

JAERI-M

7 3 3 5

高次代数方程式の数値解法プログラム
(SSLの拡充とベンチマーク・テスト No.1)

1977年10月

朝 岡 卓 見

この報告書は、日本原子力研究所が JAERI-M レポートとして、不定期に刊行している研究報告書です。入手、複製などのお問い合わせは、日本原子力研究所技術情報部（茨城県那珂郡東海村）あて、お申しこしてください。

JAERI-M reports, issued irregularly, describe the results of research works carried out in JAERI. Inquiries about the availability of reports and their reproduction should be addressed to Division of Technical Information, Japan Atomic Energy Research Institute, Tokai-mura, Naka-gun, Ibaraki-ken, Japan.

高次代数方程式の数値解法プログラム
(SSLの拡充とベンチマーク・テスト No 1)

日本原子力研究所東海研究所原子炉工学部
朝岡卓見

(1977年9月22日受理)

高次代数方程式の主な数値解法アルゴリズムを概観し、代表的な計算プログラムを整備し、既存のルーチンも含めてベンチマーク・テストを実施した。

逆補間法のルーチンとしては、Muller法のプログラムを整備すると共に、これにChambersのアルゴリズムを取り入れたものも作成した。このMuller-Chambers法のルーチンは、3重根、3つの近接根などを除けば、特に複素係数多項式の根の計算に有用である。Newton法の変形であるMadsenアルゴリズムによるルーチンも整備したが、低次多項式の根の算出には他より時間がかかるが、すべての場合に正確な解を与えており、標準的な計算プログラムとして用いることができる。実係数多項式に対する既存のBairstow法ルーチンは、3重根などを除けば最も速いアルゴリズムになっていることも示された。なお求められた根の誤差限界の計算ルーチンの整備も行っている。

Computer Programs for Nonlinear Algebraic Equations

(Development and Benchmark Test of SSL, No.1)

Takumi ASAOKA

Division of Reactor Engineering, Tokai Research Establishment, JAERI

(Received September 22, 1977)

We have provided principal computer subroutines for obtaining numerical solutions of nonlinear algebraic equations through a review of the various methods. Benchmark tests were performed on these subroutines to grasp the characteristics of them compared to the existing subroutines.

As computer programs based on the secant method, subroutines of the Muller's method using the Chambers' algorithm were newly developed, in addition to the equipment of subroutines of the Muller's method itself. The programs based on the Muller-Chambers' method are useful especially for low-order polynomials with complex coefficients except for the case of finding the triple roots, three close roots etc. In addition, we have equipped subroutines based on the Madsen's algorithm, a variant of the Newton's method. The subroutines have revealed themselves very useful as standard programs because all the roots are found accurately for every case though they take longer computing time than other subroutines for low-order polynomials. It is shown also that an existing subroutine of the Bairstow's method gives the fastest algorithm for polynomials with complex coefficients, except for the case of finding the triple roots etc. We have provided also subroutines to estimate error bounds for all the roots produced with the various algorithms.

Keywords: Nonlinear Algebraic Equation, Polynomial, SSL (Scientific Subroutine Library), Numerical Solution, Benchmark Test, Muller's Method, Chambers' Algorithm, Newton's Method, Madsen's Algorithm, Bairstow's Method, Jarrat's Method, Error Bound.

総 論

ここ数年来、数値解析理論や新しいアルゴリズムばかりでなく、SSL (Scientific Subroutine Library, 科学用サブルーチン・ライブラリー) や、特定の問題により深く関係した数値解析プログラム (例えば、Comp. Phys. Comm.) の発表件数は、膨大なものがある。従ってこれらのルーチンを隈なくサーベイし、且つ整備しておくことは不可能であるのは勿論、効率的利用という立場から見ても、労多くして、功少なしの感がある。そこで実用上は、各ルーチンの特徴の分析やベンチマークテスト、stiffな問題への使用経験など集約された情報が要求され、これらが充分蓄積された時に、ルーチン相互の位置づけや体系化が可能となる。

我々の研究室では、数値解析の広い分野にわたる、これまでのアルゴリズム調査を土台としてベンチマークテストなどの情報集約と評価を行った。対象としたのは、主に最近発表されたプログラムとアルゴリズムで、先に行ったアルゴリズム調査の延長という形でまとめたので御利用いただきたい。

(西田雄彦, 藤村統一郎)

目 次

1. 序 論	1
2. 逆補間法ルーチン	1
3. Newton-Raphson法ルーチン	6
4. Bairstow法及びJarrat法ルーチン	16
5. 数値テストと検討	18
6. 結 論	46
参考文献	48
付録1 MULLRAルーチンとテスト用メイン・プログラム, 誤差限界 評価用PA07CDルーチンのFortranリスト	49
付録2 MULLRBルーチンとテスト用メイン・プログラム, 誤差限界 評価用PA06CDルーチンのFortranリスト	54
付録3 MUACHMルーチンのFortranプログラム・リスト	59
付録4 MUBCHMルーチンのFortranプログラム・リスト	62
付録5 PA07ADルーチンとテスト用メイン・プログラムのFortran リスト	65
付録6 PA06ADルーチンとテスト用メイン・プログラムのFortran リスト	77

CONTENTS

1. Introduction	1
2. Subroutines of Secant Method	1
3. Subroutines of Newton-Raphson's Method	6
4. Subroutines of Bairstow's and Jarrat's Methods	16
5. Numerical Tests and Discussions	18
6. Conclusions	46
References	48
Appendix 1. Fortran Lists of MULLRA Subroutine, its Test Program and PA07CD Subroutine for Error- Bounds	49
Appendix 2. Fortran Lists of MULLRB Subroutine, its Test Program and PA06CD Subroutine for Error- Bounds	54
Appendix 3. Fortran Program List of MUACHM Subroutine ...	59
Appendix 4. Fortran Program List of MUBCHM Subroutine ...	62
Appendix 5. Fortran Lists of PA07AD Subroutine and its Test Program	65
Appendix 6. Fortran Lists of PA06AD Subroutine and its Test Program	77

1. 序 論

非線形代数方程式の数値解法としては、まず1元2次代数方程式の根はよく知られている公式により簡単に求められる。3次方程式には2次項を消去するCardano法、4次方程式には3次項を消去し平方の差の形にするFerrari法が知られている。

これら低次代数方程式も以下に示す反復法で取扱う方が便利なが多いが、5次以上になると一般に代数的方法で解を求めることは不可能で、反復法がもっぱら用いられる。複素変数 $z = x + iy$ に対する $f(z)$ の根に接近する値を次々に計算する反復法では、 $f(z) = 0$ を変形して $z = F(z)$ とし、

$$z_{i+1} = F(z_i), \quad i = 0, 1, 2, \dots \quad (1)$$

の形の反復計算を行う。一つの根を α とすれば、(1)式より

$$z_{i+1} - \alpha = F'(\alpha)(z_i - \alpha) + \frac{1}{2}F''(\alpha)(z_i - \alpha)^2 + \dots \quad (2)$$

これより、 $|F'(\alpha)| < 1$ ならば反復計算は α に収束することが分る。 $|F'(\alpha)| > 1$ のときは収束しないので、 $F(z)$ の逆関数を使う $z = F^{-1}(z)$ の形の反復法を使わなければならない。反復法の収束の速さは、 $\lim_{i \rightarrow \infty} z_i = \alpha$ のとき、

$$\lim_{i \rightarrow \infty} (z_{i+1} - \alpha) / (z_i - \alpha)^\lambda = c \neq 0 \quad (3)$$

となる λ の値で表わし、収束の速さは λ 位 (λ -th order) であるという。 λ 位であるための必要十分条件は

$$\alpha = F(\alpha), \quad F^{(i)}(\alpha) = 0, \quad i = 1 \sim \lambda - 1; \quad F^{(\lambda)}(\alpha) \neq 0 \quad (4)$$

である。この場合の反復は λ 位反復といわれている。

まず次章で、古くから知られている1位反復法の「はさみうち法」を中心に逆補間法を概観し、2次のLagrange補間を用いたMuller法ルーチンと、これにChambersのアルゴリズムを取り入れて収束の速さを高めた新しいルーチンについて述べる。第3章で、2位反復法のNewton-Raphson法の変形で多項式のすべての根を求めるための標準的手法といわれているJenkins-Traub法を概観し、これと同様だが計算時間の速いMadsenアルゴリズムによる最近の計算ルーチンについて述べる。第4章でBairstow法及びJarrat法のFACOMのSSLに組み込まれているルーチンに触れ、第5章で実係数多項式に対する五つのルーチン、複素係数多項式に対する六つのルーチンのベンチマーク・テストの結果の検討を行う。

2. 逆補間法ルーチン

反復法で最も単純なのは線形逆補間法に属する「はさみうち法」(secant法、あるいはregula falsi)である。根 α をはさむ2個の近以値 x_p, x_q に対して $f(x_p)f(x_q) < 0$ であ

1. 序 論

非線形代数方程式の数値解法としては、まず1元2次代数方程式の根はよく知られている公式により簡単に求められる。3次方程式には2次項を消去するCardano法、4次方程式には3次項を消去し平方の差の形にするFerrari法が知られている。

これら低次代数方程式も以下に示す反復法で取扱う方が便利なが多いが、5次以上になると一般に代数的方法で解を求めることは不可能で、反復法がもっぱら用いられる。複素変数 $z = x + iy$ に対する $f(z)$ の根に接近する値を次々に計算する反復法では、 $f(z) = 0$ を変形して $z = F(z)$ とし、

$$z_{i+1} = F(z_i), \quad i = 0, 1, 2, \dots \quad (1)$$

の形の反復計算を行う。一つの根を α とすれば、(1)式より

$$z_{i+1} - \alpha = F'(\alpha)(z_i - \alpha) + \frac{1}{2}F''(\alpha)(z_i - \alpha)^2 + \dots \quad (2)$$

これより、 $|F'(\alpha)| < 1$ ならば反復計算は α に収束することが分る。 $|F'(\alpha)| > 1$ のときは収束しないので、 $F(z)$ の逆関数を使う $z = F^{-1}(z)$ の形の反復法を使わなければならない。反復法の収束の速さは、 $\lim_{i \rightarrow \infty} z_i = \alpha$ のとき、

$$\lim_{i \rightarrow \infty} (z_{i+1} - \alpha) / (z_i - \alpha)^\lambda = c \neq 0 \quad (3)$$

となる λ の値で表わし、収束の速さは λ 位 (λ -th order) であるという。 λ 位であるための必要十分条件は

$$\alpha = F(\alpha), \quad F^{(i)}(\alpha) = 0, \quad i = 1 \sim \lambda - 1; \quad F^{(\lambda)}(\alpha) \neq 0 \quad (4)$$

である。この場合の反復は λ 位反復といわれている。

まず次章で、古くから知られている1位反復法の「はさみうち法」を中心に逆補間法を概観し、2次のLagrange補間を用いたMuller法ルーチンと、これにChambersのアルゴリズムを取り入れて収束の速さを高めた新しいルーチンについて述べる。第3章で、2位反復法のNewton-Raphson法の変形で多項式のすべての根を求めるための標準的手法といわれているJenkins-Traub法を概観し、これと同様だが計算時間の速いMadsenアルゴリズムによる最近の計算ルーチンについて述べる。第4章でBairstow法及びJarrat法のFACOMのSSLに組み込まれているルーチンに触れ、第5章で実係数多項式に対する五つのルーチン、複素係数多項式に対する六つのルーチンのベンチマーク・テストの結果の検討を行う。

2. 逆補間法ルーチン

反復法で最も単純なのは線形逆補間法に属する「はさみうち法」(secant法、あるいはregula falsi)である。根 α をはさむ2個の近以値 x_p, x_q に対して $f(x_p)f(x_q) < 0$ であ

るとすると,

$$F(x) = [x_q f(x) - x f(x_q)] / [f(x) - f(x_q)] \quad (5)$$

を使い, 新しい近似値を

$$\bar{x} = F(x_p) \quad (6)$$

によって求めていく。ξ を α と x_q の中間の値とすると

$$f(x_q) = f(\alpha) + f'(\alpha)(x_q - \alpha) + \frac{1}{2} f''(\xi)(x_q - \alpha)^2 \quad (7)$$

より

$$F'(\alpha) = \frac{1}{2} (x_q - \alpha)^2 f''(\xi) / f(x_q) \quad (8)$$

これより x_q が α に近いならば |F'(α)| < 1 を満足させることが出来るので, 出発値が適当ならば 1 位の速さで収束することが分る。

より正確に言えば,¹⁾ f'(α) ≠ 0, f''(α) ≠ 0 ならば,

$$\lim_{i \rightarrow \infty} |x_i - \alpha| / [|x_{i-1} - \alpha| |x_{i-2} - \alpha|] = |f''(\alpha) / f'(\alpha)| / 2 \quad (9)$$

で, これより

$$\lim_{i \rightarrow \infty} |x_i - \alpha| / |x_{i-1} - \alpha|^\lambda = [|f''(\alpha) / f'(\alpha)| / 2]^{\lambda-1}, \quad (10)$$

$$\lambda = (1 + \sqrt{5}) / 2 \approx 1.62$$

後述の Newton 法は 2 位の速さで収束するが, 各反復ステップで f と f' を計算しなければならぬ。これに対し, この方法では f だけでよいので簡単である。勿論重根や複素根には適用出来ない。

はさみうち法は線形補間をしているが, これに対し, 近似値 x_i ~ x_{i-k} で f(x) と一致する k (> 1) 次多項式を作り, この多項式の 0 点の一つとして x_{i+1} を求めるのが, (Lagrange の補間公式を用いた) Muller 法である。従ってはさみうち法より収束は速いが複雑である。この Muller 法により多項式の根を求める Fortran プログラムは Lawrence²⁾ により与えられている。一般に複素数を考え, 根の近似値 z_i, z_{i-1}, z_{i-2} より

$$z_{i+1} = F(z_i, z_{i-1}, z_{i-2}) \quad (11)$$

の反復を用い, 2 次の Lagrange 多項式で補間している。実係数と複素係数多項式に対するプログラム・リストがそれぞれ示されているので, 我々はそれぞれを MULLRA, MULLRB と名付けて倍精度のルーチンとして整備した。

これら Muller 法のルーチンの計算の流れを Fig. 1 に示したが, その主要部分は⑭のブロックで, そこで⑪式の反復を行っている。すなわち多項式

$$f(z) = \sum_{j=0}^n a_j z^{n-j} \quad (a_0 \neq 0) \quad (12)$$

の根を求める場合,

$$c_0 z_j^2 + c_1 z_j + c_2 = f(z_j), \quad j = i-2, i-1, i, \quad (13)$$

を満足するように c₀, c₁, c₂ を決め, これによる Lagrange 多項式

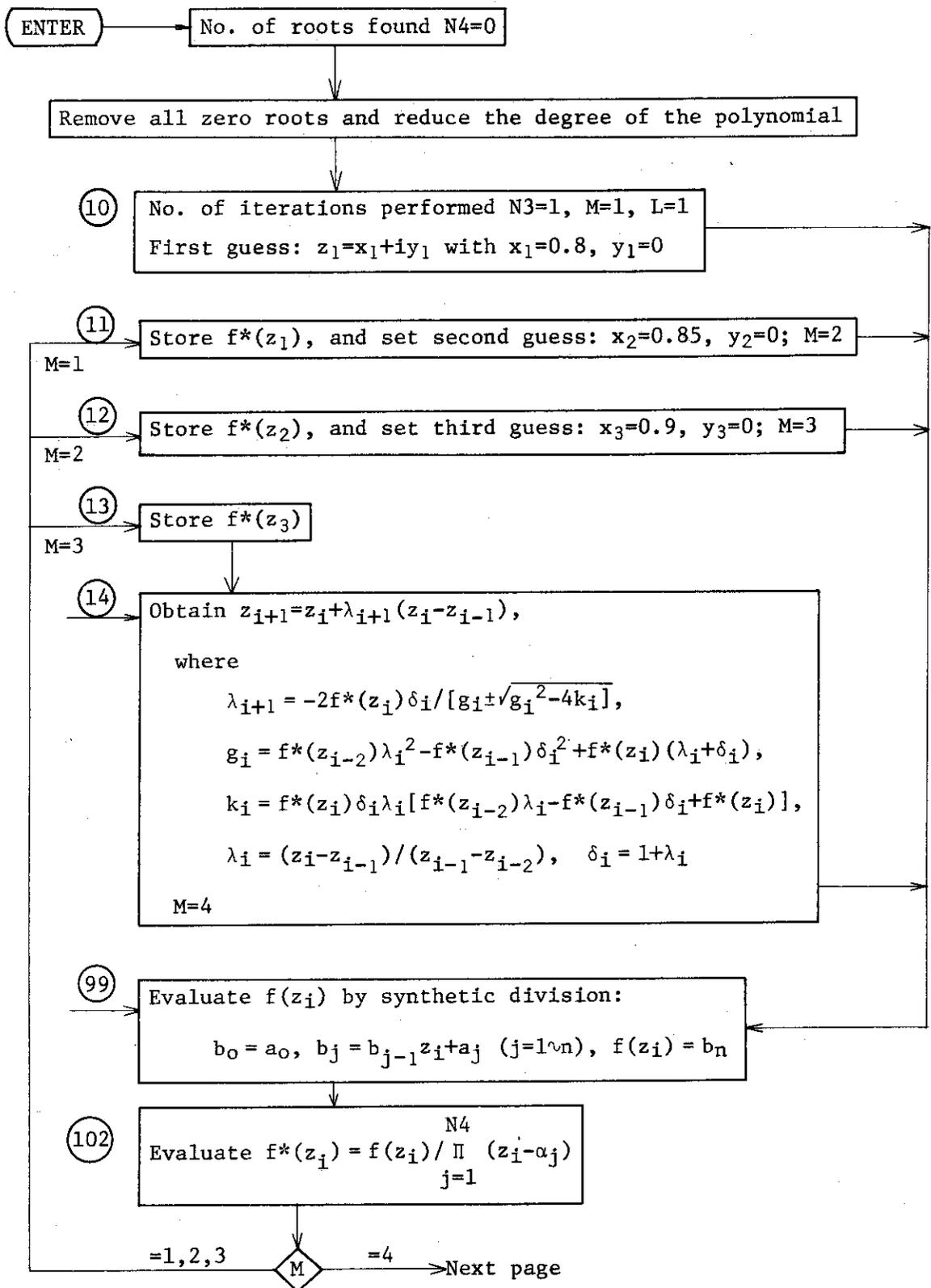


Fig. 1 Flow diagram of MULLRA or MULLRB to find all the roots

$$\alpha_j \text{'s of } f(z) = \sum_{j=0}^n a_j z^{n-j}$$

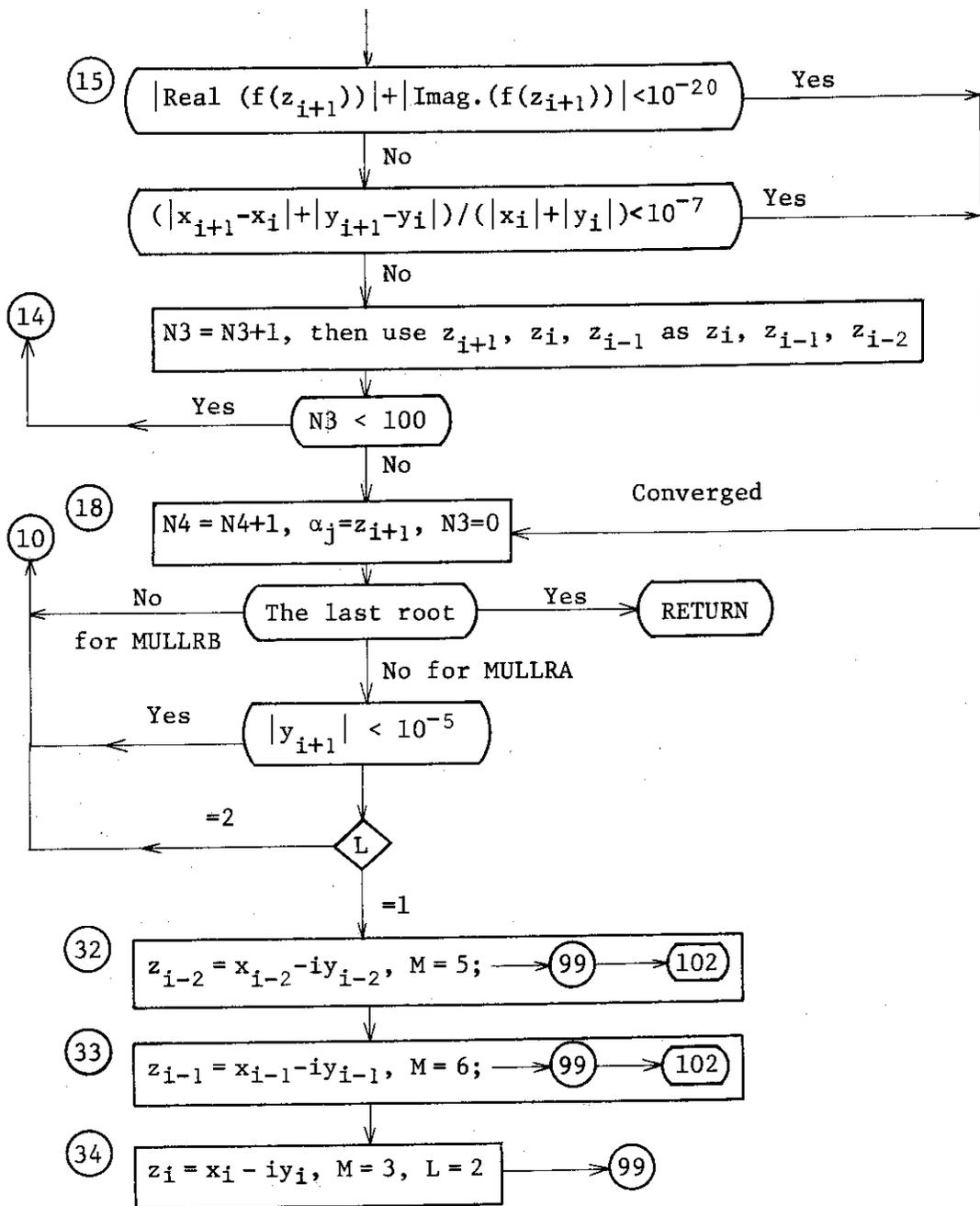


Fig. 1 (Continued)

$$L_i \{f(z)\} = c_0 z^2 + c_1 z + c_2 \quad (14)$$

の0として、次の近似値 z_{i+1} を得るのである。なお(12)式の $f(z)$ は、すでに求められた根がある場合には、Fig. 1の(102)のようにしてその根を取り去った後の(deflated)多項式となる。収束は(9)で、関数値の実数部と虚数部の絶対値の和が 10^{-20} 以下になるか、あるいは根の近似値の相対誤差が 10^{-7} 以下になるかで判定している。ただし100回反復しても収束しない際にも収束したかのように扱われるので注意を要する。なお実係数多項式に対するMULLRAルーチンでは、 $x+iy$ の根が求められ、 $y > 10^{-5}$ のときには、続けて $x-iy$ を探すようになっている。

これらのルーチンの呼び出しは以下による。

CALL MULLRA(A, N, RR, RI, N1)

CALL MULLRB(AR, AI, N, RR, RI, N1)

A, AR, AIはそれぞれ大きさN1の倍精度1次元配列で、それぞれ(12)式の実係数 a_j 、あるいは複素係数 a_j の実数部、及び虚数部が $j = 0, 1, \dots, N$ の順番に与えられる。Nは多項式の次数 n で、N1は $(n+1)$ である。そしてRR, RIがそれぞれ大きさNの倍精度1次元配列で、求められた根の実数部と虚数部がストアされる。

Chambers³⁾は(11)式のMuller法の収束の位数が1.84であるのに対し、

$$z_{i+1} = F(z_i^*, z_i, z_{i-1}) \quad (15)$$

の収束を考えている。 z_i^* は z_i と z_{i-1} の関数であるが、

$$z_i^* = g z_i + (1-g) z_{i-1} \quad (g \neq 1) \quad (16)$$

とした時、収束は2位となり、これで $g=1$ ととれば位数は2.414と増加している。又、

$$z_i^* = [z_{i-1} f(z_i) - z_i f(z_{i-1})] / [f(z_i) - f(z_{i-1})] \quad (17)$$

とregula falsi流にとると、位数は更に2.732と増えている。

従って我々は上述のMULLRA, MULLRBを改造し、(15), (17)式によるルーチンも作成した。これらは実係数、複素係数に対し、それぞれMUACHM, MUBCHMと名付けたが、Fig. 1で示されている計算の流れで変更されたのは以下の点である。まず(9)のブロックでは、以下のよう
に二つの近似値 z_i, z_{i-1} から(17)式により z_i^* を求めた。

(12) Store $f^*(z_i)$ and obtain $z_i^* = z_i + \lambda(z_i - z_{i-1})$,
where $\lambda = -f_i / (f_i - f_{i-1})$
M=3

(14)では当然、この(12)での z_i, z_i^* を、Fig. 1での z_{i-1}, z_i として扱い、新しい近似値 z_{i+1} を求め、まだ収束していなくて $N3 < 100$ で次の近似値を求める際には、 z_{i+1}, z_i^* を z_i, z_{i-1} のように扱って(Fig. 1では(14)だが) (12)に戻っていく。また実係数に対するMUACHMで、 $x+iy$ の根が求められた後で $x-iy$ を探す過程では、Fig. 1の(9)は不要で、(10), (11)から(12)を経た後(14)へ行く。これらのルーチンの呼び出しは、MULLRA, MULLRBと全く同一である。

3. Newton-Raphson 法ルーチン

α の近傍で $f'(z) \neq 0$ の際には、収束の速い Newton-Raphson 法 (Newton 法) がよく使われている。これは反復関数として

$$F(z) = z - f(z)/f'(z) \quad (18)$$

を用いる反復法である。

$$F'(z) = f(z)f''(z)/[f'(z)]^2 \quad (19)$$

なので、 $F'(\alpha) = 0$ である。又 $f'(\alpha) \neq 0$ 、 $f''(\alpha) \neq 0$ ならば、 $F''(\alpha) \neq 0$ なので、 z_i が根 α に十分近ければ 2 位の収束をする。

FACOM の SSL⁴⁾⁵⁾ には、複素係数多項式の根を求めるルーチンがあり、Newton-Raphson 法を繰り返し用いて全部の根を求めている。倍精度ルーチンの呼び出しは、

$$\text{CALL CNWTND}(AR, AI, N1, EPS, ILL)$$

と MULLRB に対するのと同様、AR, AI に (2) 式の係数を倍精度で入れるが、演算後には、根の実数部、虚数部がそれぞれ AR, AI の最初の N1-1 にセットされる。収束は $|f(z_i)| < 10^{-15}$ であつ、 $|z_i - z_{i-1}|$ あるいは $|z_i - z_{i-1}| / |z_i|$ が EPS より小、あるいは後者の $|z_i - z_{i-1}|$ に対する条件のみかのいずれかで判定しているが、この倍精度実数 EPS も入力できるようになっている。標準値は 10^{-14} 程度だが、重根の多重度に従って大きくする必要があることが記されている。なお ILL は正常に解が求められた時には 0、入力パラメータにエラーがあった時には 30000、収束しなくて演算が途中で止められた時には求められた根の数を表わし、これらの根は AR, AI の (N1-ILL+1) ~ N1 にセットされるようになっている。

α が p 重根 ($p \geq 2$) の場合には、(18) 式では収束が遅くなるので

$$F(z) = z - pf(z)/f'(z) \quad (20)$$

とすればよいことが知られている。この時

$$z_{i+1} = F(z_i) \quad (21)$$

の反復は

$$(z_{i+1} - \alpha) f'(z_i) = G(z_i), \quad (22)$$

$$G(z) = (z - \alpha) f'(z) - pf(z)$$

となり、 ζ_1, ζ_2 を z_i と α の間の値とすると

$$z_{i+1} - \alpha = (z_i - \alpha)^2 G^{(p+1)}(\zeta_1) / [p(p+1) f^{(p)}(\zeta_2)] \quad (23)$$

で、収束の速さはやはり 2 位である。p はあらかじめ分らないが、

$$\alpha_i = f'(z_i)/f''(z_i), \quad \beta_i = f(z_i)/f'(z_i) \quad (24)$$

とおくと

$$r_i = \alpha_i / (\alpha_i - \beta_i) = p + O(\zeta_i - \alpha) \quad (25)$$

であるから、 r_i に最も近い整数が p の推定値を与えることになる。

多項式の 0 点を求める反復法については、無限遠とか微分の 0 点では定義できない Newton-Raphson 反復関数の代りに、Traub⁶⁾ は同次多項式の比で定義される新しい関数を導入している。すなわち実係数多項式 (実係数であることはこの理論にとって本質的ではない)

$$f(z) = \sum_{j=0}^n a_j z^{n-j} \quad (a_0 = 1) \quad (26)$$

につき、その0点 $\alpha_1, \dots, \alpha_n$ がそれぞれ異なり、 α_1 は実数で $|\alpha_1| > |\alpha_j|$, $j = 2 \sim n$, のとき

$$B(z) z^k = G(k, z, B) + f(z) Q(k, z, B) \quad (27)$$

で $(n-1)$ 次多項式 G と $(m+k-n)$ 次多項式 Q を、左辺の $f(z)$ による剰余と商としてそれぞれ定義している。 $B(z)$ は $B(\alpha_1) \neq 0$ の $m (< n)$ 次の多項式、 k は非負の整数である。 G は Lagrange 補間多項式 $L_j(z)$ を用いて

$$G(k, z, B) = \sum_{j=1}^n B(\alpha_j) \alpha_j^k L_j(z) \quad (28)$$

と書けるので、

$$G(k+1, z, B) = zG(k, z, B) - \beta_0(k) f(z),$$

$$G(0, z, B) = B(z), \quad \beta_0(k) = \sum_{j=1}^n B(\alpha_j) \alpha_j^k / f'(\alpha_j) \quad (29)$$

により簡単に求められる。Traub は更に、これを一般化し、

$$G_p(k, z, B) = \{(-1)^{p-1} / (p-1)!\} f^p(z) [G(k, z, B) / f(z)]^{p-1} \quad (30)$$

を定義し ($G_1 = G$)、これは以下により求められることを示している。

$$G_p(k, z, B) = \sum_{j=0}^{p-1} [-f(z)]^{p-1-j} G^{(p-1-j)}(k, z, B) V_j(z) / (p-1-j)!,$$

$$V_j(z) = f'(z) V_{j-1}(z) - f(z) V'_{j-1}(z) / j,$$

$$V_0(z) = 1 \quad (31)$$

これから反復関数として

$$\begin{aligned} F_p(k, z, B) &= G_p(k+1, z, B) / G_p(k, z, B) \\ &= z - p(z) G_{p-1}(k, z, B) / G_p(k, z, B) \end{aligned} \quad (32)$$

を定義している。特に F_2 は

$$\begin{aligned} F_2 &= z - u(z) - u^2(z) W(k, z, B) / [1 - u(z) W(k, z, B)], \\ u(z) &= f(z) / f'(z), \quad W(k, z, B) = G'(k, z, B) / G(k, z, B) \end{aligned} \quad (33)$$

と書け、Newton-Raphson の反復関数 $z - u(z)$ に u^2 の項が付いた形となっている。しかし (33) の反復関数の特長は、 k が十分大きく固定された時、大域的 (global) 収束が証明できることである。以上の方法ではまず反復関数を求め、次いで一連の近似を求めるので、2ステップ・アルゴリズムと呼ばれている。Traub は例として $B(z) = 1$, $B(z) = f'(z)$ を扱い、 k としては n の数倍程度をえらんでいる。

しかしこの2ステップ・アルゴリズムは、同程度の大きさの根が沢山あるときは良くない。これの改良のため、Jenkins-Traub は3段 (three-stage) アルゴリズムを新しく導入しており、⁷⁾ このアルゴリズムが現在、実係数多項式の根を求める標準的方法となっている。

$$f(z) = \sum_{i=0}^n a_i z^{n-i} = \prod_{i=1}^j (z - \alpha_i)^{m_i}, \quad (a_0 = 1, a_n \neq 0) \quad (34)$$

に対し、まず固定 (fixed) シフト K 多項式を、実係数2次式 $\sigma(z)$ と、高々 $(n-1)$ 次の多項

式 $K^{(0)}(z)$ を用い、以下のように導入する。

$$K^{(k+1)}(z) = \left[K^{(k)}(z) + \left\{ \begin{array}{cc} f(s_1) & f(s_2) \\ K^{(k)}(s_1) & K^{(k)}(s_2) \end{array} \right\} z + \begin{array}{cc} K^{(k)}(s_1) & K^{(k)}(s_2) \\ s_1 f(s_1) & s_2 f(s_2) \end{array} \right] \\ \times f(z) / \left[\begin{array}{cc} s_1 f(s_1) & s_2 f(s_2) \\ f(s_1) & f(s_2) \end{array} \right] / \sigma(z) \quad (35)$$

ここで、 $\sigma(z) = z^2 + uz + v = (z-s_1)(z-s_2)$, $f(s_1)f(s_2) \neq 0$ (36)

$$K^{(0)}(z) = \sum_{i=1}^j c_i^{(0)} f_i(z), \quad f_i(z) = f(z)/(z-\alpha_i) \quad (37)$$

である。今 $K^{(k)}$ を leading 係数で割ったものを $\bar{K}^{(k)}$ と書くと、線形因子への収束は、(37)式の $c_1^{(0)} \neq 0$ で $|\sigma(\alpha_1)| < |\sigma(\alpha_i)|$, $i=2 \sim j$, のとき、 α_1 は実数で

$$\lim_{k \rightarrow \infty} f(z) / \bar{K}^{(k)}(z) = z - \alpha_1 \quad (38)$$

又、

$$\sigma^{(k)}(z) = \left| \begin{array}{ccc} K_0^{(k)}(s_1) & K_0^{(k)}(s_2) & z^2 \\ K_1^{(k)}(s_1) & K_1^{(k)}(s_2) & z \\ K_2^{(k)}(s_1) & K_2^{(k)}(s_2) & 1 \end{array} \right| / \left| \begin{array}{cc} K_1^{(k)}(s_1) & K_1^{(k)}(s_2) \\ K_2^{(k)}(s_1) & K_2^{(k)}(s_2) \end{array} \right| \quad (39)$$

$$K_\nu^{(k)}(z) = \sum_{i=1}^j c_i^{(k)} f_i(z) / \alpha_i^\nu, \quad c_i^{(k)} = c_i^{(0)} / \sigma^k(\alpha_i), \quad (\nu = 0, 1, 2) \quad (40)$$

とすると、2次因子への収束は $c_1^{(0)} c_2^{(0)} \neq 0$ で $|\sigma(\alpha_1)| = |\sigma(\alpha_2)| < |\sigma(\alpha_i)|$, $i=3 \sim j$ のとき、 $\lim_{k \rightarrow \infty} \sigma^{(k)}(z) = (z-\alpha_1)(z-\alpha_2)$ (41)

以上により、この K 多項式を f の1次、あるいは2次因子を求めるのに使うことができることが分るが、(40)の収束の速さは1位なので、 $\sigma^{(k)}(z)$ が収束しはじめると、(35)式の $K^{(k+1)}(z)$ 中の $\sigma(z)$ で置き換えて収束を速める。これが可変(variable)シフト K 多項式のアイデアとなっている。

さてこの3段階アルゴリズムの第1段階ではシフトなしの多項式の列を求め、小さい方の0点から求められるようにする。

$$K^{(k+1)}(z) = [K^{(k)}(z) - K^{(k)}(0) f(z) / f(0)] / z, \quad k=0, 1, \dots, M-1; \\ K^{(0)}(z) = f'(z) \quad (42)$$

第2段階(固定シフト過程)では $0 < \beta \leq \min |\alpha_i|$ をとり、 $|s_1| = |s_2| = \beta$, $s_1 \neq s_2$ で、 $|\sigma(\alpha_1)| < |\sigma(\alpha_i)|$, $i=2 \sim j$, か、 $|\sigma(\alpha_1)| = |\sigma(\alpha_2)| < |\sigma(\alpha_i)|$, $i=3 \sim j$, が成立するように(38)式の $\sigma(z)$ をえらび、(35)式で $K^{(k+1)}(z)$ を $k=M, M+1, \dots, L-1$ に対して求める。そして第3段階(可変シフト過程)で、 $|\sigma(\alpha_1)| < |\sigma(\alpha_i)|$, $i=2 \sim j$, が成立するときは、以下の1次因子に対する反復を行う。

$$K^{(k+1)}(z) = [K^{(k)}(z) - K^{(k)}(s^{(k)}) f(z) / f(s^{(k)})] / (z - s^{(k)}), \\ s^{(k+1)} = s^{(k)} - f(s^{(k)}) / \bar{K}^{(k+1)}(s^{(k)}), \quad k=L, L+1, \dots; \quad (43) \\ s^{(L)} = \text{Re}[s_1 - f(s_1) / \bar{K}^{(L)}(s_1)]$$

一方 $|\sigma(\alpha_1)| = |\sigma(\alpha_2)| < |\sigma(\alpha_i)|$, $i=3 \sim j$, が成立するときは, $\sigma(z)$ を $\sigma^{(k)}(z)$ で置換した(35)の $K^{(k+1)}(z)$, および(39)式の $\sigma^{(k+1)}(z)$ を $k=L, L+1, \dots$ に対して求める。この際 s_1, s_2 としては $\sigma^{(k)}(z)$ の0点, $s_1^{(k)}, s_2^{(k)}$ を使い, $\sigma^{(L)}(z)$ は $k=L$ の(39)式を使う。この $K^{(k+1)}(z)$, $\sigma^{(k+1)}(z)$ の反復による $\sigma^{(k)}(z)$ の(41)式の収束も, (43)の $s^{(k)}$ の $(z-\alpha_1)$ への収束も大域的であることが証明されている。

このようにして一度に一つ, あるいは二つの根を求めていくが, 計算は大きい0で多項式を deflate したときに起こる不安定性を避けるように, 大体絶対値の小さい方から計算されている。20~50次の多項式のすべての0点を, Algol W プログラムで計算した例では, 平均 $2n^2$ msec の計算時間が IBM 360/67 に対して報告されている。しかしこのアルゴリズムによる Fortran プログラムは公開されていないようである。

複素数多項式に対する3段アルゴリズム⁸⁾は, 実係数多項式の際と同様, 第1段階のシフトなしの過程は(42)式に従う。第2の固定シフト過程では $0 < \beta \leq \min |\alpha_i|$ をとり, $|s| = \beta$, $|s - \alpha_i| < |s - \alpha_j|$, $i=2 \sim j$, のように s を選び, 以下の反復を行う。

$$K^{(k+1)}(z) = [K^{(k)}(z) - K^{(k)}(s) f(z)/f(s)] / (z-s),$$

$$k = M, M-1, \dots, L-1 \quad (44)$$

最後の可変シフト過程では,

$$s^{(L)} = s - f(s) / \bar{K}^{(L)}(s) \quad (45)$$

ととって, (43)式に従って $s^{(k+1)}$, $K^{(k+1)}(z)$ を $k=L, L+1, \dots$ に対して求めていく。この $s^{(k)}$ が α_1 に大域的収束をし, その速さは2位であることが証明されている。又, この $s^{(k)}$ の反復は1次多項式に収束する有理関数に対するNewton法と同等であることも示されている。

Algol WでのIBM 360/67での計算では, 20~50次多項式のすべての根の計算時間は $8n^2$ msec 程度になっている。

これのFortranプログラムCPOLYはComm. ACMに発表されている。⁹⁾ 大体絶対値の増加の方向で一つづつ0を見つけていき, 次々に1次だけ低次の多項式に deflate している。Algol Wより計算が非常に速くなったことも報告されている。

より最近ではMadsen & Reid¹⁰⁾が, Madsenのアルゴリズム¹¹⁾によるFortranプログラムを発表している。複素係数多項式の方がいくらかアルゴリズムが簡単になるが, 実係数の際も扱えるようになっている。

$$f(z) = \sum_{j=0}^n a_j z^j \quad (46)$$

で①最初は $z_0 = 0$ から始め, $f'(0) \neq 0$ の時

$$dz_0 = -f(0)/f'(0),$$

その他の時は $dz_0 = 1$ として, これより

$$z_1 = \frac{1}{2} \min_{k>0} (|a_0/a_k|^{1/k}) dz_0 / |dz_0| \quad (47)$$

を求める。②これ以後は前の反復ステップがうまくいった時 ($z_i \approx z_{i-1}$) には, Newton 補正

$$n_i = -f(z_i)/f'(z_i)$$

を求め, $|n_i| \leq 3 |z_i - z_{i-1}|$ の時には $dz_i = n_i$ とするが, そうでなければ

$$dz_i = 3 |z_i - z_{i-1}| \exp(i\theta) n_i / |n_i| \quad (48)$$

とする。θ は任意だが $\tan^{-1}(3/4)$ ととっている。これに対して前のステップがうまくいかず $z_i = z_{i-1}$ とおいた時には以下のようにとる。

$$dz_i = -\exp(i\theta) dz_{i-1} / 2 \quad (49)$$

③次に

$$|f(z_i + dz_i)| < |f(z_i)|$$

をテストし、これを満足していれば

$$|f(z_i + t dz_i)|, \quad t=1 \sim n$$

を順に値が減少する限り求めていく。満足していない時には

$$|f(z_i + dz_i / 2^t)|, \quad t=0, 1, 2, |f(z_i + \exp(i\theta) dz_i / 4)|$$

を順に求め、これにより見付けられたところが z_{i+1} となる。④ここでNewton反復法の収束が保証されるかどうかを、以下のKantorowitzの定理に基づいて判定する。¹¹⁾

「 K_0 を $(z_i + n_i)$ を中心とし、半径 $|n_i|$ の円とするとき、

$$f(z_i) f'(z_i) \neq 0, \quad 2 \left| f(z_i) \right| \max_{z \in K_0} |f''(z)| \leq |f'(z_i)|^2$$

であれば、 z_i から出発するNewton反復は収束する。」⑤収束が保証されていれば、後の反復は普通のNewton法にする。そうでなければステップ②へ帰ってこの反復を繰り返す。このように②、③のステップをとることにより多重根への速い収束も保証される。

誤差限界の評価も含めたFortranプログラム・リストが示されており、数値例によりJenkins-Traubのアルゴリズム⁸⁾より2~4倍速いことも示されている。従って我々は、実係数に対するPA07ADプログラム、及び複素係数に対するPA06ADプログラムを倍精度のルーチンとして整備した。これらのルーチンは以下により呼び出せる。

CALL PA07AD(A, N, R, E, W, S, N1, LW)

CALL PA06AD(A, N, R, E, W, S, N1, LW)

PA06ADのA(2, N1)は(48)式の係数 a_j の実数部及び虚数部がそれぞれA(1, j), A(2, j)に入る倍精度配列〔PA07ADではA(N1)で実数部のみ〕, Nは多項式の次数n, R(2, N)は計算結果の根の実数部, 虚数部がそれぞれR(1, j), R(2, j)に入る倍精度配列, E(N1)は入力としては係数の誤差限界(計算機の精度まで正確なときには0), 出力としては最初のN個は根の近似値誤差限界がセットされる単精度配列, PA06ADのW(2, LW)は計算に使われる倍精度配列〔PA07ADではW(LW)〕, PA06ADのS(4, LW)も計算に使用される単精度配列〔PA07ADではS(2, LW)〕だがWと等価にできる。N1は(n+1), LWはPA06ADでは(5n/4+2)以上, PA07ADでは(3n/2+2)以上の整数である。

これらのルーチンは(48)式の根を求めるPA07BD, あるいはPA06BDルーチンと、求められた根の誤差限界を求めるPA07CD, あるいはPA06CDよりなっている。前者のルーチンの計算の流れをFig.2に示したが、(48), (49)式の②, ③のステップの計算が, Fig.2の(120), (160), (240)まででなされており, Newton反復が保証される場合には(160)から(240)の間のステップはとられない。なおこれらのルーチン中で使われているPA07ED(A, E, Q), PA06ED(A, E, Q)は全く同一のサブルーチン副プログラムで、複素数の割算 $Q = -A/B$ を

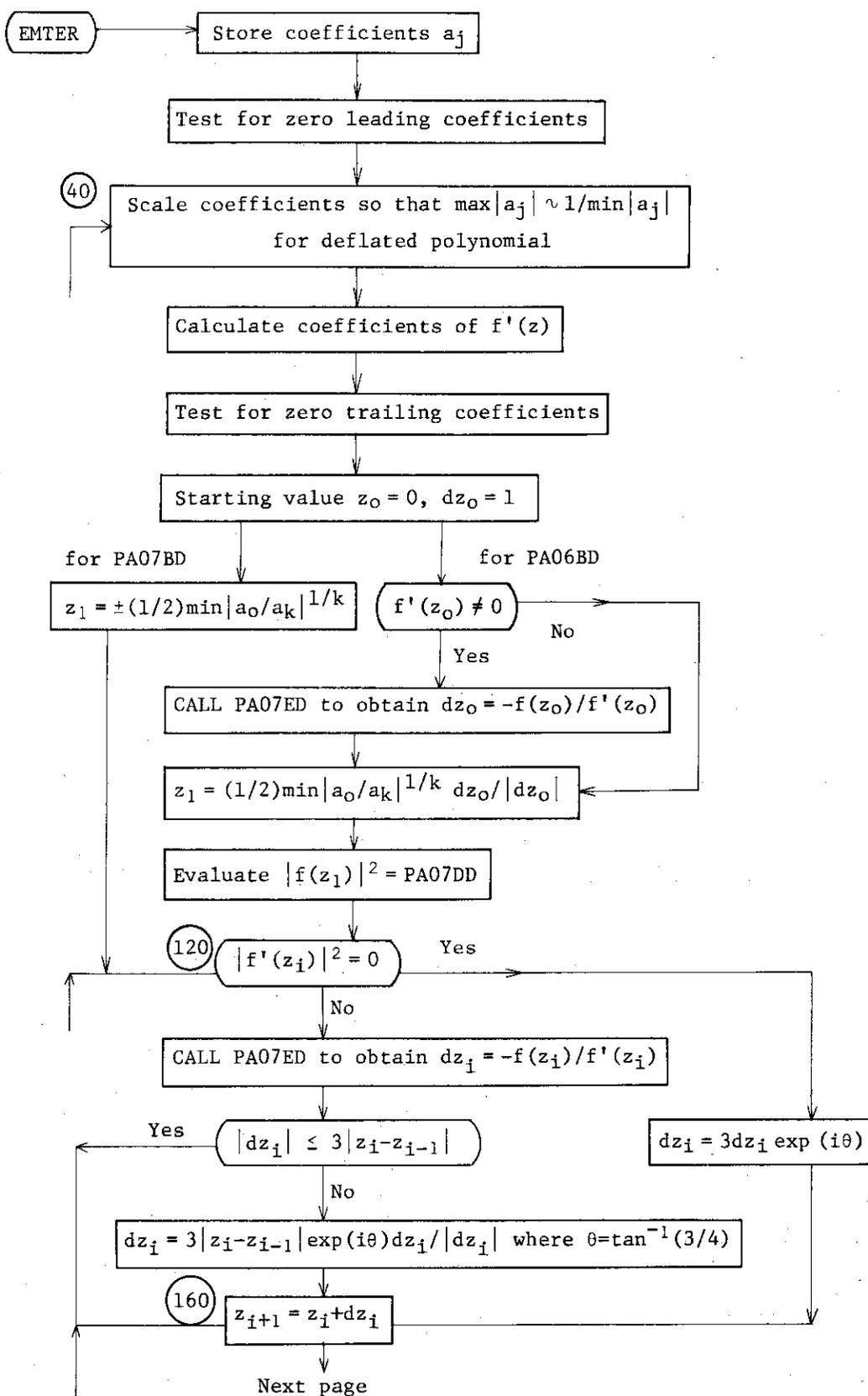


Fig. 2 Flow diagram of PA07BD or PA06BD to find all the roots of

$$f(z) = \sum_{j=0}^n a_j z^j$$

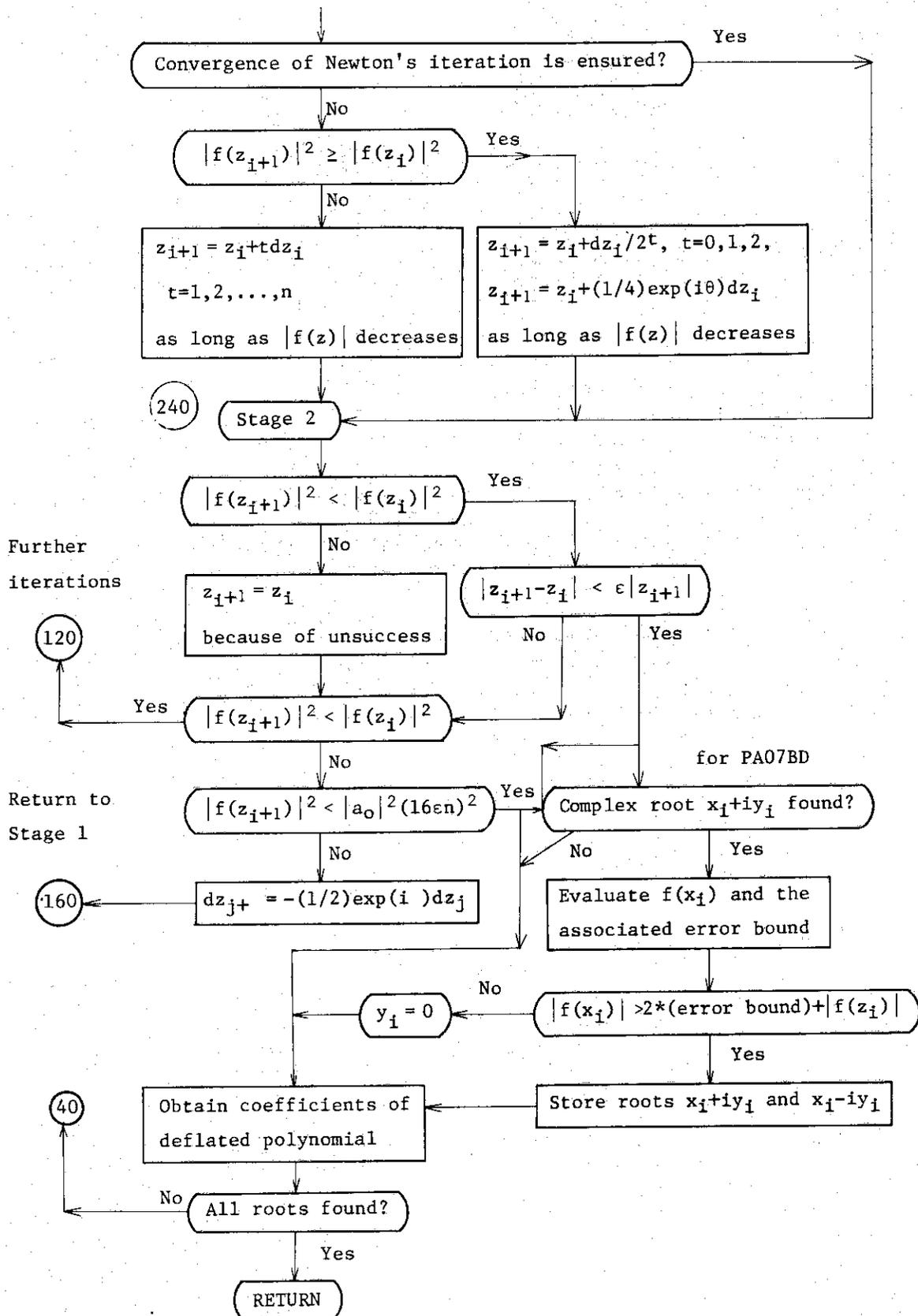


Fig. 2 (Continued)

する。又関数副プログラム PA07DD(Z, FZ, N1, A), PA06DD(Z, FZ, N1, A)は

$$f(z) = \sum_{j=0}^{N1-1} a_j z^{N1-1-j}$$

の実数部, 虚数部がそれぞれ FZ(1), FZ(2) にストアされ, 関数値としては $|f(z)|^2$ が求められる。PA07DDのAは実数で, PA06DDのAは複素数である。

収束判定は根の近似値の相対誤差が計算機の精度以下 ($\epsilon = 2.3 \times 10^{-16}$) になるか, あるいは関数値の絶対値が, 絶対値最小の根での $f(z)$ の計算精度誤差 $16n |a_0|^\epsilon$ 以下になるかで行っている。なお実係数に対する PA07BD では複素根が求められた時には, その実数部だけで根になっているかどうか確かめ, そうでなく複素根であることが確認された時にはその複素共役根をとるようにしている。これらのルーチンの呼び出しは以下による。

CALL PA07BD(A, N, R, W, N1)

CALL PA06BD(A, N, R, W, N1)

これらの引数は, PA07AD, PA06AD と同じだが, W の大きさは PA06BD では (2, N1), PA07BD では (N1) となっている。

次に求められた根の誤差限界を求めるルーチン PA07CD, PA06CD 中の計算の流れを Fig. 3 に示した。まず求められた根により作られる多項式 $P(z) = a_n \prod_{j=1}^n (z - \alpha_j)$ と(40)式との差の多項式 $Q(z) = \sum_{j=0}^n \epsilon_j z^j$ を求め, Rouche' の定理から導出された以下の定理に基づいて誤差限界を求めている。

「 $P(z)$, $Q(z)$ が求められた各々の根 α_j を中心とする半径 r_j の円の内側及び周上で $|Q(z)| < |P(z)|$ を満足するとき, $f(z)$ の根 ϕ_j は $|\alpha_j - \phi_j| \leq r_j$ である。」

今根 α_1 の誤差限界を求めているとし, 根の順序を $|\alpha_j - \alpha_1|$ が単調に増加するように並べ替えたとすると, $|Q(z)| < |P(z)|$ の条件は

$$k(r) < \prod_{j=1}^m (r - |\alpha_j - \alpha_1|), \quad (50)$$

$$k(r) = \sum_{j=0}^m |\epsilon_j| (r + |\alpha_1|)^j / \left[|a_n| \prod_{j=m+1}^n (|\alpha_j - \alpha_1| - r) \right]$$

ここで m は以下を満足するものである。

$$|\alpha_m - \alpha_1| < r < |\alpha_{m-1} - \alpha_1| \quad (51)$$

そして r の反復は

$$1.05 k(r_j) < \prod_{j=1}^m (r_{j+1} - |\alpha_j - \alpha_1|) < 1.1 k(r_j) \quad (52)$$

で行うが, これを厳密に解く必要もないので, 上限と下限の平均を用いている。また求められた r_{j+1} を直接(50)式でチェックする代りに

$$\left(\frac{r_{j+1} + |\alpha_1|}{r_j + |\alpha_1|} \frac{|\alpha_{m+1} - \alpha_1| - r_{j+1}}{|\alpha_{m+1} - \alpha_1| - r_j} \right)^n \leq 1.05 \quad (53)$$

によってもテストを実施している。

実係数に対する PA07CD では, 複素根の後にはその複素共役根があると仮定し, 複素根に対して求められた誤差半径をそのまま次の根の誤差半径としている。なお(52)の反復が 100 回

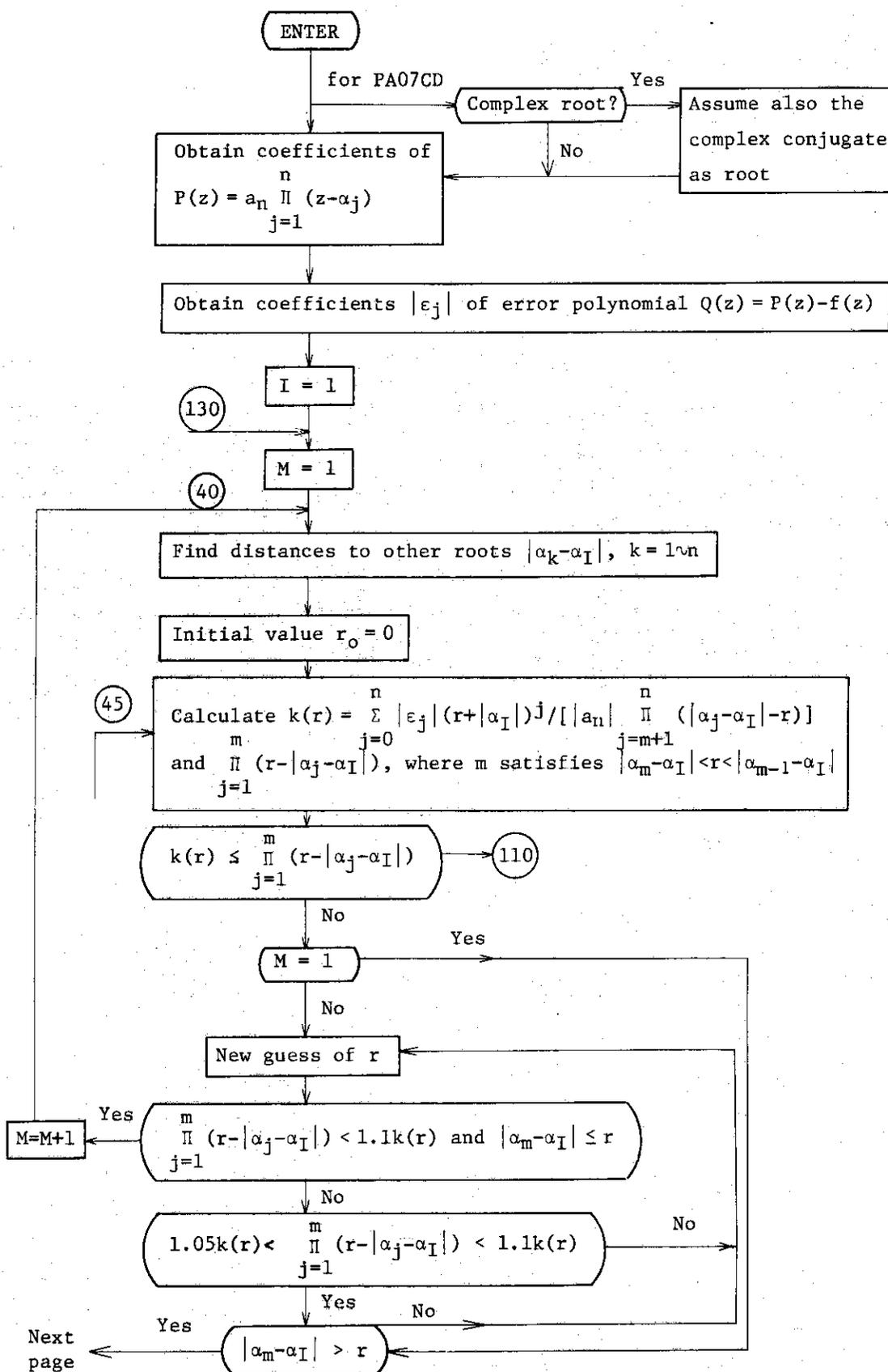


Fig. 3 Flow diagram of PA07CD or PA06CD to find error bounds of all the roots α_j 's of $f(z) = \sum_{j=0}^n a_j z^j$

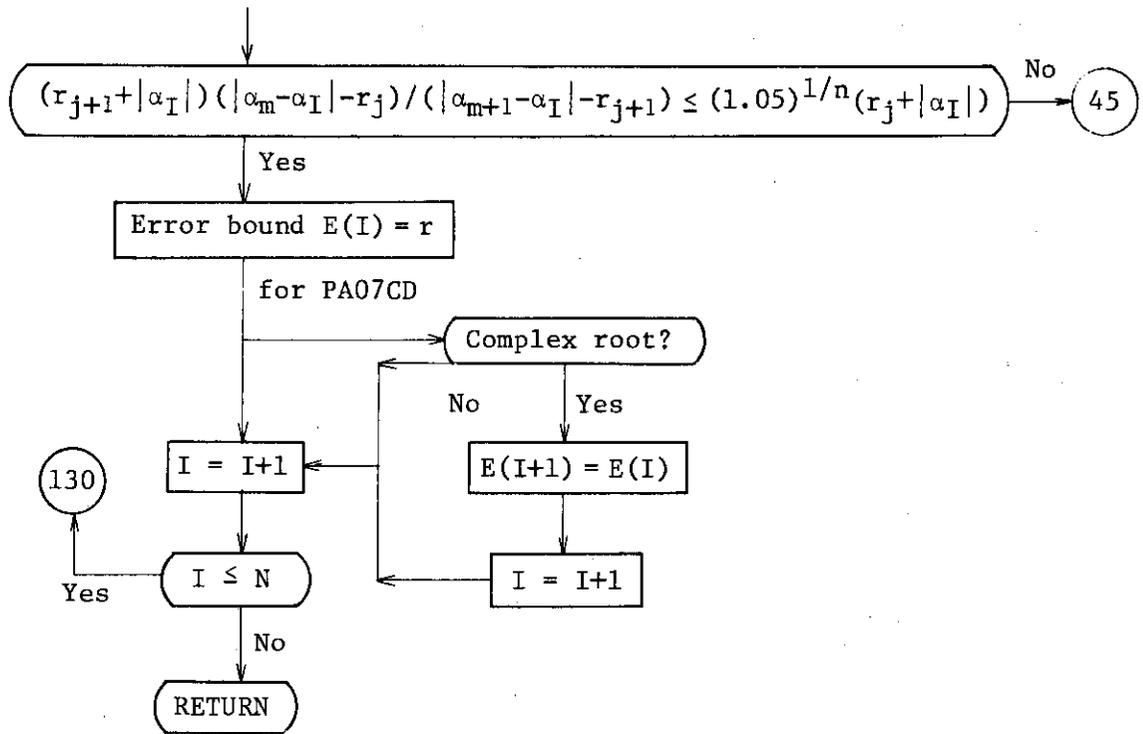


Fig. 3 (Continued)

以上になる時には誤差半径を0にセットし、次の根に対する計算に移るようにしている。これらの呼び出しは以下によりなされる。

```
CALL PA07CD(A, N, N1, R, E, W, F, IG, CR)
```

```
CALL PA06CD(A, N, N1, R, E, W, F, IG, CR)
```

ここで引数のA, N, N1, R, EはPA07AD, PA06ADのと同じだが、WはPA07CDでは大きさN1, PA06CDでは(2, N1)の倍精度配列で、P(z)の係数をストアするのに用いられる。Fは大きさN1の1次元単精度配列で、Q(z)の係数をストアするのに用いられ、IG(N)は根のグループ化のための単精度実数配列、CR(N)は一つの根から他の根までの距離をストアする単精度配列である。

4. Bairstow法及びJarrat法ルーチン

Bairstow法(Hitchcock法)は実係数多項式(2)の複素根を得るため、 $f(z)$ の2次因子 $z^2 + p^*z + q^*$ を求める手法である。まず試みの2次因子 $z^2 + pz + q$ を定め、以下の2次因子の組立て除法により b_j, c_j を求める。

$$\begin{aligned} b_j &= a_j - pb_{j-1} - qb_{j-2}, \quad j=0 \sim n; \quad b_{-1} = b_{-2} = 0, \\ c_j &= b_j - pc_{j-1} - qc_{j-2}, \quad j=0 \sim n-1; \quad c_{-1} = c_{-2} = 0 \end{aligned} \quad (54)$$

次に連立方程式

$$\begin{aligned} c_{n-2} \Delta p + c_{n-3} \Delta q &= b_{n-1}, \\ c_{n-1} \Delta p + c_{n-2} \Delta q &= b_n, \quad c_{n-1} = c_{n-1} - b_{n-1} \end{aligned} \quad (55)$$

より $\Delta p, \Delta q$ を求め

$$p^* = p + \Delta p, \quad q^* = q + \Delta q \quad (56)$$

とする。次には $z^2 + p^*z + q^*$ を新しい試みの因子とみて、 $\Delta p, \Delta q$ が十分小さくなるまで繰返すのである。この方法は2変数のNewton-Raphson法になっており、2位の収束をしている。

FACOMのSSLにはBairstow法を用いたルーチンが組み込まれている。これは実係数多項式の解を求めるとき、その次数が奇数のときにはまずNewton-Raphson法で1次根を求め、これにより次数を一つ下げて偶数次とする。そして偶数次(n)の方程式につき、 $n/2$ 組の2次式に分解し、根の公式を使ってn個の0を求めている。倍精度ルーチンは

```
CALL BAIR1D(A, M, EPS, RP, IP, ILL)
```

により呼ばれる。ここでAはMの1次元倍精度配列で(2)式の係数が a_0 から順番に入れられる。ただし演算後はAの内容は毀される。Mは $n+1 \geq 3$ で、EPSは倍精度の収束判定値で、

$|z_i - z_{i-1}| < \text{EPS}$, または $|z_i - z_{i-1}| / |z_i| < \text{EPS}$ かつ $f'(z_{i-1}), f'(z_{i-2}) > 0$ で判定している。重根、近接根のあるときにはEPSを大きくする必要があるが、標準値として 10^{-14} があげられている。RP, IPはそれぞれ(M-1)の大きさの倍精度配列で、根の実数部、虚数部がセットされる。なおILLは正常に解が求まった時0, 入力パラメータにエラーがあった時30000, 収束条件を満足する解が求められなかった時1あるいは2がセットされる。

Bairstow法は重根あるいは近接根の際には、計算時間の点で非実用的である。この際には

以上になる時には誤差半径を0にセットし、次の根に対する計算に移るようにしている。これらの呼び出しは以下によりなされる。

```
CALL PA07CD(A, N, N1, R, E, W, F, IG, CR)
```

```
CALL PA06CD(A, N, N1, R, E, W, F, IG, CR)
```

ここで引数のA, N, N1, R, EはPA07AD, PA06ADのと同じだが、WはPA07CDでは大きさN1, PA06CDでは(2, N1)の倍精度配列で、P(z)の係数をストアするのに用いられる。Fは大きさN1の1次元単精度配列で、Q(z)の係数をストアするのに用いられ、IG(N)は根のグループ化のための単精度実数配列、CR(N)は一つの根から他の根までの距離をストアする単精度配列である。

4. Bairstow法及びJarrat法ルーチン

Bairstow法(Hitchcock法)は実係数多項式(2)の複素根を得るため、 $f(z)$ の2次因子 $z^2 + p^*z + q^*$ を求める手法である。まず試みの2次因子 $z^2 + pz + q$ を定め、以下の2次因子の組立て除法により b_j, c_j を求める。

$$\begin{aligned} b_j &= a_j - pb_{j-1} - qb_{j-2}, \quad j=0 \sim n; \quad b_{-1} = b_{-2} = 0, \\ c_j &= b_j - pc_{j-1} - qc_{j-2}, \quad j=0 \sim n-1; \quad c_{-1} = c_{-2} = 0 \end{aligned} \quad (54)$$

次に連立方程式

$$\begin{aligned} c_{n-2} \Delta p + c_{n-3} \Delta q &= b_{n-1}, \\ c_{n-1} \Delta p + c_{n-2} \Delta q &= b_n, \quad c_{n-1} = c_{n-1} - b_{n-1} \end{aligned} \quad (55)$$

より $\Delta p, \Delta q$ を求め

$$p^* = p + \Delta p, \quad q^* = q + \Delta q \quad (56)$$

とする。次には $z^2 + p^*z + q^*$ を新しい試みの因子とみて、 $\Delta p, \Delta q$ が十分小さくなるまで繰返すのである。この方法は2変数のNewton-Raphson法になっており、2位の収束をしている。

FACOMのSSLにはBairstow法を用いたルーチンが組み込まれている。これは実係数多項式の解を求めるとき、その次数が奇数のときにはまずNewton-Raphson法で1次根を求め、これにより次数を一つ下げて偶数次とする。そして偶数次(n)の方程式につき、 $n/2$ 組の2次式に分解し、根の公式を使ってn個の0を求めている。倍精度ルーチンは

```
CALL BAIR1D(A, M, EPS, RP, IP, ILL)
```

により呼ばれる。ここでAはMの1次元倍精度配列で(2)式の係数が a_0 から順番に入れられる。ただし演算後はAの内容は毀される。Mは $n+1 \geq 3$ で、EPSは倍精度の収束判定値で、

$|z_i - z_{i-1}| < \text{EPS}$, または $|z_i - z_{i-1}| / |z_i| < \text{EPS}$ かつ $f'(z_{i-1}), f'(z_{i-2}) > 0$ で判定している。重根、近接根のあるときにはEPSを大きくする必要があるが、標準値として 10^{-14} があげられている。

RP, IPはそれぞれ(M-1)の大きさの倍精度配列で、根の実数部、虚数部がセットされる。なおILLは正常に解が求まった時0, 入力パラメータにエラーがあった時30000, 収束条件を満足する解が求められなかった時1あるいは2がセットされる。

Bairstow法は重根あるいは近接根の際には、計算時間の点で非実用的である。この際には

第2章で述べたMuller法が用いられるが、これも3重根、あるいは三つの近接根の際には使えない。これをJarratらは改良するため、有理関数を用いる手法を示している。最初のJarrat & Nuddsの仕事¹²⁾では(11)式の反復で、 $f(z)$ を分母、分子とも $(z-z_i)$ についての1次関数で近似している。反復アルゴリズムの有用性を、反復の各ステップで m 個の関数値とか微分の計算が必要なとき、収束の位数の m 重根と定義し、Newton-Raphson法の有用性は $\sqrt{2}$ 、regula falsi法は1.62であるのに対し、この1次有理関数はMuller法と同様1.84であることが示されている。その上Muller法より数式が簡単なので計算時間が節約でき、実根は複素数演算を使わずに見付けられる利点があると述べている。

Garsideら¹³⁾は、この方法を変形し、 $f(z)$ でなく $f(z)/f'(z)$ を $(z-a)/(b+cz)$ と近似し、反復により a を求めている。その際分母が0に近くなる場合の処理、および a の極限としての根 α の精度を上げる修正も施している。一般化して

$$f(z)/f'(z) = (z-a) / \sum_{j=0}^{k-2} c_j z^j \quad (57)$$

を考えているが、 $k=4$ は収束の位数は1.928に上るが桁落ちが出るので、位数1.839の $k=3$ が実用上ベストであると述べている($k=2$ はsecant formulaである) 初期値 z_1, z_2, z_3 のえらび方、数値例も示している。

FACOMのSSL⁴⁾⁵⁾にも、実係数高次代数方程式に対するJarrat-modify法として組込まれている。

CALL JARATD(A, N, EPS, RP, IP, ILL)

により倍精度ルーチンが呼ばれるが、 A は(12)式の係数が a_0 から順番に入れられる $(N+1)$ 以上の大きさの倍精度1次元配列、 N は多項式の次数、 EPS は $|f(z)| \leq EPS^* |a_{n+1}|$ と収束判定に用いられる倍精度実数で、標準値として $10^{-16} / |a_{n+1}|$ 、あるいは $a_{n+1}=0$ のときには、 10^{-16} があげられている。 RP, IP は根の実数部、虚数部がセットされる N 以上の大きさの1次元倍精度配列で、 ILL は正常に解が求められた時には0、入力パラメータにエラーがあった時には30000、収束しなくて計算が途中で打ち切られた時には、すでに求められた $ILL-1$ 個のみが RP, IP にセットされていることを示す整数である。

複素係数多項式をJarrat法で解くFACOM-SSLのルーチンもあり、この倍精度ルーチンは以下により呼び出される。

CALL CJARTD(A, N, EPS, RT, ILL)

ここで A は1次元倍精度複素数型配列で、JARATDと同様に係数が入れられる。 N は $n+1$ 、 EPS は収束判定に用いられる倍精度実数で、標準値は 10^{-15} であるが、0を入力した時には内部的に 10^{-21} にセットされる。 RT が求めた根がセットされる倍精度複素数型1次元配列で、 ILL はJARATDの際と同様、このサブルーチンから戻った時の状態がセットされる。

JARATDで用いられている手法を複素係数の際に適用したルーチンとして、FACOMのSSL II¹⁴⁾には、倍精度に対してはDCJARTがある。これは

CALL DCJART(ZA, N, Z, ICON)

で呼ばれるが、 ZA がCJARTDの A に対応するが、演算後には内容は保存されない。 N は次数 n 、 Z がCJARTDの RT で、 $ICON$ はエラーなしの時0、すべての根が求められなかった

時10000, 入力パラメータのエラーの時30000となる。収束は ϵ を丸め誤差の単位とするとき,

$$\left| f(z) \right| \leq \epsilon \sum_{j=0}^n \left| a_j z^{n-j} \right| \quad (58)$$

で判定している。

これらのJarrat法は, 重根, 近接根に関係なく解が求められるので, Bairstow法の欠点を補っているが, 一方多少誤差が大きく, 計算時間も長いといわれている。

5. 数値テストと検討

前節までで多項式の全ての根を求めるルーチンとして, 実係数多項式に対してはMuller法のMULLRA, これにChambersのアルゴリズムを取り入れたMUACHM, Newton反復法の収束が保証されるようになるまで特別のステップをとるMadsenのアルゴリズムによるPA07AD, Bairstow法のBAIR1D, 及びJarrat法のJARATDの五つ, 複素係数多項式に対してMuller法のMULLRB, Chambersのアルゴリズムを取り入れたMUBCHM, Newton法のCNWTND, MadsenアルゴリズムのPA06AD, Jarrat法のCJARTDとDCJARTの六つの特長を述べた。

本節ではこれらのルーチンを数値的にテストした結果を述べる。このためTable 1に示した3次から8次までの八つ(実係数に対してはExample No1から7までの七つ)の多項式を選んだ。Example No1から3, 及びNo7はTraub⁶⁾が採用した例題である。Example No4, No5はHansen & Patrick¹⁵⁾が採用したもので, Example No6はJenkins & Traub⁷⁾が1対の複素共役根, 三つの絶対値の等しい根(そのうち二つは2重根), それに重根に近い1対の根をもっている多項式として採用している。複素係数のExample No8は, Jenkins & Traub⁸⁾が2重根と三つの殆んど絶対値の等しい根(そのうちの二つは殆んど重根)をもつ多項式として採用している。この他にMadsen & Reid¹⁰⁾が用いた, 根を入力して多項式を組み立てる例題も採用した。Table 2に示したExample No9~11がそれで, そこには倍精度計算で求められた多項式の係数の値も書いておいた。付録1のメイン・プログラムがこのためのもので, 多項式の係数を入力する方のメイン・プログラムは付録2に示されている。

まず実係数多項式に対する五つのルーチンによるテスト結果をTable 3~7に示したが, これらのルーチンで求められた根の誤差限界を, 前述のFig.3で示したPA07CD(付録5参照)を通して計算した結果も併記しておいた。ただしPA07CDに入る前に, 付録1のメイン・プログラムから分るように, 求められた根の実数部と虚数部を比較し, 一方の絶対値が他方の 10^{-10} 以下の時には, それを0で置きかえ, 根の順番を絶対値の大きい順序に並べかえた。これらの全結果を計算時間の観点からまとめたのがTable 8である。

Table 3.1に示した付録1のMuller法ルーチンによる結果は, 3重根のあるExample No4に小さいが虚数部が出てきている。しかしとにかく一応収束はしているが, Table 8から分るように収束は遅い。またこのような虚数部が付いてくると, PA07CDでの根の誤差限界の

時 10000, 入力パラメータのエラーの時 30000 となる。収束は ϵ を丸め誤差の単位とするとき,

$$|f(z)| \leq \epsilon \sum_{j=0}^n |a_j z^{n-j}| \quad (59)$$

で判定している。

これらの Jarrat 法は, 重根, 近接根に関係なく解が求められるので, Bairstow 法の欠点を補っているが, 一方多少誤差が大きく, 計算時間も長いといわれている。

5. 数値テストと検討

前節までで多項式の全ての根を求めるルーチンとして, 実係数多項式に対しては Muller 法の MULLRA, これに Chambers のアルゴリズムを取り入れた MUACHM, Newton 反復法の収束が保証されるようになるまで特別のステップをとる Madsen のアルゴリズムによる PA07AD, Bairstow 法の BAIR1D, 及び Jarrat 法の JARATD の五つ, 複素係数多項式に対して Muller 法の MULLRB, Chambers のアルゴリズムを取り入れた MUBCHM, Newton 法の CNWTND, Madsen アルゴリズムの PA06AD, Jarrat 法の CJARTD と DCJART の六つの特長を述べた。

本節ではこれらのルーチンを数値的にテストした結果を述べる。このため Table 1 に示した 3 次から 8 次までの八つ (実係数に対しては Example No1 から 7 までの七つ) の多項式を選んだ。Example No1 から 3, 及び No7 は Traub⁶⁾ が採用した例題である。Example No4, No5 は Hansen & Patrick¹⁵⁾ が採用したもので, Example No6 は Jenkins & Traub⁷⁾ が 1 対の複素共役根, 三つの絶対値の等しい根 (そのうち二つは 2 重根), それに重根に近い 1 対の根をもっている多項式として採用している。複素係数の Example No8 は, Jenkins & Traub⁸⁾ が 2 重根と三つの殆んど絶対値の等しい根 (そのうちの二つは殆んど重根) をもつ多項式として採用している。この他に Madsen & Reid¹⁰⁾ が用いた, 根を入力して多項式を組み立てる例題も採用した。Table 2 に示した Example No9~11 がそれで, そこには倍精度計算で求められた多項式の係数の値も書いておいた。付録 1 のメイン・プログラムがこのためのもので, 多項式の係数を入力する方のメイン・プログラムは付録 2 に示されている。

まず実係数多項式に対する五つのルーチンによるテスト結果を Table 3~7 に示したが, これらのルーチンで求められた根の誤差限界を, 前述の Fig.3 で示した PA07CD (付録 5 参照) を通じて計算した結果も併記しておいた。ただし PA07CD に入る前に, 付録 1 のメイン・プログラムから分るように, 求められた根の実数部と虚数部を比較し, 一方の絶対値が他方の 10^{-10} 以下の時には, それを 0 で置きかえ, 根の順番を絶対値の大きい順序に並べかえた。これらの全結果を計算時間の観点からまとめたのが Table 8 である。

Table 3.1 に示した付録 1 の Muller 法ルーチンによる結果は, 3 重根のある Example No4 に小さいが虚数部が出てきている。しかしとにかく一応収束はしているが, Table 8 から分るように収束は遅い。またこのような虚数部が付いてくると, PA07CD での根の誤差限界の

Table 1 Polynomials used for the numerical examples (Example Nos. 1-8) and their roots

Example	$f(z)$	Roots
No. 1	$z^3 - 6z^2 + 11z - 6$	3, 2, 1
No. 2	$z^3 - 5.01z^2 + 8.03z - 4.02$	2.01, 2, 1
No. 3	$z^4 - 46z^3 + 528z^2 - 1090z + 2175$	29, 15, $1 \pm 2i$
No. 4	$z^6 - 6z^5 + 50z^4 - 45z^3 - 108z^2 + 108$	3, 3, 3, -2, -2, 1
No. 5	$z^6 - 4z^5 - 5z^4 + 190z^3 - 666z^2 + 944z - 600$	-6, $3 \pm 4i$, 2, $1 \pm i$
No. 6	$z^7 - 6.01z^6 + 12.54z^5 - 8.545z^4 - 5.505z^3 + 12.545z^2 - 8.035z + 2.01$	2.01, 2, 1, 1, -1, $0.5 \pm 0.5i$
No. 7	$z^8 - 36z^7 + 546z^6 - 4536z^5 + 22499z^4 - 67284z^3 + 118124z^2 - 109584z + 40320$	8, 7, 6, 5, 4, 3, 2, 1
No. 8	$z^5 - (13.999 + 5i)z^4 + (74.99 + 55.998i)z^3 - (159.959 + 260.982i)z^2 + (1.95 + 463.934i)z + (150 - 199.95i)$	$4 \pm 3i$, $3.999 + 3i$, $1 + i$, $1 + i$

Table 2 The roots and calculated coefficients of polynomials used for the numerical examples (Example Nos. 9~11)

Example	No. 9	No. 10	No. 11
Roots	1.0×10 ⁶ , 15, 14 13, 12, 11, 10, 9, 8	-10, 5.2, 5.2, 5.2, -1.1×10 ⁻⁶ , -1.1×10 ⁻⁶ 1.1×10 ⁻⁶ , 2.1×10 ⁻⁸ -2.1×10 ⁻⁸ , -2.1×10 ⁻⁸	8192, 4096, 2048, 1024, 512, 256, 128, 64, 32, 16, 8, 4, 2, 1, 0.5, 0.25, 0.125, 0.0625, 0.03125, 0.015625
Calculated coefficients (a ₀ ~a _n) for f(z) = $\sum_{j=0}^n a_j z^{n-j}$	1.0 -1.000092×10 ⁶ 9.2003682×10 ⁷ -3.68208372×10 ⁹ 8.3721182769×10 ¹⁰ -1.1827796305×10 ¹² 1.0630567354×10 ¹³ -5.9354216204×10 ¹³ 1.8820465946×10 ¹⁴ -2.594592×10 ¹⁴	1.0 -5.599998879 -7.4880006278×10 ¹ 6.7059191606×10 ² -1.4060792483×10 ³ -1.5762164762×10 ⁻³ 1.6694955234×10 ⁻⁹ 1.9079160655×10 ⁻¹⁵ 3.856536801 ×10 ⁻²³ -8.4108444074×10 ⁻³¹ -1.7331891857×10 ⁻³⁸	1.0 -1.6383984375×10 ⁴ 8.9478229333×10 ⁷ -2.0942938814×10 ¹¹ 2.2875099445×10 ¹⁴ -1.2089674533×10 ¹⁷ 3.1439870272×10 ¹⁹ -4.0557432651×10 ²¹ 2.6055367125×10 ²³ -8.3519943934×10 ²⁴ 1.3369722403×10 ²⁶ -1.0690552824×10 ²⁷ 4.2689113497×10 ²⁷ -8.5055100998×10 ²⁷ 8.439575913 ×10 ²⁷ -4.1539805391×10 ²⁷ 1.006057513 ×10 ²⁷ -1.178982643 ×10 ²⁶ 6.4475859245×10 ²⁴ -1.5111558334×10 ²³ 1.1805916207×10 ²¹

Table 3.1 The roots and the error bounds obtained with MULLRA and PA07CD for Examples Nos. 1~7

Example	Real part	Imaginary part	Error
No. 1	3.000000000000000D+00	0.0	1.52048E-14
	2.000000000000000D+00	0.0	1.52220E-14
	1.000000000000000D+00	0.0	3.05317E-15
No. 2	2.010000000000000D+00	0.0	4.99141E-12
	2.000000000000000D+00	0.0	5.01246E-12
	9.9999999999977D-01	0.0	2.58806E-14
No. 3	2.900000000000000D+01	0.0	5.71815E-14
	1.500000000000000D+01	0.0	3.55504E-14
	1.000000000000004D+00	2.000000000000002D+00	1.94463E-15
	1.000000000000000D+00	-2.000000000000000D+00	1.94463E-15
No. 4	3.000001740633287D+00	-3.160628423062742D-07	---
	3.000000241780374D+00	1.497083751848665D-06	---
	2.999998457389213D+00	-4.753328691221954D-08	---
	-2.000000000429563D+00	0.0	---
	-1.99999999439460D+00	0.0	---
	1.000000000000000D+00	0.0	---
No. 5	-6.000000000000000D+00	0.0	1.01531E-15
	3.000000000000000D+00	-4.000000000000000D+00	3.09207E-15
	3.000000000000000D+00	4.000000000000000D+00	3.09207E-15
	2.000000000000000D+00	0.0	6.47510E-15
	1.000000000000000D+00	-1.000000000000000D+00	2.51804E-15
	1.000000000000000D+00	1.000000000000000D+00	2.51804E-15
No. 6	2.00999999999997D+00	0.0	1.03272E-06
	2.000000000000000D+00	0.0	1.04612E-06
	1.000000002148208D+00	0.0	7.46687E-05
	-1.000000000000000D+00	0.0	6.43484E-11
	9.99999976703300D-01	0.0	7.46687E-05
	5.000000000000000D-01	5.000000000000000D-01	1.28326E-09
	5.000000000000000D-01	-5.000000000000000D-01	1.28326E-09
No. 7	8.000000000000000D+00	0.0	5.23441E-11
	7.000000000000014D+00	0.0	2.00690E-10
	6.000000000000019D+00	0.0	3.13118E-10
	4.9999999999983D+00	0.0	2.54458E-10
	3.9999999999984D+00	0.0	1.14153E-10
	3.000000000000010D+00	0.0	2.74417E-11
	2.000000000000000D+00	0.0	3.11045E-12
	1.000000000000000D+00	0.0	1.15991E-13

Table 3.2 The roots and the error bounds obtained with MULLRA and PA07CD for Examples Nos. 9~11

Example	Real part	Imaginary part	Error
No. 9	1.0000000000000000D+06	0.0	5.67178E-10
	1.4999999999999999D+01	0.0	5.63840E-05
	1.4000000000000000D+01	0.0	3.01251E-04
	1.2999999999999999D+01	0.0	6.82280E-04
	1.2000000000000000D+01	0.0	8.48312E-04
	1.1000000000000000D+01	0.0	5.24694E-04
	9.9999999999999999D+00	0.0	2.72125E-04
	8.9999999999999999D+00	0.0	6.48379E-05
	7.9999999999999999D+00	0.0	6.50786E-06
No. 10	-1.0000000000000000D+01	0.0	—
	5.200006476973378D+00	3.287661531595857D-08	—
	5.199996951859076D+00	5.279608635675378D-06	—
	5.199996790456326E+00	-5.539283588858762D-06	—
	-5.906091991208149D-05	-1.103038565047709D-04	—
	-5.107435096247695D-05	1.048143356361066D-04	—
	1.465663810140057D-05	-1.103883619498923D-04	—
	1.009327726203053D-04	-3.704480551914628D-06	—
	1.429522947184353D-05	9.670912787674112D-05	—
-1.008821817773585D-05	2.266554251751908D-05	—	
No. 11	6.192000000000000D+03	0.0	3.53576E-11
	4.096000000000000D+03	0.0	5.39349E-11
	2.048000000000000D+03	0.0	4.56004E-11
	1.024000000000000D+03	0.0	2.97495E-11
	5.120000000000000D+02	0.0	1.77766E-11
	2.560000000000000D+02	0.0	1.11520E-11
	1.280000000000000D+02	0.0	7.83449E-12
	6.400000000000000D+01	0.0	6.03502E-12
	3.200000000000000D+01	0.0	4.81753E-12
	1.600000000000000D+01	0.0	3.74186E-12
	7.9999999999999999D+00	0.0	2.67018E-12
	4.000000000000000D+00	0.0	1.67853E-12
	1.9999999999999999D+00	0.0	9.02603E-13
	1.000000000000000D+00	0.0	4.12288E-13
	5.000000000000000D-01	0.0	1.86500E-13
	2.500000000000000D-01	0.0	1.18710E-13
	1.250000000000000D-01	0.0	8.95999E-14
	6.250000000000000D-02	0.0	5.35170E-14
3.125000000000000D-02	0.0	2.09230E-14	
1.562500000000000D-02	0.0	4.04064E-15	

Table 4.1 The roots and the error bounds obtained with MUACHM and PA07CD for Examples Nos. 1~7

Example	Real part	Imaginary part	Error
No. 1	3.000000000000000D+00	0.0	1.51819E-14
	2.000000000000000D+00	0.0	1.51838E-14
	1.000000000000000D+00	0.0	3.03791E-15
No. 2	2.009999999999999D+00	0.0	9.59058E-12
	1.999999999999994D+00	0.0	9.62724E-12
	1.000000000000000D+00	0.0	4.61868E-14
No. 3	2.900000000000000D+01	0.0	5.55154E-14
	1.500000000000000D+01	0.0	3.63860E-14
	9.999999999999999D-01	-2.000000000000000D+00	5.89039E-15
	1.000000000000000D+00	1.999999999999997D+00	5.89039E-15
No. 4	3.000000863506398D+00	3.822504899257573D-07	—
	2.999998953755965D+00	7.766814992147976D-07	—
	2.999998806625650D+00	-1.073602425882212D-06	—
	-2.000000002522089D+00	-5.531674701203413D-09	—
	-1.99999999373623D+00	1.396045504534262D-09	—
	1.000000000000000D+00	0.0	—
No. 5	-6.000000000000000D+00	0.0	1.01667E-15
	3.000000000000000D+00	4.000000000000000D+00	3.09494E-15
	3.000000000000000D+00	-4.000000000000000D+00	3.09494E-15
	2.000000000000000D+00	0.0	6.47331E-15
	1.000000000000000D+00	-1.000000000000000D+00	2.51742E-15
	1.000000000000000D+00	1.000000000000000D+00	2.51742E-15
No. 6	2.010000000000000D+00	0.0	—
	1.999999999999995D+00	0.0	—
	1.000000000257926D+00	-1.668237853594434D-09	—
	-1.000000000000000D+00	0.0	—
	9.99999989989079D-01	4.256322020260870D-09	—
	5.000000000000000D-01	-5.000000000000000D-01	—
5.000000000000000D-01	5.000000000000000D-01	—	
No. 7	7.999999999999999D+00	0.0	6.51853E-10
	6.999999999999928D+00	0.0	2.43631E-09
	5.999999999999980D+00	0.0	3.66031E-09
	5.000000000000005D+00	0.0	2.82172E-09
	4.000000000000003D+00	0.0	1.17905E-09
	3.000000000000001D+00	0.0	2.58177E-10
	2.000000000000000D+00	0.0	2.59332E-11
	1.000000000000000D+00	0.0	8.27639E-13

Table 4.2 The roots and the error bounds obtained with MUACHM and PA07CD for Examples Nos. 9~11

Example	Real part	Imaginary part	Error
No. 9	1.0000000000000000E+06	0.0	5.49148E-10
	1.5000000000000000E+01	0.0	7.76008E-05
	1.4000000000000000E+01	0.0	4.14823E-04
	1.3000000000000000E+01	0.0	9.40025E-04
	1.1999999999999999E+01	0.0	1.16949E-03
	1.0999999999999999E+01	0.0	3.61777E-04
	9.9999999999999999E+00	0.0	3.75675E-04
	8.9999999999999999E+00	0.0	8.95816E-05
	8.0000000000000000E+00	0.0	8.99937E-06
No.10	-1.0000000000000000E+01	0.0	—
	5.2000061762550550E+00	-3.5263339058203840E-08	—
	5.1999970218989780E+00	-5.4198162452131120E-06	—
	5.1999969185108170E+00	5.7995492219622580E-06	—
	1.2614133893983020E-04	-5.5235108860109840E-06	—
	3.4548618427898430E-05	1.0759421133316790E-04	—
	-1.3711219743321010E-05	-1.0876717142337470E-04	—
	-5.4044275202143100E-05	9.1612445258274580E-05	—
	-8.2883251047904890E-06	-9.1710315937480570E-05	—
-7.6945668232761000E-05	8.3549895090060200E-06	—	
No.11	8.1920000000000000E+03	0.0	4.45170E-11
	4.0960000000000000E+03	0.0	8.10464E-11
	2.0480000000000000E+03	0.0	8.89155E-11
	1.0240000000000000E+03	0.0	7.99197E-11
	5.1199999999999999E+02	0.0	6.41057E-11
	2.5599999999999999E+02	0.0	4.64863E-11
	1.2800000000000000E+02	0.0	3.05984E-11
	8.4000000000000000E+01	0.0	1.85338E-11
	3.2000000000000000E+01	0.0	1.07919E-11
	1.5999999999999999E+01	0.0	6.05702E-12
	7.9999999999999999E+00	0.0	3.30789E-12
	4.0000000000000000E+00	0.0	1.75523E-12
	2.0000000000000000E+00	0.0	9.06053E-13
	1.0000000000000000E+00	0.0	4.56156E-13
	5.0000000000000000E-01	0.0	2.23638E-13
	2.5000000000000000E-01	0.0	1.05605E-13
	1.2500000000000000E-01	0.0	4.66913E-14
6.2500000000000000E-02	0.0	1.81714E-14	
3.1250000000000000E-02	0.0	5.45225E-15	
1.5625000000000000E-02	0.0	9.08644E-16	

Table 5.1 The roots and the error bounds obtained with PA07AD for Examples Nos. 1~7

Example	Real part	Imaginary part	Error
No. 1	3.0000000000000000+00	0.0	1.51800E-14
	2.0000000000000000+00	0.0	1.51800E-14
	1.0000000000000000+00	0.0	3.03600E-15
No. 2	2.0100000000000000+00	0.0	1.21345E-12
	1.0000000000000000+00	0.0	1.21744E-12
	1.0000000000000000+00	0.0	4.52394E-15
No. 3	2.9000000000000000+01	0.0	5.29073E-14
	1.5000000000000000+01	0.0	3.10106E-14
	1.0000000000000000+00	-2.0000000000000000+00	1.24132E-15
	1.0000000000000000+00	2.0000000000000000+00	1.24132E-15
No. 4	2.9999999999999999+00	0.0	2.78904E-05
	-1.9999999999999999+00	0.0	2.75957E-08
	-2.0000000000000000+00	0.0	2.75957E-08
	2.9999999999999999+00	0.0	2.78904E-05
	3.0000000000000000+00	0.0	2.78904E-05
	1.0000000000000000+00	0.0	1.12092E-15
No. 5	3.0000000000000000+00	-4.0000000000000000+00	3.09262E-15
	3.0000000000000000+00	4.0000000000000000+00	3.09262E-15
	-6.0000000000000000+00	0.0	1.01561E-15
	1.0000000000000000+00	1.0000000000000000+00	2.51629E-15
	1.0000000000000000+00	-1.0000000000000000+00	2.51629E-15
	2.0000000000000000+00	0.0	6.47174E-15
No. 6	2.0099999999999999+00	0.0	3.89368E-12
	-1.0000000000000000+00	0.0	1.58066E-16
	5.0000000000000000-01	-5.0000000000000000-01	2.69890E-25
	5.0000000000000000-01	5.0000000000000000-01	2.69890E-25
	2.0000000000000000+00	0.0	3.93057E-12
	9.9999999999999999+00	0.0	1.17292E-07
	1.0000000000000000+00	0.0	1.17292E-07
No. 7	8.0000000000000000+00	0.0	2.60489E-11
	6.0000000000000000+00	0.0	1.27640E-10
	4.0000000000000000+00	0.0	3.50658E-11
	5.0000000000000000+00	0.0	9.11711E-11
	7.0000000000000000+00	0.0	9.11711E-11
	2.0000000000000000+00	0.0	6.37560E-13
	3.0000000000000000+00	0.0	7.01316E-12
	1.0000000000000000+00	0.0	1.82169E-14

Table 5.2 The roots and the error bounds obtained with PA07AD for Examples Nos. 9-11

Example	Real part	Imaginary part	Error
No. 9	1.000000000000000D+06	0.0	5.06594E-10
	1.300000000002433D+01	0.0	1.32555E-07
	8.999999999995860D+00	0.0	1.04557E-08
	1.199999999998301D+01	0.0	1.57797E-07
	1.399999999998551D+01	0.0	6.10204E-08
	1.500000000000329D+01	0.0	1.18877E-08
	1.000000000000736D+01	0.0	4.61249E-08
	8.000000000000753D+00	0.0	9.95957E-10
	1.099999999999989D+01	0.0	1.11040E-07
No.10	-1.000000000000000D+01	0.0	2.24993E-15
	5.199999405469765D+00	-1.029735796194106D-06	5.32601E-05
	5.199999405469765D+00	1.029735796194106D-06	5.32601E-05
	5.200001189060471D+00	0.0	5.32601E-05
	-1.100000000005162D-06	0.0	2.48661E-14
	-1.099999999994838D-06	0.0	2.48661E-14
	1.100000000000000D-06	0.0	2.78949E-22
	2.100000000000000D-08	0.0	5.35042E-24
	-2.099999999999641D-08	0.0	4.75639E-16
-2.100000000000359D-08	0.0	4.75639E-16	
No.11	8.192000000000000D+03	0.0	3.42299E-11
	4.096000000000000D+03	0.0	5.13678E-11
	2.048000000000000D+03	0.0	4.28365E-11
	1.024000000000000D+03	0.0	2.75534E-11
	5.120000000000000D+02	0.0	1.56156E-11
	2.560000000000000D+02	0.0	8.30907E-12
	1.280000000000000D+02	0.0	4.28438E-12
	6.400000000000000D+01	0.0	2.17537E-12
	3.200000000000000D+01	0.0	1.09603E-12
	1.600000000000000D+01	0.0	5.49717E-13
	8.000000000000000D+00	0.0	2.74810E-13
	4.000000000000000D+00	0.0	1.36961E-13
	2.000000000000000D+00	0.0	6.79881E-14
	1.000000000000000D+00	0.0	3.35027E-14
	5.000000000000000D-01	0.0	1.62609E-14
	2.500000000000000D-01	0.0	7.64830E-15
	1.250000000000000D-01	0.0	3.37515E-15
6.250000000000000D-02	0.0	1.31186E-15	
3.125000000000000D-02	0.0	3.93409E-16	
1.562500000000000D-02	0.0	6.55513E-17	

Table 6.1 The roots and the error bounds obtained with BAIR1D and PA07CD for Examples Nos. 1~7

Example	Real part	Imaginary part	Error
No. 1	3.0000000000000000+00	0.0	1.51300E-14
	2.0000000000000000+00	0.0	1.51300E-14
	1.0000000000000000+00	0.0	3.03600E-15
No. 2	2.0100000000000000+00	0.0	1.21545E-12
	2.0000000000000000+00	0.0	1.21744E-12
	1.0000000000000000+00	0.0	4.52394E-15
No. 3	2.9000000000000000+01	0.0	5.29073E-14
	1.5000000000000000+01	0.0	3.10106E-14
	1.0000000000000000+00	2.0000000000000000+00	1.24139E-15
	1.0000000000000000+00	-2.0000000000000000+00	1.24139E-15
No. 4	ILL = 2		
No. 5	-6.0000000000000000+00	0.0	1.01705E-15
	3.0000000000000000+00	-4.0000000000000000+00	3.09649E-15
	3.0000000000000000+00	4.0000000000000000+00	3.09649E-15
	2.0000000000000000+00	0.0	6.47690E-15
	1.0000000000000000+00	1.0000000000000000+00	2.51515E-15
	1.0000000000000000+00	-1.0000000000000000+00	2.51515E-15
No. 6	2.0099999999999999+00	0.0	3.88823E-12
	2.0000000000000000+00	0.0	3.92505E-12
	1.0000000000000000+00	0.0	1.18719E-07
	-1.0000000000000000+00	0.0	1.57624E-16
	9.9999999999999999-01	0.0	1.18719E-07
	5.0000000000000000-01	-5.0000000000000000-01	2.68911E-15
	5.0000000000000000-01	5.0000000000000000-01	2.68911E-15
No. 7	7.9999999999999999+00	0.0	2.51128E-11
	7.0000000000000000+00	0.0	9.13797E-11
	5.9999999999999999+00	0.0	1.27904E-10
	5.0000000000000000+00	0.0	9.13352E-11
	3.9999999999999999+00	0.0	3.51190E-11
	3.0000000000000000+00	0.0	7.02266E-12
	2.0000000000000000+00	0.0	6.33694E-13
	1.0000000000000000+00	0.0	1.82975E-14

Table 6.2 The roots and the error bounds obtained with BAIR1D and PA07CD for Examples Nos. 9~11

Example	Real part	Imaginary part	Error
No. 9	1,000000000000000D+06	0,0	5,06093E-10
	1,5000000000000094D+01	0,0	1,18509E-08
	1,399999999999483D+01	0,0	6,08351E-08
	1,300000000001160D+01	0,0	1,32161E-07
	1,199999999998633D+01	0,0	1,57340E-07
	1,100000000000891D+01	0,0	1,10727E-07
	9,999999999996954D+00	0,0	4,59978E-08
	9,000000000000426D+00	0,0	1,04273E-08
	8,000000000000000D+00	0,0	9,93237E-10
No.10	2,775897117434340D+04	1,010317210302930D+04	—
	2,775897117434340D+04	-1,010317210302930D+04	—
	1,477058319182371D+04	2,558212979779863D+04	—
	1,477058319182371D+04	-2,558212979779863D+04	—
	-5,128781686402819D+03	2,909092464643274D+04	—
	-5,128781686402819D+03	-2,909092464643274D+04	—
	-2,262798947702932D+04	1,898775254340342D+04	—
	-2,262798947702932D+04	-1,898775254340342D+04	—
	-2,953896637273620D+04	0,0	—
	-1,000033854734188D+00	0,0	—
No.11	8,192000000000000D+03	0,0	3,42569E-11
	4,096000000000000D+03	0,0	5,13948E-11
	2,048000000000000D+03	0,0	4,28450E-11
	1,024000000000000D+03	0,0	2,75528E-11
	5,120000000000000D+02	0,0	1,56130E-11
	2,560000000000000D+02	0,0	8,30934E-12
	1,280000000000000D+02	0,0	4,28839E-12
	6,400000000000000D+01	0,0	2,17989E-12
	3,200000000000000D+01	0,0	1,09914E-12
	1,600000000000000D+01	0,0	5,51133E-13
	8,000000000000000D+00	0,0	2,75210E-13
	4,000000000000000D+00	0,0	1,37112E-13
	2,000000000000000D+00	0,0	6,81202E-14
	1,000000000000000D+00	0,0	3,35996E-14
	5,000000000000000D-01	0,0	1,63096E-14
	2,500000000000000D-01	0,0	7,66704E-15
	1,250000000000000D-01	0,0	3,38551E-15
	6,250000000000000D-02	0,0	1,32021E-15
3,125000000000000D-02	0,0	3,98378E-16	
1,562500000000000D-02	0,0	6,69437E-17	

Table 7.1 The roots and the error bounds obtained with JARATD and PA07CD for Examples Nos. 1~7

Example	Real part	Imaginary part	Error
No. 1	3.0000000000000000+00	0.0	1.51972E-14
	2.0000000000000000+00	0.0	1.51953E-14
	1.0000000000000000+00	0.0	3.03791E-15
No. 2	2.0100000000000000+00	0.0	1.21545E-12
	2.0000000000000000+00	0.0	1.21744E-12
	1.0000000000000000+00	0.0	4.52394E-15
No. 3	2.9000000000000000+00	0.0	5.29748E-14
	1.5000000000000000+00	0.0	3.10474E-14
	1.0000000000000000+00	2.0000000000000000+00	1.24161E-15
	1.0000000000000000+00	-2.0000000000000000+00	1.24161E-15
No. 4	3.000000000445932E+00	6.078776063508836E-10	---
	3.000000000445931E+00	6.078775993717677E-10	---
	2.999999999108137E+00	-1.215755205758015E-09	---
	-2.0000000000000000+00	0.0	---
	-1.9999999997963E+00	0.0	---
	1.0000000000000000+00	0.0	---
No. 5	-6.0000000000000000+00	0.0	1.01409E-15
	3.0000000000000000+00	4.0000000000000000+00	3.08724E-15
	3.0000000000000000+00	-4.0000000000000000+00	3.08724E-15
	2.0000000000000000+00	0.0	6.45894E-15
	1.0000000000000000+00	1.0000000000000000+00	2.51159E-15
	1.0000000000000000+00	-1.0000000000000000+00	2.51159E-15
No. 6	2.009999999999998E+00	0.0	---
	2.0000000000000000+00	0.0	---
	1.000000000692571E+00	2.734048414215541E-10	---
	-1.0000000000000000+00	0.0	---
	9.9999999995076594E-01	-2.734048469924697E-10	---
	5.0000000000000000E-01	-5.0000000000000000E-01	---
5.0000000000000000E-01	5.0000000000000000E-01	---	
No. 7	7.999999999999994E+00	0.0	2.60870E-11
	7.0000000000000000E+00	0.0	9.13031E-11
	5.999999999999997E+00	0.0	1.27819E-10
	5.0000000000000000E+00	0.0	9.12922E-11
	3.999999999999993E+00	0.0	3.51077E-11
	3.0000000000000000E+00	0.0	7.02005E-12
	2.0000000000000000E+00	0.0	6.37994E-13
	1.0000000000000000E+00	0.0	1.82254E-14

Table 7.2 The roots obtained with JARATD for Examples Nos. 9~11

Example	Real part	Imaginary part
No. 9	1.000000000000000D+06	3.990523769626639D-59
	8.000000000000000D+00	3.990523769626639D-59
	9.000000000000000D+00	3.990523769626639D-59
	1.000000000000000D+01	3.990523769626639D-59
	1.100000000000000D+01	3.990523769626639D-59
	1.200000000000000D+01	3.990523769626639D-59
	1.300000000000000D+01	3.990523769626639D-59
	1.400000000000000D+01	3.990523769626639D-59
	1.500000000000000D+01	5.732968047427759D-59
No.10	-9.406971114182650D+00	-5.185067768156051D-01
	6.253153839490055D+00	-1.224753438827893D+00
	5.973956064371962D+00	1.340512584031844D+00
	2.779861209622578D+00	4.027476291557763D-01
	-1.809534228766255D-06	2.217815816708027D-07
	8.469685229781700D-07	-6.296293039820081D-08
	-7.297369969513529D-07	-7.680796894967266D-09
	7.079703515230321D-07	6.415594150929212D-08
	4.246384931690317D-08	-2.469585307046092D-07
-1.784334421784018D-07	3.412061216970029D-08	
No.11	1.562500000000000D-02	-5.185067768156051D-01
	3.125000000000000D-02	-1.224753438827893D+00
	6.250000000000000D-02	1.340512584031844D+00
	1.250000000000000D-01	4.027476291557763D-01
	2.500000000000000D-01	2.217815816708027D-07
	5.000000000000000D-01	-6.296293039820081D-08
	1.000000000000000D+00	-7.680796894967266D-09
	2.000000000000000D+00	6.415594150929212D-08
	4.000000000000000D+00	-2.469585307046092D-07
	8.000000000000000D+00	3.412061216970029D-08
	1.600000000000000D+01	1.879696319340326D-52
	3.200000000000000D+01	1.539847222431688D-50
	6.400000000000000D+01	1.539847222431688D-50
	1.280000000000000D+02	1.539847222431688D-50
	2.560000000000000D+02	1.539847222431688D-50
	5.120000000000000D+02	1.539847222431688D-50
1.024000000000000D+03	1.539847222431688D-50	
2.048000000000000D+03	1.539847222431688D-50	
4.096000000000000D+03	1.539847222431688D-50	
8.192000000000000D+03	1.539847222431688D-50	

Table 8 Comparison of computation times (CPU msec) required for root findings and error bound calculations by various subroutines for real coefficients polynomials

Example	MULLRA		MUACHM		PAO7AD		BAIRLD		JARATD	
	Root	Error	Root	Error	Root	Error	Root	Error	Root	Error
No. 1	2	2	1	2	4	2	0	2	5	3
No. 2	2	2	1	3	4	2	0	3	7	3
No. 3	3	3	3	4	3	3	1	3	12	4
No. 4	21 ¹⁾	11 ²⁾	17 ¹⁾	10 ²⁾	11	6	2)	-	45	11 ²⁾
No. 5	6	5	6	6	8	5	2	5	35	5
No. 6	10	6	10	14 ²⁾	8	7	4	7	22	14 ²⁾
No. 7	19	7	16	7	11	6	4	7	71	7
No. 9	34	11	28	8	12	6	6	7	354 ⁴⁾	-
No.10	57 ¹⁾	24 ²⁾	47 ¹⁾	20 ²⁾	16	10	476 ³⁾	25 ²⁾	11 ¹⁾	18 ²⁾
No.11	77	27	71	23	32	19	17	19	644 ⁴⁾	-

1) Not completely correct, 2) Not found, 3) Wrong results (overflows), 4) Not normal return (ILL=1)

計算は収束していない。Example No10 については3重根は何とか求められているが、原点付近にある2重根 -1.1×10^{-6} は $-(0.6 \pm i) \times 10^{-4}$ 程度で、 1.1×10^{-6} が $(1 - 0.04i) \times 10^{-4}$ 程度、もう一つの2重根 -2.1×10^{-8} は $(0.15 \pm i) \times 10^{-4}$ 程度で、 2.1×10^{-8} が $-(1-2i) \times 10^{-5}$ 程度に求められている。

Chambers のアルゴリズムを取り入れた付録3に示したMUACHM(メイン・プログラムはMULLRAに対するのと同じ)による計算時間は、Table 8にみられるようにMULLRAのより短縮されている。計算精度の方は同程度だが、Example No6に不要な虚数部が残ったため、根の誤差限界は求められていない。Example No10もMULLRAのと同程度の収束状態である。

Madsen アルゴリズムによるNewton法の付録5のPA07ADでは、Example No10の5.2の3重根の二つが $(5.2 \pm 10^{-6}i)$ と求められている以外は、一応全て正確に求められている。計算時間も低次多項式に対しては他よりいくらか長いが、10次近くでMULLRA、MUACHMの半分程度以下になっている。Bairstow法のBAIR1Dは、 $EPS=10^{-10}$ を用いたがExample No4とNo10以外は、Table 8に示した五つの手法のうちで最も速く根が求められている。これらの中にはNo6のように2重根や近接根もあるが、問題を起していない。しかしNo4は、解が求められない状態($ILL=2$)でサブルーチンから戻っており、No10はでたらしめな値が求められている。

最後の3重根あるいは三つの近接根なども求められる手法であるJarrat法(標準のEPSを使用)は、確かにExample No4の3重根は一応求められているが、No10の方は5.2の3重根が $(6 \pm i)$ 程度と $(3+0.4i)$ 程度に求められ、0点近傍の根は $10^{-6} \sim 10^{-8}$ の値に求められているにすぎない。このNo10の計算は、Table 8から分るように非常に短時間で、正常な状態でサブルーチンから戻っている。No9と11は $ILL=1$ で戻っているが、No9の方は正解が得られている。No11の方は実数部のみならば正解を与えているが、実数部の小さい根には大きい虚数部が付いている。

なお、5次以下の実係数代数方程式を解くため、FACOMのSSL II¹⁴⁾には倍精度ルーチンとしてDLOWPが用意されている。この呼び出しは

CALL DLOWP(A, N, Z, ICON)

で、倍精度配列A(N+1)に(2)式の係数を a_0 から入れ、Nは次数、倍精度複素数配列Z(N)に根がセットされ、ICONは0ならばエラーなしで、10000は50回反復でも根が求められなかった場合、30000は入力エラーの場合である。Example No1~3にこれを適用した結果は、根を求めるために要した時間はいずれも1 msecで、根はPA07ADによるのと同じの値が得られている。

次に複素係数多項式に対する六つのルーチンによるテスト結果をそれぞれTable 9から14に示した。またこれらの計算時間はTable 15にまとめた。この際も付録2のメイン・プログラムに示されているように、各ルーチンで根を求めた後、その実数部と虚数部を比較し、一方の絶対値が他方の 10^{-10} 以下のときにはそれを0とし、そして根の順番を絶対値の大きい順に並べかえてPA06CD(付録6参照)で根の誤差限界を求めた。

付録2に示したMuller法のMULLRBによるTable 9の結果は、実係数のMULLRAのと

Table 9.1 The roots and the error bounds obtained with MULLRB and PA06CD for Examples Nos. 1~8

Example	Real part	Imaginary part	Error
No. 1	3.000000000000000D+00	0.0	1.52048E-14
	2.000000000000000D+00	0.0	1.52220E-14
	1.000000000000000D+00	0.0	3.05317E-15
No. 2	2.010000000000000D+00	0.0	4.65733E-12
	2.000000000000000D+00	0.0	4.67700E-12
	9.999999999999977D-01	0.0	2.42180E-14
No. 3	2.900000000000000D+01	0.0	9.33032E-14
	1.500000000000000D+01	0.0	7.57329E-14
	1.000000000000000D+00	2.000000000000002D+00	9.34227E-15
	1.000000000000000D+00	-2.000000000000000D+00	4.34227E-15
No. 4	3.000000690574752D+00	-2.09623+329151324D-07	3.29037E-02
	2.99999978287821D+00	1.718975559380681D-06	3.29037E-02
	2.999998791839431D+00	-1.612542703105094D-08	3.29037E-02
	-2.000000000310137D+00	0.0	1.21662E-03
	-1.99999999533189D+00	0.0	1.21662E-03
	1.000000000000000D+00	0.0	2.32862E-06
No. 5	-6.000000000000000D+00	0.0	1.01684E-15
	3.000000000000000D+00	4.000000000000000D+00	3.09565E-15
	3.000000000000000D+00	-4.000000000000000D+00	3.09565E-15
	2.000000000000000D+00	0.0	6.47792E-15
	1.000000000000000D+00	-1.000000000000000D+00	2.51995E-15
	1.000000000000000D+00	1.000000000000000D+00	2.51995E-15
No. 6	2.009999999999998D+00	0.0	7.49476E-06
	2.000000000000002D+00	0.0	7.59204E-06
	1.000000001980502D+00	0.0	2.01144E-04
	-1.000000000000000D+00	0.0	4.66998E-10
	9.999999993364434D-01	0.0	2.01144E-04
	5.000000000000000D-01	5.000000000000000D-01	9.31304E-09
5.000000000000000D-01	-5.000000000000000D-01	9.31304E-09	
No. 7	7.999999999999994D+00	0.0	3.96480E-11
	6.999999999999985D+00	0.0	1.47376E-10
	5.999999999999995D+00	0.0	2.22267E-10
	5.0000000000000027D+00	0.0	1.74089E-10
	3.999999999999997D+00	0.0	7.50460E-11
	2.99999999999999D+00	0.0	1.72751E-11
	2.000000000000000D+00	0.0	1.86535E-12
	1.000000000000000D+00	0.0	6.56283E-14
No. 8	4.000000000000007D+00	2.99999999999994D+00	3.25478E-06
	4.000000000000000D+00	-3.000000000000000D+00	2.82081E-10
	3.999000000000007D+00	3.000000000000000D+00	3.25509E-06
	1.000000001215111D+00	1.000000000125148D+00	2.14666E-05
	9.999999987427743D-01	9.99999998698642D-01	2.14666E-05

Table 9.2 The roots and the error bounds obtained with MULLRB and PA06CD for Examples Nos. 9~11

Example	Real part	Imaginary part	Error
No. 9	1.0000000000000000D+06	0.0	6.07123E-10
	1.5000000000000289D+01	0.0	8.70703E-05
	1.400000000000590D+01	0.0	4.64077E-04
	1.299999999997194D+01	0.0	1.04832E-03
	1.2000000000007529D+01	0.0	1.29977E-03
	1.100000000000024D+01	0.0	9.54249E-04
	9.999999999995988D+00	0.0	4.14320E-04
	9.0000000000005648E+00	0.0	9.83557E-05
	8.000000000000000D+00	0.0	9.83463E-06
No.10	-1.000000000000000D+01	0.0	6.52899E-05
	5.200006358477544D+00	-1.724546575483175D-08	2.38410E-01
	5.199996890206135D+00	5.265270708222738D-06	2.38410E-01
	5.199996726609440D+00	-5.638957241856744D-06	2.38411E-01
	2.780791910592487E-05	1.270877841253675D-04	---
	-4.317235325070424D-05	1.200756664749238D-04	---
	1.189638677752060E-04	4.210554723712334D-05	---
	-5.906091991206149D-05	-1.103038565047709D-04	---
	9.416379146986450D-05	-7.875835720448627D-05	---
-1.111315905640135D-04	-3.900466401313947D-05	---	
No.11	8.192000000000000D+02	0.0	3.54684E-11
	4.096000000000000D+03	0.0	5.41978E-11
	2.048000000000000D+03	0.0	4.58796E-11
	1.024000000000000D+03	0.0	2.98945E-11
	5.120000000000000D+02	0.0	1.77935E-11
	2.560000000000000D+02	0.0	1.11018E-11
	1.280000000000000D+02	0.0	7.76723E-12
	6.400000000000000D+01	0.0	5.97420E-12
	3.200000000000000D+01	0.0	4.77194E-12
	1.600000000000000D+01	0.0	3.71192E-12
	7.999999999999964D+00	0.0	2.65256E-12
	4.000000000000003D+00	0.0	1.66930E-12
	1.999999999999996D+00	0.0	8.99027E-13
	1.000000000000000D+00	0.0	4.11092E-13
	5.000000000000000D-01	0.0	1.86145E-13
	2.500000000000000D-01	0.0	1.18564E-13
1.250000000000028D-01	0.0	8.95141E-14	
6.250000000000000D-02	0.0	5.34735E-14	
3.125000000000000D-02	0.0	2.09081E-14	
1.562500000000000D-02	0.0	4.03797E-15	

Table 10.1 The roots and the error bounds obtained with MUBCHM and PA06CD for Examples Nos. 1~8

Example	Real part	Imaginary part	Error
No. 1	3.0000000000000000+00	0.0	1.60501E-14
	2.0000000000000000+00	0.0	1.62677E-14
	1.0000000000000000+00	0.0	3.32605E-15
No. 2	2.0100000000000000+00	0.0	3.68549E-12
	2.0000000000000020+00	0.0	3.69771E-12
	1.0000000000000000+00	0.0	1.68157E-14
No. 3	2.9000000000000000+01	0.0	7.86339E-14
	1.5000000000000000+01	0.0	5.95575E-14
	1.0000000000000000+00	2.0000000000000000+00	6.55071E-15
	1.0000000000000000+00	-1.9999999999999970+00	6.55071E-15
No. 4	3.0000010521478190+00	6.0648875186157540-08	1.45525E-02
	2.9999999803919090+00	9.2890556951680810-07	1.45524E-02
	2.9999989851793900+00	-6.5201351183258640-07	1.45526E-02
	-2.0000000025759020+00	-5.5888266672884430-09	3.61576E-04
	-1.999999992695180+00	1.3241529832383600-09	3.61576E-04
	1.0000000000000000+00	0.0	2.11889E-07
No. 5	-6.0000000000000000+00	0.0	1.01653E-15
	3.0000000000000000+00	4.0000000000000000+00	3.09670E-15
	3.0000000000000000+00	-4.0000000000000000+00	3.09670E-15
	2.0000000000000000+00	0.0	6.50076E-15
	1.0000000000000000+00	-1.0000000000000000+00	2.53100E-15
	1.0000000000000000+00	1.0000000000000000+00	2.53100E-15
No. 6	2.0099999999999960+00	0.0	6.43837E-06
	1.9999999999999950+00	0.0	6.52193E-06
	1.0000000001119590+00	-4.7122217431950140-10	1.86430E-04
	-1.0000000000000000+00	0.0	4.01174E-10
	9.9999999970424060-01	1.5875125023387740-09	1.86430E-04
	5.0000000000000000-01	5.0000000000000000-01	8.00036E-09
5.0000000000000000-01	-5.0000000000000000-01	8.00036E-09	
No. 7	8.0000000000000050+00	0.0	5.77444E-10
	7.00000000000001340+00	0.0	2.14841E-09
	5.9999999999999480+00	0.0	3.21096E-09
	4.9999999999999880+00	0.0	2.46040E-09
	4.0000000000000020+00	0.0	1.02083E-09
	3.0000000000000010+00	0.0	2.21676E-10
	2.0000000000000000+00	0.0	2.20478E-11
	1.0000000000000000+00	0.0	6.95506E-13
No. 8	4.0000000000000000+00	-3.0000000000000000+00	3.05235E-08
	3.9999999999999670+00	3.0000000000000330+00	1.26907E-03
	3.9989999999999640+00	3.0000000000000110+00	1.26908E-03
	9.9999999960781020-01	1.0000000008670980+00	2.23278E-04
	1.00000000012122900+00	9.9999999461851750-01	2.23278E-04

Table 10.2 The roots and the error bounds obtained with MUBCHM and PA06CD for Examples Nos. 9~11

Example	Real part	Imaginary part	Error
No. 9	1,0000000000000000D+06	0.0	5.81056E-10
	1,5000000000000155D+01	0.0	6.52107E-05
	1,400000000001115D+01	0.0	3.47690E-04
	1,29999999998673D+01	0.0	7.85703E-04
	1,2000000000000035D+01	0.0	9.74558E-04
	1,100000000004658D+01	0.0	7.15799E-04
	9.99999999983436D+00	0.0	3.10934E-04
	9,000000000001078D+00	0.0	7.38573E-05
	7.9999999999976D+00	0.0	7.38828E-06
No.10	-1,0000000000000000D+01	0.0	1.06221E-05
	5,200006329943319D+00	0.0	1.16998E-01
	5,199997085825081D+00	-5.748713092433722D-06	1.16999E-01
	5,199996902105422D+00	5.646165165656582D-06	1.16999E-01
	-2.932891346622412D-06	-1.286906263884199D-04	—
	1.263752374226593D-04	-5.272477528498482D-06	—
	3,208927630098670D-05	1.107420869623732D-04	—
	-5,404427820214310D-05	9.161244525827458D-05	—
	-1,013799961812329D-04	-1.046722694351536D-05	—
9,428414997120889D-06	-5.782687101497294D-05	—	
No.11	8,1920000000000000D+03	0.0	4.42979E-11
	4,0960000000000000D+03	0.0	8,05048E-11
	2,0480000000000000D+03	0.0	8,82402E-11
	1,0240000000000000D+03	0.0	7,93097E-11
	5,1199999999999998D+02	0.0	6,36330E-11
	2,5599999999999996D+02	0.0	4,61482E-11
	1,2800000000000000D+02	0.0	3,03683E-11
	6,4000000000000000D+01	0.0	1,84830E-11
	3,2000000000000000D+01	0.0	1,06979E-11
	1,5999999999999998D+01	0.0	6,00236E-12
	7,9999999999999999D+00	0.0	3,27876E-12
	4,0000000000000000D+00	0.0	1,74105E-12
	2,0000000000000000D+00	0.0	8,99680E-13
	1,0000000000000000D+00	0.0	4,53532E-13
	5,0000000000000000D-01	0.0	2,22680E-13
	2,5000000000000000D-01	0.0	1,05309E-13
	1,2500000000000000D-01	0.0	4,66164E-14
6,2500000000000000D-02	0.0	1,81566E-14	
3,1250000000000000D-02	0.0	5,45040E-15	
1,5625000000000000D-02	0.0	9,06563E-16	

Table 11.1 The roots and the error bounds obtained with CNWIND and PA06CD for Examples Nos. 1~8

Example	Real part	Imaginary part	Error
No. 1	3.0000000000000000+00	0.0	1.51800E-14
	2.0000000000000000+00	0.0	1.51800E-14
	1.0000000000000000+00	0.0	3.03600E-15
No. 2	2.0099999999999999+00	0.0	1.21545E-12
	2.0000000000000000+00	0.0	1.21744E-12
	1.0000000000000000+00	0.0	4.52394E-15
No. 3	2.9900000000000000+01	0.0	1.73895E-01
	1.5000000000000000+01	0.0	1.89357E-01
	1.0000000000000000+00	-2.0000000000000000+00	1.14105E-02
	1.0000000000000000+00	2.0000000000000000+00	1.14105E-02
No. 4	ILL = 2 1.0000000000000000+00		—
No. 5	-5.0000000000000000+00	0.0	—
	5.0000000000000000+00	-4.0000000000000000+00	—
	3.0000000000000000+00	4.0000000000000000+00	—
	2.0000000000000000+00	0.0	—
	1.0000000000000000+00	-1.0000000000000000+00	—
No. 6	ILL = 3 5.0000000000000000-01		—
	5.0000000000000000-01	5.0000000000000000-01	—
No. 7	7.9999999999999999+00	0.0	2.60710E-11
	7.0000000000000000+00	0.0	9.12526E-11
	5.9999999999999999+00	0.0	1.27770E-10
	5.0000000000000000+00	0.0	9.12895E-11
	3.9999999999999999+00	0.0	3.51303E-11
	3.0000000000000000+00	0.0	7.03337E-12
	2.0000000000000000+00	0.0	6.40735E-13
1.0000000000000000+00	0.0	1.83959E-14	
No. 8	ILL = 1		

Table 11.2 The roots and the error bounds obtained with CNWTND and PA06CD for Examples Nos. 9-11

Example	Real part	Imaginary part	Error
No. 9	1.0000000000000000+06	0.0	5.06593E-10
	1.5000000000000428D+01	0.0	1.18742E-08
	1.399999999998080D+01	0.0	6.09540E-08
	1.30000000003577D+01	0.0	1.32417E-07
	1.199999999996416D+01	0.0	1.57637E-07
	1.10000000002105D+01	0.0	1.10929E-07
	9.99999999992519D+00	0.0	4.60777E-08
	9.00000000001606D+00	0.0	1.04443E-08
7.99999999999822D+00	0.0	9.94731E-10	
No.10	ILL = 1		
No.11	8.19200000000000D+03	0.0	3.42587E-11
	4.09600000000000D+03	0.0	5.14373E-11
	2.04800000000000D+03	0.0	4.29188E-11
	1.02400000000000D+03	0.0	2.76314E-11
	5.12000000000000D+02	0.0	1.56749E-11
	2.56000000000000D+02	0.0	8.34410E-12
	1.28000000000000D+02	0.0	4.30168E-12
	6.40000000000000D+01	0.0	2.18302E-12
	3.20000000000000D+01	0.0	1.09909E-12
	1.60000000000000D+01	0.0	5.50687E-13
	8.00000000000000D+00	0.0	2.75111E-13
	4.00000000000000D+00	0.0	1.37160E-13
	2.00000000000000D+00	0.0	6.80974E-14
	1.00000000000000D+00	0.0	3.35567E-14
	5.00000000000000D-01	0.0	1.63256E-14
	2.50000000000000D-01	0.0	7.72722E-15
1.25000000000000D-01	0.0	3.46947E-15	
6.25000000000000D-02	0.0	1.41731E-15	
3.12500000000000D-02	0.0	4.68170E-16	
1.56250000000000D-02	0.0	8.83689E-17	

Table 12.1 The roots and the error bounds obtained with PA06AD for Examples Nos. 1~8

Example	Real part	Imaginary part	Error
No. 1	3.000000000000000D+00	0.0	1.51800E-14
	2.000000000000000D+00	0.0	1.51800E-14
	1.000000000000000D+00	0.0	3.03600E-15
No. 2	2.010000000000000D+00	0.0	1.21545E-12
	2.000000000000000D+00	0.0	1.21744E-12
	1.000000000000000D+00	0.0	4.52394E-15
No. 3	1.500000000000000D+01	0.0	3.10106E-14
	2.900000000000000D+01	0.0	5.29073E-14
	1.000000000000000D+00	-2.000000000000000D+00	1.24139E-15
	1.000000000000000D+00	2.000000000000000D+00	1.24139E-15
No. 4	2.99999999846103D+00	0.0	2.78682E-05
	-1.9999999998118D+00	0.0	2.75547E-08
	2.99999999849387D+00	0.0	2.78682E-05
	-2.00000000001882D+00	0.0	2.75547E-08
	3.00000000304510D+00	0.0	2.78682E-05
	1.000000000000000D+00	0.0	1.11834E-15
No. 5	-6.000000000000000D+00	0.0	1.01409E-15
	3.000000000000000D+00	4.000000000000000D+00	3.08724E-15
	3.000000000000000D+00	-4.000000000000000D+00	3.08724E-15
	1.000000000000000D+00	-1.000000000000000D+00	2.51159E-15
	1.000000000000000D+00	1.000000000000000D+00	2.51159E-15
	2.000000000000000D+00	0.0	6.45894E-15
No. 6	2.009999999999999D+00	0.0	3.89660E-12
	2.000000000000001D+00	0.0	3.93552E-12
	-1.000000000000000D+00	0.0	1.58041E-16
	5.000000000000000D-01	-5.000000000000000D-01	2.69614E-15
	5.000000000000000D-01	5.000000000000000D-01	2.69614E-15
	1.000000000012337D+00	0.0	1.17036E-07
	9.99999999876634D-01	0.0	1.17036E-07
No. 7	8.000000000000000D+00	0.0	2.60509E-11
	6.000000000000001D+00	0.0	1.27665E-10
	6.999999999999999D+00	0.0	9.11822E-11
	3.000000000000000D+00	0.0	7.02137E-12
	3.999999999999999D+00	0.0	3.50873E-11
	5.000000000000000D+00	0.0	9.12021E-11
	1.000000000000000D+00	0.0	1.83090E-14
	2.000000000000000D+00	0.0	6.39080E-13
No. 8	3.99999999999927D+00	3.00000000000044D+00	1.11809E-10
	4.000000000000000D+00	-3.000000000000000D+00	9.67276E-15
	3.999000000000073D+00	2.99999999999956D+00	1.11805E-10
	9.999999999688786D-01	1.000000000084193D+00	8.37839E-08
	1.000000000031121D+00	9.99999999158072D-01	8.37839E-08

Table 12.2 The roots and the error bounds obtained with PA06AD for Examples Nos. 9~11

Example	Real part	Imaginary part	Error
No. 9	1.000000000000000D+06	0.0	5.06593E-10
	1.500000000000210D+01	0.0	1.18591E-08
	1.19999999998530D+01	0.0	1.57478E-07
	8.00000000000303D+00	0.0	9.94518E-10
	1.39999999999056D+01	0.0	6.08808E-08
	8.99999999998610D+00	0.0	1.04391E-08
	1.300000000001697D+01	0.0	1.32269E-07
	1.00000000000117D+01	0.0	4.60446E-08
	1.10000000000499D+01	0.0	1.10831E-07
	No.10	-1.00000000000000D+01	0.0
5.199999405470967D+00		-1.029726705008828D-06	5.32935E-05
5.199999405470967D+00		1.029726705008828D-06	5.32935E-05
5.200001189058065D+00		0.0	5.32935E-05
-1.09999999999794D-06		0.0	2.48615E-14
1.10000000000000D-06		0.0	2.79069E-22
-1.10000000000206D-06		0.0	2.48615E-14
2.10000000000000D-08		0.0	5.34582E-24
-2.09999999999522D-08		0.0	4.75437E-16
-2.10000000000478D-08		0.0	4.75437E-16
No.11	4.09600000000000D+03	0.0	5.13872E-11
	8.19200000000000D+03	0.0	3.42560E-11
	2.04800000000000D+03	0.0	4.28224E-11
	1.02400000000000D+03	0.0	2.75219E-11
	5.12000000000000D+02	0.0	1.55913E-11
	2.56000000000000D+02	0.0	8.30188E-12
	1.28000000000000D+02	0.0	4.28794E-12
	6.40000000000000D+01	0.0	2.17934E-12
	3.20000000000000D+01	0.0	1.09703E-12
	1.60000000000000D+01	0.0	5.49478E-13
	8.00000000000000D+00	0.0	2.74656E-13
	4.00000000000000D+00	0.0	1.36954E-13
	2.00000000000000D+00	0.0	6.79854E-14
	1.00000000000000D+00	0.0	3.34939E-14
	5.00000000000000D-01	0.0	1.62512E-14
	2.50000000000000D-01	0.0	7.63689E-15
	1.25000000000000D-01	0.0	3.36851E-15
	6.25000000000000D-02	0.0	1.30981E-15
3.12500000000000D-02	0.0	3.92847E-16	
1.56250000000000D-02	0.0	6.54433E-17	

Table 13.1 The roots and the error bounds obtained with CJARTD and PA06CD for Examples Nos. 1~8

Example	Real part	Imaginary part	Error
No. 1	3.0000000000000000+00	0.0	1.52143E-14
	2.0000000000000000+00	0.0	1.52135E-14
	1.0000000000000000+00	0.0	3.03982E-15
No. 2	2.0100000000000000+00	0.0	1.21582E-12
	2.0000000000000000+00	0.0	1.21782E-12
	1.0000000000000000+00	0.0	4.52772E-15
No. 3	2.9000000000000000+00	0.0	5.30123E-14
	1.5000000000000000+00	0.0	3.10876E-14
	1.0000000000000000+00	-2.0000000000000000+00	1.24446E-15
No. 4	3.00000005216668E+00	-8.006883575702670D-09	3.16728E-05
	2.999999984076295D+00	4.652986476795700D-08	3.16642E-05
	2.999999917070370D+00	-3.852296816435021D-08	3.16728E-05
	-2.0000000000000000+00	0.0	3.65327E-06
	-2.0000000000000000+00	0.0	3.65327E-06
	1.000000000000001D+00	0.0	2.56245E-15
No. 5	-6.0000000000000000+00	0.0	1.02005E-15
	3.0000000000000000+00	4.0000000000000000+00	3.11028E-15
	3.0000000000000000+00	-4.0000000000000000+00	3.11028E-15
	2.0000000000000000+00	0.0	6.52224E-15
	1.0000000000000000+00	-1.0000000000000000+00	2.53870E-15
1.0000000000000000+00	1.0000000000000000+00	2.53870E-15	
No. 6	2.009999999999993D+00	0.0	3.89477E-12
	2.0000000000000007D+00	0.0	3.93103E-12
	1.0000000000008899D+00	0.0	1.17022E-07
	-1.0000000000000000+00	0.0	1.58023E-16
	0.9999999999913013D-01	0.0	1.17022E-07
	5.0000000000000000E-01	-5.0000000000000000D-01	2.70021E-15
5.0000000000000000E-01	5.0000000000000000D-01	2.70021E-15	
No. 7	7.99999999999999D+00	0.0	2.61006E-11
	7.000000000000024D+00	0.0	9.13544E-11
	5.99999999999981D+00	0.0	1.27898E-10
	5.00000000000004D+00	0.0	9.13544E-11
	4.00000000000007D+00	0.0	3.51346E-11
	3.00000000000000D+00	0.0	7.02645E-12
	2.00000000000000D+00	0.0	6.38625E-13
	1.00000000000000D+00	0.0	1.82757E-14
No. 8	4.0000000000000000+00	-3.0000000000000000+00	—
	4.0000000000000000+00	3.0000000000000000+00	—
	3.9990000000000000+00	3.0000000000000000+00	—
	1.000000006675609D+00	9.999999984947896D-01	—
9.999999933243908D-01	1.00000001505210D+00	—	

Table 14.1 The roots and the error bounds obtained with DCJART and PA06CD for Examples Nos. 1-8

Example	Real part	Imaginary part	Error
No. 1	3.000000000000000D+00	0.0	1.51914E-14
	2.000000000000000D+00	0.0	1.51953E-14
	1.000000000000000D+00	0.0	3.03982E-15
No. 2	2.009999999999999D+00	0.0	1.21697E-12
	2.000000000000000D+00	0.0	1.21896E-12
	1.000000000000000D+00	0.0	4.52772E-15
No. 3	2.900000000000000D+01	0.0	5.29073E-14
	1.500000000000000D+01	0.0	3.10106E-14
	1.000000000000000D+00	2.000000000000000D+00	1.24139E-15
	1.000000000000000D+00	-2.000000000000000D+00	1.24139E-15
No. 4	3.000001666457653D+00	0.0	2.97078E-05
	2.999999166771549D+00	1.443190438085828D-06	2.97078E-05
	2.999999166770798D+00	-1.443190438086262D-06	2.97078E-05
	-2.000000002950829D+00	0.0	3.09905E-08
	-1.999999997049171D+00	0.0	3.09905E-08
	1.000000000000000D+00	0.0	1.14378E-15
No. 5	-6.000000000000000D+00	0.0	1.01742E-15
	3.000000000000000D+00	-4.000000000000000D+00	3.09738E-15
	3.000000000000000D+00	4.000000000000000D+00	3.09738E-15
	2.000000000000000D+00	0.0	6.47684E-15
	1.000000000000000D+00	1.000000000000000D+00	2.51856E-15
	1.000000000000000D+00	-1.000000000000000D+00	2.51856E-15
No. 6	2.009999999999999D+00	0.0	3.90357E-12
	2.000000000000000D+00	0.0	3.94058E-12
	1.000000000440885D+00	0.0	1.18101E-07
	-1.000000000000000D+00	0.0	1.58602E-16
	9.999999995591147D-01	0.0	1.18101E-07
	5.000000000000000D-01	-5.000000000000000D-01	2.71004E-15
5.000000000000000D-01	5.000000000000000D-01	2.71004E-15	
No. 7	7.999999999999976D+00	0.0	2.61006E-11
	7.0000000000000079D+00	0.0	9.13543E-11
	5.9999999999999698D+00	0.0	1.27897E-10
	5.000000000000065D+00	0.0	9.13538E-11
	3.999999999999978D+00	0.0	3.51340E-11
	3.000000000000004D+00	0.0	7.02606E-12
	2.000000000000000D+00	0.0	6.38694E-13
	1.000000000000000D+00	0.0	1.82571E-14
No. 8	4.000000000000000D+00	-3.000000000000000D+00	—
	3.999999999999965D+00	3.000000000000017D+00	—
	3.999000000000035D+00	2.999999999999983D+00	—
	1.00000000000016203D+00	1.0000000000006727D+00	—
	9.9999999983797D-01	9.99999999932735D-01	—

Table 14.2 The roots and the error bounds obtained with DCJART and PA06CD for Examples Nos. 9~11

Example	Real part	Imaginary part	Error
No. 9	1.0000000000000000+06	0.0	5.07094E-10
	1.4999999999999920+01	0.0	1.18853E-08
	1.3999999999999060+01	0.0	6.10154E-08
	1.30000000000005290+01	0.0	1.32561E-07
	1.19999999999989750+01	0.0	1.57826E-07
	1.10000000000009700+01	0.0	1.11076E-07
	9.9999999999953180+00	0.0	4.61458E-08
	9.0000000000010150+00	0.0	1.04616E-08
	7.99999999999450+00	0.0	9.96571E-10
No.10	-1.0000000000000000+01	0.0	2.25503E-15
	5.2000063657248580+00	0.0	5.89940E-05
	5.1999968171375710+00	5.5128822787393460-06	5.89940E-05
	5.1999968171375710+00	-5.5128822787393460-06	5.89940E-05
	-1.1000000000984870-06	0.0	2.49604E-14
	1.1000000000000000-06	0.0	2.76882E-22
	-1.0999999999015130-06	0.0	2.49604E-14
	-2.1000000000694990-08	0.0	4.74781E-16
	2.1000000000000000-06	0.0	5.30012E-24
-2.0999999999305010-08	0.0	4.74781E-16	
No.11	8.1920000000000000+03	0.0	3.42842E-11
	4.0960000000000000+03	0.0	5.14787E-11
	2.0480000000000000+03	0.0	4.29228E-11
	1.0240000000000000+03	0.0	2.76082E-11
	5.1199999999999990+02	0.0	1.56598E-11
	2.5600000000000000+02	0.0	8.33842E-12
	1.2800000000000000+02	0.0	4.29911E-12
	6.4000000000000000+01	0.0	2.18153E-12
	3.2000000000000000+01	0.0	1.09794E-12
	1.6000000000000000+01	0.0	5.50056E-13
	7.9999999999999990+00	0.0	2.75073E-13
	4.0000000000000000+00	0.0	1.37408E-13
	2.0000000000000000+00	0.0	6.83280E-14
	1.0000000000000000+00	0.0	3.36520E-14
	5.0000000000000000-01	0.0	1.63021E-14
	2.5000000000000000-01	0.0	7.65239E-15
	1.2500000000000000-01	0.0	3.37588E-15
6.2499999999999990-02	0.0	1.32007E-15	
3.1250000000000010-02	0.0	4.06993E-16	
1.5624999999999990-02	0.0	7.25042E-17	

Table 15 Comparison of computation times (CPU msec) required for root findings and error bound calculations by various subroutines for complex coefficients polynomials

Example	MULLRB		MUBCHM		CNWIND		PA06AD		CJARTD		DCJART	
	Root	Error	Root	Error	Root	Error	Root	Error	Root	Error	Root	Error
No. 1	2	3	1	3	5	3	3	3	7	3	6	3
No. 2	2	3	2	2	6	3	5	2	9	3	6	3
No. 3	3	4	3	4	5	4	5	4	9	4	6	4
No. 4	21 ¹⁾	8	18 ¹⁾	8	2)	-	12	7	22	8	24 ¹⁾	8
No. 5	8	7	7	6	9	17 ³⁾	9	6	16	7	18	7
No. 6	11	8	11	8	2)	-	12	7	30	8	32	8
No. 7	19	9	16	9	26	9	12	7	44	9	44	9
No. 8	8	6	8	6	3)	-	10	5	15	14 ³⁾	29	14 ³⁾
No. 9	35	11	29	10	43	10	16	8	51	-	125	10
No.10	79 ¹⁾	27 ²⁾	64 ¹⁾	27 ²⁾	3)	-	17	13	2)	-	51	15
No.11	79	30	74	30	358	30	46	26	2)	-	222	31

1) Not completely correct, 2) Not found all, 3) Found none

同様で、3重根は一応収束しているが、Example No10 の原点付近の根は求められていない。PA06CDによる誤差限界は、Table 3のPA07CDとは異なり、Example No4に対しても、また一部だがNo10に対しても求められている。これにChambersのアルゴリズムを導入した付録4のMUBCHMによる結果も、実係数のMUACHMと同様で、計算時間もMULLRBより短縮されている。次のTable 11のNewton法ルーチンCNWTNDによる結果($EPS=10^{-10}$)は、重根のある場合が全部不成功となっている。それ以外は他のルーチン同様正解を求めているが、計算時間はTable 15から分るようにMULLRB, MUBCHMの2倍程度かかっている場合もある。一方これにMadsenアルゴリズムを導入した付録6のPA06ADによる計算はこれよりは短縮されているが、低次多項式に対してはまだ相当の時間を要している。しかしExample No4, No10などに対するMULLRB, MUBCHMでみられた不正確さは殆んどみられず、その上、8次程度以上の多項式に対しては、それらの半分程度の計算時間で根が求められている。

Jarrat法のCJARTDによる結果は、Table 13から分るように、Example No4の根は一応求められており、No8は正確に求められている。しかしNo9~11は1部しか求められておらず、特にNo10は全然だめである。計算時間もTable 15から分るように、特に低次多項式に他より時間を要している。これに対し、Jarrat-modify法のDCJARTルーチンでは、Table 14に示したように、Example No1~8の結果はCJARTDのと殆んど同じだが、低次多項式に要する計算時間がTable 15にみられるようにいくらか短縮されている。No9~11に対しては、相当時間を使っているが、CJARTDとは異なり一応正解が求められている。

以上から分るように低次多項式を除けば、MadsenアルゴリズムのNewton法のルーチンが実係数の際にも、複素係数の際にも最善である。しかし実係数多項式に対しては、重根、近接根以外(2重根、二つの近接根は本報の例題では正しく求められている)では、Bairstow法のBAIR1Dルーチンが計算時間が短くてよい。重根に対しては、これを単根にしてから解く手法も開発されていることを付記しておく。¹⁶⁾ また、低次複素係数多項式に対しては、3重根、三つの近接根などの場合を除けば、Chambersのアルゴリズムを取り入れたMuller法ルーチンMUBCHMが、MadsenのPA06ADより計算時間が短くてよい。

6. 結 論

本報では多項式のすべての根を求める数値解法アルゴリズムを概観し、その主な計算ルーチンを整備あるいは作成し、既存のSSLも含めてベンチマーク・テストを実施した。

収束の位数が1.84であるMuller法ルーチンについては、実係数多項式に対してMULLRA、複素係数多項式に対してはMULLRBを整備した。これにChambersのアルゴリズムを取り入れて新しく作成したのがMUACHMとMUBCHMで、これの収束の位数は2.73であるが、アルゴリズムが少々複雑になるため、Muller法よりいくらか計算時間が短縮された程度である。しかしこれらのルーチンは3重根、三つの近接根などを求めるには不適當である。

一方標準的な解法としてはNewton法の変形であるJenkins-Traub法があるが、これより

同様で、3重根は一応収束しているが、Example No10 の原点付近の根は求められていない。PA06CDによる誤差限界は、Table 3のPA07CDとは異なり、Example No4に対しても、また一部だがNo10に対しても求められている。これにChambersのアルゴリズムを導入した付録4のMUBCHMによる結果も、実係数のMUA CHMと同様で、計算時間もMULLRBより短縮されている。次のTable 11のNewton法ルーチンCNWTNDによる結果($\text{EPS} = 10^{-10}$)は、重根のある場合が全部不成功となっている。それ以外は他のルーチン同様正解を求めているが、計算時間はTable 15から分るようにMULLRB, MUBCHMの2倍程度かかっている場合もある。一方これにMadsenアルゴリズムを導入した付録6のPA06ADによる計算はこれよりは短縮されているが、低次多項式に対してはまだ相当の時間を要している。しかしExample No4, No10などに対するMULLRB, MUBCHMでみられた不正確さは殆んどみられず、その上、8次程度以上の多項式に対しては、それらの半分程度の計算時間で根が求められている。

Jarrat法のCJARTDによる結果は、Table 13から分るように、Example No4の根は一応求められており、No8は正確に求められている。しかしNo9~11は1部しか求められておらず、特にNo10は全然だめである。計算時間もTable 15から分るように、特に低次多項式に他より時間を要している。これに対し、Jarrat-modify法のDCJARTルーチンでは、Table 14に示したように、Example No1~8の結果はCJARTDのと殆んど同じだが、低次多項式に要する計算時間がTable 15にみられるようにいくらか短縮されている。No9~11に対しては、相当時間を使っているが、CJARTDとは異なり一応正解が求められている。

以上から分るように低次多項式を除けば、MadsenアルゴリズムのNewton法のルーチンが実係数の際にも、複素係数の際にも最善である。しかし実係数多項式に対しては、重根、近接根以外(2重根、二つの近接根は本報の例題では正しく求められている)では、Bairstow法のBAIR1Dルーチンが計算時間が短くてよい。重根に対しては、これを単根にしてから解く手法も開発されていることを付記しておく。¹⁶⁾ また、低次複素係数多項式に対しては、3重根、三つの近接根などの場合を除けば、Chambersのアルゴリズムを取り入れたMuller法ルーチンMUBCHMが、MadsenのPA06ADより計算時間が短くてよい。

6. 結 論

本報では多項式のすべての根を求める数値解法アルゴリズムを概観し、その主な計算ルーチンを整備あるいは作成し、既存のSSLも含めてベンチマーク・テストを実施した。

収束の位数が1.84であるMuller法ルーチンについては、実係数多項式に対してMULLRA、複素係数多項式に対してはMULLRBを整備した。これにChambersのアルゴリズムを取り入れて新しく作成したのがMUA CHMとMUBCHMで、これの収束の位数は2.73であるが、アルゴリズムが少々複雑になるため、Muller法よりいくらか計算時間が短縮された程度である。しかしこれらのルーチンは3重根、三つの近接根などを求めるには不適當である。

一方標準的な解法としてはNewton法の変形であるJenkins-Traub法があるが、これより

2～4倍計算の速いことが報告されているMadsenアルゴリズムによるルーチン PA07AD, PA06ADの整備がなされた。このルーチンは低次多項式の根の算出にはMuller法, Muller-Chambers法より時間がかかるが, すべての場合に正確な根を与えており, 計算時間も高次多項式では, Muller法, Muller-Chambers法より相当短縮される。なお既存の実係数多項式に対するBairstow法のルーチンは, 3重根などには不適當だが, それ以外には, 高次, 低次を問わず最も速いアルゴリズムとなっている。複素係数多項式に対するNewton法ルーチンは重根にはだめだし, それ以外に対しても計算時間がかかり過ぎる。またJarrrat法も計算時間が長いし, 精度に問題がある場合もある。

以上により, 実係数多項式はBairstow法のBAIR1D, MadsenアルゴリズムのPA07AD, 複素係数多項式は同じくMadsenアルゴリズムのPA06AD, Muller-Chambers法のMUBCHMで一応すべて処理できると考えられる。なお, これら複素係数多項式に対するルーチンは, 実係数多項式も当然扱うことができ, 実係数に対するルーチンよりいくらか計算時間がかかる程度である。

求められた根のPA07CDあるいはPA06CDによる誤差限界の算出は, このために根の計算と同程度の計算時間を消費してはいるが, 結果は適切で有用である。

参考文献

- 1) Brent R., et al.: "Optimal Iterative Process for Root-Finding", Numer. Math., 20, 327 (1973).
- 2) Lawrence J. D.: "Polynomial Root Finder", CIC Report C2.2-001 (1966).
- 3) Chambers Ll. G.: "A Quadratic Formula for Finding the Root of an Equation", Math. Comp., 25, 305 (1971).
- 4) 富士通 : "FACOM SSL (科学用サブルーチン・ライブラリ) 解法解説書," (1971)
- 5) 富士通 : "FACOM 230-60 SSL (科学用サブルーチン・ライブラリ) 使用方法解説書, FORTRAN 編," (1972)
- 6) Traub J. F.: "A Class of Globally Convergent Iterative Functions for the Solution of Polynomial Equations", Math. Comp., 20, 113 (1966).
- 7) Jenkins M. A., Traub J. F.: "A Three-Stage Algorithm for Real Polynomials Using Quadratic Iteration", SIAM J. Numer. Anal., 7, 545 (1970).
- 8) Jenkins M. A., Traub J. F.: "A Three-Stage Variable-Shift Iteration for Polynomial Zeros and its Relation to Generalized Rayleigh Iteration", Numer. Math., 14, 252 (1970).
- 9) Jenkins M. A., Traub J. F.: "Algorithm 419, Zeros of a Complex Polynomial", Comm. ACM, 15, 97 (1972).
- 10) Madsen K. and Reid J. K.: "Fortran Subroutines for Finding Polynomial Zeros", AERE-R 7986 (1975).
- 11) Madsen K.: "A Root-Finding Algorithm Based on Newton's Method", BIT, 13, 71 (1973).
- 12) Jarrat P., Nudds D.: "The Use of Rational Functions in the Iterative Solution of Equations on a Digital Computer", Comp. J., 8, 62 (1966).
- 13) Garside G. R., et al.: "A New Method for Solving Polynomial Equations", Comp. J., 11, 87 (1967).
- 14) 富士通 : "FACOM FORTRAN SSL II 使用手引書," (1976)
- 15) Hansen E., Patrick M.: "A Family of Root Finding Methods", Numer. Math., 27, 257 (1977).
- 16) Ishiguro M.: "A Numerical Method for Extraction of Multiple Roots of Algebraic Equation", Information Processing in Japan, 12, 46 (1972).

Appendix 1. Fortran Lists of MULLRA Subroutine, its Test Program
and PA07CD Subroutine for Error-Bounds

ISN	ST-NO	SOURCE PROGRAM
1		IMPLICIT DOUBLE PRECISION (A-H,O-Z)
2		DIMENSION A(41),R(2,40),*R(52),RR(41),RRR(40),RRI(40)
3		REAL RE(41),S(2,52)
4		EQUIVALENCE (*R(1),S(1,1))
	C	MAXIMUM ORDER OF REAL POLYNOMIAL = 40
5	5	READ (5,100,END=99) N
6		READ (5,103) (R(I,1),I=1,N)
7		WRITE (6,200) N,(R(I,1),I=1,N)
8		N1=N+1
9		DO 30 I=1,N1
10	30	A(I)=0.00
11		A(1)=1.00
12		DO 90 I=1,N
13		II=N1-I
14		BB=R(II,1)
15		DO 10 JJ=1,I
16		J=I+1-JJ
17	10	A(J+1)=A(J+1)-BB*A(J)
18	90	CONTINUE
19		*WRITE (6,201) (A(J),J=1,N1)
20		DO 40 J=1,N1
21	40	RR(J)=A(J)
22		CALL CLOCKM(IIJJ)
23		*WRITE (6,202) IIJJ
24		CALL MULLRA (A,N,RRR,RRI,N1)
25		CALL CLOCKM(IIJJ)
26		*WRITE (6,202) IIJJ
27		DO 51 J=1,N
28		R(1,J)=RRR(J)
29		R(2,J)=RRI(J)
30		WRITE (6,102) R(1,J),R(2,J)
31		IF(DABS(R(2,J)) .LT. DABS(R(1,J))*1.D-10) R(2,J)=0.00
32		IF(DABS(R(1,J)) .LT. DABS(R(2,J))*1.D-10) R(1,J)=0.00
33	51	A (J)=R(1,J)*R(1,J)+R(2,J)*R(2,J)
34		DO 52 I=1,N
35		DO 53 J=I,N
36		IF(A (I).GE. A (J)) GO TO 53
37		BB=A (J)
38		A (J)=A (I)
39		A (I)=BB
40		DO 54 II=1,2
41		BB=R(II,J)
42		R(II,J)=R(II,I)
43	54	R(II,I)=BB
44	53	CONTINUE
45	52	CONTINUE
46		DO 58 J=1,N1
47		II=N1+1-J
48		A(II)=RR(J)
49	58	RE(J)=0.0
	C	CHECK FOR ZERO LEADING COEFFICIENTS
50		M=N1
51	55	M=M-1
	C	IF(ABS(A(M+1)).EQ.0.0 ,AND. M.GT.0)GO TO 55
52		IF(DABS(A(M+1)).EQ.0.0 ,AND. M.GT.0)GO TO 55
	C	N2=N*2+1
53		N2=N/2

```

ISN  ST-NO          SOURCE PROGRAM      ( FTMAIN )
      C   IF(M.GE.1)CALL PA07CD(A,M,M+1,R,RR,W,W(N+2),W,W(N2+2))
      C   IF(M.GE.1)CALL PA07CD(A,M,M+1,R,E,W,W(N+2),W,W(N2+2))
54    IF(M.GE.1)CALL PA07CD(A,M,M+1,R,RE,W,S(1,N+2),S,S(1,N2+2))
55    CALL CLOCKM(I1JJ)
56    WRITE (6,202) I1JJ
      C   SET DUMMY DISCS CORRESPONDING TO INFINITE ROOTS
57    IF(M.EQ.N)GO TO 56
58    M=M+1
59    DO 57 I=M,N
C57   RR(I)= ABS(R(1,I))+ ABS(R(2,I))
60    57   RE(I)=DABS(R(1,I))+DABS(R(2,I))
61    56   CONTINUE
62    DO 2 I=1,N
63    2   WRITE (6,102) P(1,I),P(2,I),RE(I)
64    GO TO 5
65    100  FORMAT (I10)
66    102  FORMAT (5X,1P2D30,15,' WITH ERROR OF ',1PE12,5)
67    103  FORMAT (8D10,4)
68    200  FORMAT (5X,15,'-TH POLYNOMIAL WITH ROOTS',(10X,1P5D20,10))
69    201  FORMAT (5X,' COEFFICIENTS',(10X,1P5D20,10))
70    202  FORMAT (5X,' TIME ',I10,' MSEC')
71    99  STOP
72    END

```

ISN	ST-NO	SOURCE PROGRAM
		C MULLER
1		SUBROUTINE MULLRA(COE,N1,ROOTR,ROOTI,N2)
2		IMPLICIT DOUBLE PRECISION (A-H,O-Z)
3		DIMENSION COE(N2),ROOTR(N1),ROOTI(N1)
4		N4=0
5		I=N1+1
6	19	IF(COE(I))9,7,9
7	7	N4=N4+1
8		ROOTR(N4)=0,
9		ROOTI(N4)=0,
10		I=I-1
11		IF(N4-N1)19,37,19
12	9	CONTINUE
13	10	AXR=0.8
14		AXI=0,
15		L=1
16		N3=1
17		ALP1R=AXR
18		ALP1I=AXI
19		M=1
20		GOTO99
21	11	BET1R=TEMR
22		BET1I=TEMI
23		AXR=0.85
24		ALP2R=AXR
25		ALP2I=AXI
26		M=2
27		GOTO99
28	12	BET2R=TEMR
29		BET2I=TEMI
30		AXR=0.9
31		ALP3R=AXR
32		ALP3I=AXI
33		M=3
34		GOTO99
35	13	BET3R=TEMR
36		BET3I=TEMI
37	14	TE1=ALP1R-ALP3R
38		TE2=ALP1I-ALP3I
39		TE5=ALP3R-ALP2R
40		TE6=ALP3I-ALP2I
41		TEM=TE5*TE5+TE6*TE6
42		TE3=(TE1*TE5+TE2*TE6)/TEM
43		TE4=(TE2*TE5-TE1*TE6)/TEM
44		TE7=TE3+1.
45		TE9=TE3*TE3-TE4*TE4
46		TE10=2.*TE3*TE4
47		DE15=TE7*BET3R-TE4*BET3I
48		DE16=TE7*BET3I+TE4*BET3R
49		TE11=TE3*BET2R-TE4*BET2I+BET1R-DE15
50		TE12=TE3*BET2I+TE4*BET2R+BET1I-DE16
51		TE7=TE9-1.
52		TE1=TE9*BET2R-TE10*BET2I
53		TE2=TE9*BET2I+TE10*BET2R
54		TE13=TE1-BET1R-TE7*BET3R+TE10*BET3I
55		TE14=TE2-BET1I-TE7*BET3I-TE10*BET3R
56		TE15=DE15*TE3-DE16*TE4

```

ISN      ST-NO      SOURCE PROGRAM      ( MULLRA )
57          TE16=DE15*TE4+DE16*TE3
58          TE1=TE13*TE13-TE14*TE14-4.*(TE11*TE15-TE12*TE16)
59          TE2=2.*TE13*TE14-4.*(TE12*TE15+TE11*TE16)
60          IF(DABS(TE1).LT.1,D+20.AND,DABS(TE2).LT.1,D+20) GO TO 23
61          TE1=TE1*1,D-20
62          TE2=TE2*1,D-20
63          TEM=DSQRT(TE1*TE1+TE2*TE2)*1,D+20
64          TE1=TE1*1,D+20
65          TE2=TE2*1,D+20
66          GO TO 24
67      23  TEM=DSQRT(TE1*TE1+TE2*TE2)
68      24  IF(TE1)113,113,112
69      113  TE4=DSQRT(.5*(TEM-TE1))
70          TE3=.5*TE2/TE4
71          GO TO 111
72      112  TE3=DSQRT(.5*(TEM+TE1))
73          IF(TE2)110,200,200
74      110  TE3=-TE3
75      200  TE4=.5*TE2/TE3
76      111  TE7=TE13+TE3
77          TE8=TE14+TE4
78          TE9=TE13-TE3
79          TE10=TE14-TE4
80          TE1=2.*TE15
81          TE2=2.*TE16
82          IF(TE7*TE7+TE8*TE8-TE9*TE9-TE10*TE10)204,204,205
83      204  TE7=TE9
84          TE8=TE10
85      205  TEM=TE7*TE7+TE8*TE8
86          TE3=(TE1*TE7+TE2*TE8)/TEM
87          TE4=(TE2*TE7-TE1*TE8)/TEM
88          AXR=ALP3R+TE3*TE5-TE4*TE6
89          AXI=ALP3I+TE3*TE6+TE4*TE5
90          ALP4R=AXR
91          ALP4I=AXI
92          M=4
93          GO TO 99
94      15  N6=1
95      38  IF(DABS(BELL)+DABS(BELL)-1.E-20)18,18,16
96      16  TE7=DABS(ALP3R-AXR)+DABS(ALP3I-AXI)
97          IF(TE7/(DABS(AXR)+DABS(AXI))-1.E-7)18,18,17
98      17  N3=N3+1
99          ALP1R=ALP2R
100         ALP1I=ALP2I
101         ALP2R=ALP3R
102         ALP2I=ALP3I
103         ALP3R=ALP4R
104         ALP3I=ALP4I
105         BET1R=BET2R
106         BET1I=BET2I
107         BET2R=BET3R
108         BET2I=BET3I
109         BET3R=TEMR
110         BET3I=TEMI
111         IF(N3-100)14,18,18
112      18  N4=N4+1
113         ROOTR(N4)=ALP4R

```

ISN	ST-NO	SOURCE PROGRAM	(MULLA)
114		ROOTI(N4)=ALP4I	
115		N3=0	
116	41	IF(N4-N1)30,37,37	
117	37	RETURN	
118	30	IF(DABS(ROOTI(N4))-1.E-5)10,10,31	
119	31	GO TO(32,10).L	
120	32	AXR=ALP1R	
121		AXI=-ALP1I	
122		ALP1I=-ALP1I	
123		M=5	
124		GO TO 99	
125	33	BET1R=TEMR	
126		BET1I=TEMI	
127		AXR=ALP2R	
128		AXI=-ALP2I	
129		ALP2I=-ALP2I	
130		M=6	
131		GO TO 99	
132	34	BET2R=TEMR	
133		BET2I=TEMI	
134		AXR=ALP3R	
135		AXI=-ALP3I	
136		ALP3I=-ALP3I	
137		L=2	
138		M=3	
139	99	TEMR=COE(1)	
140		TEMI=0.0	
141		DO100I=1,N1	
142		TE1=TEMR*AXR-TEMI*AXI	
143		TEMI=TEMI*AXR+TEMR*AXI	
144	100	TEMR= TE1+COE(I+1)	
145		HELL=TEMR	
146		BELL=TEMI	
147	42	IF(N4)102,103,102	
148	102	DO101I=1,N4	
149		TEM1=AXR-ROOTR(I)	
150		TEM2=AXI-ROOTI(I)	
151		TE1=TEM1*TEM1+TEM2*TEM2	
152		TE2=(TEMR*TEM1+TEMI*TEM2)/TE1	
153		TEMI=(TEMI*TEM1-TEMR*TEM2)/TE1	
154	101	TEMR=TE2	
155	103	GO TO(11,12,13,15,33,34).M	
156		END	

Appendix 2. Fortran Lists of MULLRB Subroutine, its Test Program
and PA06CD Subroutine for Error-Bounds

ISN	ST-NO	SOURCE PROGRAM
1		IMPLICIT DOUBLE PRECISION (A-H,O-Z)
2		DIMENSION A(41),R(2,40),w(2,52),AI(41),C(2,41),RRR(40),RRI(40)
3		REAL RE(41),S(4,52)
4		EQUIVALENCE (*C(1,1),S(1,1))
5	C	MAXIMUM ORDER OF REAL POLYNOMIAL = 40
5	5	READ (5,100,END=99) N
6		WRITE (6,200) N
7		N1=N+1
8		READ (5,103) (A(J),J=1,N1)
9		READ (5,103) (AI(J),J=1,N1)
10		WRITE (6,201) (A(J),J=1,N1)
11		WRITE (6,201) (AI(J),J=1,N1)
12		DO 40 J=1,N1
13		II=N1+1-J
14		C(2,II)=AI(J)
15	40	C(1,II)=A(J)
16		CALL CLOCKM(IIJJ)
17		WRITE (6,202) IIJJ
18		CALL MULLRB (A,AI,N,RRR,RRI,N1)
19		CALL CLOCKM(IIJJ)
20		WRITE (6,202) IIJJ
21		DO 51 J=1,N
22		R(1,J)=RRR(J)
23		R(2,J)=RRI(J)
24		WRITE (6,102) R(1,J),R(2,J)
25		IF(DABS(R(2,J)) .LT. DABS(R(1,J))*1.D-10) R(2,J)=0.D0
26		IF(DABS(R(1,J)) .LT. DABS(R(2,J))*1.D-10) R(1,J)=0.D0
27	51	A (J)=R(1,J)*R(1,J)+R(2,J)*R(2,J)
28		DO 52 I=1,N
29		DO 53 J=1,N
30		IF(A (I).GE.A (J)) GO TO 53
31		BB=A (J)
32		A (J)=A (I)
33		A (I)=BB
34		DO 54 II=1,2
35		BB=R(II,J)
36		R(II,J)=R(II,I)
37	54	R(II,I)=BB
38	53	CONTINUE
39	52	CONTINUE
40		DO 58 J=1,N1
41	58	RE(J)=0.0
42	C	CHECK FOR ZERO LEADING COEFFICIENTS
42		M=N1
43	55	M=M-1
44	C	IF(ABS(C(1,M+1))+ABS(C(2,M+1)).EQ.0.0 .AND. M.GT.0)GO TO 55
44	C	IF(DABS(C(1,M+1))+DABS(C(2,M+1)).EQ.0.0 .AND. M.GT.0)GO TO 55
45	C	N2=N/2
45	C	N4=N/4
46	C	IF(M.GE.1)CALL PA06CD(C,M,M+1,R,RR,w,w(1,N+2),w,w(1,N2+2))
46	C	IF(M.GE.1)CALL PA06CD(C,M,M+1,R,RR,w,w(1,N+2),w,w(1,N4+2))
46	C	IF(M.GE.1)CALL PA06CD(C,M,M+1,R,RE,w,S(1,N+2),S,S(1,N4+2))
47		CALL CLOCKM(IIJJ)
48		WRITE (6,202) IIJJ
49	C	SET DUMMY DISCS CORRESPONDING TO INFINITE ROOTS
49		IF(M.EQ.N)GO TO 56
50		M=M+1

ISN	ST=NO	SOURCE PROGRAM	(FTMAIN)
51		DO 57 I=M,N	
	C57	RR(I)= ABS(R(1,I))+ ABS(R(2,I))	
52	57	RE(I)=DABS(R(1,I))+DABS(R(2,I))	
53	56	CONTINUE	
54		DO 2 I=1,N	
55	2	WRITE (6,102) R(1,I),R(2,I),RE(I)	
56		GO TO 5	
57	100	FORMAT (I10)	
58	102	FORMAT (5X,1P2D30.15,'WITH ERROR OF',1PE12,5)	
59	103	FORMAT (8D10,4)	
60	200	FORMAT (5X,I5,'-TH POLYNOMIAL WITH ROOTS'/(10X,1P5D20.10))	
61	201	FORMAT (5X,'COEFFICIENTS'/(10X,1P5D20.10))	
62	202	FORMAT (5X,'TIME',I10,'MSEC')	
63	99	STOP	
64		END	

ISN	ST-NO	SOURCE PROGRAM
		C MULLER
1		SUBROUTINE MULLRB(COE,COE1,N1,ROOTR,ROOTI,N2)
2		IMPLICIT DOUBLE PRECISION (A-H,O-Z)
3		DIMENSION COE(N2),ROOTR(N1),ROOTI(N1),COE1(N2)
4		N4=0
5		I=N1+1
6	19	IF(COE(I))9,7,9
7	7	IF(COE1(I))9,1,9
8	1	N4=N4+1
9		ROOTR(N4)=0.
10		ROOTI(N4)=0.
11		I=I-1
12		IF(N4-N1)19,37,19
13	9	CONTINUE
14	10	AXR=0.8
15		AXI=0.
16		N3=1
17		ALP1R=AXR
18		ALP1I=AXI
19		M=1
20		GOTO99
21	11	BET1R=TEMR
22		BET1I=TEMI
23		AXR=0.85
24		ALP2R=AXR
25		ALP2I=AXI
26		M=2
27		GOTO99
28	12	BET2R=TEMR
29		BET2I=TEMI
30		AXR=0.9
31		ALP3R=AXR
32		ALP3I=AXI
33		M=3
34		GOTO99
35	13	BET3R=TEMR
36		BET3I=TEMI
37	14	TE1=ALP1R-ALP3R
38		TE2=ALP1I-ALP3I
39		TE5=ALP3R-ALP2R
40		TE6=ALP3I-ALP2I
41		TEM=TE5*TE5+TE6*TE6
42		TE3=(TE1*TE5+TE2*TE6)/TEM
43		TE4=(TE2*TE5-TE1*TE6)/TEM
44		TE7=TE3+1.
45		TE9=TE3*TE3-TE4*TE4
46		TE10=2.*TE3*TE4
47		DE15=TE7*BET3R-TE4*BET3I
48		DE16=TE7*BET3I+TE4*BET3R
49		TE11=TE3*BET2R-TE4*BET2I+BET1R-DE15
50		TE12=TE3*BET2I+TE4*BET2R+BET1I-DE16
51		TE7=TE9-1.
52		TE1=TE9*BET2R-TE10*BET2I
53		TE2=TE9*BET2I+TE10*BET2R
54		TE13=TE1-BET1R-TE7*BET3R+TE10*BET3I
55		TE14=TE2-BET1I-TE7*BET3I-TE10*BET3R
56		TE15=DE15*TE3-DE16*TE4

ISN	ST-NO	SOURCE PROGRAM	(MULLRB)
57		TE16=DE15*TE4+DE16*TE3	
58		TE1=TE13*TE13-TE14*TE14-4.*(TE11*TE15-TE12*TE16)	
59		TE2=2.*TE13*TE14-4.*(TE12*TE15+TE11*TE16)	
60		IF(DABS(TE1).LT.1.D+20.AND.DABS(TE2).LT.1.D+20) GO TO 23	
61		TE1=TE1*1.D-20	
62		TE2=TE2*1.D-20	
63		TEM=DSQRT(TE1*TE1+TE2*TE2)*1.D+20	
64		TE1=TE1*1.D+20	
65		TE2=TE2*1.D+20	
66		GO TO 24	
67	23	TEM=DSQRT(TE1*TE1+TE2*TE2)	
68	24	IF(TE1)113,113,112	
69	113	TE4=DSQRT(.5*(TEM-TE1))	
70		TE3=.5*TE2/TE4	
71		GO TO 111	
72	112	TE3=DSQRT(.5*(TEM+TE1))	
73		IF(TE2)110,200,200	
74	110	TE3=-TE3	
75	200	TE4=.5*TE2/TE3	
76	111	TE7=TE13+TE3	
77		TE8=TE14+TE4	
78		TE9=TE13-TE3	
79		TE10=TE14-TE4	
80		TE1=2.*TE15	
81		TE2=2.*TE16	
82		IF(TE7*TE7+TE8*TE8-TE9*TE9-TE10*TE10)204,204,205	
83	204	TE7=TE9	
84		TE8=TE10	
85	205	TEM=TE7*TE7+TE8*TE8	
86		TE3=(TE1*TE7+TE2*TE8)/TEM	
87		TE4=(TE2*TE7-TE1*TE8)/TEM	
88		AXR=ALP3R+TE3*TE5-TE4*TE6	
89		AXI=ALP3I+TE3*TE6+TE4*TE5	
90		ALP4R=AXR	
91		ALP4I=AXI	
92		M=4	
93		GO TO 99	
94	15	N6=1	
95	38	IF(DABS(HELL)+DABS(BELL)-1.E-20)18,18,16	
96	16	TE7=DABS(ALP3R-AXR)+DABS(ALP3I-AXI)	
97		IF(TE7/(DABS(AXR)+DABS(AXI))-1.E-7)18,18,17	
98	17	N3=N3+1	
99		ALP1R=ALP2R	
100		ALP1I=ALP2I	
101		ALP2R=ALP3R	
102		ALP2I=ALP3I	
103		ALP3R=ALP4R	
104		ALP3I=ALP4I	
105		BET1R=BET2R	
106		BET1I=BET2I	
107		BET2R=BET3R	
108		BET2I=BET3I	
109		BET3R=TEMR	
110		BET3I=TEMI	
111		IF(N3-100)14,18,18	
112	18	N4=N4+1	
113		ROOTR(N4)=ALP4R	

```
ISN      ST-NO          SOURCE PROGRAM      ( MULLRB )
114          ROOT1(N4)=ALP4I
115          N3=0
116      41 IF(N4=N1)10,37,37
117          37 RETURN
118          99 TEMR=COE(1)
119          TEMI=0,0
120          DO100I=1,N1
121          TE1=TEMR*AXR-TEMI*AXI
122          TEMI=TEMI*AXR+TEMR*AXI +COE(I+1)
123      100 TEMR=      TE1+COE(I+1)
124          HELL=TEMR
125          BELL=TEMI
126      42 IF(N4)102,103,102
127      102 DO101I=1,N4
128          TEM1=AXR-ROOTR(I)
129          TEM2=AXI-ROOTI(I)
130          TE1=TEM1*TEM1+TEM2*TEM2
131          TE2=(TEMR*TEM1+TEMI*TEM2)/TE1
132          TEMI=(TEMI*TEM1-TEMR*TEM2)/TE1
133          101 TEMR=TE2
134          103 GO TO(11,12,13,15),M
135          END
```

Appendix 3. Fortran Program List of MUACHM Subroutine

ISN	ST=NO	SOURCE PROGRAM
		C MULLER
1		SUBROUTINE MUACHM(COE,N1,ROOTR,ROOTI,N2)
2		IMPLICIT DOUBLE PRECISION (A-H,O-Z)
3		DIMENSION COE(N2),ROOTR(N1),ROOTI(N1)
4		N4=0
5		I=N1+1
6	19	IF(COE(I))9,7,9
7	7	N4=N4+1
8		ROOTR(N4)=0.
9		ROOTI(N4)=0.
10		I=I-1
11		IF(N4-N1)19,37,19
12	9	CONTINUE
13	10	AXR=0.8
14		AXI=0.
15		L=1
16		N3=1
17		ALP1R=AXR
18		ALP1I=AXI
19		M=1
20		GOTO99
21	11	BET1R=TEMR
22		BET1I=TEMI
23		AXR=0.85
24		ALP2R=AXR
25		ALP2I=AXI
26		M=2
27		GOTO99
28	12	BET2R=TEMR
29		BET2I=TEMI
30		M=3
31		TE5=ALP2R-ALP1R
32		TE6=ALP2I-ALP1I
33		TE1=BET2R-BET1R
34		TE2=BET2I-BET1I
35		TEM=TE1*TE1+TE2*TE2
36		TE3=(BET2R*TE1+BET2I*TE2)/TEM
37		TE4=(BET2I*BET1R-BET2R*BET1I)/TEM
38		AXR=ALP2R+TE3*TE5-TE4*TE6
39		AXI=ALP2I+TE3*TE6+TE4*TE5
40		ALP3R=AXR
41		ALP3I=AXI
42		GOTO99
43	13	BET3R=TEMR
44		BET3I=TEMI
45	14	TE1=ALP1R-ALP3R
46		TE2=ALP1I-ALP3I
47		TE5=ALP3R-ALP2R
48		TE6=ALP3I-ALP2I
49		TEM=TE5*TE5+TE6*TE6
50		TE3=(TE1*TE5+TE2*TE6)/TEM
51		TE4=(TE2*TE5-TE1*TE6)/TEM
52		TE7=TE3+1.
53		TE9=TE3*TE3-TE4*TE4
54		TE10=2.*TE3*TE4
55		DE15=TE7*BET3R-TE4*BET3I
56		DE16=TE7*BET3I+TE4*BET3R

```

ISN      ST-NO      SOURCE PROGRAM      ( MOACHM )
57      TE11=TE3*BET2R-TE4*BET2I+BET1R-DE15
58      TE12=TE3*DE12I+TE4*BET2R+BET1I-DE16
59      TE7=TE9-1.
60      TE1=TE9*BET2R-TE10*BET2I
61      TE2=TE9*BET2I+TE10*BET2R
62      TE13=TE1-BET1R-TE7*BET3R+TE10*BET3I
63      TE14=TE2-BET1I-TE7*BET3I-TE10*BET3R
64      TE15=DE15*TE3-DE16*TE4
65      TE16=DE15*TE4+DE16*TE3
66      TE1=TE13*TE13-TE14*TE14-4.*(TE11*TE15-TE12*TE16)
67      TE2=2.*TE13*TE14-4.*(TE12*TE15+TE11*TE16)
68      IF(DABS(TE1).LT.1.D+20.AND.DABS(TE2).LT.1.D+20) GO TO 23
69      TE1=TE1*1.D-30
70      TE2=TE2*1.D-30
71      TEM=DSQRT(TE1*TE1+TE2*TE2)*1.D+30
72      TE1=TE1*1.D+30
73      TE2=TE2*1.D+30
74      GO TO 24
75      23 TEM=DSQRT(TE1*TE1+TE2*TE2)
76      24 IF(TE1)113,113,112
77      113 TE4=DSQRT(.5*(TEM-TE1))
78      TE3=.5*TE2/TE4
79      GO TO 111
80      112 TE3=DSQRT(.5*(TEM+TE1))
81      IF(TE2)110,200,200
82      110 TE3=-TE3
83      200 TE4=.5*TE2/TE3
84      111 TE7=TE13+TE3
85      TE8=TE14+TE4
86      TE9=TE13-TE3
87      TE10=TE14-TE4
88      TE1=2.*TE15
89      TE2=2.*TE16
90      IF(TE7*TE7+TE8*TE8-TE9*TE9-TE10*TE10)204,204,205
91      204 TE7=TE9
92      TE8=TE10
93      205 TEM=TE7*TE7+TE8*TE8
94      TE3=(TE1*TE7+TE2*TE8)/TEM
95      TE4=(TE2*TE7-TE1*TE8)/TEM
96      AXR=ALP3R+TE3*TE5-TE4*TE6
97      AXI=ALP3I+TE3*TE6+TE4*TE5
98      ALP4R=AXR
99      ALP4I=AXI
100     M=4
101     GO TO 99
102     15 N6=1
103     38 IF(DABS(HELL)+DABS(BELL)-1.F-20)18,18,16
104     16 TE7=DABS(ALP3R-AXR)+DABS(ALP3I-AXI)
105     IF(TE7/(DABS(AXR)+DABS(AXI))-1.E-7)18,18,17
106     17 N3=N3+1
107     ALP1R=ALP3R
108     ALP1I=ALP3I
109     ALP2R=ALP4R
110     ALP2I=ALP4I
111     BET1R=BET3R
112     BET1I=BET3I
113     IF(N3-100)12,18,18

```

ISN	ST-NO	SOURCE PROGRAM	(NUACHM)
114	18	N4=N4+1	
115		ROOTR(N4)=ALP4R	
116		ROOTI(N4)=ALP4I	
117		N3=0	
118	41	IF(N4-N1)30,37,37	
119	37	RETURN	
120	30	IF(DABS(ROOTI(N4))-1.E-5)10,10,31	
121	31	GO TO(32,10),L	
122	32	AXR=ALP1R	
123		AXI=-ALP1I	
124		ALP1I=-ALP1I	
125		M=5	
126		GO TO 99	
127	33	BET1R=TEMR	
128		BET1I=TEMI	
129		AXR=ALP2R	
130		AXI=-ALP2I	
131		ALP2I=-ALP2I	
132		M=6	
133		L=2	
134	99	TEMR=COE(I)	
135		TEMI=0.0	
136		DO100I=1,N1	
137		TE1=TEMR*AXR-TEMI*AXI	
138		TEMI=TEMI*AXR+TEMR*AXI	
139	100	TEMR= TE1+COE(I+1)	
140		BELL=TEMR	
141		BELL=TEMI	
142	42	IF(N4)102,103,102	
143	102	DO101I=1,N4	
144		TEM1=AXR-ROOTR(I)	
145		TEM2=AXI-ROOTI(I)	
146		TE1=TEM1*TEM1+TEM2*TEM2	
147		TE2=(TEMR*TEM1+TEMI*TEM2)/TE1	
148		TEMI=(TEMI*TEM1-TEMR*TEM2)/TE1	
149	101	TEMR=TE2	
150	103	GO TO(11,12,13,15,33,12),M	
151		END	

Appendix 4. Fortran Program List of MUBCHM Subroutine

ISN	ST-NO	SOURCE PROGRAM
		C MULLER
1		SUBROUTINE MUBCHM(COE,COE1,N1,ROOTR,ROOTI,N2)
2		IMPLICIT DOUBLE PRECISION (A-H,O-Z)
3		DIMENSION COE(N2),ROOTR(N1),ROOTI(N1),COE1(N2)
4		N4=0
5		I=N1+1
6	19	IF(COE(I))9,7,9
7	7	IF(COE1(I))9,1,9
8	1	N4=N4+1
9		ROOTR(N4)=0.
10		ROOTI(N4)=0.
11		I=I-1
12		IF(N4=N1)19,37,19
13	9	CONTINUE
14	10	AXR=0.8
15		AXI=0.
16		NJ=1
17		ALP1R=AXR
18		ALP1I=AXI
19		M=1
20		GOTO99
21	11	BET1R=TEMR
22		BET1I=TEMI
23		AXR=0.85
24		ALP2R=AXR
25		ALP2I=AXI
26		M=2
27		GOTO99
28	12	BET2R=TEMR
29		BET2I=TEMI
30		M=3
31		TE5=ALP2R-ALP1R
32		TE6=ALP2I-ALP1I
33		TE1=BET2R-BET1R
34		TE2=BET2I-BET1I
35		TEM=TE1*TE1+TE2*TE2
36		TE3=-(BET2R*TE1+BET2I*TE2)/TEM
37		TE4=(BET2I*BET1R-BET2R*BET1I)/TEM
38		AXR=ALP2R+TE3*TE5-TE4*TE6
39		AXI=ALP2I+TE3*TE6+TE4*TE5
40		ALP3R=AXR
41		ALP3I=AXI
42		GOTO99
43	13	BET3R=TEMR
44		BET3I=TEMI
45	14	TE1=ALP1R-ALP3R
46		TE2=ALP1I-ALP3I
47		TE5=ALP3R-ALP2R
48		TE6=ALP3I-ALP2I
49		TEM=TE5*TE5+TE6*TE6
50		TE3=(TE1*TE5+TE2*TE6)/TEM
51		TE4=(TE2*TE5-TE1*TE6)/TEM
52		TE7=TE3+1.
53		TE9=TE3*TE3-TE4*TE4
54		TE10=2.*TE3*TE4
55		DE15=TE7*BET3R-TE4*BET3I
56		DE16=TE7*BET3I+TE4*BET3R

```

ISN      ST-NO      SOURCE PROGRAM      ( MUBCHM )
57      TE11=TE3*BET2R-TE4*BET2I+BET1R-DE15
58      TE12=TE3*BET2I+TE4*BET2R+BET1I-DE16
59      TE7=TE9-1.
60      TE1=TE9*BET2R-TE10*BET2I
61      TE2=TE9*BET2I+TE10*BET2R
62      TE13=TE1-BET1R-TE7*BET3R+TE10*BET3I
63      TE14=TE2-BET1I-TE7*BET3I-TE10*BET3R
64      TE15=DE15*TE3-DE16*TE4
65      TE16=DE15*TE4+DE16*TE3
66      TE1=TE13*TE13-TE14*TE14-4.*(TE11*TE15-TE12*TE16)
67      TE2=2.*TE13*TE14-4.*(TE12*TE15+TE11*TE16)
68      IF(DABS(TE1),LT.1,D+20,AND,DABS(TE2),LT.1,D+20) GO TO 23
69      TE1=TE1*1,D-30
70      TE2=TE2*1,D-30
71      TEM=DSQRT(TE1*TE1+TE2*TE2)*1,D+30
72      TE1=TE1*1,D+30
73      TE2=TE2*1,D+30
74      GO TO 24
75      23 TEM=DSQRT(TE1*TE1+TE2*TE2)
76      24 IF(TE1)113,113,112
77      113 TE4=DSQRT(.5*(TEM-TE1))
78      TE3=.5*TE2/TE4
79      GO TO 111
80      112 TE3=DSQRT(.5*(TEM+TE1))
81      IF(TE2)110,200,200
82      110 TE3=-TE3
83      200 TE4=.5*TE2/TE3
84      111 TE7=TE13+TE3
85      TE8=TE14+TE4
86      TE9=TE13-TE3
87      TE10=TE14-TE4
88      TE1=2.*TE15
89      TE2=2.*TE16
90      IF(TE7*TE7+TE8*TE8-TE9*TE9-TE10*TE10)204,204,205
91      204 TE7=TE9
92      TE8=TE10
93      205 TEM=TE7*TE7+TE8*TE8
94      TE3=(TE1*TE7+TE2*TE8)/TEM
95      TE4=(TE2*TE7-TE1*TE8)/TEM
96      AXR=ALP3R+TE3*TE5-TE4*TE6
97      AXI=ALP3I+TE3*TE6+TE4*TE5
98      ALP4R=AXR
99      ALP4I=AXI
100     M=4
101     GO TO 99
102     15 N6=1
103     38 IF(DABS(HELL)+DABS(BELL)-1,E-20)18,18,16
104     16 TE7=DABS(ALP3R-AXR)+DABS(ALP3I-AXI)
105     IF(TE7/(DABS(AXR)+DABS(AXI)))-1,E-7)18,18,17
106     17 N3=N3+1
107     ALP1R=ALP3R
108     ALP1I=ALP3I
109     ALP2R=ALP4R
110     ALP2I=ALP4I
111     BET1R=BET3R
112     BET1I=BET3I
113     IF(N3-100)12,18,18

```

ISN	ST-NO	SOURCE PROGRAM	(MUBCHM)
114	18	N4=N4+1	
115		ROOTR(N4)=ALP4R	
116		ROOTI(N4)=ALP4I	
117		N3=0	
118	41	IF(N4=N1)10,37,37	
119	37	RETURN	
120	99	TEMR=COE(1)	
121		TEMI=0.0	
122		DO100I=1,N1	
123		TE1=TEMR*AXR-TEMI*AXI	
124		TEMI=TEMI*AXR+TEMR*AXI +COE1(I+1)	
125	100	TEMR= TE1+COE(I+1)	
126		HELL=TEMR	
127		BELL=TEMI	
128	42	IF(N4)102,103,102	
129	102	DO101I=1,N4	
130		TEMI=AXR-ROOTR(I)	
131		TEM2=AXI-ROOTI(I)	
132		TE1=TEMI*TEMI+TEM2*TEM2	
133		TE2=(TEMR*TEMI+TEMI*TEM2)/TE1	
134		TEMI=(TEMI*TEMI-TEMR*TEM2)/TE1	
135	101	TEMR=TE2	
136	103	GO TO(11,12,13,15),M	
137		END	

Appendix 5. Fortran Lists of PA07AD Subroutine and its Test Program

ISN	ST-NO	SOURCE PROGRAM
1		IMPLICIT DOUBLE PRECISION (A-H,O-Z)
2		DIMENSION A(41),R(2,40),W(62),RR(40)
3		REAL E(41),S(2,62)
4		EQUIVALENCE (W(1),S(1,1))
	C	MAXIMUM ORDER OF REAL POLYNOMIAL = 40
5	5	READ (5,100,END=99) N
6		WRITE (6,200) N
7		N1=N+1
8		READ (5,103) (A(J),J=1,N1)
9		N2=N1/2
10		DO 50 J=1,N2
11		I1=N1+1-J
12		BB=A(J)
13		A(J)=A(I1)
14	50	A(I1)=BB
15		WRITE (6,201) (A(J),J=1,N1)
16		DO 40 J=1,N1
17	40	E(J)=0,0
18		LW=3*N/2+2
19		CALL CLOCKM(I1JJ)
20		WRITE (6,202) I1JJ
21		CALL PA07AD(A,N,R,E,W,S,N1,LW)
22		CALL CLOCKM(I1JJ)
23		WRITE (6,202) I1JJ
24		DO 2 I=1,N
25	2	WRITE (6,102) R(1,I),R(2,I),E(I)
26		GO TO 5
27	100	FORMAT (I10)
28	102	FORMAT (5X,1P2D30,15,'WITH ERROR OF',1PE12,5)
29	103	FORMAT (8D10,4)
30	200	FORMAT (5X,15,'-TH POLYNOMIAL WITH ROOTS'/(10X,1P5D20,10))
31	201	FORMAT (5X,'COEFFICIENTS'/(10X,1P5D20,10))
32	202	FORMAT (5X,'TIME',I10,'MSEC')
33	99	STOP
34		END

JSN	ST-NO	SOURCE PROGRAM
	C	SUBROUTINE PA07AD(A,N,R,E,W,NP1,LW)
	C	SUBROUTINE PA07AD(A,N,R,E,W)
1	C	SUBROUTINE PA07AD(A,N,R,E,W,S,NP1,LW)
	C	REAL A(NP1),W(LW),R(2,N)
	C	DOUBLE PRECISION A(1),W(1),R(2,N)
2	C	DOUBLE PRECISION A(NP1),W(LW),R(2,N)
	C	REAL E(NP1)
	C	REAL E(1)
3	C	REAL E(NP1),S(2,LW)
	C	CALL PA07BD(A,N,R,W)
4	C	CALL PA07BD(A,N,R,W,N+1)
5	C	CALL CLOCKM(11JJ)
6	C	WRITE (6,202) 11JJ
7	C	DO 2 I=1,N
8	2	WRITE (6,102) R(1,I),R(2,I)
	C	CHECK FOR ZERO LEADING COEFFICIENTS
9	C	M=N+1
10	5	M=M-1
	C	IF(ABS(A(M+1)),E0,0,0 .AND. M.GT.0)GO TO 5
11	C	IF(DABS(A(M+1)),E0,0,0 .AND. M.GT.0)GO TO 5
	C	N2=N+2+1
12	C	N2=N/2
	C	IF(M.GE.1)CALL PA07CD(A,M,M+1,R,E,W,W(N+2),W,W(N2+2))
	C	IF(M.GE.1)CALL PA07CD(A,M,M+1,R,E,W,W(N+2),W,W(N2+2))
13	C	IF(M.GE.1)CALL PA07CD(A,M,M+1,R,E,W,S(1,N+2),S,S(1,N2+2))
	C	SET DUMMY DISCS CORRESPONDING TO INFINITE ROOTS
14	C	IF(M,E0,N)GO TO 20
15	C	M=M+1
16	C	DO 10 I=M,N
	C10	E(I)=ABS(R(1,I))+ABS(R(2,I))
17	10	E(I)=DABS(R(1,I))+DABS(R(2,I))
18	20	CONTINUE
19	102	FORMAT (5X,1P2D30,15)
20	202	FORMAT (5X,'TIME',I10,'MSEC')
21	C	RETURN
22	C	END

```

ISN  ST-NO          SOURCE PROGRAM
      C          SUBROUTINE PA07BD(A1,M,ROOT,A)
1     C          SUBROUTINE PA07BD(A1,M,ROOT,A,MP1)
      C          REAL Z0(2),F0Z(2),Z(2),DZ(2),F1Z(2),FZ(2),W(2),
      C          1 Fw(2),ROOT(2,M),DZK(2)
2     C          DOUBLE PRECISION Z0(2),F0Z(2),Z(2),DZ(2),F1Z(2),FZ(2),W(2),
      C          1 Fw(2),ROOT(2,M),DZK(2)
      C ALL THE ABOVE ARRAYS HOLD COMPLEX VARIABLES AND COMPLEX ARRAYS AND
      C WILL BE REFERRED TO IN COMMENTS AS SUCH, WITH NO MENTION OF THE FIRST
      C SUBSCRIPT
      C
      C          WHEN (M-N) ZEROS ARE FOUND, A(1), ... ,A(N+1), WILL HOLD THE
      C          COEFFICIENTS OF THE DEFLATED POLYNOMIAL. ROOT(N+1),... ,ROOT(M)
      C          WILL HOLD THE ZEROS FOUND. A1(1), ... ,A1(N) WILL HOLD THE
      C          COEFFICIENTS OF THE DERIVATIVE OF THE DEFLATED POLYNOMIAL.
      C          ROOT(1),... ,ROOT(N),A1(N+1),... ,A1(M+1) HOLD COEFFICIENTS OF
      C          THE ORIGINAL POLYNOMIAL.
      C          REAL F0,FF,F,FA,PA07DD,DZ1,A1(MP1),A(MP1),FC,FD,G,AC
      C          1,PP,W0,RR,SS,TT
      C          DOUBLE PRECISION F0,FF,F,FA,PA07DD,DZ1,A1( 1),A( 1),FC,FD,G
      C          1,AC,PP,W0,RR,SS,TT
3     C          DOUBLE PRECISION F0,FF,F,FA,PA07DD,DZ1,A1(MP1),A(MP1),FC,FD,G
      C          1,AC,PP,W0,RR,SS,TT
4     C          LOGICAL STAGE1,DIV2
      C          STAGE1 IS .TRUE. DURING STAGE1 OF THE ITERATION AND .FALSE.
      C          DURING STAGE2. DIV2 IS .TRUE. IF THE SEARCH WITH STEP-LENGTHS
      C          DZ,DZ/2,... IS IN USE AND .FALSE. IF THE SEARCH WITH STEPS DZ,2*DZ,...
      C          IS IN USE.
      C          DATA BIG/1.E70/,SMALL/1.E-70/,BASE/16./,EPS/1.0E-6/
5     C          DATA BIG/1.0E70/,SMALL/1.0E-70/,BASE/16./,EPS/2.3E-16/
      C          BIG,SMALL ARE NUMBERS NEAR OVERFLOW/UNDERFLOW LIMITS. BASE IS THE BASE
      C          OF THE FLOATING POINT ARITHMETIC IN USE. EPS IS THE SMALLEST
      C          NUMBER WHICH LEAVES UNITY UNCHANGED IN THE FLOATING-POINT ARITHMETIC
      C          IN USE.
6     C          SSMALL=SQRT(SMALL)
7     C          ALOGB=ALOG(BASE)
8     C          N=M
      C
      C          STORE ORIGINAL POLYNOMIAL IN A AND IN ROOT.
9     C          J=M+1
10    C          A(J)=A1(1)
11    C          DO 10 I=1,M
12    C          ROOT(1,I)=A1(I)
13    C          A(I)=A1(J)
14    C          10 J=J-1
      C
      C          TEST FOR ZEROS AT INFINITY
      C20  C          IF( ABS(A(1)).GT.0.)GO TO 40
15    C20  C          IF(DABS(A(1)).GT.0.)GO TO 40
16    C          DO 30 I=1,N
17    C          30 A(I)=A(I+1)
18    C          ROOT(1,N)=BIG
19    C          ROOT(2,N)=0.
20    C          N=N-1
21    C          IF (N.GT.0) GOTO 20
22    C          GOTO 310
23    C          40 IF (N.LE.1) GOTO 260
24    C          N1=N+1

```

```

ISN   ST=NO          SOURCE PROGRAM      ( PA07BD )

      C
      C SCALE THE COEFFICIENTS
25     U1=0.
26     U2=BIG
27     DO 50 K=1,N1
      C     U=ABS(A(K))
28     U=DABS(A(K))
29     IF(U.LE,U,)GO TO 50
30     IF (U.GT,U1) U1=U
31     IF (U.LT,U2) U2=U
32     50 CONTINUE
33     U=SQRT(U1)*SQRT(U2)
34     I=-ALOG(U)/ALOGB
35     U=BASE**I
36     DO 70 K=1,N
37     A(K)=A(K)*U
38     70 A1(K)=A(K)*FLOAT(N1-K)
39     A(N1)=A(N1)*U

      C
      C     TEST FOR ZEROS AT (0.,0.)
40     Z(1)=0,
41     Z(2)=0,
      C     IF(ABS(A(N1)).LE.SSMALL)GO TO 290
42     IF(DABS(A(N1)).LE.SSMALL)GO TO 290
43     Z0(1)=0,
44     Z0(2)=0,
45     F0=A(N1)**2
46     FMIN=F0*(FLOAT(N)*16.*EPS)**2

      C
      C THE FOLLOWING QUANTITIES ARE HELD AT THE START OF EACH ITERATION
      C Z IS THE CURRENT POINT, F=CABS(F(Z))**2
      C Z0 IS THE LAST POINT, S0Z=F'(Z0), F0=CABS(F(Z0))**2
      C R0=3*CABS(Z-Z0)
      C DZ IS THE LAST TENTATIVE STEP IF THE LAST ITERATION WAS SUCCESSFUL OR
      C ELSE IS THE REQUIRED NEXT TENTATIVE STEP, ON FIRST ITERATION IT IS
      C SET TO Z,
      C FF=CABS(F(ZT))**2 WHERE ZT IS THE LAST TENTATIVE POINT.
      C
      C SET INITIAL ITERATES AND QUANTITY USED IN THE CONVERGENCE TEST.
47     FF=F0
      C     U0=ABS(A(N1))
48     U0=DABS(A(N1))
49     T=BIG
50     DO 80 K=1,N
      C     U=ABS(A(K))
51     U=DABS(A(K))
52     IF (U.E0,0.) GOTO 80
53     U=ALOG(U0/U)/FLOAT(N1-K)
54     IF (U.LT,T) T=U
55     80 CONTINUE
56     T=EXP(T)
57     F0Z(1)=A(N)
58     F0Z(2)=0,
59     Z(1)=0.5*T
60     IF(A(N).LT,0.)Z(1)=-Z(1)
61     DZ(1)=Z(1)
62     DZ(2)=0.

```

```

ISN   ST-NO      SOURCE PROGRAM      ( PA07BD )
63           F=PA07DD(Z,FZ,N+1,A)
64           RU=0.5*T
      C
      C      CALCULATION OF THE TENTATIVE STEP DZ AND WHETHER IN STAGE1.
      C THIS IS WHERE THE ITERATION STARTS IF THE PREVIOUS ONE WAS SUCCESSFUL.
65   120   U=PA07DD(Z,F1Z,N ,A1)
66           IF (U.EQ.0.) GOTO 140
      C      DZ=-FZ/F1Z
67           CALL PA07ED(FZ,F1Z,DZ)
68           F2=((FOZ(1)-F1Z(1))**2+(FOZ(2)-F1Z(2))**2)/
1      ((ZO(1)-Z(1))**2+(ZO(2)-Z(2))**2)
69           STAGE1=F*F2/U.GT.U*0.25 .OR. F.NE.FF
      C
70           R= ABS(DZ(1))+ ABS(DZ(2))
71           R=DABS(DZ(1))*DABS(DZ(2))
72           IF(R.LE.R0*3.)GO TO 150
73           DZ1=DZ(1)
74           DZ(1)=(DZ1*1.8-DZ(2)*2.4)*R0/R
75           DZ(2)=(DZ1*2.4+DZ(2)*1.8)*R0/R
76   140   DZ1=DZ(1)
77           DZ(1)= DZ1*1.8-DZ(2)*2.4
78           DZ(2)= DZ1*2.4+DZ(2)*1.8
79           STAGE1=.TRUE.
80   150   FOZ(1)=F1Z(1)
81           FOZ(2)=F1Z(2)
      C
      C FIND NEXT POINT IN THE ITERATION, THIS IS WHERE THE ITERATION STARTS
      C IF THE PREVIOUS ONE WAS UNSUCCESSFUL.
82   160   ZO(1)=Z(1)
83           ZO(2)=Z(2)
84           FO=F
85           DZK(1)=DZ(1)
86           DZK(2)=DZ(2)
87           Z(1)=ZO(1)+DZ(1)
88           Z(2)=ZO(2)+DZ(2)
      C IF EITHER PART OF Z IS SMALL REPLACE BY ZERO TO AVOID UNDERFLOWS
      C IF ( ABS(Z(1)).LT.EPS* ABS(Z(2)))Z(1)=0.
89           IF (DABS(Z(1)).LT.EPS*DABS(Z(2)))Z(1)=0.
      C IF ( ABS(Z(2)).LT.EPS* ABS(Z(1)))Z(2)=0.
90           IF (DABS(Z(2)).LT.EPS*DABS(Z(1)))Z(2)=0.
91           W(1)=Z(1)
92           W(2)=Z(2)
93           F=PA07DD(Z,FZ,N+1,A)
94           FF=F
95           IF (,NOT,STAGE1) GOTO 240
      C BEGINNING OF STAGE1 SEARCH.
96           J=1
97           DIV2=F.GE.F0
98   180   IF (DIV2) GOTO 190
99           W(1)=W(1)+DZ(1)
100          W(2)=W(2)+DZ(2)
101          GOTO 200
102   190   DZ(1)=DZ(1)*0.5
103          DZ(2)=DZ(2)*0.5
104          W(1)=ZO(1)+DZ(1)
105          W(2)=ZO(2)+DZ(2)
106   200   FA=PA07DD(W,F*,N+1,A)

```

```

ISN      ST-NO      SOURCE PROGRAM      ( PA07BD )
107      IF (FA,GE,F) GOTO 240
108      F=FA
109      FZ(1)=FW(1)
110      FZ(2)=FW(2)
111      Z(1)=W(1)
112      Z(2)=W(2)
113      J=J+1
114      IF (DIV2,AND,(J,E0,3)) GOTO 220
115      IF (J,LE,N) GOTO 180
116      GOTO 240
117      220 DZ1=DZ(1)
118          DZ(1)=DZ1*0.6-DZ(2)*0.8
119          DZ(2)=DZ1*0.8+DZ(2)*0.6
120          Z(1)=Z0(1)+DZ(1)
121          Z(2)=Z0(2)+DZ(2)
122          F=PA07DD(Z,FZ,N+1,A)
          C END OF STAGE1 SEARCH
          C240 RO= ABS(Z0(1)-Z(1))+ ABS(Z0(2)-Z(2))
123      240 RO=DABS(Z0(1)-Z(1))+DABS(Z0(2)-Z(2))
          C
          C CONVERGENCE TEST,
124          IF (F,LT,F0) GO TO 250
125          Z(1)=Z0(1)
126          Z(2)=Z0(2)
          C250 R1= ABS(Z(1))+ ABS(Z(2))
127      250 R1=DABS(Z(1))+DABS(Z(2))
128          IF (RO,LT,EPS*R1) GO TO 270
129          IF (F,LT,F0) GOTO 120
130          F=F0
131          IF (F,LE,FMIN) GO TO 270
132          DZ(1)=DZK(1)*(-0.3)-DZK(2)*(-0.4)
133          DZ(2)=DZK(1)*(-0.4)+DZK(2)*(-0.3)
134          STAGE1=.TRUE.
135          GOTO 160
          C
          C DEAL WITH N=1 CASE
136      260 Z(1)=-A(2)/A(1)
137          Z(2)=0.
          C
          C DEFLATE, STORE ROOT, RESTORE COEFFICIENT OF ORIGINAL POLY AND REDUCE N
138      270 A1(N)=ROOT(1,N)
139          ROOT(1,N)=Z(1)
140          RCOT(2,N)=0.
141          IF (Z(2),EQ,0.) GO TO 277
          C
142          AC= ABS(Z(1))
143          AC=DABS(Z(1))
144          G=0.
145          FC=A(1)
146          DO 275 K=2,N1
          C
147          G=AC*(G+ ABS(FC))+ ABS(FD)
148      275 G=AC*(G+DABS(FC))+DABS(FD)
          C
149          FC=FD
          C
          IF ( ABS(FC) .GT. 2.*EPS*G+SQRT(AMIN1(F,F0))) GO TO 283
          IF (DABS(FC) .GT. 2.*EPS*G+SQRT(SNGL(DMIN1(F,F0)))) GO TO 283
          C DEFLATION WITH A REAL ROOT
150      277 G=0.

```

ISN	ST-NO	SOURCE PROGRAM	(PA078D)
151		DO 280 K=1,N	
152		G=G*Z(1)+A(K)	
153	280	A(K)=G	
154		GO TO 300	
155		C DEFLATION WITH A PAIR OF COMPLEX CONJUGATE ROOTS	
156	283	ROOT(2,N)=Z(2)	
157		RR=0.	
158		SS=0.	
159		PP=-Z(1)-Z(1)	
160		QQ=Z(1)**2+Z(2)**2	
161		N=N-1	
162		DO 287 K=1,N	
163		TT=A(K)-PP*RR-QQ*SS	
164		A(K)=TT	
165	287	SS=RR	
166	290	RR=TT	
167		A1(N)=ROOT(1,N)	
168		ROOT(1,N)=Z(1)	
169	300	ROOT(2,N)=-Z(2)	
170		N=N-1	
171	310	IF (N=1) 310,260,40	
172		RETURN	
		END	

```

1      SUBROUTINE PAUTCD(A,N,NP1,R,E,W,F,IG,CR)
2      C IG AND CR MAY BE DYNAMICALLY EQUIVALENCED WITH SEPARATE PARTS OF A
3      C REAL A(NP1),R(2,N),W(NP1),C,RS,RP
4      C DOUBLE PRECISION A(NP1),R(2,N),W(NP1),C,RS,RP
5      C REAL E(NP1),F(NP1),CR(N),IG(N)
6      C W IS A WORKSPACE ARRAY OF LENGTH N+1 USED TO HOLD THE COEFFICIENTS
7      C OF THE POLYNOMIAL FORMED FROM THE CALCULATED ROOTS.
8      C F IS A WORKSPACE ARRAY OF LENGTH N+1 USED TO HOLD THE COEFFICIENTS
9      C OF THE ERROR POLYNOMIAL.
10     C IG IS A WORKSPACE ARRAY OF LENGTH N USED TO LINK TOGETHER THE ROOTS
11     C IN A GROUP, THE FIRST IS GIG, IG(K) FOLLOWS K AND THE LAST IS 1.
12     C OTHER ROOTS HAVE IG(K)=0.
13     C CR IS A WORKSPACE ARRAY OF LENGTH N USED TO HOLD DISTANCES FROM ROOT 1
14     C TO TH REST,
15     C DATA EPS/1.0E-6 /,BIG/1.E70/
16     C DATA EPS/2.3E-16/,BIG/1.0E40/
17     C BIG IS A NUMBER NEAR THE OVERFLOW LIMIT, EPS IS THE SMALLEST
18     C NUMBER WHICH LEAVES UNITY UNCHANGED IN THE FLOWTING-POINT ARITHMETIC
19     C IN USE.
20     C FACT=1.05**(1./FLOAT(N))
21     C N1=N+1
22     C
23     C MULTIPLY OUT THE POLYNOMIAL FORMED FROM THE CALCULATED ROOTS
24     DO 5 I=1,N1
25     W(I)=0.
26     W(1)=A(N1)
27     I=1
28     II=N1-I
29     IF(R(2,II).NE,0.)GO TO 12
30     C=R(1,II)
31     DO 10 JJ=1,I
32     J=I+1-JJ
33     10 W(J+1)=W(J+1)-C*W(J)
34     GO TO 18
35     12 RS=R(1,II)+R(1,II)
36     RP=R(1,II)**2+R(2,II)**2
37     I=I+1
38     DO 16 JJ=1,I
39     J=I-JJ
40     W(J+2)=W(J+2)-RS*W(J+1)
41     IF(J,EQ,0)GO TO 16
42     W(J+2)=W(J+2)+RP*W(J)
43     16 CONTINUE
44     I=I+1
45     IF(I.LE,N)GO TO 8
46     C
47     C FIND COEFFICIENTS OF ERROR POLYNOMIAL
48     DO 20 I=1,N1
49     II=N+2-I
50     C20 F(I)= ABS(A(II)-W(I))*AMAX1(E(II),ABS(A(II)))*EPS)
51     20 F(I)=DABS(A(II)-W(I))*AMAX1(E(II),ABS(SNGL(A(II)))*EPS)
52     DO 30 I=1,N
53     30 IG(I)=0.
54     C
55     C MAIN ERROR-ROUNDING LOOP STARTS HERE AND EXTENDS TO THE END
56     C OF THE SUBROUTINE IF FINDS A BOUND FOR ROOT 1.
57     I=1

```

```

ISN   ST-NO          SOURCE PROGRAM      ( PAD7CD )
35    31      M=1
36          IJJI=1
37          IG(1)=-1
38          IG1=1
39          D=0.
      C
      C FIND DISTANCES TO OTHER ROOTS
40          R1=R(1,I)
41          R2=R(2,I)
42          IF(R2.EQ.0.)GO TO 34
43          CR1=CABS(CMPLX(R1,R2))
44          DO 32 K=1,N
45          R1=R(1,K)-R(1,I)
46          R2=R(2,K)-R(2,I)
47          32 CR(K)=CABS(CMPLX(R1,R2))
48          GO TO 40
49          34 K=1
50          CR1=ABS(R1)
51          36 R2=R(2,K)
52          R1=R(1,K)-R(1,I)
53          IF(R2.EQ.0.)GO TO 37
54          CR(K)=CABS(CMPLX(R1,R2))
55          CR(K+1)=CR(K)
56          K=K+1
57          GO TO 38
58          37 CR(K)=ABS(R1)
59          38 K=K+1
60          IF(K.LE.N)GO TO 36
      C
      C INITIALIZATION STATEMENTS WHICH ARE ALSO NEEDED WHEN M IS INCREASED
61          40 DMIN=BIG
62          S=CR1+D
63          RAD=D
      C TEST ROUCHE CONDITION WITH RADIUS RAD.
64          45 TOP=F(1)
65          PROD=1.
      C
66          BTM=ABS(A(N1))
67          BTM=DABS(A(N1))
68          IJJ=0
69          DO 80 K=1,N
70          61 IF(ABS(TOP).LT.1.E+60) GO TO 63
71          IJJ=1
72          TOP=TOP*1.E-40
73          BTM=BTM*1.E-40
74          62 TOP=S*TOP+F(K+1)*1.E-40
75          GO TO 64
76          63 TOP=S*TOP+F(K+1)
77          64 DIST=CR(K)
78          IF(IG(K).NE.0.)GO TO 70
79          BTM=BTM*(DIST-RAD)
80          IF(DMIN.LE.DIST)GO TO 80
81          DMIN=DIST
82          L=K
83          GO TO 80
84          70 PROD=PROD*(RAD-DIST)
85          IF(ABS(PROD).LT.1.E+60) GO TO 80

```

```

ISN   ST-NO           SOURCE PROGRAM      ( PA07CD )

86     PROD=PROD*1.E-20
87     BTM=BTM*1.E+20
88     50  CONTINUE
89     IF(BTM,EQ,0.0)GO TO 100
90     TB= ABS(TOP/BTM)
91     IF(PROD,GE,TB)GO TO 110
      C FIND A NEW TRIAL RADIUS.
92     OLDS=S
93     OLDR=RAD
94     RL=RAD
95     RAD=D+1.1*TB
96     IF (M,EQ,1)GO TO 98
97     RM=D+(1.1*TB)**(1./FLOAT(M))
      C BISECTION LOOP, RL AND RM HOLD UPPER AND LOWER BOUNDS.
98     83  RAD=(RL+RM)/2.
99     PROD=1.
100    K=IG1
101    90  PROD=PROD*(RAD-CR(K))
102    K=IFIX(IG(K))
103    IIII=IIII+1
104    IF(IIII,GT,100) GO TO 101
105    IF(K,GT,0)GO TO 90
106    IF(PROD,GE,1.10*TB)GO TO 94
107    IF(RAD,GE,DMIN)GO TO 100
108    IF(PROD,GT,1.05*TB)GO TO 98
109    RL=RAD
110    GO TO 83
111    94  RM=RAD
112    GO TO 83
113    98  S=S+RAD-OLDR
114    IF(RAD,GE,DMIN)GO TO 100
115    IF( S*((DMIN-OLDR)/(DMIN-RAD)),LE,FACT*OLDS)GO TO 110
116    GO TO 45
      C
      C ADD ROOT TO GROUP
117    100 M=M+1
118    IG(L)=FLOAT(IG1)
119    IG1=L
120    D=DMIN
121    GO TO 40
      C
      C STORE ERROR BOUND AND RESET IG
122    101 RAD=0.
123    110 E(I)=RAD
124    120 K=IG1
125    IG1=IFIX(IG(IG1))
126    IG(K)=0.
127    IF(K,NE,1)GO TO 120
128    IF(R(2+I),EQ,0.0)GO TO 130
129    E(I+1)=E(I)
130    I=I+1
131    130 I=I+1
132    IF(I,LE,N)GO TO 31
133    RETURN
134    END

```

ISN	ST-NO	SOURCE PROGRAM
		C REAL FUNCTION PA07DD(Z,FZ,N1,A)
1		DOUBLE PRECISION FUNCTION PA07DD(Z,FZ,N1,A)
		C REAL Z(2),FZ(2),A(N1),P,Q,R,S,T
2		DOUBLE PRECISION Z(2),FZ(2),A(N1),P,Q,R,S,T
		C Z AND FZ CONTAIN COMPLEX NUMBERS.
		C A CONTAINS REAL POLYNOMIAL COEFFICIENTS.
		C POLYNOMIAL VALUE AT Z IS SET IN FZ AND THE SQUARE OF ITS MODULUS
		C IS RETURNED AS FUNCTION VALUE.
3		N=N1-1
4		T=A(1)
5		IF(Z(2).EQ.0.)GO TO 30
6		P=Z(1)+Z(1)
7		Q=-(Z(1)**2+Z(2)**2)
8		R=0.
9		IF(N.EQ.1)GO TO 20
10		DO 10 K=2,N
11		S=R
12		R=T
13	10	T=P*R+Q*S+A(K)
14	20	FZ(1)=Z(1)*T+Q*R+A(N+1)
15		FZ(2)=Z(2)*T
16		GO TO 50
17	30	P=Z(1)
18		DO 40 I=1,N
19	40	T=T*P+A(I+1)
20		FZ(1)=T
21		FZ(2)=0.
22	50	PA07DD=FZ(1)**2+FZ(2)**2
23		RETURN
24		END

ISN	ST-NO	SOURCE PROGRAM
1		SUBROUTINE PA07ED(A,B,Q)
	C	REAL A(2),B(2),Q(2),U,A1,C,D,E
2		DOUBLE PRECISION A(2),B(2),Q(2),U,A1,C,D,E
	C	IMPLEMENTS THE COMPLEX DIVISION Q=-A/B
	C	U=ABS(B(1))+ABS(B(2))
3		U=DABS(B(1))+DABS(B(2))
4		A1=A(1)
5		C=B(1)/U
6		D=B(2)/U
7		E=-(C*C+D*D)*U
8		Q(1)=(A1*C+A(2)*D)/E
9		Q(2)=(A(2)*C-A1*D)/E
10		RETURN
	C	
11		END

Appendix 6. Fortran Lists of PA06AD Subroutine and its Test Program

ISN	ST-NO	SOURCE PROGRAM
1		IMPLICIT DOUBLE PRECISION (A-H,O-Z)
2		DIMENSION A(2,41),R(2,40),W(2,52),RR(41)
3		REAL E(41),S(4,52)
4		EQUIVALENCE (W(1,1),S(1,1))
	C	MAXIMUM ORDER OF COMPLEX POLYNOMIAL = 40
5	5	READ (5,100,END=99) N
6		READ (5,103) (R(1,I),I=1,N)
7		WRITE (6,200) N,(R(1,I),I=1,N)
8		N1=N+1
9		DO 30 I=1,N1
10	30	RR(I)=0.D0
11		RR(1)=1.D0
12		DO 90 J=1,N
13		II=N1-I
14		BB=R(1,II)
15		DO 10 JJ=1,I
16		J=I+1-JJ
17	10	RR(J+1)=RR(J+1)-BB*RR(J)
18	90	CONTINUE
19		N2=N/2
20		DO 50 J=1,N2
21		II=N1+1-J
22		BB=R(1,II)
23		RR(II)=RR(J)
24	50	RR(J)=RR
25		DO 40 J=1,N1
26		E(J)=0.0
27		A(1,J)=RR(J)
28	40	A(2,J)=0.0D0
29		WRITE (6,201) (A(1,J),J=1,N1)
30		LW=5*N/4+2
31		CALL CLOCKM(11JJ)
32		WRITE (6,202) 11JJ
33		CALL PA06AD(A,N,R,E,W,S,N1,LW)
34		CALL CLOCKM(11JJ)
35		WRITE (6,202) 11JJ
36		DO 2 I=1,N
37	2	WRITE (6,102) R(1,I),R(2,I),E(I)
38		GO TO 5
39	100	FORMAT (110)
40	102	FORMAT (5X,1P2D30.15,'WITH ERROR OF',1PE12.5)
41	103	FORMAT (8D10.4)
42	200	FORMAT (5X,15,'-TH POLYNOMIAL WITH ROOTS'/(10X,1P5D20.10))
43	201	FORMAT (5X,'COEFFICIENTS'/(10X,1P5D20.10))
44	202	FORMAT (5X,'TIME',110,'MSEC')
45	99	STOP
46		END

ISN	ST-NO	SOURCE PROGRAM
	C	SUBROUTINE PA06AD(A,N,R,E,W, NP1,LW)
	C	SUBROUTINE PA06AD(A,N,R,E,W)
1		SUBROUTINE PA06AD(A,N,R,E,W,S,NP1,LW)
	C	REAL A(2,NP1),W(2,LW),R(2,N)
	C	DOUBLE PRECISION A(2,1),W(2,1),R(2,N)
2		DOUBLE PRECISION A(2,NP1),W(2,LW),R(2,N)
	C	REL E(NP1)
	C	REAL E(1)
3		REAL E(NP1),S(4,L*)
	C	CALL PA06BD(A,N,R,W)
4		CALL PA06BD(A,N,R,W,N+1)
5		CALL CLOCKM(11JJ)
6		*WRITE (6,202) 11JJ
7		DO 2 I=1,N
8	2	WRITE (6,102) R(1,I),R(2,I)
	C	CHECK FOR ZERO LEADING COEFFICIENTS
9		M=N+1
10	5	M=M-1
	C	IF(ABS(A(1,M+1))+ABS(A(2,M+1)).EQ.0.0 ,AND. M.GT.0)GO TO 5
11		IF(DABS(A(1,M+1))+DABS(A(2,M+1)).EQ.0.0 ,AND. M.GT.0)GO TO 5
	C	N2=N/2
12		N4=N/4
	C	IF(M.GE.1)CALL PA06CD(A,M,M+1,R,E,W,W(1,N+2),W,W(1,N2+2))
	C	IF(M.GE.1)CALL PA06CD(A,M,M+1,R,E,W,W(1,N+2),W,W(1,N4+2))
13		IF(M.GE.1)CALL PA06CD(A,M,M+1,R,E,W,S(1,N+2),S,S(1,N4+2))
	C	SET DUMMY DISCS CORRESPONDING TO INFINITE ROOTS
14		IF(M.EQ.N)GO TO 20
15		M=M+1
16		DO 10 I=M,N
	C10	E(I)=ABS(R(1,I))+ABS(R(2,I))
17	10	E(I)=DABS(R(1,I))+DABS(R(2,I))
18	20	CONTINUE
19	102	FORMAT (5X,1P2D30,15)
20	202	FORMAT (5X,'TIME',110,'MSEC')
21		RETURN
22		END

```

ISN  ST-NO      SOURCE PROGRAM
      C      SUBROUTINE PA06BD(A1,M,ROOT,A)
1     C      SUBROUTINE PA06BD(A1,M,ROOT,A,MP1)
      C      REAL Z0(2),FOZ(2),Z(2),DZ(2),F1Z(2),FZ(2),W(2),
      C      1FW(2),A(2,MP1),A1(2,MP1),ROOT(2,M),DZK(2)
      C      DOUBLE PRECISION Z0(2),FOZ(2),Z(2),DZ(2),F1Z(2),FZ(2),W(2),
2     C      1FW(2),A(2,1),A1(2,1),ROOT(2,M),DZK(2)
      C      DOUBLE PRECISION Z0(2),FOZ(2),Z(2),DZ(2),F1Z(2),FZ(2),W(2),
      C      1FW(2),A(2,MP1),A1(2,MP1),ROOT(2,M),DZK(2)
      C      ALL THE ABOVE ARRAYS HOLD COMPLEX VARIABLES AND COMPLEX ARRAYS AND
      C      WILL BE REFERRED TO IN COMMENTS AS SUCH, WITH NO MENTION OF THE FIRST
      C      SUBSCRIPT
      C
      C      WHEN (M-N) ZEROS ARE FOUND, A(1), ..., A(N+1), WILL HOLD THE
      C      COEFFICIENTS OF THE DEFLATED POLYNOMIAL, ROOT(N+1),...,ROOT(M)
      C      WILL HOLD THE ZEROS FOUND, A1(1), ..., A1(N) WILL HOLD THE
      C      COEFFICIENTS OF THE DERIVATIVE OF THE DEFLATED POLYNOMIAL,
      C      ROOT(1),...,ROOT(N),A1(N+1),...,A1(M+1) HOLD COEFFICIENTS OF
      C      THE ORIGINAL POLYNOMIAL.
      C      REAL FO,FF,F,FA,PA06DD,DZ1
3     C      DOUBLE PRECISION FO,FF,F,FA,PA06DD,DZ1
4     C      LOGICAL STAGE1,Div2
      C      STAGE1 IS .TRUE. DURING STAGE1 OF THE ITERATION AND .FALSE.
      C      DURING STAGE2, Div2 IS .TRUE. IF THE SEARCH WITH STEP-LENTHS
      C      DZ,DZ/2,... IS IN USE AND .FALSE. IF THE SEARCH WITH STEPS DZ,2*DZ,...
      C      IS IN USE.
5     C      DATA BIG/1.E70/,SMALL/1.E-70/,BASE/16./,EPS/1.E-6/
      C      DATA BIG/1.E70/,SMALL/1.E-70/,BASE/16./,EPS/2.3E-16/
      C      BIG,SMALL ARE NUMBERS NEAR OVERFLOW/UNDERFLOW LIMITS, BASE IS THE BASE
      C      OF THE FLOATING POINT ARITHMETIC IN USE, EPS IS THE SMALLEST
      C      NUMBER WHICH LEAVES UNITY UNCHANGED IN THE FLOATING-POINT ARITHMETIC
      C      IN USE.
6     C      SSMALL=SQRT(SMALL)
7     C      ALOGB=ALOG(BASE)
8     C      N=M
      C
9     C      STORE ORIGINAL POLYNOMIAL IN A AND IN ROOT.
10    C      J=M+1
11    C      A(1,J)=A1(1,1)
12    C      A(2,J)=A1(2,1)
13    C      DO 10 I=1,M
14    C      ROOT(1,I)=A1(1,I)
15    C      ROOT(2,I)=A1(2,I)
16    C      A(1,I)=A1(1,J)
17    C      A(2,I)=A1(2,J)
18    C      10 J=J-1
19    C
20    C      TEST FOR ZEROS AT INFINITY
21    C20 IF(ABS(A(1,1))+ABS(A(2,1)).GT.0.0)GO TO 40
22    C20 IF(DABS(A(1,1))+DABS(A(2,1)).GT.0.0)GO TO 40
19    C      DO 30 I=1,N
20    C      A(1,I)=A(1,I+1)
21    C      A(2,I)=A(2,I+1)
22    C      ROOT(1,N)=BIG
23    C      ROOT(2,N)=BIG
24    C      N=N-1
25    C      IF (N.GT.0) GOTO 20
26    C      GOTO 310

```

```

ISN   ST-NO           SOURCE PROGRAM      ( PA06BD )
27     40 IF (N,LE,1) GOTO 260
28     N1=N+1
      C
      C SCALE THE COEFFICIENTS
29     U1=0.
30     U2=BIG
31     DO 50 K=1,N1
      C   U= ABS(A(1,K))+ ABS(A(2,K))
32     U=DABS(A(1,K))+DABS(A(2,K))
33     IF(U,LE,0.)GO TO 50
34     IF (U,GT,U1) U1=U
35     IF (U,LT,U2) U2=U
36     50 CONTINUE
37     U=SQRT(U1)*SQRT(U2)
38     I=-ALOG(U)/ALOGB
39     U=BASE**I
40     DO 70 K=1,N
41     A(1,K)=A(1,K)*U
42     A(2,K)=A(2,K)*U
43     A1(1,K)=A(1,K)*FLOAT(N1-K)
44     70  A1(2,K)=A(2,K)*FLOAT(N1-K)
45     A(1,N1)=A(1,N1)*U
46     A(2,N1)=A(2,N1)*U
      C
      C   TEST FOR ZEROS AT (0.,0.)
47     Z(1)=0.
48     Z(2)=0.
      C   IF( ABS(A(1,N1))+ ABS(A(2,N1)),LE,SSMALL)GO TO 290
49     IF(DABS(A(1,N1))+DABS(A(2,N1)),LE,SSMALL)GO TO 290
50     Z0(1)=0.
51     Z0(2)=0.
52     F0=A(1,N1)**2+A(2,N1)**2
53     FMIN=F0*(FLOAT(N)*16.,*EPS)**2
      C
      C THE FOLLOWING QUANTITIES ARE HELD AT THE START OF EACH ITERATION
      C Z IS THE CURRENT POINT. F=CABS(F(Z))**2
      C Z0 IS THE LAST POINT. F0=F'(Z0), F0=CABS(F(Z0))**2
      C R0=3*CABS(Z-Z0)
      C DZ IS THE LAST TENTATIVE STEP IF THE LAST ITERATION WAS SUCCESSFUL OR
      C ELSE IS THE REQUIRED NEXT TENTATIVE STEP, ON FIRST ITERATION IT IS
      C SET TO Z.
      C FF=CABS(F(ZT))**2 WHERE ZT IS THE LAST TENTATIVE POINT.
      C
      C SET INITIAL ITERATES AND QUANTITY USED IN THE CONVERGENCE TEST.
54     FF=FU
55     U0=F0
56     T=BIG
57     DO 80 K=1,N
58     U=A(1,K)**2+A(2,K)**2
59     IF (U,EQ,0.) GOTO 80
60     U=(ALOG(U0)-ALOG(U))/FLOAT(2*(N1-K))
61     IF (U,LT,T) T=U
62     80 CONTINUE
63     T=EXP(T)
64     F0Z(1)=A(1,N)
65     F0Z(2)=A(2,N)
66     Z(1)=1.

```

```

ISN   ST-NO          SOURCE PROGRAM      ( PA06BD )
67     Z(2)=0.
68     C             IF( ABS(FOZ(1))+ ABS(FOZ(2)),LE.0.)GO TO 100
69     C             IF(DABS(FOZ(1))+DABS(FOZ(2)),LE.0.)GO TO 100
70     C             Z=-A(N1)/A(N)
71     C100          CALL PA06ED(A(1,N1),A(1,N),Z)
72     100          U=0.5*T/( ABS(Z(1))+ ABS(Z(2)))
73     U=0.5*T/(DABS(Z(1))+DABS(Z(2)))
74     Z(1)=Z(1)*U
75     Z(2)=Z(2)*U
76     DZ(1)=Z(1)
77     DZ(2)=Z(2)
78     F=PA06DD(Z,FZ,N+1,A)
79     R0=0.5*T
80     C
81     C             CALCULATION OF THE TENTATIVE STEP DZ AND WHETHER IN STAGE1.
82     C THIS IS WHERE THE ITERATION STARTS IF THE PREVIOUS ONE WAS SUCCESSFUL.
83     120          U=PA06DD(Z,F1Z,N ,A1)
84     IF (U.EQ.0.) GOTO 140
85     C             DZ=-FZ/F1Z
86     CALL PA06ED(FZ,F1Z,DZ)
87     F2=((FOZ(1)-F1Z(1))**2+(FOZ(2)-F1Z(2))**2)/
88     1 ((Z0(1)-Z(1))**2+(Z0(2)-Z(2))**2)
89     STAGE1=F*F2/U.GT.U*0.25 .OR. F.NE.FF
90     C             R= ABS(DZ(1))+ ABS(DZ(2))
91     R=DABS(DZ(1))+DABS(DZ(2))
92     IF(R.LE.R0*3.)GO TO 150
93     DZ1=DZ(1)
94     DZ(1)=(DZ1*1.8-DZ(2)*2.4)*R0/R
95     DZ(2)=(DZ1*2.4+DZ(2)*1.8)*R0/R
96     GOTO 150
97     140          DZ1=DZ(1)
98     DZ(1)= DZ1*1.8-DZ(2)*2.4
99     DZ(2)= DZ1*2.4+DZ(2)*1.8
100    STAGE1=.TRUE.
101    150          FOZ(1)=F1Z(1)
102    FOZ(2)=F1Z(2)
103    C
104    C             FIND NEXT POINT IN THE ITERATION. THIS IS WHERE THE ITERATION STARTS
105    C IF THE PREVIOUS ONE WAS UNSUCCESSFUL.
106    160          Z0(1)=Z(1)
107    Z0(2)=Z(2)
108    F0=F
109    DZK(1)=DZ(1)
110    DZK(2)=DZ(2)
111    Z(1)=Z0(1)+DZ(1)
112    Z(2)=Z0(2)+DZ(2)
113    C             IF EITHER PART OF Z IS SMALL REPLACE BY ZERO TO AVOID UNDERFLOWS
114    C             IF( ABS(Z(1)).LT.EPS* ABS(Z(2)))Z(1)=0.
115    C             IF(DABS(Z(1)).LT.EPS*DABS(Z(2)))Z(1)=0.
116    C             IF( ABS(Z(2)).LT.EPS* ABS(Z(1)))Z(2)=0.
117    C             IF(DABS(Z(2)).LT.EPS*DABS(Z(1)))Z(2)=0.
118    W(1)=Z(1)
119    W(2)=Z(2)
120    F=PAU6DD(Z,FZ,N+1,A)
121    FF=F
122    IF (,NOT,STAGE1) GOTO 240
123    C BEGINNING OF STAGE1 SEARCH.

```

```

ISN      ST-NO      SOURCE PROGRAM      ( PA06BD )
108      J=1
109      DIV2=F,GE,FO
110      180 IF (DIV2) GOTO 190
111      W(1)=W(1)+DZ(1)
112      W(2)=W(2)+DZ(2)
113      GOTO 200
114      190 DZ(1)=DZ(1)*0.5
115      DZ(2)=DZ(2)*0.5
116      W(1)=Z0(1)+DZ(1)
117      W(2)=Z0(2)+DZ(2)
118      200 FA=PA06DD(W,F*,N+1,A)
119      IF (FA,GE,F) GOTO 240
120      F=FA
121      FZ(1)=FW(1)
122      FZ(2)=FW(2)
123      Z(1)=W(1)
124      Z(2)=W(2)
125      J=J+1
126      IF (DIV2,AND,(J,EW,3)) GOTO 220
127      IF (J,LE,N) GOTO 180
128      GOTO 240
129      220 DZ1=DZ(1)
130      DZ(1)=DZ1*0.6-DZ(2)*0.8
131      DZ(2)=DZ1*0.8+DZ(2)*0.6
132      Z(1)=Z0(1)+DZ(1)
133      Z(2)=Z0(2)+DZ(2)
134      F=PA06DD(Z,FZ,N+1,A)
C END OF STAGE1 SEARCH
C240 RO= ABS(Z0(1)-Z(1))+ ABS(Z0(2)-Z(2))
135      240 RO=DABS(Z0(1)-Z(1))+DABS(Z0(2)-Z(2))
C
C CONVERGENCE TEST.
136      IF (F,LT,FO) GO TO 250
137      Z(1)=Z0(1)
138      Z(2)=Z0(2)
C250 R1= ABS(Z(1))+ ABS(Z(2))
139      250 R1=DABS(Z(1))+DABS(Z(2))
140      IF (RO,LT,EPS*R1) GO TO 270
141      IF (F,LT,FO) GOTO 120
142      F=FO
143      IF (F,LE,FMIN) GO TO 270
144      DZ(1)=DZK(1)*(-0.3)-DZK(2)*(-0.4)
145      DZ(2)=DZK(1)*(-0.4)+DZK(2)*(-0.3)
146      STAGE1=.TRUE,
147      GOTO 160
C
C DEAL WITH N=1 CASE
C Z=-A(2)/A(1)
148      260 CALL PA06ED(A(1,2),A(1,1),Z)
149      GO TO 290
C
C DEFLATE, STORE ROOT, RESTORE COEFFICIENT OF ORIGINAL POLY AND REDUCE N
150      270 DO 280 K=2,N
151      A(1,K)=A(1,K-1)*Z(1)-A(2,K-1)*Z(2)+A(1,K)
152      280 A(2,K)=A(1,K-1)*Z(2)+A(2,K-1)*Z(1)+A(2,K)
153      290 A1(1,N)=ROOT(1,N)
154      A1(2,N)=ROOT(2,N)

```

ISN	ST-NO	SOURCE PROGRAM	(PA063D)
155		ROOT(1,N)=Z(1)	
156		ROOT(2,N)=Z(2)	
157		N=N-1	
158		IF(N-1)310,260,40	
159	310	RETURN	
160		END	

ISN	ST-NO	SOURCE PROGRAM
		C REAL FUNCTION PA06DD(Z,FZ,N1,A)
1		DOUBLE PRECISION FUNCTION PA06DD(Z,FZ,N1,A)
		C REAL Z(2),FZ(2),A(2,N1),P,Q,R,X,Y
2		DOUBLE PRECISION Z(2),FZ(2),A(2,N1),P,Q,R,X,Y
		C Z AND FZ CONTAIN COMPLEX NUMBERS.
		C A CONTAINS COMPLEX POLYNOMIAL COEFFICIENTS.
		C POLYNOMIAL VALUE AT Z IS SET IN FZ AND THE SQUARE OF ITS MODULUS
		C IS RETURNED AS FUNCTION VALUE.
3		N=N1-1
4		X=Z(1)
5		Y=Z(2)
6		P=A(1,1)
7		Q=A(2,1)
8		DO 10 I=1,N
9		R=P*X-Q*Y+A(1,I+1)
10		Q=Q*X+P*Y+A(2,I+1)
11	10	P=R
12		FZ(1)=P
13		FZ(2)=Q
14		PA06DD=P*P+Q*Q
15		RETURN
16		END

ISN	ST-NO	SOURCE PROGRAM
1		SUBROUTINE PA06ED(A,B,0)
	C	REAL A(2),B(2),Q(2),U,A1,C,D,E
2		DOUBLE PRECISION A(2),B(2),Q(2),U,A1,C,D,E
	C	INPLEMENTS THE COMPLEX DIVISION $Q=-A/B$
	C	U=ABS(B(1))+ABS(B(2))
3		U=DABS(B(1))+DABS(B(2))
4		A1=A(1)
5		C=B(1)/U
6		D=B(2)/U
7		E=-(C*C+D*D)*U
8		Q(1)=(A1*C+A(2)*D)/E
9		Q(2)=(A(2)*C-A1*D)/E
10		RETURN
	C	
11		END

```

ISN   ST-NO      SOURCE PROGRAM
1     SUBROUTINE PA06CD(A,N,NP1,R,E,W,F,IG,CR)
      C IG AND CT MAY BE DYNAMICALLY EQUIVALENCED WITH SEPARATE PARTS OF W
      C REAL A(2,NP1),R(2,N),W(2,NP1),C(2)
2     DOUBLE PRECISION A(2,NP1),R(2,N),W(2,NP1),C(2)
3     REAL E(NP1),F(NP1),CR(N),Z(2),AI(2),IG(N)
      C W IS A WORK SPACE ARRAY OF LENGTH N+1 USED TO HOLD THE COEFFICIENTS
      C OF THE POLYNOMIAL FORMED FROM THE CALCULATED ROOTS.
      C F IS A WORKSPACE ARRAY OF LENGTH N+1 USED TO HOLD THE COEFFICIENTS
      C OF THE ERROR POLYNOMIAL.
      C IG IS A WORKSPACE ARRAY OF LENGTH N USED TO LIND TOGETHER THE ROOTS
      C IN A GROUP, THE FIRST IS IG1, IG(K) FOLLOWS K AND THE LAST IS I.
      C OTHER ROOTS HAVE IG(K)=0.
      C CR IS A WORKSPACE ARRAY OF LENGTH N USED TO HOLD DISTANCES FROM ROOT I
      C TO TH REST,
      C DATA EPS/1.E-6/,BIG/1.E70/
4     DATA EPS/2.3E-16/,BIG/1.E40/
      C BIG IS A NUMBER NEAR THE OVERFLOW LIMIT, EPS IS THE SMALLEST
      C NUMBER WHICH LEAVES UNITY UNCHANGED IN THE FLOATING-POINT ARITHMETIC
      C IN USE.
5     FACT=1.05**(1./FLOAT(N))
6     N1=N+1
      C
      C MULTIPLY OUT THE POLYNOMIAL FORMED FROM THE CALCULATED ROOTS
7     W(1,1)=A(1,N1)
8     W(2,1)=A(2,N1)
9     DO 10 I=1,N
10    W(1,I+1)=0.
11    W(2,I+1)=0.
12    II=N1-I
13    C(1)=-R(1,II)
14    C(2)=-R(2,II)
15    DO 10 JJ=1,I
16    J=I+1-JJ
17    W(1,J+1)=C(1)*W(1,J)-C(2)*W(2,J)+W(1,J+1)
18    W(2,J+1)=C(1)*W(2,J)+C(2)*W(1,J)+W(2,J+1)
      C
      C FIND COEFFICIENTS OF ERROR POLYNOMIAL
19    DO 20 I=1,N1
20    II=N+2-I
21    AI(1)=A(1,II)
22    AI(2)=A(2,II)
23    Z(1)=A(1,II)-W(1,I)
24    Z(2)=A(2,II)-W(2,I)
25    20 F(I)=CABS(CMPLX(Z(1),Z(2)))+AMAX1(E(II),CABS(CMPLX(AI(1),AI(2)))
      1*EPS)
26    DO 30 I=1,N
27    30 IG(I)=0
      C
      C MAIN ERROR-BOUNDING LOOP STARTS HERE AND EXTENDS TO THE END
      C OF THE SUBROUTINE IF FINDS A BOUND FOR ROOT I.
28    DO 130 I=1,N
29    III=1
30    M=1
31    IG(I)=-1
32    IG1=I
33    C(1)=R(1,I)
34    C(2)=R(2,I)

```

```

ISN   ST-NO           SOURCE PROGRAM      ( PA06CD )

35           D=0.
      C
      C FIND DISTANCES TO OTHER ROOTS
36           DO 32 K=1,N
37           Z(1)=R(1,K)-C(1)
38           Z(2)=R(2,K)-C(2)
39   32      CR(K)=CABS(CMPLX(Z(1),Z(2)))
      C
      C INITIALIZATION STATEMENTS WHICH ARE ALSO NEEDED WHEN M IS INCREASED
40   40      DMIN=BIG
41           Z(1)=C(1)
42           Z(2)=C(2)
43           S=CABS(CMPLX(Z(1),Z(2)))+D
44           RAD=D
      C TEST ROUCHE CONDITION WITH RADIUS RAD,
45   45      TOP=F(1)
46           PROD=1.
47           Z(1)=A(1,N1)
48           Z(2)=A(2,N1)
49           BTM=CABS(CMPLX(Z(1),Z(2)))
50           IJJ=0
51           DO 80 K=1,N
52           IF(IJJ) 61,61,62
53   61      IF(CABS(TOP).LT.1.E+60) GO TO 63
54           IJJ=1
55           TOP=TOP*1.E-40
56           BTM=BTM*1.E-40
57   62      TOP=S*TOP+F(K+1)*1.E-40
58           GO TO 64
59   63      TOP=S*TOP+F(K+1)
60   64      DIST=CR(K)
61           IF(IG(K).NE.0.)GO TO 70
62           BTM=BTM*(DIST-RAD)
63           IF(DMIN.LE.DIST)GO TO 80
64           DMIN=DIST
65           L=K
66           GO TO 80
67   70      PROD=PROD*(RAD-DIST)
68           IF(ABS(PROD).LT.1.E+60) GO TO 80
69           PROD=PROD*1.E-20
70           BTM=BTM*1.E+20
71   80      CONTINUE
72           IF(BTM.EQ.0.0)GO TO 100
73           TB=ABS(TOP/BTM)
74           IF(PROD.GE.TB)GO TO 110
      C FIND A NEW TRIAL RADIUS.
75           OLDS=S
76           OLDR=RAD
77           RL=RAD
78           RAD=D+1.1*TB
79           IF(M.EQ.1)GO TO 98
80           RM=D+(1.1*TB)**(1./FLOAT(M))
      C BISECTION LOOP. RL AND RM HOLD UPPER AND LOWER BOUNDS.
81   83      RAD=(RL+RM)/2.
82           PROD=1.
83           K=IG1
84   90      PROD=PROD*(RAD-CR(K))

```

ISN	ST-NO	SOURCE PROGRAM	(PA06CD)
85		K=FIX(IG(K))	
86		IIII=IIII+1	
87		IF(IIII.GT.100) GO TO 101	
88		IF(K.GT.0)GO TO 90	
89		IF(PROD.GE.1.10*TB)GO TO 94	
90		IF(RAD.GE.DMIN)GO TO 100	
91		IF(PROD.GT.1.05*TB)GO TO 98	
92		RL=RAD	
93		GO TO 83	
94	94	KM=RAD	
95		GO TO 83	
96	98	S=S+RAD-OLDR	
97		IF(RAD.GE.DMIN)GO TO 100	
98		IF(S*((DMIN-OLDR)/(DMIN-RAD)),LE.FACT*OLDS)GO TO 110	
99		GO TO 45	
		C	
		C ADD ROOT TO GROUP	
100	100	M=M+1	
101		IG(L)=FLOAT(IG1)	
102		IG1=L	
103		D=DMIN	
104		GO TO 40	
		C	
		C STORE ERROR BOUND AND RESET IG	
105	101	RAD=0.	
106	110	E(I)=RAD	
107	120	K=IG1	
108		IG1=FIX(IG(IG1))	
109		IG(K)=0.	
110		IF(K.NE.1)GO TO 120	
111	130	CONTINUE	
112		RETURN	
113		END	