Algorithme du demi-pgcd

Bruno Grenet

Mars 2018

1 Introduction : algorithme d'Euclide

On considère le calcul du pgcd de deux polynômes R_0 et R_1 à coefficients dans un corps[1] $\mathbb{K}$, de degrés respectifs n_0 et $n_1 < n_0$. Le cas de degrés identiques est réglé par une étape de division euclidienne de R_0 par R_1 et peut donc être ignoré.

L'algorithme d'Euclide permet de calculer un[2] pgcd de R_0 et R_1 en $O(\deg(R_0)\deg(R_1))$ opérations dans $\mathbb{K}$.

EUCLIDE(R_0, R_1) :
Entrée : deux polynômes R_0 et R_1 avec $\deg(R_1) < \deg(R_0)$
Sortie : un pgcd de R_0 et R_1

1. $i = 0$
2. Tant que $R_{i+1} \neq 0$:
3. $\quad (Q_i, R_{i+2}) \leftarrow$DIVEUCL$(R_i, R_{i+1})$
4. $\quad i \leftarrow i + 1$
5. Renvoyer R_i

Soit k la valeur finale de i, de telle sorte que $R_{k+1} = 0$ et $R_j \neq 0$ pour tout $j \leq k$.

Définition 1.1. On appelle respectivement *suite des quotients* et *suite des restes* de (R_0, R_1) les suites $(Q_i)_{0\leq i<k}$ et $(R_i)_{0\leq i\leq k+1}$ définies par l'algorithme d'Euclide ci-dessus.

On vérifie aisément que la suite des restes décroit strictement en degré : $\deg(R_i) > \deg(R_{i+1})$ pour tout i. En particulier, pour tout d supérieur ou égal au degré du pgcd de R_0 et R_1, il existe un unique indice j tel que $\deg(R_j) \geq d > \deg(R_{j+1})$. Cette remarque simple est utilisée à de nombreuses reprises dans la suite.

Étant donnée la valeur de Q_i à chaque étape, on peut obtenir la suite des restes grâce au produit matrice-vecteur (les entrées étant des polynômes)

$$\begin{pmatrix} R_{i+1} \\ R_{i+2} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & -Q_i \end{pmatrix} \begin{pmatrix} R_i \\ R_{i+1} \end{pmatrix}.$$

1. On peut définir le pgcd de polynômes à coefficients dans un anneau, mais il faut alors prendre quelques précautions. Le choix de ne travailler qu'avec des polynômes sur un corps est un choix de simplicité.
2. Si G est un pgcd de R_0 et R_1, alors tout multiple de G par une constante non nulle de $\mathbb{K}$ en est également un.

De plus, le pgcd de R_{i+1} et R_{i+2} est égal au pgcd de R_i et R_{i+1}. On note T_i la matrice $\left(\begin{smallmatrix} 0 & 1 \\ 1 & -Q_i \end{smallmatrix}\right)$. On a alors pour tout $i < k$

$$\begin{pmatrix} R_{i+1} \\ R_{i+2} \end{pmatrix} = T_i \cdots T_1 \cdot T_0 \begin{pmatrix} R_0 \\ R_1 \end{pmatrix}.$$

En particulier,

$$\begin{pmatrix} G \\ 0 \end{pmatrix} = T_{k-1} \cdots T_1 \cdot T_0 \begin{pmatrix} R_0 \\ R_1 \end{pmatrix}$$

où G est un pgcd de R_0 et R_1.

L'objectif de l'algorithme dit du « demi-pgcd » est de fournir un algorithme rapide de calcul de la matrice $M = T_{k-1} \cdots T_0$, de laquelle on déduit le pgcd de R_0 et R_1 avec deux multiplications et une addition. On peut également déterminer les coefficients de Bézout sans calcul à partir de M. En effet, l'égalité précédente montrer que $G = M_{00}R_0 + M_{01}R_1$, c'est-à-dire que la première ligne de la matrice M contient les coefficients de Bézout.

On appelle M la *matrice de pgcd de R_0 et R_1*.

2 Algorithme du « demi-pgcd »

Dans l'algorithme d'Euclide, la suite des degrés des restes décroît strictement de $\deg(R_0)$ à $\deg(G)$. L'idée générale de l'algorithme du « demi-pgcd » est de factoriser le produit $M = T_{k-1} \cdots T_0$ en deux sous-produits, de telle sorte que chaque sous-produit fasse baisser le degré des restes de moitié. Plus exactement, on écrit

$$M = \left(T_{k-1} \cdots T_{j+1}\right) \cdot T_j \cdot \left(T_{j-1} \cdots T_0\right)$$

où l'indice j est choisi tel que $\deg(R_j) \geq \frac{1}{2}\deg(R_0) > \deg(R_{j+1})$. L'utilité de laisser une des matrices T_j en dehors de la factorisation apparaîtra claire ultérieurement. On note $D = T_{j-1} \cdots T_0$ et $N = T_{k-1} \cdots T_{j+1}$ de telle sorte que $M = N \cdot T_j \cdot D$. La matrice M sera alors calculée en calculant en premier lieu N, T_j et D puis en effectuant leur produit.

On remarque que par définition, $(T_j D)\left(\begin{smallmatrix} R_0 \\ R_1 \end{smallmatrix}\right) = \left(\begin{smallmatrix} R_{j+1} \\ R_{j+2} \end{smallmatrix}\right)$. Or la correction de l'algorithme d'Euclide repose sur le fait que le pgcd de R_0 et R_1 soit le même que le pgcd de R_{j+1} et R_{j+2}. Ainsi, la matrice N est la matrice de pgcd de R_{j+1} et R_{j+2}. On peut donc la calculer par un appel récursif à l'algorithme de calcul de matrice de pgcd.

Quant à D, ce n'est pas une matrice de pgcd. Appliquée à R_0 et R_1, elle fournit un couple de restes (R_j, R_{j+1}) tels que $\deg(R_j) \geq \frac{1}{2}\deg(R_0) > \deg(R_{j+1})$. On appelle cette matrice la *matrice de demi-pgcd de R_0 et R_1*. i

L'algorithme du « demi-pgcd » est constitué de deux algorithmes. L'algorithme MatricePgcd permet de calculer la matrice de pgcd de deux polyômes, étant donnée leur matrice de demi-pgcd. L'algorithme DemiPgcd calcule lui la matrice de demi-pgcd.

2.1 L'algorithme MatricePgcd

On commence par décrire l'algorithme MatricePgcd, plus simple, qui fait appel à l'algorithme DemiPgcd qui sera décrit ensuite.

MatricePgcd(R_0, R_1) :
Entrée : deux polynômes tels que $\deg(R_0) > \deg(R_1)$.
Sortie : la matrice de pgcd de R_0 *et* R_1.

1. $D \leftarrow$DemiPgcd(R_0, R_1)
2. $\begin{pmatrix} R_j \\ R_{j+1} \end{pmatrix} \leftarrow D \begin{pmatrix} R_0 \\ R_1 \end{pmatrix}$
3. Si $R_{j+1} = 0$, renvoyer D
4. $(Q_j, R_{j+2}) \leftarrow$DivEucl$(R_j, R_{j+1})$
5. $T_j \leftarrow \begin{pmatrix} 0 & 1 \\ 1 & -Q_j \end{pmatrix}$
6. Si $R_{j+2} = 0$, renvoyer $T_j \cdot D$
7. $N \leftarrow$MatricePgcd(R_{j+1}, R_{j+2})
8. Renvoyer $N \cdot T_j \cdot D$

Théorème 2.1. *Soit* $n = \deg(R_0)$, *et* $\mathsf{H}(n)$ *la complexité de l'algorithme* DemiPgcd. *On suppose que l'algorithme* DemiPgcd *est correct, et que sa complexité* $\mathsf{H}(n)$ *est telle que* $\mathsf{H}(n)/n$ *est croissante et* $\mathsf{H}(n) \geq \mathsf{M}(n)$ *où* $\mathsf{M}(n)$ *est la complexité du produit de deux polynômes de taille* n.
Alors l'algorithme MatricePgcd *est correct, et son temps de calcul* $G(n) = O(\mathsf{H}(n))$.

Démonstration. La correction de l'algorithme MatricePgcd est claire d'après celle de DemiPgcd, les étapes 3. et 6. jouant les rôles de *cas de base*.

La complexité de l'étape 1. est $\mathsf{H}(n)$ par définition. On sait que $\deg(R_j) \geq n/2 > \deg(R_{j+1})$, donc l'appel récursif à l'étape 7. est effectué sur deux polynômes de degré $< n/2$. Le coût de cette étape est borné par $G(n/2)$. Les autres étapes, division euclidienne comprise, coûtent un nombre constant de multiplications de polynômes de degré au plus n. Ainsi,

$$G(n) \leq G(n/2) + \mathsf{H}(n) + c\mathsf{M}(n)$$

pour une certaine constante c. Puisque $\mathsf{H}(n) \geq \mathsf{M}(n)$ par hypothèse, on peut réécrire $G(n) \leq G(n/2) + c'\mathsf{H}(n)$ où $c' = c + 1$. On en déduit que pour $h < \log(n)$,

$$G(n) \leq G(n/2^h) + c' \sum_{j=0}^{h-1} \mathsf{H}(n/2^j).$$

Puisque $\mathsf{H}(n) \geq \mathsf{M}(n) \geq n$, on peut écrire $\mathsf{H}(n) = n\alpha(n)$ pour une certaine fonction $\alpha(n) \geq 1$. Alors $\sum_j \mathsf{H}(n/2^j) = \sum_j (n/2^j)\alpha(n/2^j) \leq n\alpha(n) \sum_j 1/2^j \leq 2\mathsf{H}(n)$, en bornant $\alpha(n/2^j)$ par $\alpha(n)$. En prenant $h = \lceil \log(n) \rceil$, on obtient $G(n) = O(\mathsf{H}(n))$.

□

Remarque. Dans l'algorithme MatricePgcd, on peut obtenir sans calcul supplémentaire le pgcd lui-même. En effet, à l'étape 3. le pgcd est R_j et à l'étape 6. c'est R_{j+1}. Enfin, si on suppose que l'étape 7. renvoie récursivement non seulement la matrice N mais aussi le pgcd G de R_{j+1} et R_{j+2}, il suffit de renvoyer G à l'étape 8.

Si l'objectif final est d'obtenir le pgcd, le calcul de la matrice de pgcd est inutile. Si aux étapes 3., 6. et 8. on renvoie le pgcd au lieu de la matrice de pgcd (et que l'appel récursif calcule donc le pgcd que R_{j+1} et R_{j+2}), il est clair que l'on obtient encore un algorithme correct, de même complexité

asymptotique. Cette variante permet d'économiser quelques calculs (produits de matrices à l'étape 8. et calcul final du pgcd à partir de la matrice de pgcd).

Cependant, on note que la matrice de pgcd permet également de déduire (sans calcul) les coefficients de Bézout associés aux polynômes d'entrée, ce que ne permet pas la variante évoquée ci-dessus.

2.2 Calcul du demi-pgcd

On s'intéresse maintenant au calcul de la matrice D du demi-pgcd de R_0 et R_1. Comme pour l'algorithme général, on factorise la matrice D en deux sous-produits. On écrit

$$D = \left(T_{j-1} \cdots T_{i+1}\right) \cdot T_i \cdot \left(T_{i-1} \cdots T_0\right)$$

où l'indice i est choisi cette fois-ci tel que $\deg(R_i) \geq \frac{3}{4}\deg(R_0) > \deg(R_{i+1})$.

Pour exploiter pleinement l'idée du « diviser pour régner », deux problèmes se posent : d'une part, les deux matrices de la factorisation ne sont pas des matrices de demi-pgcd, et d'autre part les polynômes auxquels on les applique sont de taille de l'ordre de n. On aimerait les calculer comme matrices de demi-pgcd de polynômes de degrés environ $n/2$. Cela est rendu possible par la remarque suivante : si A et B sont deux polynômes de degrés éventuellement grands, mais proches, leur quotient ne dépend que des coefficients de poids fort de A et B.

Par exemple, si $A = 3X^{100} + 4X^{99} - 10X^{98} \cdots$ et $B = X^{99} - 2X^{98} + \cdots$, leur quotient $3X + 10$ de degré 1 et ne dépend que des deux coefficients dominants de A et B. Ainsi, ce quotient est égal au quotient dans la division de $A^\star = 3X^2 + 4X - 10$ par $B^\star = X - 2$. On peut aussi remarquer qu'en appliquant l'algorithme d'Euclide au couple (A, B) ou au couple $(A^\star, B^\star)$, le premier reste dans les deux cas a le même coefficient dominant.

Le lemme suivant quantifie ces remarques.

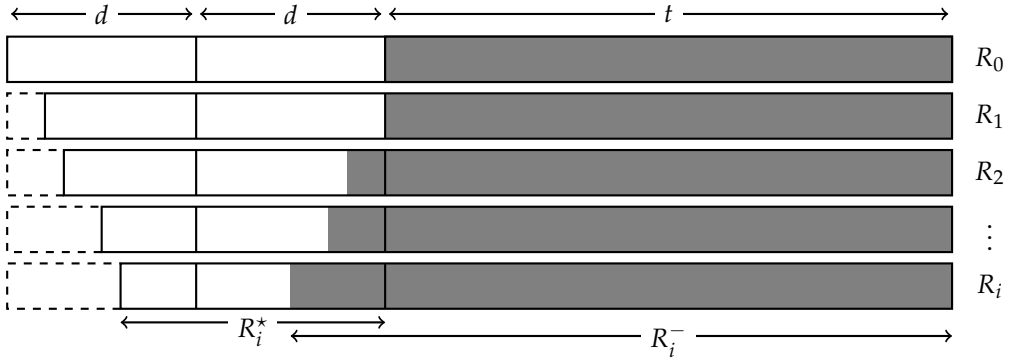

FIGURE 1 – Illustration du lemme 2.2.

Lemme 2.2. *Soit R_0 de degré $n_0 \leq t + 2d$ $(t, d > 0)$ et R_1 de degré n_1 avec $n_0 > n_1 \geq t + d$, et $R_0^\star = R_0 \operatorname{quo} X^t$ et $R_1^\star = R_1 \operatorname{quo} X^t$. Soit $(R_i)_i$ et $(R_i^\star)_i$ les suites des restes respectives de (R_0, R_1) et $(R_0^\star, R_1^\star)$, et j tel que $\deg(R_j^\star) \geq d > \deg(R_{j+1}^\star)$.*

Alors pour $0 \leq i \leq j+1$,

$$R_i = R_i^\star X^t + R_i^- \text{ avec } \begin{cases} \deg(R_i^\star) = n_i - t & \text{et} \\ \deg(R_i^-) < t + n_0 - n_{i-1} \end{cases}$$

où $n_i = \deg(R_i)$ *et* $n_{-1} = n_0$.

De plus les suites des quotients de (R_0, R_1) *et* $(R_0^\star, R_1^\star)$ *sont identiques jusqu'à l'ordre* $j-1$.

Démonstration. On montre le résultat par récurrence. Pour $i = 0$ et 1, on a bien $R_i = R_i^\star X^t + R_i^-$ avec $\deg(R_i^-) < t + n_0 - n_0 = t$ par définition de $R_i^\star$. On suppose le résultat correct jusqu'au rang $i+1$ (pour $i \geq 0$) et on le montre au rang $i+2$. On peut écrire

$$\begin{aligned} R_i &= R_i^\star X^t + R_i^- && \text{(par hypothèse de récurrence au rang } i\text{)} \\ &= R_{i+1}^\star Q_i^\star X^t + R_{i+2}^\star X^t + R_i^- && \text{(par définition de } R_{i+2}^\star\text{)} \\ &= R_{i+1} Q_i^\star - R_{i+1}^- Q_i^\star + R_{i+2}^\star X^t + R_i^- && \text{(par hypothèse de récurrence au rang } i+1\text{).} \end{aligned}$$

On pose $R_{i+2}^- = R_i^- - R_{i+1}^- Q_i^\star$ de telle sorte que $R_i = R_{i+1} Q_i^\star + R_{i+2}^\star X^t + R_{i+2}^-$. Alors puisque $\deg(R_{i+1}^-) < t + n_0 - n_i$ et $\deg(Q_i^\star) = \deg(R_i^\star) - \deg(R_{i+1}^\star) = n_i - n_{i+1}$, $\deg(R_{i+1}^- Q_i^\star) < t + n_0 - n_{i+1}$. De même, $\deg(R_i^-) < t + n_0 - n_{i-1} < t + n_0 - n_{i+1}$. Or $t + n_0 - n_{i+1} \leq t + d \leq n_{i+1}$ tant que $n_{i+1} \geq t + d$ car $n_0 \leq t + 2d$. Donc $\deg(R_{i+2}^-) < t + n_0 - n_{i+1} \leq n_{i+1}$. D'autre part, $\deg(R_{i+2}^\star) < \deg(R_{i+1}^\star)$ par propriété de la division euclidienne donc $\deg(R_{i+2}^\star X^t) < t + (n_{i+1} - t) = n_{i+1}$.

Ainsi, l'égalité $R_i = R_{i+1} Q_i^\star + (R_{i+2}^\star X^t + R_{i+2}^-)$ est la division euclidienne de R_i par R_{i+1} puisque le degré du reste est strictement inférieur au degré de R_{i+1}. Par unicité du quotient et du reste, $Q_i = Q_i^\star$, $R_{i+2} = R_{i+2}^\star X^t + R_{i+2}^-$ et $\deg(R_{i+2}^-) < t + n_0 - n_{i+1}$. Et tant que $n_{i+1} \geq t + d$, $\deg(R_{i+2}^-) < n_{i+1}$: autrement dit, $\deg(R_{i+2}) = \deg(R_{i+2}^\star X^t)$, d'où $\deg(R_{i+2}^\star) = n_{i+2} - t$.

La démonstration effectuée est valable tant que $n_{i+1} \geq t + d$, c'est-à-dire tant que $i + 1 \leq j$. Ce qui signifie que le résultat est démontré jusqu'à R_{j+1} et Q_{j-1}.

□

On peut déduire le corollaire suivant du lemme 2.2.

Corollaire 2.3. *Soit* A *et* B *deux polynômes et* t *et* d *deux entiers tels que* $\lceil (\deg(A) - t)/2 \rceil = d$ *et* $\deg(A) > \deg(B) \geq t$. *Soit* $A^\star = A \operatorname{quo} X^t$ *et* $B^\star = B \operatorname{quo} X^t$, *et* $D^\star$ *la matrice de demi-pgcd de* $A^\star$ *et* $B^\star$.

Alors $D^\star \begin{pmatrix} A \\ B \end{pmatrix} = \begin{pmatrix} R_j \\ R_{j+1} \end{pmatrix}$ *où* $(R_i)_i$ *est la suite des restes de* (A, B) *et* $\deg(R_j) \geq t + d > \deg(R_{j+1})$.

Démonstration. On remarque que la condition sur le degré de A signifie que $\deg(A)$ est égal à $t + 2d$ ou $t + 2d - 1$. Le résultat est trivial lorsque $\deg(B) < t + d$, car dans ce cas $\deg(B^\star) < \deg(A^\star)/2$ et $D^\star$ est la matrice identité.

On suppose donc $\deg(B) \geq t + d$ et on note $(Q_i)_i$ et $(Q_i^\star)_i$ les suites des quotients de (A, B) et $(A^\star, B^\star)$, respectivement. Le lemme 2.2 assure que $Q_i = Q_i^\star$ pour $i \leq j - 1$, où j est tel que $\deg(R_j^\star) \geq d > \deg(R_{j+1}^\star)$. De plus, cet indice vérifie $\deg(R_j) \geq t + d > \deg(R_{j+1})$ d'après le même lemme. Ainsi, la matrice $D^\star$ est le produit $T_{j-1} \dots T_0$ où $T_i = \begin{pmatrix} 0 & 1 \\ 1 & -Q_i \end{pmatrix}$. Donc $D^\star \begin{pmatrix} A \\ B \end{pmatrix} = \begin{pmatrix} R_j \\ R_{j+1} \end{pmatrix}$.

□

2.3 L'algorithme DemiPgcd

On peut maintenant décrire un algorithme récursif de calcul de la matrice de demi-pgcd de R_0 et R_1.

DemiPgcd(R_0, R_1) :
Entrée : deux polynômes tels que $\deg(R_0) > \deg(R_1)$.
Sortie : la matrice de demi-pgcd de R_0 *et* R_1.

1. $m \leftarrow \lceil n/2 \rceil$, où $n = \deg(R_0)$
2. Si $\deg(R_1) < m$:
3. Renvoyer I_2 (matrice identité)
4. $(R_0^\star, R_1^\star) \leftarrow (R_0 \text{ quo } X^m, R_1 \text{ quo } X^m)$
5. $D_0^\star \leftarrow$ DemiPgcd$(R_0^\star, R_1^\star)$
6. $\begin{pmatrix} R_j \\ R_{j+1} \end{pmatrix} \leftarrow D_0^\star \begin{pmatrix} R_0 \\ R_1 \end{pmatrix}$
7. Si $\deg(R_{j+1}) < m$:
8. Renvoyer $D_0^\star$
9. $(Q_j, R_{j+2}) \leftarrow$DivEucl$(R_j, R_{j+1})$
   $T_j \leftarrow \begin{pmatrix} 0 & 1 \\ 1 & -Q_j \end{pmatrix}$
10. Si $\deg(R_{j+2}) < m$:
11. Renvoyer $T_j D_0^\star$
12. $\ell \leftarrow 2m - \deg(R_{j+1})$
13. $(R_{j+1}^\star, R_{j+2}^\star) \leftarrow (R_{j+1} \text{ quo } X^\ell, R_{j+2} \text{ quo } X^\ell)$
14. $D_1^\star \leftarrow$ DemiPgcd$(R_{j+1}^\star, R_{j+2}^\star)$
15. Renvoyer $D_1^\star T_j D_0^\star$

Théorème 2.4. *L'algorithme* DemiPgcd *est correct. Sa complexité est* $\mathsf{H}(n) = O(\mathsf{M}(n)\log(n))$.

Démonstration. On suppose dans un premier temps qu'on se trouve dans un cas où l'algorithme termine à la dernière étape : en particulier $\deg(R_1) \geq m$. Soit $D = D_1^\star T_j D_0^\star$ la matrice renvoyée à l'étape 12., et R_k et R_{k+1} les polynômes définis par $D \begin{pmatrix} R_0 \\ R_1 \end{pmatrix} = \begin{pmatrix} R_k \\ R_{k+1} \end{pmatrix}$. Afin de montrer la correction de l'algorithme, on démontre que R_k et R_{k+1} appartiennent bien à la suite des restes de R_0 et R_1, et que $\deg(R_k) \geq m > \deg(R_{k+1})$.

On applique le corollaire 2.3 avec $A = R_0$, $B = R_1$, $t = m$ et $d = \lceil 3n/4 \rceil - m$. Ces valeurs vérifient bien $\lceil (\deg(A) - t)/2 \rceil = d$. Par hypothèse, $\deg(R_0) > \deg(R_1) \geq m$ donc les polynômes R_j et R_{j+1} calculés à l'étape 5. appartiennent à la suite des restes de (A, B) et $\deg(R_j) \geq t + d = \lceil 3n/4 \rceil > \deg(R_{j+1})$. Donc après une étape de division euclidienne (étape 7.), $\deg(R_{j+2}) \leq 3n/4$.

Par définition de ℓ, $\deg(R_{j+1}) = \ell + 2(m - \ell)$. On peut à nouveau appliquer le corollaire 2.3, avec cette fois $A = R_{j+1}$, $B = R_{j+2}$, $t = \ell$ et $d = m - \ell$. Les polynômes obtenus en appliquant $D_1^\star$ à $\begin{pmatrix} R_{j+1} \\ R_{j+2} \end{pmatrix}$ sont par définition R_k et R_{k+1} car $D = D_1^\star T_j D_0^\star$ et $\begin{pmatrix} R_{j+1} \\ R_{j+2} \end{pmatrix} = T_j D_0 \begin{pmatrix} R_0 \\ R_1 \end{pmatrix}$. Ainsi, R_k et R_{k+1} sont deux restes successifs de la suite des restes de R_j et R_{j+1} et $\deg(R_k) \geq \ell + (m - \ell) = m > \deg(R_{k+1})$.

Cela suffit à démontrer que D est bien la matrice de demi-pgcd de R_0 et R_1. Finalement, on se convainc aisément que les arguments présentés restent valables dans les cas où l'algorithme répond avant la dernière étape.

Pour la complexité, l'algorithme sur une entrée R_0 de degré n fait deux appels récursifs sur des polynômes de degrés $< n/2$, plus un nombre constant de multiplications en taille n. Donc il existe une constante c telle que

$$\mathsf{H}(n) \leq 2\mathsf{H}(n/2) + c\mathsf{M}(n).$$

Plus généralement, on obtient

$$\mathsf{H}(n) \leq 2^h \mathsf{H}(n/2^h) + c \sum_{i=0}^{h-1} 2^i \mathsf{M}(n/2^i).$$

En utilisant la même technique de preuve que pour l'algorithme MatricePgcd, on obtient $\mathsf{H}(n) = O(\mathsf{M}(n)\log(n))$.

□