

一般逆行列の数式処理システムによる直接解法とその評価†

野田 松太郎†† 泉 田 正 則††† 越 智 正 明†††*

数式処理システムの普及とともにいろいろの分野で誤差をとまわらない解法の必要性がいられている。特に、特異あるいは長方形に対する一般逆行列の正確解法の確立は重要である。これに関して数値計算の立場から解法の評価の試みなどはなされてもいるが、数式処理システムを用いた解法の比較研究は試みられてもいない。本論では、いくつかの一般逆行列に対する直接解法を演算量とメモリ効率の両面から解析的に比較検討する。また、それらを数式処理システムにインプリメントし、解析的に求めた比較基準の妥当性を、実行例を通じて確かめる。これらを通じ、現在発表されている直接解法の中では Greville のアルゴリズムが数式処理システムでの計算に最適であるとの結論を得る。

1. 序 論

一般逆行列が Moore により提案されて半世紀近くになる。その応用も、最小二乗法に代表されるような統計解析を初めとして、各分野でなされている。例えば、 m 個のデータ点を n 個の基底関数の一次結合 (多項式基底等) で表そうとする問題を考える。 $m=n$ の場合には連立一次方程式を解く単純な問題に帰結される。 $m>n$ の場合にはデータ過多で、連立一次方程式の係数行列は特異となり、通常の数値計算法で解を得ることはできない。この場合に一般逆行列が必要となる。

一般逆行列を求めるための数値計算法として特異値分解法が広く用いられているが、この方法は、小さな特異値が意味ある場合でも、それがしきい値以下なら誤差とみなし、無視して計算する。このため、計算結果は真の値と大きく異なる場合がある。そこで、誤差を含まず一般逆行列を求める汎用的な手法が見直されている。また、行列の要素が数値のみでなく、変数や多項式である場合には、数値解法以外の手法が必要となる。

そこで、本論では一般逆行列の解法を整理し、記号行列も含めて誤差のない計算を実行するため、数式処理システムへのインプリメントを行う。本論で扱う一般逆行列は、広く用いられるムーア・ペンローズ型一

般逆行列 (Moore-Penrose generalized inverse) に限る。この種の研究は、数値計算の立場から Shinozaki らにより直接解法と反復解法の両面から行われているが^{1),2)}、誤差のない計算をする場合には直接解法のみを考えればよい。誤差のない計算に関して、Gregory, Krishnamurthy らは、モジュラ演算を用いて、整数体上のみで一般逆行列を求めている^{3),4)}。この方法は、対象とする行列が記号行列の場合には無力である。一方、数式処理システムを用いて一般逆行列を求める試みは、現在では MACSYMA 上に実現されたもの⁵⁾が発表されているのみである。しかし、この方法は極限計算を必要とし、効率的とは思えない。

本論では、一般逆行列の直接解法として、Gauss 消去法に基づく Hermite のアルゴリズム⁶⁾、文献 5) で用いられた Albert の極限形式⁶⁾、数値計算法として提案された行列の列 (または行) 分割による Greville のアルゴリズム⁷⁾、特性多項式の計算による Decell-Leverrier のアルゴリズム⁸⁾、LU 分解法の拡張として示された Ben-Israel, Wersan のアルゴリズム¹⁰⁾、および、Gram-Schmidt の直交化法を用いた Glassey のアルゴリズム¹¹⁾をとりあげる。これらを計算量と必要メモリ量の両面から評価し、数式処理システム (MACSYMA, REDUCE) 上へインプリメントする。これにより、数式処理で有効なアルゴリズムの特徴を検討し、今後の数式処理に適するアルゴリズム開発の方向を探る。また、計算例により、上の評価の正しさを確認する。

第 2 章に一般逆行列の定義および性質を簡単にまとめ、一般逆行列の代表的な数値計算法である特異値分解法の問題点を述べる。第 3 章では上記の各アルゴリズムを概説する。それらに対する計算量、必要メモリ量の評価、検討および数式処理システムへのインプリ

† Evaluation of Exact Computation Methods of the Generalized Inverse by MATU-TAROW NODA (Department of Computer Science, Ehime University), MASANORI IZUMIDA and MASAKI OCHI (Department of Electronics Engineering, Ehime University).

†† 愛媛大学工学部情報工学科

††† 愛媛大学工学部電子工学科

* 現在 松下電器産業(株)情報機器本部コンピュータ事業部
Computer Division, Information Equipment Sector,
Matsushita Electric Industrial Co., Ltd.

メントを第4章で与える。悪条件行列、長方形等についての計算例を第5章で示す。比較結果と今後の課題を第6章で述べる。

2. 一般逆行列と数値計算法

Moore-Penrose 型一般逆行列を中心に、簡単に一般逆行列の定義、性質をまとめる。数値計算法として広く用いられている、特異値分解法の問題点を指摘する。

一般逆行列の定義 正則行列 A に対する逆行列 A^{-1} はただ一つ存在する。行列 A が特異、または長方形であるとき、逆行列 A^{-1} は存在し得ない。しかし、次の性質を満たす一般逆行列が存在する。

任意の $m \times n$ 行列 A , $n \times m$ 行列 G に対して次の行列方程式を考える。

$$AGA = A \quad (2.1)$$

$$GAG = G \quad (2.2)$$

$$(AG)^T = AG \quad (2.3)$$

$$(GA)^T = GA \quad (2.4)$$

このとき、行列 G が (2.1) 式を満たすなら、 G を行列 A の一般逆行列 (g-inverse: A^-) という。 G が (2.1) に加え、上の各式のどれかを満たすとき、以下のような特別の名称を持つ一般逆行列になる。

(2.2) 式: 反射型 (reflexive: A_R)

(2.3) 式: 最小二乗型 (least-square: A_L)

(2.4) 式: ノルム最小型 (minimum-norm: A_N)

特に、 G が (2.1) ~ (2.4) 全式を満たすとき、 G を行列 A の Moore-Penrose 型一般逆行列といい A^+ と表す。他の一般逆行列と異なり、Moore-Penrose 型一般逆行列 A^+ は一意的である。Moore-Penrose 型一般逆行列は、次の性質を持つ。

(a) A が正則なら、 $A^+ = A^{-1}$

(b) $A = 0$ なら、 $A^+ = 0$

(c) A を階数 m の $m \times n$ 行列とすると、

$$A^+ = A^T(AA^T)^{-1} \quad (2.5)$$

上記において A は任意の $m \times n$ 行列であるが、本論では $m \leq n$ と仮定する。この仮定を設けても以下の議論の一般性は失わない。

特異値分解法とその問題点 現在、一般逆行列の計算はほとんど特異値分解法⁹⁾による数値計算によって行っている。特異値分解法では、任意の $m \times n$ 行列 A が与えられたとき、二つの直交する $m \times m$ 行列と U と $n \times n$ 行列 V とを選び、

$$A = U \Sigma V^T \quad (2.6)$$

と分解する。ここで、 $m \times n$ 行列 Σ の対角部分の要素は A の特異値 σ_i であり、他の要素はすべて 0 である。このとき、 A の Moore-Penrose 型一般逆行列は

$$A^+ = V \Sigma^+ U^T \quad (2.7)$$

となる。なお、 $\Sigma^+ = [\text{diag}(1/\sigma_i); 0]^T$ である。

数値計算を行う場合の困難は、小さな特異値の出現と浮動小数演算による誤差との関連にある。特異値分解法では、あるしきい値を設定し、絶対値がその値以下であれば計算誤差による「見かけ上の特異値」とみなして 0 とする。ところが、行列が悪条件である場合、数値計算上の相対誤差はしばしば最小・最大特異値間の比を越える大きさになるため、しきい値による方法は正しく働かなくなる恐れが多い。

3. 一般逆行列を求めるアルゴリズムの概観

Moore-Penrose 型の一般逆行列を数値計算で求める方法は、大型計算機システムのライブラリとしても登録されている特異値分解法がほとんどである。そのため、悪条件の問題に対しては常に解の正確さに関する恐れを抱き続けねばならない。一方、数式処理的にこれを求める試みはあまりなく、Frawley⁵⁾によるものが実現されているのみである。数式処理システムが普及し大規模な問題も扱える今日、数式処理に適し、汎用的で、かつ取り扱いの容易なアルゴリズムの選択が必要と思われる。そのために、現在までに発表された一般逆行列を求めるための数値計算用のアルゴリズムの中から、誤差のない計算にも適するものを取りあげる。

Hermite のアルゴリズム: Hr^{3),4)}

任意の $m \times n$ 行列 A に対し、Gauss 消去法を拡張する。一般逆行列を次式の形で表現する。

$$A^+ = A^T M_R A A^T \quad (3.1)$$

ここで $M = (A A^T)^2$ であり、 M_R は行列 M の反射型一般逆行列である。 M_R は次のように求める。行列 A の階数を r とすると、 $m \times m$ 正則行列 E と F が存在し、

$$F M^T E^T = R = \begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix} \quad (3.2)$$

を満たす。ただし I_r は $r \times r$ の単位行列である。一般逆行列の性質より、 M_R は

$$M_R = F^T R E \quad (3.3)$$

と書ける。 M と単位行列 I から構成される、 $m \times 2m$ 拡大行列 $[M: I]$ に対する行基本操作により、 $[M_1: E]$

が得られる。同様に、 M_1^T と I から構成される拡大行列 $[M_1^T: I]$ に対する行基本操作により、 $[R: F]$ が求まる。(3.3), (3.1)式を用いれば、 A^+ を得るのは容易である。

Albert の極限形式: $AI^{(6)}$

a をスカラー量とし、

$$a^+ = \begin{cases} a^{-1} & (a \neq 0) \\ a=0 & (a=0) \end{cases} \quad (3.4)$$

とすると、 a^+ は Moore-Penrose 型一般逆行列の定義を満たす。 $a \rightarrow 0$ のとき、 a^+ は明らかに $a=0$ で連続ではない。また a を横ベクトルとしたときには、(2.5)(c)より、

$$a^+ = \begin{cases} a^T/aa^T & (a \neq 0) \\ a^T=0 & (a=0) \end{cases} \quad (3.5)$$

である。同様に、 A が任意の行列のとき、次の A^+ が Moore-Penrose 型一般逆行列となる。

$$A^+ = \lim_{x \rightarrow 0} A^T (AA^T + x^2 I)^{-1} \quad (3.6)$$

これが一般逆行列に関する極限形式である。

Greville のアルゴリズム: $Gr^{(9),(7)}$

アルゴリズムは入力行列の行分割、分割された行で構成される行列の一般逆行列計算、および新しい行の取り込み操作からなる。

今、 $m \times n$ 行列 A を

$$A = [a_1^T, a_2^T, \dots, a_n^T]^T$$

と行ベクトル a_i に分割し、 $i \times n$ 行列 A_i を

$$A_1 = a_1 \quad A_i = \begin{bmatrix} A_{i-1} \\ a_i \end{bmatrix} \quad (3.7)$$

と定める。 $n \times i$ 行列 A_i^+ は次式により、 $i=1, 2, \dots$ と逐次的に計算する。

$$A_i^+ = [A_{i-1}^+ - b_i^T d_i : b_i^T] \quad (3.8)$$

ここで、 d_i, c_i, b_i および、初項 A_1^+ は各々

$$d_i = a_i A_{i-1}^+ \quad (3.9)$$

$$c_i = a_i - d_i A_{i-1} \quad (3.10)$$

$$b_i = \begin{cases} \frac{c_i}{c_i c_i^T} & (c_i \neq 0) \\ \frac{d_i (A_{i-1}^+)^T}{1 + d_i d_i^T} & (c_i = 0) \end{cases} \quad (3.11)$$

$$A_1^+ = \begin{cases} a_1^T & (a_1 \neq 0) \\ a_1 a_1^T & (a_1 = 0) \\ a_1^T & (a_1 = 0) \end{cases} \quad (3.12)$$

である。この手順を $i=m$ まで反復すれば、そのときの A_i^+ が A の Moore-Penrose 型一般逆行列 A^+ となる。見てわかるとおり、この算法は 0 判定を必要とするから、浮動小数計算には不向きだが、数式処理で

は何ら問題を生じない。

Decell-Leverrier のアルゴリズム: $DI^{(9),(8)}$

任意の $m \times n$ 複素行列 A に対する特性多項式に関する Cayley-Hamilton の定理の拡大による。今、 A^H を A の Hermite 共役行列とする。行列 $B = AA^H$ は次の特性多項式を持つ。ただし、 $b_0=1$ とする。

$$B(\lambda) = (-1)^m (b_0 \lambda^m + b_1 \lambda^{m-1} + \dots + b_m) \quad (3.13)$$

r を $b_r \neq 0$ を満たす最大の整数とすると、 r は行列 A の階数となる。行列 A の一般逆行列は次式となる。

$$A^+ = \begin{cases} -(A^H (b_0 B^{r-1} + b_1 B^{r-2} + \dots + b_{r-1} I)) / b_r & (r > 0) \\ 0 & (r = 0) \end{cases} \quad (3.14)$$

b_r, A^+ の効率的計算法は次の列の計算にある⁹⁾。

$$A_0 = 0, \quad q_0 = -1, \quad B_0 = I,$$

$$A_1 = AA^T, \quad q_1 = tr A, \quad B_1 = A_1 - q_1 I,$$

$$\dots \quad (3.15)$$

$$A_r = A_1 B_{r-1}, \quad q_r = tr A_r / r, \quad B_r = A_r - q_r I,$$

ここで、 $q_i = -b_i$ である。(3.14)式により、 $r > 0$ のとき A^+ は次のようになる。

$$A^+ = \frac{1}{q_r} A^T B_{r-1} \quad (3.16)$$

ただし、 r は

$$A_1 B_r = 0 \quad (3.17)$$

が成立するまで上記のアルゴリズムを反復し決定する。

Ben-Israel, Wersan のアルゴリズム: $Bw^{(1),(10)}$

一般逆行列を定める場合に、 Hr においては反射型一般逆行列を用いたが、ここではノルム最小型一般逆行列を用いる。 A^+ は次式で求められる。

$$A^+ = A^T (AA^T)^M \quad (3.18)$$

$(AA^T)^M$ を求めるには、 AA^T の LU 分解を行う。再び、行列 A の階数を r とする。 L は $r \times r$ 下三角行列 R を含む次のような $m \times r$ 行列となる。

$$L = [R^T : S^T]^T$$

ここで、 S は $(m-r) \times r$ 行列である。一方、 U は $r \times r$ 上三角行列を含み、階数 r の長方形行列になる。 L の一般逆行列 L^- は

$$L^- = [R^{-1} : 0]$$

と得られる。 U の一般逆行列 U^+ は

$$U^+ = U^T (UU^T)^{-1}$$

となる。これらにより、

$$(AA^T)^M = U^+ L^- \quad (3.19)$$

が計算できる。(3.18)式より、 A^+ が得られる。

Glassey のアルゴリズム: GI^{(1), (11)}

Gram-Schmidt の直交化法に基づいている。 $m \times n$ 行列 A の列の張る空間に対し、その直交基底を行とする $n \times m$ 行列を B とする。 A と B から $C=BA$ を作る。このとき、 A^+ は次のように書かれる。

$$A^+ = C^T(CCT)^{-1}B \tag{3.20}$$

B を求めるには、 A の列の直交基底を計算すればよい。

$$A = [a_1^T, a_2^T, \dots, a_m^T]$$

より、ベクトル b_i を次のように求める。

$$b_1 = a_1 \tag{3.21}$$

$$b_i = a_i - \sum_{k=1}^{i-1} \{(a_i, b_k) / (b_k, b_k)\} b_k$$

ここで、 $i=2, \dots, r, j=1, \dots, m$ である。直交基底は従属した列については求められないため、 a_j が $a_i (s < j)$ に従属するとき、 $b_i = 0$ となる。

4. アルゴリズムの比較とインプリメント

3章で示したアルゴリズム (**Hr**, **Al**, **Gr**, **DI**, **Bw** および **GI**) の評価と数式処理システムへのインプリメントを行う。アルゴリズムの評価では、その演算回数 (計算量) とメモリ効率 (必要なメモリ量) を考える。

4.1 計算量の比較

計算量の尺度を次のように定める。

(乗除算の回数 + 加減算の回数) / 2

各アルゴリズム \mathcal{A} に対し、その漸近挙動 ($m, n \gg 1$) を求め、 \mathcal{A} の演算量の主要項といい、 $\phi(\mathcal{A})$ と書く。 $\phi(\mathcal{A})$ の計算は付録1に示すが、結果は次のとおりである。ただし、行列の階数を r とする。

$$\phi(\mathbf{Hr}) = m^3 + 3m^2n + 5m^2r - 3mr^2 + 2r^3/3 \tag{4.1}$$

$$\phi(\mathbf{Al}) = m^3 + 2m^2n + (\text{極限操作}) \tag{4.2}$$

$$\phi(\mathbf{Gr}) = 2m^2n - nr^2/2 \tag{4.3}$$

$$\phi(\mathbf{DI}) = m^3r - m^3 + 2m^2n + m^2r/2 \tag{4.4}$$

$$\phi(\mathbf{Bw}) = 2m^2n + 2m^2r + mr^2 + 4r^3/3 \tag{4.5}$$

$$\phi(\mathbf{GI}) = 5mnr - mr^2/2 + nr^2 + r^3 \tag{4.6}$$

なお、上記の計算においてピボット操作 (**Hr**)、行列の取り込み操作 (**Gr**) 等の付随的操作は無視した。 **Albert** の極限形式では、行列が数値行列であっても変数 x の導入による記号処理や極限操作が必要になる。(4.2) を得る場合には、記号処理部分を省略し、極限操作は $x=0$ の代入操作に置き換えた (付録2参照)。また、 $m \times n$ 長方形行列の場合、行列の形状により演算量が異

なるが、ここでは2章で述べたように、行列を $m \leq n$ の場合に限った。さらに、 **Gr**, **GI** では、一次従属な行のある位置によって計算量が変化するが、計算量が最大になる場合のみを示した。

(4.1)~(4.6)式の比較より、階数があまり低下しない行列を扱う場合には **Gr** の計算量が少なくなるといえる。階数が大きく低下する行列に対しては、他のアルゴリズムも **Gr** と比較し得るように思える。図1, 2に、 $\phi(\mathcal{A})$ を無次元化した $\phi(\mathcal{A})/n^3$ を、 $k=m/n \leq 1$ の関数として、 $r'=r/n$ を変えて示した。ただし、 **DI** は、 r が小の限られた場合以外は演算量が非常に大となるため、省略した。演算量に関しては、図1より $r'=0.2$ では、 k の如何にかかわらず、 **GI** が **Gr** より小であるが、 $r'=0.5$ に対する図2では、 **Gr** が **GI** より小となる。この逆転は、(4.3), (4.6)両式によれば、 $r' \approx 0.37$ で起こる。すなわち、階数の低下が大の行列では **GI** の演算量が小だが、それ以外では **Gr** が小となる。もちろん、階数低下のない正方行列 ($k=1, r'=1$) では **Gr** の演算量が最小である。以上より、様々な階数の行列を扱う汎用アルゴリズムとして、 **Gr** が演算量の面では優れているといえる。

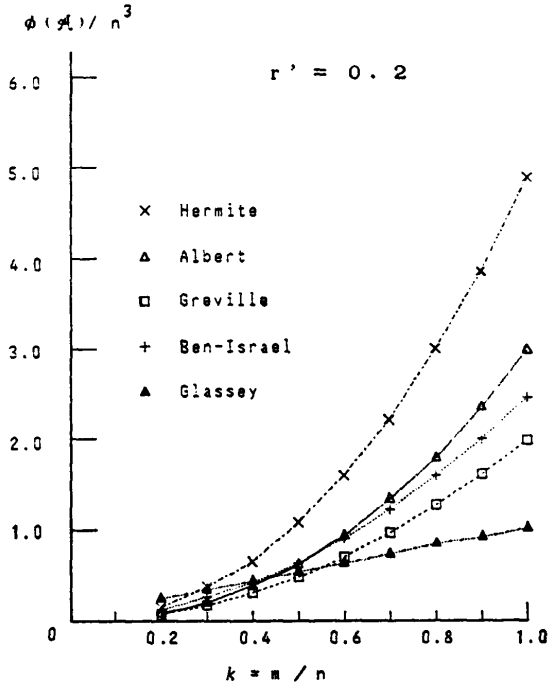


図1 計算量の解析結果 ($r'=0.2$)
 Fig. 1 Required number of operations which are obtained by analysis of six algorithms. Where $r' = \text{rank}/n = 0.2$.

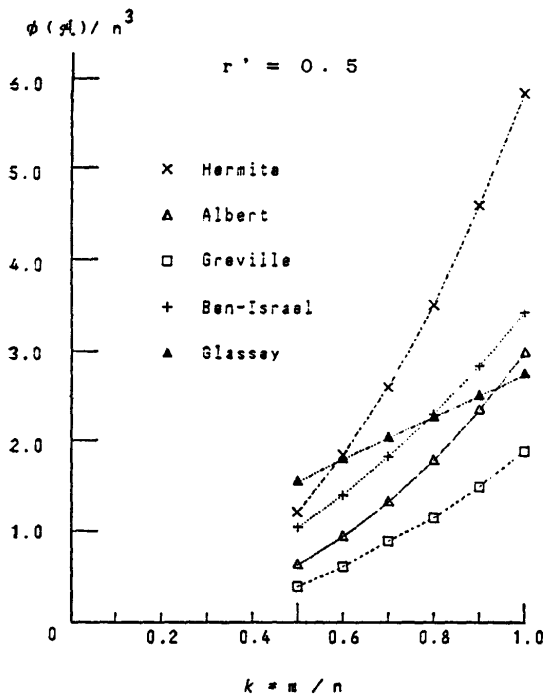


図2 計算量の解析結果 ($r'=0.5$)

Fig. 2 Required number of operations for $r'=0.5$.

4.2 メモリ効率の比較

次に、各アルゴリズムをメモリ効率の面から比較検討する。アルゴリズム \mathcal{A} の実行に際し、必要最低限のメモリ量の主要項 ($m, n \gg 1$ の場合) を $\rho(\mathcal{A})$ とする。これは、入力行列の大きさのみならず、作業領域として必要な行列の個数と大きさにもよっている。結果は以下のとおりである (参考のため、Gr に対する計算だけを付録1に示す)。

$$\rho(\mathbf{Hr}) = 4m^2 + 2mn \quad (4.7)$$

$$\rho(\mathbf{Al}) = m^2 + 2mn \quad (4.8)$$

$$\rho(\mathbf{Gr}) = 3mn \quad (4.9)$$

$$\rho(\mathbf{Dl}) = 5m^2 + 2mn \quad (4.10)$$

$$\rho(\mathbf{Bw}) = 2m^2 + 2mn \quad (4.11)$$

$$\rho(\mathbf{Gl}) = 2mn + mr + nr \quad (4.12)$$

(4.7)~(4.12)式から、Al, Gr が有利に思える。しかし、Al は、極限操作を付録2のように代入操作に置き換えても、記号処理が必要であり、メモリ量が極端に増加する。図3に、図1の場合にならば、 $\rho(\mathcal{A})/n^2$ を $k=m/n$ の関数として表す。明らかに、Gr が有利であり、他の算法よりも大次元の行列の処理を可能とする。

以上のように、演算量とメモリ効率の両面で、行列の階数が大きく低下する場合を除き、通常の数値行列

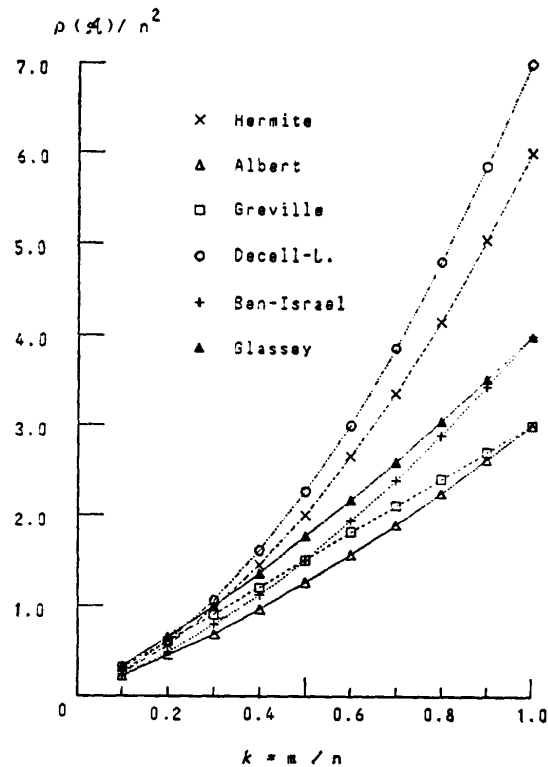


図3 最小メモリ使用量の解析結果

Fig. 3 Required storage areas which are obtained by analysis of six algorithms.

では Gr が優れているといえる。また、階数が低下する場合に有利である Gl の演算量は、階数の増加にとともに急増する。したがって、汎用アルゴリズムとしては Gr が最適であるといえる。

4.3 数式処理システムへのインプリメント

アルゴリズムの数式処理システム REDUCE, MACSYMA 上への実現に関し簡単に述べる。

Hr: (3.3)式を得るための行列基本操作 (掃き出し操作) と、(3.1)式での行列の転置および乗算とからなる。特に、MACSYMA には専用の命令があり、実現はより容易となる。

Al: (3.6)式を得るための、行列の転置、加算、乗算、逆行列の操作と極限操作からなる。Frawley⁵⁾ は、極限操作を用いて MACSYMA 上に実現している。しかし、変数 x を要素に含む行列の演算と $x \rightarrow 0$ の極限操作は、実行速度の低下とメモリ使用量の増加を招く。ここでは4.1節のように極限操作を $x=0$ の代入操作に置き換えた。いずれの場合でも個々の行列要素への操作が不用であり、行列の加算、乗算、逆転等のシステムの内部命令が使用可能である。

Gr: 反復的に行列を1列ずつ取り込み一般逆行列を求める。各段階は以下の操作で構成される。

- (1) 行ベクトル a_i への分解
- (2) (3.7)式によるベクトルの1列ずつの取り込み操作と部分行列の作成
- (3) この段階での、(3.11), (3.12)式におけるベクトルの **0** 判定

(4) (3.8)式による部分行列の一般逆行列の計算問題はベクトルの **0** 判定とベクトルの1列ずつの取り込み操作にある。これらを次のように処理した。

ベクトルの **0** 判定 REDUCE, MACSYMA とともに、内部命令を持たない。そこで、ベクトルのユークリッドノルムを求め判定した。いま、 V を $m \times 1$ ベクトル、 V^T をその転置ベクトルとすると、 V のユークリッドノルム $= \sqrt{V^T V}$ である。

行列の1行ずつの取り込み操作 MACSYMA では内部命令の使用が可能であるが、REDUCE ではこれを別に作成する必要がある。このため、作業領域が必要である。

Dl: (3.15), (3.16)式の計算と(3.17)式での **0** 判定が要点である。 **0** 判定以外は行列の加算、乗算、転置および跡の計算であり、 **A1** と同様、システムの内部命令を多用し得る。

Bw: (3.19)式の計算のために、行列の LU 分解、乗算、転置および逆行列の計算が必要である。LU 分解も **Hr** と同様、行基本操作が中心となる。以後の(3.18)式の計算は容易である。

G1: (3.21)式で、Gram-Schmidt の直交化法を用いる。ここで行列の加算、乗算以外に直交基底を決定する段階での **0** 判定が必要である。(3.20)式で、行列の乗算、転置および逆行列の計算を行い、一般逆行列を得る。

5. 各アルゴリズムの実行例

上で述べた各アルゴリズムを、具体的な数値行列に適用した場合の実行例を示す。 $A=(a_{ij})$ として、悪条件行列である Givens 行列 ($n \times n$ 行列)

$$a_{ij} = 2 * \min(i, j) - 1, \quad i, j = 1, \dots, n, \quad (5.1)$$

Givens 行列を $n \times 2n$ に拡張した長方形列

$$a_{ij} = 2 * \min(i, j) - 1, \quad i = 1, \dots, n, j = 1, \dots, 2n, \quad (5.2)$$

および階数が2に低下する $n \times n$ 正方行列

$$a_{ij} = i + j - 1, \quad i, j = 1, \dots, n, \quad (5.3)$$

の3種を考える。(5.1)~(5.3)式に対する数式処理システム MACSYMA (DEC-MICRO/VAX II 上で稼働)の計算時間を、計算可能であった行列に対してのみ図4~6に示す。

メモリの制約の中で、計算可能であった行列の次数

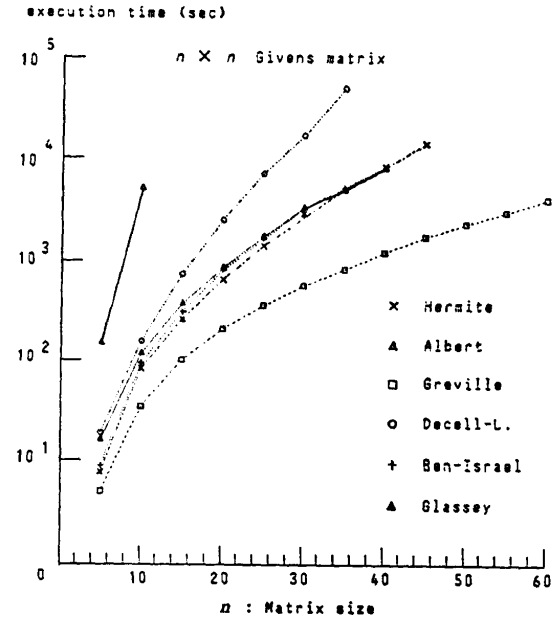


図4 一般逆行列の計算時間 ($n \times n$ Givens 行列)
Fig. 4 The computation time of the generalized inverse for $n \times n$ Givens matrix.

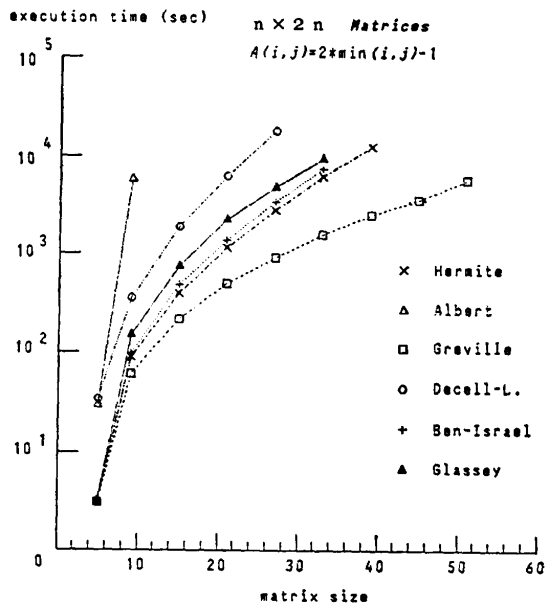


図5 一般逆行列の計算時間 ($n \times 2n$ Givens 行列)
Fig. 5 The computation time of the generalized inverse for $n \times 2n$ rectangular matrix.

は Givens 行列 (図4) では, **Hr** で 45 次, **Al** で 10 次, **Gr** で 60 次, **DI** で 36 次, **Bw** で 30 次, および **G1** で 40 次であった. 計算可能な行列の次数は **Gr** で最大であり, かつ **Gr** の実行時間が最小であることは 4 章での演算量とメモリ効率の評価をそのまま反映している. 長方形行列である (5.2) 式に対する結果は図5に示されている. この場合も, **Gr** による結果が計算時間, メモリ効率両面で最良である. 正方形行列であるが階数が大きく低下する (5.3) 式の場合は, 計算量が階数に強く依存する **Hr**, **DI**, **Bw**, **G1** 等が有利になるであろうと予想される. 図6の結果はそれを裏付ける. しかし, **DI**, **G1** は階数の増大とともに急激に不利になる. **Hr**, **Bw** は, このように極端に階数が低下する場合には, 計算時間の面で **Gr** より有利である. これは, 数式処理システム (MACSYMA) が行列演算用の内部命令を備えていることによる.

すべての場合, 文献2) で示された **Al**

は, 記号処理の必要性のため, たとえ極限操作を $x=0$ の代入操作で置き換えても, 計算時間とメモリ効率の両面から有力とはいえない. ただし, **Al** や **DI** では行列演算用の内部命令の使用が多いので, 行列要素に変数や数式を含む記号行列では相対的に有利になる. なお, メモリ容量 640KB のパソコン上で稼働している REDUCE を用いると, メモリ容量の制限により, 計算可能な行列の次数が小となる以外に上の結果とあまり差はない. この場合にも, **Gr** のアルゴリズムを用いて一般逆行列の計算を行うと, 数式処理としては十分に大きい 40 次の行列も扱うことが可能である.

以上の計算例より, 第4章の評価のごとく, Moore-Penrose 型一般逆行列を数式处理的に求める汎用的アルゴリズムとしては, **Greville** のアルゴリズム (**Gr**) が優れていることがわかる.

6. 結 び

本論では, Moore-Penrose 型一般逆行列を求める直接解法のいくつかを数式処理システムにインプリメントし, それらを演算量, メモリ効率の観点から比較した. これらにより次のような知見が得られた.

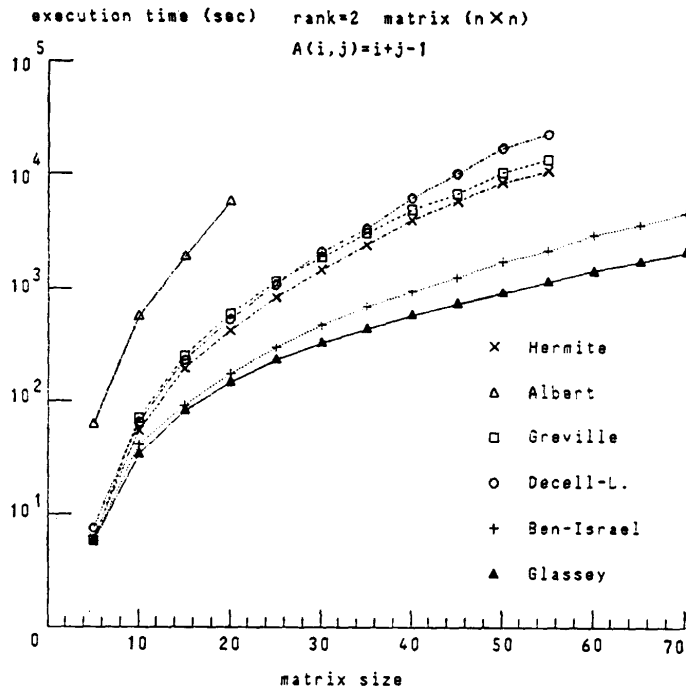


図6 一般逆行列の計算時間 ($n \times n$, 階数=2 の行列)

Fig. 6 The computation time of the generalized inverse for rank=2 square matrix.

1) 一般逆行列の直接解法を数式処理システムを用いて求める場合, 各アルゴリズムの実行に要する実行時間, メモリ効率 (必要量) の理論的な解析を行った. これらの解析結果は実際の計算例とよく一致している.

2) 比較の結果, 一般逆行列の汎用的な直接解法として, 演算量とメモリ効率の両面から, **Greville** のアルゴリズムが数式処理システムには最適である.

3) 一般逆行列を数式处理的に求めるものとして, **Frawley** により発表されている **Albert** の極限形式は, 記号処理の必要性のため効率的なアルゴリズムとはいえない.

また, 当然のことではあるが,

4) 一般逆行列の計算を数式処理システム上で逐行することにより, 数値計算で問題だった浮動小数点演算にともなう誤差を考慮する必要はなく, さらに, 記号行列の一般逆行列を求めることも可能となる.

以下の諸点は本論の結果に基づき, 今後解決されるべき課題である.

1) 数式処理システムを用いることにより, 行列要素が数値である数値行列の一般逆行列のみでなく, 変数や数式を含む記号行列の一般逆行列の計算も可能に

なる。記号行列の場合および数値行列であっても計算過程で数値が非常に大きくなる（数値の膨張が起こる）場合のアルゴリズムの良否は、本論の議論以外に、記号の長さ、種類、膨張する数値の大きさ、あるいは使用する数式処理システムのデータ構造等を、本論で求めた $\phi(\mathcal{A})$ や $\rho(\mathcal{A})$ に加える必要がある。ただし、変数の数が複数でも、次数があまり大でない記号行列の計算ではアルゴリズムの良否の順に変化はなかった。

2) アルゴリズムの数式処理システムへのインプリメントをより有効にするため、ベクトルや行列の $\mathbf{0}$ 判定の命令の数式処理システムへの付加が望まれる。これにより、Greville のアルゴリズムはさらに有効になると思われる。また、Greville, Glassey のアルゴリズム等は、厳密な $\mathbf{0}$ 判定が必要なため数値計算には適さないが、数式処理には優れた性質を有することがわかる。

3) 現在までの一般逆行列の解法研究は対象を数値計算に限っていたが、今後は誤差に対する配慮が不用の数式処理を用いたより有効なアルゴリズム開発に目を向ける必要がある。

謝辞 著者達は丁寧に本論についてご検討いただき、有益な助言をいただいた理化学研究所佐々木建昭博士に深く感謝する。また本研究の遂行にあたり、終始ご激励いただいた愛媛大学工学部相原恒博教授に感謝する。

参考文献

- 1) Shinozaki, N., Sibuya, M. and Tanabe, K.: Numerical Algorithm for the Moore-Penrose Inverse of a Matrix: Direct Methods, *Ann. Inst. Stat. Math.*, Vol. 24, No. 1, pp. 193-203 (1972).
- 2) Shinozaki, N., Sibuya, M. and Tanabe, K.: Numerical Algorithm for the Moore-Penrose Inverse of a Matrix: Iterative Methods, *Ann. Inst. Stat. Math.*, Vol. 24, No. 3, pp. 621-629 (1972).
- 3) Gregory, R. T. and Krishnamurthy, E. V.: *Methods and Applications of Error-Free Computation*, Springer-Verlag (1984).
- 4) Rao, T. M., Subramanian, K. and Krishnamurthy, E. V.: Residue Arithmetic Algorithms for Exact Computation of g-inverses of Matrices, *SIAM J. Numer. Anal.*, Vol. 13, No. 2, pp. 155-171 (1976).
- 5) Frawley, W. J.: Computer Generation of Symbolic Generalized Inverses and Applica-

- tions to Physics and Data Analysis, in *Applications of Computer Algebra* (Pavelle, R. ed.), pp. 415-426, Kluwer Academic Pub. (1985).
- 6) Albert, A.: *Regression and the Moore-Penrose Pseudoinverse*, Academic Press (1972).
 - 7) Greville, T. N. E.: Some Applications of the Pseudoinverse of a Matrix, *SIAM Review*, Vol. 2, No. 1, pp. 15-22 (1960).
 - 8) Decell, H. P., Jr.: An Application of the Cayley-Hamilton Theorem to Generalized Matrix Inversion, *SIAM Review*, Vol. 7, No. 4, pp. 526-528 (1965).
 - 9) Noble, B.: Methods for Computing the Moore-Penrose Generalized Inverse and Related Matters, in *Generalized Inverse and Applications* (Nashed, M. Z. ed.), pp. 245-301, Academic Press (1973).
 - 10) Ben-Israel, A. and Wersan, S. J.: An Elimination Method for Computing the Generalized Inverse of an Arbitrary Complex Matrix, *J. ACM*, Vol. 10, pp. 532-537 (1963).
 - 11) Glassey, C. R.: An Orthogonalization Method of Computing the Generalized Inverse of Matrix, ORC-66-10, Operations Research Center, Univ. of California, Berkeley (1966).

付録 1

Greville のアルゴリズムに対する演算量とメモリ効率の導出を行う。演算量はアルゴリズムの順を追って、メモリ効率は必要な行列の個数と次元を求めればよい。

演算量

- ① A^+ の初期値の設定 (3.12) 式 $\dots O(n)$
- ② $A^+_{:i}$ の計算 $\dots i=1, \dots, m$ に対し, (3.8)~(3.11) 式を繰り返す。
 - $d_i \dots$ 乗算: $n(i-1)$, 加算: $(n-1)(i-1)$
平均 ni 回
 - $c_i \dots$ 乗算: $n(i-1)$, 加算: $n(i-1)$
平均 ni 回
 - $b_i \dots$
 $c_i \neq 0$ 乗算: $2n$, 加算: $n-1$
平均 $O(n)$ 回
 - $c_i = 0$ 乗算: $ni+i-1$, 加算: $ni+i-2n-1$
平均 ni 回
 - $A^+_{:i} \dots$ 乗算: $n(i-1)$, 加算: $n(i-1)$
平均 ni 回

よって、行列の階数を r とすると、従属行の位置によって演算量が異なる。従属行が $2 \sim m-r+1$ にあるなら、

$$\sum_{i=2}^m 3ni + \sum_{i=2}^{m-r+1} ni = 2nm^2 - mnr + nr^2/2$$

一方、従属行が $r+1 \sim n$ にあるなら、

$$\sum_{i=2}^m 3ni + \sum_{i=r+1}^m ni = 2nm^2 - nr^2/2$$

①と②を加えると、求める演算量が得られる。②では、後者が大なので、(4.3)式を得る。

メモリ量

入力行列 $A: m \times n$

作業行列 $A_i: m \times (n-1)$

A_i の一般逆行列 $A_i^+: n \times m$

作業ベクトル d_i, b_i, c_i の大きさは高々 n 。ゆえに、 $m, n \gg 1$ のとき、必要メモリ量は(4.9)式となる。

付録 2

4章で述べたように、Albert の極限形式のインプリメントの効率化のため、極限操作を $x=0$ の代入操作に置き換える。この保証を以下に示す。任意の $m \times n$ 行列 A に対し、変数 x を含む $n \times n$ 正方行列 B を

$$B = A^T(x^2I + AA^T)^{-1}$$

と定義する。これに対し、次式が示されればよい。

$$\lim_{x \rightarrow 0} B(x) = B(x=0) \quad (\text{A.1})$$

行列 B の ij 要素は

$$(B)_{ij} = \sum_{k=1}^r q_{ik} \frac{\lambda_k}{x^2 + \lambda_k^2} p_{jk} \quad (\text{A.2})$$

と表される。ただし、 λ_k は A の特異値、 p_{jk}, q_{ik} はスカラー量、 r は A の階数とする。(A.2)式は次のように証明される。

A の特異値分解を

$$A = \sum_{k=1}^r p_i \lambda_i q_i^T$$

とする。また $n-r$ 個の適当なベクトルにより

$$P = (p_1, p_2, \dots, p_n) = \{p_{ij}\}$$

$$Q = (q_1, q_2, \dots, q_n) = \{q_{ij}\}$$

が

$$p_i^T p_j = \delta_{ij}, \quad q_i^T q_j = \delta_{ij}$$

を満たすようにする。ただし、 δ_{ij} は Kronecker の δ 記号である。これにより

$$B = \sum_{i=1}^r q_i \lambda_i p_i^T \left(\sum_{i=1}^n x^2 p_i p_i^T + \sum_{i=1}^r p_i \lambda_i^2 p_i^T \right)^{-1}$$

$$\begin{aligned} &= \sum_{i=1}^r q_i \lambda_i p_i^T \left(\sum_{i=n-r}^n x^2 p_i p_i^T + \sum_{i=1}^r \right. \\ &\quad \left. (x^2 + \lambda_i^2) p_i p_i^T \right)^{-1} \\ &= \sum_{i=1}^r q_i \lambda_i p_i^T \left(\sum_{i=n-r}^n \frac{1}{x^2} p_i p_i^T + \sum_{i=1}^r \frac{1}{x^2 + \lambda_i^2} \right. \\ &\quad \left. p_i p_i^T \right) = \sum_{i=1}^r q_i \lambda_i (x^2 + \lambda_i^2) p_i^T \end{aligned}$$

すなわち、(A.2)式が示される。

これより、明らかに行列

$$B(x) = A^T(x^2I + AA^T)^{-1}$$

に対して(A.1)式が成立する。すなわち、極限操作を $x=0$ の代入操作に置き換えてよいことがわかる。

(平成元年2月21日受付)

(平成元年9月12日採録)



野田松太郎 (正会員)

昭和44年大阪市立大学理学研究科博士課程修了。理学博士。日本学術振興会奨励研究員を経て愛媛大学工学部情報工学科に勤務。現在、同助教授。数値計算と数式処理の結合と「AI化」およびそれに基づく新しいアルゴリズムの開発に強い関心をもっている。ACM, ソフトウェア科学会, 人工知能学会, 日本物理学会各会員。



泉田 正則 (正会員)

昭和57年愛媛大学工学部電子工学科卒業。昭和59~昭和60年米カリフォルニア大学留学。昭和61年愛媛大学大学院修士課程修了。現在同大学工学部助手。連想記憶, 神経回路の研究に従事。



越智 正明

1964年生。1987年愛媛大学工学部電子工学科卒業。1989年同大学工学研究科電子工学専攻(修士課程)修了。同年、松下電器産業(株)に入社。現在、ワークステーションのハードウェア設計に従事。数式処理, コンピュータネットワーク等に興味をもつ。