# Analyse matricielle, TP n°2

27 septembre 2018

## 1 Finir le TP n°1

Et ensuite, après avoir terminé l'algorithme du pivot simple, vous pouvez passer au

## 2 Pivot partiel :

Deux raisons, vues en cours, peuvent nous conduire à améliorer la méthode du pivot simple :

**1.** le pivot simple peut échouer parce qu'on rencontre un pivot nul,

**2.** même s'il n'échoue pas, on a intérêt pour la stabilité numérique, à travailler avec un pivot maximal.

Dans les deux cas, on est amené à rechercher à chaque étape le pivot maximal (en valeur absolue) sur la colonne.

**Exercice.** Ecrivez donc la fonction `PivotMax(A,B)` en faisant un copier/coller de `PivotSimple(A,B)` et en l'améliorant.

**Exercice.** A partir d'une matrice et d'un second membre aléatoires, comparer les précisions des solutions des deux méthodes (avec ou sans stratégie de pivot partiel).

En principe, le pivot partiel doit être le meilleur.

**Exercice.** Comparez votre pivot partiel avec la méthode `solve_right` qui implante une stratégie de pivot maximal et donc qui devrait être encore meilleure en précision que votre pivot partiel. Essayez, pour voir ce qu'il en est vraiment.

## 3 Décomposition LU :

La décomposition LU d'une matrice carrée est son écriture sous forme du produit d'une matrice L triangulaire inférieure avec que des 1 sur la diagonale, par une matrice U triangulaire supérieure.

L'existence d'une telle décomposition est équivalente à ce que l'algorithme du pivot de Gauss "naïf" fonctionne. Si ce n'est pas le cas, la décomposition reste possible sous la forme $A = PLU$ ou $PA = LU$ où $P$ est dans mes deux cas une matrice de permutation (remplie uniquement de 0 et de 1 avec un seul 1 sur chaque ligne et un seul 1 sur chaque colonne).

Cette factorisation est alors équivalente à la technique du pivot partiel (recherche d'un élément non nul dans la colonne où on a trouvé un pivot nul pour remplacer ce pivot nul, ou éventuellement recherche dans tout le bas de la colonne d'un élément non nul "optimal" (par exemple le plus grand possible en valeur absolue) pour le mettre en position de pivot).

La factorisation LU fait partie des fonctions classiques d'une bibliothèque numérique. Vous allez ici directement utiliser cette fonction qui est en fait une méthode : `A.LU( )` renvoie un triplet `P,L,U` de matrices satisfaisant les conditions voulues et telles que `PLU=A`.

**Exercice.** Fabriquez une matrice aléatoire d'entiers `A`, de taille disons 5, puis calculez sa factorisation `P,L,U` et vérifiez que le produit est bien égal à `A`. Vous verrez que `P` n'est (en général) pas l'identité : c'est parce que la méthode recherche le meilleur pivot sur la colonne et donc pour cela échange des lignes. On peut cependant imposer le choix du premier pivot non nul (et ainsi obtenir - si c'est possible - la factorisation `LU`). Lisez l'aide en ligne (en tapant `A.LU?`) pour voir comment on peut interdire les permutations de lignes.

Vous remarquerez que `A` étant une matrice entière, les matrices `L` et `U` sont à coefficients rationnels. De même, si au départ `A` est à coefficients rationnels, `L` et `U` sont aussi à coefficients rationnels et leur calcul est bien sûr exact.

**Exercice.** Prendre pour `A` une matrice à coefficients dans $\mathbb{Q}$ et de taille 50 (`A=random_matrix(QQ,50)`) puis vérifier que la norme de `A-PLU` est bien nulle.

On peut aussi factoriser des matrices réelles (de type `RDF`).

**Exercice.** Faites un essai. Si vous constatez un petit problème, regardez l'aide en ligne!!

Si, après avoir factorisé une matrice `A` de type `RDF`, vous avez calculé `PLU`, vous n'avez pas retrouvé `A`. Vous pouvez alors regarder l'aide en ligne en tapant `A.LU?` Mais à présent, `A` est une matrice de réels et vous pourrez constater que l'aide en ligne vous indique que la méthode renvoie `P,L,U` telles que `PA=LU` (et pas `A=PLU`). Donc : selon que le calcul est exact (avec de rationnels) ou approché (avec des flottants), la méthode appelée n'est pas la même : ce sont deux fonctions distinctes des bibliothèques. Et c'est le type de la matrice `A` qui indique à `A.LU?` quelle descrption de fonction il doit renvoyer.

**Remarque.** $\mathtt{PA=LU} \Longleftrightarrow \mathtt{A=P}^{-1}\mathtt{LU} \Longleftrightarrow \mathtt{A=}{}^{t}\mathtt{PLU}$, car `P`, matrice de permutation, est orthogonale.

**Exercice.** Pour terminer, vous pouvez tester la méthode `LU` sur de grosses matrices (`A=random_matrix(RDF,500)`) et vérifier que la norme de `PA-LU` reste raisonnablement petite.