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

Le sujet est noté sur 27 points. Il est composé de 18 questions, souvent indépendantes les unes des autres. Ne restez donc pas bloqué sur une question. Il n'y a plus de questions après la page 5. La dernière page de l'énoncé comporte une annexe qui doit être détachée et jointe à votre copie.

1 Bases de Gröbner (14 points)

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

$$\mathscr{S} \left\{ \begin{array}{l} y - 2\,x = 0, \\ x^2 - 3\,x + 2 = 0. \end{array} \right.$$

Question 1 [1 pt]. Présenter $\mathscr{S}$ sous la forme d'un système de réécriture pour l'ordre « plex (y,x) ». Montrer, sans faire de calcul, que $\mathscr{S}$ est une base de Gröbner pour cet ordre.

SOLUTION. Sous la forme d'un système de réécriture :

$$\left\{ \begin{array}{l} y \rightarrow 2\,x, \\ x^2 \rightarrow 3\,x - 2. \end{array} \right.$$

Les termes de tête sont disjoints. D'après le premier critère de Buchberger, $\mathscr{S}$ est une base de Gröbner.

Question 2 [1 pt]. Donner les solutions de $\mathscr{S}$ (ne pas détailler les calculs).

SOLUTION. On trouve

$$\begin{pmatrix} y \\ x \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \quad \begin{pmatrix} 4 \\ 2 \end{pmatrix}.$$

1.1 Application de MAPLE

Question 3 [2 pts]. Donner une suite de commandes MAPLE permettant de calculer la base de Gröbner réduite de $\mathscr{I}$ pour l'ordre « plex (x,y) ». Aucun paquetage n'est supposé chargé en mémoire.

SOLUTION. Voici les commandes :

```
with (Groebner):
gbasis ([y - 2*x, x^2 - 3*x + 2], plex (x,y));
```

1.2 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 (x,y) ». 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 (x,y) » :

$$\begin{cases} (1) & x \to (1/2)\, y, \\ (2) & x^2 \to 3\, x - 2. \end{cases}$$

L'ensemble des paires critiques à traiter est $\{(1,\,2)\}$. Le S–polynôme défini par la paire vaut, après réduction par $\mathscr{S}$: $y^2 - 6\, y + 8$. On obtient un nouveau système

$$\mathscr{G} \begin{cases} (1) & x \to (1/2)\, y, \\ (2) & x^2 \to 3\, x - 2, \\ (3) & y^2 \to 6\, y - 8. \end{cases}$$

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 $\mathscr{I}$ pour l'ordre « plex (x,y) ».

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 (2) (les coefficients numériques sont normalisés et les parties droites des règles sont irréductibles).

1.3 Application de l'algorithme FGLM

On souhaite appliquer l'algorithme FGLM (le document fourni avant l'examen est joint en annexe) pour calculer la base de Gröbner réduite de $\mathscr{I}$ pour l'ordre « plex (x,y) », en partant de la base de Gröbner $\mathscr{S}$ connue.

Question 6 [1 pt]. Quel est l'ensemble des termes dont dépendent les formes normales énumérées par FGLM sur cet exemple ?

SOLUTION. Il s'agit de $\{1,\, x\}$.

Question 7 [1 pt]. Donner les trois premiers termes énumérés par FGLM sur cet exemple, ainsi que leur forme normale par $\mathscr{S}$.

SOLUTION. Il s'agit de 1, y et y^2. Les formes normales sont (respectivement) 1, $2\, x$ et $12\, x - 8$.

Question 8 [1 pt]. Donner la matrice M correspondant à ces trois premiers termes (ne pas effectuer le pivot de Gauss). Bien indiquer à quoi correspondent les lignes et les colonnes de la matrice.

SOLUTION. Voici au moins la matrice :

$$M = \begin{pmatrix} 1 & 0 \\ 0 & 2 \\ -8 & 12 \end{pmatrix}.$$

1.4 Compréhension de FGLM

Il est écrit dans le document joint :

« 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 déduit que le polynôme

$$\lambda_0 \, t_0 + \lambda_1 \, t_1 + \cdots + \lambda_i \, t_i \in \mathscr{I}\,. \text{ »}$$

Question 9 [2 pts]. Justifier ce point de l'algorithme à partir de théorèmes et de propositions vues en cours. Énoncer les arguments clefs. Ne pas rédiger la preuve.

SOLUTION. Il y a deux arguments clefs :

1. Une combinaison linéaire sur $\mathbb{Q}$ de formes normales est une forme normale.
2. Un polynôme appartient à $\mathscr{I}$ si et seulement si sa forme normale est 0.

Question 10 [2 pts]. Il est affirmé dans le document joint que l'algorithme FGLM ne s'applique que si l'idéal $\mathscr{I}$ admet un nombre fini de solutions complexes. Quelle propriété les bases de de Gröbner de ces idéaux ont–elles de si important ? Quel rôle joue cette propriété dans l'algorithme ?

SOLUTION. Soit $\mathscr{I}$ un idéal et $\mathscr{G}$ une base de Gröbner quelconque de $\mathscr{I}$. Alors $\mathscr{I}$ admet un nombre fini de solutions complexes si et seulement si l'ensemble des termes irréductibles par $\mathscr{G}$ est fini. Cette propriété assure l'arrêt de l'algorithme FGLM.

2 Le projet (8 points)

Dans cette partie, toutes les courbes algébriques planes $P(x, y) = 0$ considérées satisfont les mêmes hypothèses que dans le projet : elles n'admettent aucune asymptote verticale et tous leurs points critiques ont des abscisses distinctes.

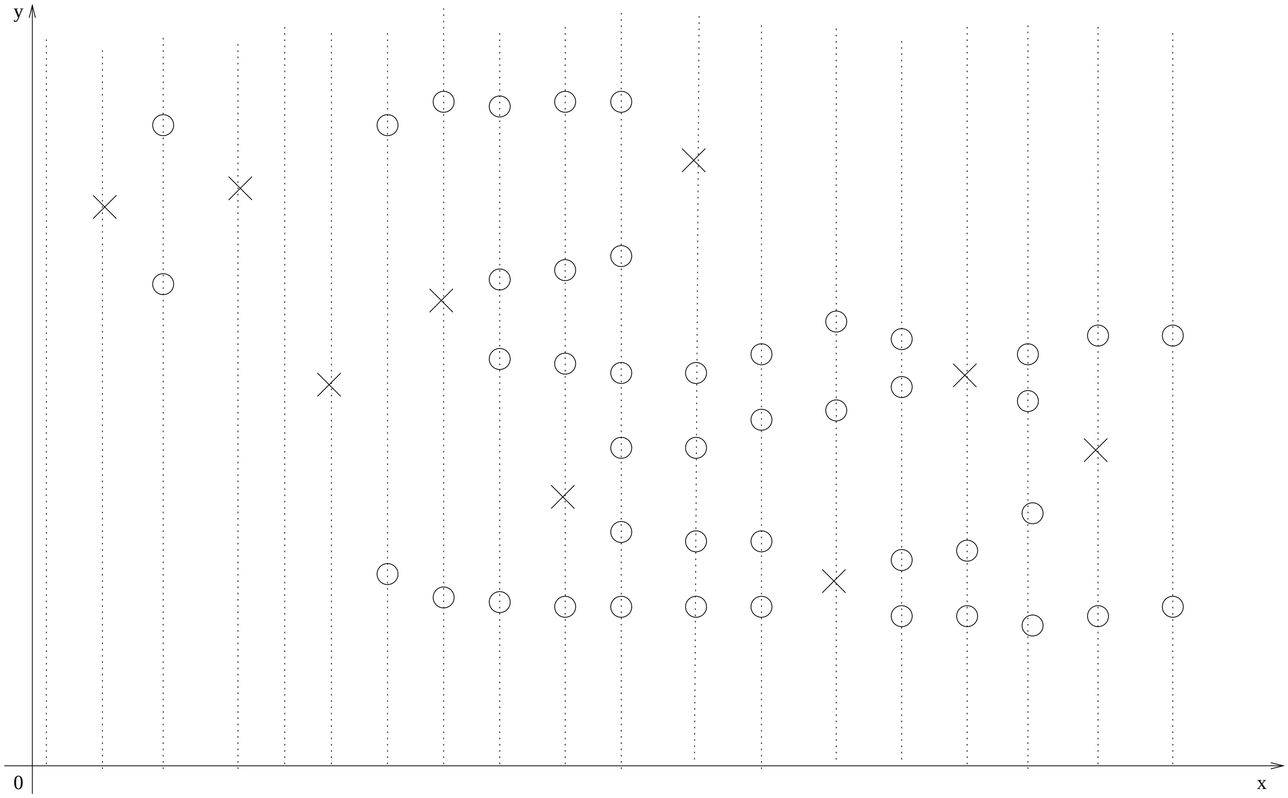

FIG. 1 – Points critiques et réguliers d'une courbe algébrique plane

Question 11 [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).

2.1 Analyse de courbe

Soit $P(x, y) = 0$ une courbe algébrique plane. On souhaite déterminer si une partie de la courbe se situe à gauche de l'axe vertical, c'est–à–dire s'il existe $x \leq 0$ et y quelconque tels que $P(x, y) = 0$. Un professeur et des étudiants débattent d'une solution.

2.1.1 Suggestion d'un étudiant

> *« Il suffit de déterminer si la courbe traverse l'axe des y, c'est–à–dire si le polynôme $P(0, y)$ admet des racines réelles. »*

Question 12 [1 pt]. Citer (ne pas donner le pseudo–code) un ou plusieurs algorithmes étudiés en cours permettant de déterminer si le polynôme $P(0, y)$ admet des racines réelles.

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

Question 13 [2 pts]. Est–ce que déterminer si la courbe traverse l'axe des y est complètement équivalent à déterminer si le polynôme $P(0, y)$ admet des racines réelles ? Si oui, justifier, si non indiquer la difficulté escamotée par l'étudiant ainsi que quelques pistes pour la prendre en compte (on ne demande pas une solution complète).

SOLUTION. Ce n'est pas tout–à–fait équivalent. Il faut s'assurer qu'au moins une racine réelle de $P(0, y)$ est de multiplicité impaire. Pour décider cela, on peut appliquer l'algorithme de Yun, qui fournit les multiplicités. Plus simplement, on peut s'assurer que le polynôme $P(0, y)$ est sans facteur carré (toutes les racines sont alors simples) en s'assurant que le pgcd de $P(0, y)$ et de sa dérivée est une constante. Mais on n'obtient alors qu'un test incomplet.

Question 14 [1 pt]. Indépendemment du problème soulevé par la question précédente, que pensez–vous de la suggestion de l'étudiant ? Justifier.

SOLUTION. Le test de l'étudiant est insuffisant. La courbe peut se situer totalement à gauche de l'axe vertical ou comporter plusieurs composantes connexes, à gauche et à droite de l'axe, sans jamais traverser cet axe.

2.1.2 Suggestion du professeur

« Il suffit, à peu de choses près, d'exploiter le résultat de l'analyse topologique de la courbe et, en particulier, l'ensemble de ses points critiques. »

Question 15 [2 pts]. Décrire (en Français) un algorithme qui résolve la question en se fondant sur l'idée du professeur. Ne pas justifier.

SOLUTION. On rappelle qu'on suppose qu'il n'y a aucune asymptote verticale. On commence par calculer les points critiques. Si l'un d'eux se situe à gauche de l'axe vertical alors la réponse est oui : une partie de la courbe se situe à gauche de l'axe vertical. Supposons que ce ne soit pas le cas. Alors la réponse est oui si et seulement si $P(x, y)$ intersecte une droite verticale tirée aléatoirement à gauche de l'axe vertical (opération implantée dans le projet).

3 Pgcd de polynômes (5 points)

La fonction de la Figure 2 calcule le développement en base b d'un entier naturel a. Elle retourne l'unique polynôme $P(x) = a_d\, x^d + \cdots + a_1\, x + a_0$ tel que $P(b) = a$ et $\underline{0 \leq a_i < b}$ pour $0 \leq i \leq d$.

Question 16 [2 pts]. Coder cette fonction en MAPLE. Utiliser les fonctions *iquo* et *irem*.

SOLUTION. Voici un codage possible :

```
devt_base := proc (a, x, b)
    local q, r;
    if a < b then
        return (a)
```

function devt_base (a, x, b)
begin
  if $a < b$ then
    return (a)
  else
    calculer le quotient q et le reste r de la division euclidienne de a par b
    return $(r + x \,*\,$ devt_base $(q,\, x,\, b))$
  fi
end

FIG. 2 – La fonction *devt_base*

```
    else
        q := iquo (a, b);
        r := irem (a, b);
        return (r + x * devt_base (q, x, b))
    fi
end:
```

3.1 Variante

On s'intéresse maintenant à une variante de la fonction précédente, appelée *devt_base2*, qui retourne l'unique polynôme $P(x) = a_d\,x^d + \cdots + a_1\,x + a_0$ tel que $P(b) = a$ et $\underline{-b/2 < a_i < b/2}$ pour $0 \leq i \leq d$. On suppose pour simplifier la base $\underline{b \text{ impaire}}$.

Question 17 [1 pt]. Quelles instructions suffit–il d'ajouter à la fonction *devt_base* après la division euclidienne, pour qu'elle satisfasse les spécifications ci–dessus ?

SOLUTION. Il suffit d'ajouter :

```
if r > b/2 then
    r := r - b;
    q := q + 1
fi;
```

3.2 Application au calcul du pgcd de deux polynômes

Soit à calculer le pgcd $P(x)$ de deux polynômes $F(x)$ et $G(x)$ de $\mathbb{Z}[x]$. On choisit un entier naturel impair b. On calcule $f = F(b)$ et $g = G(b)$. On calcule leur pgcd $p = f \wedge g$. Enfin on calcule $\overline{P} =$ *devt_base2* $(p,\, x,\, b)$. Si b est choisi convenablement, le polynôme $\overline{P}$ est, à un facteur entier près, le pgcd P recherché (un exemple est donné Figure 3).

```
> F := (2*x^2 - 3)*(x-1)*(x+2)^2;
                                2                      2
                      F := (2 x  - 3) (x - 1) (x + 2)

> G := (2*x^2 - 3)*(x+2)*(3*x-7)^3;
                               2                      3
                     G := (2 x  - 3) (x + 2) (3 x - 7)

> f := subs (x = 101, F);
                              f := 21641299100

> g := subs (x = 101, G);
                             g := 54490555566592

> p := igcd (f, g);
                                p := 8404388

> Pbar := factor (devt_base2 (p, x, 101));
                                      2
                        Pbar := 4 (2 x  - 3) (x + 2)
```

FIG. 3 – Calcul du pgcd de deux polynômes

Question 18 [2 pts]. Quelle condition la base b choisie doit–elle vérifier vis–à–vis de la hauteur de P ? Concrètement, comment peut–on choisir b ?

> SOLUTION. Il faut que b soit supérieure à deux fois la hauteur de P. On peut procéder comme en cours : calculer la borne de Landau et Mignotte sur les deux polynômes F et G et choisir b supérieur à deux fois le minimum de ces deux bornes.

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]
```

[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.

```
                  [                                                ]
                  [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]
            [                 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*$_2$, en évitant les multiples du terme $x\,y$. 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 pour la question 2

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