ANALYSE NUMERIQUE 2

Deuxième année M.I.

Prof. HAMRI NASR-EDDINE

Table des matières

ANALYSE NUMERIQUE 2

1. RESOLUTION D'UN SYSTEME LINEAIRE

1.1 METHODES DIRECTES

1.1.1 Rappel

Rang d'une matrice

Définition 1.1.1 Soit $A \in M_{m,n}(\mathbb{R})$ une matrice de m lignes et de n colonnes. Le rang de A est égal au nombre maximum de colonnes de A linéairement indépendant et on le note :

$$rang\ A$$

Le rang de A est donc aussi égal à la dimension de l'image de toute application linéaire représenté par A.

Proposition 1.1.1 - $rang\ A = rang\ A^t$ - si $A \in M_{m,n}(\mathbb{R})$ alors $rang\ A \leq \inf(m,n)$ - si $A \in M_n(\mathbb{R})$ alors A inversible $\Leftrightarrow$ $rang\ A = n$.

Définition 1.1.2 Soit $A \in M_{m,n}(\mathbb{R})$, on appelle matrice extraite de A une matrice obtenue par sélection de lignes et de colonnes. Par exemple si I (respectivement J) est un sous ensemble de $\{1,2,...m\}$ (respectivement $\{1,2,...n\}$) on définit une matrice extraite B de A par :

$$B = (a_{ij})_{i\in I, j\in J}$$

Théoreme 1.1.2 Soit $A \in M_{m,n}(\mathbb{R})$ une matrice de $rang\ r$ alors :
- le rang de toute matrice extraite de A est inférieur ou égal à r,
- toute matrice carrée inversible extraite de A est d'ordre inf érieur ou égal à r,
- il existe une matrice carrée extraite de A d'ordre r qui est inversible.

1.1.2 Systèmes linéaires

Définition 1.1.3 On appelle système de m équations à n inconnues $x_1, x_2, ..., x_n$ la famille

d'équations :

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n & = & b_1 \\ a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n & = & b_2 \\ ..... & ... & ... \\ a_{m1}x_1 + a_{m2}x_2 + ... + a_{mn}x_n & = & b_m \end{cases} \tag{1.1}$$

que l'on peut mettre sous la forme matricielle suivante :

$$Ax = b$$

où $A \in M_{m,n}(\mathbb{R})$ est une matrice donnée par ses éléments (a_{ij}), $1 \leq i \leq m, 1 \leq j \leq n$, $b \in \mathbb{R}^m$ de composantes $(b_1, b_2, ..., b_m)$ et x le vecteur inconnu de composantes $(x_1, x_2, ..., x_n)$.

R Si $b = 0$, on dit que le système (1.1) est homogène. Dans le cas où $m < n$, on dit que le système est sous determiné et dans le cas où $n > m$ on parle d'un système sur determiné.

Théoreme 1.1.3 Une condition suffisante pour que le système (1.1) admette au moins une solution est que $rang\ A = m$.

Corollaire 1.1.4 Soit $A \in M_n(\mathbb{R})$ une matrice carrée, une condition nécessaire et suffisante pour que le système $Ax = b$ admette une solution unique est que A soit inversible. Autrement dit $\det A \neq 0$ (ou $rang A = n$). Le système (1.1) est alors dit système de Cramer.

1.1.3 Résolution d'un système triangulaire supérieur

Définition 1.1.4 Soit $A \in M_n(\mathbb{R})$ une matrice carrée, A est dite triangulaire supérieure si :

$$a_{ij} = 0, \quad \forall i > j$$

Dans ce cas le système $Ax = b$ est dit triangulaire supérieur.

R On ne traitera pas le cas des systèmes triangulaires inférieurs car la technique de résolution est identique.

Le système d'équations $Ax = b$ triangulaire supérieur a la forme suivante :

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n & = & b_1 \\ a_{22}x_2 + ... + a_{2n}x_n & = & b_2 \\ ..... & ... & ... \\ a_{nn}x_n & = & b_n \end{cases}$$

Théoreme 1.1.5 Soit le système triangulaire supérieur $Ax = b$ où $A \in M_n(\mathbb{R})$ est une matrice carrée et $b \in \mathbb{R}^n$, si :

$$a_{kk} \neq 0, \quad \forall k \in [1, n]$$

alors le système admet une solution unique et cette solution x^* est telle que :

$$x_k^* = \frac{b_k - \sum_{j=k+1}^{n} a_{kj} x_j^*}{a_{kk}}, \quad k = \{n, n-1, ..., 1\} \tag{1.2}$$

Nous allons étudier les méthodes de résolution du système de n équations linéaires à n inconnues $Ax = b$, par les méthodes directes. Nous entendons par méthodes directes des méthodes qui mènent à la solution en un nombre fini d'opérations élémentaires. Ces méthodes sont utilisées seulement si le nombre d'équations du système n'est pas trop élevé (généralement $n \leq 100$). La méthode de Cramer en est une, mais elle est numériquement inacceptable. Car sa mise en oeuvre demande le calcul de $n+1$ déterminants et n divisions. Pour calculer chaque déterminant, nous devons effectuer $n!n$ multiplications et $n!-1$ additions soit un total de $(n+1)^2 n!-1$ opérations élémentaires Par exemple, pour $n = 5$ on obtient 4319 opérations élémentaires. Pour $n = 10$ on obtient à peu près 4.10^8 opérations élémentaires Or, dans la pratique, nous aurons à résoudre des systèmes d'ordres $n = 100$, $n = 1000$ voire même plus. Il est donc impossible de résoudre de tels systèmes par la méthode de Cramer. Dans ce chapitre, nous présentons essentiellement la méthode d'éliminations successives de Gauss et son interprétation matricielle, laquelle débouche sur la méthode de Cholesky pour un système à matrice définie positive.

Si la matrice A n'est plus triangulaire, nous sommes amenés à chercher une matrice M inversible telle que la matrice produit MA soit triangulaire. On résoudra alors le système :

$$MAx = Mb$$

par l'algorithme (1.2). Nous nous limitons bien entendu à des systèmes $Ax = b$ avec $detA \neq 0$.

1.2 Méthode de Gauss

Soit $A \in M_n(\mathbb{R})$ une matrice carrée donnée et $b \in \mathbb{R}^n$. On cherche x^* solution du système linéaire :

$$Ax = b$$

La méthode de Gauss consiste à construire un système équivalent plus facile à résoudre (à matrice triangulaire supérieure par exemple). Deux systèmes linéaires définis par deux matrices $A \in M_n(\mathbb{R})$ et $U \in M_n(\mathbb{R})$ sont dits équivalents si leurs solutions sont identiques.

R - Les transformations élémentaires suivantes appliquées à un système linéaire engendre un système linéaire équivalent : - Une équation peut être remplacée par cette même équation à laquelle on ajoute ou on retranche un certain nombre de fois une autre ligne. - La multiplication d'une équation par une constante non nulle. - La permutation de deux lignes ou de deux colonnes.

La représentation d'un système linéaire peut se faire à travers une matrice de dimension $n.(n+1)$ appelé matrice augmentée. La matrice est noté $\tilde{A} = [A|b]$ et a pour forme générale :

$$\tilde{A} = \left(\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \cdots & \cdots & \ddots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & b_n \end{array}\right)$$

La résolution du système linéaire ayant pour matrice augmentée $\tilde{A}$ peut se faire en appliquant des transformations élé mentaires permettant d'obtenir un système équivalent.

L'objectif de l'algorithme de Gauss est la construction d'un système triangulaire supérieur équivalent, en annulant au fur et à mesure les termes en dessous de la diagonale.

Définition 1.2.1 On appelle pivot de la transformation, l'élément a_{kk} de la matrice utilisée pour annuler les termes a_{jk}, $j > k$. La ligne k est alors appelée ligne pivot.

Théoreme 1.2.1 Soit un système linéaire défini par une matrice A d'ordre n et $b \in \mathbb{R}^n$. Si A est non singulière alors il existe une matrice U d'ordre n triangulaire supérieure et $y \in \mathbb{R}^n$ tels que $Ux = y$ soit équivalent à $Ax = b$. La résolution du système $Ax = b$ se fait ensuite par résolution du système triangulaire supérieur.

Démonstration. Construisons la matrice augmentée $\tilde{A}^{(1)} = \left[A^{(1)}|b^{(1)}\right]$

$$\tilde{A}^{(1)} = \left(\begin{array}{cccc|c} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} & b_1^{(1)} \\ a_{21}^{(1)} & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} & b_2^{(1)} \\ \cdots & \cdots & \ddots & \vdots & \vdots \\ a_{n1}^{(1)} & a_{n2}^{(1)} & \cdots & a_{nn}^{(1)} & b_n^{(1)} \end{array}\right)$$

l'exposant indiquant le nombre de fois qu'une valeur a été stockée à la location i, j donnée. La première étape de l'algorithme de Gauss est d'annuler l'ensemble des coefficients de la première colonne en dessous de la diagonale. Cela s'obtient si $a_{11} \neq 0$ en réalisant la transformation suivante sur la ligne $i > 1$:

$$a_{ij}^{(2)} = a_{ij}^{(1)} - g_{i1} a_{1j}^{(1)}, \quad j \in [1, n+1]$$

où $g_{i1} = \frac{a_{i1}^{(1)}}{a_{11}^{(1)}}$, on obtient le système équivalent à l'étape 2 suivant donné par sa matrice augment ée :

$$\tilde{A}^{(2)} = \left(\begin{array}{cccc|c} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} & b_1^{(1)} \\ 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} & b_2^{(2)} \\ \cdots & \cdots & \ddots & \vdots & \vdots \\ 0 & a_{n2}^{(2)} & \cdots & a_{nn}^{(2)} & b_n^{(2)} \end{array}\right)$$

les étapes suivantes consistent à refaire le même procédé pour les colonnes suivantes. Ainsi l'étape k consiste à éliminer l'inconnu x_k dans les équations $k+1, ..., n$. Ce qui donne les formules suivantes définies pour les lignes $i = k+1, ..., n$ en supposant que le $k^{ième}$ pivot $a_{kk}^{(k)} \neq 0$:

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - g_{ik} a_{kj}^{(k)}, \quad j \in [k, n+1] \tag{1.3}$$

avec $g_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}$. A la dernière étape c'est-à-dire à $k = n$, on obtient le système équivalent suivant :

$$\tilde{A}^{(n)} = \left(\begin{array}{ccccc|c} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & \cdots & a_{1n}^{(1)} & b_1^{(1)} \\ 0 & a_{22}^{(2)} & \cdots & \cdots & a_{2n}^{(2)} & b_2^{(2)} \\ \cdots & \cdots & \ddots & \cdots & \vdots & \vdots \\ 0 & 0 & \cdots & a_{n-1n-1}^{(n-1)} & a_{n-1n}^{(n-1)} & \cdots & b_{n-1}^{(n-1)} \\ 0 & \cdots & \cdots & 0 & a_{nn}^{(n)} & b_n^{(n)} \end{array}\right)$$

la matrice U est donc définie comme étant la matrice $\tilde{A}^{(n)}$ et y le vecteur $b^{(n)}$. ■

R - La ligne i de la matrice $\tilde{A}^{(k)}$ n'est plus modifiée par l'algorithme dès lors que $i \leq k$. - A l'étape k, on pratique l'élimination sur une matrice de taille $n-k+1$ lignes et $n-k+2$ colonnes.

R Si lors de l'élimination l'élément $a_{kk}^{(k)}$ à l'é tape k est nul alors la ligne k ne peut pas être utilisée comme ligne pivot. Dans ce cas, on cherche une ligne $j > k$ telle $a_{jk}^{(k)} \neq 0$. Si une telle ligne existe, alors on permute la ligne j et la ligne k sinon le système n'admet pas de solution.

R Pour minimiser les erreurs d'arrondi, on choisit la valeur du pivot la plus grande en valeur absolue. Pour ce faire deux stratégie sont possibles :

1. La méthode dite à *pivot partiel* : Au $k^{ième}$ pas de l'élimination, on choisit comme ligne de pivot celle qui, parmi les $n-k+1$ restantes, a l'élément de module maximum en colonne et on permute dans $\tilde{A}^{(k)}$ la $k^{ième}$ ligne naturelle et celle qui réalise ce maximum.
2. La méthode dite à *pivot total* : Au $k^{ième}$ pas de l'élimination, on choisit comme pivot l'élément de plus grand module dans la matrice d'ordre $n-k+1$ restante. On permute donc dans $\tilde{A}^{(k)}$ la $k^{ième}$ colonne naturelle et celle du pivot, ce qui modifiera l'ordre des composantes du résultat. A la fin du processus, il ne faudra pas oublier de remettre dans l'ordre initial les composantes de la solution x.

1.2.1 Interprétation matricielle de la méthode de Gauss

Supposons que l'on puisse effectuer l'élimination sans permutation des lignes et des colonnes. Considérons alors les matrices

$$G^{(k)} = \begin{pmatrix} 1 & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & \ddots & & & & \vdots \\ \vdots & \vdots & 1 & & & \vdots \\ 0 & \cdots & -g_{k+1k} & 1 & & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & -g_{nk} & 0 & \cdots & 1 \end{pmatrix}, \quad k = 1,2,...,n-1$$

avec

$$g_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}, \quad \text{pour} \quad i = k+1,...,n.$$

le système (1.3) peut se mettre sous la forme

$$\tilde{A}^{(k+1)} = G^{(k)}\tilde{A}^{(k)}$$

ce qui donne

$$\tilde{A}^{(n)} = G^{(n-1)}.G^{(n-2)}.G^{(n-3)}.....G^{(1)}.\tilde{A}^{(1)}$$

avec

$$\tilde{A}^{(n)} = \left[A^{(n)}|b^{(n)}\right]$$

Posons

$$\begin{aligned} U &= A^{(n)} \\ L &= \left(G^{(n-1)}.G^{(n-2)}.G^{(n-3)}.....G^{(1)}\right)^{-1} \end{aligned}$$

U (pour Upper) est une matrice triangulaire supérieure et L (pour Lower) est une matrice triangulaire inférieure à diagonale unité . Donc nous avons écrit A sous la forme : $A = LU$ où

$$L = \begin{pmatrix} 1 & 0 & \cdots & \cdots & \cdots & 0 \\ g_{21} & 1 & \ddots & \cdots & \cdots & \vdots \\ g_{31} & g_{32} & 1 & \ddots & & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & 0 \\ g_{n1} & \cdots & \cdots & \cdots & \cdots & 1 \end{pmatrix}, \text{ et } U = \begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & \cdots & \cdots & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & \cdots & \cdots & a_{2n}^{(2)} \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \vdots & a_{n-1n-1}^{(n-1)} & a_{n-1n}^{(n-1)} \\ 0 & \cdots & \cdots & \cdots & 0 & a_{nn}^{(n)} \end{pmatrix}$$

nous sommes donc amenés à résoudre successivement les deux systèmes triangulaires :

$$\begin{cases} Ly = b \\ Ux = y \end{cases} \quad \text{où } y = b^{(n)}$$

1.3 Méthodes LU

La première phase de la méthode de Gauss consistait à transformer le système $Ax = b$ en un système triangulaire $Ux = y$ avec U une matrice triangulaire supérieure. Supposons qu'aucune permutation n'ait été effectuée, on peut alors montrer que U et y ont été obtenus à partir de A et b en les multipliant par une même matrice R triangulaire et inversible, c'est-à-dire

$$U = RA \quad \text{et} \quad y = Rb$$

on a donc $A = R^{-1}U$. Et si on pose $L = R^{-1}$et $U = R$, on peut donc dé composer A en un produit de matrice triangulaire inférieure L et une matrice triangulaire supérieure U. La méthode de Gauss appartient donc à la classe des méthodes dites méthodes LU. Elles consistent à obtenir une décomposition de la matrice A du type LU et à résoudre le système triangulaire $Ly = b$ puis ensuite le système triangulaire $Ux = y$ (L et U étant supposées inversibles).

$$Ax = b \Longleftrightarrow LUx = b \Leftrightarrow \begin{cases} Ly = b \\ Ux = y \end{cases}$$

1.3.1 Décomposition LU

Définition 1.3.1 Une matrice A non singulière, admet une factorisation triangulaire si il existe une matrice L triangulaire inférieure et une matrice U triangulaire supérieure telles que :

$$A = LU$$

Théoreme 1.3.1 Soit le système linéaire $Ax = b$, si au cours de l'élimination de Gauss de la matrice A, aucun pivot n'est nul alors il existe une matrice L triangulaire inférieure et une matrice U triangulaire supérieure telles que :

$$A = LU$$

si de plus on impose $l_{kk} = 1$ alors la factorisation est unique.

La matrice U s'obtient en appliquant la méthode de Gauss tandis que la matrice L s'écrit de la manière suivante :

$$L = \begin{pmatrix} 1 & 0 & \cdots & 0 & 0 \\ l_{21} & 1 & \cdots & 0 & 0 \\ l_{31} & l_{32} & \ddots & 0 & 0 \\ \vdots & \cdots & \ddots & 1 & 0 \\ l_{n1} & l_{n2} & \cdots & l_{nn-1} & 1 \end{pmatrix}$$

où pour $i > 1$ on a $l_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}$. Ainsi la matrice L est composée des facteurs multiplicatifs permettant d'annuler les éléments sous le pivot. Comme il existe des problèmes simples pour lesquelles un des pivots est nul, le théorème suivant permet d'étendre la factorisation LU à un cadre plus général.

Théoreme 1.3.2 Une condition nécessaire et suffisante pour qu'une matrice A inversible puisse se factoriser sous la forme $A = LU$ est que $\det(A_k) \neq 0, \forall k = 1,2,...n-1$. Où $A_k = (a_{ij})_{\substack{i=1,2,..,k \\ j=1,2,.....,k}}$.

Démonstration. 1) Si $A = LU$, $A_k = L_k U_k$ et si A est inversible, $\det U = \prod_{i=1}^{n} u_{ii} = \prod_{i=1}^{n} a_{ii}^{(i)} \neq 0$. Donc $\det(A_k) = \det(L_k).\det(U_k) = \det(U_k) = \prod_{i=1}^{k} a_{ii}^{(i)} \neq 0$. 2) Supposons que $\det(A_k) \neq 0 \ \ \forall k = 1,2,...,n-1$. Celà est vrai en particulier pour $k = 1$, donc $a_{11}^{(1)} = \det(A_1)$ et la première étape de l'élimination de Gauss est possible. Par récurrence, si on a obtenu $A^{(k)}$ pour $k \leq n-1$

$$A^{(k)} = G^{(k-1)}.G^{(k-2)}.G^{(k-3)}.....G^{(1)}.A^{(1)}$$

alors $\det(A_k^{(k)}) = \det(G_k^{(k-1)})...\det(G_k^{(1)})\det(A^{(1)}) = \det(A^{(1)}) \neq 0$ $A_k^{(k)}$ étant triangulaire on a $\prod_{i=1}^{k} a_{ii}^{(i)} \neq 0$ donc $a_{kk}^{(k)} \neq 0$, donc la $k^{ième}$ étape de l'élimination est possible. On obtiendra finalement A=LU. ■

Théoreme 1.3.3 — méthode à pivot partiel. Soit A une matrice carrée d'ordre n inversible, alors il existe une matrice de permutation P telle que les pivots de PA soient non nuls. Ainsi il existe deux matrices L et U telles que $PA = LU$.

R le système linéaire $Ax = b$ est équivalent au système $PAx = Pb$ et la résolution du système se fait selon les étapes suivantes :

1. Construire U, L et P,
1. Calculer Pb,
2. Résoudre $Ly = Pb$ (système triangulaire inférieur),
3. Résoudre $Ux = y$ (système triangulaire supérieur).

1.4 Méthode de Cholesky

Certains systèmes présentent des propriétés particulières. Les matrices associées à ces systèmes peuvent être symètriques, à bande, etc...La méthode de cholesky a pour but la résolution de systèmes linéaires pour lesquels la matrice associée est symétrique définie positive.

Définition 1.4.1 — Matrice symétrique. Soit A une matrice carrée d'ordre n. On dit que A est symétrique si on a

$$A = A^t.$$

Définition 1.4.2 — Matrice définie positive. Soit A une matrice carrée d'ordre n. On dit que A est définie positive si elle vérifie la condition suivante :

$$\forall x \in \mathbb{R}^n \text{ et } x \neq 0, \quad \langle Ax, x\rangle > 0$$

où $\langle .,. \rangle$ désigne le produit scalaire dans $\mathbb{R}^n$. C'est-à-dire :

$$\langle x, y\rangle = \sum_{i=1}^{n} x_i y_i, \quad \forall x, y \in \mathbb{R}^n$$

On définit la norme induite par :

$$\|x\| = \langle x, x\rangle^{\frac{1}{2}}$$

Proposition 1.4.1 Si A est une matrice symétrique définie positive alors :

1. $a_{ii} > 0$,
2. $a_{ij} < a_{ii} a_{jj} \quad \forall i \neq j$,
3. $\max_{j,k} |a_{jk}| < \max_i |a_{ii}|$.

Théoreme 1.4.2 Si la matrice A est une matrice carrée définie positive alors elle est inversible.

Corollaire 1.4.3 Si la matrice A est une matrice carrée définie positive alors le système linéaire $Ax = b$ où $x, b \in \mathbb{R}^n$ admet une solution et une seule.

Théoreme 1.4.4 Soit M une matrice carrée telle et non singulière alors la matrice $A = MM^t$ est symétrique définie positive.

■ **Exemple 1.1** Soit

$$M = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

alors

$$A = MM^t = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 2 \\ 1 & 2 & 3 \end{pmatrix}$$

est définie positive. ■

1.4.1 Factorisation de Cholesky

Théoreme 1.4.5 Soit A une matrice carrée d'ordre n symétrique définie positive, A peut alors se décomposer et de manière unique en

$$A = LL^t$$

où L est une matrice triangulaire inférieure avec des éléments diagonaux positifs.

Ainsi ce théorème permet de déduire que la méthode de construction des matrices définies positives engendre en fait l'ensemble des matrices symétriques définies positives. Si A est une matrice symétrique définie positive alors le systè me $Ax = b$ peut être décomposé en $LL^t x = b$ et ce système peut se résoudre en résolvant les systèmes triangulaires :

$$\begin{cases} Ly &= b \\ L^t x &= y \end{cases}$$

1.4.2 Algorithme de décomposition de Cholesky

Soit A une matrice symétrique définie positive alors on a $A = LL^t$. Pour résoudre le système

$$Ax = b \tag{1.4}$$

le théorème précédent nous permet d'écrire (1.4) sous la forme $LL^t x = b$ avec L une matrice triangulaire inférieure inversible. On est donc amené à résoudre

$$\begin{cases} Ly &= b \\ L^t x &= y \end{cases}$$

Le problème consiste donc à construire explicitement la matrice $L = (l_{ij})$ triangulaire inférieure telle que

$$A = LL^t \quad \text{où} \quad A = (a_{ij})$$

ce qui équivaut à

$$a_{ij} = \sum_{k=1}^{j} l_{ik} l_{jk}, \quad j \leq i.$$

Soit :

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} = \begin{pmatrix} l_{11} & 0 & 0 & 0 \\ l_{21} & l_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{pmatrix} \begin{pmatrix} l_{11} & l_{12} & \cdots & l_{1n} \\ 0 & l_{22} & \cdots & l_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & l_{nn} \end{pmatrix}$$

en remarquant que a_{ij} est le produit de la ligne i de L et la colonne j de L^t alors on a :

$$a_{i1} = \sum_{k=1}^{n} l_{ik} l_{1k} = l_{i1} l_{11} + l_{i2} l_{12} + \cdots + l_{in} l_{1n} = l_{i1} l_{11}$$

en particulier pour $i = 1$, on a $l_{11} = \sqrt{a_{11}}$ (l_{11} est bien positif). la connaissance de l_{11} permet de construire la première colonne de la matrice L car :

$$l_{i1} = \frac{a_{i1}}{l_{11}}$$

En raisonnant de la même manière pour la deuxième colonne de L , on a :

$$a_{i2} = \sum_{k=1}^{n} l_{ik} l_{2k} = l_{i1} l_{21} + l_{i2} l_{22}$$

en prenant $i = 2$ alors $a_{22} = l_{21}^2 + l_{22}^2$. D'où l'on tire

$$l_{22} = \sqrt{a_{22} - l_{21}^2}$$

ensuite on a :

$$l_{i2} = \frac{a_{i2} - l_{i1} l_{21}}{l_{22}} \qquad i = 3,4,...,n$$

On peut généraliser la procédure au calcul de la colonne j en supposant que les $(j-1)$ colonnes ont déjà été calculé e. Ainsi :

$$a_{ij} = \sum_{k=1}^{n} l_{ij} l_{1j} = l_{ij} l_{j1} + l_{ij} l_{j2} + \cdots + l_{ik} l_{jk} + \cdots + l_{in} l_{jj}$$

et seul l_{ij} et l_{jj} ne sont pas connus. Si on pose $i = j$, on obtient :

$$l_{jj} = \sqrt{a_{jj} - \sum_{k=1}^{j-1} l_{jk}^2}$$

et par conséquent

$$l_{ij} = \frac{a_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk}}{l_{jj}} \qquad i > j \qquad \begin{array}{rcl} j & = & 2,.....,n \\ i & = & j+1,...,n \end{array}$$

R La décomposition de A symétrique définie positive, sous la forme $A = LL^t$ est unique à une matrice diagonale unité près. C'est-à-dire si $A = LL^t = MM^t$, alors $M = DL$ avec D matrice diagonale telle que $d_{ii} = \pm 1$.

R La méthode de Cholesky permet de calculer $\det A$ par

$$\det A = \prod_{i=1}^{n} l_{ii}^2$$

1.5 SERIE D'EXERCICES

Exercice 1.1 Résoudre le système d'équations linéaires suivant :

$$\begin{cases} -3x_1 - x_2 & = & 5 \\ -2x_1 + x_2 + x_3 & = & 0 \\ 2x_1 - x_2 + 4x_3 & = & 15 \end{cases}$$

1. En appliquant les formules de Cramer.
2. En triangularisant la matrice du système associée par la méthode de Gauss.

■

Exercice 1.2 Résoudre le système d'équations linéaires suivant :

$$\begin{cases} 2x_1 + x_2 - 5x_3 + x_4 & = & 8 \\ x_1 - 3x_2 - 6x_4 & = & 9 \\ 2x_2 - x_3 + 2x_4 & = & -5 \\ x_1 + 4x_2 - 7x_3 + 6x_4 & = & 0 \end{cases}$$

En appliquant le principe de triangularisation de Gauss. ■

Exercice 1.3 Soit le système d'équations linéaires suivant :

$$\begin{cases} x_1 + 2x_3 & = & 2 \\ 5x_2 + 4x_3 & = & 0 \\ 2x_1 + 4x_2 + 14x_3 & = & 5 \end{cases}$$

1. Montrer que la matrice associée à ce système est définie positive.
2. Résoudre ce système en utilisant la méthode de Choleski.

■

Exercice 1.4 Soit les systèmes d'équations linéaires suivants :

$$\begin{cases} x_1 + x_2 + 2x_3 & = & 1 \\ 5x_1 + 5x_2 & = & 3 \\ 3x_1 + x_2 + x_3 & = & -2 \end{cases} \quad \text{et} \quad \begin{cases} x_1 + 4x_2 + x_3 + 3x_4 & = & 2 \\ -x_2 + 3x_3 - x_4 & = & 0 \\ 3x_1 + x_2 + 2x_4 & = & 1 \\ x_1 - 2x_2 + 5x_3 + x_4 & = & -2 \end{cases}$$

Effectuer la résolution en mettant, si cela est possible, la matrice associée à chacun de ces deux systèmes, sous forme d'un produit de deux matrices triangulaires de structures différentes.

■

1.6 METHODES INDIRECTES

Introduction

Les méthodes directes de résolution de systèmes linéaires fournissent une solution x au problème $Ax = b$ en un nombre fini d'opérations. Si l'ordre n de la matrice A est élevé, le nombre d'opérations est aussi élevé et de plus, le résultat obtenu n'est pas rigoureusement exact. Par ailleurs, il existe des cas où les structures du système linéaire ne sont pas tirés à profit par les méthodes directes. C'est par exemple le cas des systèmes où la matrice A est très creuse. C'est la raison pour laquelle, dans ce cas , on préfère utiliser des méthodes itératives. L'objectif d'une méthode itérative est de construire une suite de vecteurs $\left\{x^{(k)}\right\}_{k=1,2...,n}$ qui tend vers un vecteur$\bar{x}$, solution exacte du problème $Ax = b$. Souvent, on part d'une approximation $\left\{x^{(0)}\right\}$ de $\bar{x}$ obtenue en général par une méthode directe.

1.6.1 Les méthodes itératives

L'objectif est de résoudre un système du type $Ax = b$. Pour cela, nous allons décomposer la matrice A en

$$A = M - N$$

de sorte que M soit inversible. Ainsi, le système devient :

$$Mx = Nx + b$$

et nous chercherons par récurrence une suite de vecteurs $x^{(i)}$ obtenu à partir d'un vecteur $x^{(0)}$ et de la relation

$$Mx^{(k+1)} = Nx^{(k)} + b$$

c'est-à-dire

$$x^{(k+1)} = M^{-1}Nx^{(k)} + M^{-1}b$$

Cette relation est une relation de récurrence du premier ordre. Nous pouvons en déduire une relation reliant l'erreur $e^{(k)} = x^{(k)} - \bar{x}$ à $e^{(k-1)} = x^{(k-1)} - \bar{x}$:

$$M(x^{(k)} - \bar{x}) = N(x^{(k-1)} - \bar{x})$$

puisque $M\bar{x} = N\bar{x} + b$ et donc $e^{(k)} = M^{-1}Ne^{(k-1)}$ pour $k = 1, 2, ...$ Si on pose $B = M^{-1}N$, nous avons alors

$$e^{(k)} = Be^{(0)}$$

La convergence de la suite $x^{(k)}$ vers la solution $\bar{x}$ est donné par le proposition suivant :

Proposition 1.6.1 Le choix de la décomposition de A devra obéir aux règles suivantes :

R

Proposition 1.6.2
1. Le rayon spectral $\rho(M^{-1}N)$ doit être strictement infé rieur à 1.
2. La résiolution de $Mx^{(k)} = Nx^{(k-1)} + b$ doit être simple et nécessiter le moins d'opérations possibles
3. Pour obtenir la meilleure convergence, $\rho(M^{-1}N)$ doit être le plus petit possible.

On voit que la convergence dépend de la décomposition.

1.6.2 Différentes décomposition de A

On écrit la matrice A sous la forme

$$A = D + E + F$$

avec D la matrice diagonale suivante :

$$D = \begin{pmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & 0 & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & a_{nn} \end{pmatrix}$$

E la matrice triangulaire inférieure suivante

$$E = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ a_{21} & 0 & 0 & \vdots \\ \vdots & & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn-1} & 0 \end{pmatrix}$$

et F la matrice triangulaire supérieure

$$F = \begin{pmatrix} 0 & a_{12} & \cdots & a_{1n} \\ \vdots & 0 & \ddots & \vdots \\ 0 & & \ddots & a_{n-1n} \\ 0 & \cdots & 0 & 0 \end{pmatrix}$$

Nous obtiendrons donc la décomposition $A = M - N$ à partir de diffé rents types de regroupement de ces matrices D, E et F.

1.6.3 Méthode des approximations successives

Soit le système :

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n &= b_2 \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots & \cdots \quad \cdots \\ a_{n1}x_1 + a_{n2}x_2 + ... + a_{nn}x_n &= b_n \end{cases} \tag{1.5}$$

ou

$$Ax = b$$

En supposant

$$a_{ii} \neq 0 (i = 1, 2, ...n)$$

On met le le système (1.7) sous la forme :

$$\begin{cases} x_1 &= \beta_1 + \alpha_{12}x_2 + \alpha_{13}x_3 ..... + \alpha_{1n}x_n \\ x_2 &= \beta_2 + \alpha_{21}x_1 + \alpha_{23}x_3 + ... + \alpha_{2n}x_n \\ \vdots &= \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ x_n &= \beta_n + \alpha_{n1}x_1 + \alpha_{n2}x_2 + ... + \alpha_{nn-1}x_{n-1} \end{cases} \tag{1.6}$$

où $\beta_i = \frac{b_i}{a_{ii}}$, $\alpha_{ij} = -\frac{a_{ij}}{a_{ii}}$ pour $i \neq j$ et $\alpha_{ij} = 0$ pour $i = j$ $(i, j = 1, 2, ...n)$

Soit

$$C = \begin{pmatrix} \alpha_{11} & \alpha_{12} & \cdots & \alpha_{1n} \\ \alpha_{21} & \alpha_{22} & \cdots & \alpha_{2n} \\ \vdots & & \ddots & \vdots \\ \alpha_{n1} & \alpha_{n2} & \cdots & \alpha_{nn} \end{pmatrix}$$

et $D = (\beta_1, \beta_2, ..., \beta_n)^T$. Le système (1.6) est donc sous la forme $x = D + Cx$. Nous allons prendre pour approximation initiale $x^{(0)} = D$ et nous construisons les matrices colonnes : $x^{(1)} = D + Cx^{(0)}$, $x^{(2)} = D + Cx^{(1)}$ etc... et en général $x^{(k+1)} = D + Cx^{(k)}$, $(k = 0, 1, 2, ...)$. Si la suite des approximations est convergente alors :

$$x = \lim_{k \to \infty} x^{(k)}.$$

Cette limite est solution du système (1.6). Sous forme developpée les formules s'écrivent :

$$\begin{cases} x_i^{(0)} & = & \beta_i \\ x_i^{(k+1)} & = & \beta_i + \sum_{i=1}^{\infty} \alpha_{ij} x_j^k \end{cases}$$

$(\alpha_{ii} = 0\ ,\ i = 1, 2, ...n,\ k = 0, 1, 2, ...)$

R Il est parfois commode de ramener le système (1.7) au type (1.6) de façon à ne pas annuler les coefficients α_{ii}.

■ **Exemple 1.2** L'équation :

$$1,02x_1 - 0,15x_2 = 2,7$$

s'écrit :

$$x_1 = 2,7 - 0,02x_1 + 0,15x_2$$

En général, dans le cas du système :

$$\sum_{j=1}^{n} a_{ij} x_j = \beta_i, \qquad \text{(i=1,2,.....n)}$$

On pose :

$$a_{ii} = a_{ii}^{(1)} + a_{ii}^{(2)}, \quad \text{avec} \quad a_{ii}^{(1)} \neq 0 \qquad \text{(i=1,2,.....n)}$$

Le système est équivalent au système :

$$x_i = \beta_i + \sum_{j=1}^{n} \alpha_{ij} x_j \quad (i = 1, 2, ...n)$$

où $\beta_i = \frac{b_i}{a_{ii}^{(1)}}$, $\alpha_{ii} = -\frac{a_{ii}^{(2)}}{a_{ii}^{(1)}}$, $\alpha_{ij} = -\frac{a_{ij}}{a_{ii}^{(1)}}$ pour $i \neq j$ et $(i, j = 1, 2, ...n)$. ■

■ **Exemple 1.3** Résoudre par la méthode des approximations successives le système suivant :

$$\begin{cases} 4x_1 + 0,24x_2 - 0,08x_3 & = & 8 \\ 0,09x_1 + 3x_2 - 0,15x_3 & = & 9 \\ 0,04x_1 - 0,08x_2 + 4x_3 & = & 20 \end{cases}$$

Le système se met sous la forme :

$$\begin{cases} x_1 &= 2-0,06x_2+0,02x_3 \\ x_2 &= 3-0,03x_1+0,05x_3 \\ x_3 &= 5-0,01x_1+0,02x_2 \end{cases}$$

Ou :

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \\ 5 \end{pmatrix} + \begin{pmatrix} 0 & -0,06 & 0,02 \\ -0,03 & 0 & 0,05 \\ -0,01 & 0,02 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

Nous prenons $x^{(0)} = D$ et nous construisons les matrices colonnes : $x^{(1)} = D + Cx^{(0)}$, $x^{(2)} = D + Cx^{(1)}$, c'est-à-dire : $x_1^{(0)} = 2$, $x_2^{(0)} = 3$, $x_3^{(0)} = 5$. Les premières approximations donnent :

$$\begin{cases} x_1^{(1)} &= 2-0,06\times 3+0,02\times 5 = 1,92 \\ x_2^{(1)} &= 3-0,03\times 2+0,05\times 5 = 3,19 \\ x_3^{(1)} &= 5-0,01\times 2+0,02\times 3 = 5,04 \end{cases}$$

$$\begin{cases} x_1^{(2)} &= 2-0,06\times 3,19+0,02\times 5,04 = 1,9094 \\ x_2^{(2)} &= 3-0,03\times 1,92+0,05\times 5,04 = 3,1944 \\ x_3^{(2)} &= 5-0,01\times 1,92+0,02\times 3,19 = 5,0446 \end{cases}$$

$$\begin{cases} x_1^{3)} &= 2-0,06\times 3,1944+0,02\times 5,0446 = 1,9092 \\ x_2^{(3)} &= 3-0,03\times 1,9094+0,05\times 5,0446 = 3,1949 \\ x_3^{(3)} &= 5-0,01\times 1,9094+0,02\times 3,1944 = 5,0447 \end{cases}$$

■

1.6.4 Méthode de Seidel

C'est une modification de la méthode des approximations successives. Soit :

$$x_i = \beta_i + \sum_{j=1}^{n} \alpha_{ij}x_j \quad (i = 1,2,.....n)$$

Choisissant arbitrairement $x_1^{(0)}, x_2^{(0)}, x_3^{(0)}, ...,x_n^{(0)}$. En supposant que les k^{ieme} approximations $x_i^{(k)}$, $(i = 1,2,.....n)$, sont connues, nous construisons d'après Seidel les $(k+1)^{ieme}$ approximations de la solution d'après les formules :

$$\begin{cases} x_1^{k+1} &= \beta_1 + \sum_{j=1}^{n} \alpha_{1j}x_j^k \\ x_2^{k+1} &= \beta_2 + \alpha_{21}x_1^{k+1} + \sum_{j=2}^{n} \alpha_{2j}x_j^k \\ \cdots &= \cdots\cdots\cdots\cdots\cdots\cdots \\ x_i^{k+1} &= \beta_i + \sum_{j=1}^{i-1} \alpha_{ij}x_j^{k+1} + \sum_{j=i}^{n} \alpha_{ij}x_j^k \\ \cdots &= \cdots\cdots\cdots\cdots\cdots\cdots \\ x_n^{k+1} &= \beta_n + \sum_{j=1}^{n-1} \alpha_{nj}x_j^{k+1} + \alpha_{nn}x_n^k \end{cases}$$

$$(k = 0,1,2,.....)$$

■ **Exemple 1.4** Résoudre par la méthode de Seidel, le système suivant :

$$\begin{cases} 10x_1+x_2+x_3 &= 12\\ 2x_1+10x_2+x_3 &= 13\\ 2x_1+2x_2+10x_3 &= 14 \end{cases}$$

Le système se met sous la forme :

$$\begin{cases} x_1 &= 1,2-0,1x_2-0,1x_3\\ x_2 &= 1,3-0,2x_1-0,1x_3\\ x_3 &= 1,4-0,2x_1-0,2x_2 \end{cases}$$

Ou :

$$\begin{pmatrix} x_1\\ x_2\\ x_3 \end{pmatrix} = \begin{pmatrix} 1,2\\ 1,3\\ 1,4 \end{pmatrix} + \begin{pmatrix} 0 & -0,1 & -0,1\\ -0,2 & 0 & -0,1\\ -0,2 & -0,2 & 0 \end{pmatrix} \begin{pmatrix} x_1\\ x_2\\ x_3 \end{pmatrix}$$

Nous prenons $x_1^{(0)}=1,2$, $x_2^{(0)}=0$, $x_3^{(0)}=0$, et nous construisons les matrices colonnes : $x^{(1)}=D+Cx^{(0)}$, $x^{(2)}=D+Cx^{(1)}$....., c'est-à-dire : $x_1^{(0)}=1,2$, $x_2^{(0)}=0$, $x_3^{(0)}=0$. Les premières approximations donnent :

$$\begin{cases} x_1^{(1)} &= 1,2-0,1\times 0-0,1\times 0=1,2\\ x_2^{(1)} &= 1,3-0,2\times 1,2-0,1\times 0=1,06\\ x_3^{(1)} &= 1,4-0,2\times 1,2-0,2\times 1,06=0,94 \end{cases}$$

$$\begin{cases} x_1^{(2)} &= 1,2-0,1\times 1,06-0,1\times 0,94=0,999\\ x_2^{(2)} &= 1,3-0,2\times 0,999-0,1\times 0,94=1,005\\ x_3^{(2)} &= 1,4-0,2\times 0,999-0,2\times 1,005=0,999 \end{cases}$$

$$\begin{cases} x_1^{(3)} &= 1,2-0,1\times 1,005-0,1\times 0,999=0,999\\ x_2^{(3)} &= 1,3-0,2\times 0,999-0,1\times 1,005=0,999\\ x_3^{(3)} &= 1,4-0,2\times 0,999-0,2\times 1,000=1,000 \end{cases}$$

$$\begin{cases} x_1^{(4)} &= 1,2-0,1\times 1,000-0,1\times 1,000=1,000\\ x_2^{(4)} &= 1,3-0,2\times 1,000-0,1\times 1,000=1,000\\ x_3^{(4)} &= 1,4-0,2\times 1,000-0,2\times 1,000=1,000 \end{cases}$$

La solution est donc :

$$x=(1,000;1,000;1,000)^T$$

■

1.6.5 Méthode de Jacobi

On pose

$$M=D \quad \text{et} \quad N=-(E+F)$$

ainsi, $B=M^{-1}N=D^{-1}(-E-F)$, ce qui implique :

$$x^{(k+1)}=D^{-1}(-E-F)x^{(k)}+D^{-1}b$$

si on exprime cette relation en fonction des éléments de la matrice A nous avons :

$$x_i^{(k+1)}=-\sum_{\substack{j=1\\ j\neq i}}^{n}\frac{a_{ij}}{a_{ii}}x_j^{(k)}+\frac{b_i}{a_{ii}}, \qquad i=1,2,...,n$$

1.6.6 Méthode de Gauss-Seidel

Cette méthode utilise

$$M = D + E \quad \text{et} \quad N = -F$$

D'où

$$B = -(D+E)^{-1} F,$$

et alors on a :

$$x^{(k+1)} = -(D+E)^{-1} F x^{(k)} + (D+E)^{-1} b$$

le calcul de l'inverse de $(D+E)$ peut être évité. Si on écrit $(D+E)x^{(k+1)} = -Fx^k + b$, on obtient

$$\sum_{j=1}^{n} a_{ij} x_j^{(k+1)} = -\sum_{j=i+1}^{n} a_{ij} x_j^{(k)} b_i,$$

d'où

$$x_i^{(k+1)} = -\frac{1}{a_{ii}} \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \frac{1}{a_{ii}} \sum_{j=i+1}^{i-1} a_{ij} x_j^{(k)} + \frac{b_i}{a_{ii}}, \qquad i = 1,2,...,n.$$

1.6.7 Méthode de relaxation

On donne un paramètre $\omega \in]0,2[$, appelé facteur de relaxation, et on pose

$$M = \frac{D}{\omega} + E \quad \text{et} \quad N = \left(\frac{1-\omega}{\omega}\right) D - F$$

et par conséquent

$$\left(\frac{D}{\omega} + E\right) x^{(k+1)} = \left(\left(\frac{1-\omega}{\omega}\right) D - F\right) x^{(k)} + b$$

d'où

$$x_i^{(k+1)} = (1-\omega) x_i^{(k)} + \frac{\omega}{a_{ii}} \left(-\sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} + b_i \right) \qquad i = 1,2,...,n.$$

Comme on peut le constater, la méthode de Gauss-Seidel correspond à la méthode de relaxation pour $\omega = 1$.

Autrement dit la méthode de relaxation peut se résumer en ce qui suit : Nous considérons le système :

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n &= b_2 \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots & \cdots \quad \cdots \\ a_{n1}x_1 + a_{n2}x_2 + ... + a_{nn}x_n &= b_n \end{cases} \tag{1.7}$$

ou

$$Ax = b$$

Nous transformons ce système, en transposant les termes constants à gauche et en divisant la première équation par $-a_{11}$, la deuxième équation par $-a_{22}$, etc.... Nous mettons le système (1.7) sous la forme :

$$\begin{cases} -x_1 + b_{12}x_2 + 13x_3 ..... + b_{1n}x_n + c_1 &= 0 \\ b_{21}x_1 - x_2 + b_{23}x_3 + ... + b_{2n}x_n &= 0 \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots &= 0 \\ b_{n1}x_1 + b_{n2}x_2 + ... + b_{nn-1}x_{n-1} - x_n &= 0 \end{cases} \tag{1.8}$$

où $b_{ij} = -\frac{a_i j}{a_{ii}}$, pour $i \neq j$ et $c_i = -\frac{b_i}{a_{ii}}$ $(i, j = 1, 2, ...n)$

Soit $x = (x_1^{(0)}, x_2^{(0)}, x_3^{(0)}, ..., x_n^{(0)})$, l'approximation initiale de la solution, en portant dans le système (1.8), nous obtenons les restes :

$$\begin{cases} R_1^{(0)} &= c_1 - x_1^{(0)} + \sum_{j=2}^{n} b_{1j}x_j^{(0)} \\ R_2^{(0)} &= c_2 - x_2^{(0)} + \sum_{j=1, j\neq 2}^{n} b_{2j}x_j^{(0)} \\ \cdots &= \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ R_i^{(0)} &= c_i - x_i^{(0)} + \sum_{j=1, j\neq i}^{i} b_{ij}x_j^{(0)} \\ \cdots &= \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ R_n^{(0)} &= c_n - x_n^{(0)} + \sum_{j=1}^{n-1} b_{nj}x_j^{(0)} \end{cases}$$

Si nous varions l'une des inconnues $(x_k^{(0)}$ de $(\delta_{x_k}^{(0)})$, le reste correspondant $R_k^{(0)}$ diminue de $(\delta_{x_k}^{(0)})$ alors que tous les autres restes $R_i^{(0)}$, $(i \neq k)$ augmentent de $b_{ik}\delta_{x_k}^{(0)})$.

Pour annuler le reste successif $R_k^{(1)}$, il suffit de donner à $(x_k{}^{(0)})$ l'accroissement $(\delta_{x_k}^{(0)}) = R_k^{(0)}$, pour avoir $R_k^{(1)} = 0$ et $R_i^{(1)} = R_i^{(0)} + b_{ik}\delta_{x_k}^{(0)}$, $i \neq k$.

La méthode de relaxation consiste à annuler à chaque étape le reste maximal en module, en modifiant la valeur de la composante d'approximation correspodante. Le processus s'arrête lorsque tous les restes du dernier système s'annulent.

■ **Exemple 1.5** Résoudre par la méthode de relaxation, le système suivant :

$$\begin{cases} 10x_1 - 2x_2 - 2x_3 &= 6 \\ -x_1 + 10x_2 - 2x_3 &= 7 \\ -x_1 - x_2 + 10x_3 &= 8 \end{cases}$$

Soit $x_1^{(0)} = x_2^{(0)} = x_3^{(0)} = 0$, on a donc : $R_1^{(0)} = 0,6$, $R_2^{(0)} = 0,7$ et $R_3^{(0)} = 0,8$. On pose : $\delta_{x_3}^{(0)} = 0,8$. Ce qui nous donne :

$$\begin{cases} R_1^{(1)} &= R_1^{(0)} + 0,2 \times 0,8 = 0,6 + 0,16 = 0,76 \\ R_2^{(1)} &= R_2^{(0)} + 0,2 \times 0,8 = 0,7 + 0,16 = 0,86 \\ R_3^{(1)} &= R_3^{(0)} - 0,8 = 0 \end{cases}$$

Ensuite, on pose : $\delta_{x_2}^{(1)} = 0,86$. Ce que nous pouvons mettre sous la forme du tableau suivant :

x_1	R_1	x_2	R_2	x_3	R_3
0	0,60	0	0,70	0	0,80
	0,16		0,16	0,80	−0,80
	0,76		0,86		0
	0,17	0,86	-0,86		0,09
	0,93		0		0,09
0,93	−0,93		0,09		0,09
	0		0,09		0,18
	0,04		0,04	0,18	−0,18
	0,04		0,13		0
	0,03	0,13	−0,13		0,01
	0,07		0		0,01
0,07	−0,07		0,01		0,01
	0		0,01		0,02
	0		0	0,02	−0,02
	0		0,01		0
	0	0,01	−0,01		0
	0		0		0

En additionnant les $\delta_{x_i}^{(k)}$. Pour $(i = 1,2,3)$ et $(k = 0,1,2,3,...)$. Nous obtenons la solution sous la forme :

$$\begin{cases} x_1 &= 0+0,93+0,07 = 1,00 \\ x_2 &= 0+0,86+0,13+0,01 = 1,00 \\ x_3 &= 0+0,80+0,18+0,02 = 1,00 \end{cases}$$

■

1.7 CONVERGENCE DES MÉTHODES ITÉRATIVES

La convergence des méthodes itératives dépend fortement du rayon spectral de A, Nous étudions d'abord les propriétés de certaines matrices et la localisation de leurs valeurs propres.

Définition 1.7.1 Soit $A \in \mathscr{M}_{m,n}(\mathbb{R})$ une matrice. On définit la norme matricielle induite à partir de la norme vectorielle sur $\mathbb{R}^n$ par

$$\|A\| = \max_{x \neq 0} \frac{\|Ax\|}{\|x\|}$$

Proposition 1.7.1 Soit A et B deux matrices telles que leur multiplication soit compatible alors on a :

$$\|AB\| \leq \|A\| \, \|B\|$$

pour toute norme induite.

Théoreme 1.7.2 — Gerschgorin-Hadamard. Les valeurs propres de la matrice A appartiennent à la reunion des n disques D_k pour $k = 1,2,...,n$ du plan complexe ($\lambda \in \cup_{k=1}^{n} D_k$ où D_k, appelé

disque de Gerschgorin, est défini par :

$$|z-a_{kk}| \leq \sum_{\substack{j=1 \\ j\neq i}} |a_{kj}|$$

1.7.1 Cas général

On considère une méthode itérative définie comme :

$$\begin{cases} x^{(0)} & \text{donné} \\ x^{(k+1)} & = & Cx^{(k)}+D \end{cases}$$

Théoreme 1.7.3 Soit A une matrice carré d'ordre n, pour que $\lim_{k\to\infty} A^k = 0$, il faut et il suffit que $\rho(A) < 1$.

Théoreme 1.7.4 Si il existe une norme induite telle que $\|C\| < 1$ alors la méthode itérative décrite ci-dessus est convergente quelque soit $x^{(0)}$ et elle converge vers la solution de :

$$(I_d - C)x = D$$

Théoreme 1.7.5 Une condition nécessaire et suffisante de convergence de la méthode ci-dessus est que :

$$\rho(C) < 1$$

R la condition de convergence donnée par le rayon spectral n'est pas dépendante de la norme induite, cependant elle peut être utile car le calcul du rayon spectral peut être difficile.

Cas des matrices à diagonale dominante

Définition 1.7.2 Une matrice est dite à diagonale dominante si :

$$\forall i, 1 \leq i \leq n, \quad |a_{ii}| > \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}|$$

Théoreme 1.7.6 Si A est une matrice à diagonale strictement dominante, alors A est inversible et en outre, les méthodes de Jacobi et de Gauss-Seidel convergent.

Démonstration. si A est une matrice à diagonale strictement dominante, on montre que A est inversible en démontrant que 0 n'est pas une valeur propre (c'est-à-dire $KerA = 0$). Posons $B = M^{-1}N$ est soit λ et v tels que $Bv = \lambda v$ avec $v \neq 0$. Puisque l'on s'intéresse à $\rho(B) < 1$, on s'intéresse en fait à la plus grande valeur propre de plus grand module de B. Ainsi, on peut supposer que $\lambda \neq 0$. L'équation $Bv = \lambda v$ devient :

$$\left(M - \frac{1}{\lambda}N\right) v = 0$$

- Pour Jacobi ; l'équation devient :

$$\left(D + \frac{1}{\lambda}E + \frac{1}{\lambda}F\right) v = 0$$

soit $C = D + \frac{1}{\lambda}E + \frac{1}{\lambda}F$. si $|\lambda| \geq 1$, on aurait :

$$|c_{ii}| = |a_{ii}| > \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}| \geq \sum_{\substack{j=1 \\ j\neq i}}^{n} \left|\frac{a_{ij}}{\lambda}\right| = \sum_{\substack{j=1 \\ j\neq i}}^{n} |c_{ij}|$$

donc C serait à diagonale strictement dominante et par conséquent inversible. C inversible implique que $Cv = 0$ donc $v = 0$. Or $v \neq 0$, d'où la contradiction et donc on a bien $|\lambda| < 1$. - Pour Gauss-Seidel ; l'équation devient :

$$\left(D + E + \frac{1}{\lambda}F\right) v = 0$$

en posant encore $C = D + E + \frac{1}{\lambda}F$. et en supposant $|\lambda| \geq 1$, on aurait :

$$|c_{ii}| = |a_{ii}| > \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}| \geq \sum_{j<1}^{n} \left|\frac{a_{ij}}{\lambda}\right| + \sum_{j>1}^{n} \left|\frac{a_{ij}}{\lambda}\right| = \sum_{\substack{j=1 \\ j\neq i}}^{n} |c_{ij}|$$

et on obtient le même type de contradiction. ■

Cas des matrices symétriques définies positives

Théoreme 1.7.7 Si A est une matrice symétrique définie positive, alors les mé thodes de Gauss-Seidel et de relaxation pour ($\omega \in]0,2[$) convergent.

La convergence de la méthode est d'autant plus rapide que $\rho(M^{-1}N)$ est petit. Or cette matrice $B = M^{-1}N$ dépend de ω. Une étude théorique des valeurs propres de B montre que l'allure de la courbe $\rho(B)$ en fonction de B est décroissante entre 0 et ω_{opt} et croissante entre ω_{opt} et 2. Par ailleurs, on a toujours $1 < \omega_{opt} < 2$. On a donc intérêt à choisir ω le plus proche possible de ω_{opt}.

La méthode de correction

Soit le vecteur reste en x défini comme :

$$r(x) = b - Ax$$

et $\left\{r^{(k)}\right\}$ le reste en $\left\{x^{(k)}\right\}$. On appelle également l'erreur en k le vecteur

$$e^{(k)} = x^{(k)} - \bar{x}$$

où $\bar{x}$ est la solution. si on a une approximation $\left\{x^{(0)}\right\}$ de x, la relation suivante est vérifiée :

$$Ae^{(0)} = A(x^{(0)} - \bar{x}) = A(x^{(0)}) - b = -r^{(0)}$$

ce qui signifie que $e^{(0)}$ est la solution du système $Ax = -r^{(0)}$ et théoriquement, on a $\bar{x} = x^{(0)} - e^{(0)}$. Pratiquement, en appliquant au système $Ax = -r^{(0)}$ la méthode directe qui nous a fourni $x^{(0)}$, on n'obtient pas directement $e^{(0)}$, mais une approximation $y^{(0)}$ de $e^{(0)}$. Si on pose $x^{(1)} = x^{(0)} - y^{(0)}$, $x^{(1)}$ est une nouvelle approximation de $\bar{x}$, en itérant les calculs précédents, on obtient :

$$Ae^{(1)} = A(x^{(1)} - \bar{x}) = A(x^{(1)}) - b = -r^{(1)}$$

la résolution du système $Ax = -r^{(1)}$ donnera une approximation $y^{(1)}$ de $e^{(1)}$, et une nouvelle approximation $x^{(2)}$ de $\bar{x}$:

$$x^{(2)} = x^{(1)} - y^{(1)} = x^{(0)} - y^{(0)} - y^{(1)}$$

Ces calculs peuvent être itérer autant de fois que nécessaire, pour s'arrêter lorsque le reste est suffisament petit. A la $k^{ième}$ itération, les relations suivantes sont vérifiées pour $y^{(k-1)}$ approximation de $e^{(k-1)}$:

$$x^{(k)} = x^{(k-1)} - y^{(k-1)} = x^{(0)} - \sum_{i=0}^{k-1} y^{(i)}$$

avec $y^{(i)}$ une approximation de $e^{(i)}$, solution de $Ax = -r^{(i)}$ et $i = 0,1,2,...,k-1$. Si nous nous arrêtons lorsque $k = N$, il est nécessaire de ré soudre $N+1$ systèmes linéaires : d'abord $Ax = b$, pour obtenir $x^{(0)}$ puis $Ax = -r^{(i)}$ et $i = 0,1,2,...,N-1$ afin d'obtenir $y^{(i)}$. Une fois la matrice A décomposée (en LU ou Cholesky), il s'agit donc de résoudre les systèmes $LUx = -r^{(i)}$ où $-r^{(i)}$ a été calculé par la relation $r^{(i)} = b - Ax^{(i)}$.

1.8 SERIE D'EXERCICES

Exercice 1.5 Résoudre le système d'équations linéaires suivant :

$$\begin{cases} 10x_1 - 2x_2 - 2x_3 &= 6 \\ -x_1 + 10x_2 - 2x_3 &= 7 \\ -x_1 - x_2 + 10x_3 &= 8 \end{cases}$$

Par la méthode des approximations successives. Arrêter les calculs dès que :

$$\left| x_i^{(k+1)} - x_i^{(k)} \right| < 10^{-2}$$

■

Exercice 1.6 Résoudre le système d'équations linéaires suivant :

$$\begin{cases} 10x_1 - 2x_2 - 2x_3 &= 6 \\ -x_1 + 10x_2 - 2x_3 &= 7 \\ -x_1 - x_2 + 10x_3 &= 8 \end{cases}$$

Par la méthode de Seidel. Arrêter les calculs dès que :

$$\left| x_i^{(k+1)} - x_i^{(k)} \right| < 10^{-2}$$

■

Exercice 1.7 Résoudre le système d'équations linéaires suivant :

$$\begin{cases} 10x_1 - 2x_2 - 2x_3 &= 6 \\ -x_1 + 10x_2 - 2x_3 &= 7 \\ -x_1 - x_2 + 10x_3 &= 8 \end{cases}$$

Par la méthode de relaxation. Faire les calculs avec deux décimales.

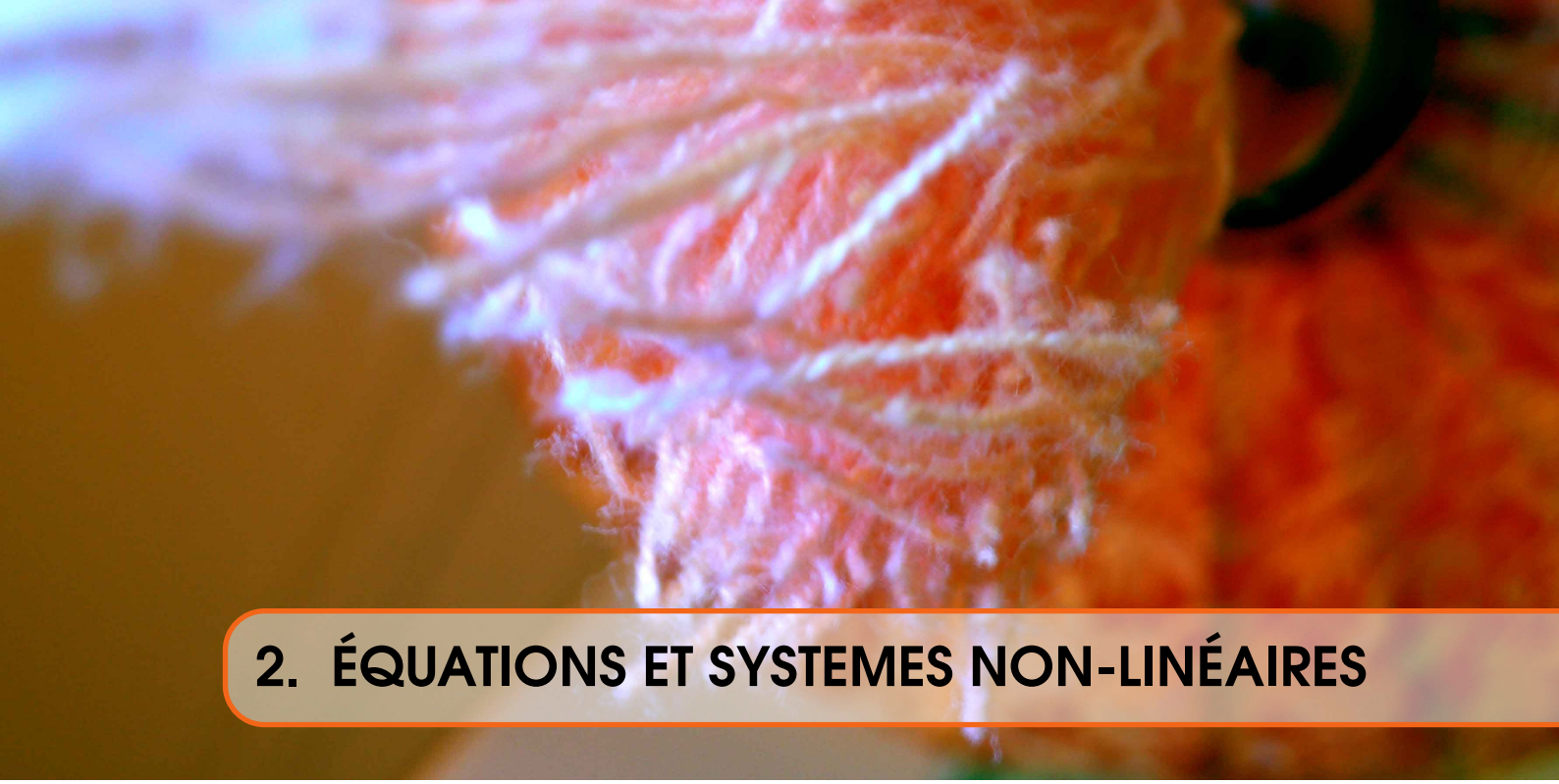
■

Exercice 1.8 Résoudre le système d'équations linéaires suivant :

$$\begin{cases} 10x_1 + x_2 + x_3 &= 12 \\ 2x_1 + 10x_2 + x_3 &= 13 \\ 2x_1 + 2x_2 + 10x_3 &= 14 \end{cases}$$

Par la méthode de relaxation. Faire les calculs avec quatre décimales.

■



2. ÉQUATIONS ET SYSTEMES NON-LINÉAIRES

2.1 RÉSOLUTION DES ÉQUATIONS ET SYSTÈMES NON-LINÉAIRES

Ce chapitre est consacré à quelques méthodes numériques de résolution des équations du type :

$$f(x) = 0 \tag{2.1}$$

où l'application : $f : [a,b] \subset \mathbb{R} \longmapsto \mathbb{R}$ est supposée suffisamment régulière (continue et dérivable) sur l'intervalle $[a,b]$.

C'est-à-dire que nous allons approcher les racines de cette équation (2.1) sur $[a,b]$.

L'équation (2.1) represente une multitude de problèmes (équations algèbriques (où f est un polynôme), trigonométriques, exponentielles...).

Le problème revient donc à trouver x vérifiant $f(x) = 0$ sans qu'on puisse déterminer x explicitement.

Une équation du type (2.1) recouvre beaucoup d'applications.

Comme exemples :

1. On veut déterminer le volume V d'un gaz à une temperature T et une pression P. L'équation d'état qui lie V,T et P est la suivante :

   $$(P + \alpha(\frac{n}{V})^2)(V - n\beta) = knT$$

   où α et β sont des coefficients qui dépendent de la nature du gaz, n le nombre de molécules contenues dans le volume V et k représente la constante de Boltzman. Il est nécessaire donc de résoudre une équation non linéaire où l'inconnue est V. Ce qui revient à trouver les racines de la fonction

   $$f(V) = (P + \alpha(\frac{n}{V})^2)(V - n\beta) - knT$$

   Il s'agit donc de résoudre une équation non linéaire dont on n'est pas capable de trouver une solution exacte.

2. Le lancement d'un projectile. Sa trajectoire est décrite (par la loi de Newton), par une fonction $t \mapsto x(t) = (x_1(t), x_2(t))$ qui doit satisfaire une équation du type

$$\ddot{x} = F(t, \dot{x}, x).$$

Où $\ddot{x} = \frac{d^2x}{dt^2}$ est l'acceleration et $\dot{x} = \frac{dx}{dt}$ la vitesse du projectile. Chercher par exemple à savoir à quel moment le projectile retombe sur le sol revient à résoudre :

$$x_2(t) = 0.$$

Ainsi, si on dispose d'une méthode numérique pour estimer $x(t)$, on pourra utiliser les méthodes de ce chapitre pour résoudre

$$x_2(t) = 0.$$

Puisqu'en général la solution d'une équation $f(x) = 0$ ne s'exprime pas par une formule, on ne peut espèrer trouver une solution exacte en un nombre fini d'étapes. Nous allons donc approcher les solutions avec une précision aussi bonne qu'on le souhaite.

Mathématiquement, cela signifie qu'on a une suite $(x_n)_{n\in\mathbb{N}}$ de solutions approchées , c'est-à-dire telle que $x_n \to x^*$ où x^* est une racine de : $f(x^*) = 0$.

Avoir des méthodes pour obtenir des solutions de $f(x) = 0$ de manière approchée est intéressant mais, si on veut les appliquer à des problèmes réels, le temps qu'il faudra attendre pour obtenir la réponse est important.

Par exemple, en ce qui concerne le projectile heurtant le sol, le coût de calcul de $x(t)$ peut être relativement élevé et on voudrait donc que la méthode de résolution de $x_2(t) = 0$ converge en aussi peu d'étapes que possible. En effet, le résultat de ce calcul est peut-être utilisé pour prendre des décisions quant à la trajectoire ultérieure du projectile.

Cette vitesse de convergence s'exprime ici par le gain de précision qu'on gagne en passant de x_n à x_{n+1}.

On s'interesse d'abord aux méthodes de séparation des racines. Il s'agit de determiner des intervalles $[a_i, b_i]$ à l'interieur desquels $f(x)$ admet une racine et une seule.

Cette séparation des racines s'effectue en général :

1. Soit sur le graphe de la fonction $y = f(x)$.
2. Soit les graphes de $y_1 = f_1(x)$ et $y_2 = f_2(x)$, si on peut mettre $f(x) = 0$ sous la forme $f_1(x) - f_2(x) = 0$.
3. Soit en se basant sur le théoreme suivant :

Théoreme 2.1.1 Pour a et b donnés :

— Si $f(a).f(b) < 0$ alors f admet au moins une racine dans $[a,b]$ si de plus $f'(x) \neq 0$ quelque soit $x \in [a,b]$, la racine est unique.

— Si $f(a).f(b) > 0$ alors f n'admet pas de racine dans $[a,b]$ ou $f(x)$ admet un nombre pair de racines dans $[a,b]$.

Apres avoir isolé une racine dans $[a,b]$, on peut en obtenir une approximation à l'aide de plusieurs méthodes numériques.

Nous allons décrire quelques unes de ces méthodes ci-dessous.

2.2 MÉTHODE DE BISSECTION OU DE DICHOTOMIE

Cette méthode permet à la fois de montrer l'existence d'une racine d'une fonction $f : [a,b] \longmapsto \mathbb{R}$ et de l'estimer numériquement.

L'idée est : si f est continue et change de signe sur $[a,b]$, f s'annule en un certain point de $[a,b]$.

Définition 2.2.1 Choisissons un point quelconque $x_0 \in]a,b[$.

1- Si $f(x_0) = 0$, x_0 est la racine et on a fini.

Sinon, supposons par exemple que $f(a) < 0$ et $f(b) > 0$.

Soit x_0 milieu de $[a,b]$, la racine x^* supposée existante se trouve dans l'un des deux intervalles $[a,x_0]$, $[x_0,b]$, pour savoir lequel, on regarde les conditions du théorème ci-dessus.

2- Si $f(x_0) < 0$, alors il y a une racine dans $]x_0,b[$. On pose dans ce cas $a_1 = x_0$, $b_1 = b$

3- Sinon, $f(x_0) > 0$ et il doit y avoir une racine dans $]a,x_0[$. On pose $a_1 = a$, $b_1 = x_0$. On recommence la procédure en choisissant x_1 dans $[a_1,b_1]$ et ainsi de suite, ce qui donne une suite décroissante d'intervalles $[a_n,b_n]$ avec $x_0 = \frac{a+b}{2}$, $x_1 = \frac{a_1+b_1}{2}$, ..., $x_n = \frac{a_n+b_n}{2}$ contenant chacun une racine. Et qui verifient :

$$|a - x_n| \leqslant \left(\frac{b-a}{2^{n+1}}\right) \tag{2.2}$$

R L'équation (2.4) permet d'estimer le nombre d'itérations nécessaires pour approcher x^* avec une précision donnée ε .

En effet, si on veut savoir à partir de quel n on a $|x_n - x^*| \leq \varepsilon$, il suffit de chercher n tel que $(1/2^n)|b-a| \leq \varepsilon$. C'est-à-dire $n \geq log_2(|b-a|/\xi)$ où ξ dénote le plus petit entier supérieur ou égal à ε .

Pour que a_n et b_n soient de bonnes approximations d'une racine x^* : $a_n \leqslant x^* \leqslant b_n$ et $a_n \to x^*, b_n \to x^*$. Il faut que la longueur de l'intervalle $[a_n,b_n]$ tende vers 0.

Le théorème donnant le résultat s'énonce comme suit :

Théoreme 2.2.1 Soit $f : [a,b] \to \mathbb{R}$ une fonction continue. Si $f(a).f(b) < 0$, la fonction f possède au moins une racine dans $]a,b[$. De plus, si on définit par récurrence $[a_0,b_0] = [a,b]$,

$$x_n = \frac{1}{2}(a_n + b_n) \text{ et } [a_{n+1},b_{n+1}] = \begin{cases} [a_n,x_n] & \text{si } f(a_n)f(x_n) < 0, \\ [x_n,x_n] = \{x_n\} & \text{si } f(x_n) = 0, \\ [x_n,b_n] & \text{si } f(x_n)f(b_n) < 0, \end{cases} \tag{2.3}$$

les trois suites (a_n), (b_n) et (x_n) convergent linéairement vers la même limite x^* avec $f(x^*) = 0$.

Démonstration. On suppose $f(a) < 0$ et $f(b) > 0$. Sinon on remplace f par $-f$.

Montrons par récurrence que $[a_n,b_n]$ est bien défini et que $f\ (a_n)\ f\ (b_n) < 0$ sauf si $a_n = b_n$ dans ce cas $f(a_n) = f(b_n) = 0$.

- Le cas $n = 0$ est trivial.

Supposons que la formule soit vraie pour n et montrons la pour $n+1$.

Soit $a_n = b_n$ sont racines et alors $a_{n+1} = b_{n+1} = a_n$ sont aussi racines. Soit $a_n \neq b_n$ et $f(a_n).f(b_n) < 0$, ce qui implique que

- si $f(a_n).f(x_n) < 0$ ou $f(x_n).f(b_n) < 0$, $a_{n+1} \neq b_{n+1}$ et $f(a_{n+1})f(b_{n+1}) < 0$;

- sinon, $f(a_n).f(x_n) \geq 0$ et $f(x_n).f(b_n) \geq 0$ d'où on déduit que $f(x_n) = 0$ et que $a_{n+1} = b_{n+1} = x_n$ sont des racines de f .

il est facile de constater que :

$$\forall n \in \mathbb{N}, [a_{n+1}, b_{n+1}] \subseteq [a_n, b_n] \quad \text{et} \quad |b_{n+1} - a_{n+1}| \leq \frac{1}{2}|b_n - a_n|.$$

Cela implique que la suite (x_n) est de Cauchy.

Soit $\varepsilon > 0$. Comme $\frac{1}{2n} \to 0$, il existe alors $n_0 \in \mathbb{N}$ tel que $n \geq n_0 \Longrightarrow (1/2n)|b_0 - a_0| \leq \varepsilon$.

Pour les $m \geq n \geq n_0$, on a $x_m \in [a_m, b_m] \subseteq [a_n, b_n]$ et alors

$$|x_m - x_n| \leq |b_n - a_n| \leq \frac{1}{2}|b_{n-1} - a_{n-1}| \leq \ldots \leq \frac{1}{2^n}|a_0 - b_0| \leq \varepsilon$$

La suite (x_n) est donc bien de Cauchy.

Par conséquent, il existe un $x^* \in [a, b]$ tel que $x_n \to x^*$.

En outre, comme $|x_n - a_n| \leq |b_n - a_n| \to_{n\to\infty} n_0$ et $|b_n - x_n| \leq |b_n - a_n| \to_{n\to\infty} n_0$, il est facile de montrer que (a_n) et (b_n) convergent aussi vers x^*. Puisque $f(a_n)f(b_n) \leq 0$ pour tout n, on en déduit en passant à la limite sur n et en utilisant la continuité de f que $f(x^*)^2 \leq 0$, c'est-à-dire $f(x^*) = 0$.

Nous avons montré que f possède une racine (x^*) et que les suites (a_n), (b_n) et (x_n) convergent toutes trois vers x^* .

Cette convergence est linéaire. Nous allons le voir pour (x_n), (il en est de même pour (a_n) et (b_n)).

Comme $(x_m)_{m\geq n} \subseteq [a_n, b_n]$ et que $x_m \to x^*$, on a $x \in [a_n, b_n]$. En conséquence

$$|x_n - x^*| \leq |b_n - a_n| \leq \frac{1}{2^n}|b_0 - a_0| \tag{2.4}$$

où $c = 1/2 \in]0, 1[$. ■

2.3 MÉTHODE DES APPROXIMATIONS SUCCESSIVES (du type $x_{n+1} = F(x_n)$)

R

L'équation (2.1) peut toujours se mettre sous la forme

$$x = F(x) \tag{2.5}$$

Il suffit par exemple de poser : $F(x) = x + f(x)$.

Définition 2.3.1 Soit $F : \mathbb{R} \to \mathbb{R}$ une fonction numérique.
Si $x \in \mathbb{R}$ est tel que $F(x) = x$, on dit que x est un point fixe de F.

Après avoir isolé une racine dans l'intervalle $[a, b]$, on peut utiliser la proposition suivante pour l'approcher :

Proposition 2.3.1 : Soit $F : [a, b] \subset \mathbb{R} \longmapsto [a, b] \subset \mathbb{R}$ une fonction Lipschitzienne de rapport k avec $0 < k < 1$ (on dit dans ce cas, strictement contractante). C'est-à-dire :

$$\forall x, y \in [a, b], |\, F(x) - F(y)\, | \leq k|x - y| \tag{2.6}$$

Alors la suite définie par :

$$x_{n+1} = F(x_n), \forall x_0 \in [a,b] \tag{2.7}$$

converge vers la racine x^* quand n tend vers l'infini.
De plus on a l'estimation de l'erreur comme suit :

$$|x_n - x^*| \leq \frac{k^n}{1-k}|x_1 - x_0| \tag{2.8}$$

Remarques :

R La condition strictement contractante peut être remplacée par :

$$|F'(x)| < 1, \forall x \in [a,b] \tag{2.9}$$

R Si $0 \leq F'(x) < 1$, la suite (x_n) converge vers x^* de façon monotone.

R Si $-1 < F'(x) \leq 0$, la suite (x_n) converge vers x^* alternativement par excès et par defaut.

R Si $|F'(x)| > 1$, la suite (x_n) diverge.

R L'écriture de l'équation (2.1) sous une forme (2.5) quelconque n'est pas unique et ne donne pas toujours une méthode convergente, en effet :
Si on cherche la racine de $\tan x - x = 0$ pour $\pi \leq x \leq \frac{3\pi}{2}$, on écrit soit :

1. $x = \tan x$ c'est-à-dire :

$$F_1(x) = \tan x$$

2. soit $x = \pi + \arctan x$ c'est-à-dire :

$$F_2(x) = \pi + \arctan x$$

On obtient :

$$F_2'(x) = \frac{1}{1+x^2}$$

et $|F_2'(x)| < 1$, la méthode converge. Mais $F_1'(x) = 1 + \tan^2 x$ et $|F_1'(x)| > 1$, la méthode diverge.

R Si on cherche les deux racines de l'équation $x^2 - 6x + 8 = 0$ qui sont $x_1 = 2$ et $x_2 = 4$, on peut écrire :

1. Soit

$$x = \frac{x^2+8}{6}$$

c'est-à-dire :

$$F_1(x) = \frac{x^2+8}{6}$$

2. Ou bien

$$x = \sqrt{6x-8}$$

c'est-à-dire :

$$F_2(x) = \sqrt{6x-8}$$

On obtient :

$$F_1'(x) = \frac{1}{3}x$$

donc

$$|F_1'(x)| < 1, \text{ seulement si } |x| < 3 \tag{2.10}$$

et

$$F_2'(x) = \frac{3}{\sqrt{6x-8}}$$

donc

$$|F_2'(x)| < 1, \text{ seulement si } x > \frac{17}{6} \tag{2.11}$$

Pour l'équation $x = F_1(x)$ on prendra l'intervalle $[0;3]$ ce qui donnera la racine $x^* = 2$, et pour l'équation $x = F_2(x)$ on prendra l'intervalle $[3;5]$ ce qui donnera la racine $x^* = 4$.

2.4 MÉTHODE DU TYPE $x_{n+1} = x_n - \frac{f(x_n)}{g(x_n)}$

On peut écrire ces algorithmes sous la forme (2.5) avec : $F(x) = x - \frac{f(x)}{g(x)}$

Donc si $|F'(x)| < 1, \forall x \in [a,b]$ ce qui veut dire : $|1 - \frac{g(x)f'(x) - g'(x)f(x)}{g(x)^2} F'(x)| < 1$, le schéma : $x_{n+1} = x_n - \frac{f(x_n)}{g(x_n)}$ converge vers la solution x^* de (2.1) pour x_0 convenablement choisi.

2.4.1 Méthode de la sécante

Soit c un point de $[a,b]$ tel que $f(c) \neq 0$. On choisit un point initial x_0 tel que $f(x_0).f(c) < 0$. La corde (ou sécante) joignant les points $M_c = (c, f(c))$ et $M_{x_0} = (x_0, f(x_0))$ coupe l'axe des x en un point dont l'abscisse est notée x_1. Et on recommence la procédure avec M_c et $M_1 = (x_1, f(x_1))$. Et ainsi de suite.

On obtient une suite (x_n) définie par :

$$x_{n+1} = x_n - \frac{f(x_n)}{f(x_n) - f(c)}(x_n - c) = F(x_n).$$

La convergence vers la solution x^* de (2.1) est assurée par un choix convenable de c tel que $|F'(x)| < 1$ dans un voisinage contenant les points (x_n) d'itération.

R Si $f''(x) > 0$ sur $[a,b]$, alors si le point c est tel que $f(c) > 0$, la suite (x_n) est monotone convergente vers la solution x^*, par excès si $f'(x) < 0$ sur $[a,b]$, par défaut si $f'(x) > 0$ sur $[a,b]$.
Si $f''(x) > 0$ sur $[a,b]$, alors si le point c est tel que $f(c) < 0$, la suite (x_n) est monotone convergente vers la solution x^*, par excès si $f'(x) > 0$ sur $[a,b]$, par défaut si $f'(x) < 0$ sur $[a,b]$.

2.4.2 Méthode de la fausse position ou de Régula-falsi

On peut améliorer la convergence de la méthode de la bissection en s'inspirant de la méthode de dichotomie. L'idée est, au lieu de prendre pour x_n le point milieu de $[a_n, b_n]$, il vaudrait peut-être mieux choisir x_n comme l'intersection du segment de droite joignant $(a_n, f(a_n))$ et $(b_n, f(b_n))$ avec l'axe des "x" : $\mathbb{R} \times \{0\}$.

Cela donne la formule :

$$x_n = a_n - \frac{b_n - a_n}{f(b_n) - f(a_n)} f(a_n).$$

On peut espérer ainsi que x_n converge plus vite vers x^* . Le désavantage de ce choix est que nous aurons besoin de plus d'hypothèses sur f pour montrer cette convergence.

Théoreme 2.4.1 Soit $f \in C([a,b];\mathbb{R}) \cap C^1(]a,b[;\mathbb{R})$ une fonction convexe ou concave et $f(a)f(b) < 0$. Définissons a_n, b_n, x_n par la récurrence suivante : $a_0 = a, b_0 = b$ et

$$x_n = a_n - \frac{b_n - a_n}{f(b_n) - f(a_n)} f(a_n), \qquad [a_{n+1}, b_{n+1}] = \begin{cases} [a_n, x_n] & \text{si} \quad f(a_n)f(x_n) < 0, \\ [x_n, b_n] & \text{si} \quad f(x_n)f(b_n) < 0, \end{cases} . \tag{2.12}$$

Alors, soit il existe un n tel que $f(x_n) = 0$, soit x_n est bien défini pour tout n et x_n converge à l'ordre 1 vers x^* où x^* est l'unique racine de f dans $[a,b]$.

Cette méthode dite de Regula -Falsi converge dans les mêmes conditions que la méthode de la sécante et en général plus vite.

2.4.3 Méthode de la tangente ou Méthode de Newton

Soit x_0 un point de $[a,b]$. La tangente à la courbe $y = f(x)$ au point $M_0 = (x_0, f(x_0))$ coupe l'axe des x en un point d'abscisses x_1. En itérant le procédé, on obtient une suite d'abscisses (x_n) définie par :

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = F(x_n)$$

La méthode de Newton peut être vue comme un cas limite de la méthode de la sécante où les deux points x_{n-1} et x_n sont tellement proches que $(f(x_n) - f(x_{n-1}))/(x_n - x_{n-1})$ se confond avec $f'(x_n)$. Ainsi on obtiendra x_{n+1} à partir de x_n en regardant l'intersection de la tangente f au point x_n avec l'axe des x. Comme cette tangente est constituée de l'ensemble des points (x,y) tels que $y = f(x_n) + f'(x_n)(x - x_n)$ et que le point recherché est du type $(x_{n+1}, 0)$. Au voisinage de la racine, cette méthode converge plus vite que la méthode de la sécante. et $x_n \to x^*$ à l'ordre 2.

La condition $|F'(x)| < 1$ dans un voisinage contenant les points (x_n) d'itération est en général satisfaite car :

$$F'(x^*) = \frac{f(x^*).f''(x^*)}{(f'(x^*))^2} = 0$$

Posons dans le voisinage contenant les points (x_n) d'itération, M=sup $|f''(x^*)|$ et m=inf $|f'(x^*)|$. La formule de Taylor dans ce voisinage contenant les points (x_n) d'itération donne l'estimation de l'erreur pour une itération

$$|x_n - x^*| \leq \frac{M}{m} |x_{n-1} - x^*|^2$$

Ce qui donne

$$|x_n - x^*| \leq \left(\frac{M}{m}\right)^{2^n-1} |x_0 - x^*|^{2^n}$$

R Lorsque la méthode converge, suivant le signe de $f''(x)$, nous avons :

1. $f''(x) > 0$, si $f'(x) > 0$ sur $[a,b]$ (respectivement $f'(x) < 0$) , alors la suite (x_n) est monotone convergente vers la solution x^*, par excès (respectivement par défaut).
2. $f''(x) < 0$, si $f'(x) < 0$ sur $[a,b]$ (respectivement $f'(x) > 0$) , alors la suite (x_n) est monotone convergente vers la solution x^*, par excès (respectivement par défaut).

R

1. Quelques faiblesses de la méthode de Newton lorsqu'on travaille sur de trop grands voisinages de la racine x^* .

   Soit $f : [-\pi/2, \pi/2] \to \mathbb{R} : x \mapsto \sin x$. Cette fonction possède une racine simple unique : $x^* = 0$.
   La méthode de Newton s'écrit :

   $$x_0 \in [-\pi/2, \pi/2], \qquad x_{n+1} = x_n - \tan x_n, \qquad n \geq 0$$

   Choisissons $x_0 = \alpha$ où α est la racine strictement positive de $\tan x = 2x$.
   Dans ce cas,

   $$x_1 = x_0 - \tan x_0 = \alpha - 2\alpha = -\alpha$$

   et

   $$x_2 = x_1 - \tan x_1 = -\alpha - \tan(-\alpha) = -\alpha + \tan\alpha = -\alpha + 2\alpha = \alpha.$$

   On est revenu à x_0. !
   Ensuite le processus recommence :

   $$x_3 = -\alpha, \quad x_4 = \alpha, \quad x_5 = -\alpha, \ldots$$

   La suite $(x_n)_{n\in\mathbb{N}}$ alterne donc entre α et $-\alpha$.
   On dit que c'est une *orbite périodique* de période 2 ou un *cycle d'ordre deux*. En conséquence (x_n) ne converge pas vers 0. Cela met en évidence qu'on doit partir suffisament près de la racine afin d'assurer la convergence de la méthode.
   Ici on peut montrer que, si $|x_0| < \alpha$, alors $x_n \to 0 = x^*$.

2. La méthode de Newton n'est pas nécessairement plus performante que les autres méthodes si on est trop loin de la racine. Il est donc important de determiner un intervalle $[a,b]$, contenant la racine x^*, le plus petit possible, de manière à ce que x_0 soit le plus proche possible de x^* . Sinon l'algorithme peut converger vers une autre racine ou même diverger.

2.5 MÉTHODE DU POINT FIXE

Si on regarde la méthode de Newton d'un point de vue abstrait, on voit qu'on obtient x_{n+1} à partir de x_n en évaluant toujours la même expression. Plus précisément, on a $x_{n+1} = F(x_n)$ avec $F(x) = x - f(x)/f'(x)$. Si $x_n \to x^*$, on déduit immédiatement de la continuité de F que $x^* = F(x^*)$.

On dit alors que x^* est un *point fixe* de F. Or, il se fait que les points fixes de F correspondent aux racines simples de f .

Nous disposons maintenant d'un cadre pour rechercher de nouveaux algorithmes.

En effet, à toute fonction F dont les points fixes correspondent aux solutions du problème, on peut associer un schéma récursif $x_{n+1} = F(x_n)$.

Quelles sont donc les propriétés que F doit posséder pour être intéressante ? C'est-à-dire :

1. On doit avoir (x_n) convergente et sa limite x^* sera alors un point fixe de F ;
2. Et (x_n) doit tendre vers x^*aussi vite que possible et l'ordre de convergence doit être aussi élevé que possible.

Théoreme 2.5.1 Soit $F : [a,b] \to [a,b]$ une fonction. On suppose qu'il existe une constante $K \in [0,1[$ telle que

$$\forall x, y \in [a,b], \qquad |F(x) - F(y)| \leq K|x-y|. \tag{2.13}$$

Alors, F possède un unique point fixe $x^* \in [a,b]$ et pour tout $x_0 \in [a,b]$, la suite $(x_n)_{n\in\mathbb{N}}$ 0 définie par $x_{n+1} = F(x_n)$ converge vers x^* .

R Une fonction qui satisfait (2.13) pour $K \in [0,+\infty[$ est dite lipchitzienne. Lorsque $K < 1$, on dit que F est une contraction.

R Le plus petit K qui satisfait (2.13) (le K optimal) est appelé la constante de Lipschitz de la fonction F et se note $Lip(F)$. Ainsi

$$|(x) - (y)|Lip(F) = Lip_{[a,b]}(F) = sup_{x,y\in[a,b]x\neq y}\frac{|F(x) - F(y)|}{|x-y|}.$$

R Les fonctions qui satisfont (2.13) sont continues. L'inverse n'est pas vrai.

R Si $F \in C^1(]a,b[;\mathbb{R})$, on peut montrer grâce au théorème de la moyenne que

$$Lip(F) = sup_{x]a,b[}|F(x)|.$$

En conséquence, F sera une contraction si et seulement si $sup_{x\in]a,b[}|F'(x)| < 1$. Si de plus F est dérivable en a et b, il découle de la compacité de $[a,b]$ que F est une contraction si et seulement si, pour tout$x \in [a,b]$, $\quad |F'(x)| < 1$.

R Du point de vue de l'existence, l'intérêt de ce théorème est qu'il est valable en dimension supérieure à 1. En effet, en dimension 1, la continuité suffit. Notons cependant que, dans ce cas, la convergence des suites (x_n) n'est pas assurée et en fait n'a pas nécessairement lieu. Leur comportement peut d'ailleurs être fort complexe .

Le théorème (2.5.1) donne un critère pour la convergence des suites sur un intervalle $[a,b]$.Et on peut l'appliquer au voisinage d'un point fixe. On en conclut que pour tout $x_0 \in I_\varepsilon$, la suite $(x_n)_{n\in\mathbb{N}}$ définie par $x_{n+1} = F(x_n)$ converge bien vers x^* .

R

Si $|F'(x^*)| < 1$, les suites qui entrent dans un petit voisinage de x^* convergent vers x^* . On dit que x^* est un point fixe *attractif*.
Si $|F'(x^*)| > 1$, même si une suite entre dans un petit voisinage de x^* , elle est forcée d'en ressortir. On dit de x^* que c'est un point fixe *répulsif*.
Si $|F'(x^*)| = 1$, on ne peut rien dire Les deux situations ci-dessus peuvent se produire. Ou aucune d'elles. Cependant on peut penser $|F'(x^*)| = 1$ comme une transition entre $|F'(x^*)| < 1$ et $|F'(x^*)| > 1$, c'est à dire entre un point fixe qui était attractif et devient répulsif. De telles situations sont communes et, typiquement, lorsque $|F'(x^*)| = 1$, une *bifurcation* a lieu.

Nous avons examiné la convergence – ou non – des suites vers un point fixe. Nous voudrions aussi connaitre la vitesse de convergence de x_n vers x^* . Globalement, le théorème (2.5.1) ne nous offre qu'une convergence linéaire. En effet, l'équation (2.13) implique

$$|x_{n+1} - x^*| = |F(x_n) - F(x^*)| \leq K|x_n - x^*|.$$

Lorsqu'on est suffisamment proche du point fixe x^*, la méthode de Newton est quadratique. En faisant un développement deTaylor avec reste de F au point x^*. On écrit

$$F(x) = F(x^*) + F(x^*)(x-x^*) + \cdots + \frac{F^{(k-1)}(x^*)}{(k-1)!}(x-x^*)^{k-1} + \frac{F'^{(k)}(\xi)}{k!}(x-x^*)^k$$

Et nous avons le théorème suivant :

Théoreme 2.5.2 Sous les hypothèses du théorème (2.5.1), si de plus on a $F \in C^k(]a,b[;\mathbb{R})$ et $F'(x^*) = 0, ..., F^{(k-1)}(x^*) = 0$, alors (x_n) converge vers x^* à l'ordre k. Plus précisément, on a

$$|x_{n+1} - x^*| \leq c|x_n - x^*|^k$$

où $c > |kF^{(k)}(x^*)|/k!$ peut être choisi arbitrairement proche de $|F^{(k)}(x^*)|/k!$.

2.6 SERIE D'EXERCICES

Exercice 2.1 Montrer en utilisant la propriété de valeur intermédiaire que toute fonction continue $f : [a,b] \to [a,b]$ possède au moins un point fixe. ■

Exercice 2.2 Utiliser l'algorithme de dichotomie pour calculer à 0.01 près la racine de :

$$f(x) = e^x \sin x - 1$$

dans l'intervalle $[0, \pi/2]$. ■

Exercice 2.3 En utilisant une méthode de convergence de la forme $x_{n+1} = F(x_n)$, trouver la racine à 0.01 près de :

$$f(x) = xe^x - 1 = 0$$

dans l'intervalle $[1/2, 1]$. ■

Exercice 2.4 On considère la fonction f définie par $f(x) = x^3 + x - 1$, $x \in \mathbb{R}$.

1. Montrer que $f(x) = 0$ admet une racine réelle unique $x^* \in]0,1[$.
2. Déterminer par la méthode de dichotomie, une approximation de x^* à 10^{-1} près en utilisant le test d'arrêt $|x_{n+1} - x_n| \leq \varepsilon$. Comparer le nombre d'itérations effectif pour avoir cette précision avec le nombre $N = (\frac{\log(\frac{b-a}{\varepsilon})}{\log 2}) = 3$.
3. Effectuer deux itérations avec la méthode de Newton en partant de $x_0 = 1$.

■

Exercice 2.5 En utilisant la méthode de Newton, chercher la racine à 0.001 près de l'équation :

$$f(x) = x^4 + x^2 + 2x - 1 = 0$$

dans l'intervalle $[0,1]$. ■

Exercice 2.6 Trouver par la méthode de Newton, la racine positive minimale de l'équation :

$$\tan x = x$$

0.0001 près. ■

Exercice 2.7

1. Trouver le nombre de racines réelles distinctes de :

$$P_3(x) = x^3 - 3x^2 - x + 3$$

2. Existent-t-ils des racines multiples ?

■

Exercice 2.8 1. Trouver le nombre de racines réelles distinctes de :

$$P_3(x) = x^3 - 5x^2 + 7x - 3$$

2. Existent-t-ils des racines multiples ?

■

Exercice 2.9 1. Trouver le nombre de racines réelles distinctes de :

$$P_3(x) = x^4 - x^3 - 3x^2 + 5x - 2$$

2. Existent-t-ils des racines multiples ?

■

Exercice 2.10 Résoudre graphiquement l'équation cubique :

$$x^3 - 1.75x + 0.75 = 0$$

■

Exercice 2.11 1. Trouver par la méthode de Krylov, le polynôme caractéristique de la matrice suivante :

$$\begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 1 \\ 3 & 1 & 2 \end{pmatrix}$$

2. Localiser les différentes valeurs propres.
3. Donner, par la méthode de Newton, une estimation de la valeur propre négative à 0.001 près.

■

Exercice 2.12 On considère la fonction $f(x) = 4 + 8x^2 - x^4$.

1. Combien f possède-t-elle de racines ? Si on décide d'utiliser la méthode de bissection, quelles sont les paires de points initiaux qu'on peut choisir pour obtenir chacune des racines ?
2. Si on opte pour la méthode de Newton, donner des intervalles autour de chacune des racines sur lesquels la méthode de Newton converge.

■

Exercice 2.13 L'équation $x^3 + 4x^2 - 10 = 0$ peut se réécrire sous la forme d'un point fixe des trois façons suivantes :

$$\begin{aligned} x &= \varphi_1(x) = \sqrt{\frac{10 - x^3}{4}} \\ x &= \varphi_2(x) = \frac{10}{x^2 + 4x} \\ x &= \varphi_3(x) = \sqrt{\frac{10}{x + 4}} \end{aligned}$$

1. Montrer que l'équation ci-dessus possède une unique racine (qui est positive) et donc que φ_i, $i = 1,2,3$, possèdent un seul point fixe.
2. Calculer les dix premières itérées des suites (x_n) définies par $x_{n+1} = \varphi_i(x_n)$ et $x_0 = 1$ pour $i = 1,2,3$. Qu'en déduire ?
3. Tracer les graphes des fonctions φ_i. Comment les comportements observés ci-dessus se voient-ils sur ces graphiques ? Observer également la vitesse de convergence.

■

Exercice 2.14 En mécanique céleste, le calcul des positions planétaires donne lieu à l'équation de Képler :

$$m = x - E\sin(x)$$

où nous allons considérer les valeurs $m = 0,8$ et $E = 0,2$. Utilisez la méthode du point fixe pour résoudre cette équation en partant des valeurs initiales $x_0 = 1$, 0 et -1 respectivement. ■

2.7 RÉSOLUTION DES SYSTÈMES D'ÉQUATIONS NON-LINÉAIRES

2.7.1 Résolution d'une équation algèbrique

Soit

$$P_x(x) = a_1x^n + a_2x^{n-1} + a_3x^{n-2} + ... + a_nx + a_{n+1} = \sum_{i=1}^{n+1} a_ix^{n+1-i}$$

un polynôme de degré inferieur ou égal à n.

On suppose que tous les coefficients a_i sont réels.

Il existe différentes méthodes pour chercher les racines de ce polynôme, c'est à dire chercher x qui vérifie

$$P(x) = 0$$

.

Nous allons donner quelques méthodes qui nous permettront de chercher les racines réels supposées existantes de ce polynôme.

2.7.2 Propriétés sur les racines d'un polynôme

Si $x_1, x_2, x_3, ...x_n$ sont les n racines de $P_n(x) = 0$, on a les propriétés suivantes qui sont vérifiées (d'après le théorème de d'Alembert).

$$\begin{cases} \sum_{i=1}^{n} x_i & = & -\frac{a_2}{a_1} \\ \sum_{i=1}^{n-1} x_i(\sum_{j=i+1}^{n} x_i) & = & \sum_{1\le i_1<i_2\le n} x_{i_1}x_{i_2} = \frac{a_3}{a_1} \\ ..... & = & ..... \\ \sum_{1\le i_1<i_2<....<i_p\le n} x_{i_1}x_{i_2}.....x_{i_{p-1}}x_{i_p} & = & (-1)^p\frac{a_{p+1}}{a_1} \\ ..... & = & ..... \\ x_1x_2.....x_{n-1}x_n & = & (-1)^n\frac{a_{n+1}}{a_1} \end{cases}$$

En posant $S_k = \sum_{i=1}^{n} x_i^k$, on obtient les relations dites de Newton :

$$\begin{cases} a_1S_1 + a_2 & = & 0 \\ a_1S_2 + a_2S_1 + 2a_3 & = & 0 \\ ..... & = & ..... \\ a_1S_n + a_2S_{n-1} + ... + a_nS_1 + na_{n+1} & = & 0 \end{cases}$$

2.7.3 Théorème de Sturm

Le théorème de Sturm permet de calculer le nombre de racines réelles distinctes d'un polynôme dans un intervalle donné.

Suite de Sturm

On se donne un polynôme $P = x^n + a_{n-1}x^{n-1} + ... + a_1x + a_0$. La suite de Sturm (ou chaîne de Sturm à partir du polynôme P) est une suite finie de polynômes $P_0, P_1, ..., P_m$. Elle est construite par récurrence :

$P_0 = P$;

$P_1 = P\prime$, où P'est la dérivée de P, c'est-à-dire le polynôme $P' = nx^{n-1} + ... + a_1$;

Pour $i \geq 2$, P_i est l'opposée du reste de la division de P_{i-2} par P_{i-1}.

La construction s'arrête au dernier polynôme non nul.

Pour obtenir cette suite, on calcule les restes intermédiaires que l'on obtient en appliquant l'algorithme d'Euclide à P_0 et sa dérivée P_1 :

$$\begin{cases} P_0 = P_1Q_1 - P_2 \\ P_1 = P_2Q_2 - P_3 \\ \vdots \quad \vdots \quad \vdots \\ \vdots \quad \vdots \quad \vdots \\ P_{m-2} = P_{m-1}Q_{m-1} - P_m \\ P_{m-1} = P_mQ_m \end{cases}$$

Si P possède uniquement des racines distinctes, le dernier terme est une constante non nulle. Si ce terme est nul, P admet des racines multiples, et on peut dans ce cas appliquer le théorème de Sturm en utilisant la suite $T_0, T_1, ..., T_{m-1}, 1$ que l'on obtient en divisant les $P_1, P_2, ..., P_{m-1}$ par P_m.

Et le nombre de racines réelles de $P_n(x) = 0$ supposées distinctes est donné donc par le théorème suivant :

Théoreme 2.7.1 — Théorème de Sturm. Le nombre de racines réelles distinctes dans un intervalle $[a,b]$ d'un polynôme à coefficients réels, dont a et b ne sont pas des racines, est égal au nombre de changements de signe de la suite de Sturm aux bornes de cet intervalle.

Plus formellement, si nous notons $N(y)$ le nombre de changements de signe (zéro n'est pas compté comme un changement de signe) observés dans la suite $P(y), P_1(y), P_2(y), \ldots, P_m(y)$ alors le nombre de racines réelles distinctes de l'équation dans l'intervalle $[a,b]$ (où a et b ne sont pas des racines de P) est donné par $N = N(a) - N(b)$.

R Si l'equation $P_n(x) = 0$ admet une racine multiple, soit $(j+1)$ le premier indice tel que $P_{j+1}(x) = 0$. Les racines de $P_n(x) = 0$ seront alors les racines simples de $P_j(x) = 0$. Le nombre de racines distinctes est donné par le théorème de Sturm en arretant la suite $(p_n(y))$ au terme $p_i(y)$.

■ **Exemple 2.1** Supposons que l'on souhaite connaître le nombre de racines dans un certain intervalle du polynôme $p(x) = x^4 + x^3 - x - 1$.

On commence par calculer les deux premiers termes.

$p_0(x) = p(x) = x^4 + x^3 - x - 1$

$p_1(x) = p\prime(x) = 4x^3 + 3x^2 - 1$ {\{\begin{aligned}p_{0}(x)& = p(x) = x^{4} + x^{3} - x - 1\\p_{1}(x)& = p\prime(x) = 4x^{3} + 3x^{2} - 1\end{aligned}\}}

En divisant p_0 par p_1 on obtient le reste $-\frac{3}{16}x^2 - \frac{3}{4}x - \frac{15}{16}$, et en le multipliant par -1 on obtient $p^2(x) = \frac{3}{16}x^2 + \frac{3}{4}x + \frac{15}{/16}$. Ensuite, on divise p_1 par p_2 et en multipliant le reste par -1, on obtient $p_3(x) = -32x - 64$. Puis on divise p_2 par p_3 et en multipliant le reste par -1, on obtient $p_4(x) = -\frac{3}{16}$.

Finalement, la suite de Sturm du polynôme P est donc :

$p_0(x) = x^4 + x^3 - x - 1$

$p_1(x) = 4x^3 + 3x^2 - 1$

$p_2(x) = \frac{3}{16}x^2 + 34x + \frac{15}{16}$

$p_3(x) = -32x - 64$ $p_4(x) = -\frac{3}{16}$

Pour trouver le nombre de racines totales, c'est à dire entre $-\infty$ et $+\infty$, on évalue p_0, p_1, p_2, p_3, et p_4 en $-\infty$ et on note la séquence de signes correspondante : $+ - + + -$. Elle contient trois changements de signe ($+$ à $-$, puis $-$ à $+$, puis $+$ à $-$).

On fait la même chose en $+\infty$ et obtient la séquence de signes $+++--$, qui contient juste un changement de signe. D'après le théorème de Sturm, le nombre total de racines du polynôme P est $3-1=2$. Nous pouvons faire une vérification en remarquant que $p(x)=x^4+x^3-x-1$ se factorise en$(x^2-1)(x^2+x+1)$, où on voit que x^2-1 a deux racines (-1 et 1) alors que x^2+x+1 n'a pas de racines réelles. ■

■ **Exemple 2.2** Soit $P_3(x)=x^3+2x^2-x-2$. Alors ■

$p_1(x)=x^3+2x^2-x-2$
$p_2(x)=3x^2+4x-1$
$p_3(x)=7x+8$ (à un facteur multiplicatif positif près)
$p_4(x)=81$ (à un facteur multiplicatif positif près)

Qu'on peut mettre sous forme du tableau suivant :

y	$p_1(y)$	$p_2(y)$	$p_3(y)$	$p_4(y)$	$N(y)$
-3	$-$	$+$	$-$	$+$	3
0	$-$	$-$	$+$	$+$	1
3	$+$	$+$	$+$	$+$	0

Il y a donc $3-1=2$ racines réelles distinctes dans $[-3,0]$
et $3-0=3$ racines réelles distinctes dans $[-3,3]$

■ **Exemple 2.3** Soit $P_6(x)=x^6+4x^5+4x^4-x^2-4x-4$. Alors ■

$p_1(x)=x^6+4x^5+4x^4-x^2-4x-4$.
$p_2(x)=6x^5+20x^4+16x^3-2x-4$
$p_3(x)=4x^4+8x^3+3x^2+14x+16$
$p_4(x)=x^3+6x^2+12x+8$
$p_5(x)=-17x^2-58x-48$
$p_6(x)=-x-2$
(Tous ces polynôme sont définis à un facteur multiplicatif positif près)
Qu'on peut mettre sous forme du tableau suivant :

y	$p_1(y)$	$p_2(y)$	$p_3(y)$	$p_4(y)$	$p_5(y)$	$p_6(y)$	$N(y)$
-3	$+$	$-$	$+$	$-$	$-$	$+$	4
-2	0	0	0	0	0	0	
-1	0	$-$	$+$	$+$	$-$	$-$	2
0	$-$	$-$	$+$	$+$	$-$	$-$	2
1	0	$+$	$+$	$+$	$-$	$-$	1
2	$+$	$+$	$+$	$+$	$-$	$-$	1
3	$+$	$+$	$+$	$+$	$-$	$-$	1

Il y a donc $3-1=2$ racines réelles négatives distinctes dans $[-3,0]$, et une racine réelle distinctes dans $[0,3]$. Comme $p_6(x)=-x-2$ la valeur -2 est une racine double.

2.8 RÈSOLUTION DE SYSTÈMES NON LINÉAIRES

Nous allons maintenant résoudre le système de m équations à m inconnues $x_1,x_2,....,x_m$ de la forme :

$$\begin{cases} f_1(x_1,x_2,.....,x_m) &= 0 \\ f_2(x_1,x_2,.....,x_m) &= 0 \\ ..... &= 0 \\ f_m(x_1,x_2,.....,x_m) &= 0 \end{cases} \tag{2.14}$$

Définition 2.8.1 Le système (2.14) peut toujours se mettre sous la forme :

$$F(x) = x\ ,\ \ x \in \mathbb{R}^m\ , F : \mathbb{R}^m \longmapsto \mathbb{R}^m$$

x est dit point fixe de $F(x)$.

Le théorème suivant assure l'existence et l'unicité d'un point fixe. Il nous permettra d'approcher les solutions de systèmes algébriques non linéaires.

Théoreme 2.8.1 — du point fixe. Soit E un espace métrique complet non vide, $F : E \longmapsto E$ une contraction stricte. Alors F admet un point fixe et un seul donné par la méthode des approximations successives :

$$x_{n+1} = F(x_n)$$

pour x_0 quelconque.

Démonstration. **1- Existence :** Soit la suite $(x_n) \in E$ définie par $x_{n+1} = F(x_n)$. Nous allons montrer que la suite (x_n) est de Cauchy.
Comme F est une contraction, nous avons :

$$\begin{aligned} d(x_2,x_1) &\leq kd(x_1,x_0) \\ d(x_3,x_2) \leq kd(x_2,x_1) &\leq k^2 d(x_1,x_0) \\ .....\cdots &\leq ..... \\ d(x_{n+1},x_n) \leq kd(x_n,x_{n-1}) &\leq k^n d(x_1,x_0) \end{aligned}$$

Ce qui nous donne

$$\begin{aligned} d(x_{n+p},x_n) &\leq d(x_{n+p},x_{n+p-1}) + d(x_{n+p-1},x_{n+p-2}) + \cdots + d(x_{n+1},x_n) \\ d(x_{n+p},x_n) &\leq k^n(k^{p-1} + k^{p-2} + k^{p-3}\cdots + k + 1)d(x_1,x_0) \leq \frac{k^n}{1-k} d(x_1,x_0) \end{aligned}$$

Nous en déduisons que $d(x_{n+p},x_n) \xrightarrow{0}$ quand $n \xrightarrow{+} \infty$. Donc la suite (x_n) est fde Cauchy et par suite admet une limite x et comme F est continue, $F(x_n)) \xrightarrow{x}$. On a donc $F(x) = x$.

2- Unicité Si x et y sont deux points fixes, on doit avoir

$$d(x,y) \leq kd(x,y) < d(x,y)$$

si $d(x,y) \neq 0$.
On a donc nécessairement $d(x,y) = 0$ et $x = y$.

■

Proposition 2.8.2 Soit $F : E \longmapsto E$. Posons $F_2 = FoF$, $F_3 = FoFoF$, ...,$F_p = FoF_{p-1}$; F_p est appelée l'iterée d'ordre p de F. Nous avons alors le résultat suivant : Si l'une des iterées F_p est strictement contractante, alors F admet un point fixe unique.

R

1. Le procédé $x_{n+1} = F(x_n)$ est un algorithme permettant de trouver le point fixe de F. De plus la suite (x_n) converge rapidement vers x, car

$$d(x_n, x) = \lim_{p \longmapsto +\infty} d(x_n, x_{n+p}) \leq \frac{k^n}{1-k} d(x_1, x_0)$$

2. L'itérée F_p contraction stricte n'implique pas nécessairement que F soit continue ou contractante.
3. La condition $k < 1$ de contraction stricte, est indispensable. Car $k \leq 1$ ne suffit pas pour garantir ni l'existence ni l'unicité du point fixe.

2.8.1 Méthode des approximations successives (type Jacobi ou Gauss-Seidel)

L'équation

$$x = F(x); \quad x \in \mathbb{R}^m, \quad F : \mathbb{R}^m \longmapsto \mathbb{R}^m$$

peut s'écrire sous la forme :

$$A(x) = b; \quad x \in \mathbb{R}^m; \quad b \in \mathbb{R}^m; \quad A : \mathbb{R}^m \longmapsto \mathbb{R}^m$$

ou bien sous la forme

$$x = B(x) + c; \quad c \in \mathbb{R}^m; \quad B : \mathbb{R}^m \longmapsto \mathbb{R}^m$$

ou sous la forme d'un système de m équations :

$$x_i = B_i(x) + c_i; \quad i = 1, 2, ..., m$$

— S'il existe un domaine Ω convexe contenant x solution de $x = F(x)$ tel que

$$\forall x \in \Omega, \quad \sum_{j=1}^{m} |\frac{\partial B_i}{\partial x_i}| \leq d < 1; \quad i = 1, 2...., m$$

alors pour tout vecteur initial $x^{(0)}$ pris dans Ω, la suite de vecteurs $x^{(k)}$ définis par le schema itératif

$$x^{(k+1)} = B(x^{(k)}) + c \tag{2.15}$$

converge vers x d'après le théorème du point fixe. Le schema (2.15) s'appelle *"Methode de Jacobi non lineaire"*.

— Sous les mêmes hypothèses, la suite de vecteurs $x^{(k)}$ définis par le schema itératif :

$$x_i^{(k+1)} = B_i(x_1^{(k+1)}, ..., x_{i-1}^{(k+1)}, x_i^{(k)}, ..., x_m^{(k)}) + c_i; \quad i = 1, 2, ..., m \tag{2.16}$$

converge vers x, d'après le théorème du point fixe.
Le schema (2.16) s'appelle *"Methode de Gauss-Seidel non lineaire"*.

3. RÉSOLUTION NUMÉRIQUE des E.D.O. d'ORDRE UN

3.1 Introduction

Les équations différentielles ordinaires ou E.D.O. sont utilisées pour modéliser un grand nombre de phénomènes mécaniques, physiques, chimiques, biologiques etc...

Soit f une fonction continue

$$f: \quad [a,b]\times\mathbb{R}^{\aleph} \to \mathbb{R}^{\aleph} \tag{3.1}$$
$$(t,y) \mapsto f(t,y) \tag{3.2}$$

telle que :

$$f = (f_1, f_2, ..., f_n)$$

et

$$f_i: \quad [a,b]\times\mathbb{R}^{\aleph} \to \mathbb{R}$$
$$(t,y) \mapsto f_i(t,y)$$

Définition 3.1.1 **1- E.D.O. du premier ordre :** On appelle équation différentielle ordinaire du premier d'ordre une équation de la forme :

$$y\prime(t) = f(t,y(t)) \qquad t \in [a,b] \tag{3.3}$$

2- E.D.O. d'ordre p : On appelle équation différentielle d'ordre p une équation de la forme :

$$y^{(p)}(t) = f(t,y(t),y\prime(t),y^{(2)}(t),.....,...,y^{(p-1)}(t)) \qquad t \in [a,b] \tag{3.4}$$

f est une fonction continue donnée

$$f: \quad [a,b]\times(R^n)^p \to \mathbb{R}^{\aleph}$$
$$(x,y) \mapsto f(x,y)$$

1. Une fonction y de classe C^1 vérifiant l'équation (3.3) est dite solution de l'équation différentielle du premier ordre.
2. Une fonction y de classe C^p vérifiant l'équation (3.4) est dite solution de l'équation différentielle d'ordre p.

Proposition 3.1.1 Toute équation différentielle d'ordre n sous forme canonique peut s'écrire comme un système de n équations différentielles du premier ordre.

R L'équation (3.3) est donc équivalente au système suivant :

$$\begin{cases} y\prime_1(t) &= f_1(t,y_1,.....,y_n) \\ ..... & \\ y\prime_n(t) &= f_n(t,y_1,.....,y_n) \end{cases} \tag{3.5}$$

Cela se fait en posant

$$\begin{cases} z_1 & = & y \\ z_2 & = & y\prime \\ \cdots\cdots & \cdots & \cdots \\ z_p & = & y^{p-1} \end{cases} \tag{3.6}$$

où $z_1, z_2, \ldots\ldots, z_p$ sont des fonctions de classe C^1 et l'équation différentielle d'ordre p (3.4) est est équivalente au système :

$$\begin{cases} z\prime_1 & = & y \\ z\prime_2 & = & y\prime \\ \cdots\cdots & \cdots & \cdots \\ z\prime_p & = & f(t, z_1, \ldots\ldots, z_p) \end{cases} \tag{3.7}$$

qui s'écrit aussi sous la forme

$$z\prime(t) = g(t, z(t))$$

Avec

$$\begin{aligned} g : [a,b] \times (R^n)^p &\to (R^n)^p \\ (t,y) &\mapsto f(t,y) \end{aligned}$$

Ce qui veut dire que l'étude d'une équation différentielle d'ordre p dans $\mathbb{R}^{\ltimes}$ est ramenée à une équation différentielle d'ordre 1 dans $\mathbb{R}^{\ltimes p}$.

Toute équation différentielle d'ordre p sous forme canonique peut s'écrire comme un système de p équations différentielles du premier ordre.

3.2 PROBLEME DE CAUCHY

Définition 3.2.1 On appelle problème de Cauchy, le problème qui consiste en la recherche d'un fonction y de classe C^1 vérifiant

$$\begin{cases} y\prime(t) = & f(t, y(t)) \\ y(a) = & y(0), \quad y_0 \quad \text{donné dans} \quad \mathbb{R}^{\ltimes} \end{cases} \tag{3.8}$$

Soit f une fonction continue

$$\begin{aligned} f : \quad [a,b] \times \mathbb{R}^{\ltimes} &\to \mathbb{R}^{\ltimes} \\ (t,y) &\mapsto f(t,y) \end{aligned}$$

Théoreme 3.2.1 Soit le problème de Cauchy (3.8). Si f vérifie en plus de la continuité, la condition de Lipchitz, c'est-à-dire

$$||f(t,y) - f(t,y^*)|| \leq k||y - y^*||; k > 0. \quad \forall t \in [a,b]; \forall y, y^* \in \mathbb{R}^{\ltimes} \tag{3.9}$$

alors le problème de Cauchy (3.8) admet une et une solution de classe C^1.

De très nombreux résultats mathématiques existent sur les problèmes de Cauchy.

Dans ce qui suit nous allons nous intéresser à certaines méthodes numériques de résolution de ce type de problème.

L'ensemble des méthodes numériques que nous allons étudier auront pour but la résolution d'un problème de Cauchy quelconque. Elles pourront donc être utilisées pour la résolution d'une très grande variété d'E.D.O.

Deux questions se posent, dans la résolution de ce type de problème (Cauchy).

1. Trouver une approximation numérique de la solution.
2. Majorer l'erreur commise à partir de cette approximation.

3.3 MÉTHODE de TAYLOR d'ORDRE 2

Pour résoudre numériquement le problème de Cauchy (3.8), On ecrit :

$$y(t) = y_0 + \frac{(t-a)}{1!}y\prime(a) + \frac{(t-a)^2}{2!}y\prime\prime(a) + \dots$$

Avec y_0 donnée

$$\begin{aligned} y\prime(a) &= f(a,y_0) = y\prime_0 \\ y\prime\prime(a) &= f(a,y_0) = y\prime_0 + \frac{\delta f}{\delta t}(a,y_0) + y\prime_0\frac{\delta f}{\delta y}(a,y_0) \\ \dots \quad \dots &\quad \dots \end{aligned}$$

Les formules de dérivation se compliquent très vite, et il est très souvent impossible d'avoir une idée sur le rayon de convergence de cette série (de Taylor).

Cette méthode est en général utilisée localement au voisinage du point $t_0 = a$.

3.4 MÉTHODES NUMERIQUES PAR PAS

Dans ce genre de méthodes, on va subdiviser l'intervalle $[a,b]$ par des points $t_1, t_2, \dots\dots, t_N$ équidistants : $t_{n+1} = t_n + h$. avec $h = \frac{b-a}{N}$ le pas de la subdivision.

On calcule N nombres $y_1, y_2, \dots\dots N$ ayant une valeur proche de celle de la solution y aux points $t_n = a_n + h$; $n = 0, \dots\dots, N$.

Ensuite, on fait une interpolation pour relier ces points et définir une fonction y_h sur $[a,b]$.

L'erreur de discretisation dépendante de h est estimée par la formule ;

$$e_n = y_n - y(t_n)$$

.

On distingue deux types d'algorithmes par :

1. Les algorithmes à pas séparés ou méthodes à un pas qui permettent de calculer y_{i+1} à partir de y_i.
2. Les algorithmes à pas liés ou méthodes à pas multiples qui permettent de calculer y_{i+1} à partir des $y_i, y_{i-1}, \dots$ précédents.

3.5 MÉTHODE d'EULER-CAUCHY

Cette méthode étant la plus simple des méthodes numériques par pas.

En partant du développement de Taylor on a :

$$y(t_{n+1}) = y(t_n) + hy\prime(t_n) + R$$

D'où :

$$\frac{y(t_{n+1}) - y(t_n)}{h} = y\prime(t_n) + \frac{R}{h}$$

Si $\frac{R}{h}$ est suffisamment petit, on peut considérer que

$$\frac{y(t_{n+1})-y(t_n)}{h}$$

est une bonne approximation de $y\prime(t_n)$ d'où **l'algorithme de Euler-Cauchy**.

$$\begin{cases} y_{n+1} = y_n + hf(t_n,y_n); \quad n=0,1....N \\ y(0) = y(a) \end{cases} \tag{3.10}$$

On peut interpréter cet algorithme comme :

connaissant y_n, on calcule y_{n+1} comme étant l'ordonnée du point d'intersection de la droite $t=t_{n+1}$ avec la droite passant par le point (t_n,y_n) ayant pour pente $f(t_n,y_n)$ c'est-à -dire la pente de la tangente en (t_n,y_n) à la courbe solution.

Théoreme 3.5.1 Si la fonction f vérifie les hypothèses suivantes :

1- f est continue

2- f est lipchitzienne de rapport $K>0$.

La methode de Euler-Cauchy (3.10) converge.

De plus on a une estimation de l'erreur sous la forme :

$$|e_n| = |y_n - y(t_n)| \leq \frac{e^{K(t_n-t_0)}-1}{K} M(h,y\prime)$$

et

$$\max_{n=1,\ldots,N} |e_n| \to 0 \text{quand} h \to 0$$

3.5.1 Estimation de l'erreur dans la méthode d'Euler-Cauchy

On va chercher une majoration de l'erreur qui ne dépendra que des données.

Soit la proposition :

Proposition 3.5.2 Soit

$$c = \sup_{t\in[a,b]} |f(t,0)|.$$

Alors

$$||y_h|| \leq |y_0| e^{K(b-a)} + c\frac{e^{K(b-a)}-1}{K} = D$$

et

$$||y_h|| \leq D$$

Théoreme 3.5.3 On pose

$$M_1 = \sup_{t\in[a,b]} |f(t,y)|$$

et

$$M_D(\delta,f) = \sup_{t,t'\in[a,b]} |f(t,y)-f(t',y)|$$

avec

$$||y|| \leq D$$

et

$$t,t' \in [a,b]$$

Alors

$$|e_n| \leq (M_D(h,f) + hKM_1)\frac{e^{K(t_n-t_0)}-1}{K}$$

La majoration de l'erreur donnée par ce théorème en fonction seulement des données est difficile a calculer numériquement.

On peut simplifier cette estimation en ajoutant une hypothèse supplémentaire.

Théoreme 3.5.4 Soit Ω le domaine défini par :

$$\Omega = \{(t,y) \in \mathbb{R}^{\not\vdash} | t \in [a,b], |y| \leq D\}$$

avec

$$D = |y_0|e^{K(b-a)} + c\frac{e^{K(b-a)}-1}{K}$$

On suppose :

1. f continue de $[a,b] \times \mathbb{R} \to \mathbb{R}$
2. f lipchitzienne en y
3. f de classe C^1 sur Ω

et on pose :

$$N(t) = \frac{1}{2}\max_{t\in[a,b]} |y"(t)|.$$

Alors

$$|e_n| \leq hN(t_n)\frac{e^{K(t_n-a)}-1}{K}$$

pour $n = 0, 1, ..., N$

3.6 MÉTHODE DE RUNGE-KUTTA

Les algorithmes de Runge-Kutta (RK) consistent à calculer à chaque pas des valeurs intermédiaires.

La méthode (RK) classique est donnée par le schéma suivant :

$$\begin{cases} y_1 = y_n + hF(t_n, y_n; h) \\ y_0 = \eta \\ \text{avec} \quad F(t,y;h) = \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ \text{où} \quad k_1 = f(t,y) \qquad k_2 = f(t+\frac{h}{2}, y+\frac{h}{2}k_1) \\ k_3 = f(t+\frac{h}{2}, y+h\frac{k_2}{2}); \qquad k_4 = f(t+h, y+hk_3) \end{cases} \tag{3.11}$$

R

1. Les méthodes (RK) sont convergentes.
2. Elles ne nécessitent pas le calcul des dérivées successives de f.
3. Elles donnent de très bons résultats notamment pour la résolution des problèmes de Cauchy.

3.7 SERIE D'EXERCICES

Exercice 3.1 On considère l'équation différentielle

$$\begin{cases} y' & = & 2y \\ y(0) & = 5 \end{cases}$$

1. Vérifier que la solution exacte est $y(t) = 5e^{2t}$.
2. Soit $h = \frac{1}{n}$, pour $i = 0, \ldots, n$, montrer que les approximations fournies par le schéma d'Euler peuvent s'écrire $y_i = 5(1+2h)^i$.
3. Représenter graphiquement l'erreur

$$e(h) = \max_{0 \le i \le n} |y(t_i) - y_i|.$$

en fonction de h, en calculant $e(0.005), e(0.01), e(0.05), e(0.1), e(0.5)$.
4. Que pensez-vous de la relation $e(h) \approx Kh$, avec K une constante ?

■

Exercice 3.2 On considere le probleme d'equation differentielle

$$\begin{cases} y' & = & -11y \\ y(0) & = 2 \end{cases}$$

Pour résoudre cette équation numériquement sur l'intervalle $[0,1]$, on se donne, pour chaque entier n, un pas $h = \frac{1}{n}$ et des noeuds $x_i = ih, i = 1, \ldots, n$.

1. En répétant le raisonnement de l'exercice précédent, on pourrait montrer que, l'application de la méthode d'Euler conduit aux approximations :

$$y_i = 2(1-11h)^i.$$

2. Représenter les approximations obtenues pour pour $h = 0.2, 0.1, 0.09, 0.1$.
Indication : la solution de ce problème est de la forme $y = Ae^{at}$, où A et a sont faciles à calculer.

■

Exercice 3.3 On considere le problème d'équations différentielles

$$y' = 2t - 3y, \quad y(0) = 1.$$

1. Montrer que la solution exacte est donnée par :

$$y = -\frac{2}{9} + \frac{2}{3}t + \frac{11}{9}e^{-3t}.$$

2. Vérifier que les approximations obtenues en prenant $h = 0.25$ et la méthode de Taylor d'ordre 2 ou la méthode d'Euler modifiée, sont égales.
3. Ecrire la formule aux différentes $y_{i+1} = y_i + h\phi(t_i, y_i)$, obtenues par chacune des deux approches. Expliquer pourquoi, dans ce cas particulier, les deux formules coıncident.

■

Exercice 3.4 L'égalité suivante découle directement du theoreme fondamental du calcul

$$y(t+h) = y(t) + \int_t^{t+h} y\prime(t)dt.$$

En appliquant la formule de Simpson à l'integrale, on obtient alors l'approximation

$$y(t+h) \approx y(t) + \frac{h}{6}(y\prime(t) + 4y\prime(t+\frac{h}{2}) + y\prime(t+h)).$$

Supposons que y soit solution de $y\prime = f(t,y)$, l'approximation précédente s'écrit

$$y(t+h) \approx y(t) + \frac{h}{6}(f(t,y(t)) + 4f(t+\frac{h}{2}) + f(t+h,y(t+h))).$$

a) En remplaçant $y(t+\frac{h}{2})$ et $y(t+h)$ par les approximations données par la formule d'Euler modifiée, déduire de l'équation précédente un schéma numérique à un pas du type

$$y_{i+1} = y_i + \frac{h}{6}(f(t_i,y_i) + 2\frac{h}{3}f(t_i+\frac{h}{2},y_i+k_1) + \frac{h}{6}f(t_i+h,y_i+k_2)))$$

pour lequel on déterminera les coefficients k_1, k_2 en fonction de h , y_i et $f(t_i,y_i)$.

b) On considère le cas particulier

$$f(t,y) = -y+t+1$$

, pour lequel la solution exacte est

$$y(t) = t + e^{-t}.$$

En vous inspirant des exemples donnés et en choisissant $t_0 = 0, x_n = 1, y_0 = 1$, calculer

$$e(h) = \max|y_i - y(t_i)||i = 1,...npourn = 2,4,8,16$$

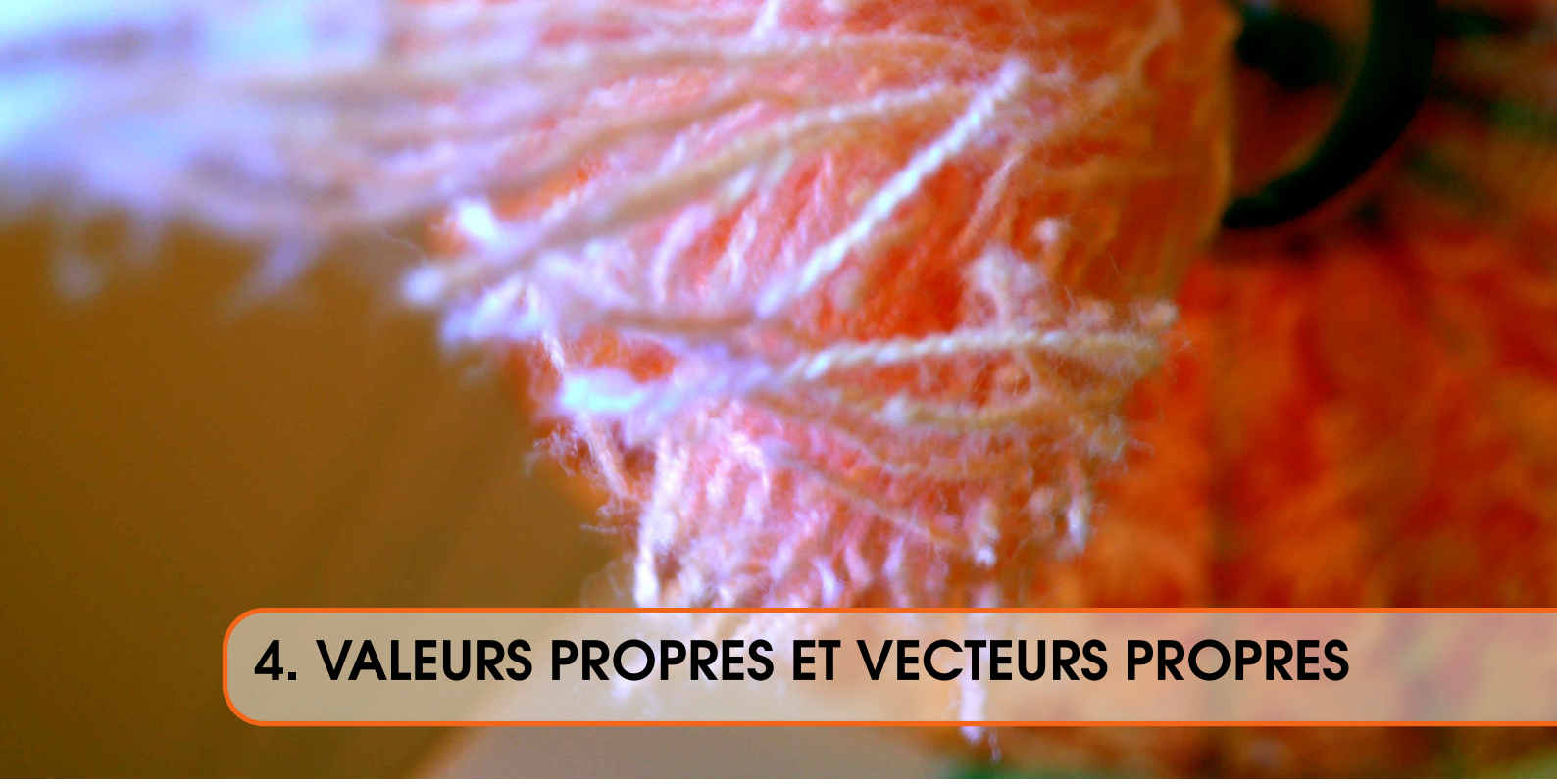
. Reporter cette fonction sur un graphe log – log et en déduire l'ordre de la méthode. ■

Exercice 3.5 On considère le problème

$$\begin{cases} y\prime & = & \frac{-3y}{t^2} \\ y(1) & = 2e^3 & \end{cases}$$

Comparer les approximations de la solution obtenues par la méthode d'Euler avec $h = 0.0016$ par la méthode du point milieu avec $h = 0.04$ et par la méthode de Runge-Kutta 4 avec $h = 0.2$. La comparaison se fera sur la base de la précision et du cout de calcul, i.e. le nombre de fois qu'il faut évaluer $f(t,y)$.

■



4. VALEURS PROPRES ET VECTEURS PROPRES

4.1 INTRODUCTION

De nombreuses méthodes numériques supposent la connaissance des valeurs propres, des vecteurs propres et du rayon spectral d'une matrice. En outre de nombreux problèmes se ramènent à la recherche des valeurs propres d'une matrice. Le plus souvent, on fait appel à deux types de méthodes numériques pour calculer les valeurs propres et les vecteurs propres (appelés aussi éléments propres) d'une matrice. *Les méthodes directes* sont celles qui permettent d'obtenir les éléments propres à partir de la connaissance explicite du polynôme caractéristique ; les autres sont essentiellement des *méthodes itératives.* Ces dénominations présentent une certaine ambiguité. En effet, le plus souvent, la détermination du polynôme caractéristique est obtenue par un procédé itératif et la recherche des racines de ce polynôme est presque toujours itérative.

Soit $A \in \mathscr{M}_n(\mathbb{C})$. Nous allons chercher ses valeurs propres λ_i dont la multiplicité sera notée m_i.

4.2 RAPPELS

Les premiers vecteurs et valeurs propres viennent des équations différentielles des matrices 6×6 dans le but de calculer les perturbations séculaires des orbites des 6 planètes connues à l'époque,

Aujourd'hui, le calcul des valeurs et vecteurs propres est indispensable dans toutes les branches de la science, en particulier pour la solution des systèmes des équations diffé rentielles linéaires, en théorie de stabilité etc...

Définition 4.2.1 Le champ de vecteurs d'une équation différentielle $y' = Ay$, présente deux directions remarquables : ce sont les directions où le vecteur Av prend la même direction que le vecteur v , c'est-à-dire., où

$$Av = \lambda v \quad \text{ou} \quad (A - \lambda I)v = 0 \tag{4.1}$$

Si cette équation est vérifiée, $\lambda \in \mathbb{C}$ s'appelle *valeur propre* de la matrice A et $v \in \mathbb{C}^n (v \neq 0)$ est le *vecteur propre* correspondant.

R L'équation (4.1) possède une solution v non nulle si et seulement si

$$P_A(\lambda) = \det((A - \lambda I) = 0$$

Le polynôme $P_A(\lambda)$ est le *polynôme caractéristique* de la matrice A . Les valeurs propres de A sont alors les z éros du polynôme caractéristique.

4.3 LA CONDITION DU CALCUL DES VALEURS PROPRES

A cause des erreurs d'arrondi, les éléments d'une matrice A , pour laquelle on cherche les valeurs propres, ne sont pas exacts. Ils sont plutôt égaux à

$$\tilde{a}_{ij} = a_{ij}(1 + \varepsilon_{ij}) \quad \text{avec} \quad |\varepsilon_{ij}| \leq eps$$

(eps étant la précision de l'ordinateur, est supposée être très petite). Il est alors très important d'étudier l'influence de ces perturbations sur les valeurs propres et sur les vecteurs propres de la matrice. Pour montrer ceci, considérons la famille de matrices

$$A(\varepsilon) = A + \varepsilon C \quad \text{où} \quad |\varepsilon| \leq eps \quad \text{et} \quad |c_{ij}| \leq |a_{ij}|$$

(souvent, la dernière hypothèse va être remplacée par $\|C\| \leq \|A\|$).

Théoreme 4.3.1 — Gershgorin. Soit A une matrice $n \times n$ (avec des éléments dans $\mathbb{R}$ ou dans $\mathbb{C}$).

a) Si λ est une valeur propre de A, alors il existe un indice i tel que

$$|\lambda - a_{ii}| \leq \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}|$$

c'est-à-dire. que toutes les valeurs propres de A se trouvent dans l'union des disques $D_i = \left\{ \lambda ; \ |\lambda - a_{ii}| \leq \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}| \right\}$

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

Démonstration. Soit $v \neq 0$ un vecteur propre et choisissons l'indice i tel que $|v_i| \geq |v_j|$ pour tout j . La ligne i de l'équation $Av = \lambda v$ donne

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

En divisant par v_i et en utilisant l'inégalité du triangle, on obtient U

$$|\lambda - a_{ii}| = \left| \sum_{\substack{j=1 \\ j \neq i}}^{n} a_{ij} \frac{v_j}{v_i} \right| \leq \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}|$$

L'affirmation (b) est vraie si A est une matrice diagonale. Le cas gén éral est obtenu par un argument de continuité en faisant tendre les éléments en dehors de la diagonale vers zéro. ■

Théoreme 4.3.2 Soit A une matrice diagonalisable, c'est-à-dire, il existe P avec $P^{-1}AP = diag(\lambda_1, \lambda_2, ..., \lambda_n)$ et soit $A(\varepsilon) = A + \varepsilon C$. Alors, pour chaque valeur propre $\lambda(\varepsilon)$ de $A(\varepsilon)$, il existe un λ_i avec

$$|\lambda(\varepsilon) - \lambda_i| \leq \varepsilon . \kappa_\infty(P) . \|C\|_\infty$$

Démonstration. Nous transformons la matrice$A(\varepsilon) = A + \varepsilon C$ par la mê me matrice, qui transforme A sous forme diagonale :

$$P^{-1}A(\varepsilon)P = diag(\lambda_1, \lambda_2, ..., \lambda_n) + \varepsilon P^{-1}CP$$

Si l'on dénote par e_{ij} les éléments de $P^{-1}CP$, le th éorème de Gershgorin implique l'existence d'un indice i tel que $|\lambda(\varepsilon) - (\lambda_i + \varepsilon e_{ii})| \leq \varepsilon \sum_{j \neq i} |e_{ij}|$. L'inégalité triangulaire donne alors

$$|\lambda(\varepsilon) - \lambda_i| \leq \varepsilon . \max_i (\sum_j |e_{ij}|) \leq \varepsilon . \left\|P^{-1}CP\right\|_\infty \leq \varepsilon . \left\|P^{-1}\right\|_\infty . \|C\|_\infty . \|P\|_\infty$$

ce qui démontre l'affirmation du théorème, car $\kappa_\infty(P) = \left\|P^{-1}\right\|_\infty . \|C\|_\infty$ (condition de T). ■

R La condition du calcul des valeurs propres depend de la condition de la matrice de transformation P. Si la matrice A est symétrique (P est orthogonale), le problème est bien conditionné.Toutefois, observons qu'on obtient seulement une estimation pour l'erreur absolue et non pour l'erreur relative.

Théoreme 4.3.3 — différentiabilité des valeurs propres. Soit λ_1 une racine simple de $P_A(\lambda) = 0$ Alors, pour $|\varepsilon|$ suffisamment petit, la matrice $A(\varepsilon) = A + \varepsilon C$ possède une valeur propre unique $\lambda_1(\varepsilon)$ proche de λ_1 . La fonction $\lambda_1(\varepsilon)$ est différentiable (même analytique) et on a

$$\lambda_1(\varepsilon) = \lambda_1 + \varepsilon \frac{u_1^* C v_1}{u_1^* v_1} + O(\varepsilon^2) \tag{4.2}$$

où v_1 est le vecteur propre à droite ($Av_1 = \lambda v_1$) et u_1 est le vecteur propre à gauche ($u_1^* A = \lambda u_1^*$). On peut supposer que $\|v_1\| = \|u_1\| = 1$

Démonstration. Soit $p(\lambda, \varepsilon) = P_{A+\varepsilon C}(\lambda) = \det(A + \varepsilon C - \lambda I)$. Comme

$$p(\lambda_1, 0) = 0 \quad \text{et} \quad \frac{\partial p(\lambda_1, 0)}{\partial \lambda} \neq 0$$

le théorème des fonctions implicites garantit l'existence d'une fonction différentiable $\lambda_1(\varepsilon)$ (même analytique), tel que $\lambda_1(0) = \lambda_1$ et $p(\lambda_1(\varepsilon), \varepsilon) = 0$. Il existe donc un vecteur $v_1(\varepsilon)$ tel que

$$(A(\varepsilon) - \lambda_1(\varepsilon)I)v_1(\varepsilon) = 0. \tag{4.3}$$

La matrice dans (4.3) étant de rang $n-1$, on peut fixer une composante à 1 et appliquer la règle de Cramer. Ceci montre que les autres composantes sont des fonctions rationnelles des éléments de la matrice $A + \varepsilon C - \lambda_1(\varepsilon)I$ et donc diff érentiables. Après la normalisation à $v_1(\lambda)^T v_1(\lambda) = 1$, la fonction $v_1(\lambda)$ reste diffé rentiable. Pour calculer $\lambda_1'(0)$, nous pouvons dériver l' équation (4.3) par rapport à ε et poser ensuite $\varepsilon = 0$. Ceci donne

$$(A - \lambda_1 I)v_1'(0) + (C - \lambda_1'(0)I)v_1 = 0 \tag{4.4}$$

En multipliant cette relation par u_1^*, on obtient $u_1^*(C - \lambda_1'(0)I)v_1 = 0$, ce qui permet de calculer $\lambda_1'(0)$ et démontre la formule (4.2). ■

Conséquences. La formule (4.2) du théorème pré cédent montre que plus le vecteur propre de droite est parallèle au vecteur propre de gauche, mieux la valeur propre correspondante est bien conditionnée (par exemple, pour les matrices symétriques les deux vecteurs sont identiques) ; plus ils se rapprochent de l'orthogonalité, plus la valeur propre est mal conditionnée. Si la matrice n'est pas symétrique (ou normale), le calcul de λ_1 (valeur propre simple) peut être mal conditionné. Considé rons par exemple la matrice

$$A = \begin{pmatrix} 1 & \alpha \\ 0 & 2 \end{pmatrix} \quad \text{où} \quad v_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad u_1 = \frac{1}{\sqrt{1+\alpha^2}} \begin{pmatrix} 1 \\ -\alpha \end{pmatrix}$$

Dans cette situation, la formule (4.2) nous donne $\lambda_1(\varepsilon) - \lambda_1 = \varepsilon.(c_{11} - \alpha c_{21}) + O(\varepsilon^2)$ et le calcul de$\lambda_1 = 1$ est mal conditionné si α est grand. Exemple 1.4 Considérons la matrice (boîte de Jordan)

$$A = \begin{pmatrix} \lambda_1 & 1 & & \\ & \lambda_1 & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_1 \end{pmatrix} \Bigg\} n \tag{4.5}$$

Le polynôme caractéristique de $A + \varepsilon C$ satisfait

$$\det(A + \varepsilon C - \lambda I) = (\lambda_1 - \lambda)^n - (-1)^n .\varepsilon.c_{n1} + O(\varepsilon^2) + O(\varepsilon.|\lambda_1 - \lambda|).$$

Si $c_{n1} \neq 0$, les termes $O(\varepsilon^2)$ et $O(\varepsilon.|\lambda_1 - \lambda|)$ sont négligeables par rapport à ε . Les valeurs propres de $A + \varepsilon C$ sont alors approximativement données par les racines de

$$(\lambda_1 - \lambda)^n - (-1)^n .\varepsilon.c_{n1} = 0 \tag{4.6}$$

c'est-à-dire $\lambda = \lambda_1 + (\varepsilon.c_{n1})^{1/n}$ (observer que $(\varepsilon.c_{n1})^{1/n}$ donne n valeurs complexes distinctes - multiples des racines de l'unité). **Expérience numérique**. Prenons la matrice (4.5) avec $\lambda_1 = 1$ et $n = 5$. Les éléments de la matrice C sont des nombres aléatoires dans l'intervalle $[-1, 1]$. Le dessin 7 ci-contre montre les 5 valeurs propres de $A + \varepsilon C$ pour $\varepsilon = 10^{-4}, 10^{-5}, ..., 10^{-10}$. L'erreur est $\approx 10^{-1}$ pour $\varepsilon = 10^{-5}$ et $\approx 10^{-2}$ pour $\varepsilon = 10^{-10}$, ce qui correspond à la formule (4.6) pour $n = 5$. **Conséquence**. Si la dimension n d'une boite de Jordan est plus grande que 1 , le calcul de la valeur propre de cette matrice est tr ès mal conditionné.

4.3.1 Condition du calcul des vecteurs propres

Considérons la situation où toutes les valeurs propres de A sont distinctes. La démonstration du théorème sur la diffé rentiabilité des valeurs propres montre (voir formule (4.3)) que les vecteurs propres normalisés $v_i(\varepsilon)$ de $A + \varepsilon C$ sont des fonctions différentiables de ε . Pour é tudier la condition du calcul des vecteurs propres, nous exprimons $v_1'(0)$ dans la base des vecteurs propres (de droite) '

$$v_1'(0) = \sum_{i=1}^{n} \alpha_i v_i. \tag{4.7}$$

La formule (4.4) donne alors

$$\sum_{j=2}^{n} (\lambda_j - \lambda_1)\alpha_j v_j + \left(C - \lambda_1'(0) I\right) v_1 = 0. \tag{4.8}$$

En multipliant (4.8) par le vecteur propre de gauche u_1^* (observer que $u_1^* v_1 = 0$ pour $i \neq j$), on obtient α_i (pour $i \geq 2$) de la relation $(\lambda_i - \lambda_1)\alpha_i u_i^* v_i + u_i^* C v_1 = 0$. La normalisation $\|v_1(\varepsilon)\|_2^2 = 1$

donne (en la dérivant) $v_1^* v_1'(0) = 0$ et on en déduit que $\alpha_1 = -\sum_{j=2}^n \alpha_i v_1^* v_i$. Si l'on insère les formules pour α_i dans (4.7), on obtient pour $v_1(\varepsilon) = v_1 + \varepsilon v_1'(0) + O(\varepsilon^2)$ la relation

$$v_1(\varepsilon) = v_1 + \varepsilon \sum_{j=2}^{n} \frac{u_i^* C v_1}{(\lambda_1 - \lambda_i) u_i^* v_i} (v_i - v_1 v_1^* v_i) + O(\varepsilon^2). \tag{4.9}$$

De cette formule, on voit que la condition du calcul du vecteur propre v_1 dépend de la grandeur $u_i^* v_i$ (comme c'est le cas pour la valeur propre ; voir la formule (4.2)) et aussi de la distance entre λ_1 & et les autres valeurs propres de A . **Un algorithme dangereux** La première méthode (déjà utilisée par Lagrange) pour calculer les valeurs propres d'une matrice A est la suivante : *calculer d'abord les coefficients du polynôme caractéristique $P_A(\lambda)$ et déterminer ' ensuite les zéros de ce polynôme*. Si la dimension de A est très petite (disons $n \leq 3$) ou si l'on fait le calcul en arithmétique exacte, cet algorithme peut être très utile. Par contre, si l'on fait le calcul en virgule flottante, cet algorithme peut donner des mauvaises surprises. Considérons, par exemple, le problème de calculer les valeurs propres de la matrice diagonale

$$A = diag(1, 2, 3, ..., n)$$

dont le polynôme caractéristique est

$$P_A(\lambda) = (1-\lambda)(2-\lambda)(3-\lambda)\cdots(n-\lambda) = (-1)^n \lambda^n + a_{n-1}\lambda^{n-1} + \cdots + a_1\lambda + a_0 \tag{4.10}$$

Les coefficients calculés satisfont $\tilde{a} = a_i(1+\varepsilon_i)$ avec $|\varepsilon_i| \leq eps$. Cette perturbation dans les coefficients provoque une grande erreur dans les zé ros de (4.10). Les résultats numériques pour $n = 9, 11, 13, 15$ (avec $eps \approx 6.10^{-8}$, simple précision) sont dessinés dans la figure V.2. **Conclusion**. Eviter le calcul des coefficients du polynôme caractéristique. Un tel algorithme est numériquement instable.

4.4 LA MÉTHODE DE LA PUISSANCE

Un algorithme simple pour calculer les valeurs propres d'une matrice A est basé sur l'itération

$$y_{k+1} = A y_k \tag{4.11}$$

où y_0 est un vecteur arbitraire. Dans le théorème suivant, on démontre que $y_k = A^k y_0$ (*méthode de la puissance*) tend vers un vecteur propre de A et que le *quotient de Rayleigh* $y_k^* A y_k / y_k^* y_k$ est une approximation d'une valeur propre de A .

Théoreme 4.4.1 Soit A une matrice diagonalisable de valeurs propres $\lambda_1, \lambda_2, ..., \lambda_n$ et de vecteurs propres $v_1, v_2, ..., v_n$ (normalisés par $\|v_i\|_2 = 1$). Si $|\lambda_1| > |\lambda_2| \geq ... \geq |\lambda_n|$, les vecteurs y_k de l'itération (4.11) vérifient

$$y_k = \lambda_1^k (a_1 v_1 + O(|\lambda_2/\lambda_1|)^k \tag{4.12}$$

(le nombre a_1 est défini par $y_0 = \sum_i a_i v_i$). Le quotient de Rayleigh satisfait (si $a_1 \neq 0$)

$$\frac{y_k^* A y_k}{y_k^* y_k} = \lambda_1 + O(|\lambda_2/\lambda_1|^k) \tag{4.13}$$

Si A est une matrice normale (c'est-à-dire. que les vecteurs propres sont orthogonaux), l'erreur dans (4.13) est $O(|\lambda_2/\lambda_1|^{2k})$

Démonstration. Exprimons le vecteur de départ y_0 dans la base des vecteurs propres, c'est-à-dire $y_0 = \sum_{i=1}^{n} a_i v_i$. Par ré currence, on voit que

$$y_k = A^k y_0 = \sum_{i=1}^{n} a_i \lambda_i^k v_i = \lambda_1^k (a_1 v_1 + \sum_{i=2}^{n} a_i \left(\frac{\lambda_i}{\lambda_1}\right)^k v_i \tag{4.14}$$

ce qui démontre la formule (4.12). De cette relation, on déduit que

$$y_k^* A y_k = y_k^* y_{k+1} = \sum_{i=1}^{n} |a_i|^2 |\lambda_i|^{2k} \lambda_i + \sum_{i \neq j}^{n} \tilde{a}_i a_j \tilde{\lambda}_i^k \lambda_j^{k+1} v_i^* v_j \tag{4.15}$$

$$y_k^* y_k = \sum_{i=1}^{n} |a_i|^2 |\lambda_i|_i^{2k} + \sum_{i \neq j}^{n} \tilde{a}_i a_j \tilde{\lambda}_i^k \lambda_j^k v_i^* v_j. \tag{4.16}$$

Si $a_1 \neq 0$, la formule (4.13) est une conséquence de

$$\frac{y_k^* A y_k}{y_k^* y_k} = \frac{|a_1|^2 . |\lambda_1|^{2k} . \lambda_1 . (1 + O(|\lambda_2/\lambda_1|^k)}{|a_1|^2 . |\lambda_1|^{2k} . (1 + O(|\lambda_2/\lambda_1|^k)}. \tag{4.17}$$

Pour une matrice normale, le deuxième terme dans les formules (4.15) et (4.16) est absent et l'expression $O(|\lambda_2/\lambda_1|^k$ peut être remplacée par $O(|\lambda_2/\lambda_1|^{2k}$ dans (4.17) et dans (4.13). ■

■ **Exemple 4.1** Considérons la matrice

$$A = \begin{pmatrix} 2 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 2 \end{pmatrix}$$

dont la valeur propre la plus grande est $\lambda_1 = 2(1 + \cos(\pi/4)) \approx 3,41421356$. Quelques itérations de la méthode de la puissance nous donnent

$$y_0 = (1,1,1)^T \quad y_1 = (3,4,3)^T \quad y_2 = (10,14,10)^T$$

et une première approximation de λ_1 est obtenue par

$$\frac{y_1^* A y_1}{y_1^* y_1} = \frac{y_1^* y_2}{y_1^* y_1} = \frac{116}{34} \approx 3,41176$$

■

Remarques. Les éléments du vecteur y_k croissent exponentiellement avec k . Il est alors recommandé de normaliser y_k après chaque itération, c'est-à-dire. de remplacer y_k par $y_k / \|y_k\|$. Sinon, on risque un "overflow". Si $|\lambda_2/\lambda_1|$ est proche de 1, la convergence est très lente. Pour accélérer la convergence, on utilise la modification suivante :

4.5 LA MÉTHODE DE LA PUISSANCE INVERSE DE WIELANDT

Supposons qu'on connaisse une approximation μ de la valeur propre cherchée λ_1 (il n'est pas · né cessaire de supposer que λ_1 soit la plus grande valeur propre de A). L'idée est d'appliquer l'itération (4.11) à la matrice $(A - \mu I)^{-1}$ (Les valeurs propres de cette matrice sont $(\lambda_i - \mu)^{-1}$. Si μ est proche de λ , on a ·

$$\frac{1}{|\lambda_1 - \mu|} \gg \frac{1}{|\lambda_i - \mu|} \quad \text{pour} \quad i \geq 2$$

et la convergence va être très rapide. L'itération devient alors $y_{k+1} = (A-\mu I)^{-1} y_k$ ou

$$(A-\mu I)y_{k+1} = y_k \tag{4.18}$$

Après avoir calculé la décomposition LU de la matrice $A-\mu I$, une itération de (4.18) ne coûte pas plus cher qu'une de (4.11). Pour la matrice A de l'exemple précédent, choisissons $\mu = 3,41$ et $y_0 = (1;1,4;1)^T$ Deux itérations de (4.18) nous donnent

$$y_1 = \begin{pmatrix} 236,134453781513 \\ 333,949579831933 \\ 236,134453781513 \end{pmatrix}, \quad y_2 = \begin{pmatrix} 56041,9461902408 \\ 79255,2785820210 \\ 56041,9461902408 \end{pmatrix}$$

et on obtient

$$\frac{1}{\lambda_1 - 3,41} \approx \frac{y_1^*(A-\mu I)^{-1} y_1}{y_1^* y_1} = \frac{y_1^* y_2}{y_1^* y_1} \approx 237,328870774159$$

De cette relation, on calcule λ_1 et on obtient l'approximation $3,41421356237333$. Les 13 premiers chiffres sont corrects. La méthode de la puissance (et celle de Wielandt) est importante pour la compréhension d'autres algorithmes. Si l'on veut calculer toutes les valeurs propres d'une matrice, on utilise des méthodes encore plus sophistiquées. En pratique, on procède de la manière suivante :

- on distingue les cas : A symétrique ou A quelconque.
- on cherche P telle que $P^{-1}AP$ devienne une matrice de Hessenberg (ou une matrice tridiagonale, si A est symétrique).
- on applique l'algorithme QR à la matrice H.
- si H est une matrice tridiagonale et symétrique, on peut é galement appliquer la méthode de bissection.

4.5.1 CALCUL DIRECT DE $\det(A-\lambda I)$

On se donne $(n+1)$ valeurs distinctes $\lambda_1, \lambda_2, ..., \lambda_{n+1}$ quelconques ; on calcule pour chacune d'elles la valeur

$$y_i = \det(A-\lambda_i I) \qquad i = 1,2,...,n+1.$$

On obtient ainsi un ensemble de valeurs $\{(\lambda_i, y_i)\}_{i=1,2,...,n+1}$. On détermine alors, le polynôme d'interpolation passant par ces points. Il sera identique, à un facteur multiplicatif près, au polynôme caractéristique de A. On peut alors chercher ses racines par l'une des méthodes connues ; ce qui aboutira à une approximation des valeurs propres de A.

4.6 MÉTHODE DE KRYLOV

La méthode consiste à calculer les coefficients du polynôme caractéristique dont on approche les racines à l'aide des méthodes connues. Les vecteurs propres associés sont alors déterminés par les formules appropriées. Plus précisément, soit :

$$P(\lambda) = (-1)^n(\lambda^n - \sum_{k=1}^{n} a_k \lambda^{n-k})$$

le polynôme caractéristique de A. D'après le théorème de Cayley-Hamilton (A annule son polynôme caractéristique), on a donc $P(A) = 0$; donc :

$$A^n = \sum_{k=1}^{n} a_k A^{n-k}$$

Prenons un vecteur quelconque x_0 non nul, on a :

$$A^n x_0 = \sum_{k=1}^{n} a_k A^{n-k} x_0 \tag{4.19}$$

Notons a le vecteur de composante (a_i) ; $i = 1,2,...,n$

$$\begin{aligned} x_1 &= Ax_0 \\ x_2 &= A^2 x_0 \\ &..... \\ x_{n-1} &= A^{n-1} x_0 \end{aligned}$$

et B la matrice dont les n colonnes sont les vecteurs

$$x_{n-1}, x_{n-2}, ..., x_1, x_0$$

L'équation (4.19) peut s'écrire :

$$A^n x_0 = Ba \tag{4.20}$$

Pour x_0 donné, on cherche a solution de (4.20). On obtiendra ainsi, si le système est inversible, les coefficients du polynôme caractéristique et on pourra utiliser une des méthodes de résolution des équations non linéaires pour calculer ses racines qui sont les valeurs propres de A. Cette méthode permet en outre de déterminer les vecteurs propres associés aux valeurs propres calculées. Pour simplifier, nous supposerons que les valeurs propres $\lambda_1, \lambda_2, ..., \lambda_n$ sont distinctes. Si nous appelons $v_1, v_2, ..., v_n$ les vecteurs propres associés respectivement à $\lambda_1, \lambda_2, ..., \lambda_n$, nous pouvons décomposer le vecteur x_0, choisi arbitrairement dans la recherche des coefficients (a_i), suivant la base des $\{v_i\}_{i=1,2,...,n}$. On a alors

$$x_0 = \alpha_1 v_1 + \alpha_2 v_2 + ... + \alpha_n v_n$$

comme

$$\begin{aligned} Av_i &= \lambda_i v_i \\ A^2 v_i &= \lambda_i^2 v_i \\ &..... \end{aligned}$$

on aura

$$\begin{aligned} x_1 &= \alpha_1 \lambda_1 v_1 + \alpha_2 \lambda_2 v_2 + ... + \alpha_n \lambda_n v_n \\ ... \quad &... \quad ..... \\ x_{n-1} &= \alpha_1 \lambda_1^{n-1} v_1 + \alpha_2 \lambda_2^{n-1} v_2 + ... + \alpha_n \lambda_n^{n-1} v_n \end{aligned}$$

Considérons une combinason linéaire des vecteurs $x_0, x_1, ..., x_{n-2}, x_{n-1}$. On a

$$\begin{aligned} x_{n-1} &= \beta_{i1} x_{n-2} + \cdots + \beta_{in-1} x_0 = \\ &\quad \alpha_1 \varphi_i(\lambda_1) v_1 + \alpha_2 \varphi_i(\lambda_2) v_2 + \cdots + \alpha_n \varphi_i(\lambda_n) v_n \\ \text{où } \varphi_i(\lambda) &= \lambda^{n-1} + \beta_{i1} \lambda^{n-2} + \cdots + \beta_{in-1} \end{aligned} \tag{4.21}$$

on peut choisir par exemple la formule (4.21) s'écrit alors

$$x_{n-1} + \beta_{i1} x_{n-2} + ... + \beta_{in-1} x_0 = \alpha_i \varphi_i(\lambda_i) v_i.$$

Donc, si $\alpha_i \neq 0$ la combinaison linéaire obtenue permet de déterminer le vecteur propre v_i. Avec le choix (**??**), les coefficients β_{ij} s'obtiennent facilement par identification. Plus précisement nous avons

$$\begin{aligned} \beta_{i0} &= 1 \\ \beta_{ij} &= \lambda_i \beta_{ij-1} - a_j \end{aligned}$$

R Si les vecteurs $x_0, x_1, ..., x_{n-2}, x_{n-1}$ sont linéairement ind épendant, la matrice B est inversible et on obtient bien les coefficients caractéristiques dont on peut calculer les valeurs propres. Mais il arrive que ces vecteurs ne soient pas linéairement indé pendants. Par exemple x_k s'exprime comme combinaison linéaire des précédents et de même des suivants. On peut alors appliquer la méthode précédente avec les vecteurs $x_0, x_1, ..., x_{k-1}$; ce qui donnera un polynôme dont les racines seront racines du polynôme caractéristique et donc valeurs propres de A. On évite en général cette complication en changeant de vecteur initial.

4.7 MÉTHODE DE LEVERRIER

Les coefficients du polynôme caractéristique sont déterminés par la formule (4.23) suivante. On utilise ensuite les méthodes de résolution des équations non linéaires pour calculer les racines de ce polynôme ; ce qui détermine les valeurs propres. Posons

$$P(x) = a_1 x^n + a_2 x^{n-1} + ... + a_{n+1} \quad \text{avec} \quad a_1 \neq 0.$$

Les relations de Newton entre les racines $x_1, x_2, ..., x_n$ et les coefficients de ce polynôme sont donnés par

$$\left\{ \begin{array}{rcl} a_2 + a_1 S_1 & = & 0 \\ 2a_3 + a_2 S_1 + a_1 S_2 & = & 0 \\ \cdots\cdots\cdots\cdots & \cdots & \cdots \\ k a_{k+1} + a_k S_1 + \cdots + a_1 S_k & = & 0 \\ \cdots\cdots\cdots\cdots & \cdots & \cdots \\ n a_{n+1} + a_n S_1 + \cdots + a_1 S_n & = & 0 \end{array} \right. \tag{4.22}$$

avec $S_k = \sum_{i=1}^{n} x_i^k$. Donc, en considérant le polynôme caractéristique de A, dont les racines sont les valeurs propres λ_i de A, on a

$$\begin{aligned} S_k &= T_r(A^k) \\ a_1 &= (-1)^n \end{aligned}$$

ce qui nous permet de calculer les coefficients a_k pour $k = 2, ..., n+1$. Plus précisément on a :

$$a_k = -\frac{1}{k-1} a_{k-1} S_1 + ... + a_2 S_{k-2} + a_1 S_{k-1} \tag{4.23}$$

R La méthode de Leverrier présente un grave inconvénient : elle impose le calcul des puissances souvent élevées de la matrice initiale. Par contre son algorithme est simple et il n'y a pas lieu d'envisager des cas particuliers

4.8 TRANSFORMATION SOUS FORME TRIDIAGONALE (ou de HESSENBERG)

Avec la transformation $v = Pu$ (où est P une matrice inversible) le problème

$$Av = \lambda v$$

devient

$$P^{-1}APu = \lambda u$$

Donc, les valeurs propres de A et de $P^{-1}AP$ sont les mêmes et les vecteurs propres v_i de A se transforment par $v_i = Pu_i$. Le but de ce paragraphe est de trouver une matrice P telle que $P^{-1}AP$ devienne "plus simple". La situation idéale serait trouvée si $P^{-1}AP$ devenait diagonale ou triangulaire - mais une telle transformation nécessiterait déjà la connaissance des valeurs propres. Alors, on cherche P tel que $P^{-1}AP$ soit sous forme de Hessenberg

$$P^{-1}AP = H = \begin{pmatrix} * & * & \cdots & \cdots & * \\ * & * & \ddots & & \vdots \\ & * & \ddots & \ddots & * \\ & & \ddots & \ddots & * \\ & & & * & * \end{pmatrix} \tag{4.24}$$

c'est-à-dire, $h_{ij} = 0$ pour $i > j+1$. Pour arriver à ce but, nous considérons deux algorithmes.

4.8.1 a) A l'aide des transformations élémentaires

Comme pour l'élimination de Gauss, nous utilisons les transformations pour faire apparaitre les zéros - colonne par colonne - dans (4.24). Dans un premier pas, nous choisissons $k \geq 2$ tel que $|a_{k1}| \geq |a_{j1}|$ pour $j \geq 2$ et nous permutons les lignes 2 et k, c'est-à-dire, nous formons PA o ù P est une matrice de permutation convenable. Pour ne pas changer les valeurs propres, il faut également permuter les colonnes 2 et k (ceci correspond au calcul de $A' = PAP^{-1}$ car $P^2 = I$ (et donc $P = P^{-1}$). Si $a'_{21} = 0$, on a aussi $a'_{i1} = 0$ pour $i \geq 3$ et le premier pas est terminé. Sinon, nous déterminons

$$L_2 = \begin{pmatrix} 1 & & & & \\ 0 & 1 & & & \\ 0 & -l_{32} & 1 & & \\ \vdots & \vdots & \ddots & \ddots & \\ 0 & -l_{n2} & \cdots & 0 & 1 \end{pmatrix} \quad \text{telle que} \quad L_2A' = \begin{pmatrix} a'_{11} & a'_{12} & \cdots & a'_{1n} \\ a'_{21} & a'_{22} & \cdots & a'_{2n} \\ 0 & * & \cdots & * \\ \vdots & \vdots & \ddots & \vdots \\ 0 & * & \cdots & * \end{pmatrix}$$

Pour ceci, on définit $l_{i2} = \frac{a'_{i1}}{a'_{21}}$. Une multiplication à droite avec

$$L_2^{-1} = \begin{pmatrix} 1 & & & & \\ 0 & 1 & & & \\ 0 & l_{32} & 1 & & \\ \vdots & \vdots & \ddots & \ddots & \\ 0 & l_{n2} & \cdots & 0 & 1 \end{pmatrix}$$

ne change pas la première colonne de L_2A'. On répète la même procédure avec la sous-matrice de $L_2A'L_2^{-1}$ de dimension $n-1$, et ainsi de suite. A cause des multiplications à droite avec L_i^{-1}, cet algorithme coûte deux fois plus cher que l'élimination de Gauss. Pour la matrice

$$A = \begin{pmatrix} 3 & 2 & 1 \\ 2 & 1 & 3 \\ 1 & 3 & 1 \end{pmatrix}$$

on prend

$$L_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1/2 & 1 \end{pmatrix}$$

et on obtient

$$L_2 A = \begin{pmatrix} 3 & 2 & 1 \\ 2 & 1 & 3 \\ 0 & 5/2 & -1/2 \end{pmatrix}, \quad \text{puis} \quad L_2 A L_2^{-1} = \begin{pmatrix} 3 & 5/2 & 1 \\ 2 & 5/2 & 3 \\ 0 & 9/4 & -1/2 \end{pmatrix} = H$$

Cet exemple montre un désavantage de cet algorithme : si l'on part avec une matrice symétrique A, la matrice de Hessenberg H, obtenue par cet algorithme, n'est plus symétrique en général.

4.8.2 b) A l'aide des transformations orthogonales

Il est souvent préférable de travailler avec des réflexions de Householder. Commençons par une réflexion pour les coordonnées $2,\ldots,n$ laissant fixe la première coordonnée :$\bar{Q}_2 = I - 2\bar{u}_2\bar{u}_2^T$ ($\|\bar{u}_2\|_2 = 1$) tel que $\bar{Q}_2\bar{A}_1 = \alpha_2 e_1$ où $\bar{A}_1 = (a_{21},\ldots,a_{n1})$. En posant $u_2 = (0,\bar{u}_2)^T$ et $Q_2 = I - 2u_2u_2^T$, la matrice Q_2A contient des zéros dans la première colonne à partir du troisième élément. La multiplication à droite avec $Q_2^{-1} = Q_2^T = Q_2$ ne change pas cette colonne :

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \xrightarrow{Q_2A} \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ \alpha_2 & * & * \\ 0 & * & * \end{pmatrix} \xrightarrow{Q_2AQ_2} \begin{pmatrix} a_{11} & * & * \\ \alpha_2 & * & * \\ 0 & * & * \end{pmatrix}$$

Dans le pas suivant, on applique la même procédure à la sous-matrice de dimension $n-1$, etc. Finalement, on arrive à la forme de Hessenberg (4.24) avec la tranformation $P^{-1} = Q_{n-1}\ldots Q_2$ qui est une matrice orthogonale (c'est-à-dire, $P^{-1} = P^T$). Nous avons un double avantage avec cet algorithme :

— il ne faut pas faire une recherche de pivot ;
— si A est symétrique, alors $P^{-1}AP$ est aussi symétrique, et donc tridiagonale.

4.8.3 Méthode de bissection pour des matrices tridiagonales

Considérons une matrice symétrique tridiagonale

$$A = \begin{pmatrix} d_1 & e_2 & & & \\ e_2 & d_2 & e_3 & & \\ & e_3 & \ddots & \ddots & \\ & & \ddots & \ddots & e_n \\ & & & e_n & d_n \end{pmatrix}$$

On observe tout d'abord que si un élément e_i est nul, la matrice A est déjà décomposée en deux sous-matrices du m ême type, qui ensemble fournissent les valeurs propres de A . On peut donc supposer, sans restreindre la généralité, que

$$c_i \neq 0 \quad \text{pour} \quad i = 2,\ldots,n. \tag{4.25}$$

Pour cette matrice, il est possible de calculer la valeur $P_A(\lambda)$ du polynôme caractéristique sans connaitre ses coefficients. En effet, si l'on pose

$$A_1 = (d_1), \qquad A_2 = \begin{pmatrix} d_1 & e_2 \\ e_2 & d_2 \end{pmatrix}, \qquad A_3 = \begin{pmatrix} d_1 & e_2 & \\ e_2 & d_2 & e_3 \\ & e_3 & d_3 \end{pmatrix}, \qquad \ldots$$

et si l'on définit

$$p_i(\lambda) = \det(A_i - \lambda I),$$

on obtient

$$\begin{aligned} p_0(\lambda) &= 1 \\ p_1(\lambda) &= d_1 - \lambda \\ p_i(\lambda) &= (d_i - \lambda)\, p_{i-1}(\lambda) - e_i^2 p_{i-2}(\lambda), \qquad i = 2, ..., n. \end{aligned} \tag{4.26}$$

La formule de récurrence dans (4.26) est obtenue en dé veloppant le déterminant de la matrice $A_i - \lambda I$ par rapport à la dernière ligne (ou colonne). En principe, on peut maintenant calculer les valeurs propres de A (c'est- à-dire. les zéros de $p_n(\lambda)$ de la manière suivante : chercher un intervalle où $p_n(\lambda)$ change de signe et localiser une racine de $p_n(\lambda) = 0$ par bissection. Les évaluations de $p_n(\lambda)$ sont faites à l'aide de la formule (4.26). Mais il existe une astuce interessante qui permet d'améliorer cet algorithme.

Théoreme 4.8.1 Si l'équation (4.25) est vérifié, les polyn ômes $p_i(\lambda)$ définis par (4.26) satisfont
a) $p_n'(\hat{\lambda}) p_{n-1}(\hat{\lambda}) < 0$ si $p_n(\hat{\lambda}) = 0$ $(\hat{\lambda} \in \mathbb{R})$
b) $p_{i-1}(\hat{\lambda}) p_{i+1}(\hat{\lambda}) < 0$ si $p_i(\hat{\lambda}) = 0$ pour un $\{i \in 1, 2, ..., n-1\}$
c) $p_0(\lambda)$ ne change pas de signe sur $\mathbb{R}$.

Démonstration. L'affirmation (c) est triviale. Si $p_i(\hat{\lambda}) = 0$ pour un $\{i \in 1, 2, ..., n-1\}$, la formule de récurrence (4.26) donne l'inégalité $p_{i-1}(\hat{\lambda}) p_{i+1}(\hat{\lambda}) \leq 0$. Pour démontrer (b), il suffit d'exclure le cas $p_{i-1}(\hat{\lambda}) p_{i+1}(\hat{\lambda}) = 0$. Si deux valeurs consécutives de la suite $\left\{ p_i(\hat{\lambda}) \right\}$ sont nulles, la formule de récurrence montre que $p_i(\hat{\lambda}) = 0$ pour tout i, ce qui contredit $p_0(\lambda) = 1$. Nous démontrons par récurrence que toutes les racines de $p_i(\lambda)$ sont réelles, simples et séparées par celles de $p_{i-1}(\lambda)$. Il n'y a rien à démontrer pour $i = 1$. Supposons la propriété vraie pour i et montrons qu'elle est encore vraie pour $i+1$. Comme les z éros $\lambda_1 < \lambda_2 < ... < \lambda_i$ sont séparés par ceux de $p_{i-1}(\lambda)$ et comme $p_{i-1}(-\infty) = +\infty$, nous avons *sign* $p_{i-1}(\lambda_j) = (-1)^{j+1}$. Alors, on d éduit de (b) que *sign* $p_{i+1}(\lambda_j) = (-1)^j$. Ceci et le fait que $p_{i+1}(\lambda) = (-1)^{j+1} \lambda^{i+1} + ...$ montrent que $p_{i+1}(\lambda)$ possède un zéro ré el dans chacun des intervalles ouverts $(-\infty, \lambda_1), (\lambda_1, \lambda_2), ..., (\lambda_i, \infty)$ L'affirmation (a) est maintenant une conséquence de (b) et du fait que toutes les racines de $p_{i-1}(\lambda)$ sont réelles simples ; ■

Définition 4.8.1 — suite de Sturm. Une suite $\{p_0, p_1, ..., p_n\}$ de polynômes à coefficients réels s'appelle une suite de Sturm, si elle vérifie les conditions (a), (b), (c) du Théorème (4.8.1)

Considérons une suite de Sturm $\{p_0, p_1, ..., p_n\}$. Si l'on définit

$$\omega(\lambda) = \text{nombre de changements de signes de } \{p_0(\lambda), p_1(\lambda), ..., p_n(\lambda)\}$$

alors le polynôme $p_n(\lambda)$ possède exactement

$$\omega(b) - \omega(a)$$

zéros dans l'intervalle $[a, b]$ (si $p_i(\lambda) = 0$, on définit $sign p_i(\lambda) = sign\ p_{i-1}(\lambda)$).

Démonstration. Par continuité, l'entier $\omega(\lambda)$ peut changer sa valeur seulement si une valeur des fonctions $p_i(\lambda)$ devient nulle. La fonction $p_0(\lambda)$ ne change pas de signe. Supposons alors que

$p_i(\tilde{\lambda}) = 0$ pour un $i \in \{1,2,...,n-1\}$. La condition (b) et la continuité de $p_j(\lambda)$ montrent que seulement les deux situations suivantes sont possibles (ε petit) :

	$\tilde{\lambda}-\varepsilon$	$\tilde{\lambda}$	$\tilde{\lambda}+\varepsilon$
$p_{i-1}(\lambda)$	$+$	$+$	$+$
$p_i(\lambda)$	$\pm$	0	$\pm$
$p_{i+1}(\lambda)$	$-$	$-$	$-$

	$\tilde{\lambda}-\varepsilon$	$\tilde{\lambda}$	$\tilde{\lambda}+\varepsilon$
$p_{i-1}(\lambda)$	$-$	$-$	$-$
$p_i(\lambda)$	$\pm$	0	$\pm$
$p_{i+1}(\lambda)$	$+$	$+$	$+$

■

Chaque fois, on a $\omega(\tilde{\lambda}+\varepsilon) = \omega(\tilde{\lambda}) = \omega(\tilde{\lambda}-\varepsilon)$ et la valeur de $\omega(\lambda)$ ne change pas si λ traverse un zéro de $p_i(\lambda)$ pour $i \in \{1,2,...,n-1\}$ Il reste à étudier la fonction $\omega(\lambda)$ dans un voisinage d'un zéro $\tilde{\lambda}$ de $p_n(\lambda)$. La propriété (a) implique que pour les signes de $p_j(\lambda)$ on a seulement les deux possibilités suivantes :

	$\tilde{\lambda}-\varepsilon$	$\tilde{\lambda}$	$\tilde{\lambda}+\varepsilon$
$p_{n-1}(\lambda)$	$+$	$+$	$+$
$p_n(\lambda)$	$+$	0	$-$

	$\tilde{\lambda}-\varepsilon$	$\tilde{\lambda}$	$\tilde{\lambda}+\varepsilon$
$p_{n-1}(\lambda)$	$-$	$-$	$-$
$p_n(\lambda)$	$-$	0	$+$

c'est-à-dire, $\omega(\tilde{\lambda}+\varepsilon) = \omega(\tilde{\lambda}-\varepsilon)+1$. Ceci démontre que la fonction $\omega(\lambda)$ est constante par morceaux et augmente de 1 sa valeur si λ traverse un zéro de $p_n(\lambda)$.

4.8.4 Méthode de bissection.

Si l'on applique ce théorème à la suite (4.26), la diff érence $\omega(b) - \omega(a)$ est égale au nombre de valeurs propres de (4.25) dans l'intervalle $[a,b]$. On obtient toutes les valeurs propres de A de la manière suivante :

— on cherche un intervalle $[a,b]$ qui contienne toutes les valeurs propres de A (par exemple, en appliquant le théorème de Gershgorin). On a donc que $\omega(a) = 0$ et $\omega(b) = n$.

— on pose $c = \frac{(a+b)}{2}$ et on calcule $\omega(c)$. Les différences $\omega(c) - \omega(a)$ et $\omega(b) - \omega(c)$ indiquent combien de valeurs propres de A sont dans $[a,c)$ et combien sont dans $[c,b)$

— on continue à diviser les intervalles qui contiennent au moins une valeur propre de A.

On peut facilement modifier cet algorithme pour calculer la valeur propre la plus petite ou la $3^{ème}$ plus grande valeur propre, etc. Pour éviter un "overflow" dans le calcul de $p_n(\lambda)$ (si n et λ sont grands), il vaut mieux travailler avec

$$f_i(\lambda) = \frac{p_i(\lambda)}{p_{i-1}(\lambda)} \qquad i = 1,2,...,n$$

et utiliser le fait que

$$\omega(\lambda) = \text{nombre d'éléments négatifs parmi } \{f_1(\lambda), f_2(\lambda), ..., f_n(\lambda)\}$$

(attention : si $p_{i-1}(\lambda)$ est zéro, on pose $f_i(\lambda) = -\infty$; cette valeur compte pour un élément négatif). Pour une programmation de l'algorithme, on utilise la récurrence

$$\begin{aligned} f_1(\lambda) &= d_1 - \lambda \\ f_i(\lambda) &= d_i - \lambda - \begin{cases} e_i^2/f_{i-1}(\lambda) & \text{si} \quad f_{i-1}(\lambda) \neq 0 \\ |e_i|/eps & \text{si} \quad f_{i-1}(\lambda) = 0 \end{cases} \end{aligned}$$

La formule pour le cas $f_{i-1}(\lambda) \neq 0$ est une conséquence de (4.26). Si $f_{i-1}(\lambda) = 0$ (c'est-à-dire $p_{i-1}(\lambda) = 0$), on remplace cette valeur par $|e_i|.eps$. Ceci correspond à ajouter la perturbation $|e_i|.eps$ à d_{i-1}

4.9 L'ITÉRATION ORTHOGONALE

Dans ce paragraphe, nous allons généraliser la méthode de la puissance afin de pouvoir calculer les deux (trois,...) valeurs propres dominantes en même temps. Cette généralisation motivera l'ité ration QR qui constitue l'algorithme le plus important pour le calcul des valeurs propres d'une matrice.

4.9.1 Généralisation de la méthode de la puissance (pour calculer les deux valeurs propres dominantes).

Considérons une matrice A dont les valeurs propres satisfont

$$|\lambda_1| > |\lambda_2| > ... > |\lambda_n| . \tag{4.27}$$

La méthode de la puissance est basée sur l'itération $y_{k+1} = Ay_k$ et nous permet d'obtenir une approximation de λ_1 à l'aide du quotient de Rayleigh. Pour calculer (en même temps) la deuxième valeur propre λ_2, nous prenons deux vecteurs y_0 et z_0 satisfaisant $y_0^* z_0 = 0$ et nous considérons l'itération

$$\begin{aligned} y_{k+1} &= Ay_k \\ z_{k+1} &= Az_k - \beta_{k+1} y_{k+1} \end{aligned} \tag{4.28}$$

où β_{k+1} est déterminé par la condition $y_{k+1}^* z_{k+1} = 0$. Par induction, on voit que

$$\begin{aligned} y_k &= A^k y_0 \\ z_k &= A^k z_0 - \gamma_k y_k \end{aligned}$$

où γ_k est tel que

$$y_k^* z_k = 0 \tag{4.29}$$

Ceci signifie que le calcul de $\{z_k\}$ correspond à la m éthode de la puissance appliquée à z_0, combinée avec une orthogonalisation (projection de $A^k z_0$ sur le complément orthogonal de y_k). En exprimant les vecteurs initiaux dans la base de vecteurs propres $v_1, v_2, ..., v_n$ de la matrice A (on suppose $\|v_i\|_2 = 1$),

$$y_0 = \sum_{i=1}^{n} a_i v_i, \qquad z_0 = \sum_{i=1}^{n} b_i v_i, \tag{4.30}$$

les vecteurs y_k, z_k deviennent

$$y_k = \sum_{i=1}^{n} a_i \lambda_i^k v_i, \qquad z_0 = \sum_{i=1}^{n} (b_i - \gamma_k a_i) \lambda_i^k v_i,$$

Comme nous l'avons constaté précédement, pour $k \to \infty$, le terme $a_1 \lambda_1^k v_1$ est dominant dans y_k (si $a_1 \neq 0$) et on obtient une approximation du premier vecteur propre v_1 . Que peut-on dire pour la suite $\{z_k\}$? La condition (4.29) d'orthogonalité implique que

$$\sum_{i=1}^{n} \sum_{i=1}^{n} a_i (b_j - \gamma_k a_j) \bar{\lambda}_i^k \lambda_j^k v_i^* v_j = 0 \tag{4.31}$$

Cette relation définit γ_k. Comme le terme avec $i = j = 1$ est dominant, on voit que $\gamma_k \approx b_1/a_1$ Par la suite, nous allons supposer que $a_1 \neq 0$ et $a_1 b_2 - a_2 b_1 \neq 0$. En divisant (4.31) par $\bar{\lambda}_1^k$ on obtient

$$\bar{a}_1 (b_1 - \gamma_k a_1) \lambda_1^k (1 + O(|\lambda_2/\lambda_1|^k)) = -\bar{a}_1 (b_2 - \gamma_k a_2) \lambda_2^k (v_1^* v_2 + O(|\lambda_2/\lambda_1|^k) + O(|\lambda_3/\lambda_2|^k)).$$

Maintenant, on peut insérer cette formule dans (4.30) et on en d éduit

$$z_k = \lambda_{21}^k (b_2 - \gamma_k a_2)(v_2 - v_1^* v_2 . v_1 + O(|\lambda_2/\lambda_1|^k) + O(|\lambda_3/\lambda_2|^k)) \tag{4.32}$$

Visiblement, le vecteur z_k s'approche (pour $k \to \infty$) d'un multiple de $v_2 - v_1^* v_2 . v_1$, qui est la projection orthogonale de v_2 à l'hyperplan $v_1^\perp$. Concernant les valeurs propres, on a le résultat suivant.

Théoreme 4.9.1 Considérons les vecteurs y_k, z_k donnés par (4.28) et notons

$$U_k = (y_k / \|y_k\|_2 , z_k / \|z_k\|_2) \tag{4.33}$$

(observer que $U_k^* U_k = 1$). Si (4.27) est vérifié , on a que

$$U_k^* A U_k \longrightarrow \begin{pmatrix} \lambda_1 & * \\ 0 & \lambda_2 \end{pmatrix} \qquad \text{pour} \qquad k \longrightarrow \infty \tag{4.34}$$

Démonstration. L'élément (1,1) de la matrice $U_k^* A U_k$ est le quotient de Rayleigh (4.13) qui converge vers λ_1 . En utilisant (4.32), on voit que l'élément (2,2) satisfait

$$\frac{z_k^* A z_k}{z_k^* z_k} \longrightarrow \frac{(v_2 - v_1^* v_2 . v_1)^* (\lambda_2 v_2 - \lambda_1 v_1^* v_2 . v_1)}{(v_2 - v_1^* v_2 . v_1)^* (v_2 - v_1^* v_2 . v_1)} = \frac{\lambda_2 (1 - |v_1^* v_2|^2)}{1 - |v_1^* v_2|^2} = \lambda_2$$

De façon similaire, on obtient pour l'élément (2,1)

$$\frac{z_k^* A y_k}{\|z_k\|_2 \|y_k\|_2} \longrightarrow \frac{(v_2 - v_1^* v_2 . v_1)^* \lambda_1 v_1}{\|v_2 - v_1^* v_2 . v_1\|_2 \|v_1\|_2} = 0$$

Finalement, l'élément (1,2) de $U_k^* A U_k$ satisfait

$$\frac{y_k^* A z_k}{\|y_k\|_2 \|z_k\|_2} \longrightarrow \frac{v_1^* (\lambda_2 v_2 - \lambda_1 v_1^* v_2 . v_1)}{\|v_1\|_2 \|v_2 - v_1^* v_2 . v_1\|_2} = \frac{(\lambda_2 - \lambda_1) v_1^* v_2}{\sqrt{1 - |v_1^* v_2|^2}} .$$

Cette expression est en général non nulle. ■

R Avec la notation (4.33), l'itération (4.28) peut être écrite sous la forme

$$A U_k = U_{k+1} R_{k+1} \tag{4.35}$$

où R_{k+1} est une matrice 2×2 qui est triangulaire supé rieure.

4.9.2 Méthode de la puissance (pour le calcul de toutes les valeurs propres)

ou simplement *itération orthogonale*. La généralisation de l'algorithme précédent au cas où l'on veut calculer toutes les á valeurs propres d'une matrice est évidente : on choisit une matrice orthogonale U_0, c'est-à-dire, on choisit n vecteurs orthogonaux (les colonnes de U_0) qui jouent le rôle de y_0, z_0, etc. Puis, on effectue l'itération

```
for k=1,2,..
    Z_{k}=AU_{k+1}          (decomposition QR)
    U_{k}R_{k}=Z_{k}
end
```

Si (4.27) est vérifié et si la matrice U_0 est bien choisie ($a_1 \neq 0, a_1b_2 - a_2b_1 \neq 0,$, etc), une géné ralisation du théorème précédent donne la convergence

$$T_k = U_k^* A U_k \tag{4.36}$$

vers une matrice triangulaire dont les éléments de la diagonale sont les valeurs propres de A . On a donc transformé A en forme triangulaire à l'aide d'une matrice orthogonale (*décomposition de Schur*). Il y a une possibilité intéressante pour calculer T_k de (4.36) directement à partir de T_{k-1}. D'une part, on déduit de (4.35) que

$$T_{k-1} = U_k^* A U_k = \left(U_{k-1}^* U_k\right) R_k \tag{4.37}$$

D'autre part, on a

$$T_k = U_k^* A U_k = U_k^* A U_{k-1}^* U_k = R_k \left(U_{k-1}^* U_k\right).$$

On calcule la décomposition QR de la matrice T_{k-1} et on é change les deux matrices de cette décomposition pour obtenir T_k

4.9.3 L' algorithme QR

La méthode QR, due à J.C.F. Francis et à V.N. Kublanovskaya, est la méthode la plus couramment utilisée pour le calcul de l'ensemble des valeurs propres . La version simple du célèbre algorithme QR n'est rien d'autre que la méthode du paragraphe précédent. En effet, si l'on pose $Q_k = U_{k-1}^* U_k$ et si l'on commence l'itération avec $U_0 = I$, les formules (4.36) et (4.37) nous permettent d' écrire l'algorithme précédent comme suit : (décomposition QR)

```
T_{0}=A
for  k=1,2,..
    Q_{k}R_{k}=T_{k-1}
    T_{k}=R_{k}Q_{k}
end
```

Les T_k qui sont les mêmes que dans le paragraphe précédent, convergent (en général) vers une matrice triangulaire. Ceci nous permet d'obtenir toutes les valeurs propres de la matrice A car les T_k ont les mêmes valeurs propres que A (voir (4.36)). Cet algorithme important a été développé indépendamment par J.G.F. Francis et par V.N. Kublanovskaya. Un algorithme similaire, qui utilise la décomposition LR à la place de la dé composition QR, a été introduit par H. Rutishauser.

■ **Exemple 4.2** Appliquons la méthode QR à la matrice

$$A = \begin{pmatrix} 10 & 2 & 3 & 5 \\ 3 & 6 & 8 & 4 \\ 0 & 5 & 4 & 3 \\ 0 & 0 & 4 & 3 \end{pmatrix}$$

On peut montrer que, pour une matrice de Hessenberg A , toutes les matrices T_k sont aussi sous forme de Hessenberg. Pour 'étudier la convergence vers une matrice triangulaire, il suffit alors de considérer les éléments $t_{i+1,i}^{(k)} (i = 1, 2, ..., n-1)$ de la sous-diagonale. On constate que

$$\frac{t_{i+1,i}^{(k+1)}}{t_{i+1,i}^{(k)}} \approx \frac{\lambda_{i+1}}{\lambda_i} \tag{4.38}$$

($\lambda_1 \approx 14,3, \lambda_2 \approx 7,86, \lambda_3 \approx 2,70, \lambda_4 \approx -1,86$). Comme, les éléments $t_{i+1,i}^{(k)}$ convergent, pour $k \longrightarrow \infty$, liné airement vers 0 (voir la figure V.4, où les valeurs sont dessinées en fonction du nombre k de l'itération). ■

R

(a) Comme le calcul de la décomposition QR d'une matrice pleine est très coûteux ($O(n^3)$ opérations), on applique l'algorithme QR uniquement aux matrices de Hessenberg. Dans cette situation une itération nécessite seulement $O(n^3)$ opérations.

(b) La convergence est très lente en général (seulement *linéaire*). Pour rendre efficace cet algorithme, il faut absolument trouver un moyen pour accélérer la convergence.

(c) Considérons la situation où A est une matrice ré elle qui possède des valeurs propres complexes (l'hypothèse (4.27) est violée). L'algorithme QR produit une suite de matrices T_k qui sont toutes réelles. Dans cette situation, les T_k ne convergent pas vers une matrice triangulaire, mais deviennent triangulaires par blocs (sans démonstration). Comme la dimension des blocs dans la diagonale vaut en général 1 ou 2, on obtient également des approximations des valeurs propres.

4.9.4 Accélération de la convergence

D'après l'observation (4.38), nous savons que

$$t_{n,n-1}^{(k)} = O(|\lambda_n/\lambda_{n-1}|^k)$$

La convergence vers zéro de cet élément ne va être rapide que si $|\lambda_n| \ll |\lambda_{n-1}|$. Une idée géniale est d'appliquer l'algorithme QR à la matrice $A - pI$ où $p \approx \lambda_n$ Comme les valeurs propres de $A - pI$ sont $\lambda_i - p$, on a la propriété $|\lambda_n - p| \ll |\lambda_i - p|$ pour $i = 1, \ldots, n-1$ et l'élément $t_{n,n-1}^{(k)}$ va converger rapidement vers zéro. Rien ne nous empêche d'amé liorer l'approximation p après chaque itération. L'algorithme QR avec "shift" devient alors :

```
T_{0}=A
for
   k=1,2,..
determiner le parametre p_{k-1}$
Q_{k}R_{k}=T_{k-1}-p_{k-1}I   (decomposition QR)
T_{k}=R_{k}Q_{k}+p_{k-1}
end
```

Les matrices T_k de cette itération satisfont

$$Q_k^* T_{k-1} Q_k = Q_k^* (Q_k R_k + p_{k-1} I) Q_k = R_k Q_k + p_{k-1} I = T_k \tag{4.39}$$

Ceci implique que, indépendamment de la suite p_k , les matrices T_k ont toutes les mêmes valeurs propres que $T_0 = A$. Pour décrire complètement l'algorithme QR avec shift, il faut encore discuter le choix du paramètre p_k et il faut donner un critère pour arrêter l'itération.

Choix du "shift"-paramètre.

On a plusieurs possibilités :

— $p_k = t_{n,n}^{(k)}$: ce choix marche très bien si les valeurs propres de la matrice sont réelles.

— on considère la matrice

$$\begin{pmatrix} t_{n-1,n-1}^{(k)} & t_{n-1,n}^{(k)} \\ t_{n,n-1}^{(k)} & t_{n,n}^{(k)} \end{pmatrix} \tag{4.40}$$

Si les valeurs propres de (4.40) sont réelles, on choisit pour p_k celle qui est la plus proche de $t_{n,n}^{(k)}$. Si elles sont de la forme $\alpha \pm i\beta$ avec $\beta \neq 0$ (donc complexes), on prend d'abord $p_k = \alpha + i\beta$ et pour l'itération suivante $p_{k+1} = \alpha - i\beta$

4.9.5 Critère pour arrêter l'itération.

L'idée est d'itérer jusqu'à ce que $t^{(k)}_{n,n-1}$ ou $t^{(k)}_{n-1,n-2}$ soit suffisamment petit. Plus précisément, on arr ête l'itération quand

$$t^{(k)}_{l,l-1} \leq eps.(\left|t^{(k)}_{l-1,l-1}\right| + \left|t^{(k)}_{l,l}\right|) \qquad \text{pour} \qquad l=n \qquad \text{ou} \qquad l=n-1 \tag{4.41}$$

— Si (4.41) est vérifié pour $l=n$ on accepte $t^{(k)}_{n,n}$ comme approximation de λ_n et on continue l'ité ration avec la matrice $\left(t^{(k)}_{i,j}\right)_{1\leq i,j\leq n-1}$

— Si (4.41) est vérifié pour $l=n-1$, on accepte les deux valeurs propres de (4.40) comme approximations de λ_n et λ_{n-1} et on continue l'itération avec la matrice $\left(t^{(k)}_{i,j}\right)_{1\leq i,j\leq n-2}$

■ **Exemple 4.3** Nous avons appliqué l'algorithme QR à la matrice (4.40) avec le shift $p_k = t^{(k)}_{n,n}$. La convergence de $t^{(k)}_{i+1,i}$ vers z éro est illustrée dans la figure V.5. Une comparaison avec la figure V.4 nous montre que la convergence est beaucoup plus rapide (convergence quadratique). Après 5 itérations, on a $\left|t^{(k)}_{4,3}\right| \leq 10^{-15}$. Encore 4 itérations pour la matrice de dimension 3 donnent $\left|t^{(k)}_{3,2}\right| \leq 10^{-15}$ Il ne reste plus que 3 itérations à faire pour la matrice de dimension 2 pour avoir $\left|t^{(k)}_{2,1}\right| \leq 10^{-15}$. En tout, 12 itérations ont donné toutes les valeurs propres avec une précision de 15 chiffres. ■

4.9.6 Le "double shift" de Francis

Dans la situation où A est une matrice réelle ayant des valeurs propres complexes, il est recommandé de choisir un shift-paramètre p_k qui soit complexe. Une application directe de l'algorithme préc édent nécessite un calcul avec des matrices complexes. L'observation suivante permet d'éviter ceci.

Proposition 4.9.2 Soit T_k une matrice réelle, $p_k = \alpha + i\beta$ et $p_{k+1} = \alpha - i\beta$. Alors, on peut choisir les décompositions dans l'algorithme QR de manière à ce que T_{k+2} soit réelle.

R La décomposition QR d'une matrice est unique sauf qu'on peut remplacer QR par $(QD)^{-1}\left(D^{-1}R\right)$ où $D = diag(d_1,...,d_n)$ avec $|d_i| = 1$.

Démonstration. La formule (4.39) montre que

$$T_{k+2} = (Q_{k+1}Q_{k+2})^* T_k (Q_{k+1}Q_{k+2}) \tag{4.42}$$

■

Il suffit alors de démontrer que le produit $Q_{k+1}Q_{k+2}$ est ré el. Une manipulation à l'aide de formules pour T_k donne

$$\begin{aligned} Q_{k+1}Q_{k+2}R_{k+2}R_{k+1} &= Q_{k+1}(T_{k+1}-p_{k+1}I)R_{k+1} = Q_{k+1}(R_{k+1}Q_{k+1}+p_{k+1}I-p_kI)R_{k+1} = \\ &= (Q_{k+1}R_{k+1})^2 + (p_k-p_{k+1})Q_{k+1}R_{k+1} = (T_k-p_kI)^2 + (p_k-p_{k+1})(T_k-p_kI) = \\ &= T_k^2 - (p_k+p_{k+1})T_k + p_kp_{k+1}I = M \end{aligned} \tag{4.43}$$

On a donc trouvé une décomposition QR de la matrice M qui, en cons équence des hypothèses du lemme, est une matrice réelle. Si, dans l'algorithme QR, la décomposition est choisie de manière à ce que les éléments diagonaux de R_{k+1} et R_{k+2} soient r éels, alors, à cause de l'unicité de la décomposition QR, les matrices $Q_{k+1}Q_{k+2}$ et $R_{k+2}R_{k+1}$ sont réelles. Une possibilité de calculer

T_{k+2} à partir de T_k est de calculer de (4.43), de faire une décomposition QR (réelle) de M et de calculer T_{k+2} à l'aide de (4.42). Cet algorithme n'est pas pratique car le calcul de T_k^2 nécessite $O(n^3)$ op érations, même si T_k est sous forme de Hessenberg. Il y a une astuce intéressante pour obtenir T_{k+2} à partir de T_k en $O(n^2)$ opérations. Elle est basée sur la propriét é suivante.

Théoreme 4.9.3 Soit une matrice donnée et supposons que

$$Q^*TQ = S \tag{4.44}$$

où Q est orthogonale et S est sous forme de Hessenberg satisfaisant $s_{i,i-1} \neq 0$ pour $i = 2,...,n$ Alors, Q et S sont déterminé es de manière "unique" par la première colonne de Q.

R On a "unicité" dans le sens suivant : si $\hat{Q}^*T\hat{Q}$ est de type Hessenberg avec une matrice orthogonale $\hat{Q}$ satisfaisant $\hat{Q}e_1$, alors $\hat{Q} = QD$ où $D = diag(d_1,...,d_n)$ avec $|d_i| = 1$.

Démonstration. Notons les colonnes de Q par q_i. Alors, la relation (4.44) implique

$$Tq_i = \sum_{j=1}^{i+1} s_{ji}q_j, \qquad q_j^*Tq_i = s_{ji}. \tag{4.45}$$

Si q_1 est fixé, la valeur s_{11} est donnée par la deuxiè me formule de (4.45). Avec cette valeur, on obtient de la premiè re formule de (4.45) que q_2 est un multiple de $Tq_1 - s_{11}q_1$. Ceci détermine q_2 à une unité prè s. Maintenant, les valeurs s_{21}, s_{12}, s_{22} sont déterminées et q_3 est un multiple de $Tq_2 - s_{21}q_1 - s_{22}q_2$ etc. ■

Si les hypothèses du lemme précédent sont vérifiées, on peut calculer la matrice réelle T_{k+2} en $O(n^2)$ opérations de la manière suivante :

— calculer $M\varepsilon_1$, la première colonne de M (formule (4.43)) ;
— déterminer une matrice de Householder H_1 telle que $H_1(M\varepsilon_1) = \alpha e_1$
— transformer $H_1^T T_k H_1$ sous forme de Hessenberg à l'aide de matrices de Householder $H_2,...,H_{n-1}$ (voir le paragraphe V.3) ; c'est-à-dire., calculer $H^T T_k H$ où $H = H_1H_2....H_{n-1}$.

Comme $H_ie_1 = e_1$ pour $i = 2,...,n-1$, la première colonne de H est un multiple de celle de M (observer $H_1^T = H_1$). Par la formule (4.43), la première colonne de $Q_{k+1}Q_{k+2}$ est aussi un multiple de Me_1. Par conséquent, pour un bon choix des dé compositions $Q_{k+1}R_{k+1}$ et $Q_{k+2}R_{k+2}$ on a $H = Q_{k+1}Q_{k+2}$ la matrice obtenue par cet algorithme est égale à T_{k+2} (voir (4.42)).

4.9.7 Etude de la convergence

Supposons d'être déjà proche de la limite et considérons, par exemple, la matrice

$$T_0 = A = \begin{pmatrix} 2 & a \\ \varepsilon & 1 \end{pmatrix}$$

où ε est un nombre petit. Avec le choix $p_0 = 1$ pour le shift-paramètre, on obtient

$$T_0 - p_0 I = \begin{pmatrix} 1 & a \\ \varepsilon & 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{1+\varepsilon^2}} & -\frac{\varepsilon}{\sqrt{1+\varepsilon^2}} \\ \frac{\varepsilon}{\sqrt{1+\varepsilon^2}} & \frac{1}{\sqrt{1+\varepsilon^2}} \end{pmatrix} = \begin{pmatrix} \sqrt{1+\varepsilon^2} & \frac{a}{\sqrt{1+\varepsilon^2}} \\ 0 & -\frac{a\varepsilon}{\sqrt{1+\varepsilon^2}} \end{pmatrix} = Q_1R_1$$

et

$$T_0 - p_0 I = R_1Q_1 = \begin{pmatrix} * & * \\ -\frac{a\varepsilon^2}{1+\varepsilon^2} & * \end{pmatrix}$$

— si A est symétrique (c'est-à-dire, $a = \varepsilon$) on a $t^{(1)}_{n,n-1} = O(\varepsilon^2)$, donc convergence *cubique*.
— si A n'est pas symétrique (p.ex. on a donc convergence quadratique.
Ces propriétés restent vraies pour des matrices générales (sans démonstration).

4.10 EXERCICES

Exercice 4.1 Calculer les valeurs propres de la matrice tridiagonale (dimension $n, b.c > 0$)

$$A = \begin{pmatrix} a & c & & & \\ b & a & c & & \\ & b & a & c & \\ & & b & \ddots & \ddots \\ & & & \ddots & \end{pmatrix}$$

Indication. Les composants du vecteur propre $(v_1, v_2, ..., v_n)^T$ satisfont une équation aux différences finies avec.$v_0 = v_{n+1} = 0$ Vérifier que $v_j = Const.(\alpha_1^j - \alpha_2^j)$ où

$$\alpha_1 - \alpha_2 = \frac{\lambda - a}{c}, \qquad \alpha_1.\alpha_2 = \frac{b}{c}, \qquad \left(\frac{\alpha_1}{\alpha_2}\right)^{n+1} = 1$$

Résultat : $\lambda_j = a - 2\sqrt{bc}.\cos\left(\frac{j\pi}{n+1}\right), \qquad j = 1, 2, .., n.$ ■

Exercice 4.2 Considérer la matrice

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

cette matrice possède une valeur propre de la forme

$$\lambda(\varepsilon) = i + \varepsilon.d + O(\varepsilon^2)$$

Calculer d et dessiner la tangente à la courbe $\lambda(\varepsilon)$ au point $\lambda(0)$
(a) Calculer par la méthode de la puissance, la plus grande valeur propre de la matrice

$$A = \begin{pmatrix} 99 & 1 & 0 \\ 1 & 100 & 1 \\ 0 & 1 & 98 \end{pmatrix}$$

(b) Pour accélerer considérablement la vitesse de convergence, appliquer la méthode de la puissance à la matrice $A - pI$ avec un choix intelligent de p.
(c) Avec quel choix de p obtient-on la valeur propre la plus petite ?

■

Exercice 4.3 Considérons la matrice tridiagonale

$$A = \begin{pmatrix} b_1 & c_1 & & \\ a_1 & b_2 & c_2 & \\ & a_2 & & \\ & & \ddots & \end{pmatrix}$$

Montrer que, si $a_i c_i > 0$ pour $i = 1, \ldots, n-1$, toutes les valeurs propres de A sont réelles. *Indication.* Trouver $D = diag(d_1, \ldots, d_n)$ telle que DAD^{-1} soit symétrique. ■

Exercice 4.4 Soit A une matrice symétrique et B quelconque. Montrer que pour chaque valeur propre λ_B de B il existe une valeur propre λ_A de A telle que

$$|\lambda_A - \lambda_B| \leq \|A - B\|_2 .$$

Indication. Montrer l'existence d'un vecteur v tel que $v = (A - \lambda_B)^{-1} (A - B) v$. En déduire que $1 \leq \left\| (A - \lambda_B)^{-1} (A - B) \right\| \leq \left\| (A - \lambda_B)^{-1} \right\| \|(A - B)\|$. ■

Exercice 4.5 (Schur, 1909). Soit A une matrice symétrique. Montrer que pour chaque indice i il existe une valeur propre λ de A telle que

$$|\lambda - a_{ii}| \leq \sqrt{\sum_{j \neq i} |a_{ij}|^2}$$

Indication. Appliquer l'exercice 5 avec une B convenable. ■

Exercice 4.6 Soit A une matrice réelle avec pour valeur propre $\alpha + i\beta$. Montrer que l'itération

$$\begin{pmatrix} \bar{\alpha} I - A & -\bar{\beta} I \\ \bar{\beta} I & \bar{\alpha} I - A \end{pmatrix} \begin{pmatrix} u_{k+1} \\ v_{k+1} \end{pmatrix} = \begin{pmatrix} u_k \\ v_k \end{pmatrix}$$

où $\bar{\alpha} \approx \alpha$ et $\bar{\beta} \approx \beta$) permet de calculer la valeur propre $\alpha + i\beta$ et le vecteur propre correspondant. Indication. Considérer les parties réelles et complexes de l'itération de Wielandt. On obtient alors

$$\frac{u_k^T A u_k + v_k^T A v_k}{u_k^T u_k + v_k^T v_k} \longrightarrow \alpha, \qquad \frac{u_k^T A v_k + v_k^T A u_k}{u_k^T u_k + v_k^T v_k} \longrightarrow \beta$$

■

Exercice 4.7 Considérons la matrice de Hilbert,

$$A = \begin{pmatrix} 1/2 & 1/3 & 1/4 \\ 1/3 & 1/4 & 1/5 \\ 1/4 & 1/5 & 1/6 \end{pmatrix}$$

(a) Transformer A en une matrice tridiagonale ayant les mêmes valeurs propres.
(b) En utilisant une suite de Sturm, montrer que toutes les valeurs propres sont positives et

qu'une valeur propre est plus petite que 0.001

(c) Calculer approximativement la condition de A pour la norme Euclidienne.

■

Exercice 4.8 La formule de récurrence

$$(k+1)\,P_{k+1}(x) = (2k+1)\,xP_k(x) - kP_{k-1}(x)$$

pour les *polynômes de Legendre* ressemble à

$$p_i(\lambda) = (d_i - \lambda)\,p_{i-1}(\lambda) - \varepsilon_i^2 p_{i-2}(\lambda), \qquad i = 2,\ldots,n.$$

pour les polynômes $\det(A_i - \lambda I)$. Trouver une matrice tridiagonale A de dimension n telle que les valeurs propres de A sont les racines de $P_n(x)$. ■

Exercice 4.9 Soit $p(x)$ un polynôme de degré n et supposons que toutes les racines soient simples. Démontrer que la suite définie par l'algorithme d'Euclide,

$$\begin{aligned} p_n(x) &= p(x), \qquad p_{n-1}(x) = -p'(x) \\ p_i(x) &= q_i(x)p_{i-1}(x) - \gamma_i^2 p_{i-2}(x), \qquad i = n,\ldots,2. \end{aligned}$$

est une suite de Sturm. Pour le polynôme $p(x) = x^5 - 6x^4 + 3x^3 + 3x^2 + 2x + 8$.

(a) déterminer le nombre de racines réelles.

(b) Combien de racines sont complexes ?

(c) Combien de racines sont réelles et positives ?

■

Exercice 4.10 Pour un φ donné notons $c = \cos\varphi$ et $s = \sin\varphi$. La matrice Ω_{kl}, définie par

$$(\Omega_{kl})_{ij} = \begin{cases} 1 & \text{si } i = j,\ j \neq k,\ j \neq l \\ c & \text{si } i = j = k, \text{ ou } i = j = l \\ s & \text{si } i = k, \text{ et } j = l \\ -s & \text{si } i = l, \text{ et } j = k \\ 0 & \text{sinon} \end{cases}$$

s'appelle rotation de Givens.

(a) Montrer qu'elle est orthogonale.

(b) Soit A une matrice symétrique.Déterminer φ tel que le (k,l)-ième élément de $A' = \Omega_{kl} A \Omega_{kl}^T$ s'annule.

Resultat. $\cot 2\varphi = (a_{kk} - a_{ll})\,/\,(2a_{kl})$. ■

Exercice 4.11 La méthode de Jacobi (1846) pour le calcul des valeurs propres d'une matrice symétrique :

i) on choisit a_{kl} $(k > l)$ tel que $|a_{kl}| = \max_{i>j} |a_{ij}|$;

ii) on détermine A' comme dans l'exercice 11.

Montrer que, si on répète cette procédure, on a convergence vers une matrice diagonale, dont les éléments sont les valeurs propres de A Indication. Montrer que $\sum_{i>j} \left|a'_{ij}\right|^2 = \sum_{i>j} |a_{ij}|^2 - |a_{kl}|^2$

■

Exercice 4.12 On considère la matrice

$$A = \begin{pmatrix} 7 & 0,5 \\ 0,0001 & 8 \end{pmatrix}$$

dont on cherche à calculer les valeurs propres.
(a) Faire une itération de l'algorithme QR sans shift.
(b) Faire une itération de l'algorithme QR avec shift.
(c) Estimer la position des valeurs propres de A à l'aide du Th éorème de Gershgorin.
(d) Calculer les valeurs propres de A à l'aide du polynôme caractéristique.

■

Exercice 4.13 Montrer que si la matrice $T_0 = A$ est une matrice de Hessenberg (ou tridiagonale), alors les matrices T_k, $k \geq 1$ construites par l'algorithme QR sont également des matrices de Hessenberg (tridiagonales). ■

Exercice 4.14 Donner une estimation grossière du nombre d'opérations qui sont n écessaires pour effectuer la décomposition QR d'une matrice de Hessenberg et pour calculer ensuite le produit RQ. ■

Exercice 4.15 Soit T_0 une matrice de Hessenberg dont tous les éléments de la sous-diagonale sont non-nuls. Montrer que, si p_0 est une valeur propre de T_0, une itération de l'algorithme QR avec shift p_0 donne

$$t^{(1)}_{n,n-1} = 0.$$

■

Exercice 4.16 Expliquer, comment le calcul de T_k à partir de T_{k-1}

$$Q_k R_k = T_{k-1} - p_{k-1} I, \qquad T_k = R_k Q_k + p_{k-1} I.$$

peut être effectué sans soustraire (et additionner) explicitement la matrice $p_{k-1} I$ ■