I.F.I.P. 論文紹介*

A-35. QD-algorithm に改良を加えて Graeffe 型の収束性をもたせること

Rutishauser, H.: On a Modification of the QD-algorithm with Graeffe-type Convergence 〔Ⅲ-1 pp. 29～30〕

QD-Algorithm によって連分数

$$f(z)=\frac{1|}{|z}-\frac{q_1|}{|1}-\frac{e_1|}{|z}-\frac{q_2|}{|1}-\cdots-\frac{q_{n-1}|}{|1}-\frac{e_{n-1}|}{|z}-\frac{q_n|}{|1}$$

の零点を計算できることは知られている．もし $f(z)$ の零点の平方を零点とする連分数

$$f(z)=\frac{1|}{|z}-\frac{\hat{q}_1|}{|1}-\frac{\hat{e}_1|}{|z}-\frac{\hat{q}_2|}{|1}-\cdots-\frac{\hat{q}_{n-1}|}{|1}-\frac{\hat{e}_{n-1}|}{|z}-\frac{\hat{q}_n|}{|1}$$

を $f(z)$ から直接計算できれば，Greaffe の方法のように収束の速さが改良されることは明らかである．さて

$$g(z)=\frac{f(z)+f(-z)}{2q_1}$$

と定義すると，$g(z)$ は偶函数であるから $g(\sqrt{z})$ を $f(z)$ にとることができる．ところが7項行列

$$Q=\begin{bmatrix}
0 & q_1+e_1 & 0 & e_1 & & & & 0\\
q_1+e_1 & 0 & e_1 & 0 & 0 & & & \\
0 & q_2 & 0 & q_2+e_2 & 0 & e_2 & & \\
q_2 & 0 & q_2+e_2 & 0 & e_2 & \ddots & \ddots & \\
 & 0 & 0 & q_3 & \ddots & \ddots & & e_{n-1}\\
 & & q_3 & & \ddots & & e_{n-1} & 0\\
 & & & \ddots & \ddots & q_n & 0 & q_n\\
0 & & & & q_n & 0 & q_n & 0
\end{bmatrix}$$

を定義すると，$(zI-Q)^{-1}$ の (1,1) 要素は $zg(z)$ であることを示すことができるから，第1行と第1列とが0のアフィン変換によって Q を3項行列

$$Q'=\begin{bmatrix}
0 & a_1 & & & & & 0\\
a_1' & 0 & b_1 & & & & \\
 & b_1' & 0 & a_2 & & & \\
 & & a_2' & 0 & b_2 & & \\
 & & & \ddots & \ddots & \ddots & \\
 & & & & \ddots & \ddots & a_n\\
0 & & & & & a_n' & 0
\end{bmatrix}$$

に変換できれば，$(zI-Q')^{-1}$ の (1, 1) 要素も $zg(z)$ に等しい．したがって

$$g(z)=\frac{1|}{|z^2}-\frac{a_1a_1'|}{|1}-\frac{b_1b_1'|}{|z^2}-\frac{a_2a_2'|}{|1}-\cdots-\frac{b_{n-1}b'_{n-1}|}{|z^2}-\frac{a_na'_n|}{|1}$$

であり，$\hat{f}(z)$ とくらべると

$\hat{q}_k=a_ka_k'$ $(k=1, 2, \cdots\cdots, n)$

$\hat{e}_k=b_kb_k'$ $(k=1, 2, \cdots\cdots, n-1)$

であることがわかる．2次元のアフィン変換をうまく重ねて Q から Q' への変換を行なうことができる．

連分数 $f(z)$ の零点を計算するのに，まず連分数

$$f_0(z)=\frac{zf(z)-1}{q_1}=\frac{1|}{|z}-\frac{q_1^{(0)}|}{|1}-\frac{e_1^{(0)}|}{|z}-\cdots-\frac{e_{n-1}^{(0)}|}{|z}-\frac{q_n^{(0)}|}{|1}$$

を QD-algorithm を一段行なって計算し，それから

$f_1(z)=\hat{f}_0(z)$

$f_2(z)=\hat{f}_1(z)$

$f_3(z)=\hat{f}_2(z)$ 等々

を計算する．$f_p(z)$ の係数を $q_k^{(p)}$ と $e_k^{(p)}$ で表わすと，次のような性質がある．

$$\lim_{p\to\infty}(|q_k^{(p)}|)^{\frac{1}{2p}}=|\lambda_k| \quad (k=1, 2, \cdots, n)$$

$$\lim\frac{e_k^{(p)}}{q_{k+1}^{(p)}}=\frac{c_{k+1}}{c_k} \quad (k=1, 2, \cdots n-1),$$

ここで

$$f(z)=\sum_1^n\frac{c_k}{z-\lambda_k} \qquad (\Sigma c_k=1)$$

である．大きな p に対して $q_k^{(p)}$ や $e_k^{(p)}$ は浮動小点数の範囲を出てしまうから，

$g_k^{(p)}=e_k^{(p)}/q_{k+1}^{(p)}$

$f_k^{(p)}=q_{k+1}^{(p)}/q_k^{(p)}$

$h_k^{(p)}=(|q_k^{(p)}|)^{\frac{1}{2p}}$

を導入して，$f_k^{(p)}$, $g_k^{(p)}$, $h_k^{(p)}$ から直接 $f_k^{(p+1)}$, $g_k^{(p+1)}$, $h_{k+1}^{(p)}$ を計算する．この方法は Gauss の数値積分公式

$$\int_0^A p(x)f(x)dx\approx\sum_1^m c_kf(x_k)$$

の座標と重みの計算に適用できる．もし

$$\int_0^A\frac{p(x)}{z-x}dx=\frac{1|}{|z}-\frac{q_1|}{|1}-\frac{e_1|}{|z}-\frac{q_2|}{|1}-\frac{e_2|}{|z}-\cdots$$

とすると，

* 前号に引続き第2回国際情報処理学会の提出論文（本誌 Vol. 3 No. 4 参照）を Preprint of the Proceedings of the IFIP CONGRESS 62 (Aug. 1962) から紹介します．

$$x_k=\lim_{p\to\infty}h_k^{(p)}$$

となり，$g_k^{(p)}$ から c_k も得られる．（清水留三郎）

A-36. 連立代数方程式の解法

Blundell, P.H.: A Method for Solving Simultaneous Polynomial Equations 〔Ⅲ-4, pp. 39～42〕

2変数の代数方程式は，

$$\Sigma a_{ij}x^iy^j=0$$
$$\Sigma b_{ij}x^iy^j=0 \qquad (1)$$
$$i,\ j=0,\ 1,\ \cdots,\ n,\ i+j\leqq n$$

という形をとり，n^2 個の根をもつ．一方差分方程式

$$\Sigma a_{ij}u(p+i,\ q+j)=0$$
$$\Sigma b_{ij}u(p+i,\ q+j)=0 \qquad (2)$$

は，A_k を任意の定数として，一般解

$$u(p,\ q)=\Sigma A_kx_k{}^py_k{}^q,\ k=1,\ 2,\ \cdots,\ n^2 \qquad (3)$$

をもつ．$q=$constant の線上では，解は

$$u(p)=\Sigma A_k{}'x_k{}^p$$

という形をとり，n^2 階の差分方程式

$$\Sigma c_iu(p+i)=0,\ i=0,\ 1,\ \cdots,\ n^2$$

を満すから，特性方程式

$$\Sigma c_ix^i=0 \qquad (4)$$

の根は x_k である．そこで差分方程式（2）の任意の特解から代数方程式（4）がわかれば，もとの代数方程式（1）の根の座標の一つが計算できる．

$n-1\leqq r+s\leqq 2n-2$ の範囲の $u(r,\ s)$ の値がわかっている場合 $r+s=2n-1$ の上の $2n$ 個の点での値は $p+q=n-1$ なる p と q との n 個の組み合わせについて式（2）を使って求められ，さらに $r-s$ 平面のあらゆる点での値がわかる．そこで解法の出発値さえわかればよい．三角形 $r+s\leqq 2n-2$ は $n(2n-1)$ 個の点を含む．一般解（3）の n^2 個の任意の定数 A_k に対応して，このうちの n^2 個の点に任意の値を与えれば，値の未知な点の数は $n(n-1)$ になる．$p+q\leqq n-2$ なる $p,\ q$ のすべての組み合わせについて式（2）を適用して，$n(n-1)$ 個の方程式が得られ特解の出発値が与えられる．

根の座標の他方の値を知るために，差分方程式（2）の点 $(r_0,\ s_0)$ と $(r_0,\ s_0+1)$ とから出発する線上での解を考える．（3）から

$$\nu_p=u(r_0+p,\ s_0)$$
$$=\Sigma A_kx_k{}^{r_0+p}y_k{}^{s_0}$$
$$=\Sigma B_kx_k{}^p, \qquad (5)$$
$$\nu_p{}'=u(r_0+p,\ s_0+1)$$
$$=\Sigma B_kx_k{}^py_k$$
$$=\Sigma B_k{}'x_k{}^p \qquad (6)$$

であるから

$$y_k=B_k{}'/B_k$$

となる．もし n^2 個の ν_p と $\nu_p{}'$ の値を知れば，（5）と（6）は B_k と $B_k{}'$ に関する n^2 個の方程式を与え，それらは解析解

$$B_k=\sigma\sum_p{}^k s_{t-p}\ \nu_p{}'(t=n^2)$$

をもつ．ここで σ は x_k の対称式であり，${}^is_{t-p}$ は，x_i を除く，x_k の $(t-p)$ 次の対称な積である．同様に，

$$B'_k=\sigma\sum_p{}^k s_{t-p}\ \nu'_p$$

であるから，

$$y_k=\frac{\Sigma^k s_{t-p}\ \nu_p{}'}{\Sigma^k s_{t-p}\ \nu_p}$$

である．さて ${}^ks_{t-p}$ は（A）の係数 c_i から漸化式

$${}^ks_{t-p}=(-1)^kc_{t-p}-x_k{}^ks_{t-p-1}$$

によって得られる．

差分方程式の解の級数展開の値は絶対値最大の根に支配されてしまって，より小さい根が貢献せず，c_i を求める連立一次方程式が高精度演算でも正則でなくなる危険がある．その場合の対策としては，絶対値最大の根 x_m のできるかぎり正しい近似値 $\bar{x}_m$ を求め，

$$u'(p,\ q)=\bar{x}_mu(p,\ q)-u(p+1,\ q)$$

を計算して x_m の貢献を消去する．しかし $\bar{x}_m$ の誤差と丸め誤差の成長によって x_m 成分がまた入って来て完全に消去することは難しい．（清水留三郎）

B-37. Dynamic Storage Allocation のための Program Organization と Record Keeping

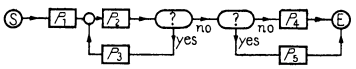
Program Organization and Record Keeping for Dynamic Storage Allocation 〔XV-1 pp. 237～240〕

これは RCA ACSI-MATIC のという programming system における core allocation についての報告である．計算機は並列処理のできるものであり，また数 level の二次記憶をもつもので，応用面はかなり複雑な data processing である．

program は segment にわかれ active なときには computation storage に segment transfer instruction によって transfer される．dynamic storage allocation では，program の一部をよみこみ，それのための allocation information を作り，また，つぎの一部をよんで，今度はこれと前の allocation

information とから第二の部分の allocation information をつくるというように，順々に直前までの allocation information を使用して，新しい allocation information を作ってゆくものである．そしてこれらの allocation と実際の computation が交互におこる．dynamnic allocation では，allocation (dump と restore) の frequency がどうかによって，process ががらりと変ってしまう．これらの allocation のほか，space をすくなくしたための speed down の penalty ももとめなければならない．

図と表とはある subprogram をしめしている．表の section I の行は use vector とよばれ，たとえば $\sum_8$ の logical entity で P_2, P_4 が 1 なのは，そのときに active であることを意味している．use vector array を use matrix とよぶ．これが subprogram の allocator に対する information の一部となる．たとえば Σ_2 と Σ_3 の core space は共用してよいが，Σ_2 と Σ_5 は共用ではこまるということがわかる．また，どの phase で segment が active になったり inactive になったりするかも use matrix からよみとれる．section II も allocator に対する記述である．つまりどの entity がどの storage medium にどのくらいの segment を必要とするかの情報である．たとえば Σ_8 が active になると core に 2 segment, disc に 5 segment 必要である．数値に付随した * は length が variable であることをあらわす．



Flow chart of a sample subprogram, $\theta_{5.1}$

以上のほかに use matrix には penalty を計算する情報が必要である． （和田英一）

Structure of Subprogram, $\theta_{5.1}$

		I. Active in phase:					II. Number of segments with active Position in:		
Logical entity	Σ	P_1	P_2	P_3	P_4	P_5	core	disc	tape
Input	Σ_1	1	1	1	0	0	*	*	
Input	Σ_2	0	1	1	0	0	*		
Output	Σ_3	0	0	0	1	1			*
Instructions for P_1	Σ_4	1	0	0	0	0	4		
Intructions and working storage co-mmon to P_1 and P_2	Σ_5	1	1	0	0	0	4 *		
Instructions for P_3	Σ_6	0	0	1	0	0	2		
A tablerefe renced by P_4 and P_5	Σ_7	0	0	0	1	1	3		1
Suboperations common to P_2 and P_4	Σ_8	0	1	0	1	0	2	5	
Instructions for P_4	Σ_9	0	0	0	1	0	4		
Instructions for P_5	Σ_{10}	0	0	0	0	1	1		
Constants and working area used throughout	Σ_{11}	1	1	1	1	1	1 *		