数理解析研究所講究録
第1955巻 2015年 188-197

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)
Calculating eigenvectors of matrices using pseudo annihilating polynomials IV

田島 慎一*
SHINICHI TAJIMA
筑波大学 数理物質系
FACULTY OF PURE AND APPLIED SCIENCES, UNIVERSITY OF TSUKUBA

照井 章†
AKIRA TERUI
筑波大学 数理物質系
FACULTY OF PURE AND APPLIED SCIENCES, UNIVERSITY OF TSUKUBA

Abstract

Based on analysis of the residues of the resolvent, we have proposed an efficient algorithm for calculating eigenvector(s) of matrices. Our algorithm uses pseudo annihilating polynomials, and the elements in eigenvector are represented as a polynomial in eigenvalue as a variable, thus we do not need to find eigenvalues by solving the characteristic equation. We focus on our fundamental algorithm that calculates an eigenvector of the eigenvalue whose multiplicity is equal to one, and show results of experiments for test data with matrices of large dimensions.

1 はじめに

これまでに，我々は，レゾルベントの留数解析に基づき，行列の固有ベクトルを効率的に計算する算法を提案した ([2], [9]). 我々の算法は行列の最小消去多項式候補を用いるものであり，計算される固有ベクトルの成分は，固有値を変数とする多項式で表され，しかもその次数は，固有値がみたす方程式の次数を超えることなく，取りうる最小値になるという特徴をもつ. 最小消去多項式候補の算法は，著者（田島）らによるレゾルベントの留数解析に基づく効率的な算法が提案されている [3]. また，我々は，この固有ベクトル算法の実装において，“行列 Horner 法” の効率化 ([4], [5]) や，それらの並列実装 ([4], [6]) も提案している. その後，我々は，固有ベクトル算法のさらなる拡張を提案し，着目する固有値に属する一般固有ベクトル空間が，固有ベクトル空間に等しいという条件下で，着目する固有値の特性方程式における重複度が1よりも大きい場合に固有ベクトルを計算する算法の拡張を提案した ([7], [8]).

本稿では，我々が最初に提案した固有ベクトルの算法（着目している固有値の重複度が1に等しい場合に，その固有値に属する固有ベクトルを計算する算法）[9] に着目し，これまでの実験例よりも大規模な行

*tajima@math.tsukuba.ac.jp
†terui@math.tsukuba.ac.jp

列に対する実験結果を示す．以下，第 2 章では，問題設定およびその解法について復習する．第 3 章では，今回新たに与えた問題例（行列）に対する実験結果を示す．

2 問題設定と固有ベクトルの算法

行列 A を有理数体 $\mathrm{K}=\mathbb{Q}$ 上の n 次正方行列とし，E_n 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，K 上の既約因数分解があらかじめ求められているものとする:

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

本稿で用いるアルゴリズムの目的は，式 (1) のある既約因子 $f_p(\lambda)$ $(1\le p\le q)$ に対し，$f_p(\alpha)=0$ をみたす 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 $m_p=1$ $(1\le p\le q)$ とする．

n 次列ベクトル $\boldsymbol{u}\in\mathrm{K}^n$ に対し，$\{p(\lambda)\mid p(A)\boldsymbol{u}=\boldsymbol{0}\}$ をみたす多項式 $p(\lambda)$ は $\mathrm{K}[\lambda]$ のイデアルをなす．このとき，このイデアルの生成元をベクトル $\boldsymbol{u}$ に関する A の最小消去多項式と呼び，$\pi_{A,\boldsymbol{u}}(\lambda)$ で表す．$\boldsymbol{e}_j={}^t(0,\ldots,0,1,0,\ldots,0)$ を，第 j 成分が 1 に等しい n 次単位ベクトルとし，列のインデックスを $J=\{1,2,\ldots,n\}$ とする．$\boldsymbol{e}_j$ に関する A の最小消去多項式 $\pi_{A,\boldsymbol{e}_j}(\lambda)$ は

$$\pi_{A,\boldsymbol{e}_j}(\lambda)=f_1(\lambda)^{l_{j,1}}f_2(\lambda)^{l_{j,2}}\cdots f_p(\lambda)^{l_{j,p}}\cdots f_q(\lambda)^{l_{j,q}},\quad 0\le l_{j,p}\le m_p,\quad j\in J \tag{2}$$

と表される．

本稿では，固有ベクトルの計算に，$\boldsymbol{e}_j$ に関する A の “最小消去多項式候補” $\pi'_{A,\boldsymbol{e}_j}(\lambda)$ を用いる．$\pi'_{A,\boldsymbol{e}_j}(\lambda)$ は

$$\pi'_{A,\boldsymbol{e}_j}(\lambda)=f_1(\lambda)^{l'_{j,1}}f_2(\lambda)^{l'_{j,2}}\cdots f_p(\lambda)^{l'_{j,p}}\cdots f_q(\lambda)^{l'_{j,q}} \tag{3}$$

と表される．ここに，我々の $\pi'_{A,\boldsymbol{e}_j}(\lambda)$ の求め方より，$\pi'_{A,\boldsymbol{e}_j}(\lambda)$ の各既約因子の多重度は $0\le l'_{j,p}\le l_{j,p}$ を満たすことに注意する．

以下，$j\in J$ に対し，$\pi_{A,\boldsymbol{e}_j}(\lambda)$ を “A の第 j 列の最小消去多項式”，$\pi'_{A,\boldsymbol{e}_j}(\lambda)$ を “A の第 j 列の最小消去多項式候補” と呼ぶことにし，それぞれ $\pi_{A,j}(\lambda)$ および $\pi'_{A,j}(\lambda)$ で表す．また，$f_p(\lambda)$ を因子にもつような $\pi_{A,j}(\lambda)$, $\pi'_{A,j}(\lambda)$ に対し，それぞれ $g_{j,p}(\lambda)=\pi_{A,j}(\lambda)/f_p(\lambda)$, $g'_{j,p}(\lambda)=\pi'_{A,j}(\lambda)/f_p(\lambda)$ とおく．

$f_p(\lambda)$ に対し，2 変数多項式 $\Psi_p(x,y)$ を

$$\Psi_p(x,y)=\frac{f_p(x)-f_p(y)}{x-y}. \tag{4}$$

で定める．このとき，$\psi_p(x,y)$ は変数 y に関して $\deg(f_p)-1$ 次の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

$\chi_A(\lambda)$, $f_p(\lambda)$, $\Psi_p(x,y)$, $\pi_{A,\boldsymbol{e}_j}(\lambda)$, $g_{j,p}(\lambda)$ 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

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

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

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

をみたす．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 の固有値を $\lambda = \alpha$, A の第 j 列の最小消去多項式候補で $f_p(\lambda)$ を因子にもつものを $\pi'_{A,j}(\lambda)$, $g'_{j,p}(\lambda) = \pi'_{A,j}(\lambda)/f_p(\lambda)$ とおく．このとき

$$\rho(\lambda) = \Psi_p(A, \lambda E)\, g'_{j,p}(A)\, \boldsymbol{e}_j \tag{5}$$

   を計算する．

2. [**最小消去多項式のチェック**] $\pi'_{A,j}(\lambda)$ 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

$$\pi'_{A,j}(A)\, \boldsymbol{e}_j = f_p(A)\, g'_{j,p}(A)\, \boldsymbol{e}_j \tag{6}$$

   が $\mathbf{0}$ に等しくなることを確かめる．

3. もし (6) が成り立つのであれば，(5) の $\rho(\lambda)$ を A の固有ベクトルとして出力する．

上記の手順で $\rho(\lambda)$ が 1 つ求まると，$f_p(\alpha) = 0$ をみたす任意の α に対し，$\rho(\alpha)$ は A の固有値 $\lambda = \alpha$ に属する固有ベクトルを表すことに注意する．

上記の手順の中で，固有ベクトル候補 $\rho(\lambda)$ を先に計算するのは，以下の理由による．(5) の $\Psi_p(A, \lambda E)$ から，Horner 法の 1 ステップの計算のみで，(6) 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（詳細は著者らによる先行論文 ([4], [5], [9]) を参照）．

固有ベクトル候補 $\rho(\lambda)$ の計算および最小消去多項式のチェックにかかる時間計算量（K 上の係数の演算回数のみを考慮する）は，行列 A の次元を n とするとき

$$O(n^2 \cdot \deg(\pi'_{A,j})) \tag{7}$$

となる [9].

3 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] を，最小消去多項式候補の計算は `taji_mat` パッケージ [4] をそれぞれ利用した．

実験環境は以下の通り: Intel Xeon E5-2690 at 2.90 GHz, RAM 128GB, Linux 2.6.32 (SMP).

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. 行列の次元と最小消去多項式の次数がともに増加する状況下での，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．

3.1 実験1：行列の次元の増加に伴う固有ベクトルの計算量の変化

本実験では，行列の次元と最小消去多項式がともに増加する場合における固有ベクトルの計算量の変化を調べた．

3.1.1 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正方行列 A を，ブロック下三角行列

$$A = (a_{ij}) = \begin{pmatrix} A_1 & & & O \\ A_1 & A_2 & & \\ \vdots & \vdots & \ddots & \\ A_1 & A_2 & \dots & A_8 \end{pmatrix} \tag{8}$$

の形で与えた．A の要領は以下の通りである．

- A の各ブロック $A_1, \dots, A_8$ は s 行 s 列とし，s を $16, 32, 48, \dots$ と 128 まで 16 ずつ増加させた．これにより，A の次元を $128, 256, 384, \dots, 1024$ と変化させた．
- A の各ブロック $A_1, \dots, A_8$ の各成分 a_{ij} は，$|a_{ij}| < 10$ なる整数で無作為に与えた（ランダムな整数の生成には，Risa/Asir 付属ライブラリの `lin_alg.random_mat` を用いた）．
- A の特性多項式 $\chi_A(\lambda)$ は無平方．すなわち，$1 \le i \ne j \le 8$ に対し，各ブロック A_i と A_j の特性多項式 $\chi_{A_i}(\lambda)$ と $\chi_{A_j}(\lambda)$ は共通因子をもたない．

行列 A の次元が $8s$ のとき，A の第 j 列最小消去多項式 $\pi_{A,j}(\lambda)$ の次数は，$ms < j \le (m+1)s$ $(0 \le m < 8)$ に対して $(8-m)s$ であることに注意する．すなわち，第 j 列がブロック A_1 内にある場合は $\deg(\pi_{A,j}(\lambda)) = \dim(A)$ であるのに対し，第 j 列がブロック A_8 内にある場合は $\deg(\pi_{A,j}(\lambda)) = \dim(A)/8$ である．

実験では，最小消去多項式の次数が最も大きい場合として，行列 A の次元に等しい場合（すなわち $\deg(\pi_{A,j}(\lambda)) = \dim(A)$ の場合），および，最小消去多項式の次数が最も小さい場合として，$\deg(\pi_{A,j}(\lambda)) = \dim(A)/8$ の場合の固有ベクトルの計算時間とメモリ使用量を測定した．計算時間は，ガーベジコレクション等の時間を除く計算時間で，最小消去多項式の次数が等しい 5 列に対する計算時間を計測し，それらの平均値を求めた．

3.1.2 実験結果

実験結果として、最小消去多項式の次数が最も大きな場合 $(\deg(\pi_{A,j}(\lambda)) = \dim(A))$ の結果を表 1，図 1，図 2 に，最小消去多項式の次数が最も小さな場合 $(\deg(\pi_{A,j}(\lambda)) = \dim(A)/8)$ の結果を表 2，図 3, 図 4 にそれぞれ示す．各表において，メモリ使用量の数値の表記は $a \times 10^b$ を “aeb” と表していることに注意．図 1–4 において，x 軸は $\dim(A)$ を表し，y 軸は計算時間（秒）またはメモリ使用量 (bytes) を表す．図 2, 4 において，y 軸のメモリ使用量の数値の表記は，$a \times 10^b$ を “aEb” と表していることに注意．

実験結果を見る限り，行列 A 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 (7) よりも大きい（$O(n^4)$ 程度と見積もられる）．この理由の一つとして，行列の次元の増加に伴い，最小消去多項式の次数も増加し，その影響で，行列の各成分も大きくなることが計算量に影響を及ぼしている可能性が考えられる．

3.2 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．

テスト用行列として，第3.1節と同じく，式(8)の行列Aを与えた．Aの次元としては128 ($s=16$) および1024 ($s=128$) の2つの場合で実験を行った．上でも述べたが，行列Aの次元が$8s$のとき，Aの第j列の最小消去多項式$\pi_{A,j}(\lambda)$の次数は，$ms < j \leq (m+1)s$ $(0 \leq m < 8)$ に対して$(8-m)s$であることに注意する．本実験では，最小消去多項式$\pi_{A,j}(\lambda)$の次数が$s, 2s, \ldots, 8s$のもとで，固有ベクトルの計算時間とメモリ使用量を測定した．いずれの場合も，計算時間については，第3.1節と同じ要領で，最小消去多項式の次数が等しい5列に対する計算時間を計測し，それらの平均値を求めた．

3.2.1 実験結果

実験結果として，$\dim(A)=128$の場合の結果を表3, 図5, 図6に，$\dim(A)=1024$の場合の結果を表4, 図7, 図8にそれぞれ示す．各表において，メモリ使用量の数値の表記は$a \times 10^b$を“$a e b$”と表していることに注意．図5–8において，x軸は$\dim(A)$を表し，y軸は計算時間（秒）またはメモリ使用量 (bytes) を表す．図6, 8において，y軸のメモリ使用量の数値の表記は，$a \times 10^b$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(7)よりも大きい（$O(\deg(\pi'_{M,j})^3)$前後）．この理由の考察は今後の課題の一つであるが，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．

4 まとめ

本稿では，レゾルベントの留数解析に基づく行列の固有ベクトル算法において，着目している固有値の重複度が1に等しい場合に，その固有値に属する固有ベクトルを計算する算法を取り上げ，これまでの実験例よりも大規模な行列に対する実験結果を示した．実験結果より，行列の次元が大きな場合は，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を示した．

今後の課題の一つとして，実験結果では，行列の次元や最小消去多項式の次数の増加に伴う計算量の増加が，理論的な見積もりよりも大きいので，その解析が望まれる．これまでの理論的な計算量の見積もりは，体K上の演算回数のみに基づいているが，行列の次元が大きくなると，各成分の絶対値も大きくなるため，計算機上の整数の多倍長演算による計算量も大きくなる．よって，固有ベクトルの成分に現われる数値の大きさを見積もった上で，それらの計算量を見積もる必要があるものと思われる．

今回の実験において，行列・ベクトル積のHorner法は素朴な方法のみ用いたが，著者らによる拡張Horner法 ([4], [5]) を用いることにより，計算効率の改善が可能である．

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 [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] 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. 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, 2009.

[3] 田島慎一, 奈良洸平. 最小消去多項式候補とその応用. 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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 (II). 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. 119–127. 九州大学マス・フォア・インダストリ研究所, 2013.

[8] 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 (III). 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 1907 巻, pp. 50–61. 京都大学数理解析研究所, 2014.

[9] 照井章, 田島慎一. 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. Computer Algebra — Design of Algorithms, Implementations and Applications, 数理解析研究所講究録, 第 1815 巻, pp. 13–22. 京都大学数理解析研究所, 2012.

$\dim(A)$	Time (sec.)	Memory usage (bytes)
128	0.39	1.81e8
256	2.21	2.12e9
384	10.47	9.76e9
512	31.77	2.92e10
640	78.66	7.11e10
768	164.90	1.51e11
896	316.58	2.79e11
1024	527.15	4.75e11

表 1: $\deg(\pi_{A,j}(\lambda)) = \dim(A)$ の場合の計算結果. 詳細は第 3.1 章を参照.

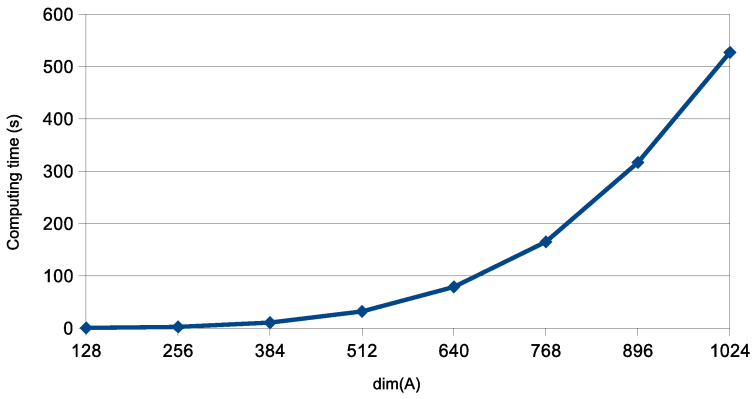

図 1: 表 1 の計算時間のグラフ. 詳細は第 3.1 章を参照.

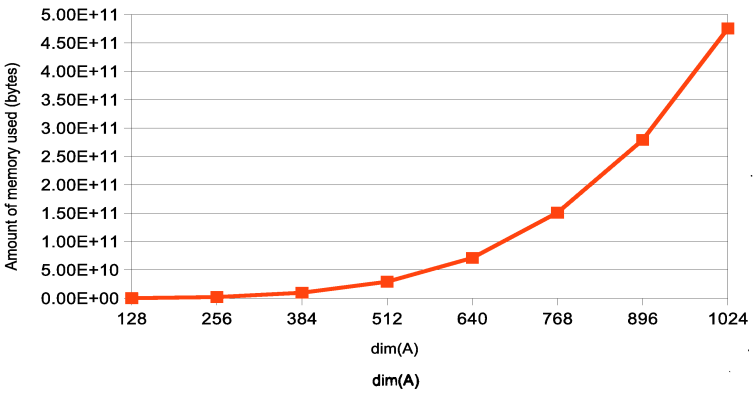

図 2: 表 1 のメモリ使用量のグラフ. 詳細は第 3.1 章を参照.

$\dim(A)$	Time (sec.)	Memory usage (bytes)
128	0.0086	5.24e6
256	0.0030	2.06e7
384	0.084	5.10e7
512	0.194	1.01e8
640	0.369	1.75e8
768	0.626	2.81e8
896	1.086	4.28e8
1024	1.559	6.26e8

表 2: $\deg(\pi_{A,j}(\lambda)) = \dim(A)/8$ の場合の計算結果. 詳細は第 3.1 章を参照.

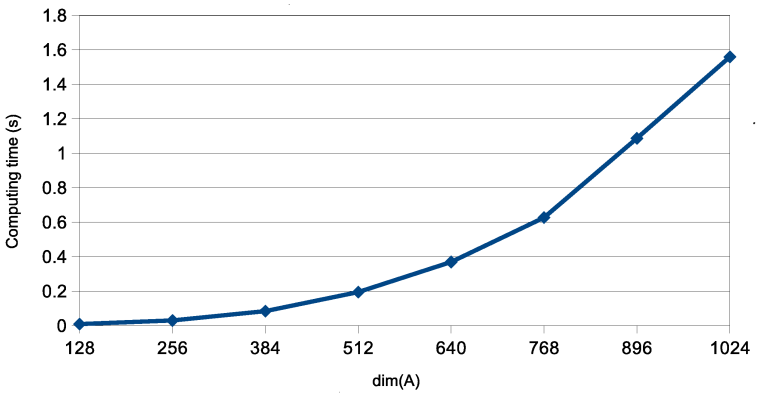

図 3: 表 2 の計算時間のグラフ. 詳細は第 3.1 章を参照.

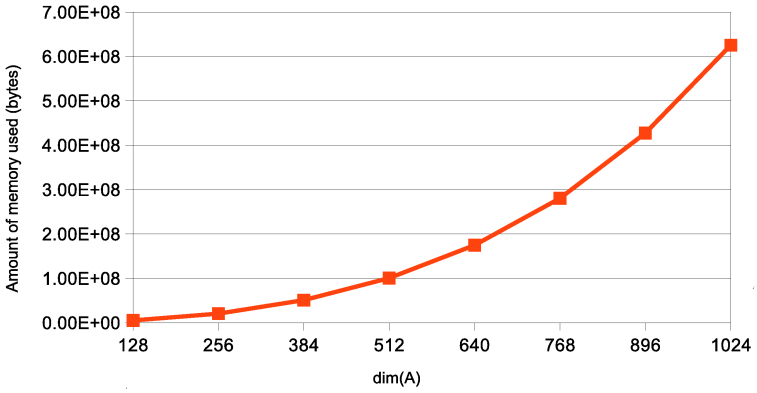

図 4: 表 2 のメモリ使用量のグラフ. 詳細は第 3.1 章を参照.

$\deg(\pi_{A,j})$	Time (sec.)	Memory usage (bytes)
16	0.0086	5.24e6
32	0.013	7.78e6
48	0.030	1.41e7
64	0.064	2.59e7
80	0.091	4.62e7
96	0.167	7.60e7
112	0.196	1.21e8
128	0.388	1.81e8

表 3: $\dim(A) = 128$ 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 3.2 節を参照．

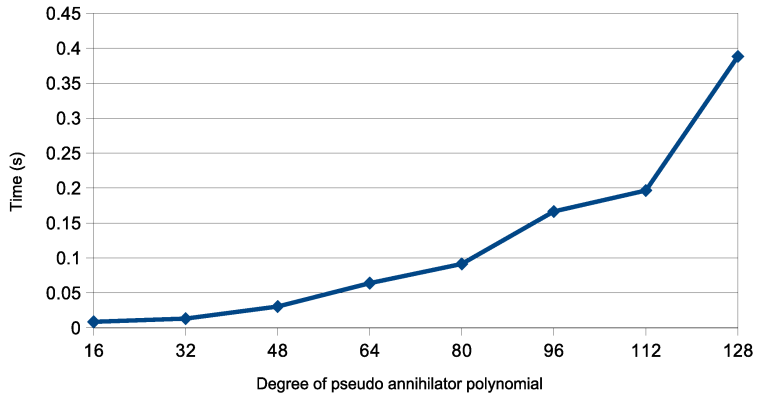

図 5: 表 3 の計算時間のグラフ．詳細は第 3.2 節を参照．

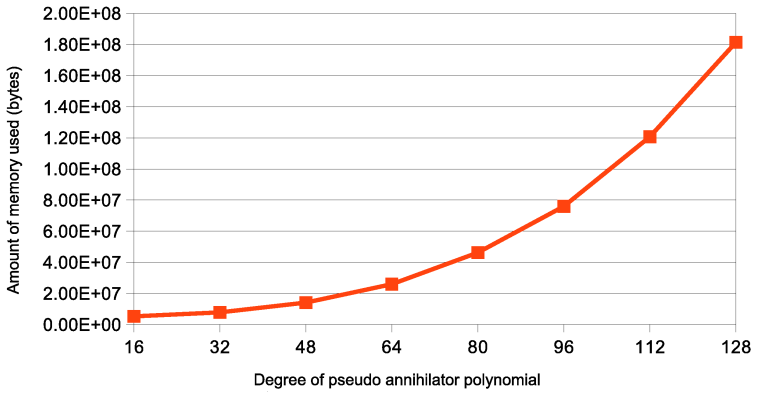

図 6: 表 3 のメモリ使用量のグラフ．詳細は第 3.2 節を参照．

$\deg(\pi_{A,j})$	Time (sec.)	Memory usage (bytes)
128	1.56	6.26e8
256	5.84	3.53e9
384	17.68	1.31e10
512	43.56	3.52e10
640	94.62	8.08e10
768	180.30	1.64e11
896	314.05	2.89e11
1024	527.15	4.75e11

表 4: $\dim(A) = 1024$ 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 3.2 節を参照．

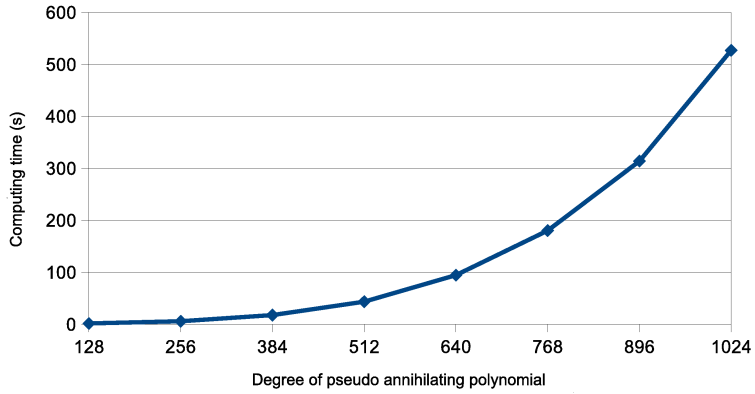

図 7: 表 4 の計算時間のグラフ．詳細は第 3.2 節を参照．

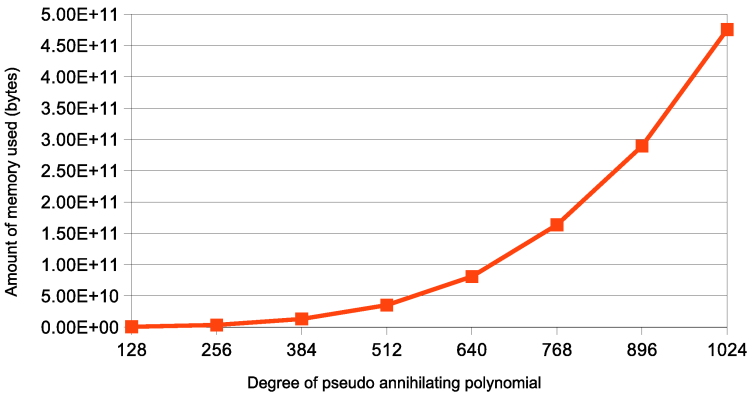

図 8: 表 4 のメモリ使用量のグラフ．詳細は第 3.2 節を参照．