CS9

THE QD-ALGORITHM AS A METHOD FOR FINDING THE ROOTS OF A POLYNOMIAL EQUATION WHEN ALL ROOTS ARE POSITIVE

BY

CHR. ANDERSEN

TECHNICAL REPORT CS9
JUNE 8, 1964

COMPUTER SCIENCE DIVISION
School of Humanities and Sciences
STANFORD UNIVERSITY

TABLE OF CONTENTS

Acknowledgements.

This paper has been worked out during my visit at Stanford. I wish to express my sincere thanks to the Danish committee for NATO and OECD stipends, to the board and the director N. I. Bech of Regnecentralen, Denmark, and to Professor George E. Forsythe, Stanford, who all helped me by sponsoring my journey and my stay in California.

Furthermore I want to thank my friend Tom Robertson, Professors Gene Golub and George Forsythe for their encouragement during my work.

Mrs. Rita Nyland and Mrs. Glee Hollander have helped very much by running the programs on the B 5000 and by typing the manuscript, respectively, and I wish to express my appreciation to both.

Chr. Andersen

Introduction,

The QD-algorithm — which stands for the quotient-difference algorithm — has been developed by H. Rutishauser, In several papers, the first of which appeared in 1954, Rutishauser has treated the theory and a number of applications of the algorithm. In this treatment the theory is based on properties of continued fractions.

In 1958 Peter Henrici based the theory of the m-algorithm on the theory of analytic functions. Furthermore Henrici gave some new results,,

The present article is a new introduction to the subject. In this paper the theory of the @-algorithm is treated by means of classical algebraic methods. The present paper however treats only a part of this theory. Although some of the results developed are general the main part of the paper is limited to a special case which, as indicated in the title, may be described as the part of the theory of the QD-algorithm needed for finding the roots of a polynomial the roots of which are known to be positive, by means of the algorithm.

With this limitation it is possible to prove some important results which cannot be proved in the general case, First the existence question of the QD-scheme can be solved; that is the QD-scheme will always exist in the case of positive roots — as may be shown by examples this is not true in the general case.

Furthermore the question of convergence of the columns of the QD-scheme can be solved, In the case of positive roots we can prove that the columns will converge to the roots under all circumstances (and not only in the case of different roots). Again this is not true in the general case, where complex roots may spoil the convergence.

Rutishauser has also developed the so-called LR algorithm which may be considered as a more general method than the QD-algorithm. The LR algorithm may be used to determine the eigenvalues and eigenvectors of matrices., Since — to a given polynomial — there corresponds a matrix the eigenvalues of which are the roots of the polynomial, the roots may be found by means of the LR-algorithm, Furthermore, to most of the results concerning one of these algorithms there corresponds a similar result concerning the other.

THE QD-ALGORITHM
AS A METHOD FOR FINDING THE ROOTS OF A
POLYNOMIAL EQUATION WHEN ALL ROOTS ARE POSITIVE *)

BY

Chr. Andersen

Summary.

In Sections 1 and 2 the QD-scheme, symmetric functions and some results from the theory of Hankel determinants are treated. Most of the results have been known for a long time. Aitken [1] and Henrici [6] have used these for the same purpose of rootfinding as treated here. However, theorem 2.4 by means of which the existence of a positive constant c such that $H_n^k > c$ (positive roots) may be proved, seems to be new.

Section 3 contains some well known relations expressing the elements of the QD-scheme by means of the Hankel determinants, and the existence theorem mentioned above.

In Section 4 the question of convergence of the columns of the QD-scheme is treated. An exact expression for q_n^k is developed for the case of different roots. This expression seems to be new. It is proved that the columns of the &D-scheme will converge not only in the well known case of different roots, but in all cases where the roots are positive,

*) A part of the present paper was first presented at the Matrix Symposium in Gatlinburg, Tennessee, April 13-18, 1964.

Reproduction in Whole or in Part is Permitted for any Purpose of the United States Government. This report was supported in part by Office of Naval Research Contract Nonr-225(37) (NR 044-211) at Stanford University.

Section 5 contains a detailed examination of the convergence to the smallest root. In this section an exact expression for q_n^N is developed. This expression, is correct in all cases of multiple positive roots,

It turns out that the convergence of the columns of the Q,D-scheme to the roots of the polynomial equation may be slow, and it becomes necessary to speed up the convergence before the QD-algorithm can be of use in practice.

In [11] Rutishauser uses the principle of replacement as a device for accelerating the QD-algorithm. This principle has also been used by Faddeev and Faddeeva [4]. They remark, that the method may be useful as soon as the QD-scheme "has stabilized". It is however not easy to give general and useful criteria for such "stability? Furthermore, Rutishauser [16] remarks that the computation practice with the method of replacement has not always been successful,,

Numerical experiments in which I have tried to use the Aitken δ^2-process on the columns of the QD-scheme has not indicated that this process will be useful in connection with the QD-algorithm in all cases.

In the case of positive roots it is however possible to use the principle of' replacement in such a way that faster convergence will be obtained. Theorems concerning this question are included in Section 5.

Finally, in Section 6, it is shown that the progressive form of the &D-algorithm is only "mildly unstable".

In Part 2, that is Sections 7 and 8, some ALGOL programs and some results obtained by means of these, are given. The examples show that the QD-algorithm works nicely in practice in cases where the roots are positive, and the difficulties which arise in cases where several roots are equal or almost equal do not give too much trouble., A few words about the practical use of the &D-algorithm as a general rootfinder may be added. In numerical experiments with real polynomials with complex roots (polynomial with real roots may be transformed into polynomials with positive roots) the algorithm works perfect in many cases; but in cases where several roots were of the same, or almost the same, modulus (apart from conjugate roots) the ALGOL programs written by the present author failed to work properly. This fact does not mean however that the QD-algorithm should not be used in such cases. But it means that the QD-algorithm should be combined with other algorithms. Used in the beginning of a general root-finding program the QD-algorithm may give some very useful information concerning the roots and this information can be used in other algorithm for the final determination of the roots.

Part 1: The QD-algorithm.

1. The QD-scheme.

1.1 Formulation of the problem.

Let

$$p_N(x) = a_N x^N + a_{N-1} x^{N-1} + \dots + a_1 x + a_0 \qquad a_N \neq 0 \tag{1.1}$$

be a polynomial of degree N, let $a_o \neq 0$ and let the roots of $p_N(x) = 0$ be numerated such that

$$|z_1| \geq |z_2| \geq \dots \geq |z_N| \quad .$$

The coefficients $a_o, a_1, \dots a_N$ may be complex.

The problem we will treat is to find the roots of $p_N(x) = 0$ by means of the QD-algorithm, or better, to find approximations to the roots by means of this algorithm.

It turns out to be difficult to treat this problem in its full generality; at least it seems to be difficult to use the w-algorithm with success for all polynomials. In the present work the problem to be considered is then limited to the following:

Let $p_N(x)$ be a polynomial with real coefficients, and let it be known that all the roots of $p_N(x) = 0$ are real and positive. Find approximations to the roots by means of the QD-algorithm.

1.2 The progressive form of the QD-algorithm.

The QD-scheme.

We begin with the formal rules for constructing a QD-scheme, which consists of two sets of elements, called q_n^k and e_n^k, written as follows:

$$
\begin{array}{ccccccccccc}
 & q_1^1 & & q_1^2 & & q_1^3 & \cdot & & \cdot & q_1^N & \\
e_1^0 & & e_1^1 & & e_1^2 & & & & e_1^{N-1} & & e_1^N \\
 & q_2^1 & & q_2^2 & & q_2^3 & \cdot & & & q_2^N & \\
e_2^0 & & e_2^1 & & e_2^2 & & & & e_2^{N-1} & & e_2^N \\
 & \vdots & & \vdots & & & & & \vdots & \vdots & \vdots
\end{array}
$$

The upper index k in q_n^k runs from $1 \leq k \leq N$ and in e_n^k, k runs from $0 \leq k \leq N$. The lower index n runs from $1 < n < \infty$ in both cases. The index k is the column number and n is the row number.

The form and the notation used in this paper is the same as Henrici has used in [7]; it differs from the notation used by Rutishauser and by Henrici in [6].

In the progressive form of the QD-algorithm the elements in the first q-row and the first e-row must be given. Furthermore the first and the last e-column has zeros in all places.

From these quantities we construct the following rows in the QD-scheme by means of the recurrence relations:

$$
(1.2) \qquad q_{n+1}^{k} = e_n^k - e_n^{k-1} + q_n^k \qquad k = 1, 2, \ldots N; \quad n = 1, 2, \ldots
$$

(1.3) $$e^k_{n+1} = q^{k+1}_{n+1}/q^k_{n+1} \times e^k_n \qquad k = 1,2,\cdots,N-1;\ n = 1,2,\cdots$$

These formulas are used as follows:

First (1,2) for $k = 1,2,\cdots,N$ to obtain the "q-part" of a new row and then (1,3) for $k = 1,2,\cdots,N-1$ to obtain the remaining "e-part" of the same row.

We remark, that the construction cannot be carried out if $q^k_n = 0$ for some $k \leq N - 1$ and some $n > 0$. In this case the Q,D scheme is said not to exist.

The formulas (1.2) and (1.3) are known as the rhombus rules (Stiefel) since they connect four elements, the configuration of which is a rhombus, in the QD-scheme.

1.3 The forward form of the QD-algorithm.

The formula (1.3) may be written in the form

(1.4) $$q^{k+1}_{n+1} = e^k_{n+1}/e^k_n \times q^k_{n+1}$$

and by putting $k + 1$ instead of k in (1.2) this may be written as

(1.5) $$e^{k+1}_n = q^{k+1}_{n+1} - q^{k+1}_n + e^k_n$$

The formulas (1.4) and (1.5) show, that a new column (k+1) may be obtained from column k; that is the QD-scheme can be built up from a given e-column and a given q-column. In this case the QD-scheme is not limited to the right, and we can only find elements q^k_n and e^k_n for which $n > k$. This form of the QD-scheme is obtained by means of the forward form of the QD-algorithm.

As we will show in Section 6, the forward form of the QD-algorithm is not suited for numerical purposes since this form is unstable.

In the remaining part of the paper we shall only use the progressive form of the QD-algorithm.

1.4 The first row of the QD-scheme.

When the QD-algorithm is used as a method for finding the roots of $p_N(x) = 0$ the first row is constructed from the polynomial,

$$p_N(x) = a_N x^N + a_{N-1} x^{N-1} + \cdots + a_1 x + a_0 ,$$

as follows:

$$q_1^1 = -\frac{a_{N-1}}{a_N}$$

$$q_1^k = 0 \qquad 2 \le k \le N \tag{1.7}$$

$$e_1^0 = e_1^N = 0$$

$$e_1^k = \frac{a_{N-k-1}}{a_{N-k}} \qquad 1 \le k \le N-1$$

Until now we have assumed that $a_N \neq 0$ and $a_0 \neq 0$. From the last of the formulas (1.7) follows that all the other coefficients must be different from zero in order to start the QD-algorithm.

By means of a simple substitution $x = x_1 + c$ it is always possible to obtain an equation where all the coefficients are different from zero.

It is more serious if one of the q-elements computed by means of the formula (1.3) becomes zero and then spoil the algorithm. By means of an example it is easy to show that this may happen.

Example 1.1

$$p_3(x) = x^3 + ax^2 + bx + c$$

QD-scheme

e^0	q^1	e^1	q^2	e^2	q^3	e^3
	$-a$		0		0	
0		$\frac{b}{a}$		$\frac{c}{b}$		0
	$\frac{b}{a} - a$		$\frac{c}{b} - \frac{b}{a}$		$-\frac{c}{b}$	

Now $q_2^1 = 0$ if $\frac{b}{a} - a = 0$ and $q_2^2 = 0$ if $\frac{c}{b} - \frac{b}{a} = 0$. In these cases the QD-scheme will not exist.

It is however possible to show, that the QD-scheme always exists, if all roots of $p_m(x) = 0$ are real and positive. This will be proved in another section.

2. Symmetric functions. Hankel determinants.

In Section 2.1 we state some well known results about the symmetric functions in the roots of a polynomial equation. These results will be used to prove a theorem which is fundamental for the solution of the existence problem.

2.1 The elementary and the complete symmetric functions.

The elementary symmetric functions in the roots $z_1, \cdots z_N$ of the polynomial equation $p_N(x) = 0$ are defined as follows:

$$\sigma_0 = 1$$
$$\sigma_1 = z_1 + \cdots + z_N$$
$$\sigma_2 = z_1 z_2 + z_1 z_3 + \cdots + z_{N-1} z_N \qquad (2.1)$$
$$\vdots$$
$$\sigma_N = z_1 z_2 \cdots z_N$$
$$\sigma_p = 0 \quad \text{for} \quad p < 0 \quad \text{or} \quad p > N$$

The polynomial

$$p_N(x) = a_N x^N + a_{N-1} x^{N-1} + \cdots + a_1 x + a_0 \qquad (a_N \neq 0)$$

may be expressed by means of the elementary symmetric functions as

$$p_N(x) = a_N(\sigma_o x^{N-1} + \sigma_2 x^{N-2} + \cdots + (-1)^N \sigma_N)$$

that is we have the relation

(2.2) $$\sigma_k = (-1)^k \frac{a_{N-k}}{a_N}$$

The complete symmetric functions in $z_1,\ldots,z_N$ are defined as follows

(2.3)
$$\begin{aligned}
S_o &= 1 \\
S_1 &= z_1 + \cdots + z_N \\
S_2 &= z_1^2 + z_1 z_2 + z_1 z_3 + \cdots + z_{N-1} z_N + z_N^2 \\
S_3 &= z_1^3 + z_1^2 z_2 + \cdots + z_N^3 \\
&\vdots \\
S_p &= 0 \quad \text{for } p < 0
\end{aligned}$$

The complete symmetric function S_n of degree n consists of the sum of all different terms of the form

(2.4) $$z_1^{\alpha_1} \cdots z_N^{\alpha_N}$$

where $0 \leq \alpha_i \leq N \qquad 1 \leq i \leq N \qquad$ and $\displaystyle\sum_{i=1}^{N} \alpha_i = n$

Theorem 2.1

Let S_n denote the complete symmetric function of degree n in the N variable $z_1,\cdots z_N$, and let $S_n^{(r)}$ denote the complete symmetric function in the (N-1) variable $z_1,\cdots z_{r-1}, z_{r+1},\cdots z_N$. Then

$$S_n = z_r S_{n-1} + S_n^{(r)} \qquad (r-1,\ldots,N; \ \text{all } n) \tag{2.5}$$

Proof

The terms of S_n may be divided into two sets, the first of which consists of all terms with z_r as a factor and the second set of all other terms. Hence (2.5) is true.

By means of a similar argument we may prove the corresponding relation between the elementary symmetric functions:

$$\sigma_n = z_r \sigma_{n-1}^{(r)} + \sigma_n^{(r)}, \qquad (r=1,\ldots,N; \ \text{all } n) \tag{2.6}$$

where $\sigma_{n-1}^{(r)}$ and $\sigma_n^{(1)}$ denote the elementary symmetric functions of degree $(n-1)$ and n, respectively in the $(N-1)$ variable $z_1,\ldots,z_{r-1}, z_{r+1},\ldots,z_N$.

Theorem 2.2

For all positive values of n the complete and the elementary symmetric functions in N variables are connected by the relation

$$S_n = \sigma_1 S_{n-1} - \sigma_2 S_{n-2} + \cdots + (-1)^{n-1} \sigma_n S_0 \tag{2.7}$$

Proof

By induction with respect to N.

N = 2, In this case $\sigma_1 = z_1 + z_2$, $\sigma_2 = z_1 z_2$ and $\sigma_p = 0$ for $p \geq 3$. Hence (2.7) has the form

$$S_n = (z_1+z_2) S_{n-1} - z_1 z_2 S_{n-2} ,$$

which, with $S_n = z_1^n + z_1^{n+1} z_2 + \cdots z_1 z_2^{n+1} + z_2^n$ and the corresponding expressions for S_{n-1} and S_{n-2}, is true.

We assume (2.7) is true for 2, 3,...N-1 variables, respectively and for all values of n in these cases, and consider the case of N variables $z_1, z_2, \ldots, z_N$. We prove that (2.7) holds in the case by induction with respect to n. n=1; that is $S_1 = \sigma_1$ which is true.

Let (2.7) be true for 1, 2,...,n and consider the case n+1. We have to prove

$$S_{n+1} = \sigma_1 S_n - \sigma_2 S_{n-1} + \sigma_3 S_{n-2} - \cdots + (-1)^n \sigma_{n+1} \tag{2.8}$$

By means of (2.5) we have - with the notation S'_p instead of $S_p^{(N)}$ - that

$$\sigma_1 S_n - \sigma_2 S_{n-1} + \sigma_3 S_{n-2} - \cdots + (-1)^n \sigma_{n+1}$$

$$= \sigma_1 (z_N S_{n-1} + S'_n) - \sigma_2 (z_N S_{n-2} + S'_{n-1}) + \sigma_3 (z_N S_{n-3} + S'_{n-2}) - \cdots (-1)^n \sigma_{n+1}$$

$$= z_N (\sigma_1 S_{n-1} - \sigma_2 S_{n-2} + \sigma_3 S_{n-3} - \cdots (-1)^{n-1} \sigma_n)$$

$$+ \sigma_1 S'_n - \sigma_2 S'_{n-1} + \sigma_3 S'_{n-2} - \cdots + (-1)^n \sigma_{n+1}$$

$$= z_N S_n$$

$$+ (z_N \sigma'_o + \sigma'_1) S'_n - (z_N \sigma'_1 + \sigma'_2) S'_{n-1} + (z_N \sigma'_2 + \sigma'_3) S'_{n-2} - \cdots + (-1)^n (z_N \sigma'_n + \sigma'_{n+1})$$

$$= z_N S_n + z_N (\sigma'_o S'_n - \sigma'_1 S'_{n-1} - \cdots + (-1)^n \sigma'_n)$$

$$+ (\sigma'_1 S'_n - \sigma'_2 S'_{n-1} + \sigma'_3 S'_{n-2} - \cdots + (-1)^n \sigma'_{n+1})$$

$$= z_N S_n + S'_{n+1} .$$

In the calculations we have used (2.7) three times, and we have used (2.6) too. The last expression however is equal to S_{n+1} and we have proved theorem 2.2 by induction.

2.2 Hankel determinants,

The Hankel determinants will be used as the basic tool in the following treatment of the QD-algorithm. The relation (2.10) which is of special importance is used by Aitken [1] and by Henrici [6] for solving the same problem as we treat, and the sketch of the proof follows the same lines as used in [6] and in Householder [8].

Definition of Hankel determinants.

Let $\cdots a_{-2}, a_{-1}, a_o, a_1, a_2 \cdots$ be any sequence of complex numbers, then we define the Hankel determinants H_n^k, for $n > 0$, as follows:

$$H_o^k = 1 \ ; \quad H_n^k = \begin{vmatrix} a_k & a_{k-1} & \cdots & a_{k-n+1} \\ a_{k+1} & a_k & & \\ & & \ddots & \\ a_{k+n-1} & & & a_k \end{vmatrix} \qquad n = 1, 2, 3, 0.0$$

We may prove the following relation:

$$(2.10) \qquad H_n^{k-1} \cdot H_n^{k+1} - (H_n^k)^2 + H_{n-1}^k H_{n+1}^k = 0 \qquad n > 1 \ ;$$

Consider the determinant of order 2n + 2:

	1	2		n	n+1	n+2			2n	2n+1	2n+2
1	a_k	a_{k-1}	$\cdots$	a_{k-n+1}	0	0	0	0	a_{k-n}	0	1
2	a_{k+1}	a_k	$\cdots$	a_{k-n}	0	0	0	0	a_{k-n+1}	0	0
	$\vdots$										$\vdots$
n	a_{k+n-1}	a_{k+n}		a_k	0	0	0	0		0	0
n+1	a_{k+n}	a_{k+n-1}		a_{k-1}	0	0	0	0	a_k	1	0
n+2	a_k	0	$\cdots$	0	a_{k-1}	a_{k-2}	a_{k-3}	$\cdots$	a_{k-n}	0	1
n+3	a_{k+1}	0		0	a_k	a_{k-1}	a_{k-2}	$\cdots$	a_{k-n+1}	0	0
					$\vdots$						$\vdots$
2n+1	a_{k+n-1}	0		0	a_{k+n-2}	a_{k+n-3}	a_{k+n-4}	$\cdots$	a_{k-1}	0	0
2n+2	a_{k+n}	0		0	a_{k+n-1}	a_{k+n-2}	a_{k+n-3}	$\cdots$	a_k	1	0

If we subtract row $(n+1+i)$ from row i for $i = 1,\ldots,n+1$ and then add column $(2+i)$ to column $(n+i)$ for $i = 0,1,\ldots,(n-2)$ we find that this determinant must be equal to zero. On the other hand if we compute the determinant by expanding by $(n+1)$-order minors we obtain two times the left side of (2.10). For further details see Householder [8].

2.3 Hankel determinants in the symmetric functions.

Hankel determinants in the elementary symmetric functions and in the complete symmetric functions are related. We prove

Theorem 2.3

Let

$$H_n^k = \begin{vmatrix} \sigma_k & \sigma_{k-1} \cdots \sigma_{k-n+1} \\ \sigma_{k+1} & \sigma_k \\ \vdots & & \ddots \\ \sigma_{k+n-1} & & \sigma_k \end{vmatrix} \quad \text{(n order)}$$

and

$$C_k^n = \begin{vmatrix} S_n & S_{n-1} \cdots S_{n-k+1} \\ S_{n+1} & S_n \\ \vdots \\ S_{n+k-1} & 000000 \; S_n \end{vmatrix} \quad \text{(k order)}$$

and let $1 \leq k \leq N$

If $H_n^k \neq 0$ for all non-negative n, then

(2011) $$H_n^k = C_k^n \qquad n = 0,1,2,\cdots$$

Proof

By induction with respect to k.

k = 1: We have to prove that $H_n^1 = S_n$.

This may be proved by induction with respect to n.

n = 0: $H_o^1 = S_o$ is correct since both sides are equal to 1.

n = 1: $H_1^1 = \sigma_1 = S_1$.

Now we may assume that $H_n^1 = S_n$ for $n = 0,1,2,\ldots,p - 1$ and we consider the case $n = p$

$$H_p^1 = \begin{vmatrix} \sigma_1 & 1 & 0 & \cdots & \\ \sigma_2 & \sigma_1 & 1 & & \\ \vdots & & \ddots & \ddots & \\ \vdots & & & \ddots & 1 \\ \sigma_n & & & & \sigma_1 \end{vmatrix} = \sigma_1 H_{p-1}^1 - \begin{vmatrix} \sigma_2 & 1 & & \\ \sigma_3 & \sigma_1 & 1 & \\ & \sigma_2 & \ddots & 1 \\ \sigma_n & & & \sigma_1 \end{vmatrix} = \cdots \quad \text{(n-1 order)}$$

$$= \sigma_1 H_{p-1}^1 - \sigma_2 H_{p-2}^1 + \cdots + (-1)^{p-1} \sigma_p$$

$$= \sigma_1 S_{p-1} - \sigma_2 S_{p-2} + \cdots + (-1)^{p-1} \sigma_p$$

$$= S_p \,.$$

The last result follows from theorem 2.2. Hence we have proved theorem 2.3 in the case $k = 1$.

Now we assume that (2.11) is true for $k = 1,2,\ldots,p$ and for all non-negative n in each case. By means of the relation (2.10) we find for $n > 0$;

$$H_n^{p+1} = [(H_n^p)^2 - H_{n+1}^p H_{n-1}^p]/H_n^{p-1}$$

$$= [(C_p^n)^2 - C_p^{n+1} C_p^{n-1}]/C_{p-1}^n$$

$$= C_{p+1}^n \,.$$

We remark that in case p = 1 we have used $H_n^{p-1} = H_n^o = 1 = C_o^n$. For n = 0 we have $H_0^{p+1} = 1 = C_{p+1}^o$, and we have proved theorem (2.3) by induction.

In the following the notation H_n^k will only be used for Hankel determinants in the elementary symmetric functions.

2.4 A fundamental theorem.

Until now $z_1, \cdots z_N$ have been arbitrary complex numbers, and this being the case the Hankel determinants may vanish, This cannot happen if $z_1 \cdots z_N$ are real and positive numbers.

Theorem 2.4

Let $z_1, z_2, \cdots z_N$ be positive. Define

$$D_n^{(N)} = \begin{vmatrix} \sigma_{\alpha 11} & \sigma_{\alpha 12} & \cdots & \sigma_{\alpha 1n} \\ \sigma_{\alpha 21} & \sigma_{\alpha 22} & \cdots & \sigma_{\alpha 2n} \\ \vdots & & & \\ \sigma_{\alpha n1} & \sigma_{\alpha n2} & & \sigma_{\alpha nn} \end{vmatrix}$$

where σ_{aij} are elementary symmetric functions.

Let

(i) $\quad a_{i1} > \alpha_{i2} > \cdots > \alpha_{in} \qquad 1 \leq i \leq n$

(ii) $\quad \alpha_{1j} < \alpha_{2j} < \cdots < \alpha_{nj} \qquad 1 \leq j \leq n$

Then

$$D_n^{(N)} > 0 \qquad \text{for all} \qquad n > 1$$

and, if

$$\alpha ii = k \qquad i = 1,\ldots,n$$

where $0 \leq k \leq N$, then $D_n^{(N)} > \min (1,(\sigma_N)^n)$.

Proof

By induction with respect to the number of variables N.

N = 1: Then $\sigma_0 = 1$, $\sigma_1 = z_1$ and $\sigma_p = 0$ for $p \neq 0, 1$.

We use induction with respect to n.

n = 1: $D_1^{(1)} = \sigma_{\alpha 11}$. The theorem is obviously true.

Assume, that the theorem is true for n = 1,2,...,p-1 and consider $D_p^{(1)}$.

If $\alpha pp \neq 0,1$ it follows from the conditions (i) and (ii) that the p-th row or the p-th column consists of zeros; that is $D_p^{(1)} = 0$.

If $\alpha pp = 0$; that is $\sigma_{\alpha pp} = 1$, we have (by means of (ii))

$$D_p^{(1)} = 1 \cdot D_{p-1}^{(1)}$$

If $\alpha pp = 1$; that is $\sigma_{\alpha pp} = z_1$, we have (by means of (i))

$$D_p^{(1)} = z_1 D_{p-1}^{(1)}$$

In all cases the theorem is true for $n = p$ and we have proved theorem (2.4) in the case $N = 1$.

Let the theorem be true for $(N-1)$ variable $z_1, \cdots z_{N-1}$ and let σ'_p denote the elementary symmetric function of degree p in these (N-1) variables, Let $z_1 \geq z_2 > \cdots \geq z_N$. We use a relation between elementary symmetric functions:

$$\sigma_p = z_N \sigma'_{p-1} + \sigma'_p \qquad p = 0, \pm 1, \pm 2, \cdots \tag{2.12}$$

To prove (2.12) we remark that the terms of- σ_p may be divided into two sets, the first of which contains all terms with z_N as a factor and the other set of the remaining terms.

By means of (2.12) we may write $D_n^{(N)}$ as follows

$$D_n^{(N)} = \begin{vmatrix} z_N \sigma'_{\alpha 1 1-1} + \sigma'_{\alpha 1 1} & z_N \sigma'_{\alpha 1 2-1} + \sigma'_{\alpha 1 2} & \cdots & z_N \sigma'_{\alpha 1 n-1} + \sigma'_{\alpha 1 n} \\ z_N \sigma'_{\alpha 2 1-1} + \sigma'_{\alpha 2 1} & & & \\ \cdot & & & \\ z_N \sigma'_{\alpha n 1-1} + \sigma'_{\alpha n 1} & \cdot & \cdot & z_N \sigma'_{nn-1} + \sigma'_{nn} \end{vmatrix} \tag{2.13}$$

From (2.13) follows that $D_n^{(N)}$ may be written as a sum of 2^n determinants, The conditions (i) and (ii) show, that each of these determinants may either have proportional columns — and then have the value zero — or the indices will again satisfy (i) and (ii), The non-zero determinants, from which z_N may be removed, are then non-negative and as a sum of these $D_n^{(N)}$ must be non-negative itself,

Now let $aii = k$, $0 \leq k \leq N$.

If $k < N$ we consider the term with z_N^o, say $D_n^{(N-1)}$. By the induction assumption $D_n^{(N-1)} \geq \min(1, (z_1 \ldots z_{N-1})^n)$. Since $\min(1, (z_1 \cdots z_N)^n) \leq \min(1, (z_1 \cdots z_{N\,1})^n)$ we have

$$D_n^{(N)} \geq D_n^{(N-1)} \geq \min(1, (\sigma_N)^n)$$

If $k = N$ we consider the term with z_N^n, that is $z_N^n \cdot \Delta_n$ where Δ_n has $(z_1 \ldots z_{N-1})$ in the diagonal, and zeros below the diagonal. Hence $z_N^n \cdot \Delta_n = (\sigma_N)^n$, and again

$$D_n^{(N)} \geq \min(1, (a,)") ,$$

and we have proved theorem (2.4) by induction.

Theorem 2.5

Let $z_1 \geq z_2 > \cdots \geq z_N > 0$.

Then

$$H_n^k \geq \min(1, (\sigma_N)^n) \qquad 1 \leq k \leq N \qquad n > 0$$

Proof

Since the Hankel determinants satisfy the conditions (i) and (ii) from theorem 2.4, and since the diagonal elements have the same index this result is. nothing but a corollary to theorem (2.4).

3. The existence theorem in the case of positive roots.

3.1 Formulas for q_n^k and e_n^k

Let the QD scheme for the polynomial $p_N(x)$ be started as in section (1.4) and continued by means of the rhombus formulas (1.2) and (1.3). Then the elements q_n^k and e_n^k may be expressed by means of the Hankel determinants H_n^k in the simple symmetric expressions.

Theorem 3.1

If the Hankel determinants H_n^k are different from zero, then

$$q_n^k = \frac{H_n^k \; H_{n-2}^{k-1}}{H_{n-1}^k \; H_{n-1}^{k-1}} \qquad \begin{array}{l} n = 2, 3, \ldots \\ k = 1, 2, \ldots N \end{array} \tag{3.1}$$

and

$$e_n^k = - \frac{H_n^{k+1} \; H_{n-1}^{k-1}}{H_n^k \; H_{n-1}^k} \qquad \begin{array}{l} n = 2, 3, \ldots \\ k = 1, 2, \ldots . N-1 \end{array} \tag{3.2}$$

Proof

By induction with respect to n

n = 2

We have to prove that $q_2^k = \dfrac{H_2^k \; H_o^{k-1}}{H_1^k \; H_1^{k-1}}$

Now

$$q_2^k = e_1^k - e_1^{k-1} + q_1^k$$

$$= \frac{a_{N-k-1}}{a_{n-k}} - \frac{a_{N-k}}{a_{N-k+1}}$$

where we have used (1.7). By means of (2.2) we find

$$q_2^k = - \frac{\sigma_{k+1}}{\sigma_k} + \frac{\sigma_k}{\sigma_{k-1}}$$

On the other hand

$$\frac{H_2^k \; H_o^{k-1}}{H_1^k \; H_1^{k-1}} = \frac{\begin{vmatrix} \sigma_k & \sigma_{k-1} \\ \sigma_{k+1} & \sigma_k \end{vmatrix}}{\sigma_k \quad \sigma_{k-1}} = \frac{\sigma_k}{\sigma_{k-1}} - \frac{\sigma_{k+1}}{\sigma_k} \; ,$$

and we have proved (3.1) for n = 2.

Since

$$e_2^k = q_2^{k+1}/q_e^k \times e_1^k$$

$$= \left(\frac{H_2^{k+1} \quad H_o^k}{H_1^{k+1} \quad H_1^k}\right) \Bigg/ \left(\frac{H_2^k \; H_o^{k-1}}{H_1^k \; H_1^{k-1}}\right) \times \frac{a_{N-k-1}}{a_{N-k}}$$

$$= - \left(\frac{H_2^{k+1} \quad H_o^k}{H_1^{k+1} \quad H_1^k}\right) \Bigg/ \left(\frac{H_2^k \; H_o^{k-1}}{H_1^k \; H_1^{k-1}}\right) \times \frac{\sigma_{k+1}}{\sigma_k}$$

$$= - \left(\frac{H_2^{k+1} \quad H_o^k}{H_1^{k+1} \quad H_1^k}\right) \Bigg/ \left(\frac{H_2^k \; H_o^{k-1}}{H_1^k \; H_1^{k-1}}\right) \times \frac{H_1^{k+1}}{H_1^k}$$

$$= - \frac{H_2^{k+1} \quad H_1^{k-1}}{H_2^k \quad H_1^k} \; ,$$

formula (3.2) is also correct for n = 2.

Now assume that (3.1) and (3.2) holds for 2, 3, ... n, and all k in question and consider the case n + 1. We obtain:

$$q_{n+1}^{k} = e_n^k - e_n^{k-1} + q_n^k$$

$$= -\frac{H_n^{k+1} H_{n-1}^{k-1}}{H_n^k H_{n-1}^k} + \frac{H_n^k H_{n-1}^{k-2}}{H_n^{k-1} H_{n-1}^{k-1}} + \frac{H_n^k H_{n-1}^{k-1}}{H_{n-1}^k H_{n-1}^{k-1}}$$

$$= \frac{H_n^k}{H_{n-1}^{k-1}} \cdot \frac{H_{n-1}^k H_{n-1}^{k-2} + H_n^{k-1} H_{n-2}^{k-1}}{H_n^{k-1} H_{n-1}^k} - \frac{H_n^{k+1} H_{n-1}^{k-1}}{H_n^k H_{n-1}^k}$$

$$= \frac{H_n^k}{H_{n-1}^{k-1}} \cdot \frac{(H_{n-1}^{k-1})^2}{H_n^{k-1} H_{n-1}^k} - \frac{H_n^{k+1} H_{n-1}^{k-1}}{H_n^k H_{n-1}^k}$$

$$= \frac{H_{n-1}^{k-1}}{H_{n-1}^k} \; \frac{(H_n^k)^2 - H_n^{k-1} H_n^{k+1}}{H_n^{k-1} H_n^k}$$

$$= \frac{H_{n-1}^{k-1}}{H_{n-1}^k} \; \frac{H_{n-1}^k H_{n+1}^k}{H_n^{k-1} H_n^k} = \frac{H_{n+1}^k}{H_n^k} \cdot \frac{H_{n-1}^{k-1}}{H_n^{k-1}} \quad ;$$

that is (3.1) holds for n + 1. We remark, that we have used (2.10) twice.

Now

$$e_{n+1}^k = q_{n+1}^{k+1} / q_{n+1}^k \times e_n^k$$

$$= \frac{H_{n+1}^{k+1} H_{n-1}^k}{H_n^{k+1} H_n^k} \cdot \frac{H_n^k H_n^{k-1}}{H_{n+1}^k H_{n-1}^{k-1}} \cdot -\left(\frac{H_n^{k+1} H_{n-1}^{k-1}}{H_n^k H_{n-1}^k}\right)$$

$$= -\frac{H_{n+1}^{k+1} H_n^{k-1}}{H_{n+1}^k H_n^k} \quad ;$$

and (3.2) has been proved for $n + 1$.

Theorem 3.2. The existence theorem.

Let the roots of $p_N(x) = o$ satisfy the conditions $z_1 \geq z_2 \geq \dots \geq z_N > o$.

Then

$$q_n^k > c > o \qquad k = 1, 2, \dots N \qquad \text{all } n > 2$$

where c is a constant.

Hence the QD-scheme always exists in the case of positive roots.

Proof

From theorem 3.1 we have

$$q_n^k = \frac{H_n^k \; H_{n-2}^{k-1}}{H_{n-1}^k H_{n-1}^{k-1}}$$

and from theorem 2.5 we know, that

$$H_n^k \geq \min\left(1, (\sigma_N)^n\right)$$

Hence we may conclude that $q_n^k > o$.

In order to prove that $q_n^k > c > o$ we use the following

Lemma 3.1

$$\sum_{k=1}^{N} q_n^k = \sigma_1 \qquad \text{for all } n > 1$$

Proof

For n = 1 this follows from the first row in the QD-scheme, where $q_1^1 = \sigma_1$ and $q_1^k = o$ for $2 < k < N$.

Let it be true for 1, 2, ... n, and consider

$$\sum_{k=1}^{N} q_{n+1}^{k} = \sum_{k=1}^{N} (e_n^k - e_n^{k-1} + q_n^k)$$

$$= e_n^N - e_o^N + \sum_{k=1}^{N} q_n^k$$

$$= o + \sum_{k=1}^{N} q_n^k$$

It follows that the lemma is true for n + 1.

Lemma 3.2

$$\prod_{k=1}^{N} q_{n+k-N}^{k} = \sigma_N \quad \text{for all } n > N$$

Proof

-

$$\prod_{k=1}^{N} q_{n+k-N}^{k} \cdot q_{n+1-N}^{1} \cdot q_{n+2-N}^{2} \cdots q_n^N$$

$$= \frac{H_{n+1-N}^{1}}{H_{n-N}^{1}}, \frac{H_{n+2-N}^{2} \; H_{n-N}^{1}}{H_{n+1-N}^{2} \; H_{n+1-N}^{1}} \cdots \frac{H_n^N \; H_{n-2}^{n-1}}{H_{n-1}^{N} \; H_{n-1}^{N-1}}$$

$$= \frac{H_n^N}{H_{n-1}^N}$$

$$= \frac{\sigma_N^n}{\sigma_N^{n-1}}$$

$$= \sigma_N$$

Lemma 3.3

$$q_n^k < \sigma_1 \qquad 1 \le k \le N \qquad n > 2$$

Proof

Since $q_n^k > o$ and $\sum_{k=1}^{N} q_n^k = \sigma_1$ the lemma is obviously true.

Lemma 3.4

$$q_n^k > \sigma_N \sigma_1^{1-N} \quad 1 \leq k \leq N \quad n > N$$

Proof

Since $q_n^k < \sigma_1$ and since $\prod_{k=1}^{N} q_{n+k-N}^k = \sigma_N$ the lemma is obviously true.

From lemma 3.4 follows that $q_n^k > c$, where $c = \sigma_N \sigma_1^{1-N}$ for $n > N$. We consider q_n^k for $2 < n < N$.

Since $H_n^k \geq \min(1, (\sigma_N)^n)$, and since $n < N$ we have

$$H_n^k \geq \min(1, (\sigma_N)^{N-1})$$

for the n's in question.

Then

$$q_n^k \geq c_1 = [\min(1, (\sigma_N)^{N-1})]^2/M$$

where

$$M = \max_{2<n<N} (H_{n-1}^k \; H_{n-1}^{k-1})$$

Hence

$$q_n^k \geq \min[\sigma_N \sigma_1^{1-N}, c_1] > o \quad 1 < k < N \quad n > 2$$

and we have proved theorem 3.2.

We remark, that polynomial equations, the roots of which are known to be real but not necessary positive, may be solved by means of the Q,D algorithm as soon as a lower bound for the roots has been found. This being the case a transformation may be carried out and the theory for positive roots can be used.

4. General convergence properties.

In this section we examine the **columns** of the QD scheme for a polynomial equation $p_N(x) = 0$. As usual we assume that $z_1 \geq z_2 \geq \cdots \geq z_N > o$. This being the case we may prove that the q-columns converge. Precisely, that $q_n^k \rightarrow z_k$ as $n \rightarrow \infty$ for $1 \leq k \leq N$. In order to prove this result we must develop some formulas for the Hankel determinants as functions of the roots $z_1, z_2, \ldots z_N$. The formulas used until now seems not to be useful since the number of terms in H_n^k tends to infinity with n.

4.1 H_n^k as a function of the roots.

The basic formula is

$$(4.1) \qquad H_n^k = \begin{vmatrix} S_n & S_{n+1} & \cdots & S_{n+k-1} \\ S_{n+1} & & & \\ \vdots & & & \\ S_{n+k-1} & \cdots & & S_n \end{vmatrix} \qquad 1 < k < N, \quad n = 0, 1, 2, \ldots$$

and we begin by finding S_n as a function of the roots.

Theorem 4.1

Let the roots be different, that is in our case $z_1 > z_2 > \cdots z_N > o$, then

(4.2)
$$S_n = \sum_{i=1}^{N} \frac{z_i^{N-1}}{\prod_{j=1, j\neq i}^{N} (z_i - z_j)} Z_i^n \qquad N > 2,\ n = 1, 2, \ldots$$

Proof

By induction with respect to the number of variables N.

N = 2

By definition

$$\begin{aligned} S_n &= z_1^n + z_1^{n-1} z_2 + \cdots + z_2^n \\ &= \frac{z_1^{n+1} - z_2^{n+1}}{z_1 - z_2} \qquad (z_1 > z_2) \\ &= \frac{z_1^{2-1}}{z_1 - z_2} z_1^n + \frac{z_2^{2-1}}{z_2 - z_1} z_2^n \qquad (N = 2) \end{aligned}$$

which is the right side of (4.2) in this case.

Let the theorem be true for 2, 3, ..., N-1 variables and for all n in each case. We consider S_n of N variables.

From theorem 2.1 we have

(4.3)
$$S_n = z_1 S_{n-1} + S_n^{(1)} = z_2 S_{n-1} + S_n^{(2)},$$

where $S_n^{(1)} = S_n^{(1)} [z_2, \ldots z_N]$; $S_n^{(2)} = S_n^{(2)} [z_1, z_3, \ldots z_N]$.

The formulas (4.3) give

$$S_{n-1} = (S_n^{(2)} - S_n^{(1)})/(z_1 - z_2)$$

or with n + 1 instead of n:

$$S_n = (S^{(2)}_{n+1} - S^{(1)}_{n+1})/(z_1 - z_2). \tag{4.4}$$

Now we may use (4.2) with N - 1 to obtain

$$S_n = \frac{1}{z_1 - z_2} \cdot \Bigg[\frac{z_1^{N-2}}{(z_1 - z_3)\cdots} z_1^{n+1} + \frac{z_3^{N-2}}{(z_3-z_1)(z_3-z_3)\ldots(z_3-z_N)} z_3^{n+1} + \cdots$$

$$+ \frac{z_N^{N-2}}{(z_N-z_1)(z_N-z_3)\cdots(z_N-z_{N-1})} z_N^{n+1} - \frac{z_2^{N-2}}{(z_2-z_3)\cdots(z_2-z_N)} z_2^{n+1} -$$

$$\frac{z_3^{N-2}}{(z_3-z_2)(z_3-z_4)\cdots(z_3-z_N)} z_3^{n+1} \cdots - \frac{z_4^{N-2}}{(z_N-z_2)(z_N-z_3)\cdots(z_N-z_{N-1})} z_N^{n+1} \Bigg]$$

$$= \frac{z_1^{N-1}}{(z_1-z_2)(z_1-z_3)\ldots(z_1-z_N)} z_1^n + \frac{z_2^{N-1}}{(z_2-z_1)(z_2-z_3)\ldots(z_2-z_N)} z_2^n$$

$$+ \sum_{i=3}^{N} \frac{z_i^{N-1}\,[z_i - z_2 - (z_i - z_1)]}{(z_1-z_2)(z_i-z_1)(z_i-z_2)\cdots(z_i-z_{i-1})(z_i-z_{i+1})\cdots(z_1-z_N)} z_i^n$$

$$= \sum_{i=1}^{N} \frac{z_i^{N-1}}{\prod_{i=1, i \neq j}^{N} (z_i - z_j)} z_i^n ,$$

and we have proved theorem 4.1 by induction.

<u>Theorem 4.2</u>

Let $z_1 > z_2 > \cdots z_N > 0$.

Then

$$(4.5)\qquad H_n^k = \sum_{\binom{N}{k}} \frac{(z_{\ell_1} \cdots z_{\ell_k})^{N-k}}{\prod_{i=1}^{k} \prod_{j=k+1}^{N} (z_{\ell_i} - z_{\ell_j})} (z_{\ell_1} \cdots z_{\ell_k})^n \qquad \begin{matrix} 1 \le k \le N \\ n > k \end{matrix}$$

where the sum is taken over all $\binom{N}{k}$ combinations $z_{\ell_1} \cdots z_{\ell_k}$ of k roots taken out of the N roots.

<u>Proof</u>

From the general formula (4.2) for S_n and the formula (4.1) follows that we may write H_n^k in the form

$$(4.6)\qquad H_n^k = \begin{vmatrix} \sum_1^N c_i z_i^n & \sum_1^N c_i z_i^{n-1} & \cdots & \sum_1^N c_i z_i^{n-k+1} \\ \sum_1^N c_i z_i^{n+1} & \sum_1^N c_i z_i^{n} & \cdots & \sum_1^N c_i z_i^{n-k+2} \\ \vdots & & & \\ \sum_1^N c_i z_i^{n+k-1} & \sum_1^N c_i z_i^{n+k-2} & \cdots & \sum_1^N c_i z_i^{n} \end{vmatrix} \qquad \text{(k rows)}$$

where the constants c_i; i = 1, ..., N are independent of k and n. At this point we have used $n \geq k$.

It follows that H_n^k may be written as a sum of N^k determinants

$$
(4.7) \qquad H_n^k = \sum \begin{vmatrix} c_{\ell_1} z_{\ell_1}^{n} & c_{\ell_2} z_{\ell_2}^{n-1} & . & . & . & c_{\ell_n} z_{\ell_k}^{n-k+1} \\ c_{\ell_1} z_{\ell_1}^{n+1} & c_{\ell_2} z_{\ell_2}^{n} & . & . & . & c_{\ell_k} z_{\ell_k}^{n-k+2} \\ . & & & & & \\ . & & & & & \\ . & & & & & \\ c_{\ell_1} z_{\ell_1}^{n+k-1} & c_{\ell_2} z_{\ell_2}^{n+k-2} & . & . & . & c_{\ell_k} z_{\ell_k}^{n} \end{vmatrix}
$$

where $1 < \ell_i < N \qquad i = 1, \ldots k$.

From (4.6) we know that the determinants, in order to be non-zero must have different roots in all columns; that is H_n^k may be written as a sum of $p(N,k)$ determinants. In (4.7) we then have to take the sum over all $p(N,k)$ permutations $(\ell_1, \ell_2, \ldots \ell_k)$ taken out of $(1, 2, \ldots N)$

Now the $p(N, k)$ determinants may be divided into $\binom{N}{k}$ sets, where the members of each set have the same k roots in their columns. Hence we may write

$$
(4.8) \qquad H_n^k = \sum_{I} \prod_{i=1}^{k} c_{\ell_i} \sum_{II} \begin{vmatrix} z_{q_1}^{n} & z_{q_2}^{n-1} & . & . & . & z_{q_k}^{n-k+1} \\ . & . & . & & & \\ . & & & & & \\ . & & & & & \\ z_{q_1}^{n+k-1} & z_{q_2}^{n+k-2} & . & . & . & z_{q_n}^{n} \end{vmatrix}
$$

where the sum $\sum_{II}$ must be taken over all $k!$ permutations $(q_1, \ldots q_k)$ of $(\ell_1, \ell_2, \ldots \ell_k)$ and the sum $\sum_{I}$ must be taken over all $\binom{N}{k}$ combinations (a,, $\ell_2, \ldots \ell_k$) of $(1, 2, \ldots N)$. Since the constants $c_{\ell_1}, \ldots c_{\ell_k}$ are the same for all members of the same set, these may be taken out as shown in (4.8). It follows that we may write (4.8) in the form

$$H_n^k = \sum_I \prod_{i=1}^{k} c_{\ell_i} \prod_{i=1}^{k} (z_{\ell_i})^{n-k+1} \sum_{II} \begin{vmatrix} z_{q_1}^{k-1} & z_{q_2}^{k-2} & \cdots & z_{q_k}^{0} \\ z_{q_1}^{k} & z_{q_2}^{k-1} & \cdots & z_{q_k}^{1} \\ \vdots & & \ddots & \\ z_{q_1}^{2k-2} & z_{q_2}^{2k-3} & \cdots & z_{q_k}^{k-1} \end{vmatrix}$$

We introduce the powers S^p of the roots $z_{\ell_1}, z_{\ell_2}, \ldots z_{\ell_k}$ by

$$S^p = z_{\ell_1}^p + \cdots z_{\ell_k}^p \qquad p = 0, 1, 2, \ldots$$

Then

$$\Delta = \begin{vmatrix} S^{k-1} & S^{k-2} & \cdot & \cdot & S^{o} \\ S^{k} & S^{k-1} & \cdot & \cdot & S^{1} \\ \cdot & & & & \\ \cdot & & & & \\ \cdot & & & & \\ S^{2k-2} & S^{2k-3} & \cdot & \cdot & S^{k-1} \end{vmatrix} = \begin{vmatrix} z_{\ell_1}^{k-1}+\cdots+z_{\ell_k}^{k-1} & \cdots & z_{\ell_1}^{o}+\cdots+z_{\ell_k}^{o} \\ z_{\ell_1}^{k}+\cdots+z_{\ell_k}^{k} & & \\ \cdot & & \\ \cdot & & \\ \cdot & & \\ z_{\ell_1}^{2k-2}+\cdots+z_{\ell_k}^{2k-2} & \cdots & z_{\ell_1}^{k-1}+\cdots+z_{\ell_k}^{k-1} \end{vmatrix}$$

It follows that Δ may be written as a sum of k^2 determinants. Of these only $k!$ are different from zero and the sum of these is $\sum_{II}$.

Now

$$\sum_{II} = \begin{vmatrix} S^{k-1} & S^{k-2} & \cdots & S^{o} \\ S^{k} & S^{k-1} & \cdots & S^{1} \\ \cdot & & & \\ \cdot & & & \\ \cdot & & & \\ S^{2k-2} & S^{2k-3} & \cdots & S^{k-1} \end{vmatrix} = \begin{vmatrix} 1 & 1 & & 1 \\ z_{\ell_1} & z_{\ell_2} & & z_{\ell_k} \\ & & & \\ z_{\ell_1}^{k-1} & & & z_{\ell_k}^{k-1} \end{vmatrix} \begin{vmatrix} z_{\ell_1}^{k-1} & & z_{\ell_1}^{1} \\ z_{\ell_2}^{k-1} & & z_{\ell_2}^{1} \\ & & \\ z_{\ell_k}^{k-1} & & z_{\ell_k}^{1} \end{vmatrix} = \Delta_1 \cdot \Delta_2$$

Since the product of the matrices corresponding to the two last determinants is the matrix corresponding to the determinant on the left side.

Since

$$\begin{vmatrix} 1 & 1 & \cdots & 1 \\ z_{\ell_1} & z_{\ell_2} & & z_{\ell_k} \\ z_{\ell_1}^{k-1} & z_{\ell_2}^{k-1} & & z_{\ell_k}^{k-1} \end{vmatrix} = \prod_{i=1, j>i}^{k} (z_{\ell_j} - z_{\ell_i}) \text{ and } \begin{vmatrix} z_{\ell_1}^{k-1} & \cdots & z_{\ell_1} & 1 \\ z_{\ell_2}^{k-1} & \cdots & z_{\ell_2} & 1 \\ z_{\ell_k}^{k-1} & \cdots & z_{\ell_k} & 1 \end{vmatrix} = \prod_{i=1, j<i}^{k} (z_{\ell_j} - z_{\ell_i})$$

follows'that

$$\Sigma_{II} = \prod_{i=1}^{k} \prod_{j=1, j\neq i}^{k} (z_{\ell_i} - z_{\ell_j}) \qquad (=(-1)^{\frac{k(k-1)}{2}} \prod_{i=1, j>i}^{k} (z_{\ell_i} - z_{\ell_j})^2 \text{ , })$$

Hence

$$H_n^k = \Sigma_I \prod_{i=1}^{k} c_{\ell_i} \prod_{i=1}^{k} (z_{\ell_i})^{n-k+1} \prod_{i=1}^{k} \prod_{j=1, j\neq i}^{k} (z_{\ell_i} - z_{\ell_j}) = \Sigma_I \prod_{i=1}^{k} (c_{\ell_i} \prod_{j=1, j\neq i}^{k} (z_{\ell_i} - z_{\ell_j}) z_{\ell_i}^{n-k+1})$$

where the sum must be taken over all $\binom{N}{k}$ combinations of k roots taken of the N roots.

With

$$c_{\ell_i} = \frac{z_{\ell_i}^{N-1}}{\prod_{j=1, j\neq \ell_i}^{N} (z_{\ell_i} - z_j)} = \frac{z_{\ell_i}^{N-1}}{\prod_{j=1, j\neq i}^{N} (z_{\ell_i} - z_{\ell_j})}$$

we obtain

$$H_n^k = \Sigma_I \prod_{i=1}^{k} \frac{z_{\ell_i}^{N-k+n}}{\prod_{j=k+1}^{N} (z_{\ell_i} - z_{\ell_j})}$$

or

$$(4.9) \qquad H_n^k = \sum_{\substack{\text{all } \binom{N}{k} \\ \text{comb}}} \frac{(z_{\ell_1} \cdots z_{\ell_k})^{N-k}}{\prod_{i=1}^{k} \prod_{j=k+1}^{N} (z_{\ell_i} - z_{\ell_j})} (z_{\ell_1} \cdots z_{\ell_k})^n$$

and we have proved theorem 4.2.

The formula (4.9) may be written as

$$(4.10) \qquad H_n^k = \sum_{\substack{\text{all } \binom{N}{k} \\ \text{comb}}} \frac{(z_{\ell_1} \cdots z_{\ell_k})^n}{\prod_{i=1}^{k} \prod_{j=k+1}^{N} (1 - z_{\ell_j}/z_{\ell_i})} .$$

4.2 General convergence theorems.

By means of the formula (4.10) we may prove

Theorem 4.3

Let $z_1 > z_2 > \cdots > z_N > 0$.

Then

$$\lim_{n\to\infty} \frac{H_n^k}{H_{n-1}^k} = z_1 \cdots z_k .$$

Proof

Since the roots are different we have

$$(4.11) \qquad \frac{H_n^k}{H_{n-1}^k} = (z_1 z_2 \cdots z_k) \frac{1 + \sum \left(\prod_{i=1}^{k} \prod_{j=k+1}^{N} \left(\frac{1 - z_j/z_i}{1 - z_{\ell_j}/z_{\ell_i}} \right) \left(\frac{z_{\ell_1} \cdots z_{\ell_k}}{z_1 \cdots z_k} \right)^n \right)}{1 + \sum \left(\prod_{i=1}^{k} \prod_{j=k+1}^{N} \left(\frac{1 - z_j/z_i}{1 - z_{\ell_j}/z_{\ell_i}} \right) \left(\frac{z_{\ell_1} \cdots z_{\ell_k}}{z_1 \cdots z_k} \right)^{n-1} \right)}, \qquad (n > k)$$

where the sums now are taken over all $\binom{N}{k} - 1$ combinations $z_{\ell_1} \cdots z_{\ell_k}$ different from $z_1 \cdots z_k$.

Since $z_1 > z_2 > \cdots > z_N > o$ it follows that

$$\frac{z_{i_1} \cdots z_{i_k}}{z_1 \cdots z_k} < 1$$

for all combinations in question. This means that all the terms in the sums in both the numerator and the denominator tend to zero, and since there are a finite number of terms in these sums the fraction in (4.11) tends to 1. Hence we have proved theorem 4.3.

<u>Multiple roots</u>

<u>Theorem 4.4</u>

Let $z_1 \geq z_2 > \cdots \geq z_N > 0$.

Then

$$\lim_{n\to\infty} \frac{H_n^k}{H_{n-1}^k} = z_1 \cdots z_k \, ;$$

that is the result from theorem 4.3 is true also for the case where one or several roots of $p_N(x) = o$ are of multiplicity greater than one.

<u>Proof</u>

We begin with the case where one of the N roots, say z_r, is of multiplicity 2, and the remaining (N - 2) roots are single roots; that is the roots of $p_N(x) = 0$ are $z_1 > z_2 > \cdots > z_{r-1} = z_r > \cdots > z_N$.

Now we consider the polynomial equation $p_N^*(x) = o$, which has the roots $z_1 > z_2 > \cdots > z_r + \epsilon > z_r > \cdots > z_N$.

Let $H_n^k(\epsilon)$ denote the Hankel determinant corresponding to this equation. From the definition of $H_n^k(\epsilon)$ as a determinant in the complete symmetric functions follows that $H_n^k(\epsilon)$ is a continuous function of ϵ. Hence we find $H_n^k(o) = \lim_{\epsilon\to\infty} H_n^k(\epsilon)$.

By means of (4.9) we may write

$$(4.11) \qquad H_n^k(\epsilon) = \frac{(z_1 z_2 \cdots z_k)^{N+n-k}}{(z_1-z_{k+1})\cdots(z_k-z_N)} + \ldots + \frac{(z_{N-k+1} \cdots z_N)^{N+n-k}}{(z_{N-k+1}-z_1)\cdots(z_N-z_{N-1})} ,$$

where $z_{r-1} = z_r + \epsilon$.

The terms of (4.11) in which ϵ occurs in the demoninator must be combined; that is we'have to consider all combinations $(z_{\ell_1} \cdots z_{\ell_k})$ of which z_{r-1} but not z_r is a factor and all combinations where z_r but not z_{r-1} is a factor.

There are $\binom{N-2}{k-1}$ combinations of each kind; we take them pairwise as in the following example where we assume $r > k$

$$u = \frac{(z_1 \cdots z_{k-1} z_{r-1})^{N+n-k}}{\prod_{i=1}^{k-1} (z_i-z_r)(z_{r-1}-z_r) \prod_{j=k, j\neq r,r-1}^{N} (z_{r-1}-z_j) \prod_2} + \frac{(z_1 \cdots z_{k-1} z_r)^{N+n-k}}{\prod_{i=1}^{k-1} (z_i-z_{r-1})(z_r-z_{r-1}) \prod_{j=k, j\neq r,r}^{N} (z_r-z_j) \prod_2},$$

where $\quad \prod_2 = \prod_{i=1}^{k-1} \prod_{j=k, j\neq r,r-1}^{N} (z_i-z_j)$.

Then with $z_{r-1} = z_r + \epsilon$ we obtain

$$u = \frac{\cdot z_1 \cdot . \bullet \; ⌘\& \;_1)^{N+n-k}}{\prod_2} \left[\frac{(z_r + \epsilon)^{N+n-k}}{\epsilon \prod_{\substack{1-r' \\ i=1}}^{k-1} z_i - z_r) \prod_{j=k,j\neq r,r-1}^{N} (z_r - z_j + \epsilon)} - \frac{z_r^{N+n-k}}{\epsilon \prod_{i=1}^{k-1} (z_i - z_r - \epsilon) \prod_{j=k,j\neq r-1,r}^{N} (z_r - z_j)} \right]$$

$$= \frac{(z_1 \cdot . \bullet \; ⌘\& \;_1)^{N+n-k}}{\prod_2} \; \frac{(z_r+\epsilon)^{N+n-k} \prod_{i=1}^{k-1}(z_i - z_r - \epsilon) \prod_{j=k,j\neq r-1,r}^{N} (z_r - z_j) - z_r^{N+n-k} \prod_{i=1}^{k-1}(z_i - z_r) \prod_{j=k,j\neq r,r-1}^{N} (z_r - z_j + \epsilon)}{\epsilon \prod_{i=1}^{k-1}(z_i - z_r) \prod_{j=k,j\neq r,r-1}^{N} (z_r - z_j + \epsilon) \prod_{i=1}^{k-1}(z_i - z_r - \epsilon) \prod_{j=k,j\neq r-1,r}^{N} (z_r - z_j)}$$

Let $t(e)$ and $b\;\epsilon)$ denote the numerator and the denominator of the last fraction, respectively.

Then $t(o) = \mathbf{b(0)} = 0$; $b'(o) = (\prod_{j=1,j\neq r,r-1}^{N} (z_r - z_j)^2)$.

We find

$$t'(o) = \binom{N+n-k}{1} z_r^{N+n-k-1} (-1)^{k-1} \prod_{j=1,j\neq r,r-1}^{N} (z_r - z_j) - (-1)^{k-1} \sum_{i=1}^{k-1} \left(\frac{1}{z_i - z_r}\right) \prod_{j=1,j=r,r-1}^{N} (z_r - z_j) z_r^{N+n-k} -$$

$$- (-1)^{k-1} \sum_{j=k,j\neq r,r-1}^{N} \left(\frac{1}{z_r - z_j}\right) \prod_{i=1,j\neq r,r-1}^{N} (z_r - z_j) \; z_r^{N+n-k} \; .$$

Hence

$$\lim_{\epsilon \to \infty} u = (-1)^{k-1} \frac{(z_1 \cdot . \cdot z_{k-1} z_r)^{N+n-k}}{\prod_{j=1,j\neq r,r-1}^{N} (z_r - z_j)^2 \prod_{i=1}^{\mathbf{k-1}} \prod_{j=k,j\neq r,r-1}^{N} (z_i - z_j)} \binom{N+n-k}{1} \frac{1}{z_r} +$$

(4.12)

$$+ (-1)^k \frac{(z_1 \cdots z_{k-1} z_r)^{N+n-k}}{\prod_{j=1,j\neq r,r-1}^{N} (z_r - z_j)^2 \prod_{i=1}^{k-1} \prod_{j=k,j\neq r,r-1}^{N} (z_i - z_j)} \left(\sum_{i=1}^{k-1} \left(\frac{1}{z_i - z_r}\right) + \sum_{j=k,j\neq r,r-1}^{N} \left(\frac{1}{z_r - z_j}\right) \right),$$

where lim u is written as the sum of two terms in order to preserve the number of terms in H_n^k. The limits of the remaining pairs obviously have the same structure as (4.12), and we may write a formula for H_n^k covering the case of N - 1 different roots.

In this formula, which again consists of $\binom{N}{k}$ terms of the form $c^{(n)}_{\ell_1 \cdots \ell_k}(z_{\ell_1} \cdots z_{\ell_k})^{N+n-k}$, the coefficients may have a factor $\binom{N+n-k}{1}$.

By means of the technique used above we may use the first result to obtain new formulas covering the cases two roots of multiplicity 2 or one root of multiplicity 3 (all other roots single in both cases) etc. until we obtain the following result:

Let r be the number of different roots of $p_N(x) = o$, and let the multiplicity of these roots be $m_1, m_2, \ldots m_r$, respectively.

Then we may write H_n^k in the form

$$H_n^k = \sum_{\substack{\text{all } \binom{N}{k} \\ \text{comb.}}} c^{(n)}_{\ell_1 \cdots \ell_k} (z_{\ell_1} \cdots z_{\ell_k})^{N+n-k} \tag{4.13}$$

In this formula $c^{(n)}_{\ell_1 \cdots \ell_k}$ is of the form

$$c^{(n)}_{\ell_1 \cdots \ell_k} = \frac{c_{\ell_1 \cdots \ell_k}}{\prod (z_i - z_j)^{\alpha}} \binom{N+n-k}{p_{\ell_1}} \cdots \binom{N+n-k}{p_{\ell_k}} , \tag{4.14}$$

where $c_{\ell_1 \cdots \ell_k}$ is a constant; $\prod (z_i - z_j)^{\alpha}$ contains powers of the differences between <u>different roots</u> and $o \leq p_{\ell_i} \leq m_i - 1$ $(1 \leq i \leq r)$. By means of (4.13) we obtain

$$(4.15)\qquad \frac{H_n^k}{H_{n-1}^k} = (z_1\cdots z_k)\,\frac{1+\sum \frac{c^{(n)}_{\ell_1\cdots\ell_k}}{c^{(n)}_{1\cdots k}}\left(\frac{z_{\ell_1}\cdots z_{\ell_k}}{z_1\cdots z_k}\right)^{N+n-k}}{1+\sum \frac{c^{(n-1)}_{\ell_1\cdots\ell_k}}{c^{(n-1)}_{1\cdots k}}\left(\frac{z_{\ell_1}\cdots z_{\ell_k}}{z_1\cdots z_k}\right)^{N+n-1-k}}$$

where the sums are taken over all $\binom{N}{k} - 1$ combinations $(z_{\ell_1}\cdots z_{\ell_k})$ different from $(z_1\cdots$ zk).

Among the combinations $(z_{\ell_1}\cdots z_{\ell_k})$ there may be some for which $z_{\ell_1}\cdots z_{\ell_k} = z_1\cdots z_k$, and among these we choose the term with $\max\left[\binom{N+n-k}{p_{\ell_1}}\cdots\binom{N+n-k}{p_{\ell_k}}\right]$. By division in the nominater and the denominater, respectively, with these functions of n, the fraction in (4.15) will tend to 1 as n tends to infinity. Since

$$\lim_{n\to\infty}\left(\left[\binom{N+n-k}{p_{\ell_1}}\cdots\binom{N+n-k}{p_{\ell_k}}\right]/\left[\binom{N+n-1-k}{p_{\ell_1}}\cdots\binom{N+n-1-k}{p_{\ell_k}}\right]\right) = 1 .$$

we have

$$\lim_{n\to\infty}(H_n^k/H_{n-1}^k) = z_1\cdots z_k$$

and we have proved theorem 4.4.

<u>Theorem 4.5</u>

Let $z_1 \geq z_2 > \cdots \geq z_N > 0$.

Then

$$q_n^k \to z_k \quad \text{as} \quad n\to\infty .$$

Proof

From theorem 3.1 we have

$$q_n^k = \frac{H_n^k}{H_{n-1}^k} \frac{H_{n-2}^{k-1}}{H_{n-1}^{n-1}}$$

$$= \frac{H_n^k}{H_{n-1}^k} \Bigg/ \frac{H_{n-1}^{k-1}}{H_{n-2}^{k-1}}$$

Hence by means of theorem 4.4 -

$$\lim_{n\to\infty} q_n^k = (z_1 \cdots z_k)/(z_1 \cdots z_{k-1}) = z_k$$

Theorem 4.6

Let $z_1 \geq z_2 \geq \cdots \geq z_N > o$

Then

$$e_n^k \to o \text{ as } n \to \infty$$

Proof

By induction with respect to k.

k = 1

From (1.2) we have

$$q_{n+1}^1 = e_n^1 - e_n^o + q_n^1$$

or - since $e_n^o = 0$ -

$$e_n^1 = q_{n+1}^1 - q_n^1$$

Hence

$$\lim_{n\to\infty} e_n^1 = \lim_{n\to\infty} q_{n+1}^1 - \lim_{n\to\infty} q_n^1$$

$$= z_1 - z_1$$

$$= 0 .$$

We assume theorem (4.6) holds for k - 1 and consider the case k. Again (1.2) may be used. We obtain

$$e_n^k = q_{n+1}^k - q_n^k + e_n^{k-1} \quad .$$

Hence

$$\lim_{n\to\infty} e_n^k = \lim_{n-9} q_{n+1}^k - \lim_{n\to\infty} q_n^k + \lim_{n\to\infty} e_n^{k-1}$$

$$= z_k - z_k + o$$

$$= o ,$$

and we have proved theorem 4.6 by induction.

In special examples the theorems 4.5 and 4.6 may be proved without using theorem 4.4. We consider two cases.

<u>Example 4.1</u>

<u>$N = 2$</u>

(i) $z_1 > z_2 > 0$

Now $H_n^1 = S_n = z_1^n + z_1^{n-1} z_2 + \cdots + z_2^n = \dfrac{z_1^{n+1} - z_2^{n+1}}{z_1 - z_2}$; $H_n^2 = (z_1 z_2)^n$ and we find directly by means of (3.1) and (3.2):

$$q_n^1 = \frac{z_1^{n+1} - z_2^{n+1}}{z_1^n - z_2^n} \quad ; \quad e_n^1 = -(z_1 \, z_2)^n \, \frac{(z_1 - z_2)^2}{(z_1^{n+1} - z_2^{n+1})(z_1^n - z_2^n)}$$

$$q_n^2 = z_1 \, z_2 \, \frac{z_1^{n-1} - z_2^{n-1}}{z_1^n - z_2^n} \quad .$$

From these formulas it is obvious that

$$q_n^1 \to z_1, \; q_n^2 \to z_2 \quad \text{and} \quad e_n^1 \to o \quad \text{as} \quad n \to \infty$$

(ii) $z_1 = z_2 > 0$

Then $H_n^1 = S_n = (n + 1) \, z_1^n$; $H_n^2 = z_1^{2n}$, and we find

$$q_n^1 = \frac{n+1}{n} \, z_1; \quad e_n^1 = - \frac{z_1^{2n} \cdot 1}{(n+1) z_1^n \; n \; z_1^{n-1}} = \frac{-z_1}{(n+1)n} \quad ;$$

$$q_n^2 = \frac{z_1^{2n}}{z_1^{2(n-1)}} \; \frac{(n-1) z_1^{n-2}}{n \, z_1^{n-1}} = \frac{n-1}{n} z_1 = \frac{n-1}{n} \, z_2 \, .$$

Again it is obvious that

$$q_n^1 \to z_1, \; q_n^2 \to z_2 (= z_1) \text{ and } e_n^1 \to o \text{ as } n \to \infty$$

Example 4.2

N arbitrary; all roots equal, that is

$$z_1 = z_2 = \cdots = z_N > 0$$

We have

$$\sigma_k = \binom{N}{k} z_1^k \qquad k = 1, \ldots, N$$

Now

$$q_2^k = \frac{H_2^k\, H_o^{k-1}}{H_1^k\, H_1^{k-1}} = \frac{\sigma_k^2 - \sigma_{k-1}\,\sigma_{k+1}}{\sigma_k\,\sigma_{k-1}}$$

(4.16)

$$= \frac{\binom{N}{k}^2 - \binom{N}{k-1}\binom{N}{k+1}}{\binom{N}{k-1}\binom{N}{k}} \quad z_1 \qquad k = 1, 2, \ldots N$$

and

$$e_2^k = -\frac{H_2^{k+1}\; H_1^{k-1}}{H_2^k\; H_1^k} = -\frac{(\sigma_{k+1})^2 - \sigma_k\,\sigma_{k+2}}{\sigma_k^2 - \sigma_{k-1}\,\sigma_{k+1}} \cdot \frac{\sigma_{k-1}}{\sigma_k}$$

$$= -\frac{\binom{N}{k+1}^2 - \binom{N}{k}\binom{N}{k+2}}{\binom{N}{k}^2 - \binom{N}{k-1}\binom{N}{k+1}} \cdot \frac{\binom{N}{k-1}}{\binom{N}{k}} \quad z_1 \qquad k = 1, 2, \ldots, N-1$$

Since

$$\binom{N}{k}^2 - \binom{N}{k-1}\binom{N}{k+1} = \binom{N}{k-1}^2 \frac{N-k+1}{k} \cdot \frac{N+1}{k(k+1)}$$

(4.16) and (4.17) may be written in the following form:

$$q_2^k = \frac{N+1}{k(k+1)} \quad z_1 \qquad k = 1, \bullet \ldots N$$

$$e_2^k = -\frac{(N-k)\,k}{(k+1)(k+2)} \; z_1 \qquad k = 1, \ldots . N-1$$

By induction we may prove that

(4.18) $$q_n^k = \frac{(n-1)(N+n-1)}{(k+n-2)(k+n-1)} z_1 \qquad k = 1, \ldots, N$$

(4.19) $$e_n^k = -\frac{(N-k)k}{(k+n-1)(k+n)} z_1 \qquad k = 1, \ldots N-1$$

For n = 2 (4.18) and (4.19) holds as we have shown above.

Now we assume that (4.18) and (4.19) holds for n and consider the case n + 1:

$$q_{n+1}^k = e_n^k - e_n^{k-1} + q_n^k$$

$$= \left[\frac{(N-k+1)(k-1)}{(k+n-2)(k+n-1)} - \frac{(N-k)k}{(k+n-1)(k+n)} + \frac{(n-1)(N+n-1)}{(k+n-2)(k+n-1)}\right]$$

$$= \frac{n(N+n)}{(k+n-1)(k+n)} z_1$$

which is (4.18) with n + 1 instead of n.

Then

$$e_{n+1}^k = (q_{n+1}^{k+1}/q_{n+1}^k)\, e_n^k$$

$$= -\frac{(N-k)k}{(k+n)(k+n+1)} z_1$$

which is (4.19) with n + 1 instead of n.

From the formulas (4.18) and (4.19) we find

$$q_n^k \to z_k (=z_1) \quad \text{and} \quad e_n^k \to o \quad \text{as} \quad n \to \infty.$$

5. The convergence to the smallest root

The formulas developed in section 4 show that the convergence of the q-columns may be very slow. In this section we shall examine the question of the speed of convergence to the smallest root z_N of $p_1(x) = o$. Furthermore we shall show that it is possible to use an acceleration technique to obtain faster convergence to the smallest root.

5.1 A formula for q_n^N

In 'section 4 we have given a qualitative formula for H_n^k valid for the case of multiple roots. In order to examine the convergence of q_n^N to z_N in detail we need a precise formula for q_n^N which cover the case of multiple roots. As usual we assume that $z_1 \geq \ldots \geq z_N > o$.

Lemma 5.1

$$(5.1) \qquad H_n^{N-1} = \sigma_N^n \, S_n\left[\frac{1}{z_1}, \frac{1}{z_2}, \cdots, \frac{1}{z_N} \right]$$

Proof

By definition

$$H_n^{N-1} = \begin{vmatrix} \sigma_{N-1} & \sigma_{N-2} & \cdot & \cdot & & \\ \sigma_N & \sigma_{N-1} & & & & \\ & \cdot & \cdot & & & \\ & & \cdot & \cdot & & \\ & & & \cdot & \cdot & \\ & & & & \sigma_N & \sigma_{N-1} \end{vmatrix}_{(n \text{ rows})} = \sigma_N^n \begin{vmatrix} \sigma_1(\frac{1}{z_1} \ldots \frac{1}{z_N}) & \sigma_2(\frac{1}{z_1} \ldots \frac{1}{z_N}) & & \\ 1 & \sigma_1(\frac{1}{z_1} \ldots \frac{1}{z_N}) & & \\ & 1 & \cdot & \cdot \\ & & \cdot & \cdot \\ & & 1 & \sigma_1(\frac{1}{z_1} \ldots \frac{1}{z_N}) \end{vmatrix}$$

In the proof of theorem 2.3 we have shown that the last determinant has the value $S_n[\frac{1}{z_1}, \frac{1}{z_2}, \ldots \frac{1}{z_N}]$ and we have proved lemma 5.1.

Lemma 5.2

Let $p(N, n)$ denote the number of terms in the complete symmetric function S_n in N variables.

Then

$$(5.2) \qquad p(N, n) = \binom{N+n-1}{N-1} = \binom{N+n-1}{n} \qquad N > 2 \qquad n \geq o$$

<u>proof</u>

By induction with respect to n.

<u>n = o</u>

Since $S_o = 1$ and $\binom{N-1}{N\ 1} = 1$ (5.2) is correct for $N \geq 2$. We assume that (5.2) holds for o, 1, 2 . . . . n-1 and all $N \geq 2$ and consider the case n. By means of the relation $S_n[z_1 \cdots z_N] = z_1 S_{n\ 1}[z_1 \cdots z_N] + S_n[z_2 \cdots z_N]$, which has been proved in theorem 2.1, we may obtain

$$\begin{aligned} p(N, n) &= p(N, n-1) + p(N-1, n) \\ &= \binom{N+n-2}{N-1} + p(N-1, n) \\ &= \cdots \\ &= \binom{N+n-2}{N-1} + \binom{N+n-3}{N-2} + \cdots + \binom{n}{1} + 1 \end{aligned}$$

where we have used that $p(1, n) = 1$.

Since

$$1 + \binom{n}{1} + \binom{n+1}{2} + .** + \binom{N+n-2}{N-1}$$

$$= \binom{n+1}{1} + \binom{n+1}{2} + \cdots + \binom{N+n-2}{N-1}$$

$$= \binom{n+2}{2} + \ldots + \binom{N+n-2}{N-1}$$

$$= \cdots$$

$$= \binom{N+n-1}{N-1} ,$$

we have proved lemma 5.2.

Lemma 5.3

Let z_1 be of multiplicity m $(1 \leq m \leq N)$ and let the other roots be different. Then

$$S_n[z_1,\ldots z_1,\ z_{m+1},\ldots z_N]$$

$$= \frac{z_1^{N+n-m}}{\prod_{j=m+1}^{N}(z_1-z_j)} \left[\binom{N+n-1}{m-1} - S_1[\frac{1}{z_1-z_{m+1}},\ldots\frac{1}{z_1-z_N}]\binom{N+n-1}{m-2}z_1 + \cdots + (-1)^{m-1}S_{m-1}[\frac{1}{z_1-z_{m+1}},\ldots\frac{1}{z_1-z_n}]z_1^{m-1}\right]$$

$$= \sum_{j=m+1}^{N} \frac{z_i^{N+n-1}}{(z_i-z_1)^m \prod_{j=m+1}^{N}(z_i-z_j)} .$$

Proof

The proof may be given by means of the limit technique used in section 4. In this case however the notation is so much handier that we may prove (5.3) by induction with respect to N.

N 2

If $z_1 > z_2$ (5.3) is nothing but (4.9) which is correct and if $z_1 = z_2$ we find $S_n[z_1 z_1] = (n+1)z_1^n$ which is correct too. We assume (5.2) is true in all cases with N-1 roots and consider the case with N roots.

If z_1 is of multiplicity N, we find by (5.3) that $S_n[z_1 \ldots z_1] = \binom{N+n-1}{N-1} z_1^n$. From lemma 5.1 follows that this is correct. If m=1 (5.3) again is (4.9) and we may assume that $1 < m < N$.

By means of (4.4) we have

$$S_n[\underbrace{z_1,\ldots z_1}_{m}, z_{m+1},\ldots z_N]$$

$$= \left[S_{n+1}[\underbrace{z_1,\ldots z_1}_{m}, z_{m+2},\ldots z_N] - S_{n+1}[\underbrace{z_1,\ldots z_1}_{m-1}, z_{m+1},\ldots z_N]\right] \Big/ (z_1 - z_{m+1})$$

The complete symmetric functions in the parentheses are functions of N-1 variables and we may use (5.3) to obtain

$$S_n = \left[-\frac{z_1^{N+n-m}}{\prod_{j=m+2}^{N}(z_1-z_j)} \left[\binom{N+n-1}{m-1} - S_1[\frac{1}{z_1-z_{m+2}},\ldots,\frac{1}{z_1-z_N}] \binom{N+n-1}{m-2} z_1 + \ldots \right]\right.$$

$$+ \sum_{i=m+2}^{N} \frac{z_i^{N+n-1}}{(z_i-z_1)^m \prod_{j=m+2, j\neq i}^{N} (z_i - z_j)} \qquad \sum_{i=m+1}^{N} \frac{z_i^{N+n-1}}{(z_i-z_1)^{m-1} \prod_{j=m+1, j\neq i}^{N} (z_i-z_j)}$$

$$\left.\frac{z_1^{N+n-m+1}}{\prod_{j=m+1}^{N}(z_1-z_j)} \left[\binom{N+n-1}{m-2} - S_1[\frac{1}{z_1-z_{m+1}},\ldots \frac{1}{z_1-z_N}] \binom{N+n-1}{m-3} z_1 + \cdots \right] \right] \Big/ (z_1 - z_{m+1})$$

By reduction of corresponding terms and by use of the formula

$$S_r[\frac{1}{z_1-z_{m+2}},\ldots,\frac{1}{z_1-z_N}] + \frac{1}{z_1-z_{m+1}} S_{r-1}[\frac{1}{z_1-z_{m+1}},\ldots \frac{1}{z_1-z_N}] = S_r[\frac{1}{z_1-z_{m+1}},\ldots,\frac{1}{z_1-z_N}]$$

we end up with (5.3). Lemma 5.3 has been proved by induction.

As an obvious consequence of lemma 5.3 we have the following general result

Lemma 5.4

Let $p_N(x) = o$ have r different roots $z_1 > z_2 > ..0 > z_r > 0$ of multiplicity $m_1, m_2, \ldots, m_r$ respectively. $(\sum m_i = N)$. Then - with the notation $S_p[\frac{m_j}{z_i - z_j}] = S_p[\underbrace{\frac{1}{z_i-z_1}, \frac{1}{z_i-z_1}, \ldots \frac{1}{z_i-z_1}}_{m_1}, \ldots, \underbrace{\frac{1}{z_i-z_r}, \ldots \frac{1}{z_i-z_r}}_{m_r}]\ (j \neq i)$ -

$$S_n[z_1 \ldots z_1, \ldots, z_r \ldots z_r]$$

$$= \sum_{i=1}^{r} \frac{z_i^{N+n-m_i}}{\prod_{j=1, j\neq i}^{r} (z_i - z_j)^{m_j}} \left[\binom{N+n-1}{m_i - 1} - S_1[\frac{m_j}{z_i - z_j}] \binom{N+n-1}{m_i - 2} z_i + \cdots + (-1)^{m_i - 1} S_{m_i - 1}[\frac{m_j}{z_i - z_j}] z_i^{m_i - 1} \right]$$

Now

$$q_n^N = \frac{H_n^N \quad H_{n-2}^{N-1}}{H_{n-1}^N \quad H_{n-1}^{N-1}}$$

$$= \sigma_N \cdot \frac{H_{n-2}^{N-1}}{H_{n-1}^{N-1}} ,$$

which by means of lemma (5.1) may be written as

(5.5) $$q_n^N = \frac{S_{n-2}[\frac{1}{z_1}, \ldots, \frac{1}{z_N}]}{S_{n-1}[\frac{1}{z_1}, \ldots, \frac{1}{z_N}]} .$$

We use the notation from lemma 5.4 and obtain

$$q_n^N = \sum_{i=1}^{r} \frac{(\frac{1}{z_1})^{N+n-m_i-2}}{\prod_{j=1,j\neq i}^{r} (\frac{1}{z_i} - \frac{1}{z_j})^{m_j}} \left[\binom{N+n-3}{m_i-1}+\cdots\right] \Big/ \sum_{i=1}^{r} \frac{(\frac{1}{z_i})^{N+n-m_i-1}}{\prod_{j=1,j\neq i}^{r} (\frac{1}{z_i} - \frac{1}{z_j})^{m_j}} \binom{N+n-2}{m_i-1}+\cdots]$$

In this formula z_r denotes the smallest root of $p_N(x) = o$. Furthermore, both the denominator and the numerator consists of N terms.

5.2 The monotonic convergence of q_n^N

We consider

$$\epsilon_n = q_n^N - z_N$$
$$= q_n^N - z_r$$

and treat the two cases $m_r = 1$ and $m_r > 1$ separately.

$m_r = 1$; that is the smallest root is a single root.

By means of (5.5) we find

$$\epsilon_n = \left[\left(\frac{(\frac{1}{z_r})^{N+n-3}}{\prod_{j=1}^{r-1}(\frac{1}{z_r} - \frac{1}{z_j})^{m_j}} + \frac{(\frac{1}{z_{r-1}})^{N+n-m_{r-1}-2}}{\prod_{j=1,j\neq r-1}^{r}(\frac{1}{z_{r-1}} - \frac{1}{z_j})^{m_j}} \left[\binom{N+n-3}{m_{r-1}-1}+\cdots + \sum_{i=1}^{r-2} \text{num}\right)\right.$$

(5.6)
$$\left. - z_r\left(\frac{(\frac{1}{z_r})^{N+n-2}}{\prod_{j=1}^{r-1}(\frac{1}{z_r} - \frac{1}{z_j})^{m_j}} + \frac{(\frac{1}{z_{r-1}})^{N+n-mr-1-1}}{\prod_{j=1,j\neq r-1}^{r}(\frac{1}{z_{r-1}} - \frac{1}{z_j})^{m_j}} \left[\binom{N+n-2}{m_{r-1}-1}+\cdots\right] + \sum_{i=1}^{r-2} \text{denom}\right)\right] \Big/$$

$$\left(\frac{(\frac{1}{z_r})^{N+n-2}}{\prod_{j=1}^{r-1}(\frac{1}{z_r} - \frac{1}{z_j})^{m_j}} + \cdots\right) .$$

From (5.6) follows, that ϵ_n may be written in the form

$$(5.7) \qquad \epsilon_n = c \left(\frac{z_r}{z_{r-1}}\right)^n b(n) ,$$

where $b(n) \to 1$ as $n \to \infty$.

Hence we have proved

Theorem 5.1

Let $z_1 \geq z_2 \geq \cdots \geq z_{N-1} > z_N > 0$.

Then

$$\frac{\epsilon_{n+1}}{\epsilon_n} \approx \frac{z_N}{z_{N-1}}$$

$m_r > 1$; that is the smallest root is a multiple root.

By means of (5.5) we find

$$(5.8) \qquad \epsilon_n = \left[\left(\frac{\left(\frac{1}{z_r}\right)^{N+n-m_r-2}}{\prod_{j=1, j\neq i}^{r}\left(\frac{1}{z_i} - \frac{1}{z_j}\right)^{m_j}}\left[\binom{N+n-3}{m_r-1}+\cdots\right] + \sum_{i=1}^{r-1} \text{num}\right) - z_r\left(\frac{\left(\frac{1}{z_r}\right)^{N+n-mr-1}}{\prod_{j=1, j\neq i}^{r}\left(\frac{1}{z_i} - \frac{1}{z_j}\right)^{m_j}}\left[\binom{N+n-2}{m_r-1}+\cdots\right] + \sum_{i=1}^{r-1} \text{denom}\right)\right] \Bigg/ \left(\frac{\left(\frac{1}{z_r}\right)^{N+n-m_r-1}}{\prod_{j=1, j\neq i}^{r}\left(\frac{1}{z_i} - \frac{1}{z_j}\right)^{m_j}}\left[\binom{N+n-2}{m_r-1}+\cdots\right] + \sum_{i=1}^{r-1} \text{denom}\right).$$

From (5.8) follows, that ϵ_n may be written in the form

$$\epsilon_n = c \frac{\binom{N+n-3}{m_r-2}}{\binom{N+n-2}{m_r-1}} z_r \quad b(n)$$

$$= c \frac{m_r-1}{(N+n-2)}$$

where $b(n) \to 1$ as $n \to \infty$.

We have proved

Theorem 5.2

Let $z_1 \geq z_2 > \cdots \geq z_{N-1} = z_N > 0$, that is the smallest root is a multiple root.

Then

$$\epsilon_n \text{ tends to zero as } \frac{1}{n} .$$

Theorem 5.3

The last column of the QD scheme forms a monotonically increasing sequence:

$$0 = q_1^N < q_2^N < \cdots < q_n^N < q_{n+1}^N < \ldots$$

Proof

Since

$$q_{n+1}^N = e_n^N - e_n^{N-1} + q_n^N$$

$$= -e_n^{N-1} + q_n^N ,$$

we have

$$q_{n+1}^N - q_n^N = -e_n^{N-1} > 0 ,$$

Since

$$e_n^{N-1} < 0 \quad \text{for all } n .$$

From theorem 5.3 and the convergence of q_n^N to z_N follows

(5.10) $$o < q_n^N < z_N \qquad n > 2$$

We remark, that a similar theorem concerning the convergence of q_n^1 to the largest root z_1 may be proved:

Let $z_1 \geq z_2 \geq \cdots \geq z_N > o$. Then

$$q_1^1 > q_2^1 > \cdots > q_n^1 > q_{n+1}^1 > \cdots > z_1$$

Theorem 5.4

Let $z_1 \geq z_2 \geq \cdots \geq z_N > o$, and let $N > 2$.

Then

(5.11) $$(N-1)\, q_n^N \geq z_N - q_n^N .$$

Proof

The proof is based on the following

lemma 5.5

For symmetric functions of N positive variables, where $N \geq 2$, and all $n > 1$

(5.12) $$S_n \leq "_1 S_{n-1}$$

For N = 2 (5.12) may be proved to hold for all n by direct calculation.

Now we assume, that (5.12) holds for N - 1 positive variables $z_1 \geq z_2 \geq \cdots \geq z_N > 0$. For n = 1 (5.12) holds. We assume (5.12) holds for N variables and for n and we have to prove that

$$S_{n+1} \leq \sigma_1 S_n \tag{5.13}$$

Now

$$\begin{aligned} S_{n+1} &= z_N S_n + S'_{n+1} \\ &\leq z_N S_n + \sigma'_1 S'_n \\ &\leq (z_N + \sigma'_1) S_n \\ &= \sigma_1 S_n \quad , \end{aligned}$$

where we have used

$$S_n = z_N S_{n-1} + S'_n \ ;$$

that is

$$S'_n \leq S_n$$

Hence we have proved lemma 5.5 by induction.

The equation (5.11) may be written in the form

$$N q_n^N \geq z_N$$

Since $N \geq \frac{z_N}{z_1} + \frac{z_N}{z_2} + \cdots + \frac{z_N}{z_N}$, we have by means of (5.5):

$$N q_n^N = N \frac{S_{n-2} [\frac{1}{z_1}, \ldots, \frac{1}{z_N}]}{S_{n-1} [\frac{1}{z_1}, \ldots, \frac{1}{z_N}]}$$

$$\geq z_n \frac{\sigma_1 [\frac{1}{z_1}, \ldots, \frac{1}{z_N}] \; S_{n-2} [\frac{1}{z_1} \ldots \frac{1}{z_N}]}{S_{n-1} [\frac{1}{z_1} \ldots \frac{1}{z_N}]} \quad ,$$

which result by means of lemma (5.5) shows that $N q_n^N \geq z_N$, and we have proved Theorem 5.4.

5.3 An acceleration device

The formulas (5.7) and (5.9), in which z_r denotes the smallest root of $p_N(x) = o$, proves the following

Theorem 5.5

Let $z_1 \geq z_2 \geq \ldots \geq z_N > o$ be the roots of $p_N(x) = o$, let $0 < c < z_N$, and let $p_N^*(x) = o$ have the roots

$$z_1 - c \geq z_2 - c \geq \cdots \geq z_N - c > o \; .$$

Then the convergence of the last column of the Q,D scheme corresponding to $p_N^*(x)$ will be faster than the convergence of the last column in the scheme corresponding to $p_N(x) = o$.

In order to use theorem 5.5 we have to find a constant c in the interval $o < c < z_N$. The formula (5.10) shows that an arbitrary element q_n^N $(n \geq 2)$ from the last column of the QD scheme may be used as the constant c in theorem 5.5.

The results from theorem 5.4 and 5.5 prove that the following algorithm may be used to find z_N within a prescribed error ϵ.

Algorithm

Let $z_1 \geq z_2 \geq \cdots \geq z_N > 0$ be the roots of $p_1(x) = 0$, and let $\epsilon > 0$ be an arbitrary real number.

Compute r_1 rows of the Q,D scheme. If $(N - 1) \times q_{r_1}^N \leq \epsilon$ then $z_N - q_{r_1}^N \leq \epsilon$ otherwise compute r_2 rows of the QD scheme corresponding to the polynomial with roots $z_1 - q_{r_1}^N, \ldots, z_N - q_{r_1}^N$. If $(N - 1) \times q_{r_2}^N \leq \epsilon$ then $z_N - q_{r_1}^N - q_{r_2}^N \leq \epsilon$ otherwise compute r_3 rows of the QD scheme corresponding to the polynomial with roots $z_1 - (q_{r_1}^N + q_{r_2}^N), \ldots, z_N - (q_{r_1}^N + q_{r_2}^N)$, etc.

6.- Stability of the QD-algorithm

6.1 The stability of the progressive form of the Q,D-algorithm

In the following considerations concerning the numerical stability of the QD algorithm we assume that the computations are carried out in floating point arithmetic on a computer for which the basic formulas of Wilkinson [18] holds. In Wilkinsons notation, if x and y are floating point numbers, then

$$fl(x+y) = (x+y)(1+\epsilon)$$

$$fl(x-y) = (x-y)(1+\epsilon)$$

$$fl(xy) = xy(1+\epsilon)$$

$$fl(x/y) = (x/y)(1+\epsilon) ,$$

where $|\epsilon| < 2^{-t}$, if the mantissa has t binary places. Since our examination will be quantitative only, the statements obtained in this section will also hold for computers for which the floating point addition and subtraction are less accurate than supposed in (6.1).

In this section we use the following notation:

(6.2) Q_n^k and E_n^k for the floating point numbers which actually are in the computer instead of q_n^k and e_n^k, respectively.

(6.3) $$r(q_n^k) = Q_n^k - q_n^k$$

(6.4) $$r(e_n^k) = E_n^k - e_n^k$$

(6.5) $$\delta(q_{n+1}^k) = Q_{n+1}^k - (E_n^k - E_n^{k-1} + Q_n^k)$$

(6.6) $$\delta(e_{n+1}^k) = E_{n+1}^k - (Q_{n+1}^{k+1}/Q_{n+1}^k \times E_n^k)$$

We want to express the errors on q_{n+1}^k and e_{n+1}^k, that is $r(q_{n+1}^k)$ and $r(e_{n+1}^k)$ by means of the errors from row n.

The formulas used in the progressive form of the QD algorithm are

(6.7) $$q_{n+1}^k = e_n^k - e_n^{k-1} + q_n^k$$

$$e_{n+1}^k = q_{n+1}^{k+1}/q_{n+1}^k \times e_n^k$$

In the computer these formulas may be substituted by means of

$$Q^k_{n+1} = [(E^k_n - E^{k-1}_n)\ (1 + \epsilon_1) + Q^k_n]\ (1 + \epsilon_2) \tag{6.9}$$

$$E^k_{n+1} = [(Q^{k+1}_{n+1}/Q^k_{n+1})\ (1 + \epsilon_3) \times E^k_n]\ (1 + \epsilon_4) \tag{6.10}$$

Now

$$\begin{aligned} r(q^k_{n+1}) &= Q^k_{n+1} - q^k_{n+1} \\ &= (E^k_n - E^{k-1}_n + Q^k_n) + \delta(q^k_{n+1}) - e^k_n - e^{k-1}_n + q^k_n) \\ &= (E^k_n - e^k_n) - (E^{k-1}_n - e^{k-1}_n) + (Q^k_n - q^k_n) + \delta(q^k_{n+1}). \end{aligned}$$

and we obtain

$$r(q^k_{n+1}) = r(e^k_n) - r(e^{k-1}_n) + r(q^k_n) + \delta(q^k_{n+1}) . \tag{6.11}$$

Furthermore

$$r(e^k_{n+1}) = \frac{Q^{k+1}_{n+1}}{Q^k_{n+1}} \times E^k_n + \delta(e^k_{n+1}) - \frac{q^{k+1}_{n+1}}{q^k_{n+1}} \times e^k_b$$

which may be approximated by

$$r(e^k_{n+1}) \approx \frac{e^k_n}{q^k_{n+1}}\, r(q^{k+1}_{n+1}) + \frac{q^{k+1}_{n+1}}{q^k_{n+1}}\, r(e^k_n) \quad \frac{q^{k+1}_{n+1}\, e^k_n}{(q^k_{n+1})^2}\, r(q^k_{n+1}) + \delta(e^k_{n+1}) \tag{6.12}$$

Before we draw any conclusions from the formulas (6.11) and (6.12) we consider the terms $\delta(q^k_{n+1})$ and $\delta(e^k_{n+1})$. By means of (6.5) and (6.9) we find

$$\delta(q_{n+1}^k) = [(E_n^k - E_n^{k-1})(1+\epsilon_1) + Q_n^k](1+\epsilon_2) - (E_n^k - E_n^{k-1} + Q_n^k)$$

$$= (E_n^k - E_n^{k-1})(\epsilon_1 + [(E_n^k - E_n^{k-1})(1+\epsilon_1) + Q_n^k]\epsilon_2$$

(6.13)

$$\approx (E_n^k - E_n^{k-1})(\epsilon_1 + \epsilon_2) + Q_n^k \epsilon_2$$

$$\approx (e_n^k - e_n^{k-1})(\epsilon_1 + \epsilon_2) + q_n^k \epsilon_2 \quad .$$

(6.6) and (6.10) may be used to obtain

$$\delta(e_{n+1}^k) = [(Q_{n+1}^{k+1}/Q_{n+1}^k)(1+\epsilon_3) \times E_n^k] \times (1+\epsilon_4) - (Q_{n+1}^{k+1}/Q_{n+1}^k \times E_n^k)$$

$$= (Q_{n+1}^{k+1}/Q_{n+1}^k \times E_n^k)\,\epsilon_3 + [(Q_{n+1}^{k+1}/Q_{n+1}^k)(1+\epsilon_3) \times E_n^k]\,\epsilon_4$$

$$\approx (Q_{n+1}^{k+1}/Q_{n+1}^k \times E_n^k)(\epsilon_3 + \epsilon_4)$$

$$\approx e_{n+1}^k (\epsilon_3 + \epsilon_4) \quad .$$

From the limit theorems we know that $e_n^k \to o$ and $q_n^k \to z_k$ as $n \to \infty$. Hence

$$\delta(e_{n+1}^k) \approx o$$

and

$$\delta(q_{n+1}^k) \approx z_k\, \epsilon \ .$$

Furthermore (6.12) give

(6.15) $$r(e^k_{n+1}) \approx \frac{z_{k+1}}{z_k} r(e^k_n).$$

These results together with (6.11) show that although the error $r(q^k_{n+1})$ may not decrease with increasing n this error will not increase very rapidly.

Hence we may conclude:

The progressive form of the QD-algorithm is only "mildly" unstable.

6.2 The stability of the forward form of the QD-algorithm.

When the formulas (1.4) and (1.5) from the forward form of the algorithm are used instead of (6.7) and (6.8) we find the relations

(6.16) $$r(q^{k+1}_{n+1}) \approx \frac{e^k_{n+1}}{e^k_n} r(q^k_{n+1}) + \frac{q^k_{n+1}}{e^k_n} r(e^k_{n+1}) - \frac{e^k_{n+1} q^k_{n+1}}{(e^k_n)^2} r(e^k_n) + \delta(q^{k+1}_{n+1})$$

(6.17) $$r(e^{k+1}_n) \approx r(q^{k+1}_{n+1}) - r(q^{k+1}_n) + r(e^k_n) + \delta(e^{k+1}_n)$$

Since $e^k_n \to 0$ and $q^k_{n+1} \to z_k$ as $n \to \infty$ we may conclude from (6.16) that the forward form of the QD algorithm is "strongly" unstable.

Part 2: ALGOL procedures and numerical experiments

7. The procedure QDPOSITIVE

7.1 Introduction

The numerical experiments with the QD-algorithm were carried out on the Burroughs B 5000 computer at Stanford. The programs were written in

Extended ALGOL for the B 5000. The part of this language used in the programs is so close to the corresponding part of the ALGOL 60, that I have chosen to show the B 5000 procedures which have been used in practice instead of ALGOL 60 procedures. In fact, the only changes needed in the following B 5000 procedure QDPOSITIVE in order to have a correct ALGOL 60 procedure are:

1) The basic symbol← should be changed to := .

2) BEGIN, COMMENT etc. should be begin, comment etc.

3) The brackets following the array identifiers in the specification should be removed.

7.2 Description of the procedure

In order to avoid to many comments in the procedure a description of the parameters, the main features of the algorithm, the' storage; requirements etc.are given below:

1. Parameters

Input parameters:

N	the degree of the polynomial.
POLY	an array which holds the (N + 1) coefficients of $a_N x^N + \cdots + a_1 x + a_0$ with a_N in POLY[o], a_{N-1} in POLY[1] etc.
EPS	a real number specifying "the tolerance." cf. section 3 below.
MAX	an integer specifying the maximum numbers of rows of the QD scheme to be used.
JUMP	a label to which exit is made when the roots are not found by means of less than MAX rows.

Output parameters:

ROOTS — an array which upon exit holds the N roots of the polynomial equation.

ROWS — an integer which upon exit holds the number of rows used in the calculations.

2. Method

In the general case Q,DPOSITIVE computes N rows of the QD scheme. Then a translation from 0 to q_N^N is carried out, and N rows of the new QD scheme are computed etc., until $(N - 1)\, q_N^N < EPS$. Now the smallest root is computed, and the process is continued with (N - 1) rows until the next root is computed etc.

Before the QD schemes are computed the procedure checks if all the remaining roots are equal. This check is carried out by means of a very simple device which consists of a comparison of the arithmetic and the geometric mean of the remaining roots. When the roots are positive these means will be equal if and only if the remaining roots are equal.

3. Accuracy

The theory of the algorithm used (chapter 5) says that the maximum error should be less than or equal to the value of the parameter EPS. Since the progressive form of the QD-algorithm is mildly unstable and since the translations used will introduce other errors this will in general not be true. In numerical experiments with equations of degrees between 4 and 10 the first five digits have been correct in all cases (see the examples in 7.4).

4. Storage requirements

The procedure uses approximately $(N + 4) \times N$ cells for storing local variables.

7.3 QDPOSITIVE

```
PROCEDURE QDPOSITIVE(N,POLY,ROOTS,EPS,MAX,JUMP,ROWS);VALUE N,EPS,MAX;
   INTEGER N,MAX,ROWS;ARRAY POLY[0],ROOTS[1];REAL EPS;LABEL  JUMP;
   BEGIN INTEGER S,K,R,I,T;REAL  AM,GM,COR,CO;LABEL STOP,AGAIN;
      ARRAY Q[1:N,1:N],E,POL,POLI[0:N];
      FOR  S←0 STEP 1 UNTIL N DO POL[S]←POLI[S]←POLY[S] ;COR←CO←0;R←0;
      FOR S←N STEP -1 UNTIL 2 DO
      BEGIN AM←ABS(POL[1]/S);GM←ABS(POL[S]*(1/S));
        'IF ABS(AM-GM)<EPS THEN
         BEGIN FOR T←1 STEP 1 UNTIL S DO ROOTS[T]←AM+COR;
            GO TO STOP
         END;
   AGAIN:
         FOR I←1 STEP 1 UNTIL S-1 DO
         BEGIN Q[1,I]←0;E[I]←POL[I+1]/POL[I];
         END;
         R←R+1;
         Q[1,1]← -POL[1]/POL[0];Q[1,S]←E[0]←E[S]←0;
         FOR  T←2 STEP 1 UNTIL S DO
         BEGIN FOR I←1 STEP 1 UNTIL S DO
            Q[T,I]←E[I]-E[I-1]+Q[T-1,I];R←R+1;
            FOR I←1 STEP 1 UNTIL S-1 DO
            E[I]←Q[T,I+1]/Q[T,I]×E[I]
         END;
         IF (N-1)×Q[S,S]<EPS THEN
            BEGIN ROOTS[S]←AM+Q[S,S]+COR;
               IF S<N THEN AM←AM-ROOTS[S+1];
               FOR  I←S STEP -1 UNTIL 1 DO
```

```
        FOR T←1 STEP 1 UNTIL I DO
            POLI[T]←POLI[T]+AM ×POLI[T−1];
    FOR  I←1 STEP 1 UNTIL S−1 DO POL[I]←POLI[I];
    IF S=2 THEN ROOTS[1]←Q[2,1]+COR;
END ELSE
BEGIN COR←COR+Q[S,S];CO←Q[S,S];
    IF R≥ MAX THEN GO TO JUMP;
    FOR I←S STEP −1 UNTIL 1 DO
        FOR T←1 STEP 1 UNTIL I DO
            POL[T]←POL[T]+CO×POL[T−1];
    GO TO AGAIN
END;
END;
STOP: ROWS←R;
END QDPOSITIVE;
```

7.4 Examples

1. $p_4(x) = x^4 - 8x^3 + 24x^2 - 32x + 16$

Exact roots: 2, 2, 2, 2.

The following output was obtained:

Table 1

COEFFICIENTS:

1.00000000	-8.00000000	24.00000000	-32.00000000	16.00000000

EPS = 0.00000001 NUMBER OF ROWS = 0

ROOTS:

2.00000000	2.00000000	2.00000000	2.00000000

2. $p_4(x) = x^4 - 8x^3 + 23.98x^2 - 31.92x + 15.9201$

Exact roots: 2.1, 2.1, 1.9, 1.9.

The following output was obtained:

Table 2

COEFFICIENTS:

1.00000000	-8.00000000	23.98000000	-31.92000000	15.92010000

EPS = 0.00000001 NUMBER OF ROWS = 54

ROOTS:

2.09999999	2.09999999	1.90004137	1989995865

The details of the computation in example 2 are shown on the next pages where the 54 q-rows and the 54 e-rows are printed.

COEFFICIENTS:

1.00000000 -8.00000000 23.96000000 -31.92000000 15.92010000

Q1	E1	Q2	E2	Q3	E3	Q4
8.00000000		0.00000000		0.00000000		0.00000000
	-2.99750000		'1.33110926		-0.49875000	
5.00250000		1.66639074		0.83235926		0.49875000
	-0.99850200		'0.66488674		-0.29885120	
4.00399600		2.00000600		1.19839480		0.79760120
	-0.49875399		-0.39839721		-0.19890279	
3.50524401		2.10036278		1.39788922		0.99650399
	-0.29885632		-0.26515190		-0.14179051	
4.01398402		0.00000000		0.00000000		0.00000000
	-1.50026143		-0.66455367		-0.24838271	
2.51372260		0.83570776		0.41617096		0.24838271
	-0.49877426		-0.33093857		-0.14824189	
2.01494834		1.00354344		0.59886764		0.39662461
	-0~24841413		-0.19748861		-0.09817926	
1.76653421		1.05446896		0.69817699		0.49480387
	-0.14828186		-0.13075966		-0.06958046	
2.03476855		0.00000000		0.00000000		0.00000000
	-0.75320908		-0.33027707		-0.12225847	
1.28155947		0.42293200		0.20801860		0.12225847
	-0~24856921		-0.16244639		-0.07185479	
1.03299026		0.50905483		0.29861020		0.19411326
	-0112249424		-0.09529062		-0.04670962	
0.91049602		0.53625845		0.34719120		0.24082288
	-0.07214592		-0.06169423		-0.03239926	
1.07147703		0.00000000		0.00000000		0.00000000
	-0.38313806		-0.16117934		-0.05763440	
0.68833896		0.22195673		0.10354493		0.05763440
	-0.12354500		-0.07519102		-0.03208003	
0.56479397		0.27031270		0.14669593		0.08971443
	-0.05912914		-0.04079427		-0.01962445	
0.50566483		0.28864757		0.16782576		0.10933887
	-0103375236		-0.02371865		-0.01278537	
0.63412153		a.00000000		0.00000000		0.00000000
	-0.20625588		-0.07336461		-0.02386274	
0.42786565		0.13269127		0.04950187		0.02386274
	-0.06406124		-0.02732825		-0.01150321	
0.36380441		0.16962426		0.06532691		0.03536595
	-0.02986863		-0.01052485		-0.00622748	
0.33393578		0.18896805		0.06962428		0.04159343
	-0.01690210		-0.00387783		-0.00372029	
0.46774781'		0.00000000		0.00000000		0.00000000
	-0.13264735		-0.02769939		-0.00785516	
0.33510047		0.10494795		0.01984424		0.00785516
	-0.04154297		-0.00523758		-0.00310939	
0.29355750		0.14125334		0.02197243		0.01096455
	-0.01998955		-0.00081472		-0.00155163	
0.27356795		0.16042817		0.02123552		0.01251618
	-0.01172245		-0.00010784		-0.00091453	
0.41768310		0.00000000		0.00000000		0~00000000
	-0.10874797		-0.00831003		-0.00216342	
0.30893513		0.10043795		0.00614661		0.00216342'
	-0.03535507		-0.00050856		-0.00076146	
0.27338006		0.13528846		0.00589371		0.00292407

0.25609709	-0.01748297	0.15274527	-0.00002216	0.00553798	-0.00037789	0.00330276
0.40447206	-0~01042745	0.00000000	-0.00000080	0.00000000	'0.00022537	0.00000000
0.30224221	-0.10222985	0.10003036	-0.00219946	0.00164432	-0.00055516	0.00055516
0.26840812	-0.03383409	0.13382829	-0.0000~3616	0.00149304	-0.00018744	0.00074260
0.25153845	-0001686968	0.15069757	-0.00000040	0~00140021	-0.00009323	0.00063583
0.40112874	-0401010668	0.00000000	0.00000000	0.00000000	-0.00005565	0.00000000
0.30056477	-0010056397	0.10000198	-0.00056200	0.00042412	-0.00013788	0.00013788
0.26710577	'0~03345900	0.13345859	-0.00000238	0.00038168	-0.00004482	0.00018270
0.25038808	-0.01671769	0.15017627	-0.00000001	0.00036023	-0.00002145	0.00020415
0.40031213	-0.01002683	0.00000000	0.00000000	0.00000000	-0.00001216	0.00000000
0.30015610	-0.10015604	0.10000015	-0.00015588	0.00012783	-0.00002805	0.00002805
0.26678806	-0103336803	0.13336799	-0.00000020	0.00012187	-0.00000616	0.00003421
0.25010730	-0~01668076	0.15004674	0.00000000	0.00012014	-0.00000173	0.00003594
0.40016838	-0.01000741	0.00000000	0.00000000	0.00000000	-0.00000052	0.00000000
0.30008420	-0.10008418	0.10000004	-0.00008414	0.00008340	-0.00000074	0.00000074
0.26673215	-0003335205	0.13335202	-0.00000007	0.00008346	-0.00000001	0.00000074
0.25005788	-0.01667427	0.15002629	0.00000000	0.00008346	0.00000000	0.00000074
0.40016540	-0.01000400	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
0.30008271	-0.10008269	0.10000004	-0.00008265	0.00008265	0.00000000	0.00000000
0.26673099	-0.03335172	0.13335169	-0.00000007	0.00008272	0.00000000	0.00000000
0.25005686	-0.01667413	0.15002582	0.00000000	0.00008272	0.00000000	0.00000000
0.40016540	-0.01000393	0.60000000	0.00000000	0.00000000	0.00000000	0.00000000
0.30008271	-0010008269	0.10000004	-0.00008265	0.00008265	0.00000000	0.00000000
0.26673099	-0.03335171	0.13335168	-0.00000007	0.00008272	0.00000000	0.00000000
0.39991723	-0001667413	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
0.29993793	-0.09997930	0.09997930	0.00000000	0.00000000	0.00000000	0.00000000
0.26661150	-0103332643	0.13330573	0.00000000	o,oooooooo	0.00000000	0.00000000
	-0.01666321		0.00000000		0.00000000	

EPS = 0.00000001 NUMBER OF ROWS = 54

ROOTS:

2.09999999 2.09999999 1.90004137 1.89995865

3. $p_4(x) = x^4 - 8x^3 + 23.9999x^2 - 31.9996x + 15.9996$

Exact roots: 2.01, 2, 2, 1.99.

The following output was obtained:

Table 3

COEFFICIENTS:

1.00000000	-8.00000000	23.99990000	-31.99960000	15.99960000

EPS = 0.00000001 NUMBER OF ROWS = 72

ROOTS:

2.00996394	2.00087 089	1.99912488	1.99004029

4. $p_{10}(x) = x^{10} - 20x^9 + 171x^8 - 816x^7 + 2380x^6 - 4368x^5$
$+ 5005x^4 - 3432x^3 + 1287x^2 - 220x + 11.$

The following output was obtained:

Table 4

COEFFICIENTS:

1.00000000	-20.00000000	171.00000000	-816.00000000	2380.00000000
-4368.00000000	5005.00000000	-3432.00000000	1287.00000000	-220.00000000
11.00000000				

EPS = 0.00000001 NUMBER OF ROWS = 191

ROOTS:

3.91898807	3.68250232	3.30972557	2.83082807	2.28463026
1.71537022	1.16916998	0.69027853	0.31749293	0.08101405

The polynomial $p_{10}(x)$ is the characteristic polynomial corresponding to the matrix considered in example 8.1 in the next chapter. In all cases the first six figures are correct and all eight figures are correct in the three smallest roots.

8. Examples of computation of eigenvalues.

8.1 Introduction

The following two examples ought to be considered as illustrations of the QD-algorithm as a rootfinder, and not as examples of the QD-algorithm as a method for finding eigenvalues. The reason for this point of view is simply that the method used in the examples merely consist of a computation of the characteristic polynomial followed by the use of a QD-procedure similar to QDPOSITIVE. This does not mean that the QD-algorithm in general cannot be considered as a good method for finding eigenvalues, but it means that the starting row of the QD-scheme should be computed directly from the elements of the given matrix and not via the coefficients of the characteristic polynomial.

8.2 An example of the computation of the eigenvalues of a symmetric three-diagonal matrix.

The matrix used was the following 10 x10 matrix

$$A = \begin{vmatrix} 2 & -1 & & & & & & & & \\ 1 & 2 & -1 & & & & & \Large 0 & & \\ & -1 & 2 & -1 & & & & & & \\ & & -1 & 2 & -1 & & & & & \\ & & & -1 & 2 & -1 & & & & \\ & & & & -1 & 2 & -1 & & & \\ & & & & & -1 & 2 & -1 & & \\ & \Large 0 & & & & & -1 & 2 & -1 & \\ & & & & & & & -1 & 2 & -1 \\ & & & & & & & & -1 & 2 \end{vmatrix}$$

The eigenvalues of A are given by means of the formula,

$$(8.1) \qquad E_p = 2 \sin^2\left(\frac{p\pi}{2(N+1)}\right), \qquad p = 1, 2, \ldots . N$$

where N is the order of the matrix (N = 10).

The following output was obtained (the numbers in the column "CORRECT EV" were computed by means of (8.1))

THE CHARACTERISTIC POLYNOMIAL HAS THE COEFFICIENTS:

1.00000000@+00

-2.00000000@+01

1.7 1000000@+02

-8.16000000@+02

2.38000000@+03

-4.36800000@+03

5.00500000@+03

-3.43200000@+03

1.28700000@+03

-2.20000000@+02

1.10000000@+01

NUMBER OF ROWS = 138 EPS = 0.00000001

EIGEN-VALUE NR	EV COMP QD-ALGORITHM	CORRECT EV	ERROR×1000000
1	3.918986773@+00	3.918985945@+00	8.276@-01
2	3.682505627@+00	3.682507063@+00	-1.436@+00
3	3.309722197@+00	3.309721464@+00	7.333@-01
4	2.830829878@+00	2.83083 0022@+00	-1.434@-01
5	2.284629734@+00	2.284629673@+00	6.103@-02
6	1.71537-294@+00	1.715370320@+00	-2.593@-02
7	1.16916997~00	1. 169169972@+00	6.956@-03
8	6.902785321@-01	6.902785306@-01	1.432@-03
9	3.174929343@-01	3.174929336@-01	6.858@-04
10	8.101405277@-02	8.101405259@-02	1.835@-04

8.3 An example of the computation of the eigenvalues of a symmetric full matrix.

The matrix used was the following 4x4 matrix, which is used in Faddeev and Faddeeva [4] (p. 281)

$$A = \begin{vmatrix} 1.00 & 0.42 & 0.54 & 0.66 \\ 0.42 & 1.00 & 0.32 & 0.44 \\ 0.54 & 0.32 & 1.00 & 0.22 \\ 0.66 & 0.44 & 0.22 & 1.00 \end{vmatrix}$$

The characteristic polynomial of A is

$$\lambda^4 - 4\lambda^3 + 4.752\lambda^2 - 2.111856\lambda + 0.28615248$$

where the coefficientsare computed exactly.

Faddeev and Faddeeva give the following eigenvalues (computed within 5.10^{-9}):

$$\lambda_1 = 2.32274880$$

$$\lambda_2 = 0.79670669$$

$$\lambda_3 = 0.63828380$$

$$\lambda_4 = 0,24226071$$

The following output was obtained:

THE CHARACTERISTIC POLYNOMIAL HAS THE COEFFICIENTS:

1.00000000@+00

-4.00000000@+00

4.7 5200000@+00

-2.111856006@+00

2.861524800@-01

NUMBERS OF ROWS = 24 EPS = 0.00001000

EV NR	EV COMP BY QD
1	2.3227488000@+00
2	7.9670668890@-01
3	6.3828380280@-01
4	2.422607 083@-01

Literature on the QD-algorithm and related topics

[1] Aitken, A. C.: On Bernoulli's numerical solution of algebraic equations, Proc. Roy. Soc. Edinburgh 46, 289-305, (1926).

[2] Aitken, A. C.: Further numerical studies in algebraic equations and matrices, Proc. Roy. Soc. Edinburgh 51, 80-90, (1931).

[3] Engeli, M., Ginsburg, Th. Rutishauser, H. and Stiefel, E.: Refined Iterative methods for computation of the solution and the eigenvalues of self-adjoint boundary value problems, Mitteilung aus dem Inst. für angew. Math. an der eidgenoss. techn. Hochschule in Zurich. Nr8. (1959).

[4] Faddeev, D. K. and Faddeeva, V. N.: Computational methods of linear algebra, W. H. Freemann and Company, San Francisco and London (1963).

[5] Golomb, M.: Zeros and poles of functions defined by Taylor series, Bull. Amer. Math. Soc. 49, 581-592, (1943).

[6] Henrici, P.: The quotient-difference algorithm, NBS Appl. Math. Series No. 49, (1958).

[7] Henrici, P.: Elementary numerical analysis,Prepublication, Wiley (1962).

[8] Householder, A. S.: Principles of numerical analysis, McGraw-Hill Book Co. (1953).

[9] Ortega, J.: On the LL^T and QR methods for symmetric tridiagonal matrices (in preparation).

[10] Parlett, B.: The development and use of methods of LR type, AEC Research and development report NYO-10, 429, Courant Institute of Mathematical Sciences, New York University, New York (1963).

[11] Rutishauser, H.: Der quotienten-differenzen-algorithmus, Birkhauser Basel (1957).

[12] Rutishauser, H.: Der quotienten-differenzen-algorithmus, Z. angew. Math. Physik 5, 233-251, (1954).

[13] Rutishauser, H.: Anvendungen des QD-Algorithmus, Z. angew. Math. Physik 5, 496-508, (1954).

b-41 Rutishauser, H.: Bestimmung der Eigenwerte und Eigenvektorem einer Matrix mit Hilfe des QD-Algorithmus, Z. angew. Math. Physik 6, 387-401 (1955).

[15] Rutishauser, H.: Solution of Eigenvalue Problems with the LR-Transformation, NBS Appl. Math. Series No. 49, (1958).

[16] Rutishauser, H.: Stabile Sonderfalle des quotienten-differenzen-algorithmus, Num. Math. 5, 93-112 (1963).

[17] Stiefel, E. L.: Einfuhrung in die numerische Mathematik, Teubner, Basel (1961)

[18] Wilkinson, J. H.: Error analysis of floating-point computation, Num. Math, 2, 319-340 (1960).

[19] Wilkinson, J. H.: Calculation of eigenvalues (assured real) of a lower Hessenberg matrix by a variant of the LR technique (in preparation).

[20] Wynn, P.: A sufficient condition for the instability of the q-d algorithm, Num. Math. 1, 203-207 (1959).