ICOT Technical Report: TR-181

TR-181

有限体上の一変数多項式の因数分解について

横山和弘，野呂正行，竹島　卓
(富士通)

May, 1986

ICOT
Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan
(03) 456-3191～5
Telex ICOT J32964

Institute for New Generation Computer Technology

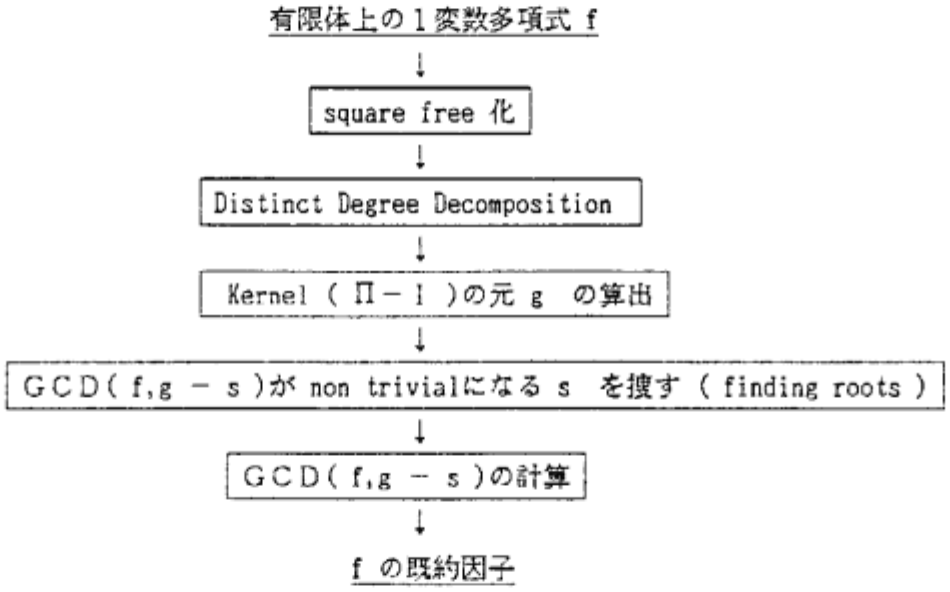
有限体上の一変数多項式の因数分解について

富士通・国際研　横山和弘　野呂正行　竹島卓

有限体上の一変数多項式の因数分解における Berlekamp-Zassenhaus のアルゴリズムに対して改良・精密化を試みた。

§1 序論

今日の有限体上の一変数多項式の因数分解のアルゴリズム研究は、1967年の E. R. Berlekamp〔1〕の発表したアルゴリズムが始点となっている。その3年後に、Berlekamp は改良版を H. Zassenhaus〔10〕の示唆を受けて発表した〔3〕。この二つの論文は現在に至るまでのアルゴリズム研究の基本的枠組を与え、以後の研究はこの改良に向けられている。基本的枠組は以下の図で示される。



Berlekamp 以後の研究の主なものは R. T. Moenck 〔7〕、D. Lazard 〔6〕、M. Rabin〔8〕、D. G. Cantor, H. Zassenhaus 〔4〕などがあるが、いずれも Berlekampのアルゴリズムの中の『 finding roots 』に関しての発展的改良となっている。

今回、本論文では、上記の『 finding roots 』の前後の処理についての改良・精密化を試み、いくつかのステップをアルゴリズムに加えることで効率化を図っている。具体的には、Berlekamp の示した DDD(distinct degree decomposition) についての改良(step 2,3)、finding roots の前の段階においての f の真に小さい因子が出せる場合の抽出(step 4)、finding roots により得た root の利用についての精密化 (step 4, 5,6) をここでは行っている。DDD の改良の部分では、効率化に成功していると思われる。step 4 以降においては、効率よりむしろ root の性質の数学的精密化に成功している。

§2 アルゴリズムの概要

p を素数とした時のＧＦ(p) 上での一変数多項式の因数分解を考える。
f(x) をＧＦ(p) 上の一変数多項式とする。以下のステップにより因数分解を行う。
(標数 p の有限体 ＧＦ(q) に対しても同様である。異なる点は、Frobenius map のところが p 乗写像のままであることである。)

Step 1. f(x)の monic 化および square-free 化

Step 2. 固有多項式による因子の次数・個数のチェック

f(x) の i 次既約因子の個数 σ_i を以下の手順でもとめる。

Π : ＧＦ(p) 〔 x 〕 / (f(x)) の上の Frobenius map に対して Π の固有多項式 Q(x) を取る。即ち、ＧＦ(p) 〔 x 〕 / (f(x))をＧＦ(p) 上の線形空間とみる時に、Π : ＧＦ(p) 〔 x 〕 / (f(x)) $\ni$ h(x) $\rightarrow$ h(x)^p をベースを決めて行列表示したものをＱとおく。この時、Q(x) = det (xＩ－Ｑ)である。

w_i = max 〔 u ; ($x^i - 1$)^u | Q(x) 〕 とおく。

また、A = ($A_{i,j}$) を次の様に定義する。

$$A_{i,j} = \begin{cases} 0 & j \nmid i \\ -1 & j \mid i \text{ かつ } j \neq i \\ 1 & j = i \end{cases}$$

この時、次の等式により σ_i は得られる。

$$A \cdot {}^t(w_n, \ldots, w_1) = {}^t(\sigma_n, \ldots, \sigma_1)$$

Step 3. ＤＤＤ (distinct degree decomposition)

f(x) の i 次既約因子全体の積 f_i を Berlekamp の方法で求める。〔2〕

しかし、ここで用いる ＧＣＤ($f, x^{p^i} - x$) については、 step 2 により ＧＣＤ の次数が判っているので ＧＣＤ の次数が予め判っている場合の ＧＣＤ 計算を行う。
(Borodin-von zur Gathen-Hopcroft の ＧＣＤ 計算〔9〕)

以下 step 3 で得られた f_i を f の代わりに考えることにより、以下の step では既約因子がすべて同じ次数であるとして f の因数分解を行う。

Step 4. 分離・非分離の判定および真に小さい因子の算出

f の既約因子の個数を r とし、因子を各々 h_i とする。degree h_i = m とおく。最初に次のことを断っておく。

Kernel ($\Pi - 1$) の元 g に対して、g の固有多項式 $G_0(x)$ を次の様に定義する。

g $\equiv c_i \bmod h_i$ ($c_i \in$ ＧＦ(p)) に対して、$G_0(x) = \prod_{i=1}^{r} (x - c_i)$

そこで、g に対して分離的・非分離的であると言うことを次の様に定義する。

定義 上記の g に対して、g が分離的であるとは $G_0(x)$ が重根を持たない時に言う。

$G_0(x)$ が重根を持つ時、g は非分離的であると言う。

さて、Kernel ($\Pi-1$) の元 g を取る。(degree $g(x) \geq 1$ に取る。)
$1, g, g^2, \ldots, g^r$ は Kernel ($\Pi-1$) の元であり、Kernel ($\Pi-1$) の次元は r であるので、$1, g, \ldots, g^{r'}$ の間に自明でない線形関係が存在するような最小の r' とその線形関係より得られる多項式を $G(x)$ とする。(Zassenhaus [10]) この $G(x)$ を g の最小多項式と呼ぶ。 r が r' に一致することと、g が分離的であることは同値である。

(4-1). $r = r'$ の時、$G(x) = G_0(x)$ となり g は分離的である。更に、$G(x)$ の根 s に対して、GCD($f, g - s$) は既約因子となる。

(4-2). $r \neq r'$ の時、g は非分離的である。

(a) $G_0(x)$ を次の手順で求める。

$1, g, \ldots, g^{r'}$ に適当な元を加えることにより、Kernel ($\Pi-1$) の基底 $g_0, g_1, \ldots, g_{r-1}$ を取る。ここで、$g_0 = 1$、$g_1 = g$ にする。

$M_i = (b^i_{j,k})$ を次の様に定義する。($b^i_{j,k}$ は M_i の j,k 成分)

$b^i_{j,k}$ は次の式により得られる： $g_i \cdot g_j \equiv \sum_{k=0}^{r-1} b^i_{j,k} \cdot g_k \bmod f(x)$

この時、$G_0(x) = \det(xI - M_1)$ となる。

(b) $G_0(x)$ を「無平方分解」する。($G(x)$ を利用する。)

$G_0(x) = G_1(x) \cdot (G_2(x))^2 \cdots (G_t(x))^t$ と分解されたとする。

この時、$G(x) = G_1(x) \cdot G_2(x) \cdots G_t(x)$ となる。

(b-1) $G_0(x) = (G_t)^t$ なる形の場合 (これを、純非分離と呼ぶ)

$G(x)$ の根 s に対して、GCD($f, g - s$) は t 個の既約因子の積となる。

(よって、次の step で得られた根 s に対して GCD($f, g - s$) を求め、これについて step 4 を行う。step 5, 5' を参照)

(b-2) 上記以外の場合、即ち純非分離でない場合。

$GCD(f, G_i(g)) = \prod_{G_i(s_j)=0} h_j$ (ここで $g \equiv s_j \bmod h_j$)

により、f は真に小さい因子に分けられる。ここで、$i = 1$ の時は、$GCD(f, G_i(g))$ に対して g は分離的であり、$i \geq 2$ の時は g は純非分離となる。(純非分離の場合には (b-1) の操作を $GCD(f, G_i(g))$ に対して行う。)

注) 上記の GCD は皆その次数が予め判っているので、step 2 と同様の計算法を用いる。

step 4 により、g は分離的であるか純非分離的であるかのいずれかに帰着される。

Step 5. finding roots of $G(x)$

$G(x) = 0$ の根 $s_1, s_2, \ldots, s_v$ を求める。(ここで、$v = r$ または r / t)

根を求める方法は、quadratic residue を利用したものを使う。

Step 6. finding irreducible factors of $f(x)$

(6-1) step 4 で g が分離的であった時
step 5 で求めた $G(x) = 0$ の根 $s_1, s_2, \ldots, s_r$ に対して、
$GCD(f, g - s_i)$ が求める既約因子である。
更に、計算の効率化のために以下の手順を加える。
$r \geq 3$ の時、$H_i(x) = G(x)/(x - s_i)$ とおく。この時、$GCD(f, H_i(g))$ は既約因子のコファクターとなるので、先に $GCD(f, H_i(g))$ を求めておき、$f / GCD(f, H_i(g))$ により既約因子を求める。

(6-2) step 4 で g が純非分離的であった時
step 5 で求めた $G(x) = 0$ の根 $s_1, s_2, \ldots, s_v$ に対して、$(v = r / t)$
$GCD(f, g - s_i)$ を求める。ここで $GCD(f, g - s_i)$ は $m \cdot t$ 次の多項式である。この $GCD(f, g - s_i)$ に対して step 4 へ戻る。

step 5, 6 については、次に置き換えることも考えられる。

Step 5'. $G(x)$ の non trivial factor を求める。(それを $H(x)$ とおく。)

Step 5'. $f(x)$ の non trivial factor を求める。
step 5' で求めた $H(x)$ に対して、それに対応する $f(x)$ の因子を求める。
即ち、$GCD(f, H(g))$ が $H(x)$ に対応する因子である。更に次のことが言える。
$\text{degree } GCD(f, H(g)) = m \cdot \text{degree } H(x)$
特に、$\text{degree } H(x) = 1$ の時、$GCD(f, H(g))$ は既約因子である。
$\text{degree } H(x) \neq 1$ の時、f の代わりに $GCD(f, H(g))$ を考えて step 4 へ戻る。

以下のセクションで各 step について詳しく述べる。(step 1 は省略する。)

§3 step 2 について

$f(x)$ を square free かつ monic な n-次の多項式とする。$f(x)$ は $GF(p)$ 上で次のように既約因子に分解されるとする。

$$f(x) = \prod_{i=1}^{r} h_i \qquad (\text{ここで } \text{degree } h_i = n_i \text{ とおく。})$$

更に $f(x)$ は i 次の既約因子を σ_i 個持つと仮定する。
$GF(p)[x] / f(x)$ を R で表し、$GF(p)[x] / f_i(x)$ を R_i で表すことにする。
Frobenius map $\Pi : R \ni h \to h^p \in R$ を R の基底 $1, x, x^2, \ldots, x^{n-1}$ により表現すれば次の行列 Q が得られる。

$$Q = \begin{bmatrix} q_{n-1,n-1}, \ldots, q_{0,n-1} \\ \vdots \qquad\qquad \vdots \\ q_{n-1,0}, \ldots\ldots, q_{0,0} \end{bmatrix}$$

ここで、$x^{pk} \equiv q_{k,n-1} \cdot x^{n-1} + \cdots + q_{k,0} \bmod f(x)$
$(k = 0, 1, \ldots, n-1)$

この時、Π の固有多項式 $Q(x)$ は次の様にして得られる。

$$Q(x) = \det(xI - Q)$$

まず、次の定理を示す。

定理 1. $Q(x) = \prod_{i=1}^{r} (x^{n_i} - 1)$

証明：Chinese Remainder Theorem により、$R = R_1 \oplus R_2 \oplus \cdots \oplus R_r$ である。R_i での Frobenius map を Π_i とする。即ち、$\Pi_i : R_i \ni h \to h^p \in R_i$ である。この Π_i を用いれば、Π は上の同形より次のようにみることができる。

$$\Pi = \Pi_1 \oplus \Pi_2 \oplus \cdots \oplus \Pi_r$$

即ち、R の元 h に対して、$h = h_1 \oplus h_2 \oplus \cdots \oplus h_r$ なる同形に対して $h^p = h_1^p \oplus h_2^p \oplus \cdots \oplus h_r^p$ となるので、Π の R への作用は

$$\Pi(h) = \Pi_1(h_1) \oplus \Pi_2(h_2) \oplus \cdots \oplus \Pi_r(h_r) \text{ となる。}$$

claim 1. 各 Π_i に対して、Π_i の最小多項式を $Q_i(x)$ とおくと、

$$Q_i(x) = x^{n_i} - 1 \text{ となる。}$$

claim 1 の証明：$GF(p)[x] / f_i(x) = GF(p^{n_i})$ である。よって、任意の $R_i = GF(p)[x] / f_i(x)$ の元 h に対して次が成り立つ。

$h_i^{p^{n_i}} - h_i = 0$ 即ち、$\Pi_i^{n_i} - I = 0$ である。このことは、$Q_i(x) \mid x^{n_i} - 1$ を意味する。

さて、degree $Q_i(x) = d < n_i$ と仮定する。

$Q_i(x) = \sum_{j=0}^{d} q^{(i)}{}_j \cdot x^j$ とおく。

この時、$Q_i(\Pi_i) = 0$ より $\sum_{j=0}^{d} q^{(i)}{}_j \cdot (\Pi_i)^j = 0$ である。

R_i の任意の元 h に対して、$\sum_{j=0}^{d} q^{(i)}{}_j \cdot (\Pi_i)^j(h) = \sum_{j=0}^{d} q^{(i)}{}_j \cdot h^{p^j} = 0$

よって、$\overline{Q_i}(x) = \sum_{j=0}^{d} q^{(i)}{}_j \cdot x^{p^j}$ とおけば、R_i の任意の元 h に対して $\overline{Q_i}(h) = 0$ となる。このことは、$GF(p^{n_i})$ の任意の元 h に対して言えるので、

$x^{p^{n_i}} - x \mid \overline{Q_i}(x)$ を意味し、$d \geq n_i$ となり矛盾である。

以上により、最小多項式 $Q_i(x)$ は次数が n_i となり、monic であることより、

$$Q_i(x) = x^{n_i} - 1 \text{ となる。} \qquad \square$$

claim 2. $Q(x) = \prod_{i=1}^{r} (x^{n_i} - 1)$

claim 2 の証明：R の基底として、次の様にとる。

$\{ \ 0 \oplus \cdots \oplus 0 \oplus a_{j,k}(x) \oplus 0 \oplus \cdots \oplus 0 \mid 1 \leq j \leq r, \ 1 \leq k \leq n_i \}$

ここで $\{ a_{j,i} \mid 1 \leq k \leq n_j \}$ は R_j の基底とする。

上の基底により、Π を行列表示すれば、その行列 Q は次の様になる。

$$Q = \begin{bmatrix} Q_1, 0, \ldots, 0 \\ 0, Q_2, \ldots, 0 \\ \vdots \qquad \vdots \\ 0, 0, \ldots, Q_r \end{bmatrix}$$

ここで各 Q_i は Π_i を上の R_i の基底により行列表示したものである。

よって、claim 1 により $\det(xI-Q_i) = Q_i(x) = x^{i}-1$ であるので、

ゆえに、$Q(x) = \det(xI-Q) = \prod_{i=1}^{r} Q_i(x) = \prod_{i=1}^{r}(x^{n_i}-1)$ となる。 □

次にセクション2で定義した w_i と A について述べる。

さて、$w_i = \max\{u ; (x^i-1)^u \mid Q(x)\}$ であり、$A = (A_{i,j})$ については以下の様であった。

$$A_{i,j} = \begin{cases} 0 & j \nmid i \\ -1 & j \mid i \quad \text{かつ } j \neq i \\ 1 & j = i \end{cases}$$

ここでは更に、$\underline{W} = {}^t(w_n,\ldots,w_1)$、$\underline{\sigma} = {}^t(\sigma_n,\ldots,\sigma_1)$ とおくことにする。

次の定理により、step 2 の正当性が言える。

定理 2. $A\cdot\underline{W} = \underline{\sigma}$ である。

証明：定理 1 により、$w_i = \#\{n_j ; i \mid n_j\}$ となる。

そこで、$A\cdot\underline{W} = \underline{\sigma'}$ とする時に、$\underline{\sigma'} = {}^t(\sigma'_n,\ldots,\sigma'_1)$ に対して次が言える。

$\sigma'_k = (\Sigma'_d - w_d) + w_k$ （ここで Σ'_d は $k \mid d$ かつ $d \neq k$ なる d を動く。）

しかし、$f(x)$ は k 次の既約因子を σ_k 個持つので、

$w_i = \sum_d \sigma_d$ と書ける。よって、これを代入して

$\sigma'_k = (\Sigma'_d - w_d) + w_k = (\Sigma'_d - (\sum_e \sigma_e)) + \sum_d \sigma_d = \sigma_k$

よって、$\underline{\sigma'} = \underline{\sigma}$ となり、$A\cdot\underline{W} = \underline{\sigma}$ を得る。 □

§4 step 3 について

step 2 により、$f(x)$ には m 次既約因子が σ_m 個あることが判っている。

この時、$GCD(f(x), x^{p^m}-x) = \prod_{d\mid m} f_d$ となる。ここで f_d は d 次既約因子全体の積とする。

さて、$\sigma_d \neq 0$ なる d を小さい順に並べたものを、$d_1,\ldots,d_r$ とする。

f_{d_i} を次のアルゴリズムにより算出する。（Berlekamp のアルゴリズムの改良）

アルゴリズム DDD （c.f. Berlekamp〔3〕）

Initialize : $F^{(0)} = f(x)$, $R^{(0)} = x$

Recursion : $R^{(i)} \equiv (R^{(i-1)})^{p^{e_i}} \mod F^{(i-1)}$ ここで $e_i = d_{i-1} - d_i$

$f_{d_i} = GCD(F^{(i-1)}, R^{(i)} - x)$

$F^{(i)} = F^{(i-1)} / f_{d_i}$

アルゴリズムの詳細について

(1) $R^{(i)} \equiv (R^{(i-1)})^{p^{e_i}} \mod F^{(i-1)}$ の求め方

$a(x) = \sum_{i=0}^{n-1} a_i \cdot x^i$ に対して、$\underline{a} = {}^t(a_{n-1},\ldots,a_0)$ を対応させる。

$R^{(i)} \to \underline{R^{(i)}}$, $F^{(i)} \to \underline{F^{(i)}}$ なる対応を考えることにより、

$\underline{R^{(i)}} \equiv Q^{e_i}\cdot\underline{R^{(i-1)}} \mod \underline{F^{(i-1)}}$ により求める。

(2) $f_{d_i} = GCD(F^{(i-1)}, R^{(i)} - x)$ の計算法

degree $f_{d_i} = d_i \cdot \sigma_i$ ($= N(i)$ とおく)、degree $F^{(i-1)} = n - \sum_{j=1}^{i-1} d_j \cdot \sigma_j$ 、かつ degree $R^{(i)} - x <$ degree $F^{(i-1)}$ であるのことに注意しておく。

GCDは、次の $S(x)$ および $T(x)$ を求めることにより計算される。

$$S(x) \cdot F^{(i-1)}(x) + T(x) \cdot (R^{(i)}(x) - x) \equiv f_{d_i}(x)$$

ここで、degree $F^{(i-1)}(x)$ − degree $f_{d_i}(x) = n - \sum_{j=1}^{i} d_j \cdot \sigma_j = M(i)$ とおけば、degree $S(x) < M(i) - 1$ かつ degree $T(x) < M(i)$ に取れる。

$S(x) = \sum_{i=0}^{M(i)-2} s_i \cdot x^i$ 、$T(x) = \sum_{i=0}^{M(i)-1} t_i \cdot x^i$ とおく。

$F^{(i-1)}(x) = \sum_{i=0}^{M(i-1)} a_i \cdot x^i$ 、$R^{(i)}(x) - x = \sum_{i=0}^{M(i-1)-1} b_i \cdot x^i$ に対して、

$$P_i = \begin{bmatrix} a_{M(i-1)} & 0 & \cdots & 0 & 0 & \cdots & 0 \\ a_{M(i-1)-1} & a_{M(i-1)} & & & b_{M(i-1)-1} & & \\ a_{M(i-1)-2} & \ddots & & & \vdots & \ddots & \\ \vdots & \ddots & & & \vdots & \ddots & \\ a_{N(i)+2} & \cdots & a_{M(i-1)} & b_{N(i)+2} & \cdots & b_{M(i-1)-1} & \\ \vdots & & \vdots & \vdots & & \vdots & \\ a_{2N(i)-M(i-1)+1} & \cdots & a_{N(i)} & b_{2N(i)-M(i-1)+1} & \cdots & b_{N(i)} & \end{bmatrix}$$

とおけば、

$$P_i \cdot \begin{bmatrix} s_{M(i)-2} \\ \vdots \\ s_0 \\ t_{M(i)-1} \\ \vdots \\ t_0 \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ \vdots \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

となる。

よって、上の線形方程式を解くことにより、$S(x)$ および $T(x)$ が得られ、最終的に f_{d_i} が求まる。(Borodin-von zur Gathen-Hopcroft の計算法〔9〕の利用)

上記の二つの step の利点は、以下に挙げられる。

(1) 既約因子の次数・個数のチェックにおいて、Gunji-Arnon〔5〕の方法がある。

Gunji-Arnon の方法は、Kernel ($\Pi^i - 1$) の次元を調べることにより、既約因子の次数・個数を求めている。我々の方法では、固有多項式の多項式による割算により求めている。一度固有多項式を求めてしまえば、後は簡単な多項式間の割算に帰着される点で、Gunji-Arnon の方法より効率的と思われる。

(2) DDD において、予め GCD の次数が判っているので、1 次より n 次までのすべての i に対して調べる必要がなくなった。更に、GCD 計算においては、Bordin-von zur Gathen-Hopcroft のアルゴリズムを用いることにより、行列の判別式 (PRS) の代わりに線形方程式の解により求めることができる。PRS は行列の成分の中に多項式を含んでおり、実際は、n 個の行列式を求めることを行っているが、上記の方法は同じサイズの行列の一個の線形方程式を解くだけである。

§5 step 4 について

以下、前の step により f は r 個の m 次既約因子の積になると仮定してよい。

即ち、$f(x) = \prod_{i=1}^{r} h_i(x)$, degree $h_i(x) = m$, degree $f = m \cdot r$ となる。

Zassenhaus の方法 (Zassenhaus [10])

Kernel ($\Pi - I$) $\ni g$ (degree $g \geq 1$) に対して、

$1, g, g^2, \ldots, g^r$ は Kernel ($\Pi - I$) の元となる。よって、$1, g, g^2, \ldots, g^r$ の間には線形関係が存在する。この線形関係の内で、次数最小の関係を $\sum_{i=0}^{r'} a_i \cdot g^i \equiv 0$ とする。

この時、$G(x) = \sum_{i=0}^{r'} a_i \cdot x^i$ を g の最小多項式と呼ぶ。

注) 最小多項式の呼び名は本論文で便宜上呼ぶだけで一般的ではない。

G(x) について次のことが言える。(c.f. Zassenhaus [10] , Berlekamp [3])

補題 3. $g \equiv c_i \bmod h_i$ ($i = 1, \ldots, r$) の時、$G(c_i) = 0$ となる。

更に、$\{ c_1, \ldots, c_r \}$ の内で異なるもの全体を $\{ c'_1, \ldots, c'_{r''} \}$ とおけば、

$r'' = r'$ であり、$G(x) = \prod_{i=1}^{r'} (x - c'_i)$ となる。

証明：$G(g) \equiv 0$ より、$G(g) = G(c_i) \equiv 0 \bmod h_i$ となる。よって、次数をみて $G(c_i) = 0$ を得る。

次に、$H(x) = \prod_{i=1}^{r''} (x - c'_i)$ とおく。

この時、$H(g) \equiv H(c_i) \equiv 0 \bmod h_i$ ($i = 1, \ldots, r$) となる。これは、$H(g) \equiv 0 \bmod f$ を意味する。よって、$G(x) \mid H(x)$ となる。

逆に、$G(c'_i) = 0$ より、$H(x) \mid G(x)$ となり、$H(x) = G(x)$ を得る。□

そこで最小多項式の根 c が求まれば、

$$GCD(f, g - c) = \prod_{g \equiv c \bmod h_i} h_i$$ となり、f の non-trivial factor が得られる。

以上が Zassenhaus の方法であるが、G のすべての根 c に対して GCD(f,g − c) により f の既約因子が求まるかどうかは、degree G = r の時のみ可能である。

または、ある根 c に対して GCD(f,g − c) により f の既約因子が求まるかどうかは、$\{ c_1, \ldots, c_r \}$ の内で c と等しいものは c 以外に存在しない時のみ可能である。

そこで、G の根に対して、その GCD(f,g − c) は何になるのかが判った方が効率的であると考えられる。よって以下の分離的・非分離的と言った概念を用いることにする。まず §2 で述べた定義をもう一度述べる。

定義 上記の記号をそのまま使う。この時、g の固有多項式 G_0 を次の様に定義する。

$$G_0 = \prod_{i=1}^{r} (x - c_i)$$

定義 Kernel ($\Pi - I$) の元 g に対して、g が分離的であるとは、最小多項式と固有

多項式が一致する時を言う。そうでない時は、非分離的であると言う。

上の定義により次がただちに言える。

定義の系 4. (1) gが分離的であるならば、g の最小多項式の任意の根 c に対して、GCD(f,g − c) は f の既約因子を与える。
(2) g が分離的であるならば、$1,g,g^2,\ldots,g^{r-1}$ は Kernel (Π − I) の基底となる。

例 (1) p < r の時は、いかなる g に対しても g は分離的には成りえない。
(2) p ≧ r の時に、Kernel (Π − I) の元の内で、分離的であるものの割合は、
$p\cdot(p-1)\cdot(p-2)\cdots(p-r+1)/(p^r-1)$ である。

固有多項式の求め方

以下 Kernel (Π − I) の元 g に対する固有多項式 G_g を求める方法を示す。
ここで degree g ≧ 1 と仮定する。
1 および g を含む Kernel (Π − I) の基底 $1,g,g_2,\ldots,g_{r-1}$ を取る。
ここで、$1 = g_0$, $g = g_1$ とおく。

さて、各 g_i は次のようになっている。

$$g_i \equiv c_{i,j} \bmod h_j \quad (0 \leqq i \leqq r-1,\ 1 \leqq j \leqq r)$$

(特に、$c_{0,j} = 1$ である。)
また、$g_i \cdot g_j$ は Kernel (Π − I) の元であるので、ある $b^i{}_{j,k} \in GF(p)$ が存在して次の様に書ける。

$$g_i \cdot g_j \equiv \sum_{k=0}^{r-1} b^i{}_{j,k} \cdot g_k \mod f$$

そこで、$M_i = (b^i{}_{j,k})_{0<i,j<r-1}$ とおく。M_i は r×r の行列である。

定理 5. $\det (xI - M_i) = \prod_{j=0}^{r-1} (x - c_{i,j})$ となる。
即ち、$\det (xI - M_i)$ は g_i の固有多項式となる。

証明 : $(1,0,\ldots,0) \cdot (M_i)^k = (a^{(i)}{}_{k,0},\ a^{(i)}{}_{k,1},\ldots,\ a^{(i)}{}_{k,r-1})$ とおく。

claim 1. $(g_i)^k \equiv \sum_{j=0}^{r-1} a^{(i)}{}_{k,j} \cdot g_j \bmod f$ となる。

claim 1 の証明 : k に関する induction により証明する。

$(g_i)^k \equiv \sum_{j=0}^{r-1} a^{(i)}{}_{k,j} \cdot g_j$ であるならば、

$$(g_i)^{k+1} \equiv \sum_{j=0}^{r-1} a^{(i)}{}_{k,j} \cdot g_i \cdot g_j \equiv \sum_{j=0}^{r-1} a^{(i)}{}_{k,j} \cdot \Big(\sum_{t=0}^{r-1} b^i{}_{j,t} \cdot g_t \Big)$$

$$\equiv \sum_{t=0}^{r-1} \Big(\sum_{j=0}^{r-1} a^{(i)}{}_{k,j} \cdot b^i{}_{j,t} \Big) \cdot g_t$$

$$\equiv \sum_{t=0}^{r-1} a^{(i)}{}_{k+1,t} \cdot g_t \quad \text{となり claim 1 は言えた。} \qquad \square$$

claim 2. $\det(xI - M_i) = \prod_{j=1}^{r}(x - c_{i,j})$ となる。

claim 2 の証明：$h_1(x),\ldots,h_r(x)$ は互いに素であるので、次を満たす $A_j(x)$ $B_j(x)$ が存在する。

$A_j(x)\cdot h_1(x)\cdots h_{j-1}(x)\cdot h_{j+1}(x)\cdots h_r(x) + B_j(x)\cdot h_j(x) = 1$

そこで、$S_j(x) = A_j(x)\cdot h_1(x)\cdots h_{j-1}(x)\cdot h_{j+1}(x)\cdots h_r(x)$ とおく。

即ち、$S_j(x) \equiv 1 \mod h_j(x)$

$\equiv 0 \mod h_t(x)\ (t\neq j)$ となる。

特に、$S_j(x)$ は Kernel $(\Pi - I)$ の元である。

よって、ある $s_{j,k}$ が存在して、$S_j(x) \equiv \sum_{k=0}^{r-1} s_{j,k}\cdot g_k$ と書ける。

そこで、$\underline{S}_j = (s_{j,0},\ldots,s_{j,r-1})$ とおけば、

$\underline{S}_j\cdot(M_i - c_{i,j}I) = 0$ となる。

なぜならば、$\underline{S}_j\cdot(M_i - c_{i,j}I)$ を多項式に置き換えれば、

$S_j(x)\cdot g_i(x) - c_{i,j}\cdot S_j(x)$ となる。

これより、$S_j(x)\cdot g_i(x) - c_{i,j}\cdot S_j(x) \equiv 0 \mod h_t\ (t\neq j)$

$\equiv c_{i,j} - c_{i,j} \equiv 0 \mod h_j$

よって、$S_j(x)\cdot g_i(x) - c_{i,j}\cdot S_j(x) \equiv 0 \mod f$ となり、

このことは、$\underline{S}_j\cdot(M_i - c_{i,j}I) = 0$ を意味する。

即ち、$\underline{S}_j$ は M_i の固有ベクターとなり、更には各 $\underline{S}_j$ は線形独立であるので、

M_i の固有値は $\{c_{i,j}\}$ であることが言える。以上により claim 2 は証明された。 □

定理 5 により、g の固有多項式を以下の方法で求めることができる。

g に対して、次数が下からの線形独立な最大集合 $\{1,g,g^2,\ldots,g^{r'}\}$ を求める。

(1) $r' = r$ の時は、最小多項式と固有多項式は一致し、g は分離的となる。

(2) $r' \neq r$ の時は、$\{1,g,g^2,\ldots,g^{r'}\}$ に基底を加えて Kernel $(\Pi - I)$ の基底を作る。それを、$\{1,g_1,g_2,\ldots,g_{r-1}\}$ とする。ここで、$g_1 = g$ とおく。

この基底に対して、先の $b^i_{i,j}$ を求め、$\det(xI - M_i)$ により g の固有多項式を得る。

非分離の場合の処理

以下、g は非分離的である場合を考える。即ち、最小多項式 C と固有多項式 G_0 は異なる場合である。この時、G_0 を無平方分解する。

つまり、$G_0 = G_1\cdot(G_2)^2\cdots(G_t)^t$ と分解する。

（ここで、G_i は i 重因子の積である。）

次の簡単な補題が成り立つ。

補題 6. $G = G_1\cdot G_2\cdots G_t$ である。

よって、G_0 を無平方分解するのに、C を利用することができる。

例えば、次の方法が考えられる。

$G_0 / G = G_2\cdots(G_t)^{t-1}$ である。

よって、GCD(G, G_0 ／ G) = $G_2 \cdots G_t$ となり、これを G'_1 とおく。
ゆえに G ／ G'_1 = G_1 となり、G_1 が取り出せる。
以下 G_0 ／ G と GCD(G, G_0 ／ G) に対して同様の操作を行う。

無平方分解の後の処置を以下で述べる。

場合 1. g が純非分離的である時

g が純非分離的であると言うことをもう一度定義する。

定義　上の無平方分解において、$G_0 = (G_t)^t$ の時 g を純非分離的であると言う。

実際 g が純非分離的である時は、G_t = G である。
この時、degree G_t = ℓ とおくと、degree G = r = $\ell \cdot t$ である。
よって、次の step で得る Gの根 s に対して
　GCD(f, g − s) は次数 ℓ になる。この GCD(f, g − s) に対して step 4 の操作を繰り返すことになる。

場合 2. g は非分離的であるが、純非分離的ではない時

無平方分解で得られた non trivial な G_i に対して、
GCD(f, G_i(g)) $= \prod_{G_i(c_{i,j})=0} h_j$ となる。これを e_i(x) とおく。
この e_i(x) に対して、g は分離的になる(i = 1)か、純非分離的になる(i ≧ 2)。
いずれにせよ、GCD(f, G_i(g)) により、f の真に小さい因子 e_i が求まり、それらに対して、g は分離的か純非分離的となる。この操作では、次の step の finding roots を必要とせずに、f の真に小さい因子が求まる。

注) step 4 において本質的な部分は上の場合 2 の時の処置である。よって、g が分離的でも純非分離的でもない場合が幸運であると言える。

§6 step 5・6 および step 5'・6' について

finding roots の方法は、現時点では quadratic residue を用いたものしかない。Berlekamp〔3〕により、quadratic residue を用いた方法が紹介されてより、いくつかの改良がなされている。(Moenck〔7〕、Cantor-Zassenhaus〔4〕)
ここでは、特にアルゴリズムを指定しないが、上の Cantor-Zassenhaus の方法を G の finding roots および finding non-trivial factor (step 5') に適用した方法を述べておく。

G(x) の non-trivial factor H(x) を以下の様にして求める。

次数が r より真に小さい B(x) をランダムに取る。
そして GCD(G, $B^{(p-1)/2} - 1$) が non-trivial になるかどうかを調べる。

もしも、GCD(G,$B^{(p-1)/2}-1$) が non-trivial であれば、それを H(x) とおく。

H に対応する f の因子は次の様にして求まる。

補題 6. degree H = t とすれば、

$$GCD(f, H(g)) \equiv \prod_{H(c_{1,i})=0} h_i$$

となる。

証明：$H(c_{1,i}) = 0$ なる h_i に対して、$H(g) \equiv 0 \bmod h_i$ であり、
$H(c_{1,i}) \neq 0$ なる h_i に対して、$H(g) \not\equiv 0 \bmod h_i$ となる。
これにより、補題は証明される。 □

ここで degree GCD(f,H(g)) = m・t であるので、上の GCD 計算は §4 で示した方法が使える。

これで得られた H に対して、degree H = 1 になる時が、step 5 の finding roots と言える。

最後に step 6 の小細工について述べる。

G の根 s が step 5 で求まったとする。この時、H(x) = G(x)/ x − s とおく。

GCD(f,g − s) の計算においては、用いる線形方程式のサイズが 2・(r − 1)・m となってしまうが、それのコファクター GCD(f,H(g)) の計算には 2・m ですむ。

よって、先にコファクター GCD(f,H(g)) を求め、f をコファクターで割ることにより既約因子を取り出す。

step 6' の H においても、degree H > ½・r の時は、H' = G/ H に対して GCD(f,H(g)) を計算した方が効率的と言える。

本研究は第五世代計算機プロジェクトの一環としてICOTの委託で行ったものである。

References

1. E. R. Berlekamp," Factoring Polynomials over Finite Fields," Bell System Tech. J. 46, 1967, 1853-1859.
2. E. R. Berlekamp, Algebraic Coding Theory, Chap. 6, McGraw-Hill, New York, 1968.
3. E. R. Berlekamp," Factoring Polynomials Over Large Finite Fields," Math. Comp., 24, 1970, 713-735.
4. D. G. Cantor, H. Zassenhaus," A New Algorithm for Polynomials over Finite Fields," Math. Comp., 36, 1981, 587-592.
5. H. Gunji, D. Arnon," On Polynomial Factorization over Finite Fields," Math. Comp., 36, 1981, 281-287.
6. D. Lazard," On Polynomial Factorization," Lecture Note in Computer Science, 72, 1979, 126-134.
7. R. T. Moenck," On the Efficiency of Algorithms for Polynomial Factoring," Math. Comp., 31, 1977, 235-250.
8. M. Rabin," Probabilistic Algorithms in Finite Fields," SIAM J. Comp., 9, 1980, 273-280.
9. J. von zur Gathen," Parallel Algorithms for Algebraic Problems," SIAM J. Comp., 13, 1984, 802-824.
10. H. Zassenhaus," On Hensal Factorization. I ," J. Number Theory, 1, 1969, 291-311.