

**norow\_mwl**

---

norow\_mwl User's Manual  
Edition 1.0  
Nov 2009

**by Masayuki Noro**

---



# 1 MWL 関連計算パッケージ `norow_mwl.rr`

このマニュアルでは, `asir-contrib` パッケージに収録されている, MWL 関連計算パッケージ '`norow_mwl.rr`' について解説する. このパッケージを使うには, まず '`norow_mwl.rr`' をロードする.

```
[1518] load("norow_mwl.rr");
```

このパッケージの関数を呼び出すには, 全て `mwl.` を先頭につける.

## 1.1 MWL 関連計算

### 1.1.1 `mwl.pdecomp`, `mwl.pdecomp_ff`

```
mwl.pdecomp(ideal,varlist[|gbcheck=yesno,f4=yesno])
mwl.pdecomp_ff(ideal,varlist,mod[|gbcheck=yesno,f4=yesno]) :: 0 次元イデア
ルideal をいくつかのイデアルの共通部分として表す.
```

`return`      二要素からなるリスト

`ideal`        多項式リスト

`varlist`      変数リスト

`mod`          素数

`yesno`        0 または 1

- 0 次元イdeal `ideal` の各変数の最小多項式を既約分解し, 各既約成分を重複度つきで `ideal` に添加することを繰り返して得られたイdealのリストを第一要素, `ideal` の全次数逆辞書式順序に関するグレブナー基底を第二要素とするリストを返す.
- `mwl.pdecomp` は有理数体上, `mwl.pdecomp_ff` は  $\text{GF}(\text{mod})$  (位数 `mod` の有限体) 上での分解を行う.
- 出力リストの要素であるイdealは, 必ずしも準素とは限らないが, 各変数の最小多項式が既約多項式のべきとなっているので, 準素に近いことが期待される. これを準素分解の入力とすることで, もとのイdealをそのまま準素分解するより効率よく準素分解できることが期待される.
- デフォルトでは, グレブナー基底計算には `nd_gr_trace` が用いられるが, オプション `f4=1` を指定すると `nd_f4_trace` が用いられる.
- オプション `gbcheck=0` を指定すると, グレブナー基底計算におけるチェックが省かれる. この場合, 大変小さい確率で正しい結果が出力されないことが有り得るが, ほとんどの場合は正しいので, 予備的な実験を繰り返す場合に有用である. 出力されたイdealリストの全ての共通部分が入力と一致すれば, 出力が入力イdealの分解になっていることは保証される.

```
[1520] load("norow_mwl.rr");
```

```
[1554] B=[(x+y+z)^2*(x+y-z)^2,(x+y*z)^2*(x-y*z)^2,
(x^2+y^2+z^2)^2*(x^2-y^2-z^2)^2]$
```

```
[1555] V=[x,y,z]$
```

```
[1556] L=mwl.pdecomp(B,V)$
```

```
[1557] C=L[0]$
```

```

[1558] G=L[1]$
[1559] length(C);
5
[1560] C0=primadec(C[0],V)$
[1561] C0[0];
[[x^2+(2*y-2*z)*x+y^2-2*z*y+z^2,...],[z^2+z+1,y-z-1,x+1]]
[1562] CM=mwl.pdecomp_ff(B,V,31991|f4=1)$
[1563] length(CM[0]);
5

```

### 1.1.2 mwl.generate\_coef\_ideal

`mwl.generate_coef_ideal(f[|simp=yesno])`

::  $x, y, t$  の多項式  $f$  の多項式零点  $(x(t), y(t))$  の係数の満たす方程式のイデアルを生成する

*return*      多項式リストと変数リストのペアからなるリスト

*f*            多項式

*yesno*       0 または 1

- $f(x, y, t) = (y^2 + c_1(t)xy + c_3(t)y) - (x^3 + c_2(t)x^2 + c_4(t)x + c_6(t))$  に対し,  $x = a_m t^m + \dots + a_0$ ,  $y = b_n t^n + \dots + b_0$  ( $a_i, b_j$  は未定係数) を  $f$  に代入したときの, 各  $t$  のべきの係数を並べたリスト *ideal* および, 未定係数のリスト  $vlist = [b_0, \dots, b_n, a_0, \dots, a_m]$  のペア  $[ideal, vlist]$  を返す.
- 各  $x, y$  の次数は,  $f$  から自動的に決定される.
- オプション *simp=1* が指定された場合,  $a_m^3 - b_n^2$  が *ideal* に含まれている場合には, 新しい変数  $v$  を導入し,  $a_m = v^2, b_n = v^3$  により  $a_m, b_n$  を消去した結果を返す.

```

[1519] load("noromwl.rr")$
[1553] F=y^2-(x^3-x+t^2)$
[1554] L=mwl.generate_coef_ideal(F);
[[b3^2-a2^3,2*b3*b2-3*a2^2*a1,2*b3*b1+b2^2-3*a2^2*a0-3*a2*a1^2,...],
[b3,b2,b1,b0,a2,a1,a0]]
[1555] L=mwl.generate_coef_ideal(F|simp=1);
[[-3*a1*v^4+2*b2*v^3,-3*a0*v^4+2*b1*v^3-3*a1^2*v^2+b2^2,...],
[b2,b1,b0,a1,a0,v]]

```

# Index

(インデックスがありません)

(インデックスがありません)

## 簡単な目次

1	MWL 関連計算パッケージ <code>norow_mwl.r</code> . . . . .	1
	Index . . . . .	3

## 目次

<b>1</b>	<b>MWL 関連計算パッケージ <code>norow_mwl.rr</code> .....</b>	<b>1</b>
1.1	MWL 関連計算 .....	1
1.1.1	<code>mwl.pdecomp</code> , <code>mwl.pdecomp_ff</code> .....	1
1.1.2	<code>mwl.generate_coef_ideal</code> .....	2
	<b>Index .....</b>	<b>3</b>

