

# APPROXIMATION DE FAIBLE RANG

Notes de cours Marie Billaud-Friess

2024 - 2025

 $\Rightarrow$  ATTENTION. Ce polycopié est régulièrement mis à jour et corrigé. Si vous découvrez des erreurs, n'hésitez pas à me les communiquer. Toutes les remarques ou questions permettant d'en améliorer le contenu peuvent être envoyées par mail à l'adresse ci-dessous.

Dernière mise à jour : 10 décembre 2024

Marie Billaud-Friess marie.billaud\_friess@centrale-med.fr

Centrale Méditerranée Institut de Mathématiques de Marseille

## TABLE DES MATIÈRES

1	Introduction	1				
1.1	Un problème modèle	1				
1.2	Exemples d'application	1				
	1.2.1 Compression d'image	1				
	1.2.2 Réduction de modèle	2				
	1.2.3 Analyse en composantes principales	3				
2	Compléments d'analyse matricielle					
2.1	Rang d'une matrice	7				
2.2	Normes	8				
	2.2.1 Exemples importants dans le cas $E = \mathbb{R}^n$	9				
	2.2.2 Exemples importants dans le cas $E = \mathbb{R}^{m \times n}$	9				
2.3	Factorisation de matrices	10				
	2.3.1 Factorisation de matrices rectangulaires	11				
	2.3.2 Factorisation des matrices de rang $r$	11				
	2.3.3 Décomposition en valeurs singulières	12				
3	Approximation de faible rang - méthodes déterministes et aléatoires 17					
3.1	Théorie	17				
3.2	Algorithmes	19				
	3.2.1 Premiers algorithmes détermistes	19				
	3.2.2 Vers des algorithmes aléatoires	19				

# Chapitre 1

INTRODUCTION

Dans ce cours nous parlerons de meilleure approximation sous format de faible rang d'une matrice. Ce problème est largement présent en mathématiques, dans le domaine de la réduction de modèle en analyse numérique, la compression d'image en traitement du signal, ou encore l'analyse en composantes principales en statistiques.

L'objectif de ce cours est de donner quelques outils de l'analyse matricielle dans but de résoudre un tel problème. En particulier nous ferons le lien avec la décomposition en valeurs singulières. D'un point de vue pratique, des algorithmes déterministes voire aléatoires seront discutés.

## 1.1 Un problème modèle

Soient  $n, m \in \mathbb{N}^*$ . Nous noterons par la suite  $\mathbb{R}^{m \times n}$  l'ensemble des matrices de taille  $m \times n$  à coefficients réels. On appellera **approximation de faible rang** d'une matrice  $A \in \mathbb{R}^{m \times n}$ , toute solution au problème de minimisation suivant

$$\min_{\substack{Z \in \mathbb{R}^{m \times n} \\ \operatorname{rang}(Z) \le r}} \|A - Z\|,\tag{1.1.1}$$

où  $\|\cdot\|$  est une norme matricielle qu'on précisera ultérieurement. On cherche la **meilleure approximation** parmi les matrices de rang r de la matrice A.

## 1.2 Exemples d'application

Nous présentons par la suite plusieurs exemples d'application pour lesquels il est utile de résoudre (3.0.1).

## 1.2.1 Compression d'image

Une image est stockée sur ordinateur dans un fichier sous forme de matrice A (2D pour les images en niveau de gris, 3D pour les images en couleur). Les entrées de la matrice A sont les valeurs des pixels, généralement des entiers entre 1 et 255 (ou un réel entre [0,1] après division par 255). Pour éviter d'envoyer une image trop volumineuse, une façon de procéder est de la compresser tout en dégradant le moins possible sa qualité. Pour réaliser cette compression, il est possible de calculer une approximation de faible rang de A au sens du problème (3.0.1). Nous donnons un exemple dans ce qui suit pour le stockage d'une image en niveau de gris (voir Figure 1.2.1) de taille 2,6Mo.



FIGURE 1.2.1 – Osaka Dotombori (3024×4032 pixels) : image originale (gauche), versions compressées (droite) pour  $r \in \{50, 100, 200\}$ .

L'image précédente  $A \in \mathbb{R}^{m \times n}$  est compressée. Pour cela on calcule son approximation comme une matrice de  $\mathbb{R}^{m \times n}$  avec différents rangs  $r \in \{50, 100, 200\}$ . Les images ainsi obtenues n'occupent plus que 1 Mo de mémoire au maximum.

### 1.2.2 Réduction de modèle

On s'intéresse désormais à la résolution d'une équation parabolique dépendant de paramètres. Etant donné  $\Omega \times I = (0, 1) \times [0, 1]$  le domaine spatio-temporel, nous cherchons  $u(\cdot; \xi)$  solution de

$$\partial_t u(x,t;\xi) - \mu(\xi)\partial_{xx}^2 u(x,t;\xi) = f(x,t), \quad \text{dans } \Omega \times I, \tag{1.2.1}$$

étant donnée la condition initiale  $u^0(\cdot,\xi): \Omega \to \mathbb{R}$ . On suppose des conditions de Dirichlet homogène i.e.  $u(x,t,\xi) = 0$  pour x = 0, 1. La solution u depend du paramètre  $\xi \in \Xi = [0.01, 0.06]$  au travers de la viscosité  $\mu(\xi) = \xi$  et de la condition initiale

$$u^{0}(x;\xi) = \sin(4x\xi)e^{-100(x-10\xi)^{2}}.$$

Le terme source  $f: \Omega \times I \to \mathbb{R}$  est défini par  $f(x,t) = \sin(4x)\cos(4t)$ .

#### Discrétisation spatiale et temporelle

Le problème (1.2.1) est discrétisé en espace par l'utilisation d'un schéma aux Différences Finies (DF) sur un maillage régulier  $\{x_i\}_{i=0}^{m+1}$  de  $\Omega$ , avec  $x_i = ih$  de pas  $h = \frac{1}{m+1}$  et n valeurs  $\{\xi_j\}_{j=1}^n$  du paramètre  $\xi$  choisies arbitrairement. Ainsi, on se ramène au système dynamique matriciel suivant

$$\frac{d}{dt}A(t) = (D_x \otimes M_\xi)A(t) + F(t), \qquad A(0) = 0,$$
(1.2.2)

où  $A(t) = [\boldsymbol{u}_h(t;\xi),\ldots,\boldsymbol{u}_h(t;\xi_m)]$  est une matrice de  $\mathbb{R}^{m \times n}$  dont les colonnes  $\boldsymbol{u}_h(t;\xi_i) \in \mathbb{R}^m$  correspondent à l'approximation DF (vecteur de  $\mathbb{R}^m$ ) de la solution pour une valeur  $\xi_i$  à l'instant t. Le vecteur  $F(t) \in \mathbb{R}^m$ est le vecteur des evaluations de  $f(t,\cdot)$  aux noeuds du maillage spatial i.e.  $F(t) = (f(t,x_1),\ldots,f(t,x_m))^T$ . L'opérateur  $D_x \otimes M_{\xi}$  est obtenu par tensorisation de  $D_x \in \mathbb{R}^{n \times n}$  l'opérateur aux différences finies centré d'ordre 2 de la dérivée de second ordre et  $M_{\xi} \in \mathbb{R}^{m \times m}$  une matrice diagonale dont les entrées sont les nvaleurs choisies de  $\xi$ . Ainsi on a  $(D_x \otimes M_{\xi})A(t) = [\xi_1 D_x \boldsymbol{u}_h(t;\xi_1),\ldots,\xi_m D_x \boldsymbol{u}_h(t;\xi_m)].$ 

Pour résoudre (1.2.2), on utilise un schéma d'intégration en temps. On veut construire une approximation  $\{A^k\}_{k=0}^K \subset \mathbb{R}^{m \times n}$  de A, sur le maillage  $\{t^k\}_{k=0}^K$  de (0, 1) t.q.  $t^k = k\Delta t$ . Cette approximation est obtenue par le biais du schéma d'Euler explicite suivant

$$A^{k+1} = \phi(A^k) := A^k + \Delta t (D_x \otimes M_{\xi} L) A^k + F(t^k), \quad k \ge 1$$

avec  $A^0 = A(0)$ . On rappelle que ce schéma est conditionnellement stable, en norme  $\infty$  ou 2, i.e. les pas d'espace et de temps doivent vérifier  $\Delta t \leq h^2/2\mu(\xi)$  pour tout  $\xi \in \Xi$ .

### Approximation de faible rang

Désormais on souhaite calculer une approximation de faible rang de A en tout instant  $t^k$  du maillage temporel. Pour cela on va chercher une approximation de faible rang  $\{A_r^k\}_{k=0}^K \subset \mathcal{M}_r(\mathbb{R}^{m \times n})$  définie par

$$A_r^{k+1} = \arg \min_{\substack{Z \in \mathbb{R}^{m \times n} \\ \operatorname{rang}(Z) \le r}} \|Z - \phi(A_r^k)\|, \ k \ge 0$$

avec  $A_r^0 = \arg \min_{\substack{Z \in \mathbb{R}^{m \times n} \\ \operatorname{rang}(Z) \leq r}} \|Z - A_r(0)\|$ . L'intérêt de cet algorithme par rapport au précédent c'est qu'on ne stocke qu'une matrice de faible rang O(r(m+n)) et non plus une matrice quelconque O(mn) coefficients.

#### **Résultats numériques**

Dans la suite nous présentons les résultats numériques pour n = m = 100. Le pas de temps est  $\Delta t = 10^{-4}$  (choisi assez petit pour assurer la stabilité du schéma explicite). Nous avons représenté la condition initiale et la solution numérique non "tronquée"  $A^k$  sur la Figure 1.2.2 pour l'instant  $t^k = 1$ .



FIGURE 1.2.2 – Condition initiale A(0) et solution numérique  $A^k$  pour  $t^k = 1$  (droite).

Nous avons ensuite calculé l'approximation de faible rang  $A_r^k$  pour  $r \in \{1, 5, 10\}$  de  $A^k$ . La solution pour  $t^k = 1$  est représentée sur la Figure 1.2.3. Nous observons qu'avec un rang 1 la solution  $A_r^k$  a globalement le même comportement que  $A^K$  mais présente des différences (valeurs max et min). A partir d'un rang r = 5,  $A_r^K$  et  $A^K$  sont en parfaite adéquation.



FIGURE 1.2.3 – Approximation de faible rang pour  $r \in \{1, 5, 10\}$  au temps t = 1.

## **1.2.3** Analyse en composantes principales

L'Analyse en Composantes Principales (ACP) est largement utilisé en statistiques pour l'exploration des données. Son but est de réduire la dimension des données multivariées (i.e. dépendant de plusieurs variables) en exprimant les variables originales en terme de nouvelles variables plus représentatives qui sont décorrélés. Ces derniers sont obtenus à partir de combinaisons linéaires des premiers.

#### Données

Supposons que nous avons pour données m observations indépendantes (échantillons)  $x_1, \ldots, x_m \in \mathbb{R}^n$  de n caractéristiques distinctes. Celles-ci sont stockées dans une matrice  $A \in \mathbb{R}^{m \times n}$  dont les lignes sont les échantillons centrés

$$A = \begin{bmatrix} (x_1 - \bar{x})^T \\ \vdots \\ (x_m - \bar{x})^T \end{bmatrix} = \begin{bmatrix} \tilde{x}_1^T \\ \vdots \\ \tilde{x}_m^T \end{bmatrix}$$

où la moyenne de l'échantillon est donnée par  $\bar{x} = \frac{1}{m} \sum_{i=1}^{m} x_i \in \mathbb{R}^n$ . On notera par la suite  $\tilde{x}_i = (x_i - \bar{x}), i = 1, \ldots, m$  les vecteurs centrés.

### Objectif

En pratique, n, m >> 1 ce qui signifie que A représente un large ensemble de données souvent corrélées. Une première idée pour mieux comprendre l'information contenue dans ces données serait d'utiliser une représentation 2D en nuages de points des caractéristiques. Ainsi, pour n caractéristiques, il faudrait  $C_2^n = \frac{n(n-1)}{2}$  graphiques pour représenter tous les échantillons pour toutes les paires possibles. Pour n = 10 cela signifie 45 graphiques différents. Ceci devient clairement inenvisageable quand n devient grand. Le but de l'ACP, est de déterminer un petit nombre de variables représentatives parmi celles observées, qui permettent d'expliquer au mieux la "variabilité".

L'objectif de l'ACP est de construire des combinaisons linéaires décorrélées des caractéristiques observées  $(x_i/\tilde{x}_i)$  qui représentent une certaine proportion de la variation dans l'échantillon appelées **composantes principales**. Ces composantes sont telles que : leur nombre r est inférieur au nombre n de variables originales (réduction de dimension), chaque compostante principale est choisie pour avoir la plus grande variance possible tout en étant décorrélée des précédentes.

### Formulation mathématique/Algorithme

Mathématiquement il s'agit de trouver le "plan" (sous-espace)  $\mathcal{V}_r \subset \mathbb{R}^n$  de dimension  $r \ll n$  dans lequel on cherche à représenter au mieux une combinaison linéaire des  $\tilde{x}_i, i = 1, \ldots, m$ . Pour cela, on s'intéresse au problème d'optimisation

$$\min_{\substack{\mathcal{V}_r \subset \mathbb{R}^n \\ \dim \mathcal{V}_r = r}} \sum_{i=1}^m \|\tilde{x}_i - P\tilde{x}_i\|_2^2, \tag{1.2.3}$$

où  $P \in \mathbb{R}^{n \times n}$  est une projection orthogonale  $(P^2 = P^T = P)$  de  $\mathbb{R}^n$  dans un sous-espace  $\mathcal{V}_r \subset \mathbb{R}^n$  de dimension r, pour le produit scalaire euclidien. On peut montrer

$$\sum_{i=1}^{m} \|\tilde{x}_i - P\tilde{x}_i\|_2^2 = \|(I_n - P)A^T\|_F^2 = \|A(I_n - P)\|_F^2$$

avec la norme de Frobenius  $\|\cdot\|_F$  (voir prochain chapitre). Ainsi le problème (1.2.4) est équivalent à trouver la matrice de projection P (de rang r i.e. dim im(P) = r)

$$\min_{\substack{P \in \mathbb{R}^{n \times n} \\ \operatorname{rang}(P) \le r}} \|A - AP\|_F,\tag{1.2.4}$$

En pratique, un minimiseur est  $P_r$  t.q.  $AP_r$  est la solution du problème d'approximation de faible rang

$$\min_{\substack{Z \in \mathbb{R}^{m \times n} \\ \operatorname{rang}(Z) \le r}} \|A - Z\|_F.$$
(1.2.5)

En effet rang $(AP_r) \leq r$  et par conséquent c'est un minimiseur si c'est l'approximation de faible rang de A définie par 1.2.5 (cf détails Chapitre 3). On verra même qu'un choix optimal est  $P := P_r = V_r V_r^T$  avec  $V_r = [v_1, \ldots, v_r] \in \mathbb{R}^{n \times r}$  la matrice dont les colonnes sont les r premiers vecteurs singuliers (orthonormaux) à droite de A. Il suffira d'effectuer une SVD tronquée à r termes de la matrice A pour obtenir  $V_r$ . Ainsi, l'espace vectoriel recherché est  $\mathcal{V}_r = \text{vect}\{v_1, \ldots, v_r\}$ .

Les r composantes principales  $\zeta_1(x), \ldots, \zeta_r(x)$  d'un vecteur  $x \in \mathbb{R}^n$  correspondent aux coordonnées de sa projection Px dans la base  $\{v_1, \ldots, v_r\}$  i.e.  $\zeta_k(x) = \langle v_k, x \rangle_2, \quad k = 1, \ldots, r.$ 

Remarque 1.2.1. Une autre interprétation de ce problème est

$$\min_{\substack{\mathcal{V}_r \subset \mathbb{R}^n \\ \dim \mathcal{V}_r = r}} \sum_{i=1}^m \|\tilde{x}_i - P\tilde{x}_i\|_2^2 \Leftrightarrow \min_{\substack{P \in \mathbb{R}^{n \times n} \\ \operatorname{rang}(P) \le r}} \|A - AP\|_F \Leftrightarrow \max_{\substack{P \in \mathbb{R}^{n \times n} \\ \operatorname{rang}(P) \le r}} \|AP\|_F \Leftrightarrow \max_{\substack{\mathcal{V}_r \subset \mathbb{R}^n \\ \dim \mathcal{V}_r = r}} \sum_{i=1}^m \|P\tilde{x}_i\|_2^2$$

En statistique, on appelle  $\sum_{i=1}^{m} \|P\tilde{x}_i\|_2$  l'inertie pour les "caractéristiques" projetées.

Les vecteurs  $v_1, \ldots, v_r$  correspondent aux vecteurs propres associés aux valeurs propres rangées par ordre décroissant de la matrice de covariance (empirique)  $C = A^T A \in \mathbb{R}^{n \times n}$ . Ainsi, il est aussi possible d'envisager un algorithme progressif pour déterminer les vecteurs  $v_1, \ldots, v_r$  en résolvant des problèmes aux valeurs propres pour C. L'idée de cet algorithme est donnée ci-après, pour plus de détails voir e.g. [5, §6].

1. Le vecteur  $v_1$  est donné par

$$v_1 = \arg \max_{v \in \mathbb{R}^n : \|v\|_2 = 1} \langle v, Cv \rangle_2 = \max_{\substack{\mathcal{V}_1 \subset \mathbb{R}^n \\ \dim \mathcal{V}_1 = 1}} \sum_{i=1}^m \|P\tilde{x}_i\|_2^2.$$

Ainsi le maximum correspond à la valeur propre  $\sigma_1^2(A) = \lambda_1(A^T A)$  et est atteint pour  $v_1$  premier vecteur propre (on retrouve le quotient de Rayleigh) de  $A^T A$ .

2. Les vecteurs  $v_k, k = 2, \ldots, r$  sont donnés par

$$v_k = \arg \max_{v \in \mathbb{R}^n : \|v\|_2 = 1} \langle v, C_{k-1}v \rangle_2$$

avec la matrice  $C_{k-1} = A_{k-1}^T A_{k-1}$  où  $A_{k-1} = A - \sum_{l=1}^{k-1} A v_l v_l^T$ . De nouveau le maximum correspond à  $\sigma_k^2(A) = \lambda_k(A)$  et est atteint pour  $v_k$  k-ième vecteur propre de  $A^T A$ .

#### Application numérique

On termine ce paragraphe en illustrant l'ACP pour deux jeux de données (voir Figure 1.2.4). On a construit, deux nuages de points en simulant n = 5000 vecteurs gaussiens en dimension 2, de moyenne  $\mu \in \mathbb{R}^2$  et matrice de covariance (s.d.p.)  $\Sigma \in \mathbb{R}^{2 \times n}$ .

- Cas test  $1: \mu = (1,3)^T$  et  $\Sigma = P\Lambda P^T$  avec  $\Lambda = \operatorname{diag}(var_1, var_2) = \operatorname{diag}(9,1)$  et  $P = \operatorname{rot}(\pi/6)$ . - Cas test  $3: \mu = (-1,2)^T$  et  $\Sigma = P\Lambda P^T$  avec  $\Lambda = \operatorname{diag}(100,1)$  et  $P = \operatorname{rot}(-\pi/4)$ .

On a représenté ces jeux de données (cercles bleus) et des vecteurs colinéaires à  $v_1, v_2$ .



FIGURE 1.2.4 – Illustration de l'ACP (données, vecteurs  $var_1v_1, var_2v_2$ ) sur deux ensembles de données. Cas test 1 (gauche) et Cas test 2 (droit).

On peut donc interpréter le jeu de données  $x_1, \ldots, x_m$  dans ce nouveau repère  $(v_1, v_2)$  et les composantes principales de  $x_i$  sont les  $x_i^T v_1, x_i^T v_2$ . Dans le premier cas, ces composantes ont des contributions similaires en effet les données sont réparties de façon homogène dans les deux directions  $v_1, v_2$ . Pour le second cas ce sont les données dans la direction de  $v_1$  qui sont prépondérentes. Ici on aurait en effet pu intuiter ce comportement car pour le Cas test 1  $var_1 \approx var_2$  alors que  $var_1 >> var_2$  pour Cas test 2. En terme de valeurs singulières, cela se traduit par  $\sigma_1 \approx \sigma_2$  pour le premier cas, et  $\sigma_1 >> \sigma_2$ . En pratique, cela permet d'envisager de réduire les données, en les représentant uniquement dans les espaces où les composantes principales sont prépondérantes. Par exemple pour le Cas Test 2, on pourrait juste consider une représentation de ces données dans l'espace engendré par  $v_1$ .

# Chapitre 2

## Compléments d'analyse matricielle

Pour commencer nous revenons sur la notion de rang d'une matrice, et ses propriétés, qui sera au coeur des problèmes que nous allons aborder par la suite. Nous rappelons également des notions importantes sur les normes vectorielles et matricielles. Par la suite, nous présentons des outils pour calculer des factorisations de matrices qui seront cruciaux pour construire les approximations de faible rang considérées ensuite. Certains résultats énoncés ci-après sont en partie des résultats de prépa que nous ne redémontrerons pas. Nous renvoyons à  $[3, \S0.4, \S5]$  ou encore [5] par exemple.

**\blacksquare** Dans la suite on pourra utiliser la notation Matlab pour les matrices de taille  $m \times n$ . Notamment, les

conséquent  $A = [a_{:1}, \dots, a_{:n}]$  ou encore  $A = \begin{bmatrix} a_{1:} \\ \vdots \\ a_{m:} \end{bmatrix}$ . On pourra considérer une matrice extraite de taille  $p \times q$ avec  $p \le m, q \le n$  sous la forme A(1:n, 1:q)

#### 2.1 Rang d'une matrice

Soient  $n, m \in \mathbb{N}^*$ . On note  $\mathcal{M}_{m,n}(\mathbb{R}) := \mathbb{R}^{m \times n}$  l'ensemble des matrices de taille  $m \times n$  à coefficients réels.

On appelle **noyau** de A l'ensemble

$$\ker(A) = \{x \in \mathbb{R}^n : Ax = 0\} \subset \mathbb{R}^n$$

et **image** de A l'ensemble

$$\operatorname{im}(A) = \{ y = Ax : x \in \mathbb{R}^n \} \subset \mathbb{R}^m$$

Le rang de A est la dimension de l'espace engendré par les colonnes de A ainsi

$$\operatorname{rang}(A) = \dim \operatorname{im}(A) \le \min(m, n).$$

On rappelle que le théorème du rang assure

$$\dim \ker(A) + \dim \operatorname{im}(A) = \dim \mathbb{R}^n \Leftrightarrow \dim \ker(A) + \operatorname{rang} A = n.$$

Dans le cas particulier où m = n, i.e. A est une matrice carré d'ordre n, alors ker $(A) = \{0\} \Leftrightarrow rang A = n$ , donc A est inversible.

On peut montrer les inégalités concernant les rangs d'une matrice.

- 1. Soit  $A \in \mathbb{R}^{m \times n}$  alors  $\operatorname{rang}(A) \leq \min(m, n)$ .
- 2. Toute sous-matrice obtenue par suppression d'une ligne ou une colonne ne peut pas être de rang supérieur à la matrice initiale.
- 3. (Inégalité de Sylester) Soient  $A \in \mathbb{R}^{m \times k}$  et  $B \in \mathbb{R}^{k \times n}$  alors

 $\operatorname{rang}(A) + \operatorname{rang}(B) - k \le \operatorname{rang}(AB) \le \min(\operatorname{rang}(A), \operatorname{rang}(B)).$ 

4. Soient  $A, B \in \mathbb{R}^{m \times n}$  alors  $\operatorname{rang}(A + B) \leq \operatorname{rang}(A) + \operatorname{rang}(B)$ .

Désormais on énonce quelques égalités.

- 1. Soit  $A \in \mathbb{R}^{m \times n}$  alors  $\operatorname{rang}(A) = \operatorname{rang}(A^T) = \operatorname{rang}(A^T A) = \operatorname{rang}(AA^T)$ . Ainsi le rang est la dimension de l'espace engendré par les lignes ou les colonnes de A.
- 2. Soient  $P \in \mathbb{R}^{m \times m}$ ,  $Q \in \mathbb{R}^{n \times n}$  deux matrices inversibles et  $A \in \mathbb{R}^{m \times n}$  alors on a

 $\operatorname{rang}(A) = \operatorname{rang}(PAQ) = \operatorname{rang}(PA) = \operatorname{rang}(AQ).$ 

On prendra par suite la convention de noter  $r \leq \min(n, m)$  le rang d'une matrice A et on notera

$$\mathcal{M}_r(\mathbb{R}^{m \times n}) = \{ Z \in \mathbb{R}^{m \times n} : \operatorname{rang}(Z) \le r \},\$$

l'ensemble des matrices de rang au plus r.

## 2.2 Normes

Soit E un espace vectoriel défini sur  $\mathbb{R}$  (e.g.,  $E = \mathbb{R}^n$  ou  $E = \mathbb{R}^{m \times n}$ ). Nous rappelons les définitions suivantes. Définition 2.2.1 (Norme). Une norme sur E est une application  $\|\cdot\|$  de E dans  $\mathbb{R}$  t.q.

- *i*)  $||x|| \ge 0, u \in E$ ,
- $ii) ||x|| = 0 \Leftrightarrow u = 0,$
- *iii)*  $\|\lambda x\| = |\lambda| \|x\|, x \in E, \lambda \in \mathbb{R}.$
- *iv)*  $||x + y|| \le ||x|| + ||y||, x, y \in E.$

Ainsi  $(E, \|\cdot\|)$  définit espace vectoriel normé.

**Proposition 2.2.2.** Si  $\|\cdot\|$  est une norme sur E alors on a  $||x|| - ||y||| \le ||x - y||$  pour tout  $x, y \in E$ .

Démonstration. Conséquence directe de l'inégalité triangulaire.

**Définition 2.2.3** (Produit scalaire). Un produit scalaire sur E est une application  $\langle \cdot, \cdot \rangle$  de  $E \times E$  dans  $\mathbb{R}$  t.q.

- i)  $\langle x, y \rangle = \langle y, x \rangle, x, y \in E$ ,
- *ii)*  $\langle x, x \rangle \ge 0, u \in E$ ,
- *iii)*  $\langle x, x \rangle = 0 \Rightarrow x = 0$ ,
- *iv*)  $\langle \lambda x + \mu y, z \rangle = \lambda \langle x, z \rangle + \mu \langle y, z \rangle, \lambda, \mu \in \mathbb{R}, x, y, z \in E.$

**Définition 2.2.4** (Orthogonalité). On dit que  $x, y \in E$  sont orthogonaux pour  $\langle \cdot, \cdot \rangle$  si  $\langle x, y \rangle = 0$ . On dit que x est orthogonal à  $F \subset E$  si pour tout  $y \in F$  on a  $\langle x, y \rangle = 0$ .

**Proposition 2.2.5** (Inégalité de Cauchy-Schwarz). Pour tout  $x, y \in \mathbb{R}^n$ , on a  $|\langle x, y \rangle| \leq \langle x, x \rangle \langle y, y \rangle$ .

Démonstration. Soient  $x, y \in \mathbb{R}^n \setminus \{0\}$ . Sinon le résultat est trivial. Introduisons  $\lambda = \frac{\langle x, y \rangle}{\langle y, y \rangle}$ . Le vecteur  $\lambda y$  est la projection orthogonale de x sur y. Les vecteurs  $x - \lambda y$  and  $\lambda y$  sont orthogonaux. Il vient  $\langle x, x \rangle = \langle (x - \lambda y) + \lambda y, (x - \lambda y) + \lambda y \rangle = \langle x - \lambda y, x - \lambda y \rangle + \lambda^2 \langle y, y \rangle \ge \lambda^2 \langle y, y \rangle = \langle x, y \rangle / \langle y, y \rangle$ . D'où le résultat. On obtient la valeur absolue en prenant x ou y.

**Corollaire 2.2.6.** Soit  $\langle \cdot, \cdot \rangle$  un produit scalaire défini sur  $E \times E$  alors l'application de  $E \to \mathbb{R}_+$  par  $x \mapsto ||x|| = \langle x, x \rangle^{1/2}$  est une norme.

Démonstration. Il est facile de vérifier toutes les propriétés de la norme. Pour l'inégalité triangulaire on voit facilement que  $||x + y||^2 \le (||x|| + ||y||)^2$  par Cauchy-Schwarz.

## **2.2.1** Exemples importants dans le cas $E = \mathbb{R}^n$

Définissons pour tout  $x, y \in \mathbb{R}^n$ , le produit scalaire euclidien

$$\langle x, y \rangle_2 = u^T v = \sum_{i=1}^n x_i y_i,$$

et sa norme associée

$$||x||_2 = \sqrt{\langle x, x \rangle_2} = \left(\sum_{i=1}^n x_i^2\right)^{1/2}$$

On peut également définie une norme définie par le biais d'une matrice de rang plein. **Proposition 2.2.7.** Soient  $m \leq n$  et  $M \in \mathbb{R}^{m \times n}$  t.q. rang(M) = n. L'application  $\|\cdot\|_M$  définie sur  $\mathbb{R}^n$  par

$$||x||_M = ||Mx||_2 \quad x \in \mathbb{R}^n$$

est une norme. On peut d'ailleurs lui associer le produit scalaire  $\langle \cdot, \cdot \rangle_M$  défini par

$$\langle x, y \rangle_M = \sqrt{\langle Mx, Mx \rangle_2} = \sqrt{\langle M^T Mx, x \rangle_2} \quad x, y \in \mathbb{R}^n.$$

Démonstration. Il suffit d'observer que  $M^T M$  est symétrie définie positive (car de rang n).

En fait, on remarquera que la norme ainsi définie est une norme équivalente à la norme euclidienne en effet  $\sqrt{\lambda_{min}(M^T M)} \|x\|_2 \le \|x\|_M \le \sqrt{\lambda_{max}(M^T M)} \|x\|_2.$ 

## 2.2.2 Exemples importants dans le cas $E = \mathbb{R}^{m \times n}$

**Définition 2.2.8** (Norme matricielle). Une norme matricielle sur  $\mathbb{R}^{m \times n}$  est une norme  $\|\cdot\|$  (au sens de la définition 2.2.1) vérifiant pour tout  $A \in \mathbb{R}^{m \times k}$ ,  $B \in \mathbb{R}^{k \times n}$  l'inégalité suivante

$$||AB|| \le ||A|| ||B||.$$

La propriété précédente est appelée sous-multiplicativité que nous n'avons pas pour les normes vectorielles. **Proposition 2.2.9.** Soit  $\|\cdot\|$  une norme vectorielle sur  $\mathbb{R}^n$ . Etant donnée  $A \in \mathbb{R}^{n \times n}$ , l'application  $A \mapsto |||A|||$ avec

$$|||A||| = \sup_{x \in \mathbb{R}^n \setminus \{0\}} \frac{||Ax||}{||x||} = \sup_{||x||=1} ||Ax||.$$

est une norme matricielle

Démonstration. On verifie assez facilement tous les points de la définition 2.2.1 qui découlent des normes vectorielles. Soient  $A \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{n \times n}$ . Pour la sous-multiplicativité pour tout  $x \neq 0$  t.q.  $Bx \neq 0$  (sinon le résultat est direct) on a

$$\frac{\|ABx\|}{\|x\|} \le \frac{\|A(Bx)\|}{\|Bx\|} \frac{\|Bx\|}{\|x\|} \le |||A||||||B|||.$$

On obtient le résultat en prenant le sup et en observant que  $im(B) \subset \mathbb{R}^n$ .

**Exemple 2.2.10.** Soit  $A \in \mathbb{R}^{n \times n}$ , considérons la norme matricielle subordonnée à la norme euclidienne ou norme spectrale dans  $\mathbb{R}^n$  *i.e.* 

$$\|A\|_{2} := \sup_{x \in \mathbb{R}^{n} \setminus \{0\}} \frac{\|Ax\|_{2}}{\|x\|_{2}} = \sup_{\|x\|_{2}=1} \|Ax\|_{2} = \sqrt{\rho(A^{T}A)} = \sup_{x \in \mathbb{R}^{n} \setminus \{0\}} \frac{\langle A^{T}Ax, x \rangle_{2}}{\|x\|_{2}}.$$

où  $\rho(A^T A)$  est le rayon spectral de  $A^T A$  (i.e. la plus grande valeur propre en module). La dernière égalité fait le lien avec le **quotient de Rayleigh**.

Un autre exemple de norme que nous manipulerons est la norme de **Frobenius**. Pour tout  $A \in \mathbb{R}^{m \times n}$  définissons

$$||A||_F = \sqrt{\text{trace}(A^T A)} = \sqrt{\sum_{i=1}^m \sum_{j=1}^n a_{ij}^2} = \sqrt{\sum_{i=1}^n \lambda_i(A^T A)}$$

Alors  $\|\cdot\|_F$  est une norme matricielle associée du produit scalaire

$$\langle A, B \rangle_F = \operatorname{trace}(A^T B), \quad \forall A, B \in \mathbb{R}^{m \times n}.$$

Démonstration. -

- Il est assez facile de voir que  $\|\cdot\|_F$  satisfait les points i) à iii) de la définition 2.2.1. Il suffit de verifier que  $\langle \cdot, \cdot \rangle_F$  est un produit scalaire et de conclure avec CS pour l'inégalité triangulaire.
- Il reste à montrer la sous-multiplicativité. Soient  $A \in \mathbb{R}^{m \times k}, B \in \mathbb{R}^{k \times n}$  alors

$$||AB||_F^2 = \operatorname{trace}((AB)^T AB).$$

Notons C = AB alors

$$\operatorname{trace}(C^{T}C) = \sum_{i=1}^{n} \sum_{i=1}^{m} c_{ij}^{2} = \sum_{i=1}^{n} \sum_{i=1}^{m} \left[ \sum_{l=1}^{k} a_{il} b_{lj} \right]^{2}$$

mais le dernier terme au carré est égal à  $[a_{i:}^T b_{:j}]^2 \leq ||a_{i:}||_2^2 ||b_{:j}||_2^2$  par C-S. Donc on a

$$\operatorname{trace}(C^{T}C) \leq \sum_{j=1}^{n} \sum_{i=1}^{m} \|a_{i:}\|_{2}^{2} \|b_{:j}\|_{2}^{2} \leq \sum_{j=1}^{n} \sum_{i=1}^{m} \left[\sum_{l=1}^{k} a_{il}^{2}\right] \left[\sum_{l=1}^{k} b_{lj}^{2}\right] = \left(\sum_{i=1}^{m} \sum_{l=1}^{k} a_{il}^{2}\right) \left(\sum_{j=1}^{n} \sum_{l=1}^{k} b_{lj}^{2}\right).$$

on a  $||AB||_F \le ||A||_F ||B||_F$ .

On termine avec quelques autres propriétés pratiques de la norme de Frobenius (les preuves sont laissées en exercice)

1. La norme de Frobenius vérifie l'invariance par rotation. Et ant données des matrices orthogonales  $U \in \mathbb{R}^{m \times m}, V \in \mathbb{R}^{n \times n}$ , on a

$$||A||_F = ||UA||_F = ||AV||_F = ||UAV||_F.$$

Démonstration. Par propriété de la trace (trace(XYZ) = trace(YZX) = trace(ZXY)) il vient

$$||UAV||_F = \operatorname{trace}(V^T A^T U^T U A V) = \operatorname{trace}(V^T A^T A V) = \operatorname{trace}(V V^T A^T A) = \operatorname{trace}(A^T A)$$

car U, V unitaires.

2. On a 
$$||A^T||_F = ||A||_F$$
 car trace $(A^T A) =$ trace $(AA^T)$ 

## 2.3 Factorisation de matrices

Dans ce paragraphe nous discutons de l'existence de factorisation de matrices éventuellement de rang r. Ces factorisations seront le point de départ pour développer des méthodes d'approximation dites de faible rang.

#### 2.3.1 Factorisation de matrices rectangulaires

Nous donnons ici le principe et la définition d'une factorisation QR d'une matrice de taille  $m \times n$ .

**Proposition 2.3.1.** *Q* Soit  $A \in \mathbb{R}^{m \times n}$ , avec  $m \ge n$ , une matrice de rang n. Alors, il existe une unique factorisation de A sous la forme

$$A = QR = \underbrace{\begin{bmatrix} q_{11} & \dots & q_{n1} \\ \vdots & & \vdots \\ q_{m1} & \dots & q_{mn} \end{bmatrix}}_{\in \mathbb{R}^{m \times n}} \underbrace{\begin{bmatrix} r_{11} & \dots & r_{n1} \\ & \ddots & \vdots \\ 0 & & r_{nn} \end{bmatrix}}_{\in \mathbb{R}^{n \times n}}.$$

De plus les vecteurs colonnes de Q forment une famille orthonormale  $Q^T Q = I_n$ , et R est triangulaire supérieure qui coïncide avec la matrice de factorisation de Cholesky de la matrice  $A^T A \in \mathbb{R}^{n \times n}$  (s.d.p.) i.e.  $R^T R = A^T A$ .

Une telle factorisation peut être obtenue par un procédé d'orthonormalisation à la Gram-Schmidt. Les détails des algorithmes peuvent être consultés dans [5, Chapitre 3.4.4].

**Remarque 2.3.2.** Si rang(A) = n, les colonnes de Q forment une base orthonormale pour im(A), ainsi la factorisation QR est une façon simple de déterminer cette base.

 $\langle \rangle$  Il est possible de calculer la factorisation QR d'une matrice A dans Python/Matlab directement avec les lignes de commandes suivantes.

```
    % MATLAB : produces the matrices of QR decomposition of A
    [Q,R] = qr(A)
    # Python : produces the matrices of QR decomposition of A
```

2 Q,R = np.linalg.qr(A)

## **2.3.2** Factorisation des matrices de rang r

On commence avec un cas idéal. Il s'agit de trouver une factorisation pour une matrice de rang r. D'un point de vue pratique, ceci peut être très interessant en terme de stockage dès lors que  $r \ll m, n$ . En effet, on verra qu'au lieu de stocker mn coefficients il suffit de se contenter de (m + n)r coefficients!

**Lemme 2.3.3.** Soit  $A \in \mathbb{R}^{m \times n}$  telle que rang $(A) \leq r$ . Alors A admet une factorisation (pas nécessairement unique) de la forme  $A = BC^T$  où  $B \in \mathbb{R}^{m \times r}$  et  $C \in \mathbb{R}^{n \times r}$ .

Démonstration. La matrice A est de rang r i.e. l'espace engendré par ses colonnes de dimension r. Autrement dit on peut trouver une famille libre de vecteurs  $\{b_1, \ldots, b_r\}$  de  $\mathbb{R}^m$  génératrice de cet espace. En particulier toutes les colonnes de a notées  $a_{:j}, j = 1, \ldots, n$  peuvent s'écrire

$$a_{:j} = \sum_{i=1}^{r} c_{ji} b_i = [b_1, \dots, b_r] \begin{bmatrix} c_{j1} \\ \vdots \\ c_{jr} \end{bmatrix}, \quad \forall j = 1, \dots, n.$$

Par concaténation des j colonnes de A, et des r vecteurs  $b_i$  il vient

$$A = [a_{:1}, \dots, a_{:n}] = \underbrace{[b_1, \dots, b_r]}_{B \in \mathbb{R}^{m \times r}} \underbrace{\begin{bmatrix} c_{11} & \dots & c_{n1} \\ \vdots & & \\ c_{1r} & \dots & c_{nr} \end{bmatrix}}_{C^T \in \mathbb{R}^{r \times n}}.$$

Bien souvent dans les applications, A n'est pas de rang r, mais de rang plein i.e.  $r = \min(m, n)$  ainsi, utiliser une "troncature" de faible rang peut s'avérer plus pertinent tout en gardant "suffisament d'information". Pour cela nous allons faire le lien avec sa décomposition en valeurs singulières (ou valeurs propres quand Acarré symétrique).

## 2.3.3 Décomposition en valeurs singulières

On énonce tout d'abord le théorème d'existence de décomposition en valeurs singulières [3, Theorem 2.6.3]. **Théorème 2.3.4** (Décomposition en valeurs singulières). Soit  $A \in \mathbb{R}^{m \times n}$  une matrice donnée. On note  $q = \min(m, n)$  et  $r = \operatorname{rang}(A)$ .

1. Il existe deux matrices orthogonales  $U \in \mathbb{R}^{m \times m}, V \in \mathbb{R}^{n \times n}$  et une matrice diagonale  $\Sigma_q \in \mathbb{R}^{q \times q}$  t.q.

$$\Sigma_q = \begin{pmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_q \end{pmatrix}$$

avec  $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0 = \sigma_{r+1} = \cdots = \sigma_q$  tels que

$$A = U\Sigma V^T.$$

avec la matrice  $\Sigma$  définie par

$$\begin{split} \Sigma &= & \Sigma_q & si \ m = n, \\ \Sigma &= & \left[ \Sigma_q & 0 \right] \in \mathbb{R}^{m \times n} & si \ n > m, \\ \Sigma &= & \left[ \Sigma_q \\ 0 \right] \in \mathbb{R}^{m \times n} & si \ n < m. \end{split}$$

2. Les réels  $\sigma_1, \ldots, \sigma_r$  sont appelés valeurs singulières de la matrice A. Elles correspondent à la racine carrée des valeurs propres rangées par ordre décroissant de  $AA^T$  ou de  $A^TA$ .

Les  $u_1, \ldots, u_m$  sont les vecteurs singulier à gauche alors que les  $v_1, \ldots, v_n$  sont les vecteurs singuliers à droite. **Remarque 2.3.5.** Le rang de la matrice A est égal on nombre de valeurs singulières non nulles de A (ou de valeurs propres non nulles de  $A^T A$ ).

#### Preuve de l'existence de la SVD

Démonstration. Preuve inspirée de [1] établie pour le cas général des matrices complexes.

1. (cas n = m) Par le théorème spectral, comme la matrice  $A^T A$  est symétrique réelle et ses vp. sont réelles positives, il existe  $V \in \mathbb{R}^{n \times n}$  orthogonale (réelle) t.q.  $A^T A = V^T \Lambda V$  avec  $\Lambda \in \mathbb{R}^{n \times n}$  matrice diagonale réelle. Ainsi

$$(AV)^T AV = V^T A^T AV = V^T V \Lambda V^T V = \Lambda.$$

On rappelle que rang $(A^T A)$  = rang(A) = r alors les v.p. de  $A^T A$  sont t.q.  $|\lambda_1| \ge \cdots \ge |\lambda_r| > 0 = \lambda_{r+1} = \cdots = \lambda_n$  (à un arrangement près par ordre décroissant des valeurs absolues). On déduit également de  $(AV)^T AV = \Lambda$  que AV a ses colonnes orthogonales. Ainsi, il existe  $U \in \mathbb{R}^{n \times n}$  orthogonale t.q.  $AV = U\Lambda^{1/2}$ . Ainsi, on peut réecrire  $A = U\Lambda^{1/2}V^T = U\Sigma_q V^T$  avec  $\Sigma_q = \Lambda^{1/2}$ .

2. (cas n > m) Tout d'abord on a  $r \le m$  ainsi dim ker $(A) = n - r \ge n - m$ . Construisons une famille orthonormée de vecteurs  $\{b_1, \ldots, b_{n-m}\} \subset \ker(A) \subset \mathbb{R}^n$ . On forme ainsi la matrice  $B_2 = [b_1, \ldots, b_{n-m}] \in \mathbb{R}^{n \times (n-m)}$ . On complète cette famille (théorème de la base incomplète) pour former une base orthonormée de  $\mathbb{R}^n$  avec des vecteurs formant les colonnes d'une matrice  $B \in \mathbb{R}^{n \times n}$  donnée par  $B = \begin{bmatrix} B_1 & B_2 \end{bmatrix}$ . Alors par définition  $AB = \begin{bmatrix} AB_1 & 0 \end{bmatrix} \in \mathbb{R}^{m \times n}$  car les colonnes de  $B_2$  sont dans ker(A). Or  $AB_1 \in \mathbb{R}^{m \times m}$ , on peut appliquer le cas du 1., et il existe des matrices  $\hat{U}, \hat{V}$  orthogonales de  $\mathbb{R}^{m \times m}$  et  $\Sigma_{q=m}$  diagonale de taille  $m \times m$  t.q.  $AB_1 = \hat{U}\Sigma\hat{V}^T$ .

Mais on a aussi en utilisant  ${\cal B}$  est orthogonale

$$A = ABB^{T} = \begin{bmatrix} AB_{1} & 0 \end{bmatrix} B^{T} = \begin{bmatrix} \hat{U}\Sigma_{q}\hat{V}^{T} & 0 \end{bmatrix} B^{T} = \underbrace{\hat{U}}_{=U} \underbrace{\begin{bmatrix} \Sigma_{q} & 0 \end{bmatrix}}_{\Sigma} \underbrace{\begin{bmatrix} \hat{V}^{T} & 0 \\ 0 & I_{m-n} \end{bmatrix} B^{T}}_{V^{T}}.$$

3. (cas n < m) Recommencer pour  $A^T$  (cf. TD).

- 4. On observe que  $A = U\Sigma V^T$  et  $A^T = V\Sigma^T U$  t.q.  $AA^T = V\Sigma^T \Sigma V^T$  avec  $\Sigma^T \Sigma = \Sigma_q^2$  si  $n \ge m$  sinon  $\Sigma^T \Sigma = \begin{bmatrix} \Sigma_q^2 & 0 \\ 0 & 0 \end{bmatrix}$ . Donc les valeurs singulières de A sont bien les racines carrées des vp. non nulles de  $AA^T$  i.e.  $\sigma_i(A) = \lambda_i(AA^T)^{1/2}, i = 1, \dots, r$ . Procéder de même pour  $A^T A$ .
- Si  $m \ge n$  seulement les n premières colonnes de U sont calculées et  $\Sigma$  est d'ordre n. C'est ce que l'on appelle **thin SVD**.
- Si  $m \leq n$  seulement les m premières colonnes de V sont calculées et  $\Sigma$  est d'ordre m.

La SVD permet notamment de définir la notion de **pseudo-inverse ou inverse généralisée** pour des matrices apriori non inversibles, ce qui peut s'avérer utile en pratique par ex. pour des problèmes de moindres carrés.

**Définition 2.3.6** (Pseudo-inverse ou inverse généralisée). On reprend les notations du théorème 2.3.4. Soit  $A \in \mathbb{R}^{m \times n}$  de rang r, et  $A = U\Sigma V^T$  sa SVD. La matrice  $A^+ = V\Sigma^+ U^T$  est appelée matrice pseudo-inverse de Moore-Penrose où

$$\begin{split} \Sigma^+ &= \Sigma^+_q & si \ m = n, \\ \Sigma^+ &= \begin{bmatrix} \Sigma^+_q & 0 \end{bmatrix} & si \ n > m, \\ \Sigma^+ &= \begin{bmatrix} \Sigma^+_q \\ 0 \end{bmatrix} & si \ n < m. \end{split}$$

avec  $\Sigma_q^+ = \operatorname{diag}(\frac{1}{\sigma_1}, \dots, \frac{1}{\sigma_r}, \underbrace{0, \dots, 0}_{q-r}).$ 

En particulier, si rang(A) = n < m alors  $A^+ = (A^T A)^{-1} A^T$  tandis que si n = m = rang(A) alors  $A^+ = A^{-1}$ .

## Quelques propriétés

**Corollaire 2.3.7.** Soit  $A \in \mathbb{R}^{m \times n}$ ,  $m \ge n$  et  $A = U\Sigma V^T$  so SVD. Alors pour tout i = 1, ..., n on a

$$\begin{cases} Av_i &= \sigma_i u_i, \\ A^T u_i &= \sigma_i v_i, \end{cases} \Rightarrow \begin{cases} A^T Av_i &= \sigma_i^2 v_i, \\ AA^T u_i &= \sigma_i^2 u_i. \end{cases}$$

Démonstration. Utiliser  $AV = U\Sigma$  et  $A^T U = V\Sigma$ .

Ce corollaire confirme ce qu'on avait pressenti dans la preuve, i.e. les vecteurs singuliers à droite sont les vecteurs propres de  $AA^T$  et les vecteurs singuliers à gauche les vecteurs propres de  $A^TA$  associés aux vp.  $\sigma_i^2$ .

On peut également donner une interprétation géométrique, i.e. les valeurs singulières de A sont les longueurs des demi-axes de l'hyperellipsoide définie par  $E = \{Ax : ||x||_2 = 1\}$ . Les directions des demi-axes sont données par les  $u_i$ .

TABLEAU 2.3.1 – Interprétation géométrique de la SVD :  $x \to V^T x \to \Sigma V^T x \to U \Sigma V^T x$ , pour x t.q.  $||x||_2 = 1$ .

La SVD d'une matrice A permet de faire un lien avec ses normes spectrale et de Frobenius.



**Corollaire 2.3.8.** Soit  $A \in \mathbb{R}^{m \times n}$ , de rang r alors

$$||A||_2 := \sigma_1 \ et \ ||A||_F := \sqrt{\sigma_1^2 + \dots + \sigma_r^2},$$

par conséquent

$$||A||_2 \le ||A||_F \le \sqrt{r} ||A||_F.$$

Démonstration. Par définition  $||A||_2^2 := \sup_{||x||_2=1} ||Ax||_2^2 = \sup_{||x||_2=1} \langle A^T A x \rangle_2 = \lambda_1(A^T A) = \sigma_1^2$ . Pour la norme de Frobenius  $||A||_F^2 = ||\Sigma||_F^2$  car U, V sont orthogonale donc  $||A||_F^2 = \sum_{i=1}^r \sigma_i^2$ . Pour montrer le second point on observe

$$||A||_{2}^{2} \leq ||A||_{F}^{2} = \sum_{i=1}^{r} \sigma_{i}^{2} \leq r\sigma_{1}^{2} = r||A||_{2}^{2}$$

D'où le résultat.

La SVD permet également de définir des bases orthonormées pour les espaces im(A) et ker(A). Corollaire 2.3.9. Soit  $A \in \mathbb{R}^{m \times n}$ , de rang r alors

$$im(A) = vect\{u_1, ..., u_r\} et ker(A) = vect\{v_{r+1}, ..., v_n\}$$

On termine avec l'interprétation d'un matrice de rang r comme un tenseur d'ordre 2. Cela signifie que les concepts vu pour les matrices s'inscrivent dans un cadre plus général : celui de l'approximation d'application multi-linéaires représentés par des tenseur d'ordre élevés par des tenseurs de faible rang.

**Corollaire 2.3.10.** Soit  $A \in \mathbb{R}^{m \times n}$ , de rang r alors

$$A = \sum_{i=1}^{r} \sigma_{i} u_{i} v_{i}^{T} := \sum_{\substack{i=1\\tenseur\ d'ordre\ 2}}^{r} \sigma_{i} u_{i} \otimes v_{i}$$

où  $u_i, v_i$  sont les i-èmes colonnes de U, V respectivement.

Démonstration. Il s'agit de réecrire la SVD sous la forme suivante

$$A = U\Sigma V^T = \begin{bmatrix} \sigma_1 u_1, \dots, \sigma_r u_r, 0, \dots, 0 \end{bmatrix} \begin{bmatrix} v_1^T \\ \vdots \\ v_n^T \end{bmatrix} = \sum_{i=1}^r \sigma_i u_i v_i^T$$

#### Implémentation de la SVD sur machine

Il est possible de calculer la SVD dans Python/Matlab directement avec les lignes de commandes suivantes.

```
1 % MATLAB : produces an economy-size SVD decomposition of A
2 [U,S,V] = svd(A,'econ')
1 # Python : produces the SVD decomposition of A
```

2 U, S, VT = np.linalg.svd(A)

On vient de voir comment calculer la SVD (compacte) i.e. qui renvoie  $U \in \mathbb{R}^{m \times k}, \Sigma \in \mathbb{R}^{k \times k}, V \in \mathbb{R}^{n \times k}$ d'une matrice  $A = U\Sigma V^T \in \mathbb{R}^{m \times n}$  de rang k. En utilisant les commandes svd() ou bien np.linalg.svd() de Matlab et Python respectivement.

Ici, nous expliquons ce qui se cache derrière ces commandes. En fait, il s'agit de résoudre un problème aux valeurs propres (cf. Corollaire 2.3.7). En pratique il est possible d'utiliser eig() ou encore np.linalg.eig()

pour résoudre ces problèmes. Nous décrivons la procédure dans l'algorithme 1 qui suit. Pouvoir en savoir plus sur des variantes de ces algorithmes voir [1, §6]

Algorithm 1	l Calcul	de la SV	'D (com	pacte) d	'une matrice $A$
-------------	----------	----------	---------	----------	------------------

-	,	
<b>Require:</b> $A \in \mathbb{R}^{m \times n}$ .		
• $\lambda, V \leftarrow \operatorname{eig}(A^T A).$	$\triangleright$ Calcul	des valeurs et vecteurs propres de $A^T A$
• $\sigma \leftarrow \sqrt{\lambda}, \sigma \leftarrow \texttt{sort}(\sigma).$	▷ Calcul des valeurs singulières e	ensuite ordonnées par ordre décroissant
• $V \leftarrow \texttt{sort}(V)$ .	$\triangleright$ Tri des vecteur	rs singuliers dans l'ordre correspondant
• $k = \operatorname{count}(\sigma \neq 0).$		$\triangleright \text{ Calcul du rang de } A$
return $V \leftarrow V(:, 1:k), \Sigma \leftarrow$	$-\operatorname{diag}(\sigma(1:k)), U \leftarrow AV\Sigma^{-1}$	$\triangleright$ Renvoi de $U, \Sigma, V$ .

Il est aussi possible d'envisager de calculer les valeurs propres de  $A^T A$  et d'utiliser les vecteurs propres associés correspondant aux vecteurs singuliers à gauche, i.e. les colonnes de U.

# Chapitre 3

Approximation de faible rang - méthodes déterministes et aléatoires

Soient  $n, m \in \mathbb{N}^*$  et  $A \in \mathbb{R}^{m \times n}$ . On rappelle le problème d'approximation de faible rang donné dans le Chapitre 1

$$\min_{\substack{Z \in \mathbb{R}^{m \times n} \\ \operatorname{rang}(Z) \le r}} \|A - Z\|, \tag{3.0.1}$$

où  $\|\cdot\|$  peut être la norme spectrale ou la norme de Frobenius

## 3.1 Théorie

**Théorème 3.1.1** (Eckart-Young-Mirsky). Soit  $A \in \mathbb{R}^{m \times n}$  une matrice de rang k admettant la SVD

$$A = \sum_{i=1}^{k} \sigma_i u_i v_i^T,$$

alors la matrice de rang r correspondant à la SVD tronquée aux r-premiers termes, notée  $T_r(A)$  satisfait la condition d'optimalité suivante

$$||A - \Im_r(A)||_2 = \min_{\substack{Z \in \mathbb{R}^{m \times n} \\ \operatorname{rang}(Z) \le r}} ||A - Z||_2 = \sigma_{r+1}.$$

de plus

$$||A - \mathfrak{I}_r(A)||_F = \min_{\substack{Z \in \mathbb{R}^{m \times n} \\ \operatorname{rang}(Z) \le r}} ||A - Z||_F = \left(\sum_{i=r+1}^k \sigma_i^2\right)^{1/2}.$$

En clair, la matrice  $\mathfrak{T}_r(A)$  s'écrit

$$\mathfrak{T}_r(A) := \sum_{i=1}^r \sigma_i u_i v_i^T = U_r \Sigma_r V_r^T$$

où  $U_r := U(:, 1:r) \in \mathbb{R}^{m \times r}, \ \Sigma_r := \Sigma(1:r, 1:r) \in \mathbb{R}^{r \times r}$  et  $V_r := V(:, 1:r) \in \mathbb{R}^{n \times r}.$ 

On remarquera que le premier résultat est un résultat en norme spectrale, alors que le second est en norme de Frobenius. Au delà du résultat d'existence de l'optimum, ce théorème donne un résultat d'erreur a priori. C'est à dire que l'erreur d'approximation de faible rang  $||A - \mathcal{T}_r(A)||$  est contrôlée par les plus petites valeurs singulières de la matrice A à partir du rang r + 1.

 $\wedge$ Pour calculer directement la matrice  $\mathcal{T}_r(A)$  en Matlab/Python, il est possible de considérer les codes

```
% MATLAB : renvoi l'approximation de rang r de A
  [U,S,V] = svd(A, 'econ')
  Ur = U(:, 1:r);
  Sr = S(1:r, 1:r);
  Vr = V(:,1:r);
5
  Ar = Ur*Sr*Vr';
6
  # Python : renvoi l'approximation de rang r de A
1
  U, S, VT = np.linalg.svd(A)
2
  Ur = U[:, 0:r]
3
  Sr = np.diag(S[0:r]) # attention S vecteur des val. singulieres
  Vr = VT.T # attention la fonction retourne la transpose de V
  Vr = V[:,0:r]
6
  Ar = Ur@Sr@Vr.T;
```

On donne un résultat intermédiaire utile voire [3, Lemma 4.2.3].

**Lemme 3.1.2** (Intersection de sous-espaces). Soient  $S_1, \ldots, S_k$  des sous espaces de  $\mathbb{R}^n$ . Si  $\delta = \dim S_1 + \cdots + \dim S_k - (k-1)n \ge 1$ , alors il existe des vecteurs orthonormés  $x_1, \ldots, x_{\delta}$  tels que  $x_1, \ldots, x_{\delta} \in S_i, i = 1, \ldots, k$  et en particulier  $S_1 \cap \cdots \cap S_k$  contient un vecteur unitaire.

Démonstration. (Démonstration du Théorème 3.1.1 : norme spectrale).

Soit  $B \in \mathcal{M}_r(\mathbb{R}^{m \times n})$  alors le théorème du rang assure dim  $\ker(B) + \dim \operatorname{im}(B) = \dim \ker(B) + \operatorname{rang}(B) = n \Rightarrow \dim \ker(B) = n - r$ . On peut donc construire une base orthonormée  $\{b_1, \ldots, b_{n-r}\}$  de  $\ker(B) \subset \mathbb{R}^n$ . Par ailleurs si on considère l'espace engendré par les colonnes de  $V_{r+1}$  i.e.  $\mathcal{V}_{r+1} = \operatorname{colspan}\{v_1, \ldots, v_{r+1}\} \subset \mathbb{R}^n$ . Par définition des  $v_i$  alors dim  $\mathcal{V}_{r+1} = r + 1$ . On peut appliquer le Lemme 3.1.2 aux espaces  $\mathcal{V}_{r+1}$  et  $\ker(B)$  de  $\mathbb{R}^n$ , avec  $\delta = n - r + r + 1 - (2 - 1)n = 1$ . Il existe donc  $\bar{x} \in \mathcal{V}_{r+1} \cap \ker(B)$  de norme unitaire i.e.  $\|\bar{x}\|_2 = 1$ . Comme  $\bar{x} \in \ker(B)$ , il vient

$$||A - B||_2^2 = \max_{||x||_2 = 1} ||(A - B)x||_2^2 \ge ||(A - B)\bar{x}||_2^2 = ||A\bar{x}||_2^2 = ||AV_{r+1}c||_2^2,$$

car  $\bar{x} \in \mathcal{V}_{r+1}$  avec  $c \in \mathbb{R}^{r+1}$  les coefficients de  $\bar{x}$  dans la base de  $\mathcal{V}_{r+1}$ . Ainsi

$$||A - B||_2^2 \ge ||U\Sigma V^T V_{r+1}c||_2^2 = ||\Sigma_{r+1}c||_2^2 \ge \sigma_{r+1}^2 ||c||_2^2 = \sigma_{r+1}^2 ||\bar{x}||_2^2 = \sigma_{r+1}^2.$$

car U, V sont orthogonales,  $\bar{x}$  est unitaire et  $\sigma_{r+1}$  est la plus petite valeur singulière. Par ailleurs on a

$$||A - \mathfrak{T}_r(A)||_2 = ||U\Sigma V^T - U_r \Sigma_r V_r^T||_2 = ||U(\Sigma - \tilde{\Sigma}_r) V^T||_2 = ||\Sigma - \tilde{\Sigma}_r||_2 = \sigma_{r+1}$$

avec la matrice  $\tilde{\Sigma}_r \in \mathbb{R}^{m \times n}$  t.q.  $\tilde{\Sigma}_r(1:r,1:r) = \Sigma_r = \text{diag}(\sigma_1,\ldots,\sigma_r)$  et 0 partout ailleurs. Donc le minimum est atteint pour  $B = \mathcal{T}_r(A)$ .

On donne un autre résultat intermédiaire liant le produit scalaire de Frobenius et les valeurs singulières d'une matrice.

**Théorème 3.1.3** (Inégalité de trace de Von Neumann's). Soit  $m \ge n$ ,  $A, B \in \mathbb{R}^{m \times n}$  deux matrices de valeurs singulières notées  $\sigma_1(A) \ge \cdots \ge \sigma_n(A)$  et  $\sigma_1(B) \ge \cdots \ge \sigma_n(B)$  respectivement. Alors on a

$$|\langle A, B \rangle_F| \le \sum_{i=1}^n \sigma_i(A) \sigma_i(B).$$
(3.1.1)

Démonstration. Voir références et détails dans les notes de cours de D. Kressner [4].

Démonstration. (Démonstration du Théorème 3.1.1 : norme de Frobenius). Soit  $B \in \mathcal{M}_r(\mathbb{R}^{m \times n})$ , une conséquence pratique du Théorème 3.1.3 est la suivante

$$||A - B||_F^2 = \langle A - B, A - B \rangle_F = ||A||_F^2 - 2\langle A, B \rangle_F + ||B||_F^2 \ge ||A||_F^2 - 2\sum_{i=1}^n \sigma_i(A)\sigma_i(B) + ||B||_F^2$$
$$= \sum_{i=1}^n (\sigma_i(A) - \sigma_i(B))^2$$

par Corollaire 2.3.8. En particulier si A est de rang k et B de rang  $r \leq k$ . Alors

$$||A - B||_F^2 \ge \sum_{i=1}^r (\sigma_i(A) - \sigma_i(B))^2 + \underbrace{\sum_{i=r+1}^k \sigma_i^2(A)}_{||A - \mathfrak{T}_r(A)||_F^2} \ge ||A - \mathfrak{T}_r(A)||_F^2$$

en effet on a

$$|A - \mathfrak{T}_r(A)||_F = ||\Sigma - \tilde{\Sigma}_r||_F = \sqrt{\sum_{i=r+1}^k \sigma_{r+1}^2}.$$

Donc le minimum est atteint pour  $B = \mathcal{T}_r(A)$ .

## **3.2** Algorithmes

### 3.2.1 Premiers algorithmes détermistes

En pratique pour calculer l'approximation  $\mathcal{T}_r(A)$  de rang r de la matrice A, il suffit de calculer une SVD tronquée à r termes. C'est à dire qu'il faut uniquement conserver les r premiers vecteurs singuliers à gauche et à droite, et valeurs singulières. La procédure est résumée dans l'algorithme 2

<b>Algorithm 2</b> Calcul de $\mathfrak{T}_r(A)$ l'approximation de rang $r$ de $A$ .							
<b>Require:</b> $A \in \mathbb{R}^{m \times n}, r \in \mathbb{R}.$							
• $\lambda, V \leftarrow \texttt{eig}(A^T A).$	⊳ Calcul de	es valeurs et vecteurs propres de $A^T A$					
• $\sigma \leftarrow \sqrt{\lambda}, \sigma \leftarrow \texttt{sort}(\sigma).$	▷ Calcul des valeurs singulières en	suite ordonnées par ordre décroissant					
• $V \leftarrow \texttt{sort}(V)$ .	$\triangleright$ Tri des vecteurs	singuliers dans l'ordre correspondant					
• $V_r \leftarrow V(:, 1:r), \Sigma_r \leftarrow \operatorname{diag}(\sigma(1))$	$(1:r)), U_r \leftarrow AV_r \Sigma_r^{-1}$	$\triangleright$ Calcul de $U_r, \Sigma_r, V_r$ .					
return $U_r \Sigma_r V_r^T$ .	$\triangleright$ Renvoi $\mathfrak{T}_r(A)$ =	$= U_r \Sigma_r V_r^T$ l'approximation de rang $r$ .					

### 3.2.2 Vers des algorithmes aléatoires

Les techniques de **randomisation** i.e. utilisant de **l'algèbre aléatoire** ont émergé ces dernières années comme un outil puissant pour calculer des approximations de faible rang des matrices. Ces techniques ont été dévelopées pour mieux exploiter les architectures modernes des ordinateurs que les méthodes classiques (déterministes) lorsqu'il s'agit de faire des calculs algébriques. Ainsi, elles permettent de traiter des ensembles de "données" vraiment massifs. Dans ce cours, nous nous intéressons plus particulièrement à des opérations sur des grandes matrices, ainsi nous donnons quelques éléments d'introduction à l'algèbre aléatoire pour le calcul de la SVD de matrices. Nous suggérons de lire par exemple [2] pour un article complet et détaillé sur le sujet.

Un algorithme randomisé pour calculer une approximation de faible rang d'une matrice  $A \in \mathbb{R}^{m \times n}$  s'écrit en deux temps.

1. D'abord on cherche un espace de petite dimension  $k \leq \min(m, n)$  (aussi petit que possible) qui capture au mieux l'action de la matrice. C'est à dire, il s'agit de trouver une matrice  $Q \in \mathbb{R}^{m \times k}$  orthonormale telle que

$$\operatorname{im}(Q) \approx \operatorname{im}(A).$$

A cette étape que nous utiliserons un algorithme avec un plongement aléatoire.

2. La seconde étape est de restreindre la matrice A à ce sous espace engendré par Q puis dans calculer la SVD avec les algorithmes vus précédemment. Ainsi on se ramène au calcul d'une SVD pour une matrice de plus petite dimension  $\mathbb{R}^{k \times n}$  et non plus  $\mathbb{R}^{m \times n}$ !

#### Approximation de l'image d'une matrice

Nous cherchons une matrice  $Q \in \mathbb{R}^{m \times k}$  t.q.  $Q^T Q = I_k$ , et vérifiant

$$\operatorname{im}(Q) \approx \operatorname{im}(A).$$

On note que  $QQ^T$  est la projection sur im(Q).

Pour trouver une telle matrice Q, on résout un problème de minimisation

$$\min_{\substack{Q \in \mathbb{R}^{m \times k} \\ Q^T Q = I_k}} \|A - QQ^T A\|,\tag{3.2.1}$$

où la norme matricielle ci-dessous peut être la norme spectrale ou la norme de Frobenius<sup>1</sup>. Une solution au problème précédent est  $Q = U_k$  avec  $U_k$  la matrice des vecteurs singuliers à gauche de A tronquée à ses premières k colonnes. En effet, on observe premièrement que

$$\operatorname{rang}(QQ^{T}A) \leq \min\left(\operatorname{rang}(QQ^{T}), \operatorname{rang}(A)\right) \leq \min(k, m, n) = k$$

car rang $(QQ^T) = \operatorname{rang}(Q) \le \min(k, m)$ . De plus, par définition de l'approximation de rang k

$$\min_{\substack{Z \in \mathbb{R}^{m \times n} \\ \operatorname{rang}(Z) \le k}} \|A - Z\| \le \|A - QQ^T A\| \Rightarrow \min_{\substack{Z \in \mathbb{R}^{m \times n} \\ \operatorname{rang}(Z) \le k}} \|A - Z\| \le \min_{\substack{Q \in \mathbb{R}^{m \times k} \\ Q^T Q = I_k}} \|A - QQ^T A\|,$$

Cependant pour le choix particulier de  $Q = U_k$  et en utilisant  $A = U\Sigma V^T$ 

$$QQ^T A = U_k U_k^T U \Sigma V^T = U_k U_k^T [U_k, \tilde{U}] \Sigma V^T = U_k [I_k, 0] \Sigma V^T = U_k \Sigma_k V_k^T$$

avec  $\tilde{U} \in \mathbb{R}^{m \times (n-k)}$  t.q.  $U_k^T \tilde{U} = 0$  et  $\Sigma_k = \Sigma(1:k,1:k), V_k = V(:,1:k)$ . En résumé, le min est donc atteint pour  $Q = U_k$ !

Dans la suite nous cherchons à développer des algorithmes **aléatoires** qui permettent de calculer une matrice Q t.q.

$$\min_{\substack{Z \in \mathbb{R}^{m \times n} \\ \operatorname{rang}(Z) \le k}} \|A - Z\| \approx \|A - QQ^T A\|.$$

Bien sur, ces algorithmes ne fourniront plus la solution exacte  $U_k$  mais une matrice Q (aléatoire) pas trop éloignée (en un certain sens)!

## Algorithme aléatoires

Algorithm 3 Calcul d'une approximation de  $\mathcal{T}_r(A)$  avec un algorithme aléatoire pour une matrice A.Require:  $A \in \mathbb{R}^{m \times n}, r, p \geq 2$  t.q.  $r + p < \min(m, n)$  et  $\Omega \in \mathbb{R}^{n \times (r+p)}$  une matrice aléatoire.•  $Y \leftarrow A\Omega$ .> Plongement aléatoire de la matrice A ainsi  $Y \in \mathbb{R}^{m \times (n+p)}$ .•  $Q, R \leftarrow qr(Y)$ .> Calcul d'une base pour im(Y).•  $B \leftarrow Q^T A$ .> Produit  $Q^T A$ .return  $Q\mathcal{T}_r(B)$ .> Renvoi une approximation de  $\mathcal{T}_r(A)$ .

Nous expliquons chaque étape de cet algorithme ci-après.

- 1. L'intuition derrière cet algorithme est la suivante. Si rang(A) = r et qu'on considère  $\omega_1, \ldots, \omega_r$  des vecteurs aléatoires alors  $y_j = A\omega_j \in im(A)$ . Comme les vecteurs  $\omega_i$  sont aléatoires, la famille  $\{\omega_1, \ldots, \omega_r\}$ est probablement libre et t.q. les  $\omega_j$  ne sont pas dans ker(A). Ainsi la famille des  $\{y_1, \ldots, y_r\}$  est probablement libre et donc génératrice pour im(A).
- 2. Les colonnes de la matrice  $\Omega \in \mathbb{R}^{n \times (r+p)}$  sont les vecteurs  $\omega_j$ . Cette matrice est un **plongement aléatoire** et permet de considérer Y au lieu de A. Le but est de réduire les couts de calculs avec  $Y = A\Omega$  de dimension plus petite tout en conservant de l'information sur l'image de A.
- 3. En pratique, on s'autorise à sur-échantillonner c'est à dire  $\Omega \in \mathbb{R}^{n \times (r+p)}$  et non  $\Omega \in \mathbb{R}^{n \times r}$ . Ainsi on peut espérer que  $\{y_1, \ldots, y_{r+p}\}$  représente mieux im(A). En pratique on prendre  $p \in \{5, 10\}$ .
- 4. A la fin de l'algorithme, il faut procéder à une étape de troncature supplémentaire sur B car rang $(Q) \le r + p$ .

**Remarque 3.2.1.** Quand le spectre (i.e. les valeurs singulières) de la matrice A décroit lentement, on peut rajouter à cet algorithme une étape itérative sur les sous espaces i.e.  $Y = (AA^T)^q A\Omega$  (voir TP). Cette étape permet d'améliorer l'efficacité de l'algorithme. En pratique q = 2 peut suffire.

<sup>1.</sup> Cela fonctionne pour toute norme unitairement invariante i.e. t.q. pour toutes matrices orthogonales U, V on ait ||UAV|| = ||A||.

#### **Résultats d'approximation**

Nous donnons un résultat d'erreur a priori. Ici, comme  $\Omega$  est aléatoire ces erreurs sont données en moyenne pour les normes de Frobenius et spectrales. Les preuves sont admises voir [2, Theorems 10.4 & 10.5].

**Théorème 3.2.2.** Soit  $A \in \mathbb{R}^{m \times n}$ . Soit un rang  $r \geq 2$  et un paramètre de sur-échantillonage  $p \geq 2$  tels que  $r + p \leq \min(m, n)$ . Soit  $\Omega \in \mathbb{R}^{m \times (r+p)}$  une matrice d'entrées i.i.d. gaussiennes centrées et réduites, i.e.  $\omega_{ij} \sim \mathcal{N}(0, 1)$ . Pour la matrice Q obtenue durant l'algorithme 3, l'erreur en moyenne et en norme de Frobenius est

$$\mathbb{E}(\|A - QQ^{T}A\|_{F}) \leq \sqrt{1 + \frac{r}{p-1}} \underbrace{\left(\sum_{j>r} \sigma_{j}^{2}\right)^{1/2}}_{=\|A - \mathfrak{I}_{r}(A)\|_{F}}.$$
(3.2.2)

L'erreur en moyenne pour la norme spectrale est

$$\mathbb{E}(\|A - QQ^T A\|_2) \le \sqrt{1 + \frac{r}{p-1}} \sigma_{k+1} + \frac{e\sqrt{r+p}}{p} \left(\sum_{j>r} \sigma_j^2\right)^{1/2} \le \left(\sqrt{1 + \frac{r}{p-1}} + \sqrt{\min(m, n) - r}\right) \underbrace{\underbrace{\sigma_{k+1}}_{=\|A - \mathcal{T}_r(A)\|_2}}_{(3.2.3)}$$

Par comparaison avec le théorème 3.1.1, nous n'avons pas exactement l'espérance l'erreur d'approximation entre A et  $QT_r(B)$  retournée par l'algorithme, mais l'espérance de l'erreur de projection sur l'espace engendré par Q. Cependant, on observe que

$$||A - Q\mathfrak{T}_r(B)|| \le ||A - QB|| + ||QB - Q\mathfrak{T}_r(B)|| = ||(I - QQ^T)A|| + ||B - \mathfrak{T}_r(B)||,$$

car Q a ses colonnes orthormales. Ainsi l'erreur obtenue d'approximation recherchée est contrôlée par la somme de l'erreur d'approximation de rang r de B (nulle si B de rang r et qu'on espère assez petite sinon.) et l'erreur de projection.

En suite, les deux bornes d'erreur nous disent qu'à un coefficient multiplicatif près (dépendant de r, p) les erreurs moyennes en norme de Frobenius et spectrale, sont controlées par les erreurs de meilleure approximation de rang r de A pour les mêmes normes. Pour l'erreur en norme spectrale, on voit même que le résultat est un peu plus fin avec un contrôle par les deux normes.

22CHAPITRE 3. APPROXIMATION DE FAIBLE RANG - MÉTHODES DÉTERMINISTES ET ALÉATOIRES

## BIBLIOGRAPHIE

- G.H. Golub and C.F. Van Loan. <u>Matrix Computations</u>. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, 2013.
- [2] N. Halko, P. G. Martinsson, and J. A. Tropp. Finding Structure with Randomness : Probabilistic Algorithms for Constructing Approximate Matrix Decompositions. SIAM Rev., May 2011.
- [3] R.A. Horn and C.R. Johnson. Matrix Analysis. Cambridge University Press, 2012.
- [4] Daniel Kressner. Lecture notes (1), epfl.
- [5] Alfio Quarteroni, Andrea Manzoni, and Federico Negri. <u>Reduced Basis Methods for Partial Differential Equations</u> Springer International Publishing, Cham, Switzerland.