数理解析研究所講究録
551巻 1985年 1-14

単因子の計算法

東大・理　森継修一　(Shuichi Moritsugu)
都立大・理　古川昭夫　(Akio Furukawa)
日大・理工　小林英恒　(Hidetsune Kobayashi)
理研　佐々木建昭　(Tateaki Sasaki)

概 要

体上1変数多項式を成分とする行列の単因子の計算法について述べる。最初に古典的アルゴリズムを示し。次にその改良について考察する。

例として、数式処理言語REDUCE2でインプリメントしたプログラムの実行結果を示す。

1. 序

行列の単因子とは、次に示す定理によって定義づけられるものである。

<u>定理1（単因子論の基本定理）</u>

R=K[x]を体K上の1変数多項式環とし、Aを、Rの元を成分とする $m\times n$ 行列とする。Aは基本変形(注)の合成によって、次のような「標準形」に変形できる。

$$A \longrightarrow \begin{pmatrix} g_1 & & & & & \\ & g_2 & & & & \\ & & \ddots & & & \\ & & & g_l & & \\ & & & & 0 & \\ & & & & & \ddots \\ & & & & & & 0 \end{pmatrix} \qquad \begin{array}{l} g_i \in R \\ g_1 \mid g_2 \mid g_3 \mid \cdots \mid g_l \end{array} \qquad 0 \leq l \leq \min(m,n)$$

ここで、$g_1, g_2, \cdots g_l$ は定数倍を除いて一意的であり、l もAによって一意的に決まる。

定義

定理1における、$g_1, g_2, \cdots g_l$を行列Aの単因子という。

(注) 以下の6種類の操作を行列の基本変形という。

(i) 0でない定数をある行に乗ずる。

(ii) 第j行と第k行を入れ換える。

(iii) 第j行に、第k行を$f(\in R)$倍したものを加える。

(iv)〜(vi) 上の(i)〜(iii)を、列について行なう。

例

$$A = \begin{pmatrix} x & x-1 & x-1 & x-2 \\ x & x^3+x & x-1 & x^3+x-1 \\ x+1 & x^3+x+2 & x+1 & x^3+2x+3 \\ x-1 & x-3 & x-3 & -6 \end{pmatrix}$$ の単因子を計算する。(文献[1]より)

$$A \xrightarrow[-\text{第}3\text{列}]{\text{第}1,2,4\text{列}} \begin{pmatrix} 1 & 0 & x-1 & -1 \\ 1 & x^3+1 & x-1 & x^3 \\ 0 & x^3+1 & x+1 & x^3+x+2 \\ 2 & 0 & x-3 & -x-3 \end{pmatrix} \xrightarrow[\text{第}4\text{行} - 2\times\text{第}1\text{行}]{\text{第}2\text{行} - \text{第}1\text{行}} \begin{pmatrix} 1 & 0 & x-1 & -1 \\ 0 & x^3+1 & 0 & x^3+1 \\ 0 & x^3+1 & x+1 & x^3+x+2 \\ 0 & 0 & -x-1 & -x-1 \end{pmatrix}$$

$$\xrightarrow[\text{第}4\text{列} + \text{第}1\text{列}]{\text{第}3\text{列} - (x-1)\times\text{第}1\text{列}} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & x^3+1 & 0 & x^3+1 \\ 0 & x^3+1 & x+1 & x^3+x+2 \\ 0 & 0 & -x-1 & -x-1 \end{pmatrix} \xrightarrow[-(\text{第}2\text{列}+\text{第}3\text{列})]{\text{第}4\text{列}} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & x^3+1 & 0 & 0 \\ 0 & x^3+1 & x+1 & 0 \\ 0 & 0 & -x-1 & 0 \end{pmatrix}$$

$$\xrightarrow[-(\text{第}2\text{行}-\text{第}4\text{行})]{\text{第}3\text{行}} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & x^3+1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -x-1 & 0 \end{pmatrix} \xrightarrow[\text{行,列の交換}]{\text{第}4\text{行}\times(-1)\text{の後}} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & x+1 & 0 & 0 \\ 0 & 0 & x^3+1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

したがって、Aの単因子は、1, $x+1$, x^3+1 である。

定理1の、一般に知られている証明法（例えば文献[2]）は、構成的でなく、具体的に単因子を求めるアルゴリズムを与えていない。そこで我々は、単因子の具体的な計算方法とその効率化について考察した。

なお、以下の章の議論では、大きさ $n \times n$ の正方行列のみを扱うことにする。

2. 古典的アルゴリズム

ここで示すアルゴリズムは文献[3]による。この方法は次の2つの事実に基く。

(I) 行ベクトル（$1 \times n$ 行列） $(a_1 \ a_2 \ \cdots \ a_n)$ の単因子は、$\mathrm{GCD}(a_1, a_2, \cdots a_n)$ である。

(II) $n \times n$ 行列Aに基本変形を施して、(1, 1)成分を除く第1行、第1列の成分が0になったとする。

$$A \longrightarrow \left(\begin{array}{c|ccc} g & 0 & \cdots & 0 \\ \hline 0 & & & \\ \vdots & & A' & \\ 0 & & & \end{array}\right)$$

このとき、(1, 1)成分 g が第1番目の単因子ならば、g は A' のすべての成分を割り切る。

この(I), (II)は、定理1の証明（文献[2]）の中間過程で得られ、その証明の根幹をなしている。

具体的なアルゴリズムは次のように表わせる。

<u>記法</u>

$A = (a_{ij})$ を $n \times n$ 正方行列（a_{ij} は体上1変数多項式）とする。基本変形を受けた成分を、（操作の回数に関係なく） $\widetilde{a_{ij}}$ と表わすことにする。

<u>アルゴリズム（古典的方法）</u>

行列Aに対して

(C1) 第1行について、$\widetilde{a_{11}} \neq 0$, $\widetilde{a_{12}} = \widetilde{a_{13}} = \cdots = \widetilde{a_{1n}} = 0$ なるよう変形する。

(C2) もし、$\tilde{a}_{21}=\tilde{a}_{31}=\cdots=\tilde{a}_{n1}=0$ ならば (C5) へ。

(C3) 第1列について、$\tilde{a}_{11}\neq 0$, $\tilde{a}_{21}=\tilde{a}_{31}=\cdots=\tilde{a}_{n1}=0$ なるよう変形する。

(C4) もし、$\tilde{a}_{12}=\tilde{a}_{13}=\cdots=\tilde{a}_{1n}=0$ でないならば (C1) へ。

(C5) 行列は $\left(\begin{array}{c|ccc} \tilde{a}_{11} & 0 & \cdots & 0 \\ \hline 0 & & & \\ \vdots & & A' & \\ 0 & & & \end{array}\right)$ という形になっている。

もし、$\tilde{a}_{11}$ が A' のすべての成分を割り切るならば、$\tilde{a}_{11}$ は単因子であるから、A' に対してこのアルゴリズムを再帰的に適用する。$\tilde{a}_{11}$ で割り切れない成分があるときは、それを含む行を第1行に加えて (C1) へ。

上述のアルゴリズムの (C1) ステップを「行の消去」といい、(C3) ステップを、列の消去」ということにする。次の第3章において、消去の具体的計算方法をべる。

. 消去計算の方法

行の消去とは、行の成分全体の GCD を求めることであるから、基本的にはユークリッドの互除法の繰り返しである。これを基本変形の合成で実行するには、次ような計算を行なえばよい。すなわち、行列を

$$\begin{pmatrix} \cdots & a_{1j} & \cdots & a_{1k} & \cdots \\ \cdots & a_{2j} & \cdots & a_{2k} & \cdots \\ & \vdots & & \vdots & \\ \cdots & a_{nj} & \cdots & a_{nk} & \cdots \end{pmatrix}$$

として、a_{1j} を a_{1k} で割ることを考えると、商 q と剰余 r は

$$a_{1j} = q\,a_{1k} + r$$

と表わせるから、「第 j 列から第 k 列の q 倍を引く」操作を行なえば、

$$\begin{pmatrix} \cdots & a_{1j}-qa_{1k} & \cdots & a_{1k} & \cdots \\ \cdots & a_{2j}-qa_{2k} & \cdots & a_{2k} & \cdots \\ & \vdots & & \vdots & \\ \cdots & a_{nj}-qa_{nk} & \cdots & a_{nk} & \cdots \end{pmatrix} = \begin{pmatrix} \cdots & r & \cdots & a_{1k} & \cdots \\ \cdots & \tilde{a}_{2j} & \cdots & a_{2k} & \cdots \\ & \vdots & & \vdots & \\ \cdots & \hat{a}_{nj} & \cdots & a_{nk} & \cdots \end{pmatrix}$$

となり、割り算が1回行なわれたことになる。

しかし、GCD計算には互除法の中間結果は不必要であり、割り算を行なうごとに第2行以下の成分も操作するのは無駄である。そこで、互除法の過程を再検討してみると、最終結果を非常に簡単な形で表わせることがわかった。

まず次の定理を示しておく。

定理3.1（拡張されたユークリッドの定理）

多項式F_1とF_2のGCDをGとするとき、次の条件を満たす多項式AとBが1組だけ存在する。

$$\begin{cases} AF_1 + BF_2 = G \\ \deg(A) < \deg(F_2) - \deg(G) \quad , \quad \deg(B) < \deg(F_1) - \deg(G) \end{cases}$$

証明は文献[4] を参照されたい。

この定理を用いて行の消去について次の定理が得られる。この定理が本論文の主要な結論の1つである。

定理3.2（行の消去）

行列$M = \begin{pmatrix} P_1 & P_2 \\ a & b \end{pmatrix}$ に対して、P_1, P_2 に互除法を施して消去した結果は、

$$\begin{pmatrix} g & 0 \\ Aa+Bb & \frac{1}{g}\begin{vmatrix} P_1 & P_2 \\ a & b \end{vmatrix} \end{pmatrix}$$

である。ただし、$\begin{cases} g = GCD(P_1, P_2) = AP_1 + BP_2 \\ \deg(A) < \deg(P_2) - \deg(g) \\ \deg(B) < \deg(P_1) - \deg(g) \end{cases}$

（すなわちA,Bは「拡張されたユークリッドの定理」におけるA,Bである。）

(注) 3×3以上の行列の計算の場合は、a,bを列ベクトルとみなして各成分について計算する。

証明]

長くなるので、4段階に分けて示す。

(i) 拡張されたユークリッドの定理の導出過程より、A, B, g は次の反復公式で計算できる。

$$\begin{cases} A_i = A_{i-2} - Q_i A_{i-1} \quad (i \geq 3) & A_1 = 1 \quad A_2 = 0 \\ B_i = B_{i-2} - Q_i B_{i-1} \quad (i \geq 3) & B_1 = 0 \quad B_2 = 1 \end{cases}$$

ただし、Q_i は $P_i = P_{i-2} - Q_i P_{i-1}$ で定義され、A_i, B_i は

$P_i = A_i P_1 + B_i P_2$ をみたす。

[(i)の証明] 文献[4] を参照されたい。

(ii) $M = \begin{pmatrix} P_1 & P_2 \\ a & b \end{pmatrix}$ の P_1, P_2 に互除法を施すときの中間結果は、割り算k回後の行列を $M^{(k)}$ とかくことにすれば、

① $2k-2$ 回の割り算の後では、$M^{(2k-2)} = \begin{pmatrix} P_{2k-1} & P_{2k} \\ A_{2k-1}a + B_{2k-1}b & A_{2k}a + B_{2k}b \end{pmatrix}$

② $2k-1$ 回の割り算の後では、$M^{(2k-1)} = \begin{pmatrix} P_{2k+1} & P_{2k} \\ A_{2k+1}a + B_{2k+1}b & A_{2k}a + B_{2k}b \end{pmatrix}$

と表わせる。(A_i, B_i は(i)で定義されたもの。)

ただし、1回目の割り算とは、P_1 を P_2 で割って $P_3 = P_1 - Q_3 P_2$ を計算することをいい、k は多項式剰余列 P_i が終了しない範囲で考える。

[(ii)の証明]

k に関する数学的帰納法

・$k=1$

$$\begin{pmatrix} P_1 & P_2 \\ a & b \end{pmatrix} \longrightarrow \begin{pmatrix} P_1 - Q_3 P_2 & P_2 \\ a - Q_3 b & b \end{pmatrix}$$

$$A_3 = A_1 - Q_3 A_2 = 1 \ , \quad B_3 = B_1 - Q_3 B_2 = -Q_3$$

であるから $M^{(1)} = \begin{pmatrix} P_3 & P_2 \\ A_3 a + B_3 b & b \end{pmatrix}$ が成り立つ。

・$k=2$

$$\begin{pmatrix} P_3 & P_2 \\ A_3a+B_3b & b \end{pmatrix} \longrightarrow \begin{pmatrix} P_3 & P_2-Q_4P_3 \\ A_3a+B_3b & b-Q_4(A_3a+B_3b) \end{pmatrix}$$

$$A_4=A_2-Q_4A_3=-Q_4A_3,\quad B_4=B_2-Q_4B_3=1-Q_4B_3$$

より、$b-Q_4(A_3a+B_3b)=-Q_4A_3a+(1-Q_4B_3)b=A_4a+B_4b$

したがって　$M^{(2)}=\begin{pmatrix} P_3 & P_4 \\ A_3a+B_3b & A_4a+B_4b \end{pmatrix}$ が成り立つ。

・$M^{(2k-2)}$が正しいと仮定する。

$(2k-1)$回目の割り算は、$P_{2k+1}=P_{2k-1}-Q_{2k+1}P_{2k}$　で決まるQ_{2k+1}を用いて、(第1列)$-Q_{2k+1}\times$(第2列) を計算することである。このとき、(2,1) 成分は、

$$A_{2k-1}a+B_{2k-1}b-Q_{2k+1}(A_{2k}a+B_{2k}b)$$
$$=(A_{2k-1}-Q_{2k+1}A_{2k})a+(B_{2k-1}-Q_{2k+1}B_{2k})b$$
$$=A_{2k+1}a+B_{2k+1}b$$

となり、$M^{(2k-1)}$が成り立つ。

・$M^{(2k-1)}$が正しいと仮定する。

$2k$回目の割り算は、$P_{2k+2}=P_{2k}-Q_{2k+2}P_{2k+1}$　で決まるQ_{2k+2}を用いて、(第2列)$-Q_{2k+2}\times$(第1列) を計算することである。このとき、(2,2)成分は、

$$A_{2k}a+B_{2k}b-Q_{2k+2}(A_{2k+1}a+B_{2k+1}b)$$
$$=(A_{2k}-Q_{2k+2}A_{2k+1})a+(B_{2k}-Q_{2k+2}B_{2k+1})b$$
$$=A_{2k+2}a+B_{2k+2}b$$

となり、$M^{(2k)}$ が成り立つ。

以上の議論により、(ii)の表記が成り立つことが示された。

(ii)の証明終)

) 多項式剰余列 P_i が終了するとき、(ii)の表記は次のように書き直せる。

①(2k-2) 回の割り算で終了するとき、

$P_{2k}=0$, $g=GCD(P_1, P_2)=P_{2k-1}$ であり、

$$M^{(2k-2)} = \begin{pmatrix} g & 0 \\ Aa+Bb & \frac{1}{g}\begin{vmatrix} P_1 & P_2 \\ a & b \end{vmatrix} \end{pmatrix}$$

②(2k-1) 回の割り算で終了するとき、

$P_{2k+1}=0$, $g=GCD(P_1, P_2)=P_{2k}$ であり、

$$M^{(2k-1)} = \begin{pmatrix} 0 & g \\ -\frac{1}{g}\begin{vmatrix} P_1 & P_2 \\ a & b \end{vmatrix} & Aa+Bb \end{pmatrix}$$

[(iii)の証明]

①の (2,1) 成分については、$A_{2k-1}=A$, $B_{2k-1}=B$ より明らか。②の (2,2) 成分についても、$A_{2k}=A$, $B_{2k}=B$ より明らかである。

残る成分、①の $A_{2k}a+B_{2k}b$ と、②の $A_{2k+1}a+B_{2k+1}b$ を、A, B, P_1, P_2 で表わすことを考える。

一般に、多項式剰余列 P_i について、$P_{n+1}=P_{n-1}-Q_{n+1}P_n=0$ とすると、$P_n=GCD(P_1, P_2)$ である。また一方

$$\begin{cases} P_{n-1} = A_{n-1}P_1 + B_{n-1}P_2 \\ P_n = A_nP_1 + B_nP_2 \end{cases}$$

であるから、これらの式を用いて、A_{n+1}, B_{n+1} を A, B, P_1, P_2 で表わしてみる。

$$\begin{aligned} A_{n+1} &= A_{n-1} - Q_{n+1}A_n \\ &= A_{n-1} - \frac{P_{n-1}}{P_n}A_n \\ &= \frac{1}{P_n}(P_nA_{n-1} - P_{n-1}A_n) \\ &= \frac{1}{P_n}\{(A_nP_1+B_nP_2)A_{n-1} - (A_{n-1}P_1+B_{n-1}P_2)A_n\} \\ &= \frac{P_2}{P_n}(A_{n-1}B_n - A_nB_{n-1}) = \frac{P_2}{P_n}\begin{vmatrix} A_{n-1} & A_n \\ B_{n-1} & B_n \end{vmatrix} \end{aligned}$$

B_{n+1} についても同様で

$$
\begin{aligned}
B_{n+1} &= B_{n-1} - Q_{n+1} B_n \\
&= B_{n-1} - \frac{P_{n-1}}{P_n} B_n \\
&= \frac{1}{P_n}\left(P_n B_{n-1} - P_{n-1} B_n \right) \\
&= \frac{1}{P_n}\left\{ (A_n P_1 + B_n P_2) B_{n-1} - (A_{n-1} P_1 + B_{n-1} P_2) B_n \right\} \\
&= \frac{P_1}{P_n}\left(A_n B_{n-1} - A_{n-1} B_n \right) = -\frac{P_1}{P_n}\begin{vmatrix} A_{n-1} & A_n \\ B_{n-1} & B_n \end{vmatrix}
\end{aligned}
$$

ここで、行列式の形をしている部分は、

$$
\begin{vmatrix} A_{n-1} & A_n \\ B_{n-1} & B_n \end{vmatrix} = \begin{vmatrix} A_{n-1} & A_{n-2} - Q_n A_{n-1} \\ B_{n-1} & B_{n-2} - Q_n A_{n-1} \end{vmatrix} = \begin{vmatrix} A_{n-1} & A_{n-2} \\ B_{n-1} & B_{n-2} \end{vmatrix} = \cdots
$$

$$
\cdots = \begin{cases} \begin{vmatrix} A_1 & A_2 \\ B_1 & B_2 \end{vmatrix} = \begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix} = 1 & n:\text{偶数のとき} \\ \begin{vmatrix} A_2 & A_1 \\ B_2 & B_1 \end{vmatrix} = \begin{vmatrix} 0 & 1 \\ 1 & 0 \end{vmatrix} = -1 & n:\text{奇数のとき} \end{cases}
$$

であるから、結局、 $A_{n+1} = \dfrac{(-1)^n P_2}{P_n}$ ， $B_{n+1} = \dfrac{(-1)^{n+1} P_1}{P_n}$ となる。

したがって、

①の場合

$$
A_{2k} a + B_{2k} b = -\frac{P_2}{P_{2k-1}} a + \frac{P_1}{P_{2k-1}} b = \frac{1}{g}\begin{vmatrix} P_1 & P_2 \\ a & b \end{vmatrix}
$$

②の場合

$$
A_{2k+1} a + B_{2k+1} b = \frac{P_2}{P_{2k}} a - \frac{P_1}{P_{2k}} b = -\frac{1}{g}\begin{vmatrix} P_1 & P_2 \\ a & b \end{vmatrix}
$$

（(iii)の証明終）

(iv) (iii)の結果から、定理の結論を得る。

[(iv)の証明]

(iii)の①は定理の結論の形を与えている。(iii)の②については、

$$M^{(2k-1)} = \begin{pmatrix} 0 & g \\ -\frac{1}{g}\begin{vmatrix} P_1 & P_2 \\ a & b \end{vmatrix} & Aa+Bb \end{pmatrix}$$ に対して、

・オ1列に -1 をかける。

・オ1列とオ2列を交換する。

という基本操作を施せば、定理の結論の形に達する。

（ⅳの証明終）

（定理の証明終）

列の消去に関しては、定理3.2 の結果を転置させた行列が消去の結果となる。
の定理3.2 を適用することにより、消去計算に伴う多項式の四則演算を大幅に
らすことができる。

. アルゴリズムの効率化

オ2章で示した古典的アルゴリズムでは、(C4), (C5) ステップから (C1) ステップ
への逆戻りの繰り返しが起こる場合が考えられる。このような逆戻りが起きない
うにして、オ1行の消去だけで単因子が得られるように、行列に前処理を施す
ことを考える。この方法の根拠となるのは、次に示す定理1の系である。

<u>定理1の系</u>

行列Aの最初の単因子は、Aの全成分のGCDである。

[証明]

定理の証明（[2]）の主張は、「基本変形を施して得られる、0でない次数最低の成分が単因子である。」ということである。基本変形で次数を下げるということはGCDを計算することであり、次数最低のものを得よとは全成分のGCDを求めよということに他ならない。

（証明終）

そこで、まず次の計算を行なう。

前処理（ステップ1）― 全成分のGCDの計算

```
begin  g0 := a11 ;
       for i:=1 to n do
              for j:=1 to n do
                     begin  k:= n(i-1)+j ;
                            gk := GCD(gk-1, aij)
                     end
end ;
```

最初の単因子 $g = GCD(a_{11}, a_{12}, \cdots a_{nn})$ を求めるには、上で計算した g_k ($k = 1, 2, \cdots n^2$) を調べて、$deg(g_k) < deg(g_{k-1})$ となるような a_{ij} を取り出して、GCD計算を行なえばよい。なぜなら、$deg(g_{k'}) = deg(g_{k'-1})$ のとき $k' = n(i'-1)+j'$ として、

$$GCD(a_{11}, a_{12}, \cdots a_{i'j'-1}) = GCD(a_{11}, a_{12}, \cdots a_{i'j'-1}, a_{i'j'})$$

であるから、$a_{i'j'}$ は全体のGCDを求めるのに不要だからである。

したがって、$deg(g_k) < deg(g_{k-1})$ となるような a_{ij} を前もって才1行に集めておけば、才1行の消去だけで単因子が見つかることが期待される。

前処理（ステップ2）― 必要な成分の才1行へのたしこみ

```
for k:=1 to n² do
       if deg(gk+1) < deg(gk)        (* ただし k=n(i-1)+j *)
            then <才i行を才1行に足す>
```

足しこむことによってGCDが変わらないという保証はないが、大部分の場合、単純に足しこんで大丈夫であろう。もし不幸にして才1行の成分のGCDが全成分のGCDにならなかったら、「適当な定数を才i行にかけて才1行に足す」ように、ステップ2を修正して再度施してみることも考えられる。

こで、前処理ステップ1の結果最初の単因子は g_{n^2} として既に求まっている
に注意されたい。したがって、第1行の消去によって、(1, 1)成分に残った g
因子であるならば、第1列の消去は不要になる。なぜなら、g は a_{i1} $(2\leq i\leq n)$
除し、かつ a_{ij} $(2\leq j\leq n)=0$ ゆえ、行に関する基本操作(第i行 − $\frac{a_{i1}}{g}$ ×
行)を行なっても、a_{ij} $(2\leq i,j\leq n)$ は不変だからである。
処理を含めたアルゴリズムは次のようになる。

<u>ルゴリズム(工夫された方法)</u>

行列Aに対して

(D1) 前処理を施す。(求まった単因子 g_{n^2} を以下では g とかく。)

(D2) 第1行を消去する。

(D3) もし $\hat{a}_{11}=g$ ならば(D7)へ。そうでないとき、次のいずれかを選べ。

・前処理ステップ2を修正して再度施して(D2)へ。

・(D4)へ。(古典的方法に切り換える。)

(D4) 第1列を消去する。

(D5) もし $\hat{a}_{11}=g$ ならば(D7)へ。

(D6) もし $\widetilde{a}_{12}=\widetilde{a}_{13}=\cdots=\widetilde{a}_{1n}=0$ でないならば(D2)へ。そうであるとき、$\hat{a}_{11}$ が割り切らない $\widetilde{a}_{ij}$ $(i,j\geq 2)$ を探して、それを含む行を第1行に加えて(D1)へ。

(D7) 行列は
$$\left(\begin{array}{c|ccc} g & 0 & \cdots & 0 \\ \hline * & & & \\ \vdots & & A' & \\ * & & & \end{array}\right)$$
(* … * は非零の場合もある)

という形になっている。A' に対してこのアルゴリズムを再帰的に適用する。

(注) 上述のアルゴリズムの(め7)ステップで、アルゴリズムを再帰的に適用する際に、A'の成分すべてをgで割った行列$\tilde{A}'$に対して適用することにすれば、行列の成分の次数の低下が起き、計算量の減少とメモリの節約が期待される。当然、Aの2番目の単因子は、$g\times$($\tilde{A}'$の最初の単因子)で与えられる。

5. REDUCE2 での実行例

古典的方法と工夫された方法の両アルゴリズムをREDUCE2でインプリメントし、いくつかの例について実行時間を比較してみた。以下にその結果を示す。

例1. (第1章で示したもの)

$$\begin{pmatrix} x & x-1 & x-1 & x-2 \\ x & x^3+x & x-1 & x^3+x-1 \\ x+1 & x^3+x+2 & x+1 & x^3+2x+3 \\ x-1 & x-3 & x-3 & -6 \end{pmatrix} \longrightarrow \begin{pmatrix} 1 & & & \\ & x+1 & & \\ & & x^3+1 & \\ & & & 0 \end{pmatrix}$$

$$\begin{cases} \text{古典的方法} & 1865 \text{ ms} \\ \text{工夫された方法} & 2309 \text{ ms} \end{cases}$$

例2. (例3の部分をとったもの)

$$\begin{pmatrix} x-2 & -1 & & & \\ & x-2 & -1 & & \\ & & x-2 & & \\ & & & x+1 & -1 \\ & & & & x+1 \end{pmatrix} \longrightarrow \begin{pmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & (x+1)^2(x-2)^3 \end{pmatrix}$$

$$\begin{cases} \text{古典的方法} & 1692 \text{ ms} \\ \text{工夫された方法} & 1668 \text{ ms} \end{cases}$$

例3 （文献[2]より）

$$\begin{pmatrix} x-2 & -1 & & & & & & & & \\ & x-2 & -1 & & & & & & & \\ & & x-2 & & & & & & & \\ & & & x+1 & -1 & & & & & \\ & & & & x+1 & & & & & \\ & & & & & x+1 & & & & \\ & & & & & & x-2 & & & \\ & & & & & & & x-2 & -1 & \\ & & & & & & & & x-2 & -1 \\ & & & & & & & & & x-2 \end{pmatrix} \longrightarrow \begin{pmatrix} 1 & & & & & & & & & \\ & 1 & & & & & & & & \\ & & 1 & & & & & & & \\ & & & 1 & & & & & & \\ & & & & 1 & & & & & \\ & & & & & 1 & & & & \\ & & & & & & 1 & & & \\ & & & & & & & x-2 & & \\ & & & & & & & & (x+1)(x-2)^3 & \\ & & & & & & & & & (x+1)^2(x-2)^3 \end{pmatrix}$$

$$\begin{cases} \text{古典的方法} & 9320 \ \mathrm{ms} \\ \text{工夫された方法} & 6493 \ \mathrm{ms} \end{cases}$$

使用した計算機はHITAC M-280Hである。

計算の途中経過をトレースしてみると、例1では古典的方法においても第1行、列の消去だけで単因子が求まっており、前のステップへの逆戻りは起きていないので、前処理を施す分だけ計算時間が余計にかかっているのが実情である。しかし、行列が大きくなっていくと逆戻りの起きる回数が増え、結果としては、工夫された方法によって計算時間の短縮が期待される。

参考文献

[1] エリ・ア・リュステルニク, ア・エル・ヤンポリスキー監修　佐藤常三監訳
麻島格次郎訳　現代応用数学ハンドブック4「高等代数」（総合図書）

[2] 斎藤正彦　線型代数入門　（東大出版会）

[3] K・ホフマン, R・クンツェ著　浅野啓三, 大林忠夫訳
線形代数学II　（培風館）

[4] 佐々木建昭　数式処理　（情報処理学会）