Devoir Maison n°1

AP3 Algorithmique et programmation — L2 mathématiques

Pour le 1er décembre 2016

Le but du devoir est de programmer un algorithme résolvant les systèmes de n équations linéaires à n inconnues ($n \geq 1$). Pour cela, on va implémenter la méthode que vous connaissez tous depuis le lycée : le pivot de Gauss. Rappelons que cette méthode consiste à soustraire un multiple d'une ligne du système donnée à toutes les autres lignes afin d'y éliminer l'occurence d'une variable. En répétant le processus astucieusement (c'est-à-dire en évitant de faire "réapparaître" les variables déjà éliminées dans une ligne), on obtient un système triangulaire de la forme

$$\begin{cases} a_{1,1}x_1 + a_{1,2}x_2 + \cdots + a_{1,n}x_n = b_1 \\ \qquad\qquad a_{2,2}x_2 + \cdots + a_{2,n}x_n = b_2 \\ \qquad\qquad\qquad\qquad\qquad \vdots \\ \qquad\qquad\qquad\qquad a_{n,n}x_n = b_n \end{cases}$$

qui est alors évident à résoudre.

La formalisation mathématique d'un système (à coefficients dans $\mathbb{R}$ disons) est la donnée d'un couple (A, b) avec $A \in \mathcal{M}_n(\mathbb{R})$ et $b \in \mathbb{R}^n$, et une solution du système est un vecteur $x \in \mathbb{R}^n$ tel que $Ax = b$. On suppose que le système admet un solution unique, c'est-à-dire que A est inversible. En particulier, aucune des colonnes de A n'est nulle.

Exercice 1.

(i) En écrivant

$$A = \begin{pmatrix} a_{1,1} & {}^{\mathrm{t}}w \\ v & A' \end{pmatrix} \qquad \text{avec } v, w \in \mathbb{R}^{n-1},\ A' \in \mathcal{M}_{n-1}(\mathbb{R}),\ \mathbf{a_{1,1} \neq 0}$$

trouver une matrice $B \in \mathcal{M}_{n-1}(\mathbb{R})$ et un vecteur $c \in \mathbb{R}^{n-1}$ tels qu'une solution $y \in \mathbb{R}^{n-1}$ de (B, c) produise une solution x du système initial (A, b) comme suit :

$$x = \begin{pmatrix} (b_1 - {}^{\mathrm{t}}wy)/a_{1,1} \\ y \end{pmatrix}$$

Indice : il s'agit juste de formaliser la *soustraction de lignes* que vous faites habituellement quand vous réalisez un pivot de Gauss. Regardons ensemble un exemple concret de 3 équations linéaires à 3 inconnues :

$$\begin{cases} 3\,x_1 + 6x_2 - x_3 = 2 \\ \qquad 5x_1 + x_3 = 7 \\ -x_1 + \dfrac{3}{4}x_2 + 2x_3 = 0 \end{cases}$$

Effectuons alors la première étape d'un pivot de Gauss avec le coefficient entouré ci-dessus : en notant L_1, L_2, L_3 les trois lignes, on doit effectuer les opérations $L_2 \leftarrow L_2 - \frac{5}{3}L_1$ et $L_3 \leftarrow L_3 + \frac{1}{3}L_1$. On se retrouve alors avec le système équivalent :

$$\begin{cases} 3x_1 + 6x_2 - x_3 = 2 \\ \quad -10x_2 + \dfrac{8}{3}x_3 = \dfrac{11}{3} \\ \quad \dfrac{11}{4}x_2 + \dfrac{5}{3}x_3 = \dfrac{2}{3} \end{cases}$$

On sait maintenant qu'il ne nous reste plus qu'à résoudre le système de deux équations linéaires à deux inconnues donné par les deux dernières lignes pour trouver x_2 et x_3. La dernière inconnue x_1 du système total s'en déduira alors : $x_1 = (2 - (6x_2 - x_3))/3$.

Essayons maintenant d'exprimer cette étape faite “à la main” dans les notations de l'énoncé. La matrice A et le vecteur b sont respectivement donnés par :

$$A = \begin{pmatrix} 3 & 6 & -1 \\ 5 & 0 & 1 \\ -1 & 3/4 & 2 \end{pmatrix} \qquad b = \begin{pmatrix} 2 \\ 7 \\ 0 \end{pmatrix}$$

On a donc également :

$$a_{1,1} = 3 \neq 0, \quad w = \begin{pmatrix} 6 \\ -1 \end{pmatrix}, \quad v = \begin{pmatrix} 5 \\ -1 \end{pmatrix}, \quad A' = \begin{pmatrix} 0 & 1 \\ 3/4 & 2 \end{pmatrix}$$

On lit la matrice B et le vecteur c cherchés dans les deux dernières lignes du système après pivot. Ils ont été produits par les opérations sur les lignes :

$$B = \begin{pmatrix} 0 - (5 \times 6)/3 & 1 - (5 \times (-1))/3 \\ 3/4 - ((-1) \times 6)/3 & 2 - ((-1) \times (-1))/3 \end{pmatrix} = \begin{pmatrix} -10 & 8/3 \\ 11/4 & 5/3 \end{pmatrix},$$
$$c = \begin{pmatrix} 7 - (5 \times 2)/3 \\ 0 - ((-1) \times 2)/3 \end{pmatrix} = \begin{pmatrix} 11/3 \\ 2/3 \end{pmatrix}$$

Supposons qu'on trouve x_2 et x_3, alors on obtient une solution globale :

$$x = \begin{pmatrix} (2 - (6 \times x_2 + (-1) \times x_3))/3 \\ x_2 \\ x_3 \end{pmatrix}$$

Étudiez bien le code couleur et généralisez à partir de là.

(ii) Que faire quand $a_{1,1} = 0$? (Dans l'exemple précédent, pouvoir diviser par 3 était crucial.)

Indice : que faites-vous habituellement quand vous ne pouvez pas choisir le premier coefficient de la première ligne comme pivot ? Rappelez-vous qu'aucune colonne de A n'est nulle.

(iii) En déduire un algorithme **récursif** résolvant le problème. L'implémenter en `python` (on utilisera des `float` pour les coefficients des matrices).

(iv) Évaluer la complexité de l'algorithme.

Exercice 2. L'algorithme précédent prend en entrée A et b et retourne un solution x du système (A, b). Ainsi, si l'on doit résoudre un grand nombre de système (A, b), (A, c), (A, d), etc. avec toujours la **même matrice** A, on doit appeler l'algorithme autant de fois que de systèmes à résoudre. Pourtant, l'on voit bien qu'on va seulement répéter un certain nombre de calcul...

Pour remédier au problème, on va exprimer ce qu'on a fait "à la main" dans l'algorithme précédent sous la forme d'un calcul matriciel. On peut montrer que toute matrice inversible A admet une décomposition LUP, c'est-à-dire qu'il existe une matrice de permutation P, une matrice triangulaire supérieure U et une matrice triangulaire inférieure L avec seulement des 1 sur sa diagonale telles que

$$PA = LU$$

(Rappelons qu'une matrice de permutation a comme coefficients seulement des 0 et des 1 et qu'elle contient un unique 1 par ligne et par colonne ; autrement dit c'est la matrice de changement de base depuis $(e_{\sigma(1)}, \dots, e_{\sigma(n)})$ vers la base canonique $(e_1, \dots, e_n)$ pour σ une permutation de $\{1, \dots, n\}$.)

1. Si $a_{k,1} \neq 0$ pour un certain $1 \leq k \leq n$, trouver une matrice de permutation Q tel que

$$QA = \begin{pmatrix} a_{k,1} & {}^{\mathrm{t}}w \\ v & A' \end{pmatrix} \qquad \text{avec } v, w \in \mathbb{R}^{n-1},\ A' \in \mathcal{M}_{n-1}(\mathbb{R})$$

   Trouver alors une matrice B telle que

$$QA = \begin{pmatrix} 1 & 0 \\ v/a_{k,1} & I_{n-1} \end{pmatrix} \begin{pmatrix} a_{k,1} & {}^{\mathrm{t}}w \\ 0 & B \end{pmatrix}$$

   (Indice : B est du même acabit qu'en exercice 1.)

2. Si B admet une décomposition LUP donnée par des matrices P',U' et L', montrer que A admet également une telle décomposition où la matrice de permutation est :

$$P = \begin{pmatrix} 1 & 0 \\ 0 & P' \end{pmatrix} Q$$

3. En déduire un algorithme récursif (et l'implémenter en python) déterminant la décomposition LUP d'une matrice inversible.

Exercice 3. En déduire une fonction `inv(A)` qui prend en paramètre une matrice A supposée inversible et qui renvoient l'inverse de A.

Note : si l'on a pas fait l'exercice 2, on peut tout de même résoudre celui-ci avec l'exercice 1 (même si c'est moins efficace).[1]

Exercice 4 (Bonus). Améliorer l'algorithme de l'exercice 2 de façon à détecter si la matrice A est réellement inversible.

1. En particulier, il ne sera pas pénalisé d'utiliser l'exercice 1 ici : ceci reviendrait à pénaliser deux fois l'échec à l'exercice 2...