

Examen, INFO601_CMI – Algorithmique numérique, Session 1

Documents autorisés : une feuille recto/verso.

Les exercices sont indépendants. Le barème est indicatif. Il dépasse volontairement 20 pour que vous ayez le choix dans les exercices.

Exercice 1. Problème bien posé, conditionnement (/4)

On définit la droite (D) comme l'ensemble des points \mathbf{x} du plan qui satisfont $\mathbf{n} \cdot \mathbf{x} = a$, avec \mathbf{n} un vecteur de longueur 1 (qui est orthogonal à la droite) et a un réel.

- [/1] Montrez que $|\mathbf{n} \cdot \mathbf{p} - a|$ est la distance du point \mathbf{p} à la droite (D).
- [/1] On s'intéresse au problème de déterminer la distance entre un point quelconque \mathbf{p} et la droite (D) définie par \mathbf{n} et a . Formaliser ce problème en écrivant sa résolvante $G(\mathbf{p}, \mathbf{n}, a)$. Quel est le domaine valide des données? Est-ce que ce problème est bien posé sur ce domaine?
- [/2] Montrez que le conditionnement absolu est borné par une fonction ne dépendant que de $\|\mathbf{p}\|$ sauf lorsque $\mathbf{p} \in (D)$ tandis que le conditionnement relatif peut exploser autour de la droite (D).

Exercice 2. Approximation de dérivées (/7)

On cherche à approcher les dérivées premières d'une fonction f , supposée suffisamment lisse (au moins C^3), en des points (x_i) , pour i de 0 à n , espacés régulièrement d'un pas h . On connaît les valeurs de f en ces points x_i , que l'on pourra noter f_i pour $f(x_i)$.

- [/2] Proposez une formule qui approche $f'(x)$ à l'ordre 2, i.e. erreur de l'ordre de $O(h^2)$, valide pour tous les points x_i internes, i.e. i de 1 à $n - 1$. Justifiez cette formule.
- [/2] En utilisant un développement de Taylor à l'ordre 3 en x_0 pour les pas h et $2h$, trouvez une formule qui approche $f'(x_0)$ à l'ordre 2, et qui utilise f_0, f_1, f_2 et h .
- [/1] Procédez similairement pour obtenir la formule pour approcher $f'(x_n)$ à l'ordre 2.
- [/2] Si maintenant les valeurs de f en ces points (x_i) forment le vecteur $\mathbf{f} = [f(x_0) \ \dots \ f(x_n)]^T$, écrivez la matrice D d'ordre $n + 1$ telle que $D\mathbf{f}$ est le vecteur approchant les dérivées de f en tous les points, d'ordre 2 partout.

Exercice 3. Conditionnement d'un système linéaire (/5)

Soit e un nombre positif (que l'on prendra petit non nul). Soit $\mathbf{A}_e = \begin{bmatrix} 1 & -e \\ 1 & e \end{bmatrix}$. On rappelle que la norme de Frobenius $\|\mathbf{A}_e\|$ de la matrice \mathbf{A}_e est simplement la racine carrée de la somme des carrés de tous les coefficients de \mathbf{A}_e .

- (/0,5) Calculez la norme $\|\mathbf{A}_e\|$. Que vaut-elle si $|e|$ est proche de 0?
- (/1,5) Montrez que \mathbf{A}_e est inversible et calculez son inverse.
- (/0,5) Calculez la norme $\|\mathbf{A}_e^{-1}\|$.
- (/0,5) Quel est environ le conditionnement de \mathbf{A}_e ?
- (/1,5) Quelle précision peut-on attendre sur la solution (x, y) du système

$$\begin{bmatrix} 1 & -0.000001 \\ 1 & 0.000001 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix},$$

en supposant une précision numérique $u = 10^{-16}$.

- (/0,5) Quelle erreur sur le résidu peut-on espérer avec un algorithme de type pivot de Gauss (PLU)?

Exercice 4. Trace d'une matrice creuse (/4)

Soit A une matrice creuse carrée de taille $n \times n$ en format CSR. Ecrivez la fonction Python `Trace(A)` qui calcule la trace la matrice A , c'est-à-dire la somme de ses éléments diagonaux, de façon le plus efficace possible selon vous.

Quelle est la complexité de cette fonction en fonction de n et/ou de n_{nz} le nombre de coefficients non nuls de A ?

Exercice 5. Soustraction de deux matrices CSR (/6)

Soit A et B deux matrices carrées au format CSR, de tailles $n \times n$. Ecrire une fonction `Soustraction(A,B)` qui retourne une nouvelle matrice au format CSR qui représente $A - B$. Attention certains coefficients peuvent s'annuler dans la soustraction.

Quelle est la complexité de `Soustraction`, en fonction de n et/ou n_A et n_B le nombre de coefficients non nuls de A et B respectivement.

Rappels et notations matrices creuses

On rappelle que les matrices creuses en mode CSR (compressed sparse row) sont représentés à l'aide des informations suivantes (A est une matrice creuse CSR) :

- `A.rows` et `A.cols` donnent le nombre de lignes et colonnes de A ,
- `A.data` est le tableau contenant les n_{nz} coefficients non nuls de A ,
- `A.indices` est le tableau des colonnes de chaque coefficient non nul de A ,
- `A.indptr` est le tableau stockant le numéro du premier coefficient sur chaque ligne,
- `x[i]` retourne la i -ème valeur du vecteur \mathbf{x} (i.e. x_i),
- `x.size()` retourne la taille du vecteur \mathbf{x} (i.e. m),
- `vector(n)` crée et retourne un vecteur de taille n , rempli de zéros.
- `ndarray((m,n))` crée et retourne une matrice pleine de taille $m \times n$
- `coo_matrix((data, (rows,cols)), shape=(m,n))` crée et retourne une matrice COO de taille $m \times n$, remplie avec les coefficients `data[k]`, à la ligne `rows[k]` et la colonne `cols[k]`.
- `csr_matrix((m,n))` crée et retourne une matrice CSR de taille $m \times n$, vide.
- `csr_matrix((data, indices, indptr), shape=(m,n))` crée et retourne une matrice CSR de taille $m \times n$, remplie exactement avec les tableaux donnés correspondant au format.
- `A.tocoo()` retourne une copie de A mais dans le format COO
- `A.tocsr()` retourne une copie de A mais dans le format CSR