ISSN 1349-127X (Online)
ISSB 1349-1261 (Paper)
http://www.math.kobe-u.ac.jp/raj

Risa/Asir Journal

Volume 2 (2007)

Risa/Asir Journal will be a step to "Journal of Free Mathematical Software". We welcome not only articles related to Computer algebra system Risa/Asir, but also any articles on free mathematical software.

Please visit `http://www.math.kobe-u.ac.jp/raj`. All articles and attached software systems will be reviewed.

Risa/Asir Journal Vol 2 (2007), 1--32
http://www.math.kobe-u.ac.jp/raj

1変数代数的局所コホモロジー類に対する Risa/Asir 用パッケージ taji_alc

庄司卓夢, 田島慎一

まえがき

taji_alc は, 1 変数有理関数に関する諸計算 (部分分数分解, ローラン展開, 留数計算) やその応用 (剰余公式, 有理数係数の線形常微分方程式のコーシー問題) に対してこれまで設計してきたアルゴリズムを, Asir 言語でプログラム化し, Asir のモジュール構造を使ってパッケージ化したものである. 本稿はそのチュートリアルである. パッケージの使用法がその主な内容となっているが, 数学的な説明も少し含まれている.

taji_alc の関数は, 全て exact な計算であり, 計算は高速である. メインは留数計算であるが, それ以外にも例えば, フーリエ・ボレル変換など珍しい機能も扱っている.

数学的には, 代数的局所コホモロジー類とネーター作用素が重要な役割を果たしているが, アルゴリズムは剰余体上での多項式演算と微分演算のみで構成されている. ユーザ側からは見えないが, アルゴリズムはこれまでにない新しいものである. その辺りの数学的な詳細については, 参考文献を参照して欲しい.

目 次

1 taji_alc パッケージ

1.1 使い方

taji_alc の各関数を使うには, Asir を起動しパッケージを読み込む必要がある. コマンドラインで読み込む場合は,

```
import("taji_alc.rr");
```

と入力する.

1.2 主な関数

- taji_alc.cpfd
  有理関数の部分分数分解を求める.
- taji_alc.snoether
  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を求める.
- taji_alc.laurent_expansion
  有理関数の極におけるローラン展開の主要部の係数を求める.
- taji_alc.residue
  有理関数の極における留数を求める.
- taji_alc.invpow
  剰余体上での逆冪を求める.
- taji_alc.rem_formula
  多項式を与えたときの剰余公式を求める.
- taji_alc.solve_ode_cp
  有理数係数の線形常微分方程式のコーシー問題の解を求める.
- taji_alc.solve_ode_cp_ps
  有理数係数の線形常微分方程式のコーシー問題の特殊解を求める.
- taji_alc.fbt
  有理関数が定める代数的局所コホモロジー類のフーリエ・ボレル変換を行う.
- taji_alc.invfbt
  指数多項式の逆フーリエ・ボレル変換を行う.
- taji_alc.diff_rat
  有理関数の n 階の導関数を求める.

taji_alc の関数は有機的であり, モジュール内の関数を呼び出して計算を行う.

関数	呼び出す関数
cpfd	invpow
snoether	invpow
laurent_expansion	snoether
residue	invpow
invpow	標準ライブラリの"gr"と"sp"の諸機能
rem_formula	cpfd, snoether
solve_ode_cp	fbt, solve_ode_cp_ps
solve_ode_cp_ps	fbt, invfbt
fbt	residue
invfbt	diff_rat

1.3 数学的な詳細について

taji_alc の関数の内部的な情報 (アルゴリズムの構成法や数学的な理論背景) については, 次の文献に説明がある.

関数	文献
cpfd	[17]
snoether	[17, 18]
laurent_expansion	[17, 18]
residue	[18, 16]
invpow	[15]
rem_formula	[19]
solve_ode_cp	[14, 20]
solve_ode_cp_ps	[14, 20]
fbt	[1, 14, 20]
invfbt	[1, 14, 20]

論文 [2, 5] は, D 加群を用いた留数計算の研究の出発点となる論文である. 代数的局所コホモロジーの概念と微分方程式を用いることで留数計算が可能となることを示し, 論文 [3, 4] で留数値を計算するアルゴリズムを与えた. 現在採用しているアルゴリズムは, [2, 5] の結果にさらにネーター作用素の概念を用いて導出したものである. ネーター作用素の構成法は, 論文 [14] で 3 通りの方法を提示した. プログラムではその中で最も効率がよい方法を採用している.

2 用語の説明

2.1 整数係数化リスト

taji_alc で扱う有理関数の各演算は, 有理数係数多項式の四則演算が中心となるが, そこにはまず解決すべき問題がある.

それは何かと言うと, 例えば, 有理数係数多項式同士の積

$$(\frac{2}{3}x^2-\frac{1}{5}x+\frac{7}{8})(\frac{2}{7}x^2-\frac{1}{11}x+\frac{1}{3})$$

を計算したいとする. Asir では,

```
[0] (2/3*x^2-1/5*x+7/8)*(2/7*x^2-1/11*x+1/3);
4/21*x^4-136/1155*x^3+971/1980*x^2-193/1320*x+7/24
```

といったように, 各係数ごとに gcd を計算し, 約分を完全に行って結果を返す. しかし, 計算効率を考えるとこれは問題である. というのは, gcd は速い部類の計算であるものの, 頻繁に行われたり, 係数が大きくなったりすれば話は別になり, この gcd 計算がボトルネックとなりうるからである.

そこで, 次のような対策をとる.

$$\frac{2}{3}x^2-\frac{1}{5}x+\frac{7}{8}=\frac{1}{120}(80x^2-24x+105)$$

$$\frac{2}{7}x^2-\frac{1}{11}x+\frac{1}{3}=\frac{1}{231}(66x^2-21x+77)$$

といったように, lcm を計算し係数部分を通分する. こうすることで gcd の計算回数が減るだけでなく, プログラム内で途中の計算を主体的に操作しやすくなる. 係数が大きくなるが, gcd を毎回計算するよりは効率的である. taji_alc のプログラムでは, 整数係数多項式と分母の整数を 2 つのプログラム変数に分けて, 別々に計算を進行させる.

この考え方で上記の積を計算すると,

```
[11] 120*231;
27720
[12] (80*x^2-24*x+105)*(66*x^2-21*x+77);
5280*x^4-3264*x^3+13594*x^2-4053*x+8085
```

より,

$$\frac{1}{27720}(5280x^4-3264x^3+13594x^2-4053x+8085)$$

となる.

taji_alc のプログラムでは, 出力の際にはこれを

```
[5280*x^4-3264*x^3+13594*x^2-4053*x+8085,27720]
```

なるリストで表現する. これを **整数係数化リスト** と呼ぶことにする. プログラム間で有理数係数多項式を受け渡しする際には, この整数係数化リストを用いる.

2.2 代数的局所コホモロジー類

$f(x)$ を $\mathbb{Q}[x]$ 上で既約, $\ell \in \mathbb{N}$ とする. このとき, $Z = \{x | f(x) = 0\}$ に台をもつ代数的局所コホモロジー群 (algebraic local cohomology groups) は, ホモロジー代数の言葉では,

$$\mathrm{H}^1_{[Z]}(\mathbb{Q}[x]) = \lim_{\ell \to \infty} \mathrm{Ext}^1_{\mathbb{Q}[x]}(\mathbb{Q}[x]/\langle f(x)^\ell \rangle, \mathbb{Q}[x])$$

と定義される.

しかし, ここでは別の定義を採用する. $h(x) \in \mathbb{Q}[x]$ とし, $\frac{h(x)}{f(x)^\ell}$ なる形の有理関数の間に次のような同値関係

$$\frac{h_1(x)}{f(x)^{\ell_1}} \equiv \frac{h_2(x)}{f(x)^{\ell_2}} \Longleftrightarrow \frac{h_1(x)}{f(x)^{\ell_1}} - \frac{h_2(x)}{f(x)^{\ell_2}} \in \mathbb{Q}[x]$$

を入れる. この同値関係による $\frac{h(x)}{f^\ell(x)}$ の同値類を, $[\frac{h(x)}{f^\ell(x)}]$ で表す. 別の表現をするなら,

$$[\frac{h(x)}{f(x)^\ell}] = \frac{h(x)}{f(x)^\ell} + \mathbb{Q}[x]$$

と書いてもよい. この同値類は $\mathrm{H}^1_{[Z]}(\mathbb{Q}[x])$ の元とみなせるので, これを有理関数 $\frac{h(x)}{f(x)^\ell}$ が定める代数的局所コホモロジー類と呼ぶ. これは, 有理関数の特異性に注目した同値類 (極でローラン展開した時に主要部が一致するような有理関数の集合) である.

例 2.1

(1) 次の 2 つの有理関数

$$\frac{2x^4 - 3x^3 + x + 7}{(x^2 + x + 1)^2}, \ \frac{x^5 + 3x^4 - 2x^3 - x^2 + 6}{(x^2 + x + 1)^2}$$

は, その差が $-x + 1$ となり, 多項式であるから, 代数的局所コホモロジー類として等しい. 即ち,

$$[\frac{2x^4 - 3x^3 + x + 7}{(x^2 + x + 1)^2}] \equiv [\frac{x^5 + 3x^4 - 2x^3 - x^2 + 6}{(x^2 + x + 1)^2}]$$

である. この代数的局所コホモロジー類の代表元としては, 例えば, proper な有理関数

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

を選べる.

(2) 次の 2 つの有理関数

$$\frac{1}{(x^3 - x - 1)^2}, \ \frac{x + 1}{(x^3 - x - 1)^2}$$

は, その差が $\frac{-x}{(x^3 - x - 1)^2}$ となり, 多項式ではないので, 代数的局所コホモロジー類として等しくない. 特異点の位置と重複度が同じでも, 特異性は異なっている.

2.3 ネーター作用素

$f(x)$ を $\mathbb{Q}[x]$ 上で既約, $h(x) \in \mathbb{Q}[x]$, $\ell \in \mathbb{N}$ とする. このとき, 代数的局所コホモロジー類 $[\frac{h(x)}{f(x)^\ell}]$ は,

$$T = (-\frac{d}{dx})^{\ell-1}t_0(x) + (-\frac{d}{dx})^{\ell-2}t_1(x) + \cdots + (-\frac{d}{dx})t_{\ell-2}(x) + t_{\ell-1}(x)$$

なる $\ell - 1$ 階の微分作用素を用いて,

$$[\frac{h(x)}{f(x)^\ell}] = T[\frac{f'(x)}{f(x)}]$$

と表せる. この T を代数的局所コホモロジー類のネーター作用素と呼ぶ.

$(-\frac{d}{dx})^{\ell-1-i}t_i(x)$ 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, $t_0(x), \ldots, t_{\ell-1}(x)$ は, 多項式ではなく, 多項式倍するという作用素 (零階の微分作用素) である. 従って例えば, $(-\frac{d}{dx})t_{\ell-2}(x)$ を $-t'_{\ell-2}(x)$ とはできないので注意.

例 2.2

代数的局所コホモロジー類 $[\frac{2x^3-2x^2-2x-2}{(x^2+1)^2}]$ のネーター作用素表示は,

$$[\frac{2x^3 - 2x^2 - 2x - 2}{(x^2+1)^2}] = \{(-\frac{d}{dx})x + (x+1)\}[\frac{2x}{x^2+1}]$$

となる.

ちなみに, 有理関数 $\frac{2x}{x^2+1}$ にこの微分作用素を施すと,

$$\begin{aligned}\{(-\frac{d}{dx})x + (x+1)\}(\frac{2x}{x^2+1}) &= \frac{2x^4 + 2x^3 + 2x^2 - 2x}{(x^2+1)^2} \\ &= \frac{2x^3 - 2x^2 - 2x - 2}{(x^2+1)^2} + 2\end{aligned}$$

となるので, 有理関数としては

$$\frac{2x^3 - 2x^2 - 2x - 2}{(x^2+1)^2} \neq \{(-\frac{d}{dx})x + (x+1)\}(\frac{2x}{x^2+1})$$

である.

2.3.1 ネーター作用素と D 加群

D 加群の理論を使えば, ネーター作用素の数学的構造が見えてくる. 結論のみを述べると, ネーター作用素 T の各右係数 $t_0(x), \ldots, t_{\ell-1}(x)$ は,

- 漸化式で逐次的に計算できる.
- 共通因子があったり, 逆冪の並びがあるなど, 規則的な構造をもつ.
- 剰余体 $\mathbb{Q}[x]/\langle f(x)\rangle$ 上で四則演算が行える.

といった性質がある.

中でも 3 番目の性質を利用することで, 計算を著しく高速化させることができる. 剰余体 $\mathbb{Q}[x]/\langle f(x)\rangle$ 上で四則演算が行えるということは, $t_0(x), \ldots, t_{\ell-1}(x)$ を求める計算過程で次数が $\deg f$ 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ということである. 計算機上での多項式の四則演算は, 多項式の次数が低い程速くなる.

2.3.2 ネーター作用素とローラン展開

これまで説明したネーター作用素表示

$$[\frac{h(x)}{f(x)^{\ell}}] = T[\frac{f'(x)}{f(x)}]$$

とローラン展開との関係を具体的に式を追って説明する.

$f(x)$ をモニックな 3 次多項式とし, その零点集合を $Z = \{\alpha, \beta, \gamma\}$ とする. (ここでは 3 次の場合で説明するが, 一般の m 次の場合でも議論は同様である.) まず, $f(x)$ を $\mathbb{C}$ 上で因数分解すると

$$\begin{aligned}
[\frac{f'(x)}{f(x)}] &= [\frac{(x-\beta)(x-\gamma)+(x-\gamma)(x-\alpha)+(x-\alpha)(x-\beta)}{(x-\alpha)(x-\beta)(x-\gamma)}] \\
&= [\frac{1}{x-\alpha}] + [\frac{1}{x-\beta}] + [\frac{1}{x-\gamma}]
\end{aligned}$$

となる.

両辺を $t(x)$ 倍すると,

$$t(x)[\frac{f'(x)}{f(x)}] = [\frac{t(x)}{x-\alpha}] + [\frac{t(x)}{x-\beta}] + [\frac{t(x)}{x-\gamma}]$$

となる.

$t(x)$ の α, β, γ におけるテイラー展開

$$\begin{aligned}
t(x) &= t(\alpha) + t'(\alpha)(x-\alpha) + \frac{1}{2!}t''(\alpha)(x-\alpha)^2 + \cdots \\
&= t(\beta) + t'(\beta)(x-\beta) + \frac{1}{2!}t''(\beta)(x-\beta)^2 + \cdots \\
&= t(\gamma) + t'(\gamma)(x-\gamma) + \frac{1}{2!}t''(\gamma)(x-\gamma)^2 + \cdots
\end{aligned}$$

を代入すると,

$$t(x)[\frac{f'(x)}{f(x)}] = [\frac{t(\alpha)}{x-\alpha}] + [\frac{t(\beta)}{x-\beta}] + [\frac{t(\gamma)}{x-\gamma}]$$

となる.

両辺に微分作用素 $(-\frac{d}{dx})^k$ を施すと,

$$
\begin{aligned}
(-\frac{d}{dx})^k t(x)[\frac{f'(x)}{f(x)}] &= [\frac{k!\ t(\alpha)}{(x-\alpha)^{k+1}}] + [\frac{k!\ t(\beta)}{(x-\beta)^{k+1}}] + [\frac{k!\ t(\gamma)}{(x-\gamma)^{k+1}}] \\
&= \sum_{z\in Z}[\frac{k!\ t(z)}{(x-z)^{k+1}}]
\end{aligned}
$$

となるから, ネーター作用素を

$$
T = (-\frac{d}{dx})^{\ell-1}t_0(x) + (-\frac{d}{dx})^{\ell-2}t_1(x) + \cdots + (-\frac{d}{dx})t_{\ell-2}(x) + t_{\ell-1}(x)
$$

とおくと,

$$
T[\frac{f'(x)}{f(x)}] = \sum_{z\in Z}\sum_{i=0}^{\ell-1}[\frac{i!\ t_{\ell-1-i}(z)}{(x-z)^{i+1}}]
$$

が導かれる.

有理関数の ℓ 位の極 $z \in Z$ におけるローラン展開の主要部の係数を,

$$
\frac{a_{-\ell}}{(x-z)^{\ell}} + \frac{a_{-(\ell-1)}}{(x-z)^{\ell-1}} + \cdots + \frac{a_{-(i+1)}}{(x-z)^{i+1}} + \cdots + \frac{a_{-2}}{(x-z)^2} + \frac{a_{-1}}{(x-z)}
$$

とおくと, ネーター作用素の係数とローラン展開の係数を見比べれば, $i = 0, \ldots, \ell-1$ に対して,

$$
a_{-(i+1)} = i!\ t_{\ell-1-i}(z)
$$

なる対応関係が得られる. この関係より, ネーター作用素は, 極におけるローラン展開の主要部を表現する微分作用素とみなせることがわかる.

2.3.3 ネーター作用素と留数

既に述べたように, 極におけるローラン展開の主要部を求める問題は, 代数的局所コホモロジー類のネーター作用素を求める問題に帰着される.

留数はローラン展開の一部であるから, これまで述べてきたようなネーター作用素 T

$$[\frac{h(x)}{f(x)^\ell}] = T[\frac{f'(x)}{f(x)}]$$

を利用することで留数値も求めることができる.

しかし, 留数のみを求めたい場合には, より効率的な計算法がある. それは, 分母はそのままで分子を 1 にした有理関数 $\frac{1}{f(x)^\ell}$ が定める代数的局所コホモロジー類 $[\frac{1}{f(x)^\ell}]$ のネーター作用素 U

$$[\frac{1}{f(x)^\ell}] = U[\frac{f'(x)}{f(x)}]$$

をまず計算し, 求めた U の形式的随伴作用素を分子 $h(x)$ に施して留数を求める方法である. 式で書くと, 極 $\alpha \in \{x|f(x) = 0\}$ における留数は,

$$\begin{aligned}
\mathrm{Res}_\alpha(\frac{h(x)}{f(x)^\ell}dx) &= \frac{1}{2\pi\sqrt{-1}}\oint \frac{h(x)}{f(x)^\ell}dx \\
&= \frac{1}{2\pi\sqrt{-1}}\oint [\frac{h(x)}{f(x)^\ell}]dx \\
&= \frac{1}{2\pi\sqrt{-1}}\oint h(x)[\frac{1}{f(x)^\ell}]dx \\
&= \frac{1}{2\pi\sqrt{-1}}\oint h(x)U[\frac{f'(x)}{f(x)}]dx \\
&= \frac{1}{2\pi\sqrt{-1}}\oint (U^*h)(x)[\frac{f'(x)}{f(x)}]dx \\
&= \frac{1}{2\pi\sqrt{-1}}\oint (U^*h)(x)\frac{f'(x)}{f(x)}dx \\
&= (U^*h)(\alpha)
\end{aligned}$$

となる.

留数計算において, T を計算する方法と, U を計算しその随伴作用素を分子に施す方法では, 後者の方が速い. その差は分子の多項式 $h(x)$ が大きいほど広がる.

2.4 剰余公式

任意の多項式 $p(x)$ を $f(x)$ で割った商を $q(x)$, 余りを $r(x)$ とすると,

$$p(x) = q(x)f(x) + r(x), \deg r < \deg f$$

と表せる. Z を $f(x)$ の零点集合とするとき, 剰余 $r(x)$ は, Z と, Z における $p(x)$ の値と, Z における $p(x)$ の導関数の値を用いて表せる. これを, 剰余公式と呼ぶことにする.

例 2.3 $f(x) = (x-\alpha)(x-\beta)$ を与えたときの剰余公式を求めよ.

$r(x)$ は 1 次以下の多項式であるから, $r(x) = Ax + B$ とおける. 連立方程式 $p(\alpha) = r(\alpha) = A\alpha + B$, $p(\beta) = r(\beta) = A\beta + B$ を解くと, 剰余公式は,

$$r(x) = \frac{p(\alpha) - p(\beta)}{\alpha - \beta}x + \frac{\alpha p(\beta) - \beta p(\alpha)}{\alpha - \beta}$$

で与えられる.

例 2.4 $f(x) = (x-\alpha)^2$ を与えたときの剰余公式を求めよ.

$p'(x) = q'(x)(x-\alpha)^2 + 2q(x)(x-\alpha) + r'(x)$ を用いて, 上の例と同じように連立方程式を解くと, 剰余公式は,

$$r(x) = p'(\alpha)x + p(\alpha) - \alpha p'(\alpha)$$

で与えられる.

このように, 多項式 $f(x)$ を与えたときの剰余公式を求めるプログラムが rem_formula である. ただし rem_formula で採用しているアルゴリズムは, 上記の例のように, 連立方程式を解く方法ではなく, エルミートの補間積分

$$r(x) = \frac{1}{2\pi\sqrt{-1}} \int_C \frac{f(y) - f(x)}{y - x} \frac{1}{f(y)} p(y) dy$$

を使う方法である. ここで, C は全ての極 $\{y | f(y) = 0\}$ の周りを一周する閉曲線とする.

2.5 指数多項式リスト

次のような (多項式) $\times$ (指数関数) の形をしている関数の線形結合

$$\sum_{\alpha_1 \in Z_1} c_1(\alpha_1, z)e^{\alpha_1 z} + \cdots + \sum_{\alpha_n \in Z_n} c_n(\alpha_n, z)e^{\alpha_n z}$$

を指数多項式 (exponential polynomials) と呼ぶ. ここで, z を独立変数, $f_1(x), \ldots, f_n(x)$ を $\mathbb{Q}$ 上で既約な多項式, $Z_1 = \{x | f_1(x) = 0\}, \ldots, Z_n = \{x | f_n(x) = 0\}$ とする. 指数多項式は, 有理数係数の線形常微分方程式などに登場する.

プログラム上では, この指数多項式を

```
[[f_1(x),c_1(x,z)],...,[f_n(x),c_n(x,z)]]
```

なるリストで表現する. これを **指数多項式リスト** と呼ぶことにする.

リストの `f_1(x),...,f_n(x)` は $\mathbb{Q}$ 上で既約な多項式 $f_1(x), \ldots, f_n(x)$ である.
リストの `c_1(x,z),...,c_n(x,z)` は $e^{\alpha_1 z}, \ldots, e^{\alpha_n z}$ の係数となる多項式 $c_1(\alpha_1, z), \ldots, c_n(\alpha_n, z)$ であるが, α_i の多項式は留数表示すると x の多項式となるので, 便宜上 α_i を $f_i(x)$ と同じ変数である x に変えて入力する.

例えば,

指数多項式	指数多項式リスト
1	`[[x,1]]`
z	`[[x,z]]`
e^z	`[[x-1,1]]`
$2e^{3z} + (3z^2 + 1)e^z$	`[[x-3,2],[x-1,3*z^2+1]]`
$\sum_{\alpha_1 \in Z} e^{\alpha_1 z},\ Z = \{x \| x^3 - x - 1 = 0\}$	`[[x^3-x-1,1]]`
$\sum_{\alpha_1 \in Z}((z^2 + 1)\alpha_1 + 2z + 1)e^{\alpha_1 z},\ Z = \{x \| x^4 + 1 = 0\}$	`[[x^4+1,(z^2+1)*x+2*z+1]]`

といったようになる.

2.6　フーリエ・ボレル変換

$n(x), d(x) \in \mathbb{Q}[x]$ とする．代数的局所コホモロジー類 $[\frac{n(x)}{d(x)}]$ のフーリエ･ボレル変換 (Fourier-Borel transformation; FBT と略記) を,

$$\begin{aligned} FB([\frac{n(x)}{d(x)}])(z) &= \mathrm{Res}([\frac{n(x)}{d(x)}]e^{zx}dx) \\ &= \frac{1}{2\pi\sqrt{-1}}\int_C [\frac{n(x)}{d(x)}]e^{zx}dx. \end{aligned}$$

で定める．ここで, C は全ての極 $\{x|d(x)=0\}$ の周りを一周する閉曲線とする．

FBT による代数的局所コホモロジー類の像 (言い換えれば, 有理形関数 $\frac{n(x)}{d(x)}e^{zx}$ の留数) は, 指数多項式となる．有理形関数 $\frac{n(x)}{d(x)}e^{zx}$ の留数計算は, 有理関数の場合とほとんど同じであり, residue を少し細工したプログラムで計算される．

FBT は, 代数的局所コホモロジー類を指数多項式に変換するが, その逆変換, 即ち, 指数多項式から代数的局所コホモロジー類を求める変換を, 逆フーリエ・ボレル変換 (inverse FBT; invFBT と略記) と呼ぶことにする．

FBT は, 有理数係数の線形常微分方程式のコーシー問題の解を求める際に役に立ち, invFBT は, 有理数係数の線形常微分方程式のコーシー問題の特殊解を求める際に役に立つ．どちらも変換テーブルを必要とせず, 統一的なアルゴリズムで計算される．

3 プログラムの使用例

3.1 cpfd

cpfd は, 有理関数

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

の部分分数分解を求めるプログラムである.

ここで, $n(x), d(x) \in \mathbb{Q}[x]$. $h(x) \in \mathbb{Q}[x]$, $f_1(x), \ldots, f_m(x)$ は $\mathbb{Q}$ 上で既約な多項式, $\ell_1, \ldots, \ell_m \in \mathbb{N}$ とする.

入力は, 左辺の形と右辺の形の両方に対応している. 右辺の形で入力する場合は, 分母はリストにする.

入力

$$\texttt{taji\_alc.cpfd(}n(x), d(x)\texttt{);}$$

$$\texttt{taji\_alc.cpfd(}h(x), [[f_1(x), \ell_1], \ldots, [f_m(x), \ell_m]]\texttt{);}$$

部分分数分解には, 分母を冪展開する complete な部分分数分解 (Complete Partial Fraction Decomposition)

$$\{\frac{h_{1,\ell_1}(x)}{f_1(x)^{\ell_1}} + \cdots + \frac{h_{1,1}(x)}{f_1(x)}\} + \cdots + \{\frac{h_{m,\ell_m}(x)}{f_m(x)^{\ell_m}} + \cdots + \frac{h_{m,1}(x)}{f_m(x)}\}$$

と, 分母を冪展開しない部分分数分解

$$\frac{h_1(x)}{f_1(x)^{\ell_1}} + \cdots + \frac{h_m(x)}{f_m(x)^{\ell_m}}$$

の 2 つがある. default では, complete な部分分数分解を返す. 分母を冪展開しない部分分数分解がほしい場合は, オプションで switch=10 とする.

出力 (default)

$$[[[h_{1,\ell_1}(x), [f_1(x), \ell_1]], \ldots, [h_{1,1}(x), [f_1(x), 1]]], \ldots, [[h_{m,\ell_m}(x), [f_m(x), \ell_m]], \ldots, [h_{m,1}(x), [f_m(x), 1]]]]$$

出力 (switch=10)

$$[[h_1(x), [f_1(x), \ell_1]], \ldots, [h_m(x), [f_m(x), \ell_m]]]$$

例 3.1 有理関数 $\frac{x^3-x-1}{x^4+2x^3+2x^2+2x+1}$ の部分分数分解を求めよ.

```
[312] taji_alc.cpfd(x^3-x-1,x^4+2*x^3+2*x^2+2*x+1);
[[[1/2*x-1,[x^2+1,1]]],[[-1/2,[x+1,2]],[1/2,[x+1,1]]]]
[313] taji_alc.cpfd(x^3-x-1,x^4+2*x^3+2*x^2+2*x+1|switch=10);
[[1/2*x-1,[x^2+1,1]],[1/2*x,[x+1,2]]]
```

$$\begin{aligned}\frac{x^3-x-1}{x^4+2x^3+2x^2+2x+1} &= \frac{x^3-x-1}{(x^2+1)(x+1)^2} \\ &= \frac{\frac{1}{2}x-1}{x^2+1}+\frac{-\frac{1}{2}}{(x+1)^2}+\frac{\frac{1}{2}}{x+1} \quad (1) \\ &= \frac{\frac{1}{2}x-1}{x^2+1}+\frac{\frac{1}{2}x}{(x+1)^2} \quad (2)\end{aligned}$$

cpfd では, 先に (1) の部分分数分解が求まる. (2) のように分母を冪展開しない部分分数分解は, (1) の部分分数分解の結果をホーナー法で足し上げて求める. (1) の部分分数分解の方が, (2) よりも多くの情報をもっている.

3.2 snoether

snoether は, 代数的局所コホモロジー類

$$[\frac{n(x)}{d(x)}] = [\frac{h(x)}{f_1(x)^{\ell_1}\cdots f_m(x)^{\ell_m}}]$$

のネーター作用素を求めるプログラムである.

ここで, $n(x), d(x) \in \mathbb{Q}[x]$. $h(x) \in \mathbb{Q}[x]$, $f_1(x), \ldots, f_m(x)$ は $\mathbb{Q}$ 上で既約な多項式, $\ell_1, \ldots, \ell_m \in \mathbb{N}$ とする.

入力は, 左辺の形と右辺の形の両方に対応している. 右辺の形で入力する場合は, 分母はリストにする.

入力

taji_alc.snoether($n(x), d(x)$);

taji_alc.snoether($h(x), [[f_1(x), \ell_1], \ldots, [f_m(x), \ell_m]]$);

出力では, 上記の代数的局所コホモロジー類のネーター作用素表示

$$T_1[\frac{f_1'(x)}{f_1(x)}] + \cdots + T_m[\frac{f_m'(x)}{f_m(x)}]$$

における各既約因子に対するネーター作用素

$$\begin{aligned}
T_1 &= (-\frac{d}{dx})^{\ell_1-1}t_{1,0}(x) + (-\frac{d}{dx})^{\ell_1-2}t_{1,1}(x) + \cdots + (-\frac{d}{dx})t_{1,\ell_1-2}(x) + t_{1,\ell_1-1}(x) \\
&\vdots \\
T_m &= (-\frac{d}{dx})^{\ell_m-1}t_{m,0}(x) + (-\frac{d}{dx})^{\ell_m-2}t_{m,1}(x) + \cdots + (-\frac{d}{dx})t_{m,\ell_m-2}(x) + t_{m,\ell_m-1}(x)
\end{aligned}$$

をリストで返す.

出力 (default)

$$[[f_1(x), [t_{1,0}(x), \ldots, t_{1,\ell_1-1}(x)]], \ldots, [f_m(x), [t_{m,0}(x), \ldots, t_{m,\ell_m-1}(x)]]]$$

例 3.2 次の代数的局所コホモロジー類のネーター作用素 T

$$[\frac{1}{(x^3-x-1)^3}] = T[\frac{3x^2-1}{x^3-x-1}]$$

を求めよ.

ネーター作用素の戻り値をどう表現するかは考える必要がある. snoether では, 4 通りの表現方法を用意し, Asir のオプション指定という機能を使って選択式とした. どの表現がいいかは一長一短がある.

```
[330] taji_alc.snoether(1,[[x^3-x-1,3]]);
[[x^3-x-1,[9/529*x^2-27/1058*x+11/1058,-81/529*x^2-9/529*x+135/529,-4905/12167*x
^2+4563/12167*x+3270/12167]]]
[331] taji_alc.snoether(1,[[x^3-x-1,3]]|switch=1);
[[x^3-x-1,[[18*x^2-27*x+11,1058],[-81*x^2-9*x+135,529],[-4905*x^2+4563*x+3270,12
167]]]]
[332] taji_alc.snoether(1,[[x^3-x-1,3]]|switch=10);
[[x^3-x-1,[[414*x^2-621*x+253,-3726*x^2-414*x+6210,-9810*x^2+9126*x+6540],24334]
]]
[333] taji_alc.snoether(1,[[x^3-x-1,3]]|switch=20);
[[x^3-x-1,[[[18*x^2-27*x+11,529],[-162*x^2-18*x+270,529],[-9810*x^2+9126*x+6540,
12167]],2]]]
```

ネーター作用素は,

$$\begin{aligned}
T &= \{(-\frac{d}{dx})^2(\frac{9}{529}x^2-\frac{27}{1058}x+\frac{11}{1058})+(-\frac{d}{dx})(-\frac{81}{529}x^2-\frac{9}{529}x+\frac{135}{529}) \\
&\quad +(-\frac{4905}{12167}x^2+\frac{4563}{12167}x+\frac{3270}{12167})\} \\
&= \{(-\frac{d}{dx})^2\frac{1}{1058}(18x^2-27x+11)+(-\frac{d}{dx})\frac{1}{529}(-81x^2-9x+135) \\
&\quad +\frac{1}{12167}(-4905x^2+4563x+3270)\} \\
&= \frac{1}{24334}\{(-\frac{d}{dx})^2(414x^2-621x+253)+(-\frac{d}{dx})(-3726x^2-414x+6210) \\
&\quad +(-9810x^2+9126x+6540)\} \\
&= \frac{1}{2}\{(-\frac{d}{dx})^2\frac{1}{529}(18x^2-27x+11)+(-\frac{d}{dx})\frac{1}{529}(-162x^2-18x+270) \\
&\quad +\frac{1}{12167}(-9810x^2+9126x+6540)\}
\end{aligned}$$

で与えられる.

switch=20 の表現で先頭の $\frac{1}{2}$ は $\frac{1}{2!}$ の意味であり, 重複度が ℓ ならこの部分は $\frac{1}{(\ell-1)!}$ となる. 階乗部分をあえて約分しないで残しておくと, 数学的な構造が綺麗に見えてよいこともある.

3.3 laurent_expansion

laurent_expansin は, 有理関数

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

の極におけるローラン展開の主要部の係数を求めるプログラムである.

ここで, $n(x), d(x) \in \mathbb{Q}[x]$. $h(x) \in \mathbb{Q}[x]$, $f_1(x), \ldots, f_m(x)$ は $\mathbb{Q}$ 上で既約な多項式, $\ell_1, \ldots, \ell_m \in \mathbb{N}$ とする.

入力は, 左辺の形と右辺の形の両方に対応している. 右辺の形で入力する場合は, 分母はリストにする.

入力

$$\texttt{taji\_alc.laurent\_expansion(}n(x), d(x)\texttt{);}$$

$$\texttt{taji\_alc.laurent\_expansion(}h(x), [[f_1(x), \ell_1], \ldots, [f_m(x), \ell_m]]\texttt{);}$$

出力では, 各極 $\alpha_1 \in \{x|f_1(x) = 0\}, \ldots, \alpha_m \in \{x|f_m(x) = 0\}$ におけるローラン展開の主要部

$$\frac{a_{1,\ell_1}(\alpha_1)}{(x-\alpha_1)^{\ell_1}} + \cdots + \frac{a_{1,1}(\alpha_1)}{(x-\alpha_1)}, \ldots, \frac{a_{m,\ell_m}(\alpha_m)}{(x-\alpha_m)^{\ell_m}} + \cdots + \frac{a_{m,1}(\alpha_m)}{(x-\alpha_m)}$$

の係数を与える多項式の組をリストで返す.

ただし, 各 $a_{i,j}(\alpha_i)$ は, 便宜上, 入力の変数と同じ x の多項式 $a_{i,j}(x)$ として返す.

出力 (default)

$$[[f_1(x), [a_{1,\ell_1}(x), \ldots, a_{1,1}(x)]], \ldots, [f_m(x), [a_{m,\ell_m}(x), \ldots, a_{m,1}(x)]]]$$

例 3.3 有理関数 $\frac{x}{(x-1)^3}$ の極におけるローラン展開の主要部を求めよ.

```
[289] taji_alc.laurent_expansion(x,(x-1)^3);
[[x-1,[1,1,0]]]
```

極 $\alpha = 1$ における主要部は,

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

である.

例 3.4 有理関数 $\frac{1}{(x^3-x-1)^3}$ の極におけるローラン展開の主要部を求めよ.

```
[290] taji_alc.laurent_expansion(1,[[x^3-x-1,3]]|switch=1);
[[x^3-x-1,[[18*x^2-27*x+11,529],[-81*x^2-9*x+135,529],[-4905*x^2+4563*x+3270,121
67]]]]
```

極 $\alpha \in \{x | x^3 - x - 1 = 0\}$ における主要部は,

$$\frac{\frac{1}{529}(18\alpha^2 - 27\alpha + 11)}{(x-\alpha)^3} + \frac{\frac{1}{529}(-81\alpha^2 - 9\alpha + 135)}{(x-\alpha)^2} + \frac{\frac{1}{12167}(-4905\alpha^2 + 4563\alpha + 3270)}{(x-\alpha)}$$

である.

例 3.5 有理関数 $\frac{x^3-x-1}{(x-1)^3(x^2-x+3)^2}$ の極におけるローラン展開の主要部を求めよ.

```
[291] taji_alc.laurent_expansion(x^3-x-1,(x-1)^3*(x^2-x+3)^2|switch=1);
[[x^2-x+3,[[23*x-6,297],[8*x-15,99]]],[x-1,[[-1,9],[8,27],[2,9]]]]
```

極 $\alpha = 1$ における主要部は,

$$\frac{-\frac{1}{9}}{(x-1)^3} + \frac{\frac{8}{27}}{(x-1)^2} + \frac{\frac{2}{9}}{(x-1)}$$

であり, 極 $\beta \in \{x | x^2 - x + 3 = 0\}$ における主要部は,

$$\frac{\frac{1}{297}(23\beta - 6)}{(x-\beta)^2} + \frac{\frac{1}{99}(8\beta - 15)}{(x-\beta)}$$

である.

ちなみに, 留数はローラン展開の一部であるから, laurent_expansion でも留数が求まる. 例 3.3 なら極における留数は 0, 例 3.4 なら極における留数は $\frac{1}{12167}(-4905\alpha^2 + 4563\alpha + 3270)$ である. しかし, 留数のみが欲しい場合には, residue を使う方がよい. residue の方が laurent_expansion よりもずっと速いからである.

3.4 residue

residue は, 有理関数

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

の極における留数を求めるプログラムである.

ここで, $n(x), d(x) \in \mathbb{Q}[x]$. $h(x) \in \mathbb{Q}[x]$, $f_1(x), \ldots, f_m(x)$ は $\mathbb{Q}$ 上で既約な多項式, $\ell_1, \ldots, \ell_m \in \mathbb{N}$ とする.

入力は, 左辺の形と右辺の形の両方に対応している. 右辺の形で入力する場合は, 分母はリストにする.

入力

$$\texttt{taji\_alc.residue(}n(x), d(x)\texttt{);}$$

$$\texttt{taji\_alc.residue(}h(x), [[f_1(x), \ell_1], \ldots, [f_m(x), \ell_m]]\texttt{);}$$

出力では, 各極 $\alpha_1 \in \{x | f_1(x) = 0\}, \ldots, \alpha_m \in \{x | f_m(x) = 0\}$ における留数

$$\mathrm{Res}_{\alpha_1}(\frac{n(x)}{d(x)}) = r_1(\alpha_1), \ldots, \mathrm{Res}_{\alpha_m}(\frac{n(x)}{d(x)}) = r_m(\alpha_m)$$

を表す多項式の組をリストで返す.

ただし, $r_1(\alpha_1), \ldots, r_m(\alpha_m)$ は, 便宜上, 入力の変数と同じ x の多項式 $r_1(x), \ldots, r_m(x)$ として返す.

出力 (default)

$$[[f_1(x), r_1(x)], \ldots, [f_m(x), r_m(x)]]$$

例 **3.6** 有理関数 $R(x)=\frac{1}{x^4+1}$ の留数を求めよ.

```
[306] taji_alc.residue(1,x^4+1);
[[x^4+1,-1/4*x]]
```

この戻り値は, $\alpha\in\{x|x^4+1=0\}$ とおいたとき,

$$\mathrm{Res}_\alpha(R(x))=-\frac{1}{4}\alpha$$

を表している.

residue で採用しているアルゴリズムでは, $\mathbb{C}$ 上の 1 点に注目するのではなく, $\mathbb{Q}$ 上での既約因子自体に注目して留数を求める. 戻り値の留数は, その因子の全ての零点が共通に満たす留数多項式である. 従って, 1 点ごとの留数値をさらに求めたい場合には, 求めた留数多項式に因子の零点 (即ち特異点) の値を代入する必要がある.

この例で言うと, 求めた留数多項式 $-\frac{1}{4}\alpha$ に, x^4+1 の 4 つある零点

$$\{\frac{1}{\sqrt{2}}(1+i),\frac{1}{\sqrt{2}}(-1+i),\frac{1}{\sqrt{2}}(-1-i),\frac{1}{\sqrt{2}}(1-i)\}$$

をそれぞれ代入したものが個別の留数値である. 例えば, $\alpha_1=\frac{1}{\sqrt{2}}(1+i)$ での留数は,

$$\mathrm{Res}_{\alpha_1}(R(x))=-\frac{1}{4}\alpha_1=-\frac{1}{4\sqrt{2}}(1+i)$$

である.

例 **3.7** 有理関数 $R(x)=\frac{x^3+1}{x^{18}-2x^{14}+x^{10}-x^8+2x^4-1}$ の留数を求めよ.

簡約化すると, $R=\frac{x^2-x+1}{(x^4+x^3+x^2+x+1)(x^4-x^3+x^2-x+1)(x^2+1)^2(x+1)^2(x-1)^3}$ となる.

```
[307] taji_alc.residue(x^3+1,x^18-2*x^14+x^10-x^8+2*x^4-1|switch=1);
[[x^4+x^3+x^2+x+1,[-2*x^2-x-2,50]],[x^4-x^3+x^2-x+1,[-2*x^3+4*x^2-x-2,50]],[x^2+
1,[8*x+5,32]],[x+1,[-39,320]],[x-1,[67,320]]]
```

switch を 1 に指定して留数を整数係数化リストで返している. $\alpha\in\{x|x^4+x^3+x^2+x+1=0\}$, $\beta\in\{x|x^4-x^3+x^2-x+1=0\}$, $\gamma\in\{x|x^2+1=0\}$ とおくと, 答えは,

$$\mathrm{Res}_\alpha(R)=\frac{1}{50}(-2\alpha^2-\alpha-2),\ \mathrm{Res}_\beta(R)=\frac{1}{50}(-2\beta^3+4\beta^2-\beta-2),$$

$$\mathrm{Res}_\gamma(R)=\frac{1}{32}(8\gamma+5),\ \mathrm{Res}_{-1}(R)=-\frac{39}{320},\ \mathrm{Res}_1(R)=\frac{67}{320}$$

である.

residue は, snoether と密接な関係があり, アルゴリズムの 8 割ぐらいが同じである. 主な違いは, 留数のみを求めたい場合には, ネーター作用素の随伴作用素を分子に施すだけでよいというところにある. なお, residue では部分分数分解 cpfd を経由せず直接ネーター作用素を求めている.

3.5 invpow

invpow は, 剰余体 $\mathbb{Q}[x]/\langle f(x)\rangle$ における $g(x)$ の逆冪 (逆元の m 乗)

$$a(x) = g(x)^{-m} \pmod{f(x)}$$

を求めるプログラムである.

ここで, $g(x) \in \mathbb{Q}[x]$, $f(x)$ は $\mathbb{Q}$ 上で既約な多項式, $m \in \mathbb{N}$, $g(x)$ と $f(x)$ は互いに素とする.

入力

$$\texttt{taji\_alc.invpow}(g(x), f(x), m)\texttt{;}$$

出力

$$a(x)$$

例 3.8 剰余体 $\mathbb{Q}[x]/\langle x^3 - x - 1\rangle$ における $3x^2 - 1$ の逆元の 30 乗を求めよ.

```
[329] taji_alc.invpow(3*x^2-1,x^3-x-1,30|switch=1);
[1857324483*x^2-2100154824*x-477264412,266635235464391245607]
```

剰余体 $\mathbb{Q}[x]/\langle x^3 - x - 1\rangle$ 上で, $3x^2 - 1$ の逆元の 30 乗は,

$$\frac{1}{266635235464391245607}(1857324483x^2 - 2100154824x - 477264412)$$

である.

ここで注目すべきこととして, $3x^2 - 1$ の逆元である $\frac{1}{23}(-6x^2 + 9x + 4)$ の 30 乗は, 単純に考えれば, 分母の整数は 23^{30} になると予想される. しかし実際は, $266635235464391245607 = 23^{15}$ で済んでいる. これは, 多項式部分に共通因数 23^{15} が現れて, キャンセレーションが起きているからである.

3.6 rem_formula

rem_formula は, 多項式

$$f(x) = f_1(x)^{\ell_1} \cdots f_m(x)^{\ell_m}$$

を与えたときの剰余公式を求めるプログラムである.

ここで, $f_1(x), \ldots, f_m(x)$ は $\mathbb{Q}$ 上で既約な多項式, $\ell_1, \ldots, \ell_m \in \mathbb{N}$ とする.

入力は, 各既約因子の零点を $\alpha_1 \in Z_1 = \{x | f_1(x) = 0\}, \ldots, \alpha_m \in Z_m = \{x | f_m(x) = 0\}$ とおいたとき, 既約分解したリストの各成分に, 零点の symbol を追加した形で行う.

入力

`taji_alc.rem_formula(`$[[f_1(x), \ell_1, \alpha_1], \ldots, [f_m(x), \ell_m, \alpha_m]]$`);`

出力では, 剰余公式

$$\begin{aligned} r(x) &= \sum_{\alpha_1 \in Z_1} \{r_{1,\ell_1-1}(x,\alpha_1)p^{(\ell_1-1)}(\alpha_1) + \cdots + r_{1,0}(x,\alpha_1)p(\alpha_1)\} \\ & \quad + \cdots + \sum_{\alpha_m \in Z_m} \{r_{m,\ell_m-1}(x,\alpha_m)p^{(\ell_m-1)}(\alpha_m) + \cdots + r_{m,0}(x,\alpha_m)p(\alpha_m)\} \end{aligned}$$

を返す. ここで, $r(x)$ は $p(x)$ を $f(x)$ で割ったときの余りを意味する.

出力 (default)

$$[[r_{1,\ell_1-1}(x,\alpha_1), \ldots, r_{1,0}(x,\alpha_1)], \ldots, [r_{m,\ell_m-1}(x,\alpha_m), \ldots, r_{m,0}(x,\alpha_m)]]$$

例 **3.9** $f(x)=(x-1)(x-2)$ を与えたときの剰余公式を求めよ.

```
[583] taji_alc.rem_formula([[x-1,1,z1],[x-2,1,z2]]);
[[-x+2],[x-1]]
[584] taji_alc.rem_formula([[x-1,1,z1],[x-2,1,z2]]|switch=20);
(-p^(0)(z1)+p^(0)(z2))*x+2*p^(0)(z1)-p^(0)(z2)
```

剰余公式は,

$$\begin{aligned} r(x) &= (-x+2)p(z_1)+(x-1)p(z_2) \\ &= (-p(z_1)+p(z_2))x+2p(z_1)-p(z_2) \end{aligned}$$

で与えられる.

例 **3.10** $f(x)=(x-1)^2$ を与えたときの剰余公式を求めよ.

```
[587] taji_alc.rem_formula([[x-1,2,z1]]);
[[x-1,1]]
[588] taji_alc.rem_formula([[x-1,2,z1]]|switch=20);
p^(1)(z1)*x-p^(1)(z1)+p^(0)(z1)
```

剰余公式は,

$$\begin{aligned} r(x) &= (x-1)p'(z_1)+p(z_1) \\ &= p'(z_1)x-p'(z_1)+p(z_1) \end{aligned}$$

で与えられる.

重複度をもつ場合は, $p(x)$ の導関数が現れる.

例 **3.11** $f(x)=x^3-x-1$ を与えたときの剰余公式を求めよ.

```
[284] taji_alc.rem_formula([[x^3-x-1,1,z1]]);
[[(-6/23*z1^2+9/23*z1+4/23)*x^2+(9/23*z1^2-2/23*z1-6/23)*x+4/23*z1^2-6/23*z1+5/2
3]]
```

剰余公式は,

$$r(x) = \sum_{z_1\in Z}((-\frac{6}{23}z_1^2+\frac{9}{23}z_1+\frac{4}{23})x^2+(\frac{9}{23}z_1^2-\frac{2}{23}z_1-\frac{6}{23})x+\frac{4}{23}z_1^2-\frac{6}{23}z_1+\frac{5}{23})p(z_1)$$

で与えられる.

$f(x)$ が 2 次以上の場合は, 零点 (代数的数) が現れる. この場合, 全ての零点に関する和をとらないといけない.

3.7 solve_ode_cp

solve_ode_cp は, 線形常微分方程式 (liner Ordinary Differential Equations; ODE と略記) のコーシー問題 (Cauchy Probrem; CP と略記)

$$\begin{cases} Pu(z) = f(z) \\ u(0) = c_0, u'(0) = c_1, \ldots, u^{(n-1)}(0) = c_{n-1} \end{cases}$$

の解を求めるプログラムである.

ここで, P を有理数係数の n 階の線形常微分作用素 $P = a_n \frac{d^n}{dz^n} + a_{n-1} \frac{d^{n-1}}{dz^{n-1}} + \cdots + a_1 \frac{d}{dz} + a_0$, $f(z)$ を指数多項式とする.

P の特性多項式を $\mathbb{Q}$ 上で既約分解し

$$p(x) = p_1(x)^{\ell_1} \cdots p_m(x)^{\ell_m}$$

とおく. 入力は, 左辺の形と右辺の形の両方に対応している.

引数は, 第 1 引数は特性多項式, 第 2 引数は関数の独立変数, 第 3 引数は $f(z)$ の指数多項式リスト

$$f(z) : [[f_1(x), c_1(x, z)], \ldots, [f_m(x), c_m(x, z)]]$$

を入力する. 斉次形の場合は, 第 3 引数に 0 を入力してもよい.

入力

taji_alc.solve_ode_cp($p(x), z, f(z)$);

taji_alc.solve_ode_cp($[[p_1(x), \ell_1], \ldots, [p_m(x), \ell_m]], z, f(z)$);

コーシー問題の一般解は, コーシー問題の基本解 $u_0(z), \ldots, u_{n-1}(z)$ とコーシーデータ $c_0, \ldots, c_{n-1}$ の線形結合にコーシー問題の特殊解 $v(z)$ を加えた

$$u(z) = c_0 u_0(z) + \cdots + c_{n-1} u_{n-1}(z) + v(z)$$

なる形をしている.

出力では, コーシー問題の基本解 $u_0(z), \ldots, u_{n-1}(z)$ は指数多項式となるので, 指数多項式リスト

$$u_0(z) : [[p_1(x), c_{1,0}(x, z)], \ldots, [p_m(x), c_{m,0}(x, z)]]$$

$$\vdots$$

$$u_{n-1}(z) : [[p_1(x), c_{1,n-1}(x, z)], \ldots, [p_m(x), c_{m,n-1}(x, z)]]$$

で返す. また, コーシー問題の特殊解 $v(z)$ も指数多項式となるので, 指数多項式リストで返す. 特殊解については, 3.8 節で詳しく述べている.

出力 (default)

$$[u_0(z), \ldots, u_{n-1}(z), v(z)]$$

例 3.12 次のコーシー問題を解け.

$$\begin{cases} (D^2-4)u(z)=0 \\ u(0)=1,\ u'(0)=-1 \end{cases}$$

```
[449] taji_alc.solve_ode_cp(x^2-4,z,0);
[[[x+2,1/2],[x-2,1/2]],[[x+2,-1/4],[x-2,1/4]]]
```

戻り値は指数多項式リストであるが, これは一般解が,

$$u(z)=c_0(\frac{1}{2}e^{-2z}+\frac{1}{2}e^{2z})+c_1(-\frac{1}{4}e^{-2z}+\frac{1}{4}e^{2z})$$

で与えられることを表している.

この例の場合, コーシーデータも具体的に $c_0=1, c_1=-1$ と与えられているので, Asir のオプションリスト機能を使ってデータを与えると,

```
[450] taji_alc.solve_ode_cp(x^2-4,z,0|data=[1,-1]);
[[x+2,3/4],[x-2,1/4]]
```

$$u(z)=\frac{3}{4}e^{-2z}+\frac{1}{4}e^{2z}$$

のように指数関数で整理した形で返される.

また, オプションリストのデータを未知数にすることもできて, その場合も指数関数で整理した形で返される.

```
[451] taji_alc.solve_ode_cp(x^2-4,z,0|data=[c0,c1]);
[[x+2,1/2*c0-1/4*c1],[x-2,1/2*c0+1/4*c1]]
```

$$u(z)=(\frac{1}{2}c_0-\frac{1}{4}c_1)e^{-2z}+(\frac{1}{2}c_0+\frac{1}{4}c_1)e^{2z}$$

例 3.13 次のコーシー問題を解け.

$$\begin{cases} (D^3-2D+1)u(z)=0 \\ u(0)=3,\ u'(0)=0,\ u''(0)=9 \end{cases}$$

```
[479] taji_alc.solve_ode_cp(x^3-2*x+1,z,0|data=[3,0,9],switch=1);
[[x^2+x-1,[-3*x-3,1]],[x-1,[6,1]]]
```

コーシー問題の解は,

$$u(z)=\sum_{\alpha\in\{x|x^2+x-1\}}(-3\alpha-3)e^{\alpha z}+6e^z$$

で与えられる.

3.8 solve_ode_cp_ps

solve_ode_cp_ps は, 線形常微分方程式 (liner Ordinary Differential Equations; ODE と略記) のコーシー問題 (Cauchy Probrem; CP と略記)

$$\begin{cases} Pu(z) = f(z) \\ u(0) = c_0, u'(0) = c_1, \ldots, u^{(n-1)}(0) = c_{n-1} \end{cases}$$

の特殊解 (Particular Solution; PS と略記) を求めるプログラムである. solve_ode_cp の中で部品として呼び出されるが, 単独で使うこともできる.

ここで, P を有理数係数の n 階の線形常微分作用素 $P = a_n \frac{d^n}{dz^n} + a_{n-1} \frac{d^{n-1}}{dz^{n-1}} + \cdots + a_1 \frac{d}{dz} + a_0$, $f(z)$ を指数多項式とする.

P の特性多項式を $\mathbb{Q}$ 上で既約分解し

$$p(x) = p_1(x)^{\ell_1} \cdots p_m(x)^{\ell_m}$$

とおく. 入力は, 左辺の形と右辺の形の両方に対応している.

引数は, 第 1 引数は特性多項式, 第 2 引数は関数の独立変数, 第 3 引数は $f(z)$ の指数多項式リスト

$$f(z) : [[f_1(x), c_1(x,z)], \ldots, [f_m(x), c_m(x,z)]]$$

を入力する.

入力

taji_alc.solve_ode_cp_ps($p(x), z, f(z)$);

taji_alc.solve_ode_cp_ps($[[p_1(x), \ell_1], \ldots, [p_m(x), \ell_m]], z, f(z)$);

出力では, コーシー問題の特殊解を指数多項式リストとして返す. ただし, $p_1(x), \ldots, p_m(x)$ と $f_1(x), \ldots, f_m(x)$ に同じ因子がある場合は一つにされる.

出力 (default)

$$[[p_1(x), c_{p1}(x,z)], \ldots, [p_m(x), c_{pm}(x,z)], [f_1(x), c_{v1}(x,z)], \ldots, [f_m(x), c_{vm}(x,z)]]$$

例 **3.14** 次のコーシー問題の特殊解を求めよ.

$$\begin{cases} (D-2)(D+3)u(z) = e^z \\ u(0) = c_0,\ u'(0) = c_1 \end{cases}$$

```
[445] taji_alc.solve_ode_cp_ps((x-2)*(x+3),z,[[x-1,1]]);
[[x-1,-1/4],[x+3,1/20],[x-2,1/5]]
```

コーシー問題の特殊解は,

$$v(z) = \frac{1}{20}e^{-3z} - \frac{1}{4}e^z + \frac{1}{5}e^{2z}$$

で与えられる.

この特殊解は, 微分方程式の解であると同時に, コーシー条件 $v(0) = 0, v'(0) = 0$ も満たしている. "コーシー問題の"と修飾している理由は, そのためである.

「コーシー条件は満たしてなくてもいいから, 簡単な形の特殊解が欲しい」という要求もあるかもしれない. その場合は, オプションで switch2=1 と指定する.

```
[446] taji_alc.solve_ode_cp_ps((x-2)*(x+3),z,[[x-1,1]]|switch2=1);
[[x-1,-1/4]]
```

簡単な形の特殊解は,

$$v(z) = -\frac{1}{4}e^z$$

で与えられる.

この特殊解は, 微分方程式の解であるが, コーシー条件は満たしていない.

例 **3.15** 次のコーシー問題の特殊解を求めよ.

$$\begin{cases} (D-2)(D+3)u(z) = e^{2z} \\ u(0) = c_0,\ u'(0) = c_1 \end{cases}$$

```
[467] taji_alc.solve_ode_cp_ps((x-2)*(x+3),z,[[x-2,1]]);
[[x+3,1/25],[x-2,1/5*z-1/25]]
```

コーシー問題の特殊解は,

$$v(z) = \frac{1}{25}e^{-3z} + (\frac{1}{5}z - \frac{1}{25})e^{2z}$$

で与えられる.

3.9 fbt

例 3.16 次の代数的局所コホモロジー類のフーリエ・ボレル変換を求めよ.

(1) $[\frac{1}{x^2}]$ (2) $[\frac{1}{x-1}]$ (3) $[\frac{3x^3-11x^2+19x-23}{(x-3)(x-1)^3}]$ (4) $[\frac{3x^2-1}{x^3-x-1}]$
(5) $[\frac{4x^{11}+8x^{10}-4x^8+8x^7-96x^6-8x^4+4x^3+24x^2-4}{(x^4+1)^3}]$

解
(1)

```
[638] taji_alc.fbt(1,[[x,2]],z);
[[x,z]]
```

$$FB([\frac{1}{x^2}])(z) = \mathrm{Res}(\frac{1}{x^2}e^{zx}dx) = z.$$

(2)

```
[639] taji_alc.fbt(1,x-1,z);
[[x-1,1]]
```

$$FB([\frac{1}{x-1}])(z) = \mathrm{Res}(\frac{1}{x-1}e^{zx}dx) = e^z.$$

(3)

```
[640] taji_alc.fbt(3*x^3-11*x^2+19*x-23,[[x-3,1],[x-1,3]],z);
[[x-1,3*z^2+1],[x-3,2]]
```

$$FB([\frac{3x^3-11x^2+19x-23}{(x-3)(x-1)^3}])(z) = \mathrm{Res}(\frac{3x^3-11x^2+19x-23}{(x-3)(x-1)^3}e^{zx}dx) = 2e^{3z}+(3z^2+1)e^z.$$

(4)

```
[641] taji_alc.fbt(3*x^2-1,[[x^3-x-1,1]],z);
[[x^3-x-1,1]]
```

$$FB([\frac{3x^2-1}{x^3-x-1}])(z) = \mathrm{Res}(\frac{3x^2-1}{x^3-x-1}e^{zx}dx) = \sum_{x_1\in Z} e^{x_1 z}.$$

(5)

```
[642] taji_alc.fbt(4*x^11+8*x^10-4*x^8+8*x^7-96*x^6-8*x^4+4*x^3+24*x^2-4,[[x^4+1
,3]],z);
[[x^4+1,(z^2+1)*x+2*z+1]]
```

$$FB([\frac{4x^{11}+8x^{10}-4x^8+8x^7-96x^6-8x^4+4x^3+24x^2-4}{(x^4+1)^3}])(z) = \sum_{x_1\in Z}((z^2+1)x_1+2z+1)e^{x_1 z}.$$

3.10 invfbt

例 3.17 次の指数多項式の逆フーリエ・ボレル変換を求めよ.

(1) z (2) e^z (3) $2e^{3z}+(3z^2+1)e^z$ (4) $\displaystyle\sum_{x_1\in Z} e^{x_1 z},\ Z=\{x|x^3-x-1=0\}$

(5) $\displaystyle\sum_{x_1\in Z}((z^2+1)x_1+2z+1)e^{x_1 z},\ Z=\{x|x^4+1=0\}$

解

(1) z の指数多項式リスト `[[x,z]]` を入力して求める.

```
[688] taji_alc.invfbt([[x,z]],z|switch=1);
[1,[[x,2]]]
```

$$FB^{-1}(z)(x)=[\frac{1}{x^2}].$$

(2) e^z の指数多項式リスト `[[x-1,1]]` を入力して求める.

```
[689] taji_alc.invfbt([[x-1,1]],z|switch=1);
[1,[[x-1,1]]]
```

$$FB^{-1}(e^z)(x)=[\frac{1}{x-1}].$$

(3) $2e^{3z}+(3z^2+1)e^z$ の指数多項式リスト `[[x-3,2],[x-1,3*z^2+1]]` を入力して求める.

```
[690] taji_alc.invfbt([[x-3,2],[x-1,3*z^2+1]],z|switch=1);
[3*x^3-11*x^2+19*x-23,[[x-3,1],[x-1,3]]]
```

$$FB^{-1}(2e^{3z}+(3z^2+1)e^z)(x)=[\frac{3x^3-11x^2+19x-23}{(x-3)(x-1)^3}].$$

(4) $\sum_{x_1\in Z} e^{x_1 z}$ の指数多項式リスト `[[x^3-x-1,1]]` を入力して求める.

```
[691] taji_alc.invfbt([[x^3-x-1,1]],z|switch=1);
[3*x^2-1,[[x^3-x-1,1]]]
```

$$FB^{-1}(\sum_{x_1\in Z} e^{x_1 z})(x)=[\frac{3x^2-1}{x^3-x-1}].$$

(5) $\sum_{x_1\in Z}((z^2+1)x_1+2z+1)e^{x_1 z}$ の指数多項式リスト `[[x^4+1,(z^2+1)*x+2*z+1]]` を入力して求める.

```
[692] taji_alc.invfbt([[x^4+1,(z^2+1)*x+2*z+1]],z|switch=1);
[4*x^11+8*x^10-4*x^8+8*x^7-96*x^6-8*x^4+4*x^3+24*x^2-4,[[x^4+1,3]]]
```

$$FB^{-1}(\sum_{x_1\in Z}((z^2+1)x_1+2z+1)e^{x_1 z})(x)=[\frac{4x^{11}+8x^{10}-4x^8+8x^7-96x^6-8x^4+4x^3+24x^2-4}{(x^4+1)^3}].$$

参考文献

[1] 森本光生：佐藤超函数入門, 共立出版, 1976.

[2] 田島慎一, 中村弥生：微分作用素を用いた有理関数の留数計算と Horowitz's algorithm, 京都大学数理解析研究所講究録 1038 「数式処理における理論と応用の研究」(1998), 23-30.

[3] 田島慎一, 中村弥生：D-加群を用いた留数計算アルゴリズムの局所化, 数式処理 7 (1999), 2-10.

[4] Tajima, S. and Nakamura, Y. : An algorithm for computing the residue of a rational function via D-Modules, Jasai Mathematical Monographes 2 - Computer Algebra - (2000), 149-158.

[5] Nakamura, Y. and Tajima, S. : Residue calculus with differential operator, Kyushu Journal of Mathematics Vol.54 No.1 (2000), 127-138.

[6] 田島慎一：多変数補間問題とホロノミック D-加群, 千葉大学数学講究録 3 「代数解析学の諸問題」 (1999), 73-94.

[7] 田島慎一：代数的局所コホモロジー類のローラン展開と L.Ehrenpreis の Noether 作用素, 京都大学数理解析研究所講究録 1138「数式処理における理論と応用の研究」(2000), 87–95.

[8] Tajima, S. : Grothendieck duality and Hermite-Jacobi formulas, Proc. Seventh International Conference on Several Complex Variables, in Finite or Infinite Dimensional Complex Analysis, (eds by Kajiwara, Li and Shon) Dekker, 503-509 (2000).

[9] Tajima, S. : An algorithm for computing the Noetherian operator representations and its applications to constant coefficients holonomic PDE's, Tools for Mathematical Modellings, (2001) St. Petersbourg, 154-160.

[10] Tajima, S. : Exponential polynomials and the Fourier-Borel transforms of algebraic local cohomology classes, in Microlocal Analysis and Complex Fourier Analysis, eds by T. Kawai and K. Fujita, World Scientific (2002), 284-296.

[11] 田島慎一：確定特異点型ホロノミック系の零次元代数的局所コホモロジー解, 京都大学数理解析研究所講究録 1336「双曲形方程式と非正則度」(2003), 121-132.

[12] Tajima, S. : On Noether differential operators attached to a zero-dimensional primary ideal - shape basis case -, in Finite or Infinite Dimensional Complex Analysis and Applications (2005), Kyushu Univ. Press, 357-366.

[13] Tajima, S. : An algorithm for computing exponential polynomial solutions of constant coefficients holonomic PDE's -generic case-, in Methods of Complex and Clifford Analysis, eds Le Hung Son, W. Tutschke and S. Jain, SAS International Pub. (2006), 335-344.

[14] 田島慎一：Holonomic な定数係数線形偏微分方程式系と Grothendieck duality, 京都大学数理解析研究所講究録 1509 「積分核の代数解析的研究」(2006), 1–23.

[15] 田島慎一：剰余体 $\mathbb{Q}[x]/\langle f(x)\rangle$ における逆冪計算, 京都大学数理解析研究所講究録 1514 「Computer Algebra-Design of Algorithms, Implementations and Applications」(2006), 171–175.

[16] 田島慎一：一変数留数計算アルゴリズムについて,
京都大学数理解析研究所講究録 1509 「積分核の代数解析的研究」(2006), 24–50.

[17] 加藤涼香, 田島慎一：有理関数のローラン展開アルゴリズムと代数的局所コホモロジー,
京都大学数理解析研究所講究録 1395 「Computer Algebra-Design of Algorithms, Implementations and Applications」(2004), 50–56.

[18] 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.

[19] 庄司卓夢, 田島慎一：多項式剰余公式の計算アルゴリズム,
京都大学数理解析研究所講究録 1514 「Computer Algebra-Design of Algorithms, Implementations and Applications」(2006), 108–114.

[20] 庄司卓夢, 田島慎一：コーシー問題の解法とアルゴリズム,
京都大学数理解析研究所講究録 1568 「Computer Algebra-Design of Algorithms, Implementations and Applications」(2007), 67-73.

[21] 庄司卓夢, 田島慎一：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, 京都大学数理解析研究所講究録 「数式処理研究の新たな発展」掲載予定.

[22] Noro, M. et al：計算機代数システム Risa/Asir, `http://www.math.kobe-u.ac.jp/Asir`