

## TP d'Algèbre Linéaire Numérique (ALN) – S. G. Petiton – Polytech'Lille, IS2A3, promo 2025

### A – Programmation et premiers tests

#### 1 - Méthode de Householder

Programmez la méthode de Householder pour résoudre un système linéaire  $Ax=b$ , d'ordre  $n$ .  
Testez la méthode à l'aide de l'exemple vue en cours (à corriger)  
Calculer la norme 2 du résidu  $r = Ax-b$ , conclure

#### 2 - Méthode de Cholesky

Programmez la méthode de Cholesky pour résoudre un système linéaire  $Cx=b$ , d'ordre  $n$ .  
Testez la méthode à l'aide de l'exemple vue en cours (en partant d'un produit  $C=A^T A$ )  
Calculer la norme 2 du résidu  $r = Cx-b$ , conclure

#### 3 - Méthode des moindres carrés à l'aide de la factorisation QR

En partant de la méthode de Householder développée, programmez la méthode des moindres carrés utilisant la factorisation QR (cas matrice  $m \times n$ ,  $m > n$ )  
Testez avec la matrice  $A$  utilisée en TD. Calculez la norme 2 du résidu  $r=Ax-b$ , conclure

#### 4 - Méthode de moindres carrés par la méthode de Cholesky

En utilisant la méthode de Cholesky développée, programmez la méthode des moindres carrés à l'aide de ce programme. Testez en utilisant la matrice du TD. Calculez le résidu du problème d'origine (bien distinguer la matrice d'origine et celle, carrée symétrique définie positive, que nous générons). Conclure

#### 5 – comparer les différents résidus obtenus

6 – Dans le cas de la méthode aux moindres carrés avec la factorisation QR, sur l'exemple traité en TD, affichez les matrices  $m$  par  $n$  et le vecteur de taille  $m$  avant la remontée. Que remarque-t-on. Expliquez.

### B – Génération de matrices et préparation des codes pour d'autres tests

1 – Génération de matrices aléatoires : matrice  $n \times n$  ou  $m \times n$ , avec des éléments compris entre  $-a$  et  $a$  pour la matrice et  $-b$  et  $b$  pour les vecteurs. Prenez au départ  $a = 10.0$  et  $b = 1.00$ . Tester les 4 méthodes développées avec  $n = 50$  (et  $m = 75$  quand requis)

2 – Implémenter la prise de temps de calcul pour chaque méthode. Attention, la construction des matrices et les affichages écran et les I/O dans des fichiers ne doivent pas être dans les parties concernées.

3 – Préparez-vous à pouvoir changer la précision des calculs entre 32 et 64 bits.

## C – Expérimentation

1 - Expérimentez avec des matrices de plus en plus grandes, jusqu'à plusieurs milliers (à  $n$  fixé ou  $m$  fixé pour les problèmes aux moindres carrés). Analysez le temps de calcul, et l'évolution des résidus pour plusieurs valeurs d' $\varepsilon$ .

2 - Expérimentez en fonction de la précision des calculs (32 et 64 bits IEEE FP)

3 – Expérimentez en fonction de plusieurs valeurs de «  $a$  » et «  $b$  ».

## D – Conclusion et rédaction

Selon les résultats obtenus, illustrés par des courbes et des tableaux, vous concluez en vous référant le plus possible au cours.

Les rapports, individuel, doivent être envoyés à [serge.petiton@univ-lille.fr](mailto:serge.petiton@univ-lille.fr) en version pdf NON COMPRESSE, avant le dimanche soir suivant la fin de votre période école actuelle. Si vous ne recevez pas un mail de bonne réception sous 3 jours, contactez-moi. Les fichiers envoyés seront intitulés NON-PRENOM-TP-ALN-2025.pdf

En annexe du rapport, et uniquement là, donnez votre code Python avec un minimum de commentaire.

**Remarque.** A la fin du prochain TP, ou du suivant, je passe voir chaque étudiant pour poser des questions sur les codes développés. De même lors du dernier TP.