数理解析研究所講究録
第1785巻 2012年 1-7

近似GCDの無平方分解への応用*
– 整数の無平方分解を格子算法に帰着させる試み –

長坂耕作
KOSAKU NAGASAKA†
神戸大学人間発達環境学研究科
GRADUATE SCHOOL OF HUMAN DEVELOPMENT AND ENVIRONMENT, KOBE UNIVERSITY

Abstract

整数係数多項式の近似GCDは，部分終結式写像の零空間に近いベクトルを格子算法で検出することにより計算が概ね可能です。また，この方法を整数の近似GCDに拡張することも可能で，そのためには格子を整数の桁毎に拡張する必要があります。今回は，これらのアプローチに基づき，整数の厳密な意味での無平方分解を行う試みについて速報します。

1 はじめに

整数係数多項式の近似GCD[5, 6]で用いたアプローチは，整数の桁毎に格子を拡張することで，整数の近似GCD[7]や様々な基数における誤差へ対応[8]させることが可能です。本報告では，これらのアプローチにより次の問題を解く試みについて扱います。なお，整数の無平方部分の計算や無平方分解などは，問題クラスが未解決となっています。本報告で解決するものではなく，格子算法による試みの経過報告となります。

問題 1 (Squarefree Part of Integer[1])
For the given positive integer $n \in \mathbb{Z}_{>0}$, find integers $r, s \in \mathbb{Z}_{>0}$ s.t. $n = r^2 s$ and s is a squarefree integer ($\forall m \in \mathbb{Z}_{>1},\ m^2 \nmid s$). ◁

2 整数係数多項式の近似GCD

まず，基本となる整数係数多項式の近似GCDの問題定義とそのアルゴリズムについて説明します。

問題 2 (Approximate Polynomial GCD Over Integers)
Let $f(\vec{x})$ and $g(\vec{x})$ be polynomials in variables $\vec{x} = x_1, \ldots, x_\ell$ over $\mathbb{Z}$, and let ε be a small positive integer. If $f(\vec{x})$ and $g(\vec{x})$ satisfy

$$f(\vec{x}) = t(\vec{x})h(\vec{x}) + \delta_f(\vec{x}),\ \ g(\vec{x}) = s(\vec{x})h(\vec{x}) + \delta_g(\vec{x}),\ \ \varepsilon = \max\{\|\delta_f\|, \|\delta_g\|\},$$

for some polynomials $\delta_f, \delta_g \in \mathbb{Z}[\vec{x}]$, then we say that the above polynomial $h(\vec{x})$ is an ***approximate GCD over integers****. We also say that $t(\vec{x})$ and $s(\vec{x})$ are* ***approximate cofactors over integers****, and we say that their* ***tolerance*** *is ε. ($\|p\|$ denotes a suitable norm of polynomial $p(\vec{x})$.)* ◁

*本研究の一部は科研費（22700011）の支援で行われている。
†nagasaka@main.h.kobe-u.ac.jp

例えば，次の 2 変数多項式の組であれば，近似 GCD は $(5x_1x_2 - 9x_2 - 3x_1 + 14)$ となります。

$$\begin{aligned} f(\vec{x}) &= 89x_1^2x_2^2 - 87x_1x_2^2 - 136x_2^2 + 15x_1^2x_2 + 132x_1x_2 + 119x_2 - 42x_1^2 + 166x_1 + 139, \\ g(\vec{x}) &= 56x_1^2x_2^2 - 45x_1x_2^2 - 98x_2^2 - 13x_1^2x_2 + 46x_1x_2 + 225x_2 - 12x_1^2 + 80x_1 - 112. \end{aligned}$$

係数において摂動された桁（$\delta_f(\vec{x})$ や $\delta_g(\vec{x})$ に対応する部分）は下線部分となっています。

$$\begin{aligned} f(\vec{x}) - \delta_f(\vec{x}) &= (18x_1x_2 + 15x_2 + 14x_1 + 10)(5x_1x_2 - 9x_2 - 3x_1 + 14) \\ &= \underline{90}x_1^2x_2^2 - 87x_1x_2^2 - 13\underline{5}x_2^2 + 1\underline{6}x_1^2x_2 + 13\underline{1}x_1x_2 + 1\underline{20}x_2 - 42x_1^2 + 166x_1 + 1\underline{40}, \\ g(\vec{x}) - \delta_g(\vec{x}) &= (11x_1x_2 + 11x_2 + 4x_1 - 8)(5x_1x_2 - 9x_2 - 3x_1 + 14) \\ &= 5\underline{5}x_1^2x_2^2 - 4\underline{4}x_1x_2^2 - 9\underline{9}x_2^2 - 13x_1^2x_2 + 4\underline{5}x_1x_2 + 22\underline{6}x_2 - 12x_1^2 + 80x_1 - 112. \end{aligned}$$

このような整数制約付きの多項式の近似 GCD に対しては，本稿の著者による方法 [5, 6] の他，J. von zur Gathen らによる方法 [9, 10] があります。前者の方法は，部分終結式写像と格子算法 [4] を使用しており大きな摂動にも対応可能なものの，理論的に計算可能な摂動や多項式の条件が与えられていません。一方，後者の方法は，Howgrave-Graham による整数の近似 GCD[3] の拡張となっており，実用的ではない（摂動が片方の多項式にしか許容できないなど）が理論的な条件が明示されています。

本稿の著者による方法を簡単に紹介します。まず，多変数の部分終結式写像を導入します。

定義 1 (部分終結式写像)
$Syl_r(f,g)$ を $f(\vec{x})$ と $g(\vec{x})$ に関する r 次の部分終結式写像と呼ぶ。なお，$r = 0, \ldots, \min\{n, m\} - 1$ である。

$$Syl_r(f,g): \begin{array}{ccc} \mathcal{P}_{m-r-1} \times \mathcal{P}_{n-r-1} & \to & \mathcal{P}_{n+m-r-1} \\ (s(\vec{x}), t(\vec{x})) & \mapsto & s(\vec{x})f(\vec{x}) + t(\vec{x})g(\vec{x}) \end{array}$$

なお，$\mathcal{P}_d$ は全次数 d 以下の多項式全体の集合，$f, g \in \mathbb{Z}[x_1, \ldots, x_\ell]$，$n = \mathrm{tdeg}(f)$，$m = \mathrm{tdeg}(g)$ とする。◁

近似 GCD の計算は，一般の GCD 計算や近似 GCD 計算でも用いられる性質（部分終結式写像が単射でない最大の r に対して，$f(\vec{x})/t(\vec{x})$ と $g(\vec{x})/s(\vec{x})$ は，$f(\vec{x})$ と $g(\vec{x})$ の GCD となる）を活用し，格子算法などと組み合わせることで（近似余因子，近似 GCD の順で）行います。

実際に，次の多項式について計算過程を説明します。

$$\begin{aligned} f(\vec{x}) &= 49x_1^2 - 2\underline{4}x_2^2 &&= (7x_1 - 5x_2)(7x_1 + 5x_2) + x_2^2, \\ g(\vec{x}) &= \underline{50}x_1^2 + 70x_1x_2 + 25x_2^2 &&= (7x_1 + 5x_2)(7x_1 + 5x_2) + x_1^2. \end{aligned}$$

まず，近似余因子を求めるために，部分終結式写像の行列を構成し，単位行列を付与してから行ベクトルの張る整数格子に対して格子算法を適用し，短いベクトルを計算します。近似余因子の係数ベクトルは，短いベクトルに含まれる形で検出されます（下線部のところが近似余因子の係数ベクトルです）。

$$\left(\begin{array}{cccccc|cccccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -24 & 0 & 0 & 0 & 0 & 49 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -24 & 0 & 0 & 0 & 0 & 49 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -24 & 0 & 0 & 49 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 25 & 0 & 0 & 70 & 0 & 50 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 25 & 0 & 0 & 70 & 0 & 50 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 25 & 0 & 70 & 50 \end{array}\right)$$

$$\Longrightarrow \left(\begin{array}{cccccc|cccccccccc} 0 & -2 & -3 & 0 & -2 & 3 & 0 & 0 & 0 & -2 & 0 & 0 & 7 & 0 & 12 & 3 \\ 0 & \underline{-5} & \underline{-7} & 0 & \underline{-5} & \underline{7} & 0 & 0 & 0 & -5 & 0 & 0 & -7 & 0 & -5 & 7 \\ 0 & 7 & 9 & 0 & 6 & -9 & 0 & 0 & 0 & -18 & 0 & 0 & -21 & 0 & 13 & -9 \\ 0 & 10 & 15 & 0 & 10 & -14 & 0 & 0 & 0 & 10 & 0 & 0 & -10 & 0 & 10 & 35 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -24 & 0 & 0 & 0 & 0 & 49 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 49 & 0 & 0 & 70 & 0 & 1 & 0 & 0 \end{array}\right)$$

部分終結式写像の行列は畳み込み行列から構成されていますが，同じく求まった近似余因子の畳み込み行列を用いて次のような行列を構成します。次に，単位行列を付与してから行ベクトルの張る整数格子に対して格子算法を適用し，短いベクトルを計算します。最終的に近似 GCD の係数ベクトルが，短いベクトルに含まれる形で検出されます（下線部のところが近似 GCD の係数ベクトルです)。結果，近似 GCD は $(7x_1+5x_2)$ で，許容度が $\varepsilon=1$ となります。

$$\left(\begin{array}{cccc|cccccccc} 1&0&0&0&0&0&-24&0&0&49&0&0&25&0&70&50\\ 0&1&0&0&0&-5&0&7&0&0&0&5&0&7&0&0\\ 0&0&1&0&0&0&-5&0&7&0&0&0&5&0&7&0\\ 0&0&0&1&0&0&0&0&-5&7&0&0&0&0&5&7 \end{array}\right) \Rightarrow \left(\begin{array}{cccc|cccccccc} 1&0&\underline{-5}&\underline{-7}&0&0&1&0&0&0&0&0&0&0&0&1\\ 0&1&0&0&0&-5&0&7&0&0&0&5&0&7&0&0\\ 0&0&1&0&0&0&-5&0&7&0&0&0&5&0&7&0\\ 0&0&0&1&0&0&0&0&-5&7&0&0&0&0&5&7 \end{array}\right).$$

なお，2009 年に投稿し 2011 年に出版された論文 [6] では，一般化された部分終結式写像を利用しているため，単変数から多変数までの多項式に対応しているのみならず，2 つを超える複数個の多項式の近似 GCD に対応しています。また，格子算法により求まる短いベクトルは多数あり，近似 GCD が求まるまで計算を繰り返すボトルネックの改善も行われています。これにより 2008 年のアルゴリズム [5] よりも 2 倍から 3 倍高速に動作するようになっています。

3 整数の近似 GCD

2010 年 7 月に行われた研究集会「数式処理研究の新たな発展 [7]」では，準同型暗号と整数及び整数多項式の近似 GCD という発表を行いました。その内容を簡単に紹介しておきます。まず，準同型暗号とは，暗号化されたデータを処理したいが，復号して平文を求めてからデータ処理を行う方法では，平文を求めないとデータ処理が不可能なため，第三者にデータ処理の委託が困難な問題を解決するものです。つまり，復号せずに平文を処理した結果の暗号文を得られるもので，特に全ての論理演算が可能な準同型暗号を，Fully Homomorphic Encryption Scheme と呼びます。

世界で最初の完全準同型暗号となった Gentry ら [2] の方法は，その安全性が整数の近似 GCD に帰着されます。この話を聞いて，多項式版の方法を整数に適用できないだろうかと試みたものです。なお，最大公約数を格子算法で求める一般的方法もありますので，その方法も一応紹介しておきます。497106617047 と 497608808351 の最大公約数を求めたい場合は，次のような行列を構成し，行ベクトルの張る格子の短いベクトルを求めますと，下線部のところに最大公約数が求まります。従って，以下で紹介する方法は，これを桁毎に拡張したものとも解釈可能です。

$$\left(\begin{array}{ccc} 1&0&497106617047\\ 0&1&497608808351 \end{array}\right) \Rightarrow \left(\begin{array}{ccc} 7927&-7919&0\\ 991&-990&\underline{-62773913} \end{array}\right)$$

本報告の著者の方法では，あくまでも多項式向けのアルゴリズムを整数に拡張することを想定していました。そのため，整数の 10 進法表記をそのまま多項式に次式で変換し，変換後の多項式の近似 GCD を求めることで，整数の近似 GCD を求めることを試みました。

$$c=\sum_{i=0}^{n} a_i\times 10^i\ (a_i\in\{0,1,\ldots,9\})\ \Rightarrow\ f(x)=\sum_{i=0}^{n} a_i x^i$$

例えば，$123456\ \Rightarrow\ x^5+2x^4+3x^3+4x^2+5x+6$，$450608\ \Rightarrow\ 4x^5+5x^4+6x^2+8$ といった変換を行うことになります。しかしながら，この方法には，整数の加減乗算時の桁上がりと桁下がりが，多項式の演算では単項式を超えた操作となり実現できないという問題がありました。

そこで，格子の基底を工夫することで，桁上がりと桁下がりの実現を行います。具体的には，加えると「桁下がり」となり，減じると「桁上がり」となる基底 $(\vec{u}=(0,\ldots,0,-1,10,0,\ldots,0))$ を格子に導入する

ことで桁上がりと桁下がりに対応しました。実際には，これをシフトしたものを必要な分だけ基底に導入することになります。この方法は単純なので既知かと思いましたが，見つけられていません。

さらに近似GCDを求めるためには，格子の基底ベクトルの要素に重みを付け，下位ビットを無視するように調整する必要があります。これらを踏まえた計算例を $c_1 = 325\times78+2 = 25352$ と $c_2 = 432\times78-1 = 33695$ で紹介します。まず，整数を多項式に変換しその部分終結式写像の行列表現に単位行列を付与したものを構成し，それに桁上りと桁下がりに対応する基底を追加し，さらに下位ビットを無視するよう上位ビットに重みを付けた次の行列を構成します。

$$\left(\begin{array}{cccccc|ccccccc}
1 & 0 & 0 & 0 & 0 & 0 & -300 & -300 & -600 & -90 & -50 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & -300 & -300 & -60 & -90 & -50 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -300 & -30 & -60 & -90 & -50 \\
0 & 0 & 0 & 1 & 0 & 0 & 200 & 500 & 300 & 50 & 20 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 200 & 500 & 30 & 50 & 20 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 200 & 50 & 30 & 50 & 20 \\
0 & 0 & 0 & 0 & 0 & 0 & -100 & 1000 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -100 & 1000 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -100 & 100 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -10 & 100 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -10 & 100 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -10 & 100
\end{array}\right)$$

この行列の行ベクトルの張る整数格子の短いベクトルを計算しますと，次のように，下線部の所に近似GCDの余因子が求まります。下線部のところは，10進整数に逆変換することにより，$(-3\ \ -1\ \ -3) \Rightarrow -313$ と $(-4\ \ -2\ \ 4) \Rightarrow -416$ になります。これらは予期していた結果とは異なりますが，$313\times81 = 25353$ と $416\times81 = 33696$ のように，予期したものよりも良い近似GCDを与えます。

$$\left(\begin{array}{cccccc|ccccccc}
0 & 0 & 0 & 1 & -10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & -10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & -10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-2 & 4 & 2 & -2 & -1 & 0 & 0 & 0 & 0 & 0 & -10 & -10 & 0 \\
6 & -3 & -2 & 7 & 5 & 5 & 0 & 0 & 0 & 20 & 0 & 0 & 0 \\
-8 & -5 & -4 & -11 & -3 & -5 & 0 & 0 & 0 & 10 & 0 & 10 & 0 \\
-2 & 3 & 3 & -2 & -2 & -2 & 0 & 0 & 0 & -10 & -10 & 20 & 10 \\
\underline{-3} & \underline{-1} & \underline{-3} & \underline{-4} & \underline{-2} & \underline{4} & 0 & 0 & 0 & 0 & 10 & 0 & 30 \\
0 & 0 & -5 & 0 & -1 & 3 & 0 & 0 & -100 & 10 & 0 & 10 & 10 \\
6 & 4 & 1 & 8 & 5 & -2 & 0 & -100 & 0 & 0 & 0 & 0 & 10 \\
0 & 2 & -3 & 1 & -4 & 2 & 100 & 0 & 0 & -10 & 0 & 10 & -10
\end{array}\right)$$

4 整数の無平方分解と格子算法

これまでのアプローチに基づき整数の最大公約数を，桁毎の大きな格子で計算することが可能であった。この方法は確かに一般の方法よりも手間はかかるが，桁毎であることを活かすことで整数の無平方分解に応用出来るのではないか，というのが本報告の試みになります。まず，今回の取り組みのスタート地点となる多項式の無平方分解アルゴリズムの1つ (Yun's Algorithm) を挙げておきますが，無平方部分だけの計算ならば，$f/\gcd(f,f')$ だけで十分な点に注意してください。

1. $u \leftarrow \gcd(f,f')$, $v_1 \leftarrow \frac{f}{u}$, $w_1 \leftarrow \frac{f'}{u}$, $i \leftarrow 1$
2. **repeat** $h_1 \leftarrow \gcd(v_i,\ w_i - v_i')$, $v_{i+1} \leftarrow \frac{v_i}{h_i}$, $w_{i+1} \leftarrow \frac{w_i - v_i'}{h_i}$, $i \leftarrow i+1$ **until** $v_i = 1$
3. **return** $(h_1,\ldots,h_{i-1})$

このアルゴリズムを確認することで，整数の無平方分解が難しいことの理由がわかります。アルゴリズムで求められている計算は，Euclid の互除法と導関数の計算のみですが，ここに問題があります。多項式と同じく Euclid 環の要素である整数には，Euclid の互除法は当然可能ですが，整数には導関数がありません。$\gcd(f, f')$ の計算が整数では困難なことが，多項式との最大の違いとなります。これをどのようにして克服するかが問題となります。

4.1 桁毎に拡張した格子の復習

導関数の問題はとりあえず脇に置き，桁毎に拡張した格子による整数の GCD 計算について，きちんと定義し直しておきます。まず，$\mathbb{Z}$ から $\mathbb{Z}^w$ への写像 $\mathrm{digits}_{b,w}(\cdot)$ と逆写像 $\mathrm{digits}_{b,w}^{-1}(\cdot)$ を次のように定義します。ここで，a は対象の整数で，b は基数（本稿の前半の例では $b = 10$），w は変換後のベクトルの長さを表しています。

$$\begin{aligned}&\mathrm{digits}_{b,w}(a) = \{a_{w-1}, \ldots, a_1, a_0\} \text{ for } a = \textstyle\sum_{i=0}^{w-1} a_i b^i,\ 0 \le \mathrm{sign}(a) a_i < b\ (i < w-1),\\ &\mathrm{digits}_{b,w}^{-1}(\vec{a}) = \textstyle\sum_{i=0}^{w-1} a_i b^i \text{ for } \vec{a} = \{a_{w-1}, \ldots, a_1, a_0\}^t \in \mathbb{Z}^w.\end{aligned}$$

この写像は例えば，$\mathrm{digits}_{10,2}(123) = \{12, 3\}$，$\mathrm{digits}_{10,3}(123) = \{1, 2, 3\}$，$\mathrm{digits}_{10,4}(123) = \{0, 1, 2, 3\}$，$\mathrm{digits}_{10,3}(-123) = \{-1, -2, -3\}$ などのような変換を表します。次に，整数格子において桁上がりと桁下がりを実現させる基底ベクトル $\vec{z}_{b,w,i}$ とそれによって構成される行列 $\mathcal{Z}_{b,w}$ を次式で定義します。基底ベクトルは，$\mathrm{digits}_{b,w}^{-1}(\vec{z}_{b,w,i}) = 0$ という重要な性質を持っています。

$$\vec{z}_{b,w,i} = \{\underbrace{0, \ldots, 0}_{i}, -1, b, \underbrace{0, \ldots, 0}_{w-i-2}\}^t \in \mathbb{Z}^w,\ \mathcal{Z}_{b,w} = \{\vec{z}_{b,w,0}\ \cdots\ \vec{z}_{b,w,w-2}\}^t \in \mathbb{Z}^{(w-1)\times w}.$$

さらに，b 進法表記の整数を多項式に関連付ける変換を次式で定義します。これにより基数 b を変数とする多項式と，b 進法表記の整数を対応付けることが可能となります。

$$\begin{aligned}&\mathrm{vect}_{b,w}^{-1}(\vec{a}) = \textstyle\sum_{i=0}^{w-1} a_i x^i \text{ for } \vec{a} = \{a_{w-1}, \ldots, a_1, a_0\} \in \mathbb{Z}^w,\\ &\mathrm{vect}_{b,w}(f) = \{f_{w-1}, \ldots, f_1, f_0\} \text{ for } f(x) = \textstyle\sum_{i=0}^{w-1} f_i x^i \in \mathbb{Z}[x]_{w-1}.\end{aligned}$$

以上の定義により，$\mathbb{Z}[x]_{w-1}$ の剰余加群，$\mathbb{Z}^w$ の剰余加群，整数集合が同型となります。剰余加群の同値類は，$\mathrm{digits}_{b,w}^{-1}(\vec{z}_{b,w,i}) = 0$ なる関係を用いた同値関係で定義されるものです。従って，整数の GCD 計算を，これらの同型で $\mathbb{Z}[x]_{w-1}$ の剰余加群上の GCD 計算に帰着できます。そのため，多項式用のアルゴリズムが加群上で実行可能ならば，整数の問題に使用できます。つまり，本稿の前半で紹介した整数の近似 GCD で用いた方法は，$\mathbb{Z}$ の部分終結式写像の核を $\mathbb{Z}^w$ の剰余加群で計算したものと解釈できます。

4.2 導関数の整数への導入

記法の準備が出来たところで，本題であった導関数がないために無平方分解が難しいことへの対応を試みます。まず，$\mathbb{Z}[x]_{w-1}$ の剰余加群上への形式的微分を，通常の定義通りに導入します。

$$\forall f(x) = \sum_{i=0}^{w-1} f_i x^i \in \mathbb{Z}[x]_{w-1},\ f'(x) = \sum_{i=0}^{w-2} (i+1) f_{i+1} x^i.$$

次に，$\mathbb{Z}[x]_{w-1}/\langle\cdot\rangle$ での $\gcd(f, f')$ の計算について考えます。これは，同型の $\mathbb{Z}^w$ 上で部分終結式の核を計算すれば，十分であることがわかります。問題は $\gcd(f, f')$ が，整数集合上で何に対応しているかになります。これが $f(x)$ に対応する整数の無平方部分に対応していれば良いのですが，残念ながら，加群としての同型な

ので，$\gcd(f, f')$ は必ずしも無平方部分には対応しません。実際，$17^2 23^1 = 6647$ を $f(x) = 6x^3+6x^2+4x+7$ と考えて，$\gcd(f, f')$ を計算しますと，$\gcd(6x^3 + 6x^2 + 4x + 7,\ 18x^2 + 12x + 4) = 1$ となり，期待される結果の $x + 7$ にはなりません（整数としての重複部分は 17 なので，$x + 7$ が得たい結果です）。

この計算（$6647 = 17^2 23^1$ の分解の試み）で何が求まっているのでしょうか。理想的には次の計算ですが，

$$\begin{aligned}\mathrm{vect}_{10,4}^{-1}(\mathrm{digits}_{10,4}(17)) = x + 7,\ \mathrm{vect}_{10,4}^{-1}(\mathrm{digits}_{10,4}(23)) = 2x + 3 &\Longrightarrow 17^2 23^1 \mapsto (x + 7)^2(2x + 3),\\ &\Longrightarrow \gcd(2x^3 + 31x^2 + 140x + 147,\ 6x^2 + 62x + 140) = x + 7.\end{aligned}$$

実際には，形式的微分が同値関係と整合性が取れないので，以下を計算していることになります。

$$\begin{aligned}\mathrm{vect}_{10,4}^{-1}(\mathrm{digits}_{10,4}(6647)) = 6x^3 + 6x^2 + 4x + 7 &\equiv 2x^3 + 31x^2 + 140x + 147 \pmod{\mathcal{Z}},\\ (6x^3 + 6x^2 + 4x + 7)' = 18x^2 + 12x + 4 &\not\equiv 6x^2 + 62x + 140,\\ \Rightarrow \gcd(6x^3 + 6x^2 + 4x + 7,\ 18x^2 + 12x + 4) = 1 &\not\equiv x + 7.\end{aligned}$$

つまり，6647 の導関数が意図したものを表現していないのが原因です。

多項式:	$6x^3 + 6x^2 + 4x + 7$	$\equiv$	$2x^3 + 31x^2 + 140x + 147$
導関数:	$18x^2 + 12x + 4$	$\not\equiv$	$6x^2 + 62x + 140$

形式的微分は，剰余加群の同値関係と不整合のため当然の結果です。この状態で期待した通りの導関数を得るには，適切な代表元に対して形式的微分を行う必要があります。

$$\begin{aligned}6x^3 + 6x^2 + 4x + 7 &\equiv 2x^3 + 31x^2 + 140x + 147 \pmod{\mathcal{Z}}\\ &= 6x^3 + 6x^2 + 4x + 7 + \textstyle\sum_{i=0}^{2} a_i \times \mathrm{vect}_{10,4}^{-1}(\vec{z}_{b,w,i})\\ &= (6 - a_2)x^3 + (6 + 10a_2 - a_1)x^2 + (4 + 10a_1 - a_0)x + (7 + 10a_0)\\ &\Rightarrow (\cdot)'|_{a_0=14,\ a_1=15,\ a_2=4} = 6x^2 + 62x + 140.\end{aligned}$$

4.3 包括的格子簡約 (Comprehensive Lattice Basis Reduction)

前節までの議論によって，剰余加群としての同型関係（$\mathbb{Z}$, $\mathbb{Z}^w$, $\mathbb{Z}[x]_{w-1}$）や部分終結式写像の核による整数の最大公約数の計算が可能（一般の格子による GCD 計算を，単純拡張したもの）なことが，桁毎格子 (Digits-wise Lattice) の導入により分かりました。一方で，整数の無平方分解を行うには，形式的微分 ($f(x) = \sum_{i=0}^{w-1} f_i x^i \Rightarrow f'(x) = \sum_{i=0}^{w-2}(i+1) f_{i+1} x^i$) が，同値関係に透過的に働かないため，$\gcd(f, f')$ の計算結果は期待した結果となりませんでした。無平方分解に必要な計算結果を得るためには，代表元を適切に選択する必要が分かったことになります。

先の $6647 = 17^2 23^1$ の無平方分解を採り上げて，具体的に示します。この分解を行うためには，次の様なパラメータ (a_0, a_1, a_2) を含む行ベクトルで張られる整数格子上で，短いベクトルとそのときのパラメータを求める必要があります（実際の解は，$a_0 = 14$, $a_1 = 15$, $a_2 = 4$）。

$$\begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 7 & 4 & 6 & 6 & 0\\
0 & 1 & 0 & 0 & 0 & 0 & 7 & 4 & 6 & 6\\
0 & 0 & 1 & 0 & 0 & g_0 & g_1 & g_2 & 0 & 0\\
0 & 0 & 0 & 1 & 0 & 0 & g_0 & g_1 & g_2 & 0\\
0 & 0 & 0 & 0 & 1 & 0 & 0 & g_0 & g_1 & g_2\\
0 & 0 & 0 & 0 & 0 & 10 & -1 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 10 & -1 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 10 & -1 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10 & -1
\end{pmatrix},\quad
\begin{cases}
g_0 = -a_0 + 10a_1 + 4,\\
g_1 = -2a_1 + 20a_2 + 12,\\
g_2 = 18 - 3a_2.
\end{cases}$$

このことから次のような問題が解ければ，この方法で整数の無平方分解が解けることがわかります。

問題 3 (包括的格子簡約 (Comprehensive Lattice Basis Reduction))
$L \subset \mathbb{Z}[a_1, \ldots, a_n]^m$ (a_i: パラメータ) に対して，十分小さい簡約基底とパラメータの値を計算せよ。 ◁

包括的格子簡約の簡単な例を挙げておきます。$\vec{u}_1 = (1,\ 1,\ 0)$, $\vec{u}_2 = (1,\ a_0,\ 1) \in \mathbb{Z}[a_0]^3$ で生成される整数格子の場合，$a_0 = 1$ で短いベクトル $(0,0,1)$ が格子に含まれます。なお，$a_0 \neq 1$ での短いベクトルはパラメータの値に関わらず，$(1,1,0)$ となります。包括的 Gröbner 基底と同じく興味深い問題と思っています。

参 考 文 献

[1] L. Adleman and K. McCurley. Open problems in number theoretic complexity, II. Lecture Notes in Comput. Sci. 877, 291–322, 1994.

[2] M. Dijk, C. Gentry, S. Halevi and V. Vaikuntanathan, Fully Homomorphic Encryption over the Integers. EUROCRYPT 2010, Lecture Notes in Comput. Sci. 6110, 24–43, 2010.

[3] N. Howgrave-Graham. Approximate integer common divisors. In: Cryptography and lattices (Providence, RI, 2001). Lecture Notes in Comput. Sci. 2146, 51–66, 2001.

[4] A. K. Lenstra, H. W. Lenstra Jr. and L. Lovász. Factoring polynomials with rational coefficients. Math. Ann. 261 (4), 515–534, 1982.

[5] K. Nagasaka. Approximate polynomial gcd over integers (ISSAC 2008 poster session). ACM Communications in Computer Algebra 42 (3), 124–126, 2008.

[6] K. Nagasaka. Approximate polynomial gcd over integers. J. Symbolic Comput. 46 (12), 1306–1317, 2011.

[7] 長坂耕作. 準同型暗号と整数及び整数多項式の近似 GCD. 研究集会 数式処理研究の新たな発展. 京都大学数理解析研究所. 2010 年 7 月. ACM Communications in Computer Algebra, Vol. 45, No. 3, Issue 177. 165, 2011. 数理解析研究所講究録 1759, 115 – 123, 2011.

[8] K. Nagasaka. An improvement in the lattice construction process of approximate polynomial gcd over integers (extended abstract). In: Proceedings of Symbolic-Numeric Computation (SNC2011). 63–64, 2011.

[9] J. von zur Gathen and I. E. Shparlinski. Approximate polynomial gcd: small degree and small height perturbations. In: LATIN 2008: Theoretical informatics. Lecture Notes in Comput. Sci. 4957, 276–283, 2008.

[10] J. von zur Gathen, M. Mignotte and I. E. Shparlinski. Approximate polynomial gcd: Small degree and small height perturbations. J. Symbolic Comput. 45 (8), 879–886, 2010.