JAERI-M
4465

高次代数方程式の多重根を求めるためのプログラム

(MROOT−ROOTP)

1971年6月

石黒美佐子

日本原子力研究所
Japan Atomic Energy Research Institute

高次代数方程式の多重根を求めるためのプログラム
(MROOT−ROOTP)

日本原子力研究所東海研究所原子炉工学部計算センター
石　黒　美　佐　子

(1971年6月受理)

要旨　実数を係数とする高次代数方程式のすべての根(虚根を含む)を求める問題をとりあげる。まず第1に重根を持つ代数方程式の根を正確に求めるための方法についてのべる。方程式を前もって多重度により因数分離し，重根を持たない方程式に対してBAIRSTOW 法等を適用することにより問題は解決できる。そのためにサブルーチンMROOTを作成した。次にFACOM230−60の科学計算用サブルーチンライブラリ(SSL)のBAIRSTOW法によるサブ・ルーチンの難点をあげ，これを改良したサブ・ルーチンROOTPを作成した。ROOTPを核にして，MROOTを適用した時の計算結果等を合わせてのべる。

Extraction of Multiple Roots of Algebraic Equation (MROOT-ROOTP)

Misako ISHIGURO

Div. of Reactor Engineering, Tokai, JAERI

(Received June 1971)

Abstract A new method is given for the numeric solution of algebraic equation. First we describe that the multiple roots of the equation are determined by existing methods (Ex.BAIRSTOW method and NEWTON method) with comparatively small accuracy. So, we provide the practical algorithm to extract the multiple roots accurately by the repeated use of Euclid Algorithm and give the computer program MROOT.

Then, we describe that the existing program by BAIRSTOW method in SSL (FACOM Scientific Subroutine Library) is not always effective even when all the roots are simple. And so we give more excellent program ROOTP by rewriting the ALGOL program in Collected Algorithm (ANL report). Numerical study of MROOT-ROOTP is also given.

目　　次

1 はじめに

n－次代数方程式

$$a_0 x^n + a_1 x^{n-1} \cdot\cdot\cdot\cdot\cdot + a_n = 0$$

はnケの根を持つ。ここでは実数を係数とする高次代数方程式のすべての根（虚根を含む）を求める問題を取り上げてみる。そのためによく知られている解法をしてBAIRSTOW 法（またはBAIRSTOW－HITCHCOCK 法）がある。これは代数方程式と2次式に分解して，2次方程式から根を求める方法である。しかしながら，重根を持つ方程式に対しては，この解法では根が正確に得られない。

以下にまず，BAIRSTOW 法の重根における問題点についてのべ，それを解決する方法として，代数方程式を前もって多重度により因数分離する。そうすれば，すべての代数方程式を解く問題は，重根を含まない代数方程式を解く問題に代えることができる。

次に現在FACOM230－60科学計算用サブルーチンライブラリ（SSL）に入っているBAIRSTOW 法により代数方程式を解くためのサブルーチン

BAIR1S（単精度）

BAIR1D（2倍精度）

そのものの問題点，つまり重根を含まない場合においても起こる問題点についてのべる。そしてこれを改良した方法として，Comm. ACMのAlgorithm の欄に記載されたプログラムをFORTRAN に翻訳したものを紹介する。

最後に同一の問題に対して，以下の4つの方法により得た計算結果を比較する。

BAIR1S	SSLのBAIRSTOW法
MROOT－BAIR1S	多重度の処理を前もって行なった上でBAIR1Sを適用する。
ROOTP	Comm. ACM のアルゴリズムに基づいて作成した改良型BAIRSTOW法
MROOT－ROOTP	多重度の処理を前もって行なった上でROOTPを適用する。

以上の計算結果から，BAIRSTOW 法の重根の処理（MROOTによる）と，BAIRSTOW 法のより優れた解法（ROOTPによる）を与えることにより，高次代数方程式のすべての根とその多重度を得るための種々の問題点が，本稿によりかなり解決されたことがわかるであろう。

この仕事は，筆者が多項式の整数の範囲での因数分解や有理式の部分分数分解の処理を行なうのに，BAIRSTOW 法を使用して代数方程式を解く必要に迫られ，代数方程式が正確に解けない場合があまりにも多いのに不便を感じ，少し掘下げて取り組もうとして始めたものである。

2 重根を含む代数方程式に対するBAIRSTOW法の問題点

2.1 BAIRSTOW 法

$$f(x)=a_0x^n+a_1x^{n-1}+\cdots\cdots+a_{n-1}x+a_n \qquad (1)$$

とおき，(1)式を2次式に分解して，2次方程式から求める方法である[1]。(1)式の $f(x)$ を2次式 $x^2+p\cdot x+q$ で除して次の関係を得る。

$$f(x)=(x^2+p\cdot x+q)\cdot Q(x)+r\cdot x+s \qquad (2)$$

もし，$r=s=0$ であれば $x^2+p\cdot x+q$ は $f(x)$ の因数である。したがって，p, q を独立変数と考えると，r, s は p, q の関数で，問題は

$$r(p,q)=0,\ s(p,q)=0 \qquad (3)$$

を満足する p，q を求めることである。これを解くには，2変数のNEWTON−RAPSON法による。2乗以上の項を無視すれば

$$\frac{\partial r}{\partial p}\Delta p+\frac{\partial r}{\partial q}\Delta q+r=0 \qquad (4)$$

$$\frac{\partial s}{\partial p}\Delta p+\frac{\partial r}{\partial q}\Delta q+s=0$$

を得る。

(4)式から Δp, Δq を求め，p，q の代りに，$p+\Delta p$, $q+\Delta q$ をそれぞれ次にためす値として選び，いままでの操作をくりかえす。これを数値的に解くには，(2)式の $Q(x)$ を $x^2+p\cdot x+q$ で除して

$$Q(x)=(x^2+p\cdot x+q)\cdot p(x)+u\cdot x+v \qquad (5)$$

を計算し，

$$\left\{\begin{aligned}\Delta p&=\frac{1}{\Delta}(r\cdot v-u\cdot s)\\ \Delta q&=\frac{1}{\Delta}\{s\cdot(v-p\cdot u)+q\cdot u\cdot r\}\\ \Delta&=v\cdot(v-p\cdot u)+q\,u^2\end{aligned}\right. \qquad (6)$$

を得る。したがって適当な近似値 p_0，q_0 から出発して

$$P_{n+1}=p_n+\Delta p,\qquad q_{n+1}=q_n+\Delta q \qquad (7)$$

をくりかえせば，$r \to 0$，$s \to 0$となる。f(x)の因数の1つとして$x^2+p \cdot x+q$が求められれば，f(x)を$x^2+p \cdot x+q$で除して，商に対して同様な試みを行ない，これらをくりかえす。以上のようにして，次数が偶数の代数方程式は2次式に分解される。f(x)の次数が奇数の場合には，NEWTON法などにより，実根を1つ求めた後でBAIRSTON法を適用する。そして2次方程式を解くことにより，虚根を含めてf(x)のすべての根を求めることができる。

2.2 BAIRSTOW法の問題点

f(x)の次数が偶数であるとして，重根を持つには二通りの場合が考えられる。第1は，f(x)が因数$x^2+p \cdot x+q$を持つとして，この2次式が重根を持つ場合である。第2は，f(x)の因数として，2次式の因数の累乗　$(x^2+p \cdot x+q)^k$，$k \geqq 2$を含む場合である。BAIRSTOW法で問題となるのは第2の場合である。

f(x)が$(x^2+p \cdot x+q)^k$，$k \geqq 2$を因数として含むと仮定する。(5)式を(2)式に代入する。

$$f(x)=(x^2+p \cdot x+q)\{(x^2+p \cdot x+q) \cdot p(x)+u \cdot x+v\}+r \cdot x+s \tag{8}$$

f(x)を微分すると，

$$\begin{aligned} f'(x)= & (x^2+p \cdot x+q)\{(x^2+p \cdot x+q) \cdot p'(x)+2(2x+p) \cdot p(x)+2u+v\} \\ & +(2v-p \cdot u)x+p \cdot v-2u \cdot q+r \end{aligned} \tag{9}$$

$(x^2+p \cdot x+q)^k$，$k \geqq 2$はf(x)の因数であるからf(x)もf'(x)も$x^2+p \cdot x+q$で割り切れる。したがって，

$$\begin{aligned} & r \to 0, \\ & s \to 0, \\ & 2 \cdot v-p \cdot u \to 0, \\ & p \cdot v-2 \cdot u \cdot q+r \to 0 \end{aligned} \tag{10}$$

となっている。したがって

$$p \doteqdot \frac{2 \cdot v}{u}, \quad q \doteqdot \frac{v^2}{u^2} \tag{11}$$

を得る。一方(6)式のΔは，

$\Delta=v \cdot (v-p \cdot u)+q \cdot u^2 \to 0$　となる。

つまりf(x)が$x^2+p \cdot x+q$の累乗を因数として含む場合には，BAIRSTOW法により近似解として因数$(x^2+p \cdot x+q)$を計算する過程で(6)式のΔp，Δqの計算をする際に，分母Δが非常に小さい値となり，数値計算上の誤差を含み易いのである。

事実筆者が，SSLのBAIRSTOW法により代数方程式を解くためのルーチンBAIR1S[2]を用いて重根を含む代数方程式を解いた場合に，表2の第2欄に示すように，大部分の方程式は解が得られなくて，途中で計算が打切られた。その原因の第1は（ILL= 2にて示す）一定限度以上のΔ（10^{-5}程度）を与える（p，q）に対してのみ(7)式の反復を実行するようになっているので，反復を何回くりかえしても（p，q）が収れんしないためである。すなわちs， rが一向に0に近づかないので，反復回数の制限を越えたというエラーメッセージにより計算が打切られる。第2の原因は，一定限度以上のΔを与える（p，q）のみの間で収れん条件を満たすものを探そうとすると，うまく見つからなくてだんだん（p，q）の値（絶対値）が大きくなり計算途中で浮動小数点overlowが続出するためである**。

表2の第3欄は，改良型BAIRSTOW法によるサブルーチンROOTPによる計算結果を示しているが，この場合BAIR1SのようにΔをある範囲以上にするという条件は取り除かれているので，いくら小さいΔに対しても（p，q）を選択できるようになっていて計算が途中で打切られるということはまれであるが，得られた結果の精度が非常に悪い。我々の実験では，解の有効桁を小数以下5桁と指定しているにかかわらず，小数以下1～2桁しか精度が得られないものもある。

重根を含む代数方程式に対するこの問題は，BAIRSTOW法のみならず，同じような解法を用いるNEWTON法で実根を求める場合にも起こる。

この場合，もっと直観的に現象を摑まえることができる。次式のくりかえしにより近似解x^*を得るが，

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \tag{12}$$

$f(x)=0$の根x^*の近傍では，x^*が重根の場合には，$f'(x^*)$も同時に0に近づくので，結局この反復方法では正確な根が得られない。このようにNEWTON法やBAIRSTOW法は重根を持たない代数方程式を解くための解法であることがわかる。

代数方程式の種々の解法のうちSTÜRMの定理を使用する方法等で，根の多重性に関係なく解が得られるものもあるが，多重性が取り除かれていることを暗目のうちに仮定しているものが多い。したがって数値計算を目的とした書物にも多重根の措置を明示しているものは少ない。代数方程式の解法を専問に取り扱った書物[3]で，NEWTON-RAPSONに立ち返えり，2乗以上の項を考慮し，(12)式の代わりに2重根の場合には，

$$f(x_n)+f'(x_n)(x_{n+1}-x_n)+\frac{1}{2}f''(x_n)(x_{n+1}-x_n)^2=0 \tag{13}$$

からx_{n+1}を計算し，

3重根の場合には，

$$f(x_n)+f'(x_n)(x_{n+1}-x_n)+\frac{1}{2}f''(x_n)(x_{n+1}-x_n)^2+\frac{1}{6}f'''(x_n)(x_{n+1}-x_n)^3=0 \tag{14}$$

からx_{n+1}を計算する方法が提案されている。

しかしながら結局多重性が，前もってわからない場合には，このように反復式を多重性により振り分けることは不可能である。その上(13)，(14)式からx_{n+1}を出すには，数値計算上の誤差が

** またBAIR1Sを2倍精度にしたもので同じくSSLの中のサブルーチンBAIR1Dを用いても，上記の2つの問題点は解決しない。

大きい。そして5重以上の根の場合には，ますます問題が難しくなる。一方NEWTON法はまだしも，上のように重根の場合の反復式が，かなり単純に記述できるが，p，q 2変数を取り扱うBAIRSTOW 法では，重根の場合の(6)式に代わる式がもっと複雑で，数値計算誤差も大きくなる。

結局，次章に示すように，多項式の多重性を前もって判定しておいて，単根しか持たない代数方程式を解くことにするのが，重根問題を解決する最もすぐれた方法であると考える。

3　多項式を多重度により因数分離するためのアルゴリズム

$p(x)$を多項式とする。$p(x)=0$を解くことを考えよう。

$$p(x)=p_1(x)\cdot p_2(x)^2\ \cdots\cdots\ p_\ell(x)^\ell\ p_m(x)^m,\qquad m\geqq 1 \tag{15}$$

とおいて，ℓ重の因数をまとめて$p_\ell(x)^\ell$と書く。ℓ重の因数がなければ$p_\ell(x)=1$とする。然らば$p_\ell(x)$は単根のみを有し，かつ，$p_1(x)$，$p_2(x)$，・・・・，$p_m(x)$は，それぞれ互いに素な多項式である。

さて，$p(x)$と$p(x)$の微分$p'(x)$の最大公約多項式を$Q_1(x)$とおくと，(15)式から

$$Q_1(x)=p_2(x)\cdot p_3(x)^2\ \cdots\cdots\ p_\ell(x)^{\ell-1}\ \cdots\ p_m(x)^{m-1}$$

となる。

$Q_1(x)$と$Q_1'(x)$の最大公約多項式を$Q_2(x)$とおく。

$$Q_2(x)=p_3(x)\cdot p_4(x)^2\ \cdots\cdots\ p_\ell(x)^{\ell-2}\ \cdots\cdots\ p_m(x)^{m-2}$$

これをくりかえすと，

$$Q_i(x)=P_{i+1}(x)\cdot p_{i+2}(x)\cdots\cdots\ p_\ell(x)^{\ell-2}\ \cdots\cdots\ p_m(x)^{m-i}$$

となり，

$$Q_{m-1}(x)=p_m(x)$$

$Q_m(x)$は，もはや定数となる。

上のような仮定のもとに，多項式$p(x)$を(15)式のように，多重度により因数を分離し，$p_1(x)$，$p_2(x)$，・・・，$p_m(x)$を得るためのアルゴリズムは次のようになる。

$$\begin{cases} Q_0(x)=p(x) \\ Q_\ell(x)=GCD\{Q_{\ell-1}(x),\ Q'_{\ell-1}(x)\},\ \ell=1,2,\cdots,m-1 \\ Q_m(x)\text{は定数}\ (=1) \end{cases}$$

$$\begin{cases} R_\ell(x)=Q_{\ell-1}(x)/Q_\ell(x), & \ell=1,2,\cdots,m-1 \\ R_m(x)=Q_{m-1}(x) \end{cases}$$

$$\begin{cases} P_{\ell-1}(x)=R_{\ell-1}(x)/R_\ell(x), & \ell=2,3,\cdots,m-1 \\ p_m(x)=R_m(x) \end{cases} \tag{16}$$

結局，重根を持つ代数方程式$p(x)=0$を解く問題は，(16)式から$p_1(x)$，$p_2(x)$，・・・，$p_m(x)$を算出し，

$$p_\ell(x)=0,\ \ell=1,2,\cdots\cdots,m \tag{17}$$

（但し$p_k(x)=1$のものを除く）

を解く問題に代わる。

(17)式の$P_{\ell}(x)=0$は，仮定から単根しか持たないので，BAIRSTOW 法を用いれば問題なく解を得ることができる。

注GCD(P, Q)は多項式P，Qの最大公約多項式のことである。

4　多項式を多重度により因数分離するためのプログラム：MROOT

4.1　目　的

MROOTは，代数方程式$A(x)=0$が解かれて，

$$A(x)=(x-a_1)^{n_1}\cdot(x-a_2)^{n_2}\cdots(x-a_N)^{n_N}$$
$$(x-\alpha_1)^{m_1}\cdot(x-\alpha_2)^{m_2}\cdots(x-\alpha_n)^{m_M} \tag{18}$$

を満たすような実根a_1，a_2……，a_N 虚根α_1，α_2，…，α_M と根の多重度それぞれn_1，n_2…，n_N とm_1，m_2，…，m_M を得ることを目的としている。プログラムは，BAIRSTOW 法により代数方程式を解くためのプログラム，たとえばSSL のBAIR1 S，BAIR1Dまたは，次に紹介するROOTP：に対するプレ・プロセッサ（Pre-processor）の形式で，(16)式のアルゴリズムが前に付加されている。

4.2　プログラムとフロチャート

図1にMROOTのフローチャート，付録1にFORTRAN によるプログラム・リストを示す。フローチャートおよびプログラムで使用している記号は，表1のとおりである。

多項式の演算は，すべて係数を高べキの順に入れて行なっている。

4.3　使用方法

サブルーチンは，次のようにして呼び出す。

```
CALL MROOT(A, M, R1, M1, N1, R2, M2, N2)
```

変数の意味は，表1のプログラム欄を参照されたい。

プログラムは一応最大次数が19まで（係数の合計は20）の代数方程式が解けるようになっている。

4.4　MROOTが参照するサブルーチン

PODIF　；多項式の微係数を求める。（プログラムリストは付録1）

POGCD　；2つの多項式の最大公約多項式を求める。互に素な時は1となる。（付録1）

PODIVS；多項式の割算を実行し，商と剰余を得る。（SSL）

ROOTP　；BAIRSTOW 法に代数方程式を解く。（付録2）

上記のサブルーチンのうち，PODIFの作成は簡単である。POGCDは，SSL の中に多項式の四則演算のためのサブルーチンが用意されているので，それらを用いて，EUCLIDの互除法により作成している。PODIVSはSSL の中にあるものを使用している。ROOTPは次章

(17)式の$P_\ell(x)=0$ は，仮定から単根しか持たないので，BAIRSTOW 法を用いれば問題なく解を得ることができる。

注 GCD(P, Q) は多項式P，Qの最大公約多項式のことである。

4　多項式を多重度により因数分離するためのプログラム：MROOT

4.1　目　的

MROOTは，代数方程式$A(x)=0$が解かれて，

$$A(x)=(x-a_1)^{n_1}\cdot(x-a_2)^{n_2}\cdots(x-a_N)^{n_N}$$
$$(x-\alpha_1)^{m_1}\cdot(x-\alpha_2)^{m_2}\cdots(x-\alpha_n)^{m_M} \tag{18}$$

を満たすような実根$a_1, a_2 \cdots\cdots, a_N$ 虚根$\alpha_1, \alpha_2, \cdots, \alpha_M$ と根の多重度それぞれ$n_1, n_2\cdots, n_N$ と$m_1, m_2, \cdots, m_M$ を得ることを目的としている。プログラムは，BAIRSTOW 法により代数方程式を解くためのプログラム，たとえばSSL のBAIR1S，BAIR1Dまたは，次に紹介するROOTP：に対するプレ・プロセッサ（Pre-processor）の形式で，(16)式のアルゴリズムが前に付加されている。

4.2　プログラムとフロチャート

図1にMROOTのフローチャート，付録1にFORTRAN によるプログラム・リストを示す。フローチャートおよびプログラムで使用している記号は，表1のとおりである。

多項式の演算は，すべて係数を高べキの順に入れて行なっている。

4.3　使用方法

サブルーチンは，次のようにして呼び出す。

CALL MROOT(A, M, R1, M1, N1, R2, M2, N2)

変数の意味は，表1のプログラム欄を参照されたい。

プログラムは一応最大次数が19まで（係数の合計は20）の代数方程式が解けるようになっている。

4.4　MROOTが参照するサブルーチン

PODIF　；多項式の微係数を求める。（プログラムリストは付録1）

POGCD　；2つの多項式の最大公約多項式を求める。互に素な時は1となる。（付録1）

PODIVS；多項式の割算を実行し，商と剰余を得る。（SSL）

ROOTP　；BAIRSTOW 法に代数方程式を解く。（付録2）

上記のサブルーチンのうち，PODIFの作成は簡単である。POGCDは，SSL の中に多項式の四則演算のためのサブルーチンが用意されているので，それらを用いて，EUCLIDの互除法により作成している。PODIVSはSSL の中にあるものを使用している。ROOTPは次章

表 1

フローチャート	プログラム	記号の用途
A	配列 A	入力多項式 A(x)
m	M	入力多項式の次数+1
R_1	配列 R1	A(x)=0の実根，すなわち(18)式の a_1, a_2, ・・・, a_N
M_1	配列 M1	実根の多重度，すなわち n_1, n_2, ・・・, n_N
N	N1	実根の数
R_2	配列 R2 (虚根の数×2)	虚根 α_1, α_2, ・・・, α_M が実部と虚部とに分かれて入る
M_2	配列 M2	虚根の多重度，すなわち m_1, m_2, ・・・, m_M
M	N2	虚根の数
Q	配列 Q	(16)式の Q(x) old*
Q′	配列 Q1	Q(x) old の微分
m_Q	MQ	Q(x) old の次数+1
Q_0	配列 Q0	(16)式の Q(x) new**
J	J	Q(x) new** の次数+1
R	配列 R	(16)式の R(x) old
m_R	MR	R(x) old の次数+1
R_0	配列 R0	(16)式の R(x) new
m_0	M0	R(x) new の次数+1
P_ℓ	2次元配列 P	(16)式の $P_\ell(x)$
$m_{d,\ell}$	配列 MD	$P_\ell(x)$の次数+1
ε	Eps	根の精度（≒10^{-5} 程度）で，方程式の最高次の係数を1になるようにノーマライズしたとき，根の精度を小数以下何桁かを指定する
R_p	配列 RP	BAIRSTOW法で方程式を解いたとき，
C_p	配列 CP	R_p には根の実部が入り，C_pには虚部が入る。実根の場合は $C_p=0$ となる

＊ old とは(16)式の反復計算における前回の値

＊＊ NEW とは(16)式の反復計算における今回の値

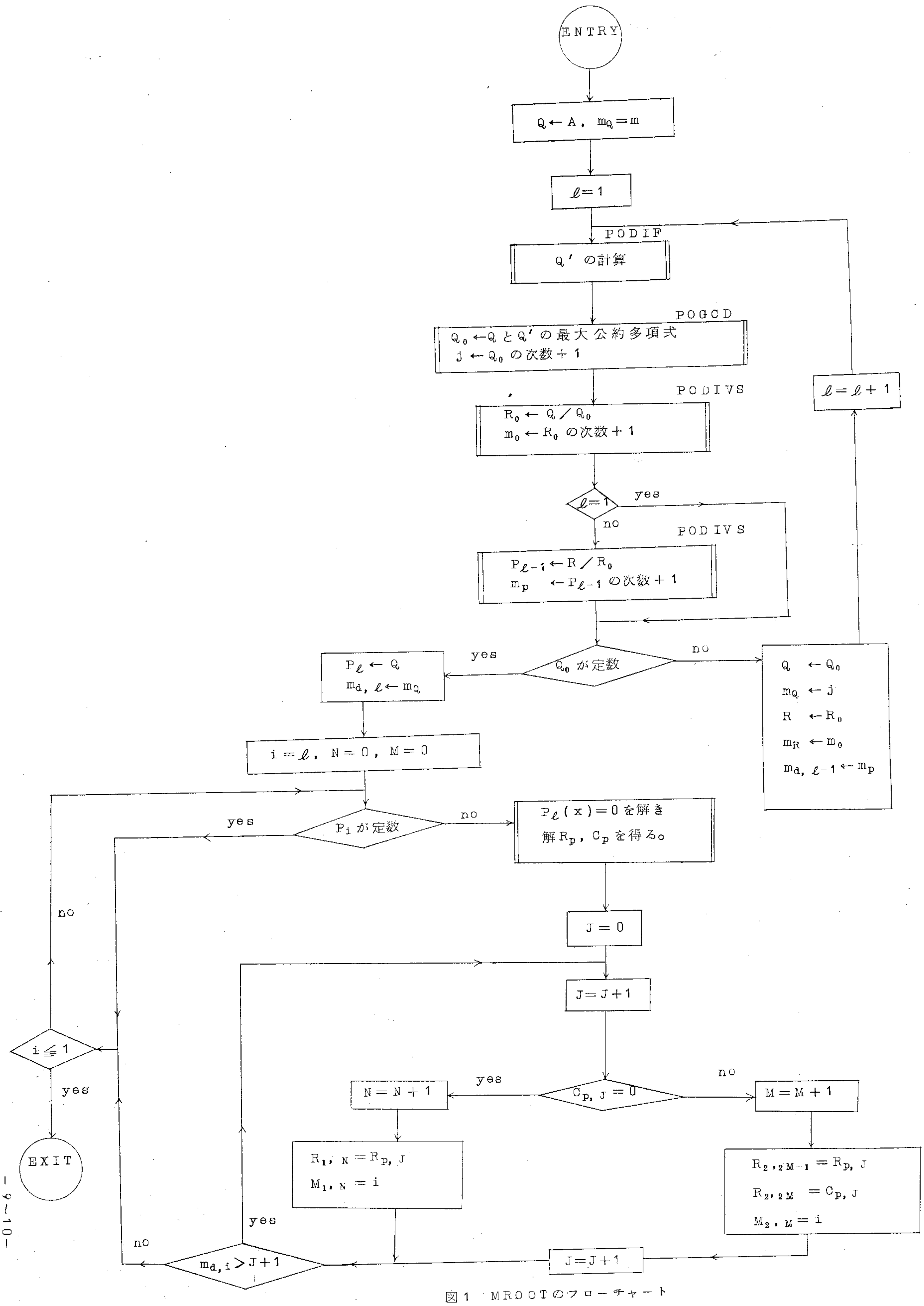

図1 MROOTのフローチャート

に述べるが，改良型BAIRSTOW 法である。ROOTPとSSL の中のBAIRSTOW 法のサブルーチンBAIR1S，BAIR1D とは変数の与え方が同一なので，ROOTPの代わりにBAIR1Sを使用してもよい。しかしROOTPの方がより優れたサブルーチンである。ROOTPまたはBAIR1S等核になるサブルーチンでILL CONDITION に出く合したときは，その理由が印刷され，演算の実行は打切られる。

4.5 計算結果

表2の第3欄，第5欄がMROOTを使用した場合の計算結果である。第3欄はBAIR1Sを核にして解いた例である。BAIR1Sのみを適用した場合13問中12問が，重根の処理が悪くて第2欄に示す理由により解けなかったのに，MROOTと組み合わせると，第3欄で見るように，13問すべて解が得られ誤差も，指定された範囲内である。第5欄はROOTPを核に解いた例である。ROOTPのみを適用すると，第4欄で見るように，一応解は得られるものの解の精度が非常に悪い。小数点以下1桁程度しか有効でないものもある。MRDOTと組合せることにより重根を避けたのが第5欄で，第3欄同様すべて正解が得られる。

表2　重根を含む代数方程式の解

サブルーチン cpu使用時間	BAIR1S	ms	MROOT－BAIR1S	ms	ROOTP	ms	MROOT－ROOTP	
x^4+4x^4+4 $=(x^2+2)^2$	ILL＝2		±1.4142i (2)	8	0.00209±1.4168i 0.00209±1.4116i	13	±1.4142i (2)	11
x^6-12x^2-16 $=(x-2)(x+2)(x^2+2)^2$	ILL＝2		±2.0000 ±1.4142i (2)	11	±2.0000 0.00125±1.4160i (2)	22	±2.0000 ±1.4142i (2)	14
x^4-2x^2+1 $=(x-1)^2(x+1)^2$	±1.0000 (2)	234	±1.0000 (2)	8	±1.000 (2)	6	±1.0000 (2)	9
$x^6+4x^4+5x^2+2$ $=(x^2+1)^2(x^2+2)$	OVERFLOW		±1.0000i (2) ±1.4142i	14	0.00591±1.0059i −0.00577±0.99411i 0.000138±1.4142i	24	±1.0000i (2) ±1.4142i	15
$x^8-x^7+7x^6-6x^5+18x^4$ $-12x^3+2x^2-8x+8$ $=(x^2+2)^3(x^2-x+1)$	OVERFLOW		±1.4142i (3) 0.50000±0.86602i	22	−0.01260±1.4159i 0.00758±1.4244i 0.00502±1.4023i 0.50000±0.86602i	44	±1.4142i (3) 0.5000±0.86602i	23
$x^4+2x^3+3x^2+2x+1$ $=(x^2+x+1)^2$	ILL＝2		−0.50000±0.86603i (2)	8	−0.50115±0.86753i −0.49885±0.86452i	15	−0.50000±0.86603i (2)	8
x^3+3x^2+3x+1 $=(x+1)^3$	ILL＝1		−1.0000 (3)	7	−1.0087±0.01502i −0.98266	13	−1.0000 (3)	7
$x^5-x^4+x^3-x^2-2x+2$ $=(x-1)^2(x+1)(x^2+2)$	OVERFLOW		1.0000 (2) −1.0000 ±1.4142i	12	1.0000 (2) −1.0000 ±1.4142i	11	1.0000 (2) −1.0000 ±1.4142i	14
$x^7+x^6+x^5-x^4-x^3-x^2$ $=x^2(x-1)(x^2+x+1)$	ILL＝2		0.0000 (2) 1.0000 −0.5000±0.86603i (2)	18	0.0000 (2) 1.0000 −0.49974±0.86665i −0.50026±0.86540i	44	0.0000 (2) 1.0000 −0.50000±0.86603i (2)	16
$x^8-x^7+6x^6-5x^5+13x^4$ $-8x^3+12x^2-4x+4$ $=(x^2+2)^2(x^2-x+1)$ (x^2+1)	ILL＝2		±1.4142i (2) 0.50000±0.86602i ±1.0000i	28	0.00014±1.4136i −0.00014±1.4149i 0.50000±0.86603i ±1.[illegible]000i	47	±1.4142i (2) 0.50000±0.86602i ±1.0000i	32
$x^4-2x^3+3x^2-2x+1$ $=(x^2-x+1)^2$	ILL＝2		0.50000±0.86603i (2)	8	0.500115±0.86752i 0.49885 ±0.86452i	12	0.50000±0.866603i (2)	9
$x^8+2x^6+3x^4+2x^2+1$ $=(x^2+x+1)^2(x^2-x)^2$	OVERFLOW		±0.50000±0.86603i (2)	20	−0.50044±0.86670i −0.49956±0.86536i 0.50004±0.86498i 0.49996±0.86708i	393	±0.50000±0.86603i (2)	20
$12x^5+12x^4+24x^3+24x^2$ $=12x^2(x+1)^2(x^2+2)^2$	OVERFLOW		0.0000 (2) −1.0000 ±1.4142i	12	0.0000 (2) −1.0000 ±1.4142i	5	0.0000 (2) −1.0000 ±1.4142i	14

(　)は根の多重度　　ILL＝2，OVERFLOWについては，2.2節を参照のこと　　根の精度はいづれも10^{-5}を指定

5 改良型BAIRSTOW法：ROOTP

5.1 BAIR1Sの問題点

BAIRSTOW 法の解法の大すじは，すでに2.1節に示した。解法の大すじは大きく変らないが細かい考慮がなされるかどうかにより方程式が解けたり解けなかったりする。SSLの富士通提供のサブルーチンBAIR1Sは，たちの良い問題については解を得ることができるが，たちの悪いものは，ILL CONDITION で打切られたり，浮動小数OVERFLOW になったりしてたちまち解けなくなる。BAIR1S で解けない例として次式を挙げる。これは浮動小数OVERFLOW となってしまう。

$$1.2073\times10^{14}x^6+4.9204\times10^{10}x^5+2.8646\times10^{10}x^4+3.8998\times10^6x^3+5.3398\times10^5x^2+1.9527\times10x+1.0000=0 \tag{19}$$

根は小数以下2桁程度のいづれも単根で虚根となるが，BAIR1S（BAIR1Dも）では，演算途中で大きな数に出くわすのでOVERFLOW になる。

また次式はこれも単根の問題であるがILL CONDITION=2（§2.2参照）にて打切られる。

$$x^7+x^6+x^5+x^4+x^3+x^2+x+1=0 \tag{20}$$

解けない原因を解消したのがサブルーチン：ROOTPである。このプログラムは文献[4]にアルゴルで書かれているものを一部分修正してFORTRANに書換えたものである。BAIR1Sと主たる違いは，方程式の係数を，OVERFLOW が起こらないようにノーマライズし，その上場合によっては，根の逆数を求めるように問題を変更したりしている。

ノーマライズの方法は，原本に少し修正を加えて，次のようになっている。まず方程式

$$a_0x^n+a_1x^{n-1}+\cdots\cdots+a_{n-1}x+a_n=0 \tag{21}$$

を $b_0x^n+b_1x^{n-1}+\cdots\cdots+b_{n-1}x+b_n=0$,

$$b_i=a_i/a_0 \tag{22}$$

と書換え，

$$s=\sum_{i=1}^{n}\log_{10}b_i \tag{23}$$

を計算し，

$$c_0x^n+c_1x^{n-1}+\cdots\cdots+c_{n-1}x+c_n=0,\qquad c_i=b_i/s \tag{24}$$

とする。

そして次の条件のときは，

$$c_1/c_0 < c_{n-1}/c_n \tag{25}$$

(24)式の係数の順序を入れかえて，

$$c_n y^n + c_{n-1} y^{n-1} + \cdots\cdots + c_1 y + c_0 = 0 \tag{26}$$

を解くことにする。そして解y^*を得たら，その逆数

$$x^* = \frac{1}{y^*}$$

をもとの方程式(24)式の根とする。

また，BAIR1Sは与えられた方程式を2次式に分解することを主眼にしているが，ROOTPではBAIRSTOW 法とNEWTON法を同時に用いて，どちらかの収れん条件を満たすまで反復を行なう。この場合，NEWTON法により収れんしたときは実根が得られ，BAIRSTOW 法からは2次式が得られ，2次式から2根を求めることになる。

その上，反復は，2次式$x^2 + px + q$ の係数p，q を求めるのに，BAIRSTOW 法の決まりにより，Δp，Δqを得て，

$$p_{n+1} \leftarrow p_n + \Delta p, \quad q_{n+1} \leftarrow q_n + \Delta q$$

が収れんするまで行なうが，Δp，Δqがうまく求められない場合の処理を，いろんな場合について考慮が払われている。もちろんBAIR1Sで起った2.2節のΔが小さすぎて正確な解が得られない場合の問題も解決されている。

ROOTPのFORTRANによるプログラムは付録2にて示す。

5.2 使用方法

ROOTPの呼び出し方法はBAIR1Sと全く同じである。

CALL ROOTP(A, M, Eps, RP, cp, ILL) 変数の意味は，表1のプログラム欄を参照されたい。

5.3 計算結果

(19)式，(20)式はいずれもBAIRIS（またはRAIR1D）をEPS=10^{-5}として解いた場合，浮動小数点OVERFLOW により解けなかったが，ROOTPを使用すると付録3のように解を得ることができた。入力方程式，それから得られる方程式，使用時間，根の実部，虚部，ILL CONDITION の順に出力されている。

6. まとめ

6.1 計算結果および計算時間の比較

5.1および5.3で述べたとおり，単根のみを持つ代数方程式に対してもBAIR1Sでは解けない場合があり，ROOTPの方が優れている。

また重根を持つ方程式の場合，BAIR1Sでは解けない場合が多い。ROOTPを使用しても一応解は得るが精度が悪い。結局MROOTによるしかない。

計算時間の上では，BAIR1Sの方がROOTPより1割程度速いがあまり差はない。重根を含む方程式を無理に，BAIR1SまたはROOTPで解いて仮りに正解が得られたとしても，MROOTを先行させ多重度を前もって判定する方が，収れんが速いので，時間的には有利な場合が多い。

結局，重根がないことがはっきりしている場合は，ROOTPにより解き，重根の出る可能性のある場合は，MROOT−ROOTPによるのが最も良いと思う。また根の精度をEPS を与えたが小さすぎて収れんしないことも考えられるのでこの場合はEPS を1桁程度大きくして（許される範囲内で）解いてみるとよい。

6.2 数式処理との関連

実数を係数とする1変数多項式は，1次式かまたはその累乗，2次式かまたはその累乗に，実数の範囲で因数分解できることが知られている。

$$P(x)=c\prod_{k=1}^{N}(x+a_k)^{n_k}\cdot\prod_{k=1}^{M}(x^2+p_k x+q_k)^{m_k}\ ,\quad c\text{は整数} \tag{27}$$

多項式P(x)を整数の範囲で因数分解することを考えよう。いろいろ方法があるが，1変数多項式の場合は，Root−finding techniqueが効率がよい[6]。(27)式を

$$P(x)=c\prod_{k=1}^{L}P_k(x) \tag{28}$$

とおく。ここで$P_k(x)$は1次式かまたは2次式でその上$k\neq k'$が$P_k(x)=P_{k'}(x)$となることもある。

一方cを素因子分解して

$$c=c_1\cdot c_2\cdot\cdots\cdot c_I \tag{29}$$

を得る。ここで$i\neq i'$のとき$c_i=c_{i'}$のこともある。$C(n_1, n_2\cdots, n_J, x_1, x_2\cdots, x_k)$を次のように定義する。

$$\begin{aligned}&c(n_1, n_2, \cdots, n_J, x_1, x_2, \cdots, x_k)\\&=x_{n_1}\cdot x_{n_2}\cdots\cdot x_{n_J}\end{aligned}$$

ここで，$J\leqq K$，$1\leqq n_1<n_2<\cdots<n_J\leqq K$ (30)

そして，計算式

$$C(i_1, i_2, \cdots, i_J, c_1, c_2, \cdots c_I)\cdot C(\ell_1, \ell_2, \cdots, \ell_H, P_1(x), P_2(x), \cdots, P_L(x)) \tag{31}$$

$$1 \leqq J \leqq I,$$

$$1 \leqq H \leqq (L+1)/2$$

なるすべての $(i_1, i_2, \cdots, i_J)$，$(\ell_1, \ell_2, \cdots, \ell_H)$ の組合せの中から(31)式が整数を係数とする多項式となるものを探す。Hを1から小さい方から順に試すことにする。こうして見つけたものを $Q_1(x)$ とすれば，$Q_1(x)$ は整係数多項式で $P(x)$ の1つの因数となっている。また $Q_1(x)$ は次数の低いものから順に探がすので既約性は保証される。1つの因数 $Q_1(x)$ が得られたら $P(x)$ を $Q_1(x)$ で割りその商を再び $P(x)$ と置けば，$P(x)$ の次数が下がる。そして(31)式を試し，次の因数を探がす。こうして次式を得る。

$$P(x) = Q_1(x)\cdot Q_2(x)\cdot \cdots, Q_k(x) \tag{32}$$

ここで $k > k'$ なら $Q_k(x)$ は次数が $Q_{k'}(x)$ の次数より小さくなく整数の範囲で既約な整係数多項式である。

次に実数係数1変数有理式の部分分数分解を考えよう[7]。。有理式の部分分数定理から，有理式 $R(x)$ は次のように部分分数が分解される。

$$R(x) = F(x) + \sum_{k=1}^{N}\sum_{i=1}^{n_k} \frac{a_{k,i}}{(x+\alpha_k)^i} + \sum_{k=1}^{M}\sum_{i=1}^{m_k} \frac{b_{k,i}x + c_{k,i}}{(x^2+p_k x+q_k)^i}$$

$a_{k,i}$, α_k, $b_{k,i}$, $c_{k,i}$, p_k, q_k はすべて実数

ここで部分分数の分母は，有理式 $R(x)$ を $P(x)/Q(x)$ と書いたとき（$P(x)$ と $Q(x)$ は互に素），$Q(x)$ の因数である。また $F(x)$ は，$P(x)$ の方が $Q(x)$ より次数が高いとき，$P(x)$ を $Q(x)$ で割った商である。

参　考　文　献

〔1〕 電子計算機ハンドブックP5－45～5－47，情報処理学会編。

〔2〕 FACOM SSL(科学計算用サブルーチン・ライブラリ)解法解説書P32～39。

〔3〕 Zaguskin V.L., Handbook of numerical methods for the solution of algebraic and transcendental equation, Pergamon press.

〔4〕 K.W. Ellenberger, Algorithm 30, Numerical solution of the polynomical equation. ANN7054 P18－19

〔5〕 高木貞治，解析概論P244－245，岩波書店。

〔6〕 石黒美佐子，多項式の数式処理と因数分解の試み，第11回プログラミングシンポジウム報告集，1970年1月

〔7〕 石黒美佐子，有理式の数式処理の標準型，第12回プログラミングシンポジウム報告集，1971年1月。

```
      SUBROUTINE  MROOT(A,M,R1,M1,N1,R2,M2,N2,EPS)
C   SOLUVE THE ALGEBRAIC EQUATION
C    INPUT POLYNOMIAL IS
C            A(1)*X**(M-1)+A(2)*X**(M-2)+.....+A(M-1)*X+A(M) .
C     RESULT IS
C        (X-R1(1))**M1(1)*(X-R1(2))**M1(2)*......*(X-R1(N1))**M1(N1)*
C  (X-CMPLX(R2(1),R2(2)))**M2(1)*(X-CMPLX(R2(3),R2(4)))**M2(2)*.....*
C   (X-CMPLX(R2(2*N2-1),R2(2*N2)))**M2(N2)
C     M1(I).GE.M1(I+1)
C     M2(I).GE.M2(I+1)
      DIMENSION A(M),R1(M),M1(M),R2(M),M2(M),Q(20),Q0(20),Q1(20),R(40)
     1RO(20),P(20,20),MD(20)
C     INITIALIZE
      DO 1 I=1,M
    1 Q(I)=A(I)
      MQ=M
      L=1
      N=MQ-1
C     GET DERIVATIVE Q(L-1)'
      CALL PODIF(Q,MQ,Q1,N)
C     GET  Q(L)=GCD(Q(L-1),Q(L-1)')
      CALL POGCD(Q,MQ,Q1,N,Q0,J)
      DO 41 I=1,J
      IF(ABS(Q0(I)).GT.    EPS) GO TO 41
      Q0(I)=0.
   41 CONTINUE
C     IF Q(L) IS CONSTANT
C    GET RO(I)=Q(L-1)/Q(L)
C   IF Q0 IS CONSTANT
      IF(J.LE.1) GO TO 35
      MO=MQ+J
      CALL PODIVS(Q,MQ,Q0,J,RO,MO,ILL)
      IF(ILL.NE.0) GO TO 18
      DO 42 I=1,MO
      IF(ABS(RO(I)).GT.    EPS) GO TO 42
      RO(I)=0.
   42 CONTINUE
C     IF REMAINDER IS 0.
      NZ=0
      J1=J-1
      DO 2 I=1,J1
      M4=MO+I
      IF(ABS(RO(M4)).LT.EPS)GO TO 2
      NZ=1
    2 CONTINUE
      IF(NZ.EQ.1) GO TO 16
   37 CONTINUE
      IF(L.EQ.1) GO TO 4
C     GET P(L-1)=R(L-1)/R(L)
C   IF RO IS CONSTANT
      IF(MO.LE.1) GO TO 38
      MP=MR+MO
      CALL PODIVS(R,MR,RO,MO,P(1,L-1),MP,ILL)
      IF(ILL.NE.0) GO TO 22
      DO 43 I=1,MP
      IF(ABS(P(I,L-1)).GT. EPS) GO TO 43
      P(I,L-1)=0.
   43 CONTINUE
C     IF REMAINDER IS 0
      NZ=0


      J1=MO-1
      DO 31 I=1,J1
      M4=MP+I
      IF(ABS(P(M4,L-1)).LT.EPS) GO TO 31
      NZ=1
   31 CONTINUE
      MD(L-1)=MP
      IF(NZ.EQ.1) GO TO 16
    4 CONTINUE
      IF(J.LE.1) GO TO 3
      DO 5 I=1,J
    5 Q(I)=Q0(I)
      MQ=J
      DO 6 I=1,MO
    6 R(I)=RO(I)
      MR=MO
      MD(L)=MP
      L=L+1
      GO TO 8
C     P(L)=Q(L-1)
    3 DO 7 I=1,MQ
    7 P(I,L)=Q(I)
      MD(L)=MQ
      MDL=MD(L)
C     SOLVE THE ALGEBRAIC EQUATION P(1),P(2),.....,P(L).
      DIMENSION RP(20),CP(20)
      N1=0
      N2=0
      I=L
   10 CONTINUE
C     SOLVE THE P(I)=0.
      IF(MD(I).LE.1)  GO TO 32
      MDJ=MD(I)-1
      IF(MD(I).EQ.2) GO TO 33
      CALL  ROOTP(P(1,I),MD(I),EPS,RP,CP,ILL)
      IF(ILL.NE.0)  GO TO 44
C     SEPARATE REAL AND COMPLEX
   55 CONTINUE
      J=0
   11 CONTINUE
C     IF CP=0
      J=J+1
      IF(CP(  J).NE.0.  ) GO TO 12
C     REAL FACTOR
      N1=N1+1
      R1(N1)=RP(J)
      M1(N1)=I
      IF(J.LT.MDJ) GO TO 11
      GO TO 32
   33 RP(1)=-P(2,I)
      CP(1)=0.
      GO TO 55
C     COMPLEX
   12 N2=N2+1
      R2(2*N2-1)=RP(J)
      R2(2*N2)=CP(J)
      M2(N2)=I
      IF(J.LT.MDJ) GO TO 11
   32 IF(I.LE.1) GO TO 13
      I=I-1
```

```
      GO TO 10
   13 CONTINUE
      RETURN
   35 MO=MQ
      DO 36 I=1,MO
   36 RO(I)=Q(I)
      GO TO 37
   38 MP=MR
      DO 39 I=1,MP
   39 P(I+L-1)=R(I)
      GO TO 4
C     ERRORS
   16 WRITE(6,17)
   17 FORMAT(1H1,' MROOT ERROR 1')
      CALL EXIT
   20 FORMAT(1H0,I6/(1H ,8E15.6))
   18 WRITE(6,19)
   19 FORMAT(1H1,' MROOT ERROR 2   PODIVS ERROR'/)
   24 WRITE(6,21) ILL
   21 FORMAT(1H0,'  ILL =',I6)
      CALL EXIT
   22 WRITE(6,23)
   23 FORMAT(1H1,' MROOT ERROR 3   PODIVS ERROR' /)
      GO TO 24
   44 WRITE(6,45)
   45 FORMAT(1H1,' MROOT ERROR 4 BAIRIS ERROR'/)
      WRITE(6,21) ILL
      CALL EXIT
      END



C
      SUBROUTINE PODIF (A,M,B,N)
C     POLYNOMIAL DIFFERENTIATION
      DIMENSION A(M),B(N)
      N=1
      B(1)=0.
      IF(M.LE.1) RETURN
      N=M-1
      DO 1  I=1,N
    1 B(I )=A(I    )*FLOAT( M-I)
      RETURN
      END



C
      SUBROUTINE POGCD(A,L,B,M,C,N)
C     COMPUTES THE GREATEST COMMON DIVISOR OF POLYNOMIALS WITH SINGLE
C    VARIABLE
C     COEFFICIENTS OF INPUT POLYNOMIALS ARE A(1),A(2),     ,A(L)
C                                       AND B(1),B(2),     ,B(M).
C     COEFFICIENTS OF RESULT ARE C(1),C(2),       ,C(N),WHERE C(1)=1.
      DIMENSION A(L),B(M),C(20),P(20),Q(20),R(40)
      ILL=0
```

付録1　サブルーチンMROOT

```
C     IF INPUT IS CONSTANT,RESULT=1.
      IF(L.EQ.1.OR.M.EQ.1) GO TO 11
      EPS=1.0E-5
C     INITIALIZE
      A1=A(1)
      B1=B(1)
      IF(L.LT.M) GO TO 7
      DO 1 I=1,L
    1 P(I)=A(I)/A1
      DO 2 I=1,M
    2 Q(I)=B(I)/B1
      NP=L
      NQ=M
      GO TO 10
    7 DO 8 I=1,M
    8 P(I)=B(I)/B1
      DO 9 I=1,L
    9 Q(I)=A(I)/A1
      NP=M
      NQ=L
   10 IF(NQ.LE.1) GO TO 11
C     EUCLIDEAN DIVISION
C     NORMALISE
      P1=P(1)
      DO 3 I=1,NP
    3 P(I)=P(I)/P1
      Q1=Q(1)
      DO 4 I=1,NQ
    4 Q(I)=Q(I)/Q1
      NR=NP+NQ
      CALL PODIVS(P,NP,Q,NQ,R,NR,ILL)
      IF(ILL.NE.0) GO TO 12
      DO 5 I=1,NQ
    5 P(I)=Q(I)
      NP=NQ
      NQ=NQ-1
      NZ=0
      DO 6 I=1,NQ
      IF(NZ.NE.0) GO TO 15
      IF(ABS(R(NR+I)).LE.EPS) GO TO 6
   15 NZ=NZ+1
      Q(NZ)=R(NR+I)
    6 CONTINUE
      IF(NZ.EQ.0) GO TO 16
      NQ=NZ
      GO TO 10
   11 N=1
      C(1)=1.
      RETURN
   12 WRITE(6,14) ILL,NP,NQ
      WRITE(6,13) (P(I),I=1,NP)
      WRITE(6,13) (Q(I),I=1,NQ)
      CALL EXIT
   16 N=NP
      DO 17 I=1,NP
   17 C(I)=P(I)
      RETURN
   13 FORMAT(1H ,8E15.5)
   14 FORMAT(1H1,'POGCD ERROR ILL='3I5 /1H0,' COEFFICIENTS OF POL.ARE')
      END
```

* SOURCE STATEMENT *

```
C
C     ALGORITHM 30
C      NUMERICAL SOLUTION OF THE POLYNOMIAL EQUATION
      SUBROUTINE ROOTP(A,L, EPS,RP,CP,ILL)
      REAL K
      DIMENSION A(L),RP(L),CP(L),H(30),D(30),B(30)
      N=L
      ILL=0
      DO 1 J=1,N
    1 H(J)=A(J)
      T=1.0
      K=1./EPS
      CONVN=K
C     ZROTST
    2 IF(H( N ).NE.0)  GO TO  3
      RP(N-1)=0.
      CP(N-1)=0.
      N=N-1
      GO TO 2
C     INIT
    3 IF(N.EQ.1) RETURN
      PS=0.
      QS=0.
      PT=0.
      QT=0.
      S=0.
      REV=1.
      K=1./EPS
      IF(N.NE.2) GO TO 4
      R=-H(2)/H(1)
      GO TO 5
    4 DO 6  J=1,N
      IF(H(J).EQ.0)  GO TO 6
      S=S+ALOG10(ABS(H(J)) )
    6 CONTINUE
      S=S/ FLOAT(N)
      S=10.**S
      DO 7 J=1,N
    7 H(J)=H(J)/S
      IF(ABS(H(2)/H(1)).GE.ABS(H(N-1)/H(N))) GO TO 8
C     REVERSE
   22 T=-T
      M=(N+1)/2
      DO 9 J=1,M
      S=H(J)
      H(J)=H(N+1-J)
    9 H(N+1-J)=S
    8 CONTINUE
      IF(QS.EQ.0.)GO TO 10
      P=PS
      Q=QS
      GO TO 11
   10 IF(H(N-2).NE.0.)GO TO 12
      Q=1.0
      P=-2.0
```


* SOURCE STATEMENT (ROOTP)*

```
      GO TO 13
   12 Q=H(N)/H(N-2)
      P=(H(N-1)-Q*H(N-3))/H(N-2)
      IF(P.EQ.0.) P=1.
   13 IF(N.EQ.3) GO TO 36
      R=0.
C     ITERATE
   11 CONTINUE
      DO 15 I=1,500
C     BAIRSTOW
      BJ1=0.
      BJ2=0.
      CJ1=0.
      CJ2=0.
      DJ1=0.
      EJ1=0.
      DO 16 J=1,N
      B(J)=H(J)-P*BJ1-Q*BJ2
      BJ=B(J)
      CJ=BJ-P*CJ1-Q*CJ2
      IF(J.EQ.N) GO TO 16
      CJ3=CJ2
      BJ2=BJ1
      CJ2=CJ1
      BJ1=BJ
      CJ1=CJ
   16 CONTINUE
      IF(H(N-1).EQ.0.)GO TO 17
      IF(B(N-1).EQ.0.)  GO TO 17
      IF(ABS(H(N-1)/B(N-1) ).LT.K)GO TO 18
      B(N)=H(N)-Q*B(N-2)
C     BNTEST
   17 IF(B(N).EQ.0.) GO TO 14
      IF(ABS(H(N)/B(N)).GT.K) GO TO 14
C     NEWTON
   18 DO 19 J=1,N
      D(J)=H(J)+R*DJ1
      DJ=D(J)
      EJ=DJ+R*EJ1
      IF (J.EQ.N) GO TO 19
      DJ1=DJ
      EJ1=EJ
   19 CONTINUE
      IF(D(N).EQ.0.) GO TO  5
      IF(ABS(H(N)/D(N)).GE.K) GO TO 5
      CJ1=-P*CJ2-Q*  CJ3
      S=CJ2*CJ2-CJ1*CJ3
      IF(S.NE.0.)GO TO 20
   35 P=P-2.
      Q=Q*(Q+1.)
      GO TO 21
   20 IF((P.EQ.0.).AND.(Q.EQ.0.)) GO TO 35
      P=P+(B(N-1)*CJ2-B(N)*CJ3)/S
      Q=Q+(-B(N-1)*CJ1+B(N)*CJ2) /S
   21 IF(EJ1.NE.0.) GO TO 29
```

付録２　サブルーチンROOTP

```
FACOM 230-60     FORTRAN D  -710110- 0009-03                    COMPILATION     71.05.27     PAGE   4

* SOURCE STATEMENT (ROOTP    )*

             R=R-1.
             GO TO 15
          29 R=R-D(M)/EJ1
          15 CONTINUE
             IF(REV.LT.0.) GO TO 37
             PS=PT
             QS=QT
             PT=P
             QT=Q
             REV=-REV
             GO TO 22
      C      LINEAR
           5 IF(T.LT.0.) R=1./R
             RP(N-1)=R
             CP(N-1)=0.
             CONVN=K
             N=N-1
             DO 23 J=1,N
          23 H(J)=B(J)
             IF(N.EQ.1)        RETURN
             GO TO 41
      C      QADRTIC
          14 IF(T.GE.0.) GO TO 24
             P=P/Q
             Q=1./Q
          24 CONTINUE
             WRITE(6,30) P,Q
             IF((Q-(P/2.)*(P/2.)).LE.0.) GO TO 25
             RP(N-1)=-P/2.
             RP(N-2)=-P/2.
             S=SQRT(Q-(P/2.)*(P/2.))
             CP(N-1)=S
             CP(N-2)=- S
             GO TO 33
          25 S=SQRT((P/2.)*(P/2.)-Q)
             IF(P.LT.0.) GO TO 26
             RP(N-1)=-P/2.-S
             GO TO 27
          26 RP(N-1)=-P/2.+S
          27 RP(N-2)=Q/RP(N-1)
             CP(N-1)=0.
             CP(N-2)=0.
          33 CONVN=K
             N=N-2
             DO 28 J=1,N
          28 H(J)=B(J)
             GO TO 3
          30 FORMAT(1H0,10X,20E15.5)
          31 FORMAT(//2X,I3)
          32 FORMAT (3X,3I10)
          36 P=H(N-1)/H(N-2)
             GO TO 14
          37 ILL=2
             RETURN
             END
```

付録３　ROOTPによる計算結果

```
M=    7
A
   0.12073E 15    0.49204E 11    0.28646E 11    0.38998E 07    0.53398E 06    0.19527E 02    0.10000E 01

DEGREE TWO POLYNOMIAL

            0.28879E-03    0.21703E-03

            0.90969E-04    0.18089E-04

            0.27779E-04    0.21097E-05

                                          TIME ELAPSED=        25MS

RP

     -0.13890E-04   -0.13890E-04   -0.45484E-04   -0.45484E-04   -0.14439E-03   -0.14439E-03

CP

     -0.14524E-02    0.14524E-02   -0.42529E-02    0.42529E-02   -0.14731E-01    0.14731E-01

ILL= 0


M=    9
A
   0.10000E 01    0.10000E 01    0.10000E 01    0.10000E 01    0.10000E 01    0.10000E 01    0.10000E 01    0.10000E 01

DEGREE TWO POLYNOMIAL

            0.14142E 01    0.10000E 01

           -0.38590E-05    0.10000E 01

           -0.14142E 01    0.10000E 01

                                          TIME ELAPSED=        34MS

RP

     -0.70711E 00    0.70711E 00    0.19295E-05    0.19295E-05   -0.70711E 00   -0.70711E 00

     -0.10000E 01

CP

     -0.70711E 00    0.70711E 00   -0.10000E 01    0.10000E 01   -0.70711E 00    0.70711E 00

      0.0
```