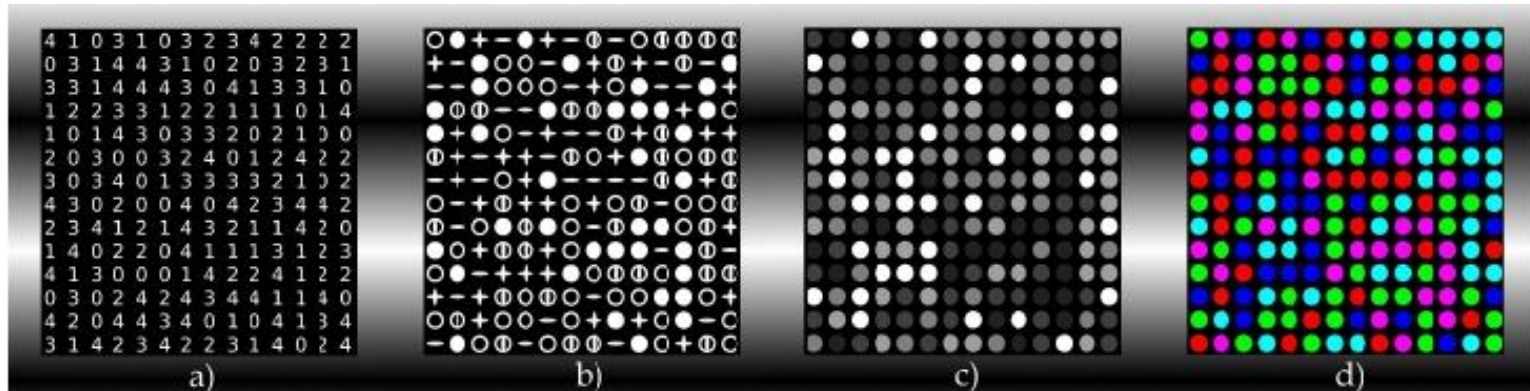


Analyse Matricielle (*Chapitre 2*)

Formation d'Ingénieurs Informatique et Réseaux
(édition 2019-2020)



A Perfect Map (14,14,3,3,5) and some associated patterns (*shape, grey and colour levels*)

Christophe DOIGNON

Professeur à Télécom Physique Strasbourg
ICube, Université de Strasbourg

Bureau C418 – Courriel : c.doignon@unistra.fr





Chapitre 2

Méthodes de décomposition





2.1 Inversion généralisée

Rappelons que l'objectif principal de ce cours est de formaliser mathématiquement des systèmes physiques par des systèmes linéaires et de les résoudre en préservant ses propres caractéristiques. Nous nous attachons donc à construire des modèles (qui sont toujours imparfaits) et à leur associer des données (qui sont entachées de bruit) afin de réduire le volume initial de données par ce qu'il convient de considérer comme une classification des données : celles qui participent à la modélisation et celles qui n'y participent pas. A ceci deux questions surgissent :

1. comment effectuer cette classification ? Avec quel critère de discrimination ?
2. cette classification est-elle sensible à la manière dont les données sont représentées ?

Prenons l'exemple de trois points géométriques (les données issues de mesures) dans un plan : nous souhaitons les rassembler sur un cercle. Nous réduirons ainsi de 6 (coordonnées) à 3 (paramètres) leur statut, part de leur représentation.

Mise en œuvre : étant donnée les coordonnées homogènes des points $P_i = (x_i \ y_i \ 1)^T, (i = 1,2,3)$ dans un plan, déterminons les coordonnées du centre du cercle $C = (x_c \ y_c)^T$ et le rayon R qui passe par ces trois points. A cette fin, exprimons ce problème sous la forme d'un système linéaire inhomogène, génériquement noté $A s = b$, où le vecteur s a ses composantes qui dépendent du modèle uniquement alors que la matrice A et le vecteur b ne dépendent que des données au contraire.



2.1 Inversion généralisée

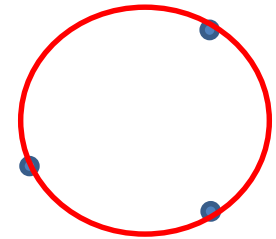
Si on considère une représentation de ces données avec l'équation cartésienne d'un cercle, il vient :

$$(x_i - x_c)^2 + (y_i - y_c)^2 = R^2 \iff (x_i \ y_i \ 1)^T \begin{pmatrix} 1 & 0 & -x_c \\ 0 & 1 & -y_c \\ -x_c & -y_c & x_c^2 + y_c^2 - R^2 \end{pmatrix} \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} = 0$$

On donne : $P_1 = (0 \ 0 \ 1)^T$; $P_2 = (5 \ 0 \ 1)^T$; $P_3 = (0 \ 5 \ 1)^T$

Dans la formulation linéaire $A s = b$, chaque ligne a_i^T de la matrice A et chaque composante b_i du vecteur b contiendront une équation cartésienne de ce type, et en développant on aura alors la formulation détaillée suivante :

$$a_i^T s = b_i \Rightarrow (-2x_i \ -2y_i \ 1) \begin{pmatrix} x_c \\ y_c \\ x_c^2 + y_c^2 - R^2 \end{pmatrix} = -(x_i^2 + y_i^2)$$



Avec 3 points, $A^T = (a_1 \ a_2 \ a_3)$ sera une matrice (3×3) . La solution (unique) est alors donnée par le calcul de l'inverse de la matrice : $s = A^{-1} b$ pourvu que la matrice ait un rang plein, ce qui est théoriquement assuré si les points P_i ne sont pas alignés. Il s'agit ici d'une inversion matricielle classique, où les cas de dégénérescence (ou singularité de la matrice du système) sont bien identifiés.



2.1 Inversion généralisée

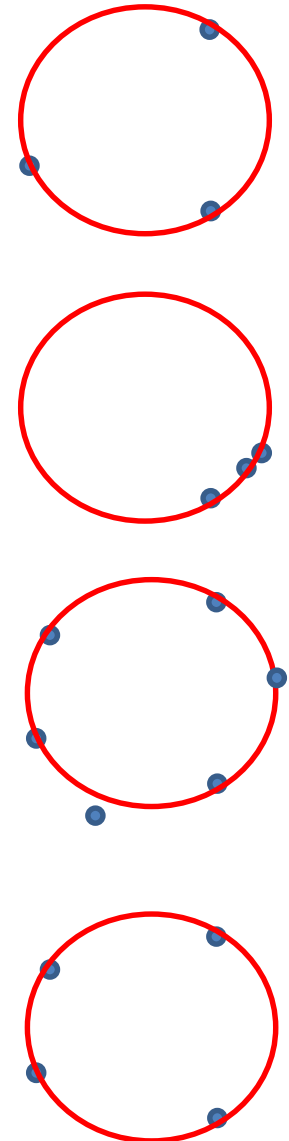
Les données étant acquises par un dispositif physique qui opère des conversions de formats la plupart du temps, obtiendrons-nous une sensibilité élevée (c'est-à-dire une variation forte des paramètres du cercle) vis-à-vis des erreurs sur les mesures qui dépendra de la configuration relative des points ?

Pour répondre à cela, il faut considérer plus de mesures, car à partir de trois points quelconques dans un plan (mais non-alignés), on peut toujours définir un cercle qui passent par eux.

Etant donné un quatrième, un cinquième, ..., n points P_i , tous étant supposés être disposés sur ce cercle, la contribution de l'ensemble de ces $2n$ données amène à considérer $n > 3$ équations du type $a_i^T s = b_i$, mais le bruit présent sur ces mesures fait qu'il sera impossible de résoudre exactement $A s = b$, car on a donc bien plus d'équations que nécessaire, ce qui est exprimé mathématiquement par la nature rectangulaire de la matrice A , précisément avec plus de lignes que de colonnes. De quelle façon pouvons-nous exploiter ce surplus de données au bénéfice de la précision de la classification ? (les valeurs des 3 paramètres caractérisant le cercle).

Etant donné le quatrième point suivant, $P_4 = (2.5 \quad 6.03 \quad 1)^T$, déterminer les paramètres du cercle qui passe au plus près de ces quatre points.

Y-a-t-il des équations « meilleures » que d'autres ? Dans le sens plus contributives pour obtenir les paramètres du cercle avec une précision et une exactitude accrues ?





2.1 Inversion généralisée

Quand $n > 3$, la matrice A n'est plus carrée, car de dimensions $(n \times 3)$. Mais elle gardera un rang toujours au plus égal à trois.

Par conséquent et comme il s'agit d'une matrice à coefficients réels, il est licite de prémultiplier de part et d'autre des deux membres du système $A s = b$ par la matrice $A^T = (a_1 \ a_2 \ \dots \ a_n)$, ce qui donne : $A^T A s = A^T b$. Si le rang de la matrice A est égal à 3, alors il en est de même du produit $A^T A$, faisant de cette matrice symétrique une matrice carrée inversible. Une solution pour s est alors donnée par :

$$s = (A^T A)^{-1} A^T b$$

Dans cette expression, la matrice $A^+ = (A^T A)^{-1} A^T$ est appelée matrice pseudo-inverse **gauche** de A . Cette matrice s'obtient également quand on minimise le critère quadratique algébrique $\|A s - b\|^2$, ce qui est une manière d'indiquer que toutes les données contribueront à la détermination de la solution, bien qu'elles ne soient pas toutes perturbées avec le même degré. Cette minimisation est appelée plus communément « minimisation par les moindres carrés » (*Least Mean Squares - LMS*). D'autres techniques - autres que le calcul de la matrice pseudo-inverse - peuvent la mettre en œuvre (cf. chapitre suivant).

```
% scilab
%
A=[0 0 1;-10 0 1;0 -10 1]; b= -[0;25;25];
s=inv(A)*b % affiche s=[2.5 2.5 0]
R = sqrt(s(1)^2+s(2)^2-s(3)) % affiche R= 3.53553

A=[A;-5 -12.06 1]; b= [b;-((2.5)^2+(6.03)^2)];
s = pinv(A)*b % s = inv(A'*A)*A'*b affiche [2.49852 2.49766 -0.00862]
R = sqrt(s(1)^2+s(2)^2-s(3)) % affiche R= 3.53406
```



2.1 Inversion généralisée

Soit une matrice $A \in M_{m,n}(\mathbb{C})$. L'inverse généralisée de A est la matrice A^+ telle que $AA^+A = A$. $A^+ \in M_{n,m}(\mathbb{C})$. Elle n'existe pas toujours.

La matrice $A^+ = (A^*A)^{-1}A^*$ est appelé la matrice **pseudo-inverse gauche** de A , ou matrice de Moore-Penrose quand $n \leq m$. Elle existe si A est de rang n .

La matrice $A^+ = A^*(AA^*)^{-1}$ est la matrice **pseudo-inverse droite** de A , quand $m \leq n$. Elle existe si A est de rang m .

L'inversion généralisée s'emploie pour résoudre des systèmes linéaires inhomogènes (c'est-à-dire avec un second membre non nul), de type $AX = b$, où $A \in M_{m,n}(\mathbb{C})$, surdéterminés ($m > n$), c'est-à-dire quand on dispose de plus d'équations linéaires que d'inconnues. $X = (x_1, \dots, x_n)^T$ et $b = (b_1, \dots, b_m)^T$ sont des vecteurs (X est le vecteur des inconnues).

Puisque toutes les équations ne peuvent pas généralement être vérifiées strictement, on applique la méthode des moindres carrés. Ceci consiste à déterminer une solution pour le vecteur X en cherchant à minimiser la quantité

$$\|AX - b\|^2$$

qui prend en compte toutes les équations du système. On montre (de manière alternative à l'aide de la dérivée des matrices et des vecteurs) que la solution à ce problème est donnée par la formule $X = A^+b$ si A^+ existe.



2.1 Inversion généralisée

- Exemple :

$$\begin{cases} x_1 + x_2 + x_3 = 1 \\ x_1 + x_2 - x_3 = 1 \\ x_1 - x_2 = 0 \\ x_1 + x_3 = 2 \end{cases} \Rightarrow \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 1 & -1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 2 \end{pmatrix}$$

correspond au système linéaire $A X = b$ avec $b = (1,1,0,2)^T$ et

$$A^T A = \begin{pmatrix} 4 & 1 & 1 \\ 1 & 3 & 0 \\ 1 & 0 & 3 \end{pmatrix} ; (A^T A)^{-1} A^T = \begin{pmatrix} 0.1 & 0.3 & 0.4 & 0.2 \\ 0.3 & 0.2333 & -0.4667 & -0.0667 \\ 0.3 & -0.433 & -0.1333 & 0.2667 \end{pmatrix}$$

On obtient alors pour $X = (A^T A)^{-1} A^T b = (0.8, 0.4, 0.4)^T$.

Le calcul de $\|A X - b\| = 1,095$ montre que la valeur de ce résidu est trop élevée pour que les équations soient (toutes) considérées comme vérifiées simultanément.



2.1 Inversion généralisée

- **Exemple** : on modifie b en $b' = (2, 1, 0, 1)^T$

$$\begin{cases} x_1 + x_2 + x_3 = 2 \\ x_1 + x_2 - x_3 = 1 \\ x_1 - x_2 = 0 \\ x_1 + x_3 = 1 \end{cases} \Rightarrow \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 1 & -1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \\ 0 \\ 1 \end{pmatrix}$$

correspond au système linéaire $A X = b'$ avec $b' = (2, 1, 0, 1)^T$ et on obtient alors pour X : $X = (A^T A)^{-1} A^T b' = (0.7, 0.7667, 0.4333)^T$.

Le calcul de $\|A X - b'\| = 0,1826$ montre que les équations sont mieux vérifiées.



2.1 Inversion généralisée

- **Exemple** : on modifie b en $b'' = (2, 2, 0, 1)^T$

$$\begin{cases} x_1 + x_2 + x_3 = 2 \\ x_1 + x_2 - x_3 = 2 \\ x_1 - x_2 = 0 \\ x_1 + x_3 = 1 \end{cases} \Rightarrow \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 1 & -1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 2 \\ 2 \\ 0 \\ 1 \end{pmatrix}$$

correspond au système linéaire $A X = b''$ avec $b'' = (2, 2, 0, 1)^T$ et on obtient alors pour X : $X = (A^T A)^{-1} A^T b'' = (1.0, 1.0, 0.0)^T$.

Le calcul de $\|A X - b''\| = 7,6919 \cdot 10^{-16}$, valeur proche de la (double) précision de la machine, ce qui montre que les équations sont bien vérifiées.



2.1 Inversion généralisée

L'inversion généralisée s'emploie aussi pour résoudre les systèmes linéaires inhomogènes sous-déterminés ($n > m$), c'est-à-dire quand on dispose de moins d'équations linéaires que d'inconnues. Dans ce dernier cas, on parle aussi de système déficient (sous-entendu de rang non plein, donc « déficient »). Les solutions sont ici généralement contenues dans un sous-ensemble dont la dimension dépendra du rang et du vecteur b . La solution générale est de la forme :

$$X = A^+b + (I_n - A^+A) v$$

où v est un vecteur quelconque de dimension n et A^+ est la pseudo-inverse de droite. Le premier terme, A^+b , est appelée solution minimale car correspondant à la fois à minimiser $\|AX - b\|^2$ mais aussi la norme euclidienne du vecteur des inconnues : $\|X\|^2$.

Dans le second terme, le vecteur $z = (I_n - A^+A) v$ correspond alors à la projection de v sur le noyau de A : $Az = 0$. Par conséquent $z \in \text{Ker}(A)$ ne modifiera pas le vecteur b . Il est parfois appelé terme d'optimisation (notamment en robotique, pour modéliser la cinématique des robots redondants). Du fait de son appartenance au noyau, il est utilisé pour satisfaire des contraintes supplémentaires.



2.1 Inversion généralisée

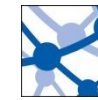
- **Exemple en robotique** : contrôle de la trajectoire par vision déportée et robot plan redondant.

Kinematic Control Algorithms for On-Line Obstacle Avoidance for Redundant Manipulators

Leon Žlajpah
Bojan Nemec

Asservissement visuel avec caméra déportée et redondance structurelle (globale).

Quelle est la tâche principale à réaliser ici ?



2.2 Méthodes d'inversion par blocs – Généralités

De par la structure de certaines matrices, des techniques spécifiques de décomposition et d'inversion sont très utiles, pour réduire fortement le nombre d'opérations à exécuter mais aussi pour conserver la structure de la matrice qui embarque une ou plusieurs caractéristiques du modèle. Il est ainsi souvent nécessaire de s'assurer de conserver les propriétés d'orthogonalité, de symétrie, ..., d'une matrice lors de l'inversion. Les différentes sources d'erreurs n'y contribuent généralement pas, au contraire.

Soit M une matrice diagonale par blocs. Comme son nom l'indique, cette matrice n'est pas strictement diagonale, avec un élément (un nb réel ou complexe) par composante sur la diagonale, mais elle l'est si on étend la nature de cet élément à une sous-matrice. Alors son inverse est obtenue en inversant chaque bloc si celui-ci est inversible.

$$M = \begin{bmatrix} A_1 & & & \\ & A_2 & & \\ & & \ddots & \\ & & & A_n \end{bmatrix}$$

$$M^{-1} = \begin{bmatrix} A_1^{-1} & & & \\ & A_2^{-1} & & \\ & & \ddots & \\ & & & A_n^{-1} \end{bmatrix}$$

Nous allons par la suite aborder plusieurs décompositions, dont des factorisations, qui concourront à atteindre cet objectif.



2.2 Méthodes d'inversion par blocs – Décomposition de Schur

Considérons la matrice carrée M correspondant partitionnée en 4 sous-matrices, selon la structure suivante :

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix};$$

$A \in M_{m,n}(\mathbb{C})$, $B \in M_{m,k}(\mathbb{C})$, $C \in M_{l,n}(\mathbb{C})$ et $D \in M_{l,k}(\mathbb{C})$. Si $m = n$ et que la matrice A est inversible, alors M peut être décomposée selon la diagonalisation par bloc suivante, appelée aussi la diagonalisation de Aitken :

$$M = \begin{bmatrix} I_m & 0 \\ CA^{-1} & I_l \end{bmatrix} \begin{bmatrix} A & 0 \\ 0 & S \end{bmatrix} \begin{bmatrix} I_n & A^{-1}B \\ 0 & I_k \end{bmatrix};$$

$S = D - CA^{-1}B$ est appelé le **complément de Schur** de A dans M . $S \in M_{l,k}(\mathbb{C})$

Si S est une matrice carrée inversible alors l'inverse de M vaut (formule de Banachiewicz) :

$$M^{-1} = \begin{bmatrix} A^{-1} + A^{-1}BS^{-1}CA^{-1} & -A^{-1}BS^{-1} \\ -S^{-1}CA^{-1} & S^{-1} \end{bmatrix}.$$

Une fois que les quantités A^{-1} et S^{-1} sont calculées, toutes les composantes de M^{-1} sont déterminées alors par de simples multiplications de matrices.

Remarque : Cette formule permet de réduire drastiquement les coûts calculatoires pour les matrices de grandes dimensions pour peu que les matrices précitées existent. Par conséquent la précision de la méthode repose sur le bon conditionnement des deux matrices A et S à inverser.



2.2 Méthodes d'inversion par blocs – Décomposition de Schur

Remarques :

1. La décomposition qui amène à la détermination du **Complément de Schur** est un outil très utile pour transformer des contraintes non-linéaires en un problème LMI convexe.
2. Le complément de Schur est aussi très utile pour mettre à jour une matrice dont les dimensions augmentent au fur et à mesure de l'acquisition des données (dimensions de A très supérieures à celles de B , C et D). En particulier, on a : $\det(M) = \det(A) \det(S)$.
3. Le complément de Schur est utilisé dans la résolution de certains systèmes d'équations linéaires, intervenant dans l'élimination d'un bloc de variables.

- **Exercice :** Inverser la matrice suivante en utilisant le complément de Schur

$$M = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 2 & 0 \\ -1 & 0 & 2 \end{bmatrix} = \begin{bmatrix} A & -1 \\ -1 & 0 & 2 \end{bmatrix}$$

$$1) \text{ Déterminer l'inverse de } A. \text{ Montrer que } A^{-1} = \frac{1}{2} \begin{bmatrix} 2 & 0 \\ -1 & 1 \end{bmatrix}$$

2) Calculer le complément de Schur S de A dans M .

3) Montrer que le rang de M est plein.

4) Déterminer l'inverse de S .

$$5) \text{ Déterminer l'inverse de } M. \text{ Montrer que } M^{-1} = \begin{bmatrix} 2 & 0 & 1 \\ -1 & 1/2 & -1/2 \\ 1 & 0 & 1 \end{bmatrix}$$



2.2 Méthodes d'inversion par blocs – Décomposition de Schur

Quatre cas particuliers sont souvent rencontrés :

- $C = B^T$: matrice à blocs symétriques, alors $S = D - B^T A^{-1} B$.
- $A = I_n$: matrice identité, alors $S = D - CB$ et on a : $M^{-1} = \begin{bmatrix} I_n + BS^{-1}C & -BS^{-1} \\ -S^{-1}C & S^{-1} \end{bmatrix}$
- D est un scalaire. Dans ce cas, $B = \underline{b}$ et $C = \underline{c}^T$ des vecteurs resp. colonnes et lignes.
- B ou C est nulle (cinématique des robots manipulateurs : $\dot{X} = J(q) \dot{q} = \frac{\partial f}{\partial q^T} \dot{q}$)

$$J(q)^{-1} = \begin{bmatrix} A^{-1} & 0 \\ -D^{-1}CA^{-1} & D^{-1} \end{bmatrix}.$$



Robot manipulateur anthropomorphe STÄUBLI RX-160 à 6 articulations rotoïdes ($X = f(q)$).

La configuration verticale de l'image de droite est une redondance locale car il y a clairement plusieurs façons de réaliser une rotation d'axe vertical de l'effecteur.



2.2 Méthodes d'inversion par blocs - Décomposition SMW

La décomposition SMW ou formule Sherman-Morrison-Woodbury s'applique à une matrice M qui peut s'exprimer sous la forme :

$$M = A + U C V$$

où

- A est une matrice carrée, $A \in M_n(\mathbb{C})$, qui contient les blocs diagonaux de M
- $U C V$ contient les blocs extra-diagonaux, où $U \in M_{n,k}(\mathbb{C})$, $C \in M_k(\mathbb{C})$ et $V \in M_{k,n}(\mathbb{C})$.

Dans ce cas, la matrice M peut être inversée en utilisant la formule d'inversion de Sherman-Morrison-Woodbury

$$M^{-1} = (A + U C V)^{-1} = A^{-1} - A^{-1} U (C^{-1} + V A^{-1} U)^{-1} V A^{-1}$$

ce qui suppose que les matrices A , C et $(C^{-1} + V A^{-1} U)$ soient inversibles.

- **Exemple** : Cas particulier très fréquent

Déterminer l'inverse de la matrice $M = A + uv^T$ en fonction de A^{-1} et des vecteurs quelconques u et v .



2.2 Méthodes d'inversion par blocs - Décomposition SMW

Remarque : Si l'inverse de la matrice A a déjà été calculée auparavant, il est seulement nécessaire de calculer l'inverse de $C^{-1} + VA^{-1}U$ pour déterminer l'inverse de la matrice M . Cette formule est par conséquent très intéressante quand la matrice C est de dimensions bien inférieure à celle de la matrice A (*low-rank update*). Le filtre de Kalman, les moindres carrés récurrents, ou encore des techniques de minimisation sans contraintes (*rank-one update, and rank-two update*) font appel à cette formule pour mettre à jour les estimations (vecteurs d'état et gain) en temps réel.

- **Exercices :** Déterminer l'inverse des matrices suivantes :

- $M = I + u u^T$ où I est la matrice identité et $u = (1,1,1,1,1,1)^T$.

- $N = \begin{bmatrix} 17 & -3 & 6 \\ 20 & -2 & 8 \\ 5 & -1 & 4 \end{bmatrix} = 2I + \begin{bmatrix} 3 \\ 4 \\ 1 \end{bmatrix} \begin{bmatrix} 5 & -1 & 2 \end{bmatrix}$

- **Exercice :** Calculer le produit des deux matrices suivantes (c est un nb réel)

$$(I + de^T)(I + c de^T)$$

et montrer que pour tous vecteurs d et e , il peut se mettre sous la forme $I + s de^T$, où s est un nb réel à déterminer. En déduire l'expression de la matrice inverse de $I + de^T$.



2.3 Méthodes par factorisation – Décomposition de Cholesky

Pour toute matrice Hermitienne définie positive A , il existe une matrice L triangulaire inférieure telle que $A = L L^*$. Cette matrice L est appelée le facteur de Cholesky, est unique avec des composantes réelles et positives sur sa diagonale.

$$\begin{pmatrix} l_{11} & & & \\ l_{21} & l_{22} & & \\ l_{31} & l_{32} & l_{33} & \\ \vdots & \vdots & \ddots & \ddots \end{pmatrix} \cdot \begin{pmatrix} l_{11}^* & l_{21}^* & l_{31}^* & \cdots \\ & l_{22}^* & l_{32}^* & \cdots \\ & & l_{33}^* & \ddots \\ & & & \ddots \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots \\ a_{21} & a_{22} & a_{23} & \cdots \\ a_{31} & a_{32} & a_{33} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

Les composantes du produit matriciel fournissent les relations non-linéaires (l_{ij} sont les inconnues) suivantes :

$$a_{jj} = \sum_{k=1}^j l_{jk} l_{jk}^*$$

$$a_{ij} = \sum_{k=1}^j l_{ik} l_{jk}^* \quad i > j$$

L'algorithme suivant traite les colonnes les unes après les autres (algorithme *left looking*) pour calculer les composantes de L , en commençant par le coin supérieur gauche :

$$l_{jj} = \sqrt{a_{jj} - \sum_{k=1}^{j-1} l_{jk} l_{jk}^*}$$

$$l_{ij} = \frac{1}{l_{jj}} \left(a_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk}^* \right) \quad i > j$$



2.3 Méthodes par factorisation – Décomposition de Cholesky

- Exemple** : matrice réelle symétrique définie positive $A \in M_4(\mathbb{R})$

$$\begin{pmatrix} l_{11} & & & \\ l_{21} & l_{22} & & \\ l_{31} & l_{32} & l_{33} & \\ l_{41} & l_{42} & l_{43} & l_{44} \end{pmatrix} \cdot \begin{pmatrix} l_{11} & l_{21} & l_{31} & l_{14} \\ & l_{22} & l_{32} & l_{24} \\ & & l_{33} & l_{34} \\ & & & l_{44} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix}$$

La détermination de L se fait itérativement, colonne par colonne :

$$\left\{ \begin{array}{l} l_{11} = \sqrt{a_{11}} \\ l_{21} = \frac{a_{21}}{l_{11}} \\ l_{31} = \frac{a_{31}}{l_{11}} \\ l_{41} = \frac{a_{41}}{l_{11}} \end{array} \right\} \left\{ \begin{array}{l} l_{22} = \sqrt{a_{22} - l_{21}^2} \\ l_{32} = \frac{1}{l_{22}} (a_{32} - l_{31}l_{21}) \\ l_{42} = \frac{1}{l_{22}} (a_{42} - l_{41}l_{21}) \end{array} \right\} \left\{ \begin{array}{l} l_{33} = \sqrt{a_{33} - l_{31}^2 - l_{32}^2} \\ l_{43} = \frac{1}{l_{33}} (a_{43} - l_{41}l_{31} - l_{42}l_{32}) \\ l_{44} = \sqrt{a_{44} - l_{41}^2 - l_{42}^2 - l_{43}^2} \end{array} \right.$$



2.3 Méthodes par factorisation – Décomposition de Cholesky

- **Exemple** : Matrice réelle symétrique définie positive

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 13 & 27 \\ 3 & 27 & 62 \end{bmatrix}; \text{ Déterminer la matrice triangulaire supérieure } L \text{ telle que } A = L^T L.$$

(réponse : $L = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 3 & 7 \\ 0 & 0 & 2 \end{bmatrix}$)

Remarque : la résolution du système linéaire $A X = b$, où $A = LL^*$ correspond à résoudre d'abord le système triangulaire $Ly = b$ puis $L^* X = y$, de manière similaire à la décomposition LU ; Cependant ici le fait que A soit hermitienne et définie positive réduit le nombre d'opérations (environ 2 fois moins).



2.3 Méthodes par factorisation – *Elimination de Gauss*

Pour certaines matrices carrées, il est possible à l'aide d'une séquences de transformations simples $H_{(k)}$ de les rendre échelonnées et réduites, voire triangulaires. Pour cela, on procède itérativement sur chacune des colonnes.

A chaque étape, au moins un élément de la j -ème colonne sous la diagonale de la matrice obtenue à l'étape précédente est annulé par combinaison linéaire des lignes, c'est la méthode dite **d'élimination de Gauss ou méthode du pivot de Gauss ou méthode de Gauss-Jordan**.

Cette méthode va permettre d'annuler les éléments situés sous la diagonale de la matrice $A \in M_n(\mathbb{C})$, en commençant par la colonne la plus à gauche.

$$\begin{array}{cc}
 A = \begin{pmatrix} X & X & X & X \\ X & X & X & X \\ X & X & X & X \\ X & X & X & X \end{pmatrix} & H_1' A = \begin{pmatrix} X & X & X & X \\ & X & X & X \\ & X & X & X \\ & X & X & X \end{pmatrix} \\
 H_2' H_1' A = \begin{pmatrix} X & X & X & X \\ & X & X & X \\ & & X & X \\ & & & X \end{pmatrix} & H_3' H_2' H_1' A = \begin{pmatrix} X & X & X & X \\ & X & X & X \\ & & X & X \\ & & & X \end{pmatrix}
 \end{array}$$



2.3 Méthodes par factorisation – *Elimination de Gauss*

- **Exercice** : Soit la matrice $A = \begin{bmatrix} 2 & 1 & 1 \\ 2 & 3 & 2 \\ 1 & 1 & 2 \end{bmatrix}$; Résoudre le système linéaire $Ax = b$, avec $b = (1 \ 1 \ 1)^T$. Trigonaliser la matrice A par la méthode par élimination de Gauss. Indiquer comment est transformé le vecteur b .

1) Déterminer les transformations $H_{(k)}$ telles $A_{(k)} = H_{(k)} A_{(k-1)}$ et on pose $A = A_0$.

Itération 1 : 2^{ème} ligne soustraite à la 1^{ère} : Calculer $H_{(1)}$, puis $A_{(1)} = \begin{bmatrix} 2 & 1 & 1 \\ 0 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}$; $\rightarrow H_{(1)} = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Itération 2 : 2 fois la 3^{ème} ligne soustraite à la 1^{ère} : Calculer $H_{(2)}$ puis $A_{(2)} = \begin{bmatrix} 2 & 1 & 1 \\ 0 & 2 & 1 \\ 0 & 1 & 3 \end{bmatrix}$; $\rightarrow H_{(2)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 2 \end{bmatrix}$

Itération 3 : 2 fois la 3^{ème} ligne soustraite à la 2^{ème} : Calculer $H_{(3)}$ puis $A_{(3)} = \begin{bmatrix} 2 & 1 & 1 \\ 0 & 2 & 1 \\ 0 & 0 & 5 \end{bmatrix}$; $\rightarrow H_{(3)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 2 \end{bmatrix}$

2) Dans le système $Ax = b$, on a $b = (1 \ 1 \ 1)^T = b_{(0)}$

$$A_{(3)} = H_{(3)}H_{(2)}H_{(1)} A_{(0)} \rightarrow H_{(3)}H_{(2)}H_{(1)} A_{(0)}x = H_{(3)}H_{(2)}H_{(1)} b_{(0)}$$

Le vecteur $b_{(3)}$ vaut donc $b_{(3)} = H_{(3)}H_{(2)}H_{(1)} b_{(0)} \rightarrow b_{(3)} = (1 \ 0 \ 2)^T$



2.3 Méthodes par factorisation – *Elimination de Gauss*

Dans ce processus itératif, chaque itération k correspond donc au traitement d'un élément dans une colonne, sous la diagonale principale. C'est une combinaison linéaire des lignes qui est réalisée pour construire une matrice de transformation $H_{(k)}$, cette combinaison ne devant pas refaire apparaître des éléments non nuls sur les colonnes déjà traitées.

Pour une matrice $A \in M_{m,n}(\mathbb{C})$, jusqu'à $s = n(n-1)/2$ itérations seront nécessaires et donc autant de transformations $H_{(k)}$ seront à calculer.

Remarques :

1. Les matrices $H_{(k)}$ sont calculées pour transformer aussi le second membre b dans le système linéaire $AX = b$.
2. A chaque itération k le facteur d'échelle dans la matrice $H_{(k)}$ peut-être imposé pour maintenir une diagonale composée uniquement de 1. Dans ce cas, ces matrices $H_{(k)}$ sont appelées matrices de transformation de Gauss et prennent la forme d'une matrice triangulaire inférieure :

$$L = \begin{bmatrix} 1 & & & & \\ -m_{21}^{(1)} & 1 & & & \\ -m_{31}^{(2)} & -m_{32}^{(n)} & 1 & & \\ \vdots & \vdots & \ddots & 1 & \\ -m_{n,1}^{(n-1)} & -m_{n,2}^{(2n-3)} & \dots & -m_{n,n-1}^{(s)} & 1 \end{bmatrix} = H_{(s)} \dots H_{(2)} H_{(1)}$$



2.3 Méthodes par factorisation – *Elimination de Gauss*

Cependant comme alternative au calcul de la matrice $H_{(k)}$, on augmente la matrice initiale du système A pour former la matrice $\tilde{A} = [A \ b]$, c'est-à-dire avec une colonne supplémentaire où on a copié le contenu du vecteur b . Cela permet d'intégrer directement les transformations $H_{(k)}$ sur A et sur b .

Algorithme (processus d'élimination de Gauss)

- Initialisation : former la matrice augmentée $\tilde{A}_{(k=0)} = \tilde{A} = [A \ b]$,
- Pour chaque colonne j ($j = 1, \dots, n$) faire :
 - Pour chaque ligne i ($i = j + 1, \dots, n$) faire :
 - I. Calculer le rapport $m_{ij}^{(k)} = a_{ij}^{(k)} / a_{jj}^{(k)}$ (l'indice i correspond à l'équation linéaire (E_i) dans $\tilde{A}_{(k)}$ et $a_{ij}^{(k)}$ se situe sous la diagonale dans $\tilde{A}_{(k)}$. Ce calcul suppose que le pivot $a_{jj}^{(k)} \neq 0$),
 - II. Remplacer la ligne (E_i) de $\tilde{A}_{(k)}$ par $(E_i - m_{ij}^{(k)} E_j) \rightarrow \tilde{A}_{(k+1)}$
 - III. Incrémenter k

Remarque : La décomposition LU d'une matrice A consiste à la considérer comme le produit d'une matrice triangulaire inférieure L (*lower*) et d'une matrice triangulaire supérieure U (*upper*) $A = L U$. Une telle décomposition n'existe cependant pas toujours telle qu'elle. Cependant, en autorisant les permutations des lignes de la matrice A , la décomposition $A = P L U$, où P est la matrice de permutation est possible, en suivant le processus d'élimination de Gauss. Cette décomposition sera unique si on impose, par exemple, que tous les éléments diagonaux de la matrice L soient égaux à 1.



2.3 Méthodes par factorisation – *Elimination de Gauss*

- **Exercice** : Soit la matrice $A = \begin{bmatrix} 2 & 1 & 1 \\ 2 & 3 & 2 \\ 1 & 1 & 2 \end{bmatrix}$; Résoudre le système $Ax = b$ où $b = (1 \ 1 \ 1)^T$ en recherchant la matrice échelonnées réduite de A par l'application de l'élimination de Gauss.

- 1) Déterminer les matrices étendues $\widetilde{A}_{(k)} = [A_{(k)} \ b_{(k)}]$ et on pose $A_{(0)} = A$ et $b_{(0)} = b$. Les valeurs de k sont incrémentées à l'issue du traitement de l'itération.

$k=1$: 2^{ème} ligne soustraite à la 1^{ère} : $m_{21}^{(0)} = 1 \rightarrow \widetilde{A}_{(1)} = [A_{(1)} \ b_{(1)}] = \begin{bmatrix} 2 & 1 & 1 & 1 \\ 0 & 2 & 1 & 0 \\ 1 & 1 & 2 & 1 \end{bmatrix}$ et $H_{(1)} = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

$k=2$: 2 x la 3^{ème} ligne soustraite à la 1^{ère} : $m_{31}^{(1)} = \frac{1}{2} \rightarrow \widetilde{A}_{(2)} = [A_{(2)} \ b_{(2)}] = \begin{bmatrix} 2 & 1 & 1 & 1 \\ 0 & 2 & 1 & 0 \\ 0 & \frac{1}{2} & \frac{3}{2} & \frac{1}{2} \end{bmatrix}$ et $H_{(2)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1/2 & 0 & 1 \end{bmatrix}$

$k=3$: 4 x la 3^{ème} ligne soustraite à la 2^{ème} : $m_{32}^{(2)} = \frac{1}{4} \rightarrow \widetilde{A}_{(3)} = [A_{(3)} \ b_{(3)}] = \begin{bmatrix} 2 & 1 & 1 & 1 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & \frac{5}{4} & \frac{1}{2} \end{bmatrix}$ et $H_{(3)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1/4 & 1 \end{bmatrix}$

2) Résoudre le système $A_{(3)} x = b_{(3)}$, avec $b_{(3)} = (1 \ 0 \ 1/2)^T \rightarrow \begin{bmatrix} 2 & 1 & 1 \\ 0 & 2 & 1 \\ 0 & 0 & \frac{5}{4} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \frac{1}{2} \end{bmatrix}$

On détermine alors successivement les valeurs des composantes du vecteur $x = (x_1 \ x_2 \ x_3)^T$, en commençant par $x_3 : x = \frac{1}{5} (2 \ -1 \ 2)^T$



2.3 Méthodes par factorisation – *Elimination de Gauss* *avec pivotage partiel*

Remarque :

Il est fréquent que la quantité a_{jj} ait une valeur très faible, comparée aux valeurs des autres composantes a_{ij} de la même colonne sous la diagonale puisque ($i = j + 1, \dots, n$), ce qui va produire une valeur élevée pour le facteur m_{ij} et peut engendrer des erreurs dans les calculs subséquents, en particulier quand les formules de remontée ou de descente seront utilisées dans la résolution du système triangulaire.

Puisqu'il est préférable de ne pas interchanger les colonnes (ce qui nécessite de redéfinir les vecteurs X et b), la stratégie la plus simple, appelée pivotage partiel, consiste à interchanger deux lignes le cas échéant à chaque itération, en commençant par sélectionner comme pivot un autre élément situé sous la diagonale de la même colonne, et qui possède la valeur (absolue ou module) la plus élevée.

Supposons que cela concerne l'itération k . A la colonne j , l'indice recherché $p \geq j$ est tel que :

$$|a_{pj}^{(k)}| = \max_{j \leq i' \leq n} \{ |a_{i'j}^{(k)}| \}$$

$$p \leftarrow i'$$

La notation i' est choisie ici pour ne pas la confondre avec l'indice i des lignes dans l'implémentation. Pour autant, la recherche du maximum va être mise en œuvre en parcourant plusieurs lignes (de j à n), mais au sein d'une boucle dont l'incrément se nomme déjà i).



2.3 Méthodes par factorisation – *Elimination de Gauss avec pivotage partiel*

- **Exemple** : Soient la matrice $A = \begin{bmatrix} 1 & 1 & 2 \\ 2 & 1 & 1 \\ 2 & 3 & 2 \end{bmatrix} = A_{(0)}$ et le vecteur $b = (1 \ 1 \ 1)^T$

Avant la première itération, à l'initialisation, $k = 0$, la recherche de la valeur maximale de la première colonne correspond à (i' varie de $j = 1$ à $n = 3$)

$$\max\{|a_{11}^{(0)}| = 1, |a_{21}^{(0)}| = 2, |a_{31}^{(0)}| = 2\} \rightarrow p = 2$$

amenant à interchanger les lignes 1 et 2 ($p = 3$ aurait pu être aussi élu, mais souvent les algorithmes de recherche du maximum ne changent la valeur retenue (et son indice) que si un autre élément lui est strictement supérieur dans un parcours séquentiel des entrées).

$$\rightarrow \tilde{A}_{(k=0)} = \begin{bmatrix} 2 & 1 & 1 & b_2 \\ 1 & 1 & 2 & b_1 \\ 2 & 3 & 2 & b_3 \end{bmatrix}$$

Au début du traitement de la colonne 2, c'est-à-dire à $k = 2$ on a : $\tilde{A}_{(k=2)} = \begin{bmatrix} 2 & 1 & 2 & b_2 \\ 0 & 1 & 2 & 2b_1 - b_2 \\ 0 & 2 & 0 & b_3 - b_2 \end{bmatrix}$.

Le pivotage partiel nous amène à interchanger les lignes 2 et 3 : $\rightarrow \tilde{A}_{(k=2)} = \begin{bmatrix} 2 & 1 & 2 & b_2 \\ 0 & 2 & 0 & b_3 - b_2 \\ 0 & 1 & 2 & 2b_1 - b_2 \end{bmatrix}$

avant que la procédure de Gauss annule le coefficient $a_{32}^{(2)} = 1$ dans $a_{32}^{(3)} = 0$.



2.3 Méthodes par factorisation – *Elimination de Gauss avec pivotage partiel*

Le processus d'élimination de Gauss est alors modifié ainsi :

Algorithme (processus d'élimination de Gauss avec pivotage partiel)

- Initialisation : former la matrice augmentée $\tilde{A}_{(k=0)} = \tilde{A} = [A \ b]$,
- Pour chaque colonne j ($j = 1, \dots, n$) faire :
 - Pour chaque ligne i ($i = j + 1, \dots, n$) faire :
 - I. Rechercher $|a_{pj}^{(k)}| = \max_{j \leq i' \leq n} \{|a_{i'j}^{(k)}|\}$ (l'indice i' correspond à parcourir les $a_{i'j}^{(k)}$ se situant sur et sous la diagonale dans $\tilde{A}_{(k)}$). Si $p \neq j$, alors interchanger les lignes j et p dans $\tilde{A}_{(k)}$.
 - II. Calculer le rapport $m_{pj}^{(k)} = a_{ij}^{(k)} / a_{pj}^{(k)}$ (l'indice i correspond à l'équation linéaire (E_i) dans $\tilde{A}_{(k)}$ et le pivot $a_{pj}^{(k)} \neq 0$ sinon la colonne j n'est plus à traiter),
 - III. Remplacer la ligne (E_i) de $\tilde{A}_{(k)}$ par $(E_i - m_{pj}^{(k)} E_j) \rightarrow \tilde{A}_{(k+1)}$
 - IV. Incrémenter k

Remarque : il est à observer que l'interchangeabilité de deux lignes n'amène pas à considérer une autre notation pour la matrice $\tilde{A}_{(k)}$ avant et après.



2.4 Méthodes itératives – Résolution d'un système triangulaire

Considérons un système triangulaire supérieur $UX = b$ où U est une matrice triangulaire supérieure inversible d'ordre n . Par exemple, considérons un système inversible (3×3) triangulaire ci-dessous :

$$\begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}.$$

La matrice étant inversible, ses termes diagonaux u_{ii} sont tous non nuls. On peut donc envisager de déterminer successivement les valeurs inconnues x_i :

$$x_3 = b_3/u_{33} \quad x_2 = (b_2 - u_{23}x_3)/u_{22} \quad x_1 = (b_1 - u_{12}x_2 - u_{13}x_3)/u_{11}$$

Dans ce cas l'algorithme s'appelle *substitution rétrograde* et s'écrit dans le cas général des systèmes de dimension n :

$x_n = b_n/u_{nn}$ et, pour $i = n - 1, \dots, 1$, on a :

$$x_i = \frac{1}{u_{ii}} \left(b_i - \sum_{j=i+1}^n u_{ij}x_j \right)$$

Logiquement, on appelle ces relations *formules de «remontée»*.



2.4 Méthodes itératives – Résolution d'un système triangulaire

On traite de manière analogue la résolution d'un système linéaire inversible (3×3) triangulaire inférieur : $LX = b$

$$\begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}.$$

La matrice étant inversible, ses termes diagonaux l_{ii} sont tous non nuls. On peut donc envisager de déterminer successivement les valeurs inconnues x_i :

$$x_1 = b_1/l_{11} \quad x_2 = (b_2 - l_{21}x_1)/l_{22} \quad x_3 = (b_3 - l_{31}x_1 - l_{32}x_2)/l_{33}$$

Cet algorithme peut être généralisé aux systèmes de dimension n . On l'appelle *substitution directe*. Dans le cas d'un système $LX = b$ de dimension n , la matrice L est une matrice inversible triangulaire inférieure d'ordre n et la méthode s'écrit :

$x_1 = b_1/l_{11}$ et, pour $i = 2, \dots, n$, on a :

$$x_i = \frac{1}{l_{ii}} \left(b_i - \sum_{j=1}^{i-1} l_{ij}x_j \right)$$

On appelle ces relations *formules de «descente»*. $n(n+1)/2$ multiplications et divisions et $n(n-1)/2$ additions et soustractions sont effectuées par cet algorithme.



2.4 Méthodes itératives – Décomposition QR

• Transformation de Householder

La transformation de Householder est une transformation linéaire qui représente une réflexion sur un hyperplan contenant l'origine. Ici ne seront traitées que les matrices réelles.

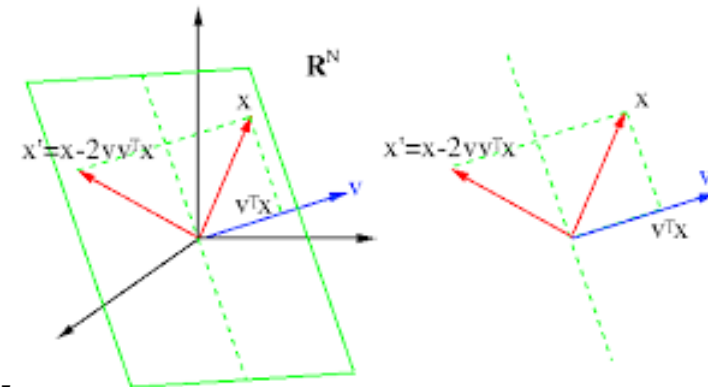
Etant donné un vecteur v à n composantes dans \mathbb{R} , une transformation de Householder est représentée par la matrice $H(v)$ suivante :

$$H(v) = I_n - 2 \frac{v v^T}{\|v\|^2}$$

La norme $\| \cdot \|$ entendue ici est la norme euclidienne.

La matrice de Householder a des propriétés importantes, elle est

- Symétrique : $H(v) = H(v)^T$
- Orthogonale : $H(v)^T H(v) = I_n - 4 \frac{v v^T}{\|v\|^2} + 4 \frac{v (v^T v) v^T}{\|v\|^4} = I_n$



- **Exemple** : $v = (9 \ 1 \ 5 \ 1)^T$ donne $H = I_4 - 2 \frac{v v^T}{\|v\|^2} = \frac{1}{54} \begin{bmatrix} -27 & -9 & -45 & -9 \\ -9 & 53 & -5 & -1 \\ -45 & -5 & 29 & -5 \\ -9 & -1 & -5 & 53 \end{bmatrix}$



2.4 Méthodes itératives – Décomposition QR

- Transformation de Householder

Par la suite, nous noterons e_i le i -ème vecteur $(\overbrace{0 \ 0 \ \dots \ 0}^{i-1} \ 1 \ 0 \ \dots \ 0 \ 0)^T$ à n composantes de la base canonique d'un espace de dimension à n .

Pour un vecteur x donné, si on choisit pour la transformation de Householder $H(v)$ un vecteur v tel que $v = x \pm \|x\| e_i$ alors $H(v)x = \mp \|x\| e_i$

En effet, $2v^T x = 2\|x\|^2 \pm 2\|x\|x_i$ et

$$\|v\|^2 = v^T v = (x \pm \|x\| e_i)^T (x \pm \|x\| e_i) = 2\|x\|^2 \pm 2\|x\|x_i = 2v^T x$$

et donc

$$H(v)x = x - 2 \frac{v v^T}{\|v\|^2} x = x - v = \mp \|x\| e_i$$

Remarque : pour éviter le phénomène d'annulation on choisit le signe qui mène à l'addition de deux nombres de même signe.

Le vecteur v est alors choisi tel que : $v = x + \text{sign}(x_i)\|x\| e_i$

- Exemple** : avec $x = (3 \ 1 \ 5 \ 1)^T$ on choisit $v = (9 \ 1 \ 5 \ 1)^T$ afin que $H(v)x = (-6 \ 0 \ 0 \ 0)^T$.



2.4 Méthodes itératives – Décomposition QR

- Exemple

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \\ 1 & 5 & 25 \\ 1 & 6 & 36 \end{bmatrix} \quad \text{avec } v^{(1)} = \begin{bmatrix} 1 + \sqrt{5} \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad \text{et} \quad Q_1 := I_5 - 2 \frac{v^{(1)} v^{(1)T}}{\|v^{(1)}\|^2}$$

$$\rightarrow Q_1 A = \begin{bmatrix} -\sqrt{5} & -7.603 & -33.541 \\ 0 & -0.658 & -6.674 \\ 0 & 0.342 & -1.674 \\ 0 & 2.342 & 14.326 \\ 0 & 3.342 & 25.326 \end{bmatrix}$$



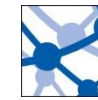
2.4 Méthodes itératives – Décomposition QR

• Exemple

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \\ 1 & 5 & 25 \\ 1 & 6 & 36 \end{bmatrix} \quad \text{avec } v^{(1)} = \begin{bmatrix} 1 + \sqrt{5} \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad \text{et} \quad Q_1 := I_5 - 2 \frac{v^{(1)} v^{(1)T}}{\|v^{(1)}\|^2}$$

$$\rightarrow Q_1 A = \begin{bmatrix} -\sqrt{5} & -7.603 & -33.541 \\ 0 & -0.658 & -6.674 \\ 0 & 0.342 & -1.674 \\ 0 & 2.342 & 14.326 \\ 0 & 3.342 & 25.326 \end{bmatrix} \quad \text{avec } v^{(2)} = \begin{bmatrix} -7.603 \\ -0.653 - 8.660 \\ 0.342 \\ 2.342 \\ 3.342 \end{bmatrix}$$

$$\text{et } Q_2 := I_5 - 2 \frac{v^{(2)} v^{(2)T}}{\|v^{(2)}\|^2}$$



2.4 Méthodes itératives – Décomposition QR

• Exemple

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \\ 1 & 5 & 25 \\ 1 & 6 & 36 \end{bmatrix} \quad \text{avec } v^{(1)} = \begin{bmatrix} 1 + \sqrt{5} \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad \text{et} \quad Q_1 := I_5 - 2 \frac{v^{(1)} v^{(1)T}}{\|v^{(1)}\|^2}$$

$$\rightarrow Q_1 A = \begin{bmatrix} -\sqrt{5} & -7.603 & -33.541 \\ 0 & -0.658 & -6.674 \\ 0 & 0.342 & -1.674 \\ 0 & 2.342 & 14.326 \\ 0 & 3.342 & 25.326 \end{bmatrix} \quad \text{avec } v^{(2)} = \begin{bmatrix} -7.603 \\ -0.653 - 8.660 \\ 0.342 \\ 2.342 \\ 3.342 \end{bmatrix}$$

$$\text{et } Q_2 := I_5 - 2 \frac{v^{(2)} v^{(2)T}}{\|v^{(2)}\|^2} \quad \rightarrow Q_2 Q_1 A = \begin{bmatrix} -0.635 & 0 & 7.419 \\ 1.963 & 8.660 & 43.532 \\ -0.072 & 0 & -3.514 \\ -0.4930 & 0 & 1.710 \\ -0.704 & 0 & 7.322 \end{bmatrix}$$

→ ce n'est pas ce qu'on voulait...



2.4 Méthodes itératives – Décomposition QR

• Exemple

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \\ 1 & 5 & 25 \\ 1 & 6 & 36 \end{bmatrix} \quad \text{avec } v^{(1)} = \begin{bmatrix} 1 + \sqrt{5} \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad \text{et} \quad Q_1 := I_5 - 2 \frac{v^{(1)} v^{(1)T}}{\|v^{(1)}\|^2}$$

$$\rightarrow Q_1 A = \begin{bmatrix} -\sqrt{5} & -7.603 & -33.541 \\ 0 & -0.658 & -6.674 \\ 0 & 0.342 & -1.674 \\ 0 & 2.342 & 14.326 \\ 0 & 3.342 & 25.326 \end{bmatrix} \quad \text{avec } v^{(2)} = \begin{bmatrix} -0.658 & -4.147 \\ 0.342 \\ 2.342 \\ 3.342 \end{bmatrix}$$

$$\text{et } Q_2 := \begin{bmatrix} 1 & 0 \\ 0 & I_4 - 2 \frac{v^{(2)} v^{(2)T}}{\|v^{(2)}\|^2} \end{bmatrix} \rightarrow Q_2 Q_1 A = \begin{bmatrix} -2.236 & -7.603 & -33.541 \\ 0 & 4.147 & 29.417 \\ 0 & 0 & -4.239 \\ 0 & 0 & -3.259 \\ 0 & 0 & 0.230 \end{bmatrix}$$



2.4 Méthodes itératives – Décomposition QR

• Exemple

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \\ 1 & 5 & 25 \\ 1 & 6 & 36 \end{bmatrix} \quad \text{avec } v^{(1)} = \begin{bmatrix} 1 + \sqrt{5} \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad \text{et } Q_1 := I_5 - 2 \frac{v^{(1)} v^{(1)T}}{\|v^{(1)}\|^2}$$

$$\rightarrow Q_1 A = \begin{bmatrix} -\sqrt{5} & -7.603 & -33.541 \\ 0 & -0.658 & -6.674 \\ 0 & 0.342 & -1.674 \\ 0 & 2.342 & 14.326 \\ 0 & 3.342 & 25.326 \end{bmatrix} \quad \text{avec } v^{(2)} = \begin{bmatrix} -0.658 - 4.147 \\ 0.342 \\ 2.342 \\ 3.342 \end{bmatrix}$$

$$\text{et } Q_2 := \begin{bmatrix} 1 & 0 \\ 0 & I_4 - 2 \frac{v^{(2)} v^{(2)T}}{\|v^{(2)}\|^2} \end{bmatrix} \rightarrow Q_2 Q_1 A = \begin{bmatrix} -2.236 & -7.603 & -33.541 \\ 0 & 4.147 & 29.417 \\ 0 & 0 & -4.239 \\ 0 & 0 & -3.259 \\ 0 & 0 & 0.230 \end{bmatrix}$$

$$\text{avec } v^{(3)} = \begin{bmatrix} -4.239 - 5.352 \\ -3.259 \\ 0.230 \end{bmatrix} \quad \text{et } Q_3 := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & I_3 - 2 \frac{v^{(3)} v^{(3)T}}{\|v^{(3)}\|^2} \end{bmatrix}$$



2.4 Méthodes itératives – Décomposition QR

- Exemple

$$\text{avec } v^{(3)} = \begin{bmatrix} -4.239 & -5.352 \\ -3.259 \\ 0.230 \end{bmatrix} \text{ et } Q_3 := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & I_3 - 2 \frac{v^{(3)} v^{(3)T}}{\|v^{(3)}\|^2} \end{bmatrix}$$

$$\rightarrow Q_3 Q_2 Q_1 A = \begin{bmatrix} -2.236 & -7.603 & -33.541 \\ 0 & 4.147 & 29.417 \\ 0 & 0 & 5.352 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Le produit des trois matrices orthogonales $Q^T = Q_3 Q_2 Q_1$ est lui-même une matrice orthogonale $Q^T Q = I$, et on voit que $Q^T A = R$ est une matrice trapézoïdale, de mêmes dimensions que la matrice A . Cet exemple nous a permis d'appréhender la décomposition QR que nous allons maintenant définir complètement.



2.4 Méthodes itératives – Décomposition QR

Les transformations de Householder ou les factorisations de Givens permettent d'exprimer la matrice $A \in M_{m,n}(\mathbb{R})$ d'un système linéaire $Ax = b$ sous la forme ($m \geq n$) :

$$A = QR = [Q_1 \quad Q_2] \begin{bmatrix} \tilde{R} \\ 0 \end{bmatrix}$$

où $0 = 0_{(m-n) \times n}$ est la matrice nulle, $Q_1 \in M_{m,n}(\mathbb{R})$, $Q_2 \in M_{m,(m-n)}(\mathbb{R})$ et la matrice triangulaire supérieure $\tilde{R} \in M_n(\mathbb{R})$, alors que la matrice orthogonale $Q \in M_m(\mathbb{R})$ et la matrice trapézoïdale supérieure $R \in M_{m,n}(\mathbb{R})$.

Si $rg(A) = n$, alors la matrice triangulaire \tilde{R} est inversible, les deux matrices ayant le même rang.

Si $rg(A) = n$, alors la pré-multiplication par A^T dans le système linéaire $Ax = b$ le transforme pour obtenir $A^T A x = A^T b$, qui peut alors s'écrire :

$$A^T A x = \tilde{R}^T Q_1^T Q_1 \tilde{R} x = \tilde{R}^T \tilde{R} x = A^T b = \tilde{R}^T Q_1^T b$$

La non-singularité de \tilde{R} entraîne que $\tilde{R}x = Q_1^T b$, qui est finalement résolu par les formules de remontées.



2.4 Méthodes itératives – Décomposition QR

- **Définition** : Une décomposition (ou factorisation) QR d'une matrice $A \in M_{m,n}(\mathbb{R})$ où $(m \geq n)$ et de rang $rg(A) = n$ est le produit qu'une matrice orthogonale Q , $Q \in M_m(\mathbb{R})$, et d'une matrice trapézoïdale supérieure $R \in M_{m,n}(\mathbb{R})$ tel que $A = QR$.

Remarques :

1. le produit $A = Q_1 \tilde{R}$ dans la décomposition $A = QR = [Q_1 \quad Q_2] \begin{bmatrix} \tilde{R} \\ 0 \end{bmatrix}$ où $Q_1 \in M_{m,n}(\mathbb{R})$ et $\tilde{R} \in M_n(\mathbb{R})$ est une matrice triangulaire supérieure ($Q_1^T Q_1 = I_n$) est appelée la factorisation réduite.
2. Si $rg(A) = r < n$, une décomposition QR est possible sous la forme modifiée :

$$Q^T AP = \begin{bmatrix} R_{11} & R_{12} \\ 0 & 0 \end{bmatrix}$$

où Q est une matrice orthogonale, P est une matrice de permutation et R_{11} est une matrice triangulaire supérieure de rang r .

3. On peut construire cette factorisation en utilisant soit le procédé d'orthogonalisation de Gram-Schmidt (peu robuste), soit les rotations de Givens, soit les matrices de transformation de Householder.



2.4 Méthodes itératives – Décomposition QR

L'algorithme suivant fournit la matrice $R = \begin{bmatrix} \tilde{R} \\ 0 \end{bmatrix}$ de mêmes dimensions que la matrice $A \in M_{m,n}(\mathbb{R})$ et une séquence de vecteurs $v^{(k=1)}, \dots, v^{(n)}$ laquelle définit implicitement la matrice Q , puisque :

$$Q_{(k)} := \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & & \vdots \\ \vdots & 0 & & 1 & 0 \\ 0 & 0 & I_{m-k+1} - 2 \frac{v^{(k)} v^{(k)T}}{\|v^{(k)}\|^2} \end{bmatrix} \quad \text{et} \quad Q^T = Q_{(m)} \dots Q_{(2)} Q_{(1)}$$

Alors que les vecteurs $v^{(k)}$ ont une dimension qui diminue proportionnellement à l'itération k , en commençant par un vecteur $v^{(1)}$ de m composantes, la construction des matrices $Q_{(k)}$ impose qu'elles aient toujours les mêmes dimensions – des matrices carrées ($m \times m$), ceci en vue de calculer la matrice orthogonale finale Q .

Algorithme QR (avec les transformations de Householder)

- Initialisation $A = R$ % Copier A dans R
- Pour $k = 1, \dots, n$
 - I. $x = R(k:m, k)$ % vecteur de $m - k + 1$ composantes, de k à m de la colonne k de la matrice R .
 - II. $v^{(k)} = x + \text{sign}(x_1) \|x\|_2 e_1$
 - III. $v^{(k)} = v^{(k)} / \|v^{(k)}\|_2$
 - IV. $R(k:m, k:n) = R(k:m, k:n) - 2v^{(k)} v^{(k)T} R(k:m, k:n)$



2.5 Décomposition en valeurs singulières - SVD

On a vu comment les valeurs propres et les vecteurs propres étaient définis pour une matrice carrée (cf. 1.2 (Espace propre)). Nous allons étudier ci-dessous le cas des matrices rectangulaires réelles.

Lorsque la matrice n'est pas carrée, on peut toujours déterminer les valeurs et vecteurs propres du produit AA^T ou du produit $A^T A$. Ainsi pour une matrice $A \in M_{m,n}(\mathbb{R})$, de rang r , les valeurs et vecteurs propres des produits AA^T et $A^T A$ vérifient :

$$(\lambda_i I_m - AA^T) u_i = 0 \quad \text{et} \quad (\lambda_i I_n - A^T A) v_i = 0 .$$

Si $n > m$, alors r ne peut être supérieur à m . La matrice AA^T , de dimensions $(m \times m)$, a donc $(m - r)$ valeurs propres nulles. La matrice $A^T A$, de dimensions $(n \times n)$, a donc $(n - r)$ valeurs propres nulles. Par conséquent, pour ces deux matrices, le nombre de valeurs propres non nulles est le même, celui du rang de la matrice A , avec $1 \leq r \leq \text{Min}\{m, n\}$.

On définit les **valeurs singulières**, σ_i , de la matrice A comme étant les racines carrées des valeurs propres de AA^T ou de $A^T A$

$$\sigma_i = \sqrt{\lambda_i} .$$

Remarque : si $\lambda_i < 0$, alors σ_i est un nombre complexe.



2.5 Décomposition en valeurs singulières - SVD

Pour la matrice $A \in M_{m,n}(\mathbb{R})$, **chaque vecteur singulier à gauche**, u_i , de dimensions $(m \times 1)$ et chaque **vecteur singulier à droite**, v_i , de dimensions $(n \times 1)$ sont associés à la même valeur singulière σ_i , et sont des vecteurs (unitaires) définis respectivement par :

$$A v_i = \sigma_i u_i \text{ et } A^T u_i = \sigma_i v_i$$

La décomposition en valeurs singulières (*Singular Value Decomposition* – SVD) de la matrice A est donnée par le produit :

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

où $U = (u_1, u_2, \dots, u_m)$; $V = (v_1, v_2, \dots, v_n)$ sont des matrices orthogonales et Σ , de dimensions $(m \times n)$, est de la forme suivante

si $n > m$:

$$\Sigma = \begin{bmatrix} \sigma_1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_m & 0 & \cdots & 0 \end{bmatrix} = [D_m \quad 0_{m \times (n-m)}] .$$



2.5 Décomposition en valeurs singulières - SVD

ou de la forme suivante si $m > n$:

$$\Sigma = \begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_n \\ 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{bmatrix} = \begin{bmatrix} D_n \\ 0_{(m-n) \times n} \end{bmatrix}.$$

D_m et D_n sont des matrices diagonales contenant les valeurs singulières. Les colonnes des matrices U et V renferment alors les vecteurs singuliers. Il s'ensuit que :

$$U^T A V = \Sigma \in M_{m,n}(\mathbb{R})$$

La plupart des algorithmes qui mettent en oeuvre la SVD rangent les valeurs singulières non-nulles (il y en a $r = \text{rg}(A)$) en ordre décroissant de leur module (ou valeur absolue si elles sont réelles). Il s'ensuit que les matrices U et V ont leurs colonnes réarrangées pour être toujours associées à la même valeur singulière :

$$|\sigma_1| \geq |\sigma_2| \geq \dots \geq |\sigma_r| > 0.$$



2.5 Décomposition en valeurs singulières - SVD

- Si $n > m$, alors $\underbrace{V}_{(n \times n)} = \left[\underbrace{V_m}_{(n \times m)} \mid \underbrace{V_{n-m} \equiv 0}_{n \times (n-m)} \right]$ avec $V_m = A^T U D_m^{-1}$

Car $A = U \Sigma V^T = U \begin{bmatrix} D_m & 0 \end{bmatrix} \begin{bmatrix} V_m^T \\ 0^T \end{bmatrix} = U D_m V_m^T$ et comme $AA^T u_i = \sigma_i^2 u_i$, on en déduit que $(AA^T - \sigma_i^2 I_m) u_i = 0$, pour chacun des vecteurs singuliers u_i de $U = (u_1, u_2, \dots, u_m)$.

- Si $m > n$, alors $\underbrace{U}_{(m \times m)} = \left[\underbrace{U_n}_{(m \times n)} \mid \underbrace{U_{m-n} \equiv 0}_{m \times (m-n)} \right]$ avec $U_n = A V D_n^{-1}$

Car $A = U \Sigma V^T = \begin{bmatrix} U_n & 0 \end{bmatrix} \begin{bmatrix} D_n \\ 0^T \end{bmatrix} V^T = U_n D_n V^T$ et comme $A^T A v_i = \sigma_i^2 v_i$, on en déduit que $(A^T A - \sigma_i^2 I_n) v_i = 0$, pour chacun des vecteurs singuliers v_i de $V = (v_1, v_2, \dots, v_n)$.



2.5 Décomposition en valeurs singulières - SVD

Remarques :

- Toute matrice réelle a une décomposition en valeurs singulières.
- si A est une matrice symétrique ($A = A^T$) alors $V = U$
- si A est une matrice de rotation, alors $A^{-1} = A^T$ et $A = \det(U) \det(V) UV^T$
- La pseudo-inverse de A , A^+ , est égale à $A^+ = V \Sigma^+ U^T$
- La SVD peut être utilisée pour résoudre des systèmes inhomogènes ($A X = b$), comme alternative à l'utilisation de la pseudo-inverse. La SVD est une réduction (d'endomorphisme) par des matrices orthogonales et par conséquent, est fondée sur la norme 2 et la norme de Frobenius. En particulier, on a :

$$\|A\|_2 = |\sigma_1| \quad \text{et} \quad \|A\|_F = \sqrt{\sigma_1^2 + \sigma_2^2 + \dots + \sigma_n^2}$$

(car $\|U^T A V\| = \|\Sigma\|$ pour les deux normes)

- La SVD s'emploie souvent pour résoudre les systèmes linéaires homogènes $A X = 0$ surdéterminés ($m > n$) pour résoudre avec plus d'équations que d'inconnues. Ceci consiste à déterminer une solution pour X , telle que $\|X\|^2 = 1$ en cherchant à minimiser la quantité $\|A X\|^2$ qui prend en compte toutes les équations du système (on impose $\|X\|^2 = 1$ pour viser les solutions autres que $X = 0$). La SVD fournit les matrices U , V et Σ . Si le rang de A vaut $r = n - 1$, La solution pour X est alors donnée par la dernière colonne de la matrice V (quand les valeurs singulières sont rangées en ordre décroissant).



2.5 Décomposition en valeurs singulières - SVD

Exercice :

Considérons la matrice $A = \begin{bmatrix} 5 & 1 & 3 \\ -1 & 2 & 0 \end{bmatrix}$;

- 1) Former $A A^T$ (ou $A^T A$). Calculer ses valeurs propres.
- 2) En déduire les valeurs singulières de A . Former alors les matrices D_2 , puis Σ .
- 3) Résoudre $AA^T u_i = \sigma_i^2 u_i$. En déduire la matrice $U = [u_1, u_2]$.

On peut alors calculer la matrice $V = A^T U D_2^{-1}$.

*Conseil : Utiliser Matlab pour mettre en œuvre cet exercice, en utilisant entre autre les commandes : **eig(A'*A) ; eig(A*A') ; et [U,D,V]=svd(A) ;***



2.5 Décomposition en valeurs singulières - SVD

Exercice :

Considérons la matrice $A = \begin{bmatrix} 5 & 1 & 3 \\ -1 & 2 & 0 \end{bmatrix}$;

- 1) Former $A A^T$ (ou $A^T A$). Calculer ses valeurs propres.
- 2) En déduire les valeurs singulières de A . Former alors les matrices D_2 , puis Σ .
- 3) Résoudre $A A^T u_i = \sigma_i^2 u_i$. En déduire la matrice $U = [u_1, u_2]$.

On peut alors calculer la matrice $V = A^T U D_2^{-1}$.

*Conseil : Utiliser Matlab pour mettre en œuvre cet exercice, en utilisant entre autre les commandes : `eig(A'*A)` ; `eig(A*A')` ; et `[U,D,V]=svd(A)` ;*

$$\text{Réponses : } \lambda_{1,2} = 20 \pm \sqrt{936}/2 \approx \begin{cases} 35,297 \\ 4,703 \end{cases} \quad \sigma_{1,2} \approx \begin{cases} 5,941 \\ 2,168 \end{cases}$$

$$D_2 = \begin{bmatrix} 5,941 & 0 \\ 0 & 2,168 \end{bmatrix} ; \Sigma = \begin{bmatrix} 5,941 & 0 & 0 \\ 0 & 2,168 & 0 \end{bmatrix} ; u_1 = \begin{bmatrix} -0,995 \\ 0,098 \end{bmatrix} ; u_2 = \begin{bmatrix} 0,098 \\ 0,995 \end{bmatrix}$$

$$V = \begin{bmatrix} -0,854 & -0,231 & -0,4657 \\ -0,134 & 0,963 & -0,2328 \\ -0,502 & 0,136 & 0,8538 \end{bmatrix}$$



2.5 Décomposition en valeurs singulières - SVD

Exercice : Cas d'un système sous-déterminé (déjà vu)

Etant donné le vecteur $X = (x_1, x_2, x_3)^T$ des inconnues, on considère le système classique suivant où $x_1^2 + x_2^2 + x_3^2 = 1$ et

$$\begin{cases} x_1 + 2x_2 + 3x_3 = 0 \\ x_1 + x_2 = 0 \end{cases}$$

- Déterminer directement toutes les solutions pour X en résolvant le $A X = 0$ à l'aide de la décomposition en valeurs singulières de A .
- Comparez avec l'exercice 3 du transparent 37



2.5 Décomposition en valeurs singulières - SVD

Remarques :

- La relation matricielle $A = U \Sigma V^T$ peut s'écrire aussi de manière développée (formule de Eckart-Young) :

$$A = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \cdots + \sigma_r u_r v_r^T = \sum_{i=1}^{i=r} \sigma_i u_i v_i^T$$

où r est le rang de A et où $U = [u_1, \dots, u_r, \dots, u_m]$ et $V = [v_1, \dots, v_r, \dots, v_n]$.

- Si k est un nombre entier positif tel que $k < rg(A)$ et si (la formule ci-dessus stoppée au terme k)

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

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

Ceci signifie que la matrice A_k est la matrice de rang k la plus proche de A au sens de la norme de Frobenius, et que la « distance » entre les deux est donnée par le module de σ_{k+1} .



2.5 Décomposition en valeurs singulières - SVD

Application à la compression :

On applique aussi la SVD pour l'approximation d'une matrice ($m \times n$) de rang r par une matrice de rang réduit, k qui peut être imposé. C'est une contrainte qui permet de contrôler le niveau d'approximation par le rang. Le nombre de valeurs propres non nulles considérées peut alors être beaucoup plus petit ($k \ll r$) :

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

(les valeurs singulières $\sigma_{k+1}, \sigma_{k+2}, \dots, \sigma_r$ sont alors négligées)

1. Comment se comporte l'approximation de la matrice (dégradation de l'image) ci-dessous si on réduit son rang ?
2. Quelle réduction du volume de données pouvons-nous obtenir ?



(50 Ko)

matrice (168x300) de rang = 168



2.5 Décomposition en valeurs singulières - SVD

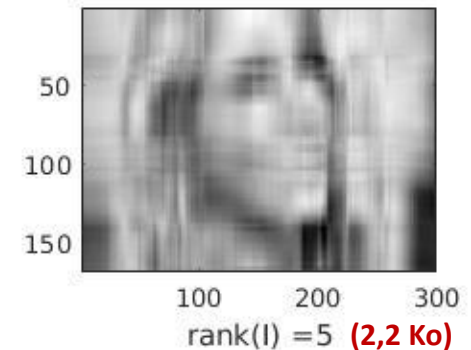
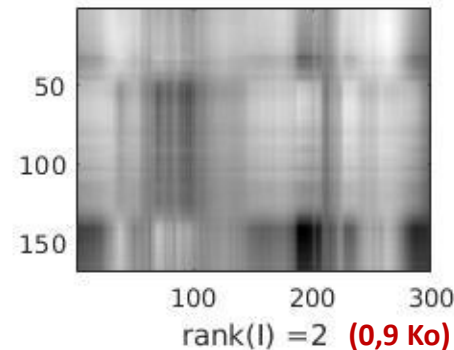
Application à la compression : (implémentation sous Matlab)

```
// build a double-precision matrix with an image
ima      = imread('./figures/chapitre-2/bl.jpg');
ima_g    = rgb2gray(ima);
X        = double(ima_g);

rank(X)

[U,D,V]  = svd(X);
vecteurk = [2 5 15 50];

// rank reduction
for i=1:length(vecteurk)
    k = vecteurk(i);
    I = U(:,1:k)*D(1:k,1:k)*V(:,1:k)';
    subplot(2,2,i);imagesc(I);
end
```



matrice (168x300) de rang = 168 (50 Ko)

