Computer Security Symposium 2014
22 - 24 October 2014

3 次元格子篩において用いられる格子点計算法の評価

早坂 健一郎 *　　青木 和麻呂 †　　小林 鉄太郎 †　　高木 剛 ‡

*九州大学大学院数理学府
819-0395　福岡市西区元岡 744

†NTT セキュアプラットフォーム研究所
180-8585　東京都武蔵野市緑町 3-9-11

‡九州大学 マス・フォア・インダストリ研究所
819-0395　福岡市西区元岡 744

あらまし 拡大体 GF(p^n) 上の離散対数問題の困難性は，ペアリング暗号の安全性基盤の一つである．数体篩法は拡大体 GF(p^n) 上の離散対数問題に対する現在最速の解法であるが，3 次元以上の領域における網羅的かつ効率的な格子点計算が課題であった．これに対して著者らは CSS2013 において 3 次元の領域における格子点計算法を提案した．また，ある条件を満たす格子に対し，上記の 3 次元格子点計算法を用いると網羅的に格子点を計算できることを実験により確かめた．本稿では，ある条件を満たす格子に対して 3 次元格子点計算法を用いれば，領域内の全ての格子点を効率的に計算可能であることを示す.

A Verification of 3-dimensional Lattice Sieve

Kenichiro HAYASAKA*　　Kazumaro AOKI†　　Tetsutaro KOBAYASHI†
Tsuyoshi TAKAGI‡

*Graduate School of Mathematics Kyushu University
744, Motooka, Nishi-ku, Fukuoka, 819-0395, Japan

†NTT Secure Platform Laboratories
3-9-11 Midori-cho, Musashino-shi, Tokyo, 180-8585, Japan

‡Institute of Mathematics for Industry
744, Motooka, Nishi-ku, Fukuoka, 819-0395, Japan

Abstract The security of pairing-based cryptosystem is based on the hardness to solve the discrete logarithm problem over an extension field GF(p^n). In CSS2013, we proposed an algorithm that enumerate lattice points of dimension three using bases which satisfy some conditions. Additionally, we confirmed that the algorithm can exhaustively enumerate the lattice points by an experiment. In this paper, we prove that the enumeration algorithm computes all lattice points in 3-dimensional region when the bases satisfy the proposed conditions.

1　はじめに

クラウドコンピューティングなどが広く利用され始めている現在において，情報の秘匿はますます重要な技術となっている．次世代公開鍵暗号であるペアリング暗号は ID ベース暗号や関数型暗号など利便性の高い暗号システムの実

現が可能であるとして注目されている実用化段階の公開鍵暗号である．ペアリング暗号の安全性は拡大体 $\mathrm{GF}(p^n)$ 上の離散対数問題を安全性の基礎としている．例として，MNT 曲線 [11] を用いた Tate pairing や BN 曲線 [1] を用いた Optimal ate pairing [13] が挙げられ，それぞれ $\mathrm{GF}(p^6)$，$\mathrm{GF}(p^{12})$ 上の離散対数問題を安全性の基盤としている．

標数 p が大きい拡大体 $\mathrm{GF}(p^n)$ 上の離散対数問題に対して，$p^n \to \infty$ とした場合漸近的に現時点で最速な解法は数体篩法 (JLSV06-NFS)[4] である．JLSV06-NFS は CRYPTO2006 において Joux らによって素体 $\mathrm{GF}(p)$ 上の離散対数問題に対する数体篩法 [3] の拡張として提案された．数体篩法には関係探索と呼ばれるステップが存在するが，500 ビットを超えるような大規模な素体 $\mathrm{GF}(p)$ に対する数体篩法では，関係探索ステップの実装手法として格子篩 [12] を用いることが主流である．この格子篩では，ある基底が与えられ 2 次元領域内に含まれる全ての格子点の計算を行うが，最小基底を用いた格子点計算法では大きなメモリ領域と複雑な計算を行う必要があった．これに対して Franke らは，特殊な基底を用いることで少ないメモリ領域と演算で 2 次元領域内の格子点の計算を行う手法 (Franke-Kleinjung 法) を提案した [2, 5]．

一方で，拡大体 $\mathrm{GF}(p^n)$ に対する数体篩法である JLSV06-NFS の格子篩では，3 次元以上の領域内に含まれる格子点の計算を行う場合がある．実際に [8, 9] では 7 次元の領域を用いている．文献 [8, 9] では最小基底を用いた格子篩であるため，広大なメモリ領域を使用していたが，従来の Franke-Kleinjung 法の 3 次元以上の領域への適用方法は知られていなかった．そこで早坂らは Franke-Kleinjung 法を拡張した 3 次元 Franke-Kleinjung 法を提案した [7]．文献 [7] では，Franke-Kleinjung 法において格子点計算に用いる基底が持つ条件を拡張し，3 次元領域での基底の条件とそれら条件を満たす基底を用いた格子点計算法を提案した．また，拡張した条件を満たすような基底がどれほど存在するか実験を行った．この結果，約 60%の確率で条件を満たすような基底を生成することができ，条件を満たすような基底を用いた場合は，ほぼ全ての格子点を計算することに成功した．ただし，理論的な証明は無かった．

本稿では，[7] において提案した条件を満たす基底を用いて格子点計算を行った場合，領域内の全ての格子点が計算できることを証明する．始めに，条件を満たした基底の向きや距離について証明を行う．次に， 条件を満たした基底と領域に含まれる格子点に対して順序を定義し，基底を加算することにより逐次的に格子点を計算できることを示す．

2　拡大体 $\mathrm{GF}(p^n)$ 上の数体篩法 (JLSV06-NFS) での格子篩

本節では，JLSV06-NFS[4] における格子篩 [7] について説明する．JLSV06-NFS は Joux らによって CRYPTO2006 において提案された拡大体 $\mathrm{GF}(p^n)$ 上の離散対数問題に対する解法であり，拡大体の標数 p が拡大次数 n に比べて大きい場合に現時点で最も効率的な解法である．JLSV06-NFS は多項式選択，関係探索，線形代数，個別離散対数計算の 4 つのステップを順に実行する．

2.1　関係探索

始めの多項式選択では，次の条件を満たすような異なる 2 つの代数体を定義する多項式 $f_1, f_2 \in \mathbb{Z}[X] \setminus \{0\}$ を選択する．$\deg f_1 = n$，f_1 は $\mathrm{GF}(p)$ 上既約，$f_1 \mid f_2 \pmod p$.

次に実行される関係探索では，整数 $t \geq 1$ に対して以降で述べる条件を満たすような $t+1$ 次元ベクトル $\mathbf{a} = (a_0, a_1, \ldots, a_t)^{\mathrm{T}}$ を多く収集する．多項式選択において選択された 2 つの多項式 f_1, f_2 と，ベクトル $\mathbf{a}$ を収集する領域の次元 $t+1$，smoothness bound $B_1, B_2 \in \mathbb{R}_{>0}$ に対して，factor base $\mathcal{B}_1, \mathcal{B}_2$ を

$$\begin{aligned}\mathcal{B}_i = \{(q, g) \mid\ & q\colon \mathrm{prime}, q \leq B_i,\\ & g\colon \text{irreducible monic factor of } f_i\\ & \text{in } \mathrm{GF}(q)[X], \deg g \leq t\}\end{aligned}$$

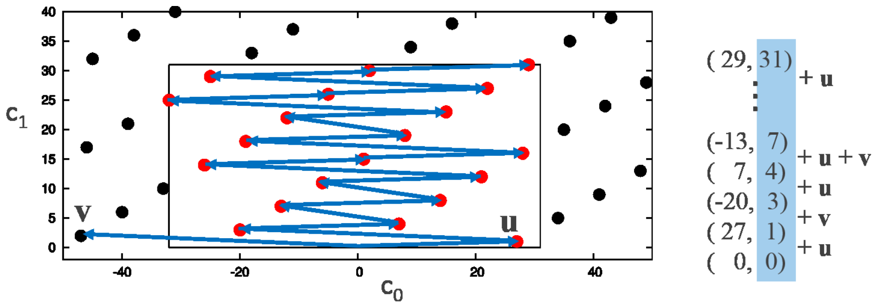

図 1: 2 次元 Franke-Kleinjung 法による格子点計算例

とする．ただし，$i = 1, 2$ である．また，ベクトル $\mathbf{a}$ と多項式 f_i $(i = 1, 2)$ に対してノルムを $N_i(\mathbf{a}) = |\mathrm{Res}(\sum_{j=0}^{t} a_j X^j, f_i(X))|$ とする．ただし，$f, g \in \mathbb{Z}[X]$ に対する $\mathrm{Res}(f, g)$ は f, g の終結式を表している．関係探索では，2 つの多項式 f_1, f_2 に対して以下を満たすようなベクトル $\mathbf{a}$ を $\sharp\mathcal{B}_1 + \sharp\mathcal{B}_2 + n$ 個以上収集する．(i) $N_1(\mathbf{a})$ が B_1-smooth, (ii) $N_2(\mathbf{a})$ が B_2-smooth, (iii) $\sum_{j=0}^{t} a_j X^j$ が $\mathbb{Z}[X]$ 上で既約．ここで，整数 $z > 0$ が B-smooth とは z の最大の素因数が B 以下であることである．本稿では上記を満たすような $\mathbf{a}$ を hit tuple と呼ぶ．

2.2 格子篩

2.1 節の関係探索では，hit tuple を収集するために両方の $i = 1, 2$ に対して $N_i(\mathbf{a})$ が B_i-smooth であるような $\mathbf{a}$ を探索する．このために JLSV06-NFS では hit tuple を探索する領域内において，B_i 以下の全ての素数 q に対して $q \mid N_i(\mathbf{a})$ であるような $\mathbf{a}$ を計算する．ここで，$\mathfrak{q} = (q, g) \in \mathcal{B}_i$ に対して $\mathfrak{q} \mid \mathbf{a} \Leftrightarrow g \mid \sum_{j=0}^{t} a_j X^j \pmod{q} \Rightarrow q \mid N_i(\mathbf{a})$ であることを利用すると，$\mathfrak{q} \mid \mathbf{a}$ であるような $\mathbf{a}$ を生成する基底を用いることで，試し割をすることなく $q \mid N_i(\mathbf{a})$ であるような $\mathbf{a}$ を計算できる．これを篩と呼ぶ．格子篩では，ある $\mathfrak{q} \in \mathcal{B}_i$ (special-$\mathfrak{q}$) に対して $\mathfrak{q} \mid \mathbf{a}$ であるような $\mathbf{a}$ の格子上で更に篩を行う．$\mathfrak{q} \in \mathcal{B}_i$ に対して $\mathfrak{q} \mid \mathbf{a}$ であるような $\mathbf{a}$ を生成する基底を列ベクトルとする $t+1$ 型正方行列を $M_{\mathfrak{q}}$ とする．また，$M_{\mathfrak{q}}$ の基底の係数 $\mathbf{c} \in \mathbb{Z}^{t+1}$ の空間 $\mathbb{Z}^{t+1}$ を $\mathbf{c}$-空間と呼ぶ．ここで，$\mathfrak{r} \in \mathcal{B}_1 \cup \mathcal{B}_2$ に対して $\mathfrak{r} \mid \mathbf{a}$ であるような $\mathbf{c}$ もまた $\mathbf{c}$-空間上で格子になっており，その基底を列ベクトルとする $t+1$ 型正方行列を $M_{\mathfrak{q},\mathfrak{r}}$ とする．格子篩では $I, J \in \mathbb{Z}_{>0}$ (ただし I は偶数) に対して $\mathbf{c}$-空間上において篩領域

$$\mathcal{H} = \{(c_0, c_1, \ldots, c_t)^{\mathrm{T}} \in \mathbb{Z}^{t+1} \mid -I/2 \le c_i < I/2 \ (0 \le i \le t-1), \ 0 \le c_t < J\}$$

を定め，$\mathcal{H}$ に含まれる $M_{\mathfrak{q},\mathfrak{r}}$ が生成する格子点を計算する．

2.3 Franke-Kleinjung 法

$\mathcal{H}$ と $M_{\mathfrak{q},\mathfrak{r}}$ に対して，$\mathcal{H}$ に含まれる $M_{\mathfrak{q},\mathfrak{r}}$ の格子点の効率的な計算方法として，$t = 1$ すなわち 2 次元篩領域では Franke-Kleinjung 法 [5] が知られている．Franke-Kleinjung 法では，特殊な形の基底を用いることで効率的に篩領域内に含まれる格子点を計算できる．また，格子点のうち第 2 成分に関して単調性をもっており，格子篩における必要なメモリ領域の大幅な削減に影響を与えた．Franke-Kleinjung 法の例を図 1 に示す．Franke-Kleinjung 法は以下の利点を持っている．

- 2 つの基底の加算によって，領域内の格子点を逐次的に全て計算できる．
- 計算される格子点の第 2 成分が単調増加である．

これに対して，$t = 2$ すなわち 3 次元篩領域ではFranke-Kleinjung 法を拡張した格子点計算法が提案された [7]．次節からこの 3 次元 Franke-Kleinjung 法について説明を行う．

2.4 3 次元 Franke-Kleinjung 法

本節では，3 次元 Franke-Kleinjung 法 [7] の概要について述べる．

文献 [7] では，**c**-空間上の基底を列ベクトルとする 3 次元正方行列 $M_{\mathfrak{q},\mathfrak{r}}$ 及び $\mathcal{H}$ を定義する閾値 I, J に対して，$M_{\mathfrak{q},\mathfrak{r}}$ を基底変換し $M_{\mathfrak{q},\mathfrak{r}}^{\mathrm{FK}}$ を生成するアルゴリズムと，$M_{\mathfrak{q},\mathfrak{r}}^{\mathrm{FK}}$ が満たすべき条件，条件を満たした場合 $\mathcal{H}$ に含まれる $M_{\mathfrak{q},\mathfrak{r}}$ の格子点を計算できるアルゴリズムを提案した．

基底 $\mathbf{u}, \mathbf{v}, \mathbf{w} \in \mathbb{Z}^3$ をそれぞれ $M_{\mathfrak{q},\mathfrak{r}}^{\mathrm{FK}}$ の列ベクトルとし，整数 I, J は $\mathcal{H}$ を定義する閾値とする．また，基底 $\mathbf{u}, \mathbf{v}$ は式 (1) を満たす列ベクトルのうち，第 1 成分がより大きい列ベクトルを $\mathbf{u}$，もう一方を $\mathbf{v}$ とし，残りの列ベクトルを $\mathbf{w}$ とする．このとき，$\mathbf{u}, \mathbf{v}, \mathbf{w}$ が以下の条件

- A1: $\forall \mathbf{x} = (x_0, x_1, x_2)^{\mathrm{T}} \in \{\mathbf{u}, \mathbf{v}, \mathbf{w}\}$,
  $|x_0| < I$ かつ $|x_1| < I$
- A2: 任意の $\mathbf{x}, \mathbf{y} \in \{\mathbf{u}, \mathbf{v}, \mathbf{w}\}, \mathbf{x} \neq \mathbf{y}$
  に対して $|x_0 - y_0| \geq I$ または $|x_1 - y_1| \geq I$
- A3: $\mathbf{x} = i\mathbf{u} + j\mathbf{v} - \mathbf{w}$ $(i, j \in \mathbb{Z}_{>0})$ としたとき，$|x_0| < I$ かつ $|x_1| < I$ を満たすような i, j が存在しない
- A4: $u_2 \geq 0$ かつ $v_2 \geq 0$ かつ $w_2 \geq 0$

を満たす場合，Algorithm 1 に示す格子点計算法により $\mathcal{H}$ に含まれる $M_{\mathfrak{q},\mathfrak{r}}$ の格子点を計算する．また，$\mathbf{u}, \mathbf{v}, \mathbf{w}$ を用いた格子点計算アルゴリズムを Algorithm 1 に示す．そして，上記条件を満たすような基底を用いて Algorithm 1 を繰り返し実行することにより，2.3 節と同様に以下の利点を備えた格子点計算を行うことが出来る．

- 3 つの基底の加算によって，領域内の格子点を逐次的に全て計算できる．
- 計算される格子点の第 3 成分が単調非減少である．

Algorithm 1：NEXTFK3($\mathcal{H}, \mathbf{u}, \mathbf{v}, \mathbf{w}, \mathbf{p}^{(i)}$)

Input: 篩領域 $\mathcal{H}$ の閾値 I，格子点 $\mathbf{p}^{(i)} = (p_0^{(i)}, p_1^{(i)}, p_2^{(i)})^{\mathrm{T}} \in \mathcal{H}$，2.4 節の基底 $\mathbf{u}, \mathbf{v}, \mathbf{w}$, $s, \bar{s} \in \{0, 1\}$ s.t. $w_s u_s \geq 0$, $s \neq \bar{s}$,
Output: $\mathbf{p}^{(i+1)} = (p_0^{(i+1)}, p_1^{(i+1)}, p_2^{(i+1)})^{\mathrm{T}} \in \mathcal{H}$

$\mathbf{q} \leftarrow \mathbf{p}^{(i)}$ /* $(q_0, q_1, q_2) \leftarrow (p_0^{(i)}, p_1^{(i)}, p_2^{(i)})$ */
while true **do**
  $\mathbf{r} \leftarrow \mathbf{q}$
  if $-I \leq q_{\bar{s}} + u_{\bar{s}} < I$ **then**
    $\mathbf{q} \leftarrow \mathbf{q} + \mathbf{u}$
  else if $-I \leq q_{\bar{s}} + w_{\bar{s}} < I$ **then**
    $\mathbf{q} \leftarrow \mathbf{q} + \mathbf{w}$
  else
    $\mathbf{q} \leftarrow \mathbf{q} + \mathbf{u} + \mathbf{w}$
  if $w_s > 0$ **then**
    if $q_s < -I$ **then continue**
    else if $q_s < I$ **then return q**
  else
    if $q_s \geq I$ **then continue**
    else if $q_s \geq -I$ **then return q**
  $\mathbf{q} \leftarrow \mathbf{r} + \mathbf{v}$
  repeat $\mathbf{q} \leftarrow \mathbf{q} + \mathbf{u}$ **until** $-I \leq q_{\bar{s}} < I$
  if $-I \leq q_s < I$ **then return q**

そして実際に，約 60%の基底に対して条件を満たすような基底が生成でき，ほぼ全ての格子点を計算できることを実験により確かめた．また，条件を満たさないような基底に関しても約 70%の格子点を計算することができた．しかし一方で，条件を満たした基底を用いて格子点計算を行うことで $\mathcal{H}$ に含まれる全ての格子点を計算できるという理論的証明はなされていなかった．

3 3 次元 Franke-Kleinjung 法の網羅的巡回性の証明

本節では，2.4 節の条件を基底 $\mathbf{u}, \mathbf{v}, \mathbf{w}$ を用いることで 2.4 節の 2 つの利点を備えた格子点計算ができることを証明する．

3.1 基底 $\mathbf{u}, \mathbf{v}, \mathbf{w}$ の方向や距離の証明

2.4 節の条件 A1 及び A2 から以下の定理が成り立つ．

定理 3.1 *3* つの基底 $\mathbf{u}=(u_0,u_1,u_2)^{\mathrm{T}}$，$\mathbf{v}=(v_0,v_1,v_2)^{\mathrm{T}}$，$\mathbf{w}=(w_0,w_1,w_2)^{\mathrm{T}}$ は *2.4* 節の条件 *A1* 及び *A2* を満たすとする．このとき，次を満たすような $\mathbf{u},\mathbf{v},\mathbf{w}$ の並べ替え $\mathbf{x},\mathbf{y},\mathbf{z}$ が存在する．

$$(x_0y_0 \le 0) \wedge (x_1y_1 \le 0) \wedge (x_0y_0 + x_1y_1 \ne 0) \tag{1}$$

であること．また，

$$((z_sx_s < 0) \wedge (z_{\bar{s}}x_{\bar{s}} \ge 0)) \wedge ((z_sy_s \ge 0) \wedge (z_{\bar{s}}y_{\bar{s}} < 0)), \tag{2}$$

$$(|z_s - x_s| \ge I) \wedge (|z_{\bar{s}} - y_{\bar{s}}| \ge I) \tag{3}$$

であるような $s,\bar{s} \in \{0,1\}, s \ne \bar{s}$ が存在すること．

証明 3.1 始めに，基底 $\mathbf{u},\mathbf{v},\mathbf{w}$ のうち，$(x_0y_0 > 0) \wedge (x_1y_1 > 0)$ を満たす $\mathbf{x},\mathbf{y} \in \{\mathbf{u},\mathbf{v},\mathbf{w}\}$ が存在すると仮定する．このとき，条件 *A2* を満たすためには x_0,x_1,y_0,y_1 のいずれかが I 以上である必要がある．これは条件 *A1* に矛盾するため，任意の $\mathbf{x},\mathbf{y} \in \{\mathbf{u},\mathbf{v},\mathbf{w}\}$ $(\mathbf{x} \ne \mathbf{y})$ に対して

$$(x_0y_0 < 0) \vee (x_1y_1 < 0) \tag{4}$$

である．

式 *(4)* より，ある $\mathbf{x},\mathbf{y} \in \{\mathbf{u},\mathbf{v},\mathbf{w}\}$ $(\mathbf{x} \ne \mathbf{y})$，$s,\bar{s} \in \{0,1\}$ $(s \ne \bar{s})$ が存在して $x_sy_s < 0$ である．ここで，$x_{\bar{s}}y_{\bar{s}} \le 0$ のとき式 *(1)* を満たす．では以降では $x_{\bar{s}}y_{\bar{s}} > 0$ の場合を考える．$\mathbf{z} \in \{\mathbf{u},\mathbf{v},\mathbf{w}\}$ を $\mathbf{x}$ でも $\mathbf{y}$ でもない列ベクトルとする．始めに $y_sz_s = 0$ の場合，$x_sy_s < 0$ であるから $x_sz_s = y_sz_s = 0$ である．よって式 *(4)* より $(x_{\bar{s}}z_{\bar{s}} < 0) \wedge (y_{\bar{s}}z_{\bar{s}} < 0)$ である．次に $y_sz_s < 0$ の場合，$x_sy_s < 0$ であるから $x_sz_s > 0$ である．よって式 *(4)* より $x_{\bar{s}}z_{\bar{s}} < 0$ であるから，$x_{\bar{s}}y_{\bar{s}} > 0$ より $y_{\bar{s}}z_{\bar{s}} \le 0$ である．最後に $y_sz_s > 0$ の場合，$x_sy_s < 0$ であるから $x_sz_s < 0$ である．また，式 *(4)* より $y_{\bar{s}}z_{\bar{s}} < 0$ である．よって $x_{\bar{s}}y_{\bar{s}} > 0$ より $x_{\bar{s}}z_{\bar{s}} \le 0$ である．以上より，どの場合でも $(x_0y_0 \le 0) \wedge (x_1y_1 \le 0)$ を満たすような $\mathbf{x},\mathbf{y}$ が存在する．このことに加え，x_0y_0 と x_1y_1 もまた条件 *A1* 及び *A2* に対して矛盾が生じるため同時に 0 にならない．したがって，式 *(1)* を満たすような $\mathbf{x},\mathbf{y}$ が存在することが証明された．

次に， 式 *(1)* を満たすような $\mathbf{x},\mathbf{y}$ ではない $\mathbf{z}$ に対して，式 *(4)* よりある s が存在して $z_{\bar{s}}x_{\bar{s}} < 0$ である．このとき式 *(1)* の $x_sy_s \le 0$ より，$z_sy_s \ge 0$ である．よって式 *(4)* より $z_{\bar{s}}y_{\bar{s}} < 0$ であるから，$x_{\bar{s}}y_{\bar{s}} \le 0$ より $z_{\bar{s}}x_{\bar{s}} \ge 0$ である．これにより式 *(2)* が証明された．さらに，式 *(2)* より $\mathbf{z},\mathbf{x}$ の第 *1* 成分あるいは第 *2* 成分のうち，*2.4* 節の条件 *A2* を満たすことができるのは異符号の関係にある z_s,x_s だけである．よって $|z_s - x_s| \ge I$ である．また $\mathbf{z},\mathbf{y}$ の場合も同様であるため，これにより式 *(3)* が証明された．□

本稿では，式 (1) を満たすような列ベクトルのうち，第 1 成分がより大きい列ベクトルを $\mathbf{u}$，もう一方を $\mathbf{v}$ とし，残りの列ベクトルを $\mathbf{w}$ とする．

3.2 基底 $\mathbf{u},\mathbf{v},\mathbf{w}$ の方向や距離の証明

次に，2.4 節の条件 A1，A2，A3 を満たすような基底 $\mathbf{u},\mathbf{v},\mathbf{w}$ は，定理 3.2 の条件 B1 を満たすことを示す．

定理 3.2 *3* つの基底 $\mathbf{u}=(u_0,u_1,u_2)^{\mathrm{T}}$，$\mathbf{v}=(v_0,v_1,v_2)^{\mathrm{T}}$，$\mathbf{w}=(w_0,w_1,w_2)^{\mathrm{T}}$ は *2.4* 節の条件 *A1*，*A2*，*A3* を満たすとする．このとき，次を条件を満たす．

B1. $i,j,k \in \mathbb{Z}$ に対して，$|iu_0 + jv_0 + kw_0| < I$ かつ $|iu_1 + jv_1 + kw_1| < I$ であるとき，$i,j,k \ge 0$ あるいは $i,j,k \le 0$ である．

証明 3.2 基底 $\mathbf{u},\mathbf{v},\mathbf{w}$ は条件 *A1* 及び *A2* を満たすため，定理 *3.1* より $\mathbf{u},\mathbf{v}$ は式 *(1)* を，$\mathbf{w}$ は式 *(2)* 及び式 *(3)* を満たす．

条件 *B1* の待遇を証明する．始めに，i,j,k のうち *1* つが 0 である場合について述べる．式 *(4)* から $\exists s \in \{0,1\}$ *s.t.* $u_sv_s < 0$ である．$k=0, i<0, j>0$ 及び $k=0, i>0, j<0$ のとき，iu_s と jv_s は同符号であるため，条件 *A2* から，$|iu_s + jv_s| = |iu_s| + |jv_s| \ge I$ である．$j=0, i=0$ の場合も同様に証明できる．

次に，i, j, k がいずれも 0 でない場合について述べる．まず $i < 0, j > 0, k > 0$ の場合，式 *(2)* から $\exists s \in \{0, 1\}$ *s.t.* $u_s w_s < 0$ であり，式 *(1)* から $u_s v_s \leq 0$ である．このとき，$i < 0, j > 0, k > 0$ から iu_s, kw_s は同符号であり，iu_s, jv_s は $v_s = 0$ あるいは同符号である．よって条件 *A2* から，$|iu_s + jv_s + kw_s| = |iu_s| + |jv_s| + |kw_s| \geq I$ である．$i > 0, j < 0, k < 0$ の場合も同様である．また，$i > 0, j < 0, k > 0$ 及び $i < 0, j > 0, k < 0$ の場合も同様の手法で証明できる．最後に，$i > 0, j > 0, k < 0$ の場合について述べる．条件 *A3* から，任意の $i, j \in \mathbb{Z}_{>0}$ に対して $\exists s$ *s.t.* $|iu_s + jv_s - w_s| \geq I$ である．まず $w_0, w_1 < 0$ とすると，式 *(2)* より $u_0 > 0, u_1 \leq 0, v_0 \leq 0, v_1 > 0$ であり，また条件 *A1* より $u_1, v_0 > -I$ である．よって $iu_0 + jv_0 - w_0 > -I$ かつ $iu_1 + jv_1 - w_1 > -I$ である．このことから $w_0, w_1 < 0$ とすると，$\exists s$ *s.t.* $iu_s + jv_s - w_s \geq I$ であることがわかる．したがって $k \in \mathbb{Z}_{<0}$ に対して，$\exists s$ *s.t.* $iu_s + jv_s + kw_s \geq I$ であり，条件 *B1* の対偶を満たす．同様の手法で $w_0, w_1 > 0$ や $w_0 < 0 \wedge w_1 > 0$，$w_0 > 0 \wedge w_1 < 0$ の場合も証明できる．以上により，条件 *B1* の対偶を証明した．□

3.3　網羅的巡回性の証明

本節では，2.4 節の条件 A2 及び A3 を満たす場合，網羅的な格子点計算が可能であることを示す．以降では，基底 $\mathbf{u}, \mathbf{v}, \mathbf{w}$ の格子点に対して以下のような二項関係を定義する．

定義 3.1 $\mathbf{p}, \mathbf{q} \in \mathbb{Z}^3$ を基底 $\mathbf{u}, \mathbf{v}, \mathbf{w} \in \mathbb{Z}^3$ の格子点とする．よって $\mathbf{p} - \mathbf{q} = i\mathbf{u} + j\mathbf{v} + k\mathbf{w}$ となる $i, j, k \in \mathbb{Z}$ が存在する．このとき，$\mathbf{q} \preceq \mathbf{p} \Leftrightarrow i, j, k \geq 0$

上記の定義を用いて以下の定理の証明する．

定理 3.3 I, J を領域の閾値とする篩領域 $\mathcal{H}$ と，*3* つの基底 $\mathbf{u} = (u_0, u_1, u_2)^{\mathrm{T}}$, $\mathbf{v} = (v_0, v_1, v_2)^{\mathrm{T}}$, $\mathbf{w} = (w_0, w_1, w_2)^{\mathrm{T}}$ に対して定理 *3.2* の条件 *B1* を満たすとき，$\mathbf{u}, \mathbf{v}, \mathbf{w}$ を加算することにより $\mathcal{H}$ に含まれる $\mathbf{u}, \mathbf{v}, \mathbf{w}$ の全ての格子点を逐次的に計算できるような始点 $\mathbf{p}^{(0)} \in \mathcal{H}$ が存在する．

証明 3.3 篩領域 $\mathcal{H}$ に含まれる $\mathbf{u}, \mathbf{v}, \mathbf{w}$ の全ての格子点の集合を S とする．このとき，任意の格子点 $\mathbf{p}, \mathbf{q} \in S$ に対して，$\mathbf{p}$ と $\mathbf{q}$ は $\mathbf{u}, \mathbf{v}, \mathbf{w}$ の格子点であるため，ある $i, j, k \in \mathbb{Z}$ が存在して $\mathbf{p} - \mathbf{q} = i\mathbf{u} + j\mathbf{v} + k\mathbf{w}$ と表すことが出来る．また，$\mathbf{p}$ 及び $\mathbf{q}$ は $\mathcal{H}$ に含まれているため，$\mathbf{p}$ と $\mathbf{q}$ の第 *1* 成分と第 *2* 成分の距離は I よりも小さい．すなわち，$|iu_0 + jv_0 + kw_0| < I$ かつ $|iu_1 + jv_1 + kw_1| < I$ である．よって上記の条件 *B1* より，任意の $\mathbf{p}, \mathbf{q} \in S$ に対する i, j, k は $i, j, k \geq 0$ あるいは $i, j, k \leq 0$ を満たす．

以上から，$(S, \preceq)$ は全順序集合となる．よって，S に対して $\mathbf{p}^{(n)} \preceq \mathbf{p}^{(n+1)}$ を満たすような S の要素の列 $(\mathbf{p}^{(n)}), n = 1, 2, \ldots, \sharp S$ が唯一存在し，格子点 $\mathbf{p}^{(0)}$ に繰り返し $\mathbf{u}, \mathbf{v}, \mathbf{w}$ を加算することで篩領域 $\mathcal{H}$ に含まれる全ての格子点 $(\mathbf{p}^{(n)})$ を順に計算できる．□

2.4 節の条件 A4 から，$\mathbf{u}, \mathbf{v}, \mathbf{w}$ の加算によって計算される格子点 $\mathbf{p}^{(n)}$ の第 3 成分 $p_2^{(n)}$ は減少することはない．よって，$p_2^{(n)}$ が J 以上であるとき格子点計算を終了する．また，格子点計算を行うためには始点となる $\mathbf{p}^{(0)}$ が必要であるが，$u_2, v_2, w_2 > 0$ であるときは $\mathbf{p}^{(0)} = (0, 0, 0)^{\mathrm{T}}$ である．なぜならば，$u_2, v_2, w_2 > 0$ である場合，$(0, 0, 0)^{\mathrm{T}}$ からどの基底を減算しても $p_2 < 0$ となり $\mathcal{H}$ に含まれないためである．

一方で，u_2, v_2, w_2 に 0 が存在する場合，$\mathbf{p}^{(0)} \prec (0, 0, 0)^{\mathrm{T}}$ となるような $\mathbf{p}^{(0)}$ が存在する可能性がある．しかし，そのような $\mathbf{p}^{(0)}$ が存在した場合，$-\mathbf{p}^{(0)}$ は $(0, 0, 0)^{\mathrm{T}}$ を始点とした格子点計算によって計算されるため，hit tuple としては重複となる．よって，2.4 節の条件 A4 の場合でも $\mathbf{p}^{(0)} = (0, 0, 0)^{\mathrm{T}}$ として格子点計算を行う．ただし，$\mathbf{u}, \mathbf{v}, \mathbf{w}$ の第 1 成分あるいは第 2 成分が I の約数の場合，$-\mathbf{p}^{(0)} \notin \mathcal{H}$ かつ $\mathbf{p}^{(0)} \in \mathcal{H}$ となる $\mathbf{p}^{(0)}$ が存在する場合があるため，例外として計算する必要がある．

以上により，2.4 節の条件 A1，A2，A3，A4 を満たすような基底を用いて，Algorithm 1 に示す格子点計算を行うことにより，2.4 節の 2 つの利点を備えた格子点計算を行うことができる．

4 基底の具体例

本節では，2.4 節の条件を満たす基底の具体例を記す．以降の基底変換や格子点生成は，OS が Linux OS (64 ビット)，プログラミング言語が C++，コンパイラが g++ 4.7.2 である PC にて行った．

拡大体として，標数 $p = 1081034284409$ と 40 ビットの素数とし，拡大次数 $n = 6$ とした．このとき拡大体の位数は 240 ビットとなる．

$$p^6 = 1596014400197077740306072399677175 6 \backslash \\ 920259173527154533440361770633521450 41$$

多項式選択で選択される 2 つの多項式は

$$f_1(X) = x^6 - 2x^5 + x^3 - x + 2,$$
$$f_2(X) = x^6 - 2x^5 + x^3 - x + 1081034284411.$$

を使用した．また，$\mathbf{c}$-空間の篩領域 $\mathcal{H}$ の閾値は $I = 256, J = 128$ とした．

始めに，基底 $\mathbf{u}, \mathbf{v}, \mathbf{w}$ の第 3 成分が全て 0 より大きい場合について述べる．special-$\mathfrak{q}$ として $\mathfrak{q} = (6532291, X+1470092)$，$\mathfrak{r} = (751691, X+268635)$ とした．このとき，$M^{\mathrm{FK}}_{\mathfrak{q},\mathfrak{r}}$ は，

$$M^{\mathrm{FK}}_{\mathfrak{q},\mathfrak{r}} = \begin{pmatrix} 230 & -6 & -35 \\ -192 & 235 & -42 \\ 27 & 19 & 4 \end{pmatrix}.$$

であった．この基底は 2.4 節の 4 つの条件を満たしており， Algorithm 1 を用いて網羅的かつ単調非減少な格子点計算を行うことができる．以下は計算される格子点を順に記したものである．

```
#1:(0, 0, 0)
  step 7: q + w = (-35, -42, 4)
#2:(-35, -42, 4)
  step 7: q + w = (-70, -84, 8)
#3:(-70, -84, 8)
  step 7: q + w = (-105, -126, 12)
#4:(-105, -126, 12)
  step 5: q + u = (125, -318, 39)
  step 7: q + w = (-140, -168, 16)
  step 9: q + u + w = (90, -360, 43)
  step 16: q + v = (-111, 109, 31)
#5:(-111, 109, 31)
  step 5: q + u = (119, -83, 58)
#6:(119, -83, 58)
    ...
#13:(-97, -100, 101)
```

次に，$\mathbf{u}, \mathbf{v}, \mathbf{w}$ の第 3 成分に 0 が存在する場合について述べる．例としては以下のような $M^{\mathrm{FK}}_{\mathfrak{q},\mathfrak{r}}$ である．

$$M^{\mathrm{FK}}_{\mathfrak{q},\mathfrak{r}} = \begin{pmatrix} 64 & -205 & -202 \\ -39 & 247 & -240 \\ 0 & 1 & 1 \end{pmatrix}.$$

列ベクトル $\mathbf{u} = (64, -39, 0)^{\mathrm{T}}$ に注目すると，$\mathbf{u} \in \mathcal{H}$ である一方で $-\mathbf{u} \in \mathcal{H}$ でもある．しかし，$\mathbf{u}$ と $-\mathbf{u}$ は定数倍の関係であり，格子篩としては一方は重複な hit tuple となる．よって $-\mathbf{u}$ は無視して問題ないため，格子点計算を行う始点は $\mathbf{p}^{(0)} = (0, 0, 0)^{\mathrm{T}}$ としてよい．ただし，第 3 成分が 0 で第 2 成分あるいは第 1 成分のうちより大きい成分が $I/2$ を割り切りかつ正の場合は注意が必要である．具体的には，次のような場合である．

$$M^{\mathrm{FK}}_{\mathfrak{q},\mathfrak{r}} = \begin{pmatrix} 128 & -148 & -227 \\ -115 & 180 & -241 \\ 0 & 1 & 2 \end{pmatrix}.$$

この例では，$\mathbf{u} = (128, -115, 0)^{\mathrm{T}}$ は $\mathcal{H}$ に含まれないが，$-\mathbf{u} = (-120, 115, 0)^{\mathrm{T}}$ は $\mathcal{H}$ に含まれるため，例外な格子点として出力する必要がある．

5 まとめ

本稿では，3 次元 Franke-Kleinjung 法における条件を満たした基底を用いた場合，3 次元領域内の全ての格子点が計算できることを証明した．始めに，条件を満たした基底の向きや距離について証明を行った．次に条件を満たした基底と領域に含まれる格子点に対して順序を定義した．最後に基底を加算することにより逐次的に格子点を計算できること，領域の 1 つの次元

に対して単調非減少な格子点計算ができることを示した．

今後は，3 次元 Franke-Kleinjung 法により網羅的に格子点計算が行える基底はどれほど存在するのかについて理論的側面から明らかにすることである．また，3 次元 Franke-Kleinjung 法により網羅的に格子点計算が行えない基底に特化した格子点計算法を提案することである．

参考文献

[1] P.S.L.M. Barreto and M. Naehrig, "Pairing-friendly elliptic curves of prime order," SAC 2005, Lecture Notes in Comput. Sci., vol.3897, pp.319-331, Springer, 2006.

[2] J. Franke and T. Kleinjung, "Continued fractions and lattice sieve," Workshop record of SHARCS, 2005, `http://www.ruhr-uni-bochum.de/itsc/tanja/SHARCS/talks/FrankeKleinjung.pdf`.

[3] A. Joux and R. Lercier, "Improvements to the general number field sieve for discrete logarithms in prime fields. A comparison with the Gaussian integer method," Math. Comp., vol.72, pp.953-967, 2003.

[4] A. Joux, R. Lercier, N.P. Smart and F. Vercauteren, "The number field sieve in the medium prime case," CRYPTO '06, Lecture Notes in Comput. Sci., vol.4117, pp.326-344, Springer-Verlag, 2006.

[5] T. Kleinjung, K. Aoki, J. Franke, A.K. Lenstra, E. Thomé, J.W. Bos, P. Gaudry, A. Kruppa, P.L. Montgomery, D.A. Osvik, H.J.J. te Riele, A. Timofeev and P. Zimmermann, "Factorization of a 768-bit RSA modulus," CRYPTO '10, Lecture Notes in Comput. Sci., vol.6223, pp.333-350, Springer-Verlag, 2010.

[6] 早坂健一郎, 青木和麻呂, 小林鉄太郎, 高木剛, "GF(p^{12}) 上の離散対数問題に対する数体篩法の計算機実験," 2013 年暗号と情報セキュリティシンポジウム, SCIS2013, 4A1-3, 2013.

[7] 早坂健一郎, 青木和麻呂, 小林鉄太郎, 高木剛, "拡大体 GF(p^n) 上の数体篩法における 3 次元 Lattice Sieve の構成," コンピュータセキュリティシンポジウム CSS2013, 1C1-3, 2013.

[8] K. Hayasaka, K. Aoki, T. Kobayashi, T. Takagi, "An Experiment of Number Field Sieve for Discrete Logarithm Problem over GF(p^{12})," Number Theory and Cryptography 2013, Buchmann Festschrift, LNCS 8260, pp.108-120, 2013.

[9] K. Hayasaka, K. Aoki, T. Kobayashi, T. Takagi, "An Experiment of Number Field Sieve for Discrete Logarithm Problem over GF(p^{12})," JSIAM Letters, to appear.

[10] A.K. Lenstra and H.W. Lenstra, *The Development of the Number Field Sieve,* Lecture Notes in Math., vol.1554, Springer-Verlag, 1993.

[11] A. Miyaji, M. Nakabayashi and S. Takano, "New explicit conditions of elliptic curve traces for FR-reduction," In IEICE Trans. on Fund., E84-A, vol.5, pp.1234-1243. 2001.

[12] J.M. Pollard, "The lattice sieve," pp.43-49, in [10].

[13] F. Vercauteren, "Optimal pairings," IEEE Transactions on Information Theory, vol.56, pp.455-461, 2010.

[14] P. Zajac, "On the use of the lattice sieve in the 3D NFS," Tatra Mt. Math. Publ. vol.45, pp.161-172, 2010.