

Le pivot de Gauss, un algorithme qui sert à tout.

L'algorithme du pivot de Gauss est fondamental en classes préparatoires (et même dans les mathématiques en général), car il permet de résoudre efficacement les systèmes linéaires, de trouver des bases ou équations de sous-espaces et, plus tard, de calculer des déterminants. En dehors de son aspect pratique, son analyse théorique fournit des théorèmes intéressants, comme le théorème du rang ou la décomposition LU (très utile, mais pas officiellement au programme).

Comme dans une matrice, on peut voir les colonnes comme des vecteurs et les lignes comme des équations, on a en quelque sorte deux versions du pivot de Gauss : le pivot en lignes et le pivot en colonnes, en fonction de ce que l'on cherche à obtenir.

Trouver une base, c'est échelonner en colonnes

On considère la matrice 3×4 suivante : $A = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 5 & 7 \\ 1 & 2 & 1 & 2 \end{pmatrix}$.

On cherche une base de son image en effectuant un pivot de Gauss sur les *colonnes*. On ajoute une ligne supplémentaire contenant les variables c_1, c_2, c_3, c_4 : elle permet d'enregistrer, à chaque étape, les combinaisons de colonnes effectuées. On parle de *matrice augmentée*.

$$\begin{array}{c} \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 5 & 7 \\ 1 & 2 & 1 & 2 \\ c_1 & c_2 & c_3 & c_4 \end{pmatrix} \\ \begin{array}{l} C_2 \leftarrow C_2 - 2C_1 \\ C_3 \leftarrow C_3 - 3C_1 \\ C_4 \leftarrow C_4 - 4C_1 \end{array} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 2 & 0 & -1 & -1 \\ 1 & 0 & -2 & -2 \\ c_1 & c_2 - 2c_1 & c_3 - 3c_1 & c_4 - 4c_1 \end{pmatrix} \\ \begin{array}{l} C_2 \leftrightarrow C_3 \end{array} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 2 & -1 & 0 & -1 \\ 1 & -2 & 0 & -2 \\ c_1 & c_3 - 3c_1 & c_2 - 2c_1 & c_4 - 4c_1 \end{pmatrix} \end{array}$$

$$\begin{array}{c} C_2 \leftarrow -C_2 \\ \begin{array}{l} C_1 \leftarrow \tilde{C}_1 - 2C_2 \\ C_4 \leftarrow \tilde{C}_4 + C_2 \end{array} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & -1 \\ 1 & 2 & 0 & -2 \\ c_1 & 3c_1 - c_3 & c_2 - 2c_1 & c_4 - 4c_1 \end{pmatrix} \\ \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -3 & 2 & 0 & 0 \\ -5c_1 + 2c_3 & 3c_1 - c_3 & c_2 - 2c_1 & c_4 - c_3 - c_1 \end{pmatrix} \end{array}$$

Par définition, l'image d'une matrice est l'espace engendré par ses colonnes, qui en forment donc une famille génératrice. Il reste à transformer cette famille génératrice en base en éliminant les vecteurs linéairement dépendants : c'est exactement ce que fait le pivot de Gauss. Les colonnes finales non nulles forment dans tous les cas une base de l'image, ici :

$$\left(\begin{pmatrix} 1 \\ 0 \\ -3 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix} \right).$$

De plus, cette base est relativement simple à manipuler : l'algorithme du pivot de Gauss fait en sorte de mettre le plus de zéros possible dans les colonnes, ce qui facilite les calculs ultérieurs (par exemple la décomposition d'un vecteur de l'image sur cette base).

Cette opération permet aussi de trouver une base du noyau de A . En effet, pour $C = (c_1 \ c_2 \ c_3 \ c_4)^T \in \mathbb{R}^4$, on a

$$C \in \ker A \iff c_1 C_1 + c_2 C_2 + c_3 C_3 + c_4 C_4 = 0.$$

Chaque colonne nulle obtenue pendant le pivot fournit précisément une telle relation. Ici, les troisième et quatrième colonnes donnent

$$C_2 - 2C_1 = 0 \quad \text{et} \quad -C_1 - C_3 + C_4 = 0.$$

Une base de $\ker A$ est donc $\left(\begin{pmatrix} -2 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} -1 \\ 0 \\ -1 \\ 1 \end{pmatrix} \right)$.

Plus généralement, si on a une famille de p vecteurs de $\mathbb{K}^n$, rassemblés dans une matrice $A \in \mathcal{M}_{n,p}(\mathbb{K})$, on peut appliquer le même raisonnement pour trouver une base de l'image et du noyau de A .

Méthode : donner une base de l'image. On échelonne en colonnes. Les colonnes non nulles à la fin de l'algorithme forment une base de l'image.

Méthode : donner une base du noyau. Supposons que A possède p colonnes. On ajoute une ligne de variables $c_1, \dots, c_p$, puis on échelonne A en colonnes. Pour chaque colonne nulle finale d'indice j , la dernière ligne contient une expression de la forme

$$\alpha_{1j}c_1 + \dots + \alpha_{pj}c_p \text{ qu'on associe alors au vecteur } \begin{pmatrix} \alpha_{1j} \\ \vdots \\ \alpha_{pj} \end{pmatrix}.$$

La famille de tous les vecteurs ainsi obtenus, un pour chaque colonne nulle, est une base de $\ker A$.

Astuce Si vous avez déjà échelonné en colonnes la matrice sans l'avoir augmentée, vous pouvez tout de même obtenir les combinaisons linéaires égales à 0 des colonnes en "remontant" les opérations effectuées, c'est-à-dire en exprimant chaque colonne nulle finale comme combinaison linéaire des colonnes initiales. Par exemple :

$$\begin{aligned} 0 &\approx C_4 + C_2 && \text{à la dernière équivalence.} \\ &\approx C_4 - C_2 && \text{car on avait fait } C_2 \leftarrow -C_2. \\ &\approx C_4 - C_3 && \text{car on avait fait } C_2 \leftrightarrow C_3. \\ &\approx (C_4 - 4C_1) - (C_3 - 3C_1) && \text{car } C_3 \leftarrow C_3 - 3C_1 \text{ et } C_4 \leftarrow C_4 - 4C_1. \\ &= -C_1 - C_3 + C_4 \end{aligned}$$

On est revenu aux colonnes initiales, et on retrouve bien le vecteur $(-1, 0, -1, 1)^\top$.

Attention : Ce ne sont pas de véritables égalités à chaque fois, car les colonnes changent à chaque étape de l'algorithme. Comme on évite en mathématiques les symboles tels que $\approx$, gardez plutôt cette manière de faire pour votre brouillon ou pour un coin de tableau à l'oral.

Trouver des équations, c'est échelonner en lignes

On cherche maintenant des équations de $\ker A \subset \mathbb{R}^4$ et de $\text{Im } A \subset \mathbb{R}^3$. Pour le noyau, c'est facile : les lignes de A donnent directement les coefficients des équations. Cependant, on peut vouloir obtenir un ensemble minimal d'équations, avec aussi peu d'équations et aussi peu de variables que possible : effectuer un pivot sur les lignes permet de simplifier le système.

Comme précédemment, la conservation de la trace des opérations sur les colonnes donnait gratuitement une base du noyau ; ici, la conservation de la trace des opérations sur les lignes permet d'obtenir des équations de l'image. Si $L = (l_1 \ l_2 \ l_3)^\top$, on applique donc le pivot de Gauss *en lignes* à la matrice augmentée $(A \mid L)$. Les mêmes opérations doivent être effectuées sur la colonne L :

$$\begin{aligned} \left(\begin{array}{cccc|c} 1 & 2 & 3 & 4 & l_1 \\ 2 & 4 & 5 & 7 & l_2 \\ 1 & 2 & 1 & 2 & l_3 \end{array} \right) &\xrightarrow[L_3 \leftarrow L_3 - L_1]{L_2 \leftarrow L_2 - 2L_1} \left(\begin{array}{cccc|c} 1 & 2 & 3 & 4 & l_1 \\ 0 & 0 & -1 & -1 & l_2 - 2l_1 \\ 0 & 0 & -2 & -2 & l_3 - l_1 \end{array} \right). \\ &\xrightarrow{L_2 \leftarrow -L_2} \left(\begin{array}{cccc|c} 1 & 2 & 3 & 4 & l_1 \\ 0 & 0 & 1 & 1 & 2l_1 - l_2 \\ 0 & 0 & -2 & -2 & l_3 - l_1 \end{array} \right). \\ &\xrightarrow[L_3 \leftarrow L_3 + 2L_2]{L_1 \leftarrow L_1 - 3L_2} \left(\begin{array}{cccc|c} 1 & 2 & 0 & 1 & -5l_1 + 3l_2 \\ 0 & 0 & 1 & 1 & 2l_1 - l_2 \\ 0 & 0 & 0 & 0 & 3l_1 - 2l_2 + l_3 \end{array} \right). \end{aligned}$$

Le système est compatible si et seulement si la dernière ligne ne donne pas une contradiction, c'est-à-dire si et seulement si

$$3l_1 - 2l_2 + l_3 = 0.$$

Ainsi,

$$\text{Im } A = \left\{ \begin{pmatrix} l_1 \\ l_2 \\ l_3 \end{pmatrix} \in \mathbb{R}^3 \mid 3l_1 - 2l_2 + l_3 = 0 \right\}.$$

Formellement, ce sont les équations de l'image qui apparaissent dans les dernières lignes de la matrice échelonnée car :

$$L \in \text{Im } A \iff \text{le système } AX = L \text{ d'inconnue } X \text{ possède une solution.}$$

Intuitivement, trouver les équations d'un sous-espace vectoriel, c'est trouver quelles relations existent entre les coefficients. En échelonnant la matrice en

lignes, si on trouve une ligne nulle, c'est bien qu'on a trouvé une relation entre cette ligne et les précédentes : la dernière colonne enregistre ces relations, et c'est pour cela qu'on peut y lire les équations de l'image.

Méthode : donner des équations de l'espace engendré par des vecteurs. Soient $u_1, \dots, u_p \in \mathbb{R}^n$ et $E = \text{Vect}(u_1, \dots, u_p)$. On forme la matrice dont les *colonnes* sont les vecteurs u_i :

$$A = \begin{pmatrix} u_1 & \cdots & u_p \end{pmatrix}.$$

On ajoute ensuite une colonne $L = \begin{pmatrix} l_1 & \cdots & l_n \end{pmatrix}^\top$ et on échelonne en lignes la matrice augmentée $(A \mid L)$, en appliquant chaque opération à toute la ligne, y compris à la dernière colonne.

À la fin de l'algorithme, pour tout $i \in \{1, \dots, n\}$, le dernier coefficient de la ligne i contient une combinaison linéaire de la forme

$$\beta_{i,1}l_1 + \cdots + \beta_{i,n}l_n,$$

qui correspond à l'équation

$$\beta_{i,1}x_1 + \cdots + \beta_{i,n}x_n = 0.$$

Les équations de l'espace engendré (de façon équivalente, de l'image de la matrice) sont les équations associées aux lignes nulles à la fin de l'algorithme.

Note : Cette manière de faire est tout à fait équivalente à celle présentée dans le cours, où on repasse aux systèmes d'équations. La seule différence est qu'ici, on manipule une matrice et on évite ainsi d'introduire les variables notées t_i dans le cours. **Dans tous les cas, choisissez la méthode avec laquelle vous faites le moins d'erreurs.**

Astuce : Comme avant, si vous n'avez pas augmenté la matrice avant de l'échelonner, vous pouvez récupérer les équations en "remontant" les opérations élémentaires qui ont donné une ligne nulle.