東邦大学大学院　理学研究科　情報科学専攻

2015 年度　修士論文

安定化理論に基づく計算履歴法の一般逆行列計算に対する有効性について

指導教員：白柳　潔

提出日：2016 年 1 月 29 日

提出者：6514007　見田　大志

概要

数式処理のアルゴリズムは、基本的に厳密計算を前提としている。素朴に近似計算を使って実行すると、誤差の影響で真の出力に近づかないばかりか、全く異なる出力を出すことさえある。このような現象を不安定性と呼ぶ。この不安定性を解消するために、白柳らは代数的アルゴリズムの安定化手法を提案した。その安定化手法に基づき提案された ISCZ 法ではシンボル付きの区間演算を導入し、計算履歴を残すことにより厳密解を得ることができる。

本研究では、数式処理ソフト Maple14 を用いて、ISCZ 法をムーア･ペンローズ型一般逆行列を求める Karampetakis のアルゴリズムと Grevill のアルゴリズムに適用した。Karampetakis のアルゴリズムでは有効性は見られなかったが、Grevill のアルゴリズムではいくつかの有効な例が発見できた。

目次

1　はじめに

一般的に計算機では小数はある桁で切り捨て等を行い、近似計算を行っている。近似計算では厳密計算に比べて誤差が生じやすく、アルゴリズムによっては全く違う結果を出力することがある。

近似計算で誤った結果を出す例

厳密計算

$$1-\frac{1}{7}\times 7=0 \text{ は真か？}$$

$$1-\frac{1}{7}\times 7=1-1=0$$

$>$YES

近似計算

$$1-\frac{1}{7}\times 7=0 \text{ は真か？}$$

$$1-\frac{1}{7}\times 7=1-0.142857\times 7=0.000001$$

$>$NO

このような厳密計算と近似計算で異なる結果を出しやすいアルゴリズムを、「不安定なアルゴリズム」と呼ぶ。この不安定なアルゴリズムを安定させるために、白柳らは安定化手法を提案した ([1])。その安定化手法は、入力した多項式の係数を区間化して区間演算を行い、If 文のところでは、0 が含まれる区間を 0 に変換するという「ゼロ書き換え」を施すものである。

厳密計算

$$1-\frac{1}{7}\times 7=0 \text{ は真か？}$$

$$1-\frac{1}{7}\times 7=1-1=0$$

$>$YES

安定化手法を用いた計算

$1 - \frac{1}{7} \times 7 = 0$ は真か？

$$
\begin{aligned}
1 - \frac{1}{7} \times 7 &= [1,1] - [0.142857, 0.142858] \times [7,7] \\
&= [-0.000006, 0.000001] \\
&= 0
\end{aligned}
$$

>YES

安定化手法はすべてのアルゴリズムに適用可能ではなく、
・入力が多項式である
・加減乗除演算のみで構成されている
・「YES か NO か」の判定が「0 であるか？」もしくは「正 (負) であるか？」の 3 点全てを満たさなければならない。安定化手法の適用可能条件は厳しいが、適用できるアルゴリズムでは近似計算よりも厳密計算に近い値を返し、厳密計算よりも短時間で解を求めることができる。
しかし、現在の研究では精度桁を増やすことで厳密解に近い解を返すことは保証されているが、精度桁が何桁以上のときに確実に安定するかは未解決問題となっている。
ISCZ 法では、ゼロ書き換えが真に正しいかどうかをその都度確認するので、出力の正当性を確認する必要がないことが安定化手法との相違である。

2　安定化手法

次のアルゴリズムを対象に安定化理論の復習を簡単に行う。

- データは、すべて多項式環 $R[x_1, \ldots, x_m]$ の元からなる。R は実数体の部分体である。

- データ間の演算は、$R[x_1, \ldots, x_m]$ 内の加減乗除である。

- データ上の述語は、不連続点をもつとすればそれは 0 のみである。

述語の不連続点 0 という意味は、If "$C = 0''$ then ... else ... のように、値が 0 か否かによって分岐が分かれることである。従って、$C = 0$ の代わりに $C > 0$ や $C \leq 0$ などでもよい。上記のクラスのアルゴルズムを、不連続点 0 の代数的アルゴリズムと呼ぶ。ほとんどの数式処理のアルゴリズムはこのクラスに入るか、このクラスのアルゴリズムに変換可能である。
さて、安定化の 3 つのポイントは、

- アルゴリズムの構造は変えない

- データ領域において、ふつうの係数を区間係数に変える。

- 述語の評価の直前で、区間係数のゼロ書き換えを行う。

である。すなわち、安定化されたアルゴリズムは次のようになる。
区間領域　データ領域は区間係数多項式の集合。区間係数は $[A, B]$ なる形で、$A, B \in \mathbb{R}, [A, B]$ は集合 $\{r \in \mathbb{R} | A \leq r \leq B\}$ を意味する。
区間演算　二項演算 $\star \in \{+, -, \times, \div\}$ に対し、
$[A, B] \star [C, D] = [min(A \star C, A \star D, B \star C, B \star D), max(A \star C, A \star D, B \star C, B \star D)]$

ゼロ書き換え　不連続点 0 をもつ述語を評価する直前で、各区間係数 $[A, B]$ に対し、
$A \leq 0 \leq B$ ならば $[A, B]$ を $[0, 0]$ に書き換えよ。
そうでないならばそのままとせよ。
今、入力 $f \in R[x_1, \ldots, x_m]$ を

$$f = \sum_{i_1,\ldots,i_m} r_{i_1 \cdots i_m} x_1^{i_1} \cdots x_m^{i_m}$$

と表したとき、f に対する近似列 $Int(f)_j$ を

$$Int(f)_j = \sum_{i_1,\ldots,i_m} [(a_{i_1 \cdots i_m})_j, (b_{i_1 \cdots i_m})_j] x_1^{i_1} \cdots x_m^{i_m}$$

で定義する。ここに、すべての $i_1, \ldots, i_m$ について、

$$(a_{i_1 \cdots i_m})_j \leq r_{i_1 \cdots i_m} \leq (b_{i_1 \cdots i_m})_j \ for \ \forall j$$

$$(b_{i_1 \cdots i_m})_j - (a_{i_1 \cdots i_m})_j \to 0 \ as \ j \to \infty$$

このとき、単に

$$Int(f)_j \to f$$

と書く。
さて、A を安定化したアルゴリズムを $Stab(A)$ とかくと、次が安定化理論
安定化理論の基本定理である。

定理 1（安定化理論の基本定理）　A は不連続点 0 の代数的アルゴリズムで、入力 $f \in R[x_1, \ldots, x_m]$ に対し正常終了するとせよ。このとき、f に対する任意の近似列 $\{Int(f)_j\}_j$ に対し、ある n が存在して、$j \geq n$ ならば、$Stab(A)$ は $\{Int(f)_j\}_j$ に対し正常終了し、

$$Stab(A)(Int(f)_j) \to A(f)$$

簡明を期すため、入力は一つだけの多項式にしているが、入力はもちろん、多項式の有限集合でもよい。

3　ISCZ 法

3.1　シンボル付き区間

区間と形式的なシンボルを組み合わせた係数（シンボル付き区間）を導入する。区間は、従来と同様の意味の区間である。シンボルは、アルゴリズム実行中に現れる係数の log（記録）を取るのに使われる。例えば、入力係数が $\frac{1}{3}$ と $\frac{1}{7}$ だったとしよう。これらの精度 3 の区間はそれぞれ $[0.333, 0.334]$ と $[0.142, 0.143]$ である。さて、$\frac{1}{3}$ に対するシンボルを s、$\frac{1}{7}$ に対するシンボルを t として、区間を組み合わせると、それぞれ、$[[0.333, 0.334], s]$ と $[[0.142, 0.143], t]$ となる。次に、これらのあいだの演算、

$$[[0.333, 0.334], s] + [[0.142, 0.143], t] = [[0.333, 0.334] + [0.142, 0.143], s\dot{+}t]$$

と定義する。$[0.333, 0.334] + [0.142, 0.143]$ に対しては通常の区間演算を使う。シンボル部分 $s\dot{+}t$ は再び形式的なシンボルで、加算を実施したことを記録できれば何でもよい。
アルゴリズム終了後、最終的なシンボルを正確係数に復元する。先の簡単な例で言えば、もし最終的なシンボルが $s\dot{+}t$ であったとすれば、s に $\frac{1}{3}$ を、t に $\frac{1}{7}$ を代入し、$\dot{+}$ には加算の意味を与えて、$\frac{1}{3} + \frac{1}{7} = \frac{10}{21}$ と復元する。
シンボル付き区間のことを interval-symbol、あるいは単に IS と呼ぶ。

3.2　手続き

A を不連続点 0 の代数的アルゴリズムとする。ISCZ 法の手続きは次の通りである。
R-to-IS 各入力係数 r を $[[A, B], Symbol_r]$ に変換する。ここに、$[A, B]$ は r の予め定められた精度の区間、$Symbol_r$ は r を表すシンボル（以下、入力シンボルと呼ぶ）である。
IS 演算　IS 間の演算を次のように実行する:

$$[[A, B], s] + [[C, D], t] = [[A, B] + [C, D], s\dot{+}t]$$
$$[[A, B], s] - [[C, D], t] = [[A, B] - [C, D], s\dot{-}t]$$

$$[[A,B],s] \times [[C,D],t] = [[A,B] \times [C,D], s\dot{\times}t]$$

$$[[A,B],s] \div [[C,D],t] = [[A,B] \div [C,D], s\dot{\div}t]$$

すなわち、区間部分については区間演算を用い、シンボル部分については加算、減算、乗算、除算の形式的なシンボル $\dot{+}, \dot{-}, \dot{\times}, \dot{\div}$ を使って、どういう演算が行われたかを記録する。

正しいゼロ書き換え　任意の $IS[[A,B],s]$ に対し、$A \leq 0 \leq B$ ならば、s をそれぞれに対応する実数 $r(s)$ に復元する。もし、$r(s) = 0$ ならば、次のステップに進む。そうでなければ、精度を上げて R-to-IS に戻る。

IS-to-R 出力のシンボル部分の各入力シンボルのそれぞれ対応する入力係数をだいにゅうし、演算シンボルに演算の意味を与えて実数値に復元する。

この手法を ISCZ 法（IS method with correct zero rewriting）と呼ぶ。

さて、ある精度 j で $Int(f)_j$ を $Stab(A)$ に入力したときの実行過程は、もしアルゴリズム中のすべてのゼロ書き換えが正しいならば、真の入力 f を A に入力したときの実行過程と完全に一致する。ISCZ 法では、各ゼロ書き換えにおいて、それが正しいのかどうかを確認する。さらに、定理 1 により、すべてのゼロ書き換えが正しくなる精度が存在する。したがって、ISCZ 法は有限ステップで終了し、その出力の各 IS 係数のシンボルは正しい正確係数を与える。

定理 2（ISCZ 法の停止性と正当性）　A が入力 I で正常終了するとせよ。このとき、A に対する ISCZ 法は、常に有限ステップで終了し、正しい結果、すなわち、$A(I)$ の出力と同じ結果を与える。

ISCZ 法の利点は、ISZ 法と違い、出力の正当性を確認する必要がないことである。任意の $IS[[A,b],s]$ に対し、$A \leq 0 \leq B$ でない限り、正確計算をスキップすることができ、浮動小数点計算だけで済む。換言すれば、ゼロでない係数についての正確計算を省略することができる。したがって、本手法は、$A \leq 0 \leq B$ でない場合が $A \leq 0 \leq B$ である場合よりも多ければ多いほど有効であるということができる。

4　ムーア・ペンローズ型一般逆行列

4.1　定義

A を $m \times n$ 実数成分行列、A^+ を $n \times m$ 実数成分行列とする。
次の 4 つの条件、

(i)$AA^+A = A$

(ii)$A^+AA^+ = A^+$

(iii)$(AA^+)^T = AA^+$

(iv)$(A^+A)^T = A^+A$

を満たす A^+ を A のムーア・ペンローズ型一般逆行列と呼ぶ。([3])
ムーア・ペンローズ型一般逆行列は、 連想記憶やパターン認識などに工学的な応用をもつ ([4])。

ムーア・ペンローズ型一般逆行列を求めるアルゴリズムはいくつかあるが、本研究では Karampetakis のアルゴリズムと Grevill のアルゴリズムを対象に実験を行う。

4.2 Karampetakis のアルゴリズム

入力:$A \in \mathbb{R}^{n\times m}$
出力:$A^{+} \in \mathbb{R}^{m\times n}$
Step1.
$i = 0, 1, \ldots, n$ において、A_i, a_i, B_i を順次計算する。
なお、A_i, a_i, B_i は、

$$i = 0 \Rightarrow \begin{cases} A_0 = 0_{n,n} \\ a_0 = 1 \\ B_0 = I_n \end{cases}$$

$$i = 1, \ldots, n \Rightarrow \begin{cases} A_i = [AA^T]B_{i-1} \\ a_i = -\dfrac{trace(A_i)}{i} \\ B_i = A_i + a_i I_n \end{cases}$$

Step2.
$a_n = \cdots = a_{k+1} = 0$ で $a_k \neq 0$ ならば、

$$A^{+} \stackrel{\text{def}}{=} -a_k^{-1} A^T B_{k-1}$$

$a_n = \cdots = a_1 = 0$ ならば、

$$A^{+} \stackrel{\text{def}}{=} 0_{m,n}$$

4.2.1 実験

Karampetakis のアルゴリズムに ISCZ 法を適用し実験を行う。
本実験で使用する PC 及び実行環境は次の通りである。

1. 使用コンピュータ
OS:Windows 7 Home Premium
CPU:Intel(R) Core(TM) i7-4770K CPU @ 3.50GHz
実装メモリ (RAM):8.00GB
2. 使用ソフト
数式処理ソフト Maple14

- 実験方法
  - 一般逆行列を Karampetakis のアルゴリズムを用いて計算する
  - 以下のそれぞれの方法について、計算開始から出力までの時間を計測する
    - Maple14 組み込み関数 (M)
    - Maple14 自作の厳密計算 (厳密)
    - Maple14 ISCZ 法 (ISCZ)
      開始精度は 3 桁から始め、ゼロ判定に失敗した場合 1 桁ずつ上げていく。
  - 入力する行列は RandomMatrix 関数を用い生成した。

実験結果（有理数行列）

入力行列	出力精度	シンボルリスト長	実行結果		
			M	厳密	ISCZ
5×6	3	1561	0.03s	0.03s	0.39s
6×5	7	1561	0.05s	0.00s	0.58s
10×11	21	21721	0.20s	0.14s	36.72s
11×10	19	21721	0.19s	0.16s	36.12s
15×16	31	106231	1.16s	1.16s	8.21m
16×15	31	106231	1.11s	1.08s	8.19m

実験結果（無理数を含む行列）

入力行列	出力精度	シンボルリスト長	実行結果		
			M	厳密	ISCZ
5×6	6	1561	1.14s	1.11s	11.49s
6×5	5	1561	0.53s	1.06s	13.78s
6×7	10	3073	9.48s	3.69s	51.89s
7×6	9	3073	0.81s	1.66s	26.35s
7×8	8	5503	8.66s	6.19s	94.56s
8×7	12	5503	7.97m	6.69s	113.30s

考察

有理数行列では、いずれのサイズの行列に対しても、厳密 $\leq M < ISCZ$ となっており有効な結果は得ることができなった。

無理数を含むような行列に関しても組み込みの関数より ISCZ 法が早い例もあるが、いずれの場合でも自作の厳密が早くあまり有効な結果は得ることができなかった。

Karampetakis のアルゴリズムでは、最終的に評価されてないシンボルがいくつかあったため、シンボルの評価法などを改良すればもっと早くなる可能性がある。

4.3 Grevill のアルゴリズム

入力行列を $A = [a_1^T, a_2^T, \ldots, a_n^T]^T$
a_i…m 次元ベクトル
とする。A_i を、

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

と定義し、$i = 1, 2, \ldots, m$ において、

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

を順次計算することにより、A_m^+ が A^+ となる。
なお、d_i, c_i, b_i, A_1^+ は、

$$d_i = a_i A_{i-1}^+$$
$$c_i = a_i - d_i A_{i-1}$$
$$b_i = \begin{cases} \dfrac{c_i}{c_i c_i^T} & (if\ c_i \neq 0) \\ \dfrac{d_i (A_{i-1}^+)^T}{1 + d_i d_i^T} & (if\ c_i = 0) \end{cases}$$
$$A_1^+ = \begin{cases} \dfrac{a_1^T}{a_1 a_1^T} & (if\ a_1 \neq 0) \\ a_1 & (if\ a_1 = 0) \end{cases}$$

4.3.1 実験

Grevill のアルゴリズムに ISCZ 法を適用し実験を行う。
本実験で使用する PC 及び実行環境は Karampetakis のアルゴリズムの実験と同様である。

- 実験方法
  - 一般逆行列を Greville のアルゴリズムを用いて計算する
  - 以下のそれぞれの方法について、計算開始から出力までの時間を計測する
    - Maple14 組み込み関数 (M)
    - Maple14 自作の厳密計算 (厳密)
    - Maple14 ISCZ 法 (ISCZ)
      開始精度は 3 桁から始め、ゼロ判定に失敗した場合 1 桁ずつ上げていく。
  - 入力する行列は RandomMatrix 関数を用い生成した。

実験結果（有理数行列）

入力行列	出力精度	シンボルリスト長	実行結果		
			M	厳密	ISCZ
5×6	5	465	0.016s	0.016s	0.203s
6×5	4	604	0.015s	0.015s	0.234s
10×11	10	3355	0.172s	0.063s	4.119s
11×10	12	3884	0.218s	0.063s	5.647s
15×16	11	10920	1.185s	0.219s	42.729s
16×15	15	12089	1.170s	0.265s	50.154s

実験結果（無理数を含む行列）

入力行列	出力精度	シンボルリスト長	実行結果		
			M	厳密	ISCZ
5×6	9	465	39.709s	10.296s	41.933s
6×5	6	604	7.129s	5.195s	15.694s
6×7	7	777	42.541s	10.312s	31.028s
7×6	8	974	17.847s	3.464s	15.943s
7×8	6	1204	87.079s	12.402s	22.777s
8×7	8	1469	79.561s	7.566s	22.433s

$$\begin{pmatrix} -\frac{8\sqrt{31}}{11} & \frac{2}{5} & \frac{19}{8} & -\frac{32}{77} & \frac{99}{25} \\ -\frac{9}{14} & \frac{9}{10} & \frac{27}{65} & -\frac{74}{9} & \frac{11\sqrt{13}}{3} \\ -\frac{25}{8} & -\frac{87\sqrt{17}}{25} & -\frac{93}{86} & -\frac{4}{31} & \frac{11}{3} \\ -\frac{22}{9} & \frac{11}{17} & -\frac{19}{5} & -\frac{27\sqrt{43}}{50} & -46 \\ \frac{5\sqrt{2}}{11} & -\frac{49}{38} & \frac{72}{61} & -\frac{1}{10} & -\frac{31}{50} \\ -\frac{27}{20} & \frac{7}{4} & \frac{\sqrt{7}}{24} & \frac{69}{43} & \frac{67}{10} \end{pmatrix}$$

入力行列	出力精度	シンボルリスト長	実行結果	
			厳密	ISCZ
6×5	4	604	4.92m/31.27GiB	3.94m/30.85GiB

$$\begin{pmatrix} \frac{11}{9} & \frac{95}{18} & \frac{4}{19} & \frac{1}{86} & \frac{22}{69} \\ -\frac{13\sqrt{7}}{44} & 1 & -\frac{1}{11} & \frac{63}{50} & -\frac{4}{5} \\ -\frac{3}{91} & \frac{5}{43} & -\frac{91\sqrt{13}}{71} & \frac{23}{94} & \frac{45}{2} \\ 1 & \frac{61}{51} & \frac{22}{25} & \frac{63}{97} & \frac{7\sqrt{5}}{44} \\ \frac{83}{94} & -\frac{26}{51} & \frac{38}{17} & \frac{13}{19} & \frac{20}{33} \\ \frac{1}{3} & -\frac{10\sqrt{19}}{19} & -\frac{38}{35} & -\frac{5}{6} & \frac{35}{59} \\ \frac{44}{9} & \frac{39}{19} & -\frac{7}{2} & -\frac{10\sqrt{3}}{69} & \frac{21}{10} \end{pmatrix}$$

入力行列	出力精度	シンボルリスト長	実行結果	
			厳密	ISCZ
7×5	7	864	3.63m/20.34GiB	86.10s/11.81GiB

$$\begin{pmatrix} \frac{77}{90} & \frac{94}{91} & -\frac{25\sqrt{5}}{41} & -1 & -\frac{8}{11} & -\frac{69}{44} \\ \frac{9}{80} & \frac{12}{29} & -\frac{11}{36} & \frac{49}{34} & -\frac{1}{30} & \frac{33}{8} \\ \frac{31}{19} & -\frac{1}{35} & \frac{15}{14} & \frac{77\sqrt{17}}{67} & \frac{32}{95} & \frac{29}{65} \\ -\frac{25}{44} & -\frac{25\sqrt{31}}{16} & -\frac{9}{2} & \frac{57}{22} & \frac{37}{10} & \frac{22}{43} \\ \frac{40}{41} & -10 & \frac{38}{59} & \frac{27}{14} & \frac{4}{25} & \frac{23}{5} \\ -\frac{43}{70} & -\frac{4}{13} & -\frac{3}{2} & -\frac{93}{16} & \frac{9\sqrt{41}}{17} & \frac{31}{61} \\ \frac{25}{41} & \frac{9}{13} & -\frac{87}{62} & -\frac{76}{9} & \frac{2}{19} & -\frac{67}{48} \end{pmatrix}$$

入力行列	出力精度	シンボルリスト長	実行結果	
			厳密	ISCZ
7×6	6	974	2.29m/13.86GiB	114.82s/15.24GiB

$$\begin{pmatrix} -\frac{1}{23} & -\frac{31}{14} & -\frac{13}{9} & 2 & \frac{1}{5} \\ -\frac{88\sqrt{23}}{35} & \frac{83}{81} & -\frac{10}{9} & -\frac{91}{55} & -\frac{11}{47} \\ -\frac{11}{6} & -\frac{1}{4} & -\frac{26\sqrt{13}}{21} & -\frac{1}{71} & -\frac{12}{97} \\ \frac{59}{17} & -1 & -\frac{2}{43} & -\frac{63\sqrt{31}}{50} & -\frac{45}{38} \\ -\frac{2}{5} & \frac{95}{91} & -\frac{5}{51} & \frac{23}{17} & \frac{7}{18} \\ -\frac{22}{39} & -\frac{63\sqrt{17}}{62} & -\frac{91}{51} & -\frac{9}{5} & -\frac{20}{23} \\ \frac{26}{23} & -\frac{5}{47} & -\frac{22\sqrt{5}}{19} & 1 & -\frac{35}{69} \\ \frac{3}{67} & -\frac{61}{27} & 1 & -\frac{15}{43} & -\frac{7\sqrt{43}}{5} \end{pmatrix}$$

入力 行列	出力 精度	シンボル リスト長	実行結果	
			厳密	ISCZ
8×5	8	1165	42.90m/241.42GiB	30.70m/224.66GiB

$$\begin{pmatrix} \frac{12}{5} & -\frac{67}{21} & -\frac{4}{85} & \frac{10}{31} & -\frac{3\sqrt{37}}{26} & -\frac{41}{49} \\ \frac{3}{22} & -\frac{37}{95} & -\frac{3}{2} & \frac{83}{75} & \frac{59}{47} & \frac{53}{90} \\ -\frac{59}{83} & -\frac{9}{16} & \frac{66\sqrt{11}}{41} & \frac{19}{25} & \frac{9}{25} & 3 \\ \frac{6}{5} & -\frac{13}{28} & -\frac{14}{25} & -\frac{4}{7} & -\frac{43}{49} & \frac{27\sqrt{5}}{43} \\ -\frac{12}{19} & 26 & \frac{29}{37} & -\frac{6}{5} & \frac{38}{15} & -\frac{91}{10} \\ \frac{22}{9} & -\frac{25}{48} & \frac{15}{34} & -\frac{81\sqrt{23}}{13} & \frac{85}{33} & \frac{92}{5} \\ \frac{6}{7} & \frac{34}{99} & \frac{11}{31} & \frac{17}{4} & -\frac{45}{37} & \frac{26}{29} \\ \frac{44}{89} & \frac{34\sqrt{2}}{99} & -\frac{26}{11} & \frac{91}{3} & -\frac{29}{22} & \frac{43}{30} \end{pmatrix}$$

入力 行列	出力 精度	シンボル リスト長	実行結果	
			厳密	ISCZ
8×6	6	1312	45.86m/152.67GiB	28.61m/144.14GiB

$$\begin{pmatrix} \frac{99}{95} & \frac{24}{5} & -\frac{31\sqrt{13}}{23} & -\frac{25}{7} & \frac{19}{41} & -\frac{93}{52} & -\frac{4}{31} \\ \frac{20}{21} & -\frac{5}{7} & \frac{50}{63} & \frac{1}{6} & \frac{9}{35} & \frac{76}{13} & -\frac{23}{11} \\ -\frac{19}{2} & -\frac{43\sqrt{3}}{22} & \frac{40}{13} & \frac{16}{35} & \frac{87}{41} & -\frac{36}{41} & -\frac{99}{68} \\ \frac{20}{61} & -\frac{10}{19} & \frac{43}{30} & -\frac{3}{7} & \frac{33}{91} & -\frac{1}{36} & -\frac{29}{67} \\ \frac{25}{26} & \frac{61}{38} & \frac{5}{2} & -\frac{5}{9} & -\frac{98\sqrt{23}}{29} & -\frac{16}{21} & 2 \\ -\frac{51}{20} & -\frac{48}{91} & \frac{47}{11} & -\frac{11}{40} & -\frac{11}{10} & -\frac{37}{9} & \frac{46}{7} \\ -\frac{38}{39} & -77 & 1 & \frac{45}{19} & -\frac{57}{32} & \frac{4}{59} & -\frac{31\sqrt{31}}{16} \\ 11 & \frac{1}{7} & -\frac{2}{45} & -\frac{81}{88} & -27 & \frac{9}{4} & \frac{67}{9} \end{pmatrix}$$

入力 行列	出力 精度	シンボル リスト長	実行結果	
			厳密	ISCZ
8×7	7	1469	22.11m/116.80GiB	15.18m/115.04GiB

$$\begin{pmatrix} -\frac{71}{59} & \sqrt{2} & -\frac{1}{2} & -\frac{77}{10} & \frac{96}{59} & -\frac{33}{23} & \sqrt{53} \\ -\frac{36}{29} & 1 & \frac{27}{13} & \frac{87}{10} & -\frac{47}{10} & -\frac{59}{45} & \frac{33}{28} \\ \frac{72}{53} & \frac{9}{10} & -\frac{6}{5} & \frac{62}{27} & 2 & -\frac{11}{9} & \frac{15}{17} \\ -\frac{18}{55} & \frac{9}{94} & \frac{16}{15} & -\frac{23}{19} & \frac{11}{54} & -\frac{37}{62} & \frac{91}{47} \\ \frac{74}{43} & -\frac{46}{97} & -\frac{3}{40} & \frac{18}{11} & \frac{11}{9} & \sqrt{3} & -\frac{68}{27} \\ \frac{93}{41} & \frac{4}{3} & \frac{37}{70} & -1 & -\frac{2}{83} & -\frac{72}{71} & \frac{13}{2} \\ 0 & -\frac{4}{5} & \frac{19}{99} & -\frac{87}{79} & \frac{79}{26} & 0 & \frac{41}{68} \\ 4\sqrt{6} & \frac{21}{31} & -\frac{73}{90} & -\frac{39}{29} & \frac{86}{63} & \frac{57}{8} & -\frac{26}{81} \end{pmatrix}$$

入力 行列	出力 精度	シンボル リスト長	実行結果	
			厳密	ISCZ
8×7	8	1469	10.12m/40.62GiB	5.92m/39.91GiB

$$\begin{pmatrix} \frac{5}{3} & \frac{25\sqrt{3}}{88} & -\frac{11}{26} & -\frac{19}{11} & -\frac{27}{95} \\ -\frac{61}{45} & -\frac{47}{41} & -\frac{45}{13} & -\frac{27}{68} & -\frac{2}{5} \\ \frac{24}{7} & -\frac{6}{35} & -\frac{81\sqrt{5}}{82} & \frac{93}{67} & -\frac{69}{25} \\ \frac{77}{60} & -\frac{2}{41} & -\frac{19}{36} & -\frac{38}{11} & \frac{33}{17} \\ -\frac{9\sqrt{2}}{35} & \frac{50}{91} & -\frac{3}{7} & -\frac{36}{7} & \frac{29}{76} \\ \frac{31}{21} & \frac{10}{29} & \frac{29}{6} & -\frac{1}{8} & -1 \\ -\frac{5}{9} & -\frac{8}{35} & -\frac{33}{59} & -\frac{32}{9} & \frac{23}{6} \\ -1 & \frac{9}{32} & -\frac{49}{6} & -\frac{74}{99} & -\frac{31\sqrt{17}}{65} \\ \frac{43}{19} & 50 & \frac{77}{62} & -\frac{\sqrt{7}}{15} & \frac{67}{86} \end{pmatrix}$$

入力 行列	出力 精度	シンボル リスト長	実行結果	
			厳密	ISCZ
9×5	11	1507	41.83m/235.48GiB	34.30m/224.82GiB

$$\begin{pmatrix} \frac{13}{4} & -\frac{50}{79} & \frac{47}{90} & -\frac{11\sqrt{11}}{7} & -\frac{93}{28} & -\frac{47}{27} \\ \frac{89}{39} & \frac{9}{17} & \frac{30}{43} & \frac{21}{83} & -\frac{12}{11} & \frac{41}{26} \\ -\frac{74\sqrt{7}}{19} & \frac{59}{16} & -\frac{13}{10} & -\frac{32}{37} & \frac{26}{55} & -\frac{88}{37} \\ -\frac{29}{31} & 1 & -\frac{63}{5} & \frac{57}{68} & \frac{8}{13} & \frac{4}{7} \\ -\frac{27}{38} & -\frac{16}{35} & \frac{47}{45} & \frac{5}{7} & \frac{85\sqrt{13}}{33} & \frac{37}{46} \\ -\frac{2}{3} & -\frac{97}{5} & \frac{8}{7} & \frac{13}{23} & -\frac{41}{16} & \frac{1}{2} \\ -\frac{29}{25} & -\frac{85}{27} & -\frac{48}{17} & \frac{44}{57} & 7 & \frac{57}{62} \\ \frac{85}{97} & -\frac{25}{46} & -\frac{75}{31} & \frac{47}{53} & -\frac{3}{2} & -\frac{30}{37} \\ 1 & -\frac{16}{31} & -\frac{62}{3} & \frac{90}{71} & -\frac{29}{18} & \frac{67}{57} \end{pmatrix}$$

入力 行列	出力 精度	シンボル リスト長	実行結果	
			厳密	ISCZ
9×6	6	1710	38.81m/151.50GiB	28.02m/116.04GiB

$$\begin{pmatrix} -\frac{5}{19} & \frac{51}{37} & -\frac{67}{11} & \frac{18}{47} & \frac{55}{54} & -\frac{97}{68} & -\frac{59\sqrt{5}}{89} \\ \frac{39}{82} & -\frac{20}{97} & \frac{28}{61} & \frac{1}{5} & -\frac{71}{72} & -\frac{19}{29} & \frac{5}{46} \\ \frac{35}{29} & \frac{1}{2} & -\frac{81}{28} & -\frac{63\sqrt{37}}{41} & \frac{50}{79} & \frac{36}{43} & 22 \\ \frac{26}{29} & \frac{35}{73} & \frac{3}{4} & -\frac{86}{79} & -\frac{17}{75} & \frac{69}{85} & -\frac{13}{23} \\ -\frac{74\sqrt{7}}{35} & -\frac{27}{22} & \frac{88}{63} & -\frac{17}{3} & -\frac{7}{17} & -\frac{69}{85} & \frac{3}{29} \\ -\frac{13}{70} & -\frac{17}{92} & \frac{91}{27} & \frac{17}{15} & \frac{26}{19} & -\frac{15}{19} & -\frac{62}{9} \\ -\frac{32}{43} & -\frac{25}{73} & -\frac{31}{29} & -\frac{19}{5} & -\frac{86}{57} & \frac{2}{25} & -\frac{83}{81} \\ -\frac{48}{23} & -2 & -47 & \frac{38}{5} & \frac{50}{83} & -\frac{88\sqrt{19}}{17} & \frac{9}{35} \\ \frac{30}{7} & \frac{23}{62} & \frac{1}{2} & -\frac{19}{47} & \frac{94}{45} & \frac{11}{9} & \frac{11}{10} \end{pmatrix}$$

入力 行列	出力 精度	シンボル リスト長	実行結果	
			厳密	ISCZ
9×7	8	1945	3.89h/0.76TiB	2.92h/0.76TiB

$$\begin{pmatrix} -\frac{47\sqrt{2}}{46} & -\frac{6}{5} & \frac{97}{93} & \frac{73}{34} & -\frac{83}{30} & \frac{49}{81} & \frac{81}{41} & \frac{98}{83} \\ -\frac{6}{5} & 42 & \frac{39}{29} & \frac{16}{29} & \frac{43}{64} & -\frac{10}{9} & -\frac{23}{29} & -\frac{23}{5} \\ \frac{23}{2} & -\frac{14\sqrt{7}}{37} & -\frac{20}{47} & -\frac{43}{47} & -\frac{67}{64} & \frac{69}{19} & -\frac{27}{19} & \frac{10}{13} \\ \frac{45}{53} & \frac{58}{33} & -\frac{41}{81} & \frac{65}{84} & -\frac{13}{3} & -\frac{49}{76} & -\frac{11}{17} & \frac{63}{25} \\ -12 & -\frac{93}{4} & \frac{11}{14} & \frac{29}{11} & \frac{11}{72} & -\frac{74}{95} & \frac{47}{74} & \frac{47\sqrt{5}}{74} \\ -\frac{79}{47} & -\frac{75}{58} & \frac{51}{26} & -11 & \frac{91}{58} & -\frac{49}{27} & \frac{47}{14} & -\frac{17}{12} \\ -\frac{19}{42} & \frac{90}{97} & -\frac{11}{81} & -\frac{79}{47} & -\frac{17}{29} & -\frac{7}{8} & -\frac{85}{48} & -\frac{49}{37} \\ \frac{7}{38} & -\frac{24}{19} & -\frac{64}{25} & \frac{82}{87} & -\frac{58}{17} & \frac{52}{61} & -\frac{99}{97} & \frac{7}{10} \\ \frac{25}{83} & \frac{1}{3} & -\frac{2}{7} & -\frac{15}{94} & -\frac{69}{41} & \frac{55}{71} & \frac{77}{3} & \frac{95}{48} \end{pmatrix}$$

入力 行列	出力 精度	シンボル リスト長	実行結果	
			厳密	ISCZ
9×8	8	2092	2.88h/414.05GiB	2.25h/366.93GiB

$$\begin{pmatrix} -\frac{81\sqrt{2}}{76} & -\frac{49}{10} & -\frac{76}{31} & -\frac{2}{47} & -\frac{29}{16} \\ \frac{19}{22} & \frac{77}{61} & \frac{36}{25} & \frac{9}{4} & -\frac{44}{9} \\ -\frac{3}{4} & -\frac{19}{16} & \frac{1}{40} & -4 & -\frac{46}{25} \\ \frac{87}{65} & \frac{27}{77} & -\frac{32}{43} & \frac{69\sqrt{7}}{50} & \frac{31}{22} \\ \frac{33}{86} & -\frac{31}{3} & -\frac{74}{25} & \frac{99}{10} & \frac{67}{45} \\ -\frac{35}{19} & -\frac{82}{63} & 1 & \frac{22}{21} & -\frac{30}{41} \\ -\frac{32}{91} & -\frac{36\sqrt{5}}{13} & -\frac{62}{45} & \frac{7}{45} & \frac{19}{14} \\ 1 & \frac{7}{5} & \frac{33}{14} & \frac{1}{5} & -\frac{20}{41} \\ \frac{52}{63} & \frac{9}{5} & -\frac{17}{15} & \frac{9}{19} & -\frac{25}{91} \\ \frac{13}{23} & -\frac{59}{22} & \frac{67}{35} & \frac{9}{8} & \frac{51}{29} \end{pmatrix}$$

入力 行列	出力 精度	シンボル リスト長	実行結果	
			厳密	ISCZ
10×5	7	1890	3.50h/0.92TiB	3.24h/0.78TiB

$$\begin{pmatrix} \frac{29}{81} & -\frac{13\sqrt{7}}{14} & -\frac{25}{11} & -\frac{47}{45} & \frac{19}{79} & \frac{97}{85} \\ \frac{9}{11} & \frac{32}{37} & \frac{78}{61} & -\frac{27}{41} & -\frac{11}{15} & \frac{38}{85} \\ -\frac{81}{76} & -\frac{48}{97} & \frac{23}{28} & -\frac{18}{79} & -\frac{71}{85} & -\frac{36}{19} \\ \frac{35}{82} & \frac{15}{23} & \frac{67}{48} & 2 & \frac{50}{19} & -\frac{69}{25} \\ -\frac{80}{29} & \frac{51}{73} & -\frac{4}{9} & \frac{7}{5} & -\frac{17}{57} & \frac{69}{17} \\ \frac{20}{29} & \frac{5}{11} & -3 & -\frac{43}{5} & \frac{35}{83} & -\frac{5\sqrt{3}}{27} \\ \frac{39}{35} & -\frac{1}{2} & -\frac{18}{29} & \frac{51}{5} & \frac{26}{45} & \frac{2}{89} \\ \frac{1}{2} & \frac{35}{73} & -44 & \frac{51}{47} & -\frac{43}{34} & -\frac{22}{23} \\ -\frac{26}{43} & \frac{18}{13} & \frac{91}{54} & -\frac{19}{27} & \frac{25}{29} & -\frac{99}{2} \\ \frac{74}{23} & -\frac{17}{62} & -\frac{62\sqrt{19}}{47} & \frac{19}{36} & \frac{94}{43} & \frac{59}{46} \end{pmatrix}$$

入力行列	出力精度	シンボルリスト長	実行結果	
			厳密	ISCZ
10×6	10	2201	28.81h/52.35TiB	21.95h/19.37TiB

$$\begin{pmatrix} \frac{80}{97} & -\frac{40}{17} & \frac{43}{56} & \frac{5}{13} & -\frac{93}{86} & -\frac{89}{63} & -\frac{34}{27} \\ \frac{69}{68} & -\frac{19}{23} & -\frac{2}{55} & -\frac{29}{40} & \frac{12}{95} & -\frac{19}{18} & \frac{55}{46} \\ \frac{7}{66} & \frac{62}{53} & \frac{23}{20} & -\frac{75}{89} & \frac{41}{16} & -7 & -\frac{9}{10} \\ \frac{86}{55} & \frac{81}{65} & -\frac{17}{60} & -\frac{31}{24} & \frac{4}{41} & -\frac{28\sqrt{13}}{25} & -\frac{79}{86} \\ -\frac{19}{70} & -\frac{11}{27} & \frac{87}{76} & -\frac{30\sqrt{31}}{67} & \frac{2}{11} & -\frac{21}{11} & -\frac{99}{64} \\ -\frac{53}{91} & \frac{5}{4} & -\frac{37}{12} & -1 & \frac{78}{61} & \frac{32}{9} & \frac{4}{5} \\ -\frac{3}{17} & -\frac{26}{27} & \frac{11}{23} & \frac{49}{15} & \frac{25}{19} & \frac{23}{14} & \frac{9}{43} \\ -\frac{89\sqrt{7}}{96} & -\frac{8}{97} & \frac{17}{78} & -\frac{5}{28} & \frac{3}{31} & -\frac{36}{43} & 3 \\ \frac{66}{71} & \frac{45}{14} & \frac{58}{81} & -\frac{23}{82} & \frac{19}{47} & -\frac{6\sqrt{11}}{11} & \frac{31}{68} \\ -\frac{77}{20} & \frac{81}{73} & -\frac{21}{11} & -\frac{19}{68} & \frac{69}{31} & -\frac{55}{48} & 33 \end{pmatrix}$$

入力 行列	出力 精度	シンボル リスト長	実行結果	
			厳密	ISCZ
10×7	9	2478	24.04m/131.38GiB	20.10m/105.57GiB

$$\begin{pmatrix} -\frac{19\sqrt{3}}{12} & \frac{57}{77} & -\frac{32}{25} & -\frac{99\sqrt{17}}{16} \\ -\frac{18}{65} & 3 & -\frac{37\sqrt{7}}{47} & -\frac{29}{9} \\ \frac{87}{86} & -3 & -\frac{\sqrt{11}}{3} & -\frac{22}{25} \\ \frac{33\sqrt{19}}{20} & \frac{38}{25} & -\frac{27}{2} & -\frac{46\sqrt{13}}{11} \\ \frac{98}{61} & \frac{9\sqrt{5}}{10} & \frac{4}{25} & -\frac{31}{45} \\ \frac{77}{48} & -\frac{2}{43} & \frac{69}{10} & -\frac{67\sqrt{2}}{81} \end{pmatrix}$$

入力 行列	出力 精度	シンボル リスト長	実行結果	
			厳密	ISCZ
6×4	5	517	-	4.08h/6.08TiB

この行列の厳密計算では、メモリ不足のため結果を出力することができなかった。

考察

有理数行列では、Karampetakis のアルゴリズムと同様にいずれのサイズの行列に対しても、厳密 $\leq M < ISCZ$ となっており有効な結果は得ることができなった。

無理数を含むような行列では、いくつかの有効な例が発見できた。

有効な例には無理数を含むことと、アルゴリズム中の b_i の計算でゼロ書き換えがされるという共通点はあったものの入力行列に対する特徴を見つけることはできなかった。

5　まとめ

今回の実験では、Karampetakis のアルゴリズムに対しては ISCZ 法の有効な例は発見できなかったためさらなる実験が必要だと考えられる。Grevill のアルゴリズムに対しては有効な例は発見できたが、どのような行列に対し有効であるかを解明するには至らなかったためそれが今後の課題となる。

最後に、今回の研究を行うにあたりアドバイスを下さった白柳潔教授、足立智子教授、並木誠准教授、白柳研究室の皆様に感謝いたします。

参考文献

[1] Shirayanagi, K. and Sweedler, M.: A Theory of Stabilizing Algebraic Algorithms, Technical Report 95-28, Mathematical Sciences Institute, Cornell University, pp.1-92 (1995).

[2] 白柳　潔、関川　浩：安定化理論に基づく log method について, 数理解析研究所講究録 1666 巻 (2009), 98-105.

[3] 水口 寛之、甲斐 博 、野田 松太郎：「Risa/Asir による 般逆行列の計算とその連想記憶への応用」 数理解析研究所講究録 986 巻 (1997), 166-173.

[4] Therrien C. W.: Eigenvalue Properties of Projection Operators and Their Application to the Subspace Method of Feature Extraction, IEEE Trans. Comput., C-24 (1975), 944-948.

[5] Nicholas Karampetakis:The Computation and Application of the Generalized Inverse via Maple,JOURNAL OF SYMBOLIC COMPUTATION,JANUARY(1998)

付録：Maple のソースコード

Grevill のアルゴリズム（厳密計算）

```
gre := proc(n)local M, A, a, b, c, d, X, i, l, m, z, y;
M := n;
(l, m) := LinearAlgebra[Dimension](M);
a[1] := M[1, 1.. − 1];
A[1] := a[1];
z := simplify(a[1].a[1]^%T);
if z <> 0 then
  X[1] := simplify(a[1]^%T/(a[1].a[1]^%T));
else
  X[1] := a[1]^%T;
end if;
for i from 2 to l do
  a[i] := M[i, 1.. − 1];
  A[i] :=< A[i − 1], a[i] >;
  d[i] := simplify(a[i].X[i − 1]);
  c[i] := simplify(a[i] − d[i].A[i − 1]);
  y[i] := simplify(c[i].c[i]^%T);
  if y[i] <> 0 then
    b[i] := simplify(c[i]/(c[i].c[i]^%T));
  else
    b[i] := simplify(d[i].X[i−1]^%T/(1+d[i].d[i]^%T));
  end if;
  X[i] :=< simplify(X[i − 1] − b[i]^%T.d[i])|b[i]^%T >;
end do;
end proc;
```

Grevill のアルゴリズム（ISCZ 法）

行列の区間化

$kukankaM := proc(a, b)$
$local M, m, n, i, j, N, A;$
$M := a;$
$m, n := LinearAlgebra[Dimension](M);$
$A := Matrix(m, n, 0);$
$for\ i\ to\ m\ do$
$for\ j\ to\ n\ do$
$N[i, j] := [[evalf(\frac{floor(M[i,j]*10^b)}{10^b}, b), evalf(\frac{ceil(M[i,j]*10^b)}{10^b}, b)], (i-1)*n+j];$
$A[i, j] := N[i, j];$
$end\ do;$
$end\ do;$
$return(A);$
$end\ proc;$

ゼロ書き換え

$zero := proc(a)$
$local\ x;$
$x := a;$
$if\ x[1]*x[2] <= 0\ then$
$x[1] := 0;$
$x[2] := 0;$
$return(x);$
$else$
$return(x);$
$end\ if;$
$end\ proc;$

区間演算

$ke2 := proc(a, b, c, d)$
$local\ x, y, z, n, v, A;$
$n := LinearAlgebra[RowDimension](d);$
$v := Vector(3, 0)^{\%T};$
$A := Matrix(n + 1, 3, 0);$
$x := a;$
$y := b;$
$z := 0;$
$if\ c = 1\ then$
$z := [zero([x[1][1] + y[1][1], x[1][2] + y[1][2]]), n + 1];$
$v[1] := x[2];$
$v[2] := y[2];$
$v[3] := -1;$
$elif\ c = 2\ then$
$z := [zero([x[1][1] - y[1][2], x[1][2] - y[1][1]]), n + 1];$
$v[1] := x[2];$
$v[2] := y[2];$
$v[3] := -2;$
$elif\ c = 3\ then$
$z := [zero([min(x[1][1]*y[1][1], x[1][1]*y[1][2], x[1][2]*y[1][1], x[1][2]*y[1][2]),$
$max(x[1][1] * y[1][1], x[1][1] * y[1][2], x[1][2] * y[1][1], x[1][2] * y[1][2])]), n + 1];$
$v[1] := x[2];$
$v[2] := y[2];$
$v[3] := -3;$
$elif\ c = 4\ then$
$z := [zero([min(\frac{x[1][1]}{y[1][1]}, \frac{x[1][1]}{y[1][2]}, \frac{x[1][2]}{y[1][1]}, \frac{x[1][2]}{y[1][2]}),$
$max(\frac{x[1][1]}{y[1][1]}, \frac{x[1][1]}{y[1][2]}, \frac{x[1][2]}{y[1][1]}, \frac{x[1][2]}{y[1][2]})]), n + 1];$
$v[1] := x[2];$

$v[2] := y[2];$
$v[3] := -4;$
$elif\ c = 5\ then$
$z := [zero([x[1][1] + y[1][1], x[1][2] + y[1][2]]), n + 1];$
$v[1] := y[2] + 1;$
$v[2] := y[2];$
$v[3] := -5;$
$end\ if;$
$A :=< d, v >;$
$return(z, A);$
$end\ proc;$

$ke := proc(a, b, c, d)$
$local\ x, y, z, f;$
$x := a;$
$y := b;$
$z := 0;$
$f := 0;$
$if\ c = 1\ then$
$z := zero([x[1][1] + y[1][1], x[1][2] + y[1][2]]);$
$elif\ c = 2\ then$
$z := zero([x[1][1] - y[1][2], x[1][2] - y[1][1]]);$
$elif\ c = 3\ then$
$z := zero([min(x[1][1]*y[1][1], x[1][1]*y[1][2], x[1][2]*y[1][1], x[1][2]*$
$y[1][2]),$
$max(x[1][1]*y[1][1], x[1][1]*y[1][2], x[1][2]*y[1][1], x[1][2]*y[1][2])]);$
$elif\ c = 4\ then$
$z := zero([min(\frac{x[1][1]}{y[1][1]}, \frac{x[1][1]}{y[1][2]}, \frac{x[1][2]}{y[1][1]}, \frac{x[1][2]}{y[1][2]}),$
$max(\frac{x[1][1]}{y[1][1]}, \frac{x[1][1]}{y[1][2]}, \frac{x[1][2]}{y[1][1]}, \frac{x[1][2]}{y[1][2]})]);$
$elif\ c = 5\ then$

```
  z := zero([x[1][1] + y[1][1], x[1][2] + y[1][2]]);
end if;
f := d + 1;
z := [z, f];
return(z, f);
end proc;

vmul3 := proc(a, b, c, d)
local x1, x2, y, z, i, n, v, A, e;
n := LinearAlgebra[RowDimension](d);
v := Matrix(2 * c - 1, 3, 0);
A := Matrix(n + 2 * c - 1, 3, 0);
x1 := a;
x2 := b;
for i to c do
  x1[i];
  x2[i];
  y[i][1] := min(x1[i][1][1] * x2[i][1][1], x1[i][1][1] * x2[i][1][2],
  x1[i][1][2] * x2[i][1][1], x1[i][1][2] * x2[i][1][2]);
  y[i][2] := max(x1[i][1][1] * x2[i][1][1], x1[i][1][1] * x2[i][1][2],
  x1[i][1][2] * x2[i][1][1], x1[i][1][2] * x2[i][1][2]);
  y[i] := [zero([y[i][1], y[i][2]]), n + i];
  v[i, 1] := x1[i][2];
  v[i, 2] := x2[i][2];
  v[i, 3] := -3;
end do;
z[1][1] := y[1][1][1] + y[2][1][1];
z[1][2] := y[1][1][2] + y[2][1][2];
z[1] := [zero([z[1][1], z[1][2]]), n + c + 1];
v[c + 1, 1] := y[1][2];
```

```
v[c+1,2] := y[2][2];
v[c+1,3] := -1;
if 2 < c then
  for i from 2 to c-1 do
    z[i][1] := z[i-1][1][1] + y[i+1][1][1];
    z[i][2] := z[i-1][1][2] + y[i+1][1][2];
    z[i] := [zero([z[i][1], z[i][2]]), n+c+i];
    v[c+i,1] := z[i-1][2];
    v[c+i,2] := y[i+1][2];
    v[c+i,3] := -1;
  end do;
end if;
A :=< d, v >;
return(z[c-1], A);
end proc;

vmul := proc(a, b, c, d)
local x1, x2, y, z, i, f;
v := Matrix(2*c-1, 3, 0);
x1 := a;
x2 := b;
for i to c do
  x1[i];
  x2[i];
  y[i][1] := min(x1[i][1][1]*x2[i][1][1], x1[i][1][1]*x2[i][1][2],
  x1[i][1][2]*x2[i][1][1], x1[i][1][2]*x2[i][1][2]);
  y[i][2] := max(x1[i][1][1]*x2[i][1][1], x1[i][1][1]*x2[i][1][2],
  x1[i][1][2]*x2[i][1][1], x1[i][1][2]*x2[i][1][2]);
  y[i] := [zero([y[i][1], y[i][2]]), d+i];
end do;
```

```
z[1][1] := y[1][1][1] + y[2][1][1];
z[1][2] := y[1][1][2] + y[2][1][2];
z[1] := [zero([z[1][1], z[1][2]]), d + c + 1];
if 2 < c then
  for i from 2 to c − 1 do
    z[i][1] := z[i − 1][1][1] + y[i + 1][1][1];
    z[i][2] := z[i − 1][1][2] + y[i + 1][1][2];
    z[i] := [zero([z[i][1], z[i][2]]), d + c + i];
  end do;
end if;
f := d + 2 ∗ c − 1;
return(z[c − 1], f);
end proc;

   シンボルリストの作成
L := proc(a)
local m, n, i, A, x, y;
(m, n) := LinearAlgebra[Dimension](a);
A := Matrix(m ∗ n, 3, 0);
x := 1;
y := 0;
for i to m ∗ n do
  if y <> n then
    y := y + 1;
  else
    y := 1;
    x := x + 1;
  end if;
  A[i, 1] := x;
  A[i, 2] := y;
```

```
end do;
return(A);
end proc;

   シンボルの復元
P := proc(a, b, c)
local x, y, z;
x := a[2];
y := b[x];
if y[3] = 0 then
  b[x, 1] := c[y[1], y[2]];
  b[x, 2] := 0;
  b[x, 3] := 1;
elif y[3] = −1 then
  if y[1] = y[2] then
    z[y[1]] := P([[0], y[1]], b, c);
    z[x] := simplify(2 ∗ z[y[1]][1]);
    b[x, 1] := z[x];
    b[x, 2] := 0;
    b[x, 3] := 1;
  else
    z[y[1]] := P([[0], y[1]], b, c);
    z[y[2]] := P([[0], y[2]], b, c);
    z[x] := simplify(z[y[1]][1] + z[y[2]][1]);
    b[x, 1] := z[x];
    b[x, 2] := 0;
    b[x, 3] := 1;
  end if;
elif y[3] = −2 then
 if y[1] = y[2] then
```

$z[y[1]] := P([[0], y[1]], b, c);$
$z[x] := simplify(z[y[1]][1] - z[y[1]][1]);$
$b[x, 1] := z[x];$
$b[x, 2] := 0;$
$b[x, 3] := 1;$
$else$
$z[y[1]] := P([[0], y[1]], b, c);$
$z[y[2]] := P([[0], y[2]], b, c);$
$z[x] := simplify(z[y[1]][1] - z[y[2]][1]);$
$b[x, 1] := z[x];$
$b[x, 2] := 0;$
$b[x, 3] := 1;$
$end\ if;$
$elif\ y[3] = -3\ then$
$if\ y[1] = y[2]\ then$
$z[y[1]] := P([[0], y[1]], b, c);$
$z[x] := simplify(z[y[1]][1] * z[y[1]][1]);$
$b[x, 1] := z[x];$
$b[x, 2] := 0;$
$b[x, 3] := 1;$
$else$
$z[y[1]] := P([[0], y[1]], b, c);$
$z[y[2]] := P([[0], y[2]], b, c);$
$z[x] := simplify(z[y[1]][1] * z[y[2]][1]);$
$b[x, 1] := z[x];$
$b[x, 2] := 0;$
$b[x, 3] := 1;$
$end\ if;$
$elif\ y[3] = -4\ then$
$if\ y[1] = y[2]\ then$

$z[y[1]] := P([[0], y[1]], b, c);$
$z[x] := simplify(\frac{z[y[1]][1]}{z[y[1]][1]});$
$b[x, 1] := z[x];$
$b[x, 2] := 0;$
$b[x, 3] := 1;$
$else$
$z[y[1]] := P([[0], y[1]], b, c);$
$z[y[2]] := P([[0], y[2]], b, c);$
$z[x] := simplify(\frac{z[y[1]][1]}{z[y[2]][1]});$
$b[x, 1] := z[x];$
$b[x, 2] := 0;$
$b[x, 3] := 1;$
$end\ if;$
$elif\ y[3] = -5\ then$
$z[y[2]] := P([[0], y[2]], b, c);$
$z[x] := simplify(1 + z[y[2]][1]);$
$b[x, 1] := z[x];$
$b[x, 2] := 0;$
$b[x, 3] := 1;$
$elif\ y[3] = 1\ then$
$end\ if;$
$return(b[x, 1], b);$
$end\ proc;$

Grevill のアルゴリズム（ISCZ 法）

$isczgre11 := proc(p)$
$local\ M, A, a, b, xb, yb, c, d, d1, X, i, j, r, n, m, z, y, ta, tb, tX, e, SL, f, Y,$
$q, k, k1, k2;$
$q := 3;$
$k := 0;$

```
(n, m) := LinearAlgebra[Dimension](p);
SL := L(p);
k1 := n * m;
k2 := n * m;
while k <> 2 do
  M := kukankaM(p, q);
  Digits := q;
  Y := Matrix(m, n, 0);
  a[1] := M[1, 1.. - 1];
  A[1] := a[1];
  ta[1] := a[1]^%T;
  if k1 > k2 then
    (z, k2) := vmul(a[1], a[1], m, k2);
  else
    (z, SL) := vmul3(a[1], a[1], m, SL);
  end if;
  X[1] := Vector(m, 0);
  while k = 0 do
    if z[1][1] = 0 and P(z, SL, p)[1] <> 0 then
      q := q + 1;
      k1 := LinearAlgebra[RowDimension](SL);
      k2 := m * n;
      break;
    end if;
    if z[1][1] <> 0 then
      for i to m do
        if k1 > k2 then
          (X[1][i], k2) := ke(ta[1][i], z, 4, k2);
        else
          (X[1][i], SL) := ke2(ta[1][i], z, 4, SL);
```

```
      end if;
    end do;
  else
    X[1] := ta[1];
  end if;
  tX[1] := X[1]^%T;
  for i from 2 to n do
    if i = 2 then
      d[i] := 0;
      f[i] := 0;
    else
      d[i] := Vector(i − 1, 0)^%T;
    end if;
    f[i] := Vector(m, 0)^%T;
    c[i] := Vector(m, 0)^%T;
    X[i] := Matrix(m, i, 0);
  end do;
  k := 1;
end do;
while k = 1 do
  for i from 2 to n do
    if k = 0 then
      break;
    end if;
    a[i] := M[i, 1.. − 1];
    A[i] :=< A[i − 1], a[i] >;
    if i = 2 then
      if k1 > k2 then
        (d[i], k2) := vmul(a[i], X[i − 1], m, k2);
      else
```

$(d[i], SL) := vmul3(a[i], X[i-1], m, SL);$
$end\ if;$
$if\ d[i][1][1] = 0\ and\ P(d[i], SL, p)[1] <> 0\ then$
$q := q + 1;$
$k := 0;$
$k1 := LinearAlgebra[RowDimension](SL);$
$k2 := m * n;$
$break;$
$else$
$k := 2;$
$end\ if;$
$else$
$for\ j\ to\ i-1\ do$
$if\ k1 > k2\ then$
$(d[i][j], k2) := vmul(a[i], X[i-1][1..-1, j], m, k2);$
$else$
$(d[i][j], SL) := vmul3(a[i], X[i-1][1..-1, j], m, SL);$
$end\ if;$
$if\ d[i][j][1][1] = 0\ and\ P(d[i][j], SL, p)[1] <> 0\ then$
$q := q + 1;$
$k := 0;$
$k1 := LinearAlgebra[RowDimension](SL);$
$k2 := m * n;$
$break;$
$else$
$k := 2$
$end\ if;$
$end\ do;$
$if\ k = 0\ then$
$break;$

```
  end if;
end if;
for j to m do
  if i = 2 then
    if k1 > k2 then
      (f[i][j], k2) := ke(d[i], A[i − 1][j], 3, k2);
      (c[i][j], k2) := ke(a[i][j], f[i][j], 2, k2);
    else
      (f[i][j], SL) := ke2(d[i], A[i − 1][j], 3, SL);
      (c[i][j], SL) := ke2(a[i][j], f[i][j], 2, SL);
    end if;
  else
    if k1 > k2 then
      (f[i][j], k2) := vmul(d[i], A[i − 1][1.. − 1, j], i − 1, k2);
      (c[i][j], k2) := ke(a[i][j], f[i][j], 2, k2);
    else
      (f[i][j], SL) := vmul3(d[i], A[i − 1][1.. − 1, j], i − 1, SL);
      (c[i][j], SL) := ke2(a[i][j], f[i][j], 2, SL);
    end if;
  end if;
  if f[i][j][1][1] = 0 and P(f[i][j], SL, p)[1] <> 0 then
    q := q + 1;
    k := 0;
    k1 := LinearAlgebra[RowDimension](SL);
    k2 := m ∗ n;
    break;
  else
    k := 2;
  end if;
  if c[i][j][1][1] = 0 and P(c[i][j], SL, p)[1] <> 0 then
```

$q := q + 1;$
$k := 0;$
$k1 := LinearAlgebra[RowDimension](SL);$
$k2 := m * n;$
$break;$
$else$
$k := 2;$
$end\ if;$
$end\ do;$
$if\ k = 0\ then$
$break;$
$end\ if;$
$y[i] := 0;$
$if\ k1 > k2\ then$
$(y[i], k2) := vmul(c[i], c[i], m, k2);$
$else$
$(y[i], SL) := vmul3(c[i], c[i], m, SL);$
$end\ if;$
$b[i] := Vector(m, 0)^{\%T};$
$if\ y[i][1][1] = 0\ and\ P(y[i], SL, p)[1] <> 0\ then$
$q := q + 1;$
$k := 0;$
$k1 := LinearAlgebra[RowDimension](SL);$
$k2 := m * n;$
$break;$
$else$
$k := 2;$
$end\ if;$
$if\ y[i][1][1] <> 0\ and\ P(y[i], SL, p)[1] <> 0\ then$
$for\ j\ to\ m\ do$

$if\ k1 > k2\ then$
$(b[i][j], k2) := ke(c[i][j], y[i], 4, k2);$
$else$
$(b[i][j], SL) := ke2(c[i][j], y[i], 4, SL);$
$end\ if;$
$if\ b[i][j][1][1] = 0\ and\ P(b[i][j], SL, p)[1] <> 0\ then$
$q := q + 1;$
$k := 0;$
$k1 := LinearAlgebra[RowDimension](SL);$
$k2 := m * n;$
$break;$
$else$
$k := 2;$
$end\ if;$
$end\ do;$
$if\ k = 0\ then$
$break;$
$end\ if;$
$else$
$d1[i] := 0;$
$xb[i] := Vector(m, 0)^{\%T};$
$yb[i] := 0;$
$if\ i = 2\ then$
$if\ k1 > k2\ then$
$(d1[i], k2) := ke(d[i], d[i], 3, k2);$
$(yb[i], k2) := ke([[1, 1], 0], d1[i], 5, k2);$
$else$
$(d1[i], SL) := ke2(d[i], d[i], 3, SL);$
$(yb[i], SL) := ke2([[1, 1], 0], d1[i], 5, SL);$
$end\ if;$

```
if d1[i][1][1] = 0 and P(d1[i], SL, p)[1] <> 0 then
  q := q + 1;
  k := 0;
  k1 := LinearAlgebra[RowDimension](SL);
  k2 := m * n;
  break;
else
  k := 2;
end if;
if yb[i][1][1] = 0 and P(yb[i], SL, p)[1] <> 0 then
  q := q + 1;
  k := 0;
  k1 := LinearAlgebra[RowDimension](SL);
  k2 := m * n;
  break;
else
  k := 2;
end if;
for j to m do
  if k1 > k2 then
    (xb[i][j], k2) := ke(d[i], tX[i - 1][j], 3, k2);
    (b[i][j], k2) := ke(xb[i][j], yb[i], 4, k2);
  else
    (xb[i][j], SL) := ke2(d[i], tX[i - 1][j], 3, SL);
    (b[i][j], SL) := ke2(xb[i][j], yb[i], 4, SL);
  endif;
  if b[i][j][1][1] = 0 and P(b[i][j], SL, p)[1] <> 0 then
    q := q + 1;
    k := 0;
    k1 := LinearAlgebra[RowDimension](SL);
```

$k2 := m * n;$
$break;$
$else$
$k := 2;$
$end\ if;$
$if\ xb[i][j][1][1] = 0\ and\ P(xb[i][j], SL, p)[1] <> 0\ then$
$q := q + 1;$
$k := 0;$
$k1 := LinearAlgebra[RowDimension](SL);$
$k2 := m * n;$
$break;$
$else$
$k := 2;$
$end\ if;$
$end\ do;$
$if\ k = 0\ then$
$break;$
$end\ if;$
$else$
$if\ k1 > k2\ then$
$(d1[i], k2) := vmul(d[i], d[i], i - 1, k2);$
$(yb[i], k2) := ke([[1, 1], 0], d1[i], 5, k2);$
$else$
$(d1[i], SL) := vmul3(d[i], d[i], i - 1, SL);$
$(yb[i], SL) := ke2([[1, 1], 0], d1[i], 5, SL);$
$end\ if;$
$if\ d1[i][1][1] = 0\ and\ P(d1[i], SL, p)[1] <> 0\ then$
$q := q + 1;$
$k := 0;$
$k1 := LinearAlgebra[RowDimension](SL);$

```
  k2 := m * n;
  break;
else
  k := 2;
end if;
if yb[i][1][1] = 0 and P(yb[i], SL, p)[1] <> 0 then
  q := q + 1;
  k := 0;
  k1 := LinearAlgebra[RowDimension](SL);
  k2 := m * n;
  break;
else
  k := 2;
end if;
for j to m do
  if k = 0 then
    break;
  end if;
  if k1 > k2 then
    (xb[i][j], k2) := vmul(d[i], tX[i - 1][1.. - 1, j], i - 1, k2);
    (b[i][j], k2) := ke(xb[i][j], yb[i], 4, k2);
  else
   (xb[i][j], SL) := vmul3(d[i], tX[i - 1][1.. - 1, j], i - 1, SL);
    (b[i][j], SL) := ke2(xb[i][j], yb[i], 4, SL);
  end if;
  if xb[i][j][1][1] = 0 and P(xb[i][j], SL, p)[1] <> 0 then
    q := q + 1;
    k := 0;
    k1 := LinearAlgebra[RowDimension](SL);
    k2 := m * n;
```

```
      break;
    else
      k := 2
    end if;
    if b[i][j][1][1] = 0 and P(b[i][j], SL, p)[1] <> 0 then
      q := q + 1;
      k := 0;
      k1 := LinearAlgebra[RowDimension](SL);
      k2 := m * n;
      break;
    else
      k := 2
    end if;
  end do;
  if k = 0 then
    break;
  end if;
 end if;
end if;
tb[i] := b[i]^%T;
e[i] := Matrix(m, i - 1, 0);
for j to m do
 if k = 0 then
   break;
 end if;
 for r to i - 1 do
   if i = 2 then
     if k1 > k2 then
       (e[i][j, r], k2) := ke(tb[i][j], d[i], 3, k2);
     else
```

```
    (e[i][j,r],SL) := ke2(tb[i][j],d[i],3,SL);
  end if;
else
  if k1 > k2 then
    (e[i][j,r],k2) := ke(tb[i][j],d[i][r],3,k2);
  else
    (e[i][j,r],SL) := ke2(tb[i][j],d[i][r],3,SL);
  end if;
end if;
if e[i][j][r][1][1] = 0 and P(e[i][j][r],SL,p)[1] <> 0 then
  q := q + 1;
  k := 0;
  k1 := LinearAlgebra[RowDimension](SL);
  k2 := m * n;
  break;
else
  k := 2;
end if;
if i = 2 then
  if k1 > k2 then
    (X[i-1][j],k2) := ke(X[i-1][j],e[i][j][1],2,k2);
  else
    (X[i-1][j],SL) := ke2(X[i-1][j],e[i][j][1],2,SL);
  end if;
if X[i-1][j][1][1] = 0 and P(X[i-1][j],SL,p)[1] <> 0 then
    q := q + 1;
    k := 0;
    k1 := LinearAlgebra[RowDimension](SL);
    k2 := m * n;
    break;
```

```
        else
          k := 2;
        end if;
          X[i][j] :=< X[i − 1][j]|tb[i][j] >;
      else
        if k1 > k2 then
          (X[i][j, r], k2) := ke(X[i − 1][j][r], e[i][j][r], 2, k2);
        else
          (X[i][j, r], SL) := ke2(X[i − 1][j][r], e[i][j][r], 2, SL);
        end if;
        if X[i][j][r][1][1] = 0 and P(X[i][j][r], SL, p)[1] <> 0 then
          q := q + 1;
          k := 0;
          k1 := LinearAlgebra[RowDimension](SL);
          k2 := m ∗ n;
          break;
        else
          k := 2;
        end if;
        X[i][j, i] := tb[i][j]
      end if;
    end do;
    if k = 0 then
      break;
    end if;
  end do;
  tX[i] := X[i]^%T;
  X[i];
 end do;
end do;
```

```
end do;
for i to m do
  for j to n do
    Y[i,j] := P(X[n][i][j], SL, p)[1];
  end do;
end do;
return(X[n], Y, q, SL);
end proc;
```

Karampetakis のアルゴリズム（厳密計算）

```
ginverse := proc(G)
local n, m, NewG, dum, NewGtran, tag, a, k, ID, Gt, i, A, B;
(n, m) := Dimension(G);
if n > m then
  NewG := Transpose(G);
  dum := n;
  n := m;
  m := dum;
  NewGtran := copy(G);
  tag := 1;
else
  NewG := copy(G);
  NewGtran := Transpose(G);
  tag := 0;
fi;
a := 1;
k := 0;
ID := IdentityMatrix(n);
B[k] := Matrix(n, n, 0);
B[k] := copy(ID);
```

```
Gt := NewG.NewGtran;
for i from 1 to n do
  B[i] := Matrix(n, n, 0);
    if i = 1 then
      if Trace(Gt) <> 0 then
        a := -(simplify(Trace(Gt)));
        B[i] := simplify(Gt + a.ID);
        k := 1;
      else
        B[i] := Gt;
      fi;
    else
      A := Gt.B[i - 1];
      if Trace(A) <> 0 then
        a := -(simplify((simplify((Trace(A))))/(i)));
        B[i] := simplify(A + a.ID);
        k := i;
      else
        B[i] := A;
      fi;
    fi;
od;
if tag = 1 then
  if k = 0 then
    return(Matrix(n, m, 0));
  else
    return(Transpose(-(simplify(simplify((1/(a)))
 * (simplify(NewGtran.B[k - 1]))))));
  fi;
else
```

```
  if k = 0 then
    return(Matrix(m, n, 0));
  else
   return(−(simplify(simplify((1/(a)))∗(simplify(NewGtran.B[k−
1])))));
  fi;
fi;
end;
```

Karampetakis のアルゴルズム（ISCZ 法）

```
行列の区間化
kukankaM1 := proc(a, b)
local M, m, n, i, j, N, A;
M := a;
(m, n) := LinearAlgebra[Dimension](M);
A := Matrix(m, n, 0);
for i to m do
  for j to n do
   N[i, j] := [[evalf(floor(M[i,j]∗10^b)/10^b, b), evalf(ceil(M[i,j]∗10^b)/10^b, b)], (i−1)∗
n + j];
    A[i, j] := N[i, j]
  end do;
end do;
return(A);
end proc;

   区間演算
ke2 := proc(a, b, c, d)
local x, y, z, n, v, A;
n := LinearAlgebra[RowDimension](d);
```

$v := Vector(3,0)^{\%T}$;
$A := Matrix(n+1,3,0)$;
$x := a$;
$y := b$;
$z := 0$;
$if\ c = 1\ then$
$z := [zero([x[1][1] + y[1][1], x[1][2] + y[1][2]]), n+1]$;
$v[1] := x[2]$;
$v[2] := y[2]$;
$v[3] := -1$;
$elif\ c = 2\ then$
$z := [zero([x[1][1] - y[1][2], x[1][2] - y[1][1]]), n+1]$;
$v[1] := x[2]$;
$v[2] := y[2]$;
$v[3] := -2$;
$elif\ c = 3\ then$
$z := [zero([min(x[1][1]{*}y[1][1], x[1][1]{*}y[1][2], x[1][2]{*}y[1][1], x[1][2]{*}y[1][2]),$
$max(x[1][1] * y[1][1], x[1][1] * y[1][2], x[1][2] * y[1][1], x[1][2] * y[1][2])]), n+1]$;
$v[1] := x[2]$;
$v[2] := y[2]$;
$v[3] := -3$;
$elif\ c = 4\ then$
$z := [zero([min(\frac{x[1][1]}{y[1][1]}, \frac{x[1][1]}{y[1][2]}, \frac{x[1][2]}{y[1][1]}, \frac{x[1][2]}{y[1][2]}),$
$max(\frac{x[1][1]}{y[1][1]}, \frac{x[1][1]}{y[1][2]}, \frac{x[1][2]}{y[1][1]}, \frac{x[1][2]}{y[1][2]})]), n+1]$;
$v[1] := x[2]$;
$v[2] := y[2]$;
$v[3] := -4$;
$elif\ c = 5\ then$

$z := [zero([x[1][1] + y[1][1], x[1][2] + y[1][2]]), n + 1];$
$v[1] := y[2] + 1;$
$v[2] := y[2];$
$v[3] := -5;$
$elif\ c = 6\ then$
$z := [zero([min(\frac{x[1][1]}{y[1][1]}, \frac{x[1][1]}{y[1][2]}, \frac{x[1][2]}{y[1][1]}, \frac{x[1][2]}{y[1][2]}),$
$max(\frac{x[1][1]}{y[1][1]}, \frac{x[1][1]}{y[1][2]}, \frac{x[1][2]}{y[1][1]}, \frac{x[1][2]}{y[1][2]})]), n + 1];$
$v[1] := y[2] + 1;$
$v[2] := y[2];$
$v[3] := -6;$
$elif\ c = 7\ then$
$z := [zero([min(x[1][1]*y[1][1], x[1][1]*y[1][2], x[1][2]*y[1][1], x[1][2]*$
$y[1][2]),$
$max(x[1][1] * y[1][1], x[1][1] * y[1][2], x[1][2] * y[1][1], x[1][2] *$
$y[1][2])]), n + 1];$
$v[1] := y[2] + 1;$
$v[2] := y[2];$
$v[3] := -7;$
$elif\ c = 8\ then$
$z := [zero([min(\frac{x[1][1]}{y[1][1]}, \frac{x[1][1]}{y[1][2]}, \frac{x[1][2]}{y[1][1]}, \frac{x[1][2]}{y[1][2]}),$
$max(\frac{x[1][1]}{y[1][1]}, \frac{x[1][1]}{y[1][2]}, \frac{x[1][2]}{y[1][1]}, \frac{x[1][2]}{y[1][2]})]), n + 1];$
$v[1] := x[2];$
$v[2] := y[2];$
$v[3] := -8;$
$end\ if;$
$A :=< d, v >;$
$return(z, A);$
$end\ proc;$

$vmul3 := proc(a, b, c, d)$

$local\ x1, x2, y, z, i, n, v, A, e;$
$(n) := LinearAlgebra[RowDimension](d);$
$v := Matrix(2 * c - 1, 3, 0);$
$A := Matrix(n + 2 * c - 1, 3, 0);$
$x1 := a;$
$x2 := b;$
$for\ i\ to\ c\ do$
$x1[i];$
$x2[i];$
$y[i][1] := min(x1[i][1][1] * x2[i][1][1], x1[i][1][1] * x2[i][1][2],$
$x1[i][1][2] * x2[i][1][1], x1[i][1][2] * x2[i][1][2]);$
$y[i][2] := max(x1[i][1][1] * x2[i][1][1], x1[i][1][1] * x2[i][1][2],$
$x1[i][1][2] * x2[i][1][1], x1[i][1][2] * x2[i][1][2]);$
$y[i] := [zero([y[i][1], y[i][2]]), n + i];$
$v[i, 1] := x1[i][2];$
$v[i, 2] := x2[i][2];$
$v[i, 3] := -3;$
$end\ do;$
$z[1][1] := y[1][1][1] + y[2][1][1];$
$z[1][2] := y[1][1][2] + y[2][1][2];$
$z[1] := [zero([z[1][1], z[1][2]]), n + c + 1];$
$v[c + 1, 1] := y[1][2];$
$v[c + 1, 2] := y[2][2];$
$v[c + 1, 3] := -1;$
$if\ 2 < c\ then$
$for\ i\ from\ 2\ to\ c - 1\ do$
$z[i][1] := z[i - 1][1][1] + y[i + 1][1][1];$
$z[i][2] := z[i - 1][1][2] + y[i + 1][1][2];$
$z[i] := [zero([z[i][1], z[i][2]]), n + c + i];$
$v[c + i, 1] := z[i - 1][2];$

```
    v[c + i, 2] := y[i + 1][2];
    v[c + i, 3] := −1;
  end do;
end if;
A :=< dv >;
return(z[c − 1], A);
end proc;

isTrace := proc(a, b)
local m, n, i, c, d;
(n, m) := Dimension(a);
d := b;
(c[1], d) := ke2(a[1, 1], a[2, 2], 1, d);
if n = 2 then
  return(c[1], d);
else
  for i from 3 to n do
    (c[i − 1], d) := ke2(c[i − 2], a[i, i], 1, d);
  end do;
return(c[n − 1], d);
end if;
end proc;

ke6 := proc(a, b, c)
local i, j, n, m, A, l;
(n, m) := Dimension(b);
l := c;
A := Matrix(n, m, 0);
for i from 1 to n do
  for j from 1 to m do
```

```
    (A[i, j], l) := ke2(a, b[i, j], 3, l);
  end do;
end do;
return(A, l);
end proc;

Mmul2 := proc(a, b, c)
local x, y, i, j, mx, my, nx, ny, M, N;
x := a;
y := b;
N := c;
(mx, nx) := LinearAlgebra[Dimension](x);
(my, ny) := LinearAlgebra[Dimension](y);
M := Matrix(mx, ny, 0);
for i to mx do
  for j to ny do
    (M[i, j], N) := vmul3(x[i, 1.. - 1], y[1.. - 1, j], nx, N);
  end do;
end do;
return(M, N);
end proc;

ke3 := proc(a, b, c, d)
local x, y, z, f;
x := a;
y := b;
z := 0;
f := 0;
if c = 1 then
  z := zero([x[1][1] + y[1][1], x[1][2] + y[1][2]]);
```

$elif\ c = 2\ then$
$z := zero([x[1][1] - y[1][2], x[1][2] - y[1][1]]);$
$elif\ c = 3\ then$
$z := zero([min(x[1][1]*y[1][1], x[1][1]*y[1][2], x[1][2]*y[1][1], x[1][2]*y[1][2]),$
$max(x[1][1]*y[1][1], x[1][1]*y[1][2], x[1][2]*y[1][1], x[1][2]*y[1][2])]);$
$elif\ c = 4\ then$
$z := zero([min(\frac{x[1][1]}{y[1][1]}, \frac{x[1][1]}{y[1][2]}, \frac{x[1][2]}{y[1][1]}, \frac{x[1][2]}{y[1][2]}),$
$max(\frac{x[1][1]}{y[1][1]}, \frac{x[1][1]}{y[1][2]}, \frac{x[1][2]}{y[1][1]}, \frac{x[1][2]}{y[1][2]})]);$
$elif\ c = 5\ then$
$z := zero([x[1][1] + y[1][1], x[1][2] + y[1][2]]);$
$elif\ c = 6\ then$
$z := zero([min(\frac{x[1][1]}{y[1][1]}, \frac{x[1][1]}{y[1][2]}, \frac{x[1][2]}{y[1][1]}, \frac{x[1][2]}{y[1][2]}),$
$max(\frac{x[1][1]}{y[1][1]}, \frac{x[1][1]}{y[1][2]}, \frac{x[1][2]}{y[1][1]}, \frac{x[1][2]}{y[1][2]})]);$
$elif\ c = 7\ then$
$z := zero([min(x[1][1]*y[1][1], x[1][1]*y[1][2], x[1][2]*y[1][1], x[1][2]*y[1][2]),$
$max(x[1][1]*y[1][1], x[1][1]*y[1][2], x[1][2]*y[1][1], x[1][2]*y[1][2])]);$
$elif\ c = 8\ then$
$z := zero([min(x[\frac{x[1][1]}{y[1][1]}, \frac{x[1][1]}{y[1][2]}, \frac{x[1][2]}{y[1][1]}, \frac{x[1][2]}{y[1][2]}),$
$max(\frac{x[1][1]}{y[1][1]}, \frac{x[1][1]}{y[1][2]}, \frac{x[1][2]}{y[1][1]}, \frac{x[1][2]}{y[1][2]})]);$
$end\ if;$
$f := d + 1;$
$z := [z, f];$
$return(z, f);$
$end\ proc;$

$vmul2 := proc(a, b, c, d)$
$local\ x1, x2, y, z, i, f;$
$v := Matrix(2 * c - 1, 3, 0);$

$x1 := a;$
$x2 := b;$
$for\ i\ to\ c\ do$
$x1[i];$
$x2[i];$
$y[i][1] := min(x1[i][1][1] * x2[i][1][1], x1[i][1][1] * x2[i][1][2],$
$x1[i][1][2] * x2[i][1][1], x1[i][1][2] * x2[i][1][2]);$
$y[i][2] := max(x1[i][1][1] * x2[i][1][1], x1[i][1][1] * x2[i][1][2],$
$x1[i][1][2] * x2[i][1][1], x1[i][1][2] * x2[i][1][2]);$
$y[i] := [zero([y[i][1], y[i][2]]), d + i];$
$end\ do;$
$z[1][1] := y[1][1][1] + y[2][1][1];$
$z[1][2] := y[1][1][2] + y[2][1][2];$
$z[1] := [zero([z[1][1], z[1][2]]), d + c + 1];$
$if\ 2 < c\ then$
$for\ i\ from\ 2\ to\ c - 1\ do$
$z[i][1] := z[i - 1][1][1] + y[i + 1][1][1];$
$z[i][2] := z[i - 1][1][2] + y[i + 1][1][2];$
$z[i] := [zero([z[i][1], z[i][2]]), d + c + i];$
$end\ do;$
$end\ if;$
$f := d + 2 * c - 1;$
$return(z[c - 1], f);$
$end\ proc;$

$Mmul3 := proc(a, b, c)$
$local\ x, y, i, j, mx, my, nx, ny, M, N;$
$x := a;$
$y := b;$
$N := c;$

```
(mx, nx) := LinearAlgebra[Dimension](x);
(my, ny) := LinearAlgebra[Dimension](y);
M := Matrix(mx, ny, 0);
for i to mx do
  for j to ny do
    (M[i, j], N) := vmul2(x[i, 1.. - 1], y[1.. - 1, j], nx, N);
  end do;
end do;
return(M, N);
end proc;

isTrace2 := proc(a, b)
local m, n, i, c, d;
(n, m) := Dimension(a);
d := b;
(c[1], d) := ke3(a[1, 1], a[2, 2], 1, d);
if n = 2 then
  return(c[1], d);
else
  for i from 3 to n do
    (c[i - 1], d) := ke3(c[i - 2], a[i, i], 1, d);
  end do;
  return(c[n - 1], d);
end if;
end proc;

ke7 := proc(a, b, c)
local i, j, n, m, A, l;
(n, m) := Dimension(b);
l := c;
```

```
A := Matrix(n, m, 0);
for i from 1 to n do
  for j from 1 to m do
    (A[i, j], l) := ke3(a, b[i, j], 3, l);
  end do;
end do;
return(A, l);
end proc;

   シンボルの復元
P := proc(a, b, c)
local x, y, z;
x := a[2];
y := b[x];
if y[3] = 0 then
  b[x, 1] := c[y[1], y[2]];
  b[x, 2] := 0;
  b[x, 3] := 1;
elif y[3] = −1 then
  if y[1] = y[2] then
    z[y[1]] := P([[0], y[1]], b, c);
    z[x] := simplify(2 ∗ z[y[1]][1]);
    b[x, 1] := z[x];
    b[x, 2] := 0;
    b[x, 3] := 1;
  else    z[y[1]] := P([[0], y[1]], b, c);
    z[y[2]] := P([[0], y[2]], b, c);
    z[x] := simplify(z[y[1]][1] + z[y[2]][1]);
    b[x, 1] := z[x];
    b[x, 2] := 0;
```

```
    b[x, 3] := 1;
  end if;
elif y[3] = −2 then
  if y[1] = y[2] then
    z[y[1]] := P([[0], y[1]], b, c);
    z[x] := simplify(z[y[1]][1] − z[y[1]][1]);
    b[x, 1] := z[x];
    b[x, 2] := 0;
    b[x, 3] := 1;
  else
    z[y[1]] := P([[0], y[1]], b, c);
    z[y[2]] := P([[0], y[2]], b, c);
    z[x] := simplify(z[y[1]][1] − z[y[2]][1]);
    b[x, 1] := z[x];
    b[x, 2] := 0;
    b[x, 3] := 1;
  end if;
elif y[3] = −3 then
  if y[1] = y[2] then
    z[y[1]] := P([[0], y[1]], b, c);
    z[x] := simplify(z[y[1]][1] ∗ z[y[1]][1]);
    b[x, 1] := z[x];
    b[x, 2] := 0;
    b[x, 3] := 1;
  else
    z[y[1]] := P([[0], y[1]], b, c);
    z[y[2]] := P([[0], y[2]], b, c);
    z[x] := simplify(z[y[1]][1] ∗ z[y[2]][1]);
    b[x, 1] := z[x];
    b[x, 2] := 0;
```

$b[x,3] := 1;$
$end\ if;$
$elif\ y[3] = -4\ then$
$if\ y[1] = y[2]\ then$
$z[y[1]] := P([[0], y[1]], b, c);$
$z[x] := simplify(\frac{z[y[1]][1]}{z[y[1]][1]});$
$b[x,1] := z[x];$
$b[x,2] := 0;$
$b[x,3] := 1;$
$else$
$z[y[1]] := P([[0], y[1]], b, c);$
$z[y[2]] := P([[0], y[2]], b, c);$
$z[x] := simplify(\frac{z[y[1]][1]}{z[y[2]][1]});$
$b[x,1] := z[x];$
$b[x,2] := 0;$
$b[x,3] := 1;$
$end\ if;$
$elif\ y[3] = -5\ then$
$z[y[2]] := P([[0], y[2]], b, c);$
$z[x] := simplify(1 + z[y[2]][1]);$
$b[x,1] := z[x];$
$b[x,2] := 0;$
$b[x,3] := 1;$
$elif\ y[3] = -6\ then$
$z[y[2]] := P([[0], y[2]], b, c);$
$z[x] := simplify(\frac{1}{z[y[2]][1]});$
$b[x,1] := z[x];$
$b[x,2] := 0;$
$b[x,3] := 1;$
$elif\ y[3] = -7\ then$

$z[y[2]] := P([[0], y[2]], b, c);$
$z[x] := simplify(-1 * z[y[2]][1]);$
$b[x, 1] := z[x];$
$b[x, 2] := 0;$
$b[x, 3] := 1;$
$elif\ y[3] = -8\ then$
$z[y[1]] := P([[0], y[1]], b, c);$
$z[x] := simplify(\frac{z[y[1]][1]}{-b[x,2]});$
$b[x, 1] := z[x];$
$b[x, 2] := 0;$
$b[x, 3] := 1;$
$elif\ y[3] = 1\ then$
$end\ if;$
$return(b[x, 1], b);$
$end\ proc;$

Karampetakis のアルゴルズム（ISCZ 法）

$isczginverse := proc(G)$
$local\ n, m, NewG, dum, NewGtran, tag, a, k, ID, Gt, i, A, B, M, j, a1,$
$z, r, a2, a3, a4, a5, d, SL, Y, k1, k2, k3;$
$d := 3;$
$k1 := 0;$
$k2 := 0;$
$k3 := 0;$
$SL := L(G);$
$while\ k1 = 0\ do$
$(n, m) := Dimension(G);$
$Digits := d;$
$M := kukankaM1(G, d);$
$if\ n > m\ then$

$NewG := Transpose(M);$
$dum := n;$
$n := m;$
$m := dum;$
$NewGtran := copy(M);$
$tag := 1;$
$else$
$NewG := copy(M);$
$NewGtran := Transpose(M);$
$tag := 0;$
$fi;$
$a := 1;$
$k := 0;$
$ID := iD(n);$
$B[k] := Matrix(n, n, 0);$
$B[k] := copy(ID);$
$if\ k2 > k3\ then$
$(Gt, k3) := Mmul3(NewG, NewGtran, k3);$
$else$
$(Gt, SL) := Mmul2(NewG, NewGtran, SL);$
$endif;$
$for\ i\ from\ 1\ to\ n\ do$
$if\ k1 = 1\ then$
$break;$
$end\ if;$
$for\ j\ from\ 1\ to\ n\ do$
$if\ Gt[i][j][1][1] = 0\ and\ P(A[i][j], SL, G)[1] <> 0\ then$
$d := d + 1;$
$k1 := 1;$
$k2 := RowDimension(SL);$

```
    k3 := m * n;
    break;
  else
    k1 := 2;
  end if;
 end do;
end do;
for i from 1 to n do
  if k1 = 1 then
    break;
  end if;
  B[i] := Matrix(n, n, 0);
  if i = 1 then
    if k2 > k3 then
      (z, k3) := isTrace2(Gt, k3);
    else
      (z, SL) := isTrace(Gt, SL);
    end if;
    if z[1][1] = 0 and P(z, SL, G)[1] <> 0 then
      d := d + 1;
      k1 := 1;
      k2 := RowDimension(SL);
      k3 := m * n;
      break;
    else
      k1 := 2;
    end if;
    if z[1][1] <> 0 and P(z, SL, G)[1] <> 0 then
      if k2 > k3 then
        (a, k3) := ke3([[-1, -1], 0], z, 7, k3);
```

$else$
$(a, SL) := ke2([[-1, -1], 0], z, 7, SL);$
$endif;$
$if\ a[1][1] = 0\ and\ P(a, SL, G)[1] <> 0\ then$
$d := d + 1;$
$k1 := 1;$
$k2 := RowDimension(SL);$
$k3 := m * n;$
$break;$
$else$
$k1 := 2;$
$end\ if;$
$B[i] := copy(Gt);$
$for\ j\ from\ 1\ to\ n\ do$
$if\ k1 = 1\ then$
$break;$
$end\ if;$
$if\ k2 > k3\ then$
$(B[i][j, j], k3) := ke3(a, B[i][j, j], 1, k3);$
$else$
$(B[i][j, j], SL) := ke2(a, B[i][j, j], 1, SL);$
$end\ if;$
$if\ B[i][j][j][1][1] = 0\ and\ P(B[i][j][j], SL, G)[1] <> 0\ then$
$d := d + 1;$
$k1 := 1;$
$k2 := RowDimension(SL);$
$k3 := m * n;$
$break;$
$else$
$k1 := 2;$

```
      end if;
    end do;
    k := 1;
  else
    B[i] := copy(Gt);
  fi;
else
if k1 = 1 then
  break;
end if;
if k2 > k3 then
  (A, k3) := Mmul3(Gt, B[i − 1], k3);
else
  (A, SL) := Mmul2(Gt, B[i − 1], SL);
end if;
for j from 1 to n do
  if k1 = 1 then
    break;
  end if;
  for r from 1 to n do
    if A[j][r][1][1] = 0 and P(A[j][r], SL, G)[1] <> 0 then
      d := d + 1;
      k1 := 1;
      k2 := RowDimension(SL);
      k3 := m ∗ n;
      break;
    else
      k1 := 2;
    end if;
  end do;
```

$end\ do;$
$if\ k1 = 1\ then$
$break;$
$end\ if;$
$if\ k2 > k3\ then$
$(z, k3) := isTrace2(A, k3);$
$else$
$(z, SL) := isTrace(A, SL);$
$end\ if;$
$if\ z[1][1] = 0\ and\ P(z, SL, G)[1] <> 0\ then$
$d := d + 1;$
$k1 := 1;$
$k2 := RowDimension(SL);$
$k3 := m * n;$
$break;$
$else$
$k1 := 2;$
$end\ if;$
$if\ z[1][1] <> 0\ and\ P(z, SL, G)[1] <> 0\ then$
$if\ k2 > k3\ then$
$(a1, k3) := ke3([[-1, -1], 0], z, 7, k3);$
$else$
$(a1, SL) := ke2([[-1, -1], 0], z, 7, SL);$
$end\ if;$
$if\ a1[1][1] = 0\ and\ P(a1, SL, G)[1] <> 0\ then$
$d := d + 1;$
$k1 := 1;$
$k2 := RowDimension(SL);$
$k3 := m * n;$
$break;$

$else$
$k1 := 2;$
$end\ if;$
$if\ k2 > k3\ then$
$(a, k3) := ke3(a1, [[i, i], -i], 8, k3);$
$else$
$(a, SL) := ke2(a1, [[i, i], -i], 8, SL);$
$end\ if;$
$if\ a[1][1] = 0\ and\ P(a, SL, G)[1] <> 0\ then$
$d := d + 1;$
$k1 := 1;$
$k2 := RowDimension(SL);$
$k3 := m * n;$
$break;$
$else$
$k1 := 2;$
$end\ if;$
$B[i] := copy(A);$
$for\ j\ from\ 1\ to\ n\ do$
$if\ k1 = 1\ then$
$break;$
$end\ if;$
$if\ k2 > k3\ then$
$(B[i][j, j], k3) := ke3(a, B[i][j, j], 1, k3);$
$else$
$(B[i][j, j], SL) := ke2(a, B[i][j, j], 1, SL);$
$end\ if;$
$if\ B[i][j][j][1][1] = 0\ and\ P(B[i][j][j], SL, G)[1] <> 0\ then$
$d := d + 1;$
$k1 := 1;$

```
        k2 := RowDimension(SL);
        k3 := m * n;
        break;
      else
        k1 := 2;
      end if;
    end do;
      k := i;
  else
    B[i] := copy(A);
  fi;
 fi;
od;
if k1 <> 1 then
 if tag = 1 then
  if k = 0 then
   return(Matrix(n, m, 0));
  else
   if k2 > k3 then
    (a5, k3) := ke3([[−1, −1], 0], a, 7, k3);
   else
    (a5, SL) := ke2([[−1, −1], 0], a, 7, SL);
   end if;
   if a5[1][1] = 0 and P(a5, SL, G)[1] <> 0 then
    d := d + 1;
    k1 := 1;
    k2 := RowDimension(SL);
    k3 := m * n;
    break;
   else
```

$k1 := 2;$
$end\ if;$
$if\ k2 > k3\ then$
$(a2, k3) := ke3([[1, 1], 0], a5, 6, k3);$
$else$
$(a2, SL) := ke2([[1, 1], 0], a5, 6, SL);$
$end\ if;$
$if\ a2[1][1] = 0\ and\ P(a2, SL, G)[1] <> 0\ then$
$d := d + 1;$
$k1 := 1;$
$k2 := RowDimension(SL);$
$k3 := m * n;$
$break;$
$else$
$k1 := 2;$
$end\ if;$
$if\ k2 > k3\ then$
$(a3, k3) := ke7(a2, NewGtran, k3);$
$else$
$(a3, SL) := ke6(a2, NewGtran, SL);$
$end\ if;$
$for\ ifrom\ 1\ to\ m\ do$
$if\ k1 = 1\ then$
$break;$
$end\ if;$
$for\ j\ from\ 1\ to\ n\ do$
$if\ a3[i][j][1][1] = 0\ and\ P(a3[i][j], SL, G)[1] <> 0\ then$
$d := d + 1;$
$k1 := 1;$
$k2 := RowDimension(SL);$

```
      k3 := m * n;
      break;
    else
      k1 := 2;
    end if;
  end do;
end do;
if k2 > k3 then
  (a4, k3) := Mmul3(a3, B[k − 1], k3);
else
  (a4, SL) := Mmul2(a3, B[k − 1], SL);
end if;
for i from 1 to m do
  if k1 = 1 then
    break;
  endif;
  for j from 1 to n do
    if a4[i][j][1][1] = 0 and P(a4[i][j], SL, G)[1] <> 0 then
      d := d + 1;
      k1 := 1;
      k2 := RowDimension(SL);
      k3 := m * n;
      break;
    else
      k1 := 2;
    end if;
  end do;
end do;
Y := Matrix(m, n, 0);
for i from 1 to m do
```

```
      if k1 = 1 then
        break;
      end if;
      for j from 1 to n do
        Y[i, j] := P(a4[i][j], SL, G)[1];
      end do;
    end do;
    if k1 = 2 then
      return(a4^%T, Y^%T, SL, d);
    end if;
  fi;
else
  if k = 0 then
    return(Matrix(m, n, 0));
  else
    if k2 > k3 then
      (a5, k3) := ke3([[−1, −1], 0], a, 7, k3);
    else
      (a5, SL) := ke2([[−1, −1], 0], a, 7, SL);
    end if;
    if a5[1][1] = 0 and P(a5, SL, G)[1] <> 0 then
      d := d + 1;
      k1 := 1;
      k2 := RowDimension(SL);
      k3 := m ∗ n;
      break;
    else
      k1 := 2;
    end if;
    if k2 > k3 then
```

$(a2, k3) := ke3([[1,1],0], a5, 6, k3);$
$else$
$(a2, SL) := ke2([[1,1],0], a5, 6, SL);$
$end\ if;$
$if\ a2[1][1] = 0\ and\ P(a2, SL, G)[1] <> 0\ then$
$d := d + 1;$
$k1 := 1;$
$k2 := RowDimension(SL);$
$k3 := m * n;$
$break;$
$else$
$k1 := 2;$
$end\ if;$
$if\ k2 > k3\ then$
$(a3, k3) := ke7(a2, NewGtran, k3);$
$else$
$(a3, SL) := ke6(a2, NewGtran, SL);$
$end\ if;$
$for\ i\ from\ 1\ to\ m\ do$
$if\ k1 = 1\ then$
$break;$
$end\ if;$
$for\ j\ from\ 1\ to\ n\ do$
$if\ a3[i][j][1][1] = 0\ and\ P(a3[i][j], SL, G)[1] <> 0\ then$
$d := d + 1;$
$k1 := 1;$
$k2 := RowDimension(SL);$
$k3 := m * n;$
$break;$
$else$

```
      k1 := 2;
    end if;
  end do;
end do;
if k2 > k3 then
  (a4, k3) := Mmul3(a3, B[k − 1], k3);
else
  (a4, SL) := Mmul2(a3, B[k − 1], SL);
end if;
for i from 1 to m do
  if k1 = 1 then
    break;
  end if;
  for j from 1 to n do
    if a4[i][j][1][1] = 0 and P(a4[i][j], SL, G)[1] <> 0 then
      d := d + 1;
      k1 := 1;
      k2 := RowDimension(SL);
      k3 := m ∗ n;
      break;
    else
      k1 := 2;
    end if;
  end do;
end do;
Y := Matrix(m, n, 0);
for i from 1 to m do
  if k1 = 1 then
    break;
  end if;
```

```
          for jfrom 1 to n do
            Y[i, j] := P(a4[i][j], SL, G)[1];
          end do;
        end do;
        if k1 = 2 then
          return(a4, Y, SL, d);
        end if;
      fi;
    fi;
  end if;
  if k1 = 1 then
    k1 := 0;
  end if;
end do;
end;
```