Équations aux dérivées partielles et Analyse numérique : Résolution de grands systèmes linéaires provenant de la discrétisation d'une EDP
Cours et TD

Bernhard Beckermann
Laboratoire Paul Painlevé UMR 8524
Université de Lille
59655 Villeneuve d'Ascq CEDEX
e-mail : Bernhard.Beckermann@univ-lille.fr

19/3/2020 : une première version du chapitre 1

24/3/2020 : ajout de 3 exercices au §1.5

24/3/2020 : une première version des chapitres §2.1–§2.4

29/3/2020 : ajout de la fin du chapitre 2, §2.5–§2.8, et plein d'exercices.

4/4/2020 : reformulation des exos 1.5.5, 2.3.3 et 2.3.6 (exo 2.3.5 me parait ok)

6/4/2020 : correction de quelques erreurs de frappe et reformulation du théorème 2.4.2

1/5/2020 : correction de quelques noncs d'exercices dans le chapitre 2.7

Table des matières

1 Introduction 3

1.1 A propos de ce texte 3
1.2 Quelques pré-requis 3
1.3 Discrétisation d'une EDP elliptique par différences finies 6
1.4 Discrétisation d'une EDP elliptique par éléments finis 10
1.5 Les projecteurs 12

2 Résolution de grands systèmes creux 16

2.1 Motivation 16
2.2 Méthodes de projection 17
2.3 La méthode de la plus forte descente 19
2.4 Le gradient conjugué 23
2.5 La méthode d'Arnoldi 28
2.6 Les méthodes FOM et GMRES 30
2.7 D'autres exercices sur les méthodes de Krylov 33
2.8 Le gradient conjugué préconditionné 40

3 Le calcul approché de valeurs propres par des techniques de projection 43

Bibliographie 44

Chapitre 1

Introduction

1.1 A propos de ce texte

Ce document contient le cours et les exercices de la partie "Résolution de grands systèmes linéaires provenant de la discrétisation d'une EDP" du cours ''Équations aux dérivées partielles et Analyse numérique" dispensé (élecroniquement avec le site moodle) en avril 2020 dans le Master 1 de Mathématiques de l'Université de Lille. Nous commencerons à rappeler quelques pré-requis pour ce module, et donnerons ensuite l'origine et quelques propriétés des grands systèmes creux liés à résolution numérique d'un certain nombre d'EDP en dimension $d \in \{1, 2\}$. Ici nous ferons appel aux éléments finis déjà étudiés, mais nous parlerons également d'une discrétisation par différences finies.

Nous aborderons ensuite le cadre général des méthodes de projection, avec comme cas particuliers la Méthode de la plus forte pente et celle du Gradient Conjugué, applicable si la matrice de coefficients est symétrique définie positive (sdp). Nous enchaînerons ensuite avec plusieurs méthodes des sous-espaces de Krylov (Arnoldi, FOM, GMRES, Lanczos, BiCGStab), nécessitant aucune hypothèse sur la matrice. Quelques techniques de préconditionnement seront étudiés (SSOR et factorisation de Choleski incomplète) pour accélérer la vitesse de convergence de ces méthodes itératives. Finalement, nous donnerons quelques éléments sur le calcul approché des valeurs propres, et une bibliographie avec plusieurs livres trouvables sur internet.

1.2 Quelques pré-requis

Au moins implicitement, ce cours est la suite d'un module L3 d'Algèbre linéaire numérique, où vous avez vu la résolution des systèmes d'équations linéaires par le pivot de Gauss (décomposition LU, pivotage), ainsi que la résolution des problèmes de moindres carrés (décomposition QR, transformations de Givens/Householder), voir par exemple [AD08, Sections 1,6,7,8].

Dans ce document, nous utilisons seulement la norme euclidienne $\|x\| = \sqrt{\sum_{j=1}^{n} |x_j|^2}$ pour les vecteurs $x = (x_1, ..., x_n)^* \in \mathbb{C}^n$, ainsi que deux normes matricielles, la norme spectrale et la

norme de Froebenius

$$\|A\| := \sup_{x\neq 0} \frac{\|Ax\|}{\|x\|} = \sqrt{\rho(A^*A)} \leq \|A\|_F := \sqrt{\sum_{j,k} |A_{j,k}|^2},$$

avec $\rho(B) = \max\{|\lambda| : \lambda \text{ valeur propre de } B\}$ le rayon spectral d'une matrice carrée B. On remarque que, pour une matrice inversible A (et donc en particulier carrée), $\text{cond}(A) := \|A\|\,\|A^{-1}\| \geq 1$. Pour une matrice carrée E de norme $\|E\| < 1$, on peut former la série de Neumann

$$(I-E)^{-1} = \sum_{j=0}^{\infty} E^j, \quad \|(I-E)^{-1}\| \leq \sum_{j=0}^{\infty} \|E\|^j = \frac{1}{1-\|E\|}, \tag{1.1}$$

où I dénote une matrice identité (de taille appropriée), $E^0 = I$, et E^{j+1} est le produit E fois E^j (on peut montrer que les sommes partielles de cette série forment une suite de Cauchy, donc admettent une limite).

D'autres propriétés seront importantes.

1.2.1. Matrices diagonalisables et bases de vecteurs propres

Une matrice $A \in \mathbb{C}^{n\times n}$ est dite diagonalisable *si $\exists V \in \mathbb{C}^{n\times n}$ inversible $\exists D = \text{diag}\,(\lambda_1, ..., \lambda_n)$ tels que $AV = VD$. Ceci implique que les λ_j sont les valeurs propres de A, et la $j$ième colonne de V contient le vecteur propre associé (à droite) de A. Comme $V^{-1}A = DV^{-1}$, la $j$ième ligne de V^{-1} (plus précisement la $j$ième colonne de l'adjoint $(V^{-1})^*$) contient le vecteur propre à gauche de A. Donc A est diagonalisable ssi il existe une base de vecteurs propres.*

Si A est normale (ce qui veut dire $A^*A = AA^*$) ou hermitienne ($A = A^*$) alors il existe une base orthonormée de vecteurs propres, autrement dit, il existe une matrice V unitaire (ce qui veut dire ($V^*V = I$ où, en mots, V admet des colonnes orthonormées) avec $V^{-1}AV = V^*AV$ une matrice diagonale.

1.2.2. Forme de Schur, Forme de Jordan

Pour toute matrice $A \in \mathbb{C}^{n\times n}$ il existe une matrice unitaire $V \in \mathbb{C}^{n\times n}$ tel que V^*AV est une matrice triangulaire supérieure (dit forme de Schur). Notons que, forcément, cette matrice triangulaire comporte les valeurs propres de A sur la diagonale.

Finalement, pour toute matrice $A \in \mathbb{C}^{n\times n}$ il existe une matrice $V \in \mathbb{C}^{n\times n}$ inversible (un changement de base) tel que $V^{-1}AV$ est une matrice diagonale par blocs, chacun des blocs étant de la forme bidiagonale

$$J(\lambda) = \begin{bmatrix} \lambda & 1 & 0 & \cdots & 0 \\ & \ddots & \ddots & \ddots & \vdots \\ 0 & & & & \\ \vdots & \ddots & \ddots & \ddots & 0 \\ & & & & \\ \vdots & & \ddots & \ddots & 1 \\ 0 & \cdots & \cdots & 0 & \lambda \end{bmatrix},$$

de taille variable (peut être réduit au scalaire λ) la taille de ces blocs et leur nombre étant un invariant. En particulier, A est diagonalisable ssi tous ses blocs de Jordan sont d'ordre 1*, et une condition suffisante est que A admet des valeurs propres distincts. Exemple :*

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad \textit{est un bloc de Jordan et donc pas diagonalisable.}$$

On termine ce chapitre en rappelant la définition suivante : une matrice A est dite symétrique définie positive (abrégée sdp) si $A = A^*$ (symétrique dans le cas réel, hermitienne sinon), et si pour tout vecteur $x \neq 0$ nous avons $x^*Ax > 0$.

1.2.3. Exercice : *Montrer qu'une matrice hermitienne admet des valeurs propres réelles. Plus précisément, soit A sdp. En notant par $\lambda_{\min}$ et $\lambda_{\max}$ la plus petite valeur propre de A (et la plus grande, respectivement), montrer que*

$$\lambda_{\max} = \max_{x\neq 0} \frac{x^*Ax}{x^*x} = \|A\|, \quad \lambda_{\min} = \min_{x\neq 0} \frac{x^*Ax}{x^*x} = \frac{1}{\|A^{-1}\|}, \quad \mathit{cond}(A) = \frac{\lambda_{\max}}{\lambda_{\min}}.$$

1.2.4. Exercice (Inégalité de Kantorovitch) :
Soit B une matrice symétrique définie positive réelle de dimension n, et soit

$$0 < \lambda_1 \leq \cdots \leq \lambda_n,$$

ses valeurs propres. On veut montrer l'inégalité :

$$\forall x \neq 0, \quad \frac{(Bx,x)(B^{-1}x,x)}{(x,x)^2} \leq \frac{(\lambda_1+\lambda_n)^2}{4\lambda_1\lambda_n}.$$

1. *En utilisant une diagonalisation de B, mettre le membre de gauche sous la forme*

$$\left(\sum_i \beta_i\lambda_i\right)\left(\sum_i \frac{\beta_i}{\lambda_i}\right)$$

   où les β_i sont des coefficients à définir.
2. *On pose $\lambda = \sum_i \beta_i\lambda_i$. En utilisant la convexité de la fonction $1/x$ sur l'intervalle $[\lambda_1, \lambda_n]$, montrer que*

$$\left(\sum_i \beta_i\lambda_i\right)\left(\sum_i \frac{\beta_i}{\lambda_i}\right) \leq \lambda(a\lambda + b),$$

   où a et b sont deux coefficients à définir.
3. *En déduire l'inégalité de Kantorovitch.*

1.2.5. Exercice :

(a) *Pour une matrice $M \in \mathbb{C}^{n\times n}$ sdp, montrer que l'on peut définir un produit scalaire et une norme (dite* norme d'énergie*) par les expressions*

$$\forall x, y \in \mathbb{C}^n : \quad \langle x, y\rangle_M := y^*Mx, \quad \|x\|_M = \sqrt{\langle x, x\rangle_M}.$$

(b) *Supposons que l'on dispose d'une factorisation $M = F^*F$ (par exemple de Choleski). Vérifier que $\|x\|_M = \|Fx\|$.*

(c) *Supposons que A est une autre matrice sdp. En notant par $\lambda_{\min}$ et $\lambda_{\max}$ la plus petite valeur propre de $M^{-1}A$ (et la plus grande, respectivement), montrer que*

$$\lambda_{\max} = \max_{x\neq 0} \frac{x^*Ax}{x^*Mx} = \|F^{-*}AF^{-1}\|, \quad \lambda_{\min} = \min_{x\neq 0} \frac{x^*Ax}{x^*Mx} = \frac{1}{\|FA^{-1}F^*\|}, \quad \mathit{cond}(F^{-*}AF^{-1}) = \frac{\lambda_{\max}}{\lambda_{\min}}.$$

1.3 Discrétisation d'une EDP elliptique par différences finies

Dans la suite de ce cours on s'intéresse à résoudre d'une manière approchée les deux problèmes elliptiques suivants.

1.3.1. Le problème de Poisson en dimension d
Étant donné un ouvert (simplement connexe) $\Omega \subset \mathbb{R}^d$ et des fonctions f et g, f donné sur Ω, g donné sur le bord $\partial\Omega$ de Ω, on cherche une fonction u définie sur la fermeture de Ω de classe $\mathbb{C}^2$ (au moins) de sorte que

$$\begin{aligned} \forall x \in \Omega : \quad & -\Delta u(x) := -\sum_{j=1}^{d} \Big(\frac{\partial}{\partial x_j}\Big)^2 u(x) = f(x), \\ \forall x \in \partial\Omega : \quad & u(x) = g(x) \quad \textit{dites conditions au bord.} \end{aligned}$$

Dans la suite on s'intéresse en particulier au cas $d = 1$ avec $\Omega =]0, 1[$ un intervalle ouvert, avec les deux extrémités $\partial\Omega = \{0, 1\}$ où $-\Delta u(x) = -u''(x)$ fait apparaître la dérivée seconde. Si on suppose que les primitives de f sont connues (par exemple dans le cas ou f est un polynôme, ou carrément $f = 0$), ce problème à plutôt un intérêt pédagogique car pour trouver u vérifiant $-u''(x) = f(x)$ on doit juste former deux primitives, et adapter des constantes d'intégration pour vérifier les 2 conditions au bord (en 0 et en 1).

Le problème de Poisson est bien plus intéressant dans le cas $d = 2$, par exemple $\Omega =]0, 1[^2$ le carré unité, $\partial\Omega$ comportant les 4 arêtes. Pour motivation, on pourrait s'imaginer u un potentiel électrique en électro-statique (pas de dépendance du temps, autrement dit, on cherche l'équilibre) : on impose une charge électrique g sur le bord $\partial\Omega$, et dans Ω le potentiel est une fonction dite harmonique (ce qui revient à dire que le Laplacien s'annule sur Ω). Une autre interprétation peut être la répartition de la température u dans une salle Ω en imposant la température aux murs (il n'y a pas échange de température avec l'extérieur). Ici le cas $f = 0$ modélise la diffusion de la température sans sources de chaleur, sinon un terme f non nul peut être une source unitaire de chaleur. Encore une fois, on s'interesse à des solutions stationnaires ne dépendant pas du temps (l'équilibre), ce qui se voit encomparant notre EDP avec l'équation générale de chaleur.

Si le matériau n'est pas homogène, on doit faire face à un coefficient de conductivité κ qui donne lieu a un problème légèrement plus complexe.

1.3.2. Le problème de diffusion en dimension d
Étant donné un ouvert (simplement connexe) $\Omega \subset \mathbb{R}^d$ et des fonctions f, κ et g, f, κ donné sur Ω, g donné sur le bord $\partial\Omega$ de Ω, on cherche une fonction u définie sur la fermeture de Ω de classe $\mathbb{C}^2$ (au moins) de sorte que

$$\begin{aligned} \forall x \in \Omega : \quad & -\sum_{j=1}^{d} \Big(\frac{\partial}{\partial x_j}\Big(\kappa\, \frac{\partial u}{\partial x_j}\Big)\Big)(x) = f(x), \\ \forall x \in \partial\Omega : \quad & u(x) = g(x) \quad \textit{dites conditions au bord,} \end{aligned}$$

où le cas $\kappa = 1$ permet de revenir à notre problème de Poisson.

Pour Ω un produit cartésien d'intervalles (ici $\Omega =]0,1[^d$), l'idée des différences finies est que l'on discrétise la ferméture $[0,1]^d$ de Ω par un maillage d'un nombre fini de points, et au lieu de chercher une fonction on cherche juste à approcher les valeurs de la fonction sur ce maillage de points. Ceci nécessitera de remplacer des dérivées par des différences finies (autrement dit, on ne passe pas à la limite dans la définition d'une dérivée). Voir aussi [AD08, Chapitre 16.1 et 16.2] ou alors les ressources électroniques [H19, Chapitres 1.3, 1.5 et 1.8].

1.3.3. Quelques différences finies dits centrées

Montrer que, pour une fonction u d'une seule variable et suffisamment différentiable,

$$(a) \qquad \frac{u(x+h)-u(x-h)}{2h} = u'(x) + \mathcal{O}(h^2)_{h\to 0},$$

$$(b) \qquad \frac{2u(x)-u(x+h)-u(x-h)}{h^2} = -u''(x) + \mathcal{O}(h^2)_{h\to 0},$$

$$(c) \qquad \frac{u(x)(\kappa(x+\frac{h}{2})+\kappa(x-\frac{h}{2})) - u(x+h)\kappa(x+\frac{h}{2}) - u(x-h)\kappa(x-\frac{h}{2})}{h^2} = -(\kappa u')'(x) + \mathcal{O}(h^2)_{h\to 0}.$$

1.3.4. La méthode des différences finies en dimension $d = 1$

On introduit pour un entier $N \gg 1$ le maillage (ici équidistant)

$$\forall j = 0,1,...,N+1 : \quad x_j = jh, \quad h = \frac{1}{N+1} \quad \textit{(dit pas)},$$

et on cherche à trouver u_j approximation de $u(x_j)$ de sorte que

$$\forall j = 1,...,N : \quad \frac{2u_j - u_{j+1} - u_{j-1}}{h^2} = f(x_j)$$

pour le problème de Poisson (égalité en tout point à l'intérieur du maillage) ou les valeurs $u_0 = g(x_0) = g(0)$ et $u_{N+1} = g(x_{N+1}) = g(1)$ sont imposées. Autrement dit, on cherche les N composantes du vecteur x solution du système $Bx = b$, avec la matrice tridiagonale

$$B = \begin{bmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -1 \\ 0 & \cdots & 0 & -1 & 2 \end{bmatrix} \in \mathbb{R}^{N\times N}, \quad x = \begin{bmatrix} u_1 \\ \vdots \\ u_N \end{bmatrix}, \quad b = \begin{bmatrix} g_0 + h^2 f_1 \\ h^2 f_2 \\ \vdots \\ h^2 f_{N-1} \\ g_{N+1} + h^2 f_N \end{bmatrix},$$

où on a utilisé les abréviations $f_j = f(x_j)$, $g_j = g(x_j)$. Un système triangulaire similaire est obtenu pour l'équation de diffusion en dimension $d = 1$, avec

$$B = \begin{bmatrix} \kappa_{\frac{1}{2}} + \kappa_{\frac{3}{2}} & -\kappa_{\frac{3}{2}} & 0 & \cdots & 0 \\ -\kappa_{\frac{3}{2}} & \kappa_{\frac{3}{2}} + \kappa_{\frac{5}{2}} & -\kappa_{\frac{5}{2}} & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -\kappa_{N-\frac{1}{2}} \\ 0 & \cdots & 0 & -\kappa_{N-\frac{1}{2}} & \kappa_{N-\frac{1}{2}} + \kappa_{N+\frac{1}{2}} \end{bmatrix} \in \mathbb{R}^{N\times N}, \quad b = \begin{bmatrix} \kappa_{\frac{1}{2}} g_0 + h^2 f_1 \\ h^2 f_2 \\ \vdots \\ h^2 f_{N-1} \\ \kappa_{N+\frac{1}{2}} g_{N+1} + h^2 f_N \end{bmatrix}.$$

1.3.5. Exercices :
(a) *Pour le problème de Poisson en dimension* 1*, montrer que* B *est symétrique définie positive, et que les éléments propres sont donnés pour* $k = 1, ..., N$ *par*

$$\lambda_k = 2(1 - \cos\frac{k\pi}{N+1}), \quad v_k = \frac{\sqrt{2}}{\sqrt{N+1}}(\sin\frac{k\pi}{N+1}, \sin\frac{2k\pi}{N+1}, \cdots, \sin\frac{Nk\pi}{N+1})^*.$$

On pourra se servir de la formule $x^T Bx = x_1^2 + x_N^2 + \sum_{j=2}^N (x_j - x_{j-1})^2$.
(b) *Pour le problème de diffusion en dimension* 1*, montrer que* B *est symétrique définie positive.*

Il se pose alors le problème suivant : comment résoudre d'une manière efficace un système triangulaire (et en plus symétrique). Vous avez probablement vu en L3 qu'il existe un algorithme en complexité $\mathcal{O}(N)$ obtenu en spécifiant la décomposition LU (ou alors Choleski) qui fait apparaître des matrices bidiagonales, voir par exemple [H19, Chapitre 1.8].

Par contre, en dimension $d \geq 2$, le système sous-jacent devient plus compliqué. Il nous faut d'abord l'équivalent de l'Exercice 1.3.4 pour des fonctions de $d \geq 2$ variables, ce qui peut encore être obtenu par le théorème de Taylor (on fixe toutes les variables sauf une).

1.3.6. La méthode des différences finies en dimension $d = 2$ pour Poisson
On introduit pour un entier $N \gg 1$ *le maillage (ici équidistant)*

$$\forall j, k = 0, 1, ..., N+1 : \quad x_{j,k} = \begin{bmatrix} jh \\ kh \end{bmatrix} \in \mathbb{R}^2, \quad h = \frac{1}{N+1} \quad \textit{(dit pas)},$$

et on cherche à trouver $u_{j,k}$ *approximation de* $u(x_{j,k})$ *de sorte que*

$$\forall j, k = 1, ..., N : \quad \frac{2u_{j,k} - u_{j+1,k} - u_{j-1,k}}{h^2} + \frac{2u_{j,k} - u_{j,k+1} - u_{j,k-1}}{h^2} = f(x_{j,k})$$

pour le problème de Poisson (égalité en tout point à l'intérieur du maillage) où maintenant les valeurs $u_{j,k} = g(x_{j,k})$ *sont imposées pour les indices de sorte que* $x_{j,k} \in \partial\Omega$*, c'est-à-dire, pour*

$$\begin{aligned}(j,k) \quad \in \quad & \{(0,k) : k = 1, ..., N\} \cup \{(N+1,k) : k = 1, ..., N\} \\ & \cup \{(j,0) : j = 1, ..., N\} \cup \{(j,N+1) : j = 1, ..., N\}.\end{aligned}$$

Notre schéma (dit à 5 points, il y en a d'autres dans la littérature) pour discrétiser le Laplacien relie alors une inconnue $u_{j,k}$ *au centre avec ses quatre voisins* $u_{j\pm1,k}$ *et* $u_{j,k\pm1}$ *(souvent nommés en se servant des 4 directions d'une rose des vents). Pour écrire un système d'équations linéaires* $Ax = b$ *avec* x *comportant nos* N^2 *valeurs inconnues, il nous faut encore fixer un ordre dans notre maillage, pour pouvoir énumérer les équations mais aussi nos inconnues. Ici on choisit une énumeration par ordre de* k *croissant et pour chaque* k *fixe par ordre de* j *croissant, ce qui donne*

$$x = (u_{1,1}, u_{2,1}, ..., u_{N,1}, u_{1,2}, ..., u_{1,N}, u_{2,N}, ..., u_{N,N})^*,$$

avec la matrice tridiagonale par blocs

$$A = \begin{bmatrix} B+2I & -I & 0 & \cdots & 0 \\ -I & B+2I & -I & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -I \\ 0 & \cdots & 0 & -I & B+2I \end{bmatrix} \in \mathbb{R}^{N^2 \times N^2}.$$

1.3.7. Exercices :
(a) *Construire élément par élément la matrice et le second membre pour* $N = 4$.
(b) *Vérifier que* A *admet au plus* 5 *élements non nuls par ligne.*
(c) *En vous inspirant du cas* $d = 1$*, vérifier que* A *est sdp.*
(d) *Comment généraliser cette étude au cas* $d = 3$ *?*
(e) *Comment généraliser cette étude au cas d'une équation de diffusion ?*

Contrairement au cas $d = 1$, il n'existe pas une méthode en complexité $O(N^2)$ (N^2 étant le nombre d'inconnues) pour résoudre notre système $Ax = b$. Il y a juste le cas $f = 0$ des fonctions harmoniques permettant de déduire un algorithme de complexité $O(N^2 \log(N)$. Il est fortement basé sur l'analyse de Fourier discrète (plus précisément la transformée DST de sinus discrète) par laquelle on peut calculer le produit Vy avec V la matrice unitaire des vecteurs propres de $B \in \mathbb{R}^{n\times N}$ (voir Exercice 1.3.5) et $y \in \mathbb{C}^N$ en complexité $\mathcal{O}(N \log N)$.

1.3.8. Exercices :
Considérons le problème de Poisson en dimension $d = 2$ *avec avec une source* $f = 0$ *triviale, et écrivons* $B = V\Lambda V^*$ *avec* V *la matrice unitaire des vecteurs propres donnés en Exo 1.3.5, et* Λ *la matrice diagonale des valeurs propres. .*
(a) *Vérifier que* $V = V^*$.
(b) *Montrer que le système d'équations linéaires donné dans 1.3.6 peut s'écrire comme*

$$BX + XB = G$$

avec $X = (u_{j,k})_{j,k} \in \mathbb{R}^{N\times N}$ *la matrice contenant nos inconnues, et rang*$(G) \leq 4$ *(venant des 4 segments formant le bord de* Ω*). En déduire qu'il existe* $M, N \in \mathbb{R}^{N\times 4}$ *tels que* $G = MN^*$.
(c) *Montrer que l'on obtient l'équation équivalente*

$$\Lambda Y + Y\Lambda = \widetilde{M}\widetilde{N}^*, \quad \widetilde{M} := VM, \quad \widetilde{N} := VN$$

dont la solution est explicitement donné par $Y = (\frac{(\widetilde{M}\widetilde{N}^*)_{j,k}}{\lambda_j+\lambda_k})_{j,k}$.
(d) *Donner un algorithme de complexité* $O(N^2 \log(N)$ *pour trouver* X.

On peut encore donner sans preuve d'autres propriétés pour nos matrices B (pour $d = 1$) et A (pour $d = 2$), qui sont également valables pour $d > 2$.

1.3.9. Propriétés du problème de diffusion discrétisé par DF :
Notons ici par $A_d^{DF}(\kappa) \in \mathbb{R}^{N^d\times N^d}$ *la matrice de coefficients obtenue par application des différences finies centrées au problème de diffusion sur* $\Omega =]0, 1[^d$*, et supposons que*

$$0 < \kappa_{\inf} := \inf_{x\in\Omega} \kappa(x) \leq \kappa_{\sup} := \sup_{x\in\Omega} < \infty, \tag{1.2}$$

alors $A_d^{DF}(\kappa)$ *admet des propriétés suivantes :*
(a) *elle est sdp, inversible, irréductible, à diagonale dominante ;*
(b) *elle est une* M*-matrice (son inverse admet seulement des éléments* > 0 *ce qui donne lieu à une propriété de principe de maximum discret) ;*
(c) *si* $\kappa = 1$*, sa plus petite valeur propre se comporte comme* $4\pi dh^2$*, sa plus grande valeur propre se comporte comme* $4d$ *pour* N *grand, en particulier cond*$(A_d^{DF}(1)) = O(N^2)_{N\to\infty}$ *;*
(d) *finalement, pour tout vecteur* $x \neq 0$ *nous avons* $\kappa_{\inf} \leq \frac{x^* A_d^{DF}(\kappa)x}{x^* A_d^{DF}(1)x} \leq \kappa_{\max}$*, en particulier cond*$(A_d^{DF}(\kappa)) = O(N^2)_{N\to\infty}$.

1.4 Discrétisation d'une EDP elliptique par éléments finis

Pour résumer la discrétisation en éléments finis (EF), on regardera directement le problème 1.3.2 de diffusion, en commençant par le cas $g = 0$ des conditions au bord homogènes. Un des avantages de la méthode EF est qu'elle s'applique à des domaines polyédrique (une intersection finie de demi-espaces par exemple), et pas seulement à des produits cartesiens. On supposera alors de disposer d'un tel ensemble Ω, avec bord $\partial\Omega$, et on supposera que le coefficient de conductivité κ vérifie la condition (1.2) ce qui assure que les hypothèses du Lemme de Céa sont bien valables.

La première étape dans une discrétisation par éléments finis est de passer à une formulation faible, où on cherche une solution dans un espace de Hilbert V (de fonctions suffisamment différentiables qui s'annulent au bord $\partial\Omega$). En utilisant une formule de Green, on obtient la formulation : chercher $u \in V$ tel que, pour tout $v \in V$,

$$\int_\Omega \kappa(x)\langle \nabla u(x), \nabla v(x)\rangle dx = \int_\Omega f(x)v(x)dx. \tag{1.3}$$

On montre que (1.3) admet une solution unique, qui est aussi une solution de notre problème 1.3.2 de diffusion à condition que la fonction f est suffisamment différentiable.

Pour en déduire une solution approchée et un système d'équations linéaires $A_d^{EF}(\kappa)x = b$ avec n inconnues, on considère des sous-espaces $V_n \subset V$ de dimension n dont on dispose une base $\phi_1, ..., \phi_n$. Ici on cherche une solution approchée de la forme

$$u_n(x) = \sum_{k=1}^{n} c_k \phi_k(x)$$

avec coefficients $c_k \in \mathbb{R}$ inconnus de sorte que, pour $j = 1, ..., n$,

$$\int_\Omega \kappa(x)\langle \nabla u_n(x), \nabla \phi_j(x)\rangle dx = \int_\Omega f(x)\phi_j(x)dx. \tag{1.4}$$

ce qui nous amène au système $A_d^{EF}(\kappa)x = b$ avec le vecteur d'inconnus $x \in \mathbb{R}^n$ comportant les coefficients $c_1, ..., c_n$, la matrice de coefficients (dite *matrice de raideur*) et le second membre étant donnés par :

$$A_d^{EF}(\kappa) = \Big(\int_\Omega \kappa(x)\langle \nabla \phi_k(x), \nabla \phi_j(x)\rangle dx\Big)_{j,k=1,...,n}, \quad b = \Big(\int_\Omega f(x)\phi_j(x)dx\Big)_{j=1,...,n}. \tag{1.5}$$

Voila le principe, mais le travail n'est pas fini : pour une dimension d de l'espace, il faut maintenant choisir V_n et sa base $\phi_1, ..., \phi_n$, faire une étude d'erreur de $u - u_n$, spécifier comment évaluer les intégrales (par des formules de quadrature ce qui introduit d'autres erreurs), écrire le système et le résoudre. Il est peut-être utile de sous-ligner que l'on n'a pas besoin véritablement de la solution exacte du système $A_d^{EF}(\kappa)x = b$ parce que, de toute façon, u_n est seulement une approximation de la solution de notre EDP (cette remarque s'applique aussi aux systèmes venant des différences finies).

1.4.1. Propriétés du problème de diffusion discrétisé par EF :
La matrice $A_d^{EF}(\kappa)$ est symétrique définie positive.

Démonstration. La symétrie est évidente. Soit $x \in \mathbb{R}^n \setminus \{0\}$ alors, en posant $v(x) = \sum_{k=1}^n x_k \phi_k(x)$,

$$x^* A_d^{EF}(\kappa) x = \int_\Omega \kappa(x) \langle \nabla v(x), \nabla v(x) \rangle \, dx \geq \kappa_{\inf} \int_\Omega \|\nabla v(x)\|^2 \, dx.$$

Comme v est un élément non nul de l'espace V, sa semi-norme H^1 de Sobolev s'annule pas, et donc est strictement positive. □

Dans la précédente preuve on a vu que la norme énergie pour la matrix de raideur (comparer avec Exo 1.2.5) est liée à la semi-norme H^1 de Sobolev d'un élément de V. On peut aussi considérer la matrice dite *de masse*

$$M_d^{EF}(\kappa) = \Big(\int_\Omega \kappa(x) \phi_k(x) \phi_j(x) \, dx \Big)_{j,k=1,\dots,n},$$

dont la norme énergie fait le lien avec la norme L^2 (à poids si $\kappa \neq 1$).

On va ici se limiter aux éléments finis de type P_1 en dimension $d = 1$ et $d = 2$. Ici on coupera Ω en un certain nombre de morceaux, et V_n est un espace de fonctions qui sont continues sur la ferméture de Ω, et affines sur chacun des morceaux.

1.4.2. Le problème de diffusion pour $d = 1$ discrétisé par EF :
Dans le cas $\Omega =]0, 1[$ de dimension $d = 1$, on commence par introduire un maillage de Ω, ici des points équidistants $j/(n+1)$, $j = 0, 1, 2, ..., n+1$, et on coupe Ω en intervalles $[j/(n+1), [(j+1)/(n+1)]$ pour $j = 0, ..., n$. Une fonction $v \in V_n$ s'annule en 0 et en 1, est affine sur chacun des intervalles, mais on raccorde les droites de sorte que, globalement, la fonction v est continue sur Ω (on parle des splines affines*). Il n'est pas trop difficile de voir qu'une telle fonction est uniquement déterminée en fonction de ses valeurs (quelconques) aux points intérieurs du maillage. Par conséquent, on peut construire une base $\phi_1, ..., \phi_n$ de V_n à l'aide des fonctions chapeaux : la fonction ϕ_j prend la valeur 1 en $x = j/(n+1)$, et la valeur 0 en tout point $x = k/(n+1)$ pour $k = 0, 1, ..., n+1$, $k \neq j$. Un petit dessin montre que ces fonctions ϕ_j ainsi que leur gradient/dérivée sont non nuls seulement sur l'intervalle $[(j-1)/(n+1), (j+1)/(n+1)]$, et*

$$\nabla \phi_j(x) = \begin{cases} n+1 & \text{sur }](j-1)/(n+1), j/(n+1)[, \\ -n-1 & \text{sur }]j/(n+1), (j+1)/(n+1)[, \end{cases}$$

ce qui permet de vérifier que, pour une conductivité $\kappa = 1$ constante et un maillage équidistant,

$$A_1^{EF}(1) = (n+1) A_1^{DF}(1) = \frac{1}{h} A_1^{DF}(1), \tag{1.6}$$

avec $A_1^{DF}(1) = B$ donné dans 1.3.4. Aussi, il est intéressant d'observer que $\frac{1}{h} A_1^{DF}(\kappa)$ est une approximation de $A_1^{EF}(\kappa)$ où l'intégrale est remplacée par la formule de quadrature du point milieu.

1.4.3. Le problème de diffusion pour $d = 2$ discrétisé par EF :
Ici on coupe notre domaine Ω en triangles, voir par exemple ici. Formellement, une triangulation de Ω consiste à couper Ω en triangles de sorte que deux triangles distincts aient soit une arête en commun, soit un sommet en commun, soit une intersection vide. Par n on note le nombre

de sommets à l'intérieur de Ω de cette triangulation. Une fonction $v \in V_n$ s'annule sur $\partial\Omega$, avec restriction sur chacun des triangles une fonction affine (ce qui fait trois de degrés de liberté), mais on raccorde ces morceaux de sorte que, globalement, la fonction v est continue sur Ω. Il n'est pas trop difficile de voir qu'une telle fonction est uniquement déterminée en fonction de ses valeurs (quelconques) aux sommets intérieurs du maillage. Par conséquent, on peut construire une base $\phi_1, ..., \phi_n$ de V_n à l'aide des fonctions chapeaux : la fonction ϕ_j associé au sommet intérieur d'indice j prend la valeur 1 en ce sommet, et la valeur 0 en tout autre sommet. Encore une fois on observe que ces fonctions ϕ_j ainsi que leur gradient sont non nuls seulement sur la réunion des triangles qui comportent ce sommet. En comparant avec (1.5) on peut conclure qu'un élément à la position (j, k) s'annule pas si j et k sont des sommets d'un seul triangle (autrement dit, les extrémités d'une arête). Sur notre dessin ci-dessus, chaque sommet admet au plus 7 sommets intérieurs adjacents, et donc chaque ligne de $A_2^{EF}(\kappa)$ comporte au plus 8 éléments non nuls (un peu plus que dans la méthode des différences finies).

1.4.4. Exercice :
Construisons une triangulation de $\Omega =]0, 1[^2$ avec comme $n = N^2$ sommets internes le maillage uniforme décrit dans 1.3.6, ou chaque sommet au centre C est connecté par une arête avec les sommets voisin au nord, sud, ouest, est, nord-est et sud-ouest (dessin ?). Il est connu que, dans ce cas, $A_2^{EF}(1) = A_2^{DF}(1)$ (et plus généralement $A_d^{EF}(1) = A_d^{DF}(1)h^{d-2}$). Vérifier pour $N \in \{1, 2, 3\}$.

Une grande partie des propriétés énoncés en 1.3.9 pour la méthode des différences finies reste valable pour les éléments finis. Par exemple, si le plus petit angle dans la triangulation (en fonction de n) ne tend pas vers 0, alors

$$\mathrm{cond}(A_d^{EF}) = \mathcal{O}(n^{2/d})_{n\to\infty}. \tag{1.7}$$

Nous avons aussi le résultat suivant.

1.4.5. Exercice :
Pour une triangulation quelconque et pour tout vecteur $x \neq 0$, montrer que

$$\kappa_{\inf} \leq \frac{x^* A_d^{EF}(\kappa) x}{x^* A_d^{EF}(1) x} \leq \kappa_{\max}.$$

1.4.6. Remarque : Les conditions au bord non homogènes.
On peut aussi utiliser les éléments finis pour résoudre un problème de Dirichlet non homogène. Pour cela on note par ϕ_j pour $j = n+1, ..., n+p$ les fonctions chapeau associés aux sommets j à la position $x_j \in \partial\Omega$, et on utilise l'ansatz

$$u_n = \underbrace{\sum_{k=1}^{n} c_k \phi_k}_{\text{solution faible pour } f \neq 0 \text{ et } g = 0} + \underbrace{\sum_{k=n+1}^{n+p} g(x_k)\phi_k}_{\text{solution faible pour } f = 0 \text{ et } g \neq 0}.$$

1.5 Les projecteurs

Dans ce chapitre on va identifier une application linéaire $B : \mathbb{C}^n \mapsto \mathbb{C}^m$ avec sa représentation matricielle dans $\mathbb{C}^{m\times n}$ (qui dans ses colonnes comporte les images des vecteurs

canoniques). On notera

$$\begin{aligned} Im(B) &= \text{ espace engendré par les colonnes de } B \ = \{By : y \in \mathbb{C}^n\}, \\ Ker(B) &= \text{ le noyau de } B \ = \{y \in \mathbb{C}^n : By = 0\}, \end{aligned}$$

dont on rappelle la somme orthogonale

$$\mathbb{C}^m = Im(B) \oplus Ker(B^*). \tag{1.8}$$

De cette équation on peut déduire que $Im(B) = Ker(B^*)^\perp = \{y \in \mathbb{C}^n : y \perp Ker(B^*)\}$ (l'orthogonal de l'ensemble $Ker(B^*)$), de plus on sait que $dim(Im(B)) = dim(Im(B^*)) = rang(B)$.

On appelle *projecteur (oblique)* une matrice $P \in \mathbb{C}^{n\times n}$ vérifiant $P^2 = P$.

1.5.1. Exemple :
Soient $U, W \in \mathbb{C}^{n\times m}$ avec $W^*U \in \mathbb{C}^{m\times m}$ inversible, alors $P = U(W^*U)^{-1}W^*$ est un projecteur, et $Im(P) = Im(U)$, $Ker(P) = Im(W)^\perp$.

La réciproque de Exemple 1.5.1 est aussi valable, en spécifiant $Im(P)$ et $Ker(P)$ on fixe un projecteur P.

1.5.2. Théorème : projecteurs obliques
Si les colonnes de $U, W \in \mathbb{C}^{n\times m}$ formes des bases de $Im(P)$ et $Ker(P)^\perp$, respectivement, d'un projecteur P alors W^*U est une matrice inversible, et $P = U(W^*U)^{-1}W^*$.

Démonstration. Par définition de U nous avons $\forall x \in \mathbb{C}^n \, \exists z \in \mathbb{C}^m$ t.q. $Px = Uz$. Ceci donne les propriétés

$$\begin{aligned} & \forall x \in \mathbb{C}^n : P(I-P)x = (P - P^2)x = 0 \\ \Longrightarrow \quad & Im(I-P) \subset Ker(P) \\ \Longrightarrow \quad & \forall x \in \mathbb{C}^n : W^*(I-P)x = 0 \quad \text{par déf. de } W \\ \Longrightarrow \quad & \forall x \in \mathbb{C}^n \, \exists z \in \mathbb{C}^m \text{ t.q. } W^*x = W^*Uz. \end{aligned}$$

Comme $Im(W^*) = \mathbb{C}^m$, on en déduit que $\forall a \in \mathbb{C}^m \, \exists z \in \mathbb{C}^m$ avec $a = W^*Uz$, autrement dit, W^*U est inversible, et $z = (W^*U)^{-1}W^*x$. Donc $\forall x \in \mathbb{C}^n$ nous avons que $Px = Uz = U(W^*U)^{-1}W^*x$. □

On dira que le projecteur P est une *projecteur orthogonal* si $Im(P) = Ker(P)^\perp$ (ce qui permet de prendre $U = W$ dans le théorème précédent). Le résultat suivant montre qu'un projecteur orthogonal est une matrice hermitienne (mais généralement pas orthogonale ou unitaire).

1.5.3. Théorème : projecteurs orthogonaux
Pour un projecteur P on a équivalence entre les quatre propriétés suivantes :
(a) *P est un projecteur orthogonal ;*
(b) *$\exists W \in \mathbb{C}^{n\times m}$ à colonnes orthonormées ($W^*W = I$) tel que $P = WW^*$;*
(c) *$\exists (W, \widetilde{W}) \in \mathbb{C}^{n\times n}$ unitaire tel que $P = WW^*$, $I - P = \widetilde{W}\widetilde{W}^*$;*
(d) *$P = P^*$.*

On écrira dans la suite $P = P_W$ pour un projecteur orthogonal.

Démonstration. $\underline{(a) \Longrightarrow (b)}$: Choisir $U = W$ dans le théorème 1.5.2 avec colonnes formant une base orthonormée de $Im(P) = Ker(P)^{\perp}$.
$\underline{(b) \Longrightarrow (c)}$: D'après le théorème de la complétition d'une base, on trouve toujours $\widetilde{W}$ de sorte que les colonnes de $(W, \widetilde{W})$ forment une base de $\mathbb{C}^n$. Par conséquent, $(W, \widetilde{W})$ est une matrice unitaire, et, avec un produit par blocs,

$$I = (W, \widetilde{W})(W, \widetilde{W})^* = WW^* + \widetilde{W}\widetilde{W}^* = P + \widetilde{W}\widetilde{W}^*.$$

$\underline{(c) \Longrightarrow (d)}$: Trivial.
$\underline{(d) \Longrightarrow (a)}$: D'après (1.8), $Im(P) = Ker(P^*)^{\perp} = Ker(P)^{\perp}$, la dernière égalité découlant de $P = P^*$. □

Dans le cas du plan $n = 2$ nous avons seulement des projecteurs non triviaux P_W avec $m = 1$ et alors $W \in \mathbb{C}^{2\times 1}$ représente un vecteur de longueur 1. Ici, la projection Px d'un $x \in \mathbb{R}^2$ est en effet la projection orthogonale de x sur la droite passant par l'origine, de direction W. Tentez de donner une interprétation géométrique similaire pour un projecteur oblique (non orthogonal).

Le théorème de Pythagore dans $\mathbb{R}^2$ relit les longueurs des arêtes du triangle avec sommets $0, x$ et Px. Il est aussi vrai en $\mathbb{C}^n$

1.5.4. Corollaire : Pythagore
Pour un projecteur P orthogonal avec les notations du théorème 1.5.3

$$\forall x \in \mathbb{C}^n : \quad \|x\|^2 = \|Px\|^2 + \|(I-P)x\|^2 = \|W^*x\|^2 + \|\widetilde{W}^*x\|^2,$$

en particulier $\|P\| \leq 1$.

Démonstration. Pour montrer la première égalité il suffit de développer $\|x\|^2 = \|Px + (I-P)x\|^2$ et d'observer que

$$(Px, (I-P)x) = ((I-P)x)^*Px = x^*(I-P)Px = x^*(P-P^2)x = 0.$$

La deuxième égalité vient du fait que $\|Px\|^2 = x^*P^2x = x^*Px = \|W^*x\|^2$. La dernière relation en découle en observant que $\forall x \in \mathbb{C}^n$ nous avons $\|Px\| \leq \|x\|$. □

Dans le corollaire précédent, il a été important de supposer que P est un projecteur orthogonal, car pour tout projecteur $P \neq 0$ nous avons par submultiplicativité de la norme spectrale que $\|P\| = \|P^2\| \leq \|P\|^2$, ce qui implique que $\|P\| \geq 1$.

1.5.5. Exercice :
Soit P un projecteur dans $\mathbb{C}^n$.

1. *Montrer que $I - P$ est aussi un projecteur et que*

$$Ker(P) = Im(I-P)$$
$$\mathbb{C}^n = Im(P) \oplus Im(I-P)$$

2. *Montrer que tout projecteur admet comme valeurs propres* 0 *et* 1 *et en déduire que* $\exists V$ *matrice inversible telle que* $V^{-1}PV$ *est une matrice diagonale (et un projecteur orthogonal).*
3. *Construire* $U \in \mathbb{C}^{r\times n}$ *base de* $Im(P)$ *et* $W \in \mathbb{C}^{r\times n}$ *base de* $Ker(P)^{\perp}$ *avec* $W^*U = I$ *en fonction de la matrice* V *de la question précédente.*
4. *Soit* P *un projecteur orthogonal. Montrer que*

$$\forall x \in \mathbb{C}^n \qquad \min_{y\in Im(P)} \| x - y \| = \| x - Px \| .$$

1.5.6. Exercice :
Soient $W \in \mathbb{C}^{n\times m}$ *à colonnes orthonormées, et* $B \in \mathbb{C}^{m\times r}$. *Montrer que* $\|B\|_F^2 = trace(B^*B)$. *En déduire que*

$$\|WB\| = \|B\|, \quad \|WB\|_F = \|B\|_F.$$

1.5.7. Exercice :
Soient A_1, A_2 *des matrices avec un même nombre de colonnes. Pour la norme de Froebenius on vérifie aisément que*

$$\left\| \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} \right\|_F \geq \left\| \begin{bmatrix} 0 \\ A_2 \end{bmatrix} \right\|_F = \|A_2\|_F .$$

A l'aide d'un projecteur orthogonal à construire, montrer que la même propriété est valable pour la norme spectrale.

1.5.8. Exercice : *Soit* $P \in \mathbb{C}^{n\times n}$ *un projecteur orthogonal, on adapte les notations du 1.5.3.*

(a) *Soit* $A \in \mathbb{C}^{n\times r}$. *Pour la norme de Froebenius, montrer que la relation de Pythagore* $\|A\|_F^2 = \|PA\|_F^2 + \|(I-P)A\|_F^2$ *reste valable, mais que cette relation peut être fausse pour la norme spectrale.*

(b) *Soit* $A \in \mathbb{C}^{n\times n}$. *Montrer que*

$$\arg\min_B \|AW - WB\|_F = W^*AW, \quad \arg\min_B \|AW - WB\| = W^*AW.$$

Motivation : on se demande si $Im(W)$ *est* A*-invariant, c'est-à-dire,* $A\,Im(W) \subset Im(W)$ *(ce qui serait par exemple le cas si* $Im(W)$ *est engendré par certains vecteurs propres de* A*).*

Chapitre 2

Résolution de grands systèmes creux

2.1 Motivation

Le but de ce chapitre est de résoudre (d'une manière approchée) des systèmes d'équations linéaires $Ax = b$ où la matrice de coefficients est supposée être "grande" et "creuse" (et inversible). Ceci signifie que A contient tellement de zéros que, algorithmiquement, il est intéressant d'en tenir compte.

Revenons par exemple à la discrétisation par différences finies du problème de Poisson en dimension $d = 2$, voir 1.3.6. Si on utilise $N = 100$ points de discrétisation par coordonnée (ce qui n'est pas beaucoup, le pas $h = 1/(N + 1)$ devrait être "petit"), on se ramène à un système à $n = N^2 = 10000$ inconnues. Pourtant, chaque ligne de A ne comporte que au plus 5 élements non nuls. Pire encore, les phénomènes physiques nécessitent souvent une simulation dans $\mathbb{R}^3$. Ici $N = 100$ nous amène à un million d'inconnues, ou la matrice admet au plus 7 éléments non nuls par ligne. Ce même phénomène se produit pour les éléments finis : des simulations d'EDF sur le crash d'un avion sur une usine nucléaire nous amènent à des systèmes avec un milliard d'inconnues car on souhaite discrétiser avec une erreur assez petite. Ici on utilise des tetrahèdres dans $\mathbb{R}^3$, mais le nombre d'éléments non nuls dans la ligne d'indice j est borné par le nombre de tetrahèdres ayant comme sommet ce $j$ième sommet, comparer avec 1.4.3.

Dans la suite on se placera dans la situation suivante :

2.1.1. Hypothèse sur nos systèmes.
On supposera que l'entier n est si grand que l'on peut encore stocker et manipuler un petit nombre de vecteurs dans $\mathbb{C}^n$. Par contre, on n'a pas assez de mémoire de stocker la matrice inversible $A \in \mathbb{C}^{n\times n}$ en entier : on dispose seulement d'une boîte noire permettant de

- *calculer pour $y \in \mathbb{C}^n$ le vecteur Ay en complexité $\mathcal{O}(n)$;*
- *calculer le produit scalaire entre $x, y \in \mathbb{C}^n$ en complexité $\mathcal{O}(n)$;*
- saxpy [1] *: calculer pour $x, y \in \mathbb{C}^n$ et $\alpha \in \mathbb{C}$ le vecteur $\alpha x + y$ en complexité $\mathcal{O}(n)$.*

Dans le contexte de notre hypothèse 2.1.1, la notion de complexité correspond au nombre d'opérations élémentaires (additions, soustractions, multiplications et divisions dans $\mathbb{C}$), et au

1. synonyme pour "single precision a x plus y".

besoin de mémoire. Ici on ne parlera pas trop du fonctionnement de ces boîtes noires : on peut s'imaginer de stocker seulement les éléments non nuls de A ainsi que leur position, et on effectue le produit Ay en parcourant seulement les éléments non nuls de A. Cela permet d'arriver à la complexité désirée si A ne contient que $\mathcal{O}(n)$ éléments non nuls (ce qui est le cas chez nous). Il faut savoir qu'il existe des boîtes noires (dites BLAS, basic linear algebra subroutines) où nos trois opérations, très adaptées à un calcul parallèle ou un calcul par coeur/threads, sont optimisées pour chaque architecture d'ordinateur.

Notre hypothèse 2.1.1 sur n et A rend impossible d'appliquer les méthodes vues en L3 comme le calcul explicite de A^{-1} ou les décompositions LU ou QR. Ici on calculera une suite $x_0, x_1, x_2, ..$ de vecteurs dans $\mathbb{C}^n$ avec une erreur $x_m - \overline{x}$ qui on espère tendra rapidement vers 0 pour $m \to \infty$. Ici on note $\overline{x} = A^{-1}b$ la solution exacte de notre système. On écartera les méthodes itératives "classiques" basées sur un un splitting $A = M - N$ avec $\rho(M^{-1}N) < 1$ (comme les méthodes de Jacobi, de Gauss-Seidel, la méthode SOR etc) qui souvent ne convergent pas très rapidement (même si l'analyse de convergence est assez belle).

2.1.2. Définition du résidu.
Comme on ne dispose pas de $\overline{x}$, l'erreur $\overline{x} - x_m$ n'est pas calculable, par contre on peut calculer le résidu

$$r_m = A(\overline{x} - x_m) = b - Ax_m,$$

avec

$$\|\overline{x} - x_m\| \leq \|A^{-1}\| \, \|r_m\|, \quad \|r_m\| \leq \|A\| \, \|\overline{x} - x_m\|.$$

Donc si le conditionnement $cond(A) = \|A\| \, \|A^{-1}\|$ n'est pas trop élevé, l'erreur et le résidu sont du même ordre de grandeur.

2.2 Méthodes de projection

Dans une méthode de projection on part d'un $x_0 \in \mathbb{C}^n$ (que l'on veut une bonne approximation de $\overline{x}$, mais en pratique on prend souvent $x_0 = 0$). À l'étape $m \geq 0$, on détermine deux espaces vectoriels $\mathcal{K}_m, \mathcal{L}_m$ de même dimension $d = d_m$ (qui peut dépendre de m) avec comme bases les colonnes de $V_m, U_m \in \mathbb{C}^{n \times d}$, respectivement, de sorte que

$$x_{m+1} \in x_m + \mathcal{K}_m, \quad r_{m+1} = b - Ax_{m+1} \perp \mathcal{L}_m, \tag{2.1}$$

avec l'espace affine $x + \mathcal{K}_m = \{x + y : y \in \mathcal{K}_m\}$. On corrige alors notre itéré x_m par un élément dans $\mathcal{K}_m$. Notons que x_{m+1} dépend seulement du choix des espaces, mais pas du choix de leur bases. Afin d'assurer l'existence et l'unicité du x_{m+1} dans (2.1), mais aussi pour pouvoir l'exprimer (2.1) en termes de projecteurs, il nous faut supposer que

$$U_m^* A V_m \text{ est inversible,} \tag{2.2}$$

ce qui est vrai si par exemple A est sdp, et $U_m = V_m$.

2.2.1. Lemme : représentation à l'aide de projecteurs.
Nous avons

$$x_{m+1} = x_m + V_m (U_m^* A V_m)^{-1} U_m^* r_m,$$
$$r_{m+1} = (I - Q_m) r_m, \quad Q_m := A V_m (U_m^* A V_m)^{-1} U_m^*.$$

Démonstration. Notons que $x_{m+1} \in x_m + \mathcal{K}_m$ si et seulement si $\exists y \in \mathbb{C}^d$ avec $x_{m+1} = x_m + V_m y$. Donc

$$r_{m+1} = b - Ax_{m+1} = r_m - AV_m y \perp \mathcal{L}_m = Im(U_m)$$

ssi $U_m^*(r_m - AV_m y) = 0$ ssi $y = (U_m^* AV_m)^{-1} U_m^* r_m$. □

Une méthode de projection devient "intéressante" si on choisit $\mathcal{K}_m, \mathcal{L}_m$ de sorte que $U_m^* AV_m$ devient une matrice "simple" (un scalaire, une matrice diagonale, etc). Donnons deux approches pour estimer l'erreur en termes de projecteurs (orthogonaux).

2.2.2. Lemme : estimation d'erreur.
Notons par P_m le projecteur orthogonal sur $\mathcal{K}_m$, alors

$$\|r_{m+1}\| \leq \|(I - Q_m)A(I - P_m)\| \, \|(I - P_m)(\overline{x} - x_m)\|.$$

Interprétation : à l'itération m on cherche à choisir $\mathcal{K}_m$ et donc P_m de sorte que $\|(I - P_m)(\overline{x} - x_m)\|$ soit petit (et l'autre facteur pas trop grand), veut dire $\overline{x}$ pas trop loin de l'espace affine $x_m + \mathcal{K}_m$. En particulier, on en déduit la condition suffisante de terminaison suivante : si $\overline{x} \in x_m + \mathcal{K}_m$ alors $x_{m+1} = \overline{x}$.

Démonstration. D'après le lemme 2.2.1, $r_{m+1} = (1 - Q_m)r_m = (1 - Q_m)A(\overline{x} - x_m)$, avec

$$\begin{aligned}(1 - Q_m)AP_m &= (1 - Q_m)AV_m(V_m^* V_m)^{-1} V_m^* \\ &= \Big(AV_m - AV_m(U_m^* AV_m)^{-1} U_m^* AV_m\Big)(V_m^* V_m)^{-1} V_m^* = 0,\end{aligned}$$

et donc

$$r_{m+1} = (1 - Q_m)A(I - P_m)(\overline{x} - x_m) = (1 - Q_m)A(I - P_m)^2(\overline{x} - x_m),$$

ce qu'il fallait démontrer. □

Dans le deuxième résultat, au lieu de quantifier la taille du résidu, on cherche à borner une norme énergie 1.2.5 de l'erreur.

2.2.3. Théorème : estimation optimale.
Si $\exists M \in \mathbb{C}^{n \times n}$ avec $M^* A$ sdp, et si $\mathcal{L}_m = M\mathcal{K}_m$ alors notre condition (2.2) *est valable, et*

$$\|\overline{x} - x_{m+1}\|_{M^*A} = \min\{\|\overline{x} - x\|_{M^*A} : x \in x_m + \mathcal{K}_m\}.$$

Interprétation :

- si $M^* A = I$ on minimise la taille de l'erreur (choix idéal mais théorique) ;
- si $M = A$ on minimise la taille du résidu ;
- si A sdp et $M = I$ on minimise la taille de l'erreur en norme énergie.

Démonstration. Dans un premier temps, observons que les formules du lemme 2.2.1 ne dépendent pas du choix des bases U_m, V_m, on peut alors prendre $U_m = MV_m$ et alors $U_m^* AV_m = V_m^*(M^* A)V_m$ qui est sdp (car $M^* A$ l'est par hypothèse, et les colonnes de V_m sont libres). En particulier, $U_m^* AV_m$ est inversible ce qui nous donne la condition (2.2).

Ensuite, introduisons la décomposition de Choleski $M^*A = C^*C$ et le projecteur orthogonal $\widetilde{P}$ sur $Im(CV_m)$

$$\widetilde{P} = CV_m(V_m^*C^*CV_m)^{-1}V_m^*C^*,$$

et donc $\widetilde{P}CV_m = CV_m$.

Posons $y_m = (U_m^*AV_m)^{-1}U_m^*r_m$ de sorte que $x_{m+1} = x_m + V_m y_m$. Nous avons $x \in x_m + \mathcal{K}_m$ un candidat à la minimisation si et seulement si $\exists y \in \mathbb{C}^m$ avec $x = x_m + V_m y$. Donc, par Pythagore,

$$\begin{aligned} \|\overline{x} - x\|_{M^*A} &= (\overline{x} - x)^* M^* A(\overline{x} - x) = \|C(\overline{x} - x)\|^2 \\ &= \|C(\overline{x} - x_m) - CV_m y\|^2 \\ &\geq \|(I - \widetilde{P})C(\overline{x} - x_m) - (I - \widetilde{P})CV_m y\|^2 = \|(I - \widetilde{P})C(\overline{x} - x_m)\|^2, \end{aligned}$$

avec égalité si et seulement si

$$\begin{aligned} & 0 = \widetilde{P}C(\overline{x} - x_m) - \widetilde{P}CV_m y = \widetilde{P}CA^{-1}r_m - CV_m y \\ \iff \quad & CV_m(V_m^*C^*CV_m)^{-1}V_m^*C^*CA^{-1}r_m = CV_m y \\ \iff \quad & CV_m(U_m^*AV_m)^{-1}U_m^*AA^{-1}r_m = CV_m y \qquad \text{car } V_m^*C^*C = U_m^*A \\ \iff \quad & CV_m y_m = CV_m y. \end{aligned}$$

Comme les colonnes de CV_m et de V_m sont libres, cette dernière propriété est valable ssi $y = y_m$ ssi $x = x_{m+1}$. On vient donc de montrer que le minimum est atteint pour $x = x_{m+1}$ (et que ce minimiseur est unique). □

2.2.4. Remarque sur la minimisation du résidu.
Dans le théorème 2.2.3 avec $M = A$ nous avons $x_{m+1} = x_m + V_m y_m$ avec $(V_m^*A^*AV_m)y_m = V_m^*A^*r_m$ (les équations normales d'un problème aux moindres carrés), ce qui équivaut au fait que y_m est solution du problème des moindres carrés

$$\min_{y \in \mathbb{C}^d} \|AV_m y - r_m\|$$

à n équations et $d = d_m$ inconnues.

Dans la suite de ce document on donnera plusieurs exemples de méthodes de projection, d'autres exemples seront étudiés dans les exercices.

2.3 La méthode de la plus forte descente

Dans la méthode de la plus forte descente (en anglais "steepest descent", aussi parfois appelé la méthode de Cauchy) on suppose que A est une matrice sdp, et que $A, b, \overline{x}$ sont à coefficients réels. Par conséquent, on peut appliquer le formalisme du théorème 2.2.3 avec $M = I$ et $\mathcal{K}_m = \mathcal{L}_m$ de dimension 1, c'est-à-dire, on souhaite minimiser la forme quadratique

$$J(x) = \|\overline{x} - x\|_A = (\overline{x} - x)^T A(\overline{x} - x) = \overline{x}A\overline{x} - 2\overline{x}Ax + x^T Ax$$

sur l'ensemble des vecteurs de de la forme $x = x_m + \alpha v$ avec $\alpha \in \mathbb{R}$ pour un vecteur $v \in \mathbb{R}^n$ fixé. Pour choisir v on remarque que, pour $\alpha \in \mathbb{R}$ "petit",

$$\frac{J(x_m + \alpha v) - J(x_m)}{\alpha} \approx \nabla J(x_m) v$$

(chez nous, le gradient est un vecteur ligne), le second membre étant minimum parmi toutes les directions v de longueur fixe ssi

$$v = -\nabla J(x_m)^T = 2r_m$$

(il est fortement conseillé de vérifier cette dernière égalité). Le choix $\mathcal{K}_m = \mathcal{L}_m = \text{span}(r_m)$ dans 2.2.1 nous donne l'algorithme suivant.

2.3.1. Algorithme de la plus forte descente.

Choisir $x_0 \in \mathbb{R}^n$ et calculer
Pour $m = 0, 1, 2, \ldots$ jusqu'à $\|r_m\|$ "petit"
$\alpha_m = \frac{(r_m, r_m)}{(Ar_m, r_m)}$, $x_{m+1} = x_m + \alpha_m r_m$, $r_{m+1} = r_m - \alpha_m A r_m$
Besoins de stockage : trois vecteurs (on stocke sur place).
Complexité par itération : 2 produits scalaires, 2 saxpies, 1 produit matrice-vecteur.

Au lieu d'utiliser le formalisme général 2.2.1 des méthodes de projection, on aurait pu aussi minimiser à la main la fonction $\mathbb{R} \ni \alpha \mapsto J(x_m + \alpha r_m) \in \mathbb{R}$, une parabole ayant un minimum unique.

Interprétation géométrique : à l'itération m, on se trouve à l'endroit x_m sur la courbe de niveau $\{x \in \mathbb{R}^n : J(x) = J(x_m)\}$ (une ellipse dans $\mathbb{R}^2$, une ellipsoïide dans $\mathbb{R}^3$). Pour joindre la vallée (à la position $\overline{x}$) on décide de partir sur une demi-droite de direction la plus raide r_m orthogonal à cette courbe de niveau, et on s'arrête à x_{m+1}, le point le plus bas sur cette demi-droite, ce qui veut dire que r_m est tangent en x_{m+1} à la courbe de niveau $\{x \in \mathbb{R}^n : J(x) = J(x_{m+1})\}$ (faites des dessins avec des ellipses allongées). On observe alors que $r_{m+1} \perp r_m$ (à vérifier), ce qui géométriquement signifié que l'on fait des zigzags, manifestement une stratégie qui n'a pas l'air d'être la meilleure. On aurait envie de partir directement dans la direction de la vallée $\overline{x}$, mais le problème est que l'on ne sait pas où cette vallée se trouve.

2.3.2. Théorème : Estimation a priori du taux de convergence. *Posons $\kappa = cond(A)$, alors pour tout $m \geq 0$*

$$\|\overline{x} - x_{m+1}\|_A \leq \frac{\kappa - 1}{\kappa + 1} \|\overline{x} - x_m\|_A$$
$$\|\overline{x} - x_m\|_A \leq \Big(\frac{\kappa - 1}{\kappa + 1}\Big)^m \|\overline{x} - x_0\|_A \to 0 \quad si\ m \to \infty.$$

Démonstration. On observe d'abord que

$$J(x_{m+1}) = \|\overline{x} - x_{m+1}\|_A^2 = \|r_{m+1}\|_{A^{-1}}^2 = (r_{m+1}, A^{-1} r_{m+1});$$

et que $r_{m+1} \perp r_m$. Donc

$$\begin{aligned} J(x_{m+1}) &= (r_{m+1}, A^{-1} r_m - \alpha_m r_m) = (r_{m+1}, A^{-1} r_m) \\ &= (r_m - \alpha_m A r_m, A^{-1} r_m) = J(x_m) - \alpha_m (r_m, r_m) \\ &= J(x_m)\Big(1 - \frac{(r_m, r_m)}{(A^{-1} r_m, r_m)} \frac{(r_m, r_m)}{(Ar_m, r_m)}\Big) \leq J(x_m)\Big(\frac{\kappa - 1}{\kappa + 1}\Big)^2, \end{aligned}$$

où dans la dernière inégalité on a appliqué l'inégalité de Kantorovitch vue dans l'exo 1.2.4. □

Comment se servir d'une telle estimation a priori ? Le théorème 2.3.2 nous dit que si $\mathrm{cond}(A) = 1$ alors, pour tout x_0, on trouve la solution exacte $x_1 = \overline{x}$ en une itération (cette observation devient beaucoup moins étonnante si on cherche des matrices sdp avec un conditionnement égal à 1, c'est forcement l'identité). Plus sérieusement, si on dispose d'une matrice sdp A avec un conditionnement qui ne dépend pas de n, alors pour tout $\epsilon > 0$ et tout choix initial x_0, on peut trouver un m indépendant de n de sorte que $\|\overline{x} - x_m\| \leq \epsilon$, ce qui veut dire que l'on trouve la solution à une précision prescrite en complexité $O(n)$. Encore une fois un résultat étonnant, car on a une complexité proportionnelle au nombre d'inconnues, mais la constante de proportionnalité (qui dépend de κ, x_0 et ϵ) peut être grande.

Nos matrices $A \in \mathbb{R}^{n\times n}$ venant d'une discrétisation par différences finies ou éléments finis en dimension d sont bien sdp (voir 1.3.5, 1.3.7, 1.4.1) et sont bien creuses, mais ils ont un conditionnement d'ordre $c_1 n^{2/d}$ avec une constante c_1, voir 1.3.9 et (1.7). En prenant $m = \lfloor c_2 n^{2/d} \rfloor$ avec une constante $c_2 > c_1$, nous trouvons que

$$\Big(\frac{\kappa-1}{\kappa+1}\Big)^m \|\overline{x} - x_0\|_A \approx \exp(-\frac{2c_2}{c_1}) \|\overline{x} - x_0\|_A$$

ce qui devient petit si $c_2 \gg c_1$. Par conséquent, on doit s'attendre à une complexité $O(n^{1+2/d})$, donc $\mathcal{O}(n^3)$ pour $d = 1$ (comme l'algo de Gauss pour une matrice pleine) mais $\mathcal{O}(n^2)$ pour $d = 2$ et $\mathcal{O}(n^{5/3})$ pour $d = 3$, on y gagne.

La complexité devient plus favorable pour des méthodes où on dispose d'une estimation d'erreur où κ est remplacé par $\sqrt{\kappa}$. Ceci est le but du chapitre suivant.

2.3.3. Exercice. *On cherche à résoudre le système linéaire $Ax = b$ en effectuant une projection sur $K = x_k + span(v_k)$ où $v_k = A^t r_k$, orthogonalement à $L = span(Av_k)$.*

1. *Déterminer x_{k+1} et r_{k+1}.*
2. *Ecrire l'algorithme correspondant.*
3. *Posons $f(x) = \| b - Ax \|^2$. Montrer que déterminer x_{k+1} par cette méthode est équivalent à calculer x_{k+1} par la formule*

$$x_k + \zeta_k \nabla f(x_k)^T, \quad \zeta_k = \arg\min\{f(x_k + \zeta \nabla f(x_k)^T) : \zeta \in \mathbb{R}\}$$

(partant de x_k, on minimise f sur la droite de la plus forte descente de f).

4. *Montrer que déterminer x_{k+1} par cette méthode est équivalent à appliquer la méthode de la plus forte pente au système linéaire $A^tAx = A^tb$ (équations normales).*

2.3.4. Exercice. *Soit $A \in \mathbb{R}^{n\times n}$. Nous rappelons qu'une méthode de projection pour résoudre $Ax = b$ consiste à chaque itération à trouver x_{k+1} vérifiant :*

$$\begin{cases} x_{k+1} \in x_k + \mathcal{K}_k \\ b - A\tilde{x} \perp \mathcal{L}_k \end{cases} \quad \textit{(projection sur } \mathcal{K}_k \textit{ orthogonalement à } \mathcal{L}_k\textit{)}$$

1. *Supposons A symétrique définie positive. On choisit*

$$\mathcal{K}_k = \mathcal{L}_k = span\{r_k, Ar_k\}.$$

(a) *Construire une base A-orthogonale pour* $\mathcal{K}_k$.

(b) *Calculer* x_{k+1} *et* r_{k+1}. *Expliciter l'algorithme*

(c) *Montrer que*

$$\| x^* - x_{k+1} \|_A \leq \| x^* - x_k \|_A \left(1 - \frac{(r_k, r_k)}{(Ar_k, r_k)} \frac{(r_k, r_k)}{(r_k, A^{-1}r_k)}\right)$$

et en déduire la convergence.

2. *Supposons A définie positive. On applique la méthode de projection avec*

$$\mathcal{K}_k = span\{r_k, Ar_k\}, \mathcal{L}_k = A\mathcal{K}_k.$$

(a) *Construire une base* $\{r_k, p_k\}$ *de* $\mathcal{K}_k$ *telle que* $(Ar_k, Ap_k) = 0$.

(b) *Donner l'itéré d'ordre* $k+1$ *et le résidu correspondant.*

(c) *Etudier sa convergence.*

2.3.5. Exercice. *On considère le système linéaire* $Ax = b$ *avec A symétrique définie positive.*

1. *On considère la suite de projections sur un espace de dimension 1 avec* $K = L = span$ (e_i) *où la suite des indices i est arbitraire. Soit* x_+ *le nouvel itéré après une étape de projection à partir de x. On note :*

$$r = b - Ax, d = A^{-1}b - x, d_+ = A^{-1}b - x_+.$$

Montrer que

$$(Ad_+, d_+) = (Ad, d) - (r, e_i)^2/a_{ii}.$$

Cette égalité prouve-t-elle la convergence de la méthode ?

2. *Si on prend les vecteurs* e_i *dans l'ordre* $e_1, e_2, \cdots, e_n, e_1, e_2, \cdots$ *quelle méthode retrouve-t-on ?*

3. *Supposons maintenant que i est choisi à chaque étape de la méthode de projection tel que :*

$$|r_i| = \max_j |r_j|.$$

Montrer que :

$$\| d_+ \|_A \leq \left(1 - \frac{1}{n\,cond(A)}\right)^{1/2} \| d \|_A .$$

(Suggestion : utiliser l'inégalité $\left|e_i^T r\right| \geq n^{-1/2} \| r \|_2$*).*
Cette inégalité prouve-t-elle la convergence de la méthode ?

2.3.6. Exercice. *Soit K un sous–espace tel que* $AK \subset K$, *avec une base donnée par les colonnes de V. On considère une étape d'une méthode de projection*

$$x_1 \in x_0 + K, \quad r_1 \perp L$$

dont on suppose qu'elle est bien définie. Notre but est de démontrer que si $r_0 \in K$ *alors* $x_1 = \overline{x}$ *est la solution du système* $Ax = b$.

(a) *Vérifier qu'il existe une matrice B telle que* $AV = VB$. *En faisant le lien avec* U^*AV, *déduire que B est inversible.*

(b) *En déduire que* $\overline{x} - x_0 = A^{-1}r_0 \in K$, *et que* $x_1 = \overline{x}$.

2.4 Le gradient conjugué

Soit A une matrice sdp, et A, b à éléments réels. La méthode du gradient conjugué (Hestenes & Stiefel, 1952), aussi dite "conjugate gradient" ou CG, est une méthode de projection avec $\mathcal{K}_m = \mathcal{L}_m = \text{span}(p_m)$ de dimension 1, avec des vecteurs p_m bien choisis :

2.4.1. Définition : *Des vecteurs $p_0, p_1, ..., p_m$ sont dits conjugués ou A-orthogonaux si, pour $0 \leq j, k \leq m$*

$$(Ap_j, p_k) \begin{cases} = 0 & si\ j \neq k \\ \neq 0 & si\ j = k. \end{cases}$$

Un telle famille de vecteurs $\{p_0, ..., p_m\}$ peut être produite par une procédure de type Gram-Schmidt partant de la famille des vecteurs $\{r_0, ..., r_m\}$

$$p_m = r_m - \sum_{k=0}^{m-1} p_j \frac{(Ar_m, p_j)}{(Ap_j, p_j)} \tag{2.3}$$

(et donc $p_0 = r_0$), bien que l'on sache pas encore si $p_0, ..., p_m \neq 0$. Voici l'intérêt des directions conjuguées.

2.4.2. Théorème. *Soient $p_0, p_1, ..., p_m$ des directions conjuguées. Pour $\ell = 0, ..., m$ soit $x_{\ell+1}$ et son résidu $r_{\ell+1}$ calculé par une méthode de projection pour $\mathcal{K}_\ell = \mathcal{L}_\ell = span(p_0, p_1, ..., p_\ell)$. Alors les formules 2.2.1 de mise à jour des itérés x_ℓ et des résidus r_ℓ sont les mêmes que pour le cas $\mathcal{L}_\ell = \mathcal{K}_\ell = span(p_\ell)$.*

Démonstration. Par récurrence sur ℓ. Le cas $\ell = 0$ est trivial, cherchons à montrer la propriété pour $\ell = m$. Notons d'abord que $p_0, ..., p_m$ directions conjugués implique que ces vecteurs sont libres (donner une preuve). Par conséquent, les colonnes de $V_m = U_m = (p_0, p_1, ..., p_m) \in \mathbb{C}^{n\times(m+1)}$ forment une base de l'espace $\mathcal{L}_m = \mathcal{K}_m = \text{span}(p_0, p_1, ..., p_m)$, et $U_m^* A V_m = \text{diag}((Ap_0, p_0), ..., (Ap_m, p_m))$ est une matrice diagonale et inversible. Par hypothèse de récurrence $\ell = m - 1$ et construction, le résidu r_m est orthogonal à p_j pour $j < m$, et alors

$$\begin{aligned} x_{m+1} &= x_m + V_m (U_m^* A V_m)^{-1} U_m^* r_m \\ &= x_m + V_m \begin{bmatrix} \frac{1}{(Ap_0,p_0)} & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \frac{1}{(Ap_m,p_m)} \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ (r_m, p_m) \end{bmatrix} = x_m + \frac{(r_m, p_m)}{(Ap_m, p_m)} p_m, \end{aligned}$$

la formule de mise à jour du cas $\mathcal{L}_m = \mathcal{K}_m = \text{span}(p_m)$. □

Formule (2.3) donne la (fausse) impression qu'il faut stocker $p_0, ..., p_{m-1}$ pour calculer p_m. Pour obtenir une version simplifiée de mise à jour, montrons les deux résultats suivants.

2.4.3. Lemme. *Soient $r_0, r_1, ..., r_m \neq 0$, alors*
(a) *$p_0, p_1, ..., p_m \neq 0$;*
(b) *$span(p_0, ..., p_m) = span(r_0, ..., r_m)$, de dimension $m + 1$;*
(c) *$r_{m+1} \perp r_0, r_1, ..., r_m$.*

Démonstration. Par récurrence sur m. Pour $m = 0$ les propriétés (a) et (b) sont triviales car $p_0 = r_0$, et la propriété (c) provient de (b) et 2.4.2. Supposons maintenant (a)-(c) valable pour les indices $0, 1, ..., m-1$, et montrons ces propriétés pour l'indice m. L'hypothèse de récurrence (c) et l'hypothèse $r_m \neq 0$ montrent que $r_0, ..., r_m$ sont libres. Par construction (2.3) et hypothèse de récurrence (b) nous avons

$$p_m - r_m \in \text{span}(p_0, ..., p_{m-1}) = \text{span}(r_0, ..., r_{m-1})$$

d'où $p_m \neq 0$, et (a) et (b) sont valables. Finalement, (c) découle du choix de $\mathcal{L}_m$ dans la définition (2.1), et de (b). □

2.4.4. Lemme. *Soient $r_0, r_1, ..., r_m \neq 0$, alors $\alpha_m = (r_m, r_m)/(Ap_m, p_m) \neq 0$, et $p_m = r_m + \beta_{m-1} p_{m-1}$, avec $\beta_{m-1} = (r_m, r_m)/(r_{m-1}, r_{m-1})$.*

Démonstration.

$$\begin{aligned} \alpha_m(Ap_m, p_m) &= (r_m, p_m) \qquad \text{par 2.4.2} \\ &= (r_m, r_m) - \sum_{k=0}^{m-1} (p_j, r_m) \frac{(Ar_m, p_j)}{(Ap_j, p_j)} \qquad \text{par (2.3)} \\ &= (r_m, r_m) \neq 0 \qquad \text{par 2.4.3(b),(c) et hypothèse.} \end{aligned}$$

Aussi, pour $j < m$,

$$\begin{aligned} \frac{(Ar_m, p_j)}{(Ap_j, p_j)} &= \frac{(r_m, Ap_j)}{(Ap_j, p_j)} = -\frac{1}{\alpha_j} \frac{(r_m, r_{j+1} - r_j)}{(Ap_j, p_j)} \qquad \text{par 2.4.2} \\ &= -\delta_{j+1,m} \frac{(r_m, r_m)}{(r_j, r_j)} = -\delta_{j+1,m} \beta_{m-1} \qquad \text{par 2.4.3(c)} \end{aligned}$$

et donc $p_m = r_m + \beta_{m-1} p_{m-1}$ par (2.3). □

2.4.5. Algorithme CG.

Choisir $x_0 \in \mathbb{R}^n$, $tol > 0$ et calculer $p_0 = r_0 = b - Ax_0$, $\rho_0 = (r_0, r_0)$
Pour $m = 0, 1, 2, ...$
$\alpha_m = \frac{\rho_m}{(Ap_m, p_m)}$, $x_{m+1} = x_m + \alpha_m p_m$, $r_{m+1} = r_m - \alpha_m A p_m$
$\rho_{m+1} = (r_{m+1}, r_{m+1})$
Arrêt si $\rho_{m+1} \leq \rho_0\, tol^2$
$\beta_m = \frac{\rho_{m+1}}{\rho_m}$, $p_{m+1} = r_{m+1} + \beta_m p_m$.
Besoins de stockage : quatre vecteurs (on stocke sur place).
Complexité par itération : 2 produits scalaires, 3 saxpies, 1 produit matrice-vecteur.

On se rend compte qu'une itération de l'algorithme 2.4.5 du gradient conjugué ne coûte pas bien plus chère qu'une itération de l'algorithme 2.3.1 du steepest descent. En effet, CG est préférable pour des raisons suivantes :

— au moins sans les erreurs d'arrondi, l'algorithme CG s'arrête avec un indice $m < n$ avec $r_{m+1} = 0$ (ce qui équivaut à $x_{m+1} = \overline{x}$) [2] ;

2. Par conséquent, si on ignore les erreurs d'arrondi, alors pour les problèmes de discrétisation de nos EDP on obtient une complexité d'au plus $\mathcal{O}(n^2)$, mais on saura encore améliorer cette borne.

— on obtient une estimation d'erreur a priori pour CG qui est plus intéressante que celle de steepest descent ;
— les experiences numériques confirment que CG a bien souvent besoin de moins d'itérations que steepest descent, au moins pour les systèmes qui nous intéressent.

Le but du reste du chapitre est de fournir des preuves pour ces premières deux propriétés. La première est facile à démontrer : si $r_0, ..., r_m$ sont $\neq 0$, alors $r_0, ..., r_m \in \mathbb{C}^n$ sont libres par le lemme 2.4.3(c), et donc forcément $m + 1 \leq n$. Notons que sur ordinateur on a des erreurs d'arrondi, et donc les résidus r_m "récursives" peuvent être différent des "vrais" résidus $b - Ax_m$. Aussi, pour une tolérance *tol* trop petite, on risque d'avoir des boucles infinies dans nos deux algorithmes 2.4.5 et 2.3.1 : il vaut mieux introduire un compteur d'itérations que l'on limitera, et un drapeau indiquant si on n'a pas atteint la tolérance désirée. Chercher des estimations d'erreur a priori en présence des erreurs d'arrondi est un sujet actuel de recherche, dans la suite on va négliger ces erreurs.

Pour une matrice A d'ordre n et un vecteur $c \in \mathbb{C}^n$, on introduit l'espace de Krylov

$$\mathcal{K}_m(A, c) = \text{span}(c, Ac, ..., A^{m-1}c), \tag{2.4}$$

qui est de dimension $\leq m$.

2.4.6. Lemme. *Soient $r_0, r_1, ..., r_m \neq 0$, alors*

$$span(r_0, ..., r_m) = \mathcal{K}_{m+1}(A, r_0).$$

Démonstration. Par récurrence sur m, le cas $m = 0$ est trivial. Soit $m > 0$, alors le lemme 2.4.3(c) nous affirme que $r_0, ..., r_m$ sont libres. Donc pour affirmer l'égalité, il suffit de montrer que $\text{span}(r_0, ..., r_m) \subset \mathcal{K}_{m+1}(A, r_0)$. Par hypothèse de récurrence et le lemme 2.4.3(c), $r_{m-1}, p_{m-1} \in \mathcal{K}_m(A, r_0)$. Par conséquent, $Ap_{m-1} \in A\mathcal{K}_m(A, r_0) \subset \mathcal{K}_{m+1}(A, r_0)$, et alors $r_m = r_{m-1} - \alpha_{m-1}Ap_{m-1} \in \mathcal{K}_{m+1}(A, r_0)$. □

2.4.7. Théorème : Estimation a priori du taux de convergence. *Posons $\kappa = cond(A)$. Si $r_0, ..., r_{m-1} \neq 0$ alors*

$$\|\overline{x} - x_m\|_A \leq 2\left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}\right)^m \|\overline{x} - x_0\|_A.$$

Démonstration. Dans la première partie de la preuve nous souhaitons montrer que, pour tout polynôme p de degré $\leq m$ avec $p(0) \neq 0$,

$$\|\overline{x} - x_m\|_A \leq \|\overline{x} - x_0\|_A \max\Big\{\frac{|p(\lambda)|}{|p(0)|} : \lambda \text{ valeur propre de } A\Big\}. \tag{2.5}$$

Soit alors p comme dans (2.5), et $q(x) = \frac{p(x)-p(0)}{x-0}$, un polynôme de degré $\leq m - 1$ et alors $q(A)r_0 \in \mathcal{K}_m(A, r_0)$. Les formules des lemme 2.4.4, 2.4.3(b) et 2.4.6 montrent que

$$x_{m-1} - x_0 \in \text{span}(p_0, .., p_{m-1}) = \mathcal{K}_m(A, r_0).$$

Donc

$$\begin{aligned} x &= \overline{x} - \frac{p(A)}{p(0)}A^{-1}r_0 = \overline{x} - \frac{p(A) - p(0)I}{p(0)}A^{-1}r_0 - A^{-1}r_0 \\ &= x_0 - \frac{q(A)}{p(0)} \in x_0 + \mathcal{K}_m(A, r_0) = x_{m-1} + \text{span}(p_0, .., p_{m-1}). \end{aligned}$$

Par conséquent, le théorème 2.2.3 avec $m+1$ remplacé par m et $M = I$ nous permet d'affirmer que

$$\|\overline{x} - x_m\|_A^2 \leq \|\overline{x} - x\|_A^2 = r_0^* \frac{p(A)^*}{p(0)^*} A^{-1} \frac{p(A)}{p(0)} r_0.$$

Notons par $\lambda_n \geq ... \geq \lambda_1 > 0$ les valeurs propres de A, et par $v_1, ..., v_n$ la base orthonormée des vecteurs propres associés, alors en écrivant r_0 dans cette base, nous obtenons

$$r_0 = \sum_{j=1}^{n} \beta_j v_j, \quad \|\overline{x} - x_0\|_A^2 = r_0^* A^{-1} r_0 = \sum_{j=1}^{n} \frac{|\beta_j|^2}{\lambda_j},$$

et finalement

$$\|\overline{x} - x\|_A^2 = \sum_{j=1}^{n} \frac{|\beta_j|^2}{\lambda_j} \frac{|p(\lambda_j)|^2}{|p(0)|^2} \leq \|\overline{x} - x_0\|_A^2 \left(\max_k \frac{|p(\lambda_k)|}{|p(0)|}\right)^2,$$

ce qui donne (2.5).

Il reste à construire un "bon" polynôme p, où on fera appel à des polynômes de Chebyshev (que vous avez peut-être déjà vu). Posons

$$\frac{1}{2}(w + \frac{1}{w}) = \frac{\lambda_n + \lambda_1 - 2y}{\lambda_n - \lambda_1}$$

et

$$p(y) = T_m(\frac{\lambda_n + \lambda_1 - 2y}{\lambda_n - \lambda_1}) := \frac{1}{2}(w^m + \frac{1}{w^m}),$$

alors on peut montrer que les T_m vérifient une récurrence à trois termes, ce qui permet d'affirmer que T_m et alors p est un polynôme de degré $\leq m$.

Si $y \in [\lambda_1, \lambda_n]$ alors $\frac{1}{2}(w + \frac{1}{w}) = \frac{\lambda_n + \lambda_1 - 2y}{\lambda_n - \lambda_1} \in [-1, 1]$, avec solutions $w \in \mathbb{C}$, $|w| = 1$, et alors $|p(y)| \leq |\frac{1}{2}(w^n + \frac{1}{w^n})| \leq 1$.

Si par contre $y = 0$, alors l'équation $\frac{1}{2}(w + \frac{1}{w}) = \frac{\lambda_n + \lambda_1 - 0}{\lambda_n - \lambda_1} = \frac{\kappa+1}{\kappa-1} > 1$ admet une seule solution $w = \frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1} > 1$, et alors $p(0) = \frac{1}{2}(w^n + \frac{1}{w^n}) \geq w^n/2$, ce qui ensemble avec (2.5) nous donne l'estimation désirée. □

En comparant avec la discussion après le théorème 2.3.2 portant sur la complexité de steepest descent, on obtient une meilleurs constante (cachée) pour la complexité $\mathcal{O}(n)$ si le conditionnement de A ne dépend pas de n. Par contre, pour nos matrices $A \in \mathbb{R}^{n \times n}$ venant d'une discrétisation par différences finies ou éléments finis en dimension d, on a un véritable gain : ici $\kappa = c_1 n^{2/d}$, $c_1 > 0$ une constante. En prenant $m = \lfloor \sqrt{c_2 n^{2/d}} \rfloor$ avec une constante $c_2 \gg c_1$, nous trouvons que

$$\left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}\right)^m \|\overline{x} - x_0\|_A \approx \exp(-2\sqrt{\frac{c_2}{c_1}}) \|\overline{x} - x_0\|_A$$

est "petit". Autrement dit, on doit s'attendre à une complexité $\mathcal{O}(n^{1+1/d})$, donc $\mathcal{O}(n^2)$ pour $d = 1$ (ce qui n'est pas encore la complexité idéale pour des matrices qui pourraient être tridiagonales), mais $\mathcal{O}(n^{3/2})$ pour $d = 2$ et $\mathcal{O}(n^{4/3})$ pour $d = 3$, on y gagne en grandes dimensions.

Terminons par une remarque sur le choix optimum d'un polynôme p dans (2.5). Bien souvent on n'observe pas un comportement $\|\overline{x} - x_m\|_A \approx C \cdot \rho^m$ avec $C > 0, \rho \in (0,1)$ indépendant de m dit de convergence linéaire, mais plutôt une convergence super-linéaire ou le taux ρ décroît avec m. Dans notre preuve du théorème 2.4.7, nous avons construit un polynôme p en fonction des bornes du plus petit intervalle comportant toutes les valeurs propres de A, ce qui donne une borne assurant au moins convergence linéaire. En fait, des meilleurs choix de p tiennent compte de la répartition des valeurs propres. Par exemple, une valeur propre isolée bien plus grande que toutes les autres valeurs propres (on parle d'un outlier) n'influence pas vraiment le taux de convergence ρ mais plutôt C, la constante multiplicative. [3] Donner des estimations d'erreurs a priori dépendant de la répartition des valeurs propres de A reste encore a ce jour un sujet de recherche sur lequel des thèses sont soutenus, voir article1 et article2.

2.4.8. Exercice. *Soit A une matrice d'ordre N symétrique définie positive, de valeurs propres $\lambda_1, \lambda_2,$ à laquelle on applique la méthode du gradient conjugué à partir de x_0.*

1. *Montrer que pour tout polynôme Q_{m-1} de degré $m-1$ on a*

$$\| x_m - \overline{x} \|_A^2 \leq \max_{1 \leq i \leq N} (1 - \lambda_i Q_{m-1}(\lambda_i))^2 \| x_0 - \overline{x} \|_A^2,$$

   avec $\overline{x}$ la solution du système à résoudre $Ax = b$.

2. *Montrer que si A ne ne possède que $p < N$ valeurs propres distinctes, alors la méthode du gradient conjugué converge en au plus p itérations.*

2.4.9. Exercice. *Soient (x_k) la suite des itérés obtenus par l'application de la méthode du gradient conjugué à la résolution de $Ax = b$ avec A symétrique définie positive, et (r_k) la suite des résidus correspondants. On pose $q_i = r_i / \| r_i \|, i = 0, \cdots, k-1$ et Q_k la matrice $(N \times k)$ dont la j-ème colonne est le vecteur q_{j-1}.*

1. *Montrer que Q_k est orthogonale.*

2. *Montrer que x_k peut s'écrire sous la forme*

$$x_k = x_0 + Q_k (Q_k^t A Q_k)^{-1} Q_k^t r_0$$

   (la matrice $A_k = Q_k^t A Q_k$ d'ordre k est la restriction de la projection de A sur $K_k(A, r_0)$.

3. *En utilisant les relations de récurrence de la méthode du gradient conjugué :*

$$\begin{cases} r_j &= p_j - \beta_{j-1} p_{j-1} \\ r_{j+1} &= r_j - \alpha_j A p_j \end{cases}$$

   montrer que les q_j vérifient la relation :

$$A q_j = -\frac{\sqrt{\beta_j}}{\alpha_j} q_{j+1} + \left(\frac{1}{\alpha_j} + \frac{\beta_{j-1}}{\alpha_{j-1}} \right) q_j - \frac{\sqrt{\beta_{j-1}}}{\alpha_{j-1}} q_{j-1}.$$

4. *Montrer que sous forme matricielle on obtient*

$$A Q_k = Q_k T_k - \frac{\sqrt{\beta_{k-1}}}{\alpha_{k-1}} q_k e_k^t$$

   en explicitant T_k. Montrer que T_k est définie positive.

3. Il suffit de prendre dans (2.5) que des polynômes qui s'annulent en ce outlier.

5. *Montrer que l'on a la décomposition LDL^t suivante pour T_k : $T_k = L_k D_k L_k^t$ avec*

$$L_k = \begin{pmatrix} 1 & & & \\ -\sqrt{\beta_0} & 1 & & \\ & \ddots & \ddots & \\ & & -\sqrt{\beta_{k-2}} & 1 \end{pmatrix}, D_k = diag(\frac{1}{\alpha_0}, \cdots, \frac{1}{\alpha_{k-1}})$$

6. *Montrer que T_N a les mêmes valeurs propres que A. La* **méthode de Lanczos symétrique** *pour le calcul des valeurs propres d'une matrice consiste à calculer directement ces matrices T_k, ce qui permet de généraliser l'algorithme aux matrices hermitiennes mais pas nécessairement définies positives.*

2.5 La méthode d'Arnoldi

Tous les exemples étudiés dans le §1 donnaient lieu à une matrice A symétrique définie positive. Cependant, dans la discrétisation des edp, il est courant de rencontrer des matrices non symétriques. Voici par exemple une équation de diffusion-convection-advection (1), avec sa formulation faible (2). On y retrouve également une discrétisation par éléments finis (4). Il n'est pas difficile de vérifier que A est symétrique pour $b = 0$ et définie positive si de plus la fonction c est à valeurs ≥ 0. En fait, au moins pour un vecteur b de convection constant, une intégration par parties montre que la partie $A - A^*$ anti-symétrique vient de la discrétisation du terme convectif. Il est donc intéressant de considérer des systèmes $Ax = b$ sous l'hypothèse 2.1.1, mais juste avec A inversible.

Avant de discuter dans le §2.6 des méthodes de projection pour $\mathcal{K}_m = \mathcal{K}_m(A, v_1)$ un espace de Krylov, présentons la construction d'une base othonormée pour une matrice A quelconque. [4] Ici on appliquera l'algorithme de Gram-Schmidt modifié : étant donné un système libre $w_1 = v_1, w_2, ..., w_m \in \mathbb{C}^n$, v_1 de norme 1, à l'étape ℓ on retire de $w_{\ell+1}$ des multiples appropriées des vecteurs $v_1, ..., v_\ell$ déjà calculés, et on normalise pour obtenir $v_{\ell+1}$. Le choix $w_j = A^{j-1}v_1$ semble canonique, mais risque de donner une très grande perte de précision sur ordinateur (au moins pour A hermitienne, cela s'explique par le conditionnement élevé des matrices de Krylov). C'est pour cette raison que l'on prend pour la méthode d'Arnoldi le choix récursif $w_{\ell+1} = Av_\ell$.

2.5.1. L'algorithme d'Arnoldi.

On dispose de $v_1 \in \mathbb{C}^n$ avec $\|v_1\| = 1$
Pour $\ell = 1, 2, ..., m$
  $w = Av_\ell$
  Pour $j = 1, 2, ..., \ell$
    $h_{j,\ell} = (w, v_j)$, $w = w - h_{j,\ell}v_j$
  $h_{\ell+1,\ell} = \|w\|$
  Arrêt si $h_{\ell+1,\ell} = 0$
  $v_{\ell+1} = w/h_{\ell+1,\ell}$.

4. Ce travail a été déjà fait pour A sdp dans les lemmes 2.4.3(c) et 2.4.6, ici une base orthonormée est donnée par $\{r_0/\|r_0\|, ..., r_{m-1}/\|r_{m-1}\|\}$.

Besoins de stockage : $m+1$ vecteurs.
Complexité : $\mathcal{O}(m^2)$ produits scalaires, $\mathcal{O}(m^2)$ saxpies, m produits matrice-vecteur.

On pose $h_{j,\ell} = 0$ pour $j > \ell + 1$, et on range les quantités calculées dans des matrices

$$\begin{aligned} V_m &= (v_1, ..., v_m) \in \mathbb{C}^{n\times m}, \\ \underline{H}_m &= (h_{j,\ell})_{j=1,2,...,m+1,\ell=1,...,m} \in \mathbb{C}^{m+1,m}, \\ H_m &= (h_{j,\ell})_{j=1,2,...,m,\ell=1,...,m} \in \mathbb{C}^{m,m}, \end{aligned}$$

où on remarque que $\underline{H}_m$ et H_m sont des matrices de Hessenberg (supérieures) ayant que des éléments 0 en dessous de la sous-diagonale principale, et H_m est obtenu de $\underline{H}_m$ en supprimant la dernière ligne.

2.5.2. Lemme.
Supposons que l'algorithme 2.5.1 ne s'arrête pas avec $\|w\| = 0$ aux itérations $\ell = 1, ..., m-1$.
(a) *$\forall \ell = 1, ..., m$ $\exists q_{\ell-1}$ polynôme de degré $= \ell - 1$ t.q. $v_\ell = q_{\ell-1}(A)v_1$.*
(b) *$\{v_1, ..., v_m\}$ est une base orthonormée de $\mathcal{K}_m(A, v_1)$*
(c) *Nous avons la décomposition dite d'Arnoldi*

$$AV_m = V_{m+1}\underline{H}_m = V_m H_m + h_{m+1,m} v_{m+1}(0, ..., 0, 1).$$

(d) *$H_m = V_m^* A V_m$.*

Démonstration. La propriété (a) se montre par récurrence sur ℓ sachant que, par construction,

$$\forall \ell = 1, ..., m : \quad Av_\ell = \sum_{j=1}^{\ell+1} h_{j,\ell} v_j \tag{2.6}$$

(où dans le cas $h_{m+1,m} = 0$ on omet le terme $h_{m+1,m}v_{m+1}$). Pour démontrer (b) on remarque que, par construction, $\{v_1, ..., v_m\}$ est une base orthonormée de $\text{span}(v_1, .., v_m) \subset \mathcal{K}_m(A, v_1)$, l'inclusion venant de la partie (a). Comme l'espace de Krylov est au plus de dimension m, on obtient égalité. La partie (c) est une conséquence immédiate de (2.6) en observant que le second membre est égal à la ℓième colonne de $V_{m+1}\underline{H}_m$. Finalement, par (c),

$$V_m^* A V_m = V_m^* V_{m+1} \underline{H}_m = [I, 0]\underline{H}_m = H_m.$$

□

Nous déduisons du lemme 2.5.2(b) et (c) que si $h_{m+1,m} = 0$ alors l'espace de Krylov $\mathcal{K}_m(A, v_1)$ est un espace A-invariant de dimension m. Plus précisément, d'après l'exercice 1.5.8(b), $h_{m+1,m} > 0$ mesure la distance de $Im(V_m)$ à un espace A-invariant :

2.5.3. Exercice : *Montrer que*

$$\min_B \|AV_m - V_m B\| = \min_B \|AV_m - V_m B\|_F = \|AV_m - V_m H_m\| = h_{m+1,m}, \tag{2.7}$$

2.6 Les méthodes FOM et GMRES

2.6.1. Définition de FOM. *Le mième itéré de FOM(=full orthogonalization method) est obtenu par une itération d'une méthode de projection avec $\mathcal{K}_0 = \mathcal{L}_0 = \mathcal{K}_m(A, r_0)$, c'est-à-dire*

$$x_m \in x_0 + \mathcal{K}_m(A, r_0), \quad r_m \perp \mathcal{K}_m(A, r_0).$$

Pour obtenir une base orthonormée de $\mathcal{K}_m(A, r_0)$, posons $\beta = \|r_0\|$, $v_1 = r_0/\beta$ et $V_m = (v_1, ..., v_m) \in \mathbb{C}^{n\times m}$, $H_m \in \mathbb{C}^{m\times m}$ comme dans l'algo d'Arnoldi 2.5.1.

2.6.2. Lemme : propriétés de FOM.
La condition (2.2) *est valable pour le mième itéré de FOM si H_m est inversible. Dans ce cas.*
(a) $x_m = x_0 + V_m y_m$ *avec* $y_m = \beta H_m^{-1} e_1$ *et* $e_1 = (1, 0, ..., 0)^* \in \mathbb{R}^m$.
(b) $r_m = -h_{m+1,m} v_{m+1} e_m^* H_m^{-1} e_1 \beta$, *avec* $e_m = (0, ..., 0, 1)^* \in \mathbb{R}^m$.
(c) $\|r_m\|/\|r_0\| = h_{m+1,m} |e_m^* H_m^{-1} e_1|$.

Démonstration. La partie (a) découle directement du lemme 2.2.1 sachant que $V_m^* A V_m = H_m$ par le lemme 2.5.2(d), et $V_m^* r_0 = \beta V_m^* v_1 = \beta e_1$. Pour la partie (b), notons que, d'après (a),

$$r_m = r_0 - A V_m y_m = r_0 - V_{m+1} \underline{H}_m y_m \in \mathcal{K}_{m+1}(A, r_0), \quad \perp K_m(A, r_0),$$

et alors r_m est un multiple de v_{m+1} par le lemme 2.5.2(b), donc

$$r_m = v_{m+1} v_{m+1}^* r_m = -v_{m+1} v_{m+1}^* V_{m+1} \underline{H}_m y_m = -v_{m+1} e_{m+1}^* \underline{H}_m y_m = -h_{m+1,m} v_{m+1} e_m^* y_m,$$

ce qui démontre (b). Partie (c) est une conséquence immédiate de (b). □

Il est difficile de déduire du lemme 2.6.2(c) une estimation a priori où alors une simple formule pour $x_m - x_{m-1}$ sans hypothèses supplémentaires sur A. Notons toutefois que $h_{m+1,m}$ peut être petit, comparer avec (2.7).

2.6.3. Exercice : *Soit $A \in \mathbb{C}^{n\times n}$ hermitienne.*

(a) Montrer que H_m est hermitienne. En déduire une version de l'algorithme d'Arnoldi nécessitant seulement $\mathcal{O}(m)$ saxpies et produits scalaires.

(b) Donner b, x_0 et une matrice A symétrique et inversible avec $r_0 = e_1$ de sorte que le premier itéré de FOM n'existe pas (car H_1 n'est pas inversible).

(c) Soit de plus A sdp. Pour chercher le lien entre les itérés de CG et FOM, montrer que H_m est inversible, et que $x_m^{CG} \in x_0^{CG} + \mathcal{K}_m(A, r_0)$. En déduire que si $x_m^{CG} = x_m^{FOM}$ est valable pour $m = 0$ alors c'est valable pour tout m.

Notons que la propriété énoncée dans l'exercice 2.6.3(c) n'est généralement plus valable sur ordinateur car les vecteurs x_m^{CG} et x_m^{FOM} ne sont pas calculés par le même algorithme.

2.6.4. Exercice : *Soit p un polynôme de degré $< m$. Montrer (d'abord pour les monômes et ensuite pour des polynômes généraux) que*

$$p(A)v_1 = V_m p(H_m) e_1.$$

Quelle est donc la valeur de $p(H_m)e_1$ avec $p = q_{j-1}$ comme dans le lemme 2.5.2 ? En déduire que le mième résidu de FOM vérifie $r_m = q_m(A)r_0/q_m(0)$

2.6.5. Exercice : *L'image numérique d'une matrice carrée B d'ordre k est définie par*

$$W(B) = \{\frac{(By,y)}{(y,y)} : y \in \mathbb{C}^k \setminus \{0\}\}$$

(l'ensemble des quotients de Rayleigh).
(a) *Vérifier que les valeurs propres de B appartiennent à $W(B)$. En déduire que $0 \notin W(B)$ implique que B est inversible.*
(b) *Supposons de plus que B est sdp. Vérifier que $0 \notin W(B)$.*
(c) *Supposons que $0 \notin W(A)$. Montrer que la condition (2.2) est valable pour le mième itéré de FOM pour tout $m \leq n$ (avant l'arrêt d'Arnoldi).*

On observe que derrière FOM il n'y a généralement pas une propriété de minimisation. Par contre, dans l'approche suivante on minimise la taille du résidu, comparer avec le théorème 2.2.3 avec $M = A$.

2.6.6. Définition de GMRES. *Le mième itéré de GMRES(=generalized minimal residual) est obtenu par une itération d'une méthode de projection avec $\mathcal{K}_0 = \mathcal{K}_m(A, r_0)$ et $\mathcal{L}_0 = A\mathcal{K}_0$, c'est-à-dire*

$$x_m \in x_0 + \mathcal{K}_m(A, r_0), \quad r_m \perp A\mathcal{K}_m(A, r_0).$$

Exprimons les itérés de GMRES en termes des quantités $\beta = \|r_0\|$, $v_1 = r_0/\beta$, $V_m, \underline{H}_m$ de la méthode d'Arnoldi.

2.6.7. Lemme : propriétés de GMRES.
Si l'algorithme d'Arnoldi s'arrête à l'itération m alors le mième itéré de GMRES donne la solution exacte $x_m = \overline{x}$.
Si l'algorithme d'Arnoldi ne s'arrête pas avant l'itération m alors le mième itéré de GMRES est bien défini, et est donné par la formule $x_m = x_0 + V_m y_m$, avec y_m solution du problème des moindres carrés

$$\min_{y \in \mathbb{C}^m} \|\underline{H}_m y - \beta e_1\|.$$

Démonstration. Comme l'algorithme d'Arnoldi ne s'arrête pas avant l'itération m, on dispose de V_m avec colonnes qui d'après le lemme 2.5.2 engendrent $\mathcal{K}_0$. Pour appliquer le lemme 2.2.1, on prend les matrices V_m et $U_m = AV_m$. Donc

$$U_m^* A V_m = V_m^*(A^*A)V_m \quad \text{sdp car } A^*A \text{ sdp, et } \operatorname{rang}(V_m) = m.$$

Donc la condition (2.2) est valable et le $m$ième itéré de GMRES x_m est bien défini. De plus, $x_m = x_0 + V_m y_m$, où d'après les lemmes 2.2.1 et 2.5.2

$$\underline{H}_m^* \underline{H}_m y_m = (V_{m+1}\underline{H}_m)^* V_{m+1}\underline{H}_m y_m = U_m^* A V_m y_m = U_m^* r_0 = \underline{H}_m^* V_{m+1}^* v_1 \beta = \underline{H}_m^* e_1 \beta.$$

On vient donc de démontrer que y_m est la solution unique du système des équations normales du problème des moindres carrés donné ci-dessus (l'unicité vient du fait que $\operatorname{rang}(\underline{H}_m) = m$), ce qui demontre la deuxième partie. Finalement, si Arnoldi s'arrête à l'itération m alors $h_{m+1,m} = 0$ et $\underline{H}_m$ est de rang m, avec une dernière ligne égale à zéro. Par conséquent,

$$0 = \min_y \|\underline{H}_m y - \beta e_1\| = \|\underline{H}_m y_m - \beta e_1\| = \|V_{m+1}(\underline{H}_m y_m - \beta e_1)\| = \|b - Ax_m\|,$$

et alors $r_m = 0$. □

Pour résoudre le problème des moindres carrés du lemme 2.6.2, on calcule une décomposition QR dite économique $\underline{H}_m = Q_m R_m$ avec $Q_m \in \mathbb{C}^{(m+1)\times m}$ à colonnes orthonormées et R_m une matrice triangulaire supérieure inversible. On se ramène au système $R_m y_m = \beta Q_m^* e_1$ qui se résout par une remontée. Comme $\underline{H}_m$ est déjà de forme Hessenberg, il suffit d'annuler les éléments sur la sous-diagonale, sans détruire les zéros ailleurs sous la diagonale, ce qui peut se faire de manière efficace par des rotations de Givens (ou par transformations de Householder).

Pour comparer FOM et GMRES, on donne la formule suivante (sans preuve)

$$\frac{1}{\|r_m^{GMRES}\|^2} = \sum_{j=0}^{m} \frac{1}{\|r_j^{FOM}\|^2},$$

avec la convention que $\|r_j^{FOM}\|^2 = \infty$ si H_j n'est pas inversible. On en déduit la monotonie

$$\|r_m^{GMRES}\| \leq \|r_{m-1}^{GMRES}\|, \quad \|r_m^{GMRES}\| \leq \|r_j^{FOM}\|, \quad j \leq m,$$

(que l'on aurait pu aussi déduire par exemple du fait que $\|r_m^{GMRES}\|^2 = \min_y \|\underline{H}_m y - \beta e_1\|$ et du fait que $\underline{H}_{m-1}$ est emboîté dans $\underline{H}_m$). Aussi, GMRES stagne à l'itération m avec $\|r_m^{GMRES}\| = \|r_{m-1}^{GMRES}\|$ ssi le $m$ième itéré de FOM n'est pas défini.

Pour GMRES, les estimations a priori d'erreur ne sont pas si facile à obtenir, il existe des estimations dans le cas $0 \notin W(A)$. On se contentera seulement du cas de la matrice hermitienne (dont on sait déjà d'après l'exercice 2.6.3(a) que le calcul de $V_m, \underline{H}_m$ et donc x_m^{GMRES} se simplifie, voir le lien dans la littérature avec MinRes).

2.6.8. Théorème : estimation à priori pour GMRES.
Soit $A = A^*$, et $\kappa = cond(A)$.
(a) *Nous avons*

$$\|r_{m+1}\| \leq \|r_m\| \leq 2\,\|r_0\| \left(\frac{\kappa - 1}{\kappa + 1}\right)^{\lfloor \frac{m}{2} \rfloor}.$$

(b) *Si A est de plus sdp alors*

$$\|r_m\| \leq 2\,\|r_0\| \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}\right)^{m}.$$

Démonstration. Par construction,

$$r_m \in r_0 + \mathcal{L}_m(A, r_0) \subset K_{m+1}(A, r_0).$$

Il existe alors un polynôme P_m de degré $\leq m$ de sorte que $r_m = P_m(A) r_0$, $P_m(0) = 1$. Aussi, d'après le théorème 2.2.3 avec $M = A$, pour tout polynôme Q_m de degré $\leq m$ avec $Q_m(0) = 1$ nous avons

$$\|r_m\| \leq \|\frac{Q_m(A)}{Q_m(0)} r_0\|,$$

ce qui implique la première inégalité dans la partie (a) par le choix $Q_{m+1} = P_m$, et la partie (b) par le choix de Q_m en fonction des polynômes de Chebyshev comme fait dans le théorème 2.4.7. Finalement, dans le cas (a) d'une matrice A seulement hermitienne, si $m = 2k$ est pair, on se ramène par le choix $Q_m(x) = \widetilde{Q}_k(x^2)$ avec $\widetilde{Q}_k$ comme avant au cas d'une matrice A^2 sdp, avec $\mathrm{cond}(A^2) = \mathrm{cond}(A)^2$. □

Résumons un peu le contenu de ce chapitre : il est bien plus compliqué de résoudre (d'une manière approchée) un système d'équations linéaires $Ax = b$ dans le contexte 2.1.1 si A n'est pas sdp. Les méthodes FOM et GMRES s'appliquent dans le cas d'une matrice A quelconque (mais inversible), ce ne sont cependant pas des méthodes itératives. Pour obtenir les itérés $x_0, .., x_m$, il faut d'abord calculer par Arnoldi une base orthonormée de l'espace de Krylov $\mathcal{K}_m(A, r_0)$ ce qui nécessite le stockage de $m + 1$ vecteurs du $\mathbb{C}^n$ et $\mathcal{O}(m^2)$ saxpies et produits scalaires. le calcul restant, par exemple une décomposition QR de H_m ou $\underline{H}_m$ est de complexité $\mathcal{O}(m^2)$, mais le produit $V_m y_m$ nécessite encore $\mathcal{O}(m^2)$ autres saxpies (ou juste $\mathcal{O}(m)$ si on calcule le $m$ième itéré). Pour des matrices A symétriques, par une implémentation sophistiquée on peut réduire le nombre de saxpies à $\mathcal{O}(m)$, voir les exercices 2.6.3 et 2.7.3.

En vue des besoins de stockage, on peut conclure que cette approche est seulement viable si $m \ll n$. Si on n'a pas encore atteint la précisions souhaitée, on redémarre GMRES en gardant seulement en mémoire le dernier itéré de GMRES et son résidu, ce qui donne la méthode de projection suivante.

2.6.9. Algorithme GMRES(m)=GMRES redémarré après m itérations.

On dispose d'un entier $m > 0$, de $x_0 \in \mathbb{C}^n$ et de son résidu $r_0 = b - Ax_0$
Pour $\ell = 0, 1, 2, ..$ jusqu'à $\|r_\ell\|$ assez petit
Calculer $x_{\ell+1} \in x_\ell + \mathcal{K}_m(A, r_\ell)$
avec résidu $r_{\ell+1} \perp A\mathcal{K}_m(A, r_\ell)$ par m itérations de GMRES.
Besoins de stockage : $m + 2$ vecteurs.
Complexité par itération : $\mathcal{O}(m^2)$ produits scalaires, $\mathcal{O}(m^2)$ saxpies, m produits matrice-vecteur.

Aux applications, on prend souvent $m \in \{2, 10, 50\}$ suivant les besoins de stockage. Notons que, dans le cas général, il n'existe pas une étude complète de convergence de cet algorithme. Une stratégie similaire de redémarrage peut être imaginée pour d'autres algorithmes, par exemple FOM(m). On remarque (exercice) que $CG(1)$ donne la méthode de steepest descent, avec le comportement de Zigzag observé avant.

2.7 D'autres exercices sur les méthodes de Krylov

2.7.1. Exercice : *Notons par $\mathcal{K}_j(B, d)$ le sous–espace vectoriel de $\mathbb{C}^n$ engendré par $d, Bd, \cdots, B^{j-1}d$, où $d \in \mathbb{C}^n$ et $B \in \mathbb{C}^{n\times n}$. On considère la suite (x_j) définie par*

$$x_j - x_0 \in \mathcal{K}_j(A, r_0), \quad r_j = b - A.x_j \text{ orthogonal à } \mathcal{K}_j(A^*, y),$$

où $b, x_0, y \in \mathbb{C}^n$ et $A \in \mathbb{C}^{n\times n}$.

1. *Montrer que x_j est déterminé par la solution d'un système d'équations linéaires $H_j.z_j = c_j$ avec H_j à préciser.*
2. *Montrer que r_j admet le représentation $r_j = P_j(A).r_0$ avec P_j un polynôme à préciser.*
3. *Montrer que pour chaque $B \in \mathbb{C}^{n\times n}$, $d \in \mathbb{C}^n$ il existe un entier $\kappa = \kappa(B, d) \geq 0$ tel que*

$$dim\mathcal{K}_j(B, d) = \begin{cases} j & si \quad 0 \leq j \leq \kappa, \\ \kappa & si \quad j > \kappa. \end{cases}$$

4. *Supposons que A soit hermitienne définie positive, et que $y = r_0, \kappa = \kappa(A, r_0)$. Montrer que les vecteurs $x_1, \cdots, x_\kappa$ sont bien définis, $r_0, \cdots, r_{\kappa-1}$ sont deux à deux orthogonaux, et que $r_\kappa = 0$. En déduire que l'on retrouve la méthode du gradient conjugué.*

2.7.2. Exercice : Algorithme IOM (Incomplete Orthogonalization Method)
Un inconvénient de l'algorithme d'Arnoldi (et donc de FOM et GMRES) est l'augmentation de l'espace mémoire utilisé quand m croit. Pour éviter ce problème on va proposer une variante qui consiste à faire une orthogonalisation incomplète : Pour un entier $k \geq 1$ dit indice de troncature, on construira des vecteurs $v_1^Q, v_2^Q, ..., v_m^Q \in \mathbb{R}^n$ de sorte que

$$\forall j, \ell = 1, ..., m : \quad si\ |j - \ell| \leq k\ alors\ \langle v_j^Q, v_\ell^Q \rangle = \delta_{j,\ell}. \tag{2.8}$$

On dispose de $v_1^Q \in \mathbb{C}^n$ avec $\|v_1^Q\| = 1$
Pour $\ell = 1, 2, ..., m-1, m$
$w = Av_\ell^Q$
Pour $j = \max(1, \ell - k), ..., \ell - 1, \ell$
$h_{j,\ell} = (w, v_j^Q)$, $w = w - h_{j,\ell} v_j^Q$
$h_{\ell+1,\ell} = \|w\|$
Arrêt si $h_{\ell+1,\ell} = 0$
$v_{\ell+1}^Q = w / h_{\ell+1,\ell}$.

Le calcul de v_{m+1}^Q nécessite alors de stocker seulement $v_{m-k}^Q, ..., v_m^Q$.

1. *Posons l'espace de Krylov $\mathcal{K}_m := \mathcal{K}_m(A, v_1^Q)$, et supposons que l'Algorithme d'Arnoldi avec $v_1 = v_1^Q$ ne s'arrête pas avant l'itération m. Montrer que*

$$v_1^Q \in \mathcal{K}_1, \quad et\ pour\ \ell = 2, ..., m : \quad v_\ell^Q \in \mathcal{K}_\ell \setminus \mathcal{K}_{\ell-1}.$$

   En déduire que $\{v_1^Q, ..., v_m^Q\}$ est un système libre, et alors une base de $\mathcal{K}_m$.

2. *Montrer que (2.8) est valable.*

3. *Si $m \leq k+1$ ou si il existe un polynôme q de degré $\leq k$ avec $A^T = q(A)$, montrer que $\{v_1^Q, ..., v_m^Q\}$ nous donne bien la base d'Arnoldi.*

4. *Posons $V_m^Q = (v_1^Q, ..., v_m^Q)$. En analogie avec la mérhode d'Arnoldi, trouver $\underline{H}_m^Q$ une matrice bande avec une sousdiagonale et k superdiagonales de sorte que*

$$AV_m^Q = V_{m+1}^Q \underline{H}_m^Q.$$

5. *Avec $v_1^Q = r_0/\beta, \beta = \| r_0 \|$, notons par H_m^Q la matrice carrée obtenue en gardant les premières m lignes de $\underline{H}_m^Q$. Le mième itéré de la méthode IOM est calculé par la formule*

$$x_m = x_0 + V_m^Q y_m \ avec\ y_m = (H_m^Q)^{-1}(\beta e_1)$$

   dont on suppose que H_m^Q est inversible. En comparant avec FOM, vérifier que $x_m \in x_0 + \mathcal{K}_m$, que $r_m := b - Ax_m$ est un multiple de v_{m+1}^Q, mais que en général r_m n'est pas orthogonal à $\mathcal{K}_m$.

6. *Supposons que H_ℓ^Q est inversible pour $\ell = 1, 2, ..., m$ de sorte qu'il existe une décomposition LU de la forme $H_m = L_m U_m$ avec L_m triangulaire inférieure ayant des 1 sur la diagonale, et U_m triangulaires supérieure. Vérifier pour les éléments de L_m et U_m que*

$$\text{pour } j \geq \ell + 1 : |L_{j,\ell} = 0, \text{ pour } \ell \geq j + k + 1 : U_{j,\ell} = 0,$$

c'est-à-dire, L_m et U_m sont des matrices bandes.

7. *On va proposer une forme récursive pour calculer les x_m basée sur la décomposition précédente.*

   (a) *On pose $P_m = V_m U_m^{-1}$. Montrer que x_m peut s'écrire sous la forme $x_m = x_0 + P_m z_m$. Expliciter le vecteur z_m.*

   (b) *On note $p_\ell, \ell = 1, \cdots, m$ les colonnes de P_m. Montrer que*

$$p_m = \frac{1}{U_{m,m}} \left[v_m^Q - \sum_{\ell=\max(1,m-k)}^{m-1} U_{\ell,m} p_\ell \right].$$

   (c) *Montrer que $z_m = \begin{pmatrix} z_{m-1} \\ \zeta_m \end{pmatrix}$ avec $\zeta_m = -L_{m,m-1}\zeta_{m-1}$.*

   (d) *Conclure que $x_m = x_{m-1} + \zeta_m p_m$.*

   (e) *Détailler l'algorithme obtenu.*

2.7.3. Exercice : Lanczos symétrique

La méthode de Lanczos symétrique est la méthode qui résulte de l'application de la méthode FOM à la résolution d'un système avec A symétrique.

1. *Montrer que dans ce cas, avec les notations usuelles, on a :*

$$H_m = T_m = tridiag(\beta_i, \alpha_i, \beta_{i+1}).$$

2. *Montrer que l'on a la décomposition suivante :*

$$T_m = \begin{pmatrix} 1 & & & & \\ \lambda_2 & 1 & & & \\ & \lambda_3 & \ddots & & \\ & & \ddots & \ddots & \\ & & & \lambda_m & 1 \end{pmatrix} \times \begin{pmatrix} \eta_1 & \beta_2 & & & \\ & \eta_2 & \ddots & & \\ & & \ddots & \ddots & \\ & & & \ddots & \beta_m \\ & & & & \eta_m \end{pmatrix}$$

   avec $\lambda_i = \beta_i/\eta_{i-1}$, $\eta_i = \alpha_i - \lambda_i \beta_i$.

3. *Nous voulons calculer récursivement les itérés de la méthode de Lanczos pour résoudre $Ax = b$. Nous rappelons qu'ils sont donnés par*

$$x_m = x_0 + V_m y_m \text{ avec } y_m = T_m^{-1}(\beta e_1)$$

   On pose $P_m = V_m U_m^{-1}, z_m = L_m^{-1}(\beta e_1)$ et donc on a $x_m = x_0 + P_m z_m$.

(a) *Calculer p_m (la m-ième colonne de P_m) à partir des p_i ($i < m$) et v_m.*

Montrer que $z_m = \begin{pmatrix} z_{m-1} \\ \zeta_m \end{pmatrix}$ *avec* $\zeta_m = -\lambda_m \zeta_{m-1}$ *et que* $x_m = x_{m-1} + \zeta_m p_m$.

(b) *En utilisant les relations précédentes écrire l'algorithme de Lanczos pour résoudre $Ax = b$.*

4. *Soient $r_m = b - Ax_m$ les résidus produits par Lanczos, p_m les vecteurs auxiliaires obtenus à la question précédente.*

(a) *Montrer que $\forall m$, $r_m = \sigma_m v_{m+1}$, avec σ_m scalaire et conclure que les résidus sont deux à deux orthogonaux .*

(b) *Montrer que les vecteurs auxiliaires p_i sont A-orthogonaux, i.e., $(Ap_i, p_j) = 0 \; \forall i \neq j$.*

(c) *Quel algorithme retrouve-t-on ?*

2.7.4. Exercice :
Montrer que pour la méthode GMRES l'approximation x_m à l'étape m peut s'écrire

$$x_m = x_0 + V_m y$$

où y est la solution du système $(\underline{H}_m^T \underline{H}_m) y = \underline{H}_m^T (\beta e_1)$.
(indication : appliquer la formule générale pour une méthode de projection dans le cas $K = K_m(A, r_0)$, $L = AK_m(A, r_0)$)

2.7.5. Exercice :
Soit A une matrice de la forme

$$A = \begin{pmatrix} I & Y \\ 0 & I \end{pmatrix}$$

1. *Montrer que les matrices I, A, A^2 sont liées.*
2. *Quel est le nombre maximal d'étapes pour trouver la solution du système $Ax = b$ par la méthode GMRES ?*

2.7.6. Exercice : Implémentation pratique de GMRES
On rappelle que la méthode GMRES consiste à projeter sur $\mathcal{K} = K_m(A, r_0)$ orthogonalement à $\mathcal{L} = AK_m(A, r_0)$ et donc à calculer

$$\begin{aligned} x_m &= x_0 + V_m y_m \\ y_m &= argmin_y \parallel \beta e_1 - \underline{H}_m y \parallel_2 \end{aligned}$$

avec $\underline{H}_m$ matrice de Hessenberg $(m+1) \times m$ obtenue par la méthode d'Arnoldi. Le but de cet exercice est de donner une méthode efficace de résolution de ce problème de moindres carrés (sous la seule hypothèse que $\underline{H}_m \in \mathbb{R}^{(m+1)\times m}$ est une matrice de Hessenberg de rang m).

1. *Montrer par récurrence sur $\ell = 1, ..., m$ qu'il existe une rotation de Givens Ω_ℓ obtenue de l'identité d'ordre $m+1$ en remplaçant la sous-matrice d'ordre 2 formée des lignes/colonnes d'indice ℓ et $\ell + 1$ par*

$$\begin{bmatrix} c_\ell & -s_\ell \\ s_\ell & c_\ell \end{bmatrix}, \quad c_\ell, s_\ell \in \mathbb{R}, \quad c_\ell^2 + s_\ell^2 = 1,$$

de sorte que

$$\Omega_\ell ... \Omega_1 \underline{H}_m = \begin{bmatrix} R_\ell & * \\ 0 & * \end{bmatrix}$$

est une matrice de Hessenberg, avec $R_\ell \in \mathbb{R}^{\ell \times \ell}$ *une matrice triangulaire supérieure.*

2. *Montrer que* $Q_m = \Omega_m ... \Omega_1$ *est une matrice orthogonale, et que* $Q_m \underline{H}_m = \begin{bmatrix} R_m \\ 0 \end{bmatrix}$ *avec* $R_m \in \mathbb{R}^{m \times m}$ *une matrice triangulaire supérieure inversible ayant comme sous-matrice principale d'ordre* ℓ *la matrice* R_ℓ.
3. *Vérifier que*

$$\Omega_\ell ... \Omega_1 \beta e_1 = \begin{bmatrix} \underline{g}_\ell \\ 0 \end{bmatrix}, \quad \underline{g}_\ell = \begin{bmatrix} g_\ell \\ \gamma_\ell \end{bmatrix} \in \mathbb{R}^{\ell+1}, \quad \gamma_\ell \in \mathbb{R}.$$

4. *Montrer que notre problème des moindres carrés revient résoudre*

$$y_m = argmin_y \parallel \underline{g}_m - \begin{bmatrix} R_m \\ 0 \end{bmatrix} y \parallel$$

avec solution $y_m = R_m^{-1} g_m$, *et* $\|\beta e_1 - \underline{H}_m y_m\| = |\gamma_m| = |s_1 s_2 ... s_m| \beta$.
5. *Montrons une première formule récursive. En observant que*

$$\underline{g}_\ell = \begin{bmatrix} g_{\ell-1} \\ c_\ell \gamma_{\ell-1} \\ s_\ell \gamma_{\ell-1} \end{bmatrix}, \quad \text{et alors } \gamma_\ell = s_\ell \gamma_{\ell-1},$$

montrer que

$$y_m = \begin{bmatrix} y_{m-1} \\ 0 \end{bmatrix} + c_m \gamma_{m-1} R_m^{-1} e_m.$$

6. *Montrons une formule explicite pour le résidu. Vérifier que*

$$\beta e_1 - \underline{H}_m y_m = \beta e_1 - Q_m^* \begin{bmatrix} I_m \\ 0 \end{bmatrix} [I_m, 0] Q_m \beta e_1.$$

En déduire que $\beta e_1 - \underline{H}_m y_m = \gamma_m Q_m^* e_{m+1}$.
7. *Montrons une formule récursive pour le résidu. En observant que*

$$Q_m^* e_{m+1} = \begin{bmatrix} Q_{m-1}^* & 0 \\ 0 & 1 \end{bmatrix} \Omega_m^* e_{m+1} = s_m \begin{bmatrix} Q_{m-1}^* e_m \\ 0 \end{bmatrix} + c_m e_{m+1},$$

montrer que

$$\beta e_1 - \underline{H}_m y_m = s_m^2 \begin{bmatrix} \beta e_1 - \underline{H}_{m-1} y_{m-1} \\ 0 \end{bmatrix} + c_m \gamma_m e_{m+1}.$$

2.7.7. Exercice : Méthode QUASI-GMRES

Soient $V_m^Q, H_m^Q, \underline{H}_m^Q, x_0, r_0, \beta, v_1^Q = r_0/\beta$ *comme dans l'exercice 2.7.2. La méthode QUASI-GMRES est définie par*

$$x_m^Q = x_0 + V_m^Q y_m^Q, \quad y_m^Q = \arg\min_y \|\underline{H}_m^Q y - \beta e_1\|.$$

1. *Vérifier que* $r_m^Q = b - Ax_m^Q = V_{m+1}^Q(\beta e_1 - \underline{H}_m^Q y_m)$.
2. *Exploitons les résultats de l'exercice 2.7.6 appliqué à la matrice de Hessenberg* $\underline{H}_m^Q$.
   (a) *Montrer que*
   $$\|r_m^Q\| \leq \|V_{m+1}^Q\|\,|\gamma_m| \leq \sqrt{m+1}|\gamma_m|.$$
   (b) *On pose* $P_\ell = V_\ell^Q R_\ell^{-1}$. *Vérifier que* $P_\ell = (P_{\ell-1}, p_\ell)$, *et que*
   $$x_m^Q = x_{m-1}^Q + c_\ell \gamma_{\ell-1} p_m,$$
   avec p_m *calculable par*
   $$p_m = \frac{1}{R_{m,m}}\Big(v_m^G - \sum_{\ell=\max(m-k-1,1)}^{m-1} p_\ell R_{\ell,m}\Big).$$
   (c) *Montrer que*
   $$r_m^Q = s_m^2 r_{m-1}^Q + c_m \gamma_m v_{m+1}^Q.$$
   Notons par x_m^I *le* m*ième itéré de IOM, et par* $r_m^I = b - Ax_m^I$ *son résidu. En utilisant le fait qu'un résidu peut s'écrire comme* $P(A)r_0$ *avec un polynôme* P *vérifiant* $P(0) = 1$, *déduire que*
   $$r_m^Q = s_m^2 r_{m-1}^Q + c_m^2 r_m^I, \quad x_m^Q = s_m^2 x_{m-1}^Q + c_m^2 x_m^I.$$
3. *Regardons le lien entre GMRES (avec* x_m^G, r_m^G*) et QUASI-GMRES. On suppose que la méthode d'Arnoldi ne s'arrête pas avant l'itération* m.
   (a) *Montrer que l'on a la décomposition QR économique* $V_m^Q = V_m S_m$ *avec* S_m *une matrice triangulaire supérieure inversible, et* $cond(V_m^Q) := \sqrt{cond((V_m^Q)^* V_m^Q)} = cond(S_m)$.
   (b) *Avec* $\underline{H}_m$ *la matrice de Hessenberg de la méthode d'Arnoldi, montrer que*
   $$\underline{H}_m^Q = S_{m+1}^{-1} \underline{H}_m S_m.$$
   (c) *En déduire que*
   $$\min_y \|\beta e_1 - \underline{H}_m^Q y\| \leq \|S_{m+1}^{-1}\| \min_y \|\beta e_1 - \underline{H}_m y\|.$$
   (d) *Conclure que*
   $$\frac{1}{cond(V_{m+1}^G)}\|r_m^G\| \leq \|r_m^Q\| \leq cond(V_{m+1}^G)\|r_m^G\|.$$

2.7.8. Exercice : Biorthogonalisation de Lanczos

L'algorithme de biorthogonalisation est le suivant :
Soit v_1 *et* w_1 *tel que* $(v_1, w_1) = 1$, *soit* $\beta_1 = \delta_1 = 0$, $w_0 = v_0 = 0$,
Pour $j = 1, 2, \ldots, m$

$$\begin{aligned}
\alpha_j &= (Av_j, w_j)\\
\widehat{v}_{j+1} &= Av_j - \alpha_j v_j - \beta_j v_{j-1}\\
\widehat{w}_{j+1} &= A^T w_j - \alpha_j w_j - \delta_j w_{j-1}\\
\delta_{j+1} &= |(\widehat{v}_{j+1}, \widehat{w}_{j+1})|^{1/2}. \quad \texttt{Si} \quad \delta_{j+1} = 0 \quad \texttt{Stop}\\
\beta_{j+1} &= (\widehat{v}_{j+1}, \widehat{w}_{j+1})/\delta_{j+1}\\
w_{j+1} &= \widehat{w}_{j+1}/\beta_{j+1}\\
v_{j+1} &= \widehat{v}_{j+1}/\delta_{j+1}
\end{aligned}$$

1. *Montrer par récurrence que les vecteurs v_i et w_j forment un système biorthogonal, i.e.*

$$(v_i, w_j) = \delta_{ij}, \quad 1 \leq i, j \leq m,$$

et que $\{v_i\}_{i=1,\ldots,m}$ est une base de $K_m(A, v_1)$, $\{w_i\}_{i=1,\ldots,m}$ est une base de $K_m(A^T, w_1)$.

2. *Soit $T_m = tridiag(\delta_i, \alpha_i, \beta_{i+1})$. Montrer les relations*

$$AV_m = V_m T_m + \delta_{m+1} v_{m+1} e_m^T,$$
$$A^T W_m = W_m T_m^T + \beta_{m+1} w_{m+1} e_m^T,$$
$$W_m^T A V_m = T_m.$$

2.7.9. Exercice : Gradient Biconjugué

On considère un système $Ax = b$ et un système dual $A^T x^* = b^*$. La méthode du gradient biconjugué consiste à projeter sur $K_m = span\{v_1, Av_1, \cdots, A^{m-1}v_1\}$ orthogonalement à $L_m = span\{w_1, A^T w_1, \cdots, (A^T)^{m-1} w_1\}$.

Soient x_0 et x_0^* donnés. On pose $v_1 = r_0/\|r_0\|$, $w_1 = r_0^*/\|r_0^*\|$. On applique la méthode de biorthogonalisation de Lanczos pour générer les bases biorthogonales $\{v_1, v_2, \cdots, v_m\}$ et $\{w_1, w_2, \cdots, w_m\}$, ainsi que la matrice tridiagonale T_m =tridiag$(\delta_i, \alpha_i, \beta_i)$.

1. *Montrer que les itérés x_m et x_m^* sont donnés par :*

$$x_m = x_0 + V_m y_m, \text{ avec } y_m = T_m^{-1}(\beta e_1),$$
$$x_m^* = x_0^* + W_m y_m^*, \text{ avec } y_m^* = (T_m^T)^{-1}(\delta e_1),$$

où $\beta = \|r_0\|_2$ et $\delta = \|r_0^*\|_2$.

2. *Montrer que*

$$\begin{aligned} r_j &= -\delta_{j+1} e_j^T y_j v_{j+1}, \\ r_j^* &= -\beta_{j+1} e_j^T y_j^* w_{j+1}. \end{aligned}$$

3. *L'algorithme peut s'écrire sous la forme suivante :*
   - *calcul de $r_0 = b - Ax_0$, $r_0^* = b - A^T x_0^*$*
   - *poser $p_0 = r_0$, $p_0^* = r_0^*$*
   - *pour $j = 0, 1, \cdots$ jusqu'à convergence faire*

$$\begin{aligned} \alpha_j &= (r_j, r_j^*)/(Ap_j, p_j^*) \\ x_{j+1} &= x_j + \alpha_j p_j \\ r_{j+1} &= r_j - \alpha_j A p_j \\ r_{j+1}^* &= r_j^* - \alpha_j A^T p_j^* \\ \beta_j &= (r_{j+1}, r_{j+1}^*)/(r_j, r_j^*) \\ p_{j+1} &= r_{j+1} + \beta_j p_j \\ p_{j+1}^* &= r_{j+1}^* + \beta_j p_j^* \end{aligned}$$

   (a) *Montrer les relations d'orthogonalité et A-orthogonalité suivantes (d'où le nom de la méthode) :*

$$(r_i, r_j^*) = 0 \ \ \forall i \neq j \ , \ (Ap_i, p_j^*) = 0 \text{ pour } i \neq j$$

   (b) *Obtenir une relation de récurrence à trois termes entre les résidus.*

2.8 Le gradient conjugué préconditionné

La convergence des méthodes itératives étudiées dans les chapitres 2.2–2.7 dépend bien souvent du conditionnement de A ou de la répartition de ses valeurs propres, d'où l'idée de remplacer le système $Ax = b$ par $\widehat{A}x = \widehat{b}$, $\widehat{A} = C^{-1}A$, $\widehat{b} = C^{-1}b$, avec C choisi tel que :

- la matrice $\widehat{A} = C^{-1}A$ donne un meilleur taux de convergence (par exemple $C^{-1}A \approx I$ ce qui correspond de choisir $C \approx A$) ;
- on peut facilement implémenter le produit $z = C^{-1}Ay$ ce qui équivaut à poser $d = Ay$ et résoudre le système $Cz = d$.

Ici on considère seulement l'algorithme CG pour une matrice A sdp, mais $C^{-1}A$ n'est généralement pas sdp. On va donc varier légèrement l'approche, et considérer seulement des matrices C ayant une factorisation connue $C = TT^*$ (avec T triangulaire inférieure pour pouvoir facilement résoudre le système $Cz = d$). On considère le système

$$\widetilde{A}\widetilde{x} = \widetilde{b}, \quad \widetilde{A} = T^{-1}AT^{-*}, \quad \widetilde{b} = T^{-1}b, \quad \widetilde{x} = T^*x$$

avec $\widetilde{A}$ sdp (et semblable à $\widehat{A}$) étant mieux conditionné que A, ou ayant une répartition de valeurs propres plus favorable, par exemple dans un petit intervalle bien séparé de 0, plus quelques outliers.

Notons par $\widetilde{x}_k$, $\widetilde{r}_k = \widetilde{b} - \widetilde{A}\widetilde{x}_k$, $\widetilde{p}_k$ les vecteurs obtenues par CG appliqué au système $\widetilde{A}\widetilde{x} = \widetilde{b}$, voir 2.4.5. Nous souhaitons écrire cet algorithme en termes des nouveaux vecteurs

$$\begin{aligned} x_k &:= T^{-*}\widetilde{x}_k, \quad r_k := b - Ax_k = T(\widetilde{b} - \widetilde{A}\widetilde{x}_k) = T\widetilde{r}_k, \\ p_k &:= T^{-*}\widetilde{p}_k, \quad z_k := C^{-1}r_k = T^{-*}\widetilde{r}_k, \end{aligned}$$

pour obtenir directement une suite x_k approchant la solution de $Ax = b$. Observons tout d'abord que l'on obtient la même expression pour la norme énergie (et donc les estimations d'erreurs)

$$\|x - x_k\|_A = \|T^{-*}(\widetilde{x} - \widetilde{x}_k)\|_A = \|\widetilde{x} - \widetilde{x}_k\|_{\widetilde{A}}.$$

Aussi, l'orthogonalité des résidus $\widetilde{r}_k$ se traduit en une C-orthogonalité ou C^{-1}-orthogonalité, ou alors une biorthogonalité

$$(\widetilde{r}_k, \widetilde{r}_j) = (r_k, r_j)_{C^{-1}} = (z_k, z_j)_C = (r_k, z_j) \quad (= 0 \text{ pour } j \neq k).$$

De même, $(p_j, p_k)_A = (\widetilde{p}_j, \widetilde{p}_k)_{\widetilde{A}} = (\widetilde{A}\widetilde{p}_k, \widetilde{p}_j)$, qui s'annule pour $j \neq k$.

Ces substitutions dans 2.4.5 se traduisent en l'algorithme suivant.

2.8.1. Algorithme PCG=gradient conjugué préconditionné.

Choisir $x_0 \in \mathbb{R}^n$,$tol > 0$ *et calculer* $r_0 = b - Ax_0$,
résoudre $Cz_0 = r_0$, *poser* $p_0 = z_0$ *et* $\rho_0 = (z_0, r_0)$
Pour $m = 0, 1, 2, \ldots$ *jusqu'à* $\rho_m/\rho_0 \leq tol^2$
$\alpha_m = \frac{\rho_m}{(Ap_m, p_m)}$, $x_{m+1} = x_m + \alpha_m p_m$, $r_{m+1} = r_m - \alpha_m Ap_m$
résoudre $Cz_{m+1} = r_{m+1}$, $\rho_{m+1} = (z_{m+1}, r_{m+1})$
$\beta_m = \frac{\rho_{m+1}}{\rho_m}$, $p_{m+1} = z_{m+1} + \beta_m p_m$.
Besoins de stockage : cinq vecteurs (on stocke sur place).
Complexité par itération : 2 produits scalaires, 3 saxpies, 1 produit matrice-vecteur, une résolution d'un système avec matrice C.

Comparé à l'algorithme 2.4.5 du gradient conjugué, on stocke un vecteur de plus, et par itération on doit résoudre un système avec C comme matrice de coefficients. On note que PCG 2.8.1 se réduit à CG 2.4.5 dans le cas d'un préconditionnement trivial $C = I$ et alors $r_j = z_j$.

Notons que dans PCG on n'a pas besoin de connaître la décomposition $C = TT^*$ de Choleski. Par exemple, pour $C = A_2^{DF}(1)$ sdp et d'ordre $n = N^2$, on peut se servir du solver rapide décrit dans l'exercice 1.3.8, en complexité $\mathcal{O}(n\log(n))$. Pour un problème de diffusion discrétisé avec $A = A_2^{DF}(\kappa)$, on a vu dans l'exercice 1.2.5(c) et avec la propriété 1.3.9(d) que le conditionnement de $\widetilde{A}$ ne dépend pas de n. Donc $\mathcal{O}(1)$ itérations de PCG suffisent pour obtenir notre solution $\underline{x}$ du système $Ax = b$ avec une précision fixe. Ceci donne en total une complexité $\mathcal{O}(n\log(n))$, bien moins que pour CG ou steepest descent. [5]

Une conclusion similaire est valable pour les éléments finis, au moins si on travaille sur un maillage uniforme comme dans l'exercice 1.4.4, car dans ce cas $C = A_2^{DF}(1) = A_2^{EF}(1)$. Ici, on a vu dans les exercices 1.2.5(c) et 1.4.5 que le conditionnement de $\widetilde{A}$ ne dépend pas de n.

On vient de discuter un cas ou la complexité de la résolution du système $Cz_k = r_k$ détermine la complexité totale de PCG, car elle est légèrement plus élevée que celle d'un saxpy (par un facteur $\log(n)$). Si par contre on veut explicitement travailler avec la décomposition de Choleski $C = TT^*$ il faudra alors choisir T avec au plus $\mathcal{O}(n)$ éléments non nuls. On peut alors résoudre $Cz_k = r_k$ en complexité $\mathcal{O}(n)$ par une descente $Ty = r_k$ en creux suivie d'une remontée $T^*z_k = y$ en creux.

Un préconditionnement d'un système linéaire venant de la discrétisation d'une EDP a été largement discuté dans la littérature. Cependant, pour une matrice creuse générale vérifiant l'hypothèse 2.1.1 il est difficile de donner des préconditionneurs qui fonctionnent toujours. Gene Golub, un fameux mathématicien en algèbre linéaire numérique, disait un jour que la construction d'un préconditionneur pour une matrice donnée rélève plutôt de l'art et pas des maths. Donnons ici deux boîtes noires implémentées par exemple sous Matlab, qui pour une matrice A avec $\mathcal{O}(n)$ éléments non nuls donnent aussi une matrice T avec $\mathcal{O}(n)$ éléments non nuls. Ces boîtes noires fonctionnent assez bien pour des problèmes modèles comme le problème de Poisson discrétisé par différences finis en dimension $d \in \{1, 2\}$.

2.8.2. Préconditionnement SSOR

Dans les méthodes itératives comme Gauss-Seidel, Jacobi, SOR etc on écrit $Ax = b$ comme une équation de point fixe $x = C^{-1}b + (I - C^{-1}A)x$ que l'on résout par une itération de Picard

$$x_{k+1} = C^{-1}b + (I - C^{-1}A)x_k.$$

On sait que cette méthode converge pour tout b et x_0 ssi $\rho(I - C^{-1}A) < 1$, le but est alors de trouver C associé à une méthode itérative avec $\rho := \rho(I - C^{-1}A) \in (0, 1)$ le plus petit possible (au moins pour certaines classes de matrices A comme des M–matrices). D'après l'exercice 1.2.5(c), on peut pour C sdp localiser les valeurs propres de $\widetilde{A}$ dans l'intervalle $[1 - \rho, 1 + \rho]$, avec

$$cond(\widetilde{A}) \leq \frac{1+\rho}{1-\rho}.$$

5. La complexité $\mathcal{O}(n\log(n))$ avec le solveur rapide doit être comparé avec $\mathcal{O}(n^2)$ trouvé pour steepest descent après le théorème 2.3.2, avec $\mathcal{O}(n^{3/2})$ trouvé pour CG après le théorème 2.4.7, et avec $\mathcal{O}(n^{5/4})$ trouvé pour PCG avec SSOR ci-dessous.

Dans la méthode de Jacobi on prend $C = D$, avec D une matrice diagonale comportant les éléments diagonaux de A (qui sont tous > 0 et donc C est sdp). Ici on peut voir pour $A = A_d^{EF}(\kappa)$ ou $A = A_d^{DF}(\kappa)$ que $\widetilde{A}$ se comporte asymptotiquement comme $A_d^{EF}(1)$, ou $A_d^{DF}(1)$, veut dire, avec un préconditionnement diagonal de Jacobi que le nombre d'itérations PCG nécessaires pour atteindre une précision donnée ne dépend pas du coefficient de conductivité κ.

Dans la méthode SSOR on choisit un paramètre $\omega \in]0, 2[$, et on écrit $A = D - E - E^*$ avec D comme avant, et $-E$ strictement triangulaire inférieure ne comportant que les éléments non nuls de A en dessous de la diagonale, et

$$C = \frac{\omega}{2-\omega}(\frac{D}{\omega} - E)D^{-1}(\frac{D}{\omega} - E^*)$$

et alors avec la factorisation de Choleski $D = D^{1/2}D^{1/2}$, $D^{-1/2} = (D^{1/2})^{-1}$,

$$T = \sqrt{\frac{\omega}{2-\omega}}(\frac{D}{\omega} - E)D^{-1/2}.$$

On observe bien que T est triangulaire inférieure, et que $T + T^*$ admet des coefficients non nuls à la même position que A. Aussi, on peut montrer que pour $A = A_2^{DF}(1)$ avec conditionnement $\mathcal{O}(n)$ et ω tel que $2 - \omega$ proportionnel à $1/\sqrt{n}$ on obtient une matrice $\widetilde{A}$ avec un meilleur conditionnement d'ordre $\mathcal{O}(\sqrt{n})$.

2.8.3. Préconditionnement IC(0)=Choleski incomplet.

Ici on construit $C = TT^*$ avec T une matrice triangulaire inférieure et $T + T^*$ des coefficients non nuls à la même position que A en demandant que

$$\forall (j,k) \quad \textit{avec } A_{j,k} \neq 0 : \quad (A - TT^*)_{j,k} = 0.$$

Une telle matrice T peut être calculée par un algorithme de type Choleski en parcourant seulement des éléments non nuls de A, en complexité $O(n)$, au moins si chaque ligne de A ne comporte que $O(1)$ éléments non nuls. Notons que en général T n'est pas le facteur de Choleski de A car ce dernier comporte bien plus d'éléments non nuls (comparer avec le théorème du front qui nous renseigne sur le remplissage des facteurs L et U dans une décomposition LU d'une matrice creuse).

On peut montrer que, pour le problème de Poisson ou de diffusion en dimension 1, la matrice A est tridiagonale et donc $\widetilde{A} = I$, et une itération de PCG donne la solution exacte. En dimension 2 on sait également que PCG avec IC(0) se comporte de la même manière que PCG avec SSOR avec un paramètre optimal ω. Il n'y a pas de paramètres à déterminer pour IC(0), ce préconditionneur est donc généralement préféré.

Nous renvoyons le lecteur intéressé à [S03] pour une étude des variantes de IC(0) comme IC(m) ou alors IC par seuil.

Chapitre 3

Le calcul approché de valeurs propres par des techniques de projection

Notons par $W, W_0, W_1, ... \in \mathbb{C}^{n\times m}$ des matrices à colonnes orthonormées, et par P_W le projecteur orthogonal sur $Im(W)$.

Dans les chapitres suivants on montra que $P_{W_k} \mapsto W_W$ pour $k \to \infty$. Une simple application de l'inégalité triangulaire montre que $W_k \to W$ implique la convergence des projecteurs, mais la réciproque peut être fausse (passez à $(-1)^k W_k$ qui donne le même projecteur). Le théorème suivant donne des formulations équivalentes, par exemple le fait que W^*W_k se comporte pour k grand comme une matrice unitaire. Notons que, par hypothèse, W^*W_k, est de norme $\leq \|W^*\|\,\|W_k\| = 1$, donc la quantité définie dans la partie (d) est bien $\in [0, 1]$ et peut s'écrire comme le cosinus d'un angle [1] dans $[0, \pi/2]$.

3.0.1. Théorème sur convergence de projecteurs.
Soient $W, W_0, W_1, ... \in \mathbb{C}^{n\times m}$ des matrices à colonnes orthonormées. Alors nous avons équivalence entre

(a) $\lim_{k\to\infty} P_{W_k} = P_W$;

(b) $\forall Y \in \mathbb{C}^{n\times m}$ *avec* Y^*W *inversible :* $\lim_{k\to\infty} W_k(Y^*W_k)^{-1} = W(Y^*W)^{-1}$;

(c) $\lim_{k\to\infty} W_k(W^*W_k)^{-1} = W;$.

(d) $\lim_{k\to\infty} \cos(\phi_k) = 1$, *avec* $\cos(\phi_k) := \min_{y\neq 0} \dfrac{\|W^*W_k y\|}{\|y\|}$.

De plus,

$$\sin(\phi_k) \leq \|P_{W_k} - P_W\| \leq 2\sin(\phi_k).$$

1. Si on définit le cosinus de l'angle entre $y \in \mathbb{C}^n$ et un espace vectoriel $\mathcal{K} \subset \mathbb{C}^n$ par

$$\max\{\frac{|(y,z)|}{\|y\|\,\|z\|} : z \in \mathcal{K} \setminus \{0\}\}$$

alors ϕ_k est bien le plus grand des angles entre un élément de $Im(W_k)$ et l'ensemble $Im(W)$.

Bibliographie

[AD08] L. Amodei, J.-P. Dedieu : Analyse numérique matricielle. Cours et exercices corrigés, Dunod, 2008. Au moins moi, je peux y acceder comme livre électronique au Lilliad.

[N00] S. Nicaise : Analyse numérique et équations aux dérivées partielles. Cours et problèmes résolus, Dunod, 2000.

[BR06] C. Brezinski, M. Redivo-Zaglia : Méthodes numériques itératives, Cours et exercices corrigés - Niveau M1, Ellipses, 2006.

[S03] Y. Saad, Iterative Methods for Sparse Linear Systems, Second Edition. SIAM, 2003. La prémière édition est gratuitement disponible ici.

[H19] F. Hecht, Résolution des EDP par la méthode des éléments finis, Cours 5MM30 2016-2017, Master 2, Laboratoire Jacques-Louis Lions, Université Pierre et Marie Curie (2018). lien