

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

照井 章*

AKIRA TERUI

筑波大学大学院 数理物質科学研究科

GRADUATE SCHOOL OF PURE AND APPLIED SCIENCES, UNIVERSITY OF TSUKUBA

田島 慎一†

SHIN-ICHI TAJIMA

筑波大学大学院 数理物質科学研究科

GRADUATE SCHOOL OF PURE AND APPLIED SCIENCES, UNIVERSITY OF TSUKUBA

Abstract

本稿では、整数を要素にもつ正方行列とその特性多項式が与えられており、注目している固有値の重複度が 1 であるときに、その固有値に属する固有ベクトルを固有値の多項式として厳密に計算する効率的な算法を提案する。本算法は、著者 (田島) がこれまでに行ってきたレゾルベントの留数解析に基づき、行列の最小消去多項式候補を用いるものであり、固有ベクトル候補の計算を先に行い、引き続いて最小消去多項式のチェックを行うという特徴をもつ。さらに、行列・ベクトル積の Horner 法を用いることにより、さらなる計算の効率化を図っている。

1 はじめに

これまでに、著者 (田島) ら [4] は、行列とその特性多項式が与えられており、かつ特性多項式が既約で、すべての固有値が相異なる場合に、固有ベクトルを固有値の多項式として厳密に計算する効率的な算法を与えた。行列のスペクトル分解の理論により、着目している固有値に属する固有ベクトルは、その固有値におけるレゾルベントの留数値から求まる射影行列の列ベクトルによって与えられるが、この算法では、留数計算を、特性多項式を用いた多項式の計算に帰着させて行うことにより、行列のスペクトル分解を並列に計算するとともに、固有ベクトルを従来より大幅に効率的に計算可能であることを示した。

これと並行して、著者 (田島) [3] は、レゾルベントの留数解析に基づき、行列の最小消去多項式を用いて、行列のスペクトル分解を効率的に求める算法を与えた。この算法では、行列の最小多項式を用いる場合と比較して、行列のスペクトル分解をより効率的に計算可能になる。さらに、行列の最小消去多項式の計算をより効率化するため、最小消去多項式候補を効率的に計算した上で最小消去多項式を計算する算法を提案している。

本稿では、これらの研究成果に基づき、行列の最小消去多項式候補を用いて、行列の固有ベクトルを効率的に計算する算法を提案する。本算法の特徴の一つは、固有ベクトル候補の計算を先に行い、引き続いて最小消去多項式のチェックを行っていることである。さらに、実際の算法においては、行列・ベクトル積の Horner 法を用いることにより、さらなる計算の効率化を図っている。

*terui@math.tsukuba.ac.jp

†tajima@math.tsukuba.ac.jp

2 問題設定と固有ベクトルの算法

行列 A を、整数を要素にもつ n 次正方形行列とし、 E を n 次単位行列とする。 A の特性多項式 $\chi_A(\lambda)$ は次式の形で、整数上の既約因数分解をあらかじめ求めているものとする。

$$\chi_A(\lambda) = f_1(\lambda)^{m_1} f_2(\lambda)^{m_2} \cdots f_p(\lambda)^{m_p} \cdots f_q(\lambda)^{m_q}. \quad (1)$$

本稿で提案する算法の目的は、式 (1) のある既約因子 $f_p(\lambda)$ ($1 \leq p \leq q$) に対し、 $f_p(\alpha) = 0$ をみたま A の固有値 $\lambda = \alpha$ に属する固有ベクトルを求めることである。なお、本稿では、 $m_p = 1$ 、すなわち、注目する固有値の重複度は 1 と仮定する (従来 χ_A 全体が既約である必要はない)。

$e_j = {}^t(0, \dots, 0, 1, 0, \dots, 0)$ を、第 j 成分が 1 に等しい n 次単位ベクトルとする。 A の第 j 列の最小消去多項式 $\pi_{A,j}(\lambda)$ は、イデアル $\{P(\lambda) \mid P(A)e_j = 0\}$ のモノックな生成元として定義される。ここで、 $\pi_{A,j}(\lambda)$ は $f_p(\lambda)$ を因子にもつと仮定し、 $\pi_{A,j}(\lambda) = f_p(\lambda) \cdot g_{j,p}(\lambda)$ と表す (すなわち、 $g_{j,p}(\lambda)$ は $f_p(\lambda)$ 以外の $\pi_{A,j}(\lambda)$ の因子の積である)。

$f_p(\lambda)$ に対し、 $\Psi_p(x, y)$ を次式のように定義する。

$$\Psi_p(x, y) = \frac{f_p(x) - f_p(y)}{x - y}. \quad (2)$$

$\Psi_p(x, y)$ は多項式であることに注意する。

このとき、以下の命題が成り立つ。

命題 1

$\chi_A(\lambda)$, $f_p(\lambda)$, $\Psi_p(x, y)$, $\pi_{A,j}(\lambda)$ を上記で与えられる多項式とする。このとき、列ベクトル $\rho(\lambda)$ を

$$\rho(\lambda) = \Psi_p(A, \lambda E) g_{j,p}(A) e_j$$

によって定めると、 $f_p(\alpha) = 0$ をみたま A の固有値 $\lambda = \alpha$ に対し、列ベクトル $\rho(\alpha)$ は

$$A \cdot \rho(\alpha) = \alpha \cdot \rho(\alpha),$$

をみたま。すなわち $\rho(\alpha)$ は A の固有値 $\lambda = \alpha$ に属する固有ベクトルである。 ■

さて、本稿の算法では、最小消去多項式候補を用いて固有ベクトル計算を行う。よって、与えられた最小消去多項式候補が、実際に最小消去多項式であることを確認する必要がある。本稿で提案する最小消去多項式候補を用いた固有ベクトル計算は、以下の流れになる。

1. [固有ベクトル候補の計算] 注目している A の固有値 $\lambda = \alpha$, A の第 j 列の最小消去多項式候補 $P_j(\lambda) = f_p(\lambda) \cdot g_{j,p}(\lambda)$ に対し

$$\rho(\alpha) = \Psi_p(A, \alpha E) g_{j,p}(A) e_j \quad (3)$$

を計算する。

2. [最小消去多項式のチェック] $P_j(\lambda)$ が A の第 j 列の最小消去多項式になるかどうかをチェックする。具体的には

$$P_j(A) e_j = f_p(A) g_{j,p}(A) e_j \quad (4)$$

が 0 に等しくなることを確かめる。

もし (4) が成り立つのであれば、(3) の $\rho(\alpha)$ を A の固有ベクトルとして出力する。

上記の手順の中で、固有ベクトル候補 $\rho(\alpha)$ を先に計算するのは、以下の理由による。(3) の $\Psi_p(A, \alpha E)$ から、Horner 法の 1 ステップの計算のみで、(4) の $f_p(A)$ が導かれる。この際、行列・ベクトル積の Horner 法を用いることで、固有ベクトル計算の効率化が可能になる。以下に詳細を述べる。

3 行列・ベクトル積の Horner 法による計算の効率化

本章では、(3) の $\rho(\alpha)$ の計算、および (4) の検算の効率化について説明する。

式 (3) の右辺に現われる $\Psi_p(A, \alpha E)$ と $g_{j,p}(A)$ は、それぞれ多項式に行列を代入した“行列多項式”であるが、本算法では、これらの多項式やその積を先に求める代わりに、右辺の $g_{j,p}(A)$ と e_j から行列・ベクトル積を計算することで、計算の効率化を図る。

以下では、例題を用いて具体的な計算手順を示す。なお、行列やベクトル積の計算量は、数体上の演算に基づく素朴な計算量で表す。

例 1

本章の以下の例題では、 A を n 次正方行列、 E を n 次単位行列とする。 $g_{j,p}(\lambda)$ を 4 次多項式

$$g_{j,p}(\lambda) = \lambda^4 + a_3\lambda^3 + a_2\lambda^2 + a_1\lambda + a_0, \quad (5)$$

$f_p(\lambda)$ を 4 次多項式

$$f_p(\lambda) = \lambda^4 + b_3\lambda^3 + b_2\lambda^2 + b_1\lambda + b_0 \quad (6)$$

とする。■

3.1 $g_{j,p}(A) e_j$ の計算

例 2

式 (5) より、 $g_{j,p}(A)$ を行列積による Horner 法で計算すると

$$g_{j,p}(A) = A(A(A(A + a_3E) + a_2E) + a_1E) + a_0E$$

より、その計算量は $O(n^3 \cdot \deg(g_{j,p}(\lambda)))$ となる。

一方、 $g_{j,p}(A) e_j$ を行列・ベクトル積による Horner 法で計算すると

$$\begin{aligned} g_{j,p}(A)e_j &= A^4e_j + a_3A^3e_j + a_2A^2e_j + a_1Ae_j + a_0e_j \\ &= A(A(A(Ae_j + a_3e_j) + a_2e_j) + a_1e_j) + a_0e_j \end{aligned} \quad (7)$$

より、その計算量は

$$O(n^2 \cdot \deg(g_{j,p}(\lambda))) \quad (8)$$

となる。行列・ベクトル積による Horner 法では、行列積による Horner 法と比較して、計算量の n の指数が 1 減少していることに注意する。■

3.2 $\Psi_p(A, \alpha E)$ から $f_p(A)$ の導出

例 3

$f_p(\lambda)$ は (6) で与えられた通りである。式 (2) より

$$\Psi_p(x, y) = \frac{(f_p(x) - f_p(y))}{x - y} = y^3 + (x + b_3)y^2 + (x^2 + b_3x + b_2)y + (x^3 + b_3x^2 + b_2x + b_1)$$

が成り立つので、 $\Psi_p(x, y)$ の x, y にそれぞれ $A, \alpha E$ を代入すると

$$\Psi_p(A, \alpha E) = \alpha^3E + \alpha^2(A + b_3E) + \alpha(A^2 + b_3A + b_2E) + (A^3 + b_3A^2 + b_2A + b_1E)$$

となる。ここで、 α の係数に $f_p(A)$ が高次項から順に現われる点に注意。そこで、 $\Psi_p(A, \alpha E)$ の α^0 の係数 $A^3 + b_3A^2 + b_2A + b_1E$ に対し、Horner 法の最後の 1 ステップとして

$$A(A^3 + b_3A^2 + b_2A + b_1E) + b_0E \quad (9)$$

を計算することにより、 $f_p(A)$ が求まる。

これにより、 $\Psi_p(A, \alpha E)$ の α の係数は、 $f_p(A)$ に対する Horner 法のステップにより計算できることに注意する。 ■

3.3 $\rho(\alpha)$ の計算と $P_j(\lambda)$ が A の第 j 列の最小消去多項式になるかどうかの検算

例 4

例 2 に引き続き、 $\rho(\alpha)$ の計算を行う。式 (7) で計算される $g_{j,p}(A)e_j$ に対し

$$\varphi_{j,p} = g_{j,p}(A)e_j \quad (10)$$

とおく。例 3 より、式 (3) の固有ベクトル候補 $\rho(\alpha) = \Psi_p(A, \alpha E) \varphi_{j,p}$ は以下の通り計算できる。

$$\Psi_p(A, \alpha E) \varphi_{j,p} = \alpha^3 \varphi_{j,p} + \alpha^2 (A + b_3E) \varphi_{j,p} + \alpha (A^2 + b_3A + b_2E) \varphi_{j,p} + (A^3 + b_3A^2 + b_2A + b_1E) \varphi_{j,p}.$$

このとき、 α^i の係数を、以下の通り、 α^{i+1} の係数を用いて、行列・ベクトル積の Horner 法で計算できる。

$$\begin{aligned} \alpha^2 &: (A + b_3E) \varphi_{j,p}, \\ \alpha^1 &: A(A + b_3E) \varphi_{j,p} + b_2 \varphi_{j,p}, \\ \alpha^0 &: A(A(A + b_3E) \varphi_{j,p} + b_2 \varphi_{j,p}) + b_1 \varphi_{j,p}. \end{aligned} \quad (11)$$

以上により、 $\rho(\alpha)$ の計算量は

$$O(n^2 \cdot \deg(f_p(\lambda))) \quad (12)$$

となる。

式 (11) の α^0 の係数に対し、Horner 法の最後の 1 ステップ (9) を用いると、(4), (10) により、

$$\begin{aligned} P_j(A) e_j &= f_p(A) g_{j,p}(A) e_j = f_p(A) \varphi_{j,p} \\ &= A(A(A(A + a_3 \varphi_{j,p} E) + a_2 \varphi_{j,p} E) + a_1 \varphi_{j,p} E) + a_0 \varphi_{j,p} E \end{aligned}$$

が得られる。これにより、 $P_j(\lambda)$ が A の第 j 列の最小消去多項式になるかどうかの検算が、行列・ベクトル積による Horner 法を用いて効率的に計算される。 ■

3.4 計算量解析

固有ベクトル 1 個あたりの計算量は、以下の通り見積もることができる。

式 (3) の固有ベクトル候補 $\rho(\alpha)$ の計算量は、(8), (12) より

$$O(n^2 \cdot \deg(g_{j,p}(\lambda))) + O(n^2 \cdot \deg(f_p(\lambda))) = O(n^2 (\deg(g_{j,p}) + \deg(f_p))) = O(n^2 \cdot \deg(P_j))$$

となる。一方、 $P_j(\lambda)$ が A の第 j 列の最小消去多項式になるかどうかの検算の計算量は、Horner 法の 1 ステップ (9) により、 $O(n^2)$ となる。以上により、固有ベクトル 1 個あたりの計算量は

$$O(n^2 \cdot \deg(P_j)) \quad (13)$$

となる.

この結果より, 固有ベクトルを効率的に計算するためには, 以下の通り, 最小消去多項式候補の選び方が重要であることがわかる.

1. 次数がより小さなものを選ぶ.
2. 次数が等しいものが複数ある場合は, 係数がより小さなものを選ぶ.

4 実験

上述の固有ベクトルの算法を数式処理システム Risa/Asir に実装し, 実験を行った. 固有ベクトル計算に必要な諸計算のうち, 特性多項式の計算は木村欣司氏による実装 ([1], [2]) を, 最小消去多項式候補の計算は小原功任氏による実装 (Risa/Asir) をそれぞれ利用した.

実験環境は以下の通り: Intel Core2 Duo T7400 (Apple MacBook “Mid-2007” model) at 2.16 GHz, RAM 2GB, MacOS X 10.6.

本稿の実験では, 上記の計算量解析より, 以下の点について評価することを目的とした.

1. 最小消去多項式候補の次数が一定の状況の下での, 行列の次元の増加に対する固有ベクトルの計算量の増加の振る舞い.
2. 行列の次元が一定の状況の下での, 最小消去多項式候補の次数の増加に対する固有ベクトルの計算量の増加の振る舞い.

4.1 実験 1: 行列の次元の増加に伴う固有ベクトルの計算量

本実験では, 計算される最小消去多項式候補の次数が一定の下で, 行列の次元の増加に対する固有ベクトルの計算量の増加を調べた.

4.1.1 テスト用行列の生成と実験

本実験では, 固有ベクトルを計算するためのテスト用行列を以下の手順で生成した.

行列 A_s, B_t をそれぞれ s 次, t 次の正方行列とし, 行列 $M_{s,t}$ を $M_{s,t} = \begin{pmatrix} A_s & \mathbf{O} \\ \mathbf{O} & B_t \end{pmatrix}$ で定まるブロック対角行列とする.

$M_{s,t}$ の定義より $\chi_{M_{s,t}}(\lambda) = \chi_{A_s}(\lambda)\chi_{B_t}(\lambda)$ が成り立つ. また, $j = 1, \dots, s$ に対し, $M_{s,t}$ の第 j 列の最小消去多項式 $\pi_{M_{s,t},j}(\lambda)$ は $\chi_{A_s}(\lambda)$ に等しい. そこで, A_s を固定し, B_t のサイズを変化させた上で, $j = 1, \dots, s$ に対し, $M_{s,t}$ の第 j 列の最小消去多項式候補を計算し, $\chi_{A_j}(\alpha) = 0$ をみたす $M_{s,t}$ の固有値 α に属する固有ベクトルを求めることにする.

実験では, $s = 20$, すなわち A_s は 20 次の正方行列として固定した. 一方で, $t \in \{20, 40, 60, 80, 100, 120, 140, 160, 180\}$ とし, 結果として, $M_{s,t}$ として, 次元が 20 次から 200 次まで 20 次おきに増加する合計 10 個の行列を与えた. A_s や B_t の要素は, 絶対値が 65536 (2 進 16 ビット) より小さい整数を無作為に与えた. 計算時間は, $j = 1, \dots, 5$ に対し, $M_{s,t}$ の第 j 列の最小消去多項式候補を計算した上で固有ベクトルを計算し, 固有ベクトル 1 個あたりの計算時間の平均値を求めた.

$\dim(A_s)$	$\dim(B_t)$	$\dim(M_{s,t})$	Time (sec.)
20	0	20	5.07×10^{-3}
20	20	40	6.01×10^{-3}
20	40	60	7.18×10^{-3}
20	60	80	8.93×10^{-3}
20	80	100	1.06×10^{-2}
20	100	120	1.40×10^{-2}
20	120	140	1.67×10^{-2}
20	140	160	2.04×10^{-2}
20	160	180	2.43×10^{-2}
20	180	200	2.94×10^{-2}

表 1: 最小消去多項式候補の次数が一定の下で、行列の次元が増加した場合の計算時間。詳細は第 4.1 章を参照。

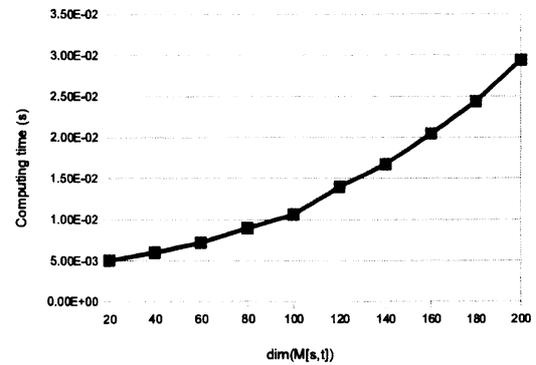


図 1: 表 1 のグラフ。詳細は第 4.1 章を参照。

4.1.2 実験結果

実験結果を表 1 および図 1 に示す。すべての実験において、計算された最小消去多項式候補の次数は 20 次であった。図 1 において、 x 軸は $\dim(M_{s,t})$ を表し、 y 軸は計算時間（秒）を表していることと、 y 軸の数値の表記は、 $a \times 10^b$ を “ aEb ” と表していることに注意。

実験結果を見る限り、行列 $M_{s,t}$ の次元の増加に伴う計算時間の増加の程度は、理論的な計算量 (13) よりも緩やかであり、行列の次元の増加に対して、固有ベクトルの計算が効率的に行われていることがわかる。

4.2 実験 2

本実験では、行列の次元が一定の下で、計算される最小消去多項式候補の次数の増加に対する固有ベクトルの計算量の増加を調べた。

4.2.1 テスト用行列の生成と実験

本実験では、固有ベクトルを計算するためのテスト用行列を以下の手順で生成した。

まず、準備として、10 個の 20 次正方行列 A_1, \dots, A_{10} を与える。ここに、 A_i ($i = 1, \dots, 10$) の各要素には絶対値が 65536 (2 進 16 ビット) より小さな整数を無作為に与える。 A_i の特性多項式 $\chi_{A_i}(\lambda)$ は既約で、 $\chi_{A_i}(\lambda)$ と $\chi_{A_j}(\lambda)$ ($i \neq j$) は互いに素になるようにする。

次に、 A_i に対し、第 1 列を除いたすべての要素を 0 で置き換えた行列を \bar{A}_i とする。そして、 A_i および \bar{A}_i を用いて、ブロック上三角行列 M を (14) のように与える。具体的には、 $i = 1, \dots, 10$ に対し、第 (i, i)

i	$\deg(\pi_{M,j}(\lambda))$	Time (sec.)
1	20	4.58×10^{-2}
2	40	8.62×10^{-2}
3	60	1.53×10^{-1}
4	80	2.57×10^{-1}
5	100	4.05×10^{-1}
6	120	6.26×10^{-1}
7	140	9.08×10^{-1}
8	160	1.26×10^0
9	180	1.75×10^0
10	200	2.34×10^0

表 2: 行列の次元が一定の下で, 最小消去多項式候補の次数が増加した場合の計算時間. 詳細は第 4.2 章を参照.

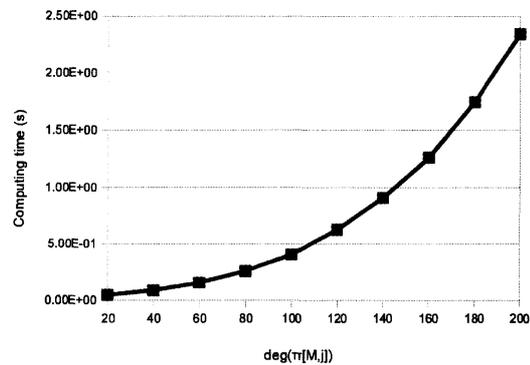


図 2: 表 2 のグラフ. 詳細は第 4.2 章を参照.

ブロック (ブロック対角要素) に A_i , 第 (i, j) ブロック ($i < j \leq 10$) に \bar{A}_i を配置する.

$$M = \begin{pmatrix} A_1 & \bar{A}_1 & \bar{A}_1 & \cdots & \bar{A}_1 & \bar{A}_1 \\ & A_2 & \bar{A}_2 & \cdots & \bar{A}_2 & \bar{A}_2 \\ & & \ddots & & \vdots & \vdots \\ & & & \ddots & \vdots & \vdots \\ & & & & A_9 & \bar{A}_9 \\ & & & & & A_{10} \end{pmatrix}. \quad (14)$$

このとき, $j = 20(i-1) + l$, $i = 1, \dots, 10$, $l = 1, \dots, 20$ に対し, M の第 j 列の最小消去多項式 $\pi_{M,j}(\lambda)$ は

$$\pi_{M,j}(\lambda) = \prod_{k=1}^i \chi_{A_k}(\lambda) \quad (15)$$

となる. すなわち, M の固有ベクトルを本稿で提案する算法で計算する際, M の固有値のうち, たとえば $\chi_{A_1}(\lambda) = 0$ の根となる固有値に属する固有ベクトルの計算に用いる最小消去多項式は $\chi_{A_1}(\lambda)$ であれば十分で, その次数は 20 であるのに対し, $\chi_{A_{10}}(\lambda) = 0$ の根となる固有値に属する固有ベクトルの計算に用いる最小消去多項式は $\chi_{A_1}(\lambda) \cdots \chi_{A_{10}}(\lambda)$ が必要で, 次数が 200 と大きくなる. 一般には, $\chi_{A_i}(\lambda) = 0$ の根となる固有値に属する固有ベクトルの計算に用いる最小消去多項式 (15) の次数は $20i$ である.

実験では, $i = 1, \dots, 10$ に対し, $\chi_{A_i}(\lambda) = 0$ の根となる固有値に属する固有ベクトルを 5 個ずつ計算した.

4.2.2 実験結果

実験結果を表 2 および図 2 に示す. すべての実験において, 与えられた行列 M の次元は 200 次である. 図 2 において, x 軸は $\deg(\pi_{M,j}(\lambda))$ を表し, y 軸は計算時間 (秒) を表していることと, y 軸の数値の表記は, $a \times 10^b$ を “aEb” と表していることに注意.

実験結果を見る限り, 最小消去多項式候補 $\pi_{M,j}$ の次数の増加に伴う計算時間の増加の程度は, 理論的な計算量 (13) よりもやや急である ($n = \deg(\pi_{M,j})$ に対し, $O(n^{1.7})$ 程度). この理由の考察は今後の課題

であるが、本稿の算法で固有ベクトルを計算する際には、最小消去多項式候補の次数がより小さい程、計算がより効率的になることがわかる。

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 [1] のソースコードを特別に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] K. Kimura. Linear Algebra PROGRAMS in Computer Algebra. <http://www-is.amp.i.kyoto-u.ac.jp/kkimur/LAPROGCA.html> (accessed December 9, 2011).
- [2] K. Kimura and H. Anai. Introduction to LAPROGCA: Linear Algebra PROGRAMS in Computer Algebra. Poster presented at The Third International Congress on Mathematical Software: ICMS 2010 (Kobe, Japan), September 13–17 2010. Available from <http://www-is.amp.i.kyoto-u.ac.jp/kkimur/ICMS2010-poster.pdf> (accessed December 9, 2011).
- [3] 田島慎一. 微分作用素を用いたレゾルベントの留数解析と行列のスペクトル分解. In *Computer Algebra — Design of Algorithms, Implementations and Applications*, 数理解析研究所講究録. 京都大学数理解析研究所, 2010. 印刷中.
- [4] 田島慎一, 樋口水紀. レゾルベントを用いた固有ベクトル計算. In *Computer Algebra — Design of Algorithms, Implementations and Applications*, 数理解析研究所講究録, 第 1666 巻, pp. 57–64. 京都大学数理解析研究所, October 2009.