CALCUL D'ELEMENTS PROPRES

BARRAULT MAXIME - SEMINAIRE CERMICS
13 NOVEMBRE 2002

Plan de l'exposé

→ Présentation du problème général

→ Problème Standard

→ Synthèse

→ Problème généralisé

→ Problèmes ouverts

→ Application au problème de la Chimie Quantique

Présentation du problème général

n la dimension du problème, A et B deux matrices de $\mathbb{C}^{\mathrm{n}\times\mathrm{n}}$

$$(\mathcal{P})\left\|\begin{array}{l}\text{Chercher les éléments propres } (u,\lambda)\in\mathbb{C}^{\mathrm{n}}\times\mathbb{C}\\ Au=\lambda Bu,\\ (u,Bu)=1.\end{array}\right.$$

$B = I_n \longmapsto$ Problème aux valeurs propres standard,

$B \neq I_n \longmapsto$ Problème aux valeurs propres généralisé.

Questions préliminaires

$\longrightarrow$ Cherche-t-on une partie ou la totalité des éléments propres de $(\mathcal{P})$?

$\longrightarrow$ Cherche-t-on les vecteurs propres ou les valeurs propres ?

$\longrightarrow$ Quelle est la nature des matrices A et B (hermitienne, normale, défective) ?

$\longrightarrow$ Quel est la taille du problème ?

$\longrightarrow$ Quel est le caractère creux des matrices A et B ?

$\longrightarrow$ Cherche-t-on des sous-espaces invariants ?

$\longrightarrow$ Quel est le type de machine utilisée (séquentielle, parallèle) ?

Décompositions fondamentales

$\longrightarrow$ Décomposition de Schur

$\exists\, Q$ unitaire et R triangulaire supérieure

$$Q^* A Q = R.$$

$\longrightarrow$ Décomposition de Schur généralisée

$\exists\, Q, Z$ unitaires et R, S triangulaires supérieures

$$\begin{aligned} Q^* A Z &= R, \\ Q^* B Z &= S. \end{aligned}$$

Autres résultats

$\longrightarrow$ Factorisation QR

$\exists\, Q$ unitaire et R triangulaire supérieure

$$A = QR.$$

$\longrightarrow$ Réduction en matrice de Hessenberg

$\exists\, Q$ unitaire et H matrice de Hessenberg

$$Q^* A Q = H.$$

$\Longrightarrow$ Outils : Transformation de Householder, Rotation de Givens

Transformation de Householder

$\longrightarrow$ Principe : Annulation des composantes d'un élément x de $\mathbb{R}^n$.

$\longrightarrow$ Forme :

$$P = I_n - 2\frac{uu^t}{u^t u}.$$

$\longrightarrow$ Fonctionnement : On veut annuler toutes les composantes de x exeptée la première. Ceci est équivalent à

$$\begin{aligned} Px = x - 2\frac{uu^t}{u^t u}x \quad // \quad e_1 \quad &\Longleftrightarrow \quad u = x + \alpha e_1, \\ &\Longleftrightarrow \quad \alpha = \pm\|x\|_2, \\ &\Longleftrightarrow \quad Px = \pm\|x\|_2 e_1. \end{aligned}$$

$\longrightarrow$ Utilisation :

$*$ Factorisation QR et Réduction de Hessenberg d'une matrice A par rotation progressive des colonnes de A.

Rotation de Givens

$\longrightarrow$ Effet sur un élément (x, y) de $\mathbb{R}^2$

$$c = \frac{x}{\sqrt{x^2+y^2}}, \ s = \frac{y}{\sqrt{x^2+y^2}},$$

$$P\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} c & s \\ -s & c \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \sqrt{x^2+y^2} \\ 0 \end{pmatrix},$$

P est appelée rotation de Givens.

$\longrightarrow$ Généralisation à un élément de $\mathbb{R}^n$

$*$ Factorisation QR d'une matrice A par élimination progressive des coefficients sous-diagonaux de A.

$*$ Réduction de A par élimination progressive des coefficients sous la sous-diagonale de A.

Quelques résultats de type perturbation (1)

$\longrightarrow$ Théorème 1 : Soit $A = D + F$ où $D = \mathrm{diag}(A) = \mathrm{diag}(d_1, \ldots, d_n)$,

$$\lambda(A) \subseteq \bigcup_{i=1}^{n} D_i$$

où

$$D_i = \left\{ z \in \mathbb{C}\,,\ |\mathrm{z} - \mathrm{d_i}| \leq \sum_{\mathrm{j=1}}^{\mathrm{n}} |\mathrm{f_{ij}}| \right\}.$$

$\longrightarrow$ Théorème 2 : Soit A une matrice symétrique et δA une perturbation symétrique, on a

$$\forall\ i = 1,\ n \quad |\tilde{\lambda}_i - \lambda_i| \leq \|\delta A\|_2.$$

Plus précisément, si $\eta_n \leq \ldots \leq \eta_1$ désignent les valeurs propres de δA,

$$\forall\ i = 1,\ n \quad \lambda_i + \eta_n \leq \tilde{\lambda}_i \leq \lambda_i + \eta_1.$$

Quelques résultats de type perturbation (2)

→ Théorème 3 : Les valeurs propres dépendent continuement des éléments de A. Si λ_i est de multiplicité 1, on récupère la différentiabilité de λ_i. Pour toute perturbation δA

$$\left| \begin{array}{lll} \delta\lambda_i & = & \dfrac{1}{y_i^* x_i} y_i^* \delta A x_i, \\ A x_i & = & \lambda_i x_i, \ \|x_i\|_2 = 1, \\ y_i^t A & = & \lambda_i y_i^t, \ \|y_i\|_2 = 1. \end{array} \right.$$

→ Théorème 4 : Pour toute perturbation δA

$$\tilde{\lambda_i} \subseteq \left\{ z \in \mathbb{C}\,,\ |\mathrm{z} - \lambda_\mathrm{i}| \leq \mathrm{n} \frac{\|\delta \mathrm{A}\|_2}{|\mathrm{y}_\mathrm{i}^* \mathrm{x}_\mathrm{i}|} \right\}$$

Quelques résultats de type perturbation (3)

→ Théorème 5 : Soient A symétrique et A_r la mineure supérieure de A de valeurs propres $(\lambda_j^r)_{j=1,r}$

$$\lambda_{r+1}^{r+1} \leq \lambda_r^r \leq \lambda_r^{r+1} \leq \ldots \leq \lambda_2^{r+1} \leq \lambda_1^r \leq \lambda_1^{r+1}.$$

→ Théorème 6 (Weinstein): Soient A symétrique, x un élément normé de $\mathbb{R}^n$ et θ un réel, il existe une valeur propre λ_i

$$\left| \begin{array}{l} r = Ax - \theta x, \\ |\lambda_i - \theta| \leq \|r\|_2. \end{array} \right.$$

Quelques résultats de type perturbation (4)

$\longrightarrow$ Théorème 7 (Kato-Temple): Soient A symétrique et x un élément normé de $\mathbb{R}^n$, il existe un élément propre (u_i, λ_i)

$$|\lambda_i - \beta| \leq \frac{\|r\|_2^2}{\mathrm{gap}(i, A)}, \quad \sin(x, u_i) \leq \frac{\|r\|_2}{\mathrm{gap}(i, A)},$$

où

$$\beta = x^t A x, \quad r = Ax - \beta x,$$
$$\mathrm{gap}(i, A) = \min_{j \neq i} |\lambda_i - \lambda_j|.$$

$\hookrightarrow$ Bornes sur l'estimation du vecteur propre seulement dans le cas où les valeurs propres sont bien séparées.

$\longrightarrow$ Clusters de valeurs propres : bornes utilisant le gap entre le cluster et le reste du spectre.

$\longrightarrow$ A non symétrique : bornes beaucoup plus difficiles à obtenir.

Hypothèses pour la suite de l'exposé

$\mathcal{H}$: Matrices A et B réelles diagonalisables et à spectre réel.

$\longmapsto$ Toutes les matrices intervenant dans les décompositions précédentes sont réelles.

Cas A symétrique et $B = I_n$

$\longrightarrow$ Décomposition de Schur

$$R \text{ est diagonale (notée } D).$$

$\longrightarrow$ Réduction de Hessenberg

$$H \text{ est tridiagonale (notée } T).$$

$\longrightarrow$ Quotient de Rayleigh : $\rho(A,x) = \frac{(x,Ax)}{(x,x)}$

$$\begin{aligned} \lambda^{max} &= \max_{0 \neq x \in \mathbb{R}^n} \rho(A,x), \\ \lambda^{min} &= \min_{0 \neq x \in \mathbb{R}^n} \rho(A,x). \end{aligned}$$

$\longrightarrow$ Théorème Min-Max de Courant-Fischer

Problème Standard : Méthodes directes

$\longrightarrow$ Méthode de Jacobi

$\longrightarrow$ Méthode QR

$\longrightarrow$ Méthode de Bissection

$\longrightarrow$ Méthode d'Homotopie

$\longrightarrow$ Méthode Divide & Conquer

$\longrightarrow$ Méthode "Holy Grail"

Méthodes directes : Jacobi (1846)

→ Matrice A symétrique

$*$ But : Former la décomposition de Schur de A en diminuant itérativement, à l'aide de rotations de Jacobi, la norme de la partie extra-diagonale de A.

$$\begin{bmatrix} J_{12} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} \\ A_{12}^t & A_{22} \end{bmatrix} \begin{bmatrix} J_{12}^t & 0 \\ 0 & I \end{bmatrix} = \begin{bmatrix} \begin{matrix} * & 0 \\ 0 & * \end{matrix} & J_{12}^t A_{12} \\ A_{12}^t J_{12} & A_{22} \end{bmatrix}.$$

$*$ Propriétés : convergence lente mais précise, peu parallélisable.

$*$ Remarque : Initiallement introduite pour rendre la matrice A à diagonale dominante.

→ Généralisation aux matrices non symétriques, non normales.

Méthodes directes : QR

$\longrightarrow$ But : Former la décomposition de Schur de A ($A = QRQ^t$) en utilisant des factorisations QR.

$*$ Algorithme :

$A_0 = A,$

Pour $i = 1, \ldots,$ jusqu'à convergence

$A_{i-1} = Q_i R_i,$

$A_i = R_i Q_i.$

Fin

$*$ Convergence :

- $A_i \longmapsto R,$
- $Q_1 \ldots Q_i \longmapsto Q.$
- Cubique si choix de shifts convenable.

Méthodes directes : Amélioration de QR

$\longrightarrow$ Réduction initiale de A sous une forme de Hessenberg H.

$\longrightarrow$ Déflation des valeurs propres convergées.

$\longrightarrow$ Utilisation de shifts

$*$ Shift explicite

- $A_{i-1} - \sigma I = Q_i R_i$,
- $A_i = R_i Q_i + \sigma I$.

$*$ Shift implicite

- théoème Q-implicite,
- $A = H \longmapsto \forall i \quad A_i$ est Hessenberg,
- Transformations de Householder.

Méthodes directes : Bissection

Soit T une matrice tridiagonale symétrique, on définit

$$P_j(z) = \det(zI_n - T_{1:j,1:j}).$$

$\rightarrow$ Propriétés

$$P_0(z) = 1,\ P_1(z) = z - \alpha_1,$$

$$\forall\ j = 1,\ n-1 \quad P_{j+1}(z) = (z - \alpha_{j+1})P_j(z) - \beta_j^2 P_{j-1}(z).$$

$\rightarrow$ Pour z fixé, le nombre d'accords de signe de deux termes consécutifs de la suite numérique $\Big(P_i(z)\Big)_{i=1,n}$ est égal au nombre de valeurs propres de T inférieures à z.

$\rightarrow$ Calcul des vecteurs propres : Itérations inverses

Méthodes directes : "Divide and Conquer"

$\longrightarrow$ Matrice tridiagonale symétrique

$$T = \begin{pmatrix} T_{11} & 0 \\ 0 & T_{22} \end{pmatrix} + \alpha v v^t.$$

Soit $T_{11} = Q_1 D_1 Q_1^t$, $T_{22} = Q_2 D_2 Q_2^t$, on pose $\tilde{v} = \mathrm{diag}(Q_1, Q_2)^t v$ et $D = \mathrm{diag}(D_1, D_2)$. D'où

$$T = \begin{pmatrix} Q_1 & 0 \\ 0 & Q_2 \end{pmatrix} (D + \alpha \tilde{v}\tilde{v}^t) \begin{pmatrix} Q_1^t & 0 \\ 0 & Q_2^t \end{pmatrix}.$$

Les valeurs propres de $D + \alpha \tilde{v}\tilde{v}^t$ sont solutions de l'équation séculaire

$$f(\lambda) = 1 + \alpha \sum_{j=1}^{n} \frac{\tilde{v}_j^2}{d_j - \lambda} = 0.$$

$\Longrightarrow$ Résolution efficace et calcul vecteurs propres : Gu et Eisenstat.

Méthodes directes : Homotopie

$\longrightarrow$ Principe : Trouver les chemins entre des valeurs propres d'une matrice connue D et les valeurs propres cherchées de la matrice A en calculant des valeurs propres de matrices intermédiaires définies par

$$\forall\, t \in [0,1] \quad M(t) = D + (A - D)t = D + Et.$$

$\longrightarrow$ Calcul de vecteurs propres :

- $*$ Itération inverse avec accélération de Rayleigh
- $*$ Solveur itératif stabilisé

$\longrightarrow$ Choix du pas de discrétisation :

- $*$ dt trop petit : algorithme très lent
- $*$ dt trop grand : chemins qui se croisent
  - $\hookrightarrow$ modèle linéaire pour λ en utilisant $\dot{\lambda} = x^t E x$.

$\longrightarrow$ Matrice tridiagonale : Combinaison avec "Divide and Conquer"

Problème Standard : Méthodes de sous-espaces

$\longrightarrow$ Introduction

$\longrightarrow$ Calcul d'un élément propre

$\longrightarrow$ Méthode de sous-espaces

$\longrightarrow$ Méthodes d'Arnoldi

$\longrightarrow$ Algorithmes préconditionnés

Méthodes de sous-espace : Introduction (1)

On cherche à calculer la plus grande valeur propre en module de A. Pour cela, on utilise l'algorithme des Puissances Itérées :

$v = v_1/\|v_1\|_2$,

Pour $i = 1, \ldots,$ jusqu'à convergence

$$v_{i+1} = Av,$$

$$\theta = v^t v_{i+1},$$

$$v = v_{i+1}/\|v_{i+1}\|.$$

Fin

A convergence, (v, θ) est l'élément propre associé à la valeur propre de plus grand module.

Méthodes de sous-espace : Introduction (2)

Dans le cas où on veut les k plus grandes valeurs propres en module de A, on utilise la méthode des Itérations Orthogonales :

U_k^1 orthonormale,

Pour $i = 1, \ldots,$ jusqu'à convergence

$V_k = AU_k^i$,

Orthonormalisation de V_k : $V_k = Q_k R_k$,

$U_k^{i+1} = Q_k$.

Fin

U_k^i converge vers une base orthonormale du sous-espace engendré par les éléments propres cherchés.

Lien avec l'algorithme QR

On suppose $k = n$ et $V_0 = I_n$. Soient $(Q_i^{QR}, R_i^{QR})_i$ les itérés de l'algorithme QR appliqué à A. Par récurrence, on montre les propriétés suivantes

$$\left| \begin{array}{lcl} U_i & = & Q_1^{QR} ... Q_i^{QR}, \\ AU_i & = & U_{i+1} R_{i+1}^{QR}, \\ R_i & = & R_i^{QR}. \end{array} \right.$$

En effet

$$\left| \begin{array}{lcl} AU^i & = & AU_{i-1} Q_i^{QR}, \\ & = & U_i R_i^{QR} Q_i^{QR}, \\ & = & U_i Q_{i+1}^{QR} R_{i+1}^{QR}, \\ & = & U_{i+1} R_{i+1}^{QR}. \end{array} \right.$$

Ainsi, $U_\infty = Q_1^{QR} ... Q_\infty^{QR}$ et $R_\infty = R_\infty^{QR}$.

Méthodes de sous-espace : Introduction (3)

$\longrightarrow$ But : Ramener le calcul de quelques éléments propres d'un système de très grande taille à un calcul d'éléments propres d'un système de taille abordable par les méthodes directes.

$\longrightarrow$ Idée : Générer à l'aide de A un sous-espace qui contient seulement l'information relative aux éléments propres désirés.

$\longrightarrow$ Principe : Méthode itérative

- $\ast$ Génération d'un sous-espace.
- $\ast$ Extraction d'informations.
- $\ast$ Mise à jour du sous-espace.

Méthodes de sous-espace : Un algorithme général

$\longrightarrow$ Un algorithme de base général

Soit $V_0 \in \mathbb{R}^{n\times p}$ où p est le nombre d'éléments propres cherchés et P_0 un polynôme.

Pour $i = 1, \ldots,$ jusqu'à convergence

$U_k = P_{k-1}(A)V_{k-1}$.

Orthonormaliser $U_k \longmapsto V_k$.

Calculer la matrice $H_k = V_k^t A V_k$.

Calculer les éléments propres $(\mu_k^i, z_k^i)_{i=1,p}$ de H_k.

Calculer P_k à l'aide des μ_k^i.

Fin

Récupérer les éléments propres désirés : $(\mu_\infty^i, u_\infty^i = V_\infty z_\infty^i)_{i=1,p}$.

Méthodes de sous-espace : Calcul d'une seule valeur propre

On suppose $|\lambda_n| < \ldots < |\lambda_1|$

$\longrightarrow$ Méthode de la Puissance

$*$ $\forall k \quad P_k(A) = A.$

$*$ Convergence : $|\lambda_1^k - \lambda_1| = \mathcal{O}\left(\left(\frac{|\lambda_2|}{|\lambda_1|}\right)^k\right).$

$\longrightarrow$ Méthode de la Puissance Inverse avec Shift

$$\forall k \quad P_k(A) = (A - \sigma^i I)^{-1}$$

$*$ Convergence : $\left|\dfrac{1}{\lambda_i^k - \sigma^i} - \dfrac{1}{\lambda_i - \sigma^i}\right| = \mathcal{O}\left(\left(\dfrac{|\lambda_i - \sigma^i|}{|\lambda_{i+1} - \sigma^i|}\right)^k\right).$

$*$ Accélération : $\forall k \quad P_k(A) = (A - \sigma_k^i I)^{-1}$ avec $\sigma_k^i = \sigma_{k-1}^i + \frac{1}{\mu_k^1}.$

$*$ Itération du Quotient de Rayleigh pour A symétrique

$$\forall k \quad P_k(A) = (A - \sigma_k^i I)^{-1} \text{ avec } \sigma_k^i = \frac{(u_k, Au_k)}{(u_k, u_k)}.$$

Méthodes de sous-espace : Sur le processus d'itération inverse

→ Paradoxe lorsque $\sigma_k^i \simeq \lambda_i$:

$*$ Le système à inverser est singulier, cependant la solution calculée reste en un sens correcte :

Même si l'estimation de la valeur propre est grossière, l'estimation du vecteur propre s'avère précise.

$*$ Obtention d'une bonne estimation du vecteur propre cherché en 1 seule itération.

→ Réponse : La solution calculée est fausse mais a la bonne direction.

→ Utilisation d'un solveur itératif délicate : stabilisation par perturbations de faible rang.

→ Problème : Valeurs propres peu séparées

$*$ Méthode de sous-espace avec itération de Rayleigh

$\Longrightarrow$ réorthogonalisation.

Méthodes de sous-espace : Remarques

$\longrightarrow$ Importance du procédé d'orthogonalisation pour la recherche de plusieurs valeurs propres.

$\longrightarrow$ Procédés d'orthogonalisation

- $*$ Algorithme de Gram-Schmidt, de Gram-Schmidt modifié.
- $*$ Algorithme de Gram-Schmidt itératif.
- $*$ Transformations orthogonales.

$\longrightarrow$ Déflation des valeurs propres convergées

Méthodes de sous-espace : Calcul du polynôme P_k

$\rightarrow$ On cherche le mode propre dominant λ_1. Soit $P_k \in \mathcal{P}^k$ (espace des polynômes de degré inférieur ou égal à k). $\mathcal{H}$ implique

$$\forall\ v_1 = \sum_{i=1}^{n} \alpha_i u_i \quad \longmapsto \quad P_k(A)v_1 = \sum_{i=1}^{n} \alpha_i P_k(\lambda_i) u_i .$$

On considère le problème continu

$$\min_{q \in \mathcal{P}^k,\ q(\lambda_1)=1} \ \max_{\lambda \in [\alpha,\beta]} |q(\lambda)|.$$

où $(\lambda_i)_{i=2,n} \in [\alpha, \beta], \lambda_1 > \beta$. La solution est

$$S_k(t) = \frac{C_k(1 + 2\frac{t-\beta}{\beta-\alpha})}{C_k(1 + 2\frac{\lambda_1-\beta}{\beta-\alpha})}.$$

avec C_k le polynôme de Chebyshev de degré k.

$\rightarrow$ $A \notin \mathcal{H}$: polynômes de Chebyshev complexes, polynômes de Faber.

Les méthodes de projection orthogonale : 2 propriétés

$\longrightarrow$ Propriété 1

Soient U le sous-espace dans lequel on projette la matrice A et $A_U = U^t AU$.

$$\left| \begin{array}{lcl} AU & \subset & U \\ A_U s & = & \lambda s \end{array} \right. \Longrightarrow \left| \begin{array}{lcl} z & = & Us \\ Az & = & \lambda z \end{array} \right. .$$

$\Longrightarrow$ Construction de sous-espaces stables par A.

$\longrightarrow$ Propriété 2

Soit un vecteur $v_1 = \sum_{i=1}^n \alpha_i u_i$ tel que $\forall i = 1,\ n \quad \alpha_i \neq 0$. Pour tout $j = 1,\ n$, il existe un polynôme Λ_j de degré $n-1$ tel que

$$u_j = \Lambda_j(A) v_1.$$

$\longrightarrow$ Méthode des sous-espaces : on approche $(u_j)_{j\in\mathcal{I}}$ par $\displaystyle\sum_{i=1}^p \alpha_i A^l v_i$.

Introduction des espaces de Krylov

$\longrightarrow$ Espace de Krylov relatif à A de dimension m associé à v_1

$$\mathcal{K}^m(A, v_1) = \mathrm{Vect}(v_1, Av_1, ..., A^{m-1}v_1).$$

$\longrightarrow$ Propriétés

$$\left|\begin{array}{lcll} \mathcal{K}^m(A, v_1) & = & \Big\{p(A)v_1,\ p \in \mathcal{P}^{m-1}\Big\}, & \\ A\mathcal{K}^m(A, v_1) & = & \mathcal{K}^m(A, v_1) & \text{pour } m \geq \mu(A, v_1), \\ \mathcal{K}^m(A, v_1) & = & \mathcal{K}^\mu(A, v_1) & \text{pour } m \geq \mu(A, v_1), \\ \dim\Big(\mathcal{K}^m(A, v_1)\Big) & = & m & \text{ssi} \quad m \leq \mu(A, v_1) \end{array}\right.$$

où $\mu(A, v_1)$ désigne le degré du polynôme minimal de v_1 par rapport à A

$$\mu(A, v_1) = \min\Big\{\text{degré}(p),\ p(A)v_1 = 0,\ p \in \mathcal{P}^n \backslash \{0\}\Big\}.$$

Optimalité de la méthode de Krylov dans le cas symétrique

Soit λ_1 la valeur propre de plus grand module de A, on a

$$\lambda_1 = \max_{0 \neq x \in \mathbb{R}^n} \rho(A, x),$$

soit, avec v_1 tel que $\mu(A, v_1) = n$,

$$\lambda_1 = \max_{0 \neq p \in \mathcal{P}^{n-1}} \rho\Big(A, p(A)v_1\Big).$$

La méthode de projection orthogonale associée à l'espace de Krylov $\mathcal{K}^m(A, v_1)$ approche λ_1 par

$$\max_{0 \neq p \in \mathcal{P}^{m-1}} \rho\Big(A, p(A)v_1\Big).$$

Méthode d'Arnoldi simple

$\longrightarrow$ Principe : méthode de projection orthogonale sur un sous-espace de Krylov $\mathcal{K}^m(A, v_1)$ pour des matrices quelconques (non symétriques).

$\longrightarrow$ Algorithme simple

Choisir un vecteur initial v_1 normé

Pour $k = 1, \ldots,$ jusqu'à convergence (au plus n itérations)

Construction d'une base orthonormale V_k de $\mathcal{K}^k(A, v_1)$.

Construction de $H_k = V_k^t A V_k$.

Calculer des éléments propres $(\mu_i^k, z_i^k)_{i=1,k}$ de H_k.

Fin

Récupérer les éléments propres désirés : $(\mu_j^\infty, u_j^\infty = V_\infty z_j^\infty)_{j \in \mathcal{I}}$.

Méthode d'Arnoldi : construction d'une base orthonormale de $\mathcal{K}^m(A, v_1)$

On suppose v_1 normé. On utilise ici l'algorithme d'orthonormalisation de Gram-Schmidt modifié.

Pour $j = 1, m-1$

$w = Av_j.$

Pour $i = 1, j$

$h_{ij} = (w, v_i).$

$w = w - h_{ij} v_i.$

Fin

$h_{j+1,j} = \|w\|.$

$$v_{j+1} = \frac{w}{h_{j+1,j}}.$$

Fin

$\longrightarrow$ possibilité d'utiliser un procédé d'orthonormalisation plus précis.

Quelques propriétés de l'algorithme d'Arnoldi simple

→ Propriété 1

$$\mu(A, v_1) = j \Longleftrightarrow h_{j+1,j} = 0 \Longrightarrow A\mathcal{K}^j(A, v_1) = \mathcal{K}^j(A, v_1).$$

→ Propriété 2

$$\forall\ m = 1,\ \mu(A, v_1)$$

$$\left| \begin{array}{lcl} AV_m & = & V_{m+1}H_{m+1,m} \quad = \quad V_mH_m + h_{m+1,m}v_{m+1}e_m^t, \\ V_m^tAV_m & = & H_m, \end{array} \right.$$

où e_m désigne le m$^{\text{ième}}$ vecteur de la base canonique de $\mathbb{R}^m$.

Sur les vecteurs de Ritz associés à A

$\longrightarrow$ Vecteurs de Ritz associés à A : $(u_i = V_m y_i, \mu_i)_{i=1,m}$

$$\left| \begin{array}{lcl} H_m y_i & = & \mu_i y_i, \\ \|y_i\| & = & 1. \end{array} \right.$$

$\longrightarrow$ Propriétés 1

$$\left| \begin{array}{lcl} V_m^t (A u_i - \mu_i u_i) & = & 0, \\ \|(A - \mu_i I_n) u_i\|_2 & = & h_{m+1,m} |e_m^t y_i|. \end{array} \right.$$

$\longrightarrow$ Propriété 2

$$\mu(A, v_1) = n \quad \Longrightarrow \quad \forall\, i = 1,\ n \quad \forall\, m = 1,\ n \quad \exists\, \epsilon_i > 0$$

$$\left| \begin{array}{lcl} \|(I - \Pi_m) u_i\|_2 & \leq & \epsilon_i \delta_i^m, \\ \delta_i^m & = & \displaystyle \min_{p \in \mathcal{P}^{m-1}} \max_{\lambda \in \sigma(A) \setminus \{\lambda_i\}} |p(\lambda)|. \end{array} \right.$$

Sur les vecteurs de Ritz associés à A (2)

$\longrightarrow$ Théorème

$\forall\, i = 1,\ n, \forall\, m = 1,\ n-1,$

il existe m valeurs propres de A numérotées $\lambda_i^1, \ldots, \lambda_i^m$ telles que

$$\delta_i^m = \left(\sum_{j=1}^{m} \prod_{k=1, k \neq j}^{m} \frac{|\lambda_i^k - \lambda_i|}{|\lambda_i^k - \lambda_i^j|} \right)^{-1}$$

$\hookrightarrow$ Les valeurs propres qui se situent à la périphérie du spectre de A sont celles qui sont susceptibles d'être approchées le plus rapidement.

Sur les vecteurs harmoniques de Ritz associés à A

$\longrightarrow$ Principe : Obtention d'informations sur le spectre de A^{-1} avec $\mathcal{K}^m(A, v_1)$

$$h_{m+1,m} \neq 0 \Longrightarrow A^{-1} : A\mathcal{K}^m(A, v_1) \longmapsto \mathcal{K}^m(A, v_1).$$

$\hookrightarrow$ Recherche d'une base orthonormale de $A\mathcal{K}^m(A, v_1)$

$$\left| \begin{array}{lcl} H_{m+1,m}^t H_{m+1,m} & = & U_m^t U_m, \\ U_m^{-t} V_m^t A^t A V_m U_m^{-1} & = & I_m, \\ U_m^{-t} (AV_m)^t A^{-1} A V_m U_m^{-1} & = & U_m^{-t} H_m^t U_m^{-1}. \end{array} \right.$$

$\longrightarrow$ Vecteurs harmoniques de Ritz

$$\left(u_i^m = A V_m U_m^{-1} y_i^m, \frac{1}{\mu_i^m} \right) \text{ tel que } U_m^{-t} H_m^t U_m^{-1} y_i^m = \mu_i^m y_i^m.$$

$\longrightarrow$ Shift : $M_{m,\tau} = (H_{m+1,m} - \tau I_{m+1,m})^t (H_{m+1,m} - \tau I_{m+1,m})$.

Refined eigenvector approximations

Soit μ une valeur (harmonique) de Ritz. On améliore le vecteur associé en résolvant

$$\min_{z \in \mathcal{K}^m(A, v_1)} \|Az - \mu z\|_2.$$

Ceci revient à résoudre

$$\min_{\|y\|_2 = 1} \left\|(AV_m - \mu V_m) y\right\|_2,$$

ou

$$\min_{\|y\|_2 = 1} \left\|(H_{m+1,m} - \mu I_{m+1,m}) y\right\|_2.$$

$\hookrightarrow$ la solution y_s est le vecteur singulier à gauche correspondant à la plus petite valeur singulière de $H_{m+1,m} - \mu I_{m+1,m}$.

Sur les différents vecteurs de Ritz

→ Synthèse

∗ Vecteurs de Ritz associés à A : on approche les éléments propres externes de A.

∗ Vecteurs de Ritz associés à $(A - \sigma I_n)^{-1}$: on approche les éléments propres dont les valeurs propres associées sont proches de σ.

∗ Vecteurs harmoniques de Ritz associés à A : on approche les éléments propres dont les valeurs propres associées sont proches de 0.

∗ Vecteurs harmoniques de Ritz associés à A avec shift τ : on approche les éléments propres de A dont les valeurs propres associées sont proches de τ.

→ Dans la pratique

∗ convergence lente des valeurs harmoniques de Ritz.

∗ Les vecteurs harmoniques de Ritz peuvent seulement apporter un intérêt lors du redémarrage de l'algorithme.

Méthode d'Arnoldi pour A symétrique : Algorithme de Lanczos

$\longrightarrow$ Construction de $\mathcal{K}^m(A, v_1)$

v_1 normé, $v_0 = 0$, $\beta_0 = 0$.

Pour $j = 1, m-1$

$w = Av_j - \beta_{j-1}v_{j-1}$.

$\alpha_j = (v_j, w)$.

$w = w - \alpha_j v_j$.

$\beta_j = \|w\|$.

$v_{j+1} = \dfrac{w}{\beta_j}$.

Fin

$\longrightarrow$ Ecriture matricielle

$$AV_{m-1} = V_m T_{m,m-1},$$

$$\forall i = 1,\ m-1 \quad Av_i = \beta_{i-1}v_{i-1} + \alpha_i v_i + \beta_i v_{i+1}.$$

Sur les vecteurs de Ritz (1)

→ Théorème Min-Max

$$\forall\, k = 1,\ m \quad \mu_k^m = \max_{\dim(S)=k}\ \min_{0\neq x\in S\subset\mathcal{K}^m(A,v_1)} \frac{x^t A x}{x^t x}.$$

↪ Bornes a priori avec le théorème de Courant-Fischer.

→ Caractérisation polynômiale

$\exists\ (P_k)_{k=0,m}$ telle que $\forall\, k = 1,\ m$

$$\left|\begin{array}{l} v_k = P_{k-1}(A)v_1, \\ \forall\, j = 0,\ k-1 \quad \Big(P_k(A)v_1, P_j(A)v_1\Big) = 0, \\ P_0 = 1 \text{ et } \forall\, k \geq 1 \quad P_k \text{ est proportionnel à } \chi(T_k), \\ \forall\, j = 1,\ k \quad P_k(\mu_j^k) = 0. \end{array}\right.$$

↪ Propriété d'entrelacement des valeurs de Ritz.

Sur les vecteurs de Ritz (2)

Soit $v_1 = \displaystyle\sum_{j=1}^{n} \gamma_j u_j$, on suppose $\mu(A, v_1) = n$ et

$$\forall\, m = 1,\ n \quad \mu_1^m > \ldots > \mu_m^m.$$

$$\forall\, m = 1,\ n \quad \forall\, j = 1,\ m$$

$$\left|\begin{array}{lcl} R_{m,j}(x) & = & \dfrac{P_m(x)}{(x - \mu_j^m)}, \\ u_j^m & = & R_{m,j}(A) v_1, \\ \mu_j^m & = & \dfrac{\sum_{l=1}^{n} \lambda_l R_{m,j}^2(\lambda_l) \gamma_j^2}{\sum_{l=1}^{n} R_{m,j}^2(\lambda_l) \gamma_j^2} \quad = \quad \dfrac{(u_j^m, A u_j^m)}{(u_j^m, u_j^m)}. \end{array}\right.$$

$\Longrightarrow$ Caractérisation du vecteur de Ritz associé à μ_j^m.

Résultats de convergence des éléments propres

Théorème 5 : $\forall\, m = 1,\ n \quad \forall\, j = 1,\ m$

$$\left|\begin{array}{l} \lambda_j \geq \mu_j^m \geq \lambda_j - (\lambda_j - \lambda_n)\left(\dfrac{\kappa_j^m \tan\Big(\theta(v_1, u_j)\Big)}{C_{m-j}(1+2\delta_j)}\right)^2, \\ \tan\Bigg(\theta\Big(u_j, \mathcal{K}^m(A, v_1)\Big)\Bigg) \leq \dfrac{\epsilon_j}{C_{m-j}(1+2\delta_j)} \tan\Big(\theta(v_1, u_j)\Big), \end{array}\right.$$

avec

$$\delta_j = \frac{\lambda_j - \lambda_{j+1}}{\lambda_{j+1} - \lambda_n},$$

$$\kappa_1^m = 1,\ \kappa_j^m = \prod_{l=1}^{j-1} \frac{(\mu_l^m - \lambda_n)}{(\mu_l^m - \lambda_j)} \quad j > 1,$$

$$\epsilon_1 = 1,\ \epsilon_j = \prod_{l=1}^{j-1} \frac{(\lambda_l - \lambda_n)}{(\lambda_l - \lambda_j)} \quad j > 1,$$

C_{m-j} est le polynôme de Chebyshev de degré $m - j$.

Cas particuliers

On suppose de plus $\lambda_1 \geq \ldots \geq \lambda_n$, pour tout $m = 1,\ n$

$$\left| \begin{array}{lllll} \lambda_1 & \geq & \mu_1^m & \geq & \lambda_1 - \dfrac{(\lambda_1 - \lambda_n)\tan(\theta_1)^2}{C_{m-1}(1+2\delta_1)^2}, \\ \\ \lambda_n & \leq & \mu_m^m & \leq & \lambda_n + \dfrac{(\lambda_1 - \lambda_n)\tan(\theta_n)^2}{C_{m-1}(1+2\delta_n)^2}, \end{array} \right.$$

avec

$$\begin{aligned} &\cos(\theta_1) = |(v_1^t, u_1)|, \\ &\gamma_1 = \frac{\lambda_1 - \lambda_2}{\lambda_2 - \lambda_n}, \\ &\cos(\theta_n) = |(v_1^t, u_n)|, \\ &\gamma_n = \frac{\lambda_{n-1} - \lambda_n}{\lambda_1 - \lambda_{n-1}}. \end{aligned}$$

Comparaison avec l'algorithme de puissance

$\longrightarrow$ Algorithme de Lanczos

$$C_m(1+2\delta_1)^2 > \frac{1}{2}\exp\left(m\sqrt{\frac{\lambda_1-\lambda_2}{\lambda_2-\lambda_n}}\right).$$

$\longmapsto$ Convergence exponentielle

$\longrightarrow$ Algorithme de puissance

$$\begin{aligned} \frac{w_m^t A w_m}{w_m^t w_m} &= \lambda_1 \frac{\displaystyle\sum_{i=1}^{n} \gamma_i^2 P_{2m-1}(\lambda_i)}{\displaystyle\sum_{i=1}^{n} \gamma_i^2 P_{2m-2}(\lambda_i)}, \\ &= \lambda_1 + c(\lambda_1-\lambda_2)\left(\frac{\lambda_2}{\lambda_1}\right)^{2m-1}\Big(1+o(1)\Big). \end{aligned}$$

$\longmapsto$ Convergence polynômiale

Sur les vecteurs harmoniques de Ritz

$\longrightarrow$ Théorème Min-Max

$$\forall\ m = 1,\ \mu(A, v_1) \quad \forall\ k = 1,\ m \quad \frac{1}{\mu_k^m} = \max_{\dim(S)=k}\ \min_{0 \neq x \in S \subset A\mathcal{K}^m(A,v_1)} \frac{x^t A^{-1} x}{x^t x}$$

$\longrightarrow$ Intervalles de Lehmann

Soit σ tel que $\lambda_{-r} < \ldots < \lambda_{-1} < \sigma < \lambda_1 \ldots < \lambda_{n-r}$. Pour tout $m = 1,\ n,\ \nu_{-t_m}^m < \ldots < \nu_{-1}^m < \sigma < \nu_1^m \ldots < \nu_{m-t_m}^m$ désignent les valeurs harmoniques de Ritz associées à $A - \sigma I_n$ à l'itération m.

- Chaque intervalle $[\sigma, \sigma + \nu_{-j}^m]_{j=1,t_m}$ contient au moins j valeurs propres de A.
- Chaque intervalle $[\sigma, \sigma + \nu_j^m]_{j=1,m-t_m}$ contient au moins j valeurs propres de A.
- En l'absence d'autres informations sur le spectre, ces intervalles sont optimaux.

Lanczos en pratique

→ Problèmes

∗ Perte d'orthogonalité des vecteurs v_j lorsque m est grand.

∗ Perte d'orthogonalité dès qu'on a convergence d'un mode propre.

∗ Coût calcul et mémoire prohibitifs pour m grand.

∗ On n'utilise pas l'information pour accélérer la convergence vers les modes propres cherchées.

→ Améliorations

∗ Réorthogonalisation totale ou partielle des vecteurs v_j.

∗ Déflation des modes propres convergés.

∗ Redémarrages périodiques en utilisant au mieux l'information pour générer un nouveau vecteur v_1.

Une alternative a Arnoldi pour les matrices non symétriques : Bi-Lanczos

$\longrightarrow$ Rappel :

$$Av_i = \sum_{j=1}^{i} h_{j,i} v_j + h_{i+1,i} v_{i+1}.$$

$\longrightarrow$ Problème : Comment construire une famille non orthogonale V_i de façon à avoir H_i tridiagonale ?

$\longrightarrow$ Remarque : Soit une famille W_i telle que $W_i^t V_i = D_i$ et $W_i^t v_{i+1} = 0$, on a

$$W_i^t A V_i = D_j H_i.$$

H_i tridiagonale $\Longrightarrow V_i^t A^t W_i$ tridiagonale

$\hookrightarrow$ W_i base de $\mathcal{K}^i(A^t, w_1)$ orthogonale à V_i.

Construction des bases V_i et W_i

Soit v_1 et w_1 normés tels que $\delta_1 = w_1^t v_1 \neq 0$.

$\beta_0 = 0, v_0 = w_0 = 0$.

Pour $i = 1, \ldots$

$$p = Av_i - \beta_{i-1} v_{i-1}.$$

$$\alpha_i = w_i^t p / \delta_i.$$

$$p = p - \alpha_i v_i.$$

$$\gamma_{i+1} = \|p\|_2.$$

$$v_{i+1} = p / \gamma_{i+1}.$$

$$w_{i+1} = (A^t w_i - \beta_{i-1} w_{i-1} - \alpha_i w_i)/\gamma_{i+1}.$$

$$\delta_{i+1} = w_{i+1}^t v_{i+1}.$$

$$\beta_i = \gamma_{i+1} \delta_{i+1} / \delta_i.$$

Fin

Quelques propriétés de l'algorithme de Bi-Lanczos

→ Ecriture matricielle

$$AV_j = V_{j+1}T_{j+1,j}.$$

→ Projection oblique : condition de Petrov-Galerkin

$$T_jy = \mu y \Longleftrightarrow W_j^t(AV_jy - \mu V_jy) = 0.$$

- (V_jy, μ) éléments de Petrov à droite,
- (W_jy, μ) éléments de Petrov à gauche.

On peut définir de la même façon des éléments harmoniques de Petrov à droite et à gauche.

Synthèse sur l'algorithme de Bi-Lanczos

$\longrightarrow$ Comportement similaire à l'algorithme de Lanczos.

$\longrightarrow$ Stratégies d'amélioration équivalentes.

$\longrightarrow$ Instabilité plus grande qu'Arnoldi en pratique.

$\longrightarrow$ Nécessité d'introduire des stratégies de type "Look-ahead" difficile d'implémentation quand $w_i^t v_i \simeq 0$.

$\longrightarrow$ Comportement encore mal compris (comme Arnoldi par ailleurs).

Problèmes de l'algorithme d'Arnoldi simple

$\longrightarrow$ Perte d'orthogonalité, coût calcul et mémoire prohibitif pour m grand.

$\hookrightarrow$ Redémarrage

$\longrightarrow$ On n'utilise pas l'information pour accélérer la convergence vers les modes propres cherchées.

$\hookrightarrow$ Redémarrage particulier

$\longrightarrow$ On garde les éléments propres convergés.

$\hookrightarrow$ Déflation de ces éléments

$\longrightarrow$ Difficulté à cerner les valeurs propres multiples (comme l'algorithme de Lanczos) et les clusters.

$\hookrightarrow$ Algorithme par bloc

Méthode d'Arnoldi avec redémarrage simple

Choisir un vecteur initial v_1 normé et m la dimension de l'espace $\mathcal{K}^m(A, v_1)$ de travail,

Pour $k = 1, \ldots,$ jusqu'à convergence

Construction d'une base orthonormale V_k de $\mathcal{K}^m(A, v_k)$.

Construction de $H_k = V_k^t A V_k$.

Calcul des éléments propres $(\mu_i^k, z_i^k)_{i=1,m}$ de H_k.

Calcul de v_{k+1} à l'aide des éléments propres de H_k.

Fin

Récupérer les éléments propres de A désirés : $(\mu_j^\infty, u_j^\infty = V_\infty z_j^\infty)_{j \in \mathcal{I}}$.

$\longrightarrow$ Remarque : Le redémarrage ralentit la convergence

$\hookrightarrow$ Compromis à réaliser à travers le choix de m.

Redémarrage : Calcul de v_{k+1}

$\longrightarrow$ Plusieurs redémarrages

- $*$ Redémarrage simple : $v_{k+1} = v_k^m$.
- $*$ Redémarrage implicite par Chebyshev : $v_{k+1} = P_k(A)v_k$.
- $*$ Redémarrage du à Saad :

$$v_{k+1} = \sum_{i=1}^{p} \alpha_i u_i$$

où u_i sont les meilleures estimations obtenues des p modes cherchés à l'itération k.

$\longmapsto$ Problème : Comment choisir les α_i ?

$\longrightarrow$ Problème lié au redémarrage : Il faut reconstruire à chaque itération $\mathcal{K}^m(A, v_{k+1})$.

Un redémarrage optimal : Implicit Restarted Arnoldi (Sorensen)

$\longrightarrow$ Proposition d'un algorithme

Choisir un vecteur v_1 normé, p le nombre de modes propres cherchés et m tel que $p+m$ soit la dimension maximale de travail autorisée. Soit $H^1_{p+1,p}$ la matrice obtenue après p itérations d'Arnoldi.

Pour $k = 1, \ldots,$ jusqu'à convergence

Obtention de H^k_{p+m} suite à m itérations d'Arnoldi supplémentaires

Calcul des estimations d'éléments propres de A avec H^k_{p+m}

Calcul de v_{k+1} à l'aide des éléments calculés.

Obtention de $H^{k+1}_{p+1,p}$ en effectuant p itérations d'Arnoldi à partir de v_{k+1}.

Fin

IRA : Principe

$\longrightarrow$ Principe : Résoudre le problème de redémarrage en évitant de reconstruire $\mathcal{K}^p(A, v_k)$ à chaque itération.

$\longrightarrow$ Calcul de v_{k+1}

$$v_{k+1} = \gamma \prod_{i=1}^{m} (A - \mu_i^k I_n) v_k$$

où γ est une constante de normalisation et μ_i^k les m moins bonnes estimations des valeurs propres cherchées.

$\longrightarrow$ calcul de $H_{p+1,p}^{k+1}$:

La forme de v_{k+1} rend possible le calcul de $H_{p+1,p}^{k+1}$ de façon implicite à l'aide de factorisation QR implicite sans effectuer p itérations d'Arnoldi à partir de v_{k+1}.

IRA : Comment cela marche (1)

$\longrightarrow$ Après Arnoldi : $AV_{p+m} = V_{p+m}H_{p+m} + h_{p+m+1,p+m}v_{p+m+1}e_{p+m}^t$.

$\longrightarrow$ Application d'une factorisation QR avec shift explicite μ

$$H_{p+m} - \mu I_n = QR \text{ avec } Q \text{ Hessenberg}$$

$$\Longrightarrow \left| \begin{array}{rcl} (A - \mu I_n)V_{p+m} - V_{p+m}QR & = & h_{p+m+1,p+m}v_{p+m+1}e_{p+m}^t, \\ AV^+ - V^+H^+ & = & h_{p+m+1,p+m}v_{p+m+1}e_{p+m}^t Q, \end{array} \right.$$

avec $V^+ = V_{p+m}Q$ et $H^+ = RQ + \mu I_n = Q^t H_{p+m} Q$.

$$\hookrightarrow (A - \mu I_n)v_1 = \rho_{11} v_1^+ \text{ avec } v_1^+ = V^+ e_1 \text{ et } \rho_{11} = e_1^t R e_1,$$

et, comme $(e_{p+m}^t Q)_l = 0$ pour $l < p+m-1$, $(v_i^+)_{i=1,p+m-1}$ sont les itérés qu'on obtiendrait après $p+m-1$ itérations d'Arnoldi avec v_1^+ pour vecteur initial.

IRA : Comment cela marche (2)

$\rightarrow$ Application de m shifts μ_j

$$AV_{p+m}^{+} = V_{p+m}^{+} H_{p+m}^{+} + h_{p+m+1,p+m} v_{p+m+1} e_{p+m}^{t} Q$$

avec $V_{p+m}^{+} = V_{p+m} Q$, $H_{p+m}^{+} = Q^{t} H_{p+m} Q$ et $Q = Q_1 \dots Q_m$.

Comme Q_i Hessenberg pour tout $i = 1,\ m$, on déduit

$$h_{p+m+1,p+m} e_{p+m}^{t} Q = (\beta e_p^{t}, b^{t}), \quad b \in \mathbb{R}^{m}.$$

soit

$$\Longrightarrow \left| \begin{array}{lcl} AV_p^{+} & = & V_p^{+} H_p^{+} + h_{p+1,p}^{+} v_{p+1}^{+} e_p^{t} + \beta v_{p+m+1} e_p^{t}, \\ AV_p^{+} & = & V_p^{+} H_p^{+} + \tilde{h}_{p+1,p}^{+} s^{+} e_p^{t} \quad \text{avec } \|s^{+}\|_2 = 1 \text{ et } s^{+} \perp V_p^{+}, \\ v_1^{+} & = & \gamma \displaystyle\prod_{i=1}^{m} (A - \mu_i I_n) v_1. \end{array} \right.$$

IRA : Algorithme

Sélectionner un vecteur normé v_1, les entiers p et m. Soient V_{p+1}^1 et $H_{p+1,p}^1$ obtenues après p itérations d'Arnoldi.

Pour $k = 1, \ldots,$ jusqu'à convergence

Obtention de V_{p+m+1}^k et $H_{p+m+1,p+m}^k$ suite à m itérations d'Arnoldi supplémentaires.

Calcul des estimations d'éléments propres de A avec H_{p+m}^k.

Sélection de m shifts μ_j.

Réalisation de m factorisations implicites QR de H_{p+m}^k avec μ_j.

$$\longmapsto Q,\ V_{p+m+1}^+,\ H_{p+m+1,p+m}^+,$$

$$\Longrightarrow \left| \begin{array}{l} V_p^{k+1} = V_p^+,\ H_{p,p}^{k+1} = H_{p,p}^+, \\ r = h_{p+1,p}^+ v_{p+1}^+ + h_{p+m+1,p+m}^k v_{p+m+1}^k e_{p+m}^t Q e_p, \\ \gamma = \|r\|_2,\ v_{p+1}^{k+1} = r/\gamma,\ h_{p+1,p}^{k+1} = \gamma. \end{array} \right.$$

Déflation

$\longrightarrow$ Principe : Prise en compte de modes propres convergés au cours d'IRA pour empêcher leur recalcul

- $*$ Purging : Modes propres non désirés.
- $*$ Locking : Modes propres désirés.

$\longrightarrow$ Question : Comment effectuer ces procédures de façon automatique ?

$\longrightarrow$ Hypothèse : $A[V_1, V_2] = [V_1, V_2] \begin{bmatrix} H_1 & G \\ \epsilon e_1 e_j^t & H_2 \end{bmatrix} + r_p e_p^t$

avec $V_1 \in \mathbb{R}^{n \times j}, \; H_1 \in \mathbb{R}^{j \times j}$.

$$(y, \mu) \text{ vecteurs de Ritz de } H_1 \longmapsto \|Ay - \mu y\|_2 = |\epsilon e_j^t y|$$

$$\hookrightarrow (y, \mu) \text{ modes propres convergés si } \epsilon \simeq 0.$$

Déflation : Locking et Purging

$\longrightarrow$ Locking : on gèle les colonnes de V_1

$$\Longrightarrow \left| \begin{array}{l} \text{redémarrages implicites effectués sur la base } V_2, \\ V_1 \text{ participe aux itérations d'Arnoldi,} \\ (I_n - V_1 V_1^t) A V_2 = V_2 H_2 + r_p e_{p-j}^t. \end{array} \right.$$

$*$ Purging : si A symétrique, on élimine les colonnes de V_1

$$V = V_2,\ H = H_2,\ G = 0 \Longrightarrow AV = VH + r_p e_{p-j}^t$$

$\hookrightarrow$ factorisation d'Arnoldi d'ordre $p-j$ à ϵ près.

$\longrightarrow$ A non symétrique :

$*$ Locking : on construit une base des vecteurs de Schur de A.

$*$ Purging : beaucoup plus complexe à mettre en œuvre.

Pourquoi le Purging ?

$\longrightarrow$ Remarque : Pourquoi ne pas faire des factorisations QR implicite avec pour shifts les valeurs propres convergés non désirées ?

$\longrightarrow$ Réponse :

$*$ Une implémentation efficace de QR inclut nécessairement la gestion de l'annulation éventuelle de termes sous-diagonaux de petites magnitudes pour décider d'une éventuelle déflation.

$*$ En arithmétique finie, l'algorithme QR présente une instabilité (dite discrète) quand le shift est proche d'une valeur propre de H au niveau des éléments sous-diagonaux de la matrice de Hessenberg générée.

Déflation : Dans la pratique

$$AV_p = V_p H_p + r_p e_p^t,$$

$\longrightarrow$ Locking

$$A[\tilde{V}_{i+j}, \tilde{V}_{m-i-j}] = [\tilde{V}_{i+j}, \tilde{V}_{m-i-j}] \begin{bmatrix} H_{i+j} & G_{i+j} \\ 0 & H_{m-i-j} \end{bmatrix} + r_p e_p^t + r_p w^t$$

avec $V_{1,i}$ et $V_{i+1,i+j}$ vecteurs à purger, respectivement à geler.

$\longrightarrow$ Purging

$$A\bar{V}_{p-i} = \bar{V}_{p-i}\bar{H}_{p-i} + \bar{r}_{p-i} e_{p-i}^t$$

où $\bar{V}_{1,j}$ base orthonormale de $V_{i+1,i+j}$ est gelée.

Accélération : Transformation spectrale de A

$\rightarrow$ Principe : on applique l'algorithme d'Arnoldi à $C = f(A)$

* Shift and Invert

$$\begin{aligned} C &= (A - \sigma I_n)^{-1}, \\ \lambda_C &= \frac{1}{\lambda_A - \sigma}. \end{aligned}$$

* Transformation rationnelle de Cayley

$$\begin{aligned} C &= (A - \sigma I_n)^{-1}(A - \tau I_n), \\ \lambda_C &= \frac{\lambda_A - \tau}{\lambda_A - \sigma}. \end{aligned}$$

* Transformation polynômiale : Chebyshev, Leja

$$\begin{aligned} C &= \rho(A), \\ \lambda_C &= \rho(\lambda_A). \end{aligned}$$

Algorithmes préconditionnés : Plan

→ Méthode de Davidson

→ Retour sur Jacobi

→ L'algorithme de Jacobi-Davidson

→ Convergence et améliorations

→ Redémarrage et déflation

→ Lanczos préconditionné

Algorithmes préconditionnés : Méthode de Davidson

$v_1, u_1 = v_1/\|v_1\|_2$.

Pour $k = 1, \ldots,$ jusqu'à convergence

$w_k = Au_k$.

$\forall\ j = 1,\ k-1 \quad B_{jk} = u_j^t w_k, B_{kj} = u_k^t w_j$.

$B_{kk} = u_k^t w_k$.

Calcul de la plus grande valeur propre de B_k : (θ, y),

$z = U_k y,\ r = Az - \theta z, t = \Big(\mathrm{diag}(A) - \theta I_n\Big)^{-1} r$.

Pour $j = 1, \ldots, k$

$t = t - (u_j^t t) u_j$.

Fin

$u_{k+1} = t/\|t\|_2$.

Algorithmes préconditionnés : Remarques sur Davidson

$\longrightarrow$ Méthode concue initiallement pour traiter les matrices hamiltoniennes rencontrées en Chimie Quantique le plus souvent à diagonale dominante.

$\longrightarrow$ Méthode d'Arnoldi généralisée : procédé différent d'expansion de l'espace de projection.

$\longrightarrow$ Comportement :

$\ast$ Convergence plus rapide qu'Arnoldi pour des matrices à diagonale dominante.

$\ast$ u_1 doit être proche du vecteur cherché.

$\longrightarrow$ Problème

$\ast$ $t = -r$: Génération des sous-espaces de projection d'Arnoldi.

$\ast$ $t = (A - I_n)^{-1} r$: Stagnation du sous-espace de projection.

Algorithmes préconditionnés : Retour sur Jacobi

$\longrightarrow$ L'approche du Complément Orthogonal : $JOCC$.

Soit $(e_1, \alpha = a_{11})$ une approximation de (u, λ) cherché avec u non normé. On pose $u = (1, z^t)^t$

$$A\begin{pmatrix} 1 \\ z \end{pmatrix} = \begin{pmatrix} \alpha & c^t \\ b & F \end{pmatrix}\begin{pmatrix} 1 \\ z \end{pmatrix} = \lambda\begin{pmatrix} \alpha & c^t \\ b & F \end{pmatrix}\begin{pmatrix} 1 \\ z \end{pmatrix}.$$

$$\text{Gauss-Jacobi} \Longrightarrow \left\{ \begin{array}{ccc} \theta_k & = & \alpha + c^t z_k, \\ (D - \theta_k I_n) z_{k+1} & = & (D - F) z_k - b. \end{array} \right.$$

$\longrightarrow$ Convergence rapide dès que A est à diagonale dominante forte.

Algorithmes préconditionnés : L'algorithme de Jacobi-Davison (1)

→ *JOCC* ou Davidson : Calcul d'une correction de u_k à l'aide de (u_k, θ_k)

→ Correction orthogonale : $t_k \perp u_k$ tel que $u = \gamma(u_k + t_k)$

$$A(u_k + t_k) = \lambda(u_k + t_k) \Longrightarrow (B - \lambda I_n)t_k = -r_k$$

avec $r_k = Au_k - \theta_k u_k$, $B = (I_n - u_k u_k^*)A(I_n - u_k u_k^*)$.

En supposant $\theta_k \simeq \lambda$, t_k est solution de

$$(I_n - u_k u_k^*)(A - \theta_k I_n)(I_n - u_k u_k^*)t_k = -r_k.$$

ou encore

$$\left\{ \begin{array}{rcl} (A - \theta_k I_n)t_k & = & -r_k + \alpha u_k, \\ \alpha & = & \dfrac{u_k^*(A - \theta_k I_n)^{-1} r_k}{u_k^*(A - \theta_k I_n)^{-1} u_k}. \end{array} \right.$$

$\hookrightarrow$ $t_k = -u_k + \alpha(A - \theta_k I_n)^{-1} u_k$: ceci est équivalent à une itération accélérée du Quotient de Rayleigh pour la matrice $(A - \theta_k I_n)^{-1}$.

Algorithmes préconditionnés : L'algorithme de Jacobi-Davison (2)

Soient $t = v_0$ et ϵ la précision souhaitée.

Pour $j = 1, \ldots,$ jusqu'à convergence

$\forall\, k = 1,\ j-1 \quad t = t - (v_k^t t) v_k.$

$v_j = t / \|t\|_2,\ v_j^A = A v_j.$

$\forall\, k = 1,\ j-1 \quad M_{k,j} = v_k^t v_j^A,\ M_{j,k} = v_j^t v_k^A,\ M_{j,j} = v_j^t v_j^A.$

Calcul de la plus grande valeur propre de M :$Ms = \theta s$.

$u = V_j s,\ r = Au - \theta u.$

Si $\|r\|_2 > \epsilon$, calcul de $t \perp u$ solution approchée de l'équation dite de correction

$$(I_n - uu^*)(A - \theta_k I_n)(I_n - uu^*) t = -r.$$

Algorithmes préconditionnés : Solution itérative de l'équation de correction

Généralement on possède un préconditionneur K à gauche de $A - \theta I_n$.

On définit

$$\begin{array}{rcl} \tilde{K} & = & (I - uu^t)K(I - uu^t), \\ \tilde{A} & = & (I - uu^t)(A - \theta I_n)(I - uu^t). \end{array}$$

On calcule $\hat{u}$ solution de $K\hat{u} = u$, on définit

$$\tilde{r} = \tilde{K}^{-1} r = \hat{r} - \frac{u^t \hat{r}}{\mu} \hat{u} \text{ avec } K\hat{r} = r \text{ et } \mu = u^* \hat{u}.$$

Application d'une méthode de sous-espace de Krylov pour la résolution de $\tilde{K}^{-1}\tilde{A}x = \tilde{K}^{-1}r$ avec pour initial guess $t_0 = 0$. Le produit $z = \tilde{K}^{-1}\tilde{A}v$ s'effectue suivant

$$\left| \begin{array}{rcl} y & = & (A - \theta I_n)v, \\ z & = & \hat{y} - \frac{u^t \hat{y}}{\mu} \hat{u} \quad \text{avec} \quad K\hat{y} = y. \end{array} \right.$$

Algorithmes préconditionnés : Convergence et Améliorations

$$\left| \begin{array}{lcl} \text{Préconditionneur exact} & \Longrightarrow & \text{Méthode de Newton exacte,} \\ \text{Préconditionneur inexact} & \Longrightarrow & \text{Méthode de Newton inexacte.} \end{array} \right.$$

$\rightarrow$ Convergence dans le cas d'un préconditionneur exact

- $*$ Matrice symétrique : cubique.
- $*$ Matrice non symétrique : quadratique.

$\rightarrow$ Améliorations

- $*$ Redémarrage : combinaison linéaire de vecteurs de Ritz.
- $*$ Calcul de plusieurs modes propres: algorithme par bloc, déflation.

Algorithmes préconditionnés : Calcul de plusieurs modes propres par déflation

$\longrightarrow$ Base de projection orthonormale :

$\hookrightarrow$ Calcul des vecteurs de Schur de A.

$\longrightarrow$ Calcul des vecteurs de Schur de A un par un

$*$ Comment tenir compte des vecteurs de Schur convergés ?

$$A - \theta I_n \longrightarrow (I - Q_k Q_k^*)(A - \theta I_n)(I - Q_k Q_k^*)$$

où Q_k désigne la matrice des k vecteurs de Schur convergés.

$*$ Quel est l'effet sur le préconditionneur ?

$$\tilde{K} = (I_n - \tilde{Q}\tilde{Q}^*)K(I_n - \tilde{Q}\tilde{Q}^*)$$

où $\tilde{Q} = (Q_k, u_{k+1}^m)$, u_{k+1}^m m$^{\text{ième}}$ itéré de Jacobi-Davidson pour le calcul de q_{k+1}.

Algorithmes préconditionnés : l'algorithme JDQR (1)

$\longrightarrow$ Algorithmes pour valeurs propres extérieures

Soient v_0 le vecteur intial, ϵ une tolérance, τ un réel, $k_{\max}$ le nombre de modes cherchés proche de la cible τ et $m_{\max}$, $m_{\min}$ les dimensions de travail minimale (aprs̀ redémarrage) et maximale.

$t = v_0$, $k = 0$, $m = 0$, $Q = [\,]$, $R = [\,]$.

Tant que $k < k_{\max}$

$m = m + 1$.

Génération de v_m et $M_{m,m}$ à partir de t, V_{m-1} et $M_{m-1,m-1}$.

Décomposition de Schur de $M_{m,m} = STS^*$ avec

$$|T_{i,i} - \tau| \leq |T_{i+1,i+1} - \tau|,$$

$u = Vs_1$, $\theta = T_{1,1}$, $r = Au - \theta u$, $\tilde{a} = Q^* r$, $\tilde{r} = r - Q\tilde{a}$.

Algorithmes préconditionnés : l'algorithme JDQR (2)

Tant que $\|\tilde{r}\|_2 \leq \epsilon$

$$R = \begin{pmatrix} R & \tilde{a} \\ 0 & \theta \end{pmatrix}, \; Q = [Q, u], \; k = k + 1.$$

Si $k = k_{\max}$, arrêt de l'algorithme.

$m = m - 1, \forall\, i = 1,\, m \quad v_i = V s_{i+1},\; s_i = e_i.$

Soit M la matrice inférieure bloc de T de taille $m \times m$

$u = v_1,\, \theta = M_{1,1},\, r = Au - \theta u,\, \tilde{a} = Q^* r,\, \tilde{r} = r - Q\tilde{a}.$

Fin

Si $m \geq m_{\max}$ alors redémarrage : $m = m_{\min}, \forall i = 2,\, m_{\min} \quad v_i = V s_i$

$v_1 = u$, M la matrice inférieure bloc de T de taille $m_{\min} \times m_{\min}$.

$\tilde{Q} = [Q, u]$, $t \perp \tilde{Q}$ solution approchée de

$$(I_n - \tilde{Q}\tilde{Q}^*)(A - \theta I_n)(I_n - \tilde{Q}\tilde{Q}^*)t = -\tilde{r}.$$

Algorithmes préconditionnés: Commentaires sur JDQR

→ Algorithmes pour valeurs propres intérieures

* Eléments de Ritz harmoniques et Quotient de Rayleigh .

→ Décomposition de Schur de $M_{m,m}$

$\hookrightarrow$ Factorisation QR

→ Valeur propre complexe :

$\hookrightarrow$ Déflation de son conjugué.

→ Test d'arrêt sur les vecteurs de Schur :

Précision similaire sur les éléments propres si les éléments propres sont simples et assez séparés.

Algorithmes préconditionnés : Méthode de Davidson par bloc

$\longrightarrow$ Calcul des l plus grandes valeurs propres

Choisir V_1 une matrice orthonormée.

Pour $k = 1, \ldots,$ jusqu'à convergence

Calcul de la matrice projetée $M_k = V_k^t A V_k$.

Calcul des l plus grands modes propres de M_k : $(y_i^k, \mu_i^k)_{k=1,l}$.

Calcul des résidus : $\forall\, i = 1,\ l \quad r_i^k = (A - \mu_i^k I_n) V_k y_i^k$.

$\forall\, i = 1,\ l \quad \|r_i^k\|_2 < \epsilon$: arrêt de l'algorithme.

$\forall\, i = 1,\ l \quad t_i^{k+1} = -(C_i^k)^{-1} r_i^k \quad$ avec $\quad C_i^k \simeq (A - \mu_i^k I_n)$.

Pour tout $i = 1,\ l$, on orthogonalise t_i^{k+1} par rapport à V_k et $(t_j^{k+1})_{j=1,i-1}$.

$V_{k+1} = (V_k, t_1^{k+1}, \ldots, t_l^{k+1})$.

Algorithmes préconditionnés : Lanczos préconditionné (Morgan et Scott)

Soit v_0 et $\rho_0 = \dfrac{(v_0^t A v_0)}{(v_0^t v_0)}$.

Pour $k = 1, \ldots,$ jusqu'à convergence

Choisir une matrice M_k définie positive estimant $A - \rho_k I_n$.

Factoriser $M_k = L_k L_k^t$.

Poser $W_k = L_k^{-1}(A - \rho_k I_n) L_k^{-t}, w = L_k^t v_k$.

Appliquer un algorithme de Lanczos avec la matrice W_k jusqu'à la convergence de la plus petite valeur propre : (y_k, μ_k)

$$\begin{aligned} v_{k+1} &= L_k^{-t} y_k, \\ \rho_{k+1} &= \frac{v_{k+1}^t A v_{k+1}}{v_{k+1}^t v_{k+1}} = \rho_k + \frac{\mu_k}{v_{k+1}^t v_{k+1}}. \end{aligned}$$

Fin

Synthèse : Remarques générales

→ Prétraitement de la matrice à diagonaliser : Balancing

→ Les versions par bloc des algorithmes de sous-espace présentés et les versions avec arithmétique complexe existent. La robustesse des versions par bloc pour isoler des valeurs propres multiples les rendent indispensables dans de tels cas.

→ Pour des petites matrices, l'algorithme QR constitue l'algorithme de choix : des algorithmes très robustes sont implémentés dans les packages standards type LAPACK.

→ Concernant les algorithmes de sous-espace, la procédure d'Arnoldi est très stable par rapport à un algorithme de Bi-Lanczos. Il est à noter que les algorithmes de type Davidson ont permis de traiter des problèmes difficiles pour Arnoldi.

→ Pour des matrices pleines, les algorithmes "Divide and Conquer" sont supérieurs aux algorithmes QR pour $n > 30$.

Synthèse : Remarque générale : Krylov versus préconditionné

$\rightarrow$ On considère une matrice A symétrique et on cherche la plus petite valeur propre.

$*$ Algorithme de Krylov :

On minimise le quotient de Rayleigh sur un espace de plus en plus gros : méthode variationnelle.

$*$ Algorithme préconditionné :

On approche la minimisation du quotient de Rayleigh sur l'espace tout entier : méthode non variationnelle.

Synthèse : Tableau de synthèse : Matrice A tridiagonale symétrique

	Comple -xité	k vecteurs propres	k valeurs propres	Parallé -lisme
Réduction	0	$\times$	$\times$	$\times$
QR	$\mathcal{O}(n^3)$	$\mathcal{O}(n^3)$	$\mathcal{O}(n^2)$	non
Bissection	$\mathcal{O}(n^2)$	non	$\mathcal{O}(kn)$	non
Shift and Invert	$\mathcal{O}(n^2)$	$\mathcal{O}(kn)$	non	oui
Divide & Conquer	$\mathcal{O}(n^3)$	non	non	oui
Holy-Grail	$\mathcal{O}(n^2)$	$\mathcal{O}(kn)$	$\mathcal{O}(kn)$	oui
Sous-Espaces	$\mathcal{O}(n^3)$	$\mathcal{O}(k^2n)$	$\mathcal{O}(k^2n)$	non

Remarque : Cas de valeurs propres non séparées

$*$ S.I. : $\mathcal{O}(k^2n)$ pour k vecteurs propres (réorthogonalisation).

$*$ DC : $\mathcal{O}(n^\alpha, \alpha \in [1,2])$ opérations (nombreuses déflations).

Synthèse : Tableau de synthèse : Matrice A dense

	Comple -xité	k vecteurs propres	k valeurs propres	Parallé -lisme
Réduction	$\mathcal{O}(n^3)$	$\times$	$\times$	$\times$
QR	$\mathcal{O}(n^3)$	$\mathcal{O}(n^3)$	$\mathcal{O}(n^3)$	non
Bissection	non	non	non	non
Shift and Invert	$\mathcal{O}(n^4)$	$\mathcal{O}(kn^3)$	non	oui
Divide & Conquer	non	non	non	non
Holy-Grail	non	non	non	non
Sous-espaces	$\mathcal{O}(n^3)$	$\mathcal{O}(kn^2 + k^2n)$	$\mathcal{O}(kn^2 + k^2n)$	non

Synthèse : Tableau de synthèse : Matrice A creuse

	Comple -xité	k vecteurs propres	k valeurs propres	Parallé -lisme
Réduction	$\mathcal{O}(n^3)$	$\times$	$\times$	$\times$
QR	$\mathcal{O}(n^3)$	$\mathcal{O}(n^3)$	$\mathcal{O}(n^3)$	non
Bissection	non	non	non	non
Shift and Invert	$\mathcal{O}(n^3)$	$\mathcal{O}(k^2n)^* + \mathcal{O}(kn^\alpha)$	non	oui
Divide & Conquer	non	non	non	non
Holy-Grail	non	non	non	non
Sous-espaces	$\mathcal{O}(n^3)$	$\mathcal{O}(k^2n)$	$\mathcal{O}(k^2n)$	non

$*$ si réorthogonalistion.

$\mathcal{O}(n^\alpha)$ complexité de l'algorithme de résolution de systèmes linéaires.

Synthèse : Tableau de synthèse pour A quelconque

On suppose que A vérifie $\mathcal{H}$ et que les éléments propres sont simples.

	A creuse sym.	A dense sym.	A creuse non sym.	A dense non sym.
k v. p.	S.E.	R.+ B.	S.E.	(R.+) S.E.
k e.p.	S.E.	R.+ H.G.	S.E.	(R.+) S.E.
n v.p.	R.+ D.C.	R+ D.C.	S.E.	R. +QR
n e.p.	R + D.C., H.G.	R + D.C., H.G.	S.E.	R. + QR

v.p. : valeurs propres

e.p. : éléments propres

Synthèse : Les voies d'avenir

→ Les algorithmes de Jacobi-Davidson

→ Les algorithmes de Davidson par bloc

→ Les algorithmes en $\mathcal{O}(n^2)$ pour des matrices bandes

→ Fast Multiple Method pour "Divide and Conquer"

→ Les méthodes d'Homotopie

→ Parallélisation des méthodes de réduction

→ Algorithmes parallèles basés sur des techniques de Décomposition de Domaines

Matrice fortement non normale

$\longrightarrow$ Conséquence : Problèmes très mal conditionnés très sensibles aux erreurs d'arrondis.

$\longrightarrow$ Développement d'outils d'analyse

- $*$ Portrait Spectral.
- $*$ Pseudo-Spectre.

$\longrightarrow$ Utilisation de procédures de réorthogonalisation très robustes.

$\longrightarrow$ Classification : introduction de la stabilité inverse

- $*$ Arnoldi associé à Gram-Schmidt Modifié n'est pas toujours stable inverse.
- $*$ Arnoldi associé à des procédures de réorthogonalisation plus robuste est stable inverse.
- $*$ QR est stable inverse.
- $*$ Les itérations de sous-espaces ne sont pas stables inverse.

Problème généralisé

→ Problème

$$\text{Chercher } (x,\lambda) \quad Ax = \lambda Bx$$

où A et B sont deux matrices de $\mathbb{R}^{n\times n}$. On les suppose réelles.

→ Différentes approches

∗ Analogue de QR : algorithme QZ.

∗ Méthodes itératives après réduction sous une forme standard : Méthodes de Krylov Rationnelles.

∗ Méthodes itératives sans réduction préalable : Généralisation de Jacobi-Davidson.

∗ Méthodes “Divide and Conquer” et d’Homotopie (moins développées).

Problème généralisé : Réduction à un problème standard (1)

→ B non singulière et bien conditionnée :

$$Ax = \lambda Bx \quad \Longleftrightarrow \quad (B^{-1}A)x = \lambda x.$$

→ B symétrique définie positive et bien conditionnée :

$$\begin{aligned} &Ax = \lambda Bx \quad \Longleftrightarrow \quad (L^{-1}AL^{-t})y = \lambda y, \\ &B = LL^t, \quad y = L^t x. \end{aligned}$$

→ B singulière ou mal conditionnée :

$$\begin{aligned} &Ax = \lambda Bx \quad \Longleftrightarrow \quad Cx = \mu c, \\ &C = (A - \sigma B)^{-1}B, \quad \mu = \frac{1}{\lambda - \mu}. \end{aligned}$$

$$\begin{aligned} &Ax = \lambda Bx \quad \Longleftrightarrow \quad Cx = \mu c, \\ &C = (A - \sigma B)^{-1}(A - \tau B), \quad \mu = \frac{\lambda - \tau}{\lambda - \mu}. \end{aligned}$$

Problème généralisé : Réduction à un problème standard (2)

$\longrightarrow$ Problème : inversion d'un système linéaire à chaque itération

- $*$ Factorisation (creuse) LU,
- $*$ Solveur itératif.

$\longrightarrow$ Méthodes de Krylov Rationnelles : génération d'un espace adéquat

$\hookrightarrow$ Inversion précise du système linéaire.

Lorsque ce n'est pas possible, il faut avoir recours à l'algorithme QZ de type Jacobi-Davidson.

Problèmes ouverts

→ Parallélisation de QR

→ Compréhension de la duplication périodique de valeurs propres avec Lanczos en arithmétique finie

→ Compréhension d'Arnoldi et de Bi-Lanczos pour A non symétrique

→ Calcul de sous-espaces invariants pour A fortement non normale

→ Développement d'algorithmes efficaces pour A tridiagonale non symétrique

→ Bonnes approximations de A^{-1} pour A de grande taille et creuse

→ Algorithmes efficaces pour la SVD de matrices de grande taille et creuse

→ Problèmes polynômiaux de grande taille et creux (acoustique)

→ Problèmes aux valeurs propres contraints (problèmes de Stokes)

Application au problème de la Chimie Quantique : plan

$\longrightarrow$ Problème

$\longrightarrow$ Approches classiques

$\longrightarrow$ Cas A possède un grand gap

$\longrightarrow$ Cas A possède un petit gap

Application au problème de la Chimie Quantique : problème (1)

$$(\mathcal{P}_0) \begin{cases} \text{Chercher } (C, \Lambda) \in \mathcal{M}(N_b, N) \times \mathcal{M}_S(N) \text{ tels que} \\ FC - SC\Lambda = 0, \\ C^* SC = I_N. \end{cases}$$

où $N_b = \mathcal{O}(N)$, F et S sont deux éléments creux de $\mathcal{M}_S(N_b)$.

Parmi les solutions de $(\mathcal{P}_0)$, on cherche celle qui rend minimale

$$\mathrm{Tr}(FCC^*).$$

Application au problème de la Chimie Quantique : problème (2)

$\longrightarrow$ Hypothèse : $S = I_d$ et $\lambda_N \neq \lambda_{N+1}$.

$\longrightarrow$ Solution : C est une base orthonormée du sous-espace propre engendré par les vecteurs propres associés aux N plus petites valeurs propres de F.

Si on choisit de calculer la base des vecteurs propres cherchés, $(\mathcal{P}_0)$ devient

$$(\mathcal{P}_1) \begin{cases} \text{Calculer } (C_i, \epsilon_i)_{i=1,N} \text{ tels que} \\ FC_i = \epsilon_i C_i, \\ C_i^* C_j = \delta_{ij}. \end{cases}$$

où $(\epsilon_i)_i$ sont les N plus petites valeurs propres de F.

Approches classiques

$\longrightarrow$ Méthodes directes : $\mathcal{O}(N^3)$

$\longrightarrow$ Méthodes H.G ou D.C : $\mathcal{O}(N^3)$ car Réduction.

$\longrightarrow$ Méthode S.E. ou S.I. : $\mathcal{O}(N^3)$ car N valeurs propres à calculer.

$\hookrightarrow$ La résolution de $(\mathcal{P}_1)$ nécessite $\mathcal{O}(N^3)$ calculs.

$\longrightarrow$ Constat : Il est donc nécessaire de développer de nouvelles méthodes pour résoudre $(\mathcal{P}_1)$ pour de grands systèmes.

$\longrightarrow$ Point clé : La grandeur pertinente est $D = CC^*$.

Vers l'ordre N

$\longrightarrow$ Gap : $\gamma = \lambda_{N+1} - \lambda_N$

$$\exists\ (m,p) \in \mathbb{N} \times P_m(\mathbb{R}) \quad D = p(F) \text{ et } m = m(\gamma).$$

$\longrightarrow$ Cas F possède un grand gap : m petit $\longmapsto$ D creuse.

$\hookrightarrow$ Ordre N est accessible

$\longrightarrow$ Cas F possède un petit gap : m grand $\longmapsto$ D pleine.

$\hookrightarrow$ Ordre N^2 a priori seulement accessible

$\longrightarrow$ Idées :

- On développe des stratégies pour le calcul de C.
- On s'intéresse aux éléments $D_{i,j}$ tels que $F_{i,j} \neq 0$.