Majoration des termes de la matrice de Blankinship

PAR PATRICK TELLER

Il est bien connu que l'algorithme d'Euclide étendu (qui devrait s'appeler « algorithme de Blankinship ») permet, étant donnés deux entiers a et b, de déterminer deux entiers relatifs (u,v) tels que au+bv=d; parmi les couples (u,v) candidats celui qui est fourni par l'algorithme vérifie une propriété de minimalité qui a été démontrée dans [1].

Il est possible d'étendre l'algorithme de Blankinship pour l'appliquer à un p-uplet d'entiers naturels et obtenir une matrice $B\in \mathrm{SL}_p(Z)$ telle que $B^t(\ x_1, \ldots., x_p\ ) = {}^t(\ d, 0., \ldots,, 0\ )$, où d est le plus grand diviseur commun du p-uplet (x1,..,xp).

Les coefficients de la matrice B fournissent à la fois u p-uplet d'entiers rationnels (u1,..,up) tels que $\sum_{i=1\ldots p} x_i u_i = d$ et p-1 relations de la forme $\sum_{i=1\ldots p} y_{\mathrm{ji}} u_i = 0$.

Nous allons établir ici une propriété de minimalité sur les coefficients de la matrice B.

1 L'algorithme

On adoptera les notations suivantes pour une matrice A$\in\mathcal{M}_{p,p+1}(Z)$: les lignes de A seront désignées par $A_1, \ldots., A_p$, sa première colonne sera notée A_1, le nombre d'éléments non nuls de A^1 sera noté $|A^1|$, et ses éléments seront désignés par $a_{i,j}$.

Description de l'algorithme:

Initialisation : construire la matrice $A = (\ X_p\ \ I_p\ ) \in \mathcal{M}_{p,p+1}(Z)$ où $X_p = \begin{pmatrix} x1 \\ . \\ . \\ . \\ x_p \end{pmatrix}$ et I_p est la matrice unité.

Procédure réduction: si $|A^1|$>1

soit J={1,..,p}

tant que |J|>1

$\alpha = \min\{i, a_{i1} = \max\{a_{j1}, j\in J\}$, $\beta = \min\{i > \alpha, a_{i1} = \max\{a_{j1}, j\in J\backslash\{\alpha\}\}$,A[$\alpha$]:A[$\alpha$]-quotient(a$_{\alpha 1}$, $a_{\beta 1}$)A[β]; J:J$\backslash\{\alpha\}$

Algorithme principal

initialisation

Tant que $|A^1|$>1 A:réduction(A)

La suite des valeurs de $a_{\alpha(A),1}$ est une suite strictement décroissante et positive donc elle est finie.

Comme à chaque itération on aura ,quel que soit $i\in\{1, \ldots, p\}$, $A_i \begin{pmatrix} -1 \\ x1 \\ . \\ . \\ . \\ x_p \end{pmatrix} = 0$, lorsque l'algorithme s'achève la première colonne de A possède un seul élément non nul, c'est $a_{\alpha(A),1} =$ d

d=$\sum a_{\alpha(A)j} x_j$ et $\forall i \neq \alpha(A)$ 0=$\sum a_{\mathrm{ij}} x_j$.

D'où

Proposition 1.

La matrice $\tilde{A} = (a_{i,j})_{(i,j)\in\{1,\ldots,p\}\times\{2,\ldots,p+1\}}$ est, à l'ordre des lignes près, une matrice de Blankinship pour le p-uplet (x1,..,xp).

```
(%i20) A:matrix([65,1,0,0],[37,0,1,0],[13,0,0,1]);
```

(%o20) $$\begin{pmatrix} 65 & 1 & 0 & 0 \\ 37 & 0 & 1 & 0 \\ 13 & 0 & 0 & 1 \end{pmatrix}$$

```
(%i21) for i:1 thru 2 do A[i]:A[i]-quotient(A[i,1],A[i+1,1])*A[i+1];A;
```

(%o21) **done**

(%o22) $$\begin{pmatrix} 28 & 1 & -1 & 0 \\ 11 & 0 & 1 & -2 \\ 13 & 0 & 0 & 1 \end{pmatrix}$$

```
(%i23) A[1]:A[1]-quotient(A[1,1],A[3,1])*A[3];A[3]:A[3]-quotient(A[3,1],A[2,
       1])*A[2];A;
```

(%o23) $[2,1,-1,-2]$

(%o24) $[2,0,-1,3]$

(%o25) $$\begin{pmatrix} 2 & 1 & -1 & -2 \\ 11 & 0 & 1 & -2 \\ 2 & 0 & -1 & 3 \end{pmatrix}$$

```
(%i9) A[2]:A[2]-quotient(A[2,1],A[1,1])*A[1];A[1]:A[1]-quotient(A[1,1],A[3,
      1])*A[3];A;
```

(%o9) $[1,-5,6,8]$

(%o10) $[0,1,0,-5]$

(%o11) $$\begin{pmatrix} 0 & 1 & 0 & -5 \\ 1 & -5 & 6 & 8 \\ 2 & 0 & -1 & 3 \end{pmatrix}$$

```
(%i12) A[3]:A[3]-quotient(A[3,1],A[2,1])*A[2];A;
```

(%o12) $[0,10,-13,-13]$

(%o13) $$\begin{pmatrix} 0 & 1 & 0 & -5 \\ 1 & -5 & 6 & 8 \\ 0 & 10 & -13 & -13 \end{pmatrix}$$

```
(%i14)
```

2 La taille des coefficients de la matrice $\tilde{A}$

Théorème 2.

Soit la matrice $\tilde{A} = (a_{i,j})_{(i,j)\in\{1,\ldots,p\}\times\{2,\ldots,p+1\}}$ obtenue à la fin de l'exécution de l'algorithme décrit au-dessus alors $\forall (i,j)\in\{1,\ldots,p\}\times\{2,\ldots,p+1\}, |a_{i,j}|\leqslant|x_i|$.

Démonstration. *Nous allons adopter les notations suivantes:*

Soit A=(X,I) on désignera ses images par les réductions successives par $A^{(i)}=(X^{(i)},M^{(i)})$ ou, pour préciser la dépendance par rapport aux données de départ, $A^{(i)}(X)=(X^{(i)},M^{(i)}(X))$.

Par ailleurs la procédure de réduction se traduit à chaque itération par $X^{(k+1)} = P_{k+1}X^{(k)}$ *et* $M^{(k+1)} = P_{k+1}M^{(k)}$

Lorsque l'algorithme s'achève $A^{(n)}(X) = (X^{(n)}, M^{(n)}(X))$ *où* $X^{(n)}$ *est un vecteur colonne dont un terme vaut d et les autres sont nuls.*

Nous allons montrer par récurrence sur le nombre de réductions effectuées que $|M^{(n)}(X)| \leqslant (X, X, ..., X)$.

Dans le cas où une réduction suffit X est de la forme $\begin{pmatrix} k_1 d \\ k_2 d \\ ...... \\ k_{p-1} d \\ d \end{pmatrix}$ *et* $A = \begin{pmatrix} k_1 d & 1 & 0 & .... & ... & 0 \\ k_2 d & 0 & 1 & 0 & & \\ ...... & .... & 0 & & & \\ k_{p-1} d & .... & .... & & & \\ d & 0 & .... & ... & 0 & 1 \end{pmatrix}$

et, après une réduction, $A^{(1)}(X) = \begin{pmatrix} 0 & 1 & 0 & .... & ... & -k1 \\ 0 & 0 & 1 & 0 & & -k2 \\ ...... & .... & 0 & & & ... \\ 0 & .... & .... & & & -\mathrm{kp}-1 \\ d & 0 & .... & ... & 0 & 1 \end{pmatrix}$.

Par ailleurs pour tout n>1 $A^{(n)}(X) = A^{(n-1)}(M^{(1)}X)$ *d'où* $M^{(n)}(X) = M^{(n-1)}(M^{(1)}(X))$.

Donc il suffit de montrer que si $|\prod_{k=2}^{n} P_k| \leqslant (X^{(1)}, X^{(1)}, ..., X^{(1)})$ *alors* $|\prod_{k=1}^{n} P_k| \leqslant (X, X, ..., X)$.

De la définition des réductions découle $P_1 = \begin{pmatrix} 1 & -q1 & 0 & ... & 0 \\ 0 & 1 & -q2 & 0 & ... \\ 0 & 0 & 1 & -q3 & ... \\ ... & ... & 0 & ... & 0 \\ 0 & ... & 0 & 1 & -q_{p-1} \end{pmatrix}$ *et nous poserons* $T = \prod_{k=2}^{n} P_k$.

et, en écrivant $T = (t_{i,j})$ *l'hypothèse de récurrence se lit «* $\forall (i,j) |t_{i,j}| \leqslant x'_i = x_i - q_i x_{i+1}$ *et on rappelle que* q_i *est le quotient (euclidien) de* x_i *par* x_{i+1}, *donc* $0 \leqslant x_k - q_k x_{k+1} < x_{i+1} \leqslant x_i$.

i) les éléments de la première colonne de TP1 vérifient

$|t'_{i1}| = |t_{i1}| \leqslant x'_i \leqslant x_{i+1} \leqslant x_i$ *ce qui est suffisant*

ii) soit maintenant $|t'_{ij+1}| = |t_{ij+1} - q_i t_{ij}| \leqslant |t_{ij+1}| + q_i |t_{ij}| \leqslant x'_i (1+q_i) = x_i + q_i x_i - q_i x_{i+1} - q_i q_i x_{i+1} = x_i + q_i (x_i - (1+q_i) x_{i+1}) < x_i$.

Ce qui établit le résultat annoncé. □

[1] Patrick Teller, http://lalgebrisant.fr/images/pdfArticles/BezoutSolutionMinimale.pdf

Paris, mi-janvier 2021