

**Houseaux Laurie**

**Jarretier Adrien**

# Projet Matriciel

## Document explicatif

### 1. Qui a codé fait quoi? comment avez-vous travaillé ensemble?

Pour les deux premières parties nous avons écrit les cartouches ensemble puis nous nous sommes réparti les fonctions à coder de cette façon :

Parties	Adrien	Laurie
<b>Partie 1 Matrices Permutations</b>	<ul style="list-style-type: none"><li>• PermutationAleatoire</li><li>• MatricePermutation</li><li>• PermutationAssociée</li><li>• ProduitPermutDG</li></ul>	<ul style="list-style-type: none"><li>• EstPermutation</li><li>• MatricePermutationInverse</li><li>• ProduitPermutD</li></ul>
<b>Partie 2 Transvections</b>	<ul style="list-style-type: none"><li>• MatriceTransvection</li><li>• ProduitTransvectionD</li></ul>	<ul style="list-style-type: none"><li>• EstTransvection</li><li>• MatriceTransvectionInverse</li><li>• ProduitTransvectionG</li></ul>

On a codé la décomposition PLU ensemble ainsi que la partie 4 (Application de la décomposition PLU).

### **2. Qu'est ce qui a été traité? testé? qu'est ce qui est fonctionnel?**

Nous avons traité toutes les questions et nous avons testé tous nos codes. Toutes les fonctions que nous avons rendu sont fonctionnelles.

### **3. Quelles sont les limitations du programme : temps d'exécution, taille des matrices...**

Nous avons rencontrés des problèmes dans la décomposition PLU avec les arrondis ; par exemple au lieu de mettre 0 il nous affichait un chiffre exposant -16.

On aurait pu améliorer le code en effectuant des calculs seulement avec des rationnels.

Mais pour régler le problème rapidement nous avons décidé d'arrondir à  $10^{-4}$ .

## 4. Les problèmes (de tous ordres ) rencontrés.

Nous avons eu un problème de return mal placé dans la fonction PLU. Puis nous avons eu d'autres petits problèmes d'inattention (erreur d'indices, erreur d'indentation,...) très facilement réglés avec des Debugs.

## 5. Toute autre chose que vous jugerez utile.

Notre code est en python 3.4.

On utilise deux fichiers en plus :

- **SimpleMatricesFonctions.py** (utilisé un peu partout) regroupant :
  - **AfficherMat** : pour afficher la matrice dans la console
  - **MatId** : pour créer une matrice identité
  - **MatNulle** : pour créer une matrice nulle
  - **Transposee** : pour transposer une matrice
- **Matrice\_inverse\_cofacteurs.py** (utilisé uniquement dans les test pour comparer) on utilise uniquement :
  - **Inverse** : qui renvoie l'inverse d'une matrice
  - **Produit** : qui renvoie le produit de deux matrices

Dans la partie transvection on a rajouté une fonction **TransvectionAssociee** qui permet de récupérer la transvection associée à une matrice de transvection donnée à la manière de permutation associée.

Le fichier **test.py** permet de lancer chaque test de manière interactive.