記号処理 12－17
(1980. 6. 21)

An Efficient Gaussian Elimination Method for Symbolic Linear Systems

村尾裕一(東大·理)　佐々木建昭(理研)

概要：多項式を係数とする線形方程式系の解を効率よく求める方法を示す。線形方程式の解法あるいは逆行列計算では、ガウス消去法が最も有力であり、本稿では、既に報告した行列式計算における効率的なガウス消去法が、この場合にも適用できることを示す。この消去法は、逆行列計算，特に back-substitution で著しい中間表式膨張を押さえ，容量・時間ともに効率的である。このことは、REDUCE 上にインプリメントし、既存の消去法と比較することにより実証された。

1．はじめに

行列を扱う場合、その行列式や，線形方程式の解および逆行列の計算において、一般にガウス消去法が用いられている。しかし、数式処理システムにおいて記号行列を扱う場合には、ガウス消去法をそのまま適用すると、途中の式は爆発的に膨張し、最後の結果よりもはるかに大きな表式を扱わねばならず、メモリという点でも、又時間的にも効率が悪い。この、所謂る中間表式膨張は、変数の数が多い程著しい。これを改善したのが Bareiss [1] で、消去の各ステップで除算を行うことにより元の行列式の要素についての整式のみを扱えばよいことを示し、中間表式膨張を押さえた。簡単に説明すると、消去を二度繰り返すと、一回前の消去で分母に現われた因子が、分子の各要素を割り切るのである（1-step algorithm）。（詳細は [1] あるいは [2] を参照されたい。）

しかし、Bareiss により導かれた式（後出）をよく見ると，小行列式の積を2つ計算し、その差をさらに小行列式で割ることにより，式の膨張を防いでいる。この除算は、Bareiss が示したように、割り切れることがわかっており，それにもかかわらず，一度大きな式を求めてから除算を行って式を小さくしているのである。筆者らは、割り切れることをうまく用いれば この無駄な計算が省けるのではないかと考え、Bareiss流の消去法をさらに改善しようと試みた。その方法が，[2] で示した我々のアルゴリズムで、既に行列式の計算に適用した。その方法とは、第一に、除算を行う際の目印となるように特別な変数を導入し、元の行列の対角要素を置き換える。即ち、割り切れる除算では（つまり商だけを求めるとき），次の例のように、分母の特定の項に注目して分子からそれに比例する項の係数をとってくれば、商に近い式が得られるのである。

例：$(\underline{ax} - ac + bc + ba \underline{- bx} - b^2) / (\underline{x} + b - c) = a - b$

第二に、それらの主変数に対して特殊な演算を定義する。即ち，導入した主変数の次数を考慮することにより、前記の乗算，除算において，不要な項は、計算する時点で取り除くのである。

行列式の計算に我々の方法を適用した結果は、多変数の場合には、Bareiss の消去法は、小行列式展開法に比肩する程度に改善されたが，実用上はアルゴリズムの簡明さという点で、小行列式展開法に軍配が上がるであろう。さて、この我々の

方法を、線形方程式の解法に適用しようというのが本稿の本題である。線形方程式を解く場合、ガウス消去法によるのが一般的であり、また、行列式計算に比べて係数からなる行列を対角化する分だけ消去の回数が多く大きな式を扱わねばならない。与えられた線形方程式系を $MX=B$, $X=(x_1,x_2,\cdots,x_n)^T$ とすると、求める解は $M^{-1}B$ であり、一方、M の逆行列の計算は $M^{-1}I$（I は単位行列）を求めることに他ならない。そこで、両方の問題を扱えるように、問題を次のように一般化しよう。

問題：与えられた２つの行列 $M(n\times n)$, $B(n\times m)$ に対して、$M^{-1}B$ を求める。但し $M=(a_{i,j})$, $B=(b_{i,j})$ とする。

以下、一般に行列 A の (i,j)要素を $A_{i,j}$ と書き，

$$M^{(1)}_{i,j}=\begin{cases} M_{i,j}\equiv a_{i,j}\,, & 1\le i,j\le n \\ B_{i,j-n}\equiv b_{i,j-n}\,, & 1\le i\le n,\ n+1\le j\le n+m \end{cases}$$

とする。

$M^{-1}B$ の計算は、次の２つの消去ステップに分けられる。

(1) Mの三角化　　(2) 対角化

(1)のステップは行列式計算における消去と全く同じであり、詳細は[２]にゆずる。

2．乗算記号⊗の導入と三角化

既に述べたように、特別な演算を行うために、$M^{(1)}$ の対角要素 $M^{(1)}_{k,k}$ を他の要素とは独立な変数 X_k $(1\le k\le n-1)$ で置き換える。

Bareissは、$M^{(1)}$の下三角部分の消去の過程を次の帰納的な式で表した。ここで $M^{(k+1)}$は $M^{(1)}$ に対して k 回消去を行った後の行列を表わす。

$$(1)\qquad M^{(k+1)}_{i,j}=(M^{(k)}_{k,k}\cdot M^{(k)}_{i,j}-M^{(k)}_{i,k}\cdot M^{(k)}_{k,j})/M^{(k-1)}_{k-1,k-1}\,,\qquad 1\le k<i,j;$$
$$M^{(k+1)}_{i,j}=M^{(k)}_{i,j}\,,\quad i\le k;\qquad M^{(0)}_{0,0}=1;$$

(Bareiss' one-step algorithm)

Bareiss はまた、(1)の第一式の右辺が割り切れ，$M^{(k+1)}_{i,j}$ は次のように表わされることを示した。

$$(2)\qquad M^{(k+1)}_{i,j}=\begin{vmatrix} X_1 & a_{1,2} & \cdots & a_{1,k} & a_{1,j} \\ a_{2,1} & X_2 & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & a_{k,k} & \vdots \\ a_{k,1} & \cdots & a_{k,k-1} & X_k & a_{k,j} \\ a_{i,1} & \cdots & \cdots & a_{i,k} & a_{i,j} \end{vmatrix}$$

ここで、$X_1,X_2,\cdots,X_{k-1}$ を主変数とする特殊な乗算を導入する。変数 $X_i(1\le i\le n-1)$ の多項式 P_1, P_2 の積が

$$P_1 \cdot P_2 = C \cdot X_1 X_2 \cdots X_{k-1} + (\text{terms not proportional to } X_1 X_2 \cdots X_{k-1})$$

となったとする。この時、特殊な乗算$\overset{k-1}{\otimes}$はCのみを計算し，その答をCとする。

$$P_1 \overset{k-1}{\otimes} P_2 = C.$$

但し、乗算記号$\otimes$の添字 $k-1$は、変数 $X_1, X_2, \cdots, X_{n-1}$ のうち $X_1, X_2, \cdots, X_{k-1}$ を主変数として扱うことを示す。

一般に、$M_{i,j}^{(k)}$は(2)で示されるように、各変数 X_i $(1 \leq i \leq n-1)$ について1次であり，特に $M_{k-1,k-1}^{(k-1)}$は、変数X_iについて最高次の項が $X_1 X_2 \cdots X_{k-1}$ である($M_{k-1,k-1}^{(k-1)} = X_1 X_2 \cdots X_{k-1} + R$)。これを $M_{k-1,k-1}^{(k-1)}$ の主項として$(X_1 X_2 \cdots X_{k-1})^2$を割り、その商，余りをそれぞれ $Q^{(k-1)}$, $R^{(k-1)}$ とする。

$$(3) \qquad (X_1 X_2 \cdots X_{k-1})^2 = Q^{(k-1)} M_{k-1,k-1}^{(k-1)} + R^{(k-1)}.$$

この$Q^{(k-1)}$は、$\otimes$を用いて計算され，主変数$X_1, X_2, \cdots, X_{k-1}$の次数を考えることにより，次のように表わされる。

$$(4) \qquad Q^{(k-1)} = \begin{cases} X_1 X_2 \cdots X_{k-1} - R + \sum\limits_{k-3 \geq 2i > 0} (-1)^{i+1} \underbrace{R \overset{k-1}{\otimes} \cdots \overset{k-1}{\otimes} R}_{i+1\text{ケ}}, & k > 1 \\ 1, \quad k = 1 \end{cases}$$

(1)の第一式と(3)とを辺々かけ合わせると，

$$(5) \qquad M_{i,j}^{(k+1)} \cdot (X_1 X_2 \cdots X_{k-1})^2 = (M_{k,k}^{(k)} M_{i,j}^{(k)} - M_{i,k}^{(k)} M_{k,j}^{(k)}) \cdot Q^{(k-1)} + M_{i,j}^{(k+1)} R^{(k-1)}.$$

ここで，主変数 $X_1, X_2, \cdots, X_{k-1}$の次数を考慮すると、右辺第二項は$M_{i,j}^{(k+1)}$に寄与せず，結局 $M_{i,j}^{(k+1)}$は、第一項の$(X_1 X_2 \cdots X_{k-1})^2$の係数として求められる。さらに，$Q^{(k-1)}$や$M_{i,j}^{(k)}$などが主変数について1次であることから，$M_{i,j}^{(k+1)}$は$\otimes$を用いて効率よく計算できる。

$$(6) \qquad \begin{aligned} M_{i,j}^{(k+1)} &= (M_{k,k}^{(k)} \overset{k-1}{\otimes} M_{i,j}^{(k)} - M_{i,k}^{(k)} \overset{k-1}{\otimes} M_{k,j}^{(k)}) \overset{k-1}{\otimes} Q^{(k-1)}, \quad 1 \leq k < i, j; \\ M_{i,j}^{(k+1)} &= M_{i,j}^{(k)}, \quad i \leq k. \end{aligned}$$

B. 対角化

(6)の計算をくり返すことにより、$M^{(1)}$は上三角行列$M^{(n)}$に変換される。次に、$M^{(n)}$の対角化，即ち上三角部分の消去を考えよう。Bareissは[1]で，(1)で表わされるような消去法を用いて対角化する方法を示しているが，$M_{n,n}^{(n)} = D \equiv |M|$ となることからわかるように，対角化にもこのような消去を行うと、最終的に(n,n)要素は$|M^{(n)}|$となり、その他の要素についても、必要以上に大きな式を扱わねばならない。なぜなら，クラメルの公式により、逆行列および線形方程式の解は、元の行列Mの要素について有理式になるが，その分母は高々Dであることがわか

っている。それゆえ、対角化する際に、対角成分が D となるように変換すれば有理式を扱わずにすむ。一般にはこのアルゴリズムが使われており、これを簡単な例で示そう。

$$M=\begin{bmatrix} x_1 & a_2 & a_3 \\ b_1 & x_2 & b_3 \\ c_1 & c_2 & c_3 \end{bmatrix} \qquad X=\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \qquad B=\begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix}$$

とし、$M^{(1)}$ を対角化すると、

$$M^{(3)}=\begin{pmatrix} x_1 & a_2 & a_3 & d_1 \\ \emptyset & x_1x_2-a_2b_1 & x_1b_3-a_3b_1 & x_1d_2-b_1d_1 \\ \emptyset & \emptyset & D & D_3 \end{pmatrix}, \quad D=\begin{vmatrix} x_1 & a_2 & a_3 \\ b_1 & x_2 & b_3 \\ c_1 & c_2 & c_3 \end{vmatrix}, D_3=\begin{vmatrix} x_1 & a_2 & d_1 \\ b_1 & x_2 & d_2 \\ c_1 & c_2 & d_3 \end{vmatrix}$$

これを解く場合、普通我々は、既に求まった $x_3=D/D_3$ を代入して次に x_2 を得るが、行列の変換としてみたとき、これは２行目の消去に対応する。この時、前述のように有理式を扱わないために、(2,2)要素が D となるように変換するのである。

$$\begin{pmatrix} x_1 & a_2 & a_3 & d_1 \\ \emptyset & D & \emptyset & \dfrac{(x_1d_2-b_1d_1)D}{x_1x_2-a_2b_1}-\dfrac{(x_1b_3-a_3b_1)D}{x_1x_2-a_2b_1}\cdot\dfrac{D_3}{D} \\ \emptyset & \emptyset & \emptyset & D_3 \end{pmatrix}$$

この(2,4)要素を実際に計算すると(省略)、確かに割り切れ、

$$D_2=\begin{vmatrix} x_1 & d_1 & a_3 \\ b_1 & d_2 & b_3 \\ c_1 & d_3 & c_3 \end{vmatrix}$$

となる。さらに、１行目についても消去を行うと、(1,4)要素は、

$$\frac{d_1D}{x_1}-\frac{a_2D}{x_1}\cdot\frac{D_2}{D}-\frac{d_1D}{x_1}\cdot\frac{D_3}{D}=\frac{d_1D-a_2D_2-d_1D_3}{x_1}=\begin{vmatrix} d_1 & a_2 & a_3 \\ d_2 & x_2 & b_3 \\ d_3 & c_2 & c_3 \end{vmatrix}$$

以上をまとめると、アルゴリズムは次の帰納的な式によって表わされる。

(7) $$M^{(d)}_{i,j}=\left(D\cdot M^{(n)}_{i,j}-\sum_{k=1}^{n-i} M^{(n)}_{i,i+k}\,M^{(d)}_{i+k,j}\right)\Big/M^{(n)}_{i,i}\ ,\quad 1\le i\le n-1,\quad n+1\le j\le n+m.$$

ここで再び、(7)式と(3)式で $k=i+1$ とした式とを辺々かけ合わせると $(M^{(n)}_{i,i}=M^{(i)}_{i,i})$

(8) $$M^{(d)}_{i,j}\cdot(x_1x_2\cdots x_i)^2=\left(D\cdot M^{(n)}_{i,j}-\sum_{k=1}^{n-i} M^{(n)}_{i,i+k}\,M^{(d)}_{i+k,j}\right)Q^{(i)}+M^{(d)}_{i,j}\cdot R^{(i)}$$

となる。$Q^{(i)}$ や $M^{(n)}_{i,j}$, $M^{(d)}_{i,j}$ などが主変数について１次であることに注意すると、

EFFICIENT GAUSSIAN ELIMINATION ALGORITHM (LINEAR SOLVE)

Input : $n \times n$ matrix M and $n \times m$ matrix B, both independent of X_i;

Output: $n \times m$ matrix $M^{(d)} = M^{-1}B$;

Step 1. [Augment M by B.]

$M^{(1)} \leftarrow M \oplus B$ (M augmented by B);

Step 2. [Replace diagonal elements by new variables.]

for $k \leftarrow 1$ to $n-1$ replace $M_{k,k}^{(1)}$ by X_k;

Step 3. [Eliminate columns.]

for $k \leftarrow 1$ to $n-1$ do

begin

if $k = 1$ then $Q^{(k-1)} \leftarrow 1$

else

begin

$R \leftarrow -M_{k-1,k-1}^{(k-1)} + X_1 X_2 \cdots X_{k-1}$;

$RR \leftarrow R$;

$Q^{(k-1)} \leftarrow X_1 X_2 \cdots X_{k-1} + R$;

while $RR \neq 0$ do

begin

$RR \leftarrow RR \overset{k-1}{\otimes} R$;

$Q^{(k-1)} \leftarrow Q^{(k-1)} + RR$

end

end; % calculation of $Q^{(k-1)}$;

for $i \leftarrow k+1$ to n do

for $j \leftarrow k+1$ to $n+m$ do

$M_{i,j}^{(k+1)} \leftarrow (M_{k,k}^{(k)} \overset{k-1}{\otimes} M_{i,j}^{(k)} - M_{i,k}^{(k)} \overset{k-1}{\otimes} M_{k,j}^{(k)}) \overset{k-1}{\otimes} Q^{(k-1)}$

end;

$D \leftarrow M_{n,n}^{(n)}$; % determinant of M;

Step 4. [Eliminate rows.(backsubstitution)]

for $i \leftarrow n-1$ to 1 step -1 do

for $j \leftarrow 1$ to m do

$M_{i,j}^{(d)} \leftarrow (D \overset{i}{\otimes} M_{i,n+j}^{(n)} - \sum_{k=1}^{n-i} M_{i,i+k}^{(n)} \overset{i}{\otimes} M_{i+k,n+j}^{(d)}) \overset{i}{\otimes} Q^{(i)}$;

Step 5. [Recover original polynomials.]

for $k \leftarrow 1$ to $n-1$ substitute $M_{k,k}^{(1)}$ for X_k in $M^{(d)}$ and D;

Step 6. [Normalize the diagonal elements to 1.]

for $i \leftarrow 1$ to n do

for $j \leftarrow 1$ to m do

$M_{i,j}^{(d)} \leftarrow M_{i,j}^{(d)}/D$;

Step 7. [Return.]

return $M^{(d)}$;

(8)式の第2項は$M_{i,j}^{(d)}$に寄与せず，右辺で$(X_1 X_2 \cdots X_i)^2$の係数のみを求めるには，

$$(9) \qquad M_{i,j}^{(d)} = \left(D \overset{i}{\otimes} M_{i,j}^{(n)} - \sum_{k=1}^{n-i} M_{i,i+k}^{(n)} \overset{i}{\otimes} M_{i+k,j}^{(d)} \right) \overset{i}{\otimes} Q^{(i)}.$$

上に示したように、我々の計算法は、対角化の場合にも適用でき，三角化・対角化と系統的に用いることにより、非常に効率よく$M^{-1}B$を計算できる。以上をまとめて前頁にアルゴリズムを示した。

4. Results and some consideration on 'Cramer's vs. Ours'

我々のアルゴリズムをREDUCE上にインプリメントし、既に組込まれて(1)+(7)のアルゴリズムと比較した。但し、どちらの場合もBの要素については全く同じ扱いをするのでB=I(単位行列)とし、逆行列M^{-1}の計算を行った。例題として用いた行列Mは次のとうりである。

(A) 一変数：

$$M_{i,j} = \begin{cases} 1+x^2 & i=j \\ x & i=j\pm1 \\ \emptyset & \text{otherwise} \end{cases}$$

(B) 多変数

$$M_{i,j} = \begin{cases} a & i=j \\ b & i=j\pm1 \\ c & i=j\pm2 \\ d & i=j\pm3 \end{cases}$$

問　題	n	(1)+(7)	our method
(A)	3	2.765	1.713
	4	10.403	5.633
	5	27.356	16.637
	6	63.192	44.144
(B)	3	3.185	2.255
	4	27.969	11.071
	5	160.823	44.862
	6	848.414	179.743
	7	-	664.800

(単位：秒)

我々の方法は、行列が多変数であるほど、又密であるほど，つまり中間表式膨張が著しくなるような行列に対して、その改善の度合は顕著になる。上の問題は比較的疎な場合だが，一変数の場合にも十分効果がでている。これは、元の行列が疎であっても、三角化することにより各要素の表式は大きくなり、対角化においてはその大きな式を消去しなければならないからである。

他の方法についても考えてみよう。例えば、クラメルの公式を用いて、各要素を2つの行列式の比として表わし、それぞれを小行列式展開法によって計算するという方法もあるが，$(n \cdot m+1)$個の$n \times n$行列式を計算しなくてはならず，一般的な問題にはむいていないと考えられる。この意味で、行列式計算ではあまり日の目を見ないガウス消去法も有用であり、我々の方法で十分効率化されたと思われる。

5. 我々の方法の拡張

一変数の場合のように中間表式膨張があまり著しくない場合は、我々の方法ではあまり効率化されず、行列式計算の場合にはBareissの方法がはるかに優れていた。そこで我々は、一変数の場合にも効率よく計算できないかと考えた。それには、2,3節で変数を導入してそれを目印として演算を行ったが，一変数の場合には、その変数自身を主変数とし、最高次の項に注目してやればよい。さらに、その変数に対して、積のうちある次数以上の項のみを計算する⊗のような乗算を定義すれば、補正因子の $Q^{(k-1)}$ も計算でき、前述と全く同様に効率化できる。

さらにこの考えを進めると、一般の場合にも、2節におけるように独立な変数を複数個導入せずに，1つだけ導入して上記の演算を行えばよいことがわかる。つまり、元の行列のすべての対角要素に全く独立な変数λを加え，消去の際にはλの次数を考慮して必要な項についてのみ計算を行うのである。インプリメントに際しても，データ構造の取り方などこの方が簡単になるが，実例をやってみればわかるように、消去の途中で不要な項がかなり現われる。このため、実際にインプリメントした結果でも、あまり良い結果が得られなかったことを報告しておく。

参考文献

1) E.H. Bareiss, "Sylvester's identity and multistep integer-preserving Gaussian elimination," *Math. Comp.* 22 (1968), pp. 565-578.

2) 佐々木建昭，村尾裕一，"記号行列式に対する効率的なガウス消去法，" 情報処理学会記号処理研究会，資料11（1980年3月）