

# Résolution d'un système linéaire par l'algorithme de Wiedemann

Soit  $K$  un corps commutatif et  $n \in \mathbb{N}$ ,  $n \geq 1$ . Nous considérons le système linéaire

$$Ax = b \tag{1}$$

où  $A$  est une matrice carrée de taille  $n$  à coefficients dans  $K$  et  $b$  un élément de  $K^n$ . De nombreux algorithmes reposent sur la résolution d'un tel système dans le cas où  $K$  est un corps fini et  $A$  une matrice *creuse* c'est-à-dire avec peu de coefficients non nuls par ligne. C'est le cas d'algorithmes de calcul de logarithme discret, de factorisation d'entiers (crible quadratique ou algébrique). L'algorithme de Wiedemann, particulièrement avantageux dans le cas des matrices creuses, permet de résoudre le système (1).

## 1 La suite de Krylov associée

Lorsque  $A$  est une matrice creuse, on souhaite tirer profit de cette qualité de sorte que l'opération à privilégier est le produit de  $A$  avec un vecteur : on considère  $A$  comme une "boîte noire" qui envoie un vecteur  $b$  sur le vecteur  $Ab$ . On considère la suite suivante appelée *suite de Krylov* associée à  $b$  :

$$b, Ab, A^2b, \dots, A^i b, \dots \quad i \in \mathbb{N} \tag{2}$$

Puisque  $n + 1$  vecteurs de cette liste sont liés, il existe une relation de dépendance linéaire

$$p_0 b + p_1 Ab + \dots + p_n A^n b = 0. \tag{3}$$

Remarquons que la suite de Krylov est une suite récurrente linéaire puisqu'elle vérifie dès lors

$$p_0 A^i b + p_1 A^{i+1} b + \dots + p_n A^{i+n} b = 0 \tag{4}$$

pour tout  $i \in \mathbb{N}$ . On dit que le polynôme  $p = p_0 + p_1 X + \dots + p_n X^n$  en est un *générateur*.

Si l'on peut trouver un polynôme générateur  $p$  de la suite de Krylov avec un coefficient constant non nul (c'est le cas si  $A$  est inversible), alors on pose

$$p^- = (p - p(0))/(Xp(0))$$

et une solution du système linéaire (1) est donnée par

$$x = -p^-(A)b. \tag{5}$$

## 2 Calcul du générateur minimal d'une suite récurrente linéaire.

Soit  $(s_i)_{i \in \mathbb{N}}$  une suite d'éléments de  $K$ . On suppose qu'elle vérifie une relation de récurrence linéaire c'est-à-dire qu'il existe un polynôme  $P = p_0 + p_1X + \dots + p_lX^l$  de degré  $l \geq 1$ , tel que pour tout  $i \in \mathbb{N}$ ,

$$p_0s_i + p_1s_{i+1} + \dots + p_ls_{i+l} = 0.$$

Un tel polynôme est dit *générateur* pour la suite. On vérifie sans difficulté que l'ensemble des polynômes générateurs de la suite  $(s_i)_{i \in \mathbb{N}}$  forme un idéal de  $K[X]$  et l'on en note  $\Pi$  le générateur unitaire. On l'appelle le *générateur minimal* et son degré  $d$  est le *degré* de la suite.

On suppose désormais que  $d \leq n$ . Notant  $H_k$  le vecteur colonne de coordonnées  $(s_k, \dots, s_{k+n})$  et  $H$  la matrice de colonnes  $H_0, \dots, H_n$ , on a la proposition suivante :

**Proposition 1** 1. Un polynôme  $P = p_0 + p_1X + \dots + p_nX^n$  génère la suite  $(s_i)_{i \in \mathbb{N}}$  si et seulement si le vecteur colonne de coordonnées  $(p_0, \dots, p_n)$  appartient au noyau de la matrice  $H$ .

2. Le rang de la matrice  $H$  est égal au degré de la suite  $(s_i)_{i \in \mathbb{N}}$ .

A la lumière de cette proposition, il apparaît que la connaissance des  $2n$  premiers termes de la suite suffit pour en déterminer le générateur minimal. En effet, on désigne par  $S$  le polynôme

$$S = s_{2n-1} + s_{2n-2}X + \dots + s_1X^{2n-2} + s_0X^{2n-1}.$$

**Proposition 2** Soit  $P$  un polynôme de degré inférieur ou égal à  $n$ . Il génère la suite  $(s_i)_{i \in \mathbb{N}}$  si et seulement si il existe  $R \in K[X]$  de degré  $< n$  tel que

$$PS = R \pmod{X^{2n}}. \quad (6)$$

Il s'agit donc de trouver le polynôme de degré minimal vérifiant la condition (6). L'algorithme (BM) suivant, justifié par les propriétés de l'algorithme d'Euclide étendu, donne le générateur minimal de la suite  $(s_i)_{i \in \mathbb{N}}$ . C'est l'algorithme de Berlekamp-Massey.

$$R_0 := X^{2n}, R_1 := S, V_0 := 0, V_1 := 1.$$

Tant que  $n \leq \deg(R_1)$  faire

$(Q, R) :=$  quotient et reste de la division euclidienne de  $R_0$  par  $R_1$  ;

$V := V_0 - QV_1$  ;

$V_0 := V_1, V_1 := V, R_0 := R_1, R_1 := R$  ;

Le polynôme  $\Pi$  est alors égal à  $V_1$  divisé par son coefficient dominant.

## 3 L'algorithme de Wiedemann

On suppose que  $A$  est inversible. D'après la première partie, calculer le générateur minimal  $\pi$  de la suite de Krylov associée à  $b$  permet de résoudre le système (1). En effet, son terme constant ne saurait être nul.

Pour pouvoir utiliser l'algorithme de Berlekamp-Massey, l'idée de Wiedemann consiste à considérer, plutôt que la suite des vecteurs, des suites de scalaires du type

$$(\langle u, A^i b \rangle)_{i \in \mathbb{N}} \quad (7)$$

où le vecteur  $u$  est choisi aléatoirement. Le polynôme  $\pi$  est générateur pour la suite (7).

Soit  $u \in K^n$ . L'algorithme de Berlekamp-Massey permet de calculer le générateur minimal de la suite  $(\langle u, A^i b \rangle)_{i \in \mathbb{N}}$  qui est un diviseur du polynôme  $\pi$ . En répétant cette procédure pour plusieurs vecteurs  $u$  choisis aléatoirement on trouve ainsi des facteurs de  $\pi$ . La question est de savoir si on finit par trouver  $\pi$  après un nombre fini, et assez petit, d'itérations de cette procédure. L'idée pour trouver rapidement  $\pi$  consiste à s'assurer que l'on balaie bien l'ensemble ses facteurs en procédant comme suit.

On pose  $b_0 = b$ . On choisit aléatoirement un vecteur  $u_1$  et l'on calcule le générateur minimal  $\pi_1$  de degré  $d_1$  de la suite  $(\langle u_1, A^i b_0 \rangle)_{i \in \mathbb{N}}$  associée. On pose  $b_1 = \pi_1(A)b_0$ . Si  $b_1 = 0$  alors c'est terminé, on a trouvé  $\pi$  et une solution au système linéaire. Sinon, la suite de Krylov associée à  $b_1$  a pour générateur minimal le quotient  $\pi/\pi_1$  de degré  $\leq n - d_1$ . On choisit alors  $u_2 \in K^n$ . A l'aide des  $2(n - d_1)$  premiers éléments de la suite  $(\langle u_2, A^i b_1 \rangle)_{i \in \mathbb{N}}$ , l'algorithme de Berlekamp-Massey fournit son générateur minimal  $\pi_2$  qui est un facteur de  $\pi/\pi_1$ . On pose  $b_2 = \pi_2(A)b_1 = (\pi_1\pi_2)(A)b$ . Et l'on réitère la discussion.

Après un "petit" nombre d'itérations, on a trouvé des polynômes  $\pi_1, \dots, \pi_r \in K[X]$  tels que  $(\pi_1 \dots \pi_r)(A)b = 0$ . Une solution au système linéaire (1) est  $x = -(\pi_1 \dots \pi_r)^-(A)b$ .

La démarche précédente peut être traduite par l'algorithme (W) suivant, au sein duquel la relation  $y_k = (\pi_1 \dots \pi_k)^-(A)b$  est conservée pour tout  $k \geq 1$  :

1.  $k := 0$ ,  $d_0 := 0$ ,  $y_0 := 0$ ,  $b_0 := b$ .
2. Si  $b_k = 0$  alors  $x := -y_k$  et c'est terminé.
3. Choisir  $u_{k+1}$  un vecteur dans  $K^n$ .
4. Calculer les  $2(n - d_k)$  premiers termes de la suite  $(\langle u_{k+1}, A^i b_k \rangle)_{i \in \mathbb{N}}$ .
5. Déterminer le polynôme minimal  $\pi_{k+1}$  de cette suite.
6.  $y_{k+1} := y_k + \pi_{k+1}^-(A)b_k$ ,  $b_{k+1} := b_0 + Ay_{k+1}$ ,  $d_{k+1} := d_k + \deg(\pi_{k+1})$ .
7.  $k := k + 1$ , et reprendre l'étape 2.

Il subsiste une incertitude sur le nombre d'itérations nécessaires pour trouver une solution au système linéaire. Supposons que  $K$  est le corps fini  $\mathbb{F}_q$ . Dans *Solving sparse linear equations over finite fields* (IEEE Trans. Inf. Theory 32, 1986) Wiedemann montre que la probabilité  $\Phi(k)$  pour que  $\pi$  soit trouvé au bout de  $k$  itérations ( $k \geq 1$ ) vérifie

$$\Phi(k) > 1 - \log\left(\frac{q^{k-1}}{q^{k-1} - 1}\right).$$

Si  $q$  est grand  $\Phi(2)$  est proche de 1. Si  $q = 2$ , on obtient des bornes inférieures assez petites :  $\Phi(2) > 0.307$ ,  $\Phi(3) > 0.712, \dots, \Phi(6) > 0.968$ . (Mais ces bornes sont valables pour des matrices aléatoirement choisies, et pas nécessairement creuses...)

## 4 Quelques pistes

1. Vérifier quelques assertions contenues dans le textes (propositions 1 et 2, véracité de l'algorithme (W)...)
2. Questions de complexité :  
Quel(s) algorithme(s) utilise-t-on en général pour inverser un système linéaire relatif à une matrice quelconque ?  
Combien de multiplications matrice  $\times$  vecteur l'algorithme de Wiedemann nécessite-t-il ? Pour une matrice creuse ayant  $l$  coefficients non nuls, quelle est la complexité d'une telle multiplication ? L'algorithme de Wiedemann est-il avantageux pour une matrice non creuse ?
3. Tester l'algorithme de Wiedemann sur différents corps finis et illustrer sa vitesse de convergence.
4. Que donne l'algorithme de Wiedemann lorsque la matrice  $A$  n'est pas inversible ? Comment peut-on trouver un élément du noyau de toute matrice de taille  $n \times n + 1$  ? Et s'il y a plus que  $n + 1$  colonnes ?
5. L'algorithme de Wiedemann peut-il permettre de calculer le polynôme minimal de la matrice  $A$  ? Son polynôme caractéristique ?