

# Méthodes numériques pour problèmes de grande taille

## Calcul de valeurs propres Résolution de système linéaire

Nahid EMAD

Université Paris Saclay/ Versailles

(Laboratoire LI-PARAD, Maison de la Simulation)

2022-2023

# Sommaire

Exemples d'application

Éléments de base pour le calcul de valeurs propres

Calcul de valeurs propres

Sélection des valeurs propres

Méthodes à étudier

Rappels

Méthode des puissances

Généralisation de la méthode des puissances

Documents et références

# Exemples d'application

## ► Phénomènes discrets

Soient  $k$  un pas de temps et un phénomène ou processus discret décrit par  $X(k) = (x_1, \dots, x_n)$  tel que:  $X(k) = A.X(k-1)$  où  $A$  est une matrice carée, donc  $X(k) = A^k X(0)$  avec  $X(0)$  l'état initial du phénomène.

- Si  $A$  diagonal, alors le calcul de  $X(k)$  est facile.
- Si  $A$  diagonalisable ( $A = VDV^{-1}$ )  $\Rightarrow A^k = VD^k V^{-1}$  avec  $D = \text{diag}(\lambda_1, \dots, \lambda_n)$ . Donc,  $X(k) = VD^k V^{-1} X(0)$ .

Un exemple d'un tel phénomène est la probabilité de la position d'un preneur sur la toile à pas de temps régulier (position initiale au premier clique, puis positions suivantes à chaque nouveau clique).

## Exemples d'application (suite)

### ► Phénomènes continus - systèmes différentiels

On considère un phénomène en évolution continue dans le temps décrit par  $X(t)$ . On suppose qu'il s'agit d'un système différentiel linéaire comme : 
$$\begin{cases} \frac{dX(t)}{dt} = A.X(t) & \text{pour } t > 0 \\ X(0) = X_0 \end{cases}$$
 où  $A$  est une matrice carée.

Le problème est alors d'étudier la stabilité asymptotique de ce système (limite de  $X(t)$  quand  $t \rightarrow +\infty$ ).

Si  $A$  est diagonalisable, la stabilité de ce système peut être vérifiée à l'aide des valeurs propres de la matrice  $A$ .

# Calcul de valeurs propres

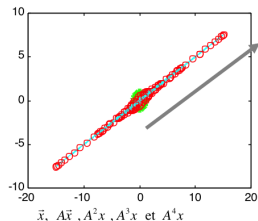
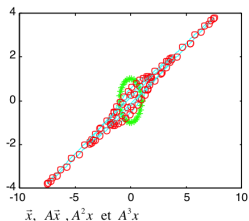
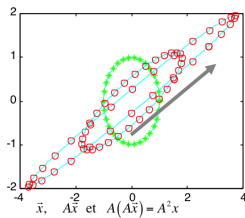
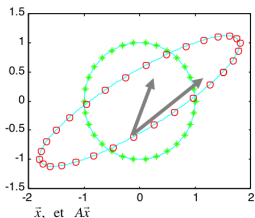
**Problème :** Soit  $A$  une matrice carrée de taille  $n$  (sauf mention du contraire, elle est à coefficients complexes). Il s'agit de calculer la valeur propre  $\lambda_i$  et le vecteur propre associé  $u_i$  (pour  $i = 1, \dots, n$ ) vérifiant:

$$Au_i = \lambda_i u_i \quad (1)$$

Nous supposons que les valeurs propres sont ordonnées de la manière suivante:

$$|\lambda_1| \geq |\lambda_2| \geq \dots \geq |\lambda_n|$$

Calcul de valeurs/vecteurs propres: Illustration avec  $A = \begin{pmatrix} 3 & 1 \\ -1 & 1 \end{pmatrix}$ .



Direction  
propre

# Calcul de valeurs propres: Disques/Cercles de Gerschgorin

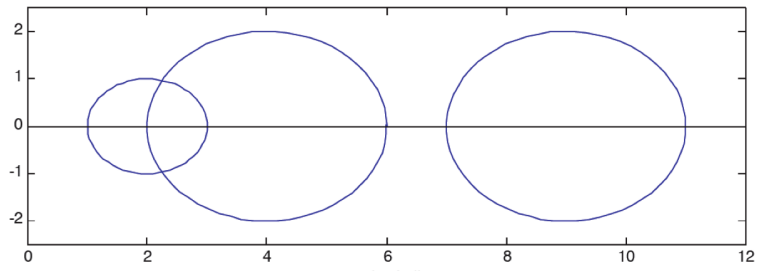
Toutes les valeurs propres de  $A$  sont incluses dans l'ensemble  $\cup_{i=1}^n D_i$  où  $D_i$  est le disque fermé du plan complexe du centre  $a(i, i)$  et du rayon  $R_i = \sum_{j \neq i} |a(i, j)|$ . Autrement dit:

Soit  $D_i$  le cercle du plan complexe défini par :

$D_i = D(a(i, i), R_i) = \{z \in \mathbb{C} \mid |z - a(i, i)| \leq R_i\}$ . Alors, toute valeur propre de  $A$  est dans au moins l'un des cercles  $D_i$ .

# Calcul de valeurs propres: Cercles de Gerschgorin avec

$$A = \begin{pmatrix} 4 & 1 & 1 \\ 0 & 2 & 1 \\ -2 & 0 & 9 \end{pmatrix} \text{ avec } D_1 = (4, 2), D_2 = (2, 1) \text{ et } D_3 = (9, 2).$$





## Cercles de Gerschgorin (démonstration)

$$\begin{aligned}
 Au_i = \lambda_i u_i &\Leftrightarrow \sum_{j=1}^n a(i, j) u_i(j) = \lambda_i u_i(i) \quad \forall i = 1, n \\
 &\Leftrightarrow \sum_{j=1, j \neq i}^n a(i, j) u_i(j) = \lambda_i u_i(i) - a(i, i) u_i(i) \\
 &= (\lambda_i - a(i, i)) u_i(i)
 \end{aligned}$$

Par conséquent:

$$\begin{aligned}
 |u_i(i)| |\lambda_i - a(i, i)| &\leq \sum_{j=1, j \neq i}^n |a(i, j)| |u_i(j)| \Rightarrow \\
 |\lambda_i - a(i, i)| &\leq \sum_{j=1, j \neq i}^n |a(i, j)| \frac{|u_i(j)|}{|u_i(i)|}
 \end{aligned}$$

En choisissant  $i$  tel que  $\frac{|u_i(j)|}{|u_i(i)|} \leq 1 \quad \forall j = 1 : n$  et  $j \neq i$ , on aura:

$$|\lambda_i - a(i, i)| \leq \sum_{j=1, j \neq i}^n |a(i, j)|.$$

Toute valeur propre de  $A$  appartient à l'un au moins des disques de Gerschgorin. Toutes les valeurs propres de  $A$  se trouvent dans la réunion des disques de Gerschgorin associés aux colonnes

# Calcul de valeurs propres: Généralité

**Décomposition de Schur<sup>1</sup> d'une matrice:** Pour toute matrice  $A$ , il existe une matrice  $U$  unitaire ( $U^H U = U U^H = I$  donc  $U^H = U^{-1}$ ) telle que la matrice  $T = U^H A U$  soit triangulaire supérieure.

- ▶ **Remarque 1:** Si  $\lambda$  est une valeur propre de  $A$ , alors  $(\lambda - \mu)$  et  $(\lambda - \mu)^{-1}$  sont les valeurs propres de  $(A - \mu I)$  et de  $(A - \mu I)^{-1}$ .
- ▶ **Remarque 2:** La forme de Schur n'est pas unique car on peut ordonner les éléments diagonaux de  $T$ .
- ▶ **Remarque 3:** Si  $A$  a  $n$  valeurs propres distinctes, elle est diagonalisable.

<sup>1</sup>Isaac Schur: Allemagne (1875-1941). Directeur de thèse: Frobenius.

# Calcul de valeurs propres: Généralité (suite)

## Forme de Jordan d'une matrice:

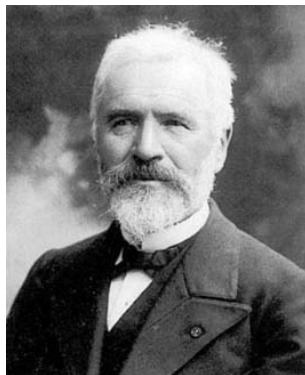


Figure: Camille Jordan, Lyon, France (1838-1922)

## Calcul de valeurs propres: Généralité (suite)

**Forme de Jordan d'une matrice:** Soit  $J_r$  une matrice de Jordan de taille  $r$  :

$$J_r = \begin{pmatrix} \lambda & 1 & 0 & 0 & \dots & 0 \\ 0 & \lambda & 1 & 0 & \dots & 0 \\ & & & \ddots & & \\ 0 & 0 & \dots & 0 & \lambda & 1 \\ 0 & 0 & \dots & 0 & 0 & \lambda \end{pmatrix} \quad \text{si } r > 1 \text{ et } J_1 = \lambda \text{ si } r = 1.$$

**Décomposition de Jordan d'une matrice:** Pour toute matrice  $A$ , il existe une matrice  $J$  diagonale par bloc (avec des blocs sous forme de Jordan) unitairement semblable à  $A$ . Autrement dit, il existe une matrice unitaire  $V$  telle que  $J = V^{-1}AV^2$ .

<sup>2</sup> Soient  $E$  et  $F$  deux espaces vectoriels sur un même corps  $K$ . Une application  $h : E$  vers  $F$  est appelé un "homomorphisme d'espaces vectoriels" ou encore une "application linéaire". Lorsque  $E = F$  on dit l'endomorphisme' de  $E$  au lieu d'application linéaire de  $E$  dans  $E$ .

## Calcul de valeurs propres: Généralité (suite)

La matrice par blocs de Jordan  $J$  a la forme suivante:

$$J = \begin{pmatrix} J_1 & & & 0 \\ & J_2 & & \\ & & \ddots & \\ 0 & & & J_k \end{pmatrix}.$$

**Définition:** La multiplicité algébrique d'une valeur propre est le nombre de fois où elle apparaît dans le spectre de la matrice. La multiplicité géométrique d'une valeur propre est le nombre de vecteurs propres linéairement indépendants associés à celle-ci.

**Exercice:** Quelle est le lien entre la MA et la MG d'une valeur propre et la taille des blocs dans  $J$ ?

## Calcul de valeurs propres: Généralité (suite)

**Similitude à une matrice quasi triangulaire** Toute matrice réelle  $A$  est orthogonalement semblable à une matrice quasi triangulaire supérieure. Autrement dit, il existe une matrice  $Q$  ( $Q^T Q = Q Q^T = I$ ) telle que  $R = Q^T A Q$  soit quasi triangulaire supérieure (ou triangulaire par blocs diagonaux). Les blocs diagonaux sont de taille 1 ou deux selon que les valeurs propres sont réelles ou complexes conjuguées.

**Rayon spectral d'une matrice**  $A$  est sa plus grande valeur propre en module  $\rho(A) = \text{Max}_{i \in \{1, \dots, n\}} (|\lambda_i|)$ .

# Calcul de valeurs propres: Généralité (suite)

**Méthode de la puissance et ses variantes** Supposons que  $A$  est réelle et que sa valeur propre dominante (en module) est simple. Cette méthode permet de déterminer cette valeur propre de  $A$ . En effet, soit  $v$  un vecteur choisi aléatoirement. On calcule la suite  $v_{k+1} = A^k v_0$  (pour  $k = 1, 2, \dots$ ). Si  $v_0$  n'est pas un vecteur orthogonal au sous-espace propre correspondant à la valeur propre dominante de  $A$ , alors  $v_k$  tend vers un vecteur de cet espace propre.

# Calcul de valeurs propres

## Quand la matrice est creuse et de grande taille

- ▶ Dans la plupart des applications industrielles, la matrice  $A$  est d'une structure creuse (a beaucoup d'éléments zéros) et sa taille  $n$  est très grande.
- ▶ Seul le calcul d'un petit nombre  $s$  de valeurs propres et leur vecteurs propres associés, est demandé

## Quelles méthodes?

- ▶ Méthodes directes ne sont pas adaptées dans ce cas (pourquoi?)
- ▶ Méthodes itératives classiques remplissent la matrice (*fill in*). La méthode  $Q.R$  par exemple **transforme**  $A$  en une matrice Hessenberge supérieure.

## Besoin de méthodes ne changeant pas la structure creuse de la matrice



# Calcul de valeurs propres

## Méthodes basées sur une suite de Krylov

### ► La puissance itérée ou des puissances

Soit  $v, Av, A^2v, A^3v, A^4v, \dots$  la suite de vecteurs de Krylov avec  $v$  un vecteur normé. Alors, l'espace  $\text{lin}(A^k v) \rightarrow \text{lin}(u_1)$  quand  $k \rightarrow +\infty$  où  $(u_1)$  est le vecteur propre dominant de  $A$ .

### ► Itérations simultanées

On démarre le processus de la méthode des puissances avec un espace engendré par une suite de vecteurs  $(v_1^0, \dots, v_r^0)$  au lieu d'un seul vecteur  $v$ . Soit  $S = \text{lin}(v_1^0, \dots, v_r^0)$  cet espace, on forme la suite  $S, AS, A^2S, A^3S, A^4S, \dots$ . Alors,  $\text{lin}(A^k S) \rightarrow \text{lin}(u_1, \dots, u_r)$  quand  $k \rightarrow +\infty$  où  $\text{lin}(u_1, \dots, u_r)$  est le sous-espace invariant dominant.

**Problème: méthodes sans mémoire** au fur et à mesure qu'elle calcule  $A^k v$ , elle oublie  $A^{k-1} v \dots$  Pour remédier à ce problème: voir plus loin.

# Calcul de valeurs propres

## Mémoriser les calculs

- ▶ On mémorise tous les vecteurs calculés  $v, Av, \dots, A^k v$  et on cherche des vecteurs propres approchés dans l'espace  $\text{lin}(v, Av, \dots, A^k v)$  engendré par ces vecteurs.
- ▶ La base de Krylov est mal-conditionnée. Mais, elle peut être améliorée en l'orthonormalisant:  $v_1, v_2, \dots, v_{k+1}$ .
- ▶ **Méthode d'Arnoldi**: L'application du procédé d'orthonormalisation de **Gram-Schmidt** sur la suite de Krylov. Orthogonalisation de  $A^k v$  contre  $v_1, v_2, \dots, v_k$  ou bien  $Av_k$  contre  $v_1, v_2, \dots, v_k$

## Algorithme RA : Réduction/projection d'Arnoldi sur un sous-espace de dimension $m$

**Input:**  $A, v, n, m, jj = 1$  (par défaut  $jj = 1$ )

**Result:**  $V_m = [v_1, v_2, \dots, v_m], H_{m \times m}, h_{m+1, m}, v_{m+1}$

```

1 initialization:  $v_1 = \frac{v}{\|v\|_2}$ 
2 for  $j = jj$  à  $m$  do
3    $v_{j+1} \leftarrow Av_j$ 
4   for  $i = 1$  à  $j$  do
5      $h_{i,j} \leftarrow (v_{j+1}, v_i)$ 
6      $v_{j+1} \leftarrow v_{j+1} - v_i \times h_{i,j}$ 
7   end
8    $h_{j+1,j} \leftarrow \|v_{j+1}\|_2$ 
9   if  $(h_{j+1,j} = 0)$  then
10     $\text{lin}(v_1, \dots, v_j)$  est le sous-espace invariant
11  end
12   $v_{j+1} \leftarrow \frac{v_{j+1}}{h_{j+1,j}}$ 
13 end
```

**Algorithm 1:** Réduction d'Arnoldi (RA) avec Gram-Schmidt modifié

# Remarque sur GS et MGS

## GS

Le procédé de Gram-Schmidt (GS) orthogonalise  $A^j v_1$  contre  $v_1, \dots, v_{j-1}$ .

## MGS

Le procédé de Gram-Schmidt modifié (GSM) orthogonalise  $A v_{j-1}$  contre  $v_1, \dots, v_{j-1}$  comme dans l'algorithme 1.

Le GS est mathématiquement équivalent à MGS. Cependant, du point de vue de calcul, MGS apporte plus de stabilité.

# Sous-espaces de Krylov

## Définition

Quelque soit  $j > 1$ , l'espace  $K_j(A, v) = \text{lin}(v, Av, \dots, A^{j-1}v)$  est appelé le  $j$ ème sous-espace de Krylov associé à  $A$  et  $v$ .

## Propriétés

Soit  $v, Av, \dots, A^{m-1}v$  une suite de vecteurs de Krylov linéairement indépendant et  $V_m = [v_1, v_2, \dots, v_m]$  les vecteurs issus de l'algorithme 1.

- ▶  $\text{lin}(v_1, \dots, v_j) = K_j(A, v)$  pour  $j = 1, \dots, m$
- ▶  $h_{j+1,j} > 0$  pour tout  $j = 1, \dots, m-1$ .
- ▶ Les vecteurs  $v, Av, \dots, A^m v$  sont linéairement dépendants si et seulement si  $h_{m+1,m} = 0$ . Dans ce cas, le sous-espace  $K_m(A, v)$  est invariant par  $A$ .

# Représentation matricielle de la projection d'Arnoldi

- Les lignes 3 – 6 de l'algorithme 1 impliquent  $v_{j+1} \leftarrow Av_j - (h_{1,j} \times v_1 + \dots + h_{j,j} \times v_j)$  et la ligne 12 permet d'écrire:  $v_{j+1} \leftarrow (Av_j - (h_{1,j} \times v_1 + \dots + h_{j,j} \times v_j)) / h_{j+1,j}$ . Par conséquent, on a:  $Av_j = \sum_{i=1}^{j+1} h_{i,j} \times v_i$
- L'égalité ci-dessus pour  $j = 1, m$  implique  $A[v_1, \dots, v_m] = [v_1, \dots, v_m, v_{m+1}]H_{m+1,m}$  ou bien  $AV_m = V_{m+1}H_{m+1,m}$  avec

$$H_{m+1,m} = \begin{pmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,m-1} & h_{1,m} \\ h_{2,1} & h_{2,2} & \cdots & h_{2,m-1} & h_{2,m} \\ 0 & h_{3,2} & \cdots & h_{3,m-1} & h_{3,m} \\ \vdots & & \ddots & & \vdots \\ 0 & & & h_{m,m-1} & h_{m,m} \\ 0 & 0 & & 0 & h_{m+1,m} \end{pmatrix}$$

une matrice de  $m + 1$  lignes et de  $m$  colonnes. Autrement dit:

# Représentation matricielle de la projection d'Arnoldi

$$AV_m = [V_m | v_{m+1}] \begin{bmatrix} H_{m,m} \\ h_{m+1,m} e_m^T \end{bmatrix}$$

ou bien

$$AV_m = V_m H_{m,m} + h_{m+1,m} v_{m+1} e_m^T \quad (2)$$

où la matrice  $H_{m,m}$  est sous forme d'Hessenberg supérieur et  $e_m^T$  est le même vecteur canonique.

- ▶ Si  $h_{m+1,m} = 0$  (*happy breakdown*), alors les vecteurs de Krylov  $v, Av, \dots, A^{m-1}v$  sont linéairement indépendants et l'espace  $\text{lin}(V_m)$  est un sous-espace invariant sous  $A$  :

$$AV_m = V_m H_{m,m}$$

$$A = V_m H_{m,m} V_m^*$$

Les valeurs propres de la matrice  $H_{m,m}$  sont donc les  $m$  vp de  $A$ .

# Représentation matricielle de la projection d'Arnoldi

*On peut aussi écrire :*

$$H_{m,m} = V_m^* A V_m$$

*Soit:*

$$H_{m,m} y_i = \tilde{\lambda}_i y_i \text{ pour } i = 1, m$$

*Avec  $\tilde{\lambda}_i$  la  $i$ ème valeur propre de  $H_{m,m}$  et  $y_i$  son vecteur propre associé. Les égalités ci-dessus conduisent à :*

$$H_{m,m} y_i = V_m^* A V_m y_i = \tilde{\lambda}_i y_i$$

$$A V_m y_i = \tilde{\lambda}_i V_m y_i \quad \text{ou} \quad A \tilde{u}_i = \tilde{\lambda}_i \tilde{u}_i$$

*avec  $\tilde{u}_i = V_m y_i$  le vecteur propre de  $A$  associé à la valeur propre  $\tilde{\lambda}_i$  (pour  $i = 1, m$ ).*



# Représentation matricielle de la projection d'Arnoldi

- Si  $h_{m+1,m} > 0$ , alors l'équation (2) peut s'écrire

$$AV_my_i = V_m H_{m,m} y_i + h_{m+1,m} v_{m+1} e_m^T y_i$$

En utilisant les égalités  $\tilde{u}_i = V_m y_i$ ,  $H_{m,m} y_i = \tilde{\lambda}_i y_i$  et  $y_i(m) = e_m^T y_i$ , on peut reformuler l'équation ci-dessus comme :

$$A\tilde{u}_i - \tilde{\lambda}_i \tilde{u}_i = h_{m+1,m} v_{m+1} y_i(m)$$

Ainsi, on peut utiliser la mesure  $\rho_i$  comme une estimation d'erreur sur le calcul de  $\tilde{\lambda}_i$  **valeur de Ritz** et  $\tilde{u}_i$  son **vecteur de Ritz** correspondant:

$$\rho_i = \|A\tilde{u}_i - \tilde{\lambda}_i \tilde{u}_i\|_2 = |h_{m+1,m}| \cdot |y_i(m)|$$

car  $\|v_{m+1}\|_2 = 1$ . Cette valeur est appelée **Ritz estimate** et sert à calculer d'erreur de la projection d'Arnoldi avec seulement une opération arithmétique élémentaire.

# Représentation matricielle de la projection d'Arnoldi

- ▶ Si  $m = n$ ?
- ▶ Généralement  $m \ll n$ . Pourquoi?
- ▶ Exécutez l'algorithme RA (algorithme 1) sur une matrice aléatoire d'ordre 50. Affichez les valeurs de Ritz (sous forme de points à deux coordonnées  $x$  et  $y$ ) calculées avec  $m = 10$  puis avec  $m = m + pas$  où  $pas$  représente l'augmentation de la taille de sous-espace de Krylov.
- ▶ Pour chacune des tailles de sous-espace  $m = m + pas$ , calculez le résidu  $\rho_i$  (pour  $i = 1, m$ ) par les deux formulations (la norme ainsi que le *Ritz estimate*).

# Comment calculer les valeurs propres aux extrémités et à l'intérieur du spectre?

- ▶ Soit  $\lambda$  une valeur propre de  $A$  et  $u$  son vecteur propre correspondant,  $\lambda^{-1}$  est une valeur propre de  $A^{-1}$  et  $u$  est son vecteur propres associé.
- ▶ Soit  $\gamma$  un scalaire. La matrice  $(A - \gamma I)^{-1}$  a la valeur propre  $(\lambda - \gamma)^{-1}$  et le vecteur propre correspondant  $u$ . Les matrices  $A$  et  $(A - \gamma I)^{-1}$  ont les mêmes sous-espaces invariants.

## Calcul des valeurs propres extrêmes et internes

- ▶ Les valeurs propres les plus grandes en modules (aux extrémités du spectre) peuvent être calculées par des méthodes des puissances, des itérations simultanées, d'Arnoldi et de Lanczos.
- ▶ Les valeurs propres à l'intérieur du spectre de  $A$  proche d'une certaine valeur cible  $\gamma$  peuvent être calculées par les mêmes méthodes appliquées à la matrice  $(A - \gamma I)^{-1}$ . En effet, les plus grandes valeurs propres de  $(A - \gamma I)^{-1}$  sont les valeurs propres les plus proches de la valeur  $\gamma$ .

# Calcul de quelques éléments propres avec Arnoldi

## Algorithme RitzArnoldi

**Input:**  $A, v, n, s, m, tol$

**Result:**  $\Lambda_s = (\tilde{\lambda}_1, \dots, \tilde{\lambda}_s), U_s = [\tilde{u}_1, \dots, \tilde{u}_s], \vec{\rho}$

- 1 Call RA (input:  $A, v, n, m, 1$ ; output:  $V_m, H_{m \times m}, h_{m+1,m}, v_{m+1}$ )
- 2 Calculer les éléments propres  $\Lambda_m = (\tilde{\lambda}_1, \dots, \tilde{\lambda}_m), Y_m = [y_1, \dots, y_m]$  de la matrice  $H_{m \times m}$
- 3 Ordonner les valeurs propres  $\tilde{\lambda}_i (i = 1, m)$ , puis sélectionner les  $s$  plus grandes en module :  $\Lambda_s = (\tilde{\lambda}_1, \dots, \tilde{\lambda}_s)$  et leur vecteur propres correspondants  $Y_s = [y_1, \dots, y_s]$
- 4 Calculer  $U_s = V_m Y_s = [\tilde{u}_1, \dots, \tilde{u}_s]$
- 5 Calculer le vecteur résidu  $\vec{\rho} = (\rho_1, \dots, \rho_s)^T$

**Algorithm 2:** Calcul de valeurs et vecteurs de Ritz avec Arnoldi

## Remarques sur l'algorithme d'Arnoldi

- ▶ En augmentant la taille de sous-espace dans l'algorithme 2, on pourra améliorer la qualité de sous espace de Krylov  $K_m(A, v)$ . Cela a été constaté par les expérimentation réalisées avec l'algorithme 1.
- ▶ La qualité de sous espace de Krylov  $K_m(A, v)$  dépend fortement du vecteur initial  $v$ . Cela est à la base de la proposition des algorithmes d'Arnoldi avec redémarrage (explicite ou implicite).

# Arnoldi avec le redémarrage

## Krylov Restard Arnoldi Method (KRAM)

**Input:**  $A, v, n, s, m, tol, nbcycle_{max}$

**Result:**  $\Lambda_s = (\tilde{\lambda}_1, \dots, \tilde{\lambda}_s), U_s = [\tilde{u}_1, \dots, \tilde{u}_s]$

```

1 for  $k = 1, 2$ , jusqu'à la convergence do
2   | Call RitzArnoldi(input: $A, v, n, s, m, tol$ ; output: $\Lambda_s, U_s, \vec{\rho}$ )
3   | Calculer  $\rho = \max(\rho_1, \dots, \rho_s)$  ou bien  $\rho = \rho_1 + \dots + \rho_s$ 
4   | if ( $\rho > tol$  et  $k < nbcycle_{max}$ ) then
5   |   | Choisir un nouveau vecteur initial  $v$  et aller à l'étape 2
6   | end
7 end
```

### Algorithm 3: Arnoldi avec le redémarrage

L'étape 2 de cet algorithme est équivalent à la réalisation de l'équation (2), la sélection des  $s$  paires de Ritz désirés et le calcul de leur vecteur résidu  $\vec{\rho} = (\rho_1, \dots, \rho_s)^T$ .

# Calcul de quelques paires de Ritz avec KRAM

## Stratégies de redémarrage

### ► Redémarrage explicite (ERAM)

L'étape 5 de l'algorithme KRA consiste à définir le nouveau vecteur initial  $v$  en fonction des vecteurs de Ritz calculés  $(\tilde{u}_1, \dots, \tilde{u}_s)$ . Par exemple  $x$  pourrait être une combinaison linéaire de ces vecteurs:  $v = \sum_{i=1}^s (\alpha_i \tilde{u}_i)$  avec  $\alpha_i$  des coefficients réels.

### ► Redémarrage implicite (IRAM)

Le choix du nouveau vecteur initial peut être fait de manière implicite. Cette stratégie implicite de redémarrage sera détaillée dans la suite.

# Calcul de toutes les valeurs propres d'une matrice dense de taille modérée

La **méthode**  $QR$  utilise la **décomposition**  $QR$  de cette matrice.

## La décomposition $QR$

- ▶ Toute matrice carée  $A$  peut être décomposée en  $QR$  avec  $Q$  une matrice unitaire ( $Q^*Q = QQ^* = I$ ), et  $R$  une matrice triangulaire supérieure.
- ▶ Si  $A$  est réel,  $Q$  est une matrice orthogonale ( $Q^* = Q^T$ ).
- ▶ Si  $A$  est inversible, la décomposition  $QR$  est unique.
- ▶ Si les  $m$  premiers vecteurs colonnes de  $A$  sont linéairement indépendants, les  $k$  ( $1 \leq k \leq m$ ) premières colonnes de  $Q$  forment une base orthogonale de l'espace engendré par les  $k$  premières colonnes de  $A$ .
- ▶ La décomposition  $QR$  est aussi applicable aux matrices rectangulaires

*[https : // en.wikipedia.org/wiki/QR\\_decomposition](https://en.wikipedia.org/wiki/QR_decomposition).*



# Calcul de toutes les valeurs propres d'une matrice dense de taille modérée

## La méthode $QR$

- ▶ Cette méthode consiste (i)– à appliquer la décomposition  $A = QR$ , (i)– à définir la nouvelle matrice  $A$  par  $A = RQ$  et, (iii)– à refaire les étapes (i) et (ii) sur cette nouvelle matrice.
- ▶ Soit  $A_1 = A$ , la méthode  $QR$  consiste à :

$$A_1 = Q_1 R_1 \iff (R_1 = Q_1^* A_1) \text{ et } A_2 = R_1 Q_1 = Q_1^* A_1 Q_1$$

$$A_2 = Q_2 R_2 \iff (R_2 = Q_2^* A_2) \text{ et } A_3 = R_2 Q_2 = Q_2^* A_2 Q_2$$

...

$$A_k = Q_k R_k \iff (R_k = Q_k^* A_k) \text{ et } A_{k+1} = R_k Q_k = Q_k^* A_k Q_k$$

- ▶ On constate que les matrices  $A_k$  ( $k > 1$ ) sont semblables à la matrice  $A$  et ont donc les mêmes valeurs propres. En effet,  $A_{k+1} = \hat{Q}_k^* A_1 \hat{Q}_k$  avec  $\hat{Q}_k = Q_1 Q_2 \cdots Q_k$ .

# Calcul de toutes les valeurs propres d'une matrice dense de taille modérée

## La méthode $QR$ (suite)

- ▶ Quand  $k$  tend vers infini, la matrice  $A_k$  tend vers une matrice quasi triangulaire supérieur dont les valeurs propres sont des valeurs situées sur sa diagonale ou bien des racines caractéristiques des blocs  $2 \times 2$  situés sur cette diagonale.
- ▶ Dans la pratique, l'algorithme  $QR$  ci-dessous (étapes (i) – (iii)) n'est jamais utilisée sous la forme présentée. Car il présente un coût trop élevé et une convergence trop lente.
- ▶ **Remède au problème de coût:** par des transformations semblables, la matrice  $A$  est d'abord mise sous la forme d'Hessenberg supérieur  $H$ . L'algorithme  $QR$  est alors appliqué à  $H$ .
- ▶ **Remède à la convergence lente:** appliquer l'algorithme  $QR$  à  $(A - \mu I)$  avec  $\mu = a_{n,n}$  (shift d'origine).

# Calcul de toutes les valeurs propres d'une matrice dense de taille modérée

## Algorithme *QR* avec shift d'origine

**Input:**  $A, n, tol$

**Result:**  $A$  transformé en matrice quasi-triangulaire supérieur

```

1 for  $k = 1, 2, \text{jusqu'à la convergence}$  do
2   Sélectionner le shift ( $\mu = a_{n,n}$  courant)
3   Décomposition :  $(A - \mu I) = QR$ 
4    $A \leftarrow RQ + \mu I$ 
5   if ( $a_{i,n} < tol$  pour tout  $i \in [1, n]$ ) then
6     Convergence de la  $n$ -ème valeur propre
7      $n \leftarrow n - 1$ 
8   end
9 end

```

Pour plus de détails voir: "Philippe G. Ciarlet, Jacques-Louis Lions  
*Introduction à l'analyse numérique matricielle et à l'optimisation - 5ème  
 édition, Dunod, 2007*".

## Arnoldi avec le redémarrage implicite (IRAM)

- ▶ L'algorithme  $QR$  calcule les plus petites valeurs propres en premier.
- ▶ Les éléments diagonaux de la matrice quasio-triangulaire supérieure issue de  $QR$  sont rangés dans l'ordre décroissant (haut en bas).
- ▶ Soient  $m$  la taille de sous-espace,  $s \leq m$  le nombre de valeurs propres désirées. Soit  $m = s + p$ , alors  $p$  est le nombre de valeurs propres non-désirées de  $H_{m,m}$ .
- ▶ Soient  $\lambda_1 \geq \dots \geq \lambda_s \geq \lambda_{s+1} \geq \dots \geq \lambda_{m=s+p}$  les valeurs propres de  $H_{m,m}$ . On nomme les valeurs propres non-désirées  $\mu_1 = \lambda_{s+1}, \dots, \mu_p = \lambda_{s+p=m}$ .
- ▶ En appliquant l'algorithme shifté  $QR$  à la matrice  $H_{m,m}$  avec des shifts  $\mu_1, \dots, \mu_p$ , elle sera transformée en une matrice dont toutes les informations concernant les valeurs propres désirées seront concentrées dans le bloc  $s \times s$  supérieur gauche de celle-ci.

## Arnoldi avec le redémarrage implicite (IRAM)

**Input:**  $A, v, n, s, m, tol, nbcycle_{max}$

**Result:**  $\Lambda_s = (\tilde{\lambda}_1, \dots, \tilde{\lambda}_s), U_s = [\tilde{u}_1, \dots, \tilde{u}_s]$

- 1 Call RA (input:  $A, v, n, m, 1$ ; output:  $V_m, H_{m \times m}, h_{m+1,m}, v_{m+1}$ )
- 2 **for**  $it = 1, 2, \dots$  jusqu'à la convergence **do**
- 3     Calculer les éléments propres de  $H_{m \times m}$
- 4     Sélectionner les  $s$  paires de Ritz dominants  $(\Lambda_s, U_s) = ((\tilde{\lambda}_1, \tilde{u}_1), \dots, (\tilde{\lambda}_s, \tilde{u}_s))$  et calculer le vecteur résidu  $\tilde{\rho}$
- 5     Si  $(\|\tilde{\rho}\|_\infty < tol)$  alors Stop
- 6     Choisir les shifts  $\mu_1 = \tilde{\lambda}_{s+1} \dots, \mu_p = \tilde{\lambda}_{s+p}$  et poser  $q^T = e_m^T$
- 7     **for**  $j = 1, \dots, p$  **do**
- 8          $(H_{m \times m} - \mu_j I) = Q_j R_j, H_{m \times m} \leftarrow Q_j^* H_{m \times m} Q_j$
- 9          $V_m \leftarrow V_m Q_j, q^T \leftarrow q^T Q_j$
- 10    **end**
- 11    Avec  $f_s$  la colonne  $s$  de  $f_m^+ \leftarrow h_{m+1,m} v_{m+1} q^T, V_k \leftarrow V_m(1:s)$  et  $H_{s \times s} \leftarrow H_{m \times m}(1:s, 1:s)$
- 12    Call RA (input:  $A, v, n, m, s$ ; output:  $V_m, H_{m \times m}, h_{m+1,m}, v_{m+1}$ )
- 13 **end**

# Méthodes de projections itératives

1. Des puissances (avec déflation)
2. Des itérations simultanées ou bien des puissances inverses?
3. Arnoldi avec redémarrage explicite (ERAM) et Multiple ERAM
4. Arnoldi avec redémarrage implicite (IRAM) et Multiple IRAM
5. Lanczos (matrice symétrique)
6. Padé-Rayleigh-Ritz (matrice symétrique)
7. Bi-Lanczos
8. Davidson (matrice symétrique)
9. Méthode GMRES
10. La décomposition en valeurs singulières (SVD)

# Rappels

- **Normes vectorielles** Pour le vecteur réel  $x \in R^n$ , les normes usuelles de  $x$  sont:

$$\|x\|_1 = |x_1| + \cdots + |x_n|$$

$$\|x\|_2 = (x_1^2 + \cdots + x_n^2)^{\frac{1}{2}}$$

$$\|x\|_\infty = \max(|x_1|, \cdots, |x_n|)$$

- **Multiplicité algébrique** La multiplicité algébrique ( $ma$ ) d'une valeur propre  $\lambda$  est le nombre de fois où elle apparaît dans le spectre de la matrice. Autrement dit, c'est la multiplicité de  $\lambda$  comme racine du polynôme caractéristique  $\det(A - \lambda I)$ .
- **Multiplicité géométrique** La multiplicité géométrique ( $mg$ ) est le nombre de ses vecteurs propres linéairement indépendants.

# Algorithme des puissances

**Input:**  $A, v, n$

**Result:**  $\lambda_1$

```

1 initialization:  $v_1 = \frac{v}{\|v\|_\infty}$ 
2 for  $k = 1, 2, \dots$  jusqu'à la convergence do
3    $v_k \leftarrow \frac{1}{\alpha_k} Av_{k-1}$ 
4    $\alpha_k \leftarrow \operatorname{argmax}_{i=1, \dots, n} |(Av_{k-1})_i|$ 
5
6 end
7  $\lambda_1 \leftarrow \alpha_k$ 

```

**Algorithm 4:** Un algorithme de la méthode des puissances

La valeur  $\alpha_k$  est la plus grande - en module - des composantes de  $Av_{k-1}$ . Dans la pratique, on normalise le vecteur  $v_k = Av_{k-1}$ .



# Méthode des puissances

## Convergence de la méthode

- ▶ Supposons:  $|\lambda_1| > |\lambda_2| \geq \dots \geq |\lambda_n|$  avec  $u_1$  le vecteur propre correspondant à la valeur propre dominante. Si le vecteur initial  $v_1$  n'est pas orthogonal au sous-espace invariant engendré par  $u_1$ , alors la suite  $\alpha_k$  converge vers  $u_1$ .

## Question:

- ▶ Pourquoi la méthode des puissances appliquée directement sur une matrice de chaîne de Markov avec des valeurs propres dominantes  $\lambda_1 = 1$  et  $\lambda_2 = -1$  ne fonctionne pas?

## Une solution

- ▶ Utiliser la version déflatée de la méthode: appliquer la méthode des puissances à la matrice  $A + \sigma I$  avec  $\sigma$  un nombre positif (appelé *shift* d'origine).

# Méthode des puissances

## La déflation

- ▶ Utiliser la version déflatée de la méthode: appliquer la méthode des puissances à la matrice  $A + \sigma I$  avec  $\sigma$  un nombre positif (appelé *shift* d'origine).
- ▶ Si les valeurs propres sont réelles et ordonnées comme:  
 $\lambda_1 > \lambda_2 \geq \lambda_3 \geq \dots \lambda_n$ , le meilleur shift d'origine est:  
 $\sigma_{best} = \frac{\lambda_2 + \lambda_n}{2}$ .
- ▶ Plus précisément, après avoir calculé le pair  $(\lambda_1, u_1)$ , la méthode de déflation permet de calculer la seconde valeur propre de plus grand module. En appliquant la même méthode à la matrice  $A^T$  qui a les mêmes valeurs propres que  $A$ , on peut calculer son vecteur propre dominant  $w_1$  associé à  $\lambda_1$ . La matrice  $B$  suivante a les valeurs propres  $\lambda_2, \dots, \lambda_n$  et zéro. Pour calculer  $\lambda_2, u_2$ , il suffit alors d'appliquer la méthode à la matrice:  $B = A - \frac{\lambda_1 u_1^T w_1}{(u_1, w_1)}$

# Méthode des itérations inverses

## Remarques

- ▶ L'idée de base est d'appliquer la méthode des puissances à  $A^{-1}$  (les valeurs propres de  $A^{-1}$  sont l'inverse de celles de  $A$ ).
- ▶ *Shift and Invert* Afin de calculer les valeurs propres les plus proches de  $\sigma$ , on peut appliquer la méthode des puissances sur la matrice  $(A - \sigma I)^{-1}$ .
- ▶ Convergence rapide en général mais nécessite la décomposition ( $LU$ ,  $QR$ , etc.) de la matrice  $(A - \sigma I)$ .

# Méthode des itérations simultanées

**Input:**  $A, n, s,$

**Result:**  $Q$

- 1 initialization: Choisir  $Q_0 = [q_1, \dots, q_s]$
- 2 **for**  $k = 1, 2, \dots$  *jusqu'à la convergence* **do**
- 3      $X \leftarrow A Q_{k-1}$
- 4      $X \leftarrow Q_k R$  (factorisation QR)
- 5 **end**
- 6  $Q = Q_k$

**Algorithm 5:** Itérations simultanées

Une seule normalisation pour l'ensemble des vecteurs.

# Méthode des itérations simultanées avec projection

**Input:**  $A, n, s$

**Result:**  $Q$

```

1 initialization: Choisir  $Q_0 = [q_1, \dots, q_s]$ 
2 for  $k = 1, 2, \dots$  jusqu'à la convergence do
3    $\tilde{Z} \leftarrow A Q_{k-1}$ 
4    $\tilde{Z} \leftarrow ZR$  (orthogonaliser  $\tilde{Z}$  pour obtenir  $Z$  / factorisation QR)
5   Calcul de  $B \leftarrow Z^H A Z$ 
6   Calcul de la forme de Schur de  $B$  et obtenir les vecteurs de
     Schur  $Y$ :  $B \leftarrow Y R Y^H$ 
7   Poser  $Q_k \leftarrow Z Y$ 
8 end
9  $Q = Q_k$ 
  
```

## Algorithm 6: Itérations simultanées

Une seule normalisation pour l'ensemble des vecteurs.

# Sélection d'articles

Un ensemble d'articles est sélectionné pour vous aider à chercher les algorithmes les plus adéquats pour la réalisation de votre projet.