

## Analyse numérique I

### Projet à réaliser sur Octave : Sur sensibilité de systèmes linéaires

Ce projet vise à analyser l'effet du conditionnement des matrices sur la stabilité numérique des solutions d'un système linéaire  $Ax = b$ . L'objectif est d'explorer comment les petites perturbations dans les données influencent la solution numérique et d'évaluer la robustesse des algorithmes de résolution selon le conditionnement de la matrice.

### Travaux demandés

#### 1. Calcul du nombre de condition

- Écrire une fonction Octave `Conditionnement1(A)` qui calcule le nombre de condition de la matrice  $A$  dans la norme 1 :  $\kappa_1(A) = \|A\|_1 \cdot \|A^{-1}\|_1$ .
- Faire de même pour les normes  $\infty$  et 2 : `ConditionnementInf(A)`, `Conditionnement2(A)`.

#### 2. Erreur relative

- Écrire une fonction `ErreurRelative(A,b,x_exact)` qui résout  $Ax = b$  et renvoie l'erreur relative  $\frac{\|x_{\text{num}} - x_{\text{exact}}\|_1}{\|x_{\text{exact}}\|_1}$ .

#### 3. Étude de perturbation du second membre

- Pour une matrice  $A$  bien conditionnée et un vecteur exact  $x_{\text{exact}}$ , calculer  $b = Ax_{\text{exact}}$ .
- Perturber  $b$  par un petit vecteur aléatoire  $\delta$  :  $\tilde{b} = b + \delta$ .
- Étudier l'effet de cette perturbation sur la solution  $x$ .

#### 4. Perturbation de la matrice

- Générer une matrice  $A$  bien conditionnée et créer  $A_\varepsilon = A + \varepsilon B$  où  $B$  est aléatoire.
- Étudier les effets sur la solution pour plusieurs valeurs de  $\varepsilon$ .
- Tracer l'évolution de l'erreur relative et du nombre de condition en fonction de  $\varepsilon$ .

#### 5. Étude de matrices classiques

- Étudier les matrices suivantes : matrice de Hilbert  $H_n$ , matrice de Vandermonde, matrice triangulaire.
- Pour chaque matrice, calculer  $\kappa_1$ ,  $\kappa_2$ ,  $\kappa_\infty$  et résoudre  $Ax = b$  avec  $x_{\text{exact}} = (1, 1, \dots, 1)^T$ .
- Analyser les erreurs relatives.

#### 6. Comparaison LU vs résolution directe

- Résoudre  $Ax = b$  via la commande directe `A\b`.
- Résoudre le même système en utilisant la factorisation LU avec `lu(A)`.
- Comparer les erreurs et les performances selon le conditionnement.

#### 7. Script de visualisation

- Écrire un script `FigureConditionnement()` qui :
  - Affiche les erreurs relatives pour différentes valeurs de  $\varepsilon$ .
  - Trace le nombre de condition et l'erreur relative (en échelle logarithmique).
  - Compare les comportements selon les types de matrices (aléatoires, Hilbert, diagonales...).

#### 8. Bonus

- Écrire une fonction `EstMalConditionnee(A,seuil)` qui renvoie vrai si  $\kappa(A) > \text{seuil}$ .
- Tester des régularisations :  $A_\lambda = A + \lambda I$  pour améliorer la stabilité.
- Étudier l'effet du bruit gaussien sur les données (matrice ou second membre).

---

**Remarque :** Tous les codes doivent être clairement commentés et accompagnés de graphiques lisibles avec titres, légendes et unités. Le rapport final devra présenter les résultats sous forme de tableaux, courbes et interprétations.

Le rapport finale doit être envoyer via le lien :

[https://docs.google.com/forms/d/e/1FAIpQLSf89NyDL\\_X5JydntTpDBfGxsf30RbGM2EDd\\_viCxxthQvwWKQ/viewform?usp=header](https://docs.google.com/forms/d/e/1FAIpQLSf89NyDL_X5JydntTpDBfGxsf30RbGM2EDd_viCxxthQvwWKQ/viewform?usp=header)

**Date limite 03 Juin 2025**

**Bonne chance.**