

ネーター作用素計算における モニックでない多項式による割り算の効率化

庄司卓夢

TAKUMU SHOJI

新潟大学自然科学研究科

GRADUATE SCHOOL OF SCIENCE AND TECHNOLOGY, NIIGATA UNIVERSITY

田島慎一

SHINICHI TAJIMA

新潟大学工学部情報工学科

FACULTY OF ENGINEERING, NIIGATA UNIVERSITY

Abstract

本稿で取り上げる問題は, Asir 上での計算機実験の中で見つかった問題である. ネーター作用素は, 有理関数の特異性を記述できる微分作用素であり, 極におけるローラン展開や留数とダイレクトに関連している. その具体的な計算アルゴリズムは論文 ([1]) で与えた. そのアルゴリズムでは, 有理関数の分母の各既約因子による割り算が何度も必要になるが, 割る多項式がモニックでない場合, その剰余の多項式は有理数係数となるため, モニックな場合に比べて, 計算が遅くなるという弱点があった. この問題点を克服するための 1 つの解決法を提示する.

1 ネーター作用素計算アルゴリズム

本稿の内容は, 論文 ([1]) で与えた留数用のネーター作用素計算アルゴリズムの部分的な効率化がテーマである. 改良部分だけを説明してもわかりにくいので, まずアルゴリズムの概要を再説する.

$f_1(x), \dots, f_m(x)$ を $\mathbb{Q}$ 上で既約な多項式, $h(x) \in \mathbb{Q}[x]$, $\ell_1, \dots, \ell_m \in \mathbb{N}$ とし, 次の有理関数

$$\frac{h(x)}{f_1(x)^{\ell_1} \cdots f_m(x)^{\ell_m}}$$

が定める代数的局所コホモロジー類

$$\left[\frac{h(x)}{f_1(x)^{\ell_1} \cdots f_m(x)^{\ell_m}} \right]$$

のネーター作用素表示

$$\left[\frac{h(x)}{f_1(x)^{\ell_1} \cdots f_m(x)^{\ell_m}} \right] = T_1 \left[\frac{f'_1(x)}{f_1(x)} \right] + \cdots + T_m \left[\frac{f'_m(x)}{f_m(x)} \right]$$

を求める問題を考える. これらの $T_1, \dots, T_m$ を有理関数の分母の既約因子 $f_1(x), \dots, f_m(x)$ に対応するネーター作用素と呼ぶ.

ここで、各 T_k は、

$$T_k = \left(-\frac{d}{dx}\right)^{\ell_k-1} t_0(x) + \left(-\frac{d}{dx}\right)^{\ell_k-2} t_1(x) + \cdots + \left(-\frac{d}{dx}\right) t_{\ell_k-2}(x) + t_{\ell_k-1}(x) \quad (1)$$

なる $\ell_k - 1$ 階の微分作用素である。

$\left(-\frac{d}{dx}\right)^{\ell-1-i} t_i(x)$ のように、微分作用素にマイナスをつけ、多項式を微分作用素の左側でなく右側につけて表現しているが、これは意図的にそうしている。この形の方が、ローラン展開との関係やネーター作用素の形式的随伴作用素を考える際に便利であるだけでなく、計算効率の面でもよいのである。なお、 $t_0(x), \dots, t_{\ell_k-1}(x)$ は、多項式ではなく、多項式倍という写像（零階の微分作用素）である。

今、有理関数の分母の k 番目の既約因子 $f_k(x)$ に対応するネーター作用素 T_k を求める問題を考える。記号を見やすくするため、以下、 $T = T_k$, $f(x) = f_k(x)$, $\ell = \ell_k$, $a(x) = f_1(x) \cdots f_{k-1}(x) f_{k+1}(x) \cdots f_m(x)$, $g(x) = f_1(x)^{\ell_1} \cdots f_{k-1}(x)^{\ell_{k-1}} f_{k+1}(x)^{\ell_{k+1}} \cdots f_m(x)^{\ell_m}$ と表すことにする。

(1) の各係数 $t_0(x), \dots, t_{\ell-1}(x)$ は、 $t_0(x) = (f'(x)^\ell g(x))^{-1}$ を共通因子にもつことが知られているので、まずそれで微分作用素全体をくくると、(1) は

$$T = \left\{ \left(-\frac{d}{dx}\right)^{\ell-1} s_0(x) + \left(-\frac{d}{dx}\right)^{\ell-2} s_1(x) + \cdots + \left(-\frac{d}{dx}\right) s_{\ell-2}(x) + s_{\ell-1}(x) \right\} t_0(x) \quad (2)$$

となる。 $t_0(x)$ は係数が複雑な多項式なので、その分 $s_0(x), \dots, s_{\ell-1}(x)$ はかなり軽くなっている。

$s_0(x), \dots, s_{\ell-1}(x)$ は、 $f'(x)a(x)$ の逆冪を因子にもつことが知られている。 $s_1(x)$ は $(f'(x)a(x))^{-1}$ を因子にもち、 $s_2(x)$ は $(f'(x)a(x))^{-2}$ を因子にもち、以下同様に、 $s_{\ell-1}(x)$ は $(f'(x)a(x))^{-(\ell-1)}$ を因子にもつ。そこで (2) を、

$$T = \left\{ \left(-\frac{d}{dx}\right)^{\ell-1} (f'(x)a(x))^{\ell-1} c_0(x) + \left(-\frac{d}{dx}\right)^{\ell-2} (f'(x)a(x))^{\ell-2} c_1(x) + \cdots + \left(-\frac{d}{dx}\right) (f'(x)a(x)) c_{\ell-2}(x) + c_{\ell-1}(x) \right\} t_0(x) (f'(x)a(x))^{-(\ell-1)} \quad (3)$$

と式変形する。見やすいように、 $B_i = \left(-\frac{d}{dx}\right)^i (f'(x)a(x))^i$, $u(x) = t_0(x) (f'(x)a(x))^{-(\ell-1)}$ とおくと、(3) は、

$$T = \{ B_{\ell-1} c_0(x) + B_{\ell-2} c_1(x) + \cdots + B_1 c_{\ell-2}(x) + B_0 c_{\ell-1}(x) \} u(x) \quad (4)$$

となる。逆冪は有理数係数の複雑な多項式となるため、その分 $c_0(x), \dots, c_{\ell-1}(x)$ は軽くなっている。

$c_0(x), \dots, c_{\ell-1}(x)$ は次の漸化式により逐次的に計算できる。 $p(x) = -f_1(x)f_2(x) \cdots f_m(x)$, $q(x) = \ell_1 f_1'(x)f_2(x) \cdots f_m(x) + \ell_2 f_1(x)f_2'(x) \cdots f_m(x) + \cdots + \ell_m f_1(x)f_2(x) \cdots f_m'(x)$ とおくと、 $i = 0, \dots, \ell-1$ に対して、

$$c_0(x) = 1, \\ c_i(x) = -\frac{1}{i} \sum_{j=1}^i \left\{ \binom{\ell-i+j}{j+1} p^{(j+1)}(x) + \binom{\ell-1-i+j}{j} q^{(j)}(x) \right\} (f'(x)a(x))^{j-1} c_{i-j}(x). \quad (5)$$

この漸化式を見てもわかるように、 $c_1(x)$ を求めるには $c_0(x)$ を使い、 $c_2(x)$ を求めるには $c_0(x), c_1(x)$ を使う。以下同様に、 $c_{\ell-1}(x)$ を求めるには $c_0(x), \dots, c_{\ell-2}(x)$ を使う。従って、先に求まるデータのサイズが小さい程、後に求まるデータの計算が楽になる。上述した共通因子のくくり出しは、(1) をそのまま求める方法と比べて、計算効率を向上させる上で有効であることがわかる。

ここからが本稿のテーマに深く関係する話である。 $c_0(x), \dots, c_{\ell-1}(x)$ は、求める過程で、剰余体 $\mathbb{Q}[x]/\langle f(x) \rangle$ 上で四則演算が行えるという性質がある。この性質により、計算過程で、次数が $\deg f$ 以上になった場合、 $f(x)$ で割り算を行い剰余を求めることで、常に次数を $\deg f - 1$ 次以下に保つことができる。その結果、計算は劇的に高速化される。しかしその一方で、 $f(x)$ がモニックでない場合には、剰余の多項式が有理数係数

となるために Asir の約分 (gcd) 計算と通分 (lcm) 計算が頻繁に発生し、計算が重くなる問題が生じる。これを本稿では、「モニックでない多項式による割り算の副作用 (以下、単に副作用)」と呼ぶことにする。この副作用を軽減することが本稿のテーマである。

例 1.1 有理関数

$$\frac{1}{(x^6 - x + 3)^2(7x^3 + x^2 - 2x + 2)^5}$$

のネーター作用素を求める問題を考える。

この有理関数が定める代数的局所コホモロジー類のネーター作用素表示は、 $f_1(x) = x^6 - x + 3$, $f_2(x) = 7x^3 + x^2 - 2x + 2$ とおくと、

$$\left[\frac{1}{f_1(x)^2 f_2(x)^5}\right] = T_1 \left[\frac{f'_1(x)}{f_1(x)}\right] + T_2 \left[\frac{f'_2(x)}{f_2(x)}\right]$$

である。

分母は 2 つの因子をもつが、ここではモニックでない方の因子 $f_2(x)$ に注目する。記号を

$$\begin{aligned} f(x) &= f_2(x) = 7x^3 + x^2 - 2x + 2, \\ f'(x) &= 21x^2 + 2x - 2, \\ a(x) &= f_1(x) = x^6 - x + 3, \\ g(x) &= f_1(x)^2 = (x^6 - x + 3)^2 \\ T &= T_2 \end{aligned}$$

と表すことにする。

この注目因子 $f(x)$ に対応するネーター作用素は、

$$T = \{B_4 c_0(x) + B_3 c_1(x) + B_2 c_2(x) + B_1 c_3(x) + B_0 c_4(x)\} u(x)$$

なる形をしている。

ここで、微分作用素 $B_4, \dots, B_0$ は次のように与えられる。

$$\begin{aligned} f'(x)a(x) &= (21x^2 + 2x - 2)(x^6 - x + 3) \pmod{7x^3 + x^2 - 2x + 2} \\ &= \frac{1}{7^5}(1128849x^2 + 5038x + 16802) \end{aligned}$$

$$B_0 = 1$$

$$\begin{aligned} B_1 &= \left(-\frac{d}{dx}\right)(f'(x)a(x)) \\ &= \left(-\frac{d}{dx}\right)\left(\frac{1}{7^5}(1128849x^2 + 5038x + 16802)\right) \end{aligned}$$

$$\begin{aligned} B_2 &= \left(-\frac{d}{dx}\right)^2(f'(x)a(x))^2 \\ &= \left(-\frac{d}{dx}\right)^2\left(\frac{1}{7^{12}}(20894882933107x^2 - 20221265530832x + 2403193227262)\right) \end{aligned}$$

$$\begin{aligned} B_3 &= \left(-\frac{d}{dx}\right)^3(f'(x)a(x))^3 \\ &= \left(-\frac{d}{dx}\right)^3\left(\frac{1}{7^{19}}(657998048809565280923x^2 - 711550370990973543808x + 367253696927888667990)\right) \end{aligned}$$

$$\begin{aligned}
B_4 &= \left(-\frac{d}{dx}\right)^4 (f'(x)a(x))^4 \\
&= \left(-\frac{d}{dx}\right)^4 \left(\frac{1}{726} (37421330292569046014853648379x^2 - 23578495073635594066150630288x \right. \\
&\quad \left. + 12986771140995011457151703126)\right)
\end{aligned}$$

$u(x)$ は、次の逆算計算で求められる.

$$\begin{aligned}
u(x) &= (f'(x))^{-5} (g(x))^{-1} (f'(x)a(x))^{-4} \bmod \langle f(x) \rangle \\
&= (21x^2 + 2x - 2)^{-9} (x^6 - x + 3)^{-6} \bmod \langle 7x^3 + x^2 - 2x + 2 \rangle \\
&= \frac{1}{(17)^6 (23)^6 (409)^6 (11334283)^2} (911231313103017791200661096401218968441x^2 \\
&\quad + 131656892697912134004021478031654433007x \\
&\quad - 559954382226892127053138867846579608128)
\end{aligned}$$

本稿で中心的に扱う計算部分は、(5) の漸化式である. この漸化式で $c_0(x), \dots, c_4(x)$ を求めると,

$$\begin{aligned}
c_0(x) &= 1 \\
c_1(x) &= \frac{1}{7^4} (2027316x^2 - 3395624x - 54620) \\
c_2(x) &= \frac{1}{7^{10}} (263204780161860x^2 - 85931976549912x + 74943002879916) \\
c_3(x) &= \frac{1}{7^{16}} (5936011123298611805880x^2 - 5982874458822140668368x \\
&\quad + 2611284323721531371976) \\
c_4(x) &= \frac{1}{7^{22}} (107791902802055995065394546320x^2 - 68409917959568499544955861520x \\
&\quad + 30339589970097307513358170320)
\end{aligned}$$

となる.

$7x^3 + x^2 - 2x + 2$ で割り算を行うことにより、次数が 2 次以下に抑えられていることがわかる. しかし割る多項式 (イデアル) がモニックでないため、その副作用として、有理数係数となってしまう.

2 副作用の対処法

この節では、前節で述べた副作用に対し、具体例を通じて、その対処法を述べる。

例 2.1 2つの剰余体 $\mathbb{Q}[x]/\langle x^3 - x - 1 \rangle$ と $\mathbb{Q}[x]/\langle 3x^3 - x - 1 \rangle$ において、 x^{30} をイデアルで割った剰余を求める問題を考える。

この2つの剰余体における割り算にかかる時間は、前者の方が速い。その理由は、これまで述べてきた通りである。実際に計算を行うと、

```
[0] srem(x^30,x^3-x-1);
1081*x^2+1432*x+816
```

$$x^{30} \equiv 1081x^2 + 1432x + 816 \pmod{\langle x^3 - x - 1 \rangle}$$

```
[1] srem(x^30,3*x^3-x-1);
20809/4782969*x^2+17720/4782969*x+8152/4782969
```

$$x^{30} \equiv \frac{20809}{4782969}x^2 + \frac{17720}{4782969}x + \frac{8152}{4782969} \pmod{\langle 3x^3 - x - 1 \rangle}$$

となる。後者は有理数係数となり、前者より複雑な多項式となることがわかる。イデアルがモニックであるかモニックでないかは、計算効率の面で、大きな違いである。

対処法 1

この問題を解決する方法として、Asir の Dalg という機能を使うことが考えられる。上記の例なら、

```
[101] Alg = newalg(3*x^3-x-1);
(#0)
[102] set_field([Alg]);
0
[103] algtoalg(subst(x^30,x,Alg));
((20809)*<<2>>+(17720)*<<1>>+(8152)*<<0>>)/4782969
```

といったように自動的に整数係数化してくれる。この機能を使えば、単純に srem を使うよりもずっと速くなるだけでなく、各係数部分を通分した値も得られるという利点がある。しかし、この機能は汎用的な機能であるため、本稿で扱うような特殊な問題に対しては、ロスが生じると思われる。

対処法 2

一時は対処法 1 を採用していたが、後になって次の方法を考えることにした。本研究では、

- 割る多項式は常に $f(x)$ で一定
- 割り算の回数は数えられる

という2つの事実に注目し、割り算によって生じる有理数の分母の整数を見積もって、あらかじめかけておく方法を考える。

多項式による割り算の回数は、高々「割られる多項式の次数 - 割る多項式の次数 + 1」回である。高々と書いた理由は、実際はこの上限より割り算の回数が減る場合があるからである。それは、割られる多項式が

スパースで、かつ割る多項式の最高次数と2番目に高い次数の項との次数差が2以上の場合などに起きる。正確に割り算の回数を数えるには、割られる多項式と割る多項式の項の情報も見なければならない。

例 2.1 でいうと、割り算の回数は、高々

$$\deg(x^{30}) - \deg(3x^3 - x - 1) + 1 = 28$$

回であることがわかるが、割られる多項式はスパースであり、割る多項式も x^2 の項がないため割り算の回数はその半分の14回である。従って、 3^{14} をあらかじめかけておけば剰余の多項式の係数に有理数はでてこない。

```
[259] srem(3^14*x^30, 3*x^3-x-1);
```

```
20809*x^2+17720*x+8152
```

```
[260] QQ/(3^14);
```

```
20809/4782969*x^2+17720/4782969*x+8152/4782969
```

正しい答えを得るには、割り算を行った後で、かけておいた数で割り直さなければならないが、この計算は一瞬である。というのは、プログラムでは、有理数係数の多項式を、整数係数化した多項式と、係数部分の通分値とに分離して計算を進行させているため、かけておいた数での割り直しの計算は、分母の整数同士の積であり、gcd を計算するわけではないからである。

3 副作用を軽減するネーター作用素計算アルゴリズム

実際に、留数用のネーター作用素計算アルゴリズムの場合で考える。副作用が最も大きく影響する計算パートは、(5) の漸化式の計算であるので、この部分を中心に考える。割り算の回数を数えるために、各係数 $c_0(x), \dots, c_{\ell-1}(x)$ の次数に注目する。見やすいように、(5) の漸化式における全体の $-\frac{1}{i}$ と $p^{(j+1)}$ と $q^{(j)}$ の係数部分を取り去って書き下すと、

$$\begin{aligned} c_0 &\simeq 1 \\ c_1 &\simeq (p^{(2)} + q^{(1)})c_0 \\ c_2 &\simeq (p^{(2)} + q^{(1)})c_1 + (p^{(3)} + q^{(2)})(f'a)c_0 \\ c_3 &\simeq (p^{(2)} + q^{(1)})c_2 + (p^{(3)} + q^{(2)})(f'a)c_1 + (p^{(4)} + q^{(3)})(f'a)^2c_0 \\ c_4 &\simeq (p^{(2)} + q^{(1)})c_3 + (p^{(3)} + q^{(2)})(f'a)c_2 + (p^{(4)} + q^{(3)})(f'a)^2c_1 + (p^{(5)} + q^{(4)})(f'a)^3c_0 \\ &\vdots \\ c_i &\simeq \sum_{j=1}^i (p^{(j+1)} + q^{(j)})(f'a)^{j-1}c_{i-j} \\ &\vdots \\ c_{\ell-1} &\simeq \sum_{j=1}^{\ell-1} (p^{(j+1)} + q^{(j)})(f'a)^{j-1}c_{\ell-1-j} \end{aligned}$$

となる。

$p(x)$ は有理関数の分母の無平方部分であり、次数は $\deg p = \deg f_1 + \dots + \deg f_m$ である。 $q(x)$ の次数は、常に $\deg q = \deg p - 1$ である。従って、 $(p^{(j+1)} + q^{(j)})$ の次数は、高々 $\deg p - 2$ である。また、 $f'(x)a(x)$ の次数は、 $\deg(f'a) = \deg p - 1$ である。これを元に、各係数 $c_0(x), \dots, c_{\ell-1}(x)$ の次数を求めていく。

まず, c_0 の次数は 0 である. c_1 の次数は $\deg p - 2$ である. c_2 の次数は, $(p^{(2)} + q^{(1)})c_1$ と $(p^{(3)} + q^{(2)})(f'a)c_0$ の高い方の次数で決まるが, どちらも次数が $2(\deg p - 2)$ であるから, $\deg c_2 = 2(\deg p - 2)$ である. c_3 の次数は, $(p^{(2)} + q^{(1)})c_2$ と $(p^{(3)} + q^{(2)})(f'a)c_1$ と $(p^{(4)} + q^{(3)})(f'a)^2c_0$ の中で最も次数が高いもので決まるが, 3 つとも次数は $3(\deg p - 2)$ であるから, $\deg c_3 = 3(\deg p - 2)$ である. c_4 の次数も同様に考えると, $\deg c_4 = 4(\deg p - 2)$ とわかる. 以下同様に, $\deg c_{\ell-1} = (\ell - 1)(\deg p - 2)$ である. まとめると,

$$\begin{aligned} \deg c_0 &= 0 \\ \deg c_1 &= \deg p - 2 \\ \deg c_2 &= 2(\deg p - 2) \\ \deg c_3 &= 3(\deg p - 2) \\ \deg c_4 &= 4(\deg p - 2) \\ &\vdots \\ \deg c_i &= i(\deg p - 2) \\ &\vdots \\ \deg c_{\ell-1} &= (\ell - 1)(\deg p - 2) \end{aligned}$$

となる.

3.1 見積もり法 1

$f(x)$ の最高次の係数を CF とすると, $f(x)$ で割った剰余の多項式の係数の分母にでてくる数は, CF の冪である. この CF の冪が最も大きくなるのは, 最も次数が高い多項式 $c_{\ell-1}(x)$ を $f(x)$ で割ったときである. その時の割り算の回数を s とすると,

$$\begin{aligned} s &= \deg c_{\ell-1} - \deg f + 1 \\ &= (\ell - 1)(\deg p - 2) - \deg f + 1 \end{aligned}$$

であるから, あらかじめかけておくべき整数の上限は CF^s と見積もれる.

ところで, 漸化式を見ればわかるように, $c_0(x), \dots, c_{\ell-1}(x)$ は, $c_0(x)$ を共通因子にもつ構造になっている. $c_0(x)$ を CF^s 倍してから漸化式を計算していくことは, 既に求まってある $c_0(x), \dots, c_{\ell-1}(x)$ を均等に CF^s 倍する (言い換えれば, 微分作用素 T を CF^s 倍する) ことと同じである. 従って, あらかじめ $c_0(x)$ を CF^s 倍しておけば, 漸化式の計算において, $c_0(x), \dots, c_{\ell-1}(x)$ の係数に有理数はでてこないことになる. 後は, 正しい答えを得るために後で CF^s で割り直せばよい.

3.2 見積もり法 2

見積もり法 1 では, 最初に上限分の整数をかけ, 漸化式を逐次的に計算していくにつれて, かけておいた整数と係数部分がキャンセルされて小さくなっていく. この細工で確かに有理数はでてこなくなるが, 最初に大きな数をかけているので冗長な計算をしている. そこで, 漸化式を 1 つ 1 つ細かくケアしていくことを考える.

上記を見てもわかるように $c_0(x), \dots, c_{\ell-1}(x)$ の次数は比例して上がっていく. そこで隣接 2 項間の次数の差分に注目し

$$L = \text{CF}^{\deg p - 2}$$

とおく. $d_i(x) = L^i c_i(x)$ とおき, 漸化式を

$$c_0(x) = d_0(x), \quad c_1(x) = \frac{d_1(x)}{L}, \quad \dots, \quad c_{\ell-1}(x) = \frac{d_{\ell-1}(x)}{L^{\ell-1}}$$

と置き換えて考えると,

$$\begin{aligned} d_0 &\simeq 1 \\ d_1 &\simeq L(p^{(2)} + q^{(1)})d_0 \\ d_2 &\simeq L(p^{(2)} + q^{(1)})d_1 + L(p^{(3)} + q^{(2)})L(f'a)d_0 \\ d_3 &\simeq L(p^{(2)} + q^{(1)})d_2 + L(p^{(3)} + q^{(2)})L(f'a)d_1 + L(p^{(4)} + q^{(3)})L^2(f'a)^2d_0 \\ &\vdots \\ d_i &\simeq \sum_{j=1}^i L(p^{(j+1)} + q^{(j)})(Lf'a)^{j-1}d_{i-j} \\ &\vdots \\ d_{\ell-1} &\simeq \sum_{j=1}^{\ell-1} L(p^{(j+1)} + q^{(j)})(Lf'a)^{j-1}d_{\ell-1-j} \end{aligned}$$

と書ける. ここで $L(p^{(j+1)} + q^{(j)})(Lf'a)^{j-1}$ は j のみで変化する値であるので, i のループの外側であらかじめ計算しておく. 特に, $j = 0, \dots, \ell-1$ に対して,

$$b_j(x) = (Lf'(x)a(x))^j \pmod{\langle f(x) \rangle}$$

と計算しておく. このデータはネーター作用素の形式的随伴作用素を分子に施す計算の際にも必要になる.

このとき, ネーター作用素は,

$$\begin{aligned} T &= \{B_{\ell-1}c_0(x) + B_{\ell-2}c_1(x) + \dots + B_1c_{\ell-2}(x) + B_0c_{\ell-1}(x)\}u(x) \\ &= \{B_{\ell-1}d_0(x) + B_{\ell-2}\frac{d_1(x)}{L} + \dots + B_1\frac{d_{\ell-2}(x)}{L^{\ell-2}} + B_0\frac{d_{\ell-1}(x)}{L^{\ell-1}}\}u(x) \\ &= \{(-\frac{d}{dx})^{\ell-1}\frac{b_{\ell-1}(x)}{L^{\ell-1}}d_0(x) + (-\frac{d}{dx})^{\ell-2}\frac{b_{\ell-2}(x)}{L^{\ell-2}}\frac{d_1(x)}{L} \\ &\quad + \dots + (-\frac{d}{dx})\frac{b_1(x)}{L}\frac{d_{\ell-2}(x)}{L^{\ell-2}} + b_0(x)\frac{d_{\ell-1}(x)}{L^{\ell-1}}\}u(x) \\ &= \{(-\frac{d}{dx})^{\ell-1}b_{\ell-1}(x) + (-\frac{d}{dx})^{\ell-2}b_{\ell-2}(x)d_1(x) + \dots + (-\frac{d}{dx})b_1(x)d_{\ell-2}(x) + d_{\ell-1}(x)\}\frac{u(x)}{L^{\ell-1}} \end{aligned}$$

となっている. 従って, 正しい答えを得るには後で $L^{\ell-1}$ で割り直せばよい.

3.3 さらに細工

$f'(x)$ を因子にもつ多項式を $f(x)$ で割った場合, 1 回目の割り算で, その剰余の多項式の最高次の係数に有理数はでてこないことになる. これを利用すれば, 若干ながら見積りの精度を上げられる. また, 前節で述べたように, 割られる多項式と割る多項式の項の情報によって割り算の回数は激減する場合がある. これを利用すれば, さらに見積りの精度を上げられる.

4 計算機実験

次の5通りの方法

方法 1	そのまま剰余をとる方法
方法 2	Asir の Dalg を使う方法
方法 3	見積もり法 1 (最初にかけておく方法)
方法 4(a)	見積もり法 2 (細かくケアしていく方法)
方法 4(b)	見積もり法 2 の改良 (割る多項式の項の情報にも注目し見積もりの精度を上げる)
参考	注目因子の最高次の係数を 1 にした場合

に基づくアルゴリズムを Asir 上に実装し, 副作用が最も影響する計算である (5) の漸化式の計算時間を計る.

ネーター作用素を求める計算において, 計算効率に影響を与える要素としては, 分子多項式の複雑さ, 注目する分母の既約因子の次数と項数, 注目する分母の既約因子の重複度, 注目する分母の既約因子以外の因子の複雑さ, 注目する分母の既約因子の係数の大きさ, などが挙げられる. しかしこの中で, 分子多項式の複雑さは (5) の漸化式の計算には関係しないのでここでは考察から除外する.

実験環境

CPU 1.30GHz, 248MB RAM.

実験データ

実験は, 次の 5 つの有理関数を使って行う.

- case1

$$\frac{1}{(x^3 + 2x - 4)^4(x^2 - 3x + 5)^5(17x^3 - x - 1)^{200}}$$

注目因子 $17x^3 - x - 1$.

- case2

$$\frac{1}{(x^3 + 2x - 4)^4(x^2 - 3x + 5)^5(499x^3 - x - 1)^{200}}$$

注目因子 $499x^3 - x - 1$. case1 より最高次の係数が大きい.

- case3

$$\frac{1}{(x^3 + 2x - 4)^4(x^2 - 3x + 5)^5(17x^3 - x - 1)^{500}}$$

注目因子 $17x^3 - x - 1$. case1 より重複度が高い.

- case4

$$\frac{1}{(x^3 + 2x - 4)^4(x^2 - 3x + 5)^5(7x^5 + 2x^4 - x^3 + 2x^2 + 3x - 7)^{200}}$$

注目因子 $7x^5 + 2x^4 - x^3 + 2x^2 + 3x - 7$. case1 より多項式の次数が高く項数も多い.

- case5

$$\frac{1}{(x^{10} + 3x^7 - x^3 - x^2 + 2x - 3)^5(x^5 + 2x^4 + 2x^3 - 3x + 5)^4(x^4 + 1)^4(17x^3 - x - 1)^{200}}$$

注目因子 $17x^3 - x - 1$. case1 より注目因子以外の因子が複雑.

上記の 5 つの case における (5) の漸化式の計算時間 (秒)

	case1	case2	case3	case4	case5
方法 1	2.94423	7.93140	1 分以上	33.0876	1 分以上
方法 2	2.17312	4.20605	26.7084	7.08018	18.8571
方法 3	0.83120	2.15310	6.44927	7.63097	28.1104
方法 4 (a)	0.18026	0.36059	1.07154	0.66095	3.01433
方法 4 (b)	0.13019	0.25036	0.81117	0.66095	1.34193
参考	0.05007	0.05007	0.30043	0.26037	0.29042

本提案手法は, 方法 3 と方法 4 であるが, 方法 4 の方がずっと速いことがわかる.

case1,2,3,5 は, 注目因子に x^2 の項がないために, 割り算の回数は上限より減っている. その分も見積もったのが方法 4(b) であり, 方法 4(a) に比べて速くなっている. 特にこの細工は, case5 のように注目因子以外の因子が, 次数が高くかつスパースな場合に威力を発揮する.

ちなみに, 方法 1 と参考を比較すると, 副作用の大きさがうかがえる. 参考のタイミングデータを限界値と考えれば, 今回の改良で副作用を大分軽減できたといえる.

5 まとめ

本稿は, 留数用のネーター作用素計算アルゴリズムにおける, モニックでない多項式による割り算の副作用の軽減をテーマに挙げ, その解決法を提示した. そして実際にプログラムも改良した. 計算機実験の結果, 本提案手法は, 計算の効率化に有効であることがわかった. 本稿では触れなかったが, ローラン展開用のネーター作用素計算アルゴリズムにも今後同様の効率化を加えたい.

参 考 文 献

- [1] 庄司卓夢, 田島慎一, 高速留数計算アルゴリズム,
京都大学数理解析研究所講究録 1456 「Computer Algebra-Design of Algorithms, Implementations and Applications」(2005), 133-143.