数理解析研究所講究録
第2019巻 2017年 18-27

拡張Hensel構成による多変数多項式の近似GCD計算とその安定化：その1*

讃岐 勝
Masaru Sanuki
筑波大学医学医療系臨床医学域
Division of Clinical Medicine, Faculty of Medicine, University of Tsukuba
稲葉 大樹
Daiju Inaba
日本数学検定協会
The Mathematics Certification Institute of Japan
佐々木 建昭
Tateaki Sasaki
筑波大学名誉教授
Professor emeritus, University of Tsukuba

Abstract

本稿では，拡張 Hensel 構成の近似 GCD 計算への適応について検討を行う．

1 はじめに

多変数多項式の厳密 GCD &近似 GCD 計算は古くから研究がされ，2000 年以降は近似 GCD でも実用的なアルゴリズムが提案されるまでになり，大きな問題は解決されたように思われた．しかし，疎な多変数多項式の取り扱いについては，因数分解においては 2005 年まで [3]，GCD 計算においては 2015 年に著者らが触れるまで検討されなかった現状がある [9].

多変数多項式の因数分解・GCD 計算において Hensel 構成（一般 Hensel 構成）を利用する方法 [1, 2] は有効であるが，展開点において多項式が特異な場合（後述：2 章を参照）にはそうではない．Hensel 構成を適応するためには初期因子が互いに素である必要があり，そうでない場合には展開点の移動が必要となる．ただし，展開点の移動を行うと項数が増え計算効率を落としてしまう（非零代入問題）.

上の Hensel 構成の問題点を解決するアルゴリズムとして**拡張 Hensel 構成**が提案された [11, 12]（詳細は 2 章を参照）．拡張 Hensel 構成は，多項式が特異な場合において展開点を移動させること

*本研究は日本学術振興会・科学研究費 (課題番号 15K00006) の援助で遂行された。

なく Hensel 構成のストラテジーを適応することができる．本アルゴリズムは，疎な多項式の因数分解において著しい効率化に成功した [3].

一方，GCD 計算においては大きな成功を収めるまでには至らなかった．GCD を計算する方法として, 互除法や Hensel 構成による方法（EZ-GCD(extended Zassenhaus GCD) 法 [6, 17]）がよく知られるが，数式処理ソフト Maple では Zippel の補題に基づく補間法を基にした方法が実装されており，実際に利用すると非常に高速なことがわかる [18, 19]．拡張 Hensel 構成による GCD 計算は 2 倍程度遅い結果であったが，工夫次第ではさらなる効率化の余地が残されている．

本稿では，拡張 Hensel 構成を振り返ると共に，いくつかの工夫について述べる．工夫は近似 GCD 計算での適応を考慮しているが，厳密 GCD でも適応できるものである．

本稿では次の記号を用いる．標数 0 の体 $\mathbb{K}$ を係数にもつ主変数 x，従変数 $\boldsymbol{u} = (u_1, \ldots, u_\ell)$ の多変数多項式集合を $\mathbb{K}[x, \boldsymbol{u}]$ で表す（$\mathbb{K}$ は有理数全体 $\mathbb{Q}$ または浮動小数全体 $\mathbb{F}$）．多項式 $F \in \mathbb{K}[x, \boldsymbol{u}]$ に対して，$\deg_u(F)$ は変数 u に関する次数，$\mathrm{lc}(F)$ は主変数 x に関する主係数をそれぞれ表す．

2 拡張 Hensel 構成

算法の解説は厳密 GCD の場合で述べるが，そのまま近似 GCD 計算にも利用できる．

EZ-GCD 法による $F, G \in \mathbb{K}[x, \boldsymbol{u}]$ の GCD$\gcd(F, G) = C$ の計算法は $H = aF + bG$ の因子分離に基づく $(a, b \in \mathbb{K})$．GCD の低次項 $C^{(0)}$ と $D^{(0)} = H^{(0)}/C^{(0)}$ を初期因子として高次項 $(\delta C^{(1)}, \delta D^{(1)}) \Rightarrow (\delta C^{(2)}, \delta D^{(2)}) \Rightarrow \cdots \Rightarrow (\delta C^{(k)}, \delta D^{(k)}) \Rightarrow \cdots$ を順に計算する：

$$H = (C^{(0)} + \delta C^{(1)} + \cdots + \delta C^{(k)} + \cdots)(D^{(0)} + \delta D^{(1)} + \cdots + \delta D^{(k)} + \cdots).$$

一般 Hensel 構成と拡張 Hensel 構成の違いは初期因子の構成法・属する多項式環である．

- 一般 Hensel 構成：初期因子は 1 変数多項式環 $\mathbb{K}[x]$
  展開点を原点 $\boldsymbol{u} = \boldsymbol{0} \in \mathbb{K}^\ell$ とし，$H^{(0)} = H(x, \boldsymbol{0}) \in \mathbb{K}[x]$ とおく．初期因子を $C^{(0)} = \gcd(F(x, \boldsymbol{0}), G(x, \boldsymbol{0})) \in \mathbb{K}[x]$，$D^{(0)} = H^{(0)}/C^{(0)} \in \mathbb{K}[x]$ とおく．ここで，$\gcd(C^{(0)}, D^{(0)}) = 1$ & $\deg_x(C) = \deg_x(C^{(0)})$ を満たす必要がある．
  満たされない場合，次の拡張 Hensel 構成にアルゴリズムを移行する．
- 拡張 Hensel 構成：初期因子は多変数多項式環 $\mathbb{K}[x, \boldsymbol{u}]$
  $H^{(0)} \in \mathbb{K}[x, \boldsymbol{u}]$ を次の手順で構成：
  ① $H = \sum_i h_i x^{e_x^{(i)}} {u_1}^{e_{u_1}^{(i)}} \cdots {u_\ell}^{e_{u_\ell}^{(i)}}$ と表す時，各項 $x^{e_x^{(i)}} {u_1}^{e_{u_1}^{(i)}} \cdots {u_\ell}^{e_{u_\ell}^{(i)}}$ の指数部なる点 $(e_x^{(i)}, e_{u_1}^{(i)} + \cdots + e_{u_\ell}^{(i)})$ を平面上にプロットし，この点集合からなる凸包（Newton Polygon）を構成する．
  ② Newton Polygon の下包において，各辺 $\mathcal{L}_1, \ldots, \mathcal{L}_d$ を Newton 線と呼ぶ．任意に選んだ Newton 線 $\mathcal{L}_i$ 上の点に対応する多項式の和を $H^{(0)}$ とおき，この多項式を Newton 線 $\mathcal{L}_i$ に対する Newton 多項式 $N_{\mathcal{L}_i}$ と呼ぶ（通常，下包の最右点を含む Newton 線 $N_{\mathcal{L}}$ を選ぶ）．
  ③ F と G について，$\mathcal{L}$ 上の点に対応する多項式 $N_{\mathcal{L}}(F)$ と $N_{\mathcal{L}}(G)$ の GCD を $C^{(0)}$，$D^{(0)} = \mathrm{quo}(H^{(0)}, C^{(0)})$ とおく．ここで，$\gcd(C^{(0)}, D^{(0)}) = 1$ & $\deg_x(C) = \deg_x(C^{(0)})$ を満たす必要がある（満たさない場合の対応策は 2.1 で解説）．

互いに素な初期因子 $C^{(0)}$ および $D^{(0)}$ から $PC^{(0)}+QD^{(0)}=R$ with $\deg_x(R)=0$ を計算し，$W_C^{(0)}C^{(0)}+W_D^{(0)}D^{(0)}=1$ を構成する (Euclid の拡張互除法で計算できる)．その後，次を満たす Moses-Yun 補間式 $(W_C^{(i)},W_D^{(i)})$ を計算する $(i=0,\ldots,\deg(H)-1)$.

$$W_C^{(i)}C^{(0)}+W_D^{(i)}H^{(0)}=x^i. \tag{1}$$

2 つの方法では，Moses-Yun 補間式 $(W_C^{(i)},W_D^{(i)})$ の属する多項式環が異なる．

- 一般 Hensel 構成：$(W_C^{(i)},W_D^{(i)})\in\mathbb{K}[x]^{(2)}$
- 拡張 Hensel 構成：$(W_C^{(i)},W_D^{(i)})\in\mathbb{K}(\boldsymbol{u})[x]^{(2)}$

Moses-Yun 補間式を構成後，$\mathrm{lc}(C)=\gcd(\mathrm{lc}(F),\mathrm{lc}(G))$ は事前に計算をし $C^{(0)}$ にかけ，$\mathrm{lc}(H)/\mathrm{lc}(C)$ を $H^{(0)}$ にかける．

$(\delta C^{(k)},\delta D^{(k)})$ まで計算されたと仮定するとき，$(\delta C^{(k+1)},\delta D^{(k+1)})$ は次のように構成できる．差 $\delta H^{(k+1)}\equiv H-(C^{(0)}+\sum_i^k\delta C^{(i)})(D^{(0)}+\sum_i^k\delta D^{(i)})\pmod{I^{k+2}}$ について，$\delta H^{(k+1)}=\delta D^{(k+1)}C^{(0)}+\delta C^{(k+1)}D^{(0)}$ より $\delta H^{(k+1)}=\sum\delta h_i^{(k+1)}x^i$ と表す時，

$$\delta C^{(k+1)}=\sum_i\delta h_i^{(k+1)}W_D^{(i)},\ \delta D^{(k+1)}=\sum_i\delta h_i^{(k+1)}W_C^{(i)}.$$

ここで，I はイデアルで $I^k=\langle\boldsymbol{u}^{k\lambda}\rangle$ で λ は Newton 線 $N_{\mathcal{L}}$ の傾きである．

注意 1 (展開点の移動：非零代入問題)
初期因子が互いに素で無い場合，展開点の移動 $\boldsymbol{u}\mapsto\boldsymbol{u}-\boldsymbol{s}$ *for some* $\boldsymbol{s}\in\mathbb{K}^\ell$ によって初期因子が互いに素となるよう変換可能である．ただし，変換によって項数が爆発的に増加してしまうため計算効率は著しく低下する．∎

例 1
次式 $F(x,y,z)$ について，平行移動 $F(x,y-1,z-1)$ を行う．

$$\begin{aligned}F(x,y,z) &= [\,x^2y^2z+x\,(y^{50}+z^{50})+3y+3z-3z^2-2y^{25}z^{25}\,]\\ &\times\ [\,x^3y^2z^2+x\,(y^{50}+z^{50})-2y-5z+4y^2+3y^{25}z^{25}\,]\end{aligned}$$

このとき，項数は 39 から 9813 へと爆発的に増加する．

例 2 (浮動小数係数多項式（有限精度の多項式）の平行移動)
次の浮動小数係数の多項式 $F(x,y,z)$ について，平行移動 $F^{(1)}(x,y,z)=F(x,y-1,z-1)$ を行い元に戻す操作 $F^{(2)}(x,y,z)=F^{(1)}(x,y+1,z+1)$ を行う．

$$F=x^2y^2z+x(y^{50}+z^{50})+3y+3z-3z^2-2.000001y^{25}z^{25}$$

このとき，$F-F^{(2)}$ は完全誤差項のため 0 にはならない．

$$
\begin{aligned}
&-xy^{34} - 8.0\cdots xy^{32} - 128.0\cdots xy^{31} + 104.0\cdots xy^{30} + 128.0\cdots xy^{29} + 1104.0\cdots xy^{28}\\
&+8704.0\cdots xy^{27} + 3772.0\cdots xy^{26} + 25440.0\cdots xy^{25} + 22548.0\cdots xy^{24} + 38880.0\cdots xy^{23}\\
&-358744.0\cdots xy^{22} + 168512.0\cdots xy^{21} + 343600.0\cdots xy^{20} + 281600.0\cdots xy^{19}\\
&-316593.0\cdots xy^{18} + 317312.0\cdots xy^{17} + 695920.0\cdots xy^{16} - 530688.0\cdots xy^{15}\\
&+1077988.0\cdots xy^{14} - 31096.0\cdots xy^{13} - 459452.0\cdots xy^{12} + 148720.0\cdots xy^{11}\\
&+182182.0\cdots xy^{10} + 4512.0\cdots xy^{9} - 4736.0\cdots xy^{8} + 5088.0\cdots xy^{7}\\
&-4828.0\cdots xy^{6} - 248.0\cdots xy^{5} + 36.0\cdots xy^{4} + 2.0\cdots xy^{2}\\
&\cdots\\
&+3.6\cdots 10^{-12}y^{25}z^{21} - 2.1\cdots 10^{-10}y^{25}z^{20} - 9.8\cdots 10^{-10}y^{25}z^{19} + 1.0\cdots 10^{-9}y^{25}z^{18}\\
&+2.4\cdots 10^{-8}y^{25}z^{17} - 6.6\cdots 10^{-8}y^{25}z^{16} - 2.8\cdots 10^{-7}y^{25}z^{15} - 6.5\cdots 10^{-8}y^{25}z^{14}\\
&+4.8\cdots 10^{-8}y^{25}z^{13} - 8.1\cdots 10^{-8}y^{25}z^{12} + 3.0\cdots 10^{-6}y^{25}z^{11} - 1.9\cdots 10^{-6}y^{25}z^{10}\ \cdots\\
&-1.8\cdots 10^{-10}z^{25} + 1.4\cdots 10^{-8}z^{24} + 2.97z^{2}3 + 1.3\cdots 10^{-6}z^{22} - 1.3\cdots 10^{-5}z^{2}1\\
&-0.0002\cdots z^{20} - 0.001\cdots z^{19} - 0.009\cdots z^{18} - 0.03\cdots z^{17} - 0.09\cdots z^{16}\\
&-0.2\cdots z^{15} - 0.5\cdots z^{14} - 1.1\cdots z^{13} - 2.1\cdots z^{12} - 3.4\cdots z^{11} - 4.5\cdots z^{10}\\
&-4.7\cdots z^{9} - 3.9\cdots z^{8} - 2.3\cdots z^{7} - 1.0\cdots z^{6} - 0.2\cdots z^{5} - 0.03\cdots z^{4} + 0.005\cdots z^{3} + 0.002\cdots z^{2}
\end{aligned}
$$

微小な係数からそうでない係数まで現れる．この例は浮動小数係数多項式において平行移動をしてはいけないことを示している．

注意 2 (近似 GCD における算法の選択)
初期因子を求めるための近似 *GCD* 計算，*Moses-Yun* 補間式の計算において，*Euclid* の拡張互除法は数値的に不安定になる．初期因子導入のための近似 *GCD* は，入力が疎でないことがほとんどなので既存の近似 *GCD* 算法を利用，*Moses-Yun* 補間式の計算は *QR* 法に基づく方法で構成すれば，精度よく計算することができる．効率に関しては現状で検討はしていない．∎

2.1 計算の工夫

疎な多項式・特異な多項式で厳密&近似 GCD を計算する上でネックとなるのは，Ⓐ 初期因子が共通因子を持つこと，ⒷMoses-Yun 補間式の式の膨張であり，本質的な問題の 1 つは Newton 多項式の取り方にある．

以下，Newton 多項式を（簡単に）再構成するための方法を紹介する．

T-0：主変数を取り替える．

T-1：（本質的でない）入力を $F(x,\boldsymbol{u}) \mapsto x^{\deg(F)}F(1/x,\boldsymbol{u})$，$G(x,\boldsymbol{u}) \mapsto x^{\deg(G)}G(1/x,\boldsymbol{u})$ と変換．GCD は $x^{\deg(C)}C(1/x,\boldsymbol{u})$ で得られる．

T-2：（従変数の重み付け：その 1）従変数の重みを変化させる，すわなち次の変換を F と G に対して行う．$F(x,u_1,\ldots,u_\ell) \mapsto F(x,u_1^{w_1},\ldots,u_\ell^{w_\ell})$，$G(x,u_1,\ldots,u_\ell) \mapsto G(x,u_1^{w_1},\ldots,u_\ell^{w_\ell})$．

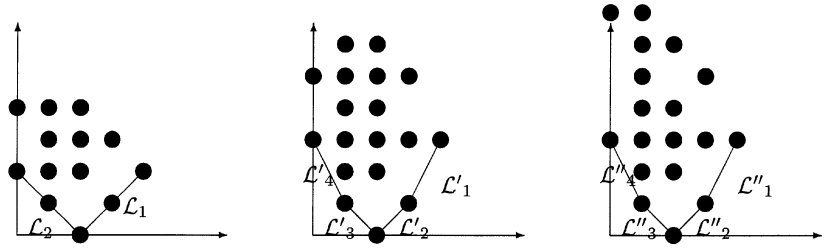
T-3：（従変数の重み付け：その 2）ある変数 u_i に関して
$F(x,u_1,\ldots,u_i,\ldots,u_\ell) \mapsto u_i^{\deg(F)}F(x,u_1,\ldots,1/u_i,\ldots,u_\ell)$，
$G(x,u_1,\ldots,u_i,\ldots,u_\ell) \mapsto u_i^{\deg(G)}G(x,u_1,\ldots,1/u_i,\ldots,u_\ell)$ とする．GCD は
$x^{\deg_{u_i}(C)}C(x,u_1,\ldots,1/u_i,\ldots,u_\ell))$ で得られる．

例 3 (拡張 Hensel 構成)
次の *GCD* を計算する（入力された多項式は展開されている:いずれも *22* 項）.

$$\begin{aligned} F(x,y,z) &= [(x+y)(xz+1)+yz]\times[(x+z)(xy+1)-z^2], \\ G(x,y,z) &= [(x+y)(xz+1)+yz]\times[(x+z)(xy+1)-y^2]. \end{aligned}$$

$F(x,0,0)=G(x,0,0)=x^2$ のため，一般 *Hensel* 構成に基づく方法は適応できない．このため，拡張 *Hensel* 構成に関する方法を適応する.

F と G の対応する *Newton* 線は同じである： $N_{\mathcal{L}}(F)=N_{\mathcal{L}}(G)=x^2yz+xy+xz+1$. 従変数の重みを $F(x,u^2,z), G(x,u^2,z)$ と変えたものを考える．このとき *Newton* 多項式は *2* つとも一緒になるが，*Newoton Polycon* の形は変わる.



Newton 線 $\mathcal{L}_1$ が $\mathcal{L}'_1$, $\mathcal{L}'_2$ および $\mathcal{L}''_1$, $\mathcal{L}''_2$ と，$\mathcal{L}_2$ が $\mathcal{L}'_3$, $\mathcal{L}'_4$ および $\mathcal{L}''_3$, $\mathcal{L}''_4$ と折れた．これは，*Newton* 多項式 $N_{\mathcal{L}_1}$,$N_{\mathcal{L}_2}$ が *2* つ以上の因子に分離することを示しており，各多項式が既約ではないことを示している．これらの情報を基に *GCD* を計算できるが，詳細は *[9]* を参照いただきたい.

この例の場合は変換 $y\mapsto 1/u$（**T-3**）を適応すると，一般 *Hensel* 構成を基にした方法が適応できる．∎

3 計算上のボトルネック

本章では次の点について改良の検討する.

1. Moses-Yun 補間式の構成法
2. 各 Hensel 因子 $\delta C^{(k+1)}, \delta D^{(k+1)}$ の構成における効率化

3.1 Moses-Yun 補間式の構成法

近似 GCD を求める場合，入力の係数は浮動小数係数である．拡張 Hensel 構成にて，Moses-Yun 補間式の計算は全体の計算精度に大きく影響する．厳密 GCD の場合は拡張 Euclid の互除法によって計算を行うが，浮動小数係数の多項式の場合は計算が破綻する．このため，精度を保持しながら計算を実行する必要がある.

$C_{\mathcal{N}}^{(0)}$ と $D_{\mathcal{N}}^{(0)}$ の Moses-Yun 補間式は Bezout identity を基に構成される.

$$AC_{\mathcal{N}}^{(0)}+BD_{\mathcal{N}}^{(0)}=R$$

ここで R は $C_{\mathcal{N}}^{(0)}$ と $D_{\mathcal{N}}^{(0)}$ の終結式である．[13] において浮動小数係数に関する終結式計算に関する種々の数値実験が行われている（主変数の次数 8 次程度まで）．数値実験では，行列の選択について，行列式の展開方法について述べられている．数値実験より，一度に R および A, B を計算することは困難なので，本アルゴリズムでは，R を計算した後に A, B を計算する．また，疎な多項式の場合については検討されていないため少し触れる．

3.1.1 R の計算

本節では，Sylvester 行列を選択すべきか Bezout 行列を選択すべきか簡単な実験を行う．以下，疎な多項式の行列について述べる．

行列のサイズは，一般に Sylvester 行列より Bezout 行列の方が小さい．ただし，疎な多項式を入力とするため，行列自身が疎な行列であるか考える必要がある．

Sylvester 行列の場合，入力が疎であれば行列は係数を並べただけの行列のため疎になる．Bezout 行列の場合，入力の多項式から見積もることができる．

Bezout 行列の各要素 $b_{i,j}$ は

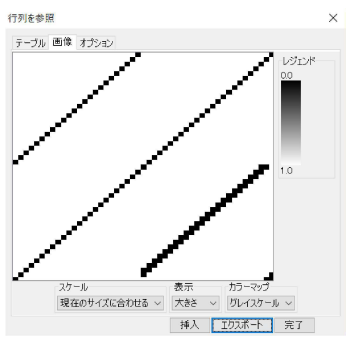
$$\frac{C_{\mathcal{N}}^{(0)}(x,\boldsymbol{u})D_{\mathcal{N}}^{(0)}(y,\boldsymbol{u}) - C_{\mathcal{N}}^{(0)}(y,\boldsymbol{u})D_{\mathcal{N}}^{(0)}(x,\boldsymbol{u})}{x-y} = \sum_{i,j} b_{i,j}x^i y^j$$

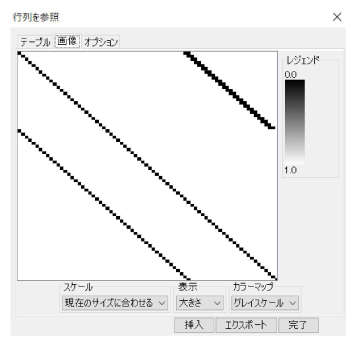
の $x^{i+1}y^{j+1}$-係数に対応する．各係数は $f_i g_j - f_j g_i$ の係数の和でかけるので，$f_i g_j - f_j g_i$ がどの程度 0 になるかでどの程度の疎な行列であるか判断することができる．例えば，$f_n g_j - f_j g_n \neq= 0 (n > j)$ であった場合は，$n-j$ 個は 0 でない要素が存在する．また，$f_{n-1} g_j - f_j g_{n-1} \neq= 0 (n-1 > j)$ の場合には，$n-j-2$ 個は 0 でない要素が存在する．

どの程度，疎な行列であるかの判定は可能であり，入力の多項式が高次と低次の項からなる場合（n/n^2 程度が 0 でない要素）は高次と中次の項からなる場合より疎でなくなることが簡単にわかる．

例 4 (疎な多項式の Bezout 行列)
次の多項式の *Sylveter* 行列（右図）と *Bezout* 行列（左図）は次のようになる（黒い部分は非零要素）

```
F(x,u,v,w)=(u+w)x^49 + ux^2 + 3wx,  G(x,u,v,w)=(u^25+v w^24)x^25 - v^2.
```





Bezout 行列に関しては，x^i の係数が *0* でない個数だけ帯があるような対称行列になっている．*Sylvester* 行列は対称にもなっていない．ただ，*2* つの行列でどちらが一層疎な行列であるかは判断できない．

いずれの行列の場合も疎な行列に見えるため，大きなサイズでも行列式は計算できると考えられる．実際に Maple2015.2 で終結式を計算すると，次のような結果が得られる．この例は整数係数のまま計算を実行した．

```
ff:=(u+w)*x^49 + u*x + 3*w*x;
gg:=(u^48+v*w^48)*x^4 -x^3- v;
t:=time(): resultant(ff,gg,x): time()-t;
                                0.469
```

すぐに結果が返ってくる（入力は高次と低次の多項式の組み合わせ）．この例において，浮動小数係数に変換し同様の計算を実行すると計算が異常終了する．

```
t:=time(): resultant(ff,gg,x): time()-t;
          hungup     %% 誤差（完全誤差項）のため計算できない
```

Maple のヘルプによると，終結式の計算は Bezout 行列の小行列式展開によって実行されると書いてあるが，それでも計算ができないようであったが，行列式展開を小行列式展開によって実行するよう明示的にコマンドを実行するとほぼ同様に実行時間で有効な結果が得られた．

ゆえに，入力がそれぞれ高次と低次の多項式であれば Bezout 行列の行列式は小行列式展開で計算できることが確認できた．

注意 3
入力がそれぞれ高次・高次の疎な多項式の場合，行列式の項数が多すぎるため計算できない．この場合は別の方法を検討する必要がある．

3.1.2 A, B の計算

$(\tilde{A}+kD_{\mathcal{N}}^{(0)})C_{\mathcal{N}}^{(0)}+(\tilde{B}-kC_{\mathcal{N}}^{(0)})D_{\mathcal{N}}^{(0)}=R$ が成り立つので，$\tilde{A}'C_{\mathcal{N}}^{(0)}+\tilde{B}'D_{\mathcal{N}}^{(0)}=R$ を満たす $\tilde{A}', \tilde{B}'$ を一つ求めたい．

$A'C_{\mathcal{N}}^{(0)}=R+kD_{\mathcal{N}}^{(0)}$ をみたす $k\in\mathbb{K}[\boldsymbol{u}][x]$ が存在するので，低次項から係数比較することにより k が一つ得られる．それから $C_{\mathcal{N}}^{(0)}$ で割ると，A' の一般形が得られ，B' も計算することができる．ゆえに，A' と B' から次数条件を満たすよう $D_{\mathcal{N}}^{(0)}$ または $C_{\mathcal{N}}^{(0)}$ で除算することによって，A, B が計算できる．

3.2 各 Hensel 因子 $\delta C^{(k+1)}, \delta D^{(k+1)}$ の構成における効率化

一般に拡張 Hensel 構成において，各計算ステップにおいて Hensel 因子は有理関するになる．ただし，GCD 計算においては GCD が多項式なので Hensel 因子も多項式になる．$\delta H^{(k+1)} \equiv H-(C^{(k-1)}+\delta C^{(k)})(D^{(k-1)}+\delta D^{(k)})$ に対して

$$\delta C^{(k+1)}=\sum_j \delta h_j^{(k+1)} W_C^{(j)}, \quad \delta D^{(k+1)}=\sum_j \delta h_j^{(k+1)} W_D^{(j)}$$

であるが実際の計算では,

$$W_C^{(j)} = \frac{\tilde{A}_j}{R} \text{ and } W_D^{(j)} = \frac{\tilde{B}_j}{R}$$

なので、

1. 積和を計算

$$\delta\tilde{C}^{(k+1)} = \sum_j \delta h_j^{(k+1)} \tilde{B}_j, \quad \delta\tilde{D}^{(k+1)} = \sum_j \delta h_j^{(k+1)} \tilde{A}_j$$

2. 除算（ここがボトルネック）

$$\delta C^{(k+1)} = \mathrm{quo}(\tilde{C}^{(k+1)}, R) \quad \delta D^{(k+1)} = \mathrm{quo}(\tilde{D}^{(k+1)}, R)$$

の手順で計算を行う．斉時多項式同士の除算が発生する．R の項数が多い時，$\tilde{A}_j, \tilde{B}_j$ も項数が多いためできる限り余分な計算は避けたい．

商の計算を高速に行う方法として，多項式の掛け算と同様の FFT を利用する方法があるが，$\tilde{A}_j, \tilde{B}_j$ も項数が多いため除算を行うまでに多くの計算をする必要がある．斉時多項式同士の除算であり，商の全次数もあらかじめわかっているため必要な項だけを利用して計算することが可能である．

例 5

$$\frac{u^5 + u^4v - 2u^3v^2 - 3u^2v^3 - 3uv^4}{u^2 - 3v^2} = u^3 + u^2v + uv^2$$

の左辺を計算する際に，全ての係数は必要ない．分子の始めの 3 つだけあれば十分である（項順序の高い順）．$\tilde{A}_j, \tilde{B}_j$ においても項順序の（除算に必要な）高いところだけピックアップしておけば，全体の計算を早くすることができる．

4 まとめ

各 Hensel 因子の構成における効率化を斉時多項式の除算の効率化によって検討を行った．浮動小数係数の多項式同士の終結式計算において，精度を落とさないための工夫を中心に検討を行ったが疎な多変数多項式の終結式であれば sparse resultant の適応など考えられるが，アルゴリズム自身は整数係数をメインの場合もあり導入の際には数値実験を行う必要があると考えられる．

また，Moses-Yun 補間式を有理式が表れない方法で導出する方法がある [10]．有理式が表れないため効率化が期待できる．実装は今後の課題である．

参 考 文 献

[1] K.O. Geddes, S.R. Czapor and G. Labahn: *Algorithms for computer algebra.* Kluwer Academic Publishers, 1992.

[2] J. von zur Gathen and J. Gerhard: *Modern Computer Algebra.* Cambridge Univ. Press, 1999.

[3] D. Inaba: Factorization of multivariate polynomials by extended Hensel construction. *ACM SIGSAM Bulletin*, **39**(1), 2-14 (2005).

[4] E. Kaltofen: Sparse Hensel lifting. *Proc. EUROCAL'85*, Springer-Verlag LNCS **2**, 4-17 (1985).

[5] T.-C. Kuo: Generalized Newton-Puiseux theory and Hensel's lemma in $\mathbf{C}[[x, y]]$. *Canad. J. Math.*, **XLI**, 1101-1116 (1989).

[6] J. Moses and D.Y.Y. Yun: The EZGCD algorithm. *Proc. 1973 ACM National Conference*, ACM, 159-166 (1973).

[7] F.K. Abu Salem, S. Gao and A.G.B. Lauder, Factoring polynomials via polytopes: *Proc. ISSAC'04*, ACM, 4-11 (2004).

[8] T. Sasaki and D. Inaba: Hensel construction of $F(x, u_1, \ldots, u_\ell)$, $\ell \geq 2$, at a singular point and its applications. *ACM SIGSAM Bulletin*, **34**(1), 9-17 (2000).

[9] M. Sanuki, D. Inaba and T. Sasaki: Computation of GCD of sparse multivariate polynomials by extended Hensel construction. *Computation of GCD of Sparse Multivariate Polynomials by Extended Hensel Construction*, 34-41 (2015).

[10] T. Sasaki and D. Inaba: Enhancing the extended Hensel construction by using Gröbner bases. *Proc. of CASC2016*, Springer, 457-472 (2016).

[11] T. Sasaki and F. Kako: Solving multivariate algebraic equation by Hensel construction. Preprint of Univ. Tsukuba, March, 1993.

[12] T. Sasaki and F. Kako: Solving multivariate algebraic equation by Hensel construction. *Japan J. Indust. Appl. Math.*, **16**(2), 257-285 (1999). (This is almost the same as [11]: the delay of publication is due to very slow reviewing process.)

[13] T. Sasaki and T. Sato: Cancellation errors in multivariate resultant computation with floating-point numbers. *ACM SIGSAM Bulletin* **32**(4), 13–20(1998)

[14] P.S. Wang and L. P. Rothschild: Factoring multivariate polynomials over the integers. Math. Comp. **29**, 935-950 (1975).

[15] P.S. Wang: Preserving sparseness in multivariate polynomial factorization. *Proc. 1977 MACSYMA Users Conference*, MIT, 55-61 (1977).

[16] P.S. Wang: An improved multivariate factoring algorithm. Math. Comp. **32**, 1215-1231 (1978).

[17] P.S. Wang: The EEZ-GCD algorithm. SIGSAM Bulletin **14**, 50–60 (1980).

[18] R. Zippel: Probabilistic algorithm for sparse polynomials. *Proc. EUROSAM'79*, Springer-Verlag LNCS **72**, 216-226 (1979).

[19] R. Zippel: Newton's iteration and the sparse Hensel lifting (extended abstract), *Proc. SYMSAC'81*, 68-72 (1981).