ALGORITHMIQUE NUMÉRIQUE

Johann PAVIS
(selon source : B. PERROT)

Table des matières

Première partie

Systèmes Linéaires

Chapitre 1

Conditionnement

1.1 Généralité

Lorsque l'on effectue une résolution numérique d'un système de la forme $A.X = b$, avec $A \in \mathcal{M}_n(\mathbb{R})$ et b un vecteur colonne de n termes, d'infimes variations de la matrice A peuvent induire de fortes différences dans les résultats calculés.

Ceci est un problème de stabilité des calculs.

<u>Définition</u> 1 *Soit $A.X = b$ un système linéaire que l'on souhaite résoudre. Soit ΔA une légère perturbation de la matrice A.*

On note X' la nouvelle solution du système $(A+\Delta A).X = b$. On a donc : $X' = X + \Delta X$

On dit que les calculs sont stables lorsque : $\dfrac{||\Delta X||}{||X||} \leqslant \mathbf{coeff}\dfrac{|||\Delta A|||}{|||A|||}$.

Le coefficient **coeff** *est* $\mathbf{coeff} = \mathbf{cond}(A)$, *qui est appelé* conditionnement de la matrice A.

<u>Théorème</u> 1 *Soit $A.X = b$ un système linéaire. Soit ΔA une perturbation de la matrice A.*

On a alors : $\dfrac{||\Delta X||}{||X||} \leqslant \mathbf{cond}(A).\dfrac{|||\Delta A|||}{|||A|||}$

1.2 Retour sur la norme triple d'une matrice

Soient $A \in \mathcal{M}_n(\mathbb{K})$ et $X \in \mathbb{K}^n$.

Par définition, on a : $|||A||| = \sup\limits_{X \neq \vec{0}} \dfrac{||A.X||}{||X||} = \sup\limits_{||X||=1} ||A.X|| = \sup\limits_{||X|| \leqslant 1} ||A.X||$

On note :

$$||X||_1 = \sum_{i=1}^{n} |x_i|$$

$$||X||_\infty = \max_{i \in [1,n]} |x_i|$$

$$||X||_2 = \sqrt{\sum_{i=1}^{n} |x_i|^2} = \sqrt{< X, X >}$$

On a donc les trois normes triples associées :

$$|||A|||_1 = \max_j \left(\sum_{i=1}^{n} |a_{i,j}| \right)$$

$$|||A|||_\infty = \max_i \left(\sum_{j=1}^{n} |a_{i,j}| \right)$$

On sait que : $A.A*$ est une matrice symétrique réelle. Ces valeurs propres, soient α_k sont donc toutes dans $\mathbb{R}^+$. On pose donc : $\mu_k = \sqrt{\alpha_k^2}$.

Si A est autoadjointe ($A = A*$), alors les α_i sont les valeurs propres de A^2. On peut donc écrire :

$$\alpha_k = \lambda_k^2$$

Ainsi : $\forall k$, $\mu_k = |\lambda_k|$

On en déduit : $|||A|||_2 = \max\limits_k |\lambda_k| = \max\limits_k \mu_k = \rho_A$

Propriétés 1 *Soit $A \in \mathcal{M}_n(\mathbb{R})$. On a alors :*

$$|||A|||_1 = \max_j \left(\sum_{i=1}^{n} |a_{i,j}| \right)$$

$$|||A|||_\infty = \max_i \left(\sum_{j=1}^{n} |a_{i,j}| \right)$$

$$|||A|||_2 = \max_k |\lambda_k| = \max_k \mu_k = \rho_A$$

(on appelle ρ_A le rayon spectral de la matrice *A)*

Supposons $||X||_2 = 1$. On a alors : $||A.X||_2^2 =< A.X, A.X >=< X, A* .A.X >$

Soit $e = (e_1, \ldots, e_n)$ une base orthonormée de vecteurs propres de $A * .A$

On a donc : $X = \sum_{i=1}^{n} x_i.e_i$

Et donc : $A * .A.X = \sum_{i=1}^{n} \alpha_i.x_i.e_i$

On en déduit : $||A.X||_2^2 =< \sum_{i=1}^{n} x_i.e_i, \sum_{i=1}^{n} \alpha_i.x_i.e_i >$

Soit : $||A.X||_2^2 \leqslant \alpha \sum_{i=1}^{n} |x_i|^2 = \alpha.||X||_2^2 = \alpha = \max_i \alpha_i$

Donc : $||A.X|| \leqslant \sqrt{\alpha} = \max_{i \in [1,n]} \mu_i$

Propriété 2 *Soient $A \in \mathcal{M}_n(\mathbb{K}) et X \in \mathbb{K}^n$.*

On a alors : $||A.X|| \leqslant |||A|||.||X||$

Soit $B \in \mathcal{M}_n(\mathbb{K})$. On a : $|||A.B||| \leqslant |||A|||.|||B|||$

Définition 2 *Soit $A \in \mathcal{GL}_n(\mathbb{K})$, c'est à dire A matrice inversible.*

On a alors par définition : $\mathbf{cond}(A) = |||A|||.|||A^{-1}|||$

Propriétés 3 *Soit $A \in \mathcal{M}_n(\mathbb{K})$. On a alors :*
- $\forall \alpha \neq 0 \ , \quad \mathbf{cond}(\alpha.A) = \mathbf{cond}(A)$
- $\mathbf{cond}(A) \geqslant 1 \qquad (A.A^{-1} = \mathrm{Id} \Rightarrow 1 = |||\mathrm{Id}||| \leqslant |||A|||.|||A^{-1}|||)$
- $\mathbf{cond}(A) = 1 \Leftrightarrow \exists Q \in \mathcal{O}_n(\mathbb{K}) \ , \quad A = \alpha.Q$
- $\mathbf{cond}_2(A) = \dfrac{\max_i \mu_i}{\min_i \mu_i}$

On a de plus, si A est une matrice inversible, et si ΔA est une perturbation de A, pour le système linéaire $A.X = b$:

$$\frac{||\Delta X||}{||X||} \leqslant \mathbf{cond}(A).\frac{||\Delta b||}{||b||}$$

Chapitre 2

Résolution

2.1 Méthodes directes

2.1.1 Système triangulaire

Il s'agit de système de la forme $A.X = b$, avec A une matrice *triangulaire supérieure*, c'est à dire $A \in \mathcal{T}_n^+(\mathbb{K})$.

On obtient alors un système de la forme :

$$\left\{ \begin{array}{rcl} a_{N,N}.x_N & = & b_N \\ a_{N-1,N-1}.x_{N-1} + a_{N-1,N}.x_N & = & b_{N-1} \\ \vdots & & \end{array} \right.$$

On choisit le *Horner* comme unité (il correspond à une addition et une multiplication). La complexité de cette méthode est alors :

$$1 + 2 + \ldots + N - 1 = \mathcal{O}(N^2)$$

2.1.1.1 Utilisation des déterminants

On doit calculer N déterminants (un pour chacune des N inconnues).

On obtient donc une complexité $C(N)$ de la forme :

$$C(N) = N.C(N-1) = N!$$

On voit donc que cette méthode est inutilisable du point de vue pratique, car elle est beaucoup trop longue. **IL NE FAUT PAS L'UTILISER**.

2.1.2 Pivot de Gauss

Il s'agit de se ramener à un système triangulaire supérieur, en partant du système $A.X = b$.

Pour cela, on remplace l'équation numéro i de ce système (i variant de 1 à N) par : $ligne\ i - \dfrac{a_{i,i}}{a_{1,1}}.ligne\ i$

Ceci a pour effet d'éliminer la variable x_1 de toutes les lignes, **à l'exception de la première ligne**.

On réitère ensuite le procédé sur les lignes suivantes. On élimine alors successivement x_2, x_3, ...

Remarque Si $a_{i,1} = 3.\varepsilon$ ("zéro machine"), on peut avoir un problème.

On peut donc adopter *la stratégie du pivot avec seuil*, ou encore celle du *pivot avec maximum partiel* (on prend le plus grand coefficient de chaque ligne).

La complexité est en $\mathcal{O}(N^3)$.

2.1.3 Factorisation $P.A = L.U$

2.1.3.1 Présentation

Il s'agit de trouver une matrice P de permutation. La matrice U est la matrice finale obtenue après application de la méthode du pivot de Gauss. L est la matrice des coefficients utilisés lors du pivot Gauss.

Le système devient alors : $P.A.X = P.b$
Soit : $L.U.X = P.b$

On résout alors le système : $P.b = L.Y$, avant de résoudre : $Y = U.X$

La complexité en en $\mathcal{O}(N^2)$.

2.1.3.2 Résolution

On note A_k la matrice dont la ligne i est remplacée par : ligne $i - \dfrac{a_{i,k}}{a_{k,k}}$ligne k

On pose : $l_{i,k} = \dfrac{a_{i,k}}{a_{k,k}}$

Alors : $A_{k+1} = P_k.A_k - l_k.(P_k.A_k)$, l_k désignant la matrice dont les coefficients des lignes de numéro inférieur à k sont tous nuls et dont les autres autres coefficients (ligne $i > k$, colonne k) valent $l_{i,k}$.

On pose : $L_k = l_k + I_N$. Seule la ligne i intervient, on peut donc écrire :

$$A_{k+1} + l_k.A_{k+1} = P_k.A_k \Longleftrightarrow L_k.A_{k+1} = P_k.A_k$$

A l'étape k, considérons $k' < k$. On pose alors : $L^{'}_{k'} = P_k L_{k'} P_k$.
Ce qui est équivalent à : $P_k L^{'}_{k'} = P_k L_{k'}$
On obtient alors les $L_{k'}$ comme suit : $L_{k'} = I + l_{k'}$

$L^{'}_{k'}$ est la matrice $L_{k'}$ dans laquelle les coefficients de la ligne k, colonne k', et ligne i, colonne k', ont été inversés.

2.1.4 Factorisation de Choleski

Il s'agit du cas où la matrice A est symétrique. On pose alors $A = B.^tB$.

La factorisation de Choleski consiste à trouver les matrices L et D telles que :

1. $A = L.D.^tL$
2. $L \in \mathcal{T}^{-}_N(\mathbb{K})$, *i.e.* L est une matrice **triangulaire inférieure**
3. $\forall i \in [1, N],\ l_{i,j} = 1$
4. $D \in \mathcal{D}_N(\mathbb{K})$, *i.e.* D est une **matrice diagonale**

Le système à résoudre $A.X = b$ est donc équivalent à celui-ci :

$$L.D.^tL.X = b$$

On doit donc résoudre : $L.D.Z = b$ et $L.Y = b$

Les complexité de résolutions sont les suivantes :

$$\begin{cases} L.Y = b & \longrightarrow & \frac{N^2}{2} \\ D.Z = Y & \longrightarrow & N \\ ^tL.X = Z & \longrightarrow & \frac{N^2}{2} \end{cases}$$

On sait aussi $A = B.^tB = L.D.^tL$. Donc on a :

$$\forall (i,j) \in [1, N]^2,\ a_{i,j} = \sum_{k=1}^{N} l_{i,k}.d_k.l_{j,k}$$

Résolution par ligne

Pour i donné, on cherche $l_{i,j}$ et d_i.
On suppose la ligne $i-1$ déjà faîte (*i.e.* on connaît déjà d_{i-1} et $l_{i-1,j}$). On a :

$$a_{i,j} = \sum_{k=1}^{j-1} li,k.d_k.l_{j,k} + l_{i,j}.d_j.l_{j,j}$$

$$l_{i,j} = \frac{1}{d_j} \times \left(a_{i,j} - \sum_{k=1}^{j-1} li,k.d_k.l_{j,k} \right)$$
$$d_i = a_{i,i} - \sum k = 1^{i-1} li,k^2.d_k$$

Propriété du profil 4 *On stocke tout ce qui suit, sur une ligne, le* ***premier coefficient non nul*** *de la matrice.*

Si $l_{i,j} \neq 0$ *alors (i,j) est dans le profil.*

(La preuve de cette propriété se fait par récurrence sur les colonnes)

2.2 Méthodes itératives

Il s'agit de résoudre les systèmes $A.X = b$ par une résolution en utilisant une suite X_n qui tend vers $\overline{X}$ solution su système.

On décompose la matrice A sous la forme : $A = M - N$, où $M \in \mathcal{GL}_N(\mathbb{K})$ (*i.e.* M inversible).

On a, pour n quelconque dans $\mathbb{N}$:

$$X_{n+1} = M^{-1}.N.X_n + M^{-1}.b \quad \Rightarrow \quad M.X_{n+1} - N.X_n = 0$$

Si $(X_n)_{n\in\mathbb{N}}$ converge vers l, pour la norme triple, alors : $(M - N).l = b$

On sait : $A.\overline{X} = b$ et $A = M - N$. On a donc de plus :

$$\forall n \in \mathbb{N},\ \left(X_{n+1} - \overline{X}\right) = (M^{-1}.N).(X_n - \overline{X}) = (M^{-1}.N)^{n+1}.(X_0 - \overline{X})$$

Théorème 2 *Soit $B \in \mathcal{M}_N(\mathbb{K})$ une matrice non nulle. On a note λ_i ces différentes valeurs propres. On a alors :* $\lim\limits_{n\to\infty} B^n = O \Leftrightarrow \max\limits_{i} |\lambda_i| < 1$

La preuve est aisée dans le cas où B est diagonalisable.

2.2.1 Méthode de Jacobi

On prend $A = D - E - F$, avec D une matrice **diagonale**.

On pose : $M = D$ et $N = E + F$. La composante i du système est alors :

$$a_{i,i}.X_{n+1,i} = -\sum_{j\neq i} a_{i,j}.X_{n,j}$$

2.2.2 Méthode de Gauss - Seidel

On prend $M = D - E$ et $N = F$, où E et F sont les mêmes matrices que celles de la méthode précédente.

La composante numéro i du système est alors :

$$a_{i,i}.X_{n+1,i} = -\left(\sum_{j=1}^{i-1} a_{i,j}.X_{n+1,j} + \sum_{j=i+1}^{N} a_{i,j}.X_{n,j}\right)$$

2.2.3 Méthode de relaxation

Pour $\omega \neq 0$ $(0 < \omega < 2)$, on pose : $M = \omega^{-1}.D - E$, avec les notations précédentes.

On a alors la relation suivante :

$$\frac{1}{w}a_{i,i}.X_{n+1,i} = -\left(\sum_{j=1}^{i-1} a_{i,j}.X_{n+1,j} - \left(\frac{1}{w} - 1\right) a_{i,i}.X_{n+1,i} + \sum_{j=1}^{N} a_{i,j}.X_{n,j} + b_i\right)$$

Dans la pratique, on prend, ρ_M désignant le rayon spectral d'une matrice M : $\omega = \dfrac{2}{1 + \sqrt{1 - \left(\rho_{D^{-1}.(E+F)}\right)^2}}$

On notera que dans la méthode de Gauss-Seidal on prend $\omega = 1$.

2.2.4 Méthode du gradient

Dans cette partie, la matrice A sera supposée *symétrique définie positive.*

On définit la fonction $\mathbf{J}$ par : $\forall X,\ \mathbf{J}(X) =< A.X, X > -2. < X, b >$

Théorème 3 $\overline{X}$ *vérifie* $A.\overline{X} = b$ ***ssi*** $\overline{X}$ *rend minimum* $\mathbf{J}$

*(*i.e. *:* $\forall\ Y$ *solution,* $\mathbf{J}(Y) \geqslant \mathbf{J}(\overline{X})$*)*

Preuve

Si $\overline{X}$ minimise $\mathbf{J}$, le vecteur des dérivées partielles de $\mathbf{J}$ appliquées en $\overline{X}$ est nul. On a donc : $\vec{grad}_{\mathbf{J}}(\overline{X}) = \vec{0}$

Et donc : $2.(A.\overline{X} - b) = \vec{0}$, soit $A.\overline{X} = b$

Réciproquement, si $A.\overline{X} = b$, on a : $\vec{grad}_{\mathbf{J}}(\overline{X}) = \vec{0}$

$\overline{X}$ est donc un extremum de $\mathbf{J}$. On pose $X = \Delta + \overline{X}$.

$\mathbf{J}(X) \ = < A.(\overline{X} + \Delta), \overline{X} + \Delta > - \ 2. < \overline{X} + \Delta, b >$

$\mathbf{J}(X) \ = \ \mathbf{J}(\overline{X}) + < A.\Delta, \Delta > \ \geqslant \ \mathbf{J}(\overline{X})$ (car A est est *positive*)

Le problème initial devient alors :

"*Déterminer* $\overline{X}$ *un minimum de* $\mathbf{J}$".

Soit X_0 quelconque. Pour $k \in [0, N-1]$, on pose $X_{k+1} = f(X_k)$, cette suite convergeant vers la solution recherchée.

La courbe de niveau de J est (pour X_k) : $\{X/J(X) = J(X_k)\}$

On cherche X_{k+1} sur la perpendiculaire à cette courbe de niveau. On aura donc X_{k+1} de la forme : $X_{k+1} = X_k + \mathit{coeff}.\vec{grad}_J$

Or, on a vu que : $\vec{grad}_J(X_k) = 2.(A.X_k - b) = 2.g_k$. Donc : $X_{k+1} = X_k - \theta_k.g_k$. Reste à déterminer les θ_k.

On sait : $J(X_{k+1}) = < A.(X_k - \theta_k.g_k) \ , \ X_k - \theta_k.g_k > -2. < X_k - \theta_k.g_k \ , \ b >$

Soit : $J(X_{k+1}) =$ polynôme de degré 2 en θ_k

La valeur de θ_k rendant ce polynôme minimal est : $\theta_k = \dfrac{< g_k \ , \ g_k >}{< A.g_k \ , \ g_k >}$

La suite $(X_k)_{k \in [0,N-1]}$ est donc la suite définie par : $\begin{cases} X_0 \ \text{ quelconque} \\ X_{k+1} = X_k - \theta_k.g_k \end{cases}$

où les θ_k sont définis par la formule précédemment donnée.

Formule

On note $\overline{X}$ la solution du problème. On a l'inégalité suivante :

$$< A.(X_{k+1} - \overline{X}) \ , X_{k+1} - \overline{X} > \ \leqslant \left(\frac{\mathbf{cond}(A) - 1}{\mathbf{cond}(A) + 1} \right)^2 \times < A.(X_{k+1} - \overline{X}) \ , \ X_{k+1} - \overline{X} >$$

On a également : $< A.U \ , \ U > = ||U||_A^2$

Donc : $||X_{k+1} - \overline{X}||_A \quad \leqslant \left(\dfrac{\mathbf{cond}(A) - 1}{\mathbf{cond}(A) + 1} \right) \times ||X_{k+1} - \overline{X}||_A$

On en déduit donc aisément :

$$||X_{k+1} - \overline{X}||_A \quad \leqslant \left(\frac{\mathbf{cond}(A) - 1}{\mathbf{cond}(A) + 1} \right)^{k+1} \times ||X_0 - \overline{X}||_A$$

Le problème $A.X = b$ est donc remplacé par le problème suivant : $\tilde{A}.X = \tilde{b}$, de sorte que $\tilde{A}$ ait un conditionnement "*bon*" et que l'on puisse passer de l'un à l'autre des problèmes facilement.

On prend : $C = E.E^*$, avec $E \in \mathcal{T}_n^+$ (E matrice triangulaire inférieure).

On pose $\tilde{A} = E^{-1}.A.E^{-t}$, $\tilde{b} = E.b$

Donc, si on a : $\tilde{A}.\overline{Y} = \tilde{b}$, alors on a : $\overline{X} = E^{-t}.\overline{Y}$

Calcul des Y_k

On choisit Y_0 quelconque. Ensuite, on pose : $\tilde{g_k} = \tilde{A}.y_k - \tilde{b}$

On obtient alors les $\tilde{\theta_k}$: $\theta_k = \dfrac{< \tilde{g_k} \ , \ \tilde{g_k} >}{< \tilde{A}.\tilde{g_k} \ , \ \tilde{g_k} >}$

D'où : $Y_k = Y_k - \tilde{\theta_k}.\tilde{g_k}$

Calcul des X_k

On prend donc X_0 quelconque. On utilise ensuite la formule de récurrence :

$$\begin{cases} g_k = A.X_k - b \\ h_k = C^{-1}.g_k \\ \tilde{\theta_k} = \frac{<g_k, h_k>}{<A.h_k, h_k>} \\ X_{k+1} = X_k - \tilde{\theta_k}.h_k \end{cases}$$

2.2.5 Méthode du gradient conjugué

On pose : $d_k = -g_k + \beta_k.d_{k-1}$ et $X_{k+1} = X_k + \alpha_k.d_k$

On cherche toujours à rendre $J(X_{k+1})$ minimum. On en déduit alors les valeurs de α_k et β_k qui correspondent :

$$\alpha_k = \frac{< g_k, g_k >}{A.d_k, d_k}$$

$$\beta_k = \frac{< g_{k+1}, g_{k+1} >}{< g_k, g_k >}$$

Si on suppose connus X_k, d_k et g_k. On peut calculer d_{k+1} et g_{k+1} :

$$g_{k+1} = A.X_{k+1} - b \qquad \text{et} \qquad d_{k+1} = -g_{k+1} + \beta_k.d_k$$

On prend X_0 quelconque. On pose : $g_0 = A.X_0 - b$,$h_0 = C^{-1}.g_0$ et $d_0 = -h_0$.

Ensuite, on applique les différentes formules de récurrences :

$$\alpha_k = \frac{< g_k, h_k >}{< A.d_k, d_k >} \qquad \text{et} \qquad X_{k+1} = X_k + \alpha_k.d_k$$

$$g_{k+1} = A.X_{k+1} - b \qquad h_{k+1} = C^{-1}.g_{k+1} \qquad \text{et} \qquad \beta_k = \frac{< g_{k+1}, h_{k+1} >}{< g_k, h_k >}$$

$$d_{k+1} = -h_{k+1} + \beta_k.d_k$$

Test d'arrêt

Soit ε fixé. On prend une suite X_n qui converge vers $\overline{X}$ la solution à notre problème. Le test d'arrêt est le suivant : $||X_n - \overline{X}|| < \dfrac{\varepsilon}{2}$

On en déduit que l'on s'arrête si : $||X_{n+1} - X_n|| < \varepsilon$ ou si $||A.X_n - b|| < \varepsilon$

2.2.6 Méthode des moindres carrés

On cherche à rendre $||A.X - b||_2$ minimum. On écrit les *équations normales* : $J = ||A.X - b||_2^2 = \displaystyle\sum_{i=1}^{m} \left(\sum_{j=1}^{n} a_{i,j}.x_j - b_i \right)^2$

On cherche J de sorte à avoir : $\forall k \in [1, n]$, $\dfrac{\partial J}{\partial x_k} = 0$

Chose que l'on peut récrire comme suit : ${}^tA.A.X ={}^t A.b$. Donc, si $X = \overline{X} + \Delta X$, on a : $J(X) = J(\overline{X}) + <^t A.A.\Delta, \Delta >$

Donc : $\forall X$, $J(X) \geqslant J(\overline{X})$

Il existe toujours au moins une solution. Elle est unique ssi $rg(A) = n$

2.2.7 Méthode de la factorisation $Q.R$

Soit $A \in \mathcal{M}_{n,m}(\mathbb{R})$. La factorisation $Q.R$ consiste à trouver une matrice $Q \in \mathcal{O}_n(\mathbb{R})$ et R un matrice rectangulaire, dont les n première lignes et colonnes forment une matrice triangulaire supérieure, les autres coefficients étant tous nuls.

On a alors la propriété suivante :

$\overline{X}$ rend minimum $||A.X - b||$ ssi $\overline{X}$ est solution de $\overline{R}.X = C_1$, où C_1 désigne les n premières composantes de ${}^tQ.b$

Preuve :

On a les relations suivantes :

$$||Q.R.X - b||_2^2 = ||{}^tQ.Q.R.R.X -^t Q.b||_2^2$$

$$||R.X -^t Q.b||_2^2 = \sum_{i=1}^{n} (A.X -^t Q.b)_i^2 + \sum_{i=n+1}^{m} (R.X -^t Q.b)_i^2$$

$$\implies \quad ||R.X - C_1||^2 + ||C_2||^2 > ||R.X - C_1||^2$$

Ceci achève la démonstration.

2.2.7.1 Obtention de $A = Q.R$

On va définir les *matrices de Householder*.

Soit u un vecteur de $\mathbb{R}^n$ qui vérifie : $||u||_2 = 1$.

On définit les matrices de Householder par la formule suivante :

Soit $\vec{v} = c.\vec{u}$, c un coefficient quelconque dans $\mathbb{R}^+$ ($c = \sqrt{2.\beta}$). On définit H_u par la formule : $H_u = I_n - \dfrac{v.^tv}{\beta}$

Ces matrices sont **symétriques** et **orthogonales** (${}^tH = H$ et ${}^tH.H = H.{}^tH = I$).

Propriété 5 *Soit $a \in \mathbb{R}^n$ tel que $a \neq \vec{0}$. On a alors : $\exists H$ matrice de Householder telle que* $H_a = \begin{pmatrix} \alpha \\ 0 \\ \vdots \\ 0 \end{pmatrix}$

Preuve :

On pose : $\beta = \alpha^2 - \alpha.a_1$, avec α quelconque.

Soit le vecteur $\vec{v} = \begin{pmatrix} a_1 - \alpha \\ a_2 \\ \vdots \\ a_n \end{pmatrix}$

On peut aisément vérifier que : $||v||^2 = \alpha^2 - 2.\alpha.a_1 + a_2^2 + \ldots$. De plus, on a : $H_c = c - \dfrac{v.^tv.c}{\beta}$

En posant $\gamma = \frac{^tv.c}{\beta}$, on obtient : $H_c = c - \gamma.v$

On pose $c = a$. On en déduit : $\gamma = \dfrac{1}{\beta}\left(-\alpha.a_1 + a_2^2 + a_n^2\right) = -\alpha.a_1 + \alpha^2 = 1$

On a donc : $H_a = a - v = \begin{pmatrix} a-(a-\alpha)=\alpha \\ 0 \\ \vdots \\ 0 \end{pmatrix}$

On obtient A par multiplication successives par des matrices de Householder, en partant d'une matrice A_0.

On prend la matrice H_2 de sorte que :

$$\begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & & & \\ \vdots & & \tilde{H}_2 & \\ 0 & & & \end{pmatrix} . \begin{pmatrix} \alpha_1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} \alpha_2 & 0 & 0 & \dots & 0 \\ 0 & \alpha_1 & \times & \dots & \times \\ \vdots & \vdots & \vdots & A_2 & \vdots \\ 0 & \times & \times & \dots & \times \end{pmatrix}$$

On a H_2 matrice symétrique orthogonale.

On obtient, au final : $A = A_0 = H_1.A_1 = \ldots = H_1 \ldots H_N.A_N$. On prend : $Q = \prod_{i=1}^{N} H_i$ et $R = A_N$

Comme on connaît déjà R, il reste à résoudre : $\overline{R}.X = C_1$, ce qui va donner directement C_1.

2.2.7.2 Lien entre $A = Q.R$ et l'orthonormalisation par la méthode de Shmidt

On suppose $rg(A) = n$. On choisit $(a_1, \ldots, a_n) \in \mathbb{R}^n$ linéairement indépendants.

On a $A = Q.R$. Donc, on a : $\forall (i,j) \in [1,n]$, $a_{i,j} = \sum_{k=1}^{j} q_k.r_{k,j}$

De même, pour un j quelconque dans $[1,n]$, on a : $a_j = \sum_{k=1}^{j} q_k.r_{k,j}$

Ce qui permet de déduire : $q_j = \left(a_j - \sum_{k=1}^{j} q_k.r_{k,j} \right) \times \dfrac{1}{r_{j,j}}$

Chapitre 3

Résolution de $F(X) = 0$

Dans ce problème, F est une fonction de $\mathcal{F}(\mathbb{R}^n, \mathbb{R}^n)$.

3.1 Méthode de la dichotomie

Ce cas concerne la dimension 1. Elle consiste à isoler la racine, par divisions successives de la longueur de l'intervalle d'étude.

On choisit tout d'abord a et b deux réels vérifiant : $F(a).F(b) < 0$

On pose : $m = \frac{|b-a|}{2}$[1] et on regarde si $F(m) = 0$.

Si ce n'est pas le cas, on réitère le procédé en modifiant l'intervalle d'étude :

- si $F(a).F(m) > 0$, on choisit l'intervalle $[m, b]$
- si $F(a).F(m) < 0$, on choisit l'intervalle $[a, m]$

Après k étapes, l'intervalle d'étude est de longueur : $\dfrac{b-a}{2^k}$

[1] Dans la pratique, on prend $a < b$

3.2 Méthode de Newton

C'est la méthode qui utilise *les tangentes.*

Étant donné le point x_n connu, on pose : $x_{n+1} = x_n - \dfrac{f(x_n)}{f'(x_n)}$

Cette méthode peut s'appliquer à des dimensions supérieures à 1.

On pose : $X = X - F(X) = \Phi(X)$

<u>Théorème</u> 4 *Soit Φ une fonction K-lipschitzienne, $K < 1$. On a donc :* $\forall (X_1, X_2) \ , \quad ||\Phi(X_1) - \Phi(X_2)|| \leqslant K.||X_1 - X_2||$.

On a alors : $\exists (X_n)_{n \in \mathbb{N}} \ , \quad \lim\limits_{n \to \infty} X_n = \overline{X} \quad$ *avec* $\quad \Phi(X) = \overline{X}$

Preuve :

On pose : $X_{n+1} = \Phi(X_n)$

Alors : $||X_{n+1} - \overline{X}|| = ||\Phi(X_n) - \Phi(\overline{X})|| \leqslant K.||X_n - \overline{X}||$

Ainsi : $||X_{n+1} - \overline{X}|| \leqslant K^{n+1}.||X_0 - \overline{X}||$

K étant strictement inférieur, cette inégalité termine la démonstration.

On utilise la formule de récurrence : $X_{n+1} = X_n - F(X_n)$

3.3 Méthode de Newton - Raphson

On pose, pour F une fonction à valeur vectorielle : $J = \left(\dfrac{\partial F_j}{\partial x_j} \right)_{j \in [1,n]}$

On obtient alors le système : $\left\{ \begin{array}{l} X_{n+1} = X_n - \Delta_n \\ J(\Delta_n) = F(X_n) \end{array} \right. \quad \rightarrow \quad$ système à résoudre

Avec : $\Delta_n = J^{-1}(X_n).F(X_n)$

Ainsi, à **l'ordre 2** : $\exists K \ , \quad ||X_{n+1} - \overline{X}|| \leqslant ||X_n - \overline{X}||^2$

3.4 Algorithme de Bairstow

Soit P le polynôme : $a_0 + a_1.X + \ldots + a_{n-1}.X^{n-1} + X^n$. On cherche les réels α et β tel que le polynôme $A(X) = X^2 + \alpha.X + \beta$ divise P. On obtient ainsi deux racines du polynôme P.

On sait alors que $P = A.Q$ et on réitère le procédé sur le nouveau polynôme Q ainsi déterminé.

On a : $P(X) = (X^2 + \alpha.X + \beta) \times Q(X) + r_1.X + r_0$

On note F la fonction de $\mathbb{R}^2$ dans $\mathbb{R}^2$ qui associe à tout couple (α,β) le couple (r_0,r_1). On cherche à résoudre l'équation : $F(\alpha, \beta) = (0, 0)$. On va pour cela utiliser la méthode de Newton - Raphson.

On choisit α_0 et β_0 quelconque dans $\mathbb{R}$, puis on calcule les termes suivants de cette suite par la formule de récurrence suivante : $(\alpha_{n+1}, \beta_{n+1}) = (\alpha_n, \beta_n) + \Delta_n$, où Δ_n est tel que : $J.\Delta_n = (r_{0,n}, r_{1,n})$

J est la matrice des dérivées partielles de F (matrice *Jacobienne* de F).

On a donc : $J = \begin{pmatrix} \frac{\partial r_0}{\partial \alpha} & \frac{\partial r_0}{\partial \beta} \\ \frac{\partial r_1}{\partial \alpha} & \frac{\partial r_1}{\partial \beta} \end{pmatrix}$

Calcul de la matrice J

On écrit les coefficients correspondants à chaque X^i :

$$\begin{array}{lll} X^0 & a_0 & = \beta.q_0 \\ X^1 & a_1 & = \alpha.q_0 + \beta.q_1 \\ X^2 & a_2 & = q_0 + \alpha.q_1 + \beta.q_2 \\ X^3 & a_3 & = q_1 + \alpha.q_2 + \beta.q_3 \\ \vdots & & \\ X^{n-3} & a_{n-3} & = q_{n-5} + \alpha.q_{n-4} + \beta.q_{n-3} \\ X^{n-2} & a_{n-2} & = q_{n-4} + \alpha.q_{n-3} + \beta \\ X^{n-1} & a_{n-1} & = q_{n-3} + \alpha \end{array}$$

On obtient ainsi un système à n équations et n inconnues.

On pose : $c_k = \dfrac{\partial q_{n-k}}{\partial \beta}$ et $d_k = \dfrac{\partial q_{n-k}}{\partial \alpha}$

On sait : $\dfrac{\partial q_{n-3}}{\partial \beta} = 0$

On en déduit : $d \begin{cases} c_3 = 0 \\ d_3 = -1 \end{cases}$ et $\begin{cases} c_4 = -1 \\ d_4 = q_{n-3} + \alpha \end{cases}$

Par récurrence, on montre facilement : $\begin{cases} c_i = -q_{n+2-i} - \beta.c_{i-2} - \alpha.c_{i-1} \\ d_i = -q_{n+1-i} - \beta.d_{i-2} - \alpha.d_{i-1} \end{cases}$

Ceci permet de calculer J. On obtient alors :

$$J = \begin{pmatrix} -\beta.d_n & -q_0 - \beta.d_n \\ -q_0 - \alpha.d_n - \beta.d_{n-1} & -q_1 - \beta.c_{n-1} - \alpha.c_n \end{pmatrix}$$

Il ne reste plus alors qu'à résoudre le système : $J.\begin{pmatrix} \Delta_1 \\ \Delta_2 \end{pmatrix} = \begin{pmatrix} r_0 \\ r_1 \end{pmatrix}$.

On obtient alors les valeurs de α et β à l'itération suivante : $\begin{pmatrix} \alpha' \\ \beta' \end{pmatrix} = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} - \begin{pmatrix} \Delta_1 \\ \Delta_2 \end{pmatrix}$

Chapitre 4

Valeurs propres et vecteurs propres

4.1 Conditionnement

Du fait des limites physiques du calcul (le *zéro machine* n'est pas le *zéro mathématique*), il arrive parfois que l'on calcule des valeurs propres nulles alors qu'elles ne le sont pas.

<u>Théorème</u> 5 *Soit $A \in \mathcal{M}_n(\mathbb{R})$ tel que A soit diagonalisable.*

On a donc : $\exists D \in \mathcal{D}_n(\mathbb{R})$, $\exists P \in \mathcal{GL}_n(\mathbb{R})$, $D = P^{-1}.A.P$.

On a alors : pour toute perturbation ΔA de la matrice A et pour toute valeur propre λ' de $A + \Delta A$, il existe une valeur propre λ de la matrice A qui vérifie : $|\lambda - \lambda'| \leqslant \mathbf{cond}(A) \times |||\Delta A|||$

Cas où A est symétrique :

On a : $\mathbf{cond}(A) = 1$. On trigonalise la matrice A par le biais de matrices de Houselder. On obtient une première matrice A_1 :

$$A_1 = \left(\begin{array}{c|c} 1 & 0 \\ \hline 0 & \tilde{H}_0 \end{array}\right) . \left(\begin{array}{c|c} A_{1,1} & A_{1,2} \\ \hline A_{2,1} & A_{2,2} \end{array}\right) . \left(\begin{array}{c|c} 1 & 0 \\ \hline 0 & \tilde{H}_0 \end{array}\right)$$

On a donc : $A_1 = \left(\begin{array}{c|ccccc} A_{1,1} & \alpha & 0 & \ldots & dots & 0 \\ \hline \alpha & & & & & \\ 0 & & & & & \\ \vdots & & \tilde{H}_0 & A_{2,2} & \tilde{H}_0 & \\ 0 & & & & & \end{array}\right)$

La première colonne vaut : $\tilde{H}_0 A_{2,1}$.

On a : $\alpha^2 = \sum_{i=1}^{n-1} A_{(2,2),i}^2 = ||A_{2,2}||^2$

Dans le cas où $\alpha^2 \neq 0$, on pose $\alpha = -\text{signe}(A_{(2,2),1}).\sqrt{\alpha^2}$. On pose alors : $\beta = \alpha^2 - \alpha.A_{(2,2),1}$.

Soit le vecteur $V = \begin{pmatrix} A_{(2,2),1} - \alpha \\ A_{(2,2),2} \\ \vdots \\ A_{(2,2),N} \end{pmatrix}$.

On a alors : $\tilde{H}_0.A_{(1,2),1}.\tilde{H}_0 = \left(Id - \frac{V.^tV}{\beta}\right).A_{2,2}.\left(Id - \frac{V.^tV}{\beta}\right)$

En développant, on obtient :

$$\tilde{H}_0.A_{(1,2),1}.\tilde{H}_0 = A_{2,2} - \frac{1}{\beta} \times (V.^tV.A_{2,2} + A_{2,2}.V.^tV) + \frac{V.^tV.A_{2,2}.V.^tV}{\beta^2}$$

En posant : $Y = A_{2,2}.V$ et $\lambda =^t V.Y$, on obtient :

$$\tilde{H}_0.A_{(1,2),1}.\tilde{H}_0 = A_{2,2} - \frac{1}{\beta} \times (Y.^tV + V.^tY) + \frac{\lambda.V.^tV}{\beta^2}$$

Et si on pose $Z = Y.^tV$, on a :

$$\tilde{H}_0.A_{(1,2),1}.\tilde{H}_0 = A_{2,2} - \frac{1}{\beta} \times (Z +^t Z) + \frac{\lambda.V.^tV}{\beta^2}$$

On réitère le procédé sur la "*matrice intérieure*". Au final, on obtient une matrice de la forme : $\begin{pmatrix} b_1 & c_1 & & \\ c_1 & b_2 & c_2 & \\ \vdots & & & c_{n-1} \\ \vdots & & & b_{n-1} \end{pmatrix}$, avec : $b = b_1 = \ldots = b_n$ et $c = c_1 = \ldots = c_{n-1}$

Lorsque l'on a les c_i nuls, on découpe le problème en deux sous-problèmes pour lesquels les c_j sont **non nuls**.

Polynôme de STURM :

On note M_i la matrice constitué par les i premières lignes de la matrice précédentes. On introduit alors les polynômes P_i, $i \in [1,n]$, tels que :

- $P_0(\lambda) = 1$
- $P_1(\lambda) = b_1 - \lambda \longrightarrow$ polynôme caractéristique de M_1
- $P_i(\lambda) = (b_i - \lambda).P_{i-1}(\lambda) - c_{i-1}^2.P_{i-2} \longrightarrow$ polynôme caractéristique de M_i

Ainsi, on a : $P_n(\lambda)$ est le **le polynôme caractéristique** de A.

De plus, le coefficient dominant du polynôme P_i est $(-1)^i\lambda^i$.

On a également la propriété suivante : $P_i(\lambda) = 0 \quad \Rightarrow \quad P_{i-1}(\lambda).P_{i+1}(\lambda) < 0$

P_i possède i racines réelles distinctes qui séparent les $i+1$ racines du polynôme P_{i+1}, lorsque i décrit $[1, n-1]$.

Propriété 6 *Soit $\mu \in \mathbb{R}$. On a alors :* $\textbf{signe}[P_i(\mu)] = \begin{cases} \textbf{signe}[P_i(\mu)] \text{ , si } P_i(\mu) \neq 0 \\ \textbf{signe}[P_{i-1}(\mu)] \text{ , sinon} \end{cases}$

On définit l'ensemble $E(i,\mu)$ de la manière suivante :

$$\forall i \in [1,n] \quad , \quad E(i,\mu) = \{+, \textbf{signe}[P_1(\mu)], \ldots, \textbf{signe}[P_i(\mu)]\}$$

On définit l'entier $N(i,\mu)$ comme étant le nombre d'inversions de signe dans l'ensemble $E(i,\mu)$. C'est donc le nombre de racines de P_i qui sont inférieures à μ.

Algorithme de bissection de Givens :

Cet algorithme permet de calculer les λ_i, à i fixé.

On choisit deux réels α et β de manière à avoir $\lambda_i \in [\alpha, \beta]$. On vérifie le nombre de racines comprises entre α et *beta* $(N(i,\beta) - N(i,\alpha))$ et on réduit l'intervalle d'étude jusqu'à obtenir la racine souhaitée.

4.2 Calcul de la plus grande valeur propre en module et des vecteurs propres associés

Théorème 6 *Soit q_0 un vecteur de $\mathbb{R}^n$ tel que q_0* est non orthogonal aux sous-espaces propres à gauche associés à λ_i, valeur propre d'une matrice A. *On a donc : $A^*.v = \overline{\lambda_1}.v$*

v est un vecteur propre gauche *de A^* pour λ_1 et u vecteur propre de A pour $\lambda_2 \neq \lambda_1$.*

On a donc : $< v, u >= 0$

$(\lambda_1 < u, v >=< u, \overline{\lambda_1}.v >=< u, A^*.v >=< A.u, v >= \lambda_2 < u, v >$
$\Rightarrow \; < u, v >= 0$ car $\lambda_1 \neq \lambda_2)$

On pose : $x_{k+1} = A.x_k$ et $q_{k+1} = \frac{x_{k+1}}{||x_{k+1}}$.

Pour j tel que $q_j \neq 0$, on veut $\frac{x_{k+1}(j)}{q_k(j)} \longrightarrow \lambda_i$.

On a donc : $\left(\frac{\overline{\lambda_i}}{|\lambda_i|}\right)^k .q_k \longrightarrow$ vecteurs propres

On suppose A diagonalisable. Soit u_i une base de vecteurs propres associés à A. On a donc : $q_0 = \sum_{i=1}^{n} \alpha_i.u_i = \alpha_1.u_1 + \sum_{i=2}^{n} \alpha_i.u_i$

(on suppose $\alpha_1 \neq 0$)

On a donc : $A^k.q_0 = \lambda_1^k.\alpha_1.u_1 + \sum_{i=2}^{n} \alpha_i.\lambda_i^k.u_i$

$$A^k.q_0 = \lambda_1^k. \left(\alpha_1.u_1 + \sum_{i=2}^{n} \left(\frac{\lambda_i}{\lambda_1}\right)^k .\alpha_i.u_i\right)$$

La somme est négligeable (on la pose égale à ε_k).

4.3 Calcul de la plus petite valeur propre en module

Soit μ un réel donné. On cherche à calculer la valeur propre la plus proche de μ. on a : "**vp**$(A-\mu.Id) = \mu - \lambda_i$ *est la plus petite valeur propre en module de $A - \mu.Id$*".

Chapitre 5

Interpolation polynomiale

<u>Théorème</u> 7 *Soit f une fonction continue sur un intervalle donné $[a, b]$.*

On découpe cet intervalle en N intervalles plus petits $[x_i, x_{i+1}]$ de sorte que :
- $i \in [0, N-1]$
- $x_0 = a$ *et* $x_N = b$

Dans ces conditions, on a l'affirmation suivante :

$$\exists! \quad P \in \mathbb{R}_n[X] \ , \quad \forall i \in [0, N] \ , \quad P(x_i) = f(x_i) = y_i$$

Ce polynôme P est le Nième **polynôme d'interpolation de Lagrange**. *Le $i$ème polynôme de* Lagrange *W_i est défini par :* $W_i(X) = \dfrac{\prod_{i \neq j} X - x_j}{\prod_{i \neq j} x_i - x_j}$

On a alors : $P(x) = \displaystyle\sum_{i=0}^{N} y_i.W_i(x)$

5.1 Calcul numérique d'une intégrale

On cherche à calculer l'intégrale : $I = \displaystyle\int_a^b f(x).dx$

5.1.1 Méthode des rectangles

On pose : $\mathbf{Re}(h) = h.\displaystyle\sum_{i=0}^{N} f\left(x_i - \frac{h}{2}\right)$

On a alors : $|I - \mathbf{Re}(h)| \leqslant \dfrac{|b-a|}{24}.M^2.h^2$, où : $M^k = \sup_{[a,b]}\{|f^{(j)}|/j \in [1,k]\}$.

Cette méthode est d'ordre 2.

5.2 Méthode des trapèzes

On pose : $\mathbf{T}(h) = h.\left(\dfrac{1}{2}f(x_0) + \dfrac{1}{2}f(X_N) + \displaystyle\sum_{i=1}^{N-1} f(x_i)\right)$

On a alors : $|i - \mathbf{T}(h) \leqslant \dfrac{|b-a|}{12}.M^2.h^2$

Cette méthode est du même ordre que la précédente.

5.3 Formule de Simpson

La méthode de Simpson consiste à faire une approximation de la courbe de f par des arcs de paraboles et non pas par des segments de droite comme dans les deux méthodes précédentes.

On pose : $\mathbf{S}(h) = \dfrac{h}{6}.\left[f(x_0) + f(x_N) + 2\displaystyle\sum_{i=1}^{N-1} f(x_i) + 4\sum_{i=1}^{N-1} f\left(x_i + \frac{h}{2}\right)\right]$

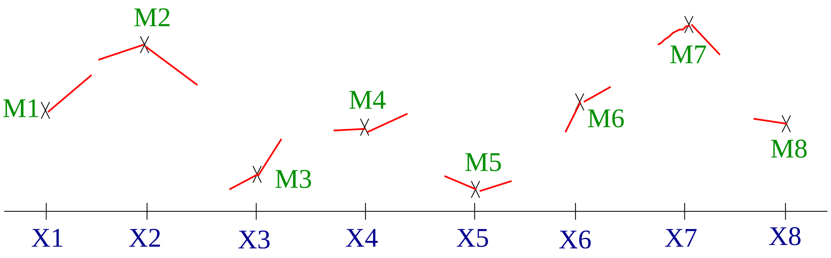
On a alors : $|I - \mathbf{S}(h)| \leqslant M^4.h^4$

Cette méthode est donc d'ordre 4.

5.4 Spline cubique

On cherche à déterminer les polynômes P_k tel que :

- $k \in [1,n]$
- P_k est de degré 3 sur $[x_{k-1}, x_k]$
- on effectue un raccordement $\mathcal{C}^2$ aux points x_k, $k \in [1, n-1]$
- $P_1(x_1) = P_n(x_n) = 0$



On a : $P_k''(x) = M_k \times \dfrac{x_{k-1} - x}{-h_k} + M_{k+1} \times \dfrac{x_k - x}{h_k}$

Donc : $P_k(x) = M_k \times \dfrac{(x_{k-1} - x)^2}{-6.h_k} + M_{k+1} \times \dfrac{(x_k - x)^2}{6.h_k} + C_k.(x_k - x) + D_k.(x_{k-1} - x)$

C_k et D_k sont donnés par les couples (x_k, y_k) et (x_{k-1}, y_{k-1}).

On obtient : $C_k = \dfrac{1}{6.h_k} \times \left(y_{k-1} - M_k.\dfrac{h_k^2}{6}\right)$ et $D_k = -\dfrac{1}{6.h_k} \times \left(y_k - M_k.\dfrac{h_k^2}{6}\right)$

On veut : $P_k'(x_k) = P_{k+1}'(x_k)$. On obtient alors une relation entre M_{k-1}, M_k et M_{k+1} $\longrightarrow$ équation linéaire

Le système à résoudre possède $n-2$ équations et $n-3$ inconnues. Sa matrice est la matrice symétrique $S = (s_{i,j})_{(i,j)}$ définie par :

- $\forall i \in [1, n-3]$, $s_{i,i} = \dfrac{h_{i+1} + h_{i+2}}{3}$
- $\forall i \in [1, n-3]$, $s_{i+1,i} = s_{i,i+1} = \dfrac{h_{i+1}}{6}$
- $s_{i,j} = 0$ sinon

Le second membre du système est, à la ligne k : $\dfrac{y_{k+1} + y_k}{h_k} - \dfrac{y_k - y_{k-1}}{h_k}$

5.5 Algorithme de Remes

Soit f une fonction $\mathcal{C}^0$ sur un intervalle $[a, b] \subset \mathbb{R}$. On utilisera ici la *norme infinie* : $||f||_\infty = \sup\limits_{x \in [a,b]} |f(x)|$

Soit $r \in \mathbb{R}$. On cherche à déterminer un polynôme P de degré r tel que :

$$\forall Q \in \mathbb{R}_r[X] \ , \ \ ||f - P||_\infty \leqslant ||f - Q||_\infty + \varepsilon$$

On choisit $r+2$ points de contrôle : $x_1, \dots, x_{r+2}$ de manière quelconque.

Équioscillation :

$\forall i \in [i, r+2]$, $(f - P)(x_i) = (-1)^i.\alpha$

On cherche P de la forme : $P(X) = \displaystyle\sum_{i=0}^{r} a_i.X^i$

On trouve donc : $f(x_i) = \displaystyle\sum_{j=0}^{r} a_j.x_i^j + (-1)^i.\alpha$

Dans le cas où le second membre est nul, on a toutes les inconnues qui sont nulles. On en déduit : $P(x_i) = (-1)^i.\alpha$

Théorème 8 *Soit α le réel précédemment défini. Soient f une fonction $\mathcal{C}^0$ et x_i l'un des points de contrôle précédemment défini. On a alors :*

$$\forall Q \in \mathbb{R}_r[X] \ , \quad |\alpha| \leqslant \max_i |f(x_i) - Q(x_i)|$$

Dans le cas contraire, on aurait : $(P-Q)(x_i) = (P-f)(x_i) - (Q-f)(x_i)$ (du signe de $(P-f)(x_i)$). Donc : $P = Q$

Lorsque l'on trouve : $||f - P||_\infty \leqslant |\alpha| + \varepsilon$, on s'arrête[1].

Procédure d'échange :

Dans le cas contraire du précédent, on a : $\exists y$, $|F(y) - P(y)| \geqslant |\alpha| + \varepsilon$

On remplace alors le x_i qui permet l'oscillation par le y défini précédemment. Puis on réitère.

[1]En effet : $||f - P||_\infty \leqslant |\alpha| + \varepsilon \leqslant \max_i |f(x_i) - P(x_i)| + \varepsilon$

Deuxième partie

Équations Différentielles

Chapitre 6

Problème de Cauchy

6.1 Définitions

On cherche à résoudre le problème suivant, où y est la fonction inconnue à déterminer ($y : [x_0, x_0 + a] \to \mathbb{R}$)et f une fonction $\mathcal{C}^0(\mathbb{R}, \mathbb{R})$:

$$\begin{cases} y'(x) &= f(x).y(x) \\ y(x_0) &= y_0 \end{cases}$$

De manière plus générale, on veut résoudre le système :

$$\begin{cases} y^{(p)} = y[x, y', y'', \ldots, y^{(p-1)}] \\ y(x_0) = y_0 \\ y'(x_0) = y_{1,0} \\ \vdots \\ y^{(p-1)}(x_0) = y_{p-1,0} \end{cases}$$

On définit le vecteur de vecteurs $z \in (\mathbb{R}^n)^n$ de la manière suivante :

$$\forall i \in [1, n] \ , \quad z_i = y^{(i)}$$

On pose que : $z'_{p-1} = y[x, y, y', \ldots, y^{(p-1)}] = g(x, z)$

On pose également : $y'(x) = f(x, y(x))$.

On a la relation suivante :

$$\forall (x, y_1, y_2) \in [x_0, x_0 + a]^3 \ , \quad \exists k \ , \quad ||f(x - y)|| \leqslant k.||y_1 - y_2||$$

Supposons connue y la solution. On pose : $x_n = x_0 + n.h$

On cherche alors y_n tel que $y_n \simeq y(x_n)$

Définition 3 *La méthode est dite convergente ssi* $\lim_{h \leftarrow 0} \left[\max_n ||y(x_n) - y_n|| \right] = 0$

Définition 4 *On dit que la méthode est d'ordre p lorsque l'on a :* $\exists k \ , \quad ||y(x_n) - y_n|| \leqslant k.h^p$

Obtention des y_n par récurrence :

On choisit y_0 "quelconque". Ensuite, on applique la formule : $y_{n+1} = y_n + h.\phi(x_n, y_n, h)$

C'est une *formule à un pas*.

Définition 5 *On définit la suite* $(z_n)_{n \in \mathbb{N}}$ *de la manière suivante :*

$$\begin{cases} z_0 = y_0 & \textbf{\textit{(condition initiale)}} \\ \forall n \in \mathbb{N} \quad , & z_{n+1} = z_n + h.\phi(x_n, z_n, h) + h.\varepsilon_n \end{cases}$$

On dit que la méthode est stable ssi $\exists k \ , \quad \max_n ||y_n - z_n|| \leqslant k.\varepsilon$

Définition 6 ***Consistance***

On pose : $J(h) = \max_n \left|\left| \dfrac{y(x_{n+1}) - y(x_n)}{h} - \phi()x_n, y(x_n), h) \right|\right|$

La méthode est dite consistante ssi $\lim_{h \to 0} J(h) = 0$

Elles est dite consistante d'ordre p ssi $\exists K \ , \quad J(h) \leqslant K.h^p$

Théorème 9 *On a le résultat suivant :*

stabilité $+$ *consistance d'ordre p* $\Rightarrow$ *convergence d'ordre p*

La preuve vient du fait que :

$$y(x_{n+1}) = y(x_n) + h.\phi(x_n, y(x_n), h) + h.\varepsilon_n$$

Donc : $\max_n ||y_n - y(x_n)|| \leqslant K. \max_n ||\varepsilon_n|| \leqslant K.h^p$

6.2 Formule de **Taylor**

$$y(x_{n+1}) = y(x_n) + h.y' + \frac{h^2}{2!}.y'' + \frac{h^3}{3!}.y''' + \dots$$

On en déduit :

$$\frac{y(x_{n+1}) - y(x_n)}{h} = f(x_n, y(x_n)) + \frac{h}{2}.\frac{\mathrm{d}f}{\mathrm{d}x} + \frac{h^2}{6}.\frac{\mathrm{d}^2 f}{\mathrm{d}x^2} + \dots$$

De même, on trouve :

$$\phi(x_n, y(x_n), h) = \phi(x_n, y_n, 0) + h.\frac{\partial \phi}{\partial h}(x_n, y(x_n), 0) + \frac{h^2}{\phi}.\frac{\partial^2 \phi}{\partial h^2}(x_n, y(x_n), 0) + \dots$$

Dans le cas où $f(x, y) = \phi(x, y, 0)$, la méthode est **d'ordre 1**

Elle est d'ordre p lorsque : $\forall i \in [1, p]$, $\dfrac{\partial^i \phi}{\partial h^i}(x, y, 0) = \dfrac{1}{i+1}.\dfrac{\mathrm{d}^i f}{\mathrm{d}x^i(x, y)}$

Cette méthode n'est cependant pas très commode.

6.3 Méthode **d'Euler**

$$\frac{y_{n+1} - y_n}{n} = \phi(x_n, y_n, h) \qquad \text{et} \qquad \frac{y_{n+1} - y_n}{h} = f(x_n, y_n)$$

On peut également utiliser *la méthode d'Euler modifiée*. Dans ce cas, on utilise un prédicteur : $\begin{cases} y_{n+\frac{1}{2}} &= y_n + \frac{h}{2}.f(x_n, y_n) \quad \rightarrow \quad \text{prédicteur} \\ y_{n+1} &= y_n + h.f(x_n + \frac{h}{2}, y_{n+\frac{1}{2}}) \end{cases}$

On a donc : $y_{n+1} = y_n + h.f\left(x_n + \dfrac{h}{2}, y_n + \dfrac{h}{2}.f(x_n, y_n)\right)$

CECI EST UNE MÉTHODE D'ORDRE 2

Variante pour le prédicteur :

Au lieu de prendre le prédicteur précédent, on peut prendre le prédicteur y^*_{n+1} défini par : $y^*_{n+1} = y_n + h.f(x_n, y_n)$

On calcule alors le terme suivant de la suite $(y_n)_n$ de la manière suivante : $y_{n+1} = y_n + \frac{h}{2}.[f(x_n, y_n) + f(x_n + h, y^*_{n+1})]$

La méthode **est toujours d'ordre 2**.

6.4 Méthode de Runge - Kutta

On utilise la définition suivante pour ϕ : $\phi(x, y, h) = \frac{1}{6}.(P_1 + 2.P_2 + 2.P_3 + P_4)$

Les fonctions P_1, P_2, P_3 et P_4 étant définies comme suit :

$$P_1(x, y, h) = f(x, y)$$

$$P_2(x, y, h) = f\left(x + \frac{h}{2}, y + \frac{h}{2}.P_1(x, y, h)\right)$$

$$P_3(x, y, h) = f\left(x + \frac{h}{2}, y + \frac{h}{2}.P_2(x, y, h)\right)$$

$$P_4(x, y, h) = f(x + h, y + h.P_3x, y, h))$$

Cette méthode est d'ordre **4**.

Test de h :

Soit ε une précision donnée. On souhaite avoir : $||y_n - y(x_n)|| \leqslant \varepsilon$.

On considère un h "*raisonnable*". On relance l'algorithme en remplaçant h par $\frac{h}{2}$. On compare alors les résultats obtenus :

Si la différence entre les deux mesures est inférieure à ε*, alors on arrête.*

NB : On compare $y_n(h)$ et $y_{2n}(h)$

Chapitre 7

Problème de Dirichlet

On cherche à résoudre le problème :

$$\begin{cases} f(x) = -u''(x) + a(x).u(x) \\ (x \in [0,1] oux \in]0,1[) eta(x) \geqslant 0 \\ u(0) = u(1) = 0 \quad \rightarrow \quad \text{conditions aux bords} \end{cases}$$

On va écrire les équations aux points x_i (i variant de 1 à n), après avoir discrétisé la formule de la dérivée : $-\dfrac{u_{i+1} + uu - 1 - 2u_i}{h^2}$

On pose : $a(x_i) = a_i$ et $f(x_i) = f_i$

Une fois ceci posé, on obtient :

$$\forall i \in [1, N] \ , \quad f(x_i) = f_i = -\frac{u_{i+1} + u_{i-1} - 2.u_i}{h^2} + a_i.u_i$$

La matrice M du système que l'on doit résoudre est donc **tridiagonale symétrique**. La sur- et la sous- diagonale ne contiennent que la valeur -1. Pour les termes diagonaux, on a : $\forall i \in [1, N] \ , \quad m_{i,i} = 2 + h^2.a_i$

Méthode des différences finies :

On utilise un *maillage carré* pour le calcul. On calcule alors aisément les dérivées partielles de f : $\dfrac{\partial f}{\partial x}$ et $\dfrac{\partial f}{\partial y}$

Méthodes des éléments finis :

Cette fois, on utilise un *maillage triangulaire*

Formulation variationnelle de l'équation

$$\int_0^1 -u''.v + \int_0^1 a.u.v = \int_0^1 f.v$$

Ce qui donne : $\forall v \in \mathbb{R}^n$, $\displaystyle\int_0^1 f.v = -[u'.v]_0^1 + \int_0^1 u'.v' + \int_0^1 a.u.v$

Le premier terme ($[u'.v]_0^1$) est nul. Pour le dernier, on écrit : $u(x) = \displaystyle\sum_{i=0}^{N} u_i.v_i(x)$

On va donc devoir calculer les intégrales : $\displaystyle\int_0^1 a(x).v_j(x)\mathrm{d}x$ et $\displaystyle\int_0^1 v_i'(x).v_j'(x)\mathrm{d}x$

Ce qui fait donc, pour chaque choix de v_i, 1 équation.

Au final, on a N équations.

Troisième partie

Transformée de **Fourier** discrète

Généralités

Soit $x = (x_0, \ldots, x_{N-1})$ un vecteur de $\mathbb{C}^N$. Par définition, la *transformée de Fourier discrète* de ce vecteur x est : $F_x(f) = \displaystyle\sum_{k=0}^{N-1} x_k.\mathbf{e}^{-2\mathbf{i}kf\pi}$

En notant $f_k = \frac{k}{N}$ pour $k \in [0, N-1]$, on obtient : $F_x(f_k) = \displaystyle\sum_{n=0}^{N-1} x_k.\mathbf{e}^{-\frac{2\mathbf{i}kn\pi}{N}}$

Propriété 7 *F est un endomorphisme de $\mathbb{C}^N$, de matrice $M_F = (m_{k,n})_{(k,n)\in[0,N-1]^2}$ telle que : $\forall (k,n) \in [0, N-1]^2 \ , \quad m_{k,n} = \mathbf{e}^{-\frac{2\mathbf{i}kn\pi}{N}}$*

On pose $\omega = \mathbf{e}^{-\frac{2\mathbf{i}\pi}{N}}$. On a alors : $\forall (k,n) \in [0, N-1]^2 \ , \quad m_{k,n} = \omega^{k.n}$

Propriété 8 *La matrice M_F de la transformée de Fourier est* ***symétrique****.*

De plus, on a : $M_F.\overline{M_F} = N.Id$

Soit : $F^{-1} = \dfrac{1}{N}.\overline{F}$

Preuve :

On pose $C = M_F.\overline{M_F}$. On a alors : $\forall (l,j) \in [0, N-1]^2 \ , \quad c_{l,j} = \displaystyle\sum_{k=0}^{N-1} \omega^{k.l}.\omega^{-k.j}$

$$\forall (l,j) \in [0, N-1]^2 \ , \quad c_{l,j} = \sum_{k=0}^{N-1} \omega^{k.(l-j)}$$

C'est donc une somme de N racines $N^{\text{ème}}$ de 1. On a donc le résultat suivant : $\forall (l,j) \in [0, N-1]^2 \ , \quad c_{l,j} = \begin{cases} 0 \ , & \text{si } \ l \neq j \\ N \ , & \text{sinon} \end{cases}$

Algorithme naïf

On peut penser à l'algorithme suivant pour calculer la transformée de Fourier d'un vecteur donné.

```
pour k = 0 à N − 1 faire
      δ = 0
          pour n = 0 à N − 1 faire
              δ = δ + x_n.W[n, k]
```

où $W[n,k]$ est une tableau contient les $\omega^{n.k}$

La complexité de cet algorithme est : N^2 HORNERS.

Algorithme rapide

Principe

On suppose que N est une puissance de 2, *i.e.* : $\exists q \in \mathbb{N}\ ,\quad N = 2^q = 2.N'$

On définit alors les deux suites : $\begin{cases} X_0(k) & = & x(2.k) \\ X_1(k) & = & x(2.k+1) \end{cases}$

Formule de synthèse

Propriété 9 *Pour tout vecteur* $x \in \mathbb{C}^N$*, on a :* $F_{x,N}(k) = F_{X_0,N'}(k) + \mathbf{e}^{-\frac{2\mathrm{i}k\pi}{N}}.F_{X_1,N'}(k)$

Preuve :

On a : $F_{X_0,N'}(k) = \displaystyle\sum_{l=0}^{N'-1} x_{2l}.\mathbf{e}^{-\frac{2\mathrm{i}k.(2l).\pi}{N}}$

$$F_{X_1,N'}(k) = \sum_{l=0}^{N'-1} x_{2l+1}.\mathbf{e}^{-\frac{2\mathrm{i}k.(2l).\pi}{N}}$$

Ainsi, on en déduit : $\mathbf{e}^{-\frac{2\mathrm{i}k\pi}{N}}.F_{X_1,N'}(k) = \displaystyle\sum_{l=0}^{N'-1} x_{2l+1}.\mathbf{e}^{-\frac{2\mathrm{i}k.\pi}{N}.(2l+1)}$

Ce qui achève la démonstration.

Principe : Dans la pratique on utilise cette formule pour n'avoir à calculer qu'un seul terme à la fois. Ceci est toujours possible, à condition que N soit une puissance de 2.

La complexité de cet algorithme est de : $q.N$ HORNERS (on le prouve par récurrence sur q).

Remarques : Optimisation

Pour $k \in [0, N'-1]$, on a toujours : $F_{x,N}(k) = F_{X_0,N'}(k) + \mathbf{e}^{-\frac{2\mathbf{i}k\pi}{N}}.F_{X_1,N'}(k)$

$$F_{x,N}(k+N') = F_{X_0,N'}(k) - \mathbf{e}^{-\frac{2\mathbf{i}k\pi}{N}}.F_{X_1,N'}(k)$$

On peut donc économiser une multiplication sur deux, à condition de poser : $Z = \exp\left(-\dfrac{2\mathbf{i}k\pi}{N}\right).F_{X_1,N'}(k)$

Dans ces conditions, on a alors : $F_{x,N}(k) = F_{X_0,N'}(k) + Z$ et $F_{x,N}(k+N') = F_{X_0,N'}(k) - Z$

On notera également que les fréquences $\dfrac{k}{N}$ sont stratégiques pour la représentation du spectre de $F_{x,N}$.

Dans le cas où N n'est pas une puissance de 2, on prend M, la plus petite puissance de 2 supérieur N, puis on pose $\forall i \in [N+1, M-1]\ ,\quad x_i = 0$

Produit de convolution de deux suites

Soient u et v deux suites de $\mathbb{K}^{\mathbb{N}}$. On définit et on note $w = u*v$ le produit de convolution de u et v par : $\forall n \in \mathbb{N}\ ,\quad w_n = \sum_{k=0}^{n} u_k.v_{n-k}$

On remarquera que pour des suites à support fini (*i.e.* à nombre fini de termes) de mesure N, on a : $w_{2N} = u_N.v_0 + u_0.v_N$

La complexité de l'algorithme du produit de convolution est : $2.N$ HORNERS.

Propriété 10 *Soient u et v deux suites. On note w le produit de convolution de ces deux suites. On a alors : $F_w = F_u.F_v$*

Cette propriété permet de déterminer “facilement” la suite w.