Università di Pisa

Dipartimento di Informatica

Technical Report : TR-96-12

A Fast Iterative Method for Determining the Stability of a Polynomial

L. Gemignani

May 16, 1996

ADDR: Corso Italia 40,56125 Pisa,Italy. TEL: +39-50-887111. FAX: +39-50-887226

A fast iterative method for determining the stability of a polynomial

Luca Gemignani *

Dipartimento di Informatica, Università di Pisa
Corso Italia 40, 56125 Pisa, Italy.

ABSTRACT

We present an iterative numerical method for solving two classical stability problems for a polynomial $p(x)$ of degree n: the Routh-Hurwitz and the Schur-Cohn problems. This new method relies on the construction of a polynomial sequence $\{p^{(k)}(x)\}_{k\in\mathbf{N}}$, $p^{(0)}(x) = p(x)$, such that $p^{(k)}(x)$ quadratically converges to $(x-1)^p(x+1)^{n-p}$ whenever the starting polynomial $p(x)$ has p zeros with positive real parts and $n-p$ zeros with negative real parts. By combining some new results on structured matrices with the fast polynomial arithmetic, we prove that the coefficients of $p^{(k)}(x)$ can be computed starting from the coefficients of $p^{(k-1)}(x)$ at the computational cost of $O(n\log^2 n)$ arithmetical operations. Moreover, by means of numerical experiments, we show that the $O(n\log n)$ bit precision of computations suffices to support the stated computational properties. In this way, apart from a logarithmic factor, we arrive at the current best upper bound of $O(n^3\log^4 n)$ for the bit complexity of the mentioned stability problems.

1. INTRODUCTION

This paper deals with a fast iterative method for the solution of two classical stability problems for a polynomial $p(x) = \prod_{i=1}^{n}(x-x_i)$ of degree n. The first one, called the Routh-Hurwitz stability problem, is that of evaluating the number of its zeros with positive, zero and negative real parts. The second one, termed the Schur-Cohn stability problem, is that of computing the number of its zeros within, on and outside the unit circle. Numerical methods for solving these problems are important for the determination of the stability of both linear differential and difference equations (see [18] for general references).

Recently, many authors (see [9], [10], [20], [17], [19]) have proposed direct methods for the solution of both Routh-Hurwitz and Schur-Cohn

*Supported by 40% funds of the Progetto Analisi numerica e matematica computazionale of MURST and by G.N.I.M. of C.N.R..

problems which rely on the evaluation of the inertia of Hankel and Bezout matrices. In the view of the *Sylvester's law of inertia*, this computation can be reduced to determining the block triangular factorization of a $n \times n$ Hankel matrix H, i.e., $H = LDL^T$ where L is a lower triangular matrix with unit diagonal entries and D is a block diagonal matrix. In this way we may solve both the Routh-Hurwitz and the Schur-Cohn zero-location problems by using the fast algorithms for solving Hankel-like linear systems at the computational cost of $O(n^2)$ arithmetical operations.

In finite precision arithmetic fast Hankel system solvers generally suffer from numerical instabilities. Although stable modifications of fast algorithms for well-conditioned Hankel systems has been recently developed [8], [13], we note that Hankel matrices arising in location problems are usually ill-conditioned [22] and, therefore, direct methods either explicitly assume exact arithmetic or implicitly do so by not considering the effects of rounding errors. In this respect, by combining subresultant theory and factorization properties of Hankel matrices, in [11] the author derived a fast algorithm for the block triangular factorization of Hankel matrices which restricts the coefficient length growth to an almost linear rate (which is the minimal coefficient growth we can expect without GCD computations). Then, by assuming that the classical algorithm is used for multiplications over $\mathbf{Q}$, we can expect to compute the block triangular decomposition of an Hankel matrix in Boolean time $O(n^4 \log^2 n)$. At present, some faster algorithms performing the n-bit integer multiplication, $n > 1000$, at the Boolean cost of $O(n \log n \log(\log n))$ have been implemented as a part of a high performance multiprecision package [1], [2].

Differently, an iterative procedure for the solution of either of the above stability problems was developed and investigated by J. L. Howland [16] in connection with root-squaring iteration

$$S_{k+1} = \frac{S_k + S_k^{-1}}{2}, \quad S_0 = A, \tag{1}$$

used to calculate the sign matrix associated with a given matrix A. If we set

$$\det(S_k - xI) = p^{(k)}(x), \quad \det(S_0 - xI) = p(x), \tag{2}$$

then $p^{(k+1)}(x)$ is a polynomial whose zeros, expressed in terms of the zeros $x_i^{(k)}$ of $p^{(k)}(x)$, $x_i^{(0)} = x_i$, are just $(x_i^{(k)} + (x_i^{(k)})^{-1})/2$. In this way, the computation of $p^{(k+1)}(x)$ is equivalent to n independent applications of the square root iteration

$$x_i^{(k+1)} = \frac{x_i^{(k)} + (x_i^{(k)})^{-1}}{2} \tag{3}$$

which is known to be convergent to $+1$ from starting value in the open right half plane, and -1 from starting value in the open left half plane.

This means that the polynomial sequence $\{p^{(k)}(x)\}_{k\in\mathbf{N}}$ theoretically converges to $(x-1)^p(x+1)^{n-p}$ whenever the starting polynomial $p(x)$ has p zeros with positive real parts and $n-p$ zeros with negative real parts. Since the classical perturbation theorems for polynomial zeros depend on the absolute values of the coefficients [15] and, moreover, the coefficients of $(x+1)^n$ have length of order $O(n\log n)$, then it can easily be seen that a realistic computation of $\{p^{(k)}(x)\}_{k\in\mathbf{N}}$ requires at least the precision of $O(n\log n)$ bits.

Howland's algorithm adapts the resultant procedure, introduced by Bareiss [3], to the computations of the polynomials $p^{(k)}(x)$. In this way the evaluation of the coefficients of $p^{(k)}(x)$ in a suitable polynomial basis is reduced to calculating the determinant of a structured matrix, namely a resultant matrix. Then, a classical method of elimination may be applied to obtain the coefficients of $p^{(k)}(x)$ at the computational cost of $O(n^2)$ arithmetical operations or $O(n^4\log^2 n)$ Boolean ones.

In this paper, we present a numerical method for computing the sequence $\{p^{(k)}(x)\}_{k\in\mathbf{N}}$ which decreases the computational cost to $O(n\log^2 n)$ arithmetical operations or $O(n^3\log^4 n)$ Boolean ones.

Firstly, in section 2 we consider a slight modification of Howland's approach by showing that the computation of the polynomials $p^{(k)}(x)$ can be reduced to the evaluation of the determinant of a Bezout matrix $B(p^{(k-1)},q)$ generated by $p^{(k-1)}(x)$ and a suitable polynomial $q(x,y)=x^2-2yx+1$. The use of Bezout matrices, rather than resultant matrices, leads to better complexity estimates. In fact, in view of the results of [5] and [6], the Gaussian elimination method for the block triangular factorization applied to $B(p^{(k-1)},q)$ is equivalent in a polynomial setting to performing the Euclidean algorithm applied to $p^{(k-1)}(x)$ and to the polynomial $q(x,y)$ with respect to the variable x. In this way, we obtain the deteminant of the Bezout matrix generated by $p^{(k-1)}(x)$ and $q(x,y)$ in terms of certain coefficients of the polynomial remainder sequence generated by the Euclidean algorithm applied to these polynomials.

Secondly, in section 3 we describe our basic algorithms. We prove that the evaluation of the needed coefficients, which are polynomials in the variable y, at the Chebyshev points $\cos(\frac{(2k-1)\pi}{2(n+1)})$, $1\le k\le n+1$, can be performed by the fast Fourier transform method in $O(n\log n)$ arithmetical operations. Then, the interpolation problem of finding the coefficients of $p^{(k)}(x)$ given its values at Chebyshev points is solved in a stable way by means of Pan's algorithm [21] at the cost of $O(n\log^2 n)$ arithmetical operations. Finally, the resulting algorithm has been implemented in $Mathematica^{TM}$ with multiprecision arithmetic and its effectiveness has

been confirmed by means of numerical experiments which are reported and discussed.

2. THEORETICAL RESULTS

Let

$$u(x) = \sum_{i=0}^{n} u_i x^i, \ u_n = 1, \quad v(x) = \sum_{i=0}^{m} v_i x^i$$

be two polynomials of degree n and m, respectively, where $m < n$. The expression

$$\frac{u(x)v(y) - v(x)u(y)}{x - y} = \sum_{i,j=1}^{n} b_{i,j} x^{i-1} y^{j-1} \tag{1}$$

which is easily verified to be a bivariate polynomial in x and y, is called *Bezoutian* of $u(x)$ and $v(x)$. The Bezoutian is a rather special bilinear form which appears in the context of the theory of equations, in stability theory and in the classical elimination theory [18]. The $n \times n$ matrix $B(u,v) = (b_{i,j})$ is called the *Bezout matrix* or, more simply, the *Bezoutian* generated by $u(x)$ and $v(x)$. Hereafter, the $k \times k$ submatrix of $B(u,v)$ which has entries in the first k rows and columns of $B(u,v)$ will be referred to as the $k \times k$ leading principal submatrix of $B(u,v)$ and will be denoted $(B(u,v))_k$.

The following Gohberg-Semencul-like formula for the Bezout matrix $B(u,v)$ holds [14] :

$$B(u,v) = \begin{pmatrix} u_1 & u_2 & \dots & u_n \\ u_2 & \iddots & \iddots & \\ \vdots & \iddots & \bigcirc & \\ u_n & & & \end{pmatrix} \begin{pmatrix} v_0 & v_1 & v_2 & \dots \\ & v_0 & v_1 & \ddots \\ & \bigcirc & \ddots & \ddots \\ & & & v_0 \end{pmatrix} -$$

$$- \begin{pmatrix} v_1 & v_2 & \dots & v_n \\ v_2 & \iddots & \iddots & \\ \vdots & \iddots & \bigcirc & \\ v_n & & & \end{pmatrix} \begin{pmatrix} u_0 & u_1 & u_2 & \dots \\ & u_0 & u_1 & \ddots \\ & \bigcirc & \ddots & \ddots \\ & & & u_0 \end{pmatrix}, \tag{2}$$

where we assume $v_j = 0$ if $j > m$.

Let

$$F(u) = \begin{pmatrix} 0 & 1 & & \\ & \ddots & \ddots & \\ & & 0 & 1 \\ -u_0 & \dots & \dots & -u_{n-1} \end{pmatrix} \tag{3}$$

be the $n \times n$ Frobenius matrix associated with $u(x)$. The well known Barnett's decomposition [4] states that

$$B(u, v) = B(u, 1)v(F(u)), \tag{4}$$

where

$$B(u, 1) = \begin{pmatrix} u_1 & u_2 & \dots & u_n \\ u_2 & \iddots & \iddots & \\ \vdots & \iddots & O & \\ u_n & & & \end{pmatrix}. \tag{5}$$

Now let us consider the polynomial sequence $\{p^{(k)}(y)\}_{k \in \mathbf{N}}$ defined as in 1.2, namely,

$$S_{k+1} = \frac{S_k + S_k^{-1}}{2}, \quad S_0 = A,$$

$$\det(S_k - yI) = p^{(k)}(y), \quad \det(S_0 - yI) = p(y),$$

where A is any starting matrix with characteristic polynomial $u(x)$. Since the monic polynomial $p^{(k+1)}(y)$,

$$p^{(k+1)}(y) = \sum_{i=0}^{n} p_i^{(k+1)} y^i, \quad p_0^{(k+1)} \neq 0, \quad p_n^{(k+1)} = 1,$$

and the monic polynomial

$$\det\left(\frac{F(p^{(k)}) + F(p^{(k)})^{-1}}{2} - yI\right)$$

have the same roots with the same algebraic multiplicities, then we find that

$$p^{(k+1)}(y) = \det(\frac{F(p^{(k)}) + F(p^{(k)})^{-1}}{2} - yI). \tag{6}$$

In view of 2.4, there follows that

$$p^{(k+1)}(y) = \frac{\det(B(p^{(k)},1)^{-1}B(p^{(k)},x^2) + I - 2yB(p^{(k)},1)^{-1}B(p^{(k)},x))}{2^n \det(F(p^{(k)}))}$$

which implies

$$p^{(k+1)}(y) = \frac{\det(B(p^{(k)},1)^{-1})\det(B(p^{(k)},x^2) + B(p^{(k)},1) - 2yB(p^{(k)},x))}{2^n p_0^{(k)}}.$$

By means of 2.5, we finally obtain

$$p^{(k+1)}(y) = 2^{-n}(1/p_0^{(k)})(-1)^{n+\lceil n/2 \rceil}\det(B(p^{(k)},x^2 - 2yx + 1)). \tag{7}$$

This equality also follows from the well-known representation of the resultant matrix in terms of the Bezoutian one [14].

In order to compute the determinant of the Bezout matrix generated by $p^{(k)}(x)$ and $q(x,y) = x^2 - 2yx + 1$, we first observe that

$$\det B(p^{(k)},q) = \det JB(p^{(k)},q)J$$

where J is the permutation matrix having 1 in the antidiagonal, i.e., its (i,j) entry is 1 for $j = n-i-1$, $i,j = 1,\ldots,n$. Then, we recall the following result [6] which describes certain properties of the matrix $JB(u,v)J$ in terms of the remainder sequence generated by the Euclidean algorithm applied to $u(x)$ and $v(x)$, that is,

$$\begin{aligned} r_0(x) = u(x), \quad r_1(x) = v(x), \\ r_{i-1}(x) = r_i(x)q_i(x) - r_{i+1}(x), \\ \deg\ (r_{i+1}(x)) = \delta_{i+1} < \delta_i,\ i = 1,\ldots,L-1, \end{aligned}$$

where $-r_{i+1}(x)$ is the remainder of the division of $r_{i-1}(x)$ and $r_i(x)$ and $r_L(x)$ is the greatest common divisor (GCD) of $r_0(x)$ and $r_1(x)$.

PROPOSITION 2.1. *The nonsingular leading principal submatrices of* $JB(u,v)J$ *have sizes* $m_i = n - \deg\ (r_i(x))$, *that is,* $\det((JB(u,v)J)_k) \neq 0$ *if and only if there exists* i, $1 \leq i \leq L$, *such that* $k = n - \delta_i$. *Moreover, for the Schur complements* S_{m_i} *of the* $m_i \times m_i$ *leading principal submatrix* $(JB(u,v)J)_{m_i}$ *of* $JB(u,v)J$ *the following relation holds*

$$S_{m_i} = JB(r_i, r_{i+1})J.$$

Now, from relation 2.2 there follows the following description of the matrix $JB(p^{(k)}, q)J$,

$$JB(p^{(k)}, q)J = \begin{pmatrix} P^{(k)} & T^{(k)} \\ {T^{(k)}}^T & Z^{(k)} \end{pmatrix},$$

where $Z^{(k)} \in C[y]^{2\times 2}$, $P^{(k)} \in C[y]^{n-2\times n-2}$ and $P^{(k)}$ is lower triangular (with respect to the antidiagonal) having $p_n^{(k)} = 1$ along the antidiagonal. Let us consider the Euclidean Algorithm applied to $p^{(k)}(x)$ and $q(x,y)$ with respect to the variable x,

$$p^{(k)}(x) = q_1^{(k)}(x,y)q(x,y) - s^{(k)}(x,y), \tag{8}$$

$$q(x,y) = q_2^{(k)}(x,y)s^{(k)}(x,y) - w^{(k)}(y), \tag{9}$$

where $s^{(k)}(x,y) = \alpha^{(k)}(y)x + \beta^{(k)}(y)$. By using both the standard block decomposition

$$\begin{pmatrix} P^{(k)} & T^{(k)} \\ {T^{(k)}}^T & Z^{(k)} \end{pmatrix} = \begin{pmatrix} I & 0 \\ {T^{(k)}}^T {P^{(k)}}^{-1} & I \end{pmatrix} D^{(k)} \begin{pmatrix} I & {P^{(k)}}^{-1}T^{(k)} \\ 0 & I \end{pmatrix},$$

$$D^{(k)} = \begin{pmatrix} P^{(k)} & 0 \\ 0 & Z^{(k)} - {T^{(k)}}^T {P^{(k)}}^{-1} T^{(k)} \end{pmatrix},$$

and Proposition 2.1 we find that

$$\det(JB(p^{(k)}, q)J) = (-1)^{n-2}\det(JB(q, s^{(k)})).$$

Finally, a new application of Proposition 2.1 yields the following result.

PROPOSITION 2.2. *Let the functions* $\alpha^{(k)}(y)$, $\beta^{(k)}(y)$ *and* $w^{(k)}(y)$ *be defined by 2.8 and 2.9. Then we find that*

$$\det(JB(p^{(k)}, q)J) = (-1)^{n-2} \begin{cases} -{\beta^{(k)}(y)}^2 & \text{if } \alpha^{(k)}(y) = 0 \\ {\alpha^{(k)}(y)}^2 w^{(k)}(y) & \text{otherwise.} \end{cases}$$

In the next section we will present a fast algorithm for computing the coefficients of $p^{(k+1)}(x)$, which relies on the Proposition 2.2.

3. ALGORITHMS AND NUMERICAL EXPERIMENTS

In this section we describe an iterative algorithm for the solution of the Routh-Hurwitz problem for a n degree polynomial $p(x)$, that is, for determining the number of its zeros with positive, zero and negative real parts. Without loss of generality, hereafter we will assume that $n+1 = 2^h$ is an integer power of 2. Since the transformation

$$z = \frac{x+1}{x-1}$$

maps the left half-plane to the unit circle, then the solution of the Schur-Cohn problem for $p(x)$ is the solution of the Routh-Hurwitz problem for $p(z)(x-1)^n$.

The relations 2.8, 2.9 and Proposition 2.2 are the means of constructing the coefficients of the consecutive elements of the sequence $\{p^{(k)}(x)\}_{k\in\mathbf{N}}$ defined by 1.2. In fact, it should be noted that

$$q(x, y_k) = x^2 - 2y_k x + 1 = (x - e^{i\theta_k})(x - e^{-i\theta_k}),$$

where $y_k = \cos(\theta_k)$ and $\theta_k = ((2k-1)/(2(n+1)))\pi$, $k = 1, \ldots, n+1$. In this way, we evaluate the functions $\alpha^{(k)}(y)$, $\beta^{(k)}(y)$ and $w^{(k)}(y)$ at the Chebyshev set

$$\mathcal{C}_h = \{y_k^{(h)} = \cos(\frac{(2k-1)\pi}{2(n+1)}) = \cos(\frac{(2k-1)\pi}{2^{h+1}}),\ k = 1, \ldots, n+1\} \qquad (1)$$

at the cost of the evaluation of $p^{(k-1)}(x)$ at the Fourier points on the unit circle in the complex plane

$$\mathcal{F}_{h+1} = \{w_k = w^{k-1},\ w = e^{2i\pi/(4n+4)} = e^{2i\pi/(2^{h+2})},\ k = 1, \ldots, 4n+4\}.$$

Specifically, from 2.8 and 2.9 we find that

$$\begin{cases} \alpha^{(k)}(y_i) = \dfrac{p^{(k)}(w_{2i}) - p^{(k)}(w_{4n+6-2i})}{w_{4n+6-2i} - w_{2i}} \\ \beta^{(k)}(y_i) = -p^{(k)}(w_{2i}) - \alpha^{(k)}(y_i) w_{2i} \\ w^{(k)}(y_i) = (\beta^{(k)}(y_i)/\alpha^{(k)}(y_i))^2 + 2y_i^{(h+1)}(\beta^{(k)}(y_i)/\alpha^{(k)}(y_i)) + 1 \end{cases} \qquad (2)$$

for $i = 1, \ldots, n+1$.

In view of Proposition 2.2, in order to recover the coefficients of $p^{(k)}(x)$, we have now to solve the interpolation problem of finding the coefficients of a polynomial of degree n given its values at the set 3.1. Recently, V.

Pan [21] gave a computationally efficient solution of this problem. He observed that the substitution $y = 2x^2 - 1$ transforms two distinct points $x_i^{(h)}$ and $x_j^{(h)}$ of the set $\mathcal{C}_h$ into a point $y_s^{(h-1)}$ of the set $\mathcal{C}_{h-1}$ since we have $\cos(2\theta) = 2\cos^2(\theta) - 1$. In this way, if we write

$$\begin{aligned} p^{(k)}(x) &= p_0^{(k)}(x^2) + x p_1^{(k)}(x^2) = \\ &= p_0^{(k)}(\frac{1+y}{2}) + x p_1^{(k)}(\frac{1+y}{2}) = \\ &= t_0^{(k)}(y) + x t_1^{(k)}(y), \end{aligned}$$

then we obtain that

$$p^{(k)}(x_l^{(h)}) = t_0^{(k)}(y_s^{(h-1)}) + x_l^{(h)} t_1^{(k)}(y_s^{(h-1)})$$

for $l = i$ and $l = j$. This reduces the interpolation problem of finding the coefficients of a polynomial $p^{(k)}(x)$ of degree $n = 2^h - 1$, given its values at the set $\mathcal{C}_h$, to finding the coefficients of two polynomials $t_0^{(k)}(x)$ and $t_1^{(k)}(x)$ of degree $2^{h-1} - 1$, given their values at the set $\mathcal{C}_{h-1}$.

By using $Mathematica^{TM}$, we have implemented a function **InterChe** which applies the above reduction recursively returning the coefficients of $p^{(k)}(x)$ given its values at the Chebyshev points. At each step **InterChe** performs two calls of itself at half-size sets, and then three FFT and two IDFT for evaluating the coefficients of the shifted polynomials $p_i^{(k)}(\frac{1+y}{2}) = t_i^{(k)}(y)$, $i = 0, 1$ (compare [7]). Thus, if $S(n+1)$ denotes the number of arithmetical operations needed to compute the coefficients of $p^{(k)}(x)$ by means of the recursive function **InterChe**, we find that

$$S(n+1) = 2S(\frac{n+1}{2}) + 9n\log n + 4n,$$

which implies that $S(n+1)$ is less than $9n\log^2 n + 4n\log n$.

The resulting algorithm for the computation of the polynomial sequence $\{p^{(k)}(x)\}_{k\in\mathbf{N}}$ proceeds as follows:

Algorithm 1
Input: the coefficients of $p(x) = p^{(0)}(x)$;
Output: the coefficients of $p^{(m)}(x)$;
for k=0, ..., m-1,

- evaluate the functions $\alpha^{(k)}(y)$, $\beta^{(k)}(y)$ and $w^{(k)}(y)$ at the set $\mathcal{C}_h$ by means of 3.2;
- evaluate $p^{(k+1)}(x)$ at the set $\mathcal{C}_h$ by means of Proposition 2.2;
- call **InterChe** at the set $\{p^{(k+1)}(y_s^{(h)})\}$.

The correctness of the algorithm easily follows from relations 2.8, 2.9 and Proposition 2.2. Relations 3.2 involve the evaluation of a polynomial at the $4(n+1)$ roots of unity, which can be performed by means of an FFT of size $4(n+1)$ using only $O(n \log n)$ complex arithmetical operations. In the case where $p(x)$ is a real polynomial we need the real parts of a real FFT only. Using the decomposition of the Fourier matrix in terms of the sine and cosine matrices, we are able to compute these values by means of a real cosine transform of size $8n$. In this way, the computational cost of each step of Algorithm 1 is that of computing the recursive function **InterChe** at a set whose cardinality is $n+1$.

Although refined criteria based on the argument principle have been supplied for checking the convergence of the sequence $\{p^{(k)}(x)\}_{k \in \mathbf{N}}$, in our applications the very simple one

$$|\frac{p_{n-1}^{(k)}}{p_n^{(k)}} - \frac{p_{n-1}^{(k-1)}}{p_n^{(k-1)}}| < \epsilon \tag{6}$$

has been considered. When the program stops, the number p of zeros with positive real parts and the number q of zeros with negative real parts are evaluated by means of

$$p+q=n, \quad q-p = Round(\frac{p_{n-1}^{(k)}}{p_n^{(k)}}),$$

where $Round(x)$ gives the integer closest to x.

We have implemented Algorithm 1 complemented with the stop criterion 3.6 by using $Mathematica^{TM}$ with multiprecision arithmetic. Recently, D. H. Bailey [1], [2], has proposed a high performance multiprecision Fortran package implementing the fast polynomial arithmetic. Probably, a multiprecision Fortran translation of our algorithm will decrease the computation times with respect to our implementation.

Firstly, we have performed 20 numerical experiments with polynomials of degree 31 with random complex coefficients lying in the rectangle $[0,1] \times [0,1]$ in the complex plane. We have set the precision of the computation, that is, the number of decimal digits which are treated as significant for computations, equal to 40. Table 1 below reports the maximum number *maxiter* of iterations, the minimum number *miniter* of iterations and the average number, rounded to the nearest integer, *averiter* of iterations performed during the computations. Concerning the stop criterium, we assume a threshold value $\epsilon = 10^{-4}$.

Table 1			
n	$miniter$	$maxiter$	$averiter$
31	8	16	11

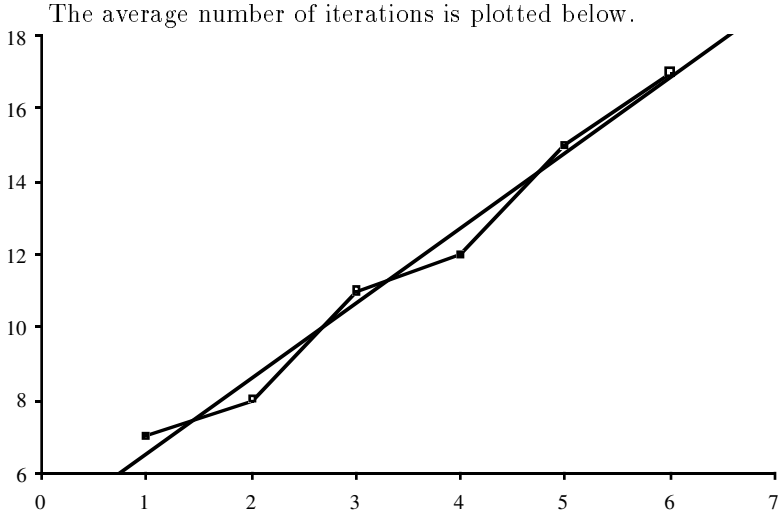
Secondly, in order to investigate better the convergence properties of the polynomial sequence generated by the Algorithm 1, we have generated 60 monic polynomials of degree 63 belonging to 6 different sets A_i, $i = 1, \ldots, 6$, such that

$$u(x) = \prod_{i=1}^{63}(x - z_i) \in A_j \Rightarrow \ \forall i \ \frac{1}{4^j} \leq Re[z_i] \leq \frac{1}{4^{j-1}}, \ 0 \leq Im[z_i] \leq 1.$$

We have set the precision of the computation equal to 90 and $\epsilon = 10^{-4}$. Table 2 below reports $miniter$, $maxiter$ and $averiter$, for each set A_j, $j = 1, \ldots, 6$.

Table 2			
j	$miniter$	$maxiter$	$averiter$
1	6	7	7
2	8	9	8
3	10	11	11
4	12	13	12
5	14	15	15
6	16	17	17

The average number of iterations is plotted below.



The reported linear least-squares fit to data is $y = 4.4667 + 2.0571x$.

Clearly, these results agree with the convergence properties of the square root iteration, which are theoretically proved. In particular, they show how the global quadratic convergence right from the start of the square root iteration depends on the geometry of the zeros. In fact, the rate of convergence of the square root iteration 1.3 is entirely dependent on the moduli of

$$\frac{x_i^{(0)} + 1}{x_i^{(0)} - 1} = \frac{x_i + 1}{x_i - 1}, \quad 1 \leq i \leq n,$$

and then the slowest convergence may be expected when a comparatively large value of $|Im[x_i]|$ is associated with a small value of $|Re[x_i]|$ [16].

Thus, when $p(x)$ admits zeros lying near or on the imaginary axis, we can expect an almost arbitrarily slow convergence. Under these circumstances, the stop criterion 3.6 will never be fulfilled. At the same time, Algorithm 1 may be complemented with suitable techniques in order to extract the factor $(x-1)^p(x+1)^q$ whenever the starting polynomial $p(x)$ has p zeros with positive real parts and q zeros with negative real parts. In this respect, we have generated 10 monic polynomials of degree 7, with 6 random zeros lying in the rectangle $[0,1] \times [0,1]$ in the complex plane and 1 random zero lying on the imaginary axis in the interval $[-1,1]$. We terminate Algorithm 1 when either the condition 3.6 where $\epsilon = 10^{-4}$ or the condition $k > 7$ is satisfied; then we compute the polynomial remainder sequence generated by the Euclidean algorithm applied to $(x-1)^7$ and $p^{(8)}(x)$. We have run the combined algorithm by using machine precision, i.e., about 18 decimal digits. In all the experiments the maximum of the absolute values of the differences between the coefficients of $(x-1)^6$ and the coefficients of the first polynomial remainder is less than 10^{-14}.

REFERENCES

1 D.H. Bailey, *A Portable High Performance Multiprecision Package*, RNR Technical Report RNR-90-022, NASA Ames Research Center, Moffet Fields, CA 1992.

2 D.H. Bailey, *Multiprecision Translation and Execution of Fortran Programs*, ACM Transactions on Mathematical Software, 1992.

3 E. H. Bareiss, *The Numerical Solution of Polynomial Equation and the Resultant Procedure*, Mathematical Methods for Digital Computers, Vol. II, pp. 185-214, 1967. Edited by A. Ralston and H. S. Wilf, John Wiley & Sons Inc., New York, 1967.

4 S. Barnett, P. Lancaster, *Some properties of the Bezoutian for polynomial matrices*, Linear and Multilinear Algebra, 9, pp. 99-111, 1980.

5 D. Bini, L.Gemignani, *On the Euclidean scheme for polynomials having interlaced real zeros*, Proc. 2nd Ann. ACM Symp. on Parallel Algorithms and Architectures, pp. 254-258, ACM Press, 1990.

6 D. Bini, L. Gemignani, *Fast parallel computation of polynomial remainder sequence via Bezout and Hankel matrices* , SIAM J. Comput., **24**, pp. 63-77, 1995.

7 D. Bini, V. Pan, *Numeric and algebraic Computations with Matrices and Polynomials: Fundamental Algorithms*, Volume 1, Birkhäuser, Boston, MA, 1994.

8 R. W. Freund, H. Zha, *A look-ahead algorithm for the solution of general Hankel systems*, Numer. Math., **64**, pp. 295-321, 1993.

9 L. Gemignani, *Computing the Inertia of Bezout and Hankel Matrices*, CALCOLO, **28**, pp. 267-274, 1991.

10 L. Gemignani, *Computationally Efficient Applications of the Euclidean Algorithm to Zero Location*, Tech. Report, Department of Mathematics, University of Parma, 1994. To appear on Linear Algebra Appl..

11 L. Gemignani, *Solving Hankel Systems over the Integers*, J. Symbolic Computation, **18**, pp. 573-584, 1994.

12 W. B. Gragg, A. Lindquist, *On the partial realization problem*, Linear Algebra Appl., **50**, pp. 277-319, 1983.

13 I. Gohberg, T. Kailath, V. Olshevsky, *Fast Gaussian elimination with partial pivoting for matrices with displacement structure*, Math. Comp. **64**, pp. 1557-1576, 1995.

14 G. Heinig, K. Rost, *Algebraic Methods for Toeplitz-like Matrices and Operators*, Birkhäuser, Boston, MA, 1984.

15 A.S. Householder, *The Numerical Treatment of a Single Nonlinear Equation*, McGraw-Hill, Boston, 1970.

16 J. L. Howland, *The Resultant Iteration for Determining the Stability of a Polynomial*, Math. Computation, **143**, pp. 779-789, 1978.

17 T. Kailath, A. H. Sayed, *Displacement Structure: Theory and Applications*, SIAM Review, **37**, pp. 297-386, 1985.

18 M. G. Krein, M. A. Naimark, *The Method of Symmetric and Hermitian Forms in the Theory of the Separation of the Roots of Algebraic Equations*, Linear and Multilinear Algebra, **10**, pp. 265-308, 1981.

19 H. Lev-Ari, Y. Bistritz, T. Kailath, *Generalized Bezoutians and families of efficient zero-location procedures*, IEEE Transactions on Circuits and Systems, **38**, pp. 170-186, 1991.

20 D. Pal, T. Kailath, *Fast Triangular Factorization of Hankel and related Matrices with Arbitrary Rank Profile*, SIAM J. Matrix Anal. Appl., **15**, pp. 451-478, 1994.

21 V. Pan, *Fast Evaluation and Interpolation at the Chebyshev Sets of Points*, Appl. Math. Lett., **2**, pp. 255-258, 1989.

22 E. E. Tyrtyshnikov, *How bad are Hankel matrices?*, Numer. Math., **67**, pp. 261-269, 1994.