行列の最小多項式候補と拡張Horner法を用いた逆行列計算について II

Calculating matrix inverse by the extended Horner's rule with pseudo minimal polynomial II

田島 慎一 *

Shinichi Tajima

筑波大学 数理物質系

Faculty of Pure and Applied Sciences, University of Tsukuba

小原 功任 †

Katsuyoshi Ohara

金沢大学 理工研究域

Faculty of Mathematics and Physics, Kanazawa University

照井 章 ‡

Akira Terui

筑波大学 数理物質系

Faculty of Pure and Applied Sciences, University of Tsukuba

Abstract

本稿では、整数を成分にもつ正方行列の逆行列の計算を取り上げる．逆行列計算には様々な方法が知られているが，ここでは，行列の特性多項式もしくは最小多項式の変数に行列を代入する方法に着目する．我々はこれまで，行列のレゾルベントの留数解析に基づき，行列の（一般）固有空間やスペクトル分解等の効率的算法を提案してきたが，その中で，行列の最小消去多項式の算法，ならびに，1 変数多項式の値を評価する Horner 法に対し，変数に行列を代入する「行列 Horner 法」を効率化する「拡張 Horner 法」を提案した．本稿では，これまでの研究成果に基づき，行列の最小消去多項式候補から最小多項式候補を求め，拡張 Horner 法を用いる逆行列計算法を提案し，計算例により，その効果を示す．

For a given matrix with integers, we consider calculating its matrix inverse. From many algorithms for that task, we pick up one by putting the matrix into the variable in its characteristic or the minimal polynomial. Based on analysis of the residues of the resolvent, we have proposed efficient algorithms for matrices such as spectral decomposition and calculating (generalized) eigenspaces, etc. Among them, we have proposed an algorithm for calculating minimal annihilating polynomials and an extended Horner's rule for efficient evaluation of univariate polynomial with a matrix. With this result, we propose an algorithm for calculating matrix inverse by evaluating pseudo minimal polynomial via pseudo annihilating polynomial of the matrix using the extended Horner's rule, and show its effects by examples.

*tajima@math.tsukuba.ac.jp

†ohara@air.s.kanazawa-u.ac.jp

‡terui@math.tsukuba.ac.jp

1　はじめに

本稿では，整数や有理数などを成分にもち，各成分に対して厳密計算を基本とする行列計算において，行列の特性多項式や最小多項式候補と拡張 Horner 法を用いた逆行列の計算法について論ずる．厳密な線型計算における逆行列の計算は，たとえば連立 1 次方程式を解く際，特定の解のみを求めるなど，有用な場合が考えられる．

我々はこれまで，行列のレゾルベントの留数解析に基づき，行列に対する厳密な数値計算を効率的に行う算法を提案してきた．それらには，行列の固有空間や一般固有空間の計算 [9]，スペクトル分解 [8]，固有ベクトル計算 [12] が含まれる．また，これらの算法の研究においては，行列の最小消去多項式やその候補の計算法 [13]，1 変数多項式の変数に行列を代入して評価するための拡張 Horner 法 ([6], [10]) などを用いることにより，算法の効率化を図ってきた．

本稿では，これまでの成果に基づき，行列の最小多項式候補と拡張 Horner 法を用いた逆行列の計算法を提案する．行列の特性多項式から逆行列を計算する方法は，大学初年度の線形代数の授業における演習問題としてもよく取り上げられるが，計算量の面においては，通常よく用いられる分数なし Gauss 消去法 [1] に劣るという見方が一般的であろう．そのような中で，我々は，以下の道具を用いて，特性多項式（もしくは最小多項式）を用いた逆行列計算の効率化を図る．

一つは，行列の最小多項式候補である．最小多項式は一般に特性多項式と比べて多項式としての次数がより小さく，算法の効率化には有利であるが，一般に最小多項式を求めるのは容易ではない．しかし，我々がこれまでの研究で提案した最小多項式候補を用いることにより，これまでよりも効率的に最小多項式を求め，逆行列計算の効率化を図る．

もう一つは，1 変数多項式を行列で評価する際に用いる拡張 Horner 法である．拡張 Horner 法は，特に行列どうしの乗算の回数を抑えることにより，Horner 法を効率化する方法である．さらに，拡張 Horner 法に並列処理 [11] を用いることにより，算法のより一層の効率化を目指す．

以下，本稿では次の内容を述べる．第 2 章では，行列の特性多項式から逆行列を計算する方法と，その計算量について復習する．第 3 章では．拡張 Horner 法およびその並列処理による実装と実験結果について述べる．第 4 章では．行列の最小多項式候補を用いた逆行列の計算法を述べる．

2　行列の特性多項式を用いた逆行列の計算

A を整数を成分にもつ n 次正方行列とする．A の特性多項式が

$$\chi_A(\lambda) = f_1(\lambda)^{m_1} f_2(\lambda)^{m_2} \cdots f_q(\lambda)^{m_q}, \tag{1}$$

A の最小多項式が

$$\pi_A(\lambda) = f_1(\lambda)^{l_1} f_2(\lambda)^{l_2} \cdots f_q(\lambda)^{l_q} \tag{2}$$

でそれぞれ与えられているものとする．このとき，$i = 1, \ldots, q$ に対し，$1 \leq l_i \leq m_i$ であることに注意する．

本稿では，上述の行列 A に対し，有理数を成分とする A の逆行列 A^{-1} の計算を考える．ここでは，A の特性多項式 $\chi_A(\lambda)$ を用いて A^{-1} の計算を行う．A の特性多項式 $\chi_A(\lambda)$ が

$$\begin{aligned}
\chi_A(\lambda) &= \lambda^n + a_{n-1}\lambda^{n-1} + a_{n-2}\lambda^{n-2} + \cdots + a_1\lambda + a_0,\ a_0 \neq 0 \\
&= \lambda(\lambda^{n-1} + a_{n-1}\lambda^{n-2} + \cdots + a_1 E) + a_0 E, \\
&= \lambda\, h(\lambda) + a_0 E
\end{aligned}$$

（ここに $h(\lambda) = \lambda^{n-1} + a_{n-1}\lambda^{n-2} + \cdots + a_1 E$）と表されるとする．ハミルトン-ケーリーの定理により，$\chi_A(A) = A\,h(A) + a_0 E = O$ が成り立つ．これより $a_0 E = -A\,h(A)$ が成り立つので，

$$A^{-1} = -\frac{1}{a_0} h(A) \tag{3}$$

が成り立つ．

$h(A)$ は n 次正方行列であるので，その (i,j) 成分を h_{ij} で表す．式 (3) の A^{-1} の各成分は一般に互いに素とは限らない．そこで，$d = \gcd(a_0, h_{11}, \ldots, h_{nn})$ とおき，$d \neq 1$ の場合には $A^{-1} = -\frac{1}{\frac{a_0}{d}} \left(\frac{1}{d} h(A) \right)$ により A^{-1} を求める．

これにより，A^{-1} の計算は $h(A)$ の計算，すなわち，1 変数多項式 $h(\lambda)$ の $\lambda = A$ による評価に帰着される．1 変数多項式の評価の算法としては Horner 法 [4] がよく知られている．Horner 法の時間計算量は，係数の四則演算を単位時間とするとき，多項式の次数 n に対し $O(n)$ であることが知られている．よって，$h(A)$ の時間計算量は，$O(n)$ 回の行列-行列積の計算量となる．ここで，n 次正方行列の行列-行列積の時間計算量は，素朴な行列乗算を用いた場合，成分の四則演算を単位時間とすると $O(n^3)$ となる．したがって，A の成分の四則演算を単位時間とした場合の $h(A)$ の時間計算量は $O(n^4)$ となる．

3　拡張 Horner 法およびその並列処理による実装

拡張 Horner 法 [10] は，Horner 法の拡張の一つで，与えられた次数ごとに Horner 法を分割して計算することにより，係数の乗算回数を減らすというものである．特に，本稿のように係数として行列を扱う場合には，行列の乗算回数を減らすことにより，前章で述べた $h(A)$ の計算の効率化が図られる．拡張 Horner 法による時間計算量の最適値は，n 次多項式に対し，分割次数 $d = 2^b$ として $\sqrt{n}$ になるべく近い値をとることにより，行列 A の乗算回数を単位として $O(\sqrt{n})$ になる．よって，A の成分の四則演算を単位時間とする時間計算量は $O(n^3\sqrt{n})$ となる．

我々は，拡張 Horner 法の並列化を実装する環境として，数式処理システム Risa/Asir および Risa/Asir 用の並列計算フレームワーク `oh_p` を用いてきた [11] ので，本稿においてもそれに従う．`oh_p` は以下の特徴をもつ（拙稿 [11] より引用．詳細は小原 [7] を参照）．

1. OpenXM プロトコル上に Asir （言語）で実装された Asir （言語）用のソフトウェアパッケージである．(OpenXM は，同一もしくは異なる計算機上のプロセス間で，数式処理をはじめとする数学情報を扱い，情報のやりとりや計算の制御を行うための手順（プロトコル）である．)

2. OpenXM がプロセス単位での並列計算に対応することから，`oh_p` もプロセス単位での並列計算を行う．

3. `oh_p` における並列計算の単位は，1 個の client と l 個の server により構成される．

4. 並列計算の際は，client からジョブの集合の要素を各 server に投入して計算させ，計算結果を client に集める．ジョブの各要素間の依存性にも対応した処理が可能である．

並列化の対象としては，n 次実正方行列 A, B に対し，積 AB の計算を，以下の要領にて並列化する（拙稿 [11] より引用）．A を行ブロックに分割する．n を l で割った商を q, 剰余を r とするとき

$$A = {}^t\left(\begin{array}{cccc} A_1 & A_2 & \cdots & A_{l-1} \quad A_l \end{array} \right), \quad A_j \in \begin{cases} \mathbb{R}^{n\times q} & (j = 1, \ldots, l-1) \\ \mathbb{R}^{n\times r} & (j = l) \end{cases}$$

（すなわち，A を l 行毎の行ブロックに分割し，最後のブロックを A の下から r 行からなるブロックとする）とし，

$$AB = {}^t\left(\begin{array}{cccc} {}^tBA_1 & {}^tBA_2 & \cdots & {}^tBA_{l-1} \quad {}^tBA_l \end{array} \right)$$

により AB を求める．${}^tBA_j$ の計算を各 server に割り振ることで並列化する．A のブロック分割数が server の個数を上回るときは，${}^tBA_1$ から順次（計算が終わって）空いている server に計算を割り振る．

3.1 実験

我々は，特性多項式を用いた上記の逆行列の算法を数式処理システム Risa/Asir に実装した．特性多項式の計算には木村欣司氏による実装 [2] を，Risa/Asir における並列計算には上述の `oh_p` パッケージをそれぞれ用いた．そして，Gauss 消去法の実装との動作比較として，Risa/Asir 組み込み関数の `invmat`, および数式処理システム Maple 2016 [5] の線形代数パッケージに含まれる組み込み関数の `LinearAlgebra[MatrixInverse]` との動作比較を行った．

実験環境は以下の通り: Intel Xeon E5-2690 at 2.90 GHz (8 cores) × 4, RAM 128GB, Linux 3.2.0 (SMP).

本稿の実験は，以下の要領で行った．

1. 行列 $A = (a_{ij})$ は，各成分 a_{ij} を $-10 \leq a_{ij} \leq 10$ をみたす整数で無作為に与えた．
2. 実験に用いた並列計算の並列度は，1) 逐次計算の場合と 2) 16 並列の場合の 2 つで比較した．
3. A の次元は 32, 64, 128, 256, 384, 512, 640, 768, 896, 1024 とした．
4. 計算時間は，Risa/Asir および Maple の出力を用いており，表示される計算時間の精度は一般的にそれぞれ 10 進 6 桁および 10 進小数点以下 3 桁である．

3.1.1 逐次計算による Gauss 消去法との比較

まず，逐次計算により，拡張 Horner 法を Gauss 消去法と比較した際の逆行列計算の実験結果を表 1 に示す．いずれの実装においても，時間計算量は $\dim(A)^4$ から $\dim(A)^5$ 程度に比例しているように見える（本稿における逆行列計算の計算量は，$\dim(A)$ に加え，各算法の計算過程で扱う行列の成分の大きさ（多倍長数の長さ）にも依存する点に注意）．そして，$\dim(A)$ のそれぞれの値に対し，拡張 Horner 法の計算時間は，Maple (`LinearAlgebra[MatrixInverse]`) のそれの 2 倍から 3 倍程度，Risa/Asir (`invmat`) のそれに対しては 5 倍から 6 倍前後である．逐次計算においては，拡張 Horner 法の効率は Gauss 消去法のそれに劣ることがわかる．

ただし，表 1 の結果は，もし，A の最小多項式があらかじめ何らかの方法でわかっており，その次数が A の特性多項式の次数よりも十分小さいときには，拡張 Horner 法による逆行列の計算が分数なし Gauss 消去法に比べてより効率的になる場合があり得ることも示している．たとえば，$\dim(A) = 640$ のときに，A の最小多項式の次数が 384 であることがあらかじめわかっている場合，最小多項式に対して拡張 Horner 法を適用することにより，分数なし Gauss の消去法に比べてより効率的に逆行列を計算することが可能になる．

表 1 の拡張 Horner 法において，Horner 法の部分と，それ以外の部分の計算時間の内訳を，表 2 に示す．これが示すように，拡張 Horner 法による逆行列計算においては，A の次元が大きくなるほど，Horner 法が総計算時間のほとんどを占めており，Horner 法の効率化が算法全体の効率化に大きく影響することがわかる．

$\dim(A)$	拡張 Horner 法	Maple	Risa/Asir (`invmat`)
32	0.204657	0.215	0.0447891
64	0.990572	1.962	0.254013
128	13.5001	25.288	2.53097
256	312.951	371.647	63.6169
384	2303.69	1938.238	470.368
512	9615.96	6507.11	1966.36
640	30224.3	17010.724	5845.03
768	77593.1	37281.873	16239.7
896	183790	72901.564	38168.1
1024	362858	132249.564	71106.7

表 1: 逐次計算の場合の拡張 Horner 法および Gauss 消去法による逆行列計算の計算時間の比較．単位は秒．詳細は第 3.1.1 節を参照．

$\dim(A)$	Horner 法	それ以外
32	0.0754962	0.1291608
64	0.439396	0.551176
128	9.96478	3.53532
256	219.075	93.876
384	1881.19	422.50
512	8472.71	1143.25
640	27567.3	2657.0
768	71984.3	5608.80
896	172067	11723.0
1024	341143	21715.0

表 2: 逐次計算の場合の拡張 Horner 法（表 1）における Horner 法とそれ以外の部分の計算時間の内訳．単位は秒．詳細は第 3.1.1 節を参照．

3.1.2 並列計算による分数なし Gauss 消去法との比較

次に，並列計算により，拡張 Horner 法を Gauss 消去法と比較した際の実験結果を表 3 に示す [1)]．16 並列による拡張 Horner 法は，逐次計算の Gauss 消去法と比較して効率が向上していることがわかる [2)]．

dim(A)	拡張 Horner 法	Maple	Risa/Asir (`invmat`)
32	0.415794	0.215	0.0447891
64	1.10875	1.962	0.254013
128	11.0809	25.288	2.53097
256	57.6576	371.647	63.6169
384	252.994	1938.238	470.368
512	1001.42	6507.11	1966.36
640	2736.18	17010.724	5845.03
768	6342.45	37281.873	16239.7
896	13074.3	72901.564	38168.1
1024	24895.0	132249.564	71106.7

表 3: 並列計算による拡張 Horner 法および逐次計算による分数なし Gauss 消去法（表 1）による逆行列計算の計算時間の比較．並列度は 16，単位は秒．詳細は第 3.1.2 節を参照．

拡張 Horner 法による逆行列計算について，逐次計算および 16 並列の並列計算を比較した結果を表 4 に示す．逐次計算の場合の時間計算量は $\dim(A)^5$ 程度に比例する一方，並列計算の場合の時間計算量は $\dim(A)^4$ 程度に比例していることがわかる．

dim(A)	逐次計算	並列計算
32	0.204657	0.415794
64	0.990572	1.10875
128	13.5001	11.0809
256	312.951	57.6576
384	2303.69	252.994
512	9615.96	1001.42
640	30224.3	2736.18
768	77593.1	6342.45
896	183790	13074.3
1024	362858	24895.0

表 4: 拡張 Horner 法による逆行列計算の，逐次計算（表 1）と並列計算（表 3）による計算時間の比較．並列度は 16，単位は秒．詳細は第 3.1.2 節を参照．

[1)] 本稿においては，Gauss 消去法（Maple および Risa/Asir (`invmat`)）の並列化は特に行っていないので，それらの計算時間のデータは表 1 のものと同一である．

[2)] Gauss 消去法の並列化については第 5 章を参照．

4　行列の最小多項式候補を用いた逆行列の計算

次に，行列の最小多項式を用いて，第 2 章と同様の要領で逆行列の計算を行うことを考える．A の最小多項式 $\pi_A(\lambda)$ は式 (2) で与えられているが，$\pi_A(\lambda)$ が

$$\begin{aligned}\pi_A(\lambda) &= \lambda^d + b_{d-1}\lambda^{d-1} + b_{d-2}\lambda^{d-2} + \cdots + b_1\lambda + b_0, b_0 \neq 0 \\ &= \lambda(\lambda^{d-1} + b_{d-1}\lambda^{d-2} + \cdots + b_1 E) + b_0 E \\ &= \lambda\, g(\lambda) + b_0 E\end{aligned}$$

（ここに $g(\lambda) = \lambda^{n-1} + b_{n-1}\lambda^{n-2} + \cdots + b_1 E$）と表されるとする．第 2 章と同様に，$\pi_A(A) = A\, g(A) + b_0 E = O$ が成り立つので，$b_0 E = -A\, g(A)$ から

$$A^{-1} = -\frac{1}{b_0} g(A) \tag{4}$$

を得る．式 (4) の A^{-1} の各成分は一般に互いに素とは限らない．そこで，$g(A) = (g_{ij})$ で表すとき，$d = \gcd(b_0, g_{11}, \ldots, g_{nn})$ とおき，$d \neq 1$ の場合には $-\frac{1}{\frac{b_0}{d}} \left(\frac{1}{d}\, g(A) \right)$ により A^{-1} を求める．

行列の最小多項式を用いた逆行列の計算 (4) は，特性多項式を用いた逆行列の計算 (3) に比べて，いくつかの利点がある．第一に，$\deg(g) = d - 1 \leq \deg(h) = n - 1$ より，$g(A)$ はより効率的に計算できる可能性がある．第二に，b_0 が a_0 の因子である場合があり，この場合，GCD の計算による b_0 の計算の方がより効率的である可能性がある．

しかしながら，行列の最小多項式の計算は，特性多項式の計算と比べて一般に容易とは言えない．そのため，行列の最小多項式を用いた逆行列の計算の効果を出すためには，最小多項式をなるべく効率的に計算することが必要となる．我々は，これまでの研究で，最小多項式候補の効率的な算法を提案した [9]．そこで，本章では，最小多項式候補を用いた逆行列の計算を提案する．

まず，最小多項式候補を用いた逆行列計算のアルゴリズムを以下に示す．

アルゴリズム 1 (最小多項式候補を用いた逆行列の計算)
入力: A: 整数上の n 次正方行列;

出力: A^{-1}: A の逆行列;

1. $\pi'_A(\lambda) \leftarrow A$ の最小多項式候補;
2. if $\pi'_A(A) = O$ then
   (a) $\pi'_A(A)$ を用いて A^{-1} を求める;
   else
   (a) それまでの計算結果を用いて，A の真の最小多項式 $\pi_A(A)$ を求め，そこから A^{-1} を求める;
3. return A^{-1}; ∎

A の最小多項式候補を $\pi'_A(\lambda)$ とし，

$$\pi'_A(\lambda) = \lambda g'(\lambda) + b'_0, \quad \deg(g') = \deg(\pi') - 1, \quad b'_0 \neq 0$$

とおく．このとき，$\pi'_A(A) = O$ であれば，$\pi'_A(\lambda)$ は最小多項式 $\pi_A(\lambda)$ に等しいか，$\pi_A(\lambda)$ を因子にもつ．このときに逆行列を計算するアルゴリズムを以下に示す．

アルゴリズム 2 (最小多項式候補の検証と逆行列の計算)
入力: A: 整数上の n 次正方行列, $\pi'_A(\lambda) = \lambda g'(\lambda) + b'_0$: A の最小多項式候補

出力: A^{-1}: （もし計算できれば）A の逆行列;

1. $g'(A)$ を拡張 Horner 法によって計算する;
2. $\pi'_A(A) \leftarrow A\,g'(A) + b'_0E$;
3. if $\pi'_A(A) = A\,g'(A) + b'_0E = O$ then
   (a) $d \leftarrow \gcd(b_0, g'_{11}, \ldots, g'_{nn})$ for $g'(A) = (g_{ij})$;
   (b) return $-\frac{1}{\frac{b_0}{d}}\left(\frac{1}{d}\,g'(A)\right)$; ∎

4.1 最小多項式候補が真の最小多項式に一致しない場合の逆行列の計算

次に，$\pi'_A(A) = A\,g'(A) + b'_0E \neq O$ の際に，真の最小多項式および逆行列の計算を行う方法を示す．以下では

$$\pi_A(\lambda) = f_1(\lambda)^{l_1} f_2(\lambda)^{l_2} \cdots f_q(\lambda)^{l_q}, \quad \pi'_A(\lambda) = f_1(\lambda)^{l'_1} f_2(\lambda)^{l'_2} \cdots f_q(\lambda)^{l'_q} \tag{5}$$

とおく．

まず，A の最小多項式 $\pi_A(\lambda)$ を求める．$R = (\boldsymbol{r}_1, \ldots, \boldsymbol{r}_n) = \pi'_A(A) = A\,g'(A) + b'_0E$, すなわち，$R$ の第 j 列を $\boldsymbol{r}_j$ とおく．そして，$\boldsymbol{r}_{j_1}, \ldots, \boldsymbol{r}_{j_s}$ を R の非ゼロの列 $(s < n, 1 \leq j_1 < \cdots < j_s)$ とおく．

さらに，$\boldsymbol{r}_{j_k}$ $(k = 1, \ldots, s)$ に対し，$\boldsymbol{r}_{j_k}$ の最小消去多項式 [13] を

$$\pi_{A,\boldsymbol{r}_{j_k}}(\lambda) = f_1(\lambda)^{\nu_{j_k,1}} f_2(\lambda)^{\nu_{j_k,2}} \cdots f_q(\lambda)^{\nu_{j_k,q}}$$

で求める．このとき，$\pi_{A,\boldsymbol{r}_{j_k}}(\lambda)$ の因子 $f_i(\lambda)$ の指数 $\nu_{j_k,i}$ は，$\pi_{A,\boldsymbol{r}_{j_k}}(R)$ の第 j_k 列を $\mathbf{0}$ にするような $f_i(\lambda)$ の指数の最小値であることに注意する．すると，最小消去多項式の定義より

$$l_i = l'_i + \max_{k=1,\ldots,s}(\nu_{j_k,i})$$

が成り立つ（$\max_{k=1,\ldots,s}(\nu_{j_k,i})$ は，R のすべての列を $\mathbf{0}$ にするような $f_i(\lambda)$ の指数の最小値である）．$\nu'_{j_k,i} = \max_{k=1,\ldots,s}(\nu_{j_k,i})$ とおき，

$$\pi_A(\lambda) = q(\lambda)\pi'_A(\lambda), \quad q(\lambda) = f_1(\lambda)^{\nu'_{j_k,1}} f_2(\lambda)^{\nu'_{j_k,2}} \cdots f_q(\lambda)^{\nu'_{j_k,q}} \tag{6}$$

によって最小多項式 $\pi_A(\lambda)$ を求めることができる．

注意 1
実際に最小多項式 $\pi_A(\lambda)$ を求める際には，すべての $\boldsymbol{r}_{j_1}, \ldots, \boldsymbol{r}_{j_s}$ に対して，$\boldsymbol{r}_{j_k}$ の最小消去多項式 $\pi_{A,\boldsymbol{r}_{j_k}}(\lambda)$ を求める必要はない．ある $j \in \{j_1, \ldots, j_s\}$ に対して，$\boldsymbol{r}_j$ の最小消去多項式 $\pi_{A,\boldsymbol{r}_j}(\lambda)$ に $f_i(\lambda)^{\nu_{j,i}}$ が因子として現れたならば，$f_i(A)^{\nu_{j,i}}$ を R にかけ，R を更新しながら上記の手順を繰り返せばよい．

次に，これまでに求めた各式から A^{-1} を求める．式 (5), (6) より

$$\pi_A(\lambda) = \lambda g(\lambda) + b_0, \quad \pi'_A(\lambda) = \lambda g'(\lambda) + b'_0, \quad q(\lambda) = \lambda r(\lambda) + c_0 \tag{7}$$

（ただし b_0, b'_0, c_0 はそれぞれ非ゼロの定数）とおく．式 (6) に $\pi'_A(\lambda)$ を代入すると

$$\begin{aligned}\pi_A(\lambda) &= q(\lambda)\{\lambda g'(\lambda)+b'_0\} = \lambda q(\lambda)g'(\lambda)+q(\lambda)b'_0 = \lambda q(\lambda)g'(\lambda)+\{\lambda\, r(\lambda)+c_0\}b'_0 \\ &= \lambda\{q(\lambda)g'(\lambda)+r(\lambda)b'_0\}+c_0b'_0, \end{aligned} \tag{8}$$

ゆえに

$$g(\lambda) = q(\lambda)g'(\lambda)+b'_0\, r(\lambda) \tag{9}$$

を得る．これにより，$g(A) = q(A)g'(A)+b'_0\, r(A)$ から $g(A)$ を求め，これを用いて A^{-1} を求めればよい．

注意 2
逆行列を求めるために必要な $g(A) = q(A)g'(A)+b'_0\, r(A)$ の計算中，$g'(A)$ は，アルゴリズム 2 の中で計算済みであり，今回新たに計算しなければならないのは $q(A)$ および $r(A)$ であるが，式 (7) より $q(A) = A\, r(A)+c_0E$（E は n 次単位行列）であり，ひとたび $r(A)$ を（拡張）Horner 法で計算すれば，$q(A)$ はさらに 1 回分の Horner 法の反復で求まる．また，$q(\lambda)$ は，最小多項式候補 $\pi'_A(\lambda)$ から真の最小多項式 $\pi_A(\lambda)$ を求める際に構成した多項式であり，その次数は，最小多項式候補の次数や真の最小多項式の次数と比較して小さいと考えられる．ゆえに，$q(A)$ および $r(A)$ の時間計算量は，$\pi'_A(\lambda)$ のそれと比較して比較的少ないと期待される．

4.2 最小多項式（候補）を用いた逆行列の計算例

最小多項式（候補）を用いた逆行列の計算例を示す．以下では n 次の Jordan 細胞

$$\begin{pmatrix} \lambda & 1 & & & O \\ & \lambda & 1 & & \\ & & \ddots & \ddots & \\ & & & \ddots & 1 \\ O & & & & \lambda \end{pmatrix}$$

を $J_n(\lambda)$ で表す．

例 1
行列 A を

$$A = \left(\begin{array}{c|c} J_{128}(2) & O \\ \hline O & J_{128}(2) \end{array}\right)$$

で定める．このとき，$\chi_A(\lambda) = (\lambda-2)^{256}$, $\pi_A(\lambda) = (\lambda-2)^{128}$ である．$\chi_A(\lambda)$ を用いて A^{-1} を求めた時の計算時間は 4.34492 秒であった．一方，$\chi_A(\lambda)$ から $\pi_A(\lambda)$ を求めた上で A^{-1} を求めた時の計算時間は 2.86926 秒であった．$\chi_A(\lambda))$ の次数は 256 であるのに対し，$\pi_A(\lambda)$ の次数は 128 であるから，最小多項式を用いて逆行列を求める方がより効率的であることがうかがえる．

なお，本計算例において，$\chi_A(\lambda)$ を用いて A^{-1} を求めた際の計算時間が，第 3.1.1 節において，256 次元の行列に対し，特性多項式に対する拡張 Horner 法を用いた逆行列の計算時間（312.951 秒）に比べて大幅に短くなっているが，これは，本節で用いた行列 A が，第 3.1.1 節で用いた行列に比べて疎であることに起因するものと思われる．参考までに，本例における A と，第 3.1.1 節で用いた行列の，それぞれの 2 乗の計算時間を比較したところ，前者の計算時間は 0.101835 秒だったのに対し，後者の計算時間は 1.54546 秒であった．

例 2
行列 B を

$$B = \left(\begin{array}{c|c|c} J_3(2) & O & O \\ \hline O & J_2(2) & O \\ \hline O & O & J_5(3) \end{array}\right)$$

で定める．このとき，$\chi_B(\lambda) = (\lambda-2)^5(\lambda-3)^5$, $\pi_B(\lambda) = (\lambda-2)^3(\lambda-3)^5$ である．ここでは，最小多項式候補が $\pi'_B(\lambda) = (\lambda-2)^2(\lambda-3)^3$ と求まったものとし，最小多項式の計算を経て逆行列の計算を行う．$R = \pi'_B(B) = (\boldsymbol{r}_1, \ldots, \boldsymbol{r}_{10})$ を求めると，R の非零の列は以下の通りである．

$$\boldsymbol{r}_3 = {}^t(-1, 0, \cdots, 0), \quad \boldsymbol{r}_9 = {}^t(0,0,0,0,0,1,0,0,0,0), \quad \boldsymbol{r}_{10} = {}^t(0,0,0,0,0,2,1,0,0,0).$$

そこで，$\boldsymbol{r}_3$, $\boldsymbol{r}_9$, $\boldsymbol{r}_{10}$ に対し，B の最小消去多項式を求めると，

$$\pi_{B,\boldsymbol{r}_3}(\lambda) = \lambda - 2, \quad \pi_{B,\boldsymbol{r}_9}(\lambda) = \lambda - 3, \quad \pi_{B,\boldsymbol{r}_{10}}(\lambda) = (\lambda-3)^2$$

と求まる．これより，多項式 $q(\lambda)$ として，$q(A)\pi'_B(A)$ を $\boldsymbol{r}_3$, $\boldsymbol{r}_9$, $\boldsymbol{r}_{10}$ の左側からかけてすべて $\mathbf{0}$ にするようなもので次数最小のものを求めると，$q(\lambda) = (\lambda-2)(\lambda-3)^2$ となる．これにより $\pi_B(\lambda) = q(\lambda)\pi'_B(A)$ が求まる．

次に，式 (9) に従い，$g(A)$ を求める．これまでの計算で，

$$\begin{aligned} q(\lambda) &= (\lambda-2)(\lambda-3)^2 = \lambda^3 - 8\lambda^2 + 21\lambda - 18, \quad r(\lambda) = \lambda^2 - 8\lambda + 21, \\ g'(\lambda) &= \lambda^4 - 13\lambda^3 + 67\lambda^2 - 171\lambda + 216, \\ \pi'_A(\lambda) &= (\lambda-2)^2(\lambda-3)^3 = \lambda(\lambda^4 - 13\lambda^3 + 67\lambda^2 - 171\lambda + 216) + (-108) \end{aligned}$$

より $b'_0 = -108$, よって

$$g(\lambda) = \lambda^7 - 21\lambda^6 + 192\lambda^5 - 998\lambda^4 + 3225\lambda^3 - 6633\lambda^2 + 8478\lambda - 6156$$

となる．実際の $g(A)$ の計算は，すでに求めていた $q(A)$, $r(A)$, $g'(A)$ および $r(A)$ から，1 回の行列-行列積，1 回の行列の定数倍，そして 1 回の行列-行列和によって行われる．

5 まとめ

本稿では，行列の特性多項式および最小多項式候補と拡張 Horner 法を用いた逆行列の算法を提案した．

アルゴリズムの実装は数式処理システム Risa/Asir 上に行った．実験は，特性多項式を用いた算法に対して行い，Risa/Asir および数式処理システム Maple に組み込みの Gauss 消去法と効率を比較した．我々が提案する算法については，逐次計算および並列計算の場合の効率を比較した．逐次計算の場合，提案した算法の効率は，Gauss の消去法のそれに対して及ばなかった．一方，16 並列の並列計算の場合は，提案した算法の方が，Gauss 消去法に比べてより効率的だった．

本稿において，比較対象にした Gauss 消去法の実装はすべて逐次計算に基づくものであったが，Gauss 消去法はブロック分割による並列計算法も存在する [3] ので，今後，そのような実装を用いる機会があれば，それらとの比較も検討対象になるだろう．

最小多項式候補を用いた逆行列計算については，いくつかの計算例を示した．逆行列計算に最小多項式を用いた例では，特性多項式を用いるよりも効率的に逆行列を計算することが示された．また，最小多項式候補から最小多項式を求めた上で，逆行列を計算する例を示した．今後は，最小多項式候補から最小多項式を求める逆行列算法の実装を行い，その効果を確かめる所存である．

謝　辞

本稿の実験にあたり，行列の特性多項式計算プログラム [2] のソースコードをご提供下さった木村欣司氏に感謝いたします.

参 考 文 献

[1] E. H. Bareiss. Sylvester's identity and multistep integer-preserving Gaussian elimination. *Math. Comp.*, Vol. 22, pp. 565–578, 1968.

[2] K. Kimura. Linear Algebra PROGrams in Computer Algebra. http://www-is.amp.i.kyoto-u.ac.jp/kkimur/LAPROGCA/LAPROGCA.html (accessed July 7, 2016).

[3] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Forth edition, 2013.

[4] D. Knuth. *The Art of Computer Programming*, Vol. 2: Seminumerical Algorithms. Addison-Wesley, Third edition, 1998.

[5] Maplesoft, a division of Waterloo Maple Inc. *Maple 2016* (Computer software). Waterloo, Ontario, Canada, 2016.

[6] Shinichi Tajima, Katsuyoshi Ohara, and Akira Terui. An extension and efficient calculation of the horner's rule for matrices. In *Proceedings of the 4th International Congress on Mathematical Software (ICMS 2014)*, *Lecture Notes in Computer Science* 8592, pp. 346–351. Springer, 2014.

[7] 小原功任. OpenXM を用いた Risa/Asir 並列計算フレームワークの開発. 数式処理, Vol. 18, No. 1, pp. 20–26, 2011.

[8] 小原功任, 田島慎一. 最小消去多項式を用いた行列スペクトル分解計算の並列化. In *Computer Algebra: Design of Algorithms, Implementations and Applications*, 数理解析研究所講究録, 第 1815 巻, pp. 21–28. 京都大学数理解析研究所, 2012.

[9] 小原功任, 田島慎一. 最小消去多項式候補を用いた行列の一般固有空間の構造の計算法について. 数式処理研究と産学研究の新たな発展, MI レクチャーノート, 第 49 巻, pp. 113–118. 九州大学マス・フォア・インダストリ研究所, 2013.

[10] 田島慎一, 小原功任, 照井章. 行列 Horner 法の拡張と効率化. 数式処理研究の新たな発展, 数理解析研究所講究録, 第 1930 巻, pp. 26–38. 京都大学数理解析研究所, 2015.

[11] 田島慎一, 小原功任, 照井章. 行列 Horner 法の並列化の実装について. 数式処理研究の新たな発展, 数理解析研究所講究録, 第 1930 巻, pp. 51–59. 京都大学数理解析研究所, 2015.

[12] 田島慎一, 照井章. 行列の最小消去多項式候補を利用した固有ベクトル計算 (IV). 数式処理とその周辺分野の研究, 数理解析研究所講究録, 第 1955 巻, pp. 188–197. 京都大学数理解析研究所, 2015.

[13] 田島慎一, 奈良洸平. 最小消去多項式候補とその応用. In *Computer Algebra — Design of Algorithms, Implementations and Applications*, 数理解析研究所講究録, 第 1815 巻, pp. 1–12. 京都大学数理解析研究所, 2012.