from Talbot's algorithm and indicate a method of obtaining the number of right half-plane zeros from it. In the real, "normal" case the TT is seen to be identical to the ERT (both reducing to the old RT). In the complex "normal" case the TT and ERT are equivalent, whereas in the singular case the TT is shorter and easier to construct than the ERT. Following an exposition of Hanzon [10], we show how to obtain stability information about a matrix from the TT associated with its characteristic polynomial. As a final consequence we are able to show that this information can be obtained from the old RT also, even in case it breaks down.

2 Talbot's algorithm

Let $F(s)$ be a complex polynomial of degree $\delta F = n$. We are interested in calculating the number of right half-plane zeros (including multiplicities) of F, denoted by $r(F)$. It will be convenient to apply a rotation to the variable space, thus obtaining $f(s)$ from $F(s)$, defined by :

$$f(s) = i^n F(-is). \tag{2.1}$$

It is clear that the number $u(f)$ of upper half plane zeros of f satisfies $u(f) = r(F)$. We define the real polynomials $f_0(s)$ and $f_1(s)$ as the real and imaginary part of $f(s)$ respectively, so that

$$f(s) = f_0(s) + if_1(s), \tag{2.2}$$

We assume that $\delta f_0 \geq \delta f_1$. (If this does not hold, we can consider polynomial $if(s)$ instead of $f(s)$, which has the same zeros.)

Now, apply the H.C.F. algorithm to f_0 and f_1 to obtain their highest common factor $f_\mu = \mathrm{HCF}(f_0, f_1)$:

$$\begin{array}{lll} f_0(s) = q_1(s)f_1(s) - f_2(s) & \text{with} & \delta f_2 < \delta f_1 \leq \delta f_0 \\ \vdots & & \vdots \\ f_{k-1}(s) = q_k(s)f_k(s) - f_{k+1}(s) & & \delta f_{k+1} < \delta f_k \\ \vdots & & \vdots \\ f_{\mu-2}(s) = q_{\mu-1}(s)f_{\mu-1}(s) - f_\mu(s) & & \delta f_\mu < \delta f_{\mu-1} \\ f_{\mu-1}(s) = q_\mu(s)f_\mu(s) & & \end{array} \tag{2.3}$$

where all polynomials $f_2, \ldots, f_\mu$ and $q_1, \ldots, q_\mu$ are defined by the above scheme in a unique way, due to the requirements on the degrees of the f_k. Actually, the H.C.F. algorithm is a version of Euclid's algorithm for polynomials.
Let $q_k(s) = c_k s^{p_k} + \cdots$, so that $p_k = \delta f_{k-1} - \delta f_k$ and $\mathrm{sign}(c_k) = \mathrm{sign}(f_{k-1}^0 / f_k^0)$, where in general f_ℓ^0 denotes the leading coefficient of f_ℓ.
We then have, according to Talbot [15], (see also [7]) :

$$u(f) = u(f_\mu) + \frac{1}{2}\sum_{k=1}^{\mu}\left(p_k - \mathrm{sign}(c_k)\cdot\frac{1-(-1)^{p_k}}{2}\right). \tag{2.4}$$

Thus, $u(f) - u(f_\mu)$ can be obtained by mere inspection of the signs of the leading coefficients of the f_k, $k = 0, \ldots, \mu$, and the degrees δf_k. For a proof conform Talbot we refer to Appendix A.
The second part of Talbot's algorithm consists of an application of the following well-known lemma (see e.g. [12], [15] or [1] for a proof; to make this article self-contained there is also one added in Appendix A).

Lemma For any real polynomial $g(s)$ we have that

$$u(g) = u(g + ig'), \tag{2.5}$$

where the prime denotes differentiation with respect to s.

Thus, applying this lemma to f_μ, we can put $f_{\mu+1} := f'_\mu$ and restart the H.C.F. algorithm. This procedure is repeated over and over again until we arrive after a finite number of steps, say ν, at f_ν with $\delta f_\nu = 0$. We then have that $u(f)$ is given by :

$$\begin{aligned} u(f) &= \frac{1}{2}\sum_{k=1}^{\nu}\left(p_k - \mathrm{sign}(c_k)\cdot\frac{1-(-1)^{p_k}}{2}\right) \\ &= \frac{1}{2}(n - \sum_{k=1}^{\nu}\mathrm{sign}(c_k)\cdot\frac{1-(-1)^{p_k}}{2}). \end{aligned} \tag{2.6}$$

This shows how $u(f)$ (and thus also $r(F)$) can be obtained directly from the sequence of polynomials $f_0, \ldots, f_\nu$.
In fact the formula allows for an interpretation as follows. If we consider consecutive polynomials f_{k-1}, f_k, f_{k+1} that are related by the corresponding line in the H.C.F. scheme, we have by putting $h_k = f_{k-1} + if_k$ and $h_{k+1} = f_k + if_{k+1}$ that

$$u(h_k) = u(h_{k+1}) + \frac{1}{2}(p_k - \text{sign}(c_k) \cdot \frac{1-(-1)^{p_k}}{2}), \tag{2.7}$$

which is essentially Talbot's formula (8). The difference between $u(h_k)$ and $u(h_{k+1})$ can be interpreted as just $\frac{1}{2}p_k$ rounded to the nearest integer. For odd p_k there are two possibilities and it is the sign of c_k that determines which integer must be chosen. Since the difference in degrees between h_k and h_{k+1} is p_k one can think of it as that there are p_k zeros "disappearing", which are distributed as equal as possible over the upper and lower half-plane. For odd p_k the sign of c_k determines which half plane "receives" the remaining one. Here one should observe (as Talbot does) that all real zeros of $f(s)$ are zeros of $f_\mu(s)$.
One can obtain the number of real zeros of f and the number of lower half-plane zeros of f also from Talbot's algorithm. For this one must notice that f_μ is a real polynomial and therefore its number of upper half-plane zeros is equal to its number of lower half-plane zeros. This number $u(f_\mu)$ is obtained directly from Talbot's algorithm via formula (2.4) since in the end $u(f)$ is known. Then from $u(f_\mu)$ and δf_μ we can obtain the number of real zeros of f_μ which is equal to the number of real zeros of f. See also Hanzon [10] for a similar observation in case of Agashe's algorithm.

3 Construction of Talbot's Table

We propose the following construction of what we call Talbot's Table (TT), using the sequence of polynomials $f_0, \ldots, f_\nu$.
To polynomial f_k we associate row $k+1$ of the table. This row is filled with the coefficients of f_k, starting with its leading coefficient f_k^0 in the first column. We add two extra columns, which are filled in for $k > 0$. In the first of these we put p_k, i.e. the decrease in length when going from f_{k-1} to f_k, so from row k to $k+1$. In the second we put $\text{sign}(c_k) = \text{sign}(f_{k-1}^0/f_k^0)$ for those k where p_k is odd only. The values in these last two columns are added up. For the first extra column the result is of course n, and we assign the result of the second column to variable m. We then have that $r(F) = u(f) = \frac{1}{2}(n-m)$.

Example (Example 3 from Benidir and Picinbono [3].)
We have $F(s) = s^5 + s^4 + s + 1 + is^4$, whence $n = 5$. In Benidir and Picinbono's approach this leads to the construction of a table of 6 polynomials and even more intermediate ones. Using Talbot's algorithm we first calculate $f_0(s) = s^5 - s^4 + s$ and $f_1(s) = s^4 + 1$. Next we get :

$$f_0(s) = f_1(s)q_1(s) - f_2(s)$$

$$\text{with } q_1(s) = s - 1 \text{ and } f_2(s) = -1,$$

$$f_1(s) = f_2(s)q_2(s)$$

$$\text{with } q_2(s) = -s^4 - 1.$$

This gives the following TT :

k	polynomials f_k						p_k	sign(c_k)
0	1	-1	0	0	1	0		
1	1	0	0	0	1		1	1
2	-1						4	
						totals	$n = 5$	$m = 1$

Hence we find that $r(F) = \frac{1}{2}(5-1) = 2$. Notice that this table involves only three rows.

The additional work of filling in two extra columns to obtain the desired information is also present in Benidir and Picinbono's algorithm where one has to find the correct quantity h (formulas (2.6) and (2.11) in [3]).
The relation between Benidir and Picinbono's $A(s)$ and $B(s)$ and Talbot's $f_0(s)$ and $f_1(s)$ is given by $A(s) = (-1)^n f_0(-s)$ and $B(s) = -(-1)^n f_1(-s)$. It follows that if all p_k are 1, then Benidir and Picinbono's algorithm yields exactly the same polynomials when applied to $F(s)$ as Talbot's algorithm applied to $\bar{F}(s)$ (the polynomial in s with coefficients that are complex conjugates of the coefficients of $F(s)$). Notice that $r(F) = r(\bar{F})$.
Moreover, when all p_k are equal to 1 we have Routh's ("normal") case, and the sign changes in the first column of the TT determine the number r(F). This is seen from the fact that $\text{sign}(c_k) = \text{sign}(f^0_{k-1})\text{sign}(f^0_k)$, whence $p_k - \text{sign}(c_k)\frac{1-(-1)^{p_k}}{2}$ is equal to $1 - \text{sign}(f^0_{k-1})\text{sign}(f^0_k)$, which is zero if f^0_{k-1} and f^0_k have the same sign and two if their signs are different.

4 Application to stability

If we consider a complex matrix A, its stability properties depend not only on the location of the zeros of its characteristic polynomial, but also on the Jordan structure associated with its eigenvalues on the imaginary axis. As pointed out by Hanzon [10], it is possible to derive from intermediate results of the Agashe algorithm applied to the characteristic polynomial $F(s)$ of A conclusions about the stability of A. It is rather straightforward to show that Agashe's algorithm is essentially identical to Talbot's, their differences being on the level of notation only. Therefore as an immediate corollary we can draw conclusions about the stability of A if we apply Talbot's algorithm to $F(s)$. We have, if we write $f_\mu = HCF(f_0, f_1)$ and $f_\lambda = HCF(f_\mu, f'_\mu)$ (or $f_\lambda = 1$ if $\delta f_\mu = 0$) :

1. A has no eigenvalues in the open right half plane if and only if $u(f) = 0$.

2. A is asymptotically stable (all its eigenvalues are in the open left half plane, or equivalently $\lim_{t\to\infty} e^{tA} = 0$) if and only if $u(f) = 0$ and $\delta f_\mu = 0$.

3. A has no eigenvalues in the open right half plane and its eigenvalues on the imaginary axis have multiplicity one if and only if $u(f) = 0$ and $\delta f_\lambda = 0$.

4. A is stable (none of its eigenvalues are in the open right half plane and to its eigenvalues on the imaginary axis correspond only diagonal Jordan blocks, or

equivalently e^{tA} is bounded for $t \to \infty$) if and only if $u(f) = 0$ and $K(A) = 0$, where $K(s) = i^{-n+\delta f_\lambda} k(is)$ with $k(s) = f(s)/f_\lambda(s)$.

5. A is unstable if and only if $u(f) \neq 0$, or $u(f) = 0$ and $K(A) \neq 0$. This is an immediate consequence of the foregoing.

The key argument in the proof of these statements lies in the observation that f_μ is the HCF of f_0 and f_1 and thus contains all real zeros of $f(s)$ and all pairs of zeros that lie symmetric with respect to the real axis. Morever one must notice that the zeros of f_λ are the same as those of f_μ but with multiplicities decreased by one. Hence, if $u(f) = 0$ we have that f_μ has real zeros only, and the same is true for f_λ. Notice that if $u(f) \neq 0$, we will find some contribution in the first λ steps. Therefore, when addressing stability questions only it is sufficient to run Talbot's algorithm until f_λ has been obtained.
Investigation of the structure of Benidir and Picinbono's algorithm shows that in their case we can draw similar conclusions (they only present the first three) because if $u(f) = 0$ we necessarily have that $p_k = 1$ throughout so that the ERT is essentially the TT applied to $\bar{F}(s)$ instead of $F(s)$.
A final remark can be made about the original RT. When Routh's algorithm breaks down (i.e. some p_k is larger than one) and if we are addressing stability questions only, then there must be zeros in both half-planes (due to the equal distribution of "disappearing" zeros discussed before), so $u(f) \neq 0$. Thus A is unstable, and all situations where Routh's algorithm does not break down can still be treated as shown above. This shows that the RT is still useful for this application.

Example
Consider the following two matrices A and B :

$$A = \begin{pmatrix} 31 & 0 & 24 & -\frac{8}{3} & 6 & 0 & -\frac{16}{3} \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -40 & 0 & -31 & -\frac{10}{3} & -8 & 0 & \frac{20}{3} \\ -12 & 0 & -9 & -1 & 0 & -2 & 0 \\ 5 & 0 & 4 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 1 \\ 12 & 0 & 9 & 0 & 0 & 1 & -1 \end{pmatrix},$$

$$B = \begin{pmatrix} 31 & 0 & 24 & -\frac{17}{3} & 6 & 0 & -\frac{34}{3} \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -40 & 0 & -31 & \frac{22}{3} & -8 & 0 & \frac{44}{3} \\ -12 & 0 & -9 & -1 & 0 & -2 & 0 \\ 5 & 0 & 4 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 1 \\ 12 & 0 & 9 & 0 & 0 & 1 & -1 \end{pmatrix}.$$

Straightforward calculation shows that both matrices have the same characteristic polynomial

$$F(s) = \det(sI - A) = \det(sI - B) = s^7 + 3s^6 + 6s^5 + 8s^4 + 9s^3 + 7s^2 + 4s + 2.$$

This leads to :

$$f(s) = i^7 F(-is) = s^7 + 3is^6 - 6s^5 - 8is^4 + 9s^3 + 7is^2 - 4s - 2i,$$

whence

$$f_0(s) = s^7 - 6s^5 + 9s^3 - 4s,$$

$$f_1(s) = 3s^6 - 8s^4 + 7s^2 - 2.$$

Application of Talbot's algorithm gives the following TT :

k	polynomials f_k								p_k	sign(c_k)
0	1	0	-6	0	9	0	-4	0		
1	3	0	-8	0	7	0	-2		1	1
2	$\frac{10}{3}$	0	$-\frac{20}{3}$	0	$\frac{10}{3}$	0			1	1
3	2	0	-4	0	2				1	1
4	8	0	-8	0					1	1
5	2	0	-2						1	1
6	4	0							1	1
7	2								1	1
								totals	$n = 7$	$m = 7$

Thus we find that $u(f) = \frac{1}{2}(n - m) = 0$. Notice that the number of sign changes in the first column of the f_k polynomials is indeed zero. We have that all p_k are equal to 1, which must be the case if $u(f) = 0$. Notice also that for a real polynomial $F(s)$ the decomposition of $f(s)$ into its real and imaginary part corresponds to the decomposition of $F(s)$ into its even and odd part, as usual.
In the above scheme we have that the algorithm was restarted via $f_4(s) = f_3'(s)$ and $f_6(s) = f_5'(s)$. Hence we have that :

$$f_\mu = \text{HCF}(f_0, f_1) = f_3, \quad f_3(s) = 2s^4 - 4s^2 + 2$$

and

$$f_\lambda = \text{HCF}(f_\mu, f_\mu') = f_5, \quad f_5(s) = 2s^2 - 2.$$

Because $\delta f_\lambda \neq 0$ we can conclude that $F(s)$ must have purely imaginary zeros with multiplicity larger than one. In fact this multiplicity is equal to 2, which follows from the fact that $F(s)$ is a real polynomial so that its zeros are either real or occur in complex conjugate pairs. From the fact that $u(f) = 0$ we already have that f_μ only contains real zeros and since $\delta f_\mu = 4$ we know that $F(s)$ contains 4 purely imaginary zeros.
This means that if we want to establish the stability properties of A and B, we must calculate $f(s)/f_\lambda(s)$. This gives :

$$k(s) = f(s)/f_\lambda(s) = \frac{1}{2}s^5 + \frac{3}{2}is^4 - \frac{5}{2}s^3 - \frac{5}{2}is^2 + 2s + i.$$

We then rotate the variable space back, so that we get :

$$K(s) = F(s)/F_\lambda(s) = i^{-5}f(is)/f_\lambda(is) = \frac{1}{2}s^5 + \frac{3}{2}s^4 + \frac{5}{2}s^3 + \frac{5}{2}s^2 + 2s + 1.$$

Substitution of A and B in the polynomial K gives :

$$K(A) = 0$$

so that A is indeed a stable matrix, but

$$K(B) \neq 0$$

whence B is unstable.

One can verify that A has a diagonal Jordan form whereas B does not. For verification purposes we mention the zeros of $F(s)$. These are $-1, -1+i, -1-i, i, i, -i, -i$, so that $F(s)$ can be decomposed as

$$F(s) = (s+1)(s+1-i)(s+1+i)(s-i)(s-i)(s+i)(s+i).$$

5 Conclusions

Talbot's algorithm provides a nice tool for determining the number of right half-plane zeros of a complex polynomial (as well as the numbers of purely imaginary zeros and left half-plane zeros). In the "normal" case it becomes equivalent to the old RT, as does the newly proposed ERT. In the "singular" case it is shorter than the ERT. Agashe's algorithm is equivalent to Talbot's. Of course, there is no longer need for the classical ϵ-method. From the TT applied to the characteristic polynomial of a matrix one can determine stability and asymptotic stability properties. From the interpretation of what happens when Routh's original algorithm breaks down, one can obtain the same answers with respect to stability questions already from the RT.

Appendix A : Validity of the TT

In this Appendix we present a proof of the correctness of the TT, via formula (2.7) which describes the effect of one step of the H.C.F. algorithm, and via the Lemma of Section 2. Both of these proofs follow the original lines of Talbot [15] and are merely added to make this article self-contained. However, a minor correction with respect to the first proof had to be made.

Let us denote by A_X the real open interval $(-X, X)$ in the complex s-plane, by S_X the upper semicircle on A_X and by U_X the open region bounded by A_X and S_X. As X tends to infinity, A_X becomes the real axis A and U_X the upper half-plane U.

Suppose that f_{k-1}, f_k and f_{k+1} are consecutive polynomials in the TT, so that they are related by :

$$f_{k-1}(s) = f_k(s)q_k(s) - f_{k+1}(s), \tag{A.1}$$

with $\delta f_{k+1} < \delta f_k \leq \delta f_{k-1}$. Of course, $\delta q_k = \delta f_{k-1} - \delta f_k$ and we denote this quantity as before by p_k. From the identity above we can obtain :

$$f_{k-1} + if_k = (q_k + i)(f_k + if_{k+1}) - iq_k f_{k+1}, \tag{A.2}$$

of which the r.h.s. can be written formally as

$$(q_k + i)(f_k + if_{k+1})(1 - \alpha) \tag{A.3}$$

with

$$\alpha = \frac{q_k}{q_k + i} \cdot \frac{1}{1 - if_k/f_{k+1}}. \tag{A.4}$$

Since q_k, f_k and f_{k+1} are real polynomials it follows that on A_X we always have that $|\alpha(s)| < 1$.
Since $\delta f_{k+1} < \delta f_k$ we find on S_X for $X \to \infty$ that $\alpha(s) \to 0$ (uniformly with X). Therefore, for sufficiently large X we find that $|\alpha(s)| < 1$ on the boundary of U_X.
According to Cauchy's Principle of the Argument we then find that, for large enough X, the number of poles of $(1 - \alpha)$ is equal to the number of zeros of $(1 - \alpha)$. Thus,

$$u(f_{k-1} + if_k) = u(f_k + if_{k+1}) + u(q_k + i) \tag{A.5}$$

which is clear from writing $(1 - \alpha)$ as $\frac{f_{k-1}+if_k}{(q_k+i)(f_k+if_{k+1})}$.
We next proceed to calculate $u(q_k + i)$. Again according to the Argument Principle we have for large enough X that

$$u(q_k + i) = \frac{1}{2\pi}\Delta\arg(q_k + i), \tag{A.6}$$

where Δ denotes the increment in a positive description of the boundary S_X, A_X of U_X. (Of course $(q_k + i)$ has no poles.) We find :

$$u(q_k + i) = \frac{1}{2\pi}(p_k\pi - \frac{1}{2}\text{sign}(c_k)\pi + \frac{1}{2}\text{sign}((-1)^{p_k}c_k)\pi), \tag{A.7}$$

where, as in Section 2, c_k denotes the leading coefficient of q_k. (Here it is convenient to treat the case where $\delta q_k = 0$, which can only occur in the first step of the TT, so for $k = 1$, separately.) The first term in the expression between brackets is the contribution of boundary segment S_X, the other two come from A_X, where it is noticed that $q_k(s) + i$ with $s \in A_X$ lies entirely in the upper half-plane.
We can write our final result as :

$$u(f_{k-1} + if_k) = u(f_k + if_{k+1}) + \frac{1}{2}(p_k - \text{sign}(c_k)\frac{1 - (-1)^{p_k}}{2}), \tag{A.8}$$

which proves formula (2.7) and by summation over $k = 1, \dots, \mu$ we obtain formula (2.4).

The second thing we want to prove is the Lemma of Section 2. For this purpose, let ξ be a (possibly complex) zero of the real polynomial g of multiplicity κ, say. Thus, $g(s) = (s - \xi)^\kappa h(s)$ for some complex polynomial $h(s)$ satisfying $h(\xi) \neq 0$. Obviously, ξ is a $(\kappa - 1)$-fold zero of g' and therefore also a $(\kappa - 1)$-fold zero of $g + i\epsilon g'$, for any nonzero value of ϵ. Suppose that ϵ is real, positive and close to zero. Then, in addition $g + i\epsilon g'$ will have a zero $\xi + \eta$ where $\eta = O(\epsilon)$. In fact we have that

$$\begin{aligned} 0 &= g(\xi+\eta) + i\epsilon g'(\xi+\eta) \\ &= \eta^\kappa h(\xi+\eta) + i\epsilon(\kappa\eta^{\kappa-1}h(\xi+\eta) + \eta^\kappa h'(\xi+\eta)) \\ &= \eta^{\kappa-1}\{(i\epsilon\kappa+\eta)h(\xi+\eta) + i\epsilon\eta h'(\xi+\eta)\} \\ &= \eta^{\kappa-1}\{(i\epsilon\kappa+\eta)[h(\xi)+\eta h'(\xi)+O(\eta^2)] + i\epsilon\eta[h'(\xi)+O(\eta)]\} \\ &= \eta^{\kappa-1}\{(i\epsilon\kappa+\eta)h(\xi) + i\epsilon(\kappa+1)\eta h'(\xi) + O(\eta^2)\} \end{aligned} \tag{A.9}$$

whence

$$\eta = -i\epsilon\kappa + O(\epsilon^2). \tag{A.10}$$

This shows that the zeros of $g + i\epsilon g'$ are either the same as the corresponding zeros of g or else have lower imaginary parts. Thus

$$u(g) = u(g + i\epsilon g'). \tag{A.11}$$

From the H.C.F. algorithm applied to $g + i\epsilon g'$ we have as an immediate corollary (also from Talbot) that

$$u(g + i\epsilon g') = u(g + ig'), \tag{A.12}$$

since the corresponding TT's are related by that the rows of one of them can be expressed as (alternatingly) ϵ and $\frac{1}{\epsilon}$ times the rows of the other; moreover we can already use formula (2.4).
This completes the proof of the lemma and as a corollary we obtain the validity of the TT.

Appendix B : Computer codes

Below we list MATLAB codes for calculating the number of right half-plane zeros of a complex polynomial. The main routine is provided by function TALBOT(F) which generates the number of right half-plane zeros, the number of real zeros and the corresponding TT with respect to polynomial F. The other listings give routines that are invoked by function TALBOT. Added also is a separate routine, called UHP_ROOTS for calculating the number of upper half-plane zeros of a complex polynomial.

```
function [m,ℓ,TT] = talbot(F)
%
% Function TALBOT.
%
% Via this function we calculate the number of right half-plane roots
% of the (complex) polynomial equation F(s) = 0.
% The coefficients of F must be stored in variable F according to
% MATLAB's standard convention, i.e. the first component F(1) of F
```

```
% denotes the coefficient of the highest power of s and the last
% component F(n + 1) denotes the constant term. (Here F(s) is assumed to
% be of degree n, so represented by an (n + 1)-vector.)
% We follow Talbot's algorithm (1960), which is equivalent to Agashe's
% (1985).
% The first argument m of the output denotes the number of RHP-roots, the
% second denotes the number ℓ of imaginary roots. Of course the number of
% left half-plane roots can be calculated as n - m - ℓ.
% In output variable TT we store the resulting TT (Talbot's Table).
% We make use of subroutines (functions) DEG, DERIV and EUCL_STEP.
% This routine is a modified version of routine UHP_ROOTS.
%
% Programmed by Ralf Peeters, Free University, Amsterdam, April 1991.
%

eps=1e-10;                          % for controlling machine round-off.
F=F(:).';
inz=find(abs(F));                   % find the first nonzero coefficient.
F=F(inz(1):max(size(F)));
f=F;                                % f is calculated as (i^ n)*F(-i*s),
n=max(size(F))-1;                   % for this we distinguish four cases.
for j=1:n+1,
    jmod4=j-4*round(j/4-0.5);
    if jmod4==2,
        f(j)=i*F(j);
    end;
    if jmod4==3,
        f(j)=-F(j);
    end;
    if jmod4==0,
        f(j)=-i*F(j);
    end;
end;
fn=f(1);                            % consider the leading coefficient of f.
if abs(real(fn))<eps,               % when necessary, reverse the real
    f=f*i;                          % and imaginary part of f.
end;
p=real(f);                          % the real and imaginary part of f
q=imag(f);                          % are displayed on screen.
n=deg(p);
TT=zeros(1,n+3);                    % first row of the TT.
TT(1,1:n+1)=p;
k=0;
while norm(q)>eps,                  % first round of Talbot's algorithm.
    [b,q,r,v]=eucl_step(p,q);
    e=1;
```

```
    if b==2*round(b/2),
        e=-1;
    end
    k=k+v*(1+e)/2;
    nn=deg(q);
    TT=[TT; q, zeros(1:n-nn),b,v*(1+e)/2];       % updating of the TT.
    p=q;
    q=r;
end
n1=n-deg(p);                                   % n1 denotes the drop in degree.
m1=(n1-k)/2;                                   % m1 denotes the number of RHP-roots
while deg(p)>0,                                % found in the first round.
    q=deriv(p);                                % restart of the algorithm.
    while norm(q)>eps,                         % next round.
        [b,q,r,v]=eucl_step(p,q);
        e=1;
        if b==2*round(b/2),
            e=-1;
        end
        k=k+v*(1+e)/2;
        nn=deg(q);
        TT=[TT; q, zeros(1:n-nn),b,v*(1+e)/2];   % updating of the TT.
        p=q;
        q=r;
    end
end
m=(n-k)/2;                                     % m = number of RHP-roots.
l=n-n1-2*(m-m1);                               % ℓ = number of imaginary roots.

%
% End of function TALBOT.
```

```
function [b,q,r,v] = eucl_step(p,q)
%
% Function EUCL_STEP.
%
% In this function we perform one step of the Euclidean division
% algorithm for polynomials.
% Input are two vectors p and q corresponding to polynomials following
% MATLAB's convention. (See e.g. UHP_ROOTS for an explanation.)
% Output are the quantities b, q, r and v, denoting (respectively):
%   b: degree of the quotient (= degree of p - degree of q),
%   q: denoting the original polynomial q, which will take p's place,
```

```
%  r: denoting the remainder polynomial, which will take q's place,
%  v: denoting the sign of the quotient polynomial
%
% Programmed by Ralf Peeters, Delft University of Technology, January 1989,
%                                revised at Free University, Amsterdam, April 1991.
%

eps=norm(p)*1e-8;                        % for controlling machine round-off.
r=-p;
inz=find(abs(q));                        % find the leading coefficient of q.
q=q(inz(1):max(size(q)));
c=deg(p);
d=deg(q);
b=c-d;
v=sign(q(1))*sign(p(1));
for i=1:b+1,                             % perform division "with remainder"
    a=r(i)/q(1);                         % to p and q. Remainder is stored in r.
    for j=1:d+1,
        r(i+j-1)=r(i+j-1)-a*q(j);
        if abs(r(i+j-1))<eps,            % use eps to remove leading "zeros"
            r(i+j-1)=0;                  % by setting them to zero exactly.
        end
    end
end

%
% End of function EUCL_STEP.
```

```
function [d] = deg(f)
%
% Function DEG.
%
% To calculate the degree of a polynomial with nonzero leading
% coefficient. For use in UHP_ROOTS and TALBOT.
%
% Programmed by Ralf Peeters, Delft University of Technology, January 1989.
%

d=max(size(f))-1;

%
% End of function DEG.
```

```
function [g] = deriv(f)
%
% Function DERIV.
%
% For calculation of the derivative of the polynomial argument f.
% To be used in UHP_ROOTS and TALBOT.
%
% Programmed by Ralf Peeters, Delft University of Technology, January 1989.
%

n=max(size(f));
for i=1:n-1,
g(i)=f(i)*(n-i);
end;

%
% End of function DERIV.
```

```
function [m,ℓ] = uhp_roots(f)
%
% Function UHP_ROOTS.
%
% Via this function we calculate the number of upper half-plane roots
% of the (complex) polynomial equation f(s) = 0.
% The coefficients of f must be stored in variable f according to
% MATLAB's standard convention, i.e. the first component f(1) of f
% denotes the coefficient of the highest power of s and the last
% component f(n + 1) denotes the constant term. (Here f(s) is assumed to
% be of degree n, so represented by an (n + 1)-vector.)
% We follow Talbot's algorithm (1960), which is equivalent to Agashe's
% (1985).
% The first argument m of the output denotes the number of UHP-roots, the
% second denotes the number ℓ of real roots. Of course the number of
% lower half-plane roots can be calculated as n - m - ℓ.
% We make use of subroutines (functions) DEG, DERIV and EUCL_STEP.
%
% Programmed by Ralf Peeters, Delft University of Technology, January 1989,
%                                  revised at Free University, Amsterdam, April 1991.
%
```

```
eps=1e-10;                          % for controlling machine round-off.
f=f(:).';
inz=find(abs(f));                   % find the first nonzero coefficient.
f=f(inz(1):max(size(f)));
fn=f(1);                            % the first coefficient is put to 1.
f=f/fn;
p=real(f)                           % the real and imaginary part of f
q=imag(f)                           % are displayed on screen.
n=deg(p);
k=0;
while norm(q)>eps,                  % first round of Talbot's algorithm.
    [b,q,r,v]=eucl_step(p,q);
    e=1;
    if b==2*round(b/2),
        e=-1;
    end
    k=k+v*(1+e)/2;
    p=q;
    q=r;
end
n1=n-deg(p);                        % n1 denotes the drop in degree.
m1=(n1-k)/2;                        % m1 denotes the number of UHP-roots
while deg(p)>0,                     % found in the first round.
    q=deriv(p);                     % restart of the algorithm.
    while norm(q)>eps,              % next round.
        [b,q,r,v]=eucl_step(p,q);
        e=1;
        if b==2*round(b/2),
            e=-1;
        end
        k=k+v*(1+e)/2;
        p=q;
        q=r;
    end
end
m=(n-k)/2;                          % m denotes the number of UHP-roots.
l=n-n1-2*(m-m1);                    % ℓ denotes the number of real roots.

%
% End of function UHP_ROOTS.
```

References

[1] S.D. Agashe, A New General Routh-Like Algorithm to Determine the Number of RHP Roots of a Real or Complex Polynomial, *IEEE Transactions on Automatic Control*, Vol. AC-30, No. 4, pp. 406–409, 1985.

[2] M. Benidir and B. Picinbono, Extended Table for Eliminating the Singularities in Routh's Array, *IEEE Transactions on Automatic Control*, Vol. AC-35, No. 2, pp. 218–222, 1990.

[3] M. Benidir and B. Picinbono, The Extended Routh's Table in the Complex Case, *IEEE Transactions on Automatic Control*, Vol. AC-36, No. 2, pp. 253–256, 1991.

[4] E. Frank, On the zeros of polynomials with complex coefficients, *Bulletin of the American Mathematical Society*, Vol. 52, pp. 144-157, 1946.

[5] F.G. Frobenius, Über das Trägheitsgesetz der quadratischen Formen, *Sitzungsberichte der Königlich Preußischen Akademie der Wissenschaften zu Berlin*, pp. 241–256, 407–431, 1894; also in: *Journal für die reine und angewandte Mathematik*, Vol. 114, pp. 187–230, 1895; also in: *Gesammelte Abhandlungen*, Band II, pp. 591–631.

[6] F.G. Frobenius, Über das trägheitsgesetz der quadratischen Formen II, *Sitzungsberichte der Königlich Preußischen Akademie der Wissenschaften zu Berlin*, pp. 657–663, 1906; also in: *Gesammelte Abhandlungen*, Band III, pp. 387–393.

[7] P.A. Fuhrmann and P.S. Krishnaprasad, Towards a Cell Decomposition for Rational Functions, *IMA Journal of Mathematical Control & Information*, Vol. 3, pp. 137–150, 1986.

[8] F.R. Gantmacher, *The Theory of Matrices*, Vol. I: pp. 294–304, 331–348, Vol. II: pp. 172–250. New York: Chelsea, 1959.

[9] W.B. Gragg and A. Lindquist, On the Partial Realization Problem, *Linear Algebra and its Applications*, Vol. 50, pp. 277–319, 1983.

[10] B. Hanzon, Some new Results on and Applications of an Algorithm of Agashe, in: R.F. Curtain (ed.), *Modelling, Robustness and Sensitivity Reduction in Control Systems*, NATO ASI Series F: Computer and Systems Sciences Vol. 34, pp. 285–303. Berlin: Springer-Verlag, 1987.

[11] A. Hurwitz, Über die Bedingungen, unter welchen eine Gleichung nur Wurzeln mit negativen reellen Teilen besitzt, *Mathematische Annalen*, Vol. 46, pp. 273–284, 1895; also in: *Mathematische Werke, Band II : Zahlentheorie, Algebra, Geometrie*, pp. 533–545.

[12] M. Marden, *The geometry of the Zeros of a Polynomial in a Complex Variable*. Providence, RI: American Mathematical Society, 1949.

[13] E.J. Routh, *A Treatise on the Stability of a Given State of Motion Particularly Steady Motion.* New York: Macmillan, 1877.

[14] E.J. Routh, *The Advanced Part of a Treatise on the Dynamics of a System of Rigid Bodies*, 6th ed., revised and enlarged. New York: Macmillan, 1905.

[15] A. Talbot, The number of zeros of a polynomial in a half-plane, *Proceedings of the Cambridge Philosophical Society*, Vol. 56, pp. 132–147, 1960.

[16] H.S. Wall, *Analytic Theory of Continued Fractions.* New York: Van Nostrand, 1948.

1990-1	B. Vogelvang	Testing For Co-Integration with Spot Prices of Some Related Agricultural Commodities
1990-2	J.C.J.M. van den Bergh P. Nijkamp	Ecologically Sustainable Economic Development Concepts and Model Implications
1990-3	J.C.J.M. van den Bergh P. Nijkamp	Ecologically Sustainable Economic Development in a Regional System: A Case Study in Agri cultural Development Planning in the Netherlands
1990-4	C.Gorter P.Nijkamp P.Rietveld	Employers' Recruitment Behaviour and Re-Employment Probabilities of Unemployed
1990-5	K.Burger	Off-farm income and the farm-household the case of Kenyan smallholders
1990-6	H. Visser	Crowding out and the Government Budget
1990-7	P. Rietveld	Ordinal Data in Multicriteria Decision Making, a Stochastic Dominance Approach to Siting Nuclear Power Plants
1990-8	G. van der Laan P.H.M. Ruys D.J.J. Talman	Signaling devices for the supply of semi-public goods
1990-9	F.A.G. den Butter	Labour Productivity Slowdown and Technical Progress: An empirical analysis for The Netherlands
1990-10	R.W. van Zijp	Neo-Austrian Business Cycle Theory
1990-11	J.C. van Ours	Matching Unemployment and Vacancies: The Efficiency of the Dutch Labour Market
1990-12	B. Vogelvang	Hypotheses Testing Concerning Relationships between Spot Prices of Various Types of Coffee
1990-13	A.F. de Vos I.J. Steyn	Stochastic Nonlinearity: A Firm Basis for the Flexible Functional Form
1990-14	Y.H. van Emmerik D. de Jong W.W.A. Zuurmond D.N. Dukkers-van Emden	Opereren in overleg: geprotocolleerde samenwerking 1e-2e-lijn bij dagchirurgie
1990-15	T.J.J.B. Wolters	Mediation and Collective Bargaining: A Diagnostic Approach
1990-16	E.M.A. Scholten J. Koelewijn	Financieringsproblematiek van startende ondernemingen: een mogelijke verklaring op basis van empirisch onderzoek.
1990-17	E. Hüner H.P. Smit	Saturation and Model Specification of Passenger car Ownership
1990-18	F.A.G. den Butter	Sociale zekerheid, de wig en economische groei
1990-19	F.A.G. den Butter R.F. v.d. Wijngaert	Who is Correcting the Error ? A Co-integration Approach for Wages, Wage Space and Labour Conflicts in the Netherlands
1990-20	J.P. de Groot R. Ruben	Sistemas de Producción y Transferencia de Tecnología en la Economía Cafetalera de Centroamerica
1990-21	R. Ruben	Campesinado y Reforma Agraria en El Salvador
1990-22	J. van Ours G. Ridder	Vacancies and the Recruitment of New Employees
1990-23	A.F. de Vos J.J. de Vries	The Likelihood Function of a Generalized Gravity Model; Handling the Implicit Singularity of a Nonlinear Transformation
1990-24	D. van der Wal	Geloofwaardigheid en het Nederlandse wisselkoersbeleid
1990-25	R.J. Veldwijk M. Boogaard M.V. van Dijk E.R.K. Spoor	EDSOs, implosion and explosion: concepts to automate a part of application maintenance
1990-26	B. Hanzon	The area enclosed by the (oriented) Nyquist diagram and the Hilbert-Schmidt-Hankel norm of a linear system
1990-27	R.W. van Zijp	Why Lucas is not a Hayekian
1990-28	J. Rouwendal	On discrete choice under uncertainty: A generalization of the logit model and its application
1990-29	J. Rouwendal	On the equitable distribution of the housing stock
1990-30	J. Rouwendal	Stochastic market equilibria with efficient rationing (with an application to the Dutch housing market)
1990-31	J.A. Vijlbrief	The effects of unemployment insurance on the labour market
1990-32	J.G.W. Simons H.P. Wansink	Traffic Ban, a means to combat smog?
1990-33	J.C. van Ours T. Zoethout	De Interne Arbeidsmarkt van de Gemeente Amsterdam
1990-34	H.J. Bierens	A Note On the Limiting Distribution of Sample Autocorrelations in the Presence of a Unit Root
1990-35	T. Kuhlman	The Economic Integration of Refugees in Developing Countries: A Research Model.
1990-36	T. Kuhlman	Towards a Definition of Refugees.
1990-37	T. Kuhlman	Organized versus Spontaneous Settlement of Refugees in Africa.
1990-38	R. Zuidema	The Neo-Austrian View on Interest
1990-39	G.v.d.Laan	General Equilibrium in a Closed International Trade Model