数理解析研究所講究録
第1568巻 2007年 135-141

$\mathbb{K}[x]$上Smith標準形の変換行列の最小化

大倉 安孝
Ookura Yasutaka *
筑波大学 数理物質科学研究科
Graduate School of Pure and Applied Sciencces, University of Tsukuba

Abstract

Smith標準形は, PID(principal ideal domain)上で定義される対角行列で, 整数論や群論での計算において重要である. PID上の任意の行列をSmith標準形に変換することができるが, 本論文では一変数多項式環上において, 変換行列のノルムを最小化する手順を示す. 最小化は格子算法で行なうが, 格子算法による縮小基底とPID上加群の極小Gröbner基底が同一であることを示す.

1 はじめに

Smith標準形はPID(principal ideal domein)上で定義される対角行列で, 整数論や群論での計算において重要である. PID上の任意の行列をSmith標準形に変換することができる. ただし, Smith標準形は一意であるが, 変換行列は一意でない. 応用上の扱い易さを考えれば, 変換行列のノルムは小さい方が望ましい. Smith標準形とその変換行列を求める最初の多項式時間アルゴリズムは[KB79]($\mathbf{Z}$上)と, [Kan85]($\mathbb{K}[x]$上)で与えられた. その後, 計算速度の向上や大規模行列の計算のために様々な改良が研究された. 変換行列に関しては, $\mathbf{Z}$上の場合には格子算法によってノルムの小さい変換行列を求める研究がなされているが[Jag05], $\mathbb{K}[x]$上の場合には同様の研究はない. 本論文では$\mathbf{Z}$上でノルムの小さい変換行列を求める方法を$\mathbb{K}[x]$上に拡張する. 変換行列のノルムを下げるために$\mathbb{K}[x]$上の格子算法を使う. これは, $\mathbf{Z}$上格子の短ベクトルと縮小基底をもとめるLLLアルゴリズム[LLL82]を拡張した算法で[Len85], この場合には最短ベクトルとノルム最小の縮小基底が多項式時間で計算できる. 本論では, 整数の場合と同様に変換行列の要素が張る格子に注目することで, 変換行列のノルムを小さくする方法を示す. しかし, 整数の場合と違い, ノルムの最小化が可能である. また, 関連する算法として, 縮小基底とPID上加群の極小Gröbner基底が定数倍を除いて同一であることを示す.

2 Smith標準形

2.1 Smith標準形の定義

Smith標準形はPID(principal ideal domain)上のある種の対角行列で, PID上の任意の行列をSmith標準形に変換できることが1861年にSmithによって示された[Smi61]. AはPID上の(m,n)行列, ただし

*yasutaka@math.tsukuba.ac.jp

$m \leq n$, でランクが r とする. A の Smith 標準形は

$$S = UAV = \begin{pmatrix} d_1 & & & & & 0 \\ & \ddots & & & & \\ & & d_r & & & \\ & & & 0 & & \\ 0 & & & & \ddots & \\ & & & & \cdots & 0 \end{pmatrix}$$

である. ここで, $d_1, \cdots, d_r$ は $d_i | d_j$ $(1 \leq i < j \leq r)$ を満たす. 変換行列 U と V はそれぞれ (m, m) 行列, (n, n) 行列であり, ユニモジュラである. (ユニモジュラ行列というのは, その行列式の値が $\mathbf{Z}$ 上であれば ± 1, $\mathbb{K}[x]$ 上であれば $\mathbb{K}$ の元, となる正方行列である.) ここで, $\mathbb{K}$ は任意の体である.

2.2 変換行列のノルム

ベクトルと行列のノルムにはいくつかの定義があるが, 本論ではノルムの最小化 (縮小) が可能なようにノルムを定義する. 以下, $U = (u_1, \cdots, u_n)$ とし, U のノルムを $|U|$ と表す.

定義 2.1. $\mathbf{Z}$ 要素の場合

$|u_i|$: ユークリッドノルム

$|U|$: $|u_1| + \cdots + |u_n|$

定義 2.2. $\mathbb{K}[x]$ 要素の場合

$u_i = (u_{i1}, \cdots, u_{in}) \in \mathbb{K}[x]^n$ とする.

$|u_{ij}|$: 多項式 u_{ij} の次数

$|u_i|$: $\max\{|u_{i1}|, \cdots, |u_{in}|\}$

$|U|$: $|u_1| + \cdots + |u_n|$

Smith 標準形 S は一意であるが, 変換行列 U と V は一意でない. よって, 応用上の扱い易さを考えれば, U と V はノルムが小さい方が望ましい.

3 格子算法

3.1 格子の定義と格子に関する問題

定義 3.1. $\mathbf{Z}$ を整数環, $\{f_1, \cdots, f_n\}$ を互いに独立な $\mathbf{Z}$ 上 n 次元ベクトルとする. このとき $L = \sum_{i=1}^{n} \mathbf{Z} f_i$ を基底 $\{f_1, \cdots, f_n\}$ によって生成される格子, もしくは $\mathbf{Z}$-*module* という. ■

$f_i = (f_{i1}, \cdots, f_{in})$ を格子点とみなせば L は無限個の格子点の集合である. (注意 : 格子は元々は $f_1, \cdots, f_n$ を実数体 $\mathbb{R}$ 上のベクトルとして定義されたが, $\mathbf{Z}$ と考えた方が分かり易く, 本論文ではそれで十分である.) 一組の基底によって格子 L が生成されたとき, 最短ベクトルの探索は NP 困難と証明されている [Mic01]. しかし比較的短い「短ベクトル」は多項式時間で求まり, これは縮小基底を求めることに帰着する (LLL アルゴリズム, [LLL82]).

本論文では, 一変数多項式 $\mathbb{K}[x]$ 上の n 次ベクトル $f_1, \cdots, f_n$ を基底とする $\mathbb{K}[x]$ 上の格子 $L = \sum_{i=1}^{n} \mathbb{K}[x] f_i$ を扱う. $\mathbb{K}[x]$ 上の格子の場合, 格子に含まれる最短ベクトルの探索が多項式時間で可能である. 最短ベクトルの探索は, $\mathbf{Z}$ 上格子の場合と同様に縮小基底を求めることに帰着される. よって, 次に $\mathbb{K}[x]$ 上格子の縮小基底について述べる.

3.2 $\mathbb{K}[x]$ 上格子の縮小基底

ベクトル $f = (f_1, \cdots, f_n) \in \mathbb{K}[x]^n$ を考える.

定義 3.2. $|f_i|$ を多項式 f_i の次数として, ベクトル $\boldsymbol{f}$ のノルムを次式で定義する.

$$|\boldsymbol{f}| = \max\{|f_1|, \cdots, |f_n|\}.$$

■

定義 3.3. $B = \{\boldsymbol{f_1}, \cdots, \boldsymbol{f_n}\}$ を $\mathbb{K}[x]$ 上格子の基底として, 基底のノルムを次式で定義する.

$$|B| = |\boldsymbol{f_1}| + \cdots + |\boldsymbol{f_n}|$$

■

[Len85] はこの定義により, 基底 $\{\boldsymbol{f_1}, \cdots, \boldsymbol{f_n}\}$ で張られる格子の最短ベクトルと, 縮小基底を多項式時間で求めるアルゴリズム ($\mathbb{K}[x]$ 上格子算法) を示した. (実際は係数を有限体 $\mathbb{F}_q$ に限定したアルゴリズムであるが, 任意の体 $\mathbb{K}$ としても機能する [Gat84].)

まず縮小基底の定義を述べる. 格子 L の基底のベクトル $\boldsymbol{f_1}, \cdots, \boldsymbol{f_n}$ を各行にもつ行列 M を考え, M の列交換演算 φ によって得られる行を $\hat{\boldsymbol{f}}_1, \cdots, \hat{\boldsymbol{f}}_n$ とする:

$$M = \begin{pmatrix} \boldsymbol{f_1} \\ \vdots \\ \boldsymbol{f_n} \end{pmatrix} \quad \stackrel{\varphi}{\Longrightarrow} \quad \hat{M} = \begin{pmatrix} \hat{\boldsymbol{f}}_1 \\ \vdots \\ \hat{\boldsymbol{f}}_n \end{pmatrix}. \tag{1}$$

定義 3.4. ベクトル $\hat{\boldsymbol{f}}_i$ $(1 \le i \le n)$ を $(\hat{f}_{i1}, \cdots, \hat{f}_{in})$ と表す. いま, $\hat{\boldsymbol{f}}_1, \cdots, \hat{\boldsymbol{f}}_n$ が次の条件を満たすとき, $\{\boldsymbol{f_1}, \cdots, \boldsymbol{f_n}\}$ を縮小基底という.

$$|\hat{f}_{ij}| < |\hat{f}_{ii}| \quad \text{for} \quad 1 \le j < i \le n, \tag{2}$$

$$|\hat{f}_{ij}| \le |\hat{f}_{ii}| \quad \text{for} \quad 1 \le i < j \le n, \tag{3}$$

$$|\hat{\boldsymbol{f}}_1| \le |\hat{\boldsymbol{f}}_2| \le \cdots \le |\hat{\boldsymbol{f}}_n|. \tag{4}$$

(注意 : (2), (3) から行列 $\hat{M}$ の対角成分はそれぞれの行のなかでノルムが最大の成分であることが分かる.)

■

L の縮小基底 $\{\boldsymbol{f_1}, \cdots, \boldsymbol{f_n}\}$ は以下の性質をもつことが示されている [Len85].

- $\boldsymbol{f_1}$ は L の最短ベクトルである,
- 縮小基底はノルムが最小の基底である,

よって, 縮小基底が求まれば格子の最短ベクトルが決定できる. 縮小基底の最小性を使うことで, 本論文の主題である Smith 標準形の変換行列の最小化が実現できる. 縮小基底と PID 上加群の極小 Gröbner 基底が同一であることからも最小性は証明できるが, これは次節で示す.

3.3 基底縮小アルゴリズム -$\mathbb{K}[x]$ 上格子算法 -

アルゴリズムは n 個のベクトルの集合 $\{\boldsymbol{f_1}, \cdots, \boldsymbol{f_n}\}$ を入力として, 縮小基底を与える n 個のベクトル集合 $\{\hat{\boldsymbol{f}}_1, \cdots, \hat{\boldsymbol{f}}_n\}$ と, 縮小基底からの変換 φ を出力する. アルゴリズムは n 個のステップで構成される. 各ステップの計算は同じで, 第 k ステップ $(1 \le k \le n)$ では

$$|\hat{f}_{ij}| < |\hat{f}_{ii}| \quad \text{for} \quad 1 \le j < i \le k, \tag{5}$$

$$|\hat{f}_{ij}| \le |\hat{f}_{ii}| \quad \text{for} \quad 1 \le i \le k, \quad i < j \le n, \tag{6}$$

$$|\hat{\boldsymbol{f}}_1| \le \cdots \le |\hat{\boldsymbol{f}}_k|, \tag{7}$$

$$|\hat{\boldsymbol{f}}_k| \le |\hat{\boldsymbol{f}}_j| \quad \text{for} \quad k < j \le n, \tag{8}$$

が満たされるように計算が行われる. つまり, $k = n$ のとき定義 3.4 が満たされ, 正常に終了する. 最後に φ^{-1} を作用させることで, 縮小基底を求めることができる.

4 縮小基底とPID上加群の極小Gröbner基底の同一性

$\mathbb{K}[x]$ 上格子の縮小基底がノルム-位置順序に対するPID上加群の極小Gröbner基底と定数倍を除いて同一であることを示す. $\mathbb{K}[x]^n$ 上のGröbner基底は次式で定義されるS-ベクトルを計算することで求まる.

定義 4.1 (S-ベクトル). $f_1, f_2 \in \mathbb{K}[x]^n$ のS-ベクトル $S(f_1, f_2)$ は以下である.

- $\mathrm{LP}(f_1) = \mathrm{LP}(f_2)$ の場合

$$S(f_1, f_2) = \frac{\mathrm{LCM}(\mathrm{LM}(f_1), \mathrm{LM}(f_2))}{\mathrm{LT}(f_1)} f_1 - \frac{\mathrm{LCM}(\mathrm{LM}(f_1), \mathrm{LM}(f_2))}{\mathrm{LT}(f_2)} f_2$$

- $\mathrm{LP}(f_1) \neq \mathrm{LP}(f_2)$ の場合

$$S(f_1, f_2) = 0$$

■

ただし, ベクトル f に対して $\mathrm{LP}(f)$ は f の最高次項を与える要素の位置とする. $F = \{f_1, \cdots, f_n\}$ が縮小基底であるとする. このとき, 縮小基底の定義から, $\mathrm{LP}(f_i) \neq \mathrm{LP}(f_j)$ $(\forall i \neq j)$ である. よって F に含まれる任意の二つのS-ベクトルは0である. また, 任意の $f_i \in F$ に対して $\mathrm{LM}(f_i)$ は $\mathrm{LM}(F \backslash \{f_i\})$ のどの元でも割れない. これより, 次の命題が成り立つ.

命題 4.2. (大倉-小副川-佐々木) 格子算法による縮小基底は, ノルム-位置順序に対するPID上加群の極小Gröbner基底と定数倍を除いて一意である. ■

縮小基底の元は, 最高次係数が1に規格化されていないので, 極小Gröbner基底の元とは定数 $(\in \mathbb{K})$ 倍の違いがある. 縮小基底の最小性は [Len85] で示されているが, 極小Gröbner基底であることを用いれば, より簡潔に最小性がいえる. 格子算法は多項式時間で終了することが保証されている一方でGröbner基底による既存の表現は構造を理解するうえで明快なので, 両者が算法として等しいことを確認するのは重要である.

5 Smith標準形の変換行列の最小化

行列 $A \in \mathbb{K}[x]^{m,n}$ のSmith標準形 S と, 変換行列 U, V が与えられたとする. このとき, ノルムの合計が最小となる変換行列 U^* と V^* を求める手順を示す.

5.1 行列 W と Z

補題 5.1. W と Z を下記のようにブロック化された行列とするとき, $U^* = WU, V^* = VZ$ もふたたび同一のSmith標準形の変換行列である.

$$W = \begin{pmatrix} W_{11} & * \\ 0 & W_{22} \end{pmatrix} \in \mathbb{K}[x]^{m,m}, \qquad Z = \begin{pmatrix} Z_{11} & 0 \\ * & Z_{22} \end{pmatrix} \in \mathbb{K}[x]^{n,n}.$$

ここで, W_{11} と Z_{11} は (r, r) 行列で, $W_{11} D Z_{11} = D$ を満たし, $*$ は $\mathbb{K}[x]$ 上の任意の行列を表す. また, $\det(W_{ii}), \det(Z_{ii}) \in \mathbb{K}, (i = 1, 2)$ である. ■

行列 Z によって得られる新しい変換行列 V^* について述べる. なお, Z による変換と W による変換は転置すればまったく同一であるので, 以下では Z に関して説明する. Smith標準形の変換行列を $V = (v_1, \cdots, v_n)$ とする. $V^* = VZ = (v_1^*, \cdots, v_n^*)$ は次のような構造をしている.

$$\begin{cases} v_1^* = \sum_{i=1}^{r}(Z_{11})_{i1}v_i + \sum_{i=r+1}^{n}(*)_{i-r,1}v_i, \\ \qquad\vdots \qquad\qquad\qquad \vdots \\ v_r^* = \sum_{i=1}^{r}(Z_{11})_{ir}v_i + \sum_{i=r+1}^{n}(*)_{i-r,n-r}v_i, \\ v_{r+1}^* = \sum_{i=r+1}^{n}(Z_{22})_{i-r,1}v_i, \\ \qquad\qquad\vdots \\ v_n^* = \sum_{i=r+1}^{n}(Z_{22})_{i-r,n-r}v_i. \end{cases} \tag{9}$$

補題 5.2. $\{v_{r+1}^*,\cdots,v_n^*\}$ を $\{v_{r+1},\cdots,v_n\}$ によって張られる格子の縮小基底とすれば, ノルム $|v_{r+1}^*|+\cdots+|v_n^*|$ を最小化できる.

次に $v_1^*,\cdots,v_r^*$ について考える. 関係式 (9) の一部を下に示す.

$$\begin{cases} v_1^* = \sum_{i=1}^{r}(Z_{11})_{i1}v_i + \sum_{i=r+1}^{n}(*)_{i-r,1}v_i, \\ \qquad\vdots \qquad\qquad\qquad \vdots \\ v_r^* = \sum_{i=1}^{r}(Z_{11})_{ir}v_i + \sum_{i=r+1}^{n}(*)_{i-r,n-r}v_i, \end{cases} \tag{10}$$

また (10) の右辺の第一項を次式のように書き直す.

$$\begin{cases} \hat{v}_1 = \sum_{i=1}^{r}(Z_{11})_{i1}v_i, \\ \vdots \qquad\qquad\qquad \vdots \\ \hat{v}_r = \sum_{i=1}^{r}(Z_{11})_{ir}v_i. \end{cases} \tag{11}$$

$\hat{v}_1,\cdots,\hat{v}_r$ を $v_{r+1},\cdots,v_n$ によって張られる格子によって簡約することで, 次がいえる.

補題 5.3. $\hat{v}_1,\cdots,\hat{v}_r$ が与えられたとき, $|v_i^*| \le |\hat{v}_i|$ $(1 \le i \le r)$ を満たすノルム最小のベクトル $v_1^*,\cdots,v_r^*$ を計算することができる.

最後に, $\hat{v}_1,\cdots,\hat{v}_r$ の定め方について述べる. 関係式 (11) の変換は行列 Z_{11} によるものであるが, $W_{11}DZ_{11}=D$ という制約条件があるので, W_{11} による U の変換結果 $\hat{u}_1,\cdots,\hat{u}_r$ とともに考える. $1 \le s < t \le r$ とし, $\hat{u}_s$ を考える. 右辺の和は $t=1$ から順に $t=r$ まで行うので, その一つの和である $\hat{u}_s = u_s + wu_t$, を考える. ただし $w \in \mathbb{K}[x]$ である. また, U の他の行の変換は考えない. このとき, 制約条件 $W_{11}DZ_{11}=D$ があるので V の列の計算 $\hat{v}_t = v_t - w\frac{d_t}{d_s}u_s = v_t - wu_s'$ $(u_s' = (d_t/d_s)u_s)$, も同時に行わなくてはならない. そこで, $\hat{u}_i$ $(i \ne s)$ と $\hat{v}_j$ $(j \ne t)$ を固定した上でノルム $|\hat{u}_s|+|\hat{v}_t|$ を最小化することを考える. これは次に述べるプロシジャReduceELM で行う. $t < s$ の場合も同様にする.

5.2 ReduceELM プロシジャ

$1 \le s < t \le r$ とする. このとき, $u_s, u_t, v_t, v_s \in \mathbb{K}[x]^n$ に対して, $\hat{u}_s = u_s + wu_t$, $\hat{v}_t = v_t - w\frac{d_t}{d_s}v_s = v_t - wv_s'$ という計算を考える. ただし $w \in \mathbb{K}[x]$ である. ReduceELM プロシジャは, $\hat{u}_i$ $(i \ne s)$ と $\hat{v}_j$ $(j \ne t)$ を固定した上でノルム $|\hat{u}_s|+|\hat{v}_t|$ を最小化する w を返す. ただし, ノルムの和 $|u_s|+|v_s|$ を下げることができない場合は 0 を返す. はじめに, $a_0 = |u_s|+|v_t|$ として下記の 3 つの場合に分ける.

(1) $\mathrm{LP}(u_s) \ne \mathrm{LP}(u_t)$ かつ $\mathrm{LP}(v_t) \ne \mathrm{LP}(v_s')$ の場合.

(2) $\mathrm{LP}(u_s) = \mathrm{LP}(u_t)$ あるいは $\mathrm{LP}(v_t) = \mathrm{LP}(v_s')$ の場合.

(3) $\mathrm{LP}(u_s)=\mathrm{LP}(u_t)$ かつ $\mathrm{LP}(v_t)=\mathrm{LP}(v'_s)$ の場合.

(1) の場合, ノルムの和 $|u_s|+|v_t|$ を下げることはできない. よって, $w=0$ を返す. (2) の場合, $\mathrm{LP}(u_s)=\mathrm{LP}(u_t)$ ならば, $w_1=-\mathrm{quo}(u_s,u_t)$, $\hat{u}_s=\mathrm{rem}(u_s,u_t)$, $\hat{v}_t=v_t-w_1v'_s$ とする. そして, $a_1=|\hat{u}_s|+|\hat{v}_t|$ と a_0 のうち小さい方を選び, a_0 ならば 0 を, a_1 ならば w_1 を返す. (3) の場合, さらに (3-1)$\mathrm{LT}(u_s)/\mathrm{LT}(u_t)\neq -\mathrm{LT}(v_t)/\mathrm{LT}(v'_s)$, (3-2)$\mathrm{LT}(u_s)/\mathrm{LT}(u_t)=-\mathrm{LT}(v_t)/\mathrm{LT}(v'_s)$ の二通りに分ける. (3-1) の場合, u_s を u_t によって簡約することができるうえ, v_t を v'_s によって簡約することもできる. しかし, 二つのノルム $|u_s|$ と $|v_t|$ を同時に下げることはできない. よって (2) と同様に w_1,a_1 を計算するとともに, $w_2=\mathrm{quo}(v_t,v'_s)$, $a_2=|\hat{u}_s|+|\hat{v}_t|=|u_s+w_2u_t|+|\mathrm{rem}(v_t,v'_s)|$ を計算する. 最後に a_0,a_1,a_2 を比較し, a_0 が最小の場合は 0 を返し, a_i $(i=1,2)$ の場合は w_i を返す. (3-2) の場合, u_s と v_t の最大次単項式を同時に消去できる. 消去した後に再び条件 (3-2) を満たす場合は, 同様に消去を繰り返す. 消去が n 回できた場合 $w_3=-\mathrm{LT}(u_s)/\mathrm{LT}(u_t)-\mathrm{LT}(u_s^1)/\mathrm{LT}(u_t)-\cdots-\mathrm{LT}(u_s^n)/\mathrm{LT}(u_t)$ とする. またノルムの和 $|\hat{u}_s|+|\hat{v}_t|$ を $a_3=|u_s+w_1u_t|+|v_t-w_1v'_s|$ とする. a_0,a_1,a_2 は (3-1) と同様にする. a_0,a_1,a_2,a_3 の大小関係は計算しない限り分からない. よって a_0 が最小であれば $w=0$ を返し, a_i $(1\leq i\leq 3)$ が最小であれば w_i を返す.

5.3 ノルム $|U|+|V|$ の最小化

$\{u_{r+1},\cdots,u_m\}$ によって張られる格子を L_1, $\{v_{r+1},\cdots,v_n\}$ によって張られる格子を L_2 とする. ReduceELM プロシジャと補題 5.3 を使い, ノルム $|u_1^*|+\cdots+|u_r^*|+|v_1^*|+\cdots+|v_r^*|$ を最小化するアルゴリズムを図示する :

Step $idx=\mathrm{finish}$
1 : for $s=1,\cdots,r\{$
2 : for $t=1,\cdots,r\{$
3 : if $s<t$
$w=\mathrm{ReduceELM}(u_s,u_t,v_t,v'_s)$
$\hat{u}_s=u_s+wu_t,\quad u_s\leftarrow\hat{u}_s$
$\hat{v}_t=v_t-wv'_s,\quad v_t\leftarrow\hat{v}_t$
4 : else if $s>t$
$w=\mathrm{ReduceELM}(u_s,u'_t,v_t,v_s)$
$\hat{u}_s=u_s-wu'_t,\quad u_s\leftarrow\hat{u}_s$
$\hat{v}_t=v_t+wv_s,\quad v_t\leftarrow\hat{v}_t$
5 : if $w\neq 0$
$idx=\mathrm{notyet}\}\}$
6 : for $s=1,\cdots,r\{$
7 : for $t=1,\cdots,r\{$
8 : $\hat{u}_i\xrightarrow{L_1}u_i^*,\quad \hat{v}_i\xrightarrow{L_2}v_i^*$
9 : if $(|u_i^*|<|\hat{u}_i|$ or $|v_i^*|<|\hat{v}_i|)$
$idx=\mathrm{notyet}\}\}$
10 : if $idx=\mathrm{notyet}$
goto **Step**

上記のアルゴリズムが終了したとき, ノルムを下げる計算はそれ以上できない. また, ノルムは整数で下限があるので, このアルゴリズムは必ず終了する.

補題 5.4. 上のアルゴリズムは終了し, $|u_1^*| + \cdots + |u_r^*| + |v_1^*| + \cdots + |v_r^*|$ を最小化できる.

6 結論と今後の課題

本論文では, $\mathbb{K}[x]$ 上で Smith 標準形とその変換行列が与えられたとき, ノルム最小の変換行列を求めるアルゴリズムを与えた. また, 関連する算法として, 縮小基底と PID 上加群の極小 Gröbner 基底が同一であることを示した.

今回, ベクトルのノルムを最大次単項式の次数として考えた. しかし, 計算の過程で多項式の係数が膨張しないかの検討をしていない. 今後大きな行列を扱う場合, 係数をどのように扱えば良いか考える必要がある.

また, PID 上有限生成加群の計算において, 本論文がどの程度の意味を持つのかを考える.

参 考 文 献

[Smi61] Henry J. Stephen Smith : On Systems of Linear Indeterminate Equations and Congruences, Phil. Trans. Royal Society **151** 1, 293-326 (1861).

[Min96] H. Minkowski. Geometrie der Zahlen. Teubner, Leipzig, 1896. Reprinted by Johnson, New York, NY 1968.

[LLL82] A.K. Lenstra, H.W. Lenstra, Jr., and L. Lovász : Factoring Polynomials with Rational Coefficients, Math. Ann. **261**, 515-524 (1982).

[KB79] R. Kannan and A. Bachem : Polynomial Algorithms for Computing the Smith and Hermite Normal Forms of an Integer Matrix, SIAM J. Comput **8**, no.4, 499-507 (1979).

[Kan85] R. Kannan : Solving System of Linear Equations over Polynomials, Theor. Comp. Sci. **39**, 69-88 (1985).

[Jag05] G. Jäger : Reduction of Smith Normal Form Transformation Matrices, Computing **74**, 377-388 (2005).

[Mic01] D. Micciancio : The Shortest Vector in a Lattice is Hard to Approximate to within Some Constant, SIAM J. Comput. **30**, no.6, 2008-2035 (2001).

[Len85] A.K. Lenstra : Factoring Multivariate Polynomials over Finite Fields, J. Comp. Sys. Sci. **30**, 235-248 (1985).

[Gat84] Joachim von zur Gathen : Hensel and Newton Methods in Valuation Rings, Math. Comp. **42**, no.166, 637-661 (1984).