

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITÉ ABDELHAMID BEN BADIS DE MOSTAGANEM
FACULTÉ DES SCIENCES EXACTES ET DE L'INFORMATIQUE
DÉPARTEMENT DE MATHÉMATIQUES ET INFORMATIQUE
FILÈRE : MATHÉMATIQUES



UNIVERSITE
Abdelhamid Ibn Badis
MOSTAGANEM

MÉMOIRE DE FIN D'ÉTUDES

Pour l'Obtention du Diplôme de Master en Mathématiques délivré par

Université de Mostaganem

Spécialité "Modélisation, Contrôle et Optimisation"

présenté par :

BELLACHEHEB NADIA

La décomposition en valeurs singulières et ses applications

soutenu publiquement le 26 /06 / 2019 devant le jury composé de :

| | | | |
|--------------------|----------------------|------------|------|
| Président : | BELHAMITI OMAR | Professeur | UMAB |
| Examineur : | AMIR ABDESSAMAD | Professeur | UMAB |
| Encadreur : | ABLAOUI-LAHMAR NAIMA | MCB | UMAB |

Année Universitaire : 2018 / 2019

M
A
S
T
E
R

Remerciements

Tout d'abord, je remercie mon Dieu qui a bien voulu me donner la force pour effectuer le présent travail.

J'ai le plaisir de formuler mes plus humbles remerciements à :

En premier lieu, à madame **Ablaoui Naima**, mon encadreur. Je la remercie pour ses conseils, sa compréhension, sa disponibilité, son aide ainsi que sa patience.

En second lieu, messieurs les membres du jury, qui m'ont honoré en acceptant d'examiner, de juger et d'évaluer mon mémoire.

Ensuite, un grand merci à mes très chers parents (**Ammour, Fatima**) qui m'ont toujours aidé, soutenu et encouragé au cours de mes études.

Je remercie aussi mes sœurs et mon frère : **Yamina, Sanaa, Laid**.

Je remercie mes amies : **Nour el houda, Souad, Kaoutar**.

Enfin, à toute personne qui a contribué de près ou de loin à l'élaboration de ce travail, je dis, MERCI.

Table des matières

| | |
|--|-------------|
| Liste des tableaux | v |
| Table des figures | vii |
| Liste des abréviations | viii |
| Index des notations | ix |
| Introduction | 1 |
| 1 La décomposition en valeurs singulières | 2 |
| 1 Introduction | 2 |
| 2 Rappels Mathématiques | 2 |
| 3 Le théorème de la décomposition en valeurs singulières | 4 |
| 4 Décompositions réduites | 8 |
| 4.1 La SVD «fine» | 8 |
| 4.2 La SVD «compacte» | 9 |
| 4.3 La SVD «tronquée» | 9 |
| 5 Interprétation Géométrique | 10 |
| 6 Les propriétés de la décomposition en valeurs singulières | 11 |
| 2 Quelques applications de la SVD | 15 |
| 1 Introduction | 15 |
| 2 Résolution d'un système linéaire par la SVD | 15 |
| 2.1 Généralisation de la notion de solution | 16 |
| 2.2 Expression de la solution | 17 |
| 3 L'image numérique | 22 |
| 3.1 Définition de l'image | 22 |
| 3.2 Opérations sur les images | 24 |
| 4 Approximation de rang faible d'une matrice | 25 |
| 5 Le tatouage visible et invisible | 27 |
| 5.1 Introduction | 27 |
| 5.2 Le tatouage invisible | 27 |
| 3 Les résultats expérimentaux | 29 |
| 1 Introduction | 29 |
| 2 Les opérations sur les images | 29 |
| 3 Les résultats numériques de l'approximation de rang faible | 31 |
| 4 L'erreur de reconstruction | 34 |
| 5 Les expériences de tatouage | 36 |
| 5.1 Le tatouage visible | 36 |

| | | |
|-----|---------------------------------|-----------|
| 5.2 | Le tatouage invisible | 37 |
| | Conclusion | 41 |
| | Bibliographie | 42 |
| | Annexe | 42 |

Liste des tableaux

| | |
|--|----|
| 3.1 Les valeurs de k à partir des tolérances e données | 35 |
|--|----|

Table des figures

| | | |
|------|--|----|
| 1.1 | Schéma de la SVD fine. | 9 |
| 1.2 | Schéma de la SVD compacte. | 9 |
| 1.3 | Schéma de la SVD tronquée. | 10 |
| 1.4 | L'effet de la transformation en dimension 2. | 11 |
| 2.1 | Échantillonnage d'une partie d'image. | 23 |
| 2.2 | Quantification des niveaux de gris. | 23 |
| 2.3 | Quantification de la partie d'image échantillonnée. | 23 |
| 2.4 | Transformation d'une image en matrice | 24 |
| 3.1 | (a). Logo de l'université de Abdelhamid Ibn Badis ; (b).Image de Abdelhamid Ibn Badis | 29 |
| 3.2 | (a). Image originale ; (b). Logo binaire ;(c). La somme des deux images | 30 |
| 3.3 | (a). Logo binaire ;(b). Même logo avec un changement ;(c). La soustraction des deux images | 30 |
| 3.4 | Multiplication de l'image de Abdelhamid Ibn Badis par un scalaire M (a). $M = 0.1$; (b). $M = 6$;(c). $M = 20$;(d). $M = 300$ | 30 |
| 3.5 | (a).Image originale avec défaut d'éclairage ; (b). Éclairage ; (c). Image corrigée. | 31 |
| 3.6 | (a). Image originale ; (b).Image transposée | 31 |
| 3.7 | (a). Logo binaire ; (b). La rotation du logo | 31 |
| 3.8 | L'image de Felix Klein | 32 |
| 3.9 | L'approximation du portrait de Klein pour les différents valeurs du rang (a). $k=1$; (b). $k=3$; (c). $k=5$; (d). $k=10$ | 33 |
| 3.10 | L'approximation du portrait de Klein pour les différents valeurs du rang (a). $k=20$; (b). $k=80$; (c). $k=100$; (d). $k=300$ | 33 |
| 3.11 | L'approximation du portrait de Klein pour les différents valeurs du rang (a). $k=400$; (b). $k=500$; (c). $k=524$; (d).image original | 34 |
| 3.12 | Variation de l'erreur relative en fonction de k | 35 |
| 3.13 | Variation du logarithme de l'erreur relative en fonction de k | 36 |
| 3.14 | (a). Image de ABD ALHAMID IBN BADIS ; (b). La marque. | 36 |
| 3.15 | (a). Image originale ; (b). La marque ;(c).Image tatouée | 37 |
| 3.16 | Tatouage invisible pour différentes valeurs du coefficient d'ajustement α (avec la matrice Σ) (a). $\alpha = 10$; (b). $\alpha = 5$;(c). $\alpha = 3$ | 37 |
| 3.17 | Tatouage invisible pour différentes valeurs du coefficient d'ajustement α (avec la matrice Σ) (a). $\alpha = 1$; (b). $\alpha = 0.5$;(c). $\alpha = 0.1$ | 37 |
| 3.18 | Tatouage invisible pour différentes valeurs du coefficient d'ajustement α (avec la matrice Σ) (a). $\alpha = 0.01$; (b). $\alpha = 0.001$;(c). image originale | 38 |
| 3.19 | (a).Image originale ; (b). Image tatoué ;(c).La marque | 38 |
| 3.20 | Tatouage invisible pour différentes valeurs du coefficient d'ajustement α (avec la matrice U). (a). $\alpha = 10$; (b). $\alpha = 5$; (c). $\alpha = 3$ | 38 |

| | |
|---|----|
| 3.21 Tatouage invisible pour différentes valeurs du coefficient d'ajustement α (avec la matrice U). (a). $\alpha = 1$; (b). $\alpha = 0.5$; (c). $\alpha = 0.1$ | 39 |
| 3.22 Tatouage invisible pour différentes valeurs du coefficient d'ajustement α (avec la matrice U). (a). $\alpha = 0.01$; (b). $\alpha = 0.001$; (c). image originale | 39 |
| 3.23 Tatouage invisible pour différentes valeurs du coefficient d'ajustement α (avec la matrice V). (a). $\alpha = 10$; (b). $\alpha = 5$; (c). $\alpha = 3$ | 39 |
| 3.24 Tatouage invisible pour différentes valeurs du coefficient d'ajustement α (avec la matrice V). (a). $\alpha = 1$; (b). $\alpha = 0.5$; (c). $\alpha = 0.1$ | 39 |
| 3.25 Tatouage invisible pour différentes valeurs du coefficient d'ajustement α (avec la matrice V). (a). $\alpha = 0.01$; (b). $\alpha = 0.001$; (c). image originale | 40 |

Liste des abréviations

- SVD** : Singular Value Decomposition
- AC-SVD** : Algorithme de Chandra par la SVD
- Pixel** : picture element

Index des notations

| | |
|----------------------------------|---|
| $\text{vect}(\{u_i\}_{i \in I})$ | : Le sous-espace engendré par $\{u_i\}_{i \in I}$ où I est un ensemble des indices fini. |
| $\text{Im}(A)$ | : L'image de la matrice A . |
| $\text{Ker}(A)$ | : Le noyau de la matrice A . |
| $\text{cond}_2(A)$ | : Le conditionnement de la matrice A par rapport à la norme 2. |
| A^{-1} | : L'inverse de la matrice A . |
| $ A $ | : Déterminant de la matrice A . |
| $\text{rg}(A)$ | : Le rang de la matrice A . |
| I_m | : La matrice identité de taille $m \times m$ (s'il n'y a pas de confusion on la note I). |
| A^t | : Transposé de la matrice A (A est symétrique $\Leftrightarrow A^t=A$). |
| $\text{tr}(A)$ | : La trace de la matrice A . |

Introduction

Le calcul matriciel s'impose aujourd'hui dans tous les domaines de la science, comme cela est le cas en économie, en chimie, en imagerie et pour tous les phénomènes qui sont modélisés par des équations différentielles ordinaires ou des équations aux dérivées partielles. L'application des différentes méthodes numériques à ces modèles (comme les différences finies, les éléments finis, les volumes finis, ...), ramène l'étude de ces derniers à la résolution de système linéaire et donc au calcul matriciel. Il est cependant intéressant de remarquer que le développement des outils «classiques» comme les décompositions matricielles qui remonte au début du 18^{ème} siècle, est remis à jour grâce à la puissance et la performance de l'outil informatique.

Parmi celles-ci, la décomposition en valeurs singulières d'une matrice (SVD), qui est considérée comme étant un outil puissant, grâce aux multiples propriétés qu'elle offre. Les domaines d'applications de cette décomposition sont nombreux, on peut citer

- La compression d'image (pour gagner l'espace de stockage dans la mémoire de l'ordinateur).
- L'analyse de la sensibilité d'un modèle (pour choisir une bonne paramétrisation d'un problème inverse).
- Le tatouage numérique (pour l'authentification des images numériques).
- Le filtrage c'est à dire la diminution des bruits (pour améliorer la qualité d'image).
- La résolution des systèmes linéaires (pour résoudre les problèmes mal posés), ...etc.

Dans ce mémoire, on s'est intéressé à l'application de la SVD dans les trois domaines suivants

- La résolution des systèmes linéaires.
- La compression d'image.
- Le tatouage numérique.

Ce mémoire est organisé en trois chapitres.

Dans le premier chapitre, on commence par rappeler quelques définitions et théorèmes utiles. Ensuite, on introduit la décomposition en valeurs singulières et on termine par énumérer les propriétés qui en découlent.

Le 2^{ème} chapitre est consacré aux différentes applications de la décomposition en valeurs singulières (SVD) et qui sont la résolution des systèmes linéaires, la compression d'image et le tatouage numérique.

Dans le dernier chapitre, on va faire des expériences sous MatLab pour la compression d'image et le tatouage numérique.

Chapitre 1

La décomposition en valeurs singulières

1 Introduction

La décomposition en valeurs singulières est un outil de factorisation de matrice carré ou rectangulaire. Elle est considérée comme la généralisation du procédé de diagonalisation pour les matrices carrées.

La décomposition en valeurs singulières d'une matrice quelconque, c'est une factorisation de la matrice en deux matrices orthogonales et une matrice pseudo-diagonale qui contient les valeurs singulières de la matrice décomposée classées en ordre décroissant.

2 Rappels Mathématiques

- Pour deux entiers non nuls m et n , $\mathcal{M}_{m,n}(\mathbb{R})$ est l'espace vectoriel des matrices à m lignes et n colonnes (à coefficients réels). Cet espace est de dimension $m \times n$.
- Soit $A \in \mathcal{M}_{m,m}(\mathbb{R})$. A est dite orthogonale si et seulement si

$$A^t A = A A^t = I \text{ c'est à dire } A^t = A^{-1}. \quad (1.1)$$

- Un produit scalaire sur E est une forme bilinéaire symétrique et définie positive sur E noté $\langle \cdot, \cdot \rangle$, tel que
 $\langle \cdot, \cdot \rangle : E \times E \rightarrow \mathbb{R}$, qui vérifie :
 - $\langle x, x \rangle > 0 \forall x \in E$ et $x \neq 0$.
 - $\langle x, y \rangle = \langle y, x \rangle \forall x, y \in E$.
 - $\langle \alpha_1 x_1 + \alpha_2 x_2, y \rangle = \alpha_1 \langle x_1, y \rangle + \alpha_2 \langle x_2, y \rangle \forall \alpha_1, \alpha_2 \in \mathbb{R}$ et $\forall x_1, x_2 \in E$.
- Une norme $\| \cdot \|$ sur E est une application de $E \rightarrow \mathbb{R}_+$ qui vérifie :
 - $\|x\| = 0 \Leftrightarrow x = 0_E$.
 - $\|\lambda x\| = |\lambda| \|x\| \forall \lambda \in \mathbb{R}$ et $\forall x \in E$.
 - $\|x + y\| \leq \|x\| + \|y\|, \forall x, y \in E$.
- Un réel positif σ est appelé valeur singulière de A si et seulement si il existe un vecteur unitaire u dans \mathbb{R}^m ($\|u\| = 1$) et un vecteur unitaire v dans \mathbb{R}^n tels que :

$$A v = \sigma u. \quad (1.2)$$

Le vecteur u (respectivement v) est appelé vecteur singulier à gauche (respectivement à droite) pour σ .

- Soit $A \in \mathcal{M}_{m,n}(\mathbb{R})$. On note par a_1, a_2, \dots, a_n , les vecteurs colonnes de la matrice A ($a_i \in \mathbb{R}^m$). On appelle rang de A , la dimension du sous-espace de \mathbb{R}^m engendré par les vecteurs a_1, a_2, \dots, a_n .
- Soit $A \in \mathcal{M}_{n,n}(\mathbb{R})$. A est dite définie positive si :

$$\forall x \in \mathbb{R}^n, \langle Ax, x \rangle \geq 0 \text{ c-à-d } x^t Ax \geq 0. \quad (1.3)$$

- Une famille de vecteurs $\{u_i\}_{i \in I}$ est dite orthogonale si les vecteurs de cette famille sont deux à deux orthogonaux. C'est à dire

$$\langle u_i, u_j \rangle = 0 \text{ si } i \neq j, \forall i, j \in I. \quad (1.4)$$

Elle est dite orthonormée si de plus elle vérifie

$$\langle u_i, u_i \rangle = 1 \text{ si } i \in I. \quad (1.5)$$

- Soit $A \in \mathcal{M}_{m,m}(\mathbb{R})$. S'il existe un scalaire $\lambda \in \mathbb{R}$ (ou \mathbb{C}) et un vecteur non nul $v \in \mathbb{R}^m$ tels que $Av = \lambda v$, on dit que λ est une valeur propre de A et v vecteur propre de A associé à λ .

Le spectre de A , noté $Sp(A)$ est l'ensemble défini par :

$$Sp(A) = \{\lambda, \lambda \text{ valeur propre de } A\}. \quad (1.6)$$

Le rayon spectrale de A , noté $\rho(A)$ est défini par :

$$\rho(A) = \max\{|\lambda|, \lambda \in Sp(A)\}. \quad (1.7)$$

Le polynôme caractéristique de la matrice A , noté $P(\lambda)$ est défini par :

$$P(\lambda) = |A - \lambda I|. \quad (1.8)$$

Toute valeur propre de la matrice A est racine du polynôme caractéristique $P(\lambda)$.

- On définit la norme matricielle subordonnée à une norme vectorielle par :

$$\forall A \in \mathcal{M}_{m,n}(\mathbb{R}), \|A\|_p = \sup_{v \in \mathbb{R}^n; \|v\|_p \neq 0} \frac{\|Av\|_p}{\|v\|_p}. \quad (1.9)$$

En particulier si $p = 2$ alors $\|A\|_2 = \sqrt{\rho(A^t A)}$.

- Soit $A \in \mathcal{M}_{m,m}(\mathbb{R})$. On dit que A est diagonalisable s'il existe une matrice V inversible telle que :

$$A = VDV^{-1}, \quad (1.10)$$

où D est une matrice diagonale contenant les valeurs propres de A .

- Soit $A \in \mathcal{M}_{m,n}(\mathbb{R})$, son pseudo-inverse A^+ est l'unique matrice de $A \in \mathcal{M}_{n,m}(\mathbb{R})$ vérifiant les conditions suivantes :

1. $AA^+A = A$;
2. $A^+AA^+ = A^+$;
3. $(AA^+)^t = AA^+$;
4. $(A^+A)^t = A^+A$.

- **Théorème 1.1** [6](page368)

Les colonnes d'une matrice A de type $m \times n$ sont orthonormées si et seulement si $A^t A = I_n$.

• **Théorème 1.2 (Spectral des matrices symétriques)** [6](page 426)

Soit $A \in \mathcal{M}_{m,m}(\mathbb{R})$; si A est symétrique et définie positive alors A est diagonalisable dans une base orthonormée. Autrement dit, il existe une base orthonormée qui consiste en des vecteurs propres de A .

• **Théorème 1.3 (Du rang)** [6](page 168)

Toute matrice A à n colonnes vérifie la relation suivante :

$$rg(A) + dim(ker(A)) = n \quad (1.11)$$

• **Théorème 1.4** [6](page 439)

Soit A une matrice symétrique. Alors M est égal à la plus grande valeur propre de A , m à la plus petite,

où $m = \min\{x^t Ax : \|x\| = 1\}$ et $M = \max\{x^t Ax : \|x\| = 1\}$.

3 Le théorème de la décomposition en valeurs singulières

Théorème 1.5 [2](page 72) Soit $A \in \mathcal{M}_{m,n}(\mathbb{R})$, m et n deux entiers. Il existe deux matrices orthogonales $U \in \mathcal{M}_{m,m}(\mathbb{R})$ et $V \in \mathcal{M}_{n,n}(\mathbb{R})$ telles que :

$$A = U \Sigma V^t, \quad (1.12)$$

où Σ prend les formes suivantes selon le rang de la matrice A :

-si le rang de A est m alors

$$\Sigma = \begin{pmatrix} \Sigma_1 & 0 \end{pmatrix} \in \mathbb{R}^{m \times n} \quad (1.13)$$

-si le rang de A est n alors

$$\Sigma = \begin{pmatrix} \Sigma_1 \\ 0 \end{pmatrix} \in \mathbb{R}^{m \times n} \quad (1.14)$$

-si le rang de A est inférieur strictement à $\min(m, n)$

$$\Sigma = \begin{pmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{pmatrix} \in \mathbb{R}^{m \times n} \quad (1.15)$$

où Σ_1 est une matrice diagonale qui contient les valeurs singulières de A classées en ordre décroissant.

Preuve.

- On commence par remarquer que la matrice $A^t A \in \mathcal{M}_{n,n}(\mathbb{R})$ est symétrique :

$$(A^t A)^t = A^t (A^t)^t = A^t A. \quad (1.16)$$

Elle est aussi semi-définie positive car : $\forall x \in \mathbb{R}^n$,

$$x^t A^t A x = \langle Ax, Ax \rangle = \|Ax\|_2^2 \geq 0. \quad (1.17)$$

D'après le théorème spectral, la matrice $A^t A$ est diagonalisable dans une base orthonormée de vecteurs propres (v_1, v_2, \dots, v_n) :

$$A^t A = V D V^t. \quad (1.18)$$

où D est une matrice diagonale de taille $n \times n$ contenant les valeurs propres de $A^t A$ et $V = [v_1 v_2 \dots v_n]$.

$A^t A$ étant définie positive, alors ses valeurs propres λ_i pour $i = 1, \dots, r$ ($r \leq n$) sont positives.

On pose $\sigma_i = \sqrt{\lambda_i}$ pour $i = 1, \dots, r$ avec $\sigma_1 > \sigma_2 > \dots > \sigma_r$.

Pour les valeurs σ_i non nulles, on introduit le vecteur u_i , $i = 1, \dots, r$ défini par :

$$u_i = \frac{1}{\sigma_i} A v_i; i = 1, \dots, r. \quad (1.19)$$

On peut vérifier facilement que les vecteurs u_i sont orthonormés

$$\begin{aligned} \langle u_i, u_j \rangle &= \left\langle \frac{1}{\sigma_i} A v_i, \frac{1}{\sigma_j} A v_j \right\rangle \\ &= \frac{1}{\sigma_i \sigma_j} \langle A v_i, A v_j \rangle \\ &= \frac{1}{\sigma_i \sigma_j} v_j^t A^t A v_i \\ &= \frac{1}{\sigma_i \sigma_j} v_j^t \lambda_i v_i. \end{aligned} \quad (1.20)$$

Si $i = j$ alors,

$$\langle u_i, u_i \rangle = \frac{\lambda_i}{\sigma_i^2} = 1, \quad (1.21)$$

sinon

$$\langle u_i, u_j \rangle = \frac{\lambda_i}{\sigma_i \sigma_j} v_j^t v_i = 0. \quad (1.22)$$

Donc la famille $\{u_1, \dots, u_r\}$ est orthonormée dans \mathbb{R}^m .

En notant U la matrice qui a pour colonnes les vecteurs u_i et V la matrice qui a pour colonnes les vecteurs v_i , $1 \leq i \leq r$; avec $r = \min(m, n)$, ces matrices sont orthogonales d'après le **théorème 1.1**.

On a d'une part

$$AV = \begin{bmatrix} A v_1 & \dots & A v_r & 0 & \dots & 0 \end{bmatrix}, \quad (1.23)$$

qui s'écrit selon la relation (1.19)

$$AV = \begin{bmatrix} \sigma_1 u_1 & \dots & \sigma_r u_r & 0 & \dots & 0 \end{bmatrix}. \quad (1.24)$$

D'autre part, en définissant Σ comme étant la matrice diagonale de coefficients diagonaux $\sigma_1, \dots, \sigma_r$, alors

$$U \Sigma = \begin{bmatrix} \sigma_1 u_1 & \dots & \sigma_r u_r & 0 & \dots & 0 \end{bmatrix}. \quad (1.25)$$

Par conséquent,

$$U \Sigma = AV. \quad (1.26)$$

Comme V est une matrice orthogonale, on obtient :

$$U \Sigma V^t = AV V^t = A. \quad (1.27)$$

Finalement, $A = U \Sigma V^t$, où U et V sont des matrices orthogonales.

■

Remarque 1.1

1. Soit $A \in \mathcal{M}_{m,n}(\mathbb{R})$ avec $A = U \Sigma V^t$, alors pour $i = 1, \dots, r$:

$$\|A v_i\| = \sigma_i. \quad (1.28)$$

Cela signifie que la longueur du vecteur $A v_i$ est égale à la valeur singulière σ_i de A pour $i = 1, \dots, r$.

2. Connaissant la décomposition en valeurs singulières de A ; $A = U\Sigma V^t$, son pseudo inverse est déterminé par $A^\dagger = V\Sigma^\dagger U^t$.
3. Soit $A \in \mathcal{M}_{m,n}(\mathbb{R})$ avec $A = U\Sigma V^t$, alors on peut écrire A sous la forme :

$$A = \sum_{i=1}^{\min(m,n)} \sigma_i u_i v_i^t = \sum_{i=1}^r \sigma_i u_i v_i^t \text{ avec } r = \text{rg}(A). \quad (1.29)$$

Car si on note les vecteurs suivants

$$u_i = \begin{pmatrix} u_{1i} \\ \vdots \\ u_{mi} \end{pmatrix}, v_i^t = (v_{i1} \quad \dots \quad v_{in}), \quad (1.30)$$

alors on a

$$\begin{aligned} U\Sigma &= [\sigma_1 u_1 \dots \sigma_1 u_m] \\ &= [\sigma_1 u_1 \quad \dots \quad \sigma_r u_r \quad 0 \quad \dots \quad 0] \\ &= \begin{pmatrix} \sigma_1 u_{11} & \sigma_2 u_{12} & \dots & \sigma_r u_{1r} & 0 & \dots & 0 \\ \sigma_1 u_{21} & \sigma_2 u_{22} & \dots & \sigma_r u_{2r} & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ \sigma_1 u_{m1} & \sigma_2 u_{m2} & \dots & \sigma_r u_{mr} & 0 & \dots & 0 \end{pmatrix}. \end{aligned} \quad (1.31)$$

On multiplie cette dernière par la matrice V^t , on obtient

$$U\Sigma V^t = \begin{pmatrix} \sigma_1 u_{11} v_{11} + \dots + \sigma_r u_{1r} v_{r1} & \sigma_1 u_{11} v_{12} + \dots + \sigma_r u_{1r} v_{r2} & \dots & \sigma_1 u_{11} v_{1n} + \dots + \sigma_r u_{1r} v_{rn} \\ \sigma_1 u_{21} v_{11} + \dots + \sigma_r u_{2r} v_{r1} & \sigma_1 u_{21} v_{12} + \dots + \sigma_r u_{2r} v_{r2} & \dots & \sigma_1 u_{21} v_{1n} + \dots + \sigma_r u_{2r} v_{rn} \\ \vdots & \vdots & & \vdots \\ \sigma_1 u_{m1} v_{11} + \dots + \sigma_r u_{mr} v_{r1} & \sigma_1 u_{m1} v_{12} + \dots + \sigma_r u_{mr} v_{r2} & \dots & \sigma_1 u_{m1} v_{1n} + \dots + \sigma_r u_{mr} v_{rn} \end{pmatrix}$$

on remarque que

$$\sigma_i u_i v_i^t = \sigma_i \begin{pmatrix} u_{1i} v_{i1} & u_{1i} v_{i2} & \dots & u_{1i} v_{in} \\ \vdots & \vdots & & \vdots \\ u_{mi} v_{i1} & u_{mi} v_{i2} & \dots & u_{mi} v_{in} \end{pmatrix}.$$

Alors

$$A = U\Sigma V^t = \sigma_1 u_1 v_1^t + \dots + \sigma_r u_r v_r^t.$$

Exemple 1.1

Pour trouver la décomposition en valeurs singulières de la matrice A , on suit les étapes :

- Calcul des valeurs propres de AA^t .
- Calcul de la matrice U , ($U = [u_1, \dots, u_m]$ avec u_i sont des vecteurs propres de la matrice AA^t).
- Calcul de la matrice V , ($V = [v_1, \dots, v_n]$ avec v_i sont des vecteurs propres de la matrice $A^t A$).

Soit A la matrice rectangulaire.

$$A = \begin{pmatrix} 3 & 2 & 2 \\ 2 & 3 & -2 \end{pmatrix} \quad (1.32)$$

a) On commence par calculer les valeurs propres de AA^t .

$$AA^t = \begin{pmatrix} 17 & 8 \\ 8 & 17 \end{pmatrix}. \quad (1.33)$$

Pour trouver les valeurs propres de AA^t , on résout l'équation $P(\lambda) = 0$

$$\begin{aligned} P(\lambda) = 0 &\Leftrightarrow |AA^t - \lambda I| = 0. \\ &\Leftrightarrow \lambda^2 - 34\lambda + 225 = 0. \\ &\Leftrightarrow (\lambda - 25)(\lambda - 9) = 0. \end{aligned} \quad (1.34)$$

Donc les valeurs propres de AA^t sont $\lambda_1 = 25$ et $\lambda_2 = 9$.

b) Calcul de U .

La construction de la matrice U passe par le calcul des vecteurs propres u_1 et u_2 associés aux valeurs propres λ_1 et λ_2 respectivement de la matrice AA^t .

Par définition on a,

$$AA^t u_1 = 25u_1. \quad (1.35)$$

En posant

$$u_1 = \begin{pmatrix} x \\ y \end{pmatrix}, \quad (1.36)$$

cela revient à résoudre le système suivant :

$$\begin{cases} 17x + 8y = 25x \\ 8x + 17y = 25y \end{cases} \Rightarrow x = y. \quad (1.37)$$

Pour $x=1$, la normalisation du vecteur $(1, 1)^t$ donne le vecteur unitaire u_1 ,

$$u_1 = \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix}. \quad (1.38)$$

De la même manière on obtient

$$u_2 = \begin{pmatrix} -1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix}. \quad (1.39)$$

Finalement la matrice U est

$$U = \begin{pmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{pmatrix}. \quad (1.40)$$

c) Calcul de V :

La construction de la matrice V passe par le calcul des vecteurs propres v_1 , v_2 et v_3 associés aux valeurs propres λ_1 , λ_2 et λ_3 de la matrice $A^t A$.

$$A^t A = \begin{pmatrix} 13 & 12 & 2 \\ 12 & 13 & -2 \\ 2 & -2 & 8 \end{pmatrix}. \quad (1.41)$$

Puisque les valeurs propres de AA^t (2×2) et $A^t A$ (3×3) sont les mêmes, par complétion par des zéro on a, $\lambda_1 = 25$, $\lambda_2 = 9$ et $\lambda_3 = 0$. Il reste à calculer les vecteurs propres associés v_i pour $i = 1, 2, 3$. On résout donc le système

$$A^t A v_1 = 25v_1. \quad (1.42)$$

En posant

$$v_1 = \begin{pmatrix} x \\ y \\ z \end{pmatrix}, \quad (1.43)$$

ce dernier est équivalent à

$$\begin{cases} 13x + 12y + 2z = 25x \\ 12x + 13y - 2z = 25y \\ 2x - 2y + 8z = 25z \end{cases} \Rightarrow x = y \text{ et } z = 0. \quad (1.44)$$

On prend $x = 1$ et avec la normalisation de ce vecteur on obtient,

$$v_1 = \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \\ 0 \end{pmatrix}. \quad (1.45)$$

De la même manière on obtient

$$v_2 = \begin{pmatrix} -1/\sqrt{18} \\ 1/\sqrt{18} \\ -4/\sqrt{18} \end{pmatrix} \text{ et } v_3 = \begin{pmatrix} -2/3 \\ 2/3 \\ 1/3 \end{pmatrix}. \quad (1.46)$$

Finalement

$$V = \begin{pmatrix} 1/\sqrt{2} & -1/\sqrt{18} & -2/3 \\ 1/\sqrt{2} & 1/\sqrt{18} & 2/3 \\ 0 & -4/\sqrt{18} & 1/3 \end{pmatrix}. \quad (1.47)$$

Donc la décomposition en valeurs singulières de la matrice A est

$$A = U \Sigma V^t. \quad (1.48)$$

C'est à dire

$$A = \begin{pmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{pmatrix} \times \begin{pmatrix} 5 & 0 & 0 \\ 0 & 3 & 0 \end{pmatrix} \times \begin{pmatrix} 1/\sqrt{2} & 1/2 & 0 \\ -1/\sqrt{18} & 1/\sqrt{18} & -4/\sqrt{18} \\ -2/3 & 2/3 & 1/3 \end{pmatrix}. \quad (1.49)$$

4 Décompositions réduites

Dans la plupart des applications la forme complète de la décomposition en valeurs singulières n'est pas employée, notamment pour des raisons de temps de calcul et d'allocation de la mémoire d'ordinateur. Des versions réduites de la SVD existent et sont le plus souvent utilisées. [2] (page page75)

Soient donc $A \in \mathcal{M}_{m,n}(\mathbb{R})$ avec m et n deux entiers et sa décomposition en valeurs singulières suivante

$$A = U \Sigma V^t. \quad (1.50)$$

4.1 La SVD «fine»

On calcule seulement les n vecteurs colonnes de U et les n vecteurs lignes de V^t . Les vecteurs colonnes restants sont omis, ce qui permet d'éviter une quantité importante de calculs dans le cas $n \ll m$.

La figure suivante 1.1 illustre comment éviter les calculs dans le cas $n \ll m$.

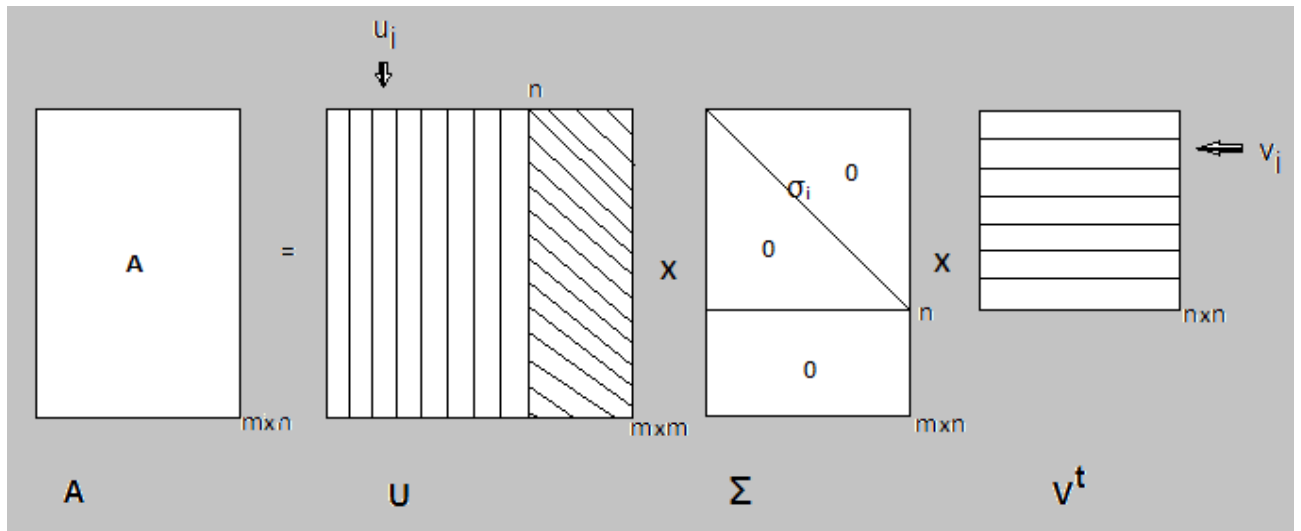


FIGURE 1.1 – Schéma de la SVD fine.

4.2 La SVD «compacte»

On calcule seulement les r vecteurs colonnes de U et les r vecteurs lignes de V^t correspondants aux valeurs singulières non-nulles. L'algorithme de calcul de la SVD compacte est plus rapide que le précédent si $r \ll n$, avec $r = \text{rg}(A)$.

Voir la figure 1.2

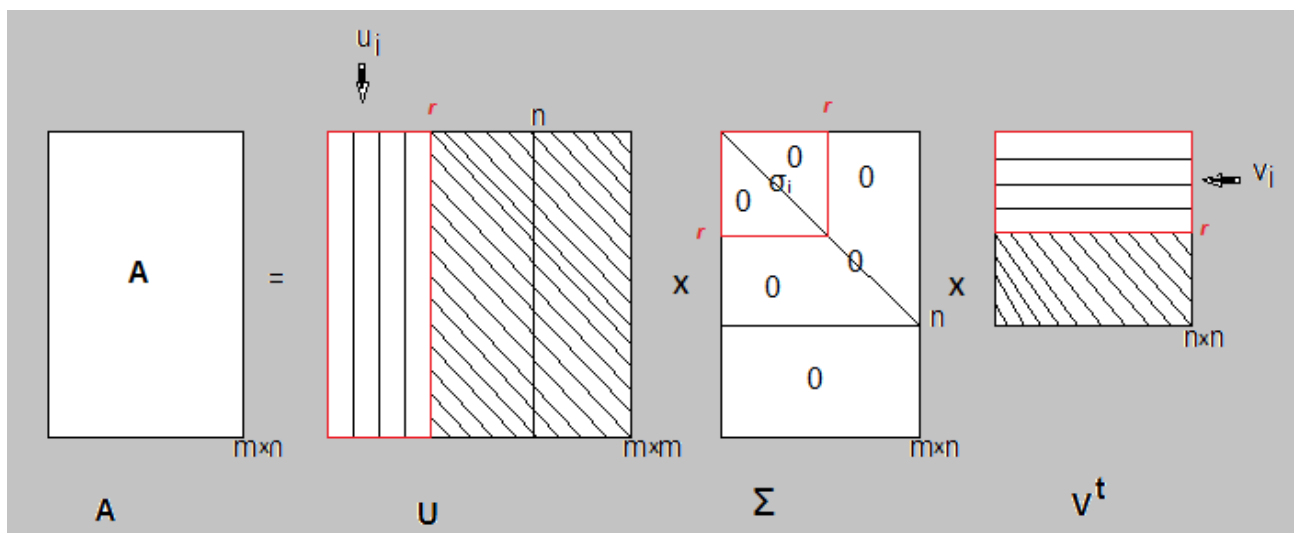


FIGURE 1.2 – Schéma de la SVD compacte.

4.3 La SVD «tronquée»

Seuls les k vecteurs colonnes de U et les k vecteurs lignes de V^t correspondants aux k plus grandes valeurs singulières sont calculés. Autrement dit, nous avons $A \approx \sum_{i=1}^k u_i \sigma_i v_i^t$ avec $k \ll \min(m, n)$. Ce calcul est encore plus rapide que celui de la SVD «fine» si le rang r de la matrice A vérifie $k \ll r$.

Voir la figure 1.3

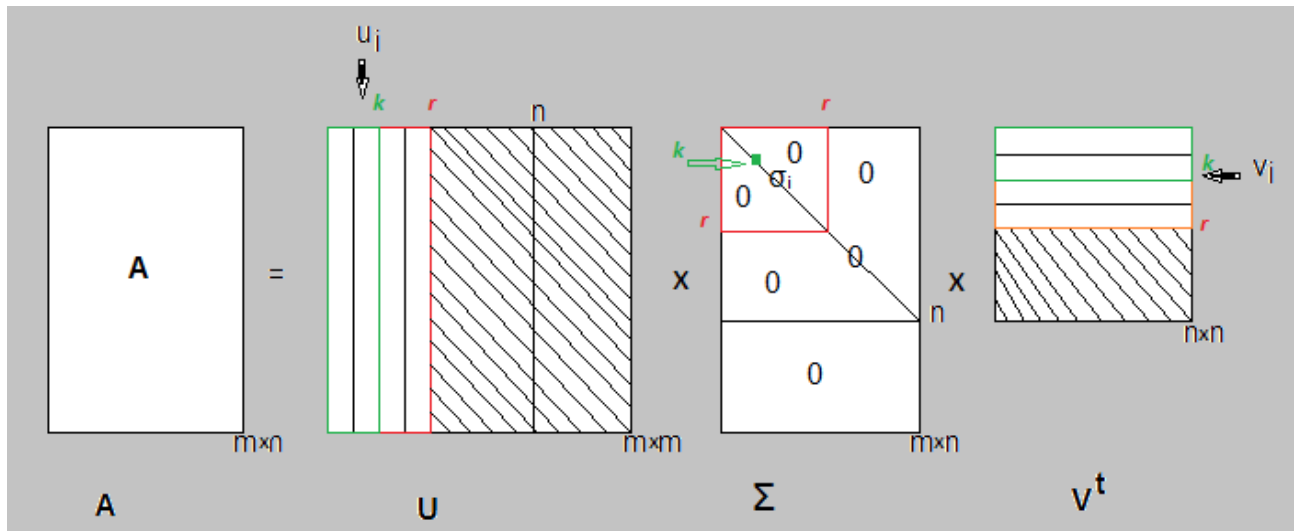


FIGURE 1.3 – Schéma de la SVD tronquée.

5 Interprétation Géométrique

Soit $A \in \mathcal{M}_{m,n}(\mathbb{R})$.

L'application linéaire $x \rightarrow Ax$ transforme la sphère unité $\{x : \|x\| = 1\}$ de \mathbb{R}^n en une hyperellipse dans \mathbb{R}^m .

Les valeurs de x qui maximisent la quantité $\|Ax\|$ sont les mêmes que celles qui maximisent $\|Ax\|^2$. Or $\|Ax\|^2$ s'étudie facilement car :

$$\|Ax\|^2 = (Ax)^t(Ax) = x^t A^t Ax = x^t (A^t A)x \quad (1.51)$$

On sait que $A^t A$ est symétrique donc d'après le **théorème 1.4**, ce maximum est égal à la plus grande valeur propre λ_1 de $A^t A$. De plus il est atteint en un vecteur propre de $A^t A$ associé à la valeurs propre λ_1 .

D'après la **Remarque 1.1**, les σ_i de A sont les longueurs des demi-axes de l'ellipse et qui sont $A v_i$. C'est ce qui détermine entièrement une ellipse.

Pour $m = n = 2$, on va voir à travers un exemple comment l'application linéaire $x \rightarrow Ax$ transforme le cercle unité en une ellipse et ceci en utilisant la SVD de A .

Exemple 1.2

Soit la matrice

$$A = \begin{pmatrix} 2 & 1 \\ 0 & 3 \end{pmatrix}. \quad (1.52)$$

Sa décomposition en valeurs singulières est :

$$A = \begin{pmatrix} 0.4719 & -0.8817 \\ 0.8817 & 0.4719 \end{pmatrix} \times \begin{pmatrix} 3.2566 & 0 \\ 0 & 1.8424 \end{pmatrix} \times \begin{pmatrix} 0.2898 & -0.9571 \\ 0.9571 & 0.2898 \end{pmatrix} \quad (1.53)$$

Les vecteurs propres de $A^t A$ sont :

$$v_1 = \begin{pmatrix} 0.2898 \\ 0.9571 \end{pmatrix}; v_2 = \begin{pmatrix} -0.9571 \\ 0.2898 \end{pmatrix}. \quad (1.54)$$

Donc le vecteur Av_1 correspond au point de l'ellipse le plus éloigné de l'origine

$$Av_1 = \begin{pmatrix} 1.5367 \\ 2.8713 \end{pmatrix} \quad (1.55)$$

et le vecteur Av_2 correspond au point de l'ellipse le plus proche de l'origine

$$Av_2 = \begin{pmatrix} -1.6244 \\ 0.8694 \end{pmatrix}. \quad (1.56)$$

La figure suivante représente l'effet de la transformation en dimension 2.

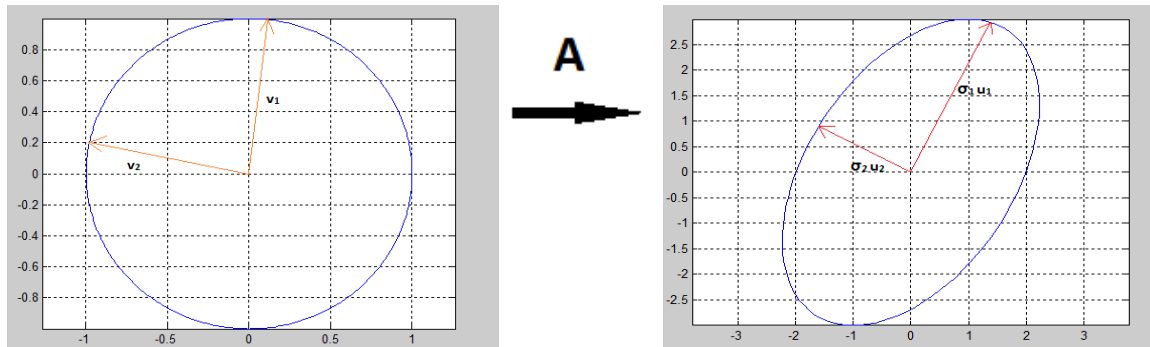


FIGURE 1.4 – L'effet de la transformation en dimension 2.

6 Les propriétés de la décomposition en valeurs singulières

Beaucoup de propriétés découlent de la décomposition d'une matrice en valeurs singulières, parmi lesquelles on a choisi de citer celles qui sont le plus fréquemment utilisées.

Soit $A \in \mathcal{M}_{m,n}(\mathbb{R})$ de rang r avec $A = U\Sigma V^t$, où $U = [u_1, u_2, \dots, u_m]$, $V = [v_1, v_2, \dots, v_n]$ et Σ est la matrice contenant les valeurs singulières de A $\sigma_i, i = 1, \dots, r$.

- A) Le rang de la matrice A coïncide avec le nombre des valeurs singulières non nulles.
- B) $Im(A) = vect(u_1, u_2, \dots, u_r)$.
- C) $ker(A) = vect(v_{r+1}, v_{r+2}, \dots, v_n)$.
- D) $\|A\|_2 = \sigma_1$.
- E) $\|A\|_F^2 = tr(AA^t) = \sum_{i=1}^r \sigma_i^2$.
- F) Si le rang de $A = m$ alors $AA^\dagger = I_m$.
- G) Si le rang de $A = n$ alors $A^\dagger A = I_n$.
- H) Pour toute matrice d'ordre n inversible, le conditionnement de A , $cond_2(A) = \frac{\sigma_1}{\sigma_r}$.

Preuve.

• **preuve de A)**

Pour démontrer la propriété A), il suffit de démontrer que $rg(A) = rg(A^t A)$.
D'après le **théorem 1.3** appliqué aux matrices A et $A^t A$ on a :

$$n = dim(ker(A)) + rg(A) \quad (1.57)$$

et

$$n = dim(ker(A^t A)) + rg(A^t A). \quad (1.58)$$

Donc on peut déduire que :

$$rg(A) = rg(A^t A) \Leftrightarrow dim(ker(A)) = dim(ker(A^t A)) \quad (1.59)$$

Autrement dit il suffit de démontrer que

$$ker(A) = ker(A^t A). \quad (1.60)$$

C'est à dire il suffit de démontrer la double inclusion. La première inclusion $ker(A) \subset ker(A^t A)$ est évidente car

pour $x \in ker(A)$ c-à-d $Ax = 0_{\mathbb{R}^n} \Rightarrow A^t Ax = 0_{\mathbb{R}^n}$ donc $x \in ker(A^t A)$.

Inversement ; pour

$$x \in ker(A^t A) \Leftrightarrow A^t Ax = 0_{\mathbb{R}^n} \Leftrightarrow x^t A^t Ax = 0_{\mathbb{R}} \Leftrightarrow \langle Ax, Ax \rangle = 0_{\mathbb{R}} \Leftrightarrow Ax = 0_{\mathbb{R}^n} \quad (1.61)$$

donc $x \in ker(A)$.

Finalement $ker(A) = ker(A^t A)$ donc

$$rg(A) = rg(A^t A). \quad (1.62)$$

D'autre part ; le rang d'une matrice diagonalisable est égale au nombre des éléments diagonaux non nuls donc le rang de $A^t A$ est égale au nombre de valeur propre $\lambda_i \neq 0$. Comme $\sigma_i = \sqrt{\lambda_i}$, donc le rang de $A^t A$ (qui est le rang de A) est égale au nombre de $\sigma_i \neq 0$.

• **preuve de B)**

De la décomposition en valeurs singulières, on a la famille (v_1, \dots, v_n) est une base de \mathbb{R}^n et $Av_i = \sigma_i u_i$ si $1 \leq i \leq r$.

Donc pour démontrer que $Im(A) = vect\{u_1, \dots, u_r\}$, il suffit de démontrer que $Im(A) = vect\{Av_1, \dots, Av_r\}$ car $\sigma_i \neq 0$ pour $1 \leq i \leq r$.

En fait, on va démontrer que (Av_1, \dots, Av_r) est une base orthogonale de $Im(A)$.

D'une part, les vecteurs v_i et $\lambda_j v_j$ sont orthogonaux, on a pour $i \neq j$

$$(Av_i)^t (Av_j) = v_i^t A^t Av_j = v_i^t \lambda_j v_j = 0. \quad (1.63)$$

Alors la famille (Av_1, \dots, Av_r) est orthogonale. Et comme $\|Av_i\| \neq 0$ ($Av_i \neq 0$ si $1 \leq i \leq r$) donc on déduit que les vecteurs (Av_1, \dots, Av_r) sont linéairement indépendants.

D'autre part, $vect\{Av_1, \dots, Av_r\} \subset Im(A)$ car $Av_i \in Im(A) \forall i$. Il reste à démontrer l'inclusion inverse.

Pour cela, soit $y \in Im(A)$ donc il existe $x \in \mathbb{R}^n$ telle que $y = Ax$ où x est lui même de la forme

$$x = c_1 v_1 + \dots + c_n v_n \quad (1.64)$$

on a alors ;

$$\begin{aligned} y = Ax &= c_1 Av_1 + \dots + c_r Av_r + c_{r+1} Av_{r+1} + \dots + c_n Av_n \\ &= c_1 Av_1 + \dots + c_r Av_r. \end{aligned} \quad (1.65)$$

car $c_i Av_i = c_i \sigma_i u_i = 0 \forall i = r+1, \dots, n$. Donc $y \in \text{vect}\{Av_1, \dots, Av_r\}$, ce qui montre que

$$(Av_1, \dots, Av_r) \text{ est une base de } \text{Im}(A) \quad (1.66)$$

D'après (1.66) on déduit que

$$\text{Im}(A) = \text{vect}\{Av_1, \dots, Av_r\} \quad (1.67)$$

• **preuve de C)**

On va démontrer que la famille $(v_{r+1}, v_{r+2}, \dots, v_n)$ est une base orthonormée de $\text{Ker}(A)$.

D'après le **théorème 1.2**, la famille (v_1, v_2, \dots, v_n) forme une base orthonormée de \mathbb{R}^n .

On sait que, toute sous famille d'une famille orthonormée est orthonormée donc la famille $(v_{r+1}, v_{r+2}, \dots, v_n)$ est orthonormée. D'où cette famille est libre.

Il reste à démontrer que $(v_{r+1}, v_{r+2}, \dots, v_n)$ est génératrice.

On a pour $x \in \text{ker}(A) \subset \mathbb{R}^n$

$$x = \sum_{i=1}^n \alpha_i v_i \text{ avec } \alpha_i \in \mathbb{R}.$$

En multipliant x par A , on obtient ;

$$\begin{aligned} 0 = Ax &= \sum_{i=1}^n A\alpha_i v_i \\ &= \sum_{i=1}^r A\alpha_i v_i + \sum_{i=r+1}^n A\alpha_i v_i. \end{aligned} \quad (1.68)$$

De la décomposition en valeurs singulières, $Av_i = \sigma_i u_i = 0$ pour $i = r+1, \dots, n$.

Donc

$$\sum_{i=1}^r A\alpha_i v_i = 0 \Leftrightarrow \sum_{i=1}^r \alpha_i \sigma_i u_i = 0 \quad (1.69)$$

Comme $rg(A) = r$ donc $\sigma_i \neq 0$ pour $1 \leq i \leq r$. Cela implique que les coefficients $\alpha_i = 0 \forall i, i = 1, \dots, r$.

Finalement, $x = \sum_{i=r+1}^n \alpha_i v_i \Rightarrow x \in \text{vect}\{v_{r+1}, \dots, v_n\}$.

Donc on a démontré que $\{v_i\}_{i=r+1, \dots, n}$ est une base de $\text{ker}(A)$. Alors $\text{ker}(A) = \text{vect}\{v_{r+1}, \dots, v_n\}$

• **preuve de D)**

On sait que $tr(AB) = tr(BA)$, que $UU^t = I_m$ et que $\|A\|_F^2 = tr(AA^t)$.

Comme

$$AA^t = U\Sigma^2 U^t \Leftrightarrow tr(AA^t) = tr((U\Sigma^2)U^t) = tr(U^t U\Sigma^2) \Leftrightarrow tr(AA^t) = tr(\Sigma^2). \quad (1.70)$$

Donc

$$tr(AA^t) = tr(\Sigma^2) \Leftrightarrow tr(AA^t) = \sum_{i=1}^r \sigma_i^2. \quad (1.71)$$

D'où $\|A\|_F^2 = \sum_{i=1}^r \sigma_i^2$.

• **preuve de E)**

Comme $A = U\Sigma V^t$ et $A^\dagger = V\Sigma^\dagger U^t$, on a alors

$$\begin{aligned} AA^\dagger &= U\Sigma V^t V\Sigma^\dagger U^t \Leftrightarrow AA^\dagger = UI_m U^t \\ &\Leftrightarrow AA^\dagger = I_m \end{aligned} \quad (1.72)$$

avec $\Sigma\Sigma^\dagger = I_m$ d'après l'hypothèse ($rg(A) = m$).

• **preuve de F)**

Comme $A = U\Sigma V^t$ et $A^\dagger = V\Sigma^\dagger U^t$, on a alors

$$\begin{aligned} A^\dagger A &= V\Sigma^\dagger U^t U \Sigma V^t \Leftrightarrow A^\dagger A = V I_n V^t \\ &\Leftrightarrow A^\dagger A = I_n \end{aligned} \quad (1.73)$$

avec $\Sigma^\dagger \Sigma = I_n$ d'après l'hypothèse ($rg(A) = n$).

• **preuve de G)**

Soit $A \in \mathcal{M}_{m,n}(\mathbb{R})$ et sa SVD $A = U\Sigma V^t$ avec $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_r)$,
où $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r \geq 0$ (Σ complétée par des zéros). On a

$$\begin{aligned} \|A\|_2 = \sqrt{\rho(A^t A)} &\Leftrightarrow \|A\|_2 = \sqrt{\max\{\lambda_i, i = \overline{1, r}\}} \\ &\Leftrightarrow \|A\|_2 = \sqrt{\max\{\sigma_i^2, i = \overline{1, r}\}} \\ &\Leftrightarrow \|A\|_2 = \sqrt{\sigma_1^2}. \end{aligned} \quad (1.74)$$

Donc $\|A\|_2 = \sigma_1$

• **preuve de H)** On sait que le conditionnement d'une matrice A , par rapport à la norme 2, est défini par :

$$\text{cond}_2(A) = \|A\|_2 \|A^{-1}\|_2. \quad (1.75)$$

Soit la SVD de la matrice A ; $A = U\Sigma V^t$. Comme A est inversible, alors $A^{-1} = A^\dagger = V\Sigma^\dagger U^t$. Il est facile de vérifier que

$$\|A^\dagger\|_2 = \|V\Sigma^\dagger U^t\|_2 = \|\Sigma^\dagger\|_2 = \frac{1}{\sigma_r}. \quad (1.76)$$

D'après la propriété D) et l'équation (1.76), on déduit que

$$\text{cond}_2(A) = \|A\|_2 \|A^{-1}\|_2 = \frac{\sigma_1}{\sigma_r} \quad (1.77)$$

■

Chapitre 2

Quelques applications de la SVD

1 Introduction

Le but de ce chapitre est d'appliquer la décomposition en valeurs singulières sur différents domaines (théorique et numérique). On va voir l'utilité de cette décomposition

- Dans la résolution des systèmes linéaires mal conditionnés.
- Dans la compression d'image par l'approximation de rang faible.
- Dans la protection de l'information par le tatouage numérique.

2 Résolution d'un système linéaire par la SVD

Il est fréquent, dans beaucoup de domaines scientifiques (en économie, en chimie, en théorie des graphes pour contrôler le flux dans les réseaux...), de devoir résoudre des systèmes linéaires rectangulaires du type

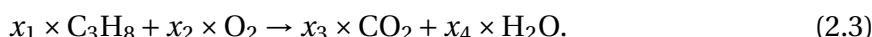
$$AX = Y, \tag{2.1}$$

où A est une matrice ($m \times n$) donnée, X est un vecteur inconnu de \mathbb{R}^n et Y est un vecteur donné de \mathbb{R}^m .

En chimie par exemple [6] (page 56), on est souvent appelé à équilibrer les équations des réactions chimiques et ceci suivant "la loi de la conservation de la matière" c'est à dire que "rien ne se crée et rien ne se perd". Cette opération conduit à la résolution d'un système linéaire qui peut être rectangulaire. Par exemple, quand on brûle du propane (C_3H_8), il se combine avec l'oxygène (O_2) pour former du dioxyde de carbone (CO_2) et de l'eau (H_2O). L'équation de cette réaction est



Pour équilibrer l'équation (2.2), on doit trouver des entiers x_1, x_2, x_3, x_4 tels que



On note par $(n_c, n_h, n_o)^t$ le vecteur dont les composantes sont respectivement le nombre de molécules de carbone, hydrogène et oxygène dans une substance.

C_3H_8 est représenté par le vecteur $(3, 8, 0)^t$.

O_2 est représenté par le vecteur $(0, 0, 2)^t$.

CO_2 est représenté par le vecteur $(1, 0, 2)^t$.

H_2O est représenté par le vecteur $(0, 2, 1)^t$.

Équilibrer l'équation (2.2), revient donc à trouver x_1, x_2, x_3, x_4 tels que

$$x_1 \begin{pmatrix} 3 \\ 8 \\ 0 \end{pmatrix} + x_2 \begin{pmatrix} 0 \\ 0 \\ 2 \end{pmatrix} = x_3 \begin{pmatrix} 1 \\ 0 \\ 2 \end{pmatrix} + x_4 \begin{pmatrix} 0 \\ 2 \\ 1 \end{pmatrix}. \quad (2.4)$$

Qui est équivalent au système rectangulaire

$$\begin{pmatrix} 3 & 0 & -1 & 0 \\ 8 & 0 & 0 & -2 \\ 0 & 2 & -2 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}. \quad (2.5)$$

On va voir que la résolution du système (2.1), nécessite la connaissance de la matrice pseudo inverse de A qui est notée A^\dagger et qui est donnée (sous certaines conditions) par

$$A^\dagger = (A^t A)^{-1} A^t, \quad (2.6)$$

car la solution de système (2.1) est $X^* = A^\dagger Y$.

Il est difficile de calculer directement cette matrice. Mais connaissant sa décomposition en valeurs singulières $A = U \Sigma V^t$, la tâche est facilitée, puisque l'inverse d'une matrice orthogonale est sa transposée. Donc l'expression de A^\dagger est simplement donnée par :

$$A^\dagger = V \Sigma^\dagger U^t,$$

avec Σ^\dagger la pseudo inverse de Σ où tout coefficient non nul est remplacé par son inverse.

2.1 Généralisation de la notion de solution

On veut résoudre le problème suivant :

$$\left\{ \begin{array}{l} \text{Trouver } X \in \mathbb{R}^n \text{ solution de} \\ AX = Y, \end{array} \right. \quad (2.7)$$

avec A une matrice de $\mathbb{R}^{m \times n}$ et Y un vecteur de \mathbb{R}^m donnés.

- Dans le cas où $n = m$ et A est inversible, on sait qu'il existe une unique solution $X^* \in \mathbb{R}^n$ du système (2.7), caractérisée par $X^* = A^{-1}Y$.

- Dans le cas où $n \neq m$, pour se ramener au cas des systèmes avec matrice carrée, on multiplie l'équation de (2.7) par A^t . On se retrouve à résoudre le système carré d'ordre n (dit normal)

$$\left\{ \begin{array}{l} \text{Trouver } X \in \mathbb{R}^n \text{ solution de} \\ A^t A X = A^t Y, \end{array} \right. \quad (2.8)$$

Si $A^t A$ est inversible, (c'est le cas si $\text{rg}(A)=n$), alors la solution du système normal est caractérisée par

$$X^* = (A^t A)^{-1} A^t Y. \quad (2.9)$$

- Dans les autres cas, afin de résoudre le problème (2.7), on est amené à généraliser la notion de solution.

Définition 2.1 On dit que le vecteur $X^* \in \mathbb{R}^n$ est solution au sens des moindres carrés du système (2.7) s'il est solution du problème d'optimisation suivant

$$\|AX^* - Y\| = \min_{X \in \mathbb{R}^n} \frac{1}{2} \|AX - Y\|. \quad (2.10)$$

Remarque 2.1 Si A est inversible, alors la solution $X^* = A^{-1}Y$ vérifie le problème (2.10) car $\|AX^* - Y\| = 0$.

L'existence de la solution du problème d'optimisation c'est à dire le problème (2.10) est assurée par le théorème qui suit où on pose $F(X) = \frac{1}{2} \|Y - AX\|_2^2$:

Remarque 2.2 On a

$$\begin{aligned} F(X) &= \frac{1}{2} \|Y - AX\|_2^2 \\ &= \frac{1}{2} \langle Y - AX, Y - AX \rangle \\ &= \frac{1}{2} (\langle X, A^t A X \rangle - 2 \langle A^t Y, X \rangle + \langle Y, Y \rangle). \end{aligned} \quad (2.11)$$

Comme F est une forme quadratique alors elle est différentiable. De plus, la matrice $A^t A$ symétrique et semi-définie positive alors le gradient de cette forme quadratique est donné par

$$\nabla F(X) = A^t A X - A^t Y \quad (2.12)$$

Théorème 2.1 [1](page 51) Si F est une fonction convexe continument différentiable, une condition nécessaire et suffisante pour que X^* soit un optimum globale de F sur \mathbb{R}^n est que $\nabla F(X^*) = 0$. Autrement dit, dans le cas convexe, la stationnarité à elle seule constitue une condition nécessaire et suffisante d'optimalité globale.

Du fait que le problème d'optimisation (2.10) admet une solution, cela assure l'existence de la solution du système linéaire (2.7).

2.2 Expression de la solution

On va distinguer deux cas. On note par r le rang de la matrice A .

La matrice A est de rang complet ; $r = \min(m, n)$

Dans ce cas, on considère deux possibilités :

1. Si $n < m$, on est dans le cas sur-déterminé autrement dit on a plus d'équations que d'inconnues.

Pour trouver la solution X^* du problème (2.10), il suffit, selon (2.12), de résoudre l'équation :

$$\begin{aligned} \nabla F(X^*) = 0 &\Leftrightarrow -A^t(Y - AX^*) = 0 \\ &\Leftrightarrow -A^t Y + A^t A X^* = 0 \\ &\Leftrightarrow X^* = (A^t A)^{-1} A^t Y. \end{aligned} \quad (2.13)$$

Donc la solution de $AX = Y$ au sens des moindres carrés, est caractérisée par

$$X^* = (A^t A)^{-1} A^t Y. \quad (2.14)$$

Il reste à calculer $(A^t A)^{-1} A^t$. Connaissant la SVD de A , on a

$$A^t A = V \Sigma^t \Sigma V^t. \quad (2.15)$$

En notant

$$\Sigma^\dagger = \begin{pmatrix} \frac{1}{\sigma_1} & & & 0 & \dots & 0 \\ & \frac{1}{\sigma_2} & 0 & \vdots & & \vdots \\ & 0 & \ddots & \vdots & & \vdots \\ & & & \frac{1}{\sigma_n} & 0 & \dots & 0 \end{pmatrix},$$

on peut exprimer l'inverse de $A^t A$ par

$$(A^t A)^{-1} = V \Sigma^\dagger (\Sigma^t)^\dagger V^t. \quad (2.16)$$

Par multiplication par A^t , on obtient enfin l'expression de la solution

$$X^* = A^\dagger Y \text{ avec } A^\dagger = V \Sigma^\dagger U^t. \quad (2.17)$$

Il est à remarquer que Σ est de taille $(m \times n)$ alors que son pseudo-inverse est de taille $(n \times m)$.

Exemple 2.1 On prend

$$A = \begin{pmatrix} 1 & 2 \\ 4 & 0 \\ 3 & 2 \end{pmatrix} \text{ et } Y = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$$

On remarque que $rg(A) = n = 2$, donc on peut résoudre ce système par trois méthodes
 – Normalisation du système : On multiplie le système $AX = Y$ par la matrice A^t , on obtient l'équation normale suivante

$$A^t AX = A^t Y, \text{ avec } (A^t A) \in \mathbb{R}^2 \times \mathbb{R}^2. \quad (2.18)$$

Puisque A est de rang plein ($rg(A) = 2$) donc $rg(A^t A) = 2$ c'est à dire $(A^t A)^{-1}$ existe, donc on calcule la quantité suivante $((A^t A)^{-1} A^t)$

$$(A^t A)^{-1} A^t = \begin{pmatrix} -0.0556 & 0.2222 & 0.556 \\ 0.3056 & -0.2222 & 0.1944 \end{pmatrix}. \quad (2.19)$$

La solution est donnée par

$$X = (A^t A)^{-1} A^t Y = \begin{pmatrix} 0 \\ \frac{1}{2} \end{pmatrix}. \quad (2.20)$$

– La méthode des moindres carrés : On cherche le vecteur X^* qui minimise $\|Y - AX\|$. D'après (2.13) la solution de $AX = Y$ au sens de moindres carrés est

$$X^* = (A^t A)^{-1} A^t Y = \begin{pmatrix} 0 \\ \frac{1}{2} \end{pmatrix}. \quad (2.21)$$

– La pseudo inverse : On calcule la matrice A^\dagger

$$A^\dagger = \begin{pmatrix} -0.0556 & 0.2222 & 0.556 \\ 0.3056 & -0.2222 & 0.1944 \end{pmatrix}. \quad (2.22)$$

Et on retrouve la même solution

$$X^* = A^\dagger Y = \begin{pmatrix} 0 \\ \frac{1}{2} \end{pmatrix}. \quad (2.23)$$

Remarque 2.3

- (a) $(A^t A)^{-1}$ existe si A est de rang complet c'est à dire $rg(A) = n$.
- (b) En particulier si $Y \in Im(A)$ alors la solution X^* existe.
- (c) L'erreur, $\|Y - AX^*\| = 0$. (X^* est la solution exacte du système normal).

2. Si $m < n$, on est dans le cas sous-déterminé, c'est à dire le nombre d'inconnues est supérieur au nombre d'équations. Dans ce cas, on a une infinité de solutions. Pour assurer l'unicité, on cherche la solution X^* qui minimise la norme de tous les vecteurs X qui réalise $AX = Y$.

Le théorème suivant assure l'existence et l'unicité de cette solution et donne son expression.

Théorème 2.2 [5](page 200)

L'unique solution X^* de $AX = Y$ qui minimise la norme $\|X\|$ est donnée par

$$X^* = A^t(AA^t)^{-1}Y. \tag{2.24}$$

Preuve.

Comme il y a une infinité de solutions, l'idée est de trouver parmi tous les vecteurs X qui réalisent $AX = Y$, celui qui a la norme la plus petite. On transforme donc ce problème ($AX = Y$) en un problème d'optimisation avec contrainte.

$$\left\{ \begin{array}{l} \text{Trouver } X \text{ solution de } \min_{X \in \mathbb{R}^n} \|X\| \\ AX = Y. \end{array} \right. \tag{2.25}$$

Toutefois, minimiser $\|X\|$ revient à minimiser $\frac{1}{2}\|X\|^2$, ce qui nous amène au problème équivalent

$$\left\{ \begin{array}{l} \text{Trouver } X \text{ solution de } \min_{X \in \mathbb{R}^n} \frac{1}{2}\|X\|^2 \\ AX = Y. \end{array} \right. \tag{2.26}$$

Le système (2.26) est équivalent, par le principe de relaxation Lagrangienne, à

$$\min_{X \in \mathbb{R}^n} \left[\frac{1}{2}\|X\|^2 + \lambda^t(Y - AX) \right] = \min_{X \in \mathbb{R}^n} \left[\frac{1}{2}X^t X + \lambda^t(Y - AX) \right], \tag{2.27}$$

où λ est un multiplicateur de Lagrange (un vecteur à déterminer).

On pose $G(X) = \frac{1}{2}X^t X + \lambda^t(Y - AX)$.

Pour trouver la solution X^* , on résout l'équation suivante :

$$\begin{aligned} \nabla G(X^*) = 0 &\Leftrightarrow X^* - A^t \lambda = 0 \\ &\Leftrightarrow X^* = A^t \lambda. \end{aligned} \tag{2.28}$$

Afin de déterminer λ , on multiplie (2.28) par A , on obtient

$$Y = AA^t \lambda \Leftrightarrow \lambda = (AA^t)^{-1}Y. \tag{2.29}$$

La substitution de λ dans (2.28) donne :

$$X^* = A^t(AA^t)^{-1}Y. \tag{2.30}$$

Il reste à calculer $A^t(AA^t)^{-1}$.

Connaissant la SVD de A, on a

$$AA^t = U\Sigma\Sigma^tU^t. \quad (2.31)$$

En notant

$$\Sigma^\dagger = \begin{pmatrix} \frac{1}{\sigma_1} & & & \\ & \frac{1}{\sigma_2} & 0 & \\ & 0 & \ddots & \\ & & & \frac{1}{\sigma_m} \\ 0 & \dots & \dots & 0 \\ \vdots & & & \vdots \\ 0 & \dots & \dots & 0 \end{pmatrix},$$

on a

$$(AA^t)^{-1} = U(\Sigma^\dagger)^\dagger \Sigma^\dagger U^t. \quad (2.32)$$

Par multiplication par A^t , on obtient

$$X^* = A^\dagger Y \text{ avec } A^\dagger = V\Sigma^\dagger U^t. \quad (2.33)$$

■

Exemple 2.2 On prend

$$A = \begin{pmatrix} 2 & -1 & 0 & 1 \\ 3 & 1 & 4 & 1 \\ 6 & 2 & 1 & 3 \end{pmatrix} \text{ et } Y = \begin{pmatrix} 2 \\ -1/2 \\ 3/2 \end{pmatrix}. \quad (2.34)$$

On remarque que $r = m = 3$, on ne peut pas résoudre le système $AX = Y$ par l'équation normale car la matrice A^tA est de taille (4×4) mais de rang 3, donc cette matrice n'est pas inversible alors on a que deux possibilités pour résoudre ce système

– La méthode des moindres carrés : En suivant les étapes (2.26) - (2.30), on obtient

$$X^* = A^t(AA^t)^{-1}Y = \begin{pmatrix} 0.1657 & -0.0057 & 0.0800 \\ -0.6019 & -0.559 & 0.2267 \\ 0.0095 & 0.2794 & -0.1333 \\ 0.0667 & -0.0444 & 0.0667 \end{pmatrix} \begin{pmatrix} 2 \\ -1/2 \\ 3/2 \end{pmatrix} = \begin{pmatrix} 0.4543 \\ -0.8359 \\ -0.3206 \\ 0.2556 \end{pmatrix} \quad (2.35)$$

– La pseudo inverse : On calcule A^\dagger

$$A^\dagger = \begin{pmatrix} 0.1657 & -0.0057 & 0.0800 \\ -0.6019 & -0.559 & 0.2267 \\ 0.0095 & 0.2794 & -0.1333 \\ 0.0667 & -0.0444 & 0.0667 \end{pmatrix}. \quad (2.36)$$

Donc la solution est simplement donnée par

$$X^* = A^\dagger Y = \begin{pmatrix} 0.4543 \\ -0.8359 \\ -0.3206 \\ 0.2556 \end{pmatrix} \quad (2.37)$$

Remarque 2.4

(a) $(AA^t)^{-1}$ existe si A est de rang complet c'est à dire $rg(A) = m$.

(b) Pour cet exemple, l'erreur $\|Y - AX^*\| = 3 \times 10^{-4}$.

La matrice A n'est pas de rang complet ; $r < \min(m, n)$

Dans ce cas, la seule méthode pour résoudre le système (2.7) est la décomposition en valeurs singulières car la SVD est la seule méthode à fournir la pseudo-inverse $A^\dagger = V\Sigma^\dagger U^t$.

Tout d'abord, on transforme le système (2.7) en un problème d'optimisation du type (2.10).

On a le résultat

Théorème 2.3 [5](page 210)

On considère un système d'équations linéaires $AX = Y$ avec $A \in \mathbb{R}^{m \times n}$ et $rg(A) = r$. Le vecteur $X^* = A^\dagger Y$ minimise $\|AX - Y\|^2$ sur \mathbb{R}^n . Parmi tous les vecteurs de \mathbb{R}^n qui minimisent $\|AX - Y\|^2$, le vecteur $X^* = A^\dagger Y$ est l'unique vecteur de norme minimale.

Pour démontrer ce théorème, on a besoin des deux résultats suivants

Lemme 2.1 [5](page 205)

Soit A une matrice de $\mathbb{R}^{m \times n}$ de $rg(A) = r < \min(m, n)$. Alors il existe deux matrices de rang r, $B \in \mathbb{R}^{m \times r}$ et $C \in \mathbb{R}^{r \times n}$ telles que

$$A = BC. \tag{2.38}$$

Théorème 2.4 [5](page 208)

Soit la matrice $A \in \mathbb{R}^{m \times n}$ qui admet une factorisation $A = BC$, avec $rg(A) = rg(B) = rg(C) = r$, et $B \in \mathbb{R}^{m \times r}$ et $C \in \mathbb{R}^{r \times n}$ alors

$$A^\dagger = C^\dagger B^\dagger. \tag{2.39}$$

où $C^\dagger = C^t(CC^t)^{-1}$ et $B^\dagger = (B^tB)^{-1}B^t$.

Preuve. du théorème(2.3) [5](page 210)

On pose $X^* = A^\dagger Y$. On veut démontrer que X^* réalise le minimum de $\|Y - AX\|^2$.

Pour tout $X \in \mathbb{R}^n$, cette dernière prend la forme

$$\begin{aligned} \|Y - AX\|^2 &= \|Y - AX + AX^* - AX^*\|^2 \\ &= \|Y - AX^* + A(X - X^*)\|^2 \\ &= \|A(X - X^*)\|^2 + \|Y - AX^*\|^2 + 2[A(X - X^*)]^t(Y - AX^*). \end{aligned} \tag{2.40}$$

Il est facile de vérifier que

$$[A(X - X^*)]^t(Y - AX^*) = 0. \tag{2.41}$$

En effet, par substitution de $X^* = A^\dagger Y$, on a

$$\begin{aligned} [A(X - X^*)]^t(Y - AX^*) &= (X - X^*)^t(A^tY - A^tAX^*) \\ &= (X - X^*)^t(A^tY - A^tAA^\dagger Y) \\ &= (X - X^*)^t[A^t - A^tAA^\dagger]Y. \end{aligned} \tag{2.42}$$

C'est ce qui est entre crochets qui s'annule car

$$A^tAA^\dagger = C^tB^tBCC^t(CC^t)^{-1}(B^tB)^{-1}B^t = A^t. \tag{2.43}$$

Donc

$$\|Y - AX\|^2 = \|A(X - X^*)\|^2 + \|Y - AX^*\|^2. \tag{2.44}$$

Puisque $\|A(X - X^*)\|^2 \geq 0$, alors

$$\|Y - AX\|^2 \geq \|Y - AX^*\|^2. \tag{2.45}$$

Donc $X^* = A^\dagger Y$ est le vecteur qui réalise le minimum de $\|Y - AX\|^2$.

■

Exemple 2.3 On prend

$$A = \begin{pmatrix} 2 & 0 & 0 \\ 1 & 1 & 2 \\ 0 & 3 & 6 \\ 1 & 2 & 4 \end{pmatrix} \text{ et } Y = \begin{pmatrix} 1 \\ 1 \\ 4/5 \\ 2 \end{pmatrix}. \quad (2.46)$$

Le rang de A $r = 2 < \min(4, 3)$, dans ce cas on ne peut pas résoudre ce système par la méthode des moindres carrés ($(AA^t)^{-1}$ n'existe pas), ni en passant par les équations normales ($(A^tA)^{-1}$ n'existe pas), donc on prend le chemin unique qui est le calcul direct de A^\dagger

$$A^\dagger = \begin{pmatrix} 0.3733 & 0.1467 & -0.1200 & 0.1067 \\ -0.0160 & 0.0080 & 0.0480 & 0.0240 \\ -0.0320 & 0.0160 & 0.0960 & 0.0480 \end{pmatrix}, \quad (2.47)$$

donc la solution de ce système est :

$$X^* = A^\dagger Y = \begin{pmatrix} 0.4907 \\ 0.0704 \\ 0.1408 \end{pmatrix} \quad (2.48)$$

Remarque 2.5

1. Pour cet exemple, l'erreur $\|Y - AX^*\| = 0.8597$.
2. D'après ces résultats, on voit que si la matrice A est de rang plein, alors on peut résoudre le système linéaire ($AX = Y$) par les deux méthodes (moindres carrés et pseudo inverse). Par contre si la matrice A n'est pas de rang plein, on ne peut résoudre le système que par la pseudo inverse.
3. En résumé, quelque soit la nature de la matrice A , la solution du système (2.7) est donnée par $X = A^\dagger Y$.

3 L'image numérique

Avant de citer les autres applications, on va donner quelques définitions sur les images.

3.1 Définition de l'image

L'image est une représentation visuelle ou mentale d'un objet ou d'une action. Il existe deux types d'images

L'image analogique (continue), c'est celle qui est liée à un support matériel comme la radiographie, la peinture, Elle peut être représentée par une fonction continue, c'est à dire il n'est pas possible de la reproduire à l'identique par la machine.

L'image numérique (discrète) est représentée par un ensemble fini de points appelés pixels (contraction des mots anglais "picture element", c'est à dire élément d'image).

Les pixels sont situés sur une grille, à chaque pixel de la grille est associé une couleur ou une nuance de gris. Le passage d'une image continue à une grille de pixels s'appelle l'échantillonnage.

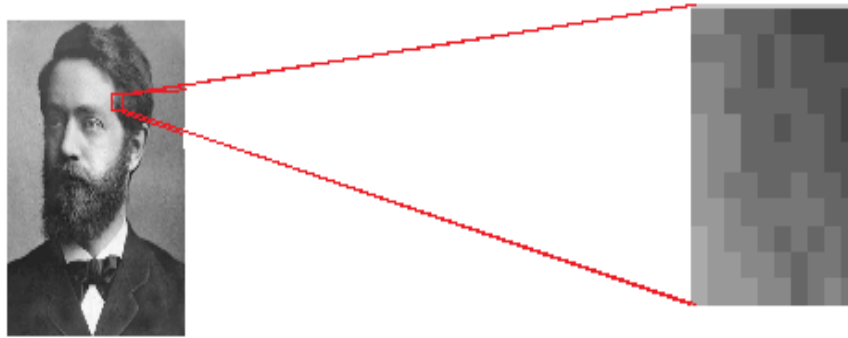


FIGURE 2.1 – Échantillonnage d'une partie d'image.

Autrement dit, l'image numérique (en deux dimensions) est un tableau (matrice), chaque élément de cette matrice représente un pixel et contient des valeurs entières pour les images en niveau de gris, cette opération est appelé la quantification. La figure (2.2) illustre ça.



FIGURE 2.2 – Quantification des niveaux de gris.

La quantification de la partie d'image de la figure (2.1) est représentée dans la figure suivante

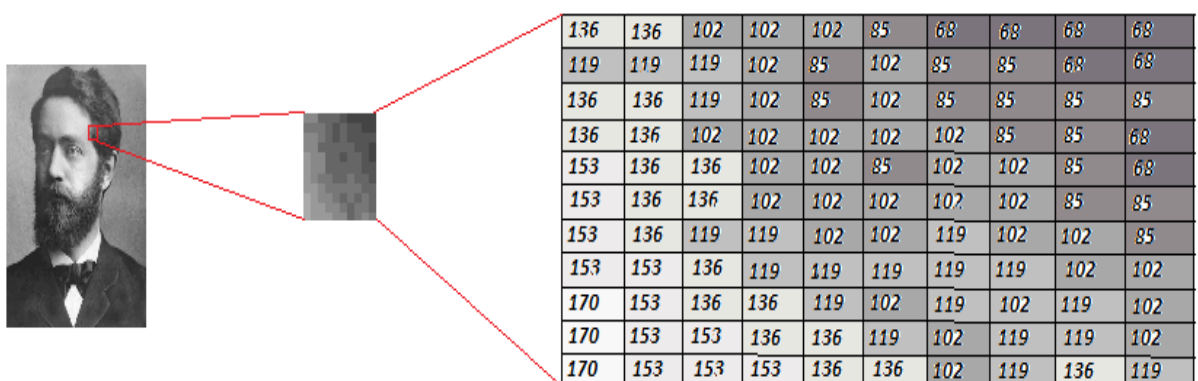


FIGURE 2.3 – Quantification de la partie d'image échantillonnée.

Les commandes sur MatLab qui réalisent la représentation d'une image par une matrice et vis versa, est donnée par le schéma suivant

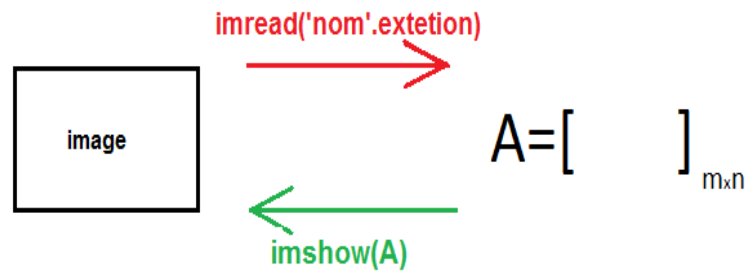


FIGURE 2.4 – Transformation d'une image en matrice

Pour pouvoir stocker et transmettre cette image comme n'importe quelle autre donnée informatique, il faut la coder en binaire, c'est-à-dire la décrire par une suite de 0 et de 1. Autrement dit on remplace chaque valeur entière de niveau de gris par son code en binaire(en base 2). Cette opération est appelée codage de l'image.

3.2 Opérations sur les images

Comme une image est caractérisée par une matrice réelle, alors on peut faire beaucoup d'opérations sur cette image.

Soient Im_1 et Im_2 deux images représentées respectivement par les matrices A_1 et A_2 de même taille.

L'addition de deux images :

La somme des deux images Im_1 et Im_2 est obtenue par la somme des deux matrices $A_1 + A_2$. Le résultat est la superposition des deux images.

La soustraction de deux images :

La soustraction de deux images est définie par $(A_1 - A_2)$. Elle est utilisée pour détecter les différences qu'il y a entre les images Im_1 et Im_2 .

La multiplication d'une image par un scalaire :

Le résultat de la multiplication de la matrice A_1 par un scalaire M ($M * A_1$) est l'image Im_1 avec des nuances différentes du niveau de gris.

La division de deux images :

La division de l'image Im_1 par l'image Im_2 est définie par la division matricielle $A_1 ./ A_2$. Elle est utilisée pour supprimer l'ombre dans une image.

La multiplication de deux images :

La multiplication de deux images Im_1 et Im_2 est définie par $A_1 * A_2$

La transposée d'une image :

Le résultat de la transposée de la matrice A_1 est l'image Im_1 orientée d'un angle $\frac{\pi}{2}$ dans le sens positif.

La rotation d'une image :

La rotation dans le sens des aiguilles de la montre de l'image Im_1 représentée par la matrice $(A_1)_{i,j}$ pour $i = 1 : m; j = 1 : n$, est une image notée Im_r représentée par la matrice $(A_r)_{j,i}$ et qui est définie par

$$A_r(j, i) = A_1(m - i + 1, j) \text{ pour } j = 1 : n \text{ et } i = 1 : m. \quad (2.49)$$

4 Approximation de rang faible d'une matrice

La compression d'image est une application de la compression des données sur des images numériques. Cette compression a pour utilité de réduire la redondance des données d'une image afin de pouvoir l'emmagasiner sans occuper beaucoup d'espace ou la transmettre rapidement. L'un des types de compression est l'approximation de rang faible. Elle consiste à trouver une matrice \tilde{A} de même taille que A (A étant la matrice associée à l'image originale) mais de rang inférieur strictement au rang de la matrice A .

Cette matrice c'est à dire \tilde{A} existe, et est liée à la SVD de A comme le stipule le théorème d'Eckart-Young .

Théorème 2.5 (Eckart-Young-Mirsky) [2](page 77)

Soit $A \in \mathcal{M}_{m,n}(\mathbb{R})$ et soit sa décomposition en valeurs singulières donnée par $A = U \Sigma V^t$.
Si $k < r$ avec $r = \text{rg}(A)$, la matrice définie par

$$A_k = \sum_{i=1}^k \sigma_i u_i v_i^t, \quad (2.50)$$

vérifie

$$\min_{\text{rg}(B)=k} \|A - B\|_2 = \|A - A_k\|_2 = \sigma_{k+1}. \quad (2.51)$$

D'après ce théorème, parmi toute les matrices d'ordre k , la matrice A_k définie par (2.50) est la plus proche de la matrice A au sens de la norme 2.

Preuve.

Soit k un entier choisi.

- On remarque que si $k \geq r$, $A_k = U \Sigma V^t = A$, d'où $\|A - A_k\|_2 = 0$.
- Dans le cas $k < r$:

I- On démontre que $\|A - A_k\|_2 = \sigma_{k+1}$:

Pour faciliter les notations, on écrira $A_k = U_k \Sigma_k V_k^t$.

$A_{i:j}$ désigne la restriction de la matrice A aux colonnes de i à j .

En scindant les colonnes de U et V^t en deux blocs (selon la valeur de k) et Σ en quatre blocs, l'expression de la matrice A devient

$$\begin{aligned} A &= (U_{1:k} \quad U_{k+1:r}) \begin{pmatrix} \Sigma_{1:k} & 0 \\ 0 & \Sigma_{k+1:r} \end{pmatrix} \begin{pmatrix} V_{1:k}^t \\ V_{k+1:r}^t \end{pmatrix} \\ &= U_{1:k} \Sigma_{1:k} V_{1:k}^t + U_{k+1:r} \Sigma_{k+1:r} V_{k+1:r}^t \end{aligned} \quad (2.52)$$

et celle de A_k devient

$$\begin{aligned} A_k &= (U_{1:k} \quad 0) \begin{pmatrix} \Sigma_{1:k} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} V_{1:k}^t \\ 0 \end{pmatrix} \\ &= U_{1:k} \Sigma_{1:k} V_{1:k}^t \end{aligned} \quad (2.53)$$

Alors

$$\begin{aligned} A - A_k &= U_{1:k} \Sigma_{1:k} V_{1:k}^t + U_{k+1:r} \Sigma_{k+1:r} V_{k+1:r}^t - U_{1:k} \Sigma_{1:k} V_{1:k}^t \\ &= U_{k+1:r} \Sigma_{k+1:r} V_{k+1:r}^t \end{aligned} \quad (2.54)$$

On rappelle que

$$\|A - A_k\|_2 = \sqrt{\rho[(A - A_k)(A - A_k)^t]}$$

Or, comme $V_{k+1:r}^t V_{k+1:r} = I_n$ alors

$$(A - A_k)(A - A_k)^t = U_{k+1:r} \Sigma_{k+1:r}^2 U_{k+1:r}^t$$

et donc

$$\|A - A_k\|_2 = \sqrt{\sigma_{k+1}^2} = \sigma_{k+1}$$

II- Il reste à démontrer que $\min_{rg(B)=k} \|A - B\|_2 = \|A - A_k\|_2$. Pour cela on procède par l'absurde c'est à dire :

on suppose qu'il existe une matrice B de rang k telle que $\|A - B\|_2 < \|A - A_k\|_2$. Soit $x \in \mathbb{R}^n$ et $x \neq 0$. On sait que $\|(A - B)x\| \leq \|A - B\|_2 \|x\|$ (d'après la propriété de la norme).

Alors $\|(A - B)x\| < \|A - A_k\|_2 \|x\| = \sigma_{k+1} \|x\|$.

Si on prend $y \in \ker(B)$ (sous espace de \mathbb{R}^n) $y \neq 0$, alors

$$\|Ay\| = \|Ay - By\| = \|(A - B)y\| < \sigma_{k+1} \|y\| \quad (2.55)$$

Si on prend aussi $z \in \text{Vect}(v_1, v_2, \dots, v_{k+1})$ (sous espace de \mathbb{R}^n), donc z s'écrit sous la forme suivant : $z = V_{1:k+1} \alpha$, avec $\alpha \in \mathbb{R}^{k+1}$. Alors d'une part on a :

$$\begin{aligned} \|Az\|^2 &= \|U \Sigma V_{1:k+1}^t \alpha\|^2 \\ &= \|U \Sigma \begin{pmatrix} V_{1:k+1}^t \\ V_{k+2:r}^t \end{pmatrix} \alpha\|^2 \\ &= \|U_{1:k+1} \Sigma_{1:k+1} \begin{pmatrix} I_{1:k+1} \\ 0 \end{pmatrix} \alpha\|^2 \\ &= \alpha^t \Sigma_{1:k+1} U_{1:k+1}^t U_{1:k+1} \Sigma_{1:k+1} \alpha \\ &= \alpha^t \Sigma_{1:k+1}^2 \alpha \\ &= \sum_{i=1}^{k+1} \alpha_i^2 \sigma_i^2 \geq \sum_{i=1}^{k+1} \alpha_i^2 \sigma_{k+1}^2 = \|\alpha\|^2 \sigma_{k+1}^2 \end{aligned}$$

d'autre part

$$\|z\|^2 = \|V_{1:k+1} \alpha\|^2 = \alpha^t V_{1:k+1}^t V_{1:k+1} \alpha = \alpha^t \alpha = \|\alpha\|^2$$

Donc

$$\|Az\| \geq \sigma_{k+1} \|z\|. \quad (2.56)$$

De (2.55) et (2.56) on déduit que $\ker(B)$ et $\text{Vect}(v_1, v_2, \dots, v_{k+1})$ sont deux espaces disjoints de \mathbb{R}^n c.à.d $\ker(B) \cap \text{Vect}(v_1, v_2, \dots, v_{k+1}) = \emptyset$.

De plus, du théorème de rang on sait que $n = rg(B) + \dim(\ker(B)) \Rightarrow \dim(\ker(B)) = n - k$.

Et comme $\dim(\text{Vect}(v_1, v_2, \dots, v_{k+1})) = k + 1$, donc

$\dim(\ker B) + \dim(\text{Vect}(v_1, v_2, \dots, v_{k+1})) = n + 1$ contradiction avec le fait que $\dim(\mathbb{R}^n) = n$.

donc

$$\min_{rg(B)=k} \|A - B\|_2 = \|A - A_k\|_2.$$

Le théorème de Eckart-Young-Mirsky est prouvé ■

On vient ainsi de démontrer que la meilleure approximation de rang faible d'une matrice au sens de la norme 2 est une SVD tronquée de cette matrice.

Le chapitre suivant illustre différents aspects de cette approximation de rang faible.

5 Le tatouage visible et invisible

5.1 Introduction

Définition 2.2 *Le tatouage consiste à insérer une marque¹ appelée aussi signature, ou tatouage, dans une image (appelé image hôte) ou dans d'autres documents numériques. Il peut être visible ou invisible.*

Le tatouage numérique est un domaine scientifique récent apparu au début des années 90 (avec l'explosion du partage de toute sorte de documents numériques).

Historiquement, le tatouage est apparu sur du papier il y a presque 700 ans. Le plus ancien document tatoué trouvé remonte à 1292 à Fabriano en Italie. Vers la fin de ce siècle, 40 fabricants se partageaient le marché du papier. Pour connaître la provenance de ce dernier, le tatouage était la méthode utilisée.

Quand au tatouage d'image numérique, les premiers travaux publiés reviennent à Tanaka et al.[4](page 12) en 1990. Depuis 1995, le tatouage numérique a gagné beaucoup d'attention et a très rapidement évolué. On compte une vingtaine de publications sur le tatouage numérique en 1995 alors que dix ans après, on en compte plus de deux milles [4](page 13).

Le but du tatouage numérique est la lutte contre la fraude, le piratage informatique et la protection des droits d'auteurs. La marque insérée dans le cas visible, est essentiellement un logo comme cela est le cas pour les sociétés, les banques, les universités, Le tatouage visible est obtenu par une simple addition des deux matrices associées aux deux images ; l'image hôte et l'image marque.

Dans le cas invisible, la marque insérée doit être connue uniquement par le propriétaire ou par le diffuseur. C'est ce qui permet de protéger de document numérique avant de le transmettre. Cette marque doit être imperceptible par le système visuel humain et ne doit pas affecter la qualité de l'image originale. L'image tatouée doit être visuellement équivalente à l'image originale.

La décomposition en valeurs singulières (SVD) est un outil mathématique très utilisé dans le tatouage d'image numériques.

5.2 Le tatouage invisible

Le tatouage invisible est l'approche la plus développée qui attire la plupart des chercheurs. La majorité des techniques concernant la protection de propriété intellectuelle suit la branche du tatouage invisible.

Il y a plusieurs algorithmes de tatouage invisible basés sur la décomposition en valeurs singulières et parmi lesquels, l'algorithme de Chandra.

Algorithme Global-SVD de Chandra (AC-SVD)

D.V.Satish Chandra propose cet algorithme tel que la SVD est appliquée sur toute l'image hôte et la marque, ces dernières ont la même taille. Les valeurs singulières de la marque sont multipliées par un scalaire et ensuite ajoutées aux valeurs singulières de l'image hôte. L'image tatouée est ainsi obtenue. Pour l'extraction de la marque à partir de l'image tatouée, on suit le chemin inverse comme cela est explicité dans l'algorithme [3].

Algorithme d'insertion

L'objectif de cet algorithme est l'insertion d'une marque dans une image hôte.

1. la marque :watermark

Entrées :

- f : image hôte de taille $m \times n$.
- W : la marque insérée de taille $m \times n$.
- α : la force de tatouage(on choisit α pour maintenir une bonne qualité de l'image tatouée).

Sorties :

- f_t : image tatouée.
- Les matrices :
 - Σ : matrice diagonale de l'image hôte.
 - U_w et V_w : matrices orthogonales de la marque.

Les étapes

1. La décomposition de f en valeurs singulières :

$$f = U \Sigma V^t$$

2. La décomposition de W en valeurs singulières :

$$W = U_w \Sigma_w V_w^t$$

3. La construction d'une nouvelle matrice diagonale Σ_t dont les valeurs diagonales sont :

$$\sigma_{t_i} = \sigma_i + \alpha \sigma_{w_i}$$

où σ_i sont les éléments diagonaux de Σ et σ_{w_i} les éléments diagonaux de Σ_w .

4. La construction de l'image tatouée f_t en utilisant Σ_t et les matrices orthogonales (U, V) de l'image originale

$$f_t = U \Sigma_t V^t.$$

Algorithme d'extraction

L'objectif de cet algorithme est de récupérer la marque insérée (cachée).

Entrées :

- f_t : image tatouée
- Les matrices :
 - Σ : matrice diagonale de l'image hôte.
 - U_w et V_w : matrices orthogonales de la marque.

Sorties :

- W : la marque .

Les étapes

1. la décomposition de f_t en valeurs singulières :

$$f_t = U_1 \Sigma_1 V_1^t$$

2. Calcul de la matrice diagonale Σ_w :

$$\Sigma_w = \frac{\Sigma_1 - \Sigma}{\alpha}$$

3. La construction de la marque W en utilisant Σ_w , U_w et V_w comme suit :

$$W = U_w \Sigma_w V_w^t.$$

Remarque 2.6 *Il existe d'autre versions de cet algorithme.*

Première version : utilisation de la matrice U

L'image tatouée peut être aussi obtenue par $f_t = U_t \Sigma V^t$ où $U_t = U + \alpha U_w$.

Deuxième version : utilisation de la matrice V

L'image tatouée peut être obtenue par $f_t = U \Sigma V_t^t$ avec $V_t = V + \alpha V_w$.

Chapitre 3

Les résultats expérimentaux

1 Introduction

La compression d'image et le tatouage numérique sont basés sur des techniques algorithmiques. On utilise langage MatLab pour faire des expériences sur différents images.

- Pour la compression d'image, on fait une expérience visuelle sur une image en niveau de gris représentant le portrait de Felix Klein ¹.
- Et pour le tatouage numérique, on fait deux expériences pour le tatouage : visible et invisible sur deux images (image en niveau de gris représentant le portrait de Abdelhamid Ibn Badis et image binaire [logo binaire de l'université de Abdelhamid Ibn Badis]).

Mais avant cela, on va voir, via des exemples l'effet de certaines opérations sur les images.

Des programmes (pour la compression de l'image, le tatouage numérique et les différents opérations) ont été élaborés en langage MatLab sur un PC de type LENOVO, Processeur Intel(R) Core(TM) i3, 2.00Ghz, RAM 4 Go.

2 Les opérations sur les images

On fait des expériences pour quelques opérations sur les deux images [l'image de Abdelhamid Ibn Badis] et [logo binaire de l'université de Abdelhamid Ibn Badis].



FIGURE 3.1 – (a). Logo de l'université de Abdelhamid Ibn Badis; (b).Image de Abdelhamid Ibn Badis

La figure suivante (3.2) représente la somme des deux matrices c'est à dire (deux images).

1. Felix Klein : Mathématicien allemand (25 avril 1849 à *Düsseldorf* - 22 juin 1925 à *Göttingen*)

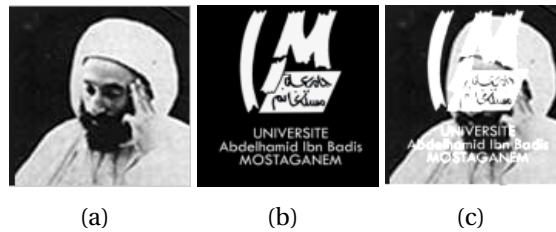


FIGURE 3.2 – (a). Image originale ; (b). Logo binaire ;(c). La somme des deux images

On remarque que la somme des deux images [(a) et (b)] donne la superposition de l'image (b) sur l'image (a).

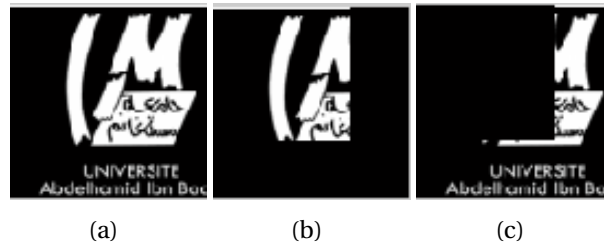


FIGURE 3.3 – (a). Logo binaire ;(b). Même logo avec un changement ;(c). La soustraction des deux images

On voit bien que la soustraction des deux images ne laisse que la différence qu'il y a entre ces deux images.

La figure suivante représente la multiplication de l'image de Abdelhamid Ibn Badis par un scalaire noté M. On fait l'expérience pour différentes valeurs de M, on obtient

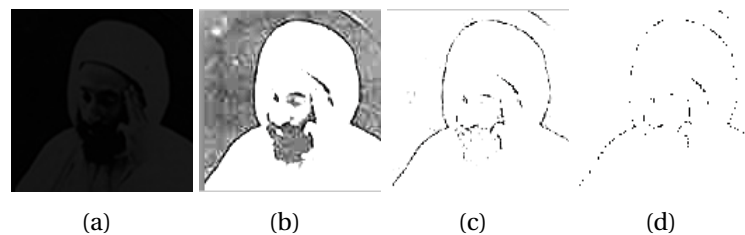


FIGURE 3.4 – Multiplication de l'image de Abdelhamid Ibn Badis par un scalaire M
(a). $M = 0.1$; (b). $M = 6$;(c). $M = 20$;(d). $M = 300$

On remarque que pour les petites valeurs de M, l'image devient de plus en plus sombre, car les petites valeurs dans les nuances du gris correspondent au "noir" et quand M prend de grandes valeurs, l'image devient de plus en plus blanche, car les grandes valeurs dans les nuances du gris correspondent au "blanc".

La figure suivante représente la division des deux images (a) par (b).

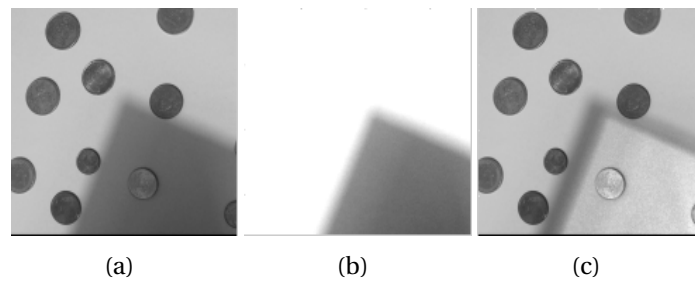


FIGURE 3.5 – (a).Image originale avec défaut d'éclairage ; (b). Éclairage ; (c). Image corrigée.

On remarque que la division des deux images donne une image (c) qui est plus claire que l'image (a).

La figure (3.6) représente la transposée de la matrice associée à l'image (logo binaire de l'université de Abdelhamid Ibn Badis).



FIGURE 3.6 – (a). Image originale ; (b).Image transposée

La rotation est illustrée dans la figure suivante

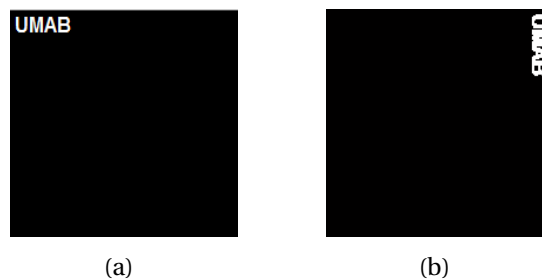


FIGURE 3.7 – (a). Logo binaire ; (b). La rotation du logo

3 Les résultats numériques de l'approximation de rang faible

On considère l'image de Felix Klein, représentée par une matrice A de dimension 720×524 pixels et de rang 524.



FIGURE 3.8 – L'image de Felix Klein

On réalise une expérience visuelle et ceci en approchant la matrice A par une matrice A_k de rang k , pour des valeurs de k comprises entre 1 et $rg(A) = 524$. (Application du théorème d'Eckart-Young-Mirsky, section précédente).

Le procédé de l'approximation est décrit dans l'algorithme suivant

Entrée :

- A : La matrice associée à l'image de "Klein".
- j : un entier choisi.

Sortie :

- d : l'image compressée.

Les étapes :

Début de l'algorithme

1. Initialisation :

$U_j = []$;

$V_j = []$;

$[m, n] = \text{size}(A)$;

2. Conversion de la matrice A en type réel.

$$h = \text{double}(A). \quad (3.1)$$

3. Décomposition de h en valeurs singulières,

$$h = U \Sigma V^t.$$

4. Approximation de rang faible

pour $i=j+1 : n$

$$\Sigma(i, i) = 0;$$

endpour

pour $i=1 : m$

pour $k=1 : j$

$$U_j(i, k) = U(i, k);$$

endpour

endpour

pour $i=1 : n$

pour k=1 :j

$$V_j(i, k) = V(i, k);$$

endpour

endpour

5. La reconstruction de la matrice d

$$d = U_j \Sigma V_j'. \quad (3.2)$$

Fin de l'algorithme

Les résultats obtenus par un programme sur MatLab sont représentés dans les figures suivantes.

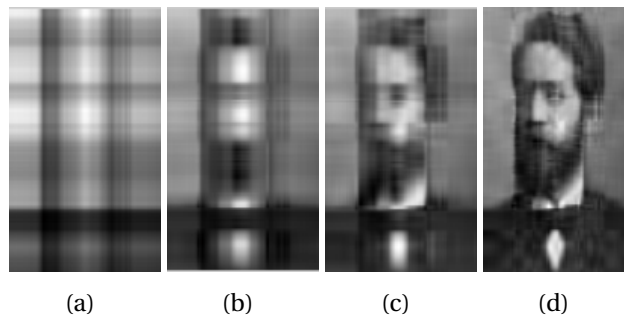


FIGURE 3.9 – L'approximation du portrait de Klein pour les différents valeurs du rang (a).k=1 ; (b).k=3 ; (c).k=5 ; (d).k=10

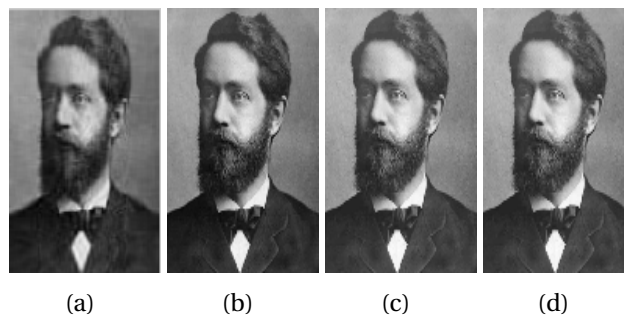


FIGURE 3.10 – L'approximation du portrait de Klein pour les différents valeurs du rang (a).k=20 ; (b).k=80 ; (c).k=100 ; (d).k=300

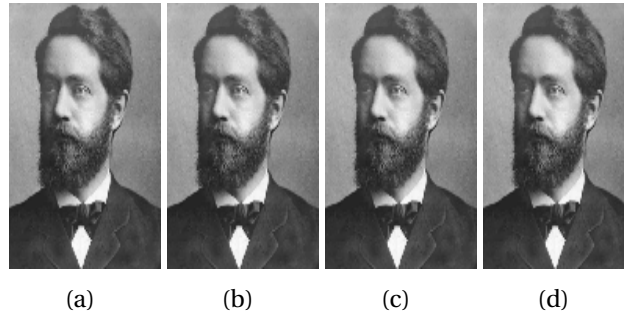


FIGURE 3.11 – L'approximation du portrait de Klein pour les différents valeurs du rang (a). $k=400$; (b). $k=500$; (c). $k=524$; (d).image original

On remarque que dans l'image (b) de la figure 3.9, les formes (nœud papillon, cheveux, yeux, barbe, épaules) sont des éléments forts de l'image puisque associés aux trois plus grandes valeurs singulières ($k=3$).

L'approximation de rang 20 donne déjà une image assez nette comparée à celle de rang 524 et que l'approximation de rang 80 donne une image proche visuellement de l'image originale qui est de rang 524. Cela signifie que les informations les plus importantes de l'image sont contenues dans les premières valeurs singulières de la matrice associée à cette image.

A partir de cette expérience, on voit qu'on a gagné en espace de stockage dans la mémoire. Si on veut par exemple approcher cette image de "Klein" par la matrice A_{80} , il suffit de stocker trois matrices de taille :

- 720 × 80 pixels pour la matrice U,
 - 80 × 80 pixels pour la matrice Σ et
 - 80 × 524 pixels pour la matrice V.
- Cette matrice va donc occuper

$$720 \times 80 + 80 \times 80 + 80 \times 524 = 105920 \text{ pixels} . \quad (3.3)$$

On voit bien qu'on a gagné trois fois plus d'espace par rapport à l'image d'origine qui occupe

$$720 \times 524 = 377280 \text{ pixels} \quad (3.4)$$

4 L'erreur de reconstruction

D'après l'expérience précédente, pour chaque k (le rang) donné, l'image associée à la matrice approchée A_k devient de plus en plus proche de l'image de Klein au fur et à mesure que le rang augmente. On note par e_ϵ l'erreur relative associée à cette approximation. Elle est définie comme étant l'erreur calculée à partir de la différence entre l'image A et sa version tronquée A_k et normalisée avec la norme de l'image A. Cette erreur est donc donnée par :

$$e_\epsilon = \frac{\|A - A_k\|_2}{\|A\|_2} \quad (3.5)$$

En s'appuyant sur les propriétés de la SVD et sur (2.51), cette erreur prend l'expression

$$e_\epsilon = \frac{\sigma_{k+1}}{\sigma_1} \quad (3.6)$$

4. L'ERREUR DE RECONSTRUCTION

Pour une erreur e_ϵ donnée, on peut calculer via la formule précédente, le rang nécessaire k , telle que l'erreur d'approximation de la matrice A par A_k soit inférieure à cette tolérance donnée.

En suivant l'algorithme du procédé de reconstruction ci dessous, on obtient les valeurs de k qui sont classés dans le tableau (3.1).

Entrées :

- A : la matrice associée à l'image de Klein.
- ϵ : tolérance donnée.

Sorties :

- K : le rang de la matrice compressée.

Les étapes

Début de l'algorithme

1. Conversion de la matrice A en type réel.
2. S est le vecteur contenant les valeurs singulières σ_i classées en ordre décroissant et m la taille de ce vecteur.

pour $i=2 :m$

$$L(i - 1) = S(i)/S(1); \quad (3.7)$$

endpour

3. On récupère l'indice k de la composante du vecteur L qui est inférieure strictement à ϵ par la commande "find".

Fin de l'algorithme

| le seuil " ϵ " | $5 * 10^{-2}$ | $3 * 10^{-2}$ | 10^{-2} | $5 * 10^{-3}$ | $4 * 10^{-3}$ | $2 * 10^{-3}$ | 10^{-3} | 10^{-4} | $3 * 10^{-5}$ |
|-------------------------|---------------|---------------|-----------|---------------|---------------|---------------|-----------|-----------|---------------|
| k | 8 | 11 | 35 | 66 | 81 | 146 | 216 | 360 | 510 |

TABLEAU 3.1 – Les valeurs de k à partir des tolérances ϵ données

Ce tableau est représenté graphiquement dans la figure suivante

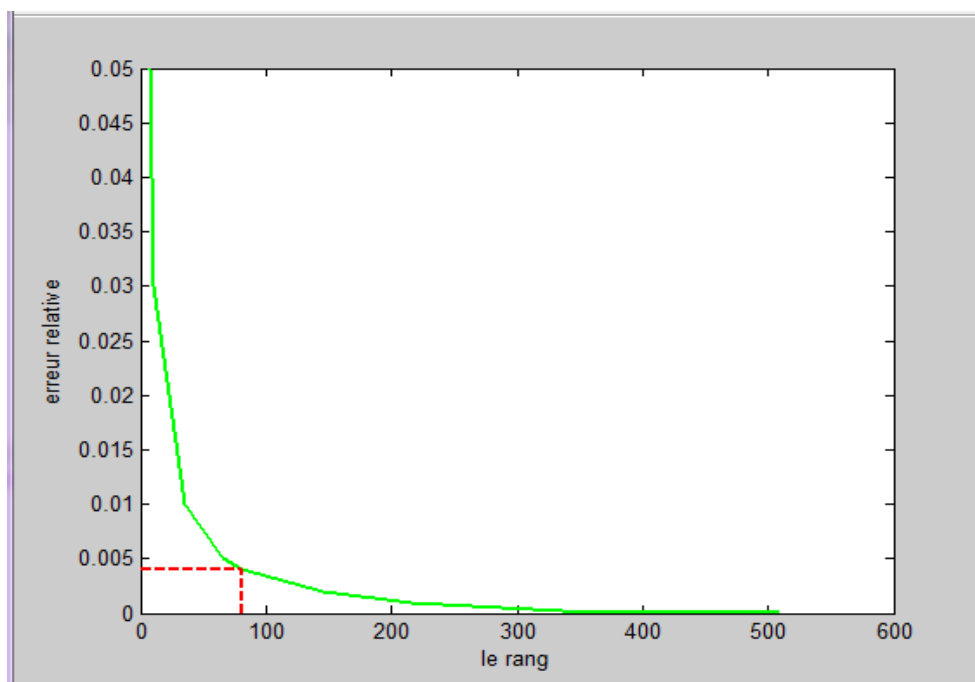


FIGURE 3.12 – Variation de l'erreur relative en fonction de k

Pour plus de visibilité, on représente les variations de logarithme de l'erreur en fonction du rang

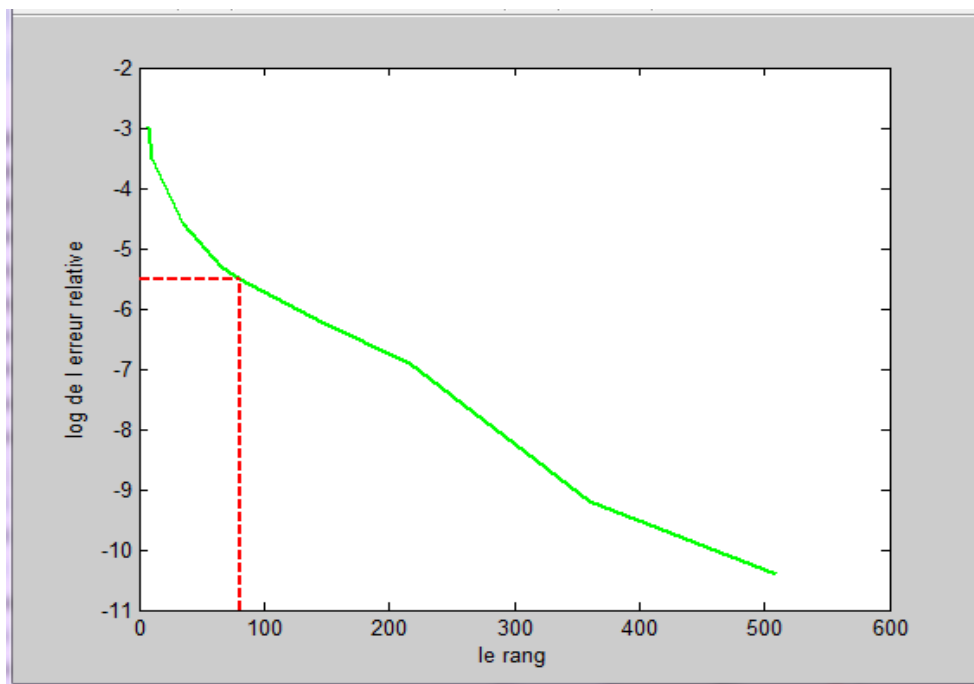


FIGURE 3.13 – Variation du logarithme de l'erreur relative en fonction de k

On voit bien que l'erreur relative diminue quand le rang de la matrice tronquée augmente.

5 Les expériences de tatouage

5.1 Le tatouage visible

On considère l'image de ABD ALHAMID IBN BADIS, et la marque (logo binaire de l'Université de Mostaganem).

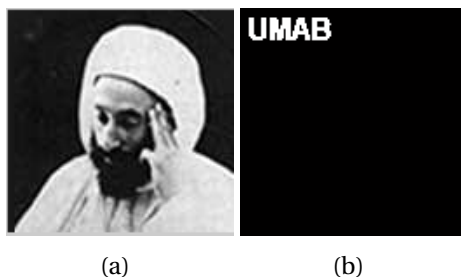


FIGURE 3.14 – (a). Image de ABD ALHAMID IBN BADIS ; (b). La marque.

On fait l'expérience sur les images, en appliquant l'algorithme présenté dans la section 5. pour le tatouage visible les résultats sont obtenus par un programme sur MatLab.

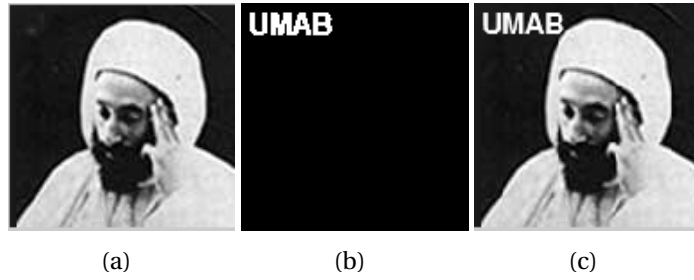


FIGURE 3.15 – (a). Image originale ; (b). La marque ;(c).Image tatouée

L'image tatouée (c) est le résultat de la somme de l'image originale (a) et de la marque (b).

5.2 Le tatouage invisible

Pour le tatouage invisible on fait l'expérience sur la même image hôte (ABD ALHAMID IBN BADIS) et la même marque (logo binaire de l'Université de Mostaganem) avec différentes valeurs de α . On va faire varier α jusqu'à obtention d'une qualité d'image tatouée imperceptible de l'image originale.

On va appliquer l'algorithme global de Chandra et ses deux versions

Les résultats obtenus par l'utilisation la matrice Σ (//insertion de la marque).

On rappelle que l'image tatouée f_t est obtenue comme suit

$$f_t = U \Sigma_t V^t \text{ avec } \Sigma_t = \Sigma + \alpha \Sigma_w. \quad (3.8)$$

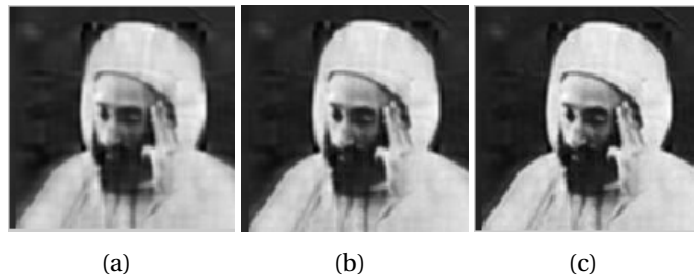


FIGURE 3.16 – Tatouage invisible pour différentes valeurs du coefficient d'ajustement α (avec la matrice Σ) (a). $\alpha = 10$; (b). $\alpha = 5$;(c). $\alpha = 3$

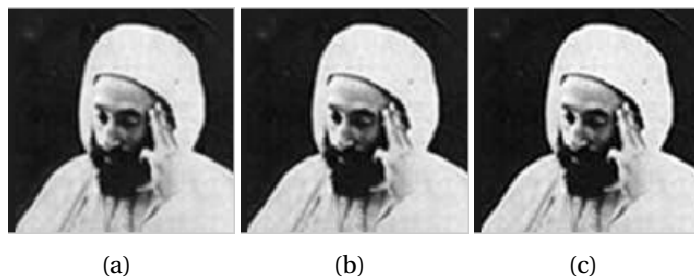


FIGURE 3.17 – Tatouage invisible pour différentes valeurs du coefficient d'ajustement α (avec la matrice Σ) (a). $\alpha = 1$; (b). $\alpha = 0.5$;(c). $\alpha = 0.1$

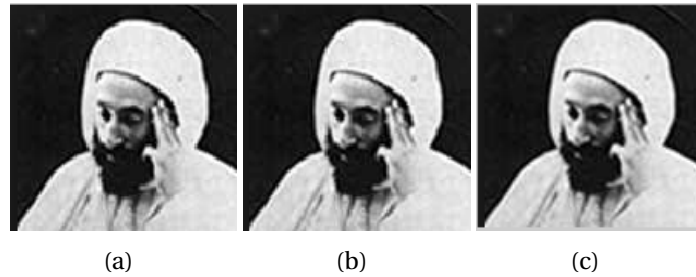


FIGURE 3.18 – Tatouage invisible pour différentes valeurs du coefficient d’ajustement α (avec la matrice Σ) (a). $\alpha = 0.01$; (b). $\alpha = 0.001$; (c). image originale

On remarque que si α prend des valeurs très grandes, la qualité de l’image tatouée est mauvaise par rapport à l’image originale (image hôte). Par contre, si α est très petit, on remarque que la qualité d’image tatouée et l’image originale sont les mêmes visuellement.

Pour extraire la marque, on suit l’algorithme présenté dans le chapitre précédent. On va donc donner l’image tatouée et l’image originale et par application de l’algorithme de l’extraction de la marque avec $\alpha = 0.001$ (coefficient correspondant à l’image la plus nette (b)), on obtient

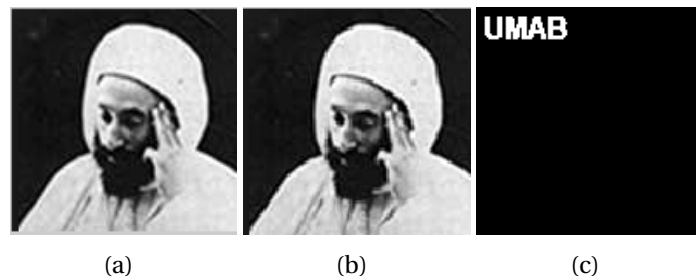


FIGURE 3.19 – (a).Image originale ; (b). Image tatoué ; (c).La marque

Le fait d’avoir récupéré la vraie marque (c), cela assure que l’image tatouée (b) est authentique (elle n’a été ni copiée, ni altérée, ...)

On rappelle que pour la première version, l’image tatouée f_t est obtenue comme suit

$$f_t = U_t \Sigma V^t \text{ avec } U_t = U + \alpha U_w. \quad (3.9)$$

Les résultats du tatouage obtenus en utilisant la matrice U.

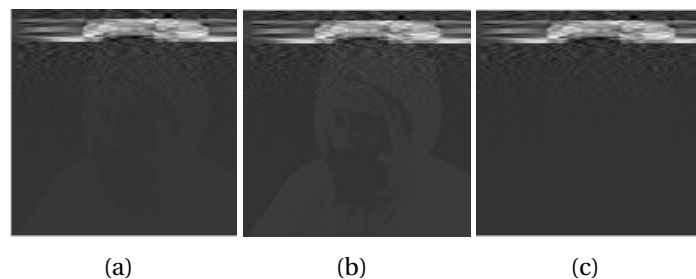


FIGURE 3.20 – Tatouage invisible pour différentes valeurs du coefficient d’ajustement α (avec la matrice U). (a). $\alpha = 10$; (b). $\alpha = 5$; (c). $\alpha = 3$

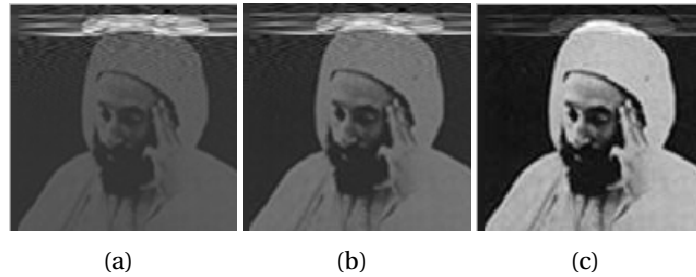


FIGURE 3.21 – Tatouage invisible pour différentes valeurs du coefficient d’ajustement α (avec la matrice U). (a). $\alpha = 1$; (b). $\alpha = 0.5$; (c). $\alpha = 0.1$

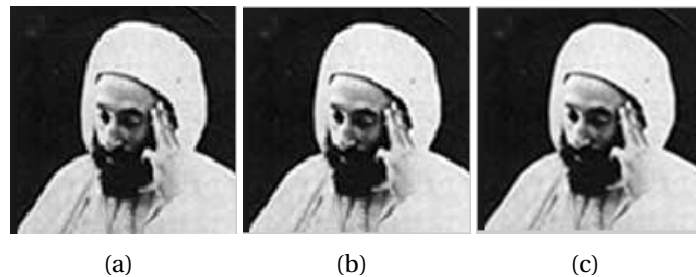


FIGURE 3.22 – Tatouage invisible pour différentes valeurs du coefficient d’ajustement α (avec la matrice U). (a). $\alpha = 0.01$; (b). $\alpha = 0.001$; (c). image originale

On rappelle que pour la deuxième version, l’image tatouée f_t est obtenue comme suit

$$f_t = U \Sigma V_t^t \text{ avec } V_t = V + \alpha V_w. \quad (3.10)$$

Les résultats du tatouage obtenus en utilisant la matrice V.

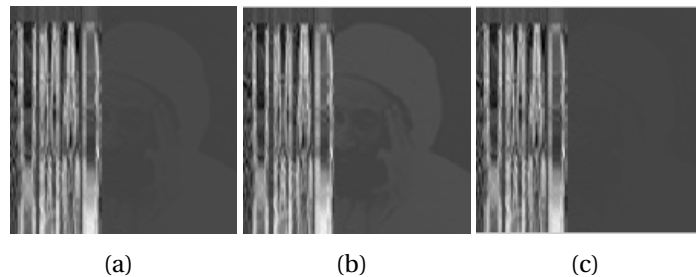


FIGURE 3.23 – Tatouage invisible pour différentes valeurs du coefficient d’ajustement α (avec la matrice V). (a). $\alpha = 10$; (b). $\alpha = 5$; (c). $\alpha = 3$

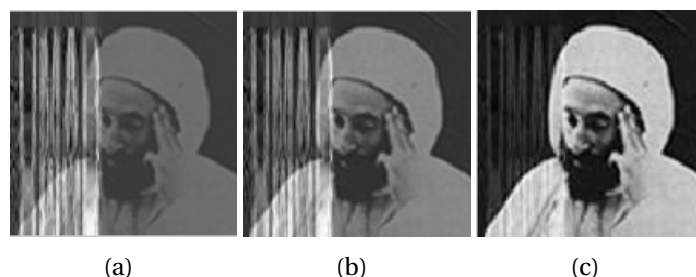


FIGURE 3.24 – Tatouage invisible pour différentes valeurs du coefficient d’ajustement α (avec la matrice V). (a). $\alpha = 1$; (b). $\alpha = 0.5$; (c). $\alpha = 0.1$

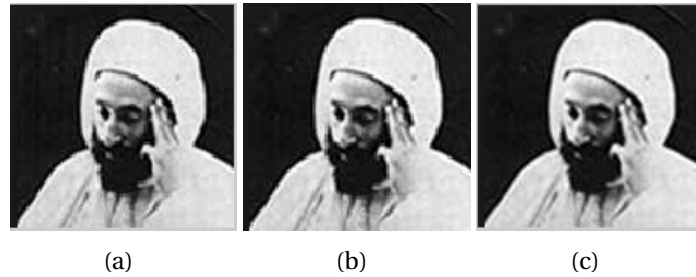


FIGURE 3.25 – Tatouage invisible pour différentes valeurs du coefficient d’ajustement α (avec la matrice V). (a). $\alpha = 0.01$; (b). $\alpha = 0.001$;(c). image originale

On remarque que le comportement du coefficient d’ajustement α est le même que l’on utilise la matrice Σ , U ou V , c’est à dire que l’image tatouée et l’image originale sont indiscernables visuellement pour la même valeur de $\alpha = 0.001$.

Les défauts des images tatouées pour les grandes valeurs de α prennent des aspects différents selon qu’on utilise la matrice Σ , U et V .

Conclusion

Dans ce travail, on s'est intéressé à la méthode de décomposition en valeurs singulières d'une matrice. Cette méthode de factorisation est appliquée pour n'importe quelle matrice qu'elle soit carrée inversible ou non inversible ou bien même rectangulaire.

On a vu que cet outil mathématique est fort de part les propriétés qu'il offre. Quelques unes de ces propriétés ainsi que leurs preuves ont été présentées dans ce travail.

Parmi tous les domaines d'application de cette méthode, on a parcouru quelques uns tels que :

- La résolution des systèmes linéaires où on a vu, à travers des exemples que cette méthode est toujours applicable alors que les autres (calcul de l'inverse, l'équation normale, la méthode de moindres carrés) ont atteint leurs limites.
- La compression de l'image par l'approximation de rang faible où on a vu que le théorème d'Ecart Young s'appuie sur cette méthode pour approcher une matrice (qui représente une image) par une autre matrice de rang plus petit (qui représente l'image compressée) tout en conservant une bonne qualité de l'image dans cette approximation.
- Le tatouage numérique qui a pour rôle la protection des images et la vérification de l'intégrité des données, est basé sur des techniques algorithmiques. Ces dernières s'appuient sur la décomposition en valeurs singulières de l'image hôte ainsi que de la marque insérée.

Des programmes ont été élaborés en langage MatLab sur PC i3, pour faire des essais sur la compression de l'image et le tatouage numérique. Les résultats sont donnés dans le dernier chapitre.

Bibliographie

- [1] **Al Kharboutly. Mira**, (2018)(Thèse pour obtenir le diplôme de doctorat en Mathématiques Appliquées) *Résolution d'un problème quadratique non convexe avec contraintes mixtes par les techniques de l'optimisation D.C*, Université Le Havre Normandie, France. [17](#)
- [2] **Ayvazyan. Vigen**, (2012)(Thèse pour obtenir le grade de docteur en Mécanique et Énergétique) *Etude de champs de température séparables avec une double décomposition en valeurs singulières. Quelques applications à la caractérisation des propriétés thermophysiques des matériaux et au contrôle non destructif*, Université BORDEAUX 1, Talence. [4](#), [8](#), [25](#)
- [3] **GOLEA. Nour El-Houda**, (Magister en informatique) *Tatouage numérique des images couleurs RGB*, Université ELHADJ LAKHDAR, batna, Algérie, page60-62. [27](#)
- [4] **Hetatache. Karima**, (2014)(Magister en communication) *Développement d'algorithmes de tatouage d'images basés sur la SVD et les transformées discrètes*, Université FERHAT ABBAS, Setif, Algérie. [27](#)
- [5] **K.P.Chong. Edwin. and H.Zak. Stanislaw**. (2001), *An Introduction to Optimization*, Édition 2, Canada. [19](#), [21](#)
- [6] **Lay. David C.** (2012 ; Pearsons France), *Algèbre Linéaire et applications*, 4^e édition, Montreuil. [3](#), [4](#), [15](#)

La décomposition en valeurs singulières et ses applications

Résumé :

L'objectif de cette étude est de présenter la méthode de décomposition en valeurs singulières d'une matrice et de l'illustrer par quelques unes de ses applications telles que la résolution des systèmes linéaires, la compression de l'image et le tatouage numérique.

Mots-Clés :

SVD, calcul matriciel, tatouage numérique, compression d'image.

The singular value decomposition and its applications

Abstract :

The aim of this study is to present the singular value decomposition (SVD) method and illustrate it from some of its applications such that the resolution of linear systems, image compression and digital tattooing.

Key Words :

SVD, matrix calculation, digital tattoo, image compression.

