暗号アルゴリズムの詳細評価に関する報告書

2002 年 3月 18日

立 教 大 学

木田 祐司

Chapter 1

素因数分解実験プロジェクト

2002/1/19　CRYPTREC 研究会 at 立教大学 S302 教室
2003/3/31 改訂
木田 祐司

この文書について
ここに集めた文書は2001年度に立教大学で行われたセミナーの配布資料である[1]。
Chapter 1 で全体の概論について述べ、Chapter 2-5 で各論の概論を述べる。これ以上の詳細な内容は2002,2003年度のセミナーで研究される予定である。

1.1　このプロジェクトの目的

1024bits の合成数の素因数分解にかかる時間を評価すること。そのために

1. 一般数体ふるい法(GNFS) のプログラムを独自に開発し、数値実験を行う。

2. GNFS の理論的解析を行い、改良の余地を研究する。

より広い観点からの評価はすでに次の論文で行われている。我々の目的はこれらと（ある程度）独立に評価を行うことである。言いかえると、これらの論文の関連する部分の妥当性を判断することでもある。

- [Arjen K. Lenstra, Eric R. Verheul] "Selecting Cryptographic Key Sizes", `http://www.cryptosavvy.com/Joc.pdf`
- [Robert D. Silverman] "A Cost-Based Security Analysis of Symmetric and Asymmetric Key Lengths", RSA Laboratories
  `ftp://ftp.rsasecurity.com/pub/pdfs/bulletn13.pdf`

1.2　素因数分解の現状

Scott Contini の home page は非常に役に立つ。この小論でもここにある文書をたびたび紹介する。
`http://www.crypto-world.com/FactorAnnouncements.html`

[1] 実際のセミナーで補充した内容は収められていない。また冗長なため削除した部分もそのまま残している。読みやすくはないことをあらかじめお断りしておく。

1.2.1 各種の分解法

1. 試し割り算
2. $p-1$ 法
3. モンテカルロ法(ρ 法)
4. 連分数法
5. 楕円曲線法
6. 二次ふるい法
7. 数体ふるい法

1.2.2 1970 年以降の記録

TABLE 1: Historical Factoring Records
Silverman の前記報告文書より引用および追加

Year	Size	Number	Who	Method	Hardware
1970	39	$2^{128}+1$	Brillhart/Morrison	CFRAC	IBM Mainframe
1978	45	$2^{223}-1$	Wunderlich	CFRAC	IBM Mainframe
1981	47	$3^{225}-1$	Gerver	QS	HP-3000
1982	51	$5^{91}-1$	Wagstaff	CFRAC	IBM Mainframe
1983	63	$11^{93}+1$	Davis/Holdridge	QS	Cray
1984	71	$10^{71}-1$	Davis/Holdridge	QS	Cray
1986	87	$5^{128}+1$	Silverman	MPQS	LAN Sun-3's
1987	90	$5^{160}+1$	Silverman	MPQS	LAN Sun-3's
1988	100	$11^{104}+1$	Internet	MPQS	Distributed
1990	111	$2^{484}+1$	Lenstra/Manasse	MPQS	Distributed
1991	116	$10^{142}+1$	Lenstra/Manasse	MPQS	Distributed
1992	129	RSA-129	Atkins	MPQS	Distributed
1996	130	RSA-130	Montgomery	GNFS	Distributed
1998	140	RSA-140	Montgomery	GNFS	Distributed
1999	155	RSA-512	Montgomery	GNFS	Distributed
2002	158	$2^{953}+1$	Bahr/Franke/Kleinjung	GNFS	LAN PC's ?

CFRAC = Continued FRACtion = 連分数
QS = Quadratic Sieve = 二次ふるい
MPQS = Multiple Polynomial Quadratic Sieve = 複数多項式二次ふるい
GNFS = General Number Field Sieve = 一般数体ふるい

TABLE 2: Historical Factoring Records: Special Types
Silverman の前記報告文書より引用および追加

1990	148(155)	$2^{2^9}+1$	Lenstra/Manasse	SNFS	Distributed
1999	211	$10^{211}-1$	The Cabal	SNFS	Distributed
2000	233	$2^{773}+1$	The Cabal	SNFS	Distributed
2003	244	$2^{809}-1$	Franke+Kleinjung+Montgomery	SNFS	LAN PC's?

素因数分解の限界のもうひとつの見方として「全体の桁数に制約されない方法」を用いて「何桁の素因数が見つかったか」に注目する方向もある。これは 2001 年に 55 桁の素因数が山梨大学の宮本泉氏によって ECM(Elliptic Curve Method) で見つけられたのが最高である。

1.3 なぜ GNFS か

1024bits の RSA 暗号で使われる合成数 n は確実に 500bits 以上の二つの素数 p, q の積である。
この素因数 p, q を群論的方法で見つけることができるだろうか。
$p-1$, $p+1$ 法では見つからないように $p\pm1$, $q\pm1$ は大きな素因数を持っていることだろう。
ECM では現状で55桁が限界であり、その進展のスピードは非常に遅い。
256bits(77桁) の素因数を見つけることでさえ当面は達成できないと思われる。
残るは (PP)MPQS と GNFS であるが歴史的にも実際的にも GNFS が有利なことは疑いない。そこで GNFS が対象となるわけである。

1.4 素因数分解法のポイント

『ふるい(sieve)』系の素因数分解法ではどういうことを目標に新しい方法を探すのだろうか。

1.4.1 基本原理

平方差法の原理 合成数 n に対してなんらかの方法で

$$x^2 \equiv y^2 \bmod n, \quad x \not\equiv y \bmod n$$

となる整数 x, y を見つければ

$$\mathrm{GCD}(x-y, n)$$

は n の約数となる。

こういう x, y は

$$x_i \equiv y_i \bmod n \quad \text{かつ} \quad x_i \neq y_i$$

となる x_i, y_i を集めて組み合わせて平方数になるものを探すという手順をとる。
例

$$\begin{aligned} 14\cdot 67 &\equiv 3 \quad \mathrm{mod}187 \\ 31\cdot 67 &\equiv 20 \quad \mathrm{mod}187 \\ 14\cdot 31 &\equiv 60 \quad \mathrm{mod}187 \end{aligned}$$

より

$$(14\cdot 31\cdot 67)^2 \equiv 60^2 \bmod 187$$

となる。
代表元を変えるというのは「加法的」な性質、素因数分解は「乗法的」な性質、それらをブレンドさせるところがポイントと言えるかもしれない。

ポイント $\bmod n$ での代表元を二通りにとってその素因数分解の違いを利用する。

1.4.2 線形代数

問題 一般にたくさんの整数 $r_1, r_2, \ldots, r_s$ が与えられたとき、その中から組み合わせて平方数になるものを見つけるにはどうするか。

- まず適当な個数の素数の集合 $\{p_1, p_2, \ldots, p_B\}$ を決める。factor base という。
- 次にその factor base で完全に素因数分解できてしまうもの、smooth な数という、つまり

$$r_i = \prod_{j=1}^{B} p_j^{e(i,j)}$$

となる r_i を $B+1$ 個以上捜す。これらをあらためて $r_0, r_1, \ldots, r_B$ としよう。

- これらのべき指数部分に注目し、さらにそれを mod 2 で考えて次のような ($\mathbf{Z}/2\mathbf{Z}$ 上の) B 次元ベクトルにする。

$$v_i = (e(i,1) \bmod 2,\ e(i,2) \bmod 2,\ \ldots,\ e(i,B) \bmod 2)$$

- 連立方程式を解き、自明でない関係式を求める。

$$c_0 v_0 + c_1 v_1 + \cdots + c_B v_B \equiv 0 \bmod 2$$

- 元に戻して

$$r_0^{c_0} \cdot r_1^{c_1} \cdots r_B^{c_B} \text{ は平方数}$$

ポイント $r_1, r_2, \ldots, r_s$ の絶対値が小さいほど良い。

1.4.3 定数倍法

アイディア 分解したい数 n に対して $n/2$ の近くの数を 2 倍して n を引けば代表元が "ずれる"。

例 $n = 253$ の場合

$$\begin{array}{lll}
(1) & 2 \cdot 127 = 2 \cdot 127 & \equiv 1 \\
(2) & 2 \cdot 128 = 2^8 & \equiv 3 \\
(3) & 2 \cdot 129 = 2 \cdot 3 \cdot 43 & \equiv 5 \\
(4) & 2 \cdot 130 = 2^2 \cdot 5 \cdot 13 & \equiv 7 \\
(5) & 2 \cdot 131 = 2 \cdot 131 & \equiv 9 \\
(6) & 2 \cdot 132 = 2^3 \cdot 11 & \equiv 11 \\
(7) & 2 \cdot 133 = 2 \cdot 7 \cdot 19 & \equiv 13 \\
(8) & 2 \cdot 134 = 2 \cdot 2 \cdot 67 & \equiv 15 \\
(9) & 2 \cdot 135 = 2 \cdot 3^3 \cdot 5 & \equiv 17 \\
(10) & 2 \cdot 136 = 2^4 \cdot 17 & \equiv 19
\end{array}$$

これより連立方程式を解けば (2),(4),(7),(9),(10) を組み合わせると良いことが分かる。実際

$$\begin{array}{rcl}
2^8\ 2^2\ 5\ 13\ 2\ 7\ 19\ 2\ 3^3\ 5\ 2^4\ 17 & \equiv & 3\ 7\ 17\ 19 \\
2^{16}\ 3^3\ 5^2\ 7\ 17\ 19 & \equiv & 3\ 7\ 17\ 19
\end{array}$$

より

$$2^{16} \cdot 3^2 \cdot 5^2 = (2^8 \cdot 3 \cdot 5)^2 \equiv 1$$

これから次を得る。

$$\mathrm{GCD}(2^8 \cdot 3 \cdot 5 - 1, 253) = 11$$

ポイント smooth になってほしい数は $n/2$ 程度の大きさ。

1.4.4 二次ふるい法

アイディア 分解したい奇数 n に対して $\sqrt{n}$ の近くの数を 2 乗して n を引けば代表元が "ずれる"。

$$x = [\sqrt{n}] \pm m,\ m = 1, 2, \ldots$$

に対して

$$Q(x) = x^2 - n \sim \pm 2m\sqrt{n}$$

を考えるということ。

$n = 3937$ の場合

$$\begin{aligned}
63^2 - n &= 32 &&= 2^5 \\
64^2 - n &= 159 &&= 3 \cdot 53 \\
65^2 - n &= 288 &&= 2^5 \cdot 3^2 \\
66^2 - n &= 419 &&= 419 \\
67^2 - n &= 552 &&= 2^3 \cdot 3 \cdot 23 \\
&\ldots
\end{aligned}$$

の 63, 65 を組み合わせて $(63 \cdot 65)^2 \equiv (2^5 \cdot 3)^2 \bmod n$ を得る。これより $\mathrm{GCD}(63 \cdot 65 - 2^5 \cdot 3, n) = 31$。

ポイント smooth になってほしい数は $H\sqrt{n}$ 程度の大きさ。ただし m を $\pm H$ の範囲で動かすとする。

1.4.5 数体ふるい法

アイディア 代数方程式の複素数体での解を θ、$\bmod n$ での解を M、とすると θ と M の式は $\bmod n$ で異なる代表元を与える。

正確に言うと、多項式 $f(X)$ に対して

$$\begin{cases}
f(\theta) &= 0 &&, \theta \in \mathbf{C} \\
f(M) &\equiv 0 \bmod n &&, M \in \mathbf{Z}
\end{cases}$$

のとき、次の ϕ は環準同型である

$$\begin{aligned}
\phi : \ \mathbf{Z}[\theta] &\longrightarrow \mathbf{Z}/n\mathbf{Z} \\
g(\theta) &\mapsto g(M) \bmod n
\end{aligned}$$

これによって次の図式を2通りに進んで、その分解の違いを利用する。この図式は $\mathbf{Z}/n\mathbf{Z}$ で考えれば可換だが、$\mathbf{Z}$ では可換とは限らない。

$$\begin{array}{ccc}
a + b\theta & \longrightarrow & \mathbf{Z}[\theta] \text{ での分解} \\
\phi \downarrow & & \downarrow \phi \\
a + bM & \longrightarrow & \mathbf{Z} \text{ での分解}
\end{array}$$

$a + b\theta$ と $a + bM$ が同時に smooth な a, b のペアがほしい。$a + b\theta$ の smoothness は $a + b\theta$ のノルム（後述）の絶対値の smoothness と同じである。

ポイント smooth になってほしい数は $f(x)$ の次数を d、係数の最大値を c、変化させる a, b の最大値を H とすると

- $a + bM$ は $n^{1/d}H$ 程度、
- $a + b\theta$ のノルムの絶対値は $c \cdot H^d$ 程度

の大きさである。

d の増加に対して逆向きに反応する。最適なものをとると非常に良いものになる。

1.5 NFS

1.5.1 SNFS 概論

Special Number Field Sieve は、ターゲットの合成数 N が係数の小さな多項式 $f(x)$ と整数 M により $f(M)$ が N の倍数となっている場合を扱う。

$f(x)$ の根の1つを θ とし、$K = \mathbf{Q}(\theta)$ とする。

仮定として次を置く

- $\mathbf{Z}[\theta]$ は UFD(素元分解環) である。

$N = f(M)$ の大きさに応じて $\mathbf{Z}$ の素数の集合 F と O_K の素元と単数の集合 G を適当にとる。

a, b を互いに素な有理整数で $a + bM$ は F-smooth とする。つまり

$$a + bM = \prod_{p \in F} p^{e(p)} \tag{1-1}$$

と完全に分解するものとする。かつ $a + b\theta$ は G-smooth とする。つまり

$$a + b\theta = \prod_{\pi \in G} \pi^{e(\pi)} \tag{1-2}$$

と完全に素元分解されるものとする。

このようなペア (a, b) を $\#F + \#G$ 個より多く集める。すると各べき指数を mod 2 で考えたベクトル

$$\left((e(p) \bmod 2)_{p \in F},\ (e(\pi) \bmod 2)_{\pi \in G} \right)$$

は線形従属となり、適当に組み合せると、つぎのような関係式が出る。

$$\prod (a_i + b_i M) = \left(\prod_{p \in F} p^{e(p)} \right)^2 \tag{1-3}$$

と $\mathbf{Z}$ で平方数になり、かつ

$$\prod (a_i + b_i \theta) = \left(\prod_{\pi \in G} \pi^{e(\pi)} \right)^2 \tag{1-4}$$

と O_K で平方数になる。

$$\prod (a_i + b_i \theta) = s(\theta)^2, \quad s(x) \in \mathbf{Z}[x] \tag{1-5}$$

であったとしてこれを準同型

$$\begin{array}{rcl} \phi : \mathbf{Z}[\theta] & \longrightarrow & \mathbf{Z}/N\mathbf{Z} \\ g(\theta) & \mapsto & g(M) \bmod N \end{array}$$

で写せば

$$\left(\prod_{p\in F} p^{e(p)}\right)^2 \equiv s(M)^2 \bmod N \tag{1-6}$$

となり、ふるい法の目標である $x^2 \equiv y^2 \bmod N$ 型の合同式が得られる。

まとめると、SNFS の原理は、

$$\begin{array}{rcl} \text{環準同型} \quad \phi : \mathbf{Z}[\theta] & \longrightarrow & \mathbf{Z}/N\mathbf{Z} \\ g(\theta) & \mapsto & g(M) \bmod N \end{array}$$

による次の図式を2通りに進んで、その分解の違いを利用することと言える。この図式は $\mathbf{Z}/N\mathbf{Z}$ で考えれば可換だが、$\mathbf{Z}$ では可換とは限らない。

$$\begin{array}{ccc} a+b\theta & \longrightarrow & O_K\text{での分解} \\ \phi\downarrow & & \downarrow\phi \\ a+bM & \longrightarrow & \mathbf{Z}\text{での分解} \end{array}$$

そこで、するべきことは $a+b\theta$ と $a+bM$ が同時に smooth な a,b のペアをたくさん探すこととなる。

”ふるい”系の素因数分解法ではこの smooth な数 をたくさん見つけるところが実行時間の主要な部分となる。高速にするためにはこの検査の対象となる数の絶対値を小さくするのが非常に有効である。Number Field Sieve では、$a+b\theta$ と $a+bM$ という二つの対象があって、前者は $f(x)$ の次数の増加に比例して絶対値が増加し、後者はその逆になる。そこで両者の中間を探ると最適な次数があって（幸いにも）計算量が優れたものとなったのである。

1.5.2 GNFS 概論

簡単のため $O_K = \mathbf{Z}[\theta]$ を仮定する。これが成立しない場合でも若干の補正で以下の議論はそのまま有効になる。補正の方法は後に述べる。

一般に整数環 O_K では素元分解ができない。しかし素イデアル分解は可能である。

そこで O_K については有理素数の集合 F と素イデアルの集合 G を適当にとる。さらに結果を平方数にするために何個かの平方剰余記号の集合 H を付け加える。

a,b を互いに素な有理整数で $a+bM$ は F-smooth とする。つまり

$$a+bM = \prod_{p\in F} p^{e(p)} \tag{2-1}$$

であり、かつ単項イデアル $< a + b\theta >$ は G-smooth とする[2]。つまり

$$< a + b\theta >= \prod_{\mathcal{P} \in G} \mathcal{P}^{e(\mathcal{P})} \tag{2-2}$$

と完全に素イデアル分解されるものとする。このとき $\chi \in H$ に関する値を次のようにおく。

$$\chi(a + b\theta) = (-1)^{e(\chi)}$$

ただし $a + b\theta$ は χ の modulus とは互いに素であるとする。このためには modulus を G に含まれる素イデアルで割れないようにとる必要がある[3]。

このようなペア (a, b) を $\#F + \#G + \#H$ 個より多く集める。すると各べき指数を $\bmod 2$ で考えたベクトル

$$(\ (e(p) \bmod 2)_{p \in F},\ (e(\mathcal{P}) \bmod 2)_{\mathcal{P} \in G},\ (e(\chi))_{\chi \in H}\ )$$

は一次従属となり、適当に組み合せると、つぎのような関係式が出る。

$$\prod (a_i + b_i M) = \left(\prod_{p \in F} p^{e(p)} \right)^2 \tag{2-3}$$

と $\mathbf{Z}$ で平方数になり、かつ

$$\prod < a_i + b_i\theta >= \left(\prod_{\mathcal{P} \in G} \mathcal{P}^{e(\mathcal{P})} \right)^2 \tag{2-4}$$

と O_K で平方イデアルになり、かつ

$$\chi\left(\prod (a_i + b_i\theta) \right) = 1 \quad \text{for } \forall \chi \in H \tag{2-5}$$

となる。

$(2-4), (2-5)$ 式は $\prod(a_i + b_i\theta)$ が高い確率で O_K の平方数になっていることを示している。もし実際に

$$\prod (a_i + b_i\theta) = s(\theta)^2, \quad s(x) \in \mathbf{Z}[x] \tag{2-6}$$

であったとすれば後は Special NFS の場合とまったく同様である。

以上での最大の問題点は (2-6) において平方根 $s(\theta)$ の計算をいかにして記憶容量も時間も少なく済ませることができるかである。これは後に述べる。

O_K がたとえ UFD であってもそのことを用いないで、この節の方法を適用するのが良い。イデアルの生成元と単数を求める必要がなく、プログラム作りは非常に簡明になる。

1.5.3 多項式の選択

ターゲットの合成数 N と多項式 $f(x)$ および M の間には次の関係があった。

$$f(M) \equiv 0 \mod N$$

$f(x)$ の次数は N が数十桁から二百桁程度ならば 3 から 5 にとるのが良い。

[2] ここでは区別をはっきりさせるため単項イデアルは生成元を $< >$ で囲んで表すことにする。
[3] 後述の "large prime" を用いる場合はそれらよりも大きくする。

係数を小さくする

$f(x)$ にまず課せられる条件は $a+b\theta$ のノルムが小さいことである。

$$f(x) = c_d x^d + c_{d-1}x^{d-1} + \cdots + c_1 x + c_0$$

とすると

$$c_d \mathcal{N}(a+b\theta) = (-b)^d f(a/(-b)) = c_d a^d + c_{d-1}a^{d-1}(-b) + \cdots + c_1 a(-b)^{d-1} + c_0(-b)^d$$

であるのでこれを小さくするには、係数の絶対値を小さくすることが有効であろう。絶対値の積分を小さくすることでも良い[4]。

Knuth-Schroeppel 関数値を大きくする

ふるいの効率は二次ふるい法と同様に Knuth-Schroeppel 関数によって推定することができる。それは

$$\sum_{\mathcal{P}\in G} \frac{\log P}{P-1}, \quad \text{ただし } P \text{ はイデアル } \mathcal{P} \text{ に含まれる素数とする}$$

である。これが大きいほど smooth data を多く集められるということになる。

この値を多項式の係数の簡単な性質から判定することはおそらく不可能であろう。少なくとも今のところは多項式の分解（＝素数の素イデアル分解）から実際に計算するしかない。

適当な実根を持たせる

$a+b\theta$ のノルムを小さくするには $f(a/(-b))$ の絶対値を小さくすることが有効。a, b が動く範囲である $[-H_a, H_a]\times[1, H_b]$ で考えると、

$f(x)$ に実根があってその中に $\pm H_a/H_b$ に近いものがある

ことが望ましいことになる。

より詳細で実践的な考察が

- [Brian Murphy] "Polynomial Selection for the Number Field Sieve Integer Factorisation Algorithm", Doctorial Thesis, 1999

においてなされている。このセミナーでも先に進んでからこの問題について扱う予定である。

1.6 RSA-155

August 22, 1999 に CWI のチームによって行われた RSA-155 の分解については次の文書が詳しい。

`http://www.crypto-world.com/announcements/RSA155.txt`

[4] 絶対値の積分は難しいので 2 乗の積分を用いるのが良い。

1.7 another C155

Simon Singh: "The Code Book"(サイモン・シン、青木薫訳「暗号解読」）の巻末付録には10個のいろいろな種類の暗号が問題として出されている[5]。

最後の Stage 10 は 155 桁の RSA 暗号であった。これは Sweden のチームによって解かれた。同じ 155 桁でもこちらの方が若干小さい。

彼らは CWI のプログラムを用いた。線形代数の部分は Alpha の 4-processor machine 用に改造し 13 日で解を求めた。詳しくは以下の文書を参照のこと。

`http://answers.codebook.org`

1.8 C158

2003年3月現在の世界記録は2002年1月に行われた F. Bahr, J. Franke, T. Kleinjung による158桁である。これについては以下の文書を参照されたい。これ以上の詳しい報告は発表されていないようである。

`http://www.crypto-world.com/announcements/c158.txt`
`http://www.ercim.org/publication/Ercim_News/enw49/franke.html`

1.9 GNFS の実行の段階分け

1. 多項式探索
2. ふるい
3. "filtering"
4. 線形代数
5. 代数的数の平方根

1.10 GNFS の実行に必要なリソース

1.10.1 ハードウェア

多項式探索

通常の PC で十分。

ふるい

通常の PC を多数用意すれば良い。ただしメモリアクセスの速いものが望ましい。

"filtering"

まだ解明できていないが特別なものが必要ないことは確かである。

[5] これは日本語訳が出たときにはすべて解かれていた。

線形代数

64-bit multi CPU 機、あるいは LAN で結合された複数台 (16 台以上) の PC が必要

平方根

通常の PC で十分。

1.10.2 ソフトウェア

多項式探索

現状は全数探索に近い。良い多項式を見つけるのは多分に偶然である。そのためいかに多数の多項式をチェックするか、あるいはいかに不要なものをスキップするかが重要となる。

ふるい

line sieve については細かな高速化の余地を探る。
lattice sieve の実装は経験がない。この実装は重要。
遠隔サーバマシンへのデータの転送、ふるい範囲の割り当てをするネットワークプログラム – 普通の TCP/IP プログラムであり、エラー処理等を付け加えるだけではあるが、使い勝手の良いものを作るのはそれなりに手間がかかる。

"filtering"

まだ本質がつかめていない。実装もまだ。

線形代数

block Lanczos 法の実装–一応できてはいる。
マルチCPU でのマルチ・スレッドのプログラム– 一応できてはいる。
ネットワーク接続PC での分散処理– MPI などの分散処理支援ソフトの導入

平方根

Montgomery の方法の実装–まだ。

1.10.3 理論

1024bits RSA 暗号の実行時間の評価という点では「多項式探索」と「ふるい」が重要である。

多項式探索

理論よりも高速な実装が重要である。
“ふるい”を高速化するために多項式に課す条件は現在のものがすべてかは検討の余地がある。

ふるい

lattice sieve, line sieve のふるいの範囲の最適化。両者の使い分け。

”filtering”

線形代数

block Lanczos 法の収束を速くする方法はないか。今はほとんど全空間を埋め尽くすまで収束しないがある部分空間内に閉じこめて収束させる方法はないか。

大規模行列処理の一般論としても有用な並列分散処理法の開発。

平方根

高速化

1.11 このプロジェクトの予定

1.11.1 第1期 2002年3月まで

基礎理論の確認のために今回も含め4回のセミナーを開催する。

- 第1回　2002/1/19　数体ふるい法概論
- 第2回　2002/2/23　代数的整数論入門
- 第3回　2002/3/9　ふるい、行列処理
- 第4回　2002/3/30　代数的数の平方根

1.11.2 第2期 2002年4月から2003年3月まで

第1期では全体をざっと眺めただけで特に高速化を意図してはいなかったが第2期では各論毎に分担者を決めて世界最高速のプログラム作成を目指す。

1.11.3 第3期 2003年4月から2004年3月まで

第3期[6] ではすべてのプログラムをつないで全体としてのチューニングを行い 320bits から 560bits くらいまでの合成数について実際の実行時間を計測し 1024bits の実行時間を推測する。

[6] 第3期については 2003年3月31日現在では、実行されるかどうか未定である。

Chapter 2

数体ふるい法（その１）

代数的整数論入門

2002/2/23　CRYPTREC 研究会 at 立教大学 S302 教室
2003/3/31 改訂
木田 祐司

この小論では代数的整数論の中から数体ふるい法の理解に必要と思われることを簡単に解説する。しかし、実際にプログラムを作る場面では代数的整数論の理論的な面に正面からぶつかると歯が立たないことが多い。あるいは細部を完全に解明することは莫大なコストをかけることになりもする。できるだけ困難を避けて通ったり、細部が詰めきれないままでも計算を行い結果オーライと割り切ることも必要である。

2.1　代数的整数

数体ふるい法では

θ を整数係数多項式 $f(x)$ の一つの根とするとき、小さな整数 a, b に対して

$$a + b\theta$$

を“素因数分解”する必要がある。

この方法を述べることがこのテキストの主要な内容である。
例として次の３つを考える。

1. $f_a(x) = x^2 - 2$
2. $f_b(x) = x^3 + 2$
   $M = 2^{43}, \quad N = f_b(M) = \mathrm{C39} = 680564733841876926926749214863536422914$
3. $f_c(x) = 99535x^5 + 219113x^4 + 1139835x^3 - 1304055x^2 + 38350500x + 65696851$
   $M = 905015890, \quad N = f_c(M) = \mathrm{C50} = 60430591373189847889033809879449779100427831121351$

参考書：
石田信「代数的整数論、数学全書５」森北出版、1974 年
藤崎源二郎「代数的整数論入門（上）」裳華房、1975,2001 年

2.1.1 最小多項式と判別式

定義 2.1.1 θ を根とする有理数係数多項式で次数が最小のものを θ の **最小多項式** あるいは **定義多項式** という。モニック（＝最高次係数が 1）のものとして一つに決めるのが普通だがここではモニックを仮定しない。
最小多項式の次数をその数の**次数** という。
最小多項式が同一な数は互いに **共役** であるという。ある数と共役な元をその数の共役元という。

例 2.1.2 $\sqrt{2}$ の最小多項式は x^2-2、$\sqrt{2}$ の次数は 2 で、共役元は $\sqrt{2}$ 自身と $-\sqrt{2}$ の 2つ。

例 2.1.3 $\sqrt{2}+\sqrt{3}$ の最小多項式は x^4-10x^2+1、$\sqrt{2}+\sqrt{3}$ の次数は 4 で、共役元は $\pm\sqrt{2}\pm\sqrt{3}$ の 4 つである。

例 2.1.4 θ の最小多項式が $f(x)=c_dx^d+c_{d-1}x^{d-1}+\cdots+c_1x+c_0$ ならば $a+b\theta$ $(a,b\in\mathbf{Q},\, b\neq 0)$ の最小多項式は

$$g(x)=f(\frac{x-a}{b})$$

である。

定義 2.1.5 θ の共役元を $\theta_1,\theta_2,\ldots,\theta_d$ とするとき $a+b\theta(a,b\in\mathbf{Q},\, b\neq 0)$ のトレース（共役和）とノルム（共役積）を次のように定義する。

$$\begin{aligned}\mathcal{T}(a+b\theta) &= (a+b\theta_1)+(a+b\theta_2)+\cdots+(a+b\theta_d)\\ \mathcal{N}(a+b\theta) &= (a+b\theta_1)(a+b\theta_2)\cdots(a+b\theta_d)\end{aligned}$$

例 2.1.6 θ の最小多項式を $f(x)=c_dx^d+c_{d-1}x^{d-1}+\cdots+c_1x+c_0$ とすると

$$\mathcal{T}(\theta)=-\frac{c_{d-1}}{c_d},\qquad \mathcal{N}(\theta)=(-1)^d\frac{c_0}{c_d}$$

である。さらに $a+b\theta$ $(a,\, b\in\mathbf{Q},\, b\neq 0)$ ならば

$$\mathcal{T}(a+b\theta)=da-b\,\frac{c_{d-1}}{c_d}$$

$$\mathcal{N}(a+b\theta)=(-1)^d\frac{f(-a/b)}{c_d/b^d}=(-b)^d\frac{f(-a/b)}{c_d}$$

よって

$$c_d\mathcal{N}(a+b\theta)=c_da^d+c_{d-1}a^{d-1}(-b)+\cdots+c_ra^r(-b)^{d-r}+\cdots+c_1a(-b)^{d-1}+c_0(-b)^d$$

ノルムは $a+b\theta$ の“素因数分解”の様子を $\mathbf{Z}$ の数で教えてくれる。

例 2.1.7

$$\mathcal{N}(1+3\sqrt{-1})=1^2+3^2=10=2\cdot 5$$

なので $1+3\sqrt{-1}$ の“素因数分解”は 2 と 5 に関係していることがわかる。実際

$$2=\mathcal{N}(1+\sqrt{-1}),\quad 5=\mathcal{N}(1+2\sqrt{-1})$$

で

$$1+3\sqrt{-1}=-(1-\sqrt{-1})(1-2\sqrt{-1})$$

定義 2.1.8 多項式 $f(x)$ の **判別式** $D(f)$ を根の差積の 2 乗で定義する。つまり根を $\theta_1, \theta_2, \ldots, \theta_d$ とするとき

$$D(f) = \left(\prod_{1 \le i < j \le d} (\theta_i - \theta_j) \right)^2$$

多項式でなく θ_i について定まるものと考えても良い。このときは $D(\theta_i)$ と記す。

判別式は直接的には多項式が重根を持つかどうかを判別するものであるが、数論的には主にイデアルが“分岐”するかどうかを判定する。我々は整数環との違いを知るために用いる。

例 2.1.9 $f(x) = ax^2 + bx + c$ の場合。

$$\begin{aligned} D(f) &= (\theta_1 - \theta_2)^2 \\ &= (\theta_1 + \theta_2)^2 - 4\theta_1\theta_2 \\ &= (-b/a)^2 - 4c/a \\ &= \frac{b^2 - 4ac}{a^2} \end{aligned}$$

である。これはおなじみの二次方程式の判別式（の $1/a^2$倍）である。

高次の多項式の場合は次の関係式を用いる。積の微分の公式によると

$$f'(x)/c_d = \sum_{j=1}^{d} \prod_{1 \le i \le d,\, i \ne j} (x - \theta_i)$$

であるから

$$f'(\theta_j)/c_d = \prod_{1 \le i \le d,\, i \ne j} (\theta_j - \theta_i)$$

となるので

命題 2.1.10

$$D(f) = (-1)^{d(d-1)/2} \frac{1}{c_d^d} \prod_{i=1}^{d} f'(\theta_i)$$

例 2.1.11 $f(x) = x^d + ax + b$ の場合。

$$\begin{aligned} \textstyle\prod_{i=1}^{d} f'(\theta_i) &= \textstyle\prod_{i=1}^{d} (d\theta_i^{d-1} + a) \\ &= \textstyle\prod_{i=1}^{d} (d\theta_i^{d} + a\theta_i)/\theta_i \\ &= \textstyle\prod_{i=1}^{d} (d(-a\theta_i - b) + a\theta_i)/\theta_i \\ &= \textstyle\prod_{i=1}^{d} ((1-d)a\theta_i - db)/\prod_{i=1}^{d} \theta_i \\ &= \mathcal{N}((1-d)a\theta_i - db)/\mathcal{N}\theta_i \qquad (i \text{ は何でも同じ}) \end{aligned}$$

である。これより

$$D(f) = (-1)^{d(d-1)/2} \{(1-d)^{d-1} a^d + d^d b^{d-1}\}$$

となる。

例 2.1. 12 $f(x) = x^5 + 2x^4 + 3x^3 + 4x^2 + 5x + 6$ の判別式を求める。厳密な計算よりも近似計算で逃げるのが簡明。*Newton*法などで根の近似値を求めると

$$\begin{aligned} &-1.49179798814 \\ &0.55168546346 \pm 1.25334886028\sqrt{-1} \\ &-0.80578646939 \pm 1.22290471337\sqrt{-1} \end{aligned}$$

となる。これらを $f'(x)$ に代入して積を作れば、その値は

$$1037231.999994567 - 6.32383 \cdot 10^{-13}\sqrt{-1}$$

である。一般に判別式は有理数であるから。この値に一番近い有理数ということで

$$D(f) = 1037232 = 2^4 3^3 7^4$$

を得る[1]。実数部が整数に近くなかったり、虚数部が 0 にあまり近くない場合は近似の精度を上げねばならない。根が代数的整数でない場合は判別式が有理整数とは限らないので分母を消す操作が必要になる。

例 2.1. 13

$$f(x) = 99535x^5 + 219113x^4 + 1139835x^3 - 1304055x^2 + 38350500x + 65696851$$

の場合は多倍長の近似計算により

$$\begin{aligned} &99535^8 D(f) \\ &= 468321481474688513582321892909069344775849393813408958 33 \\ &= 13 * 13 * 17 * 3797 * 10779623 * 39825779365841732543798347696692230293 1291 \end{aligned}$$

を得る。

また van der Monde の行列式を思い出すと

命題 2.1. 14

$$D(f) = \begin{vmatrix} 1 & 1 & \cdots & 1 \\ \theta_1 & \theta_2 & \cdots & \theta_d \\ \theta_1^2 & \theta_2^2 & \cdots & \theta_d^2 \\ \cdots & \cdots & \cdots & \cdots \\ \theta_1^d & \theta_2^d & \cdots & \theta_d^d \end{vmatrix}^2$$

2.1.2 整数環と判別式

定義 2.1. 15 整数係数多項式の根を **代数的数** という。とくに最高次係数が 1 であるものの根を **代数的整数** という。

[1] 厳密には可能性が高いとしか言えない。なお次の節の内容を先取りすれば、「$f(x)$ の根は代数的整数であるから、それらの積和である判別式も代数的整数である。しかも判別式は有理数であるから、有理整数になる。」と言えるのでより確信は深まる。

定義 2.1.16 代数的数を有理数体に添加して得られる体を **代数体** という。ある代数体に含まれる代数的整数の全体は環をなす。これをその代数体の **整数環** という。

定義 2.1.17 代数体 K の有理数体 $\mathbf{Q}$ 上のベクトル空間としての次元を K の次数という。

定理 2.1.18 代数体 K の次数 d が有限の場合、d 次の既約多項式 $f(x)$ が存在してその一つの根 θ により $K=\mathbf{Q}(\theta)$ と表される。逆に d 次の既約多項式の一つの根を添加した代数体の次数は d である。

有理数体 $\mathbf{Q}$ に、既約多項式 $f(x)=c_dx^d+c_{d-1}x^{d-1}+\cdots+c_1x+c_0$ の根 θ を添加した体 $\mathbf{Q}(\theta)$ とは $\mathbf{Q}$ の元と θ のべき乗が作る有理式の全体のことである。これに

1. 分母の有理化
2. $\theta^d=-c_d^{-1}(c_{d-1}\theta^{d-1}+\cdots+c_1\theta+c_0)$ による次数の還元

を行えば

$$\mathbf{Q}(\theta)=\{a_0+a_1\theta+a_2\theta^2+\cdots+a_{d-1}\theta^{d-1}\,|\,a_i\in\mathbf{Q}\}$$

となる。

定義 2.1.19 $K=\mathbf{Q}(\theta)$ を d 次の代数体とするとき、その元 $\alpha=a_0+a_1\theta+a_2\theta^2+\cdots+a_{d-1}\theta^{d-1}$ の K における共役元とは

$$a_0+a_1\theta_i+a_2\theta_i^2+\cdots+a_{d-1}\theta_i^{d-1},\quad i=1,2,\ldots,d$$

のこととする。ただし θ_i は θ の共役元である。
さらに K におけるノルムを

$$\mathcal{N}_K\alpha=\prod_{i=1}^{d}\left(a_0+a_1\theta_i+a_2\theta_i^2+\cdots+a_{d-1}\theta_i^{d-1}\right)$$

で定義する。
ここで α の次数を $d(\alpha)$ とすれば

$$\mathcal{N}_K\alpha=(\mathcal{N}\alpha)^{d/d(\alpha)}$$

である。

例 2.1.20 $K=\mathbf{Q}(\sqrt{2}+\sqrt{3})$ は 4 次の代数体であって

$$\begin{aligned}\mathcal{N}_K(\sqrt{2}+\sqrt{3}) &= (\sqrt{2}+\sqrt{3})(\sqrt{2}-\sqrt{3})(-\sqrt{2}+\sqrt{3})(-\sqrt{2}-\sqrt{3})=1\\ \mathcal{N}_K\sqrt{2} &= (\mathcal{N}\sqrt{2})^2=((\sqrt{2})(-\sqrt{2}))^2=4\\ \mathcal{N}_K2 &= 2^4=16\end{aligned}$$

定理 2.1. 21 整数環は $\mathbf{Z}$ 上の基底を持つ。詳しくは次の通り。
代数体 K の次数を d とすると K の d 個の代数的整数 $\theta_1, \theta_2, \ldots, \theta_d$ が存在して K の任意の代数的整数 α は

$$\alpha = c_1\theta_1 + c_2\theta_2 + \cdots + c_d\theta_d, \quad c_i \in \mathbf{Z}$$

と表される。しかもこの表し方は一意である。
注意：θ_i はここでは共役元の意味ではない。

例 2.1. 22 $\sqrt{2}$ は 2次の代数的数であり、$\mathbf{Q}(\sqrt{2})$ は $\mathbf{Q}$ 上 2次の代数体である。$\mathbf{Q}(\sqrt{2})$ の整数環の基底は $\{1, \sqrt{2}\}$ である。

例 2.1. 23 m が平方数でなければ $\sqrt{m}$ は 2次の代数的数であり、$\mathbf{Q}(\sqrt{m})$ は $\mathbf{Q}$ 上 2次の代数体である。$\mathbf{Q}(\sqrt{m})$ の整数環の基底は $m = k^2m'$, (m'は平方因子を持たない) と分けるとき $m' \equiv 2, 3 \bmod 4$ ならば $\{1, \sqrt{m'}\}$ であり、$m' \equiv 1 \bmod 4$ ならば $\{1, (1+\sqrt{m'})/2\}$ である。

例 2.1. 24 m が立方数でなければ $\sqrt[3]{m}$ は 3次の代数的数であり、$\mathbf{Q}(\sqrt[3]{m})$ は $\mathbf{Q}$ 上 3次の代数体である。$m = ab^2$ (a は平方因子を持たない) とすると整数環の基底は
$a^2 \not\equiv b^2 \bmod 3^2$ ならば

$$1, \ \sqrt[3]{ab^2}, \ \sqrt[3]{a^2b}$$

であり、
$a^2 \equiv b^2 \bmod 3^2$ ならば

$$\frac{1}{3}(1 + a\sqrt[3]{ab^2} + b\sqrt[3]{a^2b}), \ \sqrt[3]{ab^2}, \ \sqrt[3]{a^2b}$$

である。

注意：上記のように整数基底が理論的に綺麗に求まることは非常に稀である。普通は計算機プログラムによって数値計算で求めることになる。

定義 2.1. 25 $f(x)$ を d次既約多項式でその根 θ を添加した代数体 $K = \mathbf{Q}(\theta)$ の整数基底を $\theta_1, \theta_2, \ldots, \theta_d$ とするとき体 K の **判別式** $D(K)$ を

$$D(K) = \left(\det\left(\theta_i^{(j)}\right)\right)^2$$

で定義する。これは有理整数である。

例 2.1. 26 $K = \mathbf{Q}(\sqrt{2})$ の整数基底は $1, \sqrt{2}$ であるから

$$D(K) = \begin{vmatrix} 1 & 1 \\ \sqrt{2} & -\sqrt{2} \end{vmatrix}^2 = (-2\sqrt{2})^2 = 8$$

例 2.1. 27 $K = \mathbf{Q}(\sqrt[3]{2})$ の整数基底は $1, \sqrt[3]{2}, \sqrt[3]{4}$ であるから

$$D(K) = \begin{vmatrix} 1 & 1 & 1 \\ \sqrt[3]{2} & \sqrt[3]{2}\omega & \sqrt[3]{2}\omega^2 \\ \sqrt[3]{4} & \sqrt[3]{2}\omega^2 & \sqrt[3]{4}\omega \end{vmatrix}^2 = (6\omega(\omega-1))^2 = -2^2 \cdot 3^3$$

定理 2.1. 28 $\theta \in \mathcal{O}_K$ に対して

$$D(\theta)/D(K) = (O_K : \mathbf{Z}[\theta])^2$$

例 2.1. 29

$$f(x) = 99535x^5 + 219113x^4 + 1139835x^3 - 1304055x^2 + 38350500x + 65696851$$

の場合は

$$\begin{aligned}
&99535^8 D(f)\\
&= 468321481474688513582321892909069344775849393813408958 33\\
&= 13 * 13 * 17 * 3797 * 10779623 * 39825779365841732543798347696692230293 1291
\end{aligned}$$

だったので、この一つの根 θ によって $K = \mathbf{Q}(\theta)$ を作ると 99535θ は代数的整数であって

$$(O_K : \mathbf{Z}[99535\theta])^2 = D(99535\theta)/D(K) = 99535^{20} D(\theta)/D(K)$$

ここで右辺の平方因子は $99535^{12}13^2 = (5^6 \cdot 13 \cdot 17^6 \cdot 1171^6)^2$ の約数であるから $(O_K : \mathbf{Z}[99535\theta])$ は $5^6 \cdot 13 \cdot 17^6 \cdot 1171^6$ の約数である。

次に、$\eta = 1/\theta$ とするともちろん生成する体は等しい

$$K = \mathbf{Q}(\theta) = \mathbf{Q}(\eta)$$

η の最小多項式は $f(x)$ と係数の並びが逆になる。

$$g(x) = 99535 + 219113x + 1139835x^2 - 1304055x^3 + 38350500x^4 + 65696851x^5$$

すると 65696851η は代数的整数であって

$$65696851^8 D(g) = 99535^8 D(f)$$

なので

$$(O_K : \mathbf{Z}[65696851\eta])^2 = D(65696851\eta)/D(K) = 65696851^{20} D(\eta)/D(K)$$

ここで右辺の平方因子は $65696851^{12}13^2 = (11^6 \cdot 13 \cdot 19^6 \cdot 314339^6)^2$ の約数であるから $(O_K : \mathbf{Z}[65696851\eta])$ は $11^6 \cdot 13 \cdot 19^6 \cdot 314339^6$ の約数である。

両方を合わせると 13 以外の素数についてはどちらかの指数がその素数の倍数にならない。

2.2 イデアル論の基本定理

たとえば $\mathbf{Z}[\sqrt{2}]$ においては $2 = (\sqrt{2})^2$ だから 2 は $\mathbf{Z}[\sqrt{2}]$ ではもはや「素数」ではない。このように素数が代数体でどのように分解するかを調べることが代数的整数論の主要な目的の一つである。

$\mathbf{Q}(\sqrt{-5})$ の整数環では「素元」分解が一意ではない[2]。

$$6 = 2 \cdot 3 = (1 + \sqrt{-5})(1 - \sqrt{-5})$$

であって、2, 3, $1 + \sqrt{-5}$, $1 - \sqrt{-5}$ はこれ以上分解できないので「素元」である。

「素元」分解ではこれで行き詰まる。そこで素イデアル分解にするとすべてがうまく行く。

[2]正確な用語としては素元ではなく既約元という。素元は素イデアルを生成する元として定義される。ここではしかし既約元ではつまらないので「」付きで「素元」と記しておく。

定義 2.2.1 整数環 $\mathcal{O}$ のイデアル A とは
(1) A は加法に関して $\mathcal{O}$ の部分群である。
(2) $c \in \mathcal{O}$, $a \in A$ ならば $ca \in A$ である。
が満たされること。

定義 2.2.2 整数環 $\mathcal{O}$ のイデアル A, B に対して
(1) $A + B = \{a + b \,|\, a \in A, b \in B\}$
(2) $A \cdot B = \{\sum a_i b_i \,|\, a_i \in A, b_i \in B\}$
でイデアルの和と積を定義する。これらもまたイデアルになる。

$\mathbf{Q}(\sqrt{-5})$ での 6 の分解に戻る。まず

$$I = 2\mathcal{O}, \quad J = 3\mathcal{O}, \quad K = (1+\sqrt{-5})\mathcal{O}, \quad L = (1-\sqrt{-5})\mathcal{O}$$

とおいて

$$A = I + K, \quad B = I + L, \quad C = J + K, \quad D = J + L$$

とする。すると

$$\begin{aligned} A \cdot B &= 2\mathcal{O}, & C \cdot D &= 3\mathcal{O} \\ A \cdot C &= (1+\sqrt{-5})\mathcal{O}, & B \cdot D &= (1-\sqrt{-5})\mathcal{O} \end{aligned}$$

となる。つまり前の二つの分解はまだ途中であって

$$6\mathcal{O} = A \cdot B \cdot C \cdot D$$

が完全な分解であるということになる。ただし A, B は同じイデアルである。

定義 2.2.3 整数環 $\mathcal{O}_K$ のイデアル X のノルムを

$$\mathbf{N}X = (\mathcal{O}_K : X)$$

で定義する。ただし (:) は群の指数である。

例 2.2.4

$$\mathbf{N}(2\mathbf{Z}) = (\mathbf{Z} : 2\mathbf{Z}) = \#(\mathbf{Z}/2\mathbf{Z}) = 2$$

例 2.2.5 上の $\mathbf{Q}(\sqrt{-5})$ では

$$\begin{aligned} \mathbf{N}I &= (\mathcal{O} : 2\mathcal{O}) = 4 \\ \mathbf{N}J &= (\mathcal{O} : 3\mathcal{O}) = 9 \\ \mathbf{N}K &= (\mathcal{O} : (1+\sqrt{-5})\mathcal{O}) = 6 \\ \mathbf{N}L &= (\mathcal{O} : (1+\sqrt{-5})\mathcal{O}) = 6 \\ \mathbf{N}A &= (\mathcal{O} : 2\mathcal{O} + (1+\sqrt{-5})\mathcal{O}) = 2 \\ \mathbf{N}B &= (\mathcal{O} : 2\mathcal{O} + (1-\sqrt{-5})\mathcal{O}) = 2 \\ \mathbf{N}C &= (\mathcal{O} : 3\mathcal{O} + (1+\sqrt{-5})\mathcal{O}) = 3 \\ \mathbf{N}D &= (\mathcal{O} : 3\mathcal{O} + (1-\sqrt{-5})\mathcal{O}) = 3 \end{aligned}$$

定理 2.2.6 ノルムは乗法的である。つまりイデアル X, Y に対して

$$\mathbf{N}(XY) = \mathbf{N}X \cdot \mathbf{N}Y$$

K の数 α に対してそれが生成する単項イデアルのノルムは数としてのノルムの絶対値に等しい。

$$\mathbf{N}(\alpha\mathcal{O}_K) = |\mathcal{N}_K\alpha|$$

一般に次の定理が成立する。

定理 2.2.7 (イデアル論の基本定理) 整数環のイデアルは有限個の素イデアルの積に必ず分解され、しかもその分解の仕方は一通りである。

必要な例は次の小節で与える。
ここで定理は整数環に対するものであって部分環やもっと大きな環では乱れが出ることになる。

例 2.2.8 $\mathbf{Q}(\sqrt{5})$ の整数基底は $1, (1+\sqrt{5})/2$ であるが $1, \sqrt{5}$ を基底とする部分環 R では 2 の分解がうまく行かない。2 が生成する R のイデアル $2R$ を考えると

$$1+\sqrt{5} \notin 2R \quad \text{かつ} \quad (1+\sqrt{5})^2 = 2(3+\sqrt{5}) \in 2R$$

なので $2R$ は R の素イデアルではない。もし R でイデアル論の基本定理が成り立つならば $2R$ は R 全体ではないイデアルの積になる。これは計算するとありえないことがわかる。

例 2.2.9 $1, \sqrt{5}/2$ を基底とする環 R は $(\sqrt{5}/2)^2 = 5/4$ なので $1/4$ を含み、整数環より真に大きい。このとき 2 は R の可逆元であるから 2 に関係する素イデアルは存在しない。

2.3 素数の素イデアル分解

整数環 $\mathcal{O}$ は $\mathbf{Z}$ を含んでいる。詳しくは

$$\mathcal{O} \cap \mathbf{Q} = \mathbf{Z}$$

である。また $\mathcal{O}$ のイデアルは $\mathbf{Z}$ のイデアルを含んでいる。とくに素イデアルは素数の倍数の全体を含んでいる。つまり

P が $\mathcal{O}$ の素イデアルならば $P \cap \mathbf{Z} = p\mathbf{Z}$ となる素数 p が存在する。

逆に、素数 p を含む素イデアルはどのようなものだろうか。これは $p\mathcal{O}$ を含む素イデアルということだがイデアルが“含む”ということはイデアルとして“約数”になるということと同じであることが分かるので

$$p\mathcal{O} = P_1^{e_1} P_2^{e_2} \cdots P_g^{e_g}$$

を素イデアル分解とすると、ここに出てくる $P_1, P_2, \ldots, P_g$ が素数 p を含む素イデアルのすべてである。
実際の分解を計算するには次の定理を用いる。

定理 2.3. 1 既約多項式 $f(x)$ の根 θ により $K = \mathbf{Q}(\theta)$ を作り、その整数環を $\mathcal{O}_K$ とする。また p を素数で群指数 $(\mathcal{O} : \mathbf{Z}[\theta])$ を割らないものとする。$f(x)$ の $\bmod p$ での既約分解を

$$f(x) \equiv p_1(x)^{e_1} p_2(x)^{e_2} \cdots p_g(x)^{e_g} \bmod p$$

とするとき $P_i = (p, p_i(\theta))$ とすればこれは素イデアルであって

$$p\mathcal{O}_K = P_1^{e_1} P_2^{e_2} \cdots P_g^{e_g}$$

となる。

例 2.3. 2 $f(x) = x^2 - 2$ ならば $\mathcal{O}_K = \mathbf{Z}[\sqrt{2}]$ なのですべての素数について計算できる。

$$f(x) \equiv \begin{cases} x^2 & \bmod 2 & \Rightarrow & (2) = (2, \sqrt{2})^2 \\ \text{既約} & \bmod 3 & & \\ \text{既約} & \bmod 5 & & \\ (x+3)(x-3) & \bmod 7 & \Rightarrow & (3) = (7, 3+\sqrt{2})(7, 3-\sqrt{2}) \\ \text{既約} & \bmod 11 & & \\ \text{既約} & \bmod 13 & & \\ (x+6)(x-6) & \bmod 17 & \Rightarrow & (17) = (17, 6+\sqrt{2})(17, 6-\sqrt{2}) \end{cases}$$

どのような規則になっているのだろうか。実は次のような法則がある。

体の判別式は 8 であり、これが分解を統制する。

(1) $p \,|\, 8$（つまり $p = 2$）ならば素イデアルの 2 乗になる。

(2) $p \equiv 1, 7 \bmod 8$ ならば異なる素イデアルの積。

(3) $p \equiv 3, 5 \bmod 8$ ならば素イデアルのまま。

ここで

$$(2, \sqrt{2}) = (\sqrt{2}), \quad (7, 3 \pm \sqrt{2}) = (3 \pm \sqrt{2}), \quad (17, 6 \pm \sqrt{2}) = (1 \pm 3\sqrt{2})$$

なのでイデアルを出さなくても数で十分ではある。

例 2.3. 3 $f(x) = x^2 + 5$ ならば $\mathcal{O} = \mathbf{Z}[\sqrt{-5}]$ なのですべての素数について計算できる。

$$f(x) \equiv \begin{cases} (x+1)^2 & \bmod 2 & \Rightarrow & (2) = (2, 1+\sqrt{-5})^2 \\ (x+1)(x-1) & \bmod 3 & \Rightarrow & (3) = (3, 1+\sqrt{-5})(3, 1-\sqrt{-5}) \\ x^2 & \bmod 5 & \Rightarrow & (5) = (5, \sqrt{-5})^2 \\ (x+3)(x-3) & \bmod 7 & \Rightarrow & (7) = (7, 3+\sqrt{-5})(7, 3-\sqrt{-5}) \\ \text{既約} & \bmod 11 & & \\ \text{既約} & \bmod 13 & & \\ \text{既約} & \bmod 17 & & \end{cases}$$

どのような規則になっているのだろうか。実は次のような法則がある。

体の判別式は 20 であり、これが分解を統制する。

(1) $p \,|\, 20$（つまり $p = 2, 5$）ならば素イデアルの 2 乗になる。

(2) $p \equiv 1, 3, 7, 9 \bmod 20$ ならば異なる素イデアルの積。

(3) $p \equiv 11, 13, 17, 19 \bmod 20$ ならば素イデアルのまま。

ここで $(5, \sqrt{-5}) = (\sqrt{-5})$ であるが

$$(2, 1+\sqrt{-5}), \quad (3, 1 \pm \sqrt{-5}), \quad (7, 3 \pm \sqrt{-5})$$

は単項イデアルではないので数ではうまくいかない。イデアルが必須である。

ついでに書くと $(41) = (6+\sqrt{-5})(6-\sqrt{-5})$ となって単項イデアルの積になる。この体では $p \equiv 1 \bmod 20$ ならば異なる単項な素イデアルの積になるのである。

例 2.3.4 $f(x) = x^3 - 2$ ならば $\mathcal{O} = \mathbf{Z}[\sqrt[3]{2}]$ なのですべての素数について計算できる。

$(x)^3$	$\bmod 2$	$\Rightarrow$	$(2) = (2, \sqrt[3]{2})^3$
$(x+1)^3$	$\bmod 3$	$\Rightarrow$	$(3) = (3, 1+\sqrt[3]{2})^3$
$(x+2)(x^2-2x-1)$	$\bmod 5$	$\Rightarrow$	$(5) = (5, 2+\sqrt[3]{2})(5, -1-2\sqrt[3]{2}+\sqrt[3]{2}^2)$
既約	$\bmod 7$		
$(x+4)(x^2-4x+5)$	$\bmod 11$	$\Rightarrow$	$(11) = (11, 4+\sqrt[3]{2})(11, 5-4\sqrt[3]{2}+\sqrt[3]{2}^2)$
既約	$\bmod 13$		
$\cdots$			
$(x+11)(x-7)(x-4)$	$\bmod 31$	$\Rightarrow$	$(31) = (31, 11+\sqrt[3]{2})(31, 7-\sqrt[3]{2})(31, 4-\sqrt[3]{2})$

この体の判別式は $4 \cdot 27 = 108$ であるが前の二つの例と違って mod 108 で分解が決まるわけではない。この違いは方程式のガロア群が可換群かそうでないかによる。前の二つは $\mathbf{Z}/2\mathbf{Z}$ であるがこの場合は対称群 S_3 であって非可換である。

これらのイデアルはすべて単項である。

$$(2, \sqrt[3]{2}) = (\sqrt[3]{2}), \quad (3, 1+\sqrt[3]{2}) = (1+\sqrt[3]{2}), \quad (5, 2+\sqrt[3]{2}) = (1+\sqrt[3]{2}^2),$$
$$(5, -1-2\sqrt[3]{2}+\sqrt[3]{2}^2) = (-1-2\sqrt[3]{2}+\sqrt[3]{2}^2), \quad (11, 4+\sqrt[3]{2}) = (-1+\sqrt[3]{2}+\sqrt[3]{2}^2),$$
$$(11, 5-4\sqrt[3]{2}+\sqrt[3]{2}^2) = (1-3\sqrt[3]{2}-2\sqrt[3]{2}^2), \quad (31, 11+\sqrt[3]{2}) = (1-3\sqrt[3]{2}+\sqrt[3]{2}^2),$$
$$(31, 7-\sqrt[3]{2}) = (1-2\sqrt[3]{2}-\sqrt[3]{2}^2), \quad (31, 4-\sqrt[3]{2}) = (1-2\sqrt[3]{2}^2)$$

例 2.3.5 $f(x) = x^2 - 5$ ならば $\mathcal{O} = \mathbf{Z}[(1+\sqrt{5})/2]$ なので $\mathbf{Z}[\sqrt{5}]$ は指数 2 の部分環である。よって 2 についてはこの多項式では判定できない。実際、

$$f(x) \equiv (x+1)^2 \bmod 2$$

であるが 2 は $\mathcal{O}$ では素イデアルである。

2.3.1 多項式 $\bmod p$ の一次因子を求める

$\mathbf{F}_p = \mathbf{Z}/p\mathbf{Z}$ 係数で多項式を因数分解する。ここで 512bits 程度の GNFS を考えると p は $24 \sim 26$ bits 以下である。また多項式は 6 次以下である。

この処理は最初に1度行うだけなので、改良に力を入れる必要はない。

次の公式が基本である。

$$x^p - x = \prod_{s=0}^{p-1} (x-s) \quad \text{in } \mathbf{F}_p[x]$$

これにより

$$f_1(x) = \mathrm{GCD}(f(x),\, x^p - x)$$

は $f(x)$ のすべての一次因子一個ずつの積になることがわかる。次に $f_1(x)$ を分解する。

p が小さければ[3] $s = 0, 1, 2, \ldots, p-1$ と動かして $f_1(s) \equiv 0 \bmod p$ となるものを探せば良い。

p が大きいときは、次の公式を用いる。

$$x^p - x = (x+a)((x+a)^{(p-1)/2} + 1)((x+a)^{(p-1)/2} - 1) \quad \text{for } a = 0, 1, 2, \ldots, p-1$$

これにより

$$f_{11}(x) = \mathrm{GCD}(f_1(x),\, (x+a)^{(p-1)/2} + 1), \qquad f_{12}(x) = f_1(x)/f_{11}(x)$$

とすれば

$$f_1(x) = f_{11}(x) \cdot f_{12}(x)$$

と分解される。この分解されたそれぞれに次の a についての計算を行うことを繰り返して完全な分解を得る。一つの a で $f_1(x)$ の一次因子は二つのグループに分かれるので最大 6 個の一次因子を分離するには平均 3 回の計算で済む。

一般に $\mathrm{GCD}(f(x),\, g(x)^n)$ の計算は次の2段階で行う。

1. $g(x)^n \bmod f(x)$ を繰り返し2乗法で計算する。
2. その結果と $f(x)$ の GCD を計算する。

例 2.3. 6

$$f(x) = 99535x^5 + 219113x^4 + 1139835x^3 - 1304055x^2 + 38350500x + 65696851$$

を $\bmod p = 104743$ で因数分解する。まず

$$f_1(x) = \mathrm{GCD}(f(x),\, x^p - x) = x^3 + 103518x^2 + 7541x + 39422$$

であるので一次因子は 3 個ある。そして

$$\begin{aligned} f_{11}(x) &= \mathrm{GCD}(f_1(x),\, x^{(p-1)/2} + 1) &&= x^2 + 19435x + 54722 \\ f_{12}(x) &= f_1(x)/f_{11}(x) &&= 99535(x + 84083) \end{aligned}$$

次に

$$\begin{aligned} f_{111}(x) &= \mathrm{GCD}(f_{11}(x),\, (x+1)^{(p-1)/2} + 1) &&= x + 35005 \\ f_{112}(x) &= f_{11}(x)/f_{111}(x) &&= x + 89173 \end{aligned}$$

したがって

$$\begin{aligned} & 99535x^5 + 219113x^4 + 1139835x^3 - 1304055x^2 + 38350500x + 65696851 \\ = \;& 99535(x + 35005)(x + 84083)(x + 89173) \mod 104743 \end{aligned}$$

である。素イデアルとしては

$$(104743,\, \theta - 15570),\ (104743,\, \theta - 20660),\ (104743,\, \theta - 69738)$$

となる。104743 は前に求めた $(\mathcal{O}_K : \mathbf{Z}[99535\theta])$ を割らないのでこれが $\mathcal{O}_K$ における 104743 の正しい一次因子である。

[3] 実装によるが 100 とか 200 といった程度。

2.3.2 素イデアル分解の例

例 2.3. 7

$$f(x) = 99535x^5 + 219113x^4 + 1139835x^3 - 1304055x^2 + 38350500x + 65696851$$

において p が 99535 の約数でも 13 でもなければ $f(x) \bmod p$ の因数分解から p の素イデアル分解が得られる。

o が 99535 の約数の場合は $f(x)$ の根 θ から $\eta = 1/\theta$ を作るとそれは

$$g(x) = 99535 + 219113x + 1139835x^2 - 1304055x^3 + 38350500x^4 + 65696851x^5$$

の根であって、$g(x) \bmod p$ の一次因子を $x - t$ とすれば $(p, \eta - t)$ が1次の素イデアルになる。

ここで $t \not\equiv 0 \bmod p$ ならば $(p, \eta - t) = (p, \theta - t^{-1})$ なので、$f(x) \bmod p$ の一次因子として求めることができるものである。

$t \equiv 0 \bmod p$ ならば (p, η) は θ の式に変形できない。このとき $a + b\theta$ を (p, η) が割るのは $a + b\theta = \theta(a\eta + b)$ なので $b \equiv 0 \bmod p$ となる。このような素イデアルはまったく別扱いしなければいけない。また p が $f(x)$ の最高次係数も定数項も割る場合は非常に複雑になる。

2.3.3 変形ノルム

ここまで素イデアル分解の複雑な状況を説明してきたが、実は数体ふるい法のふるいの段階では $a + b\theta$ の素イデアル分解を避けて通ることができる。それはふるいの段階では $a + b\theta$ が factor base の素イデアルで完全に分解するかどうかが分かれば十分だからである。

そのために次の値を考える。ノルムに最高次係数を掛けたものである。これを **変形ノルム** と呼ぶことにする[4]。

$$c_d \mathcal{N}(a + b\theta) = c_d a^d + c_{d-1} a^{d-1}(-b) + \cdots + c_r a^r (-b)^{d-r} + \cdots + c_1 a(-b)^{d-1} + c_0(-b)^d$$

変形ノルムが素数 p で割れたとしよう。つまり、

$$c_d a^d + c_{d-1} a^{d-1}(-b) + \cdots + c_r a^r (-b)^{d-r} + \cdots + c_1 a(-b)^{d-1} + c_0(-b)^d \equiv 0 \bmod p$$

であるとする。

$p \not| b$ ならば $\bmod p$ での b の逆数を b' として $(-b')^d$ を両辺に掛けると

$$c_d(-ab')^d + c_{d-1}(-ab')^{d-1} + \cdots + c_r(-ab')^r + \cdots + c_1(-ab') + c_0 \equiv 0 \bmod p$$

となる。これは $-ab'$ が $f(x) \equiv 0 \bmod p$ の解であることを示している。つまり $x + ab'$ が $f(x) \bmod p$ の一次因子であることを示している。したがって p の素イデアル分解を意識しないで単純に多項式の $\bmod p$ での因数分解を考えれば良い。

$p|b$ ならば $c_d a^d \equiv 0 \bmod p$ となるが a と b は互いに素であるから a は p で割れず、したがって c_d が p の倍数にならねばならない。逆に言うと c_d の約数となる素数 p については $p|b$ ならば $c_d \mathcal{N}(a + b\theta)$ は必ず p で割り切れる。この p の状況を (p, ∞) と表し、射影的な素イデアル あるいは 射影的な因子 と呼ぶことにする。

変形ノルムに注目するとイデアル論を用いなくても良いので簡明であるが、数体ふるい法の最後の段階までイデアル論を避けることは今のところできない。

[4] 一般に $\mathcal{N}(a + b\theta)$ は整数とは限らないが $c_d \mathcal{N}(a + b\theta)$ は整数となる。しかしこれが何かのノルムになっているわけではない。なんとも不思議なものである。理論的には代数的整数である $c_d(a + b\theta)$ のノルムを扱うべきであるが変形ノルムの c_d^{d-1} 倍になるので計算的に不利である。

Chapter 3

数体ふるい法（その２）

ふるい(sieve)

2002/3/9　CRYPTREC 研究会 at 立教大学 S302 教室
2003/3/31 改訂
木田 祐司

3.1　ここでいう sieve とは

ここでの目的は[1]

ある範囲の整数の組 $(a,b),\ -H_a \le a < H_a,\ 0 < b \le H_b,\ \mathrm{GCD}(a,b) = 1$ の中からその素因数（素イデアル）分解

$$\begin{aligned} a + bM &= \prod_i p_i^{e_i} \\ (a + b\theta) &= \prod_i \mathcal{P}_i^{f_i} \end{aligned}$$

があらかじめ定めた素数の集合 F と素イデアルの集合 G の元だけで行われるようなものを探し出すこと。ここで $< >$ は単項イデアルを表す。

である。F を有理的な factor base、G を代数的な factor base という。すでに見たように G は 1 次の素イデアルのみからなる。

そのために、一定の範囲の素数を見つけ出す方法である“エラトステネスのふるい”をまねるのである。

たとえば $a + bM$ が p の倍数ならば $(a + ip) + (b + jp)M$ もまた p の倍数であるから、割り算を行わず単に step p で移動することにより p の倍数を次々に見つけることができる。これは $a + b\theta$ についても同様である。

1 から 100 までで $2, 3, 5$ で完全に分解されるものを探すには次のようにする。

- 100 個の場所 $w_1 \sim w_{100}$ を準備して最初は全部 1 にする。
- 2 から始めて step 2 で 2 を掛ける。
- 2 のべきについても同様にする。
- $3, 5$ についても同様にする。

以上を行った後、$w_i = i$ となっていれば i は $2, 3, 5$ で完全に分解されることになる。

[1] 前章最後の節で実際の計算では変形ノルムを扱うと書いたが、この章では理論的に簡明にするために通常のノルムで記述する。従って一般には正確な記述ではないことに注意されたい。

例 3.1.1 10001 から 10100 の中で $2, 3, 5, 7, 11, 13, 17$ で完全に素因数分解できるものを探す。log をとって足し算で行う。

```
#include <stdio.h>
#include <math.h>

#define Start 10001
#define Ns 100
#define Ps 7
#define Scale 20.0

int prime[Ps] = {2,3,5,7,11,13,17};
unsigned char work[Ns];

main()
{
    int i, ip, p, pp;
    unsigned char logp, logn;

    for(i=0; i<Ns; i++) work[i] = 0;                    ワークエリアを 0 でクリア

    for(ip=0; ip<Ps; ip++){
        pp = p = prime[ip];
        logp = (int)(log(p) * Scale + 0.5);
        while(pp < Start+Ns){
            i = Start % pp; if (i != 0) i = pp - i;     最初の index を計算。
            for(i=i; i<Ns; i+=pp) work[i] += logp;      step p で log を加える。
            pp *= p;                                    次のべき乗へ。
        }
    }

    for(i=0; i<Ns; i++){
        printf("%4d", work[i]);
    }
    printf("\n");

    logn = (int)(log(Start) * Scale * 0.95);            0.95 は誤差の考慮。
    printf("log >= %4d\n", logn);
    for(i=0; i<Ns; i++){
        if (work[i] >= logn) printf("%5d\n", Start+i); 値が十分大きいところが解。
    }
}

----- result -----

   0  36  39  28  54  14   0  86   0 184  22  28  57  36  32  70 105  14   0  82
  48  14  73  81  64  58   0  28  22 103  39 126   0  14  76  79   0  75   0  74
  22  14  96 116 110  14  79  84  51 100   0  67  44  62  32  64   0  14  61  60
   0 109   0 113 102  53   0  50   0  46  66  42  39  36 115  76  22  14   0 185
  57  14  22  28  32  36  87  93  44  46   0  50   0  92  54  56   0 185   0  92

log >=  174
10010
10080
10098
```

3.2 line sieve

単純に b を 固定して a を動かす方法を line-by-line sieve あるいは単に line sieve という。

(1) sieve のワークエリアを $2H_a$ bytes 確保し、0 クリアする。

(2) 各 $p \in F$ について p の倍数に対応するところに $\log p$ を加えていく。
これは最初の位置さえ計算すれば次の番地は p を加えるだけで済む。ワークエリアの先頭は $-H_a+bM$ に対応するので最初に p の倍数になる番地 r は

$$r \equiv H_a - bM \bmod p, \quad 0 \le r < p$$

で計算できる。次の b でのスタートは $-M \bmod p$ だけずれることに注意。

(3) すべての $p \in F$ について (3) を行なったのち $\log bM$ に近い値はそのままにするが、そうでない（= かなり小さい）値はこのテストを通らなかったので 0 クリアする。

イデアル $(a+b\theta)$ の分解は次の原理によって有理整数の分解に帰着させる。

$$a + b\theta \equiv 0 \bmod (p, \theta - s) \Longleftrightarrow a + bs \equiv 0 \bmod p$$

(4) 新たなワークエリアを $2H_a$ bytes 確保し、0 クリアする。

(5) 各 $(p, \theta - s) \in G$ についてその倍数に対応するところに $\log p$ を加えていく。これは最初の位置さえ計算すれば次の番地は p を加えるだけで良い。ワークエリアの先頭は $a = -H_a$ に対応するので最初に $(p, \theta - s)$ の倍数となる番地 r は

$$r \equiv H_a - bs \bmod p, \quad 0 \le r < p$$

で計算できる。次の b でのスタートは $-s \bmod p$ だけずれることに注意。

(6) すべての $(p, \theta - s) \in G$ について (5) を行なったのち $\log|\mathcal{N}(a+b\theta)|$ に近いものを残し、それ以外は 0 クリアする。

(7) 両方の sieve を通ったものを実際に割り算をして確認する。イデアルの割り算の方は次の原理によりノルムの割り算で済ませる。

$$(p, \theta - s)^e || (a + b\theta) \Longleftrightarrow p^e || \mathcal{N}(a + b\theta)$$

注意 $a + b\theta$, $\mathrm{GCD}(a, b) = 1$ は同一の素数 p を含む異なる素イデアル $(p, \theta - s)$, $(p, \theta - t)$ で同時に割れることはない。

注意 メモリは次のように使われる。

[TABLE] 素数、素イデアルのテーブル:16 bytes程度を最大500万セット

[WORK] sieve されるエリア:数10メガbytes(分割可)

[TABLE] から素数や位置情報を取り出しながら [WORK] を sieve する。[WORK] が頻繁にアクセスされるのでキャッシュメモリがここに効いてほしい。

例 3.2. 1 GNFS158 での sieve 領域は $-5 \cdot 10^8 \le a < 5 \cdot 10^8$, $0 < b < 2 \cdot 10^4$ であった。

3.2.1 $(1+1)$ large primes

a, b を sieve を通ったペアとする。

(8) $a+bM$ を F の素数で素因数分解する。分解しきれない部分を R とする。
高速化の手法がいろいろ考えられている。簡単なものでは
割り切れるかどうかの判定を多倍長数である $a+bM$ 自身でなく work area 中の offset で判定する
割り切れたときは work area の値を sieve のときとは逆に $\log p$ だけ減らすことにより試し割り算の終了を判定する、などである。

(9) $\mathcal{N}(a+b\theta)$ を G の素イデアルに含まれる素数で素因数分解する。分解しきれない部分を A とする。
上と同じような高速化が可能である。

ただし、R, A が factor base の最大の素数の２乗より大きいものは捨てる。そうすることにより、以下では R, A は素数として良い。
これで次のような分解が得られた。

$$\begin{cases} a+bM & = & \prod\limits_{p\in F} p^{e(p)}\, R \\ \mathcal{N}(a+b\theta) & = & \prod\limits_{q\in NG} q^{e(q)}\, A \end{cases}$$

(FF 型) $R=1$, $A=1$ の場合。
本来はこれのみを求めていた。target が大きくなるとこれだけでは十分な個数が集まらないので以下のような data も利用しなければならなくなる。

(PF 型) $R>1$, $A=1$ の場合。
このようなもので R が一致するものを見つけて積を作れば分解されない部分は平方数になる。平方数はこの先の処理で無視しても良いものなので、この積は (FF 型) と同様に扱うことができる。

$$\begin{cases} a_1+b_1M & = & \prod p^{e(p)_1}\, R \\ \mathcal{N}(a_1+b_1\theta) & = & \prod q^{e(q)_1} \end{cases}$$

$$\begin{cases} a_2+b_2M & = & \prod p^{e(p)_2}\, R \\ \mathcal{N}(a_2+b_2\theta) & = & \prod q^{e(q)_2} \end{cases}$$

から

$$\begin{cases} (a_1+b_1M)(a_2+b_2M) & = & \prod p^{e(p)_1+e(p)_2}\, R^2 \\ \mathcal{N}(a_1+b_1\theta)(a_2+b_2\theta) & = & \prod q^{e(q)_1+e(q)_2} \end{cases}$$

である。

R が一致するものは R について並べ替えをして見つける。

(FP 型) $R=1$, $A>1$ の場合。

A が一致してもノルムをとる前のイデアルが一致するとは限らない。今、$a+b\theta$ は A を含むある素イデアル $(A, \theta-s)$ で割れるが s は $s=ab^{-1} \bmod A$ で決定される。これにより 2 つの $a+b\theta$ が同じ素イデアルで割れるかどうかが判定できる。

$$\begin{cases} a_1+b_1M & = & \prod p^{e(p)_1} \\ \mathcal{N}(a_1+b_1\theta) & = & \prod q^{e(q)_1}\, A \end{cases}$$

$$\begin{cases} a_2 + b_2 M &= \prod p^{e(p)_2} \\ \mathcal{N}(a_2 + b_2\theta) &= \prod q^{e(q)_2}\, A \end{cases}$$

において $a_1 b_1^{-1} \equiv a_2 b_2^{-1} \bmod A$ ならば

$$(a_1 + b_1\theta)(a_2 + b_2\theta) = \text{factor base の素イデアルの積} \times (A, \theta - a_1 b_1^{-1})^2$$

この場合は A と $ab^{-1} \bmod A$ という二つをキーにして並べ替えを行うという面倒がある。

(PP 型) $R > 1$, $A > 1$ の場合。

ほとんどないが R, A, $ab^{-1} \bmod A$ が一致すれば (FF 型) ができる。(PF 型) を使って R を消去すれば (FP 型) ができる。(FP 型) を使って A を消去すれば (PF 型) ができる。これらを含めて (FP 型),(PF 型) の処理を再度行えば新たな (FF 型) を作ることができる。

3.2.2 $(2+2)$ large primes

"$(1+1)$ large primes" では sieve を通ったものを素因数分解あるいは素イデアル分解して分解し切れなかった部分をまた利用した。このとき残りは一つの素数あるいは素イデアルであった。これを 2 個ずつにするのが "$(2+2)$ large primes" である。つまり

$$\begin{cases} a + bM &= \displaystyle\prod_{p \in F} p^{e(p)}\, R_1 R_2 \\ \mathcal{N}(a + b\theta) &= \displaystyle\prod_{q \in NG} q^{e(q)}\, A_1 A_2 \end{cases}$$

という分解も利用するのである。

このときは sieve に合格する基準をずっと下げることになる。すると sieve において

- 素数べきの扱いを省略しても良い。
- 小さな素数は省略しても良い。

などの利点がある。反面

- 試し割り算をする候補が大幅に増える。
- 試し割り算の後で残った部分を二つに素因数分解しなければいけない。

という欠点もある。それぞれを軽減する必要がある。

一つ目は "$(1+1)$ large primes" のときとは違って巨大で不必要な素数が大量に混じることである。有理数側でいえば F の最大素数を p_{max} とすると $p_{max} < R_1 \le R_2 < p_{max}^2$ であるから

$$p_{max}^2 < R_1 R_2 < p_{max}^4$$

となる。もちろん R_1 のみのものも扱うから sieve での許容範囲は p_{max}^4 以下のすべてとなる。これは過大なので範囲を制限することが考えられる。たとえば

$$p_{max} < R_1 \le R_2 \le 50 p_{max}$$

とすれば、この範囲と

$$p_{max}^2 < R_1 R_2 \le 50^2 p_{max}^2$$

の範囲を許容することになる。

二つ目は R_1, R_2 の積を R_1, R_2 に分解するという“ミニ素因数分解”を高速に実行することである。これは Rho 法 や ECM あるいはマイナーなところであるが Shanks の SQUFOF 法が有効である。

3.2.3 Free Relations

$a+b\theta$ の中でとくに $b=0$ の数、つまり有理整数は扱っていなかった。これは別に扱うのが適当である。有理素数で K で一次の素イデアルの積に完全分解するものはそのまま smooth data として扱える。

$$p\mathcal{O}_K = (p,\, \theta - s_1)(p,\, \theta - s_2)\cdots(p,\, \theta - s_d)$$

このとき両辺の素数、素イデアルが共に factor base に入っていれば $a=p$, $b=0$ は (FF型) のペアとなる。このような関係式（あるいはそのペア）を free relation と言う。こういう素数 p がどのくらいあるかは密度定理で調べることができる。

$f(x)=x^3+2$ つまり $K=Q(-\sqrt[3]{2})$ の場合は $\#F/6$ 個あるので、かなりの助けになる。この場合の最小の p は 31 で、次のように分解される。

$$31\mathcal{O}_K = (\theta-11)(\theta-24)(\theta-27)$$

一般には $f(x)$ のすべての根を添加した体、最小分解体という、の次数で決まる。

定理 3.2. 2 (密度定理（の特殊な場合）)

$$\lim_{x\to\infty} \frac{\textit{free relation}\text{を与える素数で } x \text{ 以下のものの個数}}{x \text{ 以下の素数の個数}} = \frac{1}{\text{最小分解体の次数}}$$

しかし、d 次既約多項式をランダムにとるとほとんどの場合、最小分解体の次数は $d!$ である[2]。他の効率を悪くせずにこの次数を減らすことはかなり難しいと思われる。

3.3 lattice sieve

代数的な factor base に含まれない一次の素イデアル $Q=(q,\, \theta-t)$ を固定して

$$W = \{(a,b) \mid -H_a \le a \le H_a,\ 0 < b < H_b\}$$

の中で

$$W_q = \{(a,b) \in W \mid a+b\theta \equiv 0 \bmod Q\}$$

はどのような集合となっているかを調べる。

$$a+bt \equiv 0 \bmod q$$

であるから

$$(q,0),\ (-t,1)$$

で生成される $\mathbf{Z}$ 上の rank 2 の自由加群である。

2 次元なので通常のユークリッド除算（ガウス法）で最小ベクトルを求めることができる。

定義 3.3. 1 2 次元ベクトル $\mathbf{a}=(x_1,y_1)$ の $\mathbf{b}=(x_2,y_2)$ による整数商 $r(\mathbf{a},\mathbf{b})$ を次のように定義する。

$$r(\mathbf{a},\mathbf{b}) = \mathrm{round}\left(\frac{\mathbf{a}\cdot\mathbf{b}}{|\mathbf{b}|^2}\right) = \mathrm{round}\left(\frac{x_1x_2+y_1y_2}{x_2^2+y_2^2}\right)$$

[2]ガロア群が S_d であるということと同じ。

定理 3.3. 2 lattice L の基底 $\mathbf{a}$, $\mathbf{b}$, $|\mathbf{a}| \geq |\mathbf{b}|$ について

$$\mathbf{c} = \mathbf{a} - r(\mathbf{a}, \mathbf{b})\mathbf{b}$$

が $|\mathbf{c}| \geq |\mathbf{b}|$ となるならば $\mathbf{b}$ は L の最小ベクトルである。

例 3.3. 3 $q = 3$, $t = 1$ ならば基底として $(3, 0)$, $(-1, 1)$ をとれば

$$(3, 0) = -2(-1.1) + (1, 2)$$

となり、これ以上短くできないので終了である。

$$(-1, 1),\ (1, 2)$$

が最小基底である。

例 3.3. 4 $q = 104729$, $t = 34567$ ならば次のように計算される。

```
(104729, 0) = (-3)(-34567,1) + (1028,3)
(-34567, 1) = (-34)(1028, 3) + (385, 103)
(1028, 3)  = 2(385, 103) + (258,-203)
(385, 103) = 1(258, -203) + (127,306)
(258, -203) は (127, 306) では短縮できない。
```

$$(258, -203),\ (127, 306)$$

が最小基底である。

最小基底を $\mathbf{v}_1 = (x_1, y_1), \mathbf{v}_2 = (x_2, y_2)$ とする。$\mathbf{v}_1$ の方が絶対値が小さいものとする。また必要ならば符号を変えることによって $x_1 > 0$ とする。lattice L をこの基底で表す。

$$a + b\theta = c\mathbf{v}_1 + d\mathbf{v}_2$$

とすると

$$\begin{cases} a &= cx_1 + dx_2 \\ b &= cy_1 + dy_2 \end{cases}$$

である。

注意 $\det \begin{pmatrix} x_1 & y_1 \\ x_2 & y_2 \end{pmatrix} = \pm q$ であるので $\mathrm{GCD}(c, d) = 1$ であっても $\mathrm{GCD}(a, b) = q$ となりうる。

line sieve では a, b を動かして sieve したが lattice sieve では各 $Q = (q, \theta - t)$ について c, d を動かして sieve する。これも "$(2 + 2)$ large primes" で行う。

ひとつの Q については多くの点の sieve を行わず、Q 自身をたくさん動かすことにする。

lattice sieve の利点は代数的数の方は必ず Q で割れるのであり、割った商はランダムであると考えられるから sieve の対象の大きさが $1/q$ になった効果があることである。

例 3.3.5 RSA155 では

$$16 \cdot 10^6 < q < 308 \cdot 10^6$$

としている。素イデアルQ の個数は $15.7 \cdot 10^6$ である。$8 \cdot 10^3 \times 5 \cdot 10^3$ 個を sieve している。最小ベクトルの成分を $\sqrt{q} \sim 13 \cdot 10^3$ 程度とすれば領域としては $\pm 52 \cdot 10^6 \times 65 \cdot 10^6$ 程度ということになる。

$a = b = 1 \cdot 10^6$ における代数側の sieve 対象の大きさは 2^{206} 程度である。有理数側は 2^{115} 程度とずっと小さい。代数側が $1/q$ になることによりバランスが良くなる。

GNFS158 では

$$24 \cdot 10^6 < q < 118 \cdot 10^6$$

とし $16 \cdot 10^3 \times 8 \cdot 10^3$ 個を sieve している。最小ベクトルの成分を $\sqrt{q} \sim 8 \cdot 10^3$ 程度とすれば領域としては $\pm 64 \cdot 10^6 \times 64 \cdot 10^6$ 程度ということになる。

Chapter 4

Lanczos 法

4.1 Lanczos 法（標数 0 の場合）

4.1.1 記号

- $(v, w) = v^T w$ は通常の内積
- $(v, w)_A = v^T A w$ は対称行列 A による内積
- $L(v_1, v_2, \ldots)$ は $v_1, v_2, \ldots$ で張られる部分空間

4.1.2 Lanczos 法の考え方（１）

我々の目的は与えられた行列 A およびベクトル c に対して

$$Ax = c$$

となる x を求めることである。

簡単のため A は正則行列であり、したがって任意の c に対して上の方程式は解を持つとする。

$$v_1, v_2, \ldots, v_d$$

を全体空間 $V = \mathbf{R}^d$ の基底とする。x を求めるとは

$$x = x_1 v_1 + x_2 v_2 + \cdots + x_d v_d$$

となる係数 x_i を求めることである。x が与えられたときに x_i を決定するのは基底が直交していれば両辺の v_i との内積をとれば簡単である。

今は $Ax = c$ からそれを行いたい。v_i との内積をとってみよう。

$$(v_i, Ax) = (v_i, c)$$

右辺は計算できるのでこれを利用する。左辺は

$$(v_i, Ax) = x_1(v_i, Av_1) + x_2(v_i, Av_2) + \cdots + x_d(v_i, Av_d)$$

となる。ここで、行列 A が正定値対称行列ならば $(v, Aw) = v^T A w$ を Aによる v と w の内積と考えて通常の内積と同様に扱うことができる。以下この内積を A-内積と呼ぶことにし $(v, w)_A$ と記す。

$v_1, v_2, \ldots, v_d$ を A-内積に関する Gram-Schmidt の直交化法により A-直交化する。それもまた $v_1, \ldots, v_d$ で表すことにしよう。すると上の式では i 番目の項以外は消えてしまう。

$$(v_i, Ax) = (v_i, x)_A = x_i (v_i, v_i)_A$$

つまり、

$$x_i = (v_i, x)_A / (v_i, v_i)_A = (v_i, c)/(v_i, v_i)_A$$

であり、結局求める解は

$$x = \sum_{i=1}^{d} \frac{(v_i, c)}{(v_i, v_i)_A} v_i$$

となる。

まとめると、行列 A が正定値対称行列ならば方程式(1) は次の手順で解ける。

- $\mathbf{R}^d$ の基底から A-直交基底 $\{v_1, v_2, \ldots, v_d\}$ を作る。
- $x = \sum_{i=1}^{d} \frac{(v_i, c)}{(v_i, v_i)_A} v_i$ を計算する。

例 4.1. 1

$$A = \begin{pmatrix} 3 & 2 & -1 \\ 2 & 2 & 0 \\ -1 & 0 & 3 \end{pmatrix}, \quad c = \begin{pmatrix} 0 \\ 3 \\ 4 \end{pmatrix}$$

とすると標準基底から A-直交基底を作ると

$$v_1 = (1,\, 0,\, 0)^T, \quad v_2 = (-2/3,\, 1,\, 0)^T \quad v_3 = (1,\, -1,\, 1)^T$$

となる。すると

$$\begin{aligned} (v_1, c)/(v_1, v_1)_A &= 0/3 \\ (v_2, c)/(v_2, v_2)_A &= 3/(2/3) \\ (v_3, c)/(v_3, v_3)_A &= 1/2 \end{aligned}$$

なので

$$\begin{aligned} x &= (9/2)v_2 + (1/2)v_3 \\ &= (-5/2,\, 4,\, 1/2)^T \end{aligned}$$

が解である。

4.1.3 Lanczos 法の考え方（２）

前節の方法をよく調べてみると基底の A-直交化に時間がかかることがわかる。これは実質的に A の三角行列化と同等であって利点はない。

ただし、空間 W が次の条件を満たせば前節の議論をそのまま適用することができる。

$$\begin{cases} c & \in & W \\ AW & = & W \end{cases}$$

これは

$$\begin{cases} w_1 & = & c \\ w_{i+1} & = & Aw_i \end{cases}$$

で $W = L(w_1, w_2, \ldots)$ を作って行けば最大 d 回で増加しなくなり目的を達する。

この基底から A-直交基底を作るのは次のように簡単である。この直交化の列の長さが 2 で済むことがポイントである。

定理 4.1. 2

$$\begin{cases} w_1 &= c \\ w_i &= Aw_{i-1} - \dfrac{(w_{i-1}, Aw_{i-1})_A}{(w_{i-1}, w_{i-1})_A} w_{i-1} - \dfrac{(w_{i-2}, Aw_{i-1})_A}{(w_{i-2}, w_{i-2})_A} w_{i-2} \end{cases}$$

として作った $w_1, w_2, \ldots$ は A-直交する。

証明　命題 “Aw_{i-1} は w_{i-3} 以下と直交している。”ことを示せば良い。i についての帰納法による。$i = 1, 2$ に対しては正しい。$i-2$ まで正しかったとして $i-1$ のときに証明する。$j \geq 3$ に対して

$$(w_{i-j}, Aw_{i-1})_A = w_{i-j}^T A A w_{i-1} = (Aw_{i-j})^T A w_{i-1} = (Aw_{i-j}, w_{i-1})_A$$

であり w_{i-1} は帰納法の仮定から $w_{i-2}, w_{i-3}, \ldots$ と A-直交している。$Aw_{i-j} \subset L(w_1, w_2, \ldots, w_{i-j+1})$ であり、$j \geq 3$ であったから命題が示された（A が対称行列であることがポイント）。

4.1.4　例

$$v_1 = c = (0,\, 3,\, 4)^T$$

から始めると

$$v_2 = (2,\, -26/11,\, 28/33)^T, \quad v_3 = (-6/25,\, -2/25,\, 2/25)^T$$

である。

4.1.5　Lanczos 法の実際

前節までの考察を踏まえて Lanczos 法をきちんと記述する。
A は正定値対称行列とし、これにより定義される内積を考える。
ベクトル列

$$c,\, Ac,\, A^2c,\, \ldots, A^r c, \ldots$$

を考える[1]。全体空間は有限次元だからこの列はどこかで一定の空間 W_m を生成するようになる。つまり

$$c,\, Ac,\, A^2c,\, \ldots, A^{m-1}c$$

までは線形独立であり、これ以降はこれらの線形結合に書けるようになる。これから Gram-Schmidt の直交化法により A-直交基底を作る。

$$w_1,\, w_2, \ldots, w_m$$

とする。つまり

$$\begin{cases} w_1 &= c \\ w_i &= Aw_{i-1} - \dfrac{(w_{i-1}, Aw_{i-1})_A}{(w_{i-1}, w_{i-1})_A} w_{i-1} - \dfrac{(w_{i-2}, Aw_{i-1})_A}{(w_{i-2}, w_{i-2})_A} w_{i-2} \end{cases}$$

[1] これらで生成される部分空間を Krylov 空間という。

定理 4.1. 3

$$x = \sum_{j=1}^{m} \frac{(w_j, c)}{(w_j, w_j)_A} w_j$$

は $Ax = c$ をみたす。

証明 $\forall i$ について

$$\begin{aligned}(w_i, Ax) &= (w_i, x)_A \\ &= \sum_{j=1}^{m} \frac{(w_j, c)}{(w_j, w_j)_A}(w_i, w_j)_A \\ &= \frac{(w_i, c)}{(w_i, w_i)_A}(w_i, w_i)_A \\ &= (w_i, c)\end{aligned}$$

したがって

$$(w_i, Ax - c) = 0 \quad \forall i$$

である。ここで W_m の作り方から

$$AW_m \subset W_m$$

なので $Ax - c$ は W_m の元である。そして $\{w_i,\ i = 1, \ldots, m\}$ は W_m の基底であるからそのすべてと通常の内積で直交するベクトルは 0 ベクトルである。したがって $Ax = c$ となる。

4.2 Lanczos 法（標数 $p > 0$ の場合）

4.2.1 Lanczos 法（標数 $p > 0$ の場合）の考え方（1）

Lanczos 法を標数 $p > 0$ の場合に適用しようとすると“絶対値$= 0$ のベクトル”と“0ベクトル”が一致しないという難点に遭遇する。とくに標数$= 2$ の場合は非0ベクトルの2つに1つが絶対値$= 0$ である点が致命的に思える[2]。

この困難を克服するには単独のベクトル、つまり1次元部分空間、から2次元以上の部分空間に対象を変えるという転換が必要であった。たとえば次のような例がヒントになる。

$u = (1, 1, 0)^T$, $v = (1, 0, 1)^T$ は標数$= 2$ では共に絶対値$= 0$ である。しかし内積がつくる行列は

$$\begin{pmatrix} (u, u) & (u, v) \\ (v, u) & (v, v) \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

であるので u, v が張る空間 U の“2次元空間としての絶対値”は 0 ではない。さらに $w = (1, 1, 1)^T$ が張る空間 W を考えると w は u, v と“直交”するので全体空間は U と W の直和に直交分解できる。

実際、任意のベクトル $r = (x, y, z)^T$ の U-成分は u, v を並べた行列を $U = (u, v)$ とすると

$$(U^T U)^{-1} U^T r$$

である。

[2]逆に言うと標数が（次元に比べて）巨大な場合は“絶対値$= 0$ のベクトル”と“0ベクトル”はほとんど常に一致するので標数$= 0$ の場合と同じに考えて実用上は問題ない。

ここから一般論。A は正則行列とする。もし W が2つの部分空間の直和 $W = W_1 + W_2$ になっていてそれらが A-直交していれば任意の $w \in W$ は

$$w = w_1 + w_2, \quad w_i \in W_i$$

と書ける。W_1 の基底を $w_{11}, w_{12}, \ldots, w_{1r}$ とし、

$$w_1 = \sum_{j=1}^{r} c_{1j} w_{1j}$$

とおいて

$$(w_{1i}, w)_A = (w_{1i}, w_1)_A = \sum_{j=1}^{r} c_{1j} (w_{1i}, w_{1j})_A$$

で c_{1j} を決めたい。

$$\begin{pmatrix} (w_{11}, w)_A \\ (w_{12}, w)_A \\ \cdots \\ (w_{1i}, w)_A \end{pmatrix} = \begin{pmatrix} (w_{11}, w_{11})_A & \cdots & (w_{11}, w_{1r})_A \\ (w_{12}, w_{11})_A & \cdots & (w_{12}, w_{1r})_A \\ \cdots & \cdots & \cdots \\ (w_{1r}, w_{11})_A & \cdots & (w_{1r}, w_{1r})_A \end{pmatrix} \begin{pmatrix} c_{11} \\ c_{12} \\ \cdots \\ c_{1r} \end{pmatrix}$$

であるから、右辺の係数行列を S_1 とするとき、もし S_1 が正則ならば

$$\begin{pmatrix} c_{11} \\ c_{12} \\ \cdots \\ c_{1i} \end{pmatrix} = S_1^{-1} \begin{pmatrix} (w_{11}, w)_A \\ (w_{12}, w)_A \\ \cdots \\ (w_{1i}, w)_A \end{pmatrix}$$

である。

ここで $w_{11}, w_{12}, \ldots, w_{1r}$ を列ベクトルとする行列を W_1 とすれば以上は次のように書ける。

$$w_1 = W_1 \begin{pmatrix} c_{11} \\ c_{12} \\ \cdots \\ c_{1i} \end{pmatrix} = W_1 (W_1, W_1)_A^{-1} (W_1, w)_A$$

ここで内積の記号を自然に行列に拡張している。たとえば $(W_1, W_1)_A$ は行列であるから -1 乗は逆行列を表す。

この場合の利点は S_1, S_2 が共に正則ならば問題が二つに分割され、逆行列の計算が楽になる。さらに細分され究極の分解である1次元空間の直交和になればそれが一番最初の Lanczos 法の発想になる。

4.2.2 Block Lanczos 法（標数 $p > 0$ の場合）の考え方（1）

標数 0 の場合と同様に全体空間でなく A の作用によって部分空間を作って行くことを考えよう。

$$b,\ Ab,\ A^2 b, \ldots$$

今回は“絶対値=0 の非ゼロベクトル”が出てくるので、直交化が 1 段階ごとにできるとは限らないのである。そこで

$$b,\ Ab,\ A^2 b, \ldots, A^e b$$

の内積行列が正則になるまで e を増やし、次の $A^{e+1}b$ はこれを使って直交化することになる。

しかしこういう e がなければ議論は先に進まない。それで一般の解法は断念せざるを得ない。

では $Ax = b$ という 1 次元の解を求める問題でなく、与えられた部分空間 B に対して

$$AX \subset B$$

となる部分空間 X を求める問題にしてみる。こうすることの利点は基底を作るベクトルの“絶対値”が 0 でも部分空間としては A-正則でありうることである。

例 4.2. 1

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}, \quad B = B_0 = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix} \quad \text{とすると}$$

$$AB_0 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{pmatrix}$$

である。これから B_0 成分を消すと

$$AB_0 + B_0(B_0, B_0)_A^{-1}(B_0, AB_0)_A = \begin{pmatrix} 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}$$

となるので

$$B_1 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \end{pmatrix}, \quad \text{として、} \quad AB_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

を作り、B_0, B_1 成分を消して直交化すると

$$AB_1 + B_0(B_0, B_0)_A^{-1}(B_0, AB_1)_A + B_1(B_1, B_1)_A^{-1}(B_1, AB_1)_A = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

となって、これで解が求まる。

$$X = B_0(B_0, B_0)_A^{-1}(B_0, B) + B_1(B_1, B_1)_A^{-1}(B_1, B) = \begin{pmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}$$

とすると

$$AX = B$$

である。ここで幸運だったのは $(B_0, B_0)_A^{-1}$, $(B_1, B_1)_A^{-1}$ が存在したことである。どちらかが作れなかったならば解も得られなかったことになる。また今の場合は偶然 $AX = B$ になったがこれも非常に幸運であった。

4.2.3 Block-Lanczos 法（標数 $p > 0$ の場合）の考え方

数体ふるい法では

$$Ax = 0$$

となる $x \neq 0$ を求めることが問題である。もちろんこのとき A は正則ではない。この問題には c をいろいろ変えて $b = Ac$ を作り、これに対し

$$Ay = b$$

の解を見つける。この解から元の解が計算できる。

$$A(y - c) = 0$$

単独の c ではうまく進まないことはすでに述べた通りである。そこで c を複数（32 個あるいは 64 個程度）とって単独のベクトルから部分空間に移行する。

ランダムに列ベクトルを n 個とり $c_1, c_2, \ldots, c_n$ とし、$b_i = Ac_i$ とする。

問題を $b_1, b_2, \ldots, b_n$ の生成する部分空間を B として

$$AX \subset B$$

となる部分空間 $X \neq 0$ を求めるものに変形する[3]。

記号　以下 $v_1, v_2, \ldots, v_r$ をこの順に並べてできる行列とこれらが生成する部分空間とを同じ記号で表す。

部分空間の列の作り方は、まず

$$W_0 = B$$

から始めて

$$W_0 \text{ は } A\text{−内積に関して正則}$$

だと仮定する。さらに次々

$$AW_i \text{ は} A\text{−内積に関して正則}$$

であると仮定し、AW_i から $W_1, W_2, \ldots, W_{i-1}$ 成分を除いて A-直交化したものを W_{i+1} とする。

このプロセスを進めて行くと、有限次元なので、いつか

$$AW_m \subset W_1 + W_2 + \cdots + W_m$$

となる。つまり

$$W = W_1 + W_2 + \cdots + W_m$$

とおくと

$$AW \subset W$$

となる。

定理 4.2.2　一般に部分空間の直和分解 $V = V_1 + V_2 + \cdots + V_r$ で $(V_i, V_j)_A = 0$ if $i \neq j$ であるとき V の列ベクトルを並べて作った行列 U は次の分解を持つ[a]。

$$U = \sum_{j=1}^{r} V_j (V_j, V_j)_A^{-1} (V_j, U)_A$$

[a] V_j は定義は部分空間だが内積の式の中では基底を並べて作った行列と見ている。

[3] もちろん C と無縁な解を得たいので X を求める段階では C のことは忘れることにする。

この方法の欠点はプロセスが長続きしないことである。AW_i はすぐさま A-内積に関して正則でなくなってしまうのである。

そこで、AW_i が A-正則でなくても A-正則で maxmal な列ベクトルからなる部分行列をとり、そこから W_{i+1} を作る。そして W_{i+1} で使われなかった列ベクトルは $W^{(i+1)}$ として次の i に渡し、次では必ずこれを含めた A-正則な部分行列を作ることにする。もし次で使えなかったならこのアルゴリズムは失敗であって、まったく最初からやり直すことになる[4]。

$W^{(i+1)}$ を必ずすぐ次で使わなければいけないのは次の理由からである。

補題 4.2. 3

AW_i は $W_{i-3}, W_{i-4}, \ldots$ と $A-$直交する

証明 $(W_{i-3}, AW_i)_A = (AW_{i-3}, W_i)_A$ であって AW_{i-3} は W_{i-2} と $W^{(i-2)}$ に分かれた。後者は W_{i-1} に含まれるから W_i と A-直交する。

注意 $W^{(i-1)} = 0$ ならば $AW_{i-2} = W_{i-1}$ となるので $(W_{i-2}, AW_i)_A = (AW_{i-2}, W_i)_A = (W_{i-1}, W_i)_A = 0$ である。したがって W_{i-2} 成分を計算する必要はなく直交化の長さは2で済む。

[4]実験してみると途中の繰り返しで使えないことはまずない。一番最後の繰り返しで起こるのみである。このときは解はほとんど見つかっているのでそのまま進めて問題はない。

4.2.4 まとめ

n 行 m 列行列 A_0 に対し $A_0 v = 0$ となる m 次ベクトル v を（複数）求める方法を以下に記す。

1. m 次対称行列 $A = A_0^T A_0$ を考える。
2. s 個のランダムな m 次列ベクトル $c_1, c_2, \ldots, c_s$ から $b_i = Ac_i,\ (i = 1, 2, \ldots, s)$ を作る。
3. 以下 $AY \subset B$ となる Y を求める行程を記述する。ただし B は $b_1, b_2, \ldots, b_s$ で張られた部分空間である。
4. $\{b_1, b_2, \ldots, b_s\}$ に対し、maximal な A-正則部分集合をとり、それらを並べて作る行列を W_1 とする。使われなかった列ベクトルを並べて作る行列を $W^{(1)}$ とする。
5. $Y = W_1(W_1, W_1)_A^{-1}(W_1, B)$ を計算する。
6. i 番目の $W_i, W^{(i)}$ ができているときに $i+1$ 番目に移る操作を 7.–12. に記す。
7. AW_i の列ベクトルがそれぞれ W_i, W_{i-1}, W_{i-2} と A-直交するようにする。この列ベクトルの生成する空間、あるいは並べて作る行列を B_{i+1} とする。
8. $B_{i+1} = 0$ ならば 13. へ
9. $B_{i+1}, W^{(i)}$ の列ベクトルの集合から maximal A-正則な部分集合をとって W_{i+1} とする。このとき $W^{(i)}$ の元を優先する。A-正則な部分集合がなければ 14. へ。
10. B_{i+1} の元であって使われなかったものを並べて作る行列を $W^{(i+1)}$ とする。
11. Y に $W_{i+1}(W_{i+1}, W_{i+1})_A^{-1}(W_{i+1}, B)$ を加える。
12. i を 1 増やして 6. へ戻る。
13. $Y - C$ の s 個のベクトルが作る部分空間の元で $A_0 v = 0$ となるものが解となって終了。それを見つけるには $A_0(Y - C)$ に列についての基本変形を施して 0 ベクトルの個数を最大にする。同時に $Y - C$ に同じ変形を施せば 0 ベクトルに対応する列ベクトルが解である。ただし C は $c_1, c_2, \ldots, c_s$ を並べて作った行列である。
14. 異常終了である。幸運な場合は $Y - C$ の s 個のベクトルが作る部分空間の元で $A_0 v = 0$ となるものがあることがある。それを見つけるには $A_0(Y - C)$ に列についての基本変形を施して 0 ベクトルの個数を最大にする。同時に $Y - C$ に同じ変形を施せば 0 ベクトルに対応する列ベクトルが解である。十分な個数の解が得られない場合は 2. からやり直し。

注意 1. A を作らないのは A_0 が sparse であっても A は sparse とは限らないからである。とくに A_0 に dense な列がひとつでもあると A は非常に dense なものとなる。なお A による変換は A_0, A_0^T による変換を続けて行う。A_0, A_0^T のどちらをメモリに置くかで速度が違うことがある。NFS の場合は A_0^T を置くのが正解であるようだ。これは素元分解の要素をそのまま並べたものなのでその意味でも都合が良い。

注意 13. $Av = 0$ だが $A_0 v \neq 0$ ということは A_0 のランクが n より小さい時に起こりうる。これを避けるには m を n より大き目にとるのが簡便。NFS の場合は sieve を少し先まで行って $5\% - 10\%$ 程度多めにデータを集めておけば良い。A_0 が確定している場合は 2. からやり直すしかないだろう。

注意 14. 普通はこの型の終了となるので“異常”終了というのは適当ではないかもしれない。20 個以上の解が得られることが多い。

4.2.5 高速化

Lanczos 法の律速段階は行列×ベクトルではなくベクトルとベクトルの内積の計算にある。

そこで必要な内積をすべて直接計算するのではなく、できるだけ他の内積などから導き出すことが必要になる。

Chapter 5

数体ふるい法（その３）

代数的数の平方根

2002/3/30　CRYPTREC 研究会 at 立教大学 S302 教室
2003/3/31 改訂
木田 祐司

5.1　平方イデアルから平方数へ

Lanczos 法などにより

$$\alpha = \prod_{i \in I} (a_i + b_i \theta)$$

の生成するイデアル $<\alpha>$ が平方イデアルになるような i の集合 I を見つけることができる。

しかしまだ　“平方イデアルである単項イデアル”と“単項イデアルの平方”　には違いがある。

ここで Adleman は平方剰余記号を用いて gap を埋めることを考えついた。

- [L.M. Adleman] ”Factoring Numbers Using Singular Integers”, 23rd STOC(1991) の予稿

$K = \mathbf{Q}(\theta)$ の素イデアルで1次のもの $P = (p, \theta - s)$ をとる。ただし p はどの $a + b\theta$ をも割らないような大きなものとする。このとき $a + b\theta$ が $\bmod P$ で平方数であることは $a + bs$ が $\bmod p$ で平方数であることと同じである。

つまり $a + b\theta$ が K で平方数であるためには $a + bs$ が $\bmod p$ の平方剰余でなければならない。あるいは $a + bs$ が $\bmod p$ の平方剰余であれば $a + b\theta$ が K で平方数である可能性が高まるともいえる。

厳密には

$$S_K = \{x \in K \mid <x> \text{ は平方イデアル}\}$$

とすると $S_K / S_K \cap K^2$ の rank は高々 単数群の2-rank+イデアル類群の2-rank であることが証明できる。

つまりこの剰余群の生成系になるような複数の P に対して $\alpha \in K$ が平方剰余であれば α は K の平方数である。生成系になることを確かめるのは難しいから余分にとることで代替する。単数群の2-rank+イデアル類群の2-rank の2倍程度を Adleman は提案している。実際の GNFS での計算では多項式の次数が 5 であれば 10 個もあれば十分である。大して手間が変わらないので 32-bit CPU ならば 32 個 にすれば良い。

そこで最初の解 α が 32 個の 1 次素イデアルに関して平方剰余になれば平方数であることはほとんど確実である。もし駄目ならば複数の解 α を組み合わせて 32 個の 1 次素イデアルに対して平方剰余になるものを探せば良い。これはせいぜい 32 次行列の処理なのでまったく簡単である。

素イデアルは 31 個として、あとの 1 個は積の個数 mod 2 として積の個数が偶数になるようにすることもある。

例 5.1. 1

$$
\begin{aligned}
C60 &= 259405968614692311540344402656970765595208324844841957314041 \\
f(x) &= 316526201x^5 + 274735005x^4 + 5618090324x^3 + 6994621459x^2 + 6749441074x + 2378868742 \\
m &= 15230496453
\end{aligned}
$$

として GNFS を行ったときの例。以下これを GNFS60 として引用する。

Lanczos 法で得られた 29 個の解に 23 個の 1 次素イデアルによる平方剰余記号と偶数個数判定部分を計算したもの。

```
解 1 : 00011011010100000010001
解 2 : 01111000110001010001000
解 3 : 00011011010100000010001
解 4 : 01100011100101010010001
解 5 : 01111000110001010001000
解 6 : 00000000000000000000000
解 7 : 01111000110001010001000
解 8 : 00011011010100000010001
解 9 : 01100011100101010010001
解10 : 00000000000000000000000
解11 : 00100101001111100001000
解12 : 01100011100101010010001
解13 : 00100101001111100001000
解14 : 01011101111110100000000
解15 : 00000000000000000000000
解16 : 01111000110001010001000
解17 : 00000000000000000000000
解18 : 01100011100101010010001
解19 : 01011101111110100000000
解20 : 00100101001111100001000
解21 : 01011101111110100000000
解22 : 01000110101010110011001
解23 : 01000110101010110011001
解24 : 01000110101010110011001
解25 : 01000110101010110011001
解26 : 00011011010100000010001
解27 : 00100101001111100001000
解28 : 01000110101010110011001
解29 : 00100101001111100001000
```

この中で組み合わせて 0 になるものを *Gauss* 消去で探すと独立なものが 26 組あった。

5.2 どういう数の平方根か

この節での課題は

$$\alpha = \prod (a_i + b_i\theta)$$

が平方数であることがわかっているときに

$$\alpha = \beta^2$$

となる $\beta \in \mathbf{Z}[\theta]$ を求めることである。

整数環が UFD の場合は単数と素元によって既約分解し、各々のべき指数を半分にしたべき乗の積を作れば良い。

例 5.2. 1 前節の GNFS60 の場合の例

$$\begin{aligned}\alpha \quad = \quad & (-101418+\theta)(-5598+\theta)(-343+\theta)(-173+\theta)(-135+\theta)\\ & (67+\theta)(318+\theta)(3625+\theta)(-33+2\theta)(-21+2\theta)\\ & \cdots\\ & (-13582+57\theta)(51371+11855\theta)(37656+59923\theta)(-25772+23449\theta)(977+35646\theta)\\ & (-70601+85108\theta)(-65263+21034\theta)(-96403+57785\theta)(10792+67\theta)(94926+18839\theta)\end{aligned}$$

は 48980 個の積であって、その生成する単項イデアルの素イデアル分解は分かっている。

$$<\alpha>= P_1^{16588}P_2^{13398}P_3^{13272}\cdots P_{38093}^{2}P_{38094}^{0}P_{38095}^{4}(large\ primes)^2$$

ここで

$$P_1=(2,\,\theta),\ P_2=(3,\,\theta-2),\ P_3=(5,\,\theta-1)$$
$$\cdots$$
$$P_{38093}=(453461,\,\theta-370232),\ P_{38094}=(453527,\,\theta-293393),\ P_{38095}=(453527,\,\theta-328256)$$

である。

したがって

$$\sqrt{<\alpha>}=P_1^{8294}P_2^{6699}P_3^{6636}\cdots P_{38093}^{1}P_{38094}^{0}P_{38095}^{2}(large\ primes)$$

であり

$$|\mathcal{N}\sqrt{\alpha}|=|\mathcal{N}\beta|=2^{8294}3^{6699}5^{6636}\cdots 453461^1 453527^0 453527^2\mathcal{N}(large\ primes)$$

である。

5.3 一般論：Hensel lift を用いる

NFS とは無関係に、一般に、既約多項式 $f(x)$ のひとつの根 θ から定まる代数体 $K=\mathbf{Q}(\theta)$ において平方数

$$\alpha=a_{d-1}\theta^{d-1}+a_{d-2}\theta^{d-2}+\cdots+a_1\theta+a_0,\quad a_i\in\mathbf{Q}$$

に対してその平方根 β を

$$\beta=b_{d-1}\theta^{d-1}+b_{d-2}\theta^{d-2}+\cdots+b_1\theta+b_0,\quad b_i\in\mathbf{Q}$$

計算する方法を考える。

条件 $f(x)$ が $\bmod q$ で既約となる素数 q が存在する。

この条件を満たさない多項式が稀に存在する。SNFS では多項式が決まっているのでこの条件を満たさない場合がありうるが GNFS ならばいくらでも多項式を変えて条件を満たすようにすることができる。

奇素数 q を $f(x)$ が $\bmod q$ で既約で、かつ α は q では割れないものとする。すると q は K でも素数である。

最初の計算が容易なように小さな素数にとる。まず

$$\delta_0^2\alpha\equiv 1 \bmod q$$

となる $\delta_0 \in \mathbf{Z}[\theta]$ を求める。つまり α の逆数の $\bmod q$ での平方根である。逆数にするのは次の段階の Hensel lift における逆数計算を避けるためである。

例 5.3. 1 $f(x) = x^3 + 2x^2 + 3x + 4$ とする。

$$\alpha = 61\theta^2 + 218\theta + 396$$

の平方根を求めたい。

$q = 3$ としよう。r, s, t, u を $0, 1, \cdots, q-1$ で動かして

$$\left(r\theta^2 + s\theta + t\right)^2 (\theta^2 + \theta) \equiv 1 \bmod q$$

となるものを探すのだが、代数的数 θ を $x \bmod f(x)$ と同一視すれば

$$\left(rx^2 + sx + t\right)^2 (x^2 + x) \equiv 1 \bmod (q, f(x))$$

となるものを探すことになる。これは高々 q^3 回のループであるから工夫を凝らす必要はない。

解はすぐに見つかって $\delta = \theta^2 + 1$ である。

これから始めて modulus を $q^2, q^{2^2}, \ldots$ と上げて行く。

5.3.1 Hensel lift

命題 5.3. 2

$$\delta_j{}^2 \alpha \equiv 1 \bmod < q >^{2^j}$$

のとき

$$\delta_{j+1} = \delta_j + \delta_j \frac{1 - \delta_j^2 \alpha}{2} \bmod < q >^{2^{j+1}}$$

とすれば *(δ_j は $\bmod < q >^{2^j}$ で、α は $\bmod < q >^{2^{j+1}}$ で計算する)*

$$\delta_{j+1}^2 \alpha \equiv 1 \bmod < q >^{2^{j+1}}$$

である。

Hensel lift を繰り返して q^{2^k} が十分大きくなれば

$$\delta_k^2 \alpha \equiv 1 \bmod < q >^{2^k}$$

より

$$(\delta_k \alpha)^2 \equiv \alpha \bmod < q >^{2^k}$$

となって $\delta_k \alpha$ の係数を $[-q^{2^k}/2, q^{2^k}/2)$ でとったものが求める平方根（のひとつ）になる。

例 5.3. 3 今までの計算を続けると

$\bmod 3^2 = 9$ では

$$\delta_1 = 4\theta^2 + 6\theta + 4$$

であり、このとき

$$\delta_1 \alpha \equiv 4\theta^2 + 4\theta - 3 \bmod 3^2$$

となる。

$\mathrm{mod}\ 3^4 = 81$ では

$$\delta_2 = 13\theta^2 - 12\theta + 13$$

であり、このとき

$$\delta_2\alpha \equiv -5\theta^2 + 4\theta + 6 \bmod 3^4$$

である。

これより $-5\theta^2 + 4\theta + 6$ が解らしいことがわかる。試してみると実際にそうである。

5.4 中国の剰余定理を用いる

簡単のため θ は代数的整数であるとする。

NFS の場合に戻ると α は小さな数の多数の積になっていた。これを前節の方法で処理しようとすると積を展開しなければいけない。それは非常に大きくなって計算が不可能になる。このような時のいつもの手段である中国の剰余定理を適用してみる。つまり、複数の modulus を用いて間接的に計算するのである。

q を有理素数で $K = \mathbf{Q}(\theta)$ においても q が生成する単項イデアルは素イデアルであるものとする。こういうものを複数用意する。

$$\alpha = \prod_{i\in I}(a_i + b_i\theta) = g_{d-1}\theta^{d-1} + \cdots + g_1\theta + g_0, \quad g_i \in \mathbf{Z}$$

の g_i は巨大であっても

$$\alpha \equiv g_{q,d-1}\theta^{d-1} + \cdots + g_{q,1}\theta + g_{q,0} \bmod < q >$$

の $g_{q,i}$ は非常に小さい。

q を十分な個数だけ準備すれば g_i は $g_{q,i} \bmod q$ の集まりで間接的に表現される。

このとき

$$\alpha \equiv (\beta_q)^2 \bmod < q >$$

となる β_q は簡単に計算できる。$\alpha = \beta^2$ とすると

$$\beta \equiv \pm\beta_q \bmod < q >$$

である。しかしこの正負が q によってまちまちであって決めることができない。それは次のような有理整数の平方根の場合と同様である。

例 5.4. 1

$$841 \equiv \begin{cases} (\pm 1)^2 & \bmod 3 \\ (\pm 1)^2 & \bmod 5 \\ (\pm 1)^2 & \bmod 7 \end{cases}$$

であるが $\sqrt{841} = 29$ であって

$$29 \equiv \begin{cases} -1 & \bmod 3 \\ -1 & \bmod 5 \\ 1 & \bmod 7 \end{cases}$$

である。

5.5 Couveignes の方法

- [Jean-Marc Couveignes] "Computing a square root for the number field sieve", in Lenstra and Lenstra(Eds.) *The development of the number fiels sieve*, Lecture Notes in Math. 1554, pp 95–102

による方法を解説する。この方法では以下の仮定は必須である。

仮定　次数 d は奇数である。

前節では

$$\beta \equiv \pm\beta_q \bmod < q >$$

となる $\pm$ が各 q ごとに異なることが障害であった。

しかし、上の仮定をおけば

$$\mathcal{N}(-\beta) = (-1)^d \mathcal{N}(\beta) = -\mathcal{N}(\beta)$$

であるから二つの平方根 $\pm\beta$ はそのノルムで区別できる。そこでノルムが $b = \sqrt{\mathcal{N}(\alpha)}$ であるものを β としてこれを $\bmod q$ で定める、というのが Couveignes のアイディアである。

前節にならって α の逆数の平方根 δ を求めることから始めるが奇数次なので前節のように総当りで探すのではなくもっとスマートに答えを見つけることができる。

命題 5.5. 1

$$\delta = \alpha^{\frac{(1+q+q^2+\cdots+q^{d-1})(q-2)-1}{2}} \cdot b \bmod < q >$$

とおくと

$$(\delta\alpha)^2 \equiv \alpha \bmod < q >$$

かつ

$$\mathcal{N}(\delta\alpha) \equiv b \bmod q$$

である。

たくさんの q で計算するよりも Hensel lift でいくらか q のべきを上げて使う方が効率が良いことが多い。しかしべき指数を上げすぎると中国の剰余定理を用いた意味がなくなってしまう。

例 5.5. 2　$f(x) = x^3 + 2x^2 + 3x + 4$ とする。

$$\alpha = 61\theta^2 + 218\theta + 396$$

の平方根を求めたい。

$$\mathcal{N}\alpha = 27984100$$

なので[1] $b = 5290$ である。これより

$$\beta = \delta\alpha = 2\theta^2 + 2\theta \bmod < 3 >$$

である。

mod 11 では

$$\beta = \delta\alpha = 5\theta^2 - 4\theta + 5 \bmod < 11 >$$

[1] この場合はノルムを計算するのが面倒であるが NFS の場合は簡単に求まる。

である。

よって $\beta = a\theta^2 + b\theta + c$ とおいて

$$\begin{aligned} a &\equiv \begin{cases} 2 \mod 3 \\ 5 \mod 11 \end{cases} \\ b &\equiv \begin{cases} 2 \mod 3 \\ -4 \mod 11 \end{cases} \\ c &\equiv \begin{cases} 0 \mod 3 \\ 5 \mod 11 \end{cases} \end{aligned}$$

を解けば

$$\begin{cases} a &\equiv 5 \bmod 33 \\ b &\equiv -4 \bmod 33 \\ c &\equiv -6 \bmod 33 \end{cases}$$

となる。すると $5\theta^2 - 4\theta - 6$ が候補になる。計算すると確かに解である。

5.5.1 中国の剰余定理の変形

素因数分解で必要になる値は，ターゲットの数 N を法とした値であることが多い。そのため中国の剰余定理も実際の値そのものではなく $\bmod N$ の値を求めるものに変形しておくと巨大な数をそのまま扱う必要がなくなるというメリットが出てくる。

問題 $m_1, m_2, \ldots, m_s$ は二つずつ互いに素な自然数とし，$M = \prod_{i=1}^{s} m_i$ とする。

$-0.49M < x < 0.49M$ にある x が[2]

$$x \equiv \begin{cases} x_1 \bmod m_1 \\ x_2 \bmod m_2 \\ \cdots \\ x_s \bmod m_s \end{cases}$$

を満たしているときに $0 < N < M$ なる N に対して $x \bmod N$ を計算せよ。

通常の解法にならえば

$$M_i = M/m_i,\ a_i = 1/M_i \bmod m_i$$

とおいて[3]

$$z = \sum_{i=1}^{s} a_i x_i M_i$$

とすれば

$$x \equiv z \bmod M$$

となるのであった。これから $-0.49M < x < 0.49M$ となる x を定めて $\bmod N$ で落とせば良い。問題は M のオーダーでなく N のオーダーの計算で済ませたいということ。

[2] 0.49 は誤差を見込んでいる。本来は 0.5 である。

[3] a_i のための M_i は $\bmod m_i$ で計算すれば良いが、次の z のための M_i は $\bmod N$ の値であることに注意。つまり二通りに計算しなければいけない。

$x = z - rM,\ r \in \mathbf{Z}$ とすると

$$-0.49 < \frac{z}{M} - r < 0.49$$

より $r = round(\frac{z}{M})$ であるから

$$r = round\left(\sum_{i=1}^{s} \frac{a_i x_i}{m_i}\right)$$

となり，m_i 程度の大きさの実数の s 個の和であるから倍精度程度の計算で十分である。これで $x = z - rM$ が確定する。この式の各項を $\bmod N$ で計算すれば良い。

5.5.2 高速化

$$\alpha = \prod_{i=1}^{r}(a_i + b_i\theta)$$

の右辺の積を二つに分ける

$$\begin{aligned} \alpha_{odd} &= \prod_{i=1,\ i \text{ odd}}^{r} (a_i + b_i\theta) \\ \alpha_{even} &= \prod_{i=1,\ i \text{ even}}^{r} (a_i + b_i\theta) \end{aligned}$$

そして

$$\overline{\alpha} = c^2 \frac{\alpha_{odd}}{\alpha_{even}}$$

を考える。ここで c は両辺を代数的整数にするために掛ける数である。係数を小さくするという点では代数的数をとるべきであるがなかなか難しい。有理整数にとれば簡単ではある。

そこで

$$\overline{\gamma}^2 = \overline{\alpha}$$

となる $\overline{\gamma}$ を計算して

$$\gamma = \overline{\gamma}\alpha_{even}/c$$

とおけば

$$\gamma^2 = \alpha$$

である。

簡単のため奇数番目と偶数番目に分けたが、目的は分子・分母の相殺にあるのだから large primes が一致するペアは分子・分母に分かれるようにしたい。それにはあらかじめ $a_i + b_i\theta$ を large primes が消し合うペアが続き番号になるようにしておけば良い。

例 5.5. 3 GNFS60 の場合の例

$$\alpha = (-101418 + \theta)\cdots(94926 + 18839\theta), \quad (48980\,\text{個の積})$$

に対して

$$< \alpha >= P_1^{16588} P_2^{13398} P_3^{13272} \cdots P_{38093}^{2} P_{38094}^{0} P_{38095}^{4} (large\ primes)^2$$

であったが

$$< \frac{\alpha_{odd}}{\alpha_{even}} > = P_1^{36} P_2^{-130} P_3^{10} \cdots P_{38093}^{-2} P_{38094}^{0} P_{38095}^{0}$$

となって大幅にべき指数が小さくなっている。また *large primes* 部が消えるようにも並べ替えている *((1+1) large primes* ではこれが可能*)*。

5.6 Montgomery の方法

- [Peter L. Montgomery] "Square Roots of Product of Algebraic Numbers", Proceedings of Symposia in Applied Mathematics **48**(1994), 567–571.
- [Phong Nguyen] "A Montgomery-Like Square Root for the Number Field Sieve", in *ANT III*, Lecture Notes in Computer Sciences **1423**(1998), 151–168.

の簡単な紹介をする。話の都合上記号も変更している。

NFS の場合は各 $a_i + b_i\theta$ が生成する単項イデアルの素イデアル分解はわかっているから容易に

$$\alpha = \prod(a_i + b_i\theta)$$

の生成する単項イデアルの平方根イデアル $\sqrt{<\alpha>}$ はわかる。かつその素因子は具体的に書けている。

例 5.6.1 GNFS60 の場合の例では α は 48980 個の積であって、

$$<\beta>= P_1^{8294}P_2^{6699}P_3^{6636}\cdots P_{38093}^{1}P_{38094}^{0}P_{38095}^{2}(large\ primes)$$

であった。ここで $P_1 = (2,\, \theta),\ P_2 = (3,\, \theta - 2), \ldots$ である。

前節のように積を分けると

$$<\frac{\alpha_{odd}}{\alpha_{even}}>= \frac{P_1^{36}P_3^{10}P_4^{380}\cdots P_{38089}^{2}}{P_2^{130}P_5^{134}\cdots P_{38093}^{2}}$$

と簡単になる（このときの $a_i + b_i\theta$ の分子・分母への割り振り方も重要な問題である）。

そこで問題をこの平方根イデアル

$$\frac{P_1^{18}P_3^{5}P_4^{190}\cdots P_{38089}}{P_2^{65}P_5^{67}\cdots P_{38093}}$$

の生成元、これも β と書くことにする、を求めることにする。

一気に求めるのは無理なので段階的に行う。まず分子の一部 I をノルムがある適当な大きさ（100 桁程度）になるだけ取り出す。たとえば $I_1 = P_1^{18}P_3^{5}$ としよう。

$\beta_1 \in I_1$ とすると

$$<\beta_1>= I_1 \cdot H_1$$

という“誤差”イデアル H_1 が出てくる。これを小さくするために LLL-algorithm を用いてできるだけ short なものを見つける。すると

$$<\beta\,\frac{1}{\beta_1}>= \frac{P_4^{190}\cdots P_{38089}}{H_1P_2^{65}P_5^{67}\cdots P_{38093}^{1}}$$

となる。次は分母の一部、たとえば $H_1P_2^{65}$ を単項イデアルに置き換える。

$$<\beta\,\frac{\beta_2}{\beta_1}>= \frac{H_2P_4^{190}\cdots P_{38089}}{P_5^{67}\cdots P_{38093}^{1}}$$

これを繰り返す。最初にあった素イデアルを全部使い果たせば終了である。右辺の分子・分母のノルムの積がその前よりも常に小さくできたとすると最後に残る誤差イデアル H はノルムが非常に小さい。

このとき

$$<\alpha(\frac{\beta_2\beta_4\cdots}{\beta_1\beta_3\cdots})^2>= H^{\pm 2}$$

である。

ここで左辺は“小さな”平方数であるが多くの数の積であるので最終的には小さくても展開することは困難である。そこで適当な素数を法にして中国の剰余定理で計算することになる。

実際には、いろいろ注意しなければいけないことがある。たとえば最後の式の左辺はノルムが小さな数であるが、それは必ずしも“小さな”数であることを意味しない。たとえば単数の巨大なべきかもしれない。これを避けるためにこの数のすべての共役の絶対値が小さくなるような制約をつけた LLL-algorithm を実行することになる。