Preuve : (Suite)

Il reste à prouver que x^* est l'unique minimiseur sous l'hypothèse $\operatorname{rang} A = n$

Rap : $\forall A \in \mathbb{R}^{m \times n}$:

i) $\operatorname{Ker}(A^TA) = \operatorname{Ker}(A)$

ii) $\operatorname{Im}(A^TA) = \operatorname{Im} A^T$

iii) $\operatorname{Ker}(A) = (\operatorname{Im} A^T)^{\perp}$

Preuve : i) $\supset$: Si $x \in \operatorname{Ker} A$ $Ax = 0 \Rightarrow A^TAx = 0$

$\subset$: Si $A^TAx = 0$, alors $0 = x^TA^TAx = \|Ax\|_2^2$

et $Ax = 0$

iii) $x \in \operatorname{Ker}(A) \iff Ax = 0$

$\Leftrightarrow \forall y \in \mathbb{R}^n,\ y^TAx = 0$

$\Leftrightarrow \forall y \in \mathbb{R}^n\ (A^Ty)^Tx = 0$

$\Leftrightarrow x \in (\operatorname{Im} A^T)^{\perp}$

ii) On applique iii) à i) ■

Comme $\operatorname{Im}(A^TA) = \operatorname{Im} A^T$ et $A^Tb \in \operatorname{Im} A^T$, on obtient que $A^TAx = A^Tb$ possède toujours au moins une solution. On suppose $\operatorname{rang} A = n$. Comme A possède strictement n colonnes, ses colonnes sont linéairement indépendantes et $\operatorname{Ker} A = \{0\}$. En utilisant i), on a que A^TA est carrée à noyeau nul, donc A^TA est inversible et $A^TAx = A^Tb$ possède une unique solution.

Rem: Si A est de rang maximal ($\text{Rang}\, A = n$), alors A^TA est inversible et même sym. déf. positive (seule ?). On peut alors utiliser la décomp. de Cholesky pour résoudre les équations normales.

Algo: Méthode des moindres carrés, équations normales.

Soit $A \in \mathbb{R}^{m \times n}$ avec $m \geq n$ et $\text{rang}\, A = n$ et $\underline{b} \in \mathbb{R}^m$

Pour minimiser $\| A\underline{x} - b \|_2$

1) Construire $B = A^TA$ et le vecteur $\underline{c} = A^T\underline{b}$
2) Calculer la décomp. de Cholesky de B ($B = LL^T$)
3) Résoudre $L\underline{y} = \underline{c}$
4) Résoudre $L^T\underline{x} = \underline{y}$

Nombre opérations : $mn^2 + 1/3\, n^3$

(coût doublé avec algorithme de Gauss)

On va voir d'autres algo mieux conditionnés.

Pour des matrices mal conditionnées, l'algorithme précédent n'est pas bon.

Ex : $A = \begin{bmatrix} 1 & 1 \\ \sqrt{\varepsilon_{mach}} & 0 \\ 0 & \sqrt{\varepsilon_{mach}} \end{bmatrix}$ rang $A = 2$, A rang maximal

$$A^TA = \begin{pmatrix} 1 + \varepsilon_{mach} & 1 \\ 1 & 1 + \varepsilon_{mach} \end{pmatrix}$$

Or $fl(A^TA) = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$ pas inversible

Etudions pour simplifier le cas A carré :

Calculons $K_2(A^TA)$ (Conditionnement de A^TA pour $\|.\|_2$)

$K_2(A^TA) = \|A^TA\|_2 \, \|(A^TA)^{-1}\|_2$

$\|A^TA\|_2 = \lambda_{max}(A^TA) = \|A\|_2^2$, ainsi

$K_2(A^TA) = \|A\|_2^2 \, \|A^{-1}\|_2^2 = (K_2(A))^2$

Ainsi A mal conditionnée $\Rightarrow$ A^TA TRÈS mal conditionné.

Pour $A \in \mathbb{R}^{m \times n}$ (rectangulaire), cette conclusion reste vraie, il faut donc utiliser d'autres algorithmes adaptés au cas A mal conditionnée.

6.2 Factorisation QR :

Déf : Soit $Q \in \mathbb{R}^{m \times n}$, où $m \geq n$. si $Q^T Q = I_n$, alors Q est dite orthonormale.

De plus si $m = n$, Q est orthogonale.

---

le 28.02.19

Thm: Factorisation QR

Soit $A \in \mathbb{R}^{m \times n}$ où $m \geq n$. Il existe une matrice orthonormale $Q \in \mathbb{R}^{m \times n}$ et une matrice triangulaire supérieure $R \in \mathbb{R}^{n \times n}$ tq $A = Q \cdot R$.

Si Rang A = n (rang maximal), alors la décomposition de A est unique en choisissant $r_{jj} > 0 \quad \forall j = 1 \dots n$.

Preuve : Unicité : voir série

Existence : On peut calculer Q et R en utilisant la méthode d'orthogonalisation de Gram-Schmidt sur les vecteurs colonnes de $A = (\underline{a}_1 \; \underline{a}_2 \; \dots \; \underline{a}_n)$

- Rang A = n : les $a_i \in \mathbb{R}^m$ sont linéairement indépendants

on applique G.-S. pour transformer $\underline{a}_1, \dots, \underline{a}_n$ en $q_1, \dots q_n$ orthonormés.

On pose $\tilde{q}_1 = \underline{a}_1$ et $q_1 = \tilde{q}_1 / \| \tilde{q}_1 \|_2$

$\tilde{q}_2 = a_2 - (a_2^T q_1) q_1$ et $\langle \tilde{q}_1, \tilde{q}_2 \rangle = \tilde{q}_1^T \tilde{q}_2 = 0$

et $q_2 = \tilde{q}_2 / \| \tilde{q}_2 \|_2$

Preuve : (Suite)

- Car Rang $A = n$ (Suite)

On a $\tilde{q}_i \neq 0$ car les a_i sont linéairement indépendants.

et par "récurrence" $\tilde{q}_j = a_j - \sum_{i=1}^{j-1} (a_j^T q_i) q_i$ et $q_j = \tilde{q}_j / \|q_j\|_2$

Rem : $x \longmapsto (x^T q_i) q_i$ est une application linéaire, c'est la projection orthogonale sur $\mathrm{Vect}(q_i)$, l'espace vectoriel engendré par q_i

On réécrit de la manière suivante :

$a_1 = \tilde{q}_1 = r_{11} q_1$ avec $r_{11} = \|\tilde{q}_1\|_2$

$a_2 = (a_2^T q_1) q_1 + \tilde{q}_2 = r_{12} q_1 + r_{22} q_2$ avec $r_{22} = \|\tilde{q}_2\|_2$ et $r_{12} = a_2^T q_1$

$\vdots$

$a_j = \sum_{i=1}^{j-1} (a_j^T q_i) q_i + \tilde{q}_j = \sum_{i=1}^{j-1} r_{ij} q_i + r_{jj} q_j$

avec $r_{ij} = a_j^T q_i$ et $r_{jj} = \|\tilde{q}_j\|_2$

Matriciellement, on obtient la factorisation

$A = QR$ avec $Q = (q_1, \dots, q_n)$ et $R = \begin{pmatrix} & r_{ij} & \\ 0 & & \end{pmatrix}$ $j \geq i$

- Cas Rang $A < n$:

Il existe une étape où $\tilde{q}_{L-1} = 0$

c'est-à-dire : $a_L = \sum_{i=1}^{L-1} (a_L^T q_i) q_i + r_{LL} \underbrace{q_L}_{=0}$

On choisit un vecteur unitaire q_L quelconque, mais orthogonal à $q_1, \dots, q_{L-1}$ et on pose $r_{LL} = 0$

On a de nouveau $A = QR$.

Algorithme : Factorisation QR G.-S. classique (pas stable)

1. for $j = 1 \dots n$ do
2. $\tilde{q}_j \longleftarrow a_j$
3. for $i = 1 \dots j-1$ do
4. $r_{ij} \longleftarrow a_j^T q_i$
5. $\tilde{q}_j \longleftarrow \tilde{q}_j - r_{ij} q_i$ ⟵ permet $\tilde{q}_j \perp q_i$
6. $r_{jj} \longleftarrow \|\tilde{q}_j\|_2$
7. $q_j \longleftarrow \tilde{q}_j / r_{jj}$

Cet Algorithme est en fait instable. On peut l'améliorer en calculant :

$$\tilde{q}_j^{(1)} = a_j - (q_1^T a_j) q_1$$

$$\tilde{q}_j^{(i)} = \tilde{q}_j^{(i-1)} - (q_i^T \tilde{q}_j^{(i-1)}) q_i$$

avec $\tilde{q}_j = \tilde{q}_j^{(j-1)}$

au lieu de $\tilde{q}_j = a_j - \sum_{i=1}^{j-1} (q_i^T a_j) q_i$

Rem: Si Rang $A < n$, on remplace l'étape $q_j \leftarrow \tilde{q}_j / \|\tilde{q}_j\|_2$ lorsque $r_{jj} = \|\tilde{q}_j\|_2 = 0$ par :

- choisir $\tilde{q}_j \in \mathbb{R}^m$ au hasard
- $\tilde{q}_j \longleftarrow \tilde{q}_j - \sum_{i=1}^{j-1} (q_i^T \tilde{q}_j) q_i$ ← vect. non-nul avec proba 1
- $q_j \longleftarrow \tilde{q}_j / \|\tilde{q}_j\|_2$

Algo: Factorisation QR, G.-S. modifié (Stable)

$A \in \mathbb{R}^{m \times n}$ $m \geq n$, rang $A = n$

for $i = 1 \dots n$ do

$\tilde{q}_i \longleftarrow a_i$

for $i = 1 \dots n$ do

$r_{ii} \longleftarrow \|\tilde{q}_i\|_2$

$q_i \longleftarrow \tilde{q}_i / r_{ii}$

for $j = i+1 \dots n$ do

$r_{ij} \longleftarrow q_i^T \tilde{q}_j$

$\tilde{q}_j \longleftarrow \tilde{q}_j - r_{ij} q_i$

En l'absence d'erreur d'arrondi, les deux algo sont équivalents.

Coût : donné par $q_i^T \tilde{q}_j$ et $\tilde{q}_j - r_{ij} q_i$

donc $\sum_i \sum_j 4m = 2mn^2$

Retour sur les moindres carrés :

On peut compléter la matrice Q de la décomposition QR pour obtenir $A = \hat{Q}\hat{R}$ avec $\hat{Q} \in \mathbb{R}^{m \times m}$ orthogonale et $\hat{R} = \begin{bmatrix} R \\ 0 \end{bmatrix} \in \mathbb{R}^{m \times n}$

Pour cela on choisit $Q_\perp$ une matrice orthonormale de taille $m \times (m-n)$ dont les vecteurs colonnes sont orthogonaux avec colonnes de Q

ainsi, $\hat{Q} = (Q\ Q_\perp) \in \mathbb{R}^{m \times m}$ est orthogonale.

La factorisation $A = \hat{Q}\hat{R}$ est la factorisation QR complète.

On peut maintenant utiliser cette décomposition pour résoudre le problème des moindres carrés :

$$\|A\underline{x} - \underline{b}\|_2 = \|\hat{Q}\hat{R}\underline{x} - \underline{b}\|_2 = \|\hat{Q}^T(\hat{Q}\hat{R}\underline{x} - \underline{b})\|_2$$

$$= \|\hat{R}\underline{x} - \hat{Q}^T\underline{b}\|_2$$

$$\Rightarrow \|A\underline{x} - \underline{b}\|_2^2 = \left\| \begin{pmatrix} R \\ 0 \end{pmatrix} x - \begin{pmatrix} Q^T \\ Q_\perp^T \end{pmatrix} \underline{b} \right\|_2^2$$

$$= \|Rx - Q^Tb\|_2^2 + \|Q_\perp^T\underline{b}\|_2^2$$

Ainsi, minimiser $\|A\underline{x} - \underline{b}\|$ équivaut à minimiser $\|R\underline{x} - Q^Tb\|_2^2$

Si Rang $A = n$, on a vu que $r_{ii} > 0 \ \forall i$ et $R \in \mathbb{R}^{n \times n}$ est inversible

La solution au problème des moindres carrés est donc donnée par la solution du système triangulaire : $R\underline{x}^* = Q^T\underline{b}$

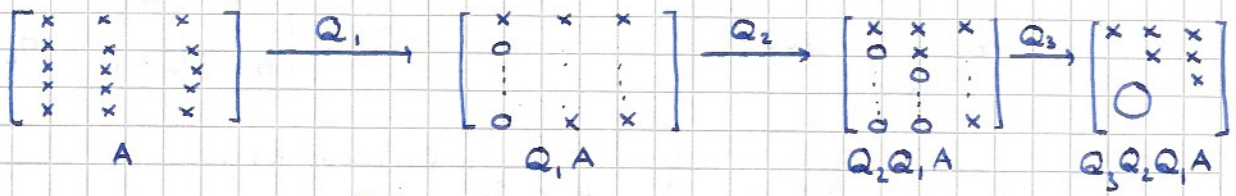
Algorithme : Méthode des moindres carrés en utilisant QR

Soit $A \in \mathbb{R}^{m \times n}$, $m \geq n$, Rang $A = n$

1) Calculer $A = QR$ (réduite)
2) Calculer $\underline{c} = Q^T\underline{b}$
3) Résoudre $R\underline{x} = \underline{c}$

Analyse Numérique : Le 28.02.19

6.3 Réflexions de Householder :



$Q_3 Q_2 Q_1 A = \begin{bmatrix} R \\ 0 \end{bmatrix}$

$A = \hat{Q}\hat{R}$ avec $\hat{R} = \begin{bmatrix} R \\ 0 \end{bmatrix}$ et $\hat{Q} = (Q_3 Q_2 Q_1)^T = Q_1^T Q_2^T Q_3^T$

$\hat{Q}$ orthogonale comme produit de Q_1^T, Q_2^T et Q_3^T orthogonales.

Déf: Soit $v \in \mathbb{R}^n$, avec $\|v\|_2 = 1$, on appelle $H = I - 2vv^T$ la réflexion de Householder associée à v.

$Hv = -v$ et $Hw = w$ si $w \perp v$ ($w^T v = 0$)

---

le 07.03.19

Rem : $H = I - 2vv^T$

Pour la décomposition QR, on cherche v tq $Hx = \alpha e_1$

On a vu en exercice $v = \dfrac{x \pm \|x\|_2 e_1}{\|x \pm \|x\|_2 e_1\|_2}$

Comme H est orthogonale, H préserve la norme $\|Hx\|_2 = \|x\|_2$

ainsi $\alpha = \pm \|X\|_2$.

Ensuite, on remarque $Hx = x - 2v(v^T x) = x + cv$

avec $c = -2v^T x$.

Ainsi, $Hx - x$ et v sont colinéaires ($Hx - x = \pm \|x\|_2 e_1 - x$)

on choisit v colinéaire à $\pm\|x_2\| e_1 - x$ (on choisit $\bigcirc$ de sorte à ne pas faire de soustraction mal conditionée) et on pose

$v = \dfrac{-\text{sign}(x_1)\|x\|_2 e_1 - x}{\|\text{sign}(x_1)\|x\|_2 e_1 + x\|_2}$ ou $v = \dfrac{\text{sign}(x_1)\|x\|_2 e_1 + x}{\|\text{sign}(x_1)\|x\|_2 e_1 + x\|_2}$

$$\begin{pmatrix} x & x & \cdots & x \\ \vdots & & & \\ x & & & x \end{pmatrix} \xrightarrow{Q_1} \begin{pmatrix} x & x & \cdots & x \\ 0 & x & & \\ \vdots & \vdots & & \\ 0 & x & & x \end{pmatrix} \xrightarrow{Q_2} \begin{pmatrix} x & x & & x \\ 0 & x & & \\ & 0 & & \\ 0 & 0 & & x \end{pmatrix}$$

A $\qquad$ $Q_1 A$ $\qquad$ $Q_2 Q_1 A$

$Q_1 = I - 2vv^T$

$v = \dots$

$$Q_2 = \begin{pmatrix} 1 & 0 & \\ 0 & & \\ & H_2 & \end{pmatrix} \qquad H_2 \in \mathbb{R}^{(n-1)\times(n-1)}$$

$$Q_k = \begin{pmatrix} I_{k-1} & \\ & H_k \end{pmatrix} \qquad H_k \in \mathbb{R}^{(n-k+1)\times(n-k+1)}$$

En pratique, on peut ne pas calculer Q explicitement, mais la liste des vecteurs v utilisés pour définir les matrices de Householder.

Algo: Factorisation QR - Householder

Soit $A \in \mathbb{R}^{m\times n}$, $m \geq n$, et $\text{rang}(A) = n$

for $k = 1 \dots n$ do

$x \longleftarrow a_{k:m,k}$

$v_k \longleftarrow x + \text{sign}(x_1) \|x\|_2 e_1$

if $\|v_k\|_2 > 0$ then

$v_k \longleftarrow v_k / \|v_k\|_2$

for $i = k \dots n$ do

$a_{k:m,i} \longleftarrow a_{k:m,i} - 2 (v_k^T a_{k:m,i}) v_k$

$R \longleftarrow A_{1:n,1:n}$

Rem: On a évité de construire H_k et Q_k explicitement car ceci représente $O((m-k)^2)$ opérations.

Coût de l'algo:

$$\sum_{k=1}^{n} 4(m-k+1)(n-k+1) \simeq 4 \int_1^n (m-x)(n-x)\,dx$$

$$\text{Coût} = \sim 2mn^2 - \frac{2}{3}n^3$$

Rem : Pour calculer Q (pas nécessaire), on calcule

$$Q = (\hat{Q}e_1, \dots, \hat{Q}e_n) \quad \text{où} \quad \hat{Q}e_k = Q_1 \dots Q_n e_k$$

$$Q_k = \begin{pmatrix} I_{k-1} & 0 \\ 0 & H_k \end{pmatrix}$$

Chapitre 7 : Équations non-linéaires.

En pratique, on veut souvent résoudre un système non-linéaire d'équations :

$$f(x^*) = 0 \quad \text{avec} \quad f : \mathbb{R}^n \longrightarrow \mathbb{R}^m \quad x^* \in \mathbb{R}^n$$

En général, il n'existe pas d'algo avec un nbr fini d'étape.

On va considérer des méthodes itératives (nombre infini d'étape) mais on ajoute un critère d'arrêt et on obtient une solution approchée.

7.1 Méthode de la bisection :

On considère $f : \mathbb{R} \to \mathbb{R}$ (cas scalaire) supposée continue.

On cherche une racine (pas forcément unique) de f.

Thm : thm des valeurs intermédiaires

Pour une fct continue $f : [a,b] \to \mathbb{R}$, si $f(a) \cdot f(b) < 0$

$\exists\, x^* \in (a,b)$ tq $f(x^*) = 0$

Algo : Méthode de la bisection

Soit $f : [a,b] \to \mathbb{R}$ continue avec $f(a)f(b) < 0$

$a^{(1)} \leftarrow a \qquad b^{(1)} \leftarrow b$

for $k = 1, \dots$

$$x^{(k)} = \frac{a^{(k)} + b^{(k)}}{2}$$

if $f(x^{(k)}) f(a^{(k)}) < 0$

$a^{(k+1)} \leftarrow a^{(k)} \qquad b^{(k+1)} \leftarrow x^{(k)}$

else if $f(x^{(k)}) f(b^{(k)}) < 0$

$a^{(k+1)} \leftarrow x^{(k)} \qquad b^{(k+1)} \leftarrow b^{(k)}$

else Arrêt car $f(x^{(k)}) = 0$ on a trouvé une racine.

Un critère d'arrêt possible : $b^{(k)} - a^{(k)} <$ tolérance

avec tolérence, une précision donnée.

On a alors $x^{(k)} \simeq x^*$ approximation à tolérance près.

Estimation de l'erreur $e_k := |x^* - x^{(k)}|$

On a $e_k \leq \frac{1}{2}(b^{(k)} - a^{(k)}) = \frac{1}{2}\cdot\frac{1}{2}(b^{(k-1)} - a^{(k-1)}) = \dots = \left(\frac{1}{2}\right)^k (b-a) \xrightarrow[k\to\infty]{} 0$

Ainsi, la méthode est convergente $x^{(k)} \xrightarrow{k\to+\infty} x^*$

On peut estimer le nbr d'itération en fonction de la tolérance

Pour avoir $|x^{(k)} - x^*| <$ tol (tol = $\varepsilon_{mach}(b-a)$)

il faut $\frac{1}{2^k}(b-a) \leq$ tol, $k \geq \log_2(b-a) - \log_2(\text{tol})$

$= \log(b-a/\text{tol}) \,/\, \log(2)$

7.2 Méthode de points fixes :

On calcule une itération $x_{k+1} = \phi(x_k)$ avec ϕ choisie

tq $x^* = \phi(x^*)$ (x^* est pt. fixe de $\phi : \mathbb{R}^n \to \mathbb{R}^n$)

pour résoudre le problème $f(x^*) = 0$

---

le 14.03.19

But : résoudre une équation non-linéaire $f(x^*) = 0$

$x^* \in \mathbb{R}^n$ avec $f : \mathbb{R}^n \to \mathbb{R}^n$.

Idée : on introduit $\phi : \mathbb{R}^n \longrightarrow \mathbb{R}^n$ de sorte que l'itération

de point fixe $x_{k+1} = \phi(x_k)$ converge vers x^*

pour $k \longrightarrow \infty$:

Ex : $\phi(x) = x - f(x)$ ou même $\phi(x) = x - Bf(x)$ $B \in \mathbb{R}^{n\times n}$ inversible

Cette méthodologie s'appelle une méthode de point fixe.

Rem : Si ϕ est continue et $(x_k)_{k \in \mathbb{N}}$ converge, alors la limite est un point fixe de ϕ.

On verra que certains algo convergent ou non et la vitesse de convergence peut différer en fonction du choix de l'algo.

Ex : $f(x) = \cos x \qquad x^* = \pi/2$

$\phi_1(x) = x + \cos x$ convergence

$\phi_2(x) = x + \frac{1}{2} \cos x$ " moins rapide

$\phi_3(x) = x + 10 \cos x$ divergence

cond. initiale : $x_0 = 1$.

4.3 Thm du pt fixe de Banach.

Il permet la convergence quelque soit la condition initiale.

Déf : Soit $f : A \subset \mathbb{R}^n \longrightarrow \mathbb{R}^m$, on dit que f est L-Lipschitzienne si $\forall x, y \in A \quad \|f(x) - f(y)\| \leq L \cdot \|x - y\|$

Rem : La valeur de L dépend de la norme choisie.

Rem : f Lipschitzienne $\Rightarrow$ f continue.

réciproque fausse ($f(x) = \sqrt{x}$ sur $\mathbb{R}_+$)

Ex : Si f est C^1 avec $\|f'(x)\| \leq L$ alors f est L-Lipschitzienne

Déf : On dit que $f : A \subset \mathbb{R}^n \longrightarrow \mathbb{R}^m$ est une contraction si elle est L-Lipschitzienne avec $L < 1$.

Thm: Point fixe de Banach 1922

Soit $\Phi : A \longrightarrow A$ une L-contraction avec $A \subset \mathbb{R}^n$ fermé non-vide

alors :

1) $\exists!\ x^* = \Phi(x^*)$ $x^* \in A$ (unique pt fixe)

2) la suite $x_{k+1} = \Phi(x_k)$ converge vers x^* $\forall\ x_0 \in A$

de + $\|x_k - x^*\| \leq L^k \|x_0 - x^*\| \leq \dfrac{L^k}{1-L} \|x_1 - x_0\|$ $k \geq 0$

Preuve: • Existance de x^* :

On considère la suite $x_{k+1} = \Phi(x_k) \in A$

mq elle est de Cauchy : $\|x_{k+1} - x_k\| = \|\Phi(x_k) - \Phi(x_{k-1})\|$

$\leq L \|x_k - x_{k-1}\|$

$\leq L^k \|x_1 - x_0\|$

Donc $\|x_{k+m} - x_k\| \leq \|x_{k+m} - x_{k+m-1}\| + \dots + \|x_{k+1} - x_k\|$

$\leq (L^{k+m-1} + \dots + L^k) \|x_1 - x_0\|$

$= (L^{m-1} + \dots + 1) L^k \|x_1 - x_0\|$

$L<1 \searrow$ $\leq \left(\sum_{j=0}^{\infty} L^j\right) L^k \|x_1 - x_0\|$

$= \dfrac{L^k}{1-L} \|x_1 - x_0\| \longrightarrow 0$

Soit $\varepsilon > 0$. Comme $\dfrac{L^k}{1-L} \xrightarrow{k \to \infty} 0$ il existe un rang $K \in \mathbb{N}$

tq $\forall\ k \geq K$ $\dfrac{L^k}{1-L} \|x_1 - x_0\| < \varepsilon$ et ainsi $\forall\ k \geq K$, $\forall m \geq 1$

$\|x_{k+m} - x_k\| \leq \varepsilon$

et $(x_k)_{k \in \mathbb{N}}$ est une suite de cauchy à valeur dans

$\mathbb{R}^n$ (complet) donc convergente vers x^* et $x^* \in A$

car A fermé

$\|x_k - x^*\| = \|\Phi(x_{k-1}) - \Phi(x^*)\| \leq L \|x_{k-1} - x^*\|$

$\leq \quad \leq L^k \|x_0 - x^*\|$

• Unicité : Soit x et y 2 pt fixes de Φ

$\|x - y\| = \|\Phi(x) - \Phi(y)\| \leq L \|x - y\|$

$(1-L) \|x - y\| \leq 0 \Rightarrow \|x - y\| = 0 \Rightarrow x = y$ $\blacksquare$ $_{>0}$

4.4 Taux de Convergence :

Déf : Soit $x_k \xrightarrow{k\to\infty} x^*$ une suite convergente. On dit que la convergence est d'ordre (au moins) m si $\exists$ un constante C tq

$$e_{k+1} := \|x_{k+1} - x^*\| \leq C \|x_k - x^*\|^m$$

$$e_{k+1} \leq C e_k^m \quad \forall k \text{ suffisament grand}$$

On dit aussi que la convergence est :

- linéaire si $m = 1$ et $C < 1$
- superlinéaire si $m > 1$
- quadratique $m = 2$
- cubique $m = 3$

Ex : On considère les suites suivantes :

a) $e_{k+1} = \frac{1}{10} e_k$ $\quad e_0 = 0.05$ linéaire

b) $e_{k+1} = \frac{1}{10} e_k + 10 e_k^2$ $\quad e_0 = 0.05$ linéaire

c) $e_{k+1} = 10 e_k^2$ $\quad e_0 = 0.09$. superlinéaire.

Rem : On suppose $e_k > 0 \ \forall k$

$$\log e_{k+1} \leq \log C + m \log e_k$$

pour $m = 1$, on obtient $\log e_k \leq \log C + \log e_k$.

Si on se place sous les hypothèses du thm du pt fixe de Banach, on a $e_k \leq C^k e_0$ ($C < 1$), la convergence est (au moins) linéaire. Mais elle peut être plus rapide.

On peut calculer l'ordre de convergence avec un développement de Taylor, avec Φ régulière. $e_k = x_k - x^*$

$$\Phi(x_k) = \Phi(x^* + e_k) = \Phi(x^*) + \Phi'(x^*) e_k + \frac{1}{2} \Phi''(x^*)(e_k, e_k) + \underbrace{\frac{1}{2} \sum_{i,j=1}^{n} \frac{\partial \Phi}{\partial x_i \partial x_j} (e_k)_i (e_k)_j}_{\|\cdot\|} + O(\|e_k\|^3)$$

Rem: $\left(\Phi^{(m)}(x^*)(a_1,\dots,a_m)\right)_i$

$$= \sum_{i_1,\dots i_m=1}^{n} \frac{\partial \Phi_i(x^*)}{\partial x_{i_1}\dots\partial x_{i_m}} (a_1)_{i_1}\dots(a_m)_{i_m}$$

$\Phi^{(m)}(x^*) : \underbrace{\mathbb{R}^n \times \mathbb{R}^n \times \dots \times \mathbb{R}^n}_{m \text{ fois}} \longrightarrow \mathbb{R}^n$ application m-linéaire

On suppose $x_k \xrightarrow{k\to\infty} x^*$, ie: $e_k \xrightarrow{k\to\infty} 0$

- si $\Phi'(x^*) \neq 0$ et $c = \|\Phi'(x^*)\| < 1$ alors

  $e_{k+1} = \Phi(x_k) - \Phi(x^*) = \Phi'(x_k)e_k + O(\|e_k\|^2)$

  $\|e_{k+1}\| \leq (C_* + O(\|e_k\|))\|e_k\|$

  La convergence est donc linéaire, car $C_* + O\|e_k\| < 1 \ \forall k$ grd

- si $\Phi'(x^*) = 0$, mais $\Phi''(x^*) \neq 0$ alors

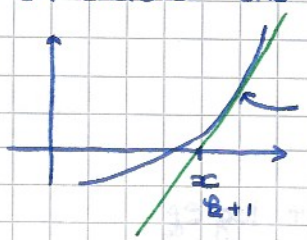
  $e_{k+1} = \frac{1}{2}\Phi''(x^*)(e_k, e_k) + O(\|e_k\|^3)$

  $\|e_{k+1}\| \leq C\|e_k\|^2 \quad \forall k$ assez grand.

  La convergence sera quadratique.

4.5 Méthode de Newton scalaire:

On cherche une racine x^* de $f: \mathbb{R}\to\mathbb{R}$ supposée régulière.



$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{1}{2} f''(x_0)(x - x_0)^2 + \dots$

(e.g la tangeante à f en x_0)

$0 = f(x_0) + f'(x_0)(x_1 - x_0)$

(de manière équivalente, si $f'(x_0) \neq 0$, $x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$

Analyse Numérique : le 14.03.19

Méthode : de Newton dans $\mathbb{R}^n$ pour résoudre $f(x^*) = 0$

$f : \mathbb{R}^n \to \mathbb{R}^n$ régulière.

On définit $(x_k)_k$ par l'itération :

$0 = f(x_k) + f'(x_k)(x_{k+1} - x_k)$ ou de manière équivalente

$x_{k+1} = x_k - f'(x_k)^{-1} f(x_k) = \Phi(x_k)$

---

le 21.03.19

Rem : Si f est C^1 au voisinage de x^* avec $f'(x^*) \neq 0$, alors $f'(x)$ reste non-nulle au voisinage de x^* par continuité de f'

(Prendre $\varepsilon = \left| \frac{f'(x^*)}{2} \right|$ dans la déf de continuité de f')

4.6 Méthode de Newton vectorielle :

Soit $f : \mathbb{R}^n \to \mathbb{R}^n$ régulière. L'idée est d'approcher f par une approximation linéaire en x_k :

$f(x) \simeq f(x_k) + f'(x_k)(x_{k+1} - x_k)$

On calcule l'itération $(x_k)_{k \in \mathbb{N}}$ par récurrence

$0 = f(x_k) + f'(x_k)(x_{k+1} - x_k)$

système linéaire d'inconue x_{k+1} qui possède une solution si $f'(x_k)$ est inversible.

De manière équivalente, $x_{k+1} = x_k - f'(x_k)^{-1} f(x_k)$

(on évite de calculer $f'(x_k)^{-1}$)

Rem : Si f est C^1 au voisinage de x^* et $f'(x^*)$ inversible alors $f'(x)$ est encore inversible pour x au voisinage de x^*

Pv : $x \longmapsto \det(f'(x))$ est continue au voisinage de x^* et non-nulle

Cette applic. reste donc non-nulle dans le voisinage et aussi inversible.

Lemme: Soit $f: \mathbb{R}^n \longrightarrow \mathbb{R}^m$, C^1 sur $D \subset \mathbb{R}^n$ ouvert convexe.

On suppose que f' (jacobienne) est lipschitzienne sur D.

$\forall x, y \in D \quad \|f'(x) - f'(y)\| \leq L \|x - y\|$

Alors $\| f(x) + f'(x)(y-x) - f(y) \| \leq \frac{L}{2} \|x - y\|^2$

Pv: On pose $\varphi : [0,1] \longrightarrow \mathbb{R}^m$
$t \longmapsto f(ty + (1-t)x)$

φ est C^1 et on a $\varphi'(t) = f'(ty + (1-t)x) \cdot (y - x)$

On pose $w(x,y) = f(x) + f'(x)(y-x) - f(y)$

$\|w(x,y)\| = \|\varphi(1) - \varphi'(0) - \varphi(0)\| = \left\| \int_0^1 (\varphi'(t) - \varphi'(0))\, dt \right\|$

$\leq \int_0^1 \|\varphi'(t) - \varphi'(0)\|\, dt \leq (*)$

$\|\varphi'(t) - \varphi'(0)\| = \|[f'(ty + (1-t)x) - f'(x)](y-x)\|$

$\leq L \cdot t \|y - x\| \, \|y - x\|$

$(*) = \frac{L}{2} \cdot \|y - x\|^2$ ▨

Rem: Si f est C^2 au voisinage de x^*, alors f est lipschitzienne au voisinage de x^*, (et $L = \sup_{x \in V} \|f''(x)\|$) et on peut appliquer le lemme précédent.

Rem: Si f est C^1 au vois de x^* alors $x \longmapsto (f'(x))^{-1}$ est bien définie, continue et bornée au vois de x^*.

Pv:
- Bien déf: pv page 73
- Continue: Formule de Cramer ($Ax = b$, fraction rationelle)
- Borné: Dans le voisinage.

Thm : Soit $f : \mathbb{R}^n \to \mathbb{R}^n$ C^1 sur $D \subset \mathbb{R}^n$ ouvert convexe et on suppose f' L-lipschitzienne sur D,

$$\forall x, y \in D, \ \|f'(y) - f'(x)\| \leq L \cdot \|y - x\|$$

Soit $x^* \in D$, avec $f(x^*) = 0$ et on suppose $f'(x^*)$ inversible. Alors pour x_0 donnée initiale assez proche de x^*, la méthode de Newton converge de manière quadratique

$$\forall k, \ \|x_{k+1} - x^*\| \leq C \|x_k - x^*\|^2$$

avec C indépendante de k.

Preuve : On suppose $x_k \in D$. comme $f(x^*) = 0$, d'après la formule

$$f(x_k) = f(x_k) - f(x^*) = f'(x_k)(x_k - x^*) + w(x_k, x^*)$$

et $\|w(x_k, x^*)\| \leq \frac{L}{2} \|x_k - x^*\|^2$

Ainsi, $x_{k+1} - x^* = x_k - x^* - (f'(x_k))^{-1} f(x_k)$

$$x_{k+1} - x^* = -(f'(x_k))^{-1} w(x_k, x^*)$$

$$\|x_{k+1} - x^*\| \leq \|(f'(x_k))^{-1}\| \, \|w(x_k, x^*)\|$$

$$\leq \tilde{\beta} \cdot \tfrac{L}{2} \|x_k - x^*\|$$

Il reste à montrer que x_{k+1} reste dans un "bon" voisinage et que la méthode converge.

On choisit x_0 tq $\tilde{\beta} \frac{L}{2} \|x_0 - x^*\| \leq 1/2$

Ainsi, on obtient par récurrence sur k que

$$\|x_{k+1} - x^*\| \leq \frac{1}{2} \|x_0 - x^*\|$$

et x_{k+1} reste dans un vois. où les majorants $\leq$ sont valables.

On obtient de même $\|x_{k+1} - x^*\| \leq 1/2 \|x_k - x^*\|$ et $(x_k)_{k \in \mathbb{N}}$ converge vers x^*.

Rem:
- On peut mq si $f''(x^*)$ est non-nul, alors la convergence est exactement quadratique, sinon elle sera au moins cubique.
- Si $f: \mathbb{R}^n \longrightarrow \mathbb{R}^n$ est linéaire alors, $x_1 = x^*$, la méthode converge en une seule itération $\forall k \geq 1 \quad x_k = x^*$.
- Si f est C^1 mais f' non Lipschitzienne, on peut montrer une convergence super linéaire : $\|x_{k+1} - x^*\| = o(\|x_k - x^*\|)$
- La question de la conv. globale, sans supposer x_0 proche de x^*, est très difficile en général.
- En général, $A(x^*) := \{ x_0 \in \mathbb{R}^n ; \; x_k \xrightarrow{k \to \infty} x^* \}$ est fractal.

7.7 Modification de la méthode de Newton:

En grande dimension, résoudre les systèmes linéaires de la méthode de Newton peut être coûteux en général.

Système lin $n \times n$ à résoudre $\longrightarrow$ n^3 opérations

Pour réduire le coût, on peut utiliser l'approximation $f'(x_k)^{-1} \simeq (f'(x_0))^{-1}$ (même jacobienne pour toutes les opérations)

Ainsi, on calcule $f'(x_0)$ une fois et sa décomposition LU (coût $O(n^3)$ opérations), ensuite on résout les systèmes lin de chaque itérations avec $O(n^2)$ opérations.

Ainsi, le coût est réduit de $O(kn^3)$ à $O(n^3 + kn^2)$ opérations où k est le nbr d'itérations de la méthode de Newton.

Le thm précédent ne s'applique plus, la convergence est seulement linéaire (mais souvent encore rapide en pratique).

Analyse Numérique :

le 21.03.19

Une autre variante, on peut approcher $f'(x_k)$ par des différences divisées.

$$\partial f_i/\partial x_j\,(x_k) = \lim_{h \to 0} \frac{f_i(x_k + h e_j) - f_i(x_k)}{h}$$

avec $f = \begin{pmatrix} f_1 \\ \vdots \\ f_n \end{pmatrix}$ et $e_j = \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} \leftarrow j$

Dans un algorithme, on fixe $h > 0$ et on approche

$$\partial f_i/\partial x_j \simeq \frac{f_i(x_k + h e_j) - f_i(x_k)}{h} = \delta f_i [x_k, x_k + h e_j]$$

Un développement de Taylor de $g : \mathbb{R} \to \mathbb{R}$ C^3 donne :

$$1/h [g(x + h) - g(x)] = g'(x) + \frac{h}{2} g''(x) + O(h^2)$$

h doit être choisit petit pour avoir une erreur petite.

⚠ h ne doit pas être trop petit, car la soustraction est mal conditionée

On obtient avec les erreurs d'arrondis :

$$\frac{g(x + h) - g(x)}{h} = g'(x) + O(h) + O\left(\frac{eps}{h}\right)$$

Bon choix $h = \sqrt{eps}$

---

le 28.03.19

Retour sur la formule de différences finies :

$$D = \frac{1}{h}(g(x + h) - g(x)) = g'(x) + \underbrace{\frac{h}{2} g''(x)}_{\text{erreur de discrétisation}} + O(h^2)$$

$\uparrow$ erreur d'arrondis $O\left(\frac{eps}{h}\right)$

erreur $\hat{D} - g'(x) = O\left(\frac{eps}{h}\right) + O(h)$

un bon choix : $eps/h \simeq h$ $\quad h \sim \sqrt{eps}$

$$\hat{D} = \frac{1}{h}(g((x + h)(1 + \varepsilon_1))(1 + \varepsilon_2) - g(x)(1 + \varepsilon_3))(1 + \varepsilon_4)$$

(en supposant $K(g) = 1$ condition de g)

$$\hat{D} \simeq \frac{g(x + h) - g(x)}{h} + 1/h\,(g'(x + h)(x + h)\varepsilon_1) + g(x + h)(\varepsilon_2 + \varepsilon_4)$$

$$- g(x)(\varepsilon_3 + \varepsilon_4) + O(eps^2)$$

$$|\hat{D} - g'(x)| = \left|\frac{1}{h}\left(g'(x+h)(x+h)\varepsilon_1 + g(x+h)(\varepsilon_2+\varepsilon_4) - g(x)(\varepsilon_3+\varepsilon_4)\right)\right.$$
$$\left. + h/2\, g''(x) + O(eps^2) + O(h^2)\right|$$
$$= O(eps/h) + O(h)$$

7.8 Minimisation et méthode de Gauss - Newton :

On considère $r: \mathbb{R}^n \to \mathbb{R}^m$ avec $m > n$ une fonction régulière et on cherche une solution de $r(x) = 0$

Pas de solution en général, on considère donc le problème des moindres carrés : $\|r(x)\|_2 = \sqrt{\sum_{i=1}^{m} r_i(x)^2} \longrightarrow \min$

avec $r(x) = \begin{pmatrix} r_1(x) \\ \vdots \\ r_m(x) \end{pmatrix}$

On pose $f(x) = \|r(x)\|_2^2$ qui est C^2 si r est C^2.

(car $x \longmapsto \|x\|_2^2$ est C^∞)

On remarque que x^* minimum de $f \Rightarrow f'(x^*) = 0$

[$f: \mathbb{R}^n \to \mathbb{R}$, $f': \mathbb{R}^n \longrightarrow \mathbb{R}^{1\times n}$]

Rap : $f'(x) = \left(\partial f/\partial x_1 (x), \ldots, \partial f/\partial x_n (x)\right)$

$\nabla f(x) = (f'(x))^T = \begin{pmatrix} \partial f/\partial x_1 (x) \\ \vdots \\ \partial f/\partial x_n (x) \end{pmatrix} \in \mathbb{R}^n$

Ainsi, on obtient $\nabla f(x^*) = 0$ $\quad \nabla f: \mathbb{R}^n \longrightarrow \mathbb{R}^n$

un système linéaire avec autant d'équations que d'inconnues.

On peut appliquer la méthode de Newton à ce système en supposant f C^3 au voisinage de x^* et la matrice Hessienne $\nabla^2 f(x^*)$ inversible $(= (\nabla f(x^*))')$

$$\nabla^2 f(x) = \begin{pmatrix} \partial^2 f/\partial x_1^2 & & \partial^2 f/\partial x_n \partial x_1 \\ & & \\ \partial^2 f/\partial x_1 \partial x_n & & \partial^2 f/\partial x_n^2 \end{pmatrix}$$

Et on écrit l'itération de Newton :

$$x_{k+1} = x_k - (\nabla^2 f(x_k))^{-1} \nabla f(x_k)$$

Analyse Numérique :

le 28.08.19

On pose $f(x) = \frac{1}{2}\|R(x)\|_2^2 = \frac{1}{2}\|r(x)\|_2^2 = \frac{1}{2}(r_1(x)^2 + \dots + r_m(x)^2)$

$$(\nabla f(x))_i = \frac{\partial f(x)}{\partial x_i} = \sum_{k=1}^{m} \frac{\partial r_k(x)}{\partial x_i} \cdot r_k(x)$$

Matriciellement, $\nabla f(x) = r'(x)^T r(x) = \nabla r(x)\, r(x)$

avec $r'(x) = \begin{pmatrix} \partial r_1/\partial x_1 & \partial r_1/\partial x_n \\ \partial r_m/\partial x_1 & \partial r_m/\partial x_n \end{pmatrix} \in \mathbb{R}^{m \times n}$

$$(\nabla^2 f(x))_{i,j} = \frac{\partial^2 f}{\partial x_i \partial x_j}(x) = \frac{\partial}{\partial x_j}\left(\frac{\partial f}{\partial x_i}(x)\right) = \sum_{k=1}^{m}\left(\frac{\partial r_k(x)}{\partial x_i \partial x_j} r_k(x) + \frac{\partial r_k(x)}{\partial x_i} \cdot \frac{\partial r_k(x)}{\partial x_j}\right)$$

Matriciellement, $\nabla^2 f(x) = H(x) + B(x)$ avec

$$B(x) = r'(x)^T r'(x) \qquad H(x) = \sum_{k=1}^{m} \nabla^2 r_k(x)\, r_k(x)$$

où $\nabla^2 r_k(x) = \begin{pmatrix} \frac{\partial^2 r_k}{\partial x_1^2} & \partial^2 r_k/\partial x_n \partial x_1 \\ \frac{\partial^2 r_k}{\partial x_1 \partial x_n} & \frac{\partial^2 r_k}{\partial x_n^2} \end{pmatrix}$

Cor : Soit $f : \mathbb{R}^n \to \mathbb{R}$ de classe C^2 sur $D \subset \mathbb{R}^n$ ouvert convexe et

$\|\nabla^2 f(x) - \nabla^2 f(y)\| \leq L \cdot \|x - y\| \quad \forall x, y \in D$

Soit $\nabla f(x^*) = 0$ avec $\nabla^2 f(x^*)$ inversible.

Alors pour x_0 assez proche de x^*, l'itération

$$x_{k+1} = x_k - (\nabla^2 f(x_k))^{-1} \nabla f(x_k)$$

converge vers x^* et $\|x_{k+1} - x^*\| \leq C \cdot \|x_k - x^*\|^2$

avec C indépendant de $k = 0, 1, 2, \dots$

Pv : C'est la méthode de Newton standard appliquée à $\nabla f(x^*) = 0$ □

Pour notre problème de moindre carré, cette itération s'écrit

$$x_{k+1} = x_k - (H(x) + B(x))^{-1} r'(x)\, r(x)$$

Rem : Pour réduire le coût, on peut négliger $H(x)$ et on obtient alors une nouvelle méthode.

(Gauss - Newton)

Rem: Une autre interprétation est d'approcher à chaque étape $r(x_k)$ par une approximation linéaire :

$$r(x_{k+1}) \approx r(x_k) + r'(x_k)(x_{k+1} - x_k)$$

et on considère $\| r(x_k) + r'(x_k)(x_{k+1} - x_k) \|_2 \longrightarrow \min$

ce qui définit x_{k+1} comme solution d'un problème de moindre carré linéaire.

C'est la méthode des moindres carrés appliquée au système sur-déterminé :

$$\overbrace{r'(x_k)}^{A} \overbrace{(x_{k+1} - x_k)}^{x} \approx \overbrace{-r(x_k)}^{b}$$

P_k

La solution est donnée par les équations normales

$$B(x_k)(x_{k+1} - x_k) = -\nabla f(x_k)$$

$$A^T A x = A^T b$$

Déf: Méthode de Gauss - Newton

x_0 approximation initiale

$\| r'(x_k) p_k + r(x_k) \|_2 \longrightarrow \min$ (moindre carré linéaire d'inconnue p_k)

$$x_{k+1} := x_k + p_k$$

Lemme: On suppose $r : \mathbb{R}^n \longrightarrow \mathbb{R}^m$ de classe C^3 $(m \geq n)$.

Soit $\nabla f(x^*) = r'(x^*) r(x^*) = 0$ $\left(f(x) = \frac{1}{2} \| r(x) \|_2^2 \right)$

avec rang $r'(x^*) = n$ (rang maximal), alors pour x_0 assez proche de x^*, l'erreur de la méthode de Gauss - Newton vérifie :

$$\| x_{k+1} - x^* \| \leq \left(\| B(x_k)^{-1} H(x_k) \| + C \| x_k - x^* \| \right) \| x_k - x^* \|$$

Une conséquence de ce lemme est la conv. superlinéaire de la méthode dans les cas suivants :

- si $r(x^*) = 0$, c-à-d le système sur déterminé $r(x) = 0$ possède une solution exacte.
- Si $r''(x^*) = 0$, c-à-d la fonction $r(x)$ n'a pas de courbure en x^*

En effet, dans ce cas on a : $B(x)^{-1} H(x) \xrightarrow[x \to x^*]{} 0$

car $B(x^*)$ est inversible (donc aussi $B(x)$ au vois de x^*)

et $H(x) \xrightarrow{x \to x^*} 0$.

On obtient $\| x_{k+1} - x^* \| = o(\| x_k - x^* \|)$

Si $r(x^*)$ est assez proche de 0, alors $B(x)^{-1} H(x)$ est bornée par une constante < 1 au voisinage de x^* et on obtient

$\| x_{k+1} - x^* \| \leq c \cdot \| x_k - x^* \|$ avec $c < 1$

La convergence est linéaire.

Si $r(x^*)$ est trop grand, on a divergence en général.

$$\| x_{k+1} - x^* \| \leq \underbrace{(\| B(x_k)^{-1} H(x_k) \| + C \| x_k - x^* \|)}_{\geq 1 \ ?} \| x_k - x^* \|$$

Pv : du Lemme

On fait comme dans la preuve de la méthode de Newton appliquée à $\nabla f(x^*) = 0$.

Comme r est C^2, ∇f est C^2 et $\nabla f'$ est lipschitzienne au vois. de x^*. De plus $B(x)$ est inversible au vois de x^* car $B(x^*) = r'(x^*)^T r'(x^*)$ est inversible.

$$\nabla f(x_k) = \nabla f(x_k) - \nabla f(x^*) = \left(B(x_k) + H(x_k) \right) (x_k - x^*) + O(\| x_k - x^* \|^2)$$

$$\begin{aligned} x_{k+1} - x^* &= x_k - x^* - B(x_k)^{-1} \nabla f(x_k) \\ &= (B(x_k))^{-1} \left(B(x_k)(x_k - x^*) - \nabla f(x_k) \right) \\ &= (B(x_k))^{-1} \left(H(x_k)(x_k - x^*) + O(\| x_k - x^* \|^2) \right) \end{aligned}$$

Rv : (Suite)

Ainsi, $\exists C$ tq $\|x_{k+1} - x^*\| \leq \|B(x_k)^{-1} H(x_k)\| \|x_k - x^*\| + C\|x_k - x^*\|^2$ □

le 04.04.19

Ch 8 : Equations différentielles ordinaires :

But : Résolution numérique de système d'EDO

$$\begin{aligned} y_1' &= f_1(y_1, \dots, y_m) & y_1(t_0) &= y_{10} \\ &\vdots \\ y_m' &= f_m(y_1, \dots, y_m) & y_m(t_0) &= y_{m0} \end{aligned}$$

d' inconnues $y_1, \dots, y_m : \mathbb{R} \to \mathbb{R}$

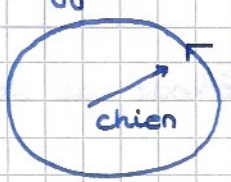
Vectoriellement $y' = f(t, y)$ $\quad y(t_0) = y_0$ (Cond. initiale)

où $y = \begin{pmatrix} y_1 \\ y_m \end{pmatrix} : \mathbb{R} \to \mathbb{R}^m$ et $f : \mathbb{R} \times \mathbb{R}^m \longrightarrow \mathbb{R}^m$ (champs de vecteurs)

Rem: En général, pas de solution explicite / exacte disponible.

On doit utiliser des méthodes numériques.

Ex : Jogger et le chien



$\text{Jog} : \begin{pmatrix} \xi(t) \\ \eta(t) \end{pmatrix} = \begin{pmatrix} \cos vt \\ \sin vt \end{pmatrix}$ v vitesse fixée

$\text{Chien} : \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} \quad \begin{pmatrix} x'(t) \\ y'(t) \end{pmatrix}$

w = vitesse chien fixée

$$\begin{pmatrix} x'(t) \\ y'(t) \end{pmatrix} = \frac{w}{\sqrt{(\xi(t) - x(t))^2 + (\eta(t) - y(t))^2}} \begin{pmatrix} \xi(t) - x(t) \\ \eta(t) - y(t) \end{pmatrix}$$

et $x(0) = 0 = y(0)$

Analyse Numérique : le 04.04.19

Ex : Evolution d'un système planétaire.

les équations sont :

$$m_i q_i''(t) = -G \sum_{j \neq i} m_i m_j \frac{q_i - q_j}{\|q_i - q_j\|^3}$$

où $q_i(t) \in \mathbb{R}^3$ la position du i-ème corps à l'instant t, $m_i > 0$ sa masse, G la cst de gravitation universelle.

ici, on a une EDO d'ordre 2, car elle implique $q_i''(t)$ $i = 1 \dots k$ pour k corps

En introduisant la vitesse $v_i(t) = q_i'(t) \in \mathbb{R}^3$,

on peut se ramener à : $\begin{cases} q_i' = v_i \\ m_i v_i' = -G \sum_{j \neq i} m_i m_j \frac{q_i - q_j}{\|q_i - q_j\|^3} \end{cases}$

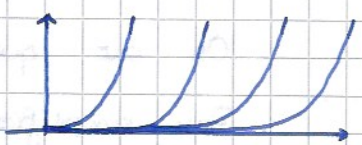
8.1 Existance et unicité :

Déf: Un problème $y' = f(y)$, $y(t_0) = y_0$ est un pb. de Cauchy.

Rem: Même si f est continue, il n'y a pas unicité en général de la solution.

Ex: $y' = \sqrt{|y|}$, $y(0) = 0$ possède une infinité de solutions.

$$y_c(t) = \begin{cases} 1/4 (t-c)^2 & t \geq c \\ 0 & 0 \leq t \leq c \end{cases}$$



⚠ $f(y) = \sqrt{|y|}$ est non-lipschitzien en 0.

Thm : Cauchy - Lipschitz

Dans un voisinage de (t_0, y_0), soit f continue et lipschitzienne par rapport à y :

$$\| f(t,x) - f(t,y) \| \leq L \cdot \| x - y \| \qquad \forall (t,x), (t,y) \text{ proche de } (t_0, y_0)$$

et L indépendant de x et y.

Alors le pb de C. possède une unique solution maximale, définie sur un ouvert $(t_-; t_+) \ni t_0$

Si de plus f est globalement lipschitzienne sur $\mathbb{R} \times \mathbb{R}^m$ alors $(t_-; t_+) = \mathbb{R}$, la solution est globale.

→ Exc Poly

En particulier, pour f C^1 dans un vois de (t_0, y_0), $\exists \alpha > 0$ tq la sol. existe et est unique sur $(t_0 - \alpha ; t_0 + \alpha)$

8.2 Méthodes de Runge - Kutta :

Idée : Pour approcher $y(t)$ sur un intervalle $[t_0, T]$, on subdivise l'intervalle en sous - intervalles $t_0 < t_1 < \dots < t_N = T$ et ensuite, on calcule des approximations $y_n \simeq y(t_n)$ par récurrence

$$y_{n+1} = y_n + h_n \, \phi(h_n, t_n, y_n) \quad \text{où} \quad h_n = t_{n+1} - t_n \ \text{(pas de temps)}$$

On remarque que y_{n+1} est calculé directement à partir de y_n sans connaître $y_{n-1}, y_{n-2}, \dots$

On dit que la méthode num. est à un pas.

Pour simplifier la notation, on regarde au premier pas de temps

$$y'(t) = f(t, y(t))$$

$$y(t_0 + h) = y(t_0) + \int_{t_0}^{t_0 + h} f(t, y(t)) \, dt$$

Idée : On approche l'intégrale par une formule de quadrature. L'approximation la + simple est la méthode des rectangles à gauche.

Analyse Numérique : le 04.04.19

$$\int_{t_0}^{t_0+h} f(t, y(t))\,dt \simeq h f(t_0, y(t_0))$$

On obtient la méthode d'Euler explicite

$$y_1 = y_0 + h f(t_0, y_0)$$
$$\vdots$$
$$y_{n+1} = y_n + h f(t_n, y_n)$$

Avec la méthode des rectangles à droites, on obtient :

$$\int_{t_0}^{t_0+h} f(t, y(t))\,dt \simeq h f(t_0+h, y(t_n+h))$$

On obtient la méthode d'Euler implicite :

$$y_{n+1} = y_n + h_n f(t_{n+1}, y_{n+1})$$

La méthode est implicite car le système est non-linéaire pour chaque pas.

On peut poser $F(4) = 4 - y_n - h_n f(t_{n+1}, 4)$

$F: \mathbb{R}^m \longrightarrow \mathbb{R}^m$ et chercher un zéro de F par une méthode (Newton p.ex).

Rem : Si f est lipschitzienne par rapport à y, du cst de lipsch. L, alors F est aussi lipschitzienne du cst $h_n L$ et F est contractante si $h_n < 1/L$

Avec la formule de quadrature du point milieu, on obtient

$$\int_{t_0}^{t_0+h} f(t, y(t))\,dt \simeq h f(t_0+h, y(t_0+\tfrac{h}{2})).$$

en utilisant $y(t_0+h/2) \simeq \frac{y_0+y_1}{2}$, on obtient la méthode implicite du point milieu : $y_{n+1} = y_n + h f\left(\frac{y_n+y_{n+1}}{2}\right)$

Méthode : de Runge

$$\int_{t_0}^{t_0+h} f(t, y(t))\,dt \simeq h f(t_0+h/2, y(t_0+h/2))$$
$$\simeq h f(t_0+h/2, y_0 + \tfrac{h}{2} f(t_0, y))$$

$$y_{n+1} = y_n + h_n f(t_n + \tfrac{h}{2}, y_n + \tfrac{h_n}{2} f(t_n, y_n))$$

Ceci conduit à une définition générale des méthodes de Runge - Kutta

Déf: Une méthode de Runge - Kutta à s étages explicite est donnée par :

$$k_1 = f(t_0 + c_1 h,\ y_0)$$
$$\vdots$$
$$k_s = f\left(t_0 + c_s h,\ y_0 + h \sum_{j=1}^{s-1} a_{sj} k_j\right)$$

$$y_1 = y_0 + h(b_1 k_1 + \dots + b_s k_s)$$

avec c_i, a_{ij}, b_j, $i,j = 1 \dots s$ les coeff de la méthode

Noté comme : $\begin{array}{c|c} c_i & a_{ij} \\ \vdots & \\ \hline & b_j \end{array}$

---

le 11.04.19

Rem :

- s évaluations de f pas à pas de temps
- On pose $c_i = \sum_{j=1}^{i-1} a_{ij}$ pour avoir $y_i \simeq y(t_0 + c_i h)$ avec égalité sur $f \equiv 1$.
- Notation $\begin{array}{c|c} c_i & a_{ij} \\ \hline & b_j \end{array}$
- $\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array}$ est la méthode d'Euler explicite. $y_1 = y_0 + h f(t_0, y_0)$
- $\begin{array}{c|cc} 0 & 0 & \\ 1/2 & 1/2 & \\ \hline & 0 & 1 \end{array}$ méthode de Runge

Déf: Une méthode de Runge - Kutta est dite d'ordre p si la solution de $y' = f(t,y)$, $y(t_0) = y_0$ avec $f : \mathbb{R} \times \mathbb{R}^d \longrightarrow \mathbb{R}^d$ régulière vérifie : $y_1 - y(t_0 + h) = O(h^{p+1})$, $h \to 0$

La quantité $y_1 - y(t_0 + h)$ s'appelle l'erreur locale (après 1 pas de temps)

Ex: avec la méthode d'Euler explicite (d'ordre 1)

$$\begin{aligned} y_1 - y(t_0 + h) &= y_1 - y(t_0) - h y'(t_0) + O(h^2) \\ &= y_1 - (y(t_0) + h f(t_0, y_0)) + O(h^2) \\ &= O(h^2) \end{aligned}$$

Analyse Numérique :

le 11.04.19

Exc : Méthode de Runge (Cas autonome $f(t,y) = f(y)$)

On doit développer $y(t_0+h)$ en série de Taylor à l'ordre 2.

$$y' = f(y) \quad . \quad y'' = f'(y)\,y' = f'(y)f(y)$$

$$y(t_0+h) = y(t_0) + h\,y'(t_0) + \frac{h^2}{2}\,y''(t_0) + O(h^3)$$

$$y(t_0+h) = y_0 + h f(y_0) + \frac{h^2}{2} f'(y_0) f(y_0) + O(h^3)$$

$$y_1 = y_0 + h f\left(y_0 + \frac{h}{2} f(y_0)\right)$$

$$= y_0 + h f(y_0) + h f'(y_0)\left(\frac{h}{2} f(y_0)\right) + O(h^3)$$

Ainsi $y_1 - y(t_0+h) = O(h^3)$ $\qquad h \longrightarrow 0$

Exc : " La méthode de Runge - Kutta d'ordre 4. "

0	0			
1/2	1/2	0		
1/2	0	1/2	0	
1	0	0	1	0
	1/6	1/3	1/3	1/6

$s = 4$.

Rem : On remarque qu'appliquer une méthode de Runge - Kutta au problème non - autonome $y' = f(t,y)$, $y(t_0) = y_0$ ($f : \mathbb{R} \times \mathbb{R}^d \longrightarrow \mathbb{R}^d$) équivaut à l'appliquer au problème autonome suivant :

$$\begin{cases} y'(t) = f(s(t), y(t)) & y(t_0) = y_0 \\ s'(t) = 1 & s(t_0) = t_0 \end{cases}$$

avec le champs de vecteur autonome suivant :

$$\tilde{f} : (s,y) \longmapsto (1, f(s,y)) \qquad \mathbb{R} \times \mathbb{R}^d \longrightarrow \mathbb{R} \times \mathbb{R}^d$$

en utilisant la relation : $c_i = \sum_{j=1}^{i-1} a_{ij} \qquad i = 1 \dots s$

Ainsi, la méthode de Runge est d'ordre 2 aussi dans le cas non - autonome.

8.3 Convergence des méthodes de Runge-Kutta :

On commence à $t = t_0$ et on applique n fois la méthode R-K avec un pas h (constant pour simplifier), pour obtenir une approximation

$y_n \simeq y(t_n)$ au temps $t_n = t_0 + n \cdot h$

et on cherche à estimer l'erreur globale

$$\| y_n - y(t_n) \| \overset{?}{=} O(h^p)$$

Thm: Soit $y(t)$ la solution de $y' = f(y)$, $y(t_0) = y_0$ sur $[t_0, T]$ avec $T > t_0$ fixé.

On considère une méthode de R-K, écrite sous la forme :

$$y_{n+1} = y_n + h \, \Phi(t_n, y_n, h)$$

On suppose :

a) l'erreur locale vérifie $\forall t \in [t_0, T]$, $h \leq h_{max}$

$$\| y(t+h) - y(t) - h \, \Phi(t, y(t), h) \| = O(h^{p+1})$$

b) $\Phi(t, y, h)$ est lipschitzienne dans un voisinage de $y(t)$ (par rapport à y) $\| \Phi(x, h) - \Phi(z, h) \| \leq L \| x - z \| \quad \forall x, z \in B_R(y(t))$

Alors pour $h \leq h_{max}$ assez petit, l'erreur globale vérifie :

$\| y(t_n) - y_n \| \leq \tilde{C} h^p = O(h^p)$ $\forall t_n \in [t_0, T]$ avec $\tilde{C}$ indép. de h, mais $\tilde{C}$ dép. de L et T.

Rem: Contrairement à la preuve pour les formules de quadrature, on ne peut pas ajouter directement les erreurs locales pour trouver l'err. globale. En effet dès le $2^{ème}$ pas, la donnée initiale n'est pas la même pour les sol. exactes et numériques

$y_1 \neq y(t_1)$

Pv: $\|y(t_{n+1}) - y_{n+1}\| = \|y(t_{n+1}) - y_n - h\Phi(y_n, h)\|$

$\leq \|y(t_{n+1}) - y(t_n) - h\Phi(y(t_n), h)\| + \|y(t_n) - y_n\| + h\|\Phi(y(t_n), h) - \Phi(y_n, h)\|$

$\leq Ch^{p+1} + \|y(t_n) - y_n\| + hL\|y(t_n) - y_n\|$

$\leq Ch^{p+1} + e^{hL} E_n$ avec $E_n = \|y(t_n) - y_n\|$ et $1 + hL \leq e^{hL}$

Ainsi $E_n \leq Ch^{p+1} + e^{hL} E_{n-1}$

$\leq Ch^{p+1} + e^{hL}(Ch^{p+1} + e^{hL} E_{n-2}) \leq \dots$

$\leq Ch^{p+1} + \sum_{j=1}^{n-1} e^{(n-j)hL} Ch^{p+1} + e^{nhL} \underbrace{E_0}_{0}$

$\leq C h^p h \sum_{j=1}^{n} e^{(t_n - t_j) L}$

On a mq $E_n \leq Ch^p h \sum_{j=1}^{n} e^{(t_n - t_j) L}$

On reconnaît une somme de Rieman pour la fonction décroissante $t \longmapsto e^{(t_n - t)L}$

$h \sum_{j=1}^{n} e^{(t_n - t_j)L} \leq \int_{t_0}^{t_n} e^{(t_n - t)L} dt = \frac{e^{(t_n - t_0)L} - 1}{L}$

Il reste à vérifier qu'on peut appliquer b) avec y_n au voisinage de $y(t_n)$

Pour cela, on chaisit h assez petit pour obtenir $h^p \underbrace{\frac{C \cdot e^{(T - t_0)L} - 1}{L}}_{\tilde{C}} \leq R$ ▨

Pour une méthode de R-K, on a :

$$\phi(y, h) = \sum_{i=1}^{s} b_i k_i \qquad k_i = f\left(y + h \sum_{j=1}^{i-1} a_{ij} k_j\right) \qquad i = 1 \dots s$$

Lemme: Si f vérifie une condition de Lipschitz $\|f(x) - f(z)\| \leq \tilde{L}\|x - z\|$ pour x, z au vois. de $y(t)$, alors la méthode de R-K vérifie b) avec $L = 1 + h\tilde{L}\left(\sum_i |b_i| + h \max \tilde{L} \sum_i |k_i a_{ij}| + (h \max \tilde{L})^2 \sum_{i,j,k} |b_i a_{ij} a_{jk}| + \dots\right)$

Pv: On note $k_i(y)$ pour insister sur la dépendance de k_i par rapport à y. La condition de Lipschitz pour f appliquée à $k_i(y) - k_i(z)$ donne :

Pv : (Suite)

$$\| k_1(y) - k_1(z) \| = \| f(y) - f(z) \| \leq \tilde{L} \| y - z \|$$

$$\| k_2(y) - k_2(z) \| \leq \tilde{L} \| y - z \| + \tilde{L} \| h a_{21} (k_1(y) - k_1(z)) \|$$

$$\leq \tilde{L} (1 + h_{max} \tilde{L} | a_{21} |) \| y - z \|$$

$$\| k_3(y) - k_3(z) \| \leq \ldots .$$

Et on insère ces estimations dans : $\Phi(y, h) - \Phi(z, h)$ ☒

---

le 18.04.19

Rem: Par la convergence globale d'une méthode de Runge-Kutta, on a montré : $E_n = \| y(t_n) - y_n \| \leq C h^p \dfrac{e^{L(T - t_0)}}{L} < R \quad t_0, t_n \in [t_0, T]$

avec $h \leq h_{max}$ et h max choisi assez petit.

Ainsi, la preuve est en deux étapes, d'abord formelle (on ne sait pas encore si y_n reste au voisinage de la solution exacte $y(t_n)$), puis rigoureuse avec h max assez petit.

8.4 Méthodes implicites :

On a vu qu'avec la formule de quadrature des rectangles à droite, appliquée à $y(t) = y_0 + \int_0^t y'(s)\, ds$, on obtient la méthode d'Euler implicite : $y_1 = y_0 + h f(t_1, y_1)$

Avec la méthode de quadrature des trapèzes, on obtient la méthode des trapèzes : $y_1 = y_0 + \frac{h}{2} f(t_0, y_0) + \frac{h}{2} f(t_1, y_1)$

Ces méthodes sont plus compliquées à implémenter, il faut résoudre à chaque pas un système non-linéaire.

Prop: On suppose f lipschitzienne par rapport à y, de cst. L. Alors pour $h < 1/L$ la méthode d'Euler implicite est bien définie, y_1 existe et est défini de manière unique.

le 18.04.19

Analyse Numérique :

Pr : On remarque que y_1 est point fixe de $y \longmapsto y_0 + hf(t, y)$ qui est lipschitzienne de cst $hL < 1$ donc contractante. Elle possède donc exactement un point fixe par thm Banach.

Rem : Pour la trapèzes, la condition est $\frac{hL}{2} < 1$

Déf : Plus généralement, on définit les méthodes de Runge-Kutta à s étages implicites :

$$k_i = f\left(t_0 + c_i h,\ y_0 + h \sum_{j=1}^{s} a_{ij} k_j\right), \quad i = 1, \dots, s$$

$$y_1 = y_0 + h \sum_{i=1}^{s} b_i k_i$$

Notation en tableau de Butcher :

$$\begin{array}{c|c} c_i & a_{ij} \\ \hline & b_j \end{array} \qquad c_i = \sum_{j=1}^{s} a_{ij}$$

Rem : La méthode devient explicite si $a_{ij} = 0$ pour $j \geq i$, $\forall i, j = 1 \dots s$ (La matrice des (a_{ij}) est triangulaire inf. stricte)

Exc : Méthode implicite des trapèzes.

$$y_1 = y_0 + h/2\, f(t_0, y_0) + \frac{h}{2} f(t_0 + h, y_1)$$

$$\Leftrightarrow \begin{cases} k_1 = f(t_0, y_0) \\ k_2 = f(t_1, y_0 + h/2 (k_1 + k_2)) \\ y_1 = y_0 + h/2 (k_1 + k_2) \end{cases} \Leftrightarrow \begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1/2 & 1/2 \\ \hline & 1/2 & 1/2 \end{array}$$

Comment évaluer l'ordre ?

Exo : Méthode des trapèzes : (Cas autonome)

$$y_1 = y_0 + h/2\, f(y_0) + h/2\, f(y_1)$$

On substitue : $y_1 = y_0 + O(h)$ et on obtient :

$$y_1 = y_0 + h/2\, f(y_0) + h/2\, f(y_0 + O(h))$$

$$= y_0 + \frac{h}{2} f(y_0) + \frac{h}{2} f(y_0) + \frac{h}{2} O(h)$$

$$= y_0 + h f(y_0) + O(h^2)$$

Exc: (Suite)

$$y_1 = y_0 + \frac{h}{2} f(y_0) + \frac{h}{2} f\left(y_0 + h f(y_0) + O(h^2)\right)$$

$$= y_0 + h/2\, f(y_0) + \frac{h}{2}\left(f(y_0) + f'(y_0)\left(h f(y_0) + O(h^2)\right) + \frac{1}{2} f''(y_0)(O(h), O(h))\right)$$

$$y_1 = y_0 + h f(y_0) + \frac{h^2}{2} f'(y_0) f(y_0) + O(h^3)$$

⇒ Méthode d'ordre au moins 2.

Exc: Euler implicite : $\begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array}$

Point milieu implicite : $\begin{array}{c|c} 1/2 & 1/2 \\ \hline & 1 \end{array}$

$$y_1 = y_0 + h f\left(t_0 + h/2,\ \frac{y_0 + y_1}{2}\right)$$

$$\begin{cases} k_1 = f\left(t_0 + h/2,\ y_0 + \frac{h}{2} k_1\right) \\ y_1 = y_0 + h k_1 \end{cases}$$

8.5 Equations différentielles raides :

Pour certains problèmes dits raides, les méthodes de R.-K. explicites ont une très forte restriction sur le pas de temps h.

Exc: Problème raide

$$y'(t) = 1/\varepsilon\,(-y + \cos t) \quad \text{avec } \varepsilon \ll 1 \text{ fixé.}$$

On cherche une solution particulière du problème in-homogène de la forme $y(t) = A\cos t + B\sin t$

On trouve A et B, et la solution générale est alors

$$y(t) = e^{-1/\varepsilon t} \cdot C + A\cos t + B\sin t$$

$$y(t) = C \cdot e^{-1/\varepsilon t} + \frac{1}{1+\varepsilon^2}\cos t + \frac{\varepsilon}{1+\varepsilon^2}\sin t$$

Comme $\frac{1}{1+\varepsilon^2} = 1 + O(\varepsilon^2)$, on obtient :

$$y(t) = \cos t + \varepsilon \sin t + O(\varepsilon^2) + C e^{-1/\varepsilon t} \quad \text{et} \quad y(t) \xrightarrow[\varepsilon \to 0]{} \cos t$$

Rem: Formellement $-y(t) + \cos t = \varepsilon \cdot y'(t) \xrightarrow[\varepsilon \to 0]{} 0$

Exc : (Suite)

Avec la méthode d'Euler explicite, on obtient :

$$y_{n+1} = (1 - h/\varepsilon)\, y_n + h/\varepsilon \cos t_n$$

et on peut mq (par récurrence) :

$$y_n = (1 - h/\varepsilon)^n \cdot C + \cos t_n + \varepsilon \cdot \sin(t_n) + O(h\varepsilon)$$

On voit que y_n reste proche de la sol. exacte si $|1 - h/\varepsilon| < 1$

ce qui équivaut à $0 < h < 2 \cdot \varepsilon$. Restriction dramatique pour $\varepsilon \ll 1$

En revanche, avec la méthode d'Euler implicite :

$$(1 + h/\varepsilon)\, y_{n+1} = y_n + h/\varepsilon \cos t$$

Et par récurrence :

$$y_n = \underbrace{(1 + h/\varepsilon)^{-n}}_{\xrightarrow{n \to \infty} 0 \text{ car } 1 + h/\varepsilon > 1} \cdot C + \cos t_n + \varepsilon \sin t_n + O(\varepsilon h)$$

Cette fois, il n'y a plus de restriction sur le temps.

- Le précédent calcul mq ce n'est pas la partie inhomogène du problème qui pose difficulté.

On considère maintenant l'équation scalaire test de Dalquist :

$$y'(t) = \lambda y(t), \quad y(0) = y_0, \quad \text{avec } \lambda \in \mathbb{C} \text{ fixé } (\lambda = \mathrm{Re}\lambda + i\,\mathrm{Im}\lambda)$$

Sol. exacte : $y(t) = e^{\lambda t} y_0 = e^{\mathrm{Re}\, t}(\cos((\mathrm{Im}\lambda)t) + i \sin((\mathrm{Im}\lambda) \cdot t))\, y_0$

Pour $y_0 \neq 0$, $y(t)$ est bornée $\forall t > 0 \iff \mathrm{Re}\lambda \leq 0$

(c-à-d $\lambda \in \mathbb{C}_- = \{ z \in \mathbb{C} \mid \mathrm{Re}\, z \leq 0 \}$)

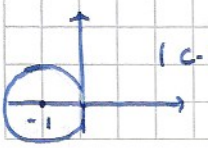
Si on applique la méthode d'Euler explicite à $y' = \lambda y$, on obtient

$$y_{n+1} = y_n + h\lambda y_n = (1 + h\lambda)\, y_n = R(z)\, y_n \quad \text{où } z = h\lambda$$

et $R(z) = 1 + z$, appellée fonction de stabilité.

On voit que $(y_n)_{n \geq 0}$ est bornée pour $y_0 \neq 0 \iff |R(z)| \leq 1$

(c-à-d $z \in$ disque de centre -1 et rayon 1 dans $\mathbb{C}$)



On obtient pour $\lambda < 0$, $z = h\lambda \in [-2, 0]$, soit $h \leq \frac{2}{|\lambda|}$

condition sur le pas de temps de la stabilité numérique.

De manière générale, pour une méthode de R.-K. implicite,

on obtient : $y_{n+1} = R(z)\, y_n$ avec $z = h\lambda$ et $R(z)$ est une fonction rationelle qui dépend des coeff de la méthode, mais pas de λ et h.

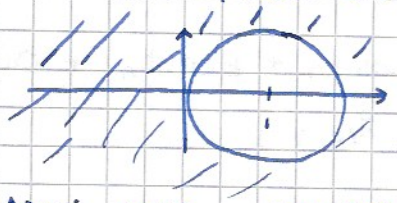
Rem : Si la méthode est explicite, $R(z)$ est un polynôme.

Exc : Euler implicite

$$y_{n+1} = y_n + h\lambda\, y_{n+1} \qquad (1 - h\lambda)\, y_{n+1} = y_n$$

$$y_{n+1} = \frac{1}{1-h\lambda}\, y_n = R(z)\, y_n \qquad R(z) = \frac{1}{1-z}$$

La condition $|R(z)| \leq 1$ devient :



$z = h\lambda$ doit être hors du disque de centre 1 et de rayon 1.

Ainsi, cette condition est vérifiée sans restriction sur h pour $\lambda \leq 0$.

Déf : Domaine de Stabilité pour une méthode de R.-K.

On pose $S = \{ z \in \mathbb{C} \mid |R(z)| \leq 1 \}$

Déf : A-Stabilité

On dit qu'une méthode de R.-K. est A-Stable si $\mathbb{C}_- \subset S$

(c-à-d : la sol. num. est bornée si la sol exacte l'est)

Exc : La méthode d'Euler implicite est A-Stable.

Thm : Si une méthode de R.-K. avec fonction de stabilité $R(z)$ vérifie : • $|R(y)| \leq 1 \quad \forall y \in \mathbb{R}$

• $R(z)$ ne possède aucun pôle dans $\mathbb{C}_-$

Alors la méthode est A-Stable.

Exc : La méthode Θ

$$y_{n+1} = y_n + (1-\Theta)\, h f(y_n) + \Theta h f(y_{n+1})$$

(voir les cas $\Theta = 0, 1/2, 1$ déjà étudiés)

$$R(z) = \frac{1 + (1-\Theta)z}{1 - \Theta z}$$

Il n'y a pas de pôles dans $\mathbb{C}_-$ pour $\Theta \geq 0$

Ensuite, on calcule : $|R(iy)|^2 = \dfrac{1 + (1-\Theta)^2 y^2}{1 + \Theta^2 y^2}$

Ainsi $|R(iy)|^2 \leq 1$ équivaut à $1 + (1-\Theta)^2 y^2 \leq 1 + \Theta^2 y^2$

on trouve $(1 - 2\Theta)\, y^2 \leq 0$, c-à-d $\Theta \geq 1/2$

La méthode Θ est A-Stable $\Leftrightarrow$ $\Theta \geq 1/2$

Pv : (thm)

On pose $\mathbb{D}_R = \{ z \in \mathbb{C}_-,\ |z| \leq R \}$

comme $R(z)$ est une fraction rationelle, avec $|R(iy)|$ bornée,

on obtient $|R(z)| \xrightarrow{z \to +\infty} |R(\infty)| \in [0,1]$

En particulier, $\forall \varepsilon > 0$, $\exists R_0 > 0$ tq $|z| \geq R_0 \Rightarrow |R(z)| \leq 1 + \varepsilon$

Comme $R(z)$ est analytique dans $\mathbb{C}_-$ (pas de pôle), on a :

$$\sup_{z \in \mathbb{C}} |R(z)| = \lim_{R \to \infty} \sup_{z \in \mathbb{D}_R} |R(z)| = \lim_{R \to \infty} \sup_{z \in \partial \mathbb{D}_R} |R(z)| \leq 1 + \varepsilon$$

∎

8.6 Equation de la chaleur : le 02.05.19

On considère l'équation de la chaleur en dimension 1 d'espace :

$$\frac{\partial u}{\partial t}(x,t) = \frac{\partial^2 u}{\partial x^2}(x,t) \qquad t>0 \,,\ x \in (0,1)$$

avec des conditions aux bords $u(0,t) = u(1,t) = 0$, $t>0$

et la condition initiale $u(x,0) = g(x)$ 0°C |——————| 0°C (0 à 1)

Pour la résolution numérique, on discrétise ... espace en approchant

$u(x_j,t) \simeq u_j(t)$ avec $x_j = j\Delta x$, $\Delta x = \frac{1}{N+1}$ $j = 0, \dots N+1$

On considère l'approximation de différences finies.

$$\frac{\partial^2 u}{\partial x^2}(x_j,t) \simeq \frac{u(x_{j-1},t) - 2u(x_j,t) + u(x_{j+1},t)}{\Delta x^2}$$

$$\simeq \frac{u_{j-1}(t) - 2u_j(t) + u_{j+1}(t)}{\Delta x^2} \qquad j = 1 \dots N$$

avec les conditions au bord, on a $u_0(t) = u_{N+1}(t) = 0$

et l'inconnue est $\underline{u} = \begin{pmatrix} u_1(t) \\ \vdots \\ u_N(t) \end{pmatrix} \in \mathbb{R}^N$

On obtient le système d'EDO :

$$\frac{\partial u_j}{\partial t} = \frac{u_{j-1} + 2u_j + u_{j+1}}{\Delta x^2} \,, \quad j = 1, \dots, N \quad \text{avec } u_0 = u_{N+1} \equiv 0$$

et la condition initiale $u_j(0) = g(x_j)$

Matriciellement, on obtient :

$$\frac{\partial \underline{u}}{\partial t} = A\,\underline{u}(t) \quad \text{avec} \quad A = \frac{1}{\Delta x^2}\begin{pmatrix} -2 & 1 & & 0 \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ 0 & & 1 & -2 \end{pmatrix} \qquad \underline{u}(0) = \begin{pmatrix} g(x_1) \\ \\ g(x_N) \end{pmatrix}$$

La matrice A est symétrique réelle et ses valeurs propres sont dans $\left(\frac{-4}{\Delta x^2}, 0\right)$. On peut diagonaliser A en base orthonormée :

$A = Q \Lambda Q^T$, $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_N)$ matrice des val. propres.

avec Q orthogonale.

On considère le changement de variable qui diagonalyse A

et on pose $\underline{u} = Q\,v$

$$\partial\, \frac{dv}{dt} = Q^T \frac{du}{dt} = Q^T A\, \underline{u}(t) = Q^T A\, Q\, v(t) = \Lambda\, v(t)$$

On obtient N équations différentielles scalaires, découplées,

$\frac{d}{dt} v_j(t) = \lambda_j v_j(t) \quad j = 1, \dots, N$

La sol. est $v_j(t) = e^{\lambda_j t} v_j(0) \xrightarrow{t \to +\infty} 0$ car $\lambda_j < 0$

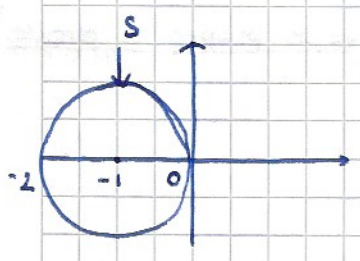
Ainsi, $\|u(t)\|_2 = \|v(t)\|_2 \xrightarrow{t \to +\infty} 0$

Considérons maintenant les méthodes d'Euler explicite, puis implicite :

E. Exc : $u_{n+1} = u_n + hAu_n = (I + hA) u_n$

ou de manière équivalente $v_{n+1} = (I + h\Lambda) v_n$

$v_{j,n+1} = R(\lambda_j h) v_{j,n} \quad j = 1 \dots N$ et $R(z) = 1 + z$



Ainsi, la sol. numérique reste bornée si $|R(h\lambda_j)|$

$= |1 + h\lambda_j| < 1 \quad j = 1 \dots N \quad \forall h$

$h\lambda_j \in S \quad j = 1 \dots N$, $S = D_{-1}(1)$

Il faut donc $h\lambda_N \in S$, $\lambda_N < \dots < \lambda_1 < 0$.

$h\lambda_N \in [-2, 0]$

la sol. numérique est bornée si $h - \frac{4}{\Delta x^2} \in [-2, 0]$

ce qui équivaut à $h \leq \frac{\Delta x^2}{2}$

Condition "CFL" de stabilité très restrictive sur le pas de temps h.

E. Im. : On a maintenant $R(z) = \frac{1}{1-z}$ et

$|R(h\lambda_j)| = \frac{1}{1 - h\lambda_j} < 1 \quad \forall h$

car les $\lambda_j < 0$

Ainsi, cette méthode donne une sol. num. bornée sans restriction sur h (pas de temps)

Plus généralement, une méthode de R-K. A-stable appliquée à l'équation de la chaleur discrétisée donne une sol. num. bornée, sans restriction sur h.

(On peut choisir $h \simeq \Delta x$)

Rappel : $S = \{ z \in \mathbb{C} \; ; \; |R(z)| \leq 1 \}$

A-stable signifie $\mathbb{C}_- \subset S$ avec $\mathbb{C}_- = \{ z \in \mathbb{C} \mid Re(z) \leq 0 \}$

Ch. 9 Valeurs et Vecteurs propres :

Déf: Soit $A \in \mathbb{R}^{n \times n}$ (ou $\mathbb{C}^{n \times n}$), le scalaire $\lambda \in \mathbb{C}$ est une val. propre de A si $\exists v \neq 0$ tq $Av = \lambda v$. (v vect. propre à droite)

Rem :
- $\lambda \in \mathbb{C}$ en général, même pour $A \in \mathbb{R}^{n \times n}$.
- $\lambda \in \mathbb{R}$ si la matrice est symétrique réelle.
- Un vect. p. à gauche est donné par u^H avec $u^H A = \lambda u^H$ où $M^H = \bar{M}^T$ et $u^H = \bar{u}^T$ ($u^H = u^*$ et $A^H = A^*$)
- u^H est vect. p. à droite de A^H pour le même λ.
- En général, on peut choisir u et v réels pour A réelle.

Prop :
1) $Av = \lambda v$ avec $v \neq 0$ équivaut à $Ker(A - \lambda I) \neq \{0\}$
   ou encore $A - \lambda I$ pas inversible
   ou encore $\det(A - \lambda I) = 0$
2) On note $p_A(\lambda) = \det(A - \lambda I)$ le polynôme caractéristique de A.
3) Les valeurs propres sont les n racines de $p_A(\lambda)$

Analyse Numérique : le 02.05.19

En général, il n'y a pas de formule explicite pour calculer les λ_i (pas de formule générale à partir de $n \geq 5$), et on doit approcher les valeurs propres par des méthodes numérique et on évite de les calculer comme zéro de $p_A(\lambda)$.

9.1 Condition du calcul des valeurs et vecteurs propres :

On rappel que les données, comme la matrice A, sont connues à perturbation près. Il faut étudier l'influence de cette perturbation sur les valeurs propres et vecteurs propres de A.

Soit $\tilde{a}_{ij} = a_{ij}(1 + \varepsilon_{ij})$ avec $\varepsilon_{ij} \leq \varepsilon_{mach}$

Matriciellement, $A(\varepsilon) = \tilde{A} = A + \varepsilon E$ avec $|\varepsilon| \leq \varepsilon_{mach}$ $\|E\| \leq \|A\|$

Thm : Gershgorin

Soit $A \in \mathbb{R}^{n \times n}$ (ou $\mathbb{C}^{n \times n}$)

a) Si λ est une val. p. de A, alors $\exists i$ indice tq $|\lambda - a_{ii}| \leq R_i$

$$R_i := \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}|$$

càd : Toutes les val. p. de A sont dans l'union des disques

$$D_i := \overline{D}_{R_i}(a_{ii}) = \{ z \in \mathbb{C} \mid |z - a_{ii}| \leq R_i \}$$

b) Si une composante connexe de $\bigcup_{i=1}^{n} D_i$ consiste de k disques, alors elle contient exactement k val. propres.

Rv : a) Soit $v \neq 0$ avec $Av = \lambda v$, soit i tq $\|v\|_\infty = |v_i| > 0$

la $i^{ème}$ composante de $Av = \lambda v$ s'écrit :

$$\sum_{j=1}^{n} a_{ij} v_j = \lambda v_i \quad \text{et donc} \quad \sum_{\substack{j=1 \\ j \neq i}}^{n} a_{ij} v_j = (\lambda - a_{ii}) v_i$$

En divisant par $v_i \neq 0$, on a :

$$|\lambda - a_{ii}| \leq \left| \sum_{j \neq i} a_{ij} \frac{v_j}{v_i} \right| \leq \sum_{j \neq i} |a_{ij}| \underbrace{\frac{|v_j|}{|v_i|}}_{\leq 1} \leq R_i$$

Rv : b) On utilise un argument de continuité, en se ramenant au cas A diagonale.

On pose $A(t) = D + t \cdot B$ avec $\begin{cases} D = \operatorname{diag}(A) = \operatorname{diag}(a_{11}, \dots, a_{nn}) \\ B := A - D \end{cases}$

Comme les racines d'un polynôme sont une fonction continue de ses coefficients (Analyse II), les val. p. de $A(t)$ varient continuement avec t.

Calculons les rayons $R_i(t)$ des disques de Gershgorine de la matrice $A(t)$ $(t > 0)$

$$R_i(t) = \sum_{j \neq i} |t b_{ij}| = t \sum_{j \neq i} |b_{ij}| = t R_i$$

Pour $t = 0$, les disques de G sont exactement les points $\{a_{ii}\}$ $i = 1 \dots n$, et pour $0 \leq t \leq 1$, ils grandissent continuement, en contenant chacun une val. p. différente de $A(t)$, avec la multiplicité.

Ainsi les val. p. de A ne peuvent "s'échapper" des $D_i(t)$ d'où le résultat.

▨

---

le 09.05.A

Thm : Bauer - Fike 1960

Soit $A \in \mathbb{C}^{n \times n}$ ou $\mathbb{R}^{n \times n}$ diagonalisable, avec $A = TDT^{-1}$ où $D = \operatorname{diag}(\lambda_1, \dots, \lambda_n)$ matrice diagonale des val. propres et T la matrice de changement de base (inversible) et soit $\tilde{A} = A + E$.

Alors pour chaque val. propre $\tilde{\lambda}$ de $\tilde{A}$, il existe une val. propre λ_i de A tq $\|\tilde{\lambda} - \lambda_i\| \leq K_\infty(T) \|E\|_\infty$ où $K_\infty(T) = \|T\|_\infty \|T^{-1}\|_\infty$ est le conditionnement de T.

le 09.05.19

Analyse Numérique :

Pv : On applique le changement de base T à $\tilde{A}$

$$T^{-1}\tilde{A}T = T^{-1}(A+E)T = \mathrm{diag}(\lambda_1,\dots,\lambda_n) + T^{-1}ET$$

On remarque que $T^{-1}\tilde{A}T$ a les mêmes val. propres que $\tilde{A}$

On pose $F = T^{-1}ET$ et on applique le thm de Gerschgorin à $T^{-1}\tilde{A}T$:

$$\exists i \text{ tq } |\tilde{\lambda} - (\lambda_i + f_{ii})| \leq \sum_{j\neq i} |f_{ij}|$$

l'inégalité triangulaire donne :

$$|\tilde{\lambda} - \lambda_i| \leq |\tilde{\lambda} - (\lambda_i + f_{ii})| + |f_{ii}| \leq \sum_{j=1}^{n} |f_{ij}| \leq \|F\|_\infty$$

$$\leq \|E\|_\infty \underbrace{\|T\|_\infty \|T^{-1}\|_\infty}_{K_\infty(T)}$$

Rem : On a ainsi montré, pour $\|A - \tilde{A}\|_\infty \leq \delta$ que $|\tilde{\lambda}_i - \lambda_i| \leq K_\infty(T)\,\delta$

Ainsi la condition absolue K du calcule de λ_i vérifie

$$K_i \leq K_\infty(T)$$

La condition est donc bonne pour des matrices T orthogonales ($K_\infty(T) \leq n$)

On va voir qu'une analyse plus précise est possible pour une seule val. propre de multiplicité simple.

Thm : Différentiabilité des val. propres.

Soit $A \in \mathbb{R}^{n\times n}$ ($\mathbb{C}^{n\times n}$), supposons que λ_* est une val. propre de A de multiplicité simple avec u_* et v_* les vecteurs propres à gauche et à droite de A respectivement,

$$u_*^H A = \lambda_* u_*^H \quad \text{et} \quad A v_* = \lambda_* v_*$$

Alors $\forall \varepsilon > 0$ assez petit, la matrice $A(\varepsilon) = A + \varepsilon \cdot E$, avec $E \in \mathbb{R}^{n\times n}$ ($\mathbb{C}^{n\times n}$) fixée, possède une val. propre simple $\lambda(\varepsilon)$ près de λ_*. De plus, $\lambda(\varepsilon)$ est dérivable dans un voisinage de 0, et on a :

$$\lambda(\varepsilon) = \lambda_* + \varepsilon \underbrace{\frac{u_*^H E v_*}{u_*^H v_*}}_{\lambda'(0)} + O(\varepsilon^2)$$

p 101

Rem : • Cette formule est en fait un développement de Taylor à l'ordre 1 de la fonction $\lambda(\varepsilon)$ pour $\varepsilon \to 0$.

• Ce théorème stipule que pour $\|\tilde{A} - A\|_2 \leq \delta$ avec λ_* une val. prop. simple de A, on obtient pour $\tilde{\lambda}_*$ val. prop. de $\tilde{A}$:

$$|\tilde{\lambda}_* - \lambda_*| = \varepsilon \left| \frac{u_*^H E v_*}{u_*^H v_*} \right| + O(\varepsilon^2) \leq \varepsilon \cdot \frac{\|E\|_2 \, \|u_*^H\|_2 \, \|v_*^H\|_2}{|u_*^H v_*|} + O(\varepsilon^2)$$

$$\leq \frac{\delta}{|u_*^H v_*|} + O(\delta^2)$$

Ainsi, le conditionnement absolu du calcul de λ_*, val. prop. simple est donné par $\kappa_* = \dfrac{1}{|u_*^H v_*|}$

en supposant $\|u_*^H\| = \|v_*\| = 1$

Ex : • Si A est normale ($AA^H = A^H A$), par le thm spectral, A est diagonalisable avec une matrice de changement de base U unitaire ($UU^H = I$) et $U^H A U = \mathrm{diag}(\lambda_1, \dots, \lambda_n) = D$

Dans ce cas, on a $u_i = v_i$ (à la fois vect. p. à droite et à gauche) puisque $AU = UD$, $U^H A = D U^H$ et on obtient

$$\kappa_i = \frac{1}{|u_*^H v_*|} = \frac{1}{\|u_*\|_2^2} = 1$$

• Si A n'est pas normale, le calcul de λ_i peut être mal conditionné :

$$A = \begin{pmatrix} 1 & \alpha \\ 0 & 2 \end{pmatrix} \qquad \lambda_1 = 1, \; v_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad u_1 = \frac{1}{\sqrt{1+\alpha^2}} \begin{pmatrix} 1 \\ -\alpha \end{pmatrix}$$

donc $\kappa_1 = \dfrac{1}{|u_1^H v_1|} = \sqrt{1+\alpha^2}$ mal cond. si α grand.

• Un cas bien pire : une val. prop. de multiplicité multiple

$$A = \begin{pmatrix} \lambda_1 & 1 \\ 0 & \lambda_1 \end{pmatrix} \qquad E = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \qquad A(\varepsilon) = A + \varepsilon E$$

$$p_{A(\varepsilon)}(\lambda) = \det(A(\varepsilon) - \lambda I) = (\lambda_1 - \lambda)^2 - \varepsilon$$

les val. prop. sont donc $\lambda_1(\varepsilon) = \lambda_1 + \sqrt{\varepsilon}$, $\lambda_2(\varepsilon) = \lambda_1 - \sqrt{\varepsilon}$

Ainsi, la perturbation des val. propres est de taille $O(\sqrt{\varepsilon})$ ($\gg O(\varepsilon)$) et $\lambda_1(\varepsilon)$, $\lambda_2(\varepsilon)$ ne sont pas dérivables en $\varepsilon = 0$

Le calcul des v. p. est alors très mal conditionné.

le 09.05.19

Analyse Numérique :

Pv : (du thm)

On considère le cas $A \in \mathbb{C}^{n\times n}$.

On définit $f : \mathbb{C}\times\mathbb{R} \longrightarrow \mathbb{C}$

$(\lambda, \varepsilon) \longmapsto p_{A(\varepsilon)}(\lambda) = \det(A(\varepsilon) - \lambda I)$

qui est un polynôme en λ, donc analytique.

Comme λ_* est une val. p. simple, c'est une racine simple de $p_{A(\varepsilon)}$ et on a : $p_A(\lambda_*) = 0 = f(\lambda_*, 0)$ et $p_A'(\lambda_*) \neq 0 \neq \frac{\partial f}{\partial \lambda}(\lambda_*, 0)$

On peut appliquer le thm des fonctions implicites et il existe une fonction $\varepsilon \longmapsto \lambda(\varepsilon)$ définie et dérivable proche de 0, tq

$f(\lambda(\varepsilon), \varepsilon) = 0 = p_{A(\varepsilon)}(\lambda(\varepsilon))$ et $\lambda(\varepsilon)$ est une val. propre de $A(\varepsilon)$ $\forall \varepsilon$ assez petit.

Il reste à construire un vect. p. correspondant $v(\varepsilon)$, dérivable par rapport à ε.

Comme $\lambda(\varepsilon)$ est une val. prop. de $A(\varepsilon)$, il existe $v(\varepsilon) \neq 0$ tq

$v(0) = v_*$ $\quad (A(\varepsilon) - \lambda(\varepsilon) I)\, v(\varepsilon) = 0$.

Comme $\partial f/\partial\lambda\,(\lambda(\varepsilon), \varepsilon)$ est non nul en $\varepsilon = 0$, cela reste vrai $\forall \varepsilon$ assez petit, par continuité par rapport à ε.

Ainsi, $\lambda(\varepsilon)$ est de multiplicité simple et la dimension de l'esp. prop. associé est 1. $\dim \ker(A(\varepsilon) - \lambda(\varepsilon) I) = 1$

On peut donc fixer une des composantes non nulles de $\lambda(\varepsilon)$ à 1 et utiliser la règle de Cramer pour calculer le reste.

Ainsi, les composantes de $v(\varepsilon)$ sont des fct rationelles de ε, donc dérivables.

On a donc construit $\lambda(\varepsilon)$ et $v(\varepsilon) \neq 0$ définis $\forall \varepsilon$ assez petit tq $A(\varepsilon)\, v(\varepsilon) = \lambda(\varepsilon)\, v(\varepsilon)$ avec $v(0) = v_*$ et $\lambda(0) = \lambda_*$

On dérive par rap. à ε et on fait $\varepsilon = 0$:

Pr : (Suite)

$(A - \lambda_* I) v'(0) = (E - \lambda'(0) I) v_*$

On multiplie à gauche par u_*^H (Vect prop à gauche pour λ_* de A)

$u_*^H E v_* - \lambda'(0) u_*^H v_* = 0$ car $u_*^H (A - \lambda_* I) = 0$

Il reste à diviser par $u_*^H v_*$ pour obtenir le résultat :

$$\lambda'(0) = \frac{u_*^H E v_*}{u_*^H v_*}$$

En posant le cas particulier $E = u_* v_*^H$, on obtient $u_*^H E v_* =$

$\|u_*\|_2^2 \|v_*\|_2^2 \neq 0 \quad \Rightarrow u_*^H v_* \neq 0$

9.2 Méthode de la puissance

le 16.05.19

Le but de ce chapitre est de trouver numériquement les val. et vect. propres de $A \in \mathbb{C}^{n \times n}$.

L'algorithme que l'on étudie est l'algo. de la puissance :

$$y^{(k+1)} = A y^{(k)} \quad \text{où } y^{(0)} \in \mathbb{C}^n \text{ donné}$$

Notation : On note $\lambda_1, \dots, \lambda_n$ les val. propres de A et on suppose $|\lambda_1| \geq |\lambda_2| \geq \dots \geq |\lambda_n|$.

Les vect. propres à droite sont $v_1, \dots, v_n$ et à gauche $u_1, \dots, u_n$ ($u_i^H A = \lambda_i u_i^H$)

Hypothèses sur A :

a) $A \in \mathbb{C}^{n \times n}$ est diagonalisable, ie : $A = T \Lambda T^{-1}$ où $\Lambda = \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix}$ et $T = (v_1 \quad v_n)$ (Sinon, il faut utiliser la représ. de A dans la base de Jordan)

b) On suppose $|\lambda_1| > |\lambda_2|$

Hypothèse sur $y^{(0)}$:

c) On suppose que $u_1^H y^{(0)} \neq 0$

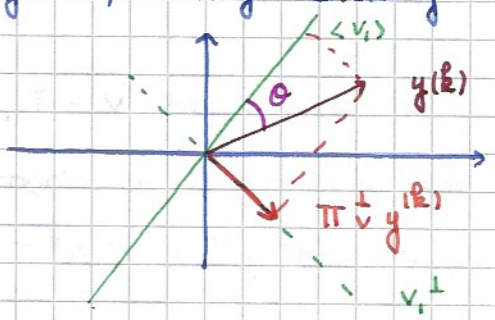
Il est difficile d'étudier $\| y^{(k)} - v \|_2$ car v n'est pas unique et cette valeur ne converge pas en général.

Ex : $A = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}$ $y^{(0)} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \dots, y^{(k)} = \begin{pmatrix} 2^k \\ 1 \end{pmatrix}$

La méthode de la puissance permet d'approximer v_1.

On étudie, comme critère de convergence, l'angle entre $y^{(k)}$ et v_1

$$\sin \Theta(y^{(k)}, v_1) = \frac{\| \Pi_{v_1}^{\perp} y^{(k)} \|_2}{\| y^{(k)} \|_2}$$



On commence par étudier le cas $A = \Lambda = \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix}$ est diagonale.

Lemme 1 : Soit $A = \Lambda = \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix}$, tq $|\lambda_1| > |\lambda_2|$.

Alors, si $y_1^{(0)} \neq 0$, la méthode de la puissance converge et on a $\sin \Theta(y^{(k)}, e_1) \leq C \cdot \left| \frac{\lambda_2}{\lambda_1} \right|^k$

Rem : les vect. propres de A sont $e_1, \dots, e_n$

Pv :

$$y^{(k)} = \Lambda^k y^{(0)} = \begin{pmatrix} \lambda_1^k y_1^{(0)} \\ \lambda_n^k y_n^{(0)} \end{pmatrix}$$

$$\Pi_{e_1}^{\perp} y^{(k)} = y^{(k)} - (e_1^T y^{(k)}) e_1 = \begin{pmatrix} 0 \\ y_2^{(k)} \\ y_n^{(k)} \end{pmatrix}$$

$$\sin \Theta(y^{(k)}, v_1)^2 = \frac{\left\| \begin{pmatrix} 0 \\ y_2^{(k)} \\ y_n^{(k)} \end{pmatrix} \right\|_2^2}{\| y^{(k)} \|_2^2} = \frac{\sum_{i=2}^n |y_i^{(k)}|^2}{\sum_{i=1}^n |y_i^{(k)}|^2} \leq \frac{\sum_{j=2}^n |y_j^{(k)}|^2}{|y_1^{(k)}|^2}$$

On obtient :

$$\sin \Theta(y^{(k)}, v_1) \leq \frac{\sqrt{\sum_{j=2}^n |\lambda_j|^k |y_j^{(0)}|^2}}{|\lambda_1|^k |y_1^{(0)}|} \leq \frac{\sqrt{|\lambda_2|^{2k} \sum_{j=2}^n |y_j^{(0)}|^2}}{|\lambda_1|^k |y_1^{(0)}|}$$

$$= \left| \frac{\lambda_2}{\lambda_1} \right|^k \underbrace{\frac{\sqrt{\sum_{j=2}^n |y_j^{(0)}|^2}}{|y_1^{(0)}|}}_{C}$$

□

Pour le cas général, on utilise le lemme suivant :

Lemme 2 : Soit $B \in \mathbb{C}^{n\times n}$ inversible et soient $x, y \in \mathbb{C}^n$, alors

$$\kappa_2(B)^{-1} \sin\Theta(x,y) \leq \sin\Theta(Bx, By) \leq \kappa_2(B) \sin\Theta(x,y)$$

Thm : Soit $A \in \mathbb{C}^{n\times n}$ diagonalisable et tq $|\lambda_1| > |\lambda_2|$.

Si $u_1^H y^{(0)} \neq 0$, alors la méthode de la puissance converge et on a l'estimation : $\sin\Theta(y^{(k)}, v_1) \leq \left|\frac{\lambda_2}{\lambda_1}\right|^k \kappa_2(T) \cdot C$

Pv : On peut prendre $v_1 = Te_1, \dots, v_n = Te_n$ et $u_1 = T^{-H} e_1, \dots, u_n = T^{-H} e_n$

Pour se ramener au cas diagonal, on pose $z^{(k)} = T^{-1} y^{(k)}$

$y^{(k)} = A^k y^{(0)} = T \Lambda^k T^{-1} y^{(0)}$, donc $z^{(k)} = \Lambda^k z^{(0)}$

Pour appliquer le lemme 1, il faut $z_1^{(0)} \neq 0$:

$z_1^{(0)} = e_1^T z^{(0)} = u_1^H T T^{-1} y^{(0)} = u_1^H y^{(0)} \neq 0$ par hypothèse.

Donc, on obtient $\sin\Theta(z^{(k)}, e_1) \leq C \left|\lambda_2/\lambda_1\right|^k$

On applique le lemme 2 avec $B = T$, bien inversible.

$\sin\Theta(\underbrace{Tz^{(k)}}_{y^{(k)}}, \underbrace{Te_1}_{v_1}) \leq \kappa_2(T) \sin\Theta(z^{(k)}, e_1) \leq \kappa_2(T)\, C \cdot \left|\lambda_2/\lambda_1\right|^k$ ∎

On observe que l'évolution de $y^{(k)}$ est donnée majoritairement par λ_1 :

$z^{(k)}$ représente le vect. $y^{(k)}$ dans la base $\langle v_1, \dots, v_n \rangle$ car $y^{(k)} = T z^{(k)}$

on a $y^{(k)} = \sum_{i=1}^n v_i z_i^{(k)}$

$$z^{(k)} = \Lambda^k z^{(0)} = \begin{pmatrix} \lambda_1^k z_1^{(0)} \\ \lambda_n^k z_n^{(0)} \end{pmatrix} = \lambda_1^k z_1^{(0)} \begin{pmatrix} 1 \\ \lambda_2^k/\lambda_1^k \, z_2^{(0)}/z_1^{(0)} \\ \vdots \end{pmatrix}$$

$$= \lambda_1^k z_1^{(0)} \cdot \begin{pmatrix} 1 \\ O(|\lambda_2/\lambda_1|^k) \\ \vdots \\ O(|\lambda_n/\lambda_1|^k) \end{pmatrix}$$

On obtient pour $y^{(k)}$:

$$y^{(k)} = \lambda_1^k z_1^{(0)} \left(v_1 + \sum_{i=2}^n v_i \, O(|\lambda_i/\lambda_1|^k) \right)$$

On déduit : $\left\| v - \frac{1}{\lambda_1^n z_1^{(0)}} y^{(k)} \right\|_2 = O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right)$

Rem : La condition $|\lambda_1| > |\lambda_2|$ est importante.

p. ex : $A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, $\lambda_1 = -1$, $\lambda_2 = 1$

On note $y^{(0)} = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$, $A \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \beta \\ \alpha \end{pmatrix}$

$\Rightarrow y^{(2k)} = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$ $y^{(2k+1)} = \begin{pmatrix} \beta \\ \alpha \end{pmatrix}$ $k \in \mathbb{N}$

Rem : La condition $u_1^H y^{(0)} \neq 0$ est "toujours" vérifiée en pratique si on choisit $y^{(k)}$ aléatoirement.

On cherche maintenant à approcher numériquement λ_1 à l'aide des $y^{(k)}$.

On veut minimiser la valeur $\| A y^{(k)} - \lambda^{(k)} y^{(k)} \| = \| y^{(k)} \lambda^{(k)} - A y^{(k)} \|_2$

Il s'agit d'un problème des moindres carrés avec n équations et une inconnue $\lambda^{(k)}$.

On utilise les équations normales :

$$y^{(k)H} y^{(k)} \lambda^{(k)} = y^{(k)H} A y^{(k)} \Rightarrow \lambda^{(k)} = \frac{y^{(k)H} A y^{(k)}}{y^{(k)H} y^{(k)}}$$

On appelle cette approximation le quotient de Rayleigh.

Thm : Sous les mêmes conditions que le théorème précédent, on a l'estimation : $\| \lambda^{(k)} - \lambda_1 \| = O\left(\left| \frac{\lambda_2}{\lambda_1} \right|^k \right)$

Si de plus A est normale ($A^H = A$), avec des vect. p. orthogonaux, alors on a $\| \lambda^{(k)} - \lambda_1 \| = O(|\lambda_2/\lambda_1|^{2k})$

Pr : On rappelle que $y^{(k)} = \lambda_1^k z_1^{(0)} \left(v_1 + \sum_{i=2}^n O\left(\left| \frac{\lambda_2}{\lambda_1} \right|^k \right) v_i \right)$

On en déduit :

$$y^{(k)H} A y^{(k)} = y^{(k)H} y^{(k+1)} = \lambda_1^k \overline{z_1^{(0)}}^H \left(v_1 + \sum_{i=2}^n O\left(\left| \frac{\lambda_2}{\lambda_1} \right|^k \right) v_i \right) \lambda_1^{k+1} z_1^{(0)} \cdot \left(v_1 + \sum_{j=2}^n O\left(\left| \frac{\lambda_2}{\lambda_1} \right|^{k+1} \right) v_j \right)$$

$$= |\lambda_1|^{2k} \lambda_1 |z_1^{(0)}|^2 \left(\|v_1\|_2^2 + \sum_{i=2}^n O\left(\left| \frac{\lambda_2}{\lambda_1} \right|^k \right) v_i v_1^H + \sum_{j=2}^n O\left(\left| \frac{\lambda_2}{\lambda_1} \right|^{k+1} \right) v_1^H v_j + \sum_{i=2}^n \sum_{j=2}^n O\left(\left| \frac{\lambda_2}{\lambda_1} \right|^{2k+1} \right) v_i^H v_j \right)$$

$$y^{(k)H} y^{(k)} = |\lambda_1|^{2k} |z_1^{(0)}|^2 \left(\|v_1\|_2^2 + O\left(\left| \frac{\lambda_2}{\lambda_1} \right|^k \right) \right)$$

Pv : (Suite)

$$\lambda^{(k)} = \frac{y^{(k)H} A y^{(k)}}{y^{(k)H} y^{(k)}} = \frac{\lambda_1 |\lambda_1|^{2k} |z_1^{(0)}|^2 (\|V_1\|_2^2 + O(|\lambda_2/\lambda_1|^k))}{|\lambda_1|^{2k} |z_1^{(0)}|^2 (\|V_1\|_2^2 + O(|\lambda_2/\lambda_1|^k))}$$

$$= \lambda_1 (1 + O(|\lambda_2/\lambda_1|^k)$$

$$\frac{1}{\|V_1\|_2^2 + O(|\lambda_2/\lambda_1|^k)} = \frac{1}{\|V_1\|_2^2} \left(\frac{1}{1 + O(|\lambda_2/\lambda_1|^k)} \right) = \frac{1}{\|V_1\|_2^2} \sum_{n \geq 0} O(|\lambda_2/\lambda_1|^k)^n$$

Si A est normale, $V_i V_1^H$ et $V_j V_1^H$ s'annulent.

□

Algo : Méthode de la puissance

le 23.05.19.

Pour calculer la valeur propre de plus grand module (supp unique)

$y^{(0)}$ donné $\in \mathbb{R}^n$ (ou $\mathbb{C}^n$) avec $\|y^{(0)}\|_2 = 1$

for $k = 0, 1, 2 \ldots$

$$\omega = A y^{(k)}$$

$$\lambda^{(k)} = \omega^H y^{(k)}$$

$$y^{(k+1)} = \frac{\omega}{\|\omega\|_2}$$

end

Rem : Si $|\lambda_1| \simeq |\lambda_2|$ la convergence de la méthode (qui est en $O(|\frac{\lambda_2}{\lambda_1}|^k)$ ↙ $2k$ si A normale) sera très lente.

L'algo de la puissance ne calcule que la valeur propre de plus grand module.

9.3 Méthode de la puissance inverse :

On suppose qu'on connaît une approximation σ d'une val. propre cherchée λ_i (pas forcément celle de plus grand module)

Idée : Appliquer formellement la méthode de la puissance à $(A-\sigma I)^{-1}$ sans calculer explicitement $(A-\sigma I)^{-1}$!

Regardons qu'elles en sont les val. propres en fonction des val. p. $\lambda_1, \dots \lambda_n$ de A :

$$Av = \lambda v \iff (A-\sigma I)v = (\lambda-\sigma)v \iff (A-\sigma I)^{-1}v = (\lambda-\sigma)^{-1}v$$

$$\Rightarrow \tilde{\lambda} = \frac{1}{\lambda-\sigma}$$ en supposant $A-\sigma I$ inversible (ie : $\sigma \neq \lambda_i \ \forall i$)

Les val. p. de $(A-\sigma I)^{-1}$ sont les valeurs $\frac{1}{\lambda_i-\sigma}$ avec les même vect. p. que A.

Supposons que λ_i est la val. p. de A la plus proche de σ est unique. On note λ_I la 2ème val p. la + proche de σ

$$|\sigma-\lambda_i| < |\sigma-\lambda_I| \leq |\sigma-\lambda_j| \quad \forall j \neq i$$

$$\frac{1}{|\sigma-\lambda_i|} > \frac{1}{|\sigma-\lambda_I|} \geqslant \frac{1}{|\sigma-\lambda_j|} \quad \forall j \neq i$$

↑ val. p. de + grd module de $(A-\sigma I)^{-1}$

Si $\sigma \simeq \lambda_i$, on obtient : $\left|\frac{1}{\sigma-\lambda_i}\right| \gg \frac{1}{|\sigma-\lambda_I|}$

et la convergence est rapide, car déterminée par

$$\frac{|\sigma-\lambda_I|^{-1}}{|\sigma-\lambda_i|^{-1}} = \left(\frac{\sigma-\lambda_i}{\sigma-\lambda_I}\right) \ll 1$$

Algo : Méthode de la puissance inverse

(Pour calculer la val p. la + proche de σ donné)

Soit donné $y^{(0)} \in \mathbb{K}^n$ avec $\|y^{(0)}\| = 1$ et $\sigma \simeq \lambda_i$

Jsq $k = 0, \dots$: Résoudre $(A-\sigma I)\,w = y^{(k)}$ $\left(\Leftrightarrow w = (A-\sigma I)^{-1} y^{(k)}\right)$

$$y^{(k+1)} = w/\|w\| \quad , \quad \lambda^{(k+1)} = \left(y^{(k+1)}\right)^H A\, y^{(k+1)}$$

Rem: On peut calculer au départ une décomposition LU de A, à réutiliser à chaque itération.

9.3 Décomposition en valeurs singulières (SVD)

Thm: (SVD)

Soit $A \in \mathbb{R}^{m\times n}$ (ou $\mathbb{C}^{m\times n}$), $\exists U, V$ orthogonales (unitaires) telles que

$A = U \Sigma V^H$ avec $\Sigma = \begin{pmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_{\min(n,m)} \\ & 0 & \end{pmatrix} \in \mathbb{R}^{m\times n}$

où $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_{\min(n,m)} > 0$

Les σ_i sont les valeurs singulière de A $(\in \mathbb{R}^+)$

Les colonnes v_i de V sont les vect. sing à droite,

—— " —— u_i de U —— " —— à gauche

(P_V) ? On suppose $m \geq n$ (sinon on regarde A^H)

Rem: On peut calculer la décomp. SVD en diagonalisant deux matrices sym. réelles. En effet, supposons $A = U\Sigma V^H$, alors $A^HA = V\Sigma^H\Sigma V^H$ et $AA^H = U\Sigma\Sigma^H U^H$ avec $\Sigma^H\Sigma = \text{diag}(\sigma_1^2, \dots, \sigma_n^2) \in \mathbb{R}^{n\times n}$ et $\Sigma\Sigma^H = \text{diag}(\sigma_1^2, \dots, \sigma_n^2, 0, \dots, 0) \in \mathbb{R}^{m\times m}$

Rem: Comme A^HA (et aussi AA^H) est sym. réelle semi déf. positive, ses val. propres sont réelles positives, ce sont les $\sigma_1^2, \dots, \sigma_n^2$

Rem: On peut aussi diagonaliser $\begin{pmatrix} 0 & A^H \\ A & 0 \end{pmatrix}$ pour retrouver la décomp. SVD

Rem: Σ dans la décomp SVD est unique grâce à $\sigma_1 \geq \dots \sigma_n \geq 0$.

(Contrairement à D matrices diag des val. p. de A)

Prop: SVD, Soit $A \in \mathbb{R}^{m\times n}$, $A = U\Sigma V^H$

1) $Av_i = \sigma_i u_i$ et $u_i^H A = \sigma_i v_i^H$

2) $\|A\|_2 = \sigma_1$

Preuve : $\|A\|_2$ est la + grande val. propre en module de $A^H A$

ou

$\|A\|_2 = \|U\Sigma V^H\|_2 = \|\Sigma\|_2 = \sigma_1$

Prop : (Suite)

3) Rang(A) = nbr de val. sing. non nulles.

Comme U et V^H sont inversibles, c'est aussi le rang de $U\Sigma V^H$

4) Soit r = rang A, alors $\mathrm{Im}(A) = \mathrm{Span}\{u_1, \dots, u_r\}$

$\ker(A) = \mathrm{Span}\{v_{r+1}, \dots, v_n\}$

Rappel : Thm du rang

dim Ker A + dim Im A = dim espace de départ

5) Soit r = rang A, alors A peut se décomposer comme une combinaison linaire de matrices de rang 1 : $A = \sum_{i=1}^{r} \sigma_i u_i v_i^H$

Pr : utiliser $A = U\Sigma V^H$

Rem : $u_i v_i^H \in \mathbb{R}^{m\times n}$, $\ker(u_i v_i^H) = v_i^\perp = \{v \in \mathbb{C}^n : v_i^H v_i = 0\}$

$\mathrm{im}(u_i v_i^H) = \mathrm{Vect}(u_i)$ ← $\mathrm{Rang}(u_i v_i^H) = 1$

Rem : Généralement, / Génétiquement, A se décomp. comme une composition d'une rot. V^H, une dilatation Σ, une rot U.

Rem : On a vu $\sigma_1 = \|A\|_2$, ainsi pour A inversible, $\sigma_n = \|A^{-1}\|_2$

Ainsi, $K_2(A) = \sigma_1/\sigma_n$ pour A inversible

Déf : Pour $A \neq 0 \in \mathbb{R}^{m\times n}$, on définit $K_2(A) = \sigma_1/\sigma_r$

Pseudo inverse de Moore - Penrose :

$A = U\Sigma V^H \in \mathbb{R}^{m\times n}$

$A^+ = V\Sigma^+ U^H$ avec $\Sigma^+ = \begin{pmatrix} \sigma_1^{-1} & & & \\ & \ddots & & \\ & & \sigma_r^{-1} & \\ & & & 0 \end{pmatrix}$

On peut vérifier que pour $Ax = b$ surdéterminé, $A \in \mathbb{R}^{m \times n}$, $m \geq n$

$\text{rang}(A) = n$, $x^* = A^+ b$ donne en fait la solution du problème

de moindre carré $\min_{x \in \mathbb{R}^n} \| Ax - b \|_2 = \| Ax^* - b \|_2$