Recherche des valeurs et vecteurs propres

A la fin du chapitre, l'étudiant doit **être capable de** :

1. Décrire la méthode de la puissance itérée pour la détermination des vecteurs propres
2. Décrire l'algorithme de Jacobi pour la diagonalisation des matrices et expliquer ses faiblesses
3. Définir la transformation de Householder pour l'annulation de presque toutes les composantes d'un vecteur quelconque
4. Expliquer comment les transformations de Housholder peuvent être utiliser pour réaliser la factorisation QR d'une matrice, pour obtenir la forme Hessenberg d'une matrice.
5. Décrire l'algorithme QR pour la détermination des valeurs propres d'une matrice
6. Faire un choix éclairé entre les méthodes du cours pour la résolution d'un problème donné

Introduction

Deux grandes classes de problèmes en algèbre linéaire :

- **Résoudre un système linéaire d'équations**

$$\mathbf{AX} = \mathbf{b}$$

  Déjà vu au chapitre précédent

- **Trouver les éléments propres d'une matrices**

$$\mathbf{AX} = \boldsymbol{\lambda}\mathbf{X}$$

  X: vecteur propre
  λ: valeur propre

  L'objectif de ce chapitre

Pourquoi faire ?

- La détermination des éléments propres d’une matrice carrée a de multiples applications:
  - résonnance (balançoire, pont),
  - traitement d’image,
  - géométrie,
  - recherche sur le web,
  - Vibration (voiture, bruit de cabine,…)
  - marché financier,
  - résolution de certaines équations aux dérivées partielles (équation de la chaleur),
  - des systèmes d’équations différentielles ordinaires (EDO),
  - …

Résolution de AX=λX

- Rappel du cours sur les systèmes linéaires:

$\mathbf{MX}=\mathbf{b}$ avec $\mathbf{M}$ matrice carrée admet :

- 1 solution unique si : $\mathbf{det(M)} \neq \mathbf{0}$

- 0 ou une infinité de solutions si : $\mathbf{det(M)} = \mathbf{0}$

Dans le cas $\mathbf{det(M)} = \mathbf{0}$, on dit que $\mathbf{M}$ est singulière

- Conséquence pour le problème aux valeurs propres:

Puisque $\mathbf{X}=\mathbf{0}$ est toujours solution de $\left(\mathbf{A}-\boldsymbol{\lambda}\mathbf{I}\right)\mathbf{X}=0$ et que l'on cherche des solutions $\mathbf{X}\neq\mathbf{0}$, on se place dans le cas où $\mathbf{M}=\mathbf{A}-\boldsymbol{\lambda}\mathbf{I}$ est singulière

Autrement dit on cherche les valeurs propres $\boldsymbol{\lambda}$ de $\mathbf{A}$ en résolvant l'équation :

$$\mathbf{det}\left(\mathbf{A}-\boldsymbol{\lambda}\mathbf{I}\right)=\mathbf{0}$$

Résolution de $AX=\lambda X$

- Théoriquement, elle se fait en <u>3 étapes </u>systématiques:

1. Calcul du déterminant de $\mathbf{A-\lambda I}$

2. Détermination des racines du polynôme caractéristique obtenu en écrivant : $\det(\mathbf{A-\lambda I})=0$

3. Pour chaque racine (chaque valeur propre), résoudre le système linéaire $\mathbf{AX=\lambda X}$ afin de déterminer le ou les vecteurs propres associés.

Quelques résultats à connaître

Théorème 1 :

Si A et T sont des matrices carrées d'ordre n et si T est inversible, alors la matrice $B = T^{-1}AT$ a les mêmes valeurs propres que la matrice A et si X est le vecteur propre de A attaché à λ, le vecteur propre de B attaché à cette même valeur propre est $Z = T^{-1}X$.

Cas particulier important : $T = R$ orthogonale ($R^{-1} = R^T$).

Théorème 2 :

La somme des carrés des éléments d'une matrice symétrique est égale à la somme des carrés de ses valeurs propres (2nd invariant)

Théorème 3 :

Si A est diagonale, les valeurs propres de A sont égales aux coefficients sur la diagonale.

Recherche des valeurs et vecteurs propres

- En pratique, cette démarche n'est pas exploitable pour les systèmes de taille supérieure à 4 ou 5 car le calcul formel de det(A-λI) est rapidement monstrueux …
- On préfère donc les méthodes itératives dans lesquelles on s'approche pas à pas des valeurs propres.
- Dans ce cours :
  - Puissance itérée: la plus simple, mais aussi très limitée
  - Méthode de Jacobi: très lente et non pratique pour des problèmes de taille supérieure à 10
  - Algorithme QR: la méthode de référence pour des problèmes de taille inférieure à quelques milliers.

I- Méthode de la puissance itérée et de déflation

I-1 Puissance itérée

- Matrice *A* réelle quelconque diagonalisable
- On note $\lambda_1 \le \lambda_2 \le \cdots \le \lambda_{n-1} < \lambda_n$ ses valeurs propres et V_1, V_2,…, V_{n-1}, V_n les vecteurs propres associés
- On se donne un vecteur initial X_0 non nul. Ce vecteur peut s'exprimer dans la base propre de A:

$$X_0 = c_1 V_1 + c_2 V_2 + \cdots + c_n V_n$$

- Si l'on calcule X_1 = A X_0 , on obtient un vecteur qui s'écrit:

$$X_1 = c_1 \lambda_1 V_1 + c_2 \lambda_2 V_2 + \cdots + c_n \lambda_n V_n$$

- Plus généralement, le k^ème itéré $X_k = A\, X_{k-1}$ s'exprime de la manière suivante dans la base propre de A:

$$X_k = c_1 \lambda_1^k V_1 + c_2 \lambda_2^k V_2 + \cdots + c_n \lambda_n^k V_n$$

- Deux utilisations possibles de la puissance itérée:

1. Si la valeur propre λ_n est déjà connue (calculée par une autre méthode comme QR), la suite des vecteurs X_k / λ_n^k tend vers un vecteur colinéaire au vecteur propre V_n car:

$$\frac{X_k}{\lambda_n^k} = c_1 \left(\frac{\lambda_1}{\lambda_n} \right)^k V_1 + c_2 \left(\frac{\lambda_2}{\lambda_n} \right)^k V_2 + \cdots + c_{n-1} \left(\frac{\lambda_{n-1}}{\lambda_n} \right)^k V_{n-1} + c_n V_n$$

2. Dans le cas contraire, la suite des vecteurs $X_k / \|X_k\|$ tend vers un vecteur unitaire, colinéaire au vecteur propre . La valeur propre est la limite de la suite $AX_k \bullet X_k / \|X_k\|^2$

I-2 Exemple 2×2

$$A = \begin{pmatrix} 10 & 0 \\ -9 & 1 \end{pmatrix}$$

- Les valeurs propres sont $\lambda_1 = 1$ et $\lambda_2 = 10$, et les vecteurs propres associés sont respectivement $V_1 = (0,1)^t$ et $V_2 = (1,-1)^t$,
- On choisit $X_0 = (1,2)^t$ et on obtient:

Itération (k)	$X_k / \lVert X_k \rVert$	$AX_k \bullet X_k / \lVert X_k \rVert^2$
1	0.4472 0.8944	-0.8000
2	0.8192 -0.5735	11.2685
3	0.7178 -0.6963	10.1349
4	0.7082 -0.7060	10.0135
5	0.7072 -0.7070	10.0013
6	0.7071 -0.7071	10.0001
7	0.7071 -0.7071	10.0000

Remarques

- La convergence est d'autant plus rapide que le rapport λ_n/λ_{n-1} est grand
- La méthode nécessite un produit matrice-vecteur à chaque itération; *A* doit donc de préférence être de faible dimension, ou très creuse
- La convergence peut être accélérer en utilisant la méthode « shifted inverse » dans laquelle on applique la puissance itérée à la matrice $(A-\alpha I)^{-1}$, où α est une (bonne) approximation de la valeur propre recherchée.
- Malgré ces améliorations, la puissance itérée conserve le défaut principal de ne converger que vers une seule valeur propre à la fois (même si une technique appelée déflation permet d'éliminer la convergence vers la valeur propre de plus grand module)

Principes de base des méthodes de calcul des valeurs propres

- Deux matrices semblables partagent les mêmes valeurs propres
- Les valeurs propres d'une matrice triangulaire sont ses éléments diagonaux
- **<u>Idée générale</u>**: Faire apparaître, à l'aide d'une succession de transformations élémentaires, des zéros sur les éléments extra-diagonaux et obtenir:

$$\mathbf{B} = \mathbf{T}^{-1}\,\mathbf{A}\,\mathbf{T}$$

  avec **D** matrice triangulaire (supérieure) et **T** inversible

- Pour se faciliter les choses, on demande également que **T** soit orthogonale de sorte que son inversion ne coûte rien. On a alors:

$$\mathbf{B} = \mathbf{T}^{t}\,\mathbf{A}\,\mathbf{T}$$

- La méthode de Jacobi est la plus simple des méthodes de cette catégorie

Principe de la méthode de Jacobi (sur une matrice 3×3)

L'idée est d'éliminer rotation après rotation, les « poids » hors diagonale. Comme le poids global se conserve, tout ce qui disparaît hors diagonal a des chances de se retrouver sur la diagonale….

par exemple, a_{23} est le terme hors diagonal le plus « lourd »;
la rotation R associée s'écrit :

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{12} & a_{22} & a_{23} \\ a_{13} & a_{23} & a_{33} \end{pmatrix} \qquad R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{pmatrix} \qquad R^T = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix}$$

on calcule alors $B = R^T A R$

$$B = \begin{pmatrix} a_{11} & a_{21}\cos\theta - a_{31}\sin\theta & a_{21}\sin\theta + a_{31}\cos\theta \\ a_{21}\cos\theta - a_{31}\sin\theta & a_{22}\cos^2\theta - a_{23}\sin 2\theta + a_{33}\sin^2\theta & a_{23}\cos 2\theta + \dfrac{(a_{22}-a_{33})}{2}\sin 2\theta \\ a_{21}\sin\theta + a_{31}\cos\theta & a_{23}\cos 2\theta + \dfrac{(a_{22}-a_{33})}{2}\sin 2\theta & a_{22}\sin^2\theta + a_{23}\sin 2\theta + a_{33}\cos^2\theta \end{pmatrix}$$

$$b_{23} = a_{23}\cos 2\theta + \frac{(a_{22}-a_{33})}{2}\sin 2\theta$$

et on détermine alors θ pour que b_{23}=0

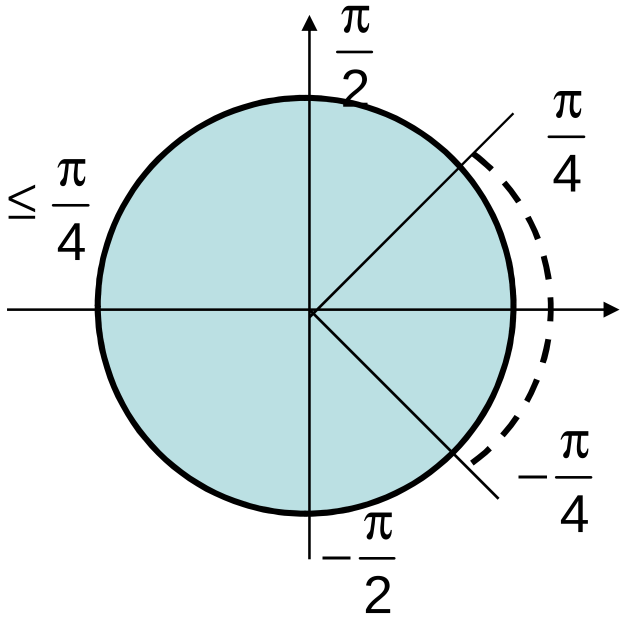
$$\boxed{\mathrm{tg}\,2\theta = \frac{-a_{23}}{\frac{1}{2}(a_{22}-a_{33})}}$$

en fait, on a seulement besoin de connaître cosθ et sinθ pour construire R.

On pose alors :

$$\begin{cases} \nu = -a_{23} \\ \mu = \frac{1}{2}(a_{22}-a_{33}) \end{cases}$$

$$\tan 2\theta = \frac{\nu}{\mu} \in \mathbf{R} \qquad -\frac{\pi}{4} \leq \theta \leq \frac{\pi}{4}$$



$$1+\mathrm{tg}^2 2\theta = \frac{1}{\cos^2 2\theta} = 1+\frac{\nu^2}{\mu^2} = \frac{\mu^2+\nu^2}{\mu^2}$$

D'où

$$\cos 2\theta = \frac{|\mu|}{\sqrt{\mu^2+\nu^2}} \qquad \text{et comme} \qquad \cos^2 2\theta + \sin^2 2\theta = 1$$

$$\sin 2\theta = \frac{\nu}{\sqrt{\mu^2+\nu^2}} \operatorname{sgn}\mu = \omega$$

Or : $\cos^2\theta = \frac{1}{2}(1+\cos 2\theta) = \frac{1}{2}\left(1+\sqrt{1-\sin^2 2\theta}\right) = \frac{1}{2}\left(1+\sqrt{1-\omega^2}\right)$

$$\boxed{\cos\theta = \sqrt{\frac{1}{2}\left(1+\sqrt{1-\omega^2}\right)}}$$

Par ailleurs, $\sin 2\theta = 2\sin\theta\cos\theta$, on obtient :

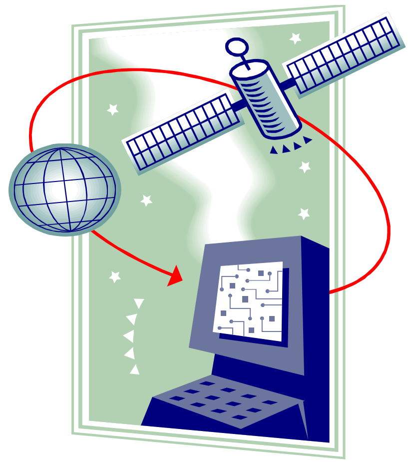
$$\boxed{\sin\theta = \frac{\omega}{\sqrt{2\left(1+\sqrt{1-\omega^2}\right)}}}$$

b– cas général

Les calculs précédents se généralisent aux cas n×n.
A chaque nouvelle itération (r), on veut éliminer le terme hors diagonale de plus fort poids.

Soit, $a_{pq}^{(r)}$ ce terme hors diagonale. On construit alors $R_{(pq)}^{(r)}$

$$R_{(pq)}^{(r)} = \begin{array}{cc}
\begin{pmatrix}
1 & 0 & \cdots & \cdots & \cdots & 0 & 0 \\
0 & \cos\theta & \cdots & \cdots & \sin\theta & \cdots & 0 \\
\vdots & 0 & 1 & 0 & \cdots & \cdots & \vdots \\
\vdots & \cdots & \cdots & \ddots & \cdots & \cdots & \vdots \\
\vdots & -\sin\theta & \cdots & \cdots & \cos\theta & \cdots & \vdots \\
0 & \cdots & \cdots & \cdots & \cdots & \ddots & 0 \\
0 & 0 & \cdots & \cdots & \cdots & 0 & 1
\end{pmatrix}
&
\begin{array}{c} \\ p \\ \\ \\ q \\ \\ \\ \end{array}
\\
\begin{array}{ccccccc} & p & & & q & & \end{array} &
\end{array}$$



Dans le cas général, on peut montrer que dans le produit

$$A^{(r+1)} = R^{\mathrm{T}(r)}_{(pq)} A^{(r)} R^{(r)}_{(pq)}$$

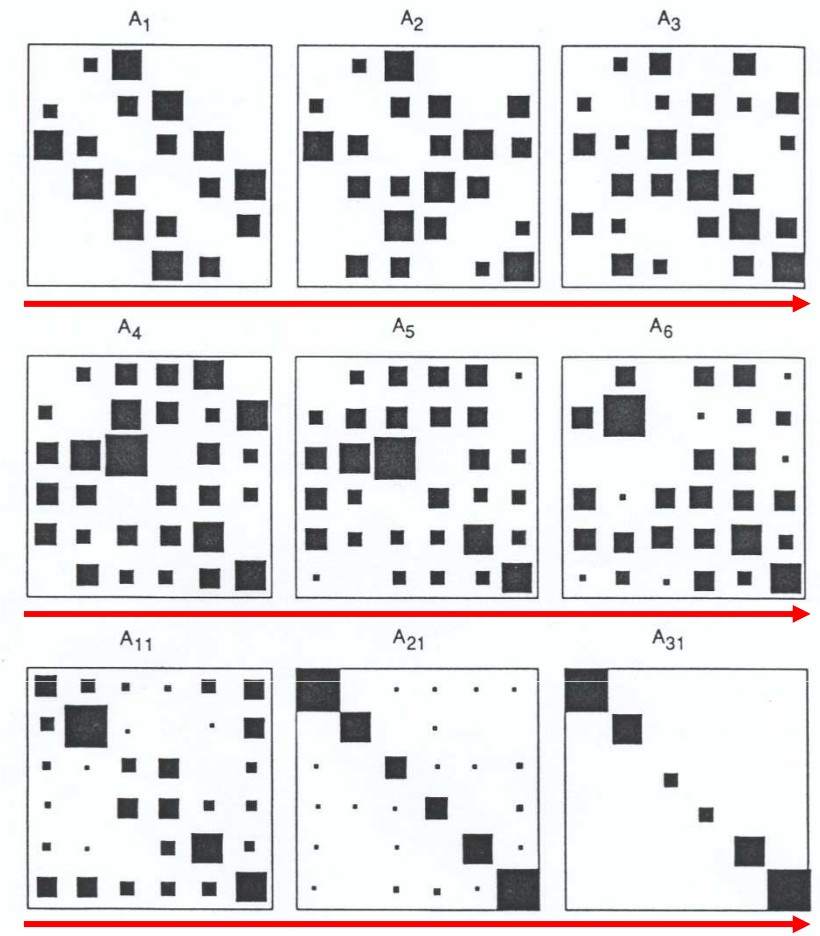
les coefficients de $A^{(r+1)}$ ne diffèrent de ceux de $A^{(r)}$ que dans les lignes et les colonnes p, q.

$$A^{(r+1)} = \begin{array}{cc} \left(\begin{array}{ccccccc} a^{(r)} & \cdots & \cdots & \cdots & \cdots & \cdots & a^{(r)} \\ \vdots & a^{(r+1)} & \cdots & \cdots & a^{(r+1)} & \cdots & \vdots \\ \vdots & & \ddots & \cdots & \cdots & \cdots & \vdots \\ \vdots & \cdots & \cdots & \ddots & \cdots & \cdots & \vdots \\ \vdots & a^{(r+1)} & \cdots & \cdots & a^{(r+1)} & \cdots & \vdots \\ \vdots & \cdots & \cdots & \cdots & \cdots & \ddots & \cdots \\ a^{(r)} & \cdots & \cdots & \cdots & \cdots & \cdots & a^{(r)} \end{array}\right) & \begin{array}{c} \\ p \\ \\ \\ q \\ \\ \\ \end{array} \\ \begin{array}{ccccccc} \quad & p & \quad & \quad & q & \quad & \quad \end{array} & \end{array}$$

On retrouve en particulier un résultat de la forme :

$$a_{pq}^{(r+1)} = a_{pq}^{(r)} \cos 2\theta + \frac{(a_{pp}^{(r)} - a_{qq}^{(r)})}{2} \sin 2\theta$$

$$A^{(1)} = \begin{pmatrix} 0 & 1 & 6 & 0 & 0 & 0 \\ 1 & 0 & 2 & 7 & 0 & 0 \\ 6 & 2 & 0 & 3 & 8 & 0 \\ 0 & 7 & 3 & 0 & 4 & 9 \\ 0 & 0 & 8 & 4 & 0 & 5 \\ 0 & 0 & 0 & 9 & 5 & 0 \end{pmatrix}$$



$$A^{(31)} = \begin{pmatrix} 16.61 & 0 & 0 & 0 & 0 & 0 \\ 0 & 5.94 & 0 & 0 & 0 & 0 \\ 0 & 0 & -2.46 & 0 & 0 & 0 \\ 0 & 0 & 0 & -12.13 & 0 & 0 \\ 0 & 0 & 0 & 0 & -10.06 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2.20 \end{pmatrix}$$

Détermination des vecteurs propres :

Il existe deux stratégies possibles:

1. Utiliser la méthode de la puissance itérée en mode « shift and invert » pour déterminer les vecteurs propres associés à chaque valeur propre

2. Poursuivre l'algorithme de Jacobi jusqu'à obtenir une matrice diagonale semblable à **A**:

   $$\mathbf{D} = \mathbf{T}^t\,\mathbf{A}\,\mathbf{T}, \qquad \text{avec } \mathbf{D} \text{ matrice diagonale.}$$

   Les vecteurs propres sont alors les colonnes de **T**.

Remarques

- Chaque rotation modifie certains des coefficients précédemment annulés puisque l'ensemble des entrées des lignes et colonnes p et q est modifié

- Globalement la contribution des termes extra-diagonaux diminue donc l'algorithme converge. On ne sait cependant pas dire a priori combien de rotations devront être faites pour obtenir une matrice triangulaire de la matrice

- Pour éviter ce problème, et pour gagner en efficacité, une autre stratégie consiste à ne pas annuler les entrées les unes après les autres, mais plutôt colonne (ou bout de colonne) par colonne …

Transformation de Householder

- Une transformation orthogonale permet en un seul coup de faire apparaître des zéros sur toute une colonne: c'est la transformation de Householder
- Pour **X** vecteur j quelconque de taille *m*, on définit:

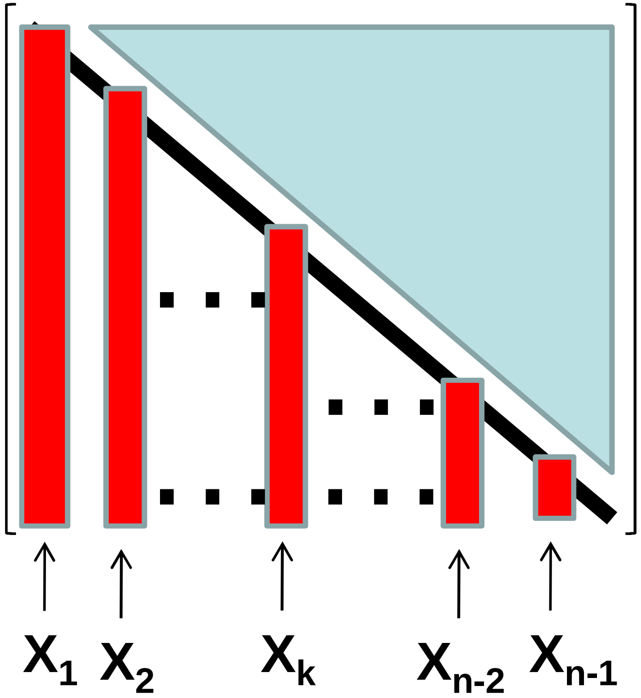
$$\mathbf{H} = \mathbf{I} - 2\frac{\mathbf{V}\mathbf{V}^{\mathbf{t}}}{\|\mathbf{V}\|^{2}} \quad \text{avec} \quad \mathbf{V} = \mathbf{X} + \|\mathbf{X}\|\mathbf{e_1}, \text{ où } \mathbf{e_1} = \underbrace{(1,0,0,\cdots,0)^t}_{m \text{ composantes}}$$

- On vérifie facilement que $\mathbf{H}=\mathbf{H}^{\mathbf{t}}$ puis que $\mathbf{H}\mathbf{H}^{\mathbf{t}} = \mathbf{I}$. Par suite, on obtient $\mathbf{H}^{-1}=\mathbf{H}^{\mathbf{t}}$ et donc **H** est bien orthogonale en plus d'être symétrique
- Un calcul simple permet alors de vérifier que:

$$\mathbf{H}\mathbf{X} = -\|\mathbf{X}\|\mathbf{e_1} = \left(-\|\mathbf{X}\|,0,0,\cdots,0\right)^t$$

Factoristion QR

- Partant d'une matrice **A** quelconque, on peut faire apparaître les vecteurs colonnes suivants



Les vecteurs colonne $\mathbf{X_1}$, $\mathbf{X_2}$, …,$\mathbf{X_k}$, …et $\mathbf{X_{n-1}}$ ont pour taille n, n-1, …, n-k+1, …et 2 respectivement,

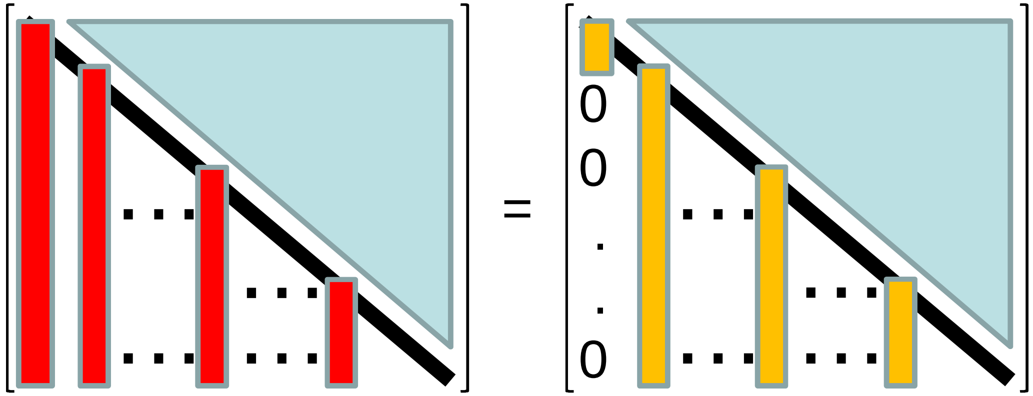
L'idée est d'appliquer une transformation de Householder adéquate successivement à chacun de ces vecteurs afin d'annuler tous les termes sous diagonaux

Plus précisément …

Factoristion QR

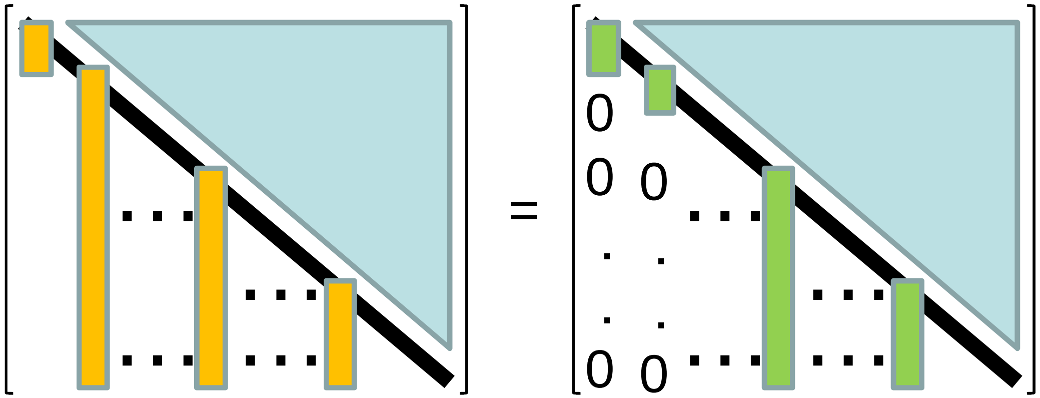
- Pour annuler les n-1 dernières composantes de $\mathbf{X_1}$:

$$\underbrace{\left[\begin{array}{l} \mathbf{H}_1 = \mathbf{I} - 2\dfrac{\mathbf{V}\mathbf{V}^t}{\|\mathbf{V}\|^2} \\ \text{avec } \mathbf{V} = \mathbf{X}_1 + \|\mathbf{X}_1\|\mathbf{e_1}, \\ \text{où } \mathbf{e_1} = \underbrace{(1,0,0,\cdots,0)^t}_{n \text{ composantes}} \end{array}\right]}_{\mathbf{U_1}}$$



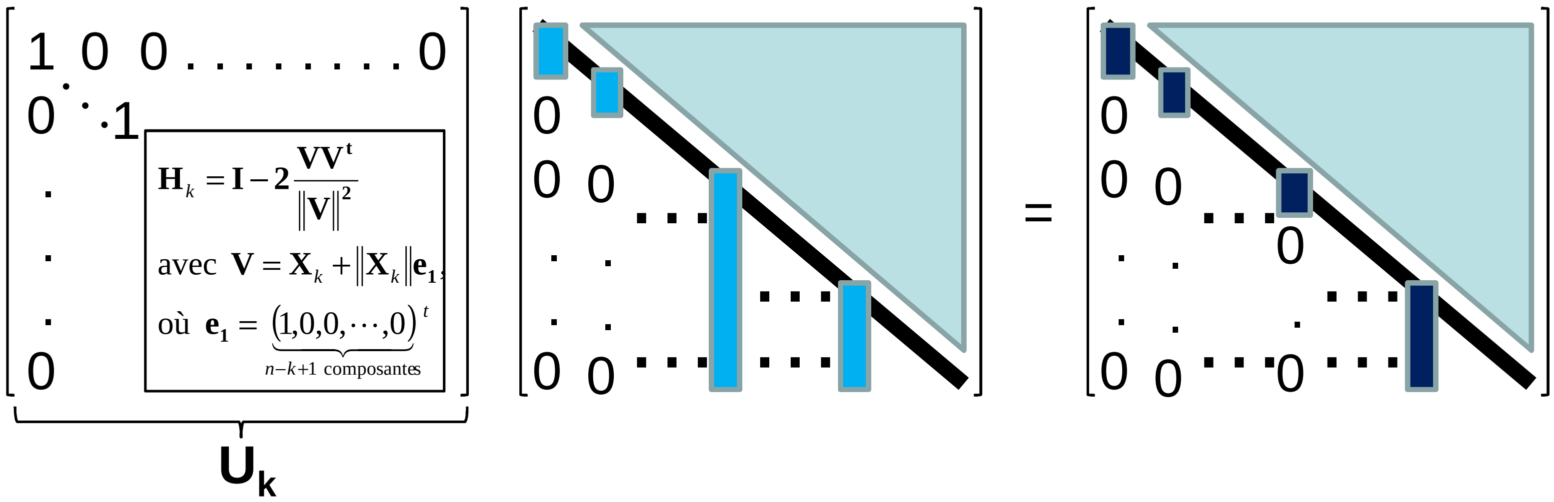
- Pour annuler les n-2 dernières composantes du nouveau $\mathbf{X_2}$:

$$\underbrace{\left[\begin{array}{ll} 1 & 0 \quad 0 \;.\;.\;.\;.\;.\;.\;.\;.\; 0 \\ 0 & \boxed{\begin{array}{l} \mathbf{H}_2 = \mathbf{I} - 2\dfrac{\mathbf{V}\mathbf{V}^t}{\|\mathbf{V}\|^2} \\ \text{avec } \mathbf{V} = \mathbf{X}_2 + \|\mathbf{X}_2\|\mathbf{e_1}, \\ \text{où } \mathbf{e_1} = \underbrace{(1,0,0,\cdots,0)^t}_{n-1 \text{ composantes}} \end{array}} \\ . & \\ . & \\ . & \\ 0 & \end{array}\right]}_{\mathbf{U_2}}$$



Factoristion QR

- Pour annuler les n-k dernières composantes du nouveau $\mathbf{X_k}$:



- Au final, après avoir appliqué n-1 transformations de Householder bien choisies on obtient (en posant $\mathbf{Q^t = U_{n-1}U_{n-2} \ldots U_2U_1}$) la factorisation **QR** de la matrice **A**:

$$\mathbf{A = QR}$$

avec **Q** matrice orthogonale et **R** matrice triangulaire supérieure

Exemple sur une matrice de taille 4:

On fait d'abord apparaitre des zéros sur la première colonne:

$$\mathbf{A} = \begin{bmatrix} 1 & 3 & 2 & 1 \\ 1 & 2 & 1 & 2 \\ 2 & -1 & 2 & 1 \\ -1 & -2 & 3 & 4 \end{bmatrix} \qquad \mathbf{X_1} = \begin{bmatrix} 1 \\ 1 \\ 2 \\ -1 \end{bmatrix} \qquad \mathbf{V_1} = \mathbf{X_1} + \|\mathbf{X_1}\| \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 4.6458 \\ 1 \\ 2 \\ -1 \end{bmatrix}$$

$$\mathbf{U_1} = \mathbf{H_1} = \mathbf{I} - 2\frac{\mathbf{V_1 V_1^t}}{\|\mathbf{V_1}\|^2} = \begin{bmatrix} -0.7559 & -0.3780 & 0.3780 & 0.3780 \\ -0.3780 & 0.9186 & 0.0814 & 0.0814 \\ 0.3780 & 0.0814 & 0.9186 & -0.0814 \\ 0.3780 & 0.0814 & -0.0814 & 0.9186 \end{bmatrix}$$

$$\mathbf{U_1 A} = \begin{bmatrix} -2.6458 & -4.1576 & 0.0000 & 0.3780 \\ 0 & 0.4593 & 0.5695 & 1.8661 \\ 0 & 0.5407 & 2.4305 & 1.1339 \\ 0 & -0.4593 & 3.4305 & 4.1339 \end{bmatrix}$$

Exemple sur une matrice de taille 4:

On procède de la même manière pour les colonnes 2 et 3 avec:

$$\mathbf{X}_2 = \begin{bmatrix} 0.4593 \\ 0.5407 \\ -0.4593 \end{bmatrix} \quad \mathbf{V}_2 = \mathbf{X}_2 + \|\mathbf{X}_2\| \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1.3045 \\ 0.5407 \\ -0.4593 \end{bmatrix} \quad \mathbf{H}_2 = \mathbf{I} - 2\frac{\mathbf{V}_2 \mathbf{V}_2^t}{\|\mathbf{V}_2\|^2}$$

$$\mathbf{U}_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & & & \\ 0 & & \mathbf{H}_2 & \\ 0 & & & \end{bmatrix} \qquad \mathbf{U}_2\mathbf{U}_1\mathbf{A} = \begin{bmatrix} -2.6458 & -4.1576 & 0.0000 & 0.3780 \\ 0 & -0.8452 & 0.0000 & 0.5071 \\ 0 & 0 & 2.1945 & 0.5706 \\ 0 & 0 & 3.6310 & 4.6124 \end{bmatrix}$$

$$\mathbf{X}_3 = \begin{bmatrix} 2.1945 \\ 3.6310 \end{bmatrix} \quad \mathbf{V}_3 = \mathbf{X}_3 + \|\mathbf{X}_3\| \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 6.4371 \\ 3.6310 \end{bmatrix} \quad \mathbf{H}_3 = \mathbf{I} - 2\frac{\mathbf{V}_3 \mathbf{V}_3^t}{\|\mathbf{V}_3\|^2}$$

$$\mathbf{U}_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & & \\ 0 & 0 & \mathbf{H}_3 & \end{bmatrix} \qquad \underbrace{\mathbf{U}_3\mathbf{U}_2\mathbf{U}_1}_{\mathbf{Q}^t}\mathbf{A} = \underbrace{\begin{bmatrix} -2.6458 & -4.1576 & 0.0000 & 0.3780 \\ 0 & -0.8452 & 0.0000 & 0.5071 \\ 0 & 0 & -4.2426 & -4.2426 \\ 0 & 0 & 0 & 1.8974 \end{bmatrix}}_{\mathbf{R}}$$

Remarques sur la factorisation QR

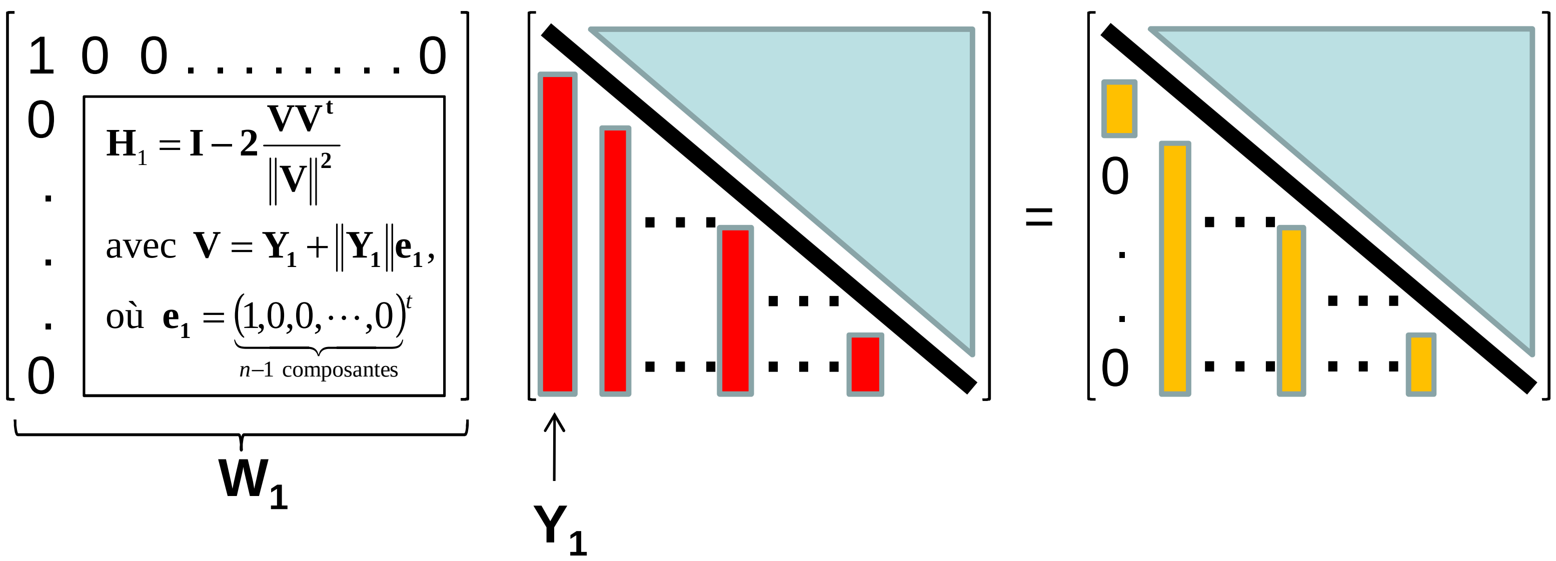
- Toute matrice peut être écrite sous la forme **QR**
- On utilise pour cela une combinaison de transformations de Householder
- Cette factorisation est très importante en analyse linéaire (un peu comme la factorisation **LU**)
- Comme det(Q)=±1, elle permet notamment de calculer le déterminant de **A** comme le produit des termes diagonaux de **R**
- Pour le calcul des valeurs propres, elle est à la base d’un algorithme très performant: l’algorithme **QR**

La suite …

- Partant de **A** on a donc obtenu par transformations orthogonale une matrice **R** dont les valeurs propres se calculent de manière immédiate (les termes diagonaux de **R**)
- Oui mais les valeurs propres de **A** et de **R** ne sont pas les mêmes car ces matrices ne sont pas semblables !
- Pour obtenir une matrice semblable à **A** il faut une matrice de la forme $\mathbf{Q}^t\mathbf{AQ}$
- Avec **A = QR** on obtient facilement**:** $\mathbf{RQ} = \mathbf{Q}^t\mathbf{AQ}$
- Donc **A** est semblable à **RQ**, qui n’est pas triangulaire !!
- Le problème est que la multiplication à droite par la transformation **Q** détruit la triangularisation obtenue par $\mathbf{Q}^t$ …

La suite …

- Pour surmonter cette difficulté, l'idée consiste à être un petit peu moins ambitieux et générer des zéros en dessous de la sous diagonale
- Par exemple pour la première colonne de **A**, on ne cherche à annuler que les n-2 dernières composantes en excluant le terme diagonal du vecteur colonne $\mathbf{Y_1}$ sur lequel est basé la transformation de Householder :

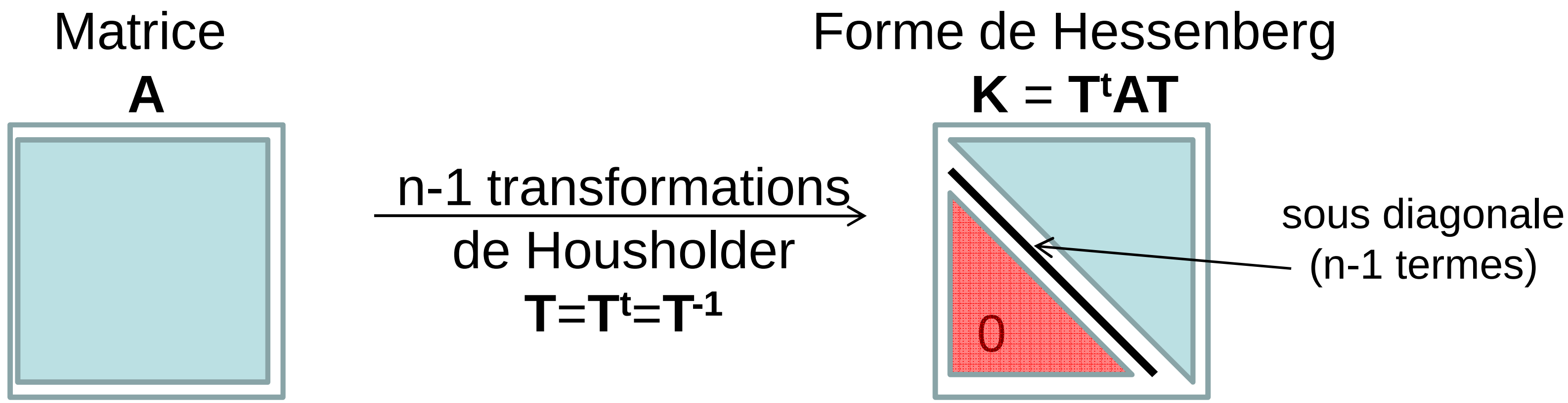


Forme de Hessenberg d'un matrice

- On procède alors comme pour la factorisation **QR**, en traitant successivement toutes les colonnes de **A** mais en excluant tous les termes diagonaux pour construire les transformations de Householder successives $\mathbf{W_1}, \mathbf{W_2}$, **…** et $\mathbf{W_{n-2}}$
- En posant $\mathbf{T^t = W_{n-2} \ldots W_2\, W_1}$ on définit donc une transformation orthogonale telle que **A** = **TB** où **B** est une matrice dont tous les coefficients en dessous de la sous-diagonale sont nuls. C'est une matrice de Hessenberg
- L'intérêt est que la matrice **K** = **BT** = $\mathbf{T^tAT}$ est semblable à **A** tout en conservant une forme de Hessenberg
- Evidemment, l'intérêt est que **A** et **K** sont semblables et qu'elles partagent les mêmes valeurs propres, avec **K** ayant presque tous ses éléments sous diagonaux égaux à zéros

Forme de Hessenberg d’un matrice

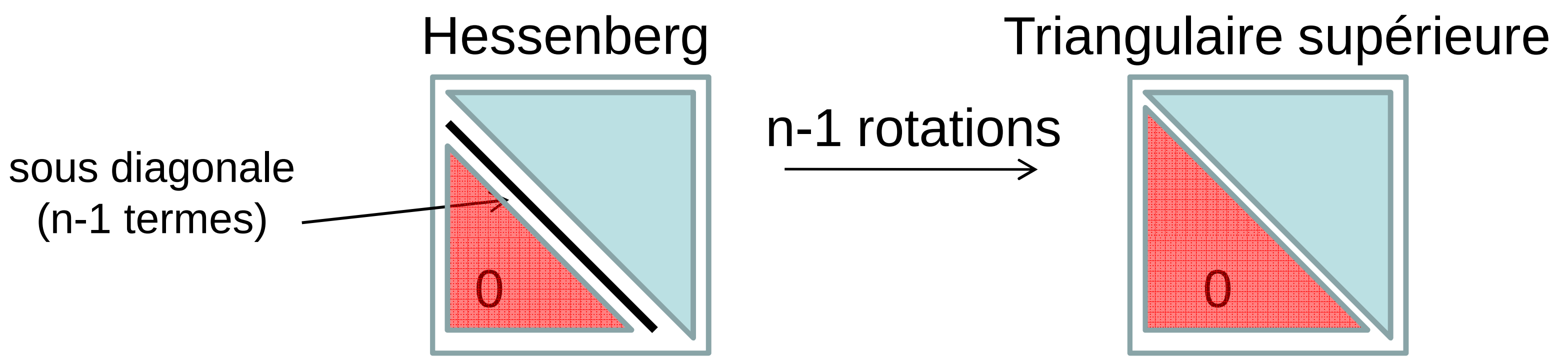
- Pour résumer:



- Les matrices de Hessenberg ne sont pas triangulaires et l’on ne connait donc pas directement leur valeurs propres
- Leur intérêt réside dans le fait que l’on peut facilement les rendre triangulaire avec une méthode déjà rencontrée: la méthode de Jacobi

Factorisation QR

- Partant d’une matrice de Hessenberg, il faut exactement n-1 rotations pour générer une matrice triangulaire:



- C’est un processus rapide puisque nécessitant O(n) opérations arithmétiques
- On obtient alors la factorisation **QR** de la matrice de Hessenberg

$$\mathbf{K} = \mathbf{QR}$$

avec **K** Hessenberg, **Q** orthogonale, **R** triangulaire supérieure

Algorithme QR

- L'idée de base est très simple:
- Partant de **K** = **QR** on obtient **R**=**Q**^{t}**K** puis **RQ**=**Q**^{t}**KQ**
- Autrement dit la factorisation **QR** de **K** permet de produire une matrice semblable à **K**, soit la matrice **RQ**
- L'algorithme est alors le suivant:

$$\mathbf{K}^{(k)} = \mathbf{Q}^{(k)}\,\mathbf{R}^{(k)}$$

$$\mathbf{K}^{(k+1)} = \mathbf{R}^{(k)}\,\mathbf{Q}^{(k)}$$

- A convergence, les valeurs propres de **K** sont les termes diagonaux de **R**