数理解析研究所講究録
第1930巻 2015年 15-25

The Berlekamp Algorithm

– サーベイと試み/Survey and Refinement – *

長坂耕作
KOSAKU NAGASAKA†
神戸大学人間発達環境学研究科
GRADUATE SCHOOL OF HUMAN DEVELOPMENT AND ENVIRONMENT, KOBE UNIVERSITY

1 Introduction

本稿では，有限体上の（そして整数環上の）多項式の因数分解を行う上で，重要な（古典的な）アルゴリズムの1つであるBerlekampアルゴリズムについて，近年のサーベイとその改善についての報告を行う。

まず，Elwyn R. Berlekampにより提案された（まとめられた）因数分解法（現在では，Berlekampアルゴリズムと呼称）について復習しておく。多くの論文や書籍で引用され，提案された論文とされるものは，1967年[2]と1970年[3]の2つの論文である。大まかなアルゴリズムの流れとしては，(1) Petr-Berlekamp行列と呼ばれる行列を構成，(2) Berlekamp (sub)algebraと呼ばれる線形空間の基底を計算，(3) 与式を既約因子に分離，となる。もう少し詳しい説明は次章で，詳細については原著を参考にして頂くとして，取り扱われている問題について整理しておく。

問題 1 (Factoring Univariate Polynomial over Finite Fields)
モニックかつ無平方な $f(x) \in \mathbb{F}_q[x]$ に対して，$\mathbb{F}_q$ 上の既約分解 $f(x) = f_1(x) \cdots f_r(x)$ を求めよ。 ◁

以下，本稿では，$n = \deg f$ とし，素数 p と自然数 k に対して $q = p^k$ とする。

1.1 Factoring Algorithms over $\mathbb{F}_q$

本稿では，Berlekampアルゴリズムを取り上げるが，問題1の解法は他にも様々なものがあるので，それらについて簡単にまとめておく。基本的に，有限体上の因数分解は，次の3つのステップから構成される。

SFF:	SquareFree Factorization 無平方分解（無平方な多項式の積に分解する）
DDF:	Distinct Degree Factorization 同じ次数の既約因子のみからなる多項式の積に分解する（異なる既約因子の次数毎に分解される）
EDF:	Equal Degree Factorization 同じ次数の既約因子のみからなる多項式を既約因子に分解する（同じ次数の既約因子に分解する）

*本研究の一部は科研費（22700011）の支援で行われている。
†nagasaka@main.h.kobe-u.ac.jp

現在知られている非決定的アルゴリズムで最速と考えられるものは，SFFを良く知られているYunのアルゴリズム[11]で，DDFを1998年のKaltofenとShoupの方法[10]で，EDFを1992年のvon zur GathenとShoupの方法[14]で行うものである。計算量は，これらの計算量を順に足し合わせた，$O^{\sim}(n\log(q)) + O(n^{1.815}(\log(q))^{0.407}) + O^{\sim}(n^{(\omega+1)/2} + n\log(q))$ となる。ただし，このうち非決定的なのはEDFのみであり，SFFとDDFは決定的アルゴリズムである。一方，Riemann仮説を仮定しない決定的アルゴリズムで最速と考えられるものは，1992年のvon zur GathenとShoup[14]に記載されている決定的なEDFで，その計算量は，$O^{\sim}(n^2 + n^{\frac{3}{2}}k + n^{\frac{3}{2}}k^{\frac{1}{2}}p^{\frac{1}{2}})$ となる。なお，$f \in O^{\sim}(g) \Rightarrow f \in O(g\log(g)^{O(1)})$ であることと，決定的なBerlekampアルゴリズム（ただし，良く知られているのはvon zur Gathen[13]が修正したもの）の計算量が，$O^{\sim}(n^{\omega} + qn^2)$ であることに注意されたい。

2 The Berlekamp Algorithm

本章では，Berlekampアルゴリズムについて復習する。念のため，次の定理を述べておく。

定理 1 (中国剰余定理（Chinese Remainder Theorem）)
$f(x) \in \mathbb{F}_q[x]$ を無平方，$f_1(x), \ldots, f_r(x)$ をその $\mathbb{F}_q$ 上の既約因子とする。このとき，次式が成り立つ。

$$\mathbb{F}_q[x]/\langle f\rangle \simeq \mathbb{F}_q[x]/\langle f_1\rangle \times \cdots \times \mathbb{F}_q[x]/\langle f_r\rangle$$

◁

Berlekampアルゴリズムでは，Frobenius写像 $\Phi(a) = a^q$ により定義される次のBerlekamp (sub)algebra $\mathcal{B}$ が，既約分解において重要な役割を担っている。

$$\mathcal{B} = \{\, h \in \mathbb{F}_q[x]/\langle f\rangle \mid \Phi(h) = h \,\} \simeq \mathbb{F}_q \times \cdots \times \mathbb{F}_q = (\mathbb{F}_q)^r$$

Fermatの小定理より $\Phi(h) - h = \prod_{\alpha\in\mathbb{F}_q}(h-\alpha)$ であるので，既約因子の分離は次のように行える。

$$f(x) = \prod_{\alpha\in\mathbb{F}_q} \gcd(f, h-\alpha)$$

以下では，具体的な手順について確認していく。

1. Petr-Berlekamp 行列の構成

$x^q \bmod f(x)$ のべき乗計算を行い，次式を満たす係数 $q_{i,j}$ を求める。

$$x^{iq} \bmod f(x) = \sum_{j=0}^{n-1} q_{i,j}x^j \quad (i = 0, 1, \ldots, n-1)$$

この係数を要素に持つ次の行列 Q を構成する。この行列は，Petr-Berlekamp行列と呼ばれる。

$$Q = (q_{i,j}) = \begin{pmatrix} q_{0,0} & \cdots & q_{0,n-1} \\ \vdots & \ddots & \vdots \\ q_{n-1,0} & \cdots & q_{n-1,n-1} \end{pmatrix} \in \mathbb{F}_q^{n\times n}$$

Petr-Berlekamp行列から単位行列を引いた行列の零空間（核）は係数ベクトルと多項式を同一視することで，Berlekamp (sub)algebraと同型であり，このことから次のステップが容易に計算可能となる。

$$\mathrm{Ker}(Q - I) \simeq \mathcal{B} \ \text{(coefficient vector} \Leftrightarrow \text{polynomial)}$$

2. Berlekamp (sub)algebra の基底計算

$(Q-I)$ の零空間の基底の組を掃き出し法などで計算する (次の線形方程式の解空間の基底)。

$$(Q-I)\vec{h}=\vec{0}$$

零空間の基底から対応する多項式を求めることで, その多項式の組 $\{h_1(x)=1,\ldots,h_r(x)\}$ は, Berlekamp (sub)algebra の基底多項式の組となる。

$$\mathcal{B}=\left\{\sum_{i=1}^{r}a_i\times h_i(x)\,\middle|\,a_i\in\mathbb{F}_q\right\},\quad (h_1,\ldots,h_r\in\mathbb{F}_q[x]/\langle f\rangle)$$

このステップが最も計算量が多く, その計算量は $O(n^{\omega})\approx O(n^{2.376})$ となる。なお, 1991 年の Kaltofen と Saunders[9] による Wiedemann アルゴリズムを用いれば, $O^{\sim}(n^2\log(q))$ に低減することは可能であるが, 非決定的となる。

3. 既約因子への分解

既約因子への分解方法として, Berlekamp による基本的な方法をまず説明する。冒頭にも述べた

$$f(x)=\prod_{\alpha\in\mathbb{F}_q}\gcd(f(x),h(x)-\alpha)$$

という関係式が成立しているため, $f(x)$ やその因子に対して, 基底多項式と有限体の元を動かしながら, 次のように最大公約因子を計算していくことで, 最終的にすべての既約因子を求められる (Berlekamp (sub)algebra の次元が既約因子の個数なので, いつ分解が終了となるか判定可能)。

$$\gcd(\cdot,h_i(x)-\alpha_j)\text{ for }i=1,\ldots,r\text{ and }\alpha_j=0,1,\ldots,q-1$$

ところが, この方法は総当たりになるため計算効率が悪い。一般的には, Zassenhaus[15] により提案[1]された最小多項式を求め, その根に限定して最大公約因子を求める方法が取られる (日本語の文献としては, [16] の 49 項も参照のこと)。

$$f(x)=\prod_{\alpha\in\left\{\alpha\in\mathbb{F}_q\,\middle|\,g(\alpha)=0\right\}}\gcd(f(x),h(x)-\alpha)\ \text{ where }\ g(x)\in\mathbb{F}_q[x]\text{ s.t. }g(h(x))\equiv 0\pmod{f(x)}$$

2.1 Berlekamp アルゴリズムの計算例

具体的に次の多項式の既約分解を, Berlekamp アルゴリズムで行ってみる。

$$f(x)=x^7+2x^5+x^4+2x^3+x^2+x+1\in\mathbb{F}_5[x]$$

まず, Petr-Berlekamp 行列を構成するために, x^5 の累乗を計算する。

$$\begin{aligned}
x^{0\times 5}&\equiv 1\\
x^{1\times 5}&\equiv x^5\\
x^{2\times 5}&\equiv 2x^6+3x^5+4x^4+3x^3+3x^2+3x+1\\
x^{3\times 5}&\equiv 4x^5+4x^4+4x^2+4x+4\\
x^{4\times 5}&\equiv 3x^6+x^5+x^4+x^3+2x^2+x+3\\
x^{5\times 5}&\equiv 4x^6+3x^5+4x^4+3x^3+4x^2+4x\\
x^{6\times 5}&\equiv 2x^6+x^5+x^3+x+4
\end{aligned}$$

[1]原典については, 神戸大学の野呂先生に示唆を頂きました。ここに感謝致します。

この結果により，Petr-Berlekamp 行列は次のように構成される。

$$Q \equiv \begin{pmatrix} 2 & 4 & 3 & 0 & 2 & 0 & 0 \\ 1 & 3 & 1 & 4 & 3 & 1 & 0 \\ 0 & 4 & 1 & 4 & 4 & 0 & 0 \\ 1 & 3 & 1 & 0 & 3 & 0 & 0 \\ 0 & 4 & 2 & 4 & 3 & 0 & 0 \\ 1 & 4 & 1 & 4 & 3 & 0 & 0 \\ 4 & 0 & 3 & 4 & 1 & 0 & 1 \end{pmatrix}$$

次の線形方程式を解き，Berlekamp (sub)algebra の基底多項式を求める。

$$(Q - I_7)\vec{h} \equiv \vec{0},\ Q - I_7 \equiv \begin{pmatrix} 1 & 4 & 3 & 0 & 2 & 0 & 0 \\ 1 & 2 & 1 & 4 & 3 & 1 & 0 \\ 0 & 4 & 0 & 4 & 4 & 0 & 0 \\ 1 & 3 & 1 & -1 & 3 & 0 & 0 \\ 0 & 4 & 2 & 4 & 2 & 0 & 0 \\ 1 & 4 & 1 & 4 & 3 & -1 & 0 \\ 4 & 0 & 3 & 4 & 1 & 0 & 0 \end{pmatrix}$$

結果，解空間の基底が下記左のものになるので，右のような基底多項式が求まる。

$$(\vec{h}_1, \vec{h}_2, \vec{h}_3) \equiv \begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 4 & 4 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \qquad \begin{aligned} h_1(x) &\equiv 1, \\ h_2(x) &\equiv x^6 + x^5 + 4x^3 + x, \\ h_3(x) &\equiv x^4 + 4x^3 + x^2 \end{aligned}$$

以上の基底多項式から，最後の既約因子への分解は次のように計算される。

$\gcd(f(x), h_2(x)) \equiv 1,$
$\gcd(f(x), h_2(x) - 1) \equiv 1,$
$\gcd(f(x), h_2(x) - 2) \equiv 1,$
$\gcd(f(x), h_2(x) - 3) \equiv x^5 + 4x^4 + 2x^3 + 1 \Longrightarrow f(x) \equiv (x^5 + 4x^4 + 2x^3 + 1)(x^2 + x + 1)$
$\gcd(x^5 + 4x^4 + 2x^3 + 1, h_3(x)) \equiv x^2 + 4x + 1 \Longrightarrow f(x) \equiv (x^2 + 4x + 1)(x^3 + x + 1)(x^2 + x + 1)$

最終的に，解空間の次元と同じ 3 つの因子が求まったところで，既約分解が得られたことになる。

3 Survey Recent Articles

最近の論文から Berlekamp アルゴリズムについて取り上げられているものをサーベイする。

3.1 大きな拡大次数での改善

1 つ目は，拡大次数の大きな有限体上($q = p^k$ の k が十分大きい場合) における改善で，2007 年に Genovese により行われた研究 [7] である。改善のポイントは，$\mathbb{F}_q$ 上の基底計算を $\mathbb{F}_p$ 上の基底計算として行うこと。

そして，部分可群の零空間の計算を Gröbner 基底の計算で用いられる項順序の変更(Change Ordering) を取り入れて行うことである。これにより，古典的な方法では，$\mathbb{F}_p$ 上で $O(n^3k^3)$ のコストがかかる基底計算を，$O(n^3k^2+nk^3)$ に減らすことが出来ている。特に，拡大次数 k の指数が小さくなっているので，拡大次数の大きな有限体上の分解において効果が高くなると考えられる。

なお，$\mathbb{F}_q$ を $\mathbb{F}_p$ 上の線形空間として扱うことは既知のものであり，この場合，Berlekamp (sub)algebra は，$\mathcal{B}=\{h\in\mathbb{F}_q[x]/\langle f\rangle \mid h^q=h\}$ でなく，$\mathcal{B}_p=\{h\in\mathbb{F}_q[x]/\langle f\rangle \mid h^p=h\}$ となる。これらは，世界的な数式処理の教科書とも言える書籍 Modern Computer Algebra[6] の 417 項 Exercise 14.40 (i) にも，因数分解が結果として次式で与えられることを示せという問題が掲載されている。

$$f(x)=\prod_{\alpha\in\mathbb{F}_p,\ h\in\mathcal{B}_p}\gcd(f,h-\alpha)$$

論文には著者 Genovese による計算機実験の結果も掲載されているので，結果については原論文を参照されたい。なお，比較対象には，1992 年の von zur Gathen と Shoup の非決定的な方法 [14] の EDF(計算量は $n^2k^2(k+d)\log(r)$，d は既約因子の次数) と Shoup の NTL に実装されている Berlekamp アルゴリズム(計算量は n^3k^3) が挙げられている。

3.2 既約因子の分離での改善

2 つ目は，Berlekamp アルゴリズムの最後のステップの改善の論文である。2008 年に Insua と Ladra が発表した論文 [8] では，既約因子を最大公約数の計算により分離するステップの改善を，Gröbner 基底を用いて行っている。論文タイトルに「 Note」とあり，残念ながら計算量や深い考察は行われていない(本報告で，この論文の計算量を求め，かつ考察を行っているのはこのため)。

Insua と Ladra が論文中で述べていることを簡単に紹介する。まず，Berlekamp アルゴリズムの分離ステップにおいて次式を用いて既約因子を求めるのは，総当たりなので他の方法を考えるべきとある。

$$f(x)=\prod_{\alpha\in\mathbb{F}_q}\gcd(f(x),h(x)-\alpha)\implies \gcd(\cdot,h_i(x)-\alpha_j)\text{ for }i=1,\ldots,r\text{ and }\alpha_j=0,1,\ldots,q-1$$

その代わりに，$\mathbb{F}_q[x,z]$ の Gröbner 基底計算で既約因子を求める方法を提案している。

1. 簡約 Gröbner 基底の計算

Berlekamp (sub)algebra の基底多項式 $h(x)\in\mathcal{B}$ に対して，イデアル $\langle f(x),h(x)-z\rangle$ の辞書式順序 $(x\succ z)$ の簡約 Gröbner 基底を計算する。結果の基底を $\langle g_1(x,z),\ldots,g_m(x,z)\rangle$ $(g_{t+1}\succ g_t)$ とする。

2. 求めた簡約 Gröbner 基底の性質

求まった簡約 Gröbner 基底の基底多項式に次の性質が成立する($\mathrm{lc}_x(\cdot)$ は，x に関する主係数)。

- $\mathrm{lc}_x(g_{t+1})\mid\mathrm{lc}_x(g_t)$ $(t=1,\ldots,m-1)$ かつ $\mathrm{lc}_x(g_m)\in\mathbb{F}_q$ である。
- $g_1(x,z)=\mathrm{lc}_x(g_1)$ であり，これは $\mathrm{res}_x(f(x),h(x)-z)$ の根基である($\mathrm{res}_x(\cdot,\cdot)$ は終結式)。
- $\mathrm{lc}_x(g_t)\mid g_t(x,z)$ $(t=1,\ldots,m)$ である。

3. 簡約 Gröbner 基底による既約因子の分離

前項の性質より，既約因子の分離において最大公約因子の計算を必要とせず，以下で分離可能となる。

$$f(x)=\prod_{t=1}^{m-1}\prod_{\alpha\in\left\{\alpha\in\mathbb{F}_q\,\middle|\,\mathrm{lc}_x(g_t)/\mathrm{lc}_x(g_{t+1})(\alpha)=0\right\}}g_{t+1}/\mathrm{lc}_x(g_{t+1})(x,\alpha)$$

3.2.1 既約因子の分離を行う計算例

実際に，簡約 Gröbner 基底を使用する方法で，次の多項式の既約分解を求めてみる。

$$f(x) = x^7 + 2x^5 + x^4 + 2x^3 + x^2 + x + 1 \in \mathbb{F}_5[x]$$

零空間を計算した結果，Berlekamp (sub)algebra の基底多項式として次の 3 つを得る。

$$h_1(x) \equiv 1,\ h_2(x) \equiv x^6 + x^5 + 4x^3 + x,\ h_3(x) \equiv x^4 + 4x^3 + x^2$$

手順に基づき，$\langle f(x), h_2(x) - z\rangle$ の簡約 Gröbner 基底を辞書式順序 $(x \succ z)$ で求めると，

$$\langle z^2 + 3z + 2,\ x^2z + 2x^2 + xz + 2x + z + 2,\ x^5 + 4x^4 + 2x^3 + 2xz + 4x + 3z + 2\rangle$$

を得るが，これを既約因子の分離に用いる性質が分かりやすく書き直したものが，次の多項式集合である。

$$\langle (z+1)(z+2),\ (z+2)(x^2 + x + 1),\ x^5 + 4x^4 + 2x^3 + 2xz + 4x + 3z + 2\rangle$$

これにより，$z + 2 = 0$ の根（つまり，$z = 3$）を $x^5 + 4x^4 + 2x^3 + 2xz + 4x + 3z + 2$ に代入したものと，$(z+1)(z+2)/(z+2) = 0$ の根（つまり，$z = 4$）を $x^2 + x + 1$ に代入したものが，$f(x)$ の因子（既約とは限らない）となることが分かる。そこで，これらの因子を次のように，$f_1(x), f_2(x)$ とおく。

$$f_1(x) \equiv x^2 + x + 1,\ f_2(x) \equiv x^5 + 4x^4 + 2x^3 + 1$$

Berlekamp (sub)algebra の次元は 3 であり，引き続いて，$f_1(x)$ と $f_2(x)$ の分解をする。そこで，$\langle f_1(x), h_3(x) - z\rangle$ の簡約 Gröbner 基底を辞書式順序 $(x \succ z)$ で求めると，$\langle z + 2,\ x^2 + x + 1\rangle$ となり，$f_1(x)$ を分解することは出来ない。次に，$\langle f_2(x), h_3(x) - z\rangle$ の簡約 Gröbner 基底を辞書式順序 $(x \succ z)$ で求めると，

$$\langle z^2 + 4z,\ x^2z + 4x^2 + 4xz + x + z + 4,\ x^3 + xz + 1\rangle$$

を得るが，これを既約因子の分離に用いる性質が分かりやすく書き直したものが，次の多項式集合である。

$$\langle z(z+4),\ (z+4)(x^2 + 4x + 1),\ x^3 + xz + 1\rangle$$

よって，$z + 4 = 0$ の根（つまり，$z = 1$）を $x^3 + xz + 1$ に代入したものと，$z(z+4)/(z+4) = 0$ の根（つまり，$z = 0$）を $x^2 + 4x + 1$ に代入したものが，$f_2(x)$ の因子となることが分かる。そこで，これらの因子を次のように，$f_{21}(x), f_{22}(x)$ とおく。

$$f_{21}(x) \equiv x^2 + 4x + 1,\ f_{22}(x) \equiv x^3 + x + 1$$

次元 3 であったので，以上で必要な分解がすべて求まったことになり，既約分解として以下を得る。

$$f(x) \equiv (x^2 + x + 1)(x^2 + 4x + 1)(x^3 + x + 1)$$

3.2.2 論文には記載されていない本稿著者による留意事項

論文で提案されている方法は，他に見たことがなく，著者らによる新しい方法であることは疑いようがない。しかしながら，使われている性質自体は新たらしいものではなく，非常に良く知られているものばかりである。本報告の著者としては，このような良く知られた性質に基づく方法が，これまで提案されることがなかったことに驚きを隠せない（一方，提案した著者らにはその着眼に敬意を払いたい）。

例えば，$\langle f(x), h(x)-z\rangle$ $(h \in \mathcal{B})$ の簡約 Gröbner 基底 $\langle g_1(x,z),\ldots,g_m(x,z)\rangle$ $(g_{t+1} \succ g_t)$ に現れる $g_1(x,z)=\mathrm{lc}_x(g_1)$ が，$\mathrm{res}_x(f(x),h(x)-z)$ の根基であることを考える。そもそも，Zassenhaus[15] により提案された最小多項式を使った既約因子の分離ステップを実現するのに必要な最小多項式の計算であるが，直接的に終結式から行う方法が，Modern Computer Algebra[6] の 417 項 Exercise 14.40 (ii) に掲載されている（書籍では，既約分解が最小多項式の求根問題に帰着される理由を述べよ，という観点から取り上げられている）。そして，包括的 Gröbner 基底などで，パラメータ（変数）の条件に依存する基底であっても，そのすべての場合を含むように基底計算を行うことが可能なことが知られている。論文で提案されている内容は，これらをうまく組み合わせたものと考えられる。

4 既約因子の分離での改善を更に改善

前章で述べたように，2008 年に Insua と Ladra が発表した論文 [8] で提案した方法は非常に興味深い。しかしながら，総当たりで最大公約因子を求める方法は，既に Zassenhaus[15] により最小多項式の求根問題に帰着されており，簡約 Gröbner 基底を使用することからも，計算量の点での改善は見込めないと考えられる。ところが，著者らにより解析されていない計算量を本報告で求めて見たところ，予想に反して最小多項式による方法よりも計算量が小さくなることが判明した。本章では，この結果について報告する。

まず，一般的な最小多項式による分離も，今回の簡約 Gröbner 基底による分離も，Berlekamp (sub)algebra の自明でない基底多項式 $h_2(x),\ldots,h_r(x)\in\mathcal{B}_q$ それぞれに対して，その時点での与式の分解 $\bar{f}_1(x),\ldots,\bar{f}_\ell(x)$ s.t. $f(x)\equiv\prod_{i=1}^{\ell}\bar{f}_i(x)$ に現れるそれぞれの因子多項式に対して，次のような処理を行っていることに留意したい。

最小多項式による分離

1. 最小多項式 $g(y)\in\mathbb{F}_q[y]$ を計算する（$g(y)$ は $g(h_i(x))\equiv 0 \pmod{\bar{f}_j(x)}$ を満たす）。
2. 最小多項式の根 $c_1,\ldots,c_s\in\mathbb{F}_q$ を計算する（c_u は $g(c_u)\equiv 0$ を満たす）。
3. 自明でない因子を $\gcd(\bar{f}_j(x),h_i(x)-c_u)$ により計算する。

簡約 Gröbner 基底による分離

1. $\langle\bar{f}_j(x),\ h_i(x)-z\rangle$ の簡約 Gröbner 基底 $\langle g_1(z),\ g_2(x,z),\ \ldots,\ g_m(x,z)\rangle$ $(g_{t+1}\succ g_t)$ を求める。
2. 隣接主係数に共通しない根 $c_1,\ldots,c_s\in\mathbb{F}_q$ を求める（c_u は $\mathrm{lc}_x(g_t)/\mathrm{lc}_x(g_{t+1})(c_u)\equiv 0$ を満たす）。
3. 自明でない因子を $g_{t+1}/\mathrm{lc}_x(g_{t+1})(x,c_u)$ による求める。

つまり，計算量の比較をするにあたっては，ある基底多項式 $h(x)$ とある因子多項式 $f(x)$ を固定して，その場合でのみ比較をすれば十分であることが分かる。

そこで，最小多項式による方法の計算量を見積もる。最初のステップである最小多項式の計算は，1999 年の Shoup の方法 [12] を用いることで，$O(n^{\frac{1}{2}}M(n)+n^2)$ となる。その根の計算については，単なる代入法で $O(nq)$ となる。これは，残念ながら $\log(q)$ の多項式時間ではないが，これ以外の有力な方法は非決定的な EDF となってしまう。非決定的でも良ければ，計算量は $O(\log(n)\log(nq)M(n))$ または $O\tilde{}(n\log(q))$ となる。最後の最大公約因子の計算は，Euclid の互除法の高速算法（多くの研究があり，ここでは代表的な書籍として，Modern Computer Algebra[6] を挙げておく）を用いるとして，$O(s\log(n)M(n))$ となる（s は根の個数）。従って，トータルでは，$O(n^{\frac{1}{2}}M(n)+n^2+nq+s\log(n)M(n))$ となる。

次に，簡約 Gröbner 基底による方法の計算量を見積もる。最初のステップである簡約 Gröbner 基底の計算は，イデアルの生成多項式が特殊な形をしているため，項順序を違えれば既に簡約 Gröbner 基底となって

いる上，このイデアルは0次元である。そのため，この部分の計算量は思ったよりも低く，良く知られる項順序を変更するFGLM[5]を使っても$O(n^3)$，2003年のBasiriとFaugèreによる方法[1]を使えば，$O(s^3n^2)$で抑えられる（sは根の個数)。結果求まった基底多項式の主係数同士の除算やその根の計算は，通常の高速除算と代入法を採用すると，$O(sM(s)+sq)$となる。最後の自明でない因子の計算は，除算と代入だけなので，$O(M(s)n+n^2)$で抑えられる。従って，トータルでは，$O(n^3+sq)$または$O(s^3n^2+sq)$となる。

以上の結果から，まとめると次のようになり，簡約Gröbner基底による方法の方が良いことが分かる。ただし，決定的なBerlekampアルゴリズムでの計算量は，基本的にPetr-Berlekamp行列の掃き出しに依存しており，あまり違いはないとも言える。

方法	計算量	$s=2, M(n)=n\log(n)\log\log(n)$の場合
最小多項式	$O(n^{\frac{1}{2}}M(n)+n^2+nq+s\log(n)M(n))$	$O(n^2+nq)$
Gröbner基底	$O(n^3+sq)$ or $O(s^3n^2+sq)$	$O(n^2+q)$

5 Berlekampアルゴリズムの更なる改善の取り組み

この章では，簡約Gröbner基底を用いた方法に基づいて更なる改善が可能かを，Berlekampアルゴリズムの改善という方向性と，有限体上の求根に関する未解決問題への試みという方向性からの取り組みについて報告する。

5.1 SLC-PRSによる改善の取り組み

まず，主係数が数である範囲においてのみ項簡約を行うことで得られる剰余（Scalar Leading Coefficient Remainder）や，それによる多項式剰余列（Scalar Leading Coefficient Polynomial Remainder Sequence）を導入する。

定義 2 (Scalar Leading Coefficient Remainder)
$f(x)\in\mathbb{F}_q[z][x]$の$\mathrm{lc}_x(g)\in\mathbb{F}_q$を満たす$g(x)\in\mathbb{F}_q[z][x]$による*Scalar Leading Coefficient Remainder*を$\mathrm{slcrem}_x(f(x),g(x))$で表記し，以下のアルゴリズムの出力で定義する。

1. *While* $\deg_x f\geq\deg_x g$ *and* $\mathrm{lc}_x(f)\in\mathbb{F}_q$ *do:*
2. $f\leftarrow f-\mathrm{lc}_x(f)\mathrm{lc}_x(g)^{-1}g$
3. *EndWhile and output* f

◁

定義 3 (Scalar Leading Coefficient Polynomial Remainder Sequence)
$w_1(x),w_2(x)\in\mathbb{F}_q[z][x]$で$\mathrm{lc}_x(w_2)\in\mathbb{F}_q$を満たすとする。このとき，以下のアルゴリズムで生成される$w_1,w_2,w_3,\ldots$を，$w_1(x)$と$w_2(x)$の$\mathbb{F}_q[z]$上の**SLC-PRS***(Scalar Leading Coefficient Polynomial Remainder Sequence)*と定義する。

1. $i=1$
2. $w_{i+1}=\mathrm{slcrem}_x(w_i(x),w_{i-1}(x))$
3. *if* $\mathrm{lc}_x(w_{i+1})\in\mathbb{F}_q\setminus\{0\}$ *then* $i=i+1$ *and goto Step 2*

4. *output* $w_1(x), w_2(x), \ldots, w_{i+1}(x)$

◁

これらは通常の剰余や剰余列ではないが，基本的に項簡約なので，SLC-PRS の各多項式は $w_i(x,z) \in \langle w_1(x,z), w_2(x,z)\rangle$ という 性質を持つ。その結果，余り 有用ではないが次の補題を得る(証明は省略する)。

補題 4
$w_1(x)$, $w_2(x)$, $w_3(x)$, ...を，$F_q[z]$ 上の $w_1(x) = f(x)$ と $w_2(x) = h(x) - z$ の *SLC-PRS* とする。低い確率ではあるが，$w_{t-1}(x,c) | f(x)$ が成り 立つことがある。ここで，$\mathrm{lc}_x(w_i) \in \mathbb{F}_q$ $(i < t)$ かつ $\mathrm{lc}_x(w_t) \notin \mathbb{F}_q$ であり，$c \in \mathbb{F}_q$ は $\mathrm{lc}_x(w_t)(z)$ の根である。なお，$\deg_z \mathrm{lc}_x(w_t) = 1$ かつ $\deg_x w_t < \deg_x f$ も成立する。 ◁

具体的にこの補題が効果を発揮する例として，次の多項式の既約分解を求めてみる。

$$f(x) = x^7 + 2x^5 + x^4 + 2x^3 + x^2 + x + 1 \in \mathbb{F}_5[x]$$

零空間を計算した結果，Berlekamp (sub)algebra の基底多項式として次の 3 つを得る。

$$h_1(x) \equiv 1,\ h_2(x) \equiv x^6 + x^5 + 4x^3 + x,\ h_3(x) \equiv x^4 + 4x^3 + x^2$$

自明でない因子を見つけるのに，$f(x)$ と $h_2(x) - z$ の簡約 Gröbner 基底でなく，SLC-PRS を計算する。

$$\begin{aligned}
&w_1(x,z) = f(x),\ w_2(x,z) = h_2(x) - z,\\
&w_3(x,z) = \mathrm{slcrem}_x(w_1,\ w_2) \equiv 3x^5 + 2x^4 + x^3 + (z+2)x + 4z + 1,\\
&w_4(x,z) = \mathrm{slcrem}_x(w_2,\ w_3) \equiv (3z+1)x^2 + (3z+1)x + 3z + 1
\end{aligned}$$

結果，$w_4(x)$ の主係数に着目して $3z + 1 = 0$ を解き，$w_3(x)$ の z にその根である $z = 3$ を代入したものが，$f(x)$ の因子となる。簡約 Gröbner 基底の方法と 異なり，余因子は割り 算で直接計算する必要がある。

$$w_3(x,3) \equiv 3x^5 + 2x^4 + x^3 + 3,\ \ f_1(x) = \mathrm{lc}_x(w_3(x,3))^{-1} w_3(x,3) \equiv x^5 + 4x^4 + 2x^3 + 1$$

$$\Longrightarrow\ f_1(x) \equiv x^5 + 4x^4 + 2x^3 + 1,\ f_2(x) \equiv x^2 + x + 1$$

次に，$f_1(x)$ と $h_3(x) - z$ に対して，同様にその SLC-PRS を計算する。

$$\begin{aligned}
&w_1(x,z) = f_1(x),\ w_2(x,z) = h_3(x) - z,\\
&w_3(x,z) = \mathrm{slcrem}_x(w_1,\ w_2) \equiv x^3 + xz + 1,\\
&w_4(x,z) = \mathrm{slcrem}_x(w_2,\ w_3) \equiv (4z+1)x^2 + (z+4)x + 4z + 1
\end{aligned}$$

結果，$w_4(x)$ の主係数に着目して $4z + 1 = 0$ を解き，$w_3(x)$ の z にその根である $z = 1$ を代入したものが，$f_1(x)$ の因子となる。

$$w_3(x,1) \equiv x^3 + x + 1,\ \ f_{11}(x) = \mathrm{lc}_x(w_3(x,1))^{-1} w_3(x,1) \equiv x^3 + x + 1$$

最終的に，簡約 Gröbner 基底の方法と 同じく，以下の既約分解が得られる。

$$f(x) \equiv (x^3 + x + 1)(x^2 + 4x + 1)(x^2 + x + 1)$$

しかしながら，補題は自明でない因子を常に見つけられることは保証しておらず，因数分解アルゴリズムとしては不完全で，簡約 Gröbner 基底による方法などにフォールバックしなければならない。計算量の改善には結びつかないが，一度のコストは古典的に計算しても $O(n^2)$ であり，場合により 高速算法により $O(\log(n)M(n))$ や $O^{\sim}(n)$ で計算可能なことも 考えると，簡約 Gröbner 基底の方法の前段として実行しても良いかもしれない(SLC-PRS による方法では，根の計算が不要なことが一つのメリット)。

5.2 F_p 上の求根問題への取り組み

Berlekamp アルゴリズムに関係する未解決問題の一つに次がある (Modern Computer Algebra より)。

問題 2

Given $f(x) \in \mathbb{F}_p[x]$, of degree $n \leq p$, which is known to have n distinct roots in $\mathbb{F}_p$, and where p is prime, can we find these roots with a number of operations that is polynomial in n and $\log(p)$? ◁

この問題への既知の結果は，Modern Computer Algebra に基づき最近の結果も調べた限りにおいて，概ね次のような結果であり，Riemann 仮説を仮定してもしなくても，完全には解かれていないことが分かる。

- von zur Gathen と Shoup による 1992 年の結果 [14]: $O^{\sim}(n^2 + n^{\frac{3}{2}}p^{\frac{1}{2}})$
- Evdokimov による 1994 年の結果 [4]: $(n^{\log(n)}\log(p))^{O(1)}$ (Riemann 仮説を仮定)
- 多くの研究者による特定の場合の結果: $(n\log(p))^{O(1)}$ (Riemann 仮説を仮定)

簡約 Gröbner 基底による方法は，求根しなければいけないため，この問題へは直接的には寄与しない (基本的に，因数分解の対象の多項式と同じ次数の z に関する多項式の零点を求めなければいけないため)。本報告の SLC-PRS による方法については，求根する必要はないが，簡約 Gröbner 基底による方法にフォールバックしなければならないので同様である。なお，口頭発表時には，3 次の多項式を用いて，SLC-PRS による方法の拡張で問題 2 を解く試みも紹介 (実験から余り有用でないことが判明したと紹介) しているが，本報告では割愛する。

参 考 文 献

[1] A. Basiri and J.-C. Faugère. Changing the ordering of Gröbner bases with LLL: case of two variables. In *Proceedings of the 2003 International Symposium on Symbolic and Algebraic Computation*, pages 23–29 (electronic), New York, 2003. ACM.

[2] E. R. Berlekamp. Factoring polynomials over finite fields. *Bell System Tech. J.*, 46:1853–1859, 1967.

[3] E. R. Berlekamp. Factoring polynomials over large finite fields. *Math. Comp.*, 24:713–735, 1970.

[4] S. Evdokimov. Factorization of polynomials over finite fields in subexponential time under GRH. In *Algorithmic number theory (Ithaca, NY, 1994)*, volume 877 of *Lecture Notes in Comput. Sci.*, pages 209–219. Springer, Berlin, 1994.

[5] J. C. Faugère, P. Gianni, D. Lazard, and T. Mora. Efficient computation of zero-dimensional Gröbner bases by change of ordering. *J. Symbolic Comput.*, 16(4):329–344, 1993.

[6] J. V. Z. Gathen and J. Gerhard. *Modern Computer Algebra.* Cambridge University Press, New York, NY, USA, 2 edition, 2003.

[7] G. Genovese. Improving the algorithms of Berlekamp and Niederreiter for factoring polynomials over finite fields. *J. Symbolic Comput.*, 42(1-2):159–177, 2007.

[8] M. A. Insua and M. Ladra. A note on Gröbner bases and Berlekamp's algorithm. *Appl. Math. Comput.*, 196(1):77–85, 2008.

[9] E. Kaltofen and B. D. Saunders. On Wiedemann's method of solving sparse linear systems. In *Applied algebra, algebraic algorithms and error-correcting codes (New Orleans, LA, 1991)*, volume 539 of *Lecture Notes in Comput. Sci.*, pages 29–38. Springer, Berlin, 1991.

[10] E. Kaltofen and V. Shoup. Subquadratic-time factoring of polynomials over finite fields. *Math. Comp.*, 67(223):1179–1197, 1998.

[11] D. E. Knuth. *The Art of Computer Programming, Volume II: Seminumerical Algorithms, 2nd Edition.* Addison-Wesley, 1981.

[12] V. Shoup. Efficient computation of minimal polynomials in algebraic extensions of finite fields. In *Proceedings of the 1999 International Symposium on Symbolic and Algebraic Computation (Vancouver, BC)*, pages 53–58 (electronic), New York, 1999. ACM.

[13] J. von zur Gathen. Factoring polynomials and primitive elements for special primes. *Theoret. Comput. Sci.*, 52(1-2):77–89, 1987.

[14] J. von zur Gathen and V. Shoup. Computing Frobenius maps and factoring polynomials. *Comput. Complexity*, 2(3):187–224, 1992.

[15] H. Zassenhaus. On Hensel factorization. I. *J. Number Theory*, 1:291–311, 1969.

[16] 野呂 正行. **計算機代数入門**, *http://www.math.kobe-u.ac.jp/Asir/ca.pdf.* 2005.