数理解析研究所講究録
第406巻 1980年 13-24

数式処理の立場から見た線型代数
— 行列式 と 逆行列 —

東大・理　村尾裕一

1。はじめに

行列計算は応用分野が広く、数値計算においても記号計算においても、計算機上で頻繁に行われる計算である。なかでも、行列式や線形方程式は、不定積分アルゴリズムを初めとして需要が多いのだが，その計算過程における数式処理特有の中間表式膨張が問題となる。即ち、計算機上では、数式はポインタ等を用いてメモリ・セルの構造として表わされ、演算はそのデータ構造を変更あるいは作り変えることによりなされる。このため、式が大きくなる程多くのメモリ・セルを必要とし、又演算の度にメモリ・セルは消費される。行列式計算などのように必然的に式が大きくなる場合、最悪の場合には、計算途上の表式が最後の答の表式よりはるかに長大になり、メモリ不足により計算しきれなくなる。これが所謂る中間表式膨張である。本稿では、このような点をふまえて、行列式

及び線形方程式の解の計算を高速化した新たな方法について説明する。

行列式や線形方程式の解を求めるには、数値計算ではガウス消去法が最も高速かつ一般的であるが，記号行列に対してガウス消去法をそのまま適用すると、行列式自体は元の行列要素の整式であるにもかかわらず有理式を扱わねばならない。線形方程式の解についても、クラメルの公式により、係数行列式をかけておけば、多項式計算だけですませられることはわかっている。しかし、この点については、Bareiss[1]が示したように、除算を行うことにより途中の式は整式におさえられる。つまり、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は、分母の各要素を割り切るのである(one-step fraction-free algorithm)。

(1) $$M_{i,j}^{(k+1)} = \left(M_{k,k}^{(k)} M_{i,j}^{(k)} - M_{i,k}^{(k)} M_{k,j}^{(k)}\right) / M_{k-1,k-1}^{(k-1)}, \quad 1 \le k < i,j,$$
$$M_{i,j}^{(k+1)} = M_{i,j}^{(k)}, \quad i \le k, \quad M_{0,0}^{(0)} = 1.$$

但し、$M^{(1)} \equiv M$ は与えられた $n \times n$ 行列。

上式は、与えられた行列Mの下三角部分の消去の過程を帰納的に表した式であり、行列式は $M_{n,n}^{(n)}$ で求まる。

(1)式を用いて、消去の各ステップで除算を行うことにより、中間式の膨張をおさえることができる。しかし、(1)の右辺では、分子の(一般に)大きな式を一度計算し、分母の小行列

式で割ることにより式を小さくしている。(1)式の右辺は割り切れることから、このような無駄な計算が省けるはずである。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、元の対角要素を置き換える。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。

問題を定式化しよう。行列式については，与えられた行列Mを(1)式のように上三角化し、$M^{(n)}_{n,n}$ をもとめる。線形方程式の場合，一般に方程式系を $MX=B$ とすると，$M^{-1}B$ をもとめることに他ならない。以下、与えられた2つの行列 $M(n\times n)$, $B(n\times m)$ とし、$M=(a_{i,j})$, $B=(b_{i,j})$ とする。また，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。

(a) M の上三角化　　(b) 対角化

(a) のステップは行列式計算と共通である。

2. 乗算記号 $\otimes$ の導入と上三角化

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。そして、Bareiss [1] によれば、(1) の第一式の右辺が割り切れた結果、$M^{(k+1)}_{i,j}$ は次のように表わされる。

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

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

$$P_1 \cdot P_2 = C \cdot X_1 X_2 \cdots X_{k-1} + (X_1 X_2 \cdots X_{k-1} \text{ に比例しない項})$$

となったとする。この時、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、記号 $\otimes$ の添字 $k-1$ は、変数 $X_1, X_2, \cdots, X_{n-1}$ のうち、$X_1, X_2, \cdots, X_{k-1}$ のみを主変数を扱うことを示す。

一般に、$M^{(k)}_{i,j}$ は (2) で示されるように、各変数 X_i $(1 \le i \le n-1)$ 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 $X_1X_2\cdots X_{k-1}$ である（$M^{(k-1)}_{k-1,k-1} = X_1X_2\cdots X_{k-1} + R$）。これを $M^{(k-1)}_{k-1,k-1}$ の主項として $(X_1X_2\cdots X_{k-1})^2$ を割り、その商及び余りをそれぞれ $Q^{(k-1)}$, $R^{(k-1)}$ とする。

(3) $\quad (X_1X_2\cdots X_{k-1})^2 = Q^{(k-1)}M^{(k-1)}_{k-1,k-1} + R^{(k-1)}$.

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

(4) $$Q^{(k-1)} = \begin{cases} X_1X_2\cdots X_{k-1} - R + \sum_{k-3\geq 2i>0} (-1)^{i+1} \underbrace{R \overset{k-1}{\otimes} \cdots \overset{k-1}{\otimes} R}_{i+1\text{個}}, & k>1 \\ 1 & k=1 \end{cases}$$

(1)の第一式と(3)とを辺々かけ合わせると、

(5) $$M^{(k+1)}_{i,j}\cdot(X_1X_2\cdots X_{k-1})^2 = (M^{(k)}_{k,k}M^{(k)}_{i,j} - M^{(k)}_{i,k}M^{(k)}_{k,j})Q^{(k-1)} + M^{(k+1)}_{i,j}R^{(k-1)}.$$

ここで、主変数 $X_1, X_2, \cdots, X_{k-1}$ の次数を考慮すると、右辺第二項には $(X_1X_2\cdots X_{k-1})^2$ に比例する項は含まれず、結局 $M^{(k+1)}_{i,j}$ は、第一項の $(X_1X_2\cdots X_{k-1})^2$ の係数として求められる。さらに、$Q^{(k-1)}$ や $M^{(k)}_{i,j}$ などが主変数について1次であることから、$M^{(k+1)}_{i,j}$ は $\otimes$ を用いて効率よく計算できる。

(6) $$M^{(k+1)}_{i,j} = (M^{(k)}_{k,k} \overset{k-1}{\otimes} M^{(k)}_{i,j} - M^{(k)}_{i,k} \overset{k-1}{\otimes} M^{(k)}_{k,j}) \overset{k-1}{\otimes} Q^{(k-1)}, \quad 1\leq k < i,j$$

$$M^{(k+1)}_{i,j} = M^{(k)}_{i,j}, \quad i\leq k.$$

3. 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)式の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，逆行列及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。この過程は，次の帰納的な式によって表わされる。

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

$$1 \le i \le n-1 \ , \quad n+1 \le j \le n+m .$$

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

$$(8)\quad M^{(d)}_{i,j} (x_1 x_2 \cdots x_i)^2 = \left(D M^{(n)}_{i,j} - \sum_{k=1}^{n-i} M^{(n)}_{i,i+k} M^{(d)}_{i+k,j}\right) Q^{(i)} + M^{(d)}_{i,j} R^{(i)}.$$

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，(8)式の第二項は $M^{(d)}_{i,j}$ に寄与せず，右辺で

Input : n×n matrix M and n×m matrix B, both independent of X_i;

Output: n×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 $M^{(1)}_{k,k}$ 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

$R \leftarrow -M^{(k-1)}_{k-1,k-1} + X_1X_2\cdots X_{k-1}$;

$RR \leftarrow R$;

$Q^{(k-1)} \leftarrow X_1X_2\cdots X_{k-1} + R$;

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

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

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

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

Step 5. [Recover original polynomials.]

for $k \leftarrow 1$ to n-1 substitute $M^{(1)}_{k,k}$ 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

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

Step 7. [Return.]

return $M^{(d)}$;

$(X_1X_2\cdots X_i)^2$ の係数のみを求めるには、

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

とすればよいことがわかる。

以上のように、このような計算法は、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、$M^{-1}B$ を効率よく計算できる。以上をまとめて、前頁にアルゴリズムを示した。

4. 他の方法との比較

一般に、式の膨張は多変数になる程はげしく、また、除算におけるコストも大きくなる。このため、元の行列が密である程、また変数の数が多くなる程、除算を除いた上記の方法は効力を発揮することになる。この方法を REDUCE にインプリメントし、いくつかの簡単な例題に対して計算時間(CPU time)を計測した結果を表に示した。表中の Bareiss の方法は、(1)式を一般化し、iteration の回数を減らした 2-step algorithm [1] であるが、上記の方法による効果が充分にみられる。一変数の場合には、式の膨張が非常に小さく、除算も容易なため 2-step algorithm はかなり高速である。さらに、逆行列計算では、対角化に際しては補正因子の $Q^{(i)}$ は計算済みであり、又各式が必然的に大きく密になっているために、その効果は大きい。このように、一般に式の膨張のはげしい、多変数で密

な行列に対して上記の方法は有効である。

計算時間の計測結果（単位：秒）

（REDUCE - HLISP interpreter　H-8700）

行列式

	次数	小行列式法	上記の方法	Bareiss法
Vandermonde	3	0.422	0.460	0.429
$a_{i,j}=x_j^{i-1}$	4	1.135	1.985	5.563
	5	5.375	11.684	43.092
	6	33.114	78.590	（未計測）
対称Toeplitz	3	0.615	0.890	0.876
	4	3.706	3.639	7.472
$a_{i,j}=x_{\|i-j\|}$	5	19.579	16.696	46.606
	6	118.312	93.917	471.206
	7	822.449	579.006	—
一変数	3	2.047	2.255	1.037
$a_{i,j}=\sum_{k=0}^{i+j-2} x^k$	4	14.000	27.768	3.287
	5	137.988	278.286	8.193

逆行列

	次数	上記の方法	(1)+(7)
一変数	3	1.713	2.765
$a_{i,j}=\begin{cases}1+x^2 & i=j\\ x & i=j\pm1\\ 0 & \text{otherwise}\end{cases}$	4	5.633	10.403
	5	16.637	27.356
	6	44.144	63.192

多変数			
$a_{i,j}=$ a: $i=j$, b: $i=j\pm1$, c: $i=j\pm2$, d: $i=j\pm3$	3	2.255	3.185
	4	11.071	27.969
	5	44.862	160.823
	6	179.743	848.414
	7	664.800	

表にも示されているように、数式処理における行列式計算では、小行列式展開法も有力な方法である。これは、記号行列を扱う場合には、ガウス消去法には前述のような欠点があるのに比べ、小行列式展開法では除算の必要がなく、アルゴリズム自体簡明なためである。小行列式による展開では、より低次の小行列式を、recursiveに計算するわけだが、行列の次数が高くなると、同じ小行列式を何度も計算することになる。そのような小行列式を結果がでる毎にストアしておき、必要な時にhashingを用いてその結果をとり出すことにより、組織的に重複計算を避ける方法も知られている[2]。

逆行列計算あるいは線形方程式の解法では、他にクラメルの公式を用いて各行列式を展開して解く方法も考えられる。しかし、nm個の行列式の計算が必要となるため、一般的ではないであろう。但し、この場合にも、重複計算を避ける方法は提案されており[3]，疎な行列に対しては高速である。

総じて、小行列式による展開法は、疎な行列に対しては高速であるが、ストアしておく方法による場合密な行列に対し

では、途中の小行列式の小さくはない式を記憶しておく分だけメモリ的に不利であろう。

以上のように、行列式などのように簡単な計算でも、数式処理においては、実用上のメモリ・バウンドをいかに克服するか、又いかに高速化するかというのは依然として大きな問題である。

5. 拡張

一変数の行列式計算のように、中間表式膨張が著しくない場合、前述の方法では項の打ち消し合いが減り、又overheadにもより却って非効率化することになってしまった。しかしこの場合には、特別な変数を導入してそれに着目して行った演算を、その一変数自身について、次数を考慮して同様のことを行えばよい。さらにこの考えを進めると、(一般の場合にも)独立な変数を導入せずに、一変数λだけ導入すればよいことがわかる。つまり、元の行列の対角要素にλをたし込み、途中の式のλの次数を数えてやれば、容易に同様の式が導かれる。ところがこの方法では、本来現われるはずのない、元の要素の中乗に比例する項が現われ、インプリメントした結果でも実用的で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 multi-step integer-preserving Gaussian elimination. Math. Comp. 22 (1968), 565-578

[2] W. M. Gentleman AND S. C. Johnson, Analysis of algorithms, a case study : determinants of polynomials. Proc. 5th ACM Symp. on Theory of Computing, Austin, Texas (1973), P.P. 135-142

[3] M. L. Griss The algebraic solution of sparse linear systems via minor expansion. TOMS 2(1976), 31-49