Licence Informatique — Systèmes polynomiaux, que signifie : résoudre ? — Juin 2007
Durée 3 heures — documents et calculatrices interdits

---

1 Bases de Gröbner (14 points)

On considère le système suivant et on note $\mathscr{I}$ l'idéal qu'il engendre dans l'anneau $\mathbb{Q}[x, y]$:

$$\mathscr{S} \left\{ \begin{array}{l} y^2 - 2\,x = 0, \\ x^2 - y - 1 = 0. \end{array} \right.$$

Question 1 [2 pts]. Il existe un ordre admissible vis–à–vis duquel on peut vérifier, sans faire de calcul, que $\mathscr{S}$ est une base de Gröbner. Quel ordre admissible ? Justifier. Présenter $\mathscr{S}$ sous la forme d'un système de réécriture pour cet ordre.

SOLUTION. L'ordre « tdeg (y,x) ». Les termes de tête sont disjoints pour cet ordre. Le système est donc une base de Gröbner d'après le premier critère de Buchberger.

$$\mathscr{S} \left\{ \begin{array}{l} y^2 \to 2\,x, \\ x^2 \to y + 1. \end{array} \right.$$

Question 2 [1 pt]. Dessiner l'escalier défini par la base de Gröbner $\mathscr{S}$. En déduire le nombre de solutions (comptées avec multiplicités) de $\mathscr{I}$ dans $\mathbb{C}^2$.

SOLUTION. On trouve quatre termes irréductibles. Le système admet donc quatre solutions complexes, comptées avec multiplicités.

Question 3 [2 pts]. On suppose le paquetage *Groebner* chargé en mémoire. Donner une suite de commandes MAPLE permettant de calculer la base de Gröbner réduite $\mathscr{G}$ de l'idéal $\mathscr{I}$ pour l'ordre « plex (y,x) » puis de calculer la forme normale du monôme $2\,x\,y$ par $\mathscr{G}$.

SOLUTION.

```
G := gbasis ([y^2 - 2*x, x^2 - y - 1], plex (y,x));
normalf (2*x*y, G, plex (y,x));
```

1.1 Application de l'algorithme de Buchberger

Question 4 [2 pts]. Calculer, en appliquant l'algorithme de Buchberger, une base de Gröbner de $\mathscr{I}$ pour l'ordre « plex (y,x) ». À chaque itération, indiquer l'ensemble des paires critiques qui restent à traiter.

SOLUTION. On transforme $\mathscr{S}$ en un système de réécriture pour l'ordre « plex (y,x) ».

$$\left\{\begin{array}{ll}(1) & y^2 \rightarrow 2\,x,\\ (2) & y \rightarrow x^2 - 1.\end{array}\right.$$

L'ensemble des paires critiques à traiter est $\{(1,\,2)\}$. Le S–polynôme défini par la paire vaut, après réduction par $\mathscr{S}$: $x^4 - 2\,x^2 - 2\,x + 1$.On obtient un nouveau système

$$\mathscr{G}\left\{\begin{array}{ll}(1) & y^2 \rightarrow 2\,x,\\ (2) & y \rightarrow x^2 - 1,\\ (3) & x^4 \rightarrow 2\,x^2 + 2\,x - 1.\end{array}\right.$$

D'après le premier critère de Buchberger, il est inutile de former de nouvelles paires critiques. Le système $\mathscr{G}$ est donc une base de Gröbner de $\mathscr{I}$.

Question 5 [1 pt]. La base obtenue à la fin de la question précédente est–elle réduite ? Si non, la réduire.

SOLUTION. Non. On la réduit en supprimant la règle (1) (les coefficients numériques sont normalisés et les parties droites des règles sont irréductibles).

1.2 Programmation et FGLM

Pour implanter l'algorithme FGLM en MAPLE, on se donne, outre les paramètres formels *base1*, *ordre1* et *ordre2*, les variables suivantes :

1. une variable *base2* contenant la base de Gröbner en cours de construction,
2. une variable *tetes2* contenant la liste des termes de tête des éléments de *base2*,
3. une variable *AEnumerer* contenant une liste de termes à énumérer.

Notons X l'ensemble des indéterminées dont dépend le système à traiter. À chaque itération, on extrait le plus petit élément t de la liste *AEnumerer* pour l'ordre admissible *ordre2* et on ajoute à la liste *AEnumerer* tous les termes de la forme $x\,t$, où $x \in X$, qui ne sont multiples d'aucun des termes de *tetes2*. L'algorithme effectue ensuite d'autres opérations qui ne nous intéressent pas ici.

Question 6 [2 pts]. Écrire en MAPLE une fonction *est_irreductible*, paramétrée par un terme t et une liste de termes *tetes*, qui retourne *true* si t n'est le multiple d'aucun terme de *tetes* et *false* sinon.

On pourra utiliser la fonction *lcm* qui retourne le plus petit multiple commun des deux termes qui lui sont passés en paramètre.

SOLUTION.

```
est_irreductible := proc (t, tetes)
    local i;
    for i from 1 to nops (tetes) do
        if lcm (t, tetes [i]) = t then
```

```
            return false
        fi
    od;
    true
end:
```

Question 7 [2 pts]. Écrire en MAPLE une fonction *nouveaux_AEnumerer*, paramétrée par un terme t, la liste X des indéterminées du problème, la liste *tetes2* et qui retourne la liste des termes de la forme $x\,t$, où $x \in X$, et qui ne sont pas des multiples de l'un des termes de *tetes2*.

SOLUTION.

```
nouveaux_AEnumerer := proc (t, X, tetes2)
    local i, L;
    L := [];
    for i from 1 to nops (X) do
        if est_irreductible (X[i]*t, tetes2) then
            L := [ op (L), X[i]*t ]
        fi
    od;
    L
end:
```

1.2.1 Questions subsidiaires

Question 8 [1 pt]. La liste *AEnumerer* est en fait une *file avec priorité*. Quelle structure de données, étudiée dans le cours ALGO, permet d'implanter efficacement cette file ? Préciser ce qui doit l'être.

SOLUTION. Un tas binaire. les éléments du tas sont des termes. Tout terme est inférieur ou égal à ses fils gauche et droit vis–à–vis de l'ordre admissible *ordre2*.

Question 9 [1 pt]. Vu la façon dont on implante la fonction *nouveaux_AEnumerer*, il est possible que la liste *AEnumerer* (après ajout des nouveaux termes à énumérer) contienne des doublons. Indiquer, en quelques phrases, une méthode permettant de contourner ce problème d'existence de doublons, sans engendrer de lignes inutiles dans la matrice M et sans perdre la bonne complexité en temps dans le pire des cas offerte par la file avec priorité.

SOLUTION. On supprime les doublons au moment où on extrait l'élément minimal du tas. Si le tas est implanté par un tableau T, le plus petit terme figure en T_1 et on peut facilement consulter sa valeur. On peut donc extraire le plus petit élément du tas tant qu'on a affaire à un doublon.

2 Le projet (6 points)

Question 10 [2 pts]. Figure 1, on a mis des croix pour les points critiques et des ronds pour les points réguliers. Connecter les points (utiliser la figure fournie en annexe).

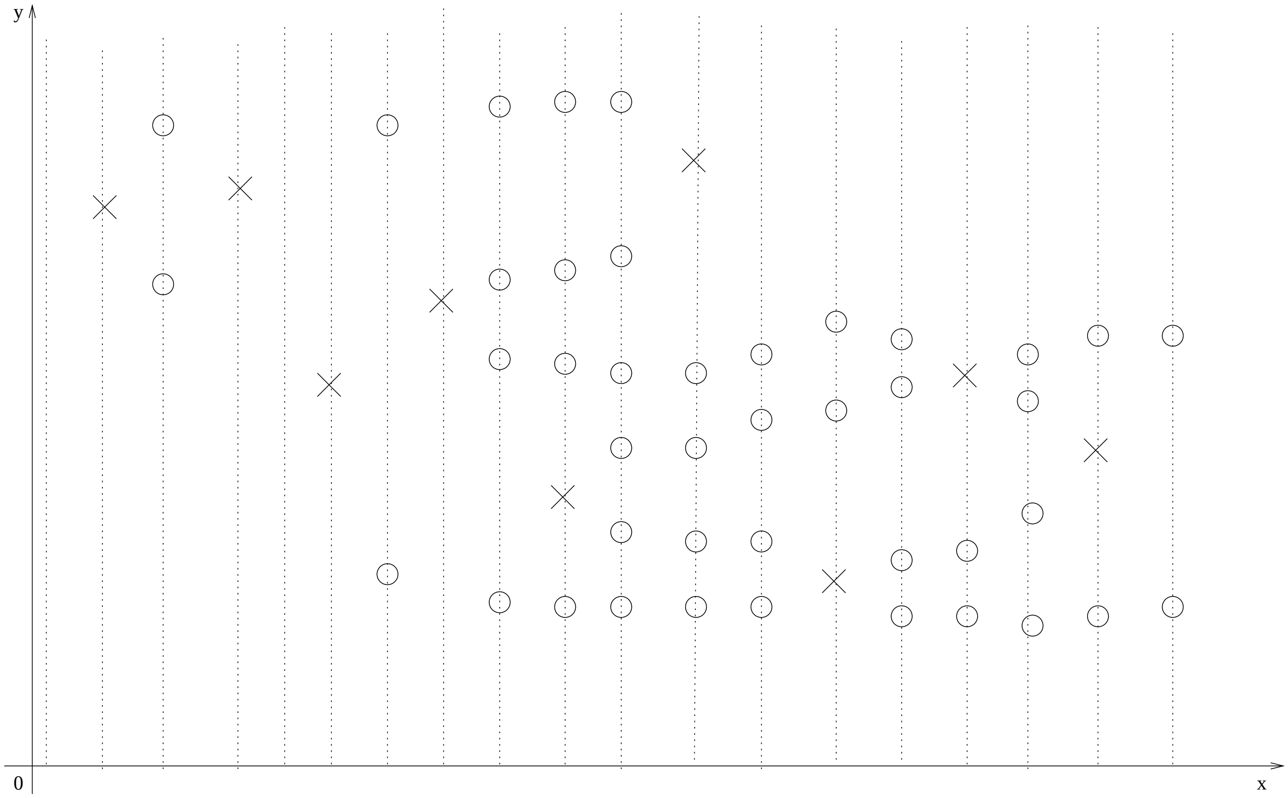

FIG. 1 – Points critiques et réguliers d'une courbe algébrique plane

2.1 Recherche de points fixes

Lorsqu'on fait ce qu'on appelle *l'analyse qualitative* d'une équation différentielle, la première information à déterminer est l'ensemble de ses *points fixes* (ou solutions constantes). Considérons par exemple l'équation différentielle

$$\dot{x}(t) = (x(t) - 1)\,(x(t) - 3)$$

où $x(t)$ désigne la fonction inconnue et $\dot{x}(t)$ sa dérivée. Ses points fixes sont les deux fonctions

$$x(t) = 1 \quad \text{et} \quad x(t) = 3.$$

Ils sont les solutions de l'équation :

$$0 = (x - 1)\,(x - 3).$$

De façon générale, les points fixes d'une équation différentielle

$$\dot{x}(t) \quad = \quad F(x(t)) \tag{1}$$

sont les solutions de l'équation non différentielle $F(x) = 0$.

Question 11 [1 pt]. Citer un algorithme vu en cours permettant de déterminer les points fixes d'une équation différentielle du type (1) dans le cas où $F \in \mathbb{Q}[x]$.

SOLUTION. L'algorithme de Vincent–Collins–Akritas.

2.2 Analyse de bifurcation

On s'intéresse maintenant à une équation différentielle

$$\dot{x}(t) \;=\; F(x(t),\, \lambda) \tag{2}$$

où F est un polynôme en deux indéterminées et λ est un paramètre. On voudrait déterminer les valeurs de λ pour lesquelles le nombre de points fixes de l'équation (2) change. Ces valeurs de λ sont des cas particuliers de *points de bifurcation* de l'équation différentielle.

Pour fixer les idées, on peut considérer l'équation suivante, où $x(t)$ représente une quantité de poissons dans l'océan, et où le paramètre λ représente une quantité constante pêchée.

$$\dot{x}(t) = x(t)\,(1 - x(t)) - \lambda.$$

Les points fixes de l'équation sont les solutions de l'équation

$$0 = x\,(1 - x) - \lambda.$$

On calcule le discriminant $\Delta = 1 - 4\,\lambda$. On voit que l'équation différentielle admet deux points fixes si $\lambda < 1/4$, un seul si $\lambda = 1/4$ et aucun si $\lambda > 1/4$. La valeur $\lambda = 1/4$ est un point de bifurcation de l'équation différentielle. On peut montrer (ça sort du cadre de l'épreuve), que la population de poisson disparaît si la quantité pêchée dépasse le point de bifurcation.

Question 12 [3 pts]. Expliquer en quelques phrases comment obtenir les points de bifurcation d'une équation différentielle du type (2) en utilisant les techniques mises en œuvre dans le cadre du projet.

SOLUTION. Il suffit de procéder à l'analyse topologique de la courbe implicite $F(x, \lambda) = 0$ en prenant λ comme abscisse (en ordonnée ça ne marche pas). Les points de bifurcation sont les points critiques. Le nombre de points fixes entre deux points de bifurcation est donné par le nombre de points réguliers entre deux points critiques.

3 Petites questions (7 points)

Question 13 [2 pts]. Donner la suite des triplets calculée par l'algorithme d'Euclide étendu, appliqué à $a = 113$ et $b = 45$. En déduire une identité de Bézout entre a et b.

SOLUTION. Identité de Bézout : $2 \times 113 - 5 \times 45 = 1$.

```
U[1] := [1, 0, 113];
                              U[1] := [1, 0, 113]

U[2] := [0, 1, 45];
                              U[2] := [0, 1, 45]

U[3] := U[1] - 2*U[2];
                              U[3] := [1, -2, 23]

U[4] := U[2] - U[3];
                              U[4] := [-1, 3, 22]

U[5] := U[3] - U[4];
                              U[5] := [2, -5, 1]
```

Question 14 [1 pt]. Quel est le nombre rationnel équivalent à 45 modulo 113, déterminé par l'algorithme de Wang ?

SOLUTION. Il s'agit de $-1/5$.

Question 15 [2 pts]. Pour remplir un tableau de taille $n \times m$ avec des 1, en une seule boucle, un étudiant propose la solution suivante. Cette solution est–elle correcte ? seulement dans certains cas ? pas du tout ? Justifier.

```
for i from 1 to m*n do
    lig := irem (i, n);
    col := irem (i, m);
    M [lig, col] := 1;
od
```

SOLUTION. La boucle n'est correcte que si $n \wedge m = 1$. D'après le théorème chinois, l'application $\mathbb{Z}/n \times m\mathbb{Z} \to \mathbb{Z}/n\mathbb{Z} \times \mathbb{Z}/m\mathbb{Z}$ est bijective si et seulement si $n \wedge m = 1$.

Question 16 [2 pts]. En appliquant un changement de variables simple ainsi que la règle des signes de Descartes, prouver que le polynôme suivant admet une unique racine supérieure à 1 :

$$P(x) = x^4 - 2\,x^2 - 2\,x + 1.$$

SOLUTION. On effectue le changement de variables $x = y + 1$. Les racines supérieures à 1 de $P(x)$ sont en bijection avec les racines réelles positives de $Q(y) = P(y + 1)$. Le nombre de variations de signe de $Q(y)$ vaut 1. D'après la règle des signes, il admet au moins une racine réelle positive. Comme ses coefficients de tête et de queue sont de signes différents, il admet exactement une racine réelle positive. CQFD.

$$Q(y) = y^4 + 4\,y^3 + 4\,y^2 - 2\,y - 2.$$

Annexe : l'algorithme FGLM

Principe. L'algorithme FGLM (des initiales de ses inventeurs : Jean-Charles Faugère, Patricia Gianni, Daniel Lazard et Teo Mora) est un algorithme de changement d'ordre admissible pour bases de Gröbner. Il résout efficacement le problème suivant : étant donnés

1. une base de Gröbner *base*$_1$ d'un idéal $\mathscr{I}$ pour un ordre admissible *ordre*$_1$ et
2. un ordre admissible *ordre*$_2 \neq$ *ordre*$_1$,

calculer la base de Gröbner réduite *base*$_2$ de $\mathscr{I}$ pour l'ordre admissible *ordre*$_2$. L'algorithme ne s'applique pas à toutes les bases de Gröbner : il ne s'applique que si l'idéal $\mathscr{I}$ admet un nombre fini de solutions complexes (le cas abordé en cours).

Illustration sur un exemple. La variable *base*$_1$ contient une base de Gröbner pour l'ordre admissible *ordre*$_1$.

```
with (Groebner):
with (LinearAlgebra):
p1 := 2*x*y^2  - 5*y^2 - y + 2;
p2 := x^2*y + x - y - 4;
ordre1 := plex (y, x);
base1 := gbasis ([p1, p2], ordre1);
                 2             4      3                             3      2
    [-74 - 29 x  + 71 x + 2 x  + 3 x , 285 x + 315 y - 368 - 26 x  - 83 x ]
```

Voici les équations de la base, sous la forme de règles de réécriture, calculées avec la fonction *regle_reec* donnée au cours 11.

```
map (regle_reec, base1, ordre1);
     4              2                  3        19      368   26   3   83   2
   [x  = 37 + 29/2 x  - 71/2 x - 3/2 x , y = --- x + --- + --- x  + --- x ]
                                              21      315   315      315
```

D'un point de vue calculatoire, l'algorithme exécute une boucle qui énumère les termes par ordre croissant pour l'ordre admissible *ordre*$_2$ en commençant à : $1 = t_0 < t_1 < t_2 < t_3 < \cdots$ À chaque itération i, la forme normale $\mathrm{NF}(t_i)$ du terme courant t_i est calculée, pour l'ordre admissible *ordre*$_1$ (on se sert de la base de Gröbner connue) et l'algorithme cherche s'il existe des rationnels $\lambda_0, \ldots, \lambda_i$ tels que $\lambda_0\ \mathrm{NF}(t_0) + \lambda_1\ \mathrm{NF}(t_1) + \cdots + \lambda_i\ \mathrm{NF}(t_i) = 0$. Dès qu'une telle combinaison linéaire est trouvée, l'algorithme en déduit que le polynôme $\lambda_0\, t_0 + \lambda_1\, t_1 + \cdots + \lambda_i\, t_i \in \mathscr{I}$ et il le rajoute à la base de Gröbner *base*$_2$ en construction.

Recherche du premier polynôme. Voici comment procéder concrètement. La variable *ordre*$_2$ reçoit l'ordre admissible cible *ordre*$_2$. Initialement, la base de Gröbner *base*$_2$ est vide. Affectons à T l'ensemble des termes irréductibles par *base*$_1$ et à *AEnumerer* la liste des (mettons) cinq premiers termes pour l'ordre *ordre*$_2$ (ils sont énumérés par ordre croissant). On voit que les formes normales des termes à énumérer ne dépendent que des termes présents dans T.

```
ordre2 := tdeg (y, x);
base2 := [];
T := [1, x, x^2, x^3];
AEnumerer := [1, x, y, x^2, x*y];
```

```
for i from 1 to nops (AEnumerer) do
    normalf (AEnumerer [i], base1, ordre1)
od;
                                        1

                                        x

                           19      368    26    3    83    2
                          --- x + --- + --- x  + --- x
                           21      315    315        315

                                        2
                                       x

                          92    2    37       44    3    962
                          --- x  - -- x + --- x  + ---
                          315       21      315       315
```

Ces formes normales sont–elles linéairement dépendantes sur $\mathbb{Q}$? Pour le savoir, on les voit comme des vecteurs dans la base T, on forme une matrice $m \times n$ où m désigne le nombre de formes normales et n le nombre d'éléments de T.

```
m := nops (AEnumerer);
n := nops (T);
M := Matrix (m, n);
for lig from 1 to m do
    nf := normalf (AEnumerer [lig], base1, ordre1);
    koeffs := [ coeffs (nf, indets (nf), 'termes') ];
    termes := [ termes ];
    for j from 1 to nops (termes) do
        member (termes [j], T, 'col');
        M [lig, col] := koeffs [j];
    od;
od;
M;
                          [ 1      0      0      0 ]
                          [                        ]
                          [ 0      1      0      0 ]
                          [                        ]
                          [368    -19    83     26 ]
                          [---    ---    ---    ---]
                          [315    21     315    315]
                          [                        ]
                          [ 0      0      1      0 ]
                          [                        ]
                          [962    -37    92     44 ]
                          [---    ---    ---    ---]
                          [315    21     315    315]
```

En appliquant un pivot de Gauss, on voit que la dernière ligne est identiquement nulle : les formes normales sont bien linéairement dépendantes[1].

```
GaussianElimination (M);
                                     [1    0     0      0 ]
                                     [                    ]
                                     [0    1     0      0 ]
                                     [                    ]
                                     [          83     26 ]
                                     [0    0    ---    ---]
                                     [          315    315]
                                     [                    ]
                                     [                 -26]
                                     [0    0     0     ---]
                                     [                 83 ]
                                     [                    ]
                                     [0    0     0      0 ]
```

On obtient les coefficients λ_i par une petite manipulation d'algèbre linéaire : on borde la matrice M par la matrice identité $m \times m$ et on applique à nouveau le pivot de Gauss. Les coefficients recherchés sont sur les m dernières colonnes de la dernière ligne de la matrice ainsi obtenue.

```
M1 := < M | IdentityMatrix (m) >;
                      [ 1      0      0      0     1    0    0    0    0]
                      [                                                 ]
                      [ 0      1      0      0     0    1    0    0    0]
                      [                                                 ]
                      [368    -19    83     26                          ]
                      [---    ---    ---    ---    0    0    1    0    0]
              M1 := [315    21     315    315                         ]
                      [                                                 ]
                      [ 0      0      1      0     0    0    0    1    0]
                      [                                                 ]
                      [962    -37    92     44                          ]
                      [---    ---    ---    ---    0    0    0    0    1]
                      [315    21     315    315                         ]


M2 := GaussianElimination (M1);
                  [1    0     0      0      1       0       0       0      0]
                  [                                                         ]
                  [0    1     0      0      0       1       0       0      0]
                  [                                                         ]
                  [          83     26     -368     19                      ]
                  [0    0    ---    ---    ----     --      1       0      0]
                  [          315    315    315      21                      ]
          M2 := [                                                         ]
                  [                 -26    368     -285    -315             ]
                  [0    0     0     ---    ---     ----    ----     1      0]
```

[1]Il n'est pas nécessaire de procéder à un pivot de Gauss puisque le nombre de lignes est supérieur au nombre de colonnes mais on ne tient pas compte de ce raisonnement qui est particulier à l'exemple.

```
               [                 83     83       83      83              ]
               [                                                         ]
               [                        -14              -22             ]
               [0    0     0      0     ---      3/13    ---      2/13    1]
               [                        13               13              ]

lambdas := Row (SubMatrix (M2, 1..m, n+1..n+m), -1);
                                [-14         -22         ]
                     lambdas := [---, 3/13, ---, 2/13, 1]
                                [13          13          ]
```

Il ne reste plus qu'à multiplier les coefficients λ_i par les termes t_i pour obtenir un premier polynôme de la base de Gröbner recherchée.

```
q1 := lambdas . Vector (m, AEnumerer);
                          14              22             2
                  q1 := - -- + 3/13 x - -- y + 2/13 x  + x y
                          13              13
base2 := [ op (base2), q1 ];
map (regle_reec, base2, ordre2);
                            14              22             2
                     [x y = -- - 3/13 x + -- y - 2/13 x ]
                            13              13
```

Recherche du deuxième polynôme. On cherche maintenant un deuxième polynôme de la base *base*$_2$. Pour cela, on commence par retirer le terme $x\,y$ de la liste *AEnumerer*. Les formes normales des termes restants sont linéairement indépendantes. Ensuite, on rajoute à cette liste, un à la fois, les termes supérieurs à $x\,y$ pour l'ordre *ordre*$_2$, <u>en évitant les multiples du terme $x\,y$</u>. On continue jusqu'à détecter une nouvelle dépendance linéaire. Sur l'exemple, on voit que la forme normale du premier terme considéré, y^2, dépend linéairement des autres formes normales.

```
AEnumerer := [1, x, y, x^2, y^2];
[M est calculée comme précédemment]
GaussianElimination (M);
                            [1    0     0      0 ]
                            [                    ]
                            [0    1     0      0 ]
                            [                    ]
                            [          83     26 ]
                            [0    0    ---    ---]
                            [          315    315]
                            [                    ]
                            [                -26]
                            [0    0     0    ---]
                            [                83 ]
                            [                    ]
                            [0    0     0      0 ]
```

Par le même procédé que précédemment, on obtient un nouveau polynôme de la base *base*$_2$.

```
M1 := < M | IdentityMatrix (m) >;
M2 := GaussianElimination (M1);
lambdas := Row (SubMatrix (M2, 1..m, n+1..n+m), -1);

                                  [-46  -10  -11  -4   ]
                       lambdas := [---, ---, ---, --, 1]
                                  [273  39   273  91   ]

q2  := lambdas . Vector (m, AEnumerer);
base2 := [ op (base2), q2 ];
map (regle_reec, base2, ordre2);

           14            22           2   2   46    10     11            2
    [x y = -- - 3/13 x + -- y - 2/13 x , y  = --- + -- x + --- y + 4/91 x ]
           13            13                   273   39     273
```

Recherche du troisième polynôme. On recommence : on retire le terme y^2 de la liste *AEnumerer*. On rajoute, en évitant les multiples de $x\,y$ et de y^2, le plus petit terme supérieur à y^2. Il s'agit de x^3. On trouve à nouveau une dépendance linéaire entre les formes normales et donc un nouveau polynôme de *base*$_2$.

```
AEnumerer := [1, x, y, x^2, x^3];
[M est calculée comme précédemment]
GaussianElimination (M);
                            [1    0     0      0 ]
                            [                    ]
                            [0    1     0      0 ]
                            [                    ]
                            [          83     26 ]
                            [0    0    ---    ---]
                            [          315    315]
                            [                    ]
                            [                 -26]
                            [0    0     0     ---]
                            [                 83 ]
                            [                    ]
                            [0    0     0      0 ]

M1 := < M | IdentityMatrix (m) >;
M2 := GaussianElimination (M1);
lambdas := Row (SubMatrix (M2, 1..m, n+1..n+m), -1);

                                 [184  -285  -315  83   ]
                      lambdas := [---, ----, ----, --, 1]
                                 [13    26    26   26   ]

q3 := lambdas . Vector (m, AEnumerer);
base2 := [ op (base2), q3 ];
map (regle_reec, base2, ordre2);

       14            22           2   2   46    10     11            2
[x y = -- - 3/13 x + -- y - 2/13 x , y  = --- + -- x + --- y + 4/91 x ,
```

```
       13            13                     273   39     273

    3      184   285     315     83  2
   x  = - --- + --- x + --- y - -- x ]
           13    26      26      26
```

Arrêt de l'algorithme. Il n'existe aucun terme supérieur à x^3 qui ne soit multiple d'aucun des termes de tête de la base *base*$_2$. L'algorithme FGLM s'arrête et retourne *base*$_2$. On peut vérifier par acquis de conscience qu'il s'agit bien de la même base que celle que l'algorithme de Buchberger aurait calculée.

```
basis2 := gbasis ([p1, p2], ordre2):
 map (regle_reec, basis2, ordre2);
       14            22           2   2   46    10     11            2
[x y = -- - 3/13 x + -- y - 2/13 x , y  = --- + -- x + --- y + 4/91 x ,
       13            13                     273   39     273

    3      184   285     315     83  2
   x  = - --- + --- x + --- y - -- x ]
           13    26      26      26
```

Annexe

Numéro de place (seul indice de recherche en cas de perte) :