解　説

因数分解アルゴリズム†

古　川　昭　夫††

1. まえがき

整係数多項式の有理数体 Q 上での因数分解，また，代数的数を係数とする代数拡大体 K 上での因数分解のアルゴリズムについては1967年に Berlekamp の算法が考案されて以来，数式処理システムの進歩とともに次々と効率的な新しい算法が考案されてきている．本稿では多変数多項式の因数分解には軽くふれるにとどめ，主として1変数多項式の因数分解のアルゴリズムについて，Berlekamp の算法から最近の Lenstra の算法までを簡単に紹介する．

2. 因数分解前史

まず，整数係数多項式 $f\in\boldsymbol{Z}[x]$ の因数分解を考えよう．n次多項式

$$f=a_0+a_1x+a_2x^2+\cdots+a_nx^n \qquad (1)$$

の因数分解の基本原理は

"f の因子である可能性があるすべての多項式 g で f を割って，割りきれるかどうか調べる" (2)

ことにつきる．したがって，"f の因子の候補" を有限個に，しかも，できるだけ少数の候補にしぼるのが因数分解の基本原理である．19世紀の数学者 Kronecker は，$\boldsymbol{Z}$ における素因数分解の一意性を利用して(1)式の f の因子の候補を有限にしぼることができることを示した[1]．一方，pを素数とするとき，

$$f=g\cdot h \Rightarrow f\equiv g\cdot h \pmod{p} \qquad (3)$$

($a-b$ が p の倍数のとき $a\equiv b$ (modp) とかく．) が成立するので，因子の候補 g が mod p で f の因子でなければ，それは真の因子でないことはすぐわかる．Kronecker 以後，多くの数学者・工学者は，mod 2, mod 3 で，因子の候補をしぼってから，Kronecker の方法を用い手計算で因数分解を実行してきた．そして，それ以来，因数分解のアルゴリズムは数学者の興味をひかなかった．しかし，1960年になって数式処理システムが開発され使用されるようになってくると高次の多変数多項式を高速に因数分解する必要性が高まり，(特に記号積分において因数分解は不可欠である) それに従って以下に述べるような算法が次々と開発されたのである．

3. 因数分解の基本原理

因数分解の基本原理の1変数多項式の場合に，図式的に表すと次のようになる．

$$\begin{array}{ccc} f & \xrightarrow{\ \varphi\ } & f\equiv g_1\cdot g_2\cdots g_r \pmod m \\ \cap & & \text{[mod } m \text{ で因数分解]} \\ \boldsymbol{Z}[x] & & \end{array}$$

$$\varphi^{-1}\ \swarrow$$

$$f=g_1\cdot g_2\cdots g_r \quad \in\boldsymbol{Z}[x]$$

ここで，φ は，fに mod m での f を対応させる写像であり，φ^{-1} はその逆写像である．しかし，困ったことにどのように大きい m をとっても φ は忠実な写像にはならない．たとえば，$f=x^4+1$ は，$\boldsymbol{Z}$ 上既約であるが，どんな素数 p を法としても，可約になってしまう．この困難を解決するのは次の定理である．

(定理1，Mignotte の不等式)[2),3)]

(1)式の f に対して，$\|f\|=(|a_0|^2+\cdots+|a_n|^2)^{1/2}$ と定めると，$g=b_0+b_1x+\cdots+b_mx^m$ が $f(x)$ の因子であるならば，次の不等式が成立する．

$$\left.\begin{array}{l} |b_i|\leqq mC_i\cdot\|f\|, \\ \|g\|(2mC_m)^{1/2}\cdot\|f\|. \end{array}\right\} \qquad (4)$$

定理1より，fが与えられたとき，fの因子 g の係数の上限 B を知ることができる．$m\geqq 2B+2$ なる法 m をとって，mod m で因数分解を実行すればよいことになる．実際には，理論上の上限は実際の因子の係数に比べてはるかに小さいのが普通なので，大きな法に達する前に，実際に割算を行ってチェックするのが良い．

さて，したがってあとは十分大きい m を法とする因数分解ができれば良いことになる．その基礎となるの

† On Polynomical Factorization by Akio FURUKAWA (Dept. of Mathematics, Tokyo Metropolitan University).

†† 東京都立大学理学部数学科

は次の2定理である.

(定理2・中国式剰余定理)[1),4),5)]

各素数 $p_i(1\leqq i\leqq k)$ について

$f\equiv g_i h_i \pmod{p_i}$, $\deg(g_i)=$一定

と因数分解できるならば,

$g\equiv g_i \pmod{p_i}$, $h\equiv h_i \pmod{p_i}$

$f\equiv gh \pmod{p_1 p_2\cdots p_k}$

なる多項式 g, h を構成することができる. □

(定理3・Hensel の補題)

$f\equiv g_1 h_1 \pmod{p}$ であれば, 任意の正整数 k に対して,

$f\equiv g_k h_k \pmod{p^k}$, $g_k\equiv g$, $h_k\equiv h \pmod{p}$

なる g_k, h_k を構成できる. □

Hensel 構成の方が一つの法だけで最初の計算がすむので, 現在では Hensel 構成によって法をあげる方法が主流である. また, 多変数多項式の場合, mod p のかわりに mod $S=(x_1-\alpha_1,\cdots,x_n-\alpha_n)$ としても同様の構成をすることができる.

4. 因数分解の前処理

1変数あるいは多変数の多項式の因数分解は通常, 次の6 step で実行される.

1° $f\rightarrow \mathrm{cont}(f)\cdot \mathrm{pp}(f)$

2° $\mathrm{pp}(f)$ の主係数を1に変換

3° $\mathrm{pp}(f)\rightarrow f_1 f_2^2\cdots f_k^k$ (無平方分解)

4° $f_i\rightarrow f_{i1}\cdots f_{il} \pmod{p}$

mod p での既約因子分解

5° $f_i\rightarrow g_{i1}\cdots g_{ir}$

6° $f\rightarrow g_1\cdot g_2\cdots g_r$

これを簡単に解説しよう.

1° $\mathrm{cont}(f)$ は content of f の略で, f の係数(または係数多項式) の GCD (最大公約因子) を表し, $\mathrm{pp}(f)$ は principal part of f の略で, $f/\mathrm{cont}(f)$ を表す記号である. これらは, f の係数(または係数多項式) の GCD 計算によって求められる.

2° 以下 $l_c(f)$ で, 最高次の係数 a_n を表す. a_n は主係数 (leading coefficient) といわれる. f の主係数を1にするのは簡単で,

$a_n^{n-1}f=a_0\cdot a_n^{n-1}+a_1\cdot a_n^{n-2}(a_n x)+\cdots+(a_n x)^n$

なので, f のかわりに $a_n^{n-1}\cdot f(x/a_n)$ を考えてやればよい. ただし, 多変数多項式の場合は a_n^{n-1} が大きい多項式になることがあるので厄介である[5)].

3° 以下, 主係数は1とする. さてこのとき, f のすべての i 重因子の積を f_i とかくと,

$$f=f_1\cdot f_2^2\cdots\cdots\cdot f_k^k \qquad (5)$$

の形にかける. 各 f_i は, もはや平方因子を含まないので(5)式の形の分解を無平方分解という.

$\mathrm{GCD}(f, df/dx)=f_2 f_3^2\cdots f_k^{k-1}$

なので, GCD 計算 (ユークリッドの算法) を繰り返すことにより無平方分解が実行できる.

4° は Berlekamp の算法によって行われる. これについては次の第5章で解説する.

5° は通常 Hensel 構成によって構成する. その留意点については第6章にて述べる.

6° 因数分解された無平方因子から, 主係数をもとにもどし与えられた多項式 f の因数分解を得る.

以下の章では, Step 4°, Step 5° に関するアルゴリズムだけを考察するので, 因数分解の対象となる多項式は平方因子をもたず, 主係数が1であると仮定しておくことにしよう.

5. Berlekamp の算法

整数係数の1変数多項式を素数を法とした mod p の有限体で因数分解する方法は Berlekamp が開発したのが最初である[6)-8)].

$$f(x)\equiv f_1(x)\cdot f_2(x)\cdots\cdots\cdot f_r(x) \pmod{p} \qquad (6)$$

が f の mod p での既約因子分解であるとする. さて, 任意の r 個の整数 $s_1, s_2, \cdots\cdots, s_r$ に対して,

$$\left.\begin{array}{l} U(x)\equiv s_i \pmod{f_i(x)} \\ \deg U<\Sigma \deg(f_i)=\deg f \end{array}\right\} \qquad (7)$$

なる多項式 U が唯一つ存在する. (6)式より,

$$f_i(x)=\mathrm{GCD}(U(x)-s_i,\ f(x)) \qquad (8)$$

なる関係があるので, (7)式をみたすような U と $s_1, \cdots, s_r$ をみつけることができれば因数分解計算は(8)式の GCD 計算に帰着されることになる. (7)式をみたす $U(x)$ は,

$U(x)^p\equiv U(x) \pmod{f_i}$ $(1\leqq i\leqq r)$

をみたすゆえ

$$U(x)^p\equiv U(x) \pmod{f(x)} \qquad (9)$$

をみたす. 逆に $U^p-U\equiv U(U-1)\cdots(U-p+1)$ を考えて, (9)式をみたす U に対して, (7)式をみたす s_i たちが存在する. したがって(9)式をみたす $U(x)$ を求めれば,

$$\mathrm{GCD}(U(x)-s, f(x)) \quad (0\leqq s\leqq p-1) \qquad (10)$$

を計算することによってどれかの既約因子 f_i を分離させて求めることができる.

$U(x)=u_{n-1}x^{n-1}+u_{n-2}x^{n-2}+\cdots\cdots+u_0$

$\vec{u}=(u_{n-1}, u_{n-2}, \cdots\cdots, u_0)$

$$x^{pk} \equiv q_{k,n-1}x^{n-1} + \cdots\cdots + q_{k,0} \pmod{f(x)}$$

$$Q = \begin{pmatrix} q_{n-1,n-1} & \cdots & q_{n-1,0} \\ \vdots & & \vdots \\ q_{0,n-1} & \cdots & q_{0,0} \end{pmatrix}$$

とおくと，$U(x)^p \equiv U(x^p) \pmod{p}$ なので，

$$(9) \Leftrightarrow \vec{u}Q = \vec{u}$$
$$\Leftrightarrow \vec{u}(Q-I) = \vec{o} \qquad (11)$$

であるから $(Q-I)^{-1}(\vec{o})$ を求めればよい．ただし，ここで I は n 次単位行列を表す．(11)式の線形方程式の線形独立な解は周知の方法で求めることができる．それらを $\vec{u}_1, \vec{u}_2, \cdots, \vec{u}_{r'}$ とし，それに対応する多項式を $U_1, U_2, \cdots, U_{r'}$ とする．このとき，(11)式をみたす $\vec{u}$ は $p^{r'}$ 個存在する．一方，(7)式をみたす $U(x)$ は s_i のとり方より p^r 個存在するので，解空間の次元 r' と既約因子の数 r が一致していることがわかる．以上より，次の算法をうる．

(Berlekamp の算法)

1° 行列 Q の計算

2° $(Q-I)^{-1}(\vec{o})$ の基本解 $\vec{u}_1, \cdots, \vec{u}_r$ の計算

3° $r=1$ ならば既約．

$r \neq 1$ ならば，

GCD $(U_i - s, f)$ $(0 \leqq s \leqq p-1)$

を計算し，r 個の因子 f_i を構成．

さて，この算法の計算量は 1° が $O(n^2 \log pn)$，2° が $O(n^3)$，3° が $O(p)n^2 r \approx O(pn^3)$ であり，全体として $O(pn^3)$ のオーダの計算量となる．

大きい素数 p を法とするとき，3° のステップにおいて，GCD $(U_i - s, f)$ はほとんどすべての $0 \leqq s < p$ について1になっているので，非常に無駄である．大きい素数 p の場合には，$U_i(x)$ が求まった時点で，s をパラメータとし $U_i(x) - s$ と $f(x)$ の終結式 Res $(U_i(x) - s, f(x)) = r(s)$ を計算し，$r(s) = 0$ なる s についてのみ GCD 計算をする方が簡単である．$r(s) = 0$ かどうかのチェックは，p が $p = L \cdot 2^l + 1$ (L は小さくて $L \neq l$) なる形をしているときには効率的に行うことができることが知られている[9]．

6. Hensel 構成

Macsyma, REDUCE などの数式処理システムでは，Hensel の補題にもとづいた Hensel 構成を用いて，mod p での因数分解から mod p^k への因数分解が作られる．

(Hensel 構成の算法)

1° 与えられた多項式 f を Berlekamp 算法により互いに素な2つの多項式の積に分解する．

$$f \equiv g_1 \cdot h_1 \pmod{p}$$

2° $f - g_k h_k \equiv p^k \cdot f_k \pmod{p^{k+1}}$ なる多項式 $f_k \in \boldsymbol{Z}[x]$ に対し，

$$A_k g_1 + B_k h_1 \equiv f_k \pmod{p}$$

なる $A_k, B_k \in \boldsymbol{Z}[x]$ をユークリッドの算法により求める．

3° $g_{k+1} = g_k + p^k A_k$,

$h_{k+1} = h_k + p^k B_k$

とおく．以下 1°, 2° をくり返し，f の因子の係数の上限を M とするとき，$p^{k+1} \geqq 2M+2$ となるまで法をあげる．

拡張されたユークリッドの算法を用いれば，$g_{11}, g_{21}, \cdots, g_{r1}$ が互いに素であれば，

$$A_{1k}g_{11} + A_{2k}g_{21} + \cdots + A_{rk}g_{r1} \equiv f_k \pmod{p}$$

なる多項式 $A_{1k}, A_{2k}, \cdots, A_{rk}$ を構成することもできるので，Hensel 構成は容易に3つ以上の互いに素な因子に対しても拡張することができる．また，法を $p \to p^2 \to p^3 \to p^4$ と1つずつあげるだけでなく，$p \to p^2 \to p^4 \to p^8$ と平方的にあげる方が一般的には効率的である[5),10)]．多変数多項式 $f(x, x_1, \cdots, x_n)$ の場合は，1つの変数 x を主変数とみて，

$$f(x, x_1, \cdots, x_n) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_0$$

の形で取り扱い，$S = (x_1 - \alpha_1, x_2 - \alpha_2, \cdots, x_n - \alpha_n)$ なるイデアルを考え，mod (p^l, s^t) で考えることによって，同様に Hensel 構成を行うことが可能である．ただし，主係数を1に調整してから行うのが厄介な場合には別の工夫が必要である[4),5),11)]．

7. 整係数多項式因数分解の標準的算法

現在多くのシステムでインプリメントされている，整係数の1変数多項式の因数分解アルゴリズムは，Zassenhaus の算法といわれている次のアルゴリズムである．

(Zassenhaus のアルゴリズム)

1° $f(x) \in \boldsymbol{Z}[x]$ から係数の GCD をとりだし，cont $(f) \cdot$ pp (f) と分解

2° 原始多項式 (cont $(f) = 1$ なる多項式) f を無平方分解 $f_1 f_2^2 \cdots f_m^m$ (GCD 計算は部分終結式算法を用いるか，mod をとって Hensel 構成を行う)．

3° 無平方多項式 $f(x)$ を主係数が1になるように変換．

4° f が無平方となるような変数 p をとり，

$$f \equiv f_1, f_2 \cdots\cdots f_r \pmod{p}$$

と因数分解 (Berlekamp 算法).

5° 十分大きい p^k まで，Hensel 構成で

$f \equiv f_1, f_2 \cdots\cdots f_r \pmod{p^k}$

と因数分解.

6° $F=\{f_1, f_2, \cdots, f_r\}$ の部分集合，$G=\{g_1, g_2, \cdots, g_l\}$ をひとつとり，$g=g_1g_2\cdots g_l$ を作り，$\boldsymbol{Z}[x]$で f が g で割りきれるかどうかチェック．もし割りきれれば，G の真部分集合 $H=\{h_1, \cdots, h_i\}$ をとり，$h=h_1 \cdot h_2 \cdots h_i$ が $\boldsymbol{Z}[x]$ で g を割りきるかを調べる．すべての h について g が h で割りきれなければ，g は $\boldsymbol{Z}$ 上 f の既約因子なので，$F=F-G$, $f=f/g$ として以下同様に既約因子を探す．ある h について g が h について割りきれれば，hをあらたなる g として同様の作業をつづけ，fの既約因子をうる．

また，どのような g も fを割りきらなければ，fは $\boldsymbol{Z}$ 上既約であるとわかる．

7° 主係数をもとにもどす.

さて，このアルゴリズムにおいて，法となる pとしてあまり小さい素数 pをとると，既約因子が分解しすぎるために非効率的であり，経験上2ケタの大きくない素数をとることが良い．このアルゴリズムは実用上非常に効率よく因数分解を実行するが，n次多項式 f は $\boldsymbol{Z}$ 上既約でも，mod p では n 個の因子をもつことがあり得るので，ステップ6の g の組み合わせとして最悪の場合 2^{n-1} 通りを調べることになる．したがって，最悪の場合の計算量は f の次数 n について指数関数的な計算になってしまうのが難点である．これに対し，Lenstra は，Hensel 構成で作った因子から，$\boldsymbol{Z}[x]$ の因子を構成するまったく別の方法を1982年から1983年にかけて発表した[12)~14)]．この方法によると，n の多項式時間（より正確には n^6 のオーダ）で因数分解を実行するアルゴリズムを作ることができる[15)]．Lenstra の算法については，第9章，第10章にて詳述する．

8. 代数的拡大体での因数分解

α を有理数体 $\boldsymbol{Q}$ 上代数的な元とし，$K=\boldsymbol{Q}(\alpha)$ を α を含む最小の体とする．α のみたす $\boldsymbol{Z}$ 係数の最小次数の多項式（すなわち最小多項式）を，$F(t)$ とする．f が K上無平方で原始的とするとき，f の K上での因数分解の標準的な方法は次のとおりである[16)]．

(Weinberger Rothchild のアルゴリズム)

1° 最小多項式 $F(t)$ を mod p で因数分解.

$F(t) \equiv F_1, F_2, \cdots, F_k \pmod{p}$

2° F_i の次数を n_i とすると，(mod p, $F_i(t)$) を考えると位数 p^{n_i} の有限体 $GF(p^{n_i})$ と同一視できる．有限体では Berlekamp の算法が使えるので，すべての i について

$f(x) \equiv f_{i1}, \cdots, f_{ij_i} \pmod{p, F_i(t)}$

と因数分解.

3° 一般化された中国式剰余定理の算法を用いて，

$f(x) \equiv f_1, \cdots, f_m \pmod{p, F(t)}$

を構成.

4° Hensel 構成で，

$f(x) \equiv f_1 \cdots f_m \pmod{p^k, F(t)}$

5° $\{f_1, \cdots, f_m\}$ の積の組み合わせを，mod $F(t)$ で割りきれるかどうかチェック．($\boldsymbol{Q}(\alpha)[x]$ は，剰余環 $\boldsymbol{Q}[t, x]/F(t)$ と同型であるので，α の多項式を考えるのは，tの多項式を mod $F(t)$ で考えるのと同じである.) この算法でもステップ5のところで，指数的計算量になってしまう．また，アルゴリズムの効率が $F(t)$ のmod p における既約性に大きく依存することも欠点である.

9. 整数格子を用いた $\boldsymbol{Z}[x]$ での因数分解

さて，$\boldsymbol{Z}[x]$ での因数分解を行う Zassenhaus のアルゴリズム，α を代数的整数とするとき $\boldsymbol{Z}[\alpha][x]$ での因数分解を行う Weinberger-Rothchild のアルゴリズムでは，共に Hensel 構成された既約因子の候補から，既約因子を発見するまでに，次数に関して指数的な計算量が（最悪の場合）必要となることはすでにみた．それに対して，Lenstra は mod で因数分解された因子の係数の作るベクトルの生成する整数格子 (lattice) から $\boldsymbol{Z}$ 上での既約因子を求める方法を考案した[12)]．以下，その方法を解説する.

$f(x)=a_0+a_1x+a_2x^2+\cdots+a_nx^n$, $a_n \neq 0$

に対して，$f=(a_0, a_1, \cdots, a_n)$ なるベクトルと同一視する．また，$(a_0, a_1, , \cdots, a_n)$ と $(a_0, a_1, \cdots, a_n, 0, \cdots, 0)$ を同一視する．さて，$f \in \boldsymbol{R}^{n+1}$ に対して，[ ] を通常のガウス記号として，

$[f]=([f_0+1/2], [f_1+1/2], \cdots, [f_n+1/2])$

と定め，fの長さ $\|f\|$ を，

$\|f\|=(f_0^2+f_1^2+\cdots+f_n^2)^{1/2}$

と定める．また，以下では $f(x)$ は原始的（係数の GCD が1）を仮定する.

さて，$\boldsymbol{b}_0, \boldsymbol{b}_1, \cdots, \boldsymbol{b}_n \in \boldsymbol{Z}^{m+1}$ に対して，その定める整数格子とは，集合

$L=\boldsymbol{Z}\boldsymbol{b}_0+\boldsymbol{Z}\boldsymbol{b}_1+\cdots+\boldsymbol{Z}\boldsymbol{b}_m$ (12)

であり，これを，$L=(\boldsymbol{b}_0, \boldsymbol{b}_1, \cdots, \boldsymbol{b}_m)$ で表す．

$$d(L)=|\boldsymbol{b}_0, \boldsymbol{b}_1, \cdots, \boldsymbol{b}_m \text{ の作る行列式}| \tag{13}$$

$$FD(L)=\{\boldsymbol{x}\in\boldsymbol{R}^{m+1} \mid {}^{\exists}c_j\in[-1/2, 1/2) \quad \boldsymbol{x}=\textstyle\sum_{j=0}^{m} c_j\boldsymbol{b}_j\} \tag{14}$$

と定める．$d(L)$ はいわば，lattice L の1区画の体積であり，FD(L)（L の基本領域という）は，原点を中心とした格子 L の1区画そのものであると考えて差しつかえない．

$$\boldsymbol{x}\equiv\boldsymbol{y} \pmod{L} \Leftrightarrow \boldsymbol{x}-\boldsymbol{y}\in L$$

と定めると，任意の $\boldsymbol{x}\in\boldsymbol{R}^{m+1}$ に対して

$$\tilde{\boldsymbol{x}}\equiv\boldsymbol{x} \pmod{L},\ \tilde{\boldsymbol{x}}\in \mathrm{FD}\ (L)$$

をみたす $\tilde{\boldsymbol{x}}$ が唯一つ存在する．具体的には，L を定める行列を M とするとき，$\tilde{\boldsymbol{x}}=\boldsymbol{x}-[\boldsymbol{x}M^{-1}]\cdot M$ なる $\tilde{\boldsymbol{x}}$ をとればよい．

さて，L の基底 $\boldsymbol{b}_0, \boldsymbol{b}_1, \cdots, \boldsymbol{b}_m$ が与えられたとき，シュミットの直交化法により

$$\boldsymbol{b}_i^*=\boldsymbol{b}_i-\textstyle\sum_{j=1}^{i-1}\mu_{ij}\boldsymbol{b}_j^* \quad (0\leqq i\leqq m)$$

$$\mu_{ij}=\frac{(\boldsymbol{b}_i, \boldsymbol{b}_j^*)}{\|\boldsymbol{b}_j\|^2} \quad (1\leqq j<i<n) \tag{15}$$

として，$\boldsymbol{b}_0^*, \boldsymbol{b}_1^*, \cdots, \boldsymbol{b}_m^*$ なる直交基底を作ることができる．ここで，$(\boldsymbol{b}_i, \boldsymbol{b}_j^*)$ はベクトル $\boldsymbol{b}_i$ と $\boldsymbol{b}_j$ の内積を表す．このとき

（定義 1） $M=(\boldsymbol{b}_0, \boldsymbol{b}_1, \cdots, \boldsymbol{b}_m)$ が簡約化基底であるとは(15)式の方法で作った直交化基底を $M^*=(\boldsymbol{b}_0^*, \boldsymbol{b}_1^*, \cdots, \boldsymbol{b}_m^*)$ とするとき，

$$\mu_{ij}\leqq 1/2 \tag{16}$$

$$\|\boldsymbol{b}_i^*+\mu_{i\,i-1}\boldsymbol{b}_{i-1}^*\|\geqq\sqrt{3}/2\cdot\|\boldsymbol{b}_{i-1}^*\| \tag{17}$$

が成立することである．

このとき，

$$\|\boldsymbol{b}_i^*\|\geqq\|\boldsymbol{b}_{i-1}^*\|/\sqrt{2} \tag{18}$$

なる不等式も成立する[15]．

（定理 4, Kaltofen の基底簡約）[12),15)]

$(m+1)$ 次元整数ベクトル $\boldsymbol{b}_0, \boldsymbol{b}_1, \cdots, \boldsymbol{b}_m$ が与えられたとき，1) $(\boldsymbol{\delta}_0, \boldsymbol{\delta}_1, \cdots, \boldsymbol{\delta}_m)=(\boldsymbol{b}_0, \boldsymbol{b}_1, \cdots, \boldsymbol{b}_m)$ 2) $\boldsymbol{\delta}_0, \boldsymbol{\delta}_1, \cdots, \boldsymbol{\delta}_m$ が簡約化基底となる整数ベクトル $\boldsymbol{\delta}_0, \boldsymbol{\delta}_1, \cdots, \boldsymbol{\delta}_m$ を m について m^6 のオーダで計算することができる． □

このアルゴリズムについては参考文献 12), 15) を参照されたい．このアルゴリズムがmについて多項式的なために，因数分解の多項式時間アルゴリズムが構成できるのである．

さて，Lenstra は 1)mod p^k での因数分解の因子から作った整数格子の最小ベクトルから既約因子を求める方法[18]，2) 1) と同じ整数格子の簡約化基底に対応する多項式の GCD から既約因子を求める方法[12]，3) $f(x)$ の複素近似根 α から整数格子を生成し，α の最小多項式として既約因子を求める方法[17]の3つの格子算法を発表している．3)の算法については参考文献 17)を参照していただくことにして，ここでは 1), 2)の算法を紹介する．

（格子算法 A・m 次既約因子を求める）

1° 原始多項式 $f(x)$ を mod p^k で因数分解し，主係数が1の既約因子の1つを h_k とする．(Berlekamp, Hensel 算法) ただし $n=\deg h_k$ とし，f の因子 g の大きさ $\|g\|$ の上限を B とするとき，

$$B^{2m}<p^{kn} \tag{19}$$

なるよう十分大きい k をとる．このとき，

$$g|h_k \pmod{p^k},\ g|h \text{ (in } \boldsymbol{Z}[x]) \tag{20}$$

なる m次既約式 g を次の手順で求める．

$$\left.\begin{array}{ll} 2^\circ\ \boldsymbol{b}_i=p^k\cdot x^i & i=0, \cdots, n-1 \\ \boldsymbol{b}_i=h_k\cdot x^{i-n} & i=n, \cdots, m \end{array}\right\} \tag{21}$$

から，$L_k=\boldsymbol{Z}\boldsymbol{b}_0+\boldsymbol{Z}\boldsymbol{b}_1+\cdots+\boldsymbol{Z}\boldsymbol{b}_m$ なる格子 L_k を生成．

3° L_k の中の最小ベクトル $\tilde{\boldsymbol{b}}_0$ を探し，それが m 次式ならば，$g=\tilde{\boldsymbol{b}}_0$ が求めるm次既約因子である．

この算法を $m=1, 2, \cdots, \deg f-1$ まで実行することによりすべての既約因子を得る．なお，$m=\deg f$, のときは，f は既約ということになる．

さて，この算法の正当性を以下に示そう．

（命題 2） f_1, f_2 が $\boldsymbol{Z}$ 上互いに素で，$n_1=\deg f_1\geqq n_2=\deg f_2\geqq 1$ とする．p^k を素数のべき乗とし，mod p^k では f_1, f_2 は首係数が1の共通因子 $h_k(\deg h_k=n\geqq 1)$ をもつものとすると，

$$p^{kn}\leqq\|f_1\|^{n_2}\|f_2\|^{n_1} \tag{22}$$

が成立する． □

（証明） $\tilde{\boldsymbol{b}}_i=f_1x^i,\ i=0, \cdots, n_2-1,$

$\tilde{\boldsymbol{b}}_i=f_2\cdot x^i,\ i=n_2, \cdots, n_2+n_1-1,$

で生成される整数格子を L とすると，L は

$$\{Af_1+Bf_2 \mid \deg A<n_2,\ \deg B<n_1\} \tag{23}$$

なる多項式の集合に対応している．(23)式の多項式は，(n_1+n_2-1) 次以下の多項式で，mod p^k では h^k を因子にもつことに注意しよう．一方，

$$\left.\begin{array}{ll} \boldsymbol{b}_i=p^k\cdot x^i, & i=0, \cdots, n_1-1, \\ \boldsymbol{b}_i=h_k\cdot x^{i-n}, & i=n_1\cdots, n_1+n_2-1, \end{array}\right\} \tag{24}$$

で生成される整数格子を L_k とすると，L_k は (n_1+n_2-1) 次以下の多項式で，mod p^k で h_k を因子にもつような多項式の全体と対応している．したがって，L は L_k の部分格子であり，d(L) は d(L_k) の整数

倍になっている．

$$d(L_k)=P^{kn}\leqq d(L)\leqq\|f_1\|^{n_2}\|f_2\|^{n_1}$$

より題意の不等式が成立する．（おわり）

（命題 3） 格子算法Aのステップ2で作られた整数格子 L_k の最小ベクトル $\vec{g}$ が m 次式ならば f の既約因子である．

（証明） (19)式で作られた整数格子 L_k は，

$$\left.\begin{aligned}&F=h_k\cdot Q+R\cdot P^k,\\&\deg Q\leqq m-n,\ \deg R<\deg h_k=n\end{aligned}\right\}\quad(25)$$

をみたす整係数多項式の全体である．すなわち，

「mod p^k で h_k を因数にもつ，

次数が m 以下の整係数多項式」． (26)

の全体にほかならない．求める既約因子 g も(26)式をみたすので，$g\in L_k$ である．したがって，あとは g と L_k の最小元が一致することをいえばよい．(26)式をみたす長さの最小元を $\vec{g}$ とし $\vec{g}$ は m 次式とする．$\vec{g}$ と g は次数が等しく，g が既約であることを考えると，$\vec{g}$ は g の整数倍であるかあるいは $\vec{g}$ と g は互いに素であるかどちらかの場合しかない．後者の場合，命題2より，$p^{kn}\leqq\|g\|^m\|\vec{g}\|^m$ が成立する．一方，$\|g\|$，$\|\vec{g}\|\leqq B$ と(19)式の不等式からこれは矛盾である．よって，$\vec{g}=l\cdot g(l\in\boldsymbol{Z})$ とかけねばならない．$\|\vec{g}\|=|l|\|g\|$ なので，$\vec{g}$ の最小性より，$l=\pm1$．よって，最小ベクトル $\vec{g}$ は既約である．（おわり）

例 $f=96x^8+80x^7-156x^6-58x^5+101x^4-39x^3-29x^2+8x-24$ の場合，

$p=5$ とすると，mod p で $f(3)\equiv0$ なので

$$f\equiv(x+2)\cdot(x^7+3x^6+3x^5+x^4+4x^3+3x^2+3)\pmod 5$$

なる因数分解を得る．Hensel 構成して，$(x-515858)\pmod{5^{10}}$ なる因子を得る．

これより

$$\boldsymbol{b}_0=(5^{10},0,0,0,0),\ \boldsymbol{b}_1=(-515858,1,0,0,0)$$
$$\boldsymbol{b}_2=(0,-515858,1,0,0),$$
$$\boldsymbol{b}_3=(0,0,-515858,1,0)$$
$$\boldsymbol{b}_4=(0,0,0,-515858,1)$$

なる基底を得る．これから最小元を含むような別の基底を構成すると，

$$\boldsymbol{b}_0=(0,23,5,-43,-3),\ \boldsymbol{b}_1=(-24,5,16,13,-1)$$
$$\boldsymbol{b}_2=(19,12,7,-2,16),\ \boldsymbol{b}_3=(-3,1,-7,0,8)$$
$$\boldsymbol{b}_4=(-5,-8,13,-10,17)$$

となり，$\boldsymbol{b}_3$ が最小ベクトルである．実際，

$$f=(8x^4-7x^3+x-3)(12x^4+10x^3-9x^2+8)$$

となっている．（例おわり）

整数格子と多項式の因数分解の関連性については，命題 2,3 の証明とこの例で理解していただけたと思う．しかし，残念ながら，最小ベクトルを求める算法は整数格子の次数について多項式的ではなく指数的である．一方，(16)，(17)式をみたす簡約化基底は多項式時間で計算できる．そこで，Lenstra は簡約化基底から既約因子を構成する次のような算法を構成した．

（格子算法B）

1° 次数 l の原始多項式 $f(x)$ を mod p^k で因数分解し，主係数が1の既約因子 h_k を求める．ここで，$n=\deg h_k$ とし，

$$p^{kn}>2^{l(l-1)/2}(2l-2C_{l-1})^{l/2}\|f\|^{2l-1}\quad(27)$$

をみたすように k を十分大きくとる．このとき，

$$g\mid h_k(\bmod p^k),\ g\mid h(\text{in } \boldsymbol{Z}[x])$$

なる m 次既約式 g を次の手順で求める．

2° $\boldsymbol{b}_i=p^kx^i,\ i=0,\cdots,n-1$

$\boldsymbol{b}_i=h_k\cdot x^{i-n},\ i=n,\cdots,m$

をみたす整数格子 L_k を作る．

3° L_k に定理4の算法で(16)，(17)式をみたす基底 $(\boldsymbol{b}_0,\boldsymbol{b}_1,\cdots,\boldsymbol{b}_m)$ を作る．（$\|\boldsymbol{b}_0\|\leqq\|\boldsymbol{b}_1\|\leqq\cdots\leqq\|\boldsymbol{b}_n\|$ となる．）

4° $M=(p^{kn}/\|f\|^n)^{1/l}$ とする．

$$\|\boldsymbol{b}_0\|\leqq\cdots\leqq\|\boldsymbol{b}_i\|<M\leqq\|\boldsymbol{b}_{i+1}\|\quad(28)$$

をみたす $\boldsymbol{b}_0,\cdots,\boldsymbol{b}_i$ をとるとき，GCD $(\boldsymbol{b}_0,\boldsymbol{b}_1,\cdots,\boldsymbol{b}_i)=g$ が求める既約因子となる．（$\|\boldsymbol{b}_0\|\geqq M$ ならば，m 次既約因子は存在しない．）

この算法の正当性の証明については参考文献 12)，15) を参照されたい．

10. 代数体上での格子算法

代数体 $K=\boldsymbol{Q}(\alpha)$ における因数分解を考えよう．代数的整数 α の最小多項式 $F(t)$ がある mod p で既約ならば (mod p, $F(t)$) で Berlekamp の算法を適用し Hensel 構成するのが最良な方法である．しかし，どんな素数 p に対しても $F(t)$ が分解されてしまう場合もある．この場合，次のような格子算法が有用である．以下，簡単のため $f\in K[x]$ は主係数が1で無平方な s 次多項式，F は m 次多項式とする．

（格子算法 C，$\boldsymbol{Q}(\alpha)$ 上での因数分解）

1° $F(t)$ を mod p で因数分解し，$H_k(t)$ を mod p^k での主係数が1の F の既約因子とする．ただし，

$$H_k\equiv H_{k-1}(\bmod p^{k-1})$$

$$m>\deg(H_k)=\cdots=\deg(H_1)=n\geqq1$$

とする．このとき，H_k の根を α_k とおくと，

$$\{\Sigma a_i\alpha_k{}^i \mid 0\leqq a_i<p, 0\leqq i<n\}$$

は $q=p^n$ を位数とする有限体 F_q とみなすことができる．これを $W_k(F_q)$ とかく．

2° f を $W_k(F_q)$ 上の多項式とみて

$$f\equiv f_1, f_2\cdots f_r \pmod{p,\ H_k(t)}$$

と既約因子分解．さて，f の u 次既約因子 $g\in K[x]$ をみつけたい．ここで，g は，

$$g\in((1/D)\boldsymbol{Z}[\alpha])[x],\ \mathrm{lc}(g)=1/D$$

と仮定してよい[16]．ただし，ここで D は f から決まるある正整数である．この g をみつけ出そう．（以下，簡単のため $D=1$ と仮定する．）

$$g=\sum_{i=0}^{u} v_i x^i\in\boldsymbol{Z}[\alpha](x)$$

$$v_i=\sum_{j=0}^{m-1} v_{ij}\alpha^j\in\boldsymbol{Z}[\alpha]$$

とする．v_i を $\boldsymbol{Z}$ 上の多項式と以下みなす．（v_i を多項式とみたとき係数ベクトルの大きさ $\|v_i\|$ は f と F によってある上限 B で押さえられる[19]）

3° Hensel 構成により

$$f=f_1, f_2\cdots f_r \pmod{p^k, H_k} \qquad (29)$$

なる分解を構成．

4° (29)式の r 個の因子から，任意の組み合わせの積を g_k とする．（このようなことを 2^r 回行う．）この中から f の既約因子 g となるものを以下の手順で発見する．

5° $i\in\{0, 1, \cdots, u=\deg g\}$ とし，$\tilde{v}_k\in\boldsymbol{Z}[\alpha]$ を g_k の i 次の係数とし，$v=v_i\in\boldsymbol{Z}[\alpha]$ を g の i 次の係数とする．$v\equiv\tilde{v}_k \pmod{p_k, H_k}$ なる v を求めたい．v は $(m-1)$ 次以下の多項式だから，

$$\left.\begin{aligned} &v=\tilde{v}_k+w_1H_k+p^k\cdot w_2\\ &\deg w_1\leqq(m-1)-n,\ \deg w_2\leqq n-1\end{aligned}\right\} \qquad (30)$$

の形にかけているはずである．

6° $\boldsymbol{b}_i=p^kx^i,\ i=0, \cdots, n-1$

$\boldsymbol{b}_i=H_k\cdot x^{m-1-i},\ i=n, \cdots, m+1$

として m 次元整数格子 L_k を構成．

7° (14)式で定められる基本領域が，半径 B の球を含むような L_k の新しい基底 $\tilde{\boldsymbol{b}}_0, \tilde{\boldsymbol{b}}_1, \cdots, \tilde{\boldsymbol{b}}_{m+1}$ を構成．この基底の作る行列を M とする．

8° $v=\tilde{v}_k-[\tilde{v}_kM^{-1}]\cdot M$ とすると，v が求める g の α^i の係数 v_i となる．

この算法の正当性の証明については参考文献 18)にゆずるとして，ここでも1つ例をあげよう．

例 代数的数 α の最小多項式を

$F(t)=t^6+3t^5+6t^4+t^3-3t^2+12t+16$ とし，

因数分解する $K=\boldsymbol{Q}(\alpha)$ 上の多項式を $f(x)=x^3-3$ とする．

$H_1=(t^3+t^2-2t+3) \pmod 7$

$H_8=(t^3-1399040t^2-1399043t-4) \pmod{7^8}$

であり，

$$f\equiv(x-2387947\alpha-2387948)\times(x+2387948\alpha+1)\times(x-\alpha+2387947) \pmod{7^8, H_8(\alpha)}$$

とわかる．これから L_5 を作ると，

$$\begin{bmatrix} 5764801, & 0, & 0, & 0, & 0, & 0\\ 0, & 5764801, & 0, & 0, & 0, & 0\\ 0, & 0, & 5764801, & 0, & 0, & 0\\ -4, & -1399043, & -1399040, & 1, & 0, & 0\\ 0, & -4, & -1399043, & -1399040, & 1, & 0\\ 0, & 0, & -4, & -1399043, & -1399040, & 0 \end{bmatrix}$$

なる整数格子を得る．基底の取り替えにより，

$$M=\begin{bmatrix} 1265, & 479, & 547, & -752, & -957, & -1299\\ -1265, & -273, & 547, & -2017, & -205, & -1299\\ -1059, & -547, & -137, & 2359, & -957, & -376\\ -1265, & 683, & -34, & -752, & -1231, & 2051\\ 0, & 2530, & 752, & 0, & 1265, & -752\\ -103, & 34, & 1641, & -171, & 205, & 376 \end{bmatrix}$$

この逆行列 M^{-1} を有効数字5桁で計算し，この場合の分母として $D=12$ をとると，

$$f=(x+(168641\alpha+168629)/12)\times(x-(168629\alpha-12)/12)$$

$$\searrow\quad\times(x-(12\alpha+168641)/12)$$

なる因数分解を得る．（例おわり）

11. 付 記

本稿では多変数多項式の因数分解，多変数多項式の代数体上での因数分解については解説できなかった．前者については，参考文献 11)，16)，20)，21)，22)を参照されたい．後者については，参考文献 19)，23)を参照されたい．また，有限体上における多変数多項式の多項式時間的算法については，参考文献 24)，25)を参照されるとよい．

参考文献

1) Van der Waerden, B.L.: Algebra I・II, Springer-Verlag, Berlin (1973).
2) Mignotte, M.: An In Equality about Factors of Polynomials, Math, Comp., 28, pp. 1153-1157 (1974).
3) Mignotte, M.: Some Inequalitys about Univariate Polynomials, Proc, 1981 ACH SYMSAC pp. 195-199 (1981).
4) 佐々木建昭他：数と式と文の処理（岩波情報科学第23巻），岩波書店，pp. 101-137 (1981).
5) 佐々木建昭：数式処理（情報処理叢書7），情報処理学会発行，オーム社 (1981).
6) Berlekamp, E.R.: Algebraic Coding Theory, McGraw-Hill, New York pp. 146-175, (1968).
7) Berlekamp, E.R.: Factoring Polynomials over Large Finite Fields, Math, Comp., 24, pp. 713-735 (1970).
8) Knuth, D.E.: The Art of Computer Programming Vol. 2. Seminumerical Algorithms, Addison Wesley, New York, pp. 381-398 (1969)
9) Moenk, R.T.: On the Efficiency for Polynomial Factoring, Math, Comp., 137, pp. 235-250 (1977).
10) Zassenhaus, H.: On Hensed Factorization I, J. Number Theory, 1, pp. 291 (1969).
11) Wang, P.S. and Rothschild, L.P.: Factoring Multivariate Polynomials over the Integers, Math. Comp., 29, pp. 935-950 (1975).
12) Lensta, A.K., Lenstra, H.W. and Lovasz, L.: Factoring Polynomials with Rational Coefficients, Math. Ann., 261, pp. 515-534 (1982).
13) Lenstra, A.K.: Lattice and Factorization of Polynomials over Algebraic Number Fields, Lecture Note in Computer Science 144, Springer, New York, pp. 32-39 (1982).
14) Lenstra, A.K.: Factoring Polynomials over Algebraic Number Fields, Lecture Note in Computer Science 162, Springer, New York pp. 245-254 (1983).
15) Kaltofen, E.: On the Complexity of Finding Short Rectors in Integer Lattices, Lecture Note in Comp. Sci. 162, Springer, New York, pp. 236-244 (1983).
16) Weinberger, P.J. and Rothchild, L.P.: Factoring Polynomials over Algebraic Number Fields, ACM Trans, Math. Soft 2, pp. 335-350 (1976).
17) Lenstra, A.K.: Polynomial Factorization by Root Approximation, Lecture Note in Comp. Sci. 174, Springer, New York, pp. 272-274 (1982).
18) Lenstra, A.K.: Lattices and Factorization of Polynomials, preprint, Mathematic Center of Amsterdam, Holland (1982).
19) Wang, P.S.: Factoring Maltivariate Polynomials over Algebraic Number Fields, Math. Comp., 30, pp. 324-336 (1976).
20) Wang, P.S.: An Improved Multivariate Integral Polynomial Factoring Algorithm, Math. Comp., 32, pp. 1215-1231 (1978).
21) Lensttra, A.K.: Factoring Multivariate Integral Polynomials, Lecture Note in Comp. Sci, 154, Springer, New York, pp. 458-465 (1983).
22) Kaltofen, E.: Polynomial-Time Reductions from Multi-variate to Bi-and Univariate Integral Polynomial Factorization, SIAM. J. Comp., 14, pp. 469-489 (1985).
23) Lenstra, A.K.: Factoring Multivariate Polynomials over Algebraic Number Fields, Lecture Note in Comp, Sci. 176, Springer, New York, pp. 389-396 (1984).
24) Gathen, J. and Kaltofen, E.: Polynomial-Time Factorization of Multivariate Polynomials over Finite Fields, Lecture Note in Comp. Sci. 154, Springer, New York, pp. 250-263 (1982).
25) Lenstra, A.K.: Factoring Multivariate Polynomials over Finite Fields, J. Computter and System Science, 30, pp. 235-248 (1985).

(昭和60年12月9日受付)