# ALGEBRE AVANCEE

Pr. Nadia IDRISSI FATMI

Département Mathématique-Informatique ENSA KHOURIBGA

# Table des matières

# Chapitre 1

# Arithmétique des ordinateurs et analyse d'erreurs

Dans tout ce cours, nous manipulerons des nombres réels. L'objet de ce premier chapitre est de décrire comment ces nombres réels sont représentés dans un ordinateur.

## 1.1 L'arithmétique flottante

### 1.1.1 Le système des nombres à virgule flottante

Théorème 1.1. Soit $\beta$ un entier strictement supérieur à 1. Tout nombre réel $x$ non nul peut se représenter sous la forme

$$x = sgn(x)\beta^e \sum_{k\geq 1} \frac{d_k}{\beta^k}$$

où $sgn(x) \in \{+,-\}$ est le signe de $x$, les $d_k$ sont des entiers tels que $0 < d_1 \leq \beta - 1$ et $0 \leq d_k \leq \beta - 1$ pour $k \geq 2$, et $e \in Z$. De plus, cette écriture est unique (sauf pour les décimaux : $2,5 = 2,499999\ldots$

D'ordinaire, nous utilisons le système décimal, i.e., $\beta = 10$ et les chiffres 0,1,2,3,4,5,6,7 8,9 pour les $d_k$. Nous avons par exemple :

- $0,0038 = 0,38.10^{-2} = +10^{-2}\left(\frac{3}{10} + \frac{8}{10^2}\right)$

Remarque : en MATLAB, on peut écrire $0.38\mathrm{e}-2$ au lieu de $0,38.10^{-2}$

- $\frac{1}{7} = 0,142857\ldots = +10^0\left(\frac{1}{10} + \frac{4}{10^2} + \frac{2}{10^3} + \frac{8}{10^4} + \cdots\right)$. Notons que le développement décimal d'un nombre rationnel est périodique (ici, $\frac{1}{7} = 0,142857142857142857\ldots\Big)$
- $-\sqrt{2} = -1,4142\ldots = -10^1\left(\frac{1}{10} + \frac{4}{10^2} + \frac{1}{10^3} + \frac{4}{10^4} + \cdots\right)$

- $\pi = 3,14159\ldots = +10^1\left(\frac{3}{10} + \frac{1}{10^2} + \frac{4}{10^3} + \frac{1}{10^4} + \cdots\right)$

Historiquement, le choix $\beta = 10$ est lié à une particularité anatomique de la race humaine (nous avons 10 doigts). Les ordinateurs utilisent quant à eux $\beta = 2$ (numération binaire), $\beta = 8$ (numération octale), ou encore $\beta = 16$ (numération hexadécimale).

Remarquons que l'on perd l'unicité si on autorise $d_1 = 0$ : en effet, on a par exemple :

$$\begin{aligned} 0,0038 \qquad &= 0,38.10^{-2} = +10^{-2}\left(\tfrac{3}{10} + \tfrac{8}{10^2}\right) \\ &= 0,038.10^{-3} = 10^{-1}\left(\tfrac{0}{10} + \tfrac{3}{10^2} + \tfrac{8}{10^3}\right) \end{aligned}$$

On définit l'ensemble $F \subset R$ par :

$$F = \left\{ y \in R \mid y = \pm\beta^e \left(\frac{d_1}{\beta} + \frac{d_2}{\beta^2} + \cdots + \frac{d_t}{\beta^t}\right), e_{\min} \leq e \leq e_{\max} \right\}$$

ou encore

$$F = \left\{ y \in R \mid y = \pm m\beta^{e-t}, e_{\min} \leq e \leq e_{\max} \right\}$$

Ceci correspond aux deux écritures $0,0038 = +10^{-2}\left(\frac{3}{10} + \frac{8}{10^2}\right) = +38.10^{-4}$ avec $e = -2$ $t = 2, e - t = -4$. Le nombre $m$ s'appelle la mantisse et on utilise la notation $m = \overline{d_1 d_2 \ldots d_t}^{\beta}$.

Notons que $0 \notin F$.

Pour $y \neq 0$, on a

$$m\beta^{e-t} = \beta^e \left(\frac{d_1}{\beta} + \frac{d_2}{\beta^2} + \cdots + \frac{d_t}{\beta^t}\right) \geq \beta^e \frac{1}{\beta}$$

car $d_1 \geq 1$.

D'où $m \geq \beta^{t-1}$. D'autre part

$$m = \overline{d_1 d_2 \dots d_t}^{\beta} = d_1\beta^{t-1} + \cdots + d_{t-k}\beta^k + \cdots + d_{t-1}\beta + d_t < \beta^t.$$

On a donc montré que $\beta^{t-1} \leq m < \beta^t$

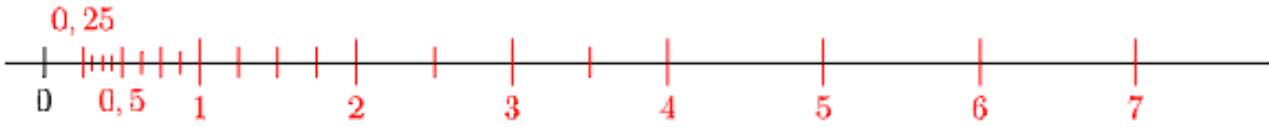
$F$ est un système de nombres à virgule flottante (floating point number system) noté $F(\beta, t, e_{\min}, e_{\max})$. Il dépend de quatre paramètres :

1. la base $\beta$ (chiffres utilisés $0, 1, \dots, \beta - 1$),

2. la précision $t$ (nombre de chiffres utilisés pour représenter la mantisse),

3. $e_{\min}$ et $e_{max}$ qui définissent le domaine des exposants.

Par exemple, pour $F(2, 3, -1, 3)$, on obtient les nombres représentés en rouge sur la figure ci-dessous :




On constate que l'écart entre deux nombres consécutifs est multiplié par 2 à chaque puissance de 2 .

Dans le standard IEEE 754 utilisé par MATLAB, on a $\beta = 2$ et :

• en simple précision : $t = 24, e_{\min} = -125, e_{\max} = 128$

• en double précision : $t = 53, e_{\min} = -1021, e_{\max} = 1024$

Définition 1.2. On appelle epsilon machine et on note $\epsilon_M$ la distance de 1

au nombre flottant suivant.

Par exemple, pour $F(2,3,-1,3)$, on a $\epsilon_M = 0,25$. Dans MATLAB, c'est eps.

Proposition 1.3. Pour $F(\beta,t,e_{\min},e_{\max})$, on a $\epsilon_M = \beta^{1-t}$.

Démonstration. On a $1 = \beta\frac{1}{\beta} = \beta^{1-t}\overline{10\ldots0}^{\beta}$. Le nombre suivant dans le système de nombres à virgule flottante $F(\beta,t,e_{\min},e_{\max})$ est alors $\beta^{1-t}\overline{10\ldots1}^{\beta} = \beta^{1-t}\left(\beta^{t-1}+1\right) = 1+\beta^{1-t}$.

Lemme 1.4. Dans le système de nombres à virgule flottante $F(\beta,t,e_{\min},e_{\max})$, l'écart $|y-x|$ entre un nombre flottant $x$ (non nul) et un nombre flottant $y$ (non nul) adjacent vérifie $\beta^{-1}\epsilon_M|x| \leq |y-x| \leq \epsilon_M|x|$.

Démonstration. On a $x = m\beta^{e-t}$, donc $|y-x| = 1\beta^{e-t}$. Or $\beta^{t-1} \leq m < \beta^t$ donc $m\beta^{-t} < 1$ et $m\beta^{1-t} \geq 1$. Il vient donc $m\beta^{-t}\beta^{e-t} < 1\beta^{e-t} \leq m\beta^{1-t}\beta^{e-t}$ d'où le résultat puisque $\epsilon_M = \beta^{1-t}$.

### 1.1.2 Représentation effective des réels et sa formalisation

• Représentation "physique" : par exemple, en simple précision 32 bits (bit = binary digit), 8 bits sont réservés à l'exposant et 24 bits (dont 1 pour le signe) à la mantisse. En double précision 64 bits, 11 bits sont réservés à l'exposant et 53 bits (dont 1 pour le signe) à la mantisse.

• Arrondi :

1. par troncature : par exemple avec 3 chiffres, $0,8573\ldots$ devient 0,857 .

2. au plus près : $0,8573\ldots$ devient 0,857 .

3. au représentant le plus proche dont la dernière décimale est paire (rounding to even) : $0,8573\ldots$ devient 0,858

• Formalisation :

Définition 1.5. Soit $G = G(\beta, t) = \{y \in R \mid y = \pm m\beta^{e-t}\}$ sans conditions sur l'exposant $e$. $L$ 'application $\mathrm{fl} : R \to G, x \mapsto \mathrm{fl}(x)$ est appelée opération d'arrondi.

Étant donné un domaine $F(\beta, t, e_{\min}, e_{\max})$ , il y a alors dépassement de capacité si :

1. $|\mathrm{fl}(x)| > \max\{|y| \mid y \in F\}$ : On parle d' overflow

2. $|\mathrm{fl}(x)| < \min\{|y| \mid y \in F\}$ : On parle d'underflow

sinon, $x$ est dans le domaine de $F$.

### 1.1.3 Unité d'erreur d'arrondi-Estimation d'erreurs

Définition 1.6. Soit $x$ un réel et $\bar{x}$ une valeur approchée de $x$. $L$ 'erreur absolue e est défini par $e = |x - \bar{x}|$, l'erreur relative est $\left|\frac{1}{x}\right|$. Le pourcentage d'erreur est l'erreur relative multipliée par 100.

En pratique, on ne connaît en général pas la valeur exacte $x$ (c'est le cas dans la plupart des mesures physiques) mais on peut souvent avoir une idée de l'erreur maximale e que l'on a pu commettre : dans ce cas, on majore la

quantité $\left|\frac{c}{7}\right|$.

Théorème 1.7 (Estimation de l'erreur d'arrondi). Soit $x$ un réel. Si $x$ est dans le domaine $F(\beta, t, e_{\min}, e_{\max})$, alors il existe $\delta \in R$ avec $|\delta| < \mathbf{u} = \frac{1}{2}\beta^{1-t} = \frac{1}{2}\epsilon_M$ tel que $\mathrm{fl}(x) = x(1+\delta)$.

Démonstration. Admis pour ce cours.

L'erreur relative sur l'arrondi est égale à $|\delta| < u$ : le nombre $u$ s'appelle unité d'erreur d'arrondi.

Par exemple, dans le standard IEEE 754 utilisé par MATLAB, on a $u = 2^{-24} \approx 5,96.10^{-8}$ en simple précision et $u = 2^{-53} \approx 1,11.10^{-16}$ en double précision.

### 1.1.4 Modèle de l'arithmétique flottante

Le modèle suivant de l'arithmétique flottante est celui utilisé par le standard IEEE.

**Modèle Standard** : Soit $x, y \in F(\beta, t, e_{\min}, e_{\max})$. Pour op $\in \{+, -, x, \div, \sqrt{}\}$, on définit $x$ [op] $y = fl(x \text{ op } y) = (x \text{ op } y)(1+\delta)$ avec $|\delta| < u = \frac{1}{2}\beta^{1-t} = \frac{1}{2}\epsilon_M$.

Nous allons maintenant nous intéresser aux erreurs faites par [op].

## 1.2 L'analyse d'erreurs

### 1.2.1 Non-associativité

En général, contrairement à op, l'opération [op] n'est pas associative. Ceci est dû aux erreurs d'arrondi. Par exemple, supposons que les réels soient calculés avec 3 chiffres significatifs et arrondis à la décimale la plus proche et cherchons à calculer la somme $x[+]y[+]z$ avec $x = 8,22, y = 0,00317$ et $z = 0,00432$.

•$x[+]y = 8,22$ donc $(x[+]y)[+]z = 8,22$

•$y[+]z = 0,01$ donc $x[+] + (y[+]z) = 8,23$.

### 1.2.2 Erreurs d'arrondi sur une somme

Supposons que l'on souhaite calculer une somme $S = u_1 + u_2 + \cdots + u_n$ de $n$ réels positifs dans $F(\beta, t, e_{\min}, e_{\max})$. On calcule alors les sommes partielles $S_i$ par la récurrence $S_0 = 0, S_i = S_{i-1} + u_i$. Si l'on suppose les $u_i$ connus exactement, alors les erreurs d'arrondi $\Delta S_i$ commises sur le calcul des sommes partielles $S_i$ vérifient $\Delta S_i \leq \Delta S_{i-1} + \delta(S_{i-1} + u_i) = \Delta S_{i-1} + \delta S_i$ où $|\delta| < u$. L'erreur globale sur $S = S_n$ vérifie donc

$$\Delta S \leq \delta(S_2 + \cdots + S_n)$$

ou encore

$$\Delta S \leq \delta \left( u_n + 2u_{n-1} + 3u_{n-2} + \cdots + (n-1)u_2 + (n-1)u_1 \right)$$

On voit donc que pour minimiser cette erreur on a tout intérêt à sommer d'abord les termes les plus petits (cf exemple de la sous-section précédente).

### 1.2.3 Erreurs d'arrondi sur un produit

Supposons que l'on souhaite calculer un produit $P = u_1 u_2 \dots u_n$ de $n$ réels positifs dans $F(\beta, t, e_{\min}, e_{\max})$. On calcule alors les produits $P_i$ par la récurrence $P_0 = 1, P_i = P_{i-1} u_i$. Si l'on suppose les $u_i$ connus exactement, alors les erreurs d'arrondi $\Delta P_i$ commises sur le calcul des produits $P_i$ vérifient $\Delta P_i \leq \Delta P_{i-1} u_i + \delta (S_{i-1} u_i) = \Delta P_{i-1} u_i + \delta P_i$ où $|\delta| < u$. L'erreur globale sur $P = P_n$ vérifie donc

$$\Delta P \leq (k-1) \delta P_n$$

On voit donc que contrairement au cas de l'addition, la majoration de l'erreur ne dépend pas de l'ordre des facteurs.

### 1.2.4 Phénomènes de compensation

Les phénomènes de compensation sont ceux qui se produisent lorsque l'on tente de soustraire des nombres très proches. Nous illustrons ces phénomènes sur deux exemples et donnons des astuces pour les contourner.

**Exemple 1** : On considère l'expression $E = \sqrt{x+1} - \sqrt{x}$ avec $x > 0$. Sous MATLAB, en calculant $E$ pour $x = 10^9$, on va obtenir $1,5811.10^{-5}$ mais pour $x = 10^{16}$, on va obtenir 0! Si l'on remarque que $E = \frac{1}{\sqrt{x+1}+\sqrt{x}}$, alors en utilisant cette nouvelle formule, on trouvera $E = 1,5811.10^{-5}$ pour $x = 10^9$ et $E = 5,000.10^{-9}$ pour $x = 10^{16}$.

**Exemple 2** : On considère l'équation du second degré $x^2 - 1634x + 2 = 0$. Supposons que les calculs soient effectués avec 10 chiffres significatifs. Les formules habituelles donnent $\Delta' = \left(\frac{1634}{2}\right)^2 - 2 = 667487$, $\sqrt{\Delta'} = 816,9987760$, d'où les solutions

$$x_1 = \frac{1634}{2} + \sqrt{\Delta'} = 817 + 816,9987760 = 1633,998776$$

$$x_2 = \frac{1634}{2} - \sqrt{\Delta'} = 817 - 816,9987760 = 0,0012240$$

On voit donc qu'en procédant ainsi on a une perte de 5 chiffres significatifs sur $x_2$. Pour y remédier, on peut utiliser la relation $x_1 x_2 = 2$ et calculer

$$x_2 = \frac{2}{x_1} = \frac{2}{1633,998776} = 0,001223991125.$$

### 1.2.5 Phénomènes d'instabilité numérique

Les phénomènes d'instabilité numérique sont des phénomènes d'amplification d'erreur d'arrondi. Ils se produisent en général pour des calculs récurrents ou itératifs. Nous illustrons ces phénomènes sur deux exemples.

**Exemple 1** : On considère l'intégrale

$$I_n = \int_0^1 \frac{x^n}{10+x} \mathrm{d}x, \quad n \in N$$

que l'on cherche à évaluer numériquement. Un calcul direct montre que $I_0 = \ln\left(\frac{11}{10}\right)$. De plus, on a

$$I_n = \int_0^1 \frac{x}{10+x} x^{n-1} \mathrm{d}x = \int_0^1 \left(1 - \frac{10}{10+x}\right) x^{n-1} \mathrm{d}x = \frac{1}{n} - 10 I_{n-1}$$

On peut donc calculer successivement les valeurs de $I_n$ en utilisant la récurrence $I_0 = \ln\left(\frac{11}{10}\right)$ $I_n = \frac{1}{n} - 10 I_{n-1}$. Numériquement, cela conduit à des résultats très mauvais. Cela provient du fait que l'erreur d'arrondi $\Delta I_n$ sur le calcul de $I_n$ vérifie $\Delta I_n \approx 10 \Delta I_{n-1}$, si l'on néglige l'erreur d'arrondi sur $\frac{1}{n}$. On voit donc que l'erreur croit exponentiellement : l'erreur sur $I_0$ est multipliée par $10^n$ sur $I_n$. Par conséquent cette formule de récurrence ne peut pas nous permettre de calculer la valeur de $I_{36}$ par exemple. Pour remédier à ce problème, on peut renverser la récurrence c'est-à-dire considérer la formule :

$$I_{n-1} = \frac{1}{10}\left(\frac{1}{n} - I_n\right)$$

Toujours en négligeant l'erreur d'arrondi sur $\frac{1}{n}$, on obtient alors $\Delta I_{n-1} \approx \frac{1}{10}\Delta I_n$. En utilisant l'encadrement $10 \leq 10 + x \leq 11$ pour $x \in [0, 1]$, on montre que

$$\frac{1}{11(n+1)} \leq I_n \leq \frac{1}{10(n+1)}$$

L'approximation $I_n \approx \frac{1}{11(n+1)}$ nous permet alors de calculer une valeur de départ pour notre récurrence renversée. Par exemple, si l'on part de $I_{46} \approx \frac{1}{11(46+1)}$, on obtiendra pour $I_{36}$ une erreur relative meilleure que $10^{-10}$.

On constate ici l'importance du coefficient d'amplification d'erreur pour ce genre de calcul.

**Exemple 2** : On considère la suite définie par :

$$\begin{cases} u_0 = 2 \\ u_1 = -4 \\ u_n = 111 - \frac{1130}{u_{n-1}} + \frac{3000}{u_{n-1}u_{n-2}} \end{cases}$$

introduite par J.M. Muller. On peut alors montrer que la limite de cette suite est égale à 6 et malgré cela, quelque soit le système et la précision utilisés, cette suite semblera tendre vers 100. L'explication de ce phénomène étrange est assez simple : la solution générale de la récurrence $u_n$ est donnée par :

$$u_n = \frac{\alpha 100^{n+1} + \beta 6^{n+1} + \gamma 5^{n+1}}{\alpha 100^n + \beta 6^n + \gamma 5^n}$$

où $\alpha, \beta$ et $\gamma$ dépendent des valeurs initiales $u_0$ et $u_1$. Par conséquent, si $\alpha \neq 0$, la suite converge vers 100 et sinon ($si\beta \neq 0$) la suite converge vers 6. Dans notre exemple, les valeurs initiales $u_0 = 2$ et $u_1 = -4$ correspondent à $\alpha = 0, \beta = -3$ et $\gamma = 4$. Par conséquent, la limite exacte de la suite est 6. Mais à cause des erreurs d'arrondi, même les premiers termes calculés seront différents des termes exacts et donc la valeur de $\alpha$ correspondant à ces termes calculés sera très petite mais non-nulle ce qui suffira à faire en sorte que la suite converge vers 100 au lieu de 6 .

### 1.2.6 Erreur amont - Erreur aval

Considérons un problème que l'on résout à l'aide d'un algorithme numérique et appelons $f$ la fonction qui fait correspondre à l'entrée $x$ de l'algorithme la solution algébrique $y = f(x)$. En pratique, compte tenu des erreurs d'arrondis, étant donnée une entrée $x$, nous allons obtenir une sortie $\bar{y}$ qui sera distincte de la solution algébrique $y = f(x)$. L'erreur aval est alors la différence entre le résultat $\bar{y}$ obtenu et la solution algébrique $y$. L'erreur amont ou erreur inverse est le plus petit $\delta x$ tel que la solution algébrique $f(x+\delta x)$ correspondant à l'entrée $x+\delta x$ soit égale à $\bar{y}$. Ces deux erreurs sont liées par le conditionnement (voir Chapitre 3) : l'erreur aval étant du même

ordre que l'erreur amont multipliée par le conditionnement. L'erreur amont est en général plus intéressante que l'erreur aval car elle nous renseigne sur le problème qui est réellement résolu par l'algorithme numérique. De plus, en pratique, nous ne connaissons en général qu'une valeur approchée de l'entrée (par exemple obtenue par une mesure).

### 1.2.7 Outils théoriques de l'analyse d'erreurs

Considérons la formule $(x \times y) + z$ pour trois réels $x, y$ et $z$ d'un domaine $F(\beta, t, e_{\min}, e_{\max})$.

On a alors :

$$\begin{aligned} \mathrm{fl}((x \times y) + z) &= [\mathrm{fl}(x \times y) + z](1 + \delta_1) \\ &= [(x \times y)(1 + \delta_2) + z](1 + \delta_1) \\ &= (x \times y)(1 + \delta_2)(1 + \delta_1) + z(1 + \delta_1) \end{aligned}$$

d'après le théorème 1.7, avec de plus $|\delta_i| < u$, pour $i = 1, 2$.

Lemme 1.8. Si pour tout $i = 1, \ldots, k$, on $a\,|\delta_i| < u$ et si $ku < 1$, alors il existe $\theta_k$ tel que $|\theta_k| \leq \frac{ku}{1-ku}$ et $\prod_{i=1}^{k}(1 + \delta_i) \leq 1 + \theta_k$.

Démonstration. La démonstration se fait par récurrence.

En utilisant la notation $< k >= \prod_{i=1}^{k}(1 + \delta_i)$ avec $< j > . < k >=< j+k >$,

on a alors

$$\mathrm{fl}((x \times y) + z) = (x \times y) < 2 > + z < 1 > \leq (x \times y)\left(1 + \tfrac{2u}{1-2u}\right) + z\left(1 + \tfrac{u}{1-u}\right).$$

# Chapitre 2

# Résolution d'un Système d'Equations Linéaires : Méthodes Directes

Beaucoup de problèmes se réduisent à la résolution numérique d'un système d'équations linéaires. Il existe deux grandes classes de méthodes pour résoudre ce type de systèmes :

1. les méthodes directes qui déterminent explicitement la solution après un nombre fini d'opérations arithmétiques

2. les méthodes itératives qui consistent à générer une suite qui converge vers la solution du système.

Remarquons que les méthodes itératives ne s'appliquent que dans le cas de systèmes à coefficients dans $R$ ou $C$ mais pas dans le cas des corps finis $F_p$. Notons qu'il existe aussi des méthodes intermédiaires telles que les méthodes de Splitting ou de décomposition incomplètes, et des méthodes probabilistes comme celle de Monte-Carlo qui ne seront pas abordées dans ce cours.

Dans ce deuxième chapitre nous nous intéressons aux méthodes directes. Les méthodes itératives seront abordées au chapitre 4.

## 2.1 Introduction et motivation

### 2.1.1 Objet

On considère un système (S) de $n$ équations à $n$ inconnues de la forme

$$(S)\begin{cases} a_{1,1}x_1 + a_{1,2}x_2 + \cdots & +a_{1,n}x_n = b_1 \\ a_{2,1}x_1 + a_{2,2}x_2 + \cdots & +a_{2,n}x_n = b_2 \\ \vdots & \\ a_{n,1}x_1 + a_{n,2}x_2 + \cdots & +a_{n,n}x_n = b_n \end{cases}$$

Les données sont les coefficients $a_{i,j}$ du système qui appartiennent à un corps $K$ avec $K = R$ ou C ainsi que les coefficients du second membre $b_1, \ldots, b_n$. Les inconnues sont $x_1, \ldots, x_n$ qui appartiennent à $K$. Un système est dit homogène si son second membre est nul c'est-à-dire si tous les $b_i$ sont nuls.

Nous écrirons ce système sous forme matricielle $(S) \quad Ax = b$ avec

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & \ldots & a_{1,n} \\ a_{2,1} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ a_{n,1} & \ldots & \ldots & a_{n,n} \end{pmatrix} \in M_{n\times n}(K), x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \in K^n$$

et $b = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix} \in K^n$

Dans tout ce chapitre, on supposera que la matrice $A$ est inversible !

### 2.1.2 Motivation

Le lecteur peut se demander pourquoi le problème de la résolution d'un tel système se pose alors que les formules de Cramer nous donnent la solution d'un tel système $(S)$ :

$$\forall i \in \{1, \ldots, n\}, \quad x_i = \frac{\begin{vmatrix} a_{1,1} & \ldots & a_{1,(i-1)} & b_1 & a_{1,(i+1)} & \ldots & a_{1,n} \\ \vdots & & & \vdots & & & \\ \vdots & & & & & & \\ a_{n,1} & \ldots & a_{n,(i-1)} & b_n & a_{n,(i+1)} & \ldots & a_{n,n} \end{vmatrix}}{det(A)}$$

Pour comprendre le problème, essayons de compter le nombre d'opérations nécessaires pour calculer la solution en utilisant ces formules de Cramer. Nous devons calculer $n+1$ déterminants de taille $n$. On sait que le déterminant de $A$ est donné par la formule

$$det(A) = \sum_{\sigma \in \mathcal{S}_n} \epsilon_\sigma a_{\sigma(1),1} a_{\sigma(2),2} \cdots a_{\sigma(n),n}$$

où $\mathcal{S}_n$ est l'ensemble des permutations de $\{1, \ldots, n\}$ et $\epsilon_\sigma$ est la signature de la permutation $\sigma$ $(\epsilon_\sigma \in \{-1, 1\})$. L'ensemble $\mathcal{S}_n$ contient $n!$ éléments donc le calcul de $det(A)$ se ramène à $n! - 1$ additions et $n!(n-1)$ multiplications

soit au total $n! - 1 + n!(n-1) = nn! - 1$ opérations à virgule flottante. Comme nous avons $n+1$ déterminants à calculer, le nombre d'opérations nécessaires pour résoudre le système à l'aide des formules de Cramer est de $(n+1)(nn!-1)$ opérations à virgule flottante qui est équivalent quand la dimension $n$ tend vers l'infini à $n(n+1)!$. Essayons d'évaluer cette quantité lorsque $n = 100$. Pour ceci on peut utiliser la formule de Stirling $n! \sim n^{n+\frac{1}{2}}\mathrm{e}^{-n}\sqrt{2\pi}$ qui est très précise pour évaluer la factorielle :

$$\begin{aligned} 100.101! &= 100.101.100! \\ &\simeq 100.101.100^{100,5} \cdot \mathrm{e}^{-100} \cdot \sqrt{2\pi} \\ &\simeq 100.101.100^{100,5} \cdot 10^{-43,43} \cdot \sqrt{2\pi} \quad (\log_{10}(\mathrm{e}) \simeq 0,4343) \\ &\simeq 10^{205-44} \cdot 10^{0,57} \cdot \sqrt{2\pi} \simeq 9,4.10^{161} \end{aligned}$$

Par exemple, avec ordinateur fonctionnant à 100 megaflops (flops = opérations à virgule flottante par secondes - floating point opérations per second), il faudrait environ $3.10^{146}$ années pour résoudre notre système !

On en déduit donc qu'il est impossible d'utiliser les formules de Cramer pour des systèmes de grande taille. En pratique, on utilise les méthodes directes pour des systèmes de dimension peu élevée ($n \leq 100$). Pour de très grands systèmes qui peuvent par exemple apparaître en économie ou pour l'approximation d'équations aux dérivées partielles (voir Chapitre 4 ), on préfère les méthodes itératives.

### 2.1.3 Résolution d'un système triangulaire

L'idée des méthodes directes est de se ramener à la résolution d'un (ou de deux) système(s) triangulaire(s).

Supposons que $A$ soit une matrice triangulaire supérieure. Le système s'écrit alors :

$$(S)\begin{cases} a_{1,1}x_1 + a_{1,2}x_2 + \cdots + a_{1,n}x_n = b_1 \\ \qquad\qquad a_{2,2}x_2 + \cdots + a_{2,n}x_n = b_2 \\ \\ \\ \qquad\qquad\qquad\qquad a_{n,n}x_n = b_n \end{cases}$$

Puisque $A$ est supposée inversible, aucun des $a_{i,i}$ n'est nul et on peut résoudre ce système en utilisant l'algorithme de substitution rétrograde (ou substitution arrière) suivant :

Entrées : $A = (a_{i,j})_{1\leq i,j\leq n} \in M_{n\times n}(K)$ triangulaire supérieure et $b = (b_i)_{1\leq i\leq n} \in K^n$

Sortie : $x = (x_i)_{1\leq i\leq n} \in K^n$ tel que $Ax = b$

1. $x_n = \frac{b_n}{a_{n,n}}$

2. Pour $i$ de $n-1$ à 1 par pas de $-1$, faire :

$$\bullet x_i = \frac{1}{a_{i,i}}\left(b_i - \sum_{j=i+1}^{n} a_{i,j}x_j\right)$$

3. Retourner $x = (x_i)_{1 \leq i \leq n} \in K^n$.

**Exemple** : On considère le système triangulaire supérieur

$$(S) \begin{cases} x_1 + 2x_2 + 5x_3 = 1 \\ -4x_2 - 16x_3 = -\frac{5}{2} \\ -17x_3 = -\frac{17}{8} \end{cases}$$

En résolvant la troisième équation, on trouve $x_3 = \frac{1}{8}$. La deuxième équation donne alors $x_2 = \frac{-5/2+16x_3}{-4} = \frac{1}{8}$ et finalement la première équation fournit $x_1 = \frac{1-2x_2-5x_3}{1} = \frac{1}{8}$.

De façon analogue, lorsque $A$ est triangulaire inférieure, on obtient l'algorithme de substitution progressive (ou substitution avant).

Proposition 2.1. La résolution d'un système d'équations linéaires triangulaire se fait en $n^2$ opérations à virgule flottante.

Démonstration. Calculer $x_n$ nécessite 1 opération (division), calculer $x_{n-1}$ nécessite 3 opérations (une multiplication, une soustraction et une division), et calculer $x_i$ nécessite $2(n-i)+1$ opérations ($n-i$ multiplications, $(n-i-1)$ additions, 1 soustraction et 1 division). Au total, le nombre d'opérations est donc

$$\sum_{i=1}^{n}(2(n-i)+1) = 2\sum_{i=1}^{n}(n-i) + n = 2\sum_{j=0}^{n-1} j + n = 2\frac{(n-1)n}{2} + n = n^2$$

Proposition 2.2. Soient $A, B \in M_{n\times n}(K)$ deux matrices triangulaires supérieures.

*On* a alors les résultats suivants :

1. $AB$ est triangulaire supérieur

2. Si A et $B$ sont à diagonale unité (i.e., n'ont que des 1 sur la diagonale), alors $AB$ est à diagonale unité

3. Si A est inversible, alors $A^{-1}$ est aussi triangulaire supérieure ;

4. Si A est inversible et à diagonale unité, alors $A^{-1}$ est aussi à diagonale unité.

Démonstration. Exercice.

### 2.1.4 Les méthodes directes étudiées

Dans la suite de ce chapitre, nous allons considérer les trois méthodes directes suivantes pour la résolution de $(S) \quad Ax = b$ :

1. Méthode de Gauss : le principe est de réduire le système à $(MA)x = Mb$ avec $MA$ triangulaire supérieure sans calculer explicitement $M$. On se ramène donc à la résolution d'un système triangulaire supérieur. Cette méthode est associée à la factorisation $A = LU$ de la matrice $A$ avec $L$ triangulaire inférieure (Lower) et $U$ triangulaire supérieure (Upper). Étant donnée une telle factorisation $A = LU$, on peut résoudre le système $Ax = b$ en résolvant successivement les deux systèmes triangulaires $Ly = b$ puis $Ux = y$.

En MATLAB, on peut appliquer la méthode de Gauss pour résoudre $(S)$ : $Ax = b$ en tapant $x = A\backslash b$ (attention, le symbole entre la matrice $A$ et le vecteur colonne $b$ est backslash et non un slash !) et calculer la factorisation $LU$ d'une matrice en utilisant$[L, U] = lu(A)$.

2. Méthode de Cholesky associée à la factorisation de Cholesky $A = R^T R$ avec $R$ triangulaire supérieure. Cette méthode est valable pour une matrice $A$ symétrique et définie positive (voir Définitions 2.14 et 2.16 ). On résout alors $Ax = b$ en résolvant successivement les systèmes triangulaires $R^T y = b$ puis $Rx = y$.

En MATLAB, on peut calculer la factorisation de Cholesky d'une matrice $A$ en tapant R = chol(A). On obtient un message d'erreur lorsque $A$ n'est pas symétrique définie positive.

3. Méthode de Householder associée à la factorisation $A = QR$ avec $R$ triangulaire supérieure et $Q$ orthogonale (voir Définition 2.20 ). La matrice $Q$ est un produit de $n - 1$ matrices de Householder $\%H_i$ (voir Définition 2.19 ). Le système $Ax = b$ s'écrit alors $H_{n-1} \cdots H_2 H_1 Ax = H_{n-1} \cdots H_2 H_1 b$ que l'on résout facilement grâce au fait que le produit $H_{n-1} \cdots H_2 H_1 A$ est une matrice triangulaire supérieure.

En MATLAB, on calcule la factorisation QR d'une matrice $A$ en tapant $[Q, R] = qr(A)$.

## 2.2 Méthode de Gauss et factorisation LU

### 2.2.1 Description de la méthode

On considère le système linéaire $(S) : Ax = b$ en supposant toujours que $A$ est inversible et on pose $b^{(1)} = b$ et $A^{(1)} = A = \left(a_{i,j}^{(1)}\right)_{1\leq i,j\leq n}$. Le système $(S)$ s'écrit alors $A^{(1)}x = b^{(1)}$ que l'on note $S^{(1)}$

**Étape 1** : Puisque $A$ est inversible, quitte à échanger la première ligne de $A^{(1)}$ avec une autre, on peut supposer que $a_{1,1}^{(1)} \neq 0$. Le nombre $a_{1,1}^{(1)}$ est le premier pivot de l'élimination de Gauss. Pour $i = 2, \ldots, n$, on multiplie la première équation de $\left(S^{(1)}\right)$ par $g_{i,1} = \frac{a_{i,1}^{(1)}}{a_{1,1}^{(1)}}$ et on retranche l'équation obtenue à la ième équation de $\left(S^{(1)}\right)$. La ième ligne $L_i$ de $\left(S^{(1)}\right)$ devient donc $L_i - g_{i,1}L_1$. On obtient alors un nouveau système $\left(S^{(2)}\right) : A^{(2)}x = b^{(2)}$ avec :

$$\begin{cases} a_{1,j}^{(2)} = a_{1,j}^{(1)}, \quad j = 1, \ldots, n \\ a_{i,1}^{(2)} = 0, \quad i = 2, \ldots, n \\ a_{i,j}^{(2)} = a_{i,j}^{(1)} - g_{i,1}a_{1,j}^{(1)}, \quad i, j = 2, \ldots, n \\ b_1^{(2)} = b_1^{(1)} \\ b_i^{(2)} = b_i^{(1)} - g_{i,1}b_1^{(1)}, \quad i = 2, \ldots, n \end{cases}$$

La matrice $A^{(2)}$ et le vecteur $b^{(2)}$ sont donc de la forme :

$$A^{(2)} = \begin{pmatrix} a_{1,1}^{(1)} & a_{1,2}^{(1)} & \dots & a_{1,n}^{(1)} \\ 0 & a_{2,2}^{(2)} & \dots & a_{2,n}^{(2)} \\ 0 & \vdots & \vdots & \\ \vdots & \vdots & & \\ 0 & a_{n,2}^{(2)} & \dots & a_{n,n}^{(2)} \end{pmatrix}, \quad b^{(2)} = \begin{pmatrix} b_1^{(1)} \\ b_2^{(2)} \\ \vdots \\ b_n^{(2)} \end{pmatrix}$$

**Étape k** : On a ramené le système à $\left(S^{(k)}\right) : A^{(k)}x = b^{(k)}$ avec

$$
\mathrm{A}^{(k)} = \begin{pmatrix}
a_{1,1}^{(1)} & & \dots & \dots & a_{1,k}^{(1)} & \dots \\
a_{1,n}^{(1)} & & & & & \\
0 & a_{2,2}^{(2)} & & & a_{2,k}^{(2)} & \dots \\
a_{2,n}^{(2)} & & & & & \\
0 & 0 & a_{3,3} & & a_{3,k}^{(3)} & \dots \\
a_{3,n}^{(3)} & & & & & \\
\vdots & \ddots & \ddots & \ddots & \vdots & \\
\vdots & & & & & \\
0 & \dots & 0 & 0 & a_{k,k}^{(k)} & \dots \\
a_{k,n}^{(k)} & & & & & \\
\vdots & & \vdots & 0 & a_{k+1,k}^{(k)} & \dots \\
a_{k+1,n}^{(k)} & & & & & \\
\vdots & & \vdots & \vdots & \vdots & \\
\vdots & & & & & \\
0 & \dots & 0 & 0 & a_{n,k}^{(k)} & \dots \\
a_{n,n}^{(k)} & & & & &
\end{pmatrix}.
$$

On peut alors se ramener au cas $a_{k,k}^{(k)} \neq 0$ et $a_{k,k}^{(k)}$ est le kième pivot de l'élimination de Gauss. Par le mème principe qu'à l'étape 1 et en utilisant

$g_{i,k} = \frac{a_{i,k}^{|\lambda|}}{a_{k,k}^{(\lambda)}}$ pour $i > k$, on obtient alors $\left(S^{(k+1)}\right) : A^{(k+1)}x = b^{(k+1)}$ avec

$$A^{(k+1)} = \begin{pmatrix}
a_{1,1}^{(1)} & & \dots & \dots & a_{1,k+1}^{(1)} & \dots & \dots \\
a_{1,n}^{(1)} & & & & & & \\
0 & a_{22}^{(2)} & & & a_{26}^{(2)} & \dots & \dots \\
a_{2,n}^{(2)} & & & & & & \\
0 & 0 & a_{3,3}^{(3)} & & a_{3k}^{(3)} & \dots & \dots \\
a_{3,n}^{(3)} & & & & & & \\
\vdots & \ddots & \ddots & \ddots & \vdots & & \\
\vdots & & & & & & \\
0 & \dots & 0 & 0 & a_{i,i}^{(k)} & \dots & \dots \\
a_{k,n}^{(k)} & & & & & & \\
\vdots & & \vdots & 0 & 0 & a_{k+1,k+1}^{(k+1)} & \dots \\
a_{k+1,n}^{(k+1)} & & & & & & \\
\vdots & & \vdots & \vdots & \vdots & \vdots & \\
\vdots & & & & & & \\
0 & \dots & 0 & 0 & 0 & a_{n,k+1}^{(k+1)} & \dots \\
a_{n,n}^{(k+1)} & & & & & &
\end{pmatrix}.$$

**Étape n-1** : Le système $\left(S^{(n)}\right) : A^{(n)}x = b^{(n)}$ obtenu est triangulaire supéricure avec

$$A^{(n)} = \begin{pmatrix} a_{1,1}^{(1)} & & \cdots & \cdots \\ a_{1,n}^{(1)} & & & \\ 0 & a_{2,2}^{(2)} & & \\ a_{2,n}^{(2)} & & & \\ 0 & 0 & a_{3,3} & \\ a_{3,n}^{(3)} & & & \\ \vdots & \ddots & \ddots & \ddots \\ \vdots & & & \\ 0 & \cdots & 0 & 0 \\ a_{n,n}^{(n)} & & & \end{pmatrix}$$

et peut donc être résolu par l'algorithme de substitution rétrograde de la sous-section 2.1.3 .

**Exemple** : Considérons le système suivant

$$(S) = \left(S^{(1)}\right) \begin{cases} x_1 + 2x_2 + 5x_3 &= 1 \\ 3x_1 + 2x_2 - x_3 &= \frac{1}{2} \\ 5x_2 + 3x_3 &= 1 \end{cases}$$

Le premier pivot de l'élimination de Gauss est donc $a_{1,1}^{(1)} = 1$ et on a $g_{2,1}^{(1)} = 3, g_{3,1}^{(1)} = 0$. La première étape fournit donc

$$\left(S^{(2)}\right) \begin{cases} x_1 + 2x_2 + 5x_3 & = 1 \\ -4x_2 - 16x_3 & = -\frac{5}{2} \\ 5x_2 + 3x_3 & = 1 \end{cases}$$

Le second pivot de l'élimination de Gauss est donc $a_{22}^{(2)} = -4$ et on a $g_{3,2}^{(2)} = -\frac{5}{4}$. On obtient donc le système

$$\left(S^{(3)}\right) \begin{cases} x_1 + 2x_2 + 5x_3 = 1 \\ -4x_2 - 16x_3 = -\frac{5}{2} \\ -17x_3 = -\frac{17}{8} \end{cases}$$

On résout alors ce système triangulaire supérieur par l'algorithme de substitution rétrograde pour trouver : $x_1 = x_2 = x_3 = \frac{1}{8}$ (voir l'exemple de la sous-section 2.1.3 ).

### 2.2.2 Point de vue numérique : stratégies de choix du pivot

Au cours de l'exécution de l'élimination de Gauss, si on tombe sur un pivot nul, alors on permute la ligne en question avec une ligne en dessous pour se ramener à un pivot non nul (ceci est toujours possible car A est supposée inversible). Cependant, certains choix de pivots peuvent s'avérer plus judicieux que d'autres.

**Exemple** :

Considérons le système $(S) : Ax = b$ où

$$A = \begin{pmatrix} \alpha & 1 \\ 1 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

avec $\alpha$ réel non nul. On suppose de plus $\alpha \neq 1$ de sorte que $A$ est inversible. La solution de ce système est alors donnée par $x_1^* = \frac{1}{1-\alpha}, x_2^* = \frac{1-2\alpha}{1-\alpha}$. Supposons maintenant que $\alpha$ est précédente. Le premier pivot est alors $\alpha$ et $g_{2,1} = \frac{1}{\alpha}$, La première étape transforme donc le système $(S)$ en $\left(S^{(2)}\right)$ : $A^{(2)}x = b^{(2)}$ avec

$$A^{(2)} = \begin{pmatrix} \alpha & 1 \\ 0 & 1 - \frac{1}{a} \end{pmatrix}, \quad b^{(2)} = \begin{pmatrix} 1 \\ 2 - \frac{1}{a} \end{pmatrix}$$

La deuxième équation entraîne $-\frac{1}{\alpha}x_2 \approx -\frac{1}{\alpha}$ d'où $x_2 \approx 1$ et la première nous donne alors si on multiplie la première ligne par une puissance de 10 quelconque, on va trouver la même erreur. Notons $x_2 = x_2^* + \delta x_2$ ou $|\delta x_2|$ est l'erreur absolue sur $x_2$. On a alors

$$x_1 = \frac{1 - x_2}{\alpha} = \frac{1 - x_2^*}{\alpha} - \frac{\delta x_2}{\alpha}$$

et on voit donc que l'erreur $\delta x_1 = \frac{1}{\alpha}\delta x_2$ sur $x_1$ est très amplifiée par rapport à celle sur $x_2$. L'anomalie provient du déséquilibré entre les coefficients de $x_1$ ct $x_2$ de la ligne du pivot. Pour y remédier, échangeons maintenant les

deux lignes de notre système et appliquons l'élimination de Gauss avec 1 comme premier pivot. On obtient alors

$$A^{(2)} = \begin{pmatrix} 1 & 1 \\ 0 & 1-\alpha \end{pmatrix}, \quad b = \begin{pmatrix} 2 \\ 1-2\alpha \end{pmatrix}$$

qui entraîne alors $x_2 \approx 1$ et $x_1 \approx 1$ ce qui est correct.

**Élimination de Gauss à pivot partiel**

A l'étape $k$, on échange les lignes $k$ et $k'\,(k' \geq k)$ de $A^{(k)}$ de telle sorte que $\left|a_{ki}^{(k)}\right| = \max\left\{\left|a_{i,k}^{(k)}\right|, i \geq k\right\}$.

Par exemple, pour le système $(S)$ de la sous-section précédente, à l'étape 1, on va permuter les lignes 1 et 2 et considérer

$$(S') : \begin{cases} 3x_1 + 2x_2 - x_3 &= \frac{1}{2} \\ x_1 + 2x_2 + 5x_3 &= 1 \\ 5x_2 + 3x_3 &= 1 \end{cases}$$

**Élimination de Gauss à pivot total**

A l'étape $k$, on échange à la fois les lignes $k$ et $k'\,(k' \geq k)$ et les colonnes $k$ et $k''\,(k'' \geq k)$ de telle sorte que : $\left|a_{k,k}^{(k)}\right| = \max\left\{\left|a_{i,j}^{(k)}\right|, i \geq k, j \geq k\right\}$.

**Attention** : Si on fait des échanges de colonnes cela modifie l'ordre des composantes du vecteur solution $x$ donc il faut penser à rétablir le bon ordre des composantes $\hat{\lambda}$ la fin. Par exemple, pour le système $(S)$ de la

sous-section précédente, à l'étape 1, on va permuter les colonnes 1 et 3 et considérer

$$(S') \begin{cases} 5x_3 + 2x_2 + x_1 & = 1 \\ -x_3 + 2x_2 + 3x_1 & = \frac{1}{2} \\ 3x_3 + 5x_2 & = 1 \end{cases}$$

### 2.2.3 Lien avec la factorisation LU d'une matrice

Définition 2.3. Soit $A \in M_{n\times n}$(K). On appelle factorisation LU de $A$ une factorisation $A = LU$ avec $L \in M_{n\times n}(K)$ triangulaire inférieure et $U \in M_{n\times n}($ K) triangulaire supérieure.

Lemme 2.4. Avec les notations de la sous-section 9.2.1, à l'étape $k$ de l'élimination de Gauss, on a $A^{(k+1)} = G_k A^{(k)}$ où

$$G_k = \begin{pmatrix} 1 & & (0) & & 0 & \dots \\ 0 & & & & & \\ & & \ddots & & \vdots & \\ \vdots & & & & & \\ & (0) & & 1 & 0 & \dots \\ 0 & & & & & \\ 0 & \dots & 0 & -g_{k+1,k} & 1 & \\ (0) & & & & & \\ \vdots & & \vdots & \vdots & & \ddots \\ & & & & & \\ 0 & \dots & 0 & -g_{n,k} & (0) & \\ 1 & & & & & \end{pmatrix}$$

et

$$g_{i,k} = \frac{a_{i,k}^{(k)}}{a_{k,k}^{(k)}, i = k+1, \dots, n}; \quad b^{(k+1)} = G_k b^{(k)}.$$

Démonstration. Il suffit d'effectuer le produit $G_k A^{(k)}$ et de vérifier que l'on retrouve $A^{(k+1)}$.

Définition 2.5. Soit $A \in M_{n\times n}(K)$. Les mineurs fondamentaux $D_k, k = 1, \dots, n$ de $A$ sont les déterminants des sous-matrices de $A$ formées par les $k$ premières lignes et les $k$ premières colonnes de $A : Dk = det((ai, j)1i, jk)pourk =$

$1, ..., n$.

Théorème 2.6. Soit $A \in M_{monn}(\mathrm{K})$ une matrice carrée inversible. Les propriétés suivantes sont équivalentes :

(i) L'élimination de Gauss s'effectue sans permutation de lignes :

(ii) Π existe $L \in M_{n\times n}$ (K) triangulaire inférieure inversible et $U \in M_{n\times n}$ (K) triangulaire supérieure inversible telles que $A = LU$;

(iii) Tous les mineurs fondamentaux de A sont non nuls.

Démonstration. Pour la suite de ce cours, la partie de la démonstration qui nous intéresse est l'implication $(i) \Rightarrow (ii)$ qui nous permettra de calculer la factorisation $LU$ de $A$. D'après le lemme 2.4, on a $A^{(n)} = G_{n-1}G_{n-2}\cdots G_1 A$. La matrice $G_{n-1}G_{n-2}\cdots G_1$ étant inversible, on peut donc écrire $A = (G_{n-1}G_{n-2}\cdots G_1)^{-1} A^{(n)}$. Les matrices $G_k$ étant triangulaires inférieures, la proposition 2.2 affirme que $(G_{m-1}G_{n-2}\cdots G_1)^{-1}$ est aussi triangulaire inférieure. De plus, par construction $A^{(n)}$ est triangulaire supérieure d'où le résultat en posant $L = (G_{m-1}G_{n-2}\cdots G_1)^{-1}$ et $U = A^{(n)}$.

Corollaire 2.7. Soit $A \in M_{n\times n}(\mathrm{K})$ une matrice carrée inversible. La matrice $A$ admet une factorisation Li si et seulement si tous ses mineurs fondamentaux sont non nuls.

Le lemme suivant nous permet d'expliciter la matrice $L$ de la factorisation LU de $A$ obtenue à partir de l'élimination de Gauss.

Lemme 2.8. Avec les notations précédentes, on $a$

$$(G_{m-1}G_{n-2}\cdots G_1)^{-1} = \begin{pmatrix} 1 & 0 & \dots & \dots & 0 \\ 9_{2,1} & 1 & \ddots & & \vdots \\ g_{3,1} & g_{3,2} & 1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ g_{n,1} & g_{n-2} & \cdots & g_{n,n-1} & 1 \end{pmatrix}$$

Démonstration. Faire le calcul.

Corollaire 2.9. Soit $A \in M_{n\times n}(\mathrm{K})$ une matrice carrée inversible. Si tous les mineurs fondamentaux de $A$ sont non nuls, alors avec les notations précédentes, l'élimination de Gauss fournit la factorisation $LU$ de $A$ suivante

$$A = \begin{pmatrix} 1 & 0 & \dots & \dots & 0 \\ g_{2,1} & 1 & \ddots & & \vdots \\ 9_{3,1} & g_{3,2} & 1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ g_{n,1} & g_{n,2} & \dots & g_{n,n-1} & 1 \end{pmatrix} \begin{pmatrix} a_{1,1}^{(1)} & & \dots & \dots \\ a_{1,n}^{(1)} & & & \\ 0 & a_{22}^{(2)} & & \\ a_{2n}^{(2)} & & & \\ 0 & 0 & a_{3,3}^{(3)} & \\ a_{3n}^{(3)} & & & \\ \vdots & \ddots & \ddots & \ddots \\ \vdots & & & \\ 0 & \dots & 0 & 0 \\ a_{n,10}^{(n)} & & & \end{pmatrix}$$

On remarque que la matrice $L$ obtenue est à diagonale unité.

**Exemple** : En reprenant l'exemple de la sous-section 2.2.1, on trouve :

$$\underbrace{\begin{pmatrix} 1 & 2 & 5 \\ 3 & 2 & -1 \\ 0 & 5 & 3 \end{pmatrix}}_{A} = \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ 0 & -\frac{5}{4} & 1 \end{pmatrix}}_{L} \underbrace{\begin{pmatrix} 1 & 2 & 5 \\ 0 & -4 & -16 \\ 0 & 0 & -17 \end{pmatrix}}_{U}$$

Proposition 2.10. Soit $A \in M_{n\times n}(K)$ une matrice carrée inversible admettant une factorisation $LU$. Alors il existe une unique factorisation $LU$ de $A$ avec $L$ à diagonale unité.

Démonstration. Si A admet une factorisation LU, alors l'existence d'une factorisation LU avec $L$ à diagonale unité est claire d'après le corollaire 2.9. Supposons maintenant que $A$ admette deux factorisations LU : $A = L_1U_1$ et $A = L_2U_2$ avec $L_1$ et $L_2$ à diagonale unité. L'égalité $L_1U_1 = L_2U_2$ implique $U_1U_2^{-1} = L_1^{-1}L_2$. D'après la proposition 2.2, $U_1U_2^{-1}$ est triangulaire supérieure et $L_1^{-1}L_2$ est triangulaire inférieure à diagonale unité. La seule possibilité pour que ces deux matrices soient égales est donc $U_1U_2^{-1} = L_1^{-1}L_2 = \mathrm{I}_n$ ce qui implique $L_1 = L_2$ et $U_1 = U_2$.

Lorsque $A$ admet une factorisation LU, la résolution du système d'équations linéaires (S) : $Ax = b$ se ramène a la résolution de deux systèmes linéaires

triangulaires, En effet

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

En pratique, on résout donc d'abord $Ly = b$ puis connaissant $y$ on résout $Ux = y$.

Définition 2.11. On appelle matrice de permutation associée a une permutation $\sigma \in \mathcal{S}_n$, la matrice P = (i(j))1i,j,n où ij = 1 si i = j et ij = 0 si i = j. Par exemple si l'on considère la permutation, $\sigma : (1, 2, 3, 4, 5) \mapsto (3, 2, 5, 1, 4)$, on obtient la "matrice de permutation élémentaire

$$P_\sigma = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Multiplier une matrice $A$ à gauche (resp. À droite) par une matrice de permutation revient alors a permuter les lignes (resp. les colonnes) de la matrice. Par exemple, en multipliant une matrice a gauche par la matrice $\mathcal{P}_\sigma$ ci-dessus, la troisième ligne devient la première, la seconde ligne reste inchangée, la cinquième ligne devient la troisième ... Les matrices

de permutation sont orthogonales (voir Définition 2.20 ) c'est-a-dire que $\forall a \in \mathcal{S}_n, \mathcal{P}_\sigma^{-1} = \mathcal{P}_\sigma^T$.

Le corollaire 2.7 nous donne une condition nécessaire et suffisante pour qu'une matrice inversible admette une factorisation LU. Lorsque cette factorisation LU n'existe pas, on peut tout de même utiliser le théorème suivant :

Théorème 2.12. Soit $A \in M_{n\times n}(K)$ une matrice carne inversible. Il existe une matrice de permutation $\mathcal{P}$ telle que $\mathcal{P}A$ admette une factorisation $LU$.

Démonstration. Admis pour ce cours.

Notons que dans ce cas, on a :

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

En pratique, on résout donc d'abord $Ly = \mathcal{P}b$ puis connaissant $y$ on résout $Ux = y$.

### 2.2.4 Coût de l'algorithme

Résolution d'un système via l'élimination de Gauss Nous allons nous intéresser au nombre d'opérations a virgule flottante nécessaires a la résolution d'un système $(S) : Ax = b$ en utilisant l'élimination de Gauss puis en résolvant le système triangulaire $A^{(n)}x = b^{(n)}$.

Soit $A \in M_{n\times n}(K)$ une matrice carrée inversible et supposons que $A$ admette une factorisation LU. A l'étape $k$ de l'élimination de Gauss, on doit faire $n-k$ divisions pour calculer les $g_{i,k}$ puis nous avons $(n-k)^2$ coefficients $a_{ij}^{(k+1)}$ à calculer et $(n-k)$ coefficients $b_i^{ik+1)}$ Cela nécessite donc $(n-k)(n-k+1)$ multiplications puis $(n-k)(n-k+1)$ soustractions.

Au total, pour toute l'élimination de Gauss, nous aurons donc à faire

$$\begin{aligned}
\textstyle\sum_{k=1}^{n-1}(n-k)+2\sum_{k=1}^{n-1}(n-k)(n-k+1) &= \textstyle\sum_{k=0}^{n-1}k+2\sum_{k=0}^{n-1}k(k+1)\\
&= \tfrac{n(n-1)}{2}+2\left(\textstyle\sum_{k=1}^{n-1}k+\sum_{k=1}^{n-1}k^2\right)\\
&= \tfrac{n(n-1)}{2}+2\left(\tfrac{n(n-1)}{2}+\tfrac{n(n-1)(2n-1)}{6}\right)\\
&= n(n-1)\left(\tfrac{3+6+2(2n-1)}{6}\right)\\
&= \left(\tfrac{n(n-1)(4n+7)}{6}\right)\\
&= \left(\tfrac{4n^3+3n^2-7n}{6}\right)
\end{aligned}$$

opérations a virgule flottante. D'après la proposition 2.1, la résolution d'un système triangulaire coûte $n^2$ opérations à virgule flottantes donc au total, la résolution du système $(S)$ par cette méthode coûtera $G(n) = n^2+\left(\frac{4n^3+3n^2-7n}{6}\right) = \frac{4n^3+9n^2-7n}{6}$ opérations a virgule flottante.

Lorsque $n$ tend vers 1 'infini, on a $G(n) \sim \frac{2n^3}{3}$ opérations à virgule flottante.

Notons que d'après ce qui précède, cette estimation représente aussi le coût asymptotique du calcul de la factorisation LU d'une matrice. On a donc :

Lemme2.13. Soit $A \in M_{n\times n}(K)$ une matrice carrée inversible. Résoudre un

système linéaire (S) : $Ax = b$ via l'élimination de Gauss nécessite un nombre d'opérations a virgule flottante équivalent à $\frac{2n^3}{3}$ lorsque $n$ tend vers l'infini. Ce coût asymptotique est aussi celui du calcul de la factorisation $LU$ de $A$. Pour comparer avec le résultat obtenu en utilisant les formules de Cramer, si l'on suppose que $n = 100$, alors cela donne environ $6,6.10^5$ opérations à virgule flottante et avec un ordinateur fonctionnant à 100 megaflops, cela prendra moins de 7 millièmes de secondes. Cela ne se fait donc pas à la main mais votre ordinateur personnel pourra le faire en quelques(dizaines de) secondes.

**Faut-il inverser une matrice ?**

Nous admettrons qu'étant donnée la factorisation LU de $A$, le coût du calcul de l'inverse $A^{-1}$ de $A$ lorsque $n$ tend vers 1 'infini est de $\frac{4n^3}{3}$ opérations à virgule flottante. Au total, lorsque $n$ tend vers l'infini, il faut donc $2n^3$ opérations a virgule flottante pour calculer l'inverse de A. D'après ce qui précède cela signifie donc qu'asymptotiquement (i.e., lorsque $n$ tend vers l'infini), il faut 3 fois plus d'opérations a virgule flottante pour calculer l'inverse de A que pour résoudre le système linéaire $Ax = b$ en utilisant l'élimination de Gauss. Cela implique qu'il ne faut pas calculer l'inverse d'une matrice pour résoudre un système linéaire.

**Cas de la résolution de plusieurs systèmes de même matrice $A$**

Soit $A \in M_{n\times n}(K)$ une matrice carrée inversible et supposons que l'on ait à résoudre $K$ systèmes linéaires avec la même matrice $A$ et $N$ seconds membres $b^{(1)}, \ldots, b^{[K]}$. Si l'on applique l'élimination de Gauss a chacun de ces systèmes pour les résoudre, alors d'après ce qui précède cela nous coûtera $K\frac{4n^3+9n^2-7n}{6}$ opérations a virgule flottante, Si maintenant on calcule une fois pour toute une factorisation LU de $A$ (via l'élimination de Gauss) et que l'on résout ensuite successivement les $2K$ systèmes triangulaires cela ne nous coûtera que $\left(\frac{4n^3+3n^2-7n}{6}\right) + 2Kn^2$ opérations à virgule flottante ce qui est asymptotiquement meilleur. Notons que si l'on calcule une fois pour toute l'inverse $A^{-1}$ de $A$ et que l'on résout ensuite chaque système en posant $x^{[i]} = A^{-1}b^{[i]}$ cela coûtera $2n^3$ opérations pour le calcul de l'inverse puis $n(2n-1)$ opérations pour calculer chacun des $K$ produits $A^{-1}b^{[i]}$ soit au total $2n^3 + 2Kn^2$ ce qui est moins avantageux que d'utiliser la factorisation LU.

## 2.3 Méthode de Cholesky

La méthode de Cholesky est une alternative a l'élimination de Gauss qui s'applique aux matrices symétriques et définies positives.

Définition 2.14. Une matrice $A \in M_{n\times n}(\mathrm{K})$ est dite symétrique si elle est

égale $\dot{a}$ sa transposée, i.e., $A^T = A$.

Définition 2.15. Soit K = $R$ ou C. Le produit scalaire canonique sur $\mathrm{K}^n$ est défini comme l'application $\langle \ldots \rangle : K^n \times K^n \to K, (u, v) \mapsto \langle u, v)$

• Si $K = R, \langle u, v) = v^T u = \sum_{i=1}^n u_i v_i$ (produit scalaire euclidien)

• Si $K = C, \langle u, v) = \bar{v}^T u = \sum_{i=1}^n u_i \bar{v}_i$ (produit scalaire hermitien).

Définition 2.16. Une matrice $A \in M_{n\times n}(\mathrm{K})$ est dite définie positive, resp. semi définie positive si pour tout $x \in R^n$ non nul, on $a(Ax, x) > 0$, resp. $\langle Ax, x) \geq 0$.

On montre facilement les propriétés suivantes :

1. Une matrice définie positive est inversible

2. Si $A \in M_{n\times n}(K)$ est inversible, alors $A^T A$ est symétrique et définie positive

3. Si $A = (a_{i,j})_{1\leq i,j\leq n} \in M_{n\times n}(\mathrm{K})$ est définie positive, alors $a_{i,i} > 0$ pour tout $i = 1, \ldots, n$.

Théorème 2.17 (Caractérisation des matrices symétriques définies positives) Une matrice réelle $A \in M_{n\times n}(\mathrm{R})$ est symétrique définie positive si et seulement si il existe une matrice $L = (l_{i,j})_{1\leq i,j\leq n} \in M_{n\times n}(R)$ triangulaire inférieure inversible telle que $A = LL^T$. De plus, si pour tout $i = 1, \ldots, n, l_{i,i} \geq 0$, alors $L$ est unique.

Démonstration. Admis pour ce cours.

Remarquons que si $L$ est triangulaire inférieure, alors $L^T$ est triangulaire supérieure de sorte que la factorisation de Cholesky $A = LL^T$ peut être vue comme un cas particulier de la factorisation LU d'une matrice étudiée dans la section précédente.

On donne maintenant l'algorithme de Cholesky qui étant donnée une matrice carrée réelle $A$ symétrique et définie positive calcule une matrice $L$ telle que $A = LL^T$. On admettra que cet algorithme est correct.

Entrée : $A = (a_{i,j})_{1 \leq i,j \leq n} \in M_{n \times n}(R)$ symétrique et définie positive.

Sortie : $L = (l_{i,j})_{1 \leq i,j \leq n} \in M_{n \times n}(R)$ tel que $A = LL^T$

1. $l_{1,1} = \sqrt{a_{1,1}}$

2. Pour $i$ de 2 à $n$ par pas de 1, faire :

• $l_{i,1} = \frac{a_{i,1}}{l_{1,1}}$

3. Pour $j$ de 2 à $n$ par pas de 1, faire :

•Pour i de 1 à j-1 par pas de l faire :

$$l_{i,j} = 0$$

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

• Pour i de j+1 à n par pas de 1, faire : $\mathrm{l}_{i,j} = \frac{a_{i,j} - \sum_{i,j}^{j-1} l_{i,k} l_{j,k}}{l_{j,j}}$

4. Retourner $L = (l_{i,j})_{1 \leq i,j \leq n} \in M_{n \times n}(R)$

Notons que dans ce cas, comme pour la factorisation LU, on a :

$$Ax = b \Longleftrightarrow LL^T x = b \Longleftrightarrow \begin{cases} Ly = b \\ L^T x = y \end{cases}$$

En pratique, on résout donc d'abord $Ly = b$ puis connaissant $y$ on résout $L^T x = y$.

Proposition 2.18. L'algorithme de Cholesky décrit ci-dessus nécessite $n$ extractions de racines carrées et un nombre d'opérations à virgule flottante équivalent à $\frac{n^3}{3}$ lorsque $n$ tend vers l'infini.

Démonstration. Admis pour ce cours.

On voit donc que cet algorithme requiert asymptotiquement presque (cela dépendra de la méthode utilisée pour le calcul des racines carrées qui peut-être plus ou moins rapide) deux fois moins d'opérations à virgule flottante que celui calculant la factorisation $LU$ via l'élimination de Gauss. Par conséquent, il est conseillé de l'utiliser lorsque $A$ est réelle symétrique et définie positive.

## 2.4 Méthode de Householder-Factorisation QR

Dans cette section, on suppose que $A \in M_{n\times n}(R)$ est une matrice réelle inversible.

### 2.4.1 Transformation (élémentaire) de Householder

Définition 2.19. On appelle matrice (élémentaire) de Householder une matrice $H$ de la forme $H_u = I_n - 2uu^T$, où $u \in R^n$ est un vecteur unitaire $c$ 'est- $\dot{a}$ -dire de norme 1 pour la norme associée au produit scalaire canonique sur $R^n$ définie par $\|u\| = \sqrt{\langle u, u \rangle}$.

Par exemple, pour $n = 3$, on peut considérer le vecteur $u = \frac{1}{\sqrt{6}} \begin{pmatrix} -1 & 1 & 2 \end{pmatrix}^T$ qui vérifie bien $\|u\| = 1$. On obtient alors la matrice de Householder

$$H_u = \frac{1}{3} \begin{pmatrix} 2 & 1 & 2 \\ 1 & 2 & -2 \\ 2 & -2 & -1 \end{pmatrix}$$

Définition 2.20. Une matrice $A \in M_{n\times n}(\mathrm{K})$ est dite orthogonale si elle est réelle, i.e., $A \in M_{n\times n}(R)$ et si $AA^T = A^T A = I_n$.

Par exemple, les matrices de permutation (voir Définition 2.11 ) sont orthogonales.

Proposition 2.21. Toute matrice de Householder $H$ est symétrique et orthogonale.

Démonstration. Facile à vérifier.

Proposition 2.22. Pour tout vecteur $u \in R^n$ tel que $\|u\| = 1$, on a $H_u u = -u$. De plus, si $v \in R^n$ est orthogonal $\dot{a}u$, i.e., $\langle u, v \rangle = 0$, alors $H_u v = v$.

Démonstration. Facile à vérifier.

Géométriquement, la matrice $H_u$ représente donc la symétrie orthogonale par rapport au plan $u^T$ orthogonal à $u$.

Lemme 2.23. Soit $x$ et $y$ deux vecteurs de $R^n$ tels que $x \neq y$ et $\|x\| = \|y\|$. Alors il existe un vecteur unitaire $u \in R^n$ tel que $H_u x = y$

Démonstration. Prendre $u = \frac{x-y}{\|x-y\|}$.

### 2.4.2 Principe de la méthode de Householder

La méthode de Householder pour la résolution d'un système linéaire $Ax = b$ est basée sur les deux propositions suivantes que nous admettrons.

Proposition 2.24. Soit $v$ un vecteur non nul de $R^n$. Alors il existe une matrice de Householder $H$ et un réel $\alpha$ tels que $Hv = \alpha e_1$, où $e_1 = (1, 0, \ldots, 0)^T$ est le premier vecteur de la base canonique de $R^n$.

Proposition 2.25. Soit $u = (u_i)$ un vecteur unitaire de $R^n$ tel que $u_1 = \cdots = u_p = 0$ pour $p < n$. On décompose alors $u$ en deux blocs : $u = (0 \quad z)^T$ avec $z \in R^{n-p}$. La matrice de Householder $H_u$ se decompose alors par blocs de la manière suivante : $H_u = \begin{pmatrix} I_p & 0 \\ 0 & H_z \end{pmatrix}$.

### 2.4.3 Exemple de résolution d'un système linéaire par la méthode de Householder

Considérons le système linéaire suivant :

$$(S)\begin{cases} 2x_1 + x_2 + 2x_3 = 1 \\ x_1 + x_2 + 2x_3 = 1 \\ 2x_1 + x_2 + x_3 = 1 \end{cases}$$

**Étape 1**: On considère le vecteur obtenu à partir de la première colonne de la matrice $A$ de $(S)$, i.e., $a_1 = \begin{pmatrix} 2 & 1 & 2 \end{pmatrix}^T$. Le but de cette étape est de trouver la matrice $H$ apparaissant dans la proposition 2.24 et correspondant à $v = a_1$ de sorte qu'en multipliant la matrice $A$ du système (S) à gauche par $H$ on obtienne des zéros sous le premier coefficient. Pour ceci, en concordance avec la preuve du lemme 2.23, on introduit le vecteur $v_1 = \frac{a_1}{|a_1|} - e_1 = \frac{1}{3}(-1 \quad 1 \quad 2)^T$, le vecteur $u_1 = \frac{v_1}{\|v_1\|} = \frac{1}{\sqrt{6}} \begin{pmatrix} -1 & 1 & 2 \end{pmatrix}^T$ et on considère la matrice élémentaire de Householder $H_{u_1} = \frac{1}{3} \begin{pmatrix} 2 & 1 & 2 \\ 1 & 2 & -2 \\ 2 & -2 & -1 \end{pmatrix}$. On a alors :

$$(S) : Ax = b \Longleftrightarrow H_{\mathrm{u}_1} Ax = H_{\mathrm{u}_1} b \Longleftrightarrow \begin{pmatrix} 9 & 5 & 8 \\ 0 & 1 & 4 \\ 0 & -1 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 5 \\ 1 \\ -1 \end{pmatrix}$$

**Étape 2** : On se place maintenant dans $R^2$ et on considère le vecteur $a_2 = (1 \quad -1)^T$ auquel nous allons appliquer la même technique qu'au vecteur $a_1$ à l'étape précédente. On utilisera ensuite la proposition 2.25 pour obtenir une matrice de Householder de la bonne taille. On pose alors $z_2' = \frac{a_2}{\|a_2\|} - e_1'$ où $e_1' = (1, 0)$ est le premier vecteur de la base canonique de $R^2$ et $z_2 = \frac{z_2'}{\|z_2^2\|}$. On considère ensuite la matrice de Householder $H_{z_2} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -1 \\ -1 & -1 \end{pmatrix}$ puis $u_2 = \begin{pmatrix} 0 & z_2 \end{pmatrix}^T$ et $H_{\mathrm{u}_2} = \begin{pmatrix} 1 & 0 \\ 0 & H_{z_2} \end{pmatrix}$ (Voir Proposition 2.25). On a alors :

$$Ax = b \Longleftrightarrow H_{w_2} H_{u_1} Ax = H_{u_2} H_{w_1} b \Longleftrightarrow \begin{pmatrix} 9 & 5 & 8 \\ 0 & \sqrt{2} & \frac{5}{\sqrt{2}} \\ 0 & 0 & -\frac{3}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 5 \\ \sqrt{2} \\ 0 \end{pmatrix}$$

d'où en résolvant ce système triangulaire $x = \begin{pmatrix} 0 & 1 & 0 \end{pmatrix}^T$.

### 2.4.4 Factorisation QR d'une matrice

Définition 2.26. Soit $A \in M_{n\times n}(R)$ une matrice carrée réelle inversible. On appelle factorisation QR de $A$ une factorisation de la forme $A = QR$ avec $Q \in M_{n\times n}(R)$ orthogonale *et* $R \in M_{n\times n}(R)$ triangulaire supérieure.

En généralisant la méthode expliquée dans la sous-section précédente pour une matrice carrée réelle inversible quelconque de taille $n$, on obtient $n - 1$

matrices de Householder $H_1, \ldots, H_{n-1}$ telles que le produit $R = H_{n-1}H_{n-2}\cdots H_1 A$ est triangulaire supérieure. On pose alors $Q = (H_{n-1}H_{n-2}\cdots H_1)^{-1}$ de sorte que $A = QR$. La matrice $Q$ ainsi défini étant orthogonale, on a obtenu une factorisation QR de $A$. On obtient alors le résultat suivant :

Théorème 2.27. Pour toute matrice réelle $A \in M_{n\times n}(R)$, il existe une matrice orthogonale $Q \in M_{n\times n}(R)$ produit d'au plus $(n-1)$ matrices de Houscholder et une matrice triangulaire supérieure $R \in M_{n\times n}(R)$ telles que $A = QR$.

Proposition 2.28. La méthode de Householder pour résoudre un système linéaire nécessite un nombre d'opérations a virgule flottante équivalent a $\frac{4\mathrm{m}^3}{3}$ lorsque $n$ tend vers l'infini.

Démonstration. Admis pour ce cours.

Le coût de cette méthode est donc relativement élevé compare a l'élimination de Gauss ou à la méthode de Cholesky. Cependant un avantage de cette méthode est qu'elle est plus stable numériquement que les deux méthodes citées précédemment.

Notons enfin que la factorisation QR d'une matrice existe aussi pour des matrices rectangulaires et qu'elle est utilisée pour des problèmes de moindres carrés.

**Factorisation QR et moindres carré linéaires**

Le problème des moindres carrés linéaires est le suivant : étants donné une

matrice $A \in M_{m\times n}(R)$ et un vecteur $b \in R^m$ minimiser la quantité $\|Ax-b\|_2^2$. Ce problème peut étre résolu en utilisant la factorisation QR de la matrice $A \in \mathrm{M}_{m\times n}(\mathrm{R})$ en procédant comme suit : la matrice $A$ étant rectangulaire (on supposera ici $m > n$), on partitionne les matrices $Q$ et $R$ de la manière suivante :

$$A = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} \begin{pmatrix} R_1 \\ \\ 0 \end{pmatrix}$$

où $Q_1 \in M_{m\times n}(R), Q_2 \in M_{m\times(m-n)}(R)$ et $R_1 \in M_{n\times n}(R)$. En remplaçant $A$ par sa factorisation $QR$, on a :

$$\|Ax - b\|_2^2 = \|QRx - b\|_2^2$$

On admettra qu'en appliquant une transformation orthogonale au vecteur des résidus $Ax - b$ on ne modifie pas la norme euclidienne du vecteur et que le problème de minimisation conduit au même résultat. On multiplie alors le résidu par $Q^T$ et on est ramené à minimiser

$$\left\|Q^TQRx - Q^Tb\right\|_2^2 = \left\|Rx - Q^Tb\right\|_2^2 = \left\| \begin{pmatrix} R_1 \\ \\ 0 \end{pmatrix} x - \begin{pmatrix} Q_1^T \\ \\ Q_2^T \end{pmatrix} b \right\|_2^2$$

ce qui revient à minimiser

$$\left\|R_1x - Q_1^Tb\right\|_2^2 + \left\|Q_2^Tb\right\|_2^2$$

On résout alors le système triangulaire $R_1 x = Q_1^T b$ et la somme des résidus au carré correspond à $\left\| Q_2^T b \right\|_2^2$.

Dans les problèmes qui apparaissent en économétrie ou statistique, on est aussi intéressé par la matrice des variances et covariances $\sigma^2 \left( A^T A \right)^{-1}$, où $\sigma^2 = \frac{\|Ax-b\|_2^2}{m-n}$. Pour obtenir cette matrice, on a donc besoin d'évaluer le résidu à la solution $x$ trouvée et de calculer l'inverse de $A^T A$. Si on possède la factorisation QR de $A$, alors $A^T A = R^T Q^T Q R = R^T R$ car $Q$ est orthogonale. On doit donc inverser la matrice $R^T R$. De plus, avec les notations précédentes, si on note $d = Q_2^T b$, on peut écrire $\sigma^2 = \frac{d^T d}{m-n}$.

**Remarque** : cela n'entre pas dans le cadre de ce cours mais il existe une méthode efficace et numériquement stable pour calculer l'inverse de matrices de la forme $A^T A$.

# Chapitre 3

# Conditionnement d'une Matrice pour la Résolution d'un Système Linéaire

## 3.1 Normes matricielles

### 3.1.1 Normes vectorielles

Soit $E$ un espace vectoriel sur $K = R$ ou $C$.

Définition 3.1. *On* appelle norme sur $E$ une application $\|.\| : E \to R_+$ telle

que :

$$\bullet \forall x \in E, (\|x\| = 0 \Rightarrow x = 0)$$

$$\bullet \forall \lambda \in K, \forall x \in E, \|\lambda x\| = |\lambda| \|x\|$$

$$\bullet \forall (x, y) \in E^2, \|x + y\| \leq \|x\| + \|y\|$$

Les exemples classiques de normes sur $R^n$ sont les normes $\|\cdot\|_1, \|\cdot\|_2$ et $\|.\|_\infty$ définies par :

$$\forall x = (x_i)_{1 \leq i \leq n} \in R^n, \quad \|x\|_1 = \sum_{i=1}^{n} |x_i|, \quad \|x\|_2 = \left( \sum_{i=1}^{n} |x_i|^2 \right)^{\frac{1}{2}} = \langle x, x \rangle^{\frac{1}{2}}, \quad \|x\|_\infty = \max_{1 \leq i \leq n} |x_i|$$

### 3.1.2 Normes matricielles et normes subordonnées

Définition 3.2. Une norme $\|.\|$ sur $\mathrm{M}_{n \times n}(\mathrm{K})$ est une norme matricielle si elle vérifie : $\forall (A, B) \in M_{n \times n}(K)^2, \quad \|AB\| \leq \|A\| \|B\|$.

L'exemple fondamental est celui des normes dites subordonnées qui sont associées à une norme vectorielle.

Théorème et Définition 3.3. Soit $\|.\|$ une norme vectorielle sur $\mathrm{K}^n$. Pour toute matrice $A \in M_{n \times n}(K)$, on définit $\|\cdot\|_M : M_{n \times n}(K) \to R_+ par \|A\|_M = \sup_{x \in \mathrm{K}^n \setminus \{0\}} \frac{\|Ax\|}{\|x\|}$.Alors$\|.\|_M$ est une norme matricielle. Elle est dite norme subordonnée a la norme vectorielle $\|.\|$.

Dans la suite on notera indifféremment $\|.\|$ pour une norme vectorielle ou

une norme matricielle.

Démonstration. On utilise le lemme suivant qui découle de la définition d'une norme subordonnée :

Lemme 3.4. Soit $\|.\|$ la norme subordonnée *à* une norme vectorielle $\|.\|$. Alors, pour tout vecteur $x \in K^n$ et pour toute matrice $M \in M_{n\times n}(K)$, on $a\|Mx\| \leq \|M\|\|x\|$ Soit $(A, B) \in M_{n\times n}(K)^2$ et $x \neq 0$. On a :

$$\frac{\|ABx\|}{\|x\|} \leq \frac{\|A\|\|Bx\|}{\|x\|} \leq \frac{\|A\|\|B\|\|x\|}{\|x\|}$$

d'où le résultat.

Les normes subordonnées respectivement aux normes vectorielles $\|\cdot\|_1, \|\cdot\|_2$ et $\|.\|_\infty$ de $R^n$ sont données par : $\forall A = (a_{i,j})_{1\leq i,j\leq n} \in M_{n\times n}(K)$ :

$$\|A\|_1 = \max_{1\leq j\leq n} \sum_{i=1}^{n} |a_{i,j}|, \quad \|A\|_2 = \sqrt{\rho\left(AA^*\right)}, \quad \|A\|_\infty = \max_{1\leq i\leq n} \sum_{j=1}^{n} |a_{i,j}|$$

où $A^* = \bar{A}^T$ désigne la matrice adjointe de $A$ et $\rho(M)$ désigne le rayon spectral d'une matrice $M$, i.e., le maximum des modules des valeurs propres de $M$.

Notons que ces trois normes matricielles sont équivalentes : pour $A \in M_{n\times n}(K)$, on a :

$$\frac{1}{\sqrt{n}}\|A\|_2 \leq \|A\|_1 \leq \sqrt{n}\|A\|_2, \quad \frac{1}{\sqrt{n}}\|A\|_\infty \leq \|A\|_2 \leq \sqrt{n}\|A\|_\infty, \quad \frac{1}{n}\|A\|_1 \leq \|A\|_\infty \leq n\|A\|_1.$$

## 3.2 Conditionnement d'une matrice

### 3.2.1 Exemple classique

Cet exemple est du à R.S.Wilson. Considérons le système linéaire $(S)$ : $Ax = b$ avec

$$A = \begin{pmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{pmatrix}, \quad b = \begin{pmatrix} 32 \\ 23 \\ 33 \\ 31 \end{pmatrix}$$

Remarquons tout d'abord que la matrice $A$ est symétrique, que son déterminant vaut 1 et que la solution de (S) est donnée par $x = \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix}^T$.

**Premier cas** : $b$ est perturbé. Perturbons légèrement le second membre $b$ et considérons

$$U = \begin{pmatrix} 32,1 \\ 22,9 \\ 33,1 \\ 30,9 \end{pmatrix}$$

Si on résout le système $(S')$ : $Ax' = b'$, on trouve $x' = (9,2 \quad -12,6 \quad 4,5 \quad -1,1)^T$. La petite perturbation sur le second membre b entraîne donc une forte perturbation sur la solution du système.

D'une manière générale, on considère les deux systèmes $Ax = b$ et $A(x + \delta x) = b + \delta b$. On a donc $A\delta x = \delta b$ de sorte que $\delta x = A^{-1}\delta b$ et on obtient la majoration suivante de l'erreur absolue sur la solution : $\|\delta x\| \leq \|A^{-1}\| . \|\delta b\|$. Or $Ax = b$ donc $\|b\| \leq \|A\| \cdot \|x\|$. Finalement, on obtient une majoration de l'erreur relative sur la solution en fonction de l'erreur relative sur la donnée

$$\frac{\|\delta x\|}{\|x\|} \leq \left\|A^{-1}\right\| \cdot \|A\| \cdot \frac{\|\delta b\|}{\|b\|}$$

Notons que cette majoration est optimale dans le sens où il n'existe pas de borne plus petite qui soit valable pour tout système. En effet, prenons $A = \begin{pmatrix} 1 & 0 \\ 0 & \frac{1}{2} \end{pmatrix}, b = \begin{pmatrix} 1 & 0 \end{pmatrix}^T$ et $\delta b = \begin{pmatrix} 0 & \frac{1}{2} \end{pmatrix}^T$. La solution de $Ax = b$ est alors $x = (1 \ 0)^T$ et celle de $A\delta x = \delta b$ est $\delta x = (0 \quad 1)^T$. On a donc $\frac{|\delta x|}{|x|} = 1, \frac{|\delta b|}{|b|} = \frac{1}{2}$. Or $\|A^{-1}\| \cdot \|A\| = 2$ donc la borne est atteinte.

**Deuxième cas** : $A$ est perturbée. De façon analogue, si on perturbe légèrement la matrice $A$ et que l'on considère

$$A'' = \begin{pmatrix} 10 & 7 & 8,1 & 7,2 \\ 7,08 & 5,04 & 6 & 5 \\ 8 & 5,98 & 9,89 & 9 \\ 6,99 & 4,99 & 9 & 9,98 \end{pmatrix}$$

alors la solution du système $(S'')$ : $A''x'' = b$ est $x'' = \begin{pmatrix} -81 & 107 & -34 & 22 \end{pmatrix}^T$.

D'une manière générale, on considère les deux systèmes $Ax = b$ et $(A + \Delta A)(x + \delta x) = b$ Il vient donc $\delta x = A^{-1}\Delta A(x + \delta x)$ d'où

$$\frac{\|\delta x\|}{\|x + \delta x\|} \leq \left\|A^{-1}\right\| . \|A\| \cdot \frac{\|\Delta A\|}{\|A\|}$$

### 3.2.2 Définition du conditionnement

Définition 3.5. Soit$\|.\|$ une norme matricielle subordonnée et A une matrice inversible. Le nombre Cond $(A) = \left\|A^{-1}\right\| . \|A\| s'appelle le conditionnement de A relatif à la norme \|.\|$. Ce nombre mesure la sensibilité de la solution par rapport aux données du problème.

Une matrice est bien conditionnée si Cond $(A) \approx 1$ est mal conditionnée ie. Cond $(A) \gg 1$. Le fait que les normes matricielles $\|\cdot\|_1, \|\cdot\|_2$ et $\|\cdot\|_\infty$ soient équivalentes implique la même propriété d'équivalence pour les conditionnements associés : pour toute matrice $A \in M_{n\times n}(K)$, on a :

$$\frac{1}{n}Cond_2(A) \leq Cond_1(A) \leq nCond_2(A), \quad \frac{1}{n}Cond_\infty(A) \leq Cond_2(A) \leq nCond_\infty(A)$$

$$\frac{1}{n^2}Cond_1(A) \leq Cond_\infty(A) \leq n^2 Cond_1(A)$$

Des exemples de matrices bien conditionnées sont les matrices de la forme

$$A = \begin{pmatrix} 4 & 1 & 0 & \dots & 0 \\ 1 & 4 & 1 & \ddots & \vdots \\ 0 & 1 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 1 \\ 0 & \dots & 0 & 1 & 4 \end{pmatrix}$$

qui vérifient que, quelque soit la dimension $n$ de $A$, Cond $_\infty(A) \leq 3$. Inversement, les matrices de Hilbert $H_n$ et les matrices de Vandermonde $V_n$ respectivement définies par

$$H_n = \left(\frac{1}{i+j-1}\right)_{1\leq i,j\leq n}, \quad V_n = \left(\begin{pmatrix} j \\ n \end{pmatrix}^{i-1}\right)_{1\leq i,j\leq n}$$

sont très mal conditionnées : leur conditionnement pour la norme $\|\cdot\|_\infty$ vérifie :

| $n$ | Cond $(H_n)$ | Cond $(V_n)$ |
|---|---|---|
| 2 | 27 | 8 |
| 4 | $2,8.10^4$ | $5,6.10^2$ |
| 6 | $2,9.10^7$ | $3,7.10^4$ |

Proposition 3.6 . Soit $A$ une matrice réelle et considérons la norme matricielle subordonnée

$$\|A\|_2 = \sqrt{\rho\left(AA^*\right)}, \qquad Cond_2(A) = \left\|A^{-1}\right\|_2 \cdot \|A\|_2 = \sqrt{\frac{\rho(AA^T)}{\sigma(AA^T)}}$$

où $\sigma(M)$ désigne le minimum des modules des valeurs propres de $M$. En particulier, si $A$ est symétrique, alors on obtient Cond $_2(A) = \frac{\rho(A)}{\sigma(A)}$.

Démonstration. Admis pour ce cours.

### 3.2.3 Estimation théorique de l'erreur a priori

**Premier cas** : $b$ est perturbé

Théorème 3.7. Soit $A \in M_{n\times n}(R)$ inversible et $b \in R^n$ tels que $Ax = b$ et $A(x + \delta x) = b + \delta b$ avec $x \neq 0$. Alors on $a$ :

$$\frac{1}{cond(A)} \cdot \frac{\|\delta b\|}{\|b\|} \leq \frac{\|\delta x\|}{\|x\|} \leq Cond(A) \cdot \frac{\|\delta b\|}{\|b\|}$$

Démonstration. La majoration a déjà été prouvée. La minoration s'obtient directement à partir des inégalités $\|\delta b\| \leq \|A\| \cdot \|\delta x\|$ et $\|x\| \leq \left\|A^{-1}\right\| .\|b\|$.

**Deuxième cas** : $A$ est perturbée

Théorème 3.8. Soit $A \in M_{n\times n}(R)$ inversible, $b \in R^n$ et $\Delta A \in M_{n\times n}(R)$ tels que $\left\|A^{-1}\right\| \cdot \|\Delta A\| < 1$. Alors $A + \Delta A$ est inversible. De plus si on suppose

$Ax = b$ et $(A + \Delta A)(x + \delta x) = b$ avec $x \neq 0$, alors on $a$ :

$$\frac{\|\delta x\|}{\|x\|} \leq \frac{Cond(A) \cdot \frac{\|\Delta A]}{\|}}{1 - Cond(A) \cdot \frac{\Delta A}{\|A}}$$

Démonstration. On a $(A + \Delta A) = A\left(I_n + A^{-1}\Delta A\right)$ avec par hypothèse $\left\|A^{-1}\Delta A\right\| < 1$. Le lemme suivant (admis pour ce cours - la preuve n'est pas difficile) implique que $(A + \Delta A)$ est inversible.

Lemme 3.9. Soit $\|.\|$ une norme matricielle subordonnée et $B \in M_{n\times n}(R)$ une matrice telle que $\|B\| < 1$, alors la matrice $I_n + B$ est inversible et on a:$\left\|(I_n + B)^{-1}\right\| \leq \frac{1}{1-\|B\|}$.

On a de plus $(A + \Delta A)^{-1} = \left(I_n + A^{-1}\Delta A\right)^{-1} A^{-1}$. Maintenant, Si $Ax = b$ et $(A + \Delta A)(x + \delta x) = b$, alors $(A + \Delta A)\delta x = -\Delta A \cdot x$ d'où $\delta x = -\left(I_n + A^{-1}\Delta A\right)^{-1} A^{-1}\Delta A \cdot x$. On a donc

$$\frac{\|\delta x\|}{\|x\|} \leq \|I_n + A^{-1}\Delta A\Big)^{-1} \|.\|A^{-1}\|.\|\Delta A\|$$

d'où d'après le lemme 3.9 ci-dessus,

$$\frac{\|\delta x\|}{\|x\|} \leq \frac{1}{1 - \|A^{-1}\Delta A\|} \cdot \left\|A^{-1}\right\| \cdot \|\Delta A\| \leq \frac{Cond(A) \cdot \frac{\|A\|}{|A\|}}{1 - Cond(A) \cdot \frac{\|A\|}{\|A\|}}$$

**Troisième cas** : $A$ et $b$ sont perturbés

Théorème 3.10. Soit $A \in M_{n\times n}(R)$ inversible, $b \in R^n$ et $\Delta A \in M_{n\times n}(R)$

vérifiant $\|A^{-1}\| \cdot \|\Delta A\| < 1$. Si l'on suppose que $Ax = b$ et $(A+\Delta A)(x+\delta x) = b + \delta b$ avec $x \neq 0$ alors on a :

$$\frac{\|\delta x\|}{\|x\|} \leq \frac{Cond(A)}{1 - Cond(A) \cdot \frac{\|A\|}{\|A\|}} \left( \frac{\|\delta b\|}{\|b\|} + \frac{\|\Delta A\|}{\|A\|} \right)$$

Démonstration. Exercice : on utilise les mêmes techniques que pour les preuves précédentes.

### 3.2.4 Estimation théorique de l'erreur a posteriori

Étant donné un système linéaire $Ax = b$, on va maintenant estimer, en fonction du conditionnement, l'erreur commise sur la solution réellement calculée. Soit $x$ la solution exacte et $y$ la solution obtenue par la machine. On pose $r = Ay - b; r$ s'appelle le résidu. On a alors le résultat suivant :

Théorème 3.11. Avec les notations précédentes, si $x \neq 0$, alors

$$\|y - x\| \leq Cond(A) \cdot \frac{\|r\|}{\|b\|} \cdot \|x\|$$

Démonstration. On a $r = Ay - b = A(y - x)$ d'où $y - x = A^{-1}r$ et $\|y - x\| \leq \|A^{-1}\| \cdot \|r\|$ ce qui implique $\|y - x\| \leq Cond(A) \cdot \frac{\|r\|}{\|A\|}$. Or $Ax = b$ implique $\|A\| \geq \frac{\|b\|}{|x|}$ d'où le résultat.

On voit donc que si le conditionnement est grand, l'erreur relative peut être

grande. Cette majoration n'est pas très facile à utiliser car le conditionnement est en général inconnu. Soit $C$ une approximation de $A^{-1}$ (que l'on peut par exemple calculer via la méthode d'élimination de Gauss) et posons $R = AC - I_n$. On a alors le résultat suivant :

Théorème 3.12. Avec les notations précédentes, *si* $\|R\| < 1$, alors

$$\|y - x\| \leq \frac{\|r\| \cdot \|C\|}{1 - \|R\|}$$

Démonstration. On a vu dans la preuve précédente que $\|y-x\| \leq \left\|A^{-1}\right\| . \|r\|$. Il suffit alors de remarquer que $A^{-1} = C\left(I_n + R\right)^{-1}$ et d'utiliser le lemme 3.9 .

# Chapitre 4

# Résolution d'un Système d'Equations Linéaires : Méthodes Itératives

## 4.1 Motivation

Les méthodes itératives deviennent indispensables dès que la taille $n$ du système est très grande. En effet, les méthodes directes exigent un nombre d'opérations à virgule flottante de l'ordre de $n^3$ lorsque $n$ tend vers l'infini ce qui les rend lentes pour de grandes valeurs de $n$. De tels systèmes apparaissent par exemple dans les techniques de résolution numérique d'équations

aux dérivées partielles. Les matrices des systèmes obtenus sont en général "creuses" ( à-dire qu'elles ont beaucoup de 0 ) et semi-définies positives. Voici un exemple classique. Étant donnée une fonction $f : R^2 \to R$, on se propose de trouver une solution approchée $\bar{u} : \Omega \subset R^2 \to R$ du problème suivant :

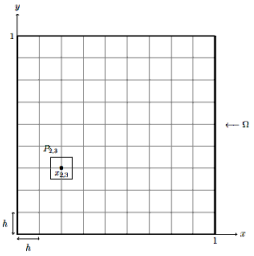
$$\begin{cases} -\Delta\bar{u} = f, \quad \forall (x,y) \in \Omega = ]0,1[\times]0,[1 \\ \bar{u} = 0, \quad \forall (x,y) \in \partial\Omega \end{cases}$$

où $\Delta\tilde{u} = \frac{\partial^2 \bar{u}}{\partial x^2} + \frac{\partial^2 \bar{u}}{\partial y^2}$ désigne le laplacien de la fonction $\tilde{u}$ et $\partial\Omega$ la frontière de $\Omega$.

Pour ce faire, on se donne un réel $h > 0$, on effectue une discrétisation de $\Omega = ]0,1[\times]0,1[$ de étagée $u$ (dépendante de $h$ ) telle que $u$ tende vers $\tilde{u}$ lorsque $h$ tend vers 0. On pose $h = \frac{1}{n+1}$ et on écrit $u = \sum_{i,j} u_{i,j}\chi_{i,j}$ où, pour $1 \leq i,j \leq n, \chi_{i,j}$ est la fonction caractéristique du pavé

$$P_{ij} = ] \left(i - \frac{1}{2}\right) h, \left(i + \frac{1}{2}\right) h[\times] \left(j - \frac{1}{2}\right) h, \left(j + \frac{1}{2}\right) h\,[.$$

On note alors $x_{i,j} = (ih, jh)$ les noeuds du quadrillage et $P_{i,j}$ est donc le pavé de centre $x_{i,j}$



En se basant sur la définition de la dérivée d'une fonction, pour $h$ suffisamment petit, on a les approximations par différences finies suivantes :

$$\frac{\partial \tilde{u}}{\partial x} \approx \frac{u\left(x+\frac{h}{2},y\right)-u\left(x-\frac{h}{2},y\right)}{h}, \quad \frac{\partial^2 \bar{u}}{\partial x^2} \approx \frac{u(x+h,y)-2u(x,y)+u(x-h,y)}{h^2}$$

De même,

$$\frac{\partial \tilde{u}}{\partial y} \approx \frac{u\left(x,y+\frac{h}{2}\right)-u\left(x,y-\frac{h}{2}\right)}{h}, \quad \frac{\partial^2 \tilde{u}}{\partial y^2} \approx \frac{u(x,y+h)-2u(x,y)+u(x,y-h)}{h^2}$$

En décomposant $f$ sous la forme $\sum_{i,j} f_{i,j}\chi_{i,j}$, notre problème se ramène alors à chercher les $u_{i,j}, 1 \leq i,j \leq n$ satisfaisants les équations suivantes :

$$\begin{cases} 4u_{i,j} - u_{i-1,j} - u_{i+1,j} - u_{i,j-1} - u_{i,j+1} = h^2 f_{i,j}, \quad 1 \leq i,j \leq n \\ u_{0,j} = u_{n+1,j} = u_{i,0} = u_{i,n+1} = 0, \quad \leq i,j \leq n \end{cases}$$

Ce schéma numérique est dit implicite par opposition à un schéma explicite pour lequel il est possible d'ordonner les inconnues de sorte que chacune d'entre elles puisse être déterminée explicitement $y$ en fonction des précédentes (système triangulaire). Les schémas numériques implicites

ont l'avantage d'être numériquement stables ce qui n'est pas toujours le cas pour les schémas explicites.

On remarque que ces équations forment un système linéaire que nous allons écrire sous forme matricielle. Pour ceci, on pose

$$M = \begin{pmatrix} 4 & -1 & 0 & \dots & 0 \\ -1 & 4 & -1 & \ddots & \vdots \\ 0 & -1 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -1 \\ 0 & \dots & 0 & -1 & 4 \end{pmatrix} \in M_{n\times n}(R)$$

$\mathrm{X}_j = \begin{pmatrix} u_{1,j} & u_{2,j} \dots u_{n,j} \end{pmatrix}^T$ pour $1 \leq j \leq n$

$X = \begin{pmatrix} X_1^T & X_2^T \dots X_n^T \end{pmatrix}^T$

ainsi que $F_j = (f_{1,j} \quad f_{2,j} \quad \dots \quad f_{n,j})^T$ pour $1 \leq j \leq n, F = \begin{pmatrix} F_1^T & F_2^T & \dots & F_n^T \end{pmatrix}^T$.

En définissant de plus $X_0 = X_{n+1} = 0$, le système précédent s'écrit

$$-X_{j-1} + MX_j - X_{j+1} = h^2 F_j, \quad 1 \leq j \leq n$$

ce qui conduit à

$$AX = h^2 F, A = \begin{pmatrix} M & -I_n & 0 & \dots & 0 \\ -I_n & M & -I_n & \ddots & \vdots \\ 0 & -I_n & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -I_n \\ 0 & \dots & 0 & -I_n & M \end{pmatrix} \in M_{n^2\times n^2}(R)$$

La matrice $A$ est symétrique réelle et on peut montrer qu'elle est définie positive donc en particulier inversible. En pratique, on a donc à résoudre un système linéaire tridiagonal par blocs de grande taille (notons que faire tendre $h$ vers 0 équivaut à faire tendre $n$ vers l'infini) à résoudre. Notons qu'il existe des méthodes efficaces (*e.g.*, l'algorithme de Thomas qui est une simplification de l'algorithme de Gauss dans le cas particulier des systèmes tridiagonaux) pour résoudre les systèmes linéaires tridiagonaux.

## 4.2 Notions générales

On rappelle que les méthodes itératives ne s'appliquent que dans le cas de systèmes à coefficients dans $R$ ou $C$ mais pas dans le cas des corps finis $F_p$.

### 4.2.1 Modèle général d'un schéma itératif

On considère une matrice $A \in M_{n\times n}(\mathrm{K})$ inversible, un vecteur $b \in K^n$ et un système linéaire

$$(S) : Ax = b$$

Le principe général d'une méthode itérative pour résoudre (S) est de générer une suite de vecteurs qui converge vers la solution $A^{-1}b$. Pour ce faire l'idée est d'écrire le système (S) sous une forme équivalente permettant de voir la

solution comme le point fixe d'une certaine fonction, *e.g.*

$$(S) \Longleftrightarrow Bx + c = x$$

avec $B \in M_{n\times n}(K)$ et $c \in K^n$ bien choisis c'est-à-dire $I - B$ inversible et $c = (I - B)A^{-1}b$ Par exemple, si $A = M - N$ pour deux matrices $M, N \in M_{n\times n}(K)$ avec $M$ inversible, on peut choisir $B = M^{-1}N$ et $c = M^{-1}b$. Dans la suite on supposera toujours que $B \in M_{n\times n}(K)$ et. $c \in K^n$ sont choisis tels que $I - B$ inversible et $c = (I - B)A^{-1}b$(e.g méthode itérative consistante). On se donne alors un vecteur $x^{(0)} \in \mathrm{K}^n$ et on construit une suite de vecteurs $x^{(k)} \in K^n$ à l'aide du schéma itératif

$$x^{(k+1)} = Bx^{(k)} + c, \quad k = 1, 2, \ldots \quad (4.2)$$

Si la suite $\left(x^{(k)}\right)_{k\in\mathrm{N}}$ est convergente, alors elle converge vers la solution $A^{-1}b$ de $(S)$. En effet, si elle existe, la limite $x^*$ est un point fixe de la fonction $x \mapsto Bx + c$, i.e., vérifie $x^* = Bx^* + c$ qui est équivalent à $Ax^* = b$ d'après (4.1)

La mise en oeuvre pratique d'une méthode itérative de la forme (4.2) nécessite la donnée d'un point de départ $x^{(0)}$ (en général, sauf si l'on possède des informations a priori sur la solution, on choisit le vecteur nul) et d'une

tolérance sur la solution que l'on cherche A calculer. On calcule ensuite les itérés $x^{(k)}, k = 1, 2, \ldots$ en utilisant la formule (4.2) jusqu'à ce que le résidu $b - Ax^{(k)}$ soit plus petit que la tolérance.

### 4.2.2 Convergence

Définition 4.1. La méthode itérative (4.2) pour résoudre $Ax = b$ est dite convergente *si* pour toute valeur initiale $x^{(0)} \in K^n$, on $a \lim_{k\to+\infty} x^{(k)} = A^{-1}b$.

Lemme 4.2. Si la méthode itérative (4.2) est convergente et si on note $x = A^{-1}b$ la solution, alors

$$x^{(k)} - x = B^t \left(x^{(0)} - x\right).$$

Démonstration. On a $c = (\mathrm{I}_n - B)\, A^{-1}b = (I_n - B)\, x$ d'où

$$x^{(k+1)} = Bx^{(k)} + (\mathrm{I}_n - B)\, x$$

ou encore $x^{(k+1)} - x = B\left(x^{(k)} - x\right)$ d'où le résultat.

Remarquons que $x^{(k)} - x$ représente l'erreur à la k-ième itération de sorte que la formule ci-dessus permet d'estimer cette erreur en fonction de l'erreur initiale.

Le résultat suivant nous donne des critères pour tester la convergence de la

méthode itérative (4.2).

Théorème 4.3. Les assertions suivantes sont équivalentes :

(i) La méthode itérative (4.2) est convergente

(ii) Pour tout $y \in K^n, \lim_{k\to+\infty} B^k y = 0$

(iii) Pour toute norme matricielle $\|.\|$ sur $\mathrm{M}_{n\times n}(\mathrm{K})$, on $a \lim_{k\to+\infty} \left\|B^t\right\| = 0$.

Démonstration. Admis pour ce cours.

En pratique, les caractérisations précédentes de la convergence d'une méthode itérative ne sont pas faciles a vérifier. On utilise plutôt le résultat suivant :

Théorème 4.4. Les assertions suivantes sont équivalentes :

(i) La méthode itérative (4.2) est convergente ;

(ii) $\rho(B) < 1$, où $\rho(B)$ désigne le rayon spectral de la matrice $B$, i.e., le maximum des modules des valeurs propres de $B$;

(iii) Il existe une norme matricielle $\|.\|$ sur $\mathrm{M}_{n\times n}(\mathrm{K})$ subordonnée à une norme vectorielle sur $\mathrm{K}^n$ telle que $\|B\| < 1$

Démonstration. Admis pour ce cours.

### 4.2.3 Vitesse de convergence

L'égalité $x^{(k)} - x = B^k \left(x^{(0)} - x\right)$ donnée précédemment implique que c'est la norme des puissances de la matrice $B$ qui va nous renseigner sur la vitesse de convergence de la méthode itérative. Nous définissons ici les outils per-

mettent de comparer les vitesses de convergence de différentes méthodes itératives.

Définition 4.5. Considérons le schéma itératif (4.2) convergent. Soit $\|.\|$ une norme matricielle sur $\mathrm{M}_n(\mathrm{K})$ et soit $k$ un entier tel que $\left\|B^k\right\| < 1$ (l'existence d'un tel $k$ découle du théorème 4.8 ). On appelle taux moyen de convergence associé à la norme $\|.|$ pour $k$ itérations de (4.2) le nombre positif

$$R_k(B) = -\ln\left(\left[\left\|B^k\right\|\right]^{\frac{1}{k}}\right)$$

Définition 4.6. Considérons des méthodes itératives consistantes ct convergentes :

(1) $x^{(k+1)} = B_1 x^{(k)} + c_1, \quad k = 1, 2, \ldots$

(2) $x^{(k+1)} = B_2 x^{(k)} + c_2, \quad k = 1, 2, \ldots$

Soit $k$ un entier tel que $\left\|B_1^k\right\| < 1$ et $\left\|B_2^k\right\| < 1$. On dit que (1) est plus rapide que (2) relativement a la norme $\|.\|$ si $R_k(B_1) \geq R_k(B_2)$.

En pratique le calcul des $R_k(B)$ est trop coûteux car il nécessite l'évaluation des $B^t$. On préfère donc estimer le taux asymptotique de convergence.

Définition 4.7. On appelle taux asymptotique de convergence le nombre

$$R_\infty(B) = \lim_{k \to +\infty} R_k(B) = -\ln(\rho(B))$$

Théorème 4.8. Avec les notations précédentes, une méthode itérative est d'autant plus rapide que son taux asymptotique de convergence est grand c'est-à-dire que $\rho(B)$ est petit.

## 4.3 Les méthodes itératives classiques

### 4.3.1 Principe

On considère un système linéaire $(S)$ : $Ax = b$ avec $A$ inversible. L'idée est de déduire un schéma itératif de la décomposition de $A$ sous la forme $\bar{A} = M - N$ ou $M$ est une matrice (par exemple $M$ diagonale, triangulaire, ... ). Le système ( $S$ ) s'écrit alors $Mx = Nx + b$ c'est-à-dire $x = Bx + c$ avec $B = M^{-1}N$ et $c = M^{-1}b$ et on considère le schéma itératif associe

$$x^{(0)} \in K^n, \quad Mx^{(k+1)} = Nx^{(k)} + b.$$

Nous allons maintenant considérer trois exemples classiques : les méthodes de Jacobi, Gauss-Seidel et de relaxation. Le point de départ de chacune de ces méthodes est l'unique décomposition de la matrice $A = (a_{i,j})_{1 \leq i,j \leq n}$ sous la forme $A = D - E - F$ avec

• $D = (d_{i,j})_{1 \leq i,j \leq n}$ diagonale, telle que $d_{i,i} = a_{i,i}$ et $d_{i,j} = 0$ pour $i \neq j$

• E=$(e_{ij})_1 \leq i_j \leq n$ triangulaire inférieure stricte telle que $e_{i,j} = -a_{i,j}$ si $i > j$ et $e_{i,j} = 0$ si $i \leq j$

• F=$(f_{i,j})_{1\leq i,j\leq n}$ triangulaire supérieure stricte telle que $f_{i,j} = -a_{ij}$ si $i < j$ et $f_{i,j} = 0$ si $i \geq j$.

**Exemple** : Considérons la matrice

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

La décomposition de $A$ sous la forme $A = D - E - F$ decrite ci-dessus s'écrit alors

$$\underbrace{\begin{pmatrix} 2 & -1 & 1 \\ 2 & 2 & 2 \\ -1 & -1 & 2 \end{pmatrix}}_{A} = \underbrace{\begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{pmatrix}}_{D} - \underbrace{\begin{pmatrix} 0 & 0 & 0 \\ -2 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix}}_{E} - \underbrace{\begin{pmatrix} 0 & 1 & -1 \\ 0 & 0 & -2 \\ 0 & 0 & 0 \end{pmatrix}}_{F}$$

On supposera de plus que $D$ est inversible et on distingue les trois méthodes suivantes :

•Méthode de Jacobi : $M = D, N = E + F$

• Méthode de Gauss-Seidel : $M = D - E, N = F$

• Méthode de relaxation : $M = \frac{1}{\omega}(D - \omega E), N = \left(\frac{1-\omega}{\omega}\right) D + F$ avec $\omega$ paramètre réel non nul.

On remarque que la méthode de Gauss-Seidel est un cas particulier de la méthode relaxation pour $\omega = 1$.

### 4.3.2 Méthode de Jacobi

**Description** : On considère un système linéaire ( $S$ ) : $Ax = b$ avec $A$ inversible. On pose $A = M - N$ avec $M = D$ inversible et $N = E + F$. Le schéma itératif s'écrit alors

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

Définition 4.9 La matrice $B_J = D^{-1}(E + F)$ s'appelle la matrice de Jacobi associée à $A$.

**Mise en œuvre et complexité arithmétique** : On se propose d'estimer le nombre d'opérations à virgule flottante nécessaires pour calculer $x^{(k+1)}$ à partir de $x^{(k)}$. On a $Dx^{(k+1)} = (E + F)x^{(k)} + b$

donc pour tout $i = 1, \ldots, n, \left(Dx^{(k+1)}\right)_i = \left((E + F)x^{(k)}\right)_i + b_i$ c'est-d-dire

$$a_{i,i}x_i^{(k+1)} = -\sum_{\substack{j=1 \\ j\neq i}}^{n} a_{i,j}x_j^{(k)} + b_i \Longleftrightarrow x_i^{(k+1)} = \frac{1}{a_{i,i}}\left[\left(-\sum_{\substack{j=1 \\ j\neq i}}^{n} a_{i,j}x_j^{(k)}\right) + b_i\right]$$

Pour calculer $x_i^{(k+1)}$ à partir de $x^{(k)}$, on a done besoin de $n - 1$ multiplications, $n - 1$ additions et 1 division soit $2n - 1$ opérations a virgule flottante. Par conséquent il nous faudra $n(2n - 1)$ opérations a virgule flottante pour calculer $x^{(k+1)}$ a partir de $x^{(k)}$ et pour $K$ itérations, on aura besoin de $Kn(2n-1)$ opérations à virgule flottante. Pour comparaison, pour

$n = 1000$, l'élimination de Gauss coûte environ $\frac{2}{3}n^3 = 6,6.10^8$ opérations à virgule flottante alors que par exemple $k = 100$ itérations de la méthode de Jacobi coûtent approximativement $2\mathrm{k}n^2 = 2.10^{\mathrm{s}}$ opérations a virgule flottante.

**Convergence** : D'après le théorème 4.4, on a le résultat suivant :

Théorème 4.10. La méthode de Jacobi converge si et seulement si $\rho(B_J) < 1$.

Exemple : Pour la matrice $A$ donnée par (4.3), on obtient :

$$B_J = D^{-1}(E+F) = \begin{pmatrix} \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{2} \end{pmatrix} \begin{pmatrix} 0 & 1 & -1 \\ -2 & 0 & -2 \\ 1 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & \frac{1}{2} & -\frac{1}{2} \\ -1 & 0 & -1 \\ \frac{1}{2} & \frac{1}{2} & 0 \end{pmatrix}$$

Les valeurs propres de la matrices $B_j$ sont 0 et $\pm\frac{i\sqrt{5}}{2}$. On a donc $\rho(B_J) = \frac{\sqrt{5}}{2} > 1$ et la méthode de Jacobi diverge.

### 4.3.3 Méthode de Gauss-Seidel

**Description** : On considère un système linéaire ( $S$ ) : $Ax = b$ avec $A$ inversible. On pose $A = M - N$ avec $M = D - E$ inversible et $N = F$. Le schéma itératif s'écrit alors

$$(D-E)x^{(k+1)} = Fx^{(k)} + b \Longleftrightarrow x^{(k+1)} = (D-E)^{-1}Fx^{(k)} + (D-E)^{-1}b.$$

Définition 4.11. La matrice $B_{GS} = (D - E)^{-1}F$ s'appelle la matrice de Gauss-Seidel associé à $A$.

**Mise en oeuvre et complexité arithmétique** : On se propose d'estimer le nombre d'opérations a virgule flottante nécessaires pour calculer $x^{(k+1)}$ à partir de $x^{(k)}$. On a $(D-$ E$)$ $x^{(k+1)} = Fx^{(k)} + b$ donc pour tout $i = 1, \ldots, n$, $\left((D - E)x^{(k+1)}\right)_i = \left(Fx^{(k)}\right)_i + b_i$ c'est-à- dire

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

ce qui entraîne

$$x_1^{(k+1)} = \frac{1}{a_{1,1}}\left[-\sum_{j=2}^{n} a_{1,j}x_j^{(k)} + b_1\right]$$

et pour $i = 2, \ldots, n$

$$x_i^{(k+1)} = \frac{1}{a_{i,i}}\left[-\sum_{j=1}^{i-1} a_{i,j}x_j^{(k)} + b_i - \sum_{j=i+1}^{n} a_{i,j}x_j^{(k)}\right].$$

La complexité arithmétique de la méthode de Gauss-Seidel est la même que celle de la méthode de Jacobi. Cependant, on peut remarquer que la méthode de Gauss-Seidel est plus intéressante en ce qui concerne la gestion de la mémoire. En effet, on peut écraser au fur et à mesure la valeur de $x_i^{(k)}$ et ne stocker au cours des calculs qu'un seul vecteur de taille $n$, e. $g.$, le vecteur $\left(x_1^{(k+1)} \ldots x_i^{(k+1)} x_{i+1}^{(k)} \ldots x_n^{(k)}\right)^T$, au lieu de deux vecteurs pour la

méthode de Jacobi.

**Convergence** : D'après le théorème 4.4, on a le résultat suivant :

Théorème 4.12. La méthode de Gauss-Seidel converge si et seulement si $\rho(B_{GS}) < 1$.

**Exemple** : Pour la matrice $A$ donnée par (4.3), on obtient :

$$B_{GS} = (D-E)^{-1}F = \begin{pmatrix} 2 & 0 & 0 \\ 2 & 2 & 0 \\ -1 & -1 & 2 \end{pmatrix}^{-1} \begin{pmatrix} 0 & 1 & -1 \\ 0 & 0 & -2 \\ 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & 0 & 0 \\ -\frac{1}{2} & \frac{1}{2} & 0 \\ 0 & \frac{1}{4} & \frac{1}{2} \end{pmatrix} \begin{pmatrix} 0 & 1 & -1 \\ 0 & 0 & -2 \\ 0 & 0 & 0 \end{pmatrix}$$

d'où

$$B_{GS} = \begin{pmatrix} 0 & \frac{1}{2} & -\frac{1}{2} \\ 0 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & 0 & -\frac{1}{2} \end{pmatrix}$$

Les valeurs propres de la matrices $B_{GS}$ sont 0 et $-\frac{1}{2}$ (de multiplicité 2 ).

On a donc $\rho(B_{GS}) = \frac{1}{2} < 1$ donc la méthode de Gauss-Seidel converge.

### 4.3.4 Méthode de relaxation

On considère un système linéaire ( $S$ ) : $Ax = b$ avec $A$ inversible. Soit $\omega$ un paramètre réel non nul. On pose $A = M - N$ avec $M = \frac{1}{\omega}(D - \omega E)$ inversible et $N = \left(\frac{1-\omega}{\omega}\right) D + F$. Le schéma itératif s'écrit alors

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

qui est équivalent à:

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

Définition 4.13. La matrice $B_R(\omega) = (D-\omega E)^{-1}[(1-\omega)D+\omega F]$ s'appelle la matrice de relaxation associée a $A$ et $\omega$ est le facteur de relaxation. Si$\omega < 1$, on parle de sous-relaxation, si $\omega = 1$, on retrouve la méthode de Gauss-Seidel et si $\omega > 1$, on parle de sur-relaxation.

D'après le théorème 4.4, on a le résultat suivant :

Théorème 4.14. La méthode de relaxation converge si et seulement si $\rho\left(B_n(\omega)\right) < 1$.

**Exemple** : Pour la matrice $A$ donné par (4.3), on obtient :

$$B_R(\omega) = \begin{pmatrix} 1-\omega & \frac{1}{2}\omega & -\frac{1}{2}\omega \\ \omega(\omega-1) & -\frac{1}{2}\omega^2 + 1 - \omega & \frac{1}{2}\omega^2 - \omega \\ \frac{1}{2}\omega(\omega-1)^2 & -\frac{1}{4}\omega^3 - \frac{1}{4}\omega^2 + \frac{1}{2}\omega & \frac{1}{4}\omega^3 - \frac{3}{4}\omega^2 + 1 - \omega \end{pmatrix}$$

Les valeurs propres de la matrice $B_R(\omega)$ dépendent en général de $\omega$ donc la convergence de la méthode de relaxation dépendra aussi de la valeur de $\bar{\omega}$.

### 4.3.5 Résultats de convergence dans des cas particuliers

On s'intéresse tout d'abord au cas des matrices symétriques définies positives.

Théorème 4.15. Soit $A$ une matrice symétrique définie positive et écrivons $A = M - N$ avec $M$ inversible et $M^T + N$ définie positive. Alors la méthode itérative

$$x^{(0)} \in K^n, \quad x^{(k+1)} = M^{-1}Nx^{(k)} + M^{-1}b$$

converge.

Démonstration. Admis pour ce cours.

Corollaire 4.16. Soit $A$ une matrice symétrique définie positive. Alors la méthode de Gauss Seidel converge.

Démonstration. Pour la méthode de Gauss-Seidel, on a $M = D - E$ et $N = F$. La matrice $M$ est inversible car $A$ est supposé définie positive et est donc inversible (voir la section 2.3 ). On a de plus $M^T + N = D - E^T + F$. Or $A$ étant supposé symétrique, on a $E^T = F$ d'où $M^T + N = D$. La matrice $M^T + N$ est donc définie positive car, pour tout $i = 1, \ldots, n$ $\langle De_i, e_i \rangle = a_{i,i}$ et $a_{i,i} > 0$ puisque $A$ est définie positive (voir la section 2.3 ). Le théorème 4.15 précédent permet alors de conclure.

Considérons maintenant le cas des matrices à diagonale strictement dominante.

Définition 4.17. Une matrice $A = (a_{i,j})_{1 \leq i,j \leq n}$ est dite à diagonale strictement dominante si :

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

Par exemple, la matrice du système linéaire obtenu à la fin de la section 4.1 pour la résolution d'un système d'équations aux dérivées partielles est à diagonale strictement dominante.

Théorème 4.18. Soit $A$ une matrice à diagonale strictement dominante. Alors $A$ est inversible et les méthodes de Jacobi et de Gauss-Seidel convergent toutes les deux.

Démonstration. Admis pour ce cours.

## 4.4 Méthode du gradient conjugué

Il s'agit d'une méthode itérative qui permet de résoudre un système linéaire $(S) : Ax = b$ lorsque $A$ est une matrice symétrique et définie positive. Dans cette méthode, la matrice $A$ du système intervient une seule fois a chaque itération, lorsqu'on calcule son produit par un vecteur. Par conséquent, la méthode du gradient conjugué est particulièrement bien adaptée aux systèmes creux et de grande taille. Pour ce type de systèmes, la méthode du

gradient conjugué est souvent plus efficace que les méthodes déjà présentées, à la fois en termes de complexité arithmétique et en termes d'espace mémoire nécessaire : elle est donc très utilisée en pratique.

Soit $(S) : Ax = b$ avec $A \in M_{n\times n}(R)$ symétrique et définie positive, et $b \in R^n$. La méthode du gradient conjugué construit une suite de vecteurs $\left(x^{(k)}\right)_{k=0,1,...}$ telle que $x^{(m)} = A^{-1}b$ pour un indice $m \leq n$. En principe, il s'agit donc d'une méthode exacte. En pratique, a cause des erreurs numériques, le gradient conjugué est considéré comme une méthode itérative. Dans les applications, le nombre d'itérations nécessaires pour atteindre la précision voulue est significativement plus petit que la taille n du système, en particulier dans le cas où on utilise des techniques de préconditionnement. Dans la suite, le gradient conjugué sera présenté comme un cas particulier d'une famille de méthodes itératives connue sous le nom de méthodes du gradient.

Définition 4.19. Soit $A \in M_{n\times n}(R)$ symétrique et définie positive. On définit la fonction

$$\|\cdot\|_A : R^n \longrightarrow R_+ par \|x\|_A = \sqrt{x^T Ax}$$

Proposition 4.20. La fonction $\|\cdot\|_A$ est une norme vectorielle.

Démonstration. En utilisant le fait que $A$ est symétrique et définie positive,

on montre que la fonction $\|\cdot\|_A$ vérifie les trois propriétés de la définition 3.1, avec $E = R^n$ et K $= R$ :

•$\forall x \in R^n, \|x\|_A = 0 \Rightarrow x^T Ax = 0 \Rightarrow x = 0$, car $A$ est définie positive

•$\forall \lambda \in R, \forall x \in R^n, \|\lambda x\|_A = \sqrt{(\lambda x)^T A(\lambda x)} = \sqrt{\lambda^2 x^T Ax} = |\lambda|\sqrt{x^T Ax} = |\lambda|\|x\|_A$

•$\forall x, y \in R^n, \|x+y\|_A = \sqrt{(x+y)^T A(x+y)} = \sqrt{x^T Ax + y^T Ay} \leq \sqrt{x^T Ax} + \sqrt{x^T Ax}$

$= \|x\|_A + \|y\|_A$

Définition 4.21. Soit $A \in M_{n\times n}(R)$ symétrique et définie positive. On dit que les vecteurs $u$ et $v$ de $R^n$ sont $A$ -conjugués si $u^T Av = 0$.

On remarque que l'application $(u, v) \in (R^n)^2 \longmapsto u^T Av \in R$ est un produit scalaire sur $R^n$. Deux vecteurs $u$ et $v$ de $R^n$ sont donc $A$ -conjugués s'ils sont orthogonaux (pour ce produit scalaire).

### 4.4.1 Méthodes du gradient

On considère le problème suivant : minimiser sur $R^n$ la fonction $\phi$ définie par

$$\phi(x) = \frac{1}{2}x^T Ax - b^T x$$

où la matrice $A$ est symétrique et définie positive. L'intérêt de la question vient du fait que le minimum de $\phi$ est atteint pour $x^* = A^{-1}b$, et cette solution est unique. En effet, le gradient de $\phi$ en $x = (x_1 \dots x_n)^T \in R^n$ est

le vecteur

$$\nabla\phi(x) = \left(\frac{\partial\phi}{\partial x_1}\frac{\partial\phi}{\partial x_2}\cdots\frac{\partial\phi}{\partial x_n}\right)^T = \frac{1}{2}Ax + \frac{1}{2}A^Tx - b = Ax - b$$

qui s'annule seulement pour $x^* = A^{-1}b$. Notons que dans ce calcul on a utilise le fait que $A$ est une matrice symétrique. Le vecteur $x^* = A^{-1}b$ est donc l'unique point critique de $\phi$ et, puisque $A$ est définie positive, il s'agit d'un minimum global (admis).

On peut donc conclure de ce qui précède que minimiser $\phi$ sur $R^n$ et résoudre le système linéaire $Ax = b$ sont deux problèmes équivalents.

Définition 4.22. Soit $(S) : Ax = b$ avec $A \in M_{n\times n}(R)$ symétrique et définie positive, *et* $b \in R^n$. La quantité $r(x)$ définie par

$$r(x) = b - Ax = -\nabla\phi(x)$$

est appelée résidu du système $(S)$ en $x$. En particulier, on notera

$$r^{(k)} = b - Ax^{(k)} = -\nabla\phi\left(x^{(k)}\right)$$

le résidu à l'itération $k$.

Remarque : on rappelle que la valeur $\nabla\phi(x)$ du gradient de $\phi$ en $x$ donne la direction de plus forte pente pour la fonction $\phi$ au point $x$.

Les méthodes du gradient procèdent généralement en choisissant à l'étape $k$ une direction de descente pour $\phi$ c'est-à-dire un vecteur $p^{(k)} \in R^n$ tel que

$p^{(k)T}\nabla\phi\left(x^{(k)}\right) < 0$. Pour calculer $x^{(k+1)}$ à partir de $x^{(k)}$, on minimise alors la fonction $\phi$ sur la droite de vecteur directeur $p^{(k)}$ et passant par $x^{(k)}$ : on choisit donc

$$x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)}$$

où $\alpha_k \in R$ est tel que

$$\phi\left(x^{(k+1)}\right) = \min_{\alpha\in R} \phi\left(x^{(k)} + \alpha p^{(k)}\right)$$

En dérivant $\phi\left(x^{(k)} + \alpha p^{(k)}\right)$ par rapport à $\alpha$, on obtient

$$\begin{aligned}
\tfrac{\partial}{\partial\alpha} \quad \left(\phi\left(x^{(k)} + \alpha p^{(k)}\right)\right) &= \tfrac{\partial}{\partial\alpha}\left(\tfrac{1}{2}\left(x^{(k)} + \alpha p^{(k)}\right)^T A\left(x^{(k)} + \alpha p^{(k)}\right) - \left(x^{(k)} + \alpha p^{(k)}\right)^T b\right) \\
&= \tfrac{\partial}{\partial\alpha}\left(\tfrac{1}{2}\left(x^{(k)T}Ax^{(k)} + \alpha p^{(k)T}Ax^{(k)} + \alpha x^{(k)T}Ap^{(k)} + \alpha^2 p^{(k)T}Ap^{(k)}\right) - x^{(k)T}b - \alpha p^{(k)T}b\right) \\
&= \tfrac{1}{2}\left(p^{(k)T}Ax^{(k)} + x^{(k)T}Ap^{(k)} + 2\alpha p^{(k)T}Ap^{(k)}\right) - p^{(k)T}b \\
&= \left(x^{(k)} + \alpha p^{(k)}\right)^T Ap^{(k)} - b^T p^{(k)}
\end{aligned}$$

car $A$ étant symétrique, $p^{(k)T}Ax^{(k)} = x^{(k)T}Ap^{(k)}$. En imposant $\tfrac{\partial}{\partial\alpha}\left(\phi\left(x^{(k)} + \alpha p^{(k)}\right)\right) = 0$, on trouve alors

$$\alpha_k = \frac{\left(b - Ax^{(k)}\right)^T p^{(k)}}{p^{(k)T}Ap^{(k)}} = \frac{r^{(k)T}p^{(k)}}{p^{(k)T}Ap^{(k)}} \quad (4.5)$$

En particulier, on observe que $\alpha_k > 0$ puisque $r^{(k)T}p^{(k)} = -p^{(k)T}\nabla\phi\left(x^{(k)}\right) > 0$.

Proposition 4.23. A chaque itération, le résidu $r^{(k+1)}$ est orthogonal à la direction de descente $p^{(k)}$ utilisée à l'étape précédente, i.e., $r^{(k+1)T}p^{(k)} = 0$.

Démonstration. D'après (4.4), pour $k \in N$, on a $b - Ax^{(k+1)} = b - Ax^{(k)} - \alpha_k Ap^{(k)}$ et donc $r^{(k+1)} = r^{(k)} - \alpha_k Ap^{(k)}$. D'où , à partir de (4.5), $r^{(k+1)T}p^{(k)} = \left(r^{(k)} - \alpha_k Ap^{(k)}\right)^T p^{(k)} = r^{(k)T}p^{(k)} - \alpha_k p^{(k)T}Ap^{(k)} = 0$.

### 4.4.2 Méthode de la plus forte pente

Dans cette méthode, on choisit à chaque itération la direction de descente

$$p^{(k)} = r^{(k)} = -\nabla\phi\left(x^{(k)}\right)$$

c'est-à-dire la direction de plus forte pente pour $\phi$ au point $x^{(k)}$. La proposition 4.23 implique alors qu'à chaque itération, on choisit une direction de descente orthogonale à la précédente, i.e., $p^{(k+1)T}p^{(k)} = 0$.

Théorème 4.24. Pour la méthode de la plus forte pente on $a$, à l'itération $k$ :

$$\left\|x^* - x^{(k)}\right\|_A \leq \left(\frac{Cond_2(A) - 1}{Cond_2(A) + 1}\right)^k \left\|x^* - x^{(0)}\right\|_A .$$

Démonstration. Admis pour ce cours.

Le choix $p^{(k)} = -\nabla\phi\left(x^{(k)}\right)$ peut paraître intuitivement assez efficace pour minimiser $\phi$ vu qu'on se déplace le long de la direction de plus forte décroissance de la fonction. Pourtant, le théorème 4.24 suggère que dans certains cas la

convergence de cette méthode pourrait être lente, notamment si la matrice $A$ est mal conditionnée, *i.e.*, lorsque $Cond_2(A) \gg 1$ (Voir Chapitre 3 ). Soient $\lambda_1 \geq \cdots \geq \lambda_n > 0$ les valeurs propres de $A$ : géométriquement, le fait que le nombre de conditionnement Cond $_2(A) = \lambda_1/\lambda_n$ (voir Proposition 3.6 ) soit grand est équivalent au fait que les courbes de niveau de $\phi$ soient des hyperellipsoïdes très allongés.

**Exemple** : Soit $A$ une matrice symétrique de taille $2 \times 2$ ayant pour valeurs propres $\lambda_1 = e^4 \geq \lambda_2 = e^{-2} > 0$, et $b$ le vecteur $(1)^T$. Le conditionnement de $A$ est alors Cond $_2(A) \approx 403,43$ de sorte que $A$ est mal conditionnée. La figure 4.1 montre la surface définie dans $R^3$ par la fonction $\phi$ : on voit que cette surface ressemble à une vallée au fond assez plat.

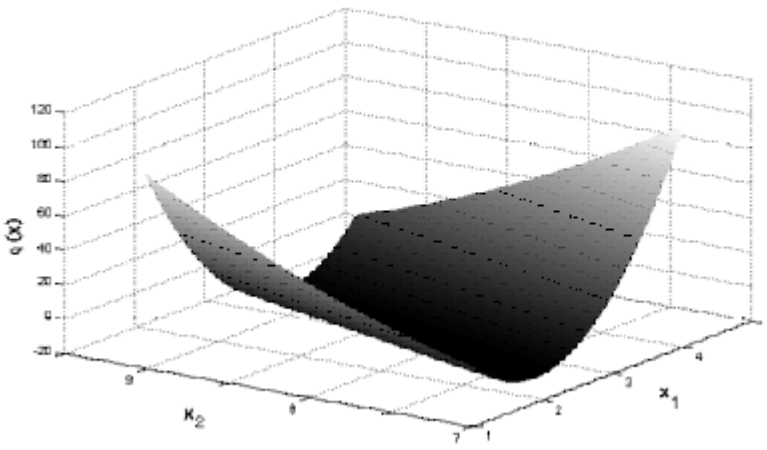
La solution du système est approximativement $x^* \approx (8,7361 \quad 2,3478)^T$. On choisit un vecteur initial $x^{(0)} = (8 \quad 2)^T$ assez proche de $x^*$ et on effectue 10 itérations de la méthode de la plus forte pente. La figure 4.2 montre, en échelle logarithmique, les résidus relatifs $\frac{\left\|r^{(k)}\right\|_2}{\|b\|_2}$ obtenus à chaque itération : on constate que 10 itérations ne suffisent pas à atteindre un résidu de norme inférieure à $10^{-6}$. Géométriquement, ceci peut s'expliquer par le fait qu'à chaque itération de la méthode de la plus forte pente on choisit une direction orthogonale à la précédente et donc on va rebondir sur les parois de la vallée, ce qui fait qu'on s'approche très lentement du minimum situé

sur le fond.

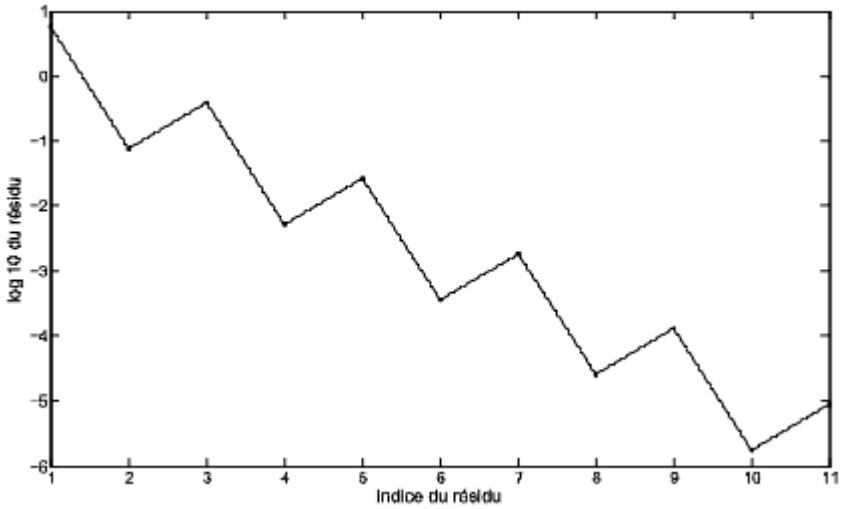
Par contre, pour ce même exemple, la méthode du gradient conjugué converge en 2 itérations avec un résidu relatif comparable à la précision machine (voir la section suivante).

### 4.4.3 Gradient conjugué

Pour la méthode du gradient conjugué, le choix de $p^{(k)}$ dans le schéma itératif (4.4) se fait en tenant compte des directions $p^{(j)}, j = 0, 1, \ldots, k-1$, calculées aux itérations précédentes.




Surface définie par la fonction $\phi$




Résidus relatifs calculés à chaque itération de la méthode de la plus forte pente

On définit

$$p^{(k)} = \begin{cases} r^{(0)} & sik = 0 \\ r^{(k)} + \beta_k p^{(k-1)} & sik \geq 1 \end{cases} \quad (4.6)$$

où $\beta_k \in R$ est tel que

$$p^{(k)T} A p^{(k-1)} = 0 \quad (4.7)$$

La condition (4.7) revient à dire que les directions de descente $p^{(k-1)}$ et $p^{(k)}$ calculées à deux itérations consécutives de la méthode du gradient conjugué

sont $A$ -conjuguées.

En utilisant (4.6) et (4.7), on peut écrire $\beta_k$ sous la forme

$$\beta_k = -\frac{r^{(k)T} A p^{(k-1)}}{p^{(k-1)T} A p^{(k-1)}}$$

On vérifie alors que $p^{(k)}$ est effectivement une direction de descente pour $\phi$ : si $r^{(k)} \neq 0$ c'est-à-dire $x^{(k)} \neq x^*$, on a

$$p^{(k)T} \nabla \phi \left( x^{(k)} \right) = -p^{(k)T} r^{(k)} = -r^{(k)T} r^{(k)} - \beta_k p^{(k-1)T} r^{(k)} = -r^{(k)T} r^{(k)} < 0$$

En particulier, on constate que $p^{(k)T} r^{(k)} = r^{(k)T} r^{(k)}$ donc l'expression (4.5) pour $\alpha_k$ devient

$$\alpha_k = \frac{r^{(k)T} r^{(k)}}{p^{(k)T} A p^{(k)}}.$$

Lemme 4.25.*Å* A chaque itération, le résidu $r^{(k)}$ est orthogonal au résidu $r^{(k-1)}$ calculé *à* l'itération précédente, i.e., $r^{(k)T} r^{(k-1)} = 0$.

Démonstration. On a à la fois $r^{(k)T} r^{(k-1)} = r^{(k)T} p^{(k-1)} - \beta_{k-1} r^{(k)T} p^{(k-2)} = -\beta_{k-1} r^{(k)T} p^{(k-2)}$ et $r^{(k)T} p^{(k-2)} = r^{(k-1)T} p^{(k-2)} - \alpha_k p^{(k-1)T} A p^{(k-2)} = 0$, d'où $r^{(k)T} r^{(k-1)} = 0$.

Lemme 4.26. La quantité $\beta_k$ peut s'écrire sous la forme

$$\beta_k = \frac{r^{(k)T} r^{(k)}}{r^{(k-1)T} r^{(k-1)}}.$$

Démonstration. Pour démontrer cette formule, on écrit la quantité $p^{(k)T} r^{(k-1)}$ de deux manières différentes. On a

$$p^{(k)T} r^{(k-1)} = r^{(k)T} r^{(k-1)} + \beta_k p^{(k-1)T} r^{(k-1)} = \beta_k p^{(k-1)T} r^{(k-1)} = \beta_k r^{(k-1)T} r^{(k-1)}$$

et

$$p^{(k)T} r^{(k-1)} = p^{(k)T} r^{(k)} + \alpha_{k-1} p^{(k)T} A p^{(k-1)} = p^{(k)T} r^{(k)} = r^{(k)T} r^{(k)}$$

d'où le résultat.

Théorème 4.27. Soit $\mathcal{S}_k$ le sous-espace vectoriel de $R^n$ engendré par les vecteurs $p^{(0)}, \ldots, p^{(k-1)}$ Alors le vecteur $x^{(k)}$ défini par la méthode du gradient conjugué à l'itération $k$ minimise la fonction $\phi$ sur $\mathcal{S}_k$ :

$$\phi\left(x^{(k)}\right) = \min_{x \in \mathcal{S}_k} \phi(x), \quad k \geq 1.$$

Démonstration. Admis pour ce cours.

Théorème 4.28. Soit $r^{(0)} \neq 0$ et $h \geq 1$ tels que $r^{(k)} \neq 0$ pour tout $k \leq h$. Alors pour $k, j \in \{0, \ldots, h\}$ avec $k \neq j$, on $a$

$$r^{(k)T} r^{(j)} = 0 \quad et \quad p^{(k)T} A p^{(j)} = 0.$$

Autrement dit, dans la méthode du gradient conjugué, les résidus forment un

ensemble de vecteurs orthogonaux et les directions de descente $p^{(k)}$ forment un ensemble de vecteurs A-conjugués.

Démonstration. Par récurrence sur $h$ (exercice).

Corollaire 4.29. Il existe $m \leq n$ tel que $r^{(m)} = 0$. Autrement dit, le gradient conjugué calcule la solution $x^* = A^{-1}b$ en au plus $n$ itérations.

Pour récapituler, l'algorithme du gradient conjugué se déroule de la manière suivante :

Entrée : $A \in M_{n\times n}(R)$ symétrique et définie positive, $b \in R^n$, et $x^{(0)} \in R^n$

Sortie : $x^* \in R^n$ tel que $Ax^* = b$

1. $k = 0$

2. $r^{(0)} = b - Ax^{(0)}$

3. Tant que $r^{(k)} \neq 0$, faire :

•$Si$k=0, $alors faire$

$p^{(0)} = r^{(0)}$

$sinon faire$ :

$\beta_k = r^{(k)T}r^{(k)}/r^{(k-1)T}r^{(k-1)}$

$p^{(k)} = r^{(k)} + \beta_k p^{(k-1)}$

•$\alpha_k = r^{(k)T}r^{(k)}/p^{(k)T}Ap^{(k)}$

•$x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)}$

•$r^{(k+1)} = r^{(k)} - \alpha_k Ap^{(k)}$

•$k = k + 1$

4. Retourner $x^* = x^{(k)}$

Les propriétés de convergence de la méthode du gradient conjugué sont données par le résultat suivant (à comparer au théorème 4.24 ) :

Théorème 4.30. Pour la méthode du gradient conjugué, on a, à l'itération $k$ :

$$\left\|x^* - x^{(k)}\right\|_A \leq 2\left(\frac{\sqrt{Cond_2(A)} - 1}{\sqrt{Cond_2(A)} + 1}\right)^k \left\|x^* - x^{(0)}\right\|_A$$

Démonstration. Admis pour ce cours.

**Quelques remarques** :

• Dans la pratique, le critère d'arrêt $r^{(k)} = 0$ est remplacé par $\left\|r^{(k)}\right\|_2 < \epsilon_M \|b\|_2$, et $k$ est borné par $k_{\max} \ll n$. En effet, la méthode du gradient conjugué est souvent appliquée à des systèmes de grande taille, où on espère atteindre une bonne approximation de la solution $x^*$ après un nombre d'itérations significativement plus petit que $n$.

• En principe, le vecteur initial $x^{(0)}$ peut être choisi de manière arbitraire, par exemple comme le vecteur nul. Évidemment, le choix de $x^{(0)}$ a un effet sur le nombre d'itérations nécessaires pour atteindre la solution.

• À chaque itération, l'opération la plus coûteuse du point de vue de la complexité arithmétique est la multiplication matrice-vecteur $Ap^{(k)}$, qui

nécessite en général $n^2$ opérations. On peut donc estimer qu'asymptotiquement le coût de la méthode du gradient conjugué est de l'ordre de $k_{\max} n^2$ opérations. De plus, si la matrice $A$ est creuse, comme c'est souvent le cas dans les applications ( *e.g.*, résolution d'équations aux dérivées partielles par discrétisation-voir Section 4.1 ), le produit matrice-vecteur $Ap^{(k)}$ peut se faire en seulement $n$ opérations, ce qui rend la méthode du gradient conjugué plus avantageuse qu'une méthode directe comme celle de Cholesky (voir la proposition 2.18 ).

• Si on veut résoudre un système linéaire $Ax = b$ où la matrice $A$ est inversible mais n'est pas symétrique et définie positive, on peut toujours appliquer la méthode du gradient conjugué au système équivalent $A^T Ax = A^T b$ (méthode des équations normales). Cependant, cette approche n'est pas recommandée lorsque $A$ est mal conditionnée car le passage aux équations normales élève le conditionnement de la matrice au carré. Dans ce cas, il existe des versions de la méthode du gradient conjugué spécialement adaptées aux matrices non symétriques : l'une des plus utilisées est la méthode GMRES.

### 4.4.4 Gradient conjugué avec préconditionnement

Le préconditionnement est une technique qui vise à accélérer la convergence d'une méthode itérative. On rappelle que, dans le cas de la méthode du gradient conjugué, la convergence est très rapide si la matrice $A$ est proche de la matrice identité, ou si ses valeurs propres sont bien regroupées (voir Théorème 4.30 ).

Description : On considère un système linéaire $(S) : Ax = b$ avec $A$ symétrique et définie positive. Étant donnée une matrice $C \in M_{n\times n}(R)$ inversible, on définit le système transformé

$$(\tilde{S}) : \tilde{A}\tilde{x} = \tilde{b}, \quad avec \quad \tilde{A} = C^{-1}A\left(C^{-1}\right)^T, \quad \tilde{x} = C^T x, \quad et \quad \tilde{b} = C^{-1}b$$

On remarque que $\tilde{A}$ est aussi symétrique et définie positive donc on peut appliquer la méthode du gradient conjugué à $(\tilde{S})$.

Définition 4.31. La matrice $M = CC^T$ est appelée préconditionneur du système $(S)$.

Le choix du préconditionneur $M$ est généralement fait de sorte que la matrice $\tilde{A}$ soit mieux conditionnée que $A$, ou idéalement proche de la matrice identité, pour que la méthode du gradient conjugué appliquée à $(\tilde{S})$ converge rapidement,

•M soit  facilement " inversible car dans l'algorithme il faut résoudre un système linéaire de matrice $M$ à chaque itération (voir ci-dessous), donc cette opération doit pouvoir être effectuée de manière stable et rapide, idéalement en un nombre d'opérations de l'ordre de $n$.

Pour écrire l'algorithme du gradient conjugué préconditionné, on observe que, si on note $s^{(k)}$ le résidu de la méthode préconditionnée à l'itération $k$, on a, avec les notations précédentes, $s^{(k)} = C^{-1}r^{(k)}$, et donc $s^{(k)T}s^{(k)} = r^{(k)T}M^{-1}r^{(k)}$. Si $z^{(k)}$ est tel que $Mz^{(k)} = r^{(k)}$, alors $s^{(k)T}s^{(k)} = z^{(k)T}r^{(k)}$. On obtient donc l'algorithme suivant :

Entrée : $A \in M_{n\times n}(R)$ symétrique et définie positive, $b \in R^n, x^{(0)} \in R^n$, et un préconditionneur $M \in M_{n\times n}(R)$ symétrique et défini positif.

Sortie : $x^* \in R^n$ tel que $Ax^* = b$

1. $k = 0$

2. $r^{(0)} = b - Ax^{(0)}$

3. Tant que $r^{(k)} \neq 0$, faire :

• résoudre le système linéaire $Mz^{(k)} = r^{(k)}$;

•Si $k = 0$, alors faire :

$\mathrm{p}^{(0)} = z^{(0)} sinonfaire$ :

$\beta_k = z^{(k)T} r^{(k)} / z^{(k-1)T} r^{(k-1)}$

$p^{(k)} = z^{(k)} + \beta_k p^{(k-1)}$

•$\alpha_k = z^{(k)T} r^{(k)} / p^{(k)T} A p^{(k)}$

•$x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)}$

•$r^{(k+1)} = r^{(k)} - \alpha_k A p^{(k)}$

•$k = k + 1$

4. *Retourner*x$^* = x^{(k)}$.

**Remarque** : la matrice $C$ est utilisée dans la description théorique de la méthode mais n'apparait pas explicitement dans l'algorithme.

Le choix d'un préconditionneur est un problème délicat et il existe une vaste littérature à ce sujet. Ici nous présentons deux exemples simples qui peuvent s'appliquer de manière assez générale.

**Exemple 1** : Préconditionnement diagonal. On note $A = (a_{i,j})_{1 \leq i,j \leq n}$, et on choisit le conditionneur $M = (m_{i,j})_{1 \leq i,j \leq n}$ défini par

$$m_{i,j} = \begin{cases} a_{i,i} & sii = j \\ 0 & sii \neq j \end{cases}$$

**Exemple 2** : Préconditionnement de Cholesky incomplet. On choisit $M = LL^T$, où $L = (\ell_{i,j})_{1 \leq i,j \leq n}$ est une matrice triangulaire inférieure définie de

la manière suivante :

$$\begin{aligned}
\ell_{i,i} &= \sqrt{a_{i,i} - \sum_{r=1}^{i-1} \ell_{i,r}^2}, \quad i = 1, \ldots, n \\
\ell_{i,j} &= \begin{cases} 0 & \text{si } a_{i,j} = 0 \\ \frac{1}{\ell_{i,j}} \left( a_{i,j} - \sum_{r=1}^{j-1} \ell_{i,r} \ell_{j,r} \right) & \text{si } a_{i,j} \neq 0, \end{cases} \qquad j = 1, \ldots, i-1, \quad i = 2, \ldots, n
\end{aligned}$$

On remarque que la matrice L est définie de manière à préserver l'éventuelle structure creuse de $A$. Les éléments non nuls de $L$ sont calculés comme pour le facteur de Cholesky de $A$ (voir l'algorithme de Cholesky dans la section 2.3).

# Chapitre 5

# Calcul de Valeurs Propres et Vecteurs Propres

## 5.1 Rappels

Soit $A$ une matrice carrée d'ordre $N$ à coefficients réels ou complexes et $\lambda$ un scalaire réel ou complexe. Alors sont équivalentes

$*$ $\exists X \in R^N$ ( ou $C^N, X \neq 0, AX = \lambda X$ (1)

$**$ $A - \lambda I$ est une matrice non inversible (2)

$***$ $det(A - \lambda I) = 0$ (3)

Si $\lambda$ satisfait à une de ces propriétés, on dit que $\lambda$ est valeur propre de $A$ et tout vecteur satisfaisant a (1) est appelé vecteur propre de $A$ associé à la valeur propre $\lambda$.

On vérifie que $\lambda \rightarrow \det\,(A - \lambda I)$ est un polynôme de degré $N$ (dit polynôme caractéristique de $A$ ) dont le coefficient de plus haut degré est $(-1)^N$. Ainsi les valeurs propres de $A$ sont les racines d'un polynôme de degré $N$. Donc $A$ admet exactement $N$ valeurs propres complexes (en comptant les racines multiples éventuelles avec leur multiplicité). D'autre part, la recherche de valeurs propres étant équivalente à la recherche des racines d'un polynôme, les méthodes ne peuvent être qu'itératives et non directes, puisque, d'après le théorème d'Abel, on ne peut "résoudre par radicaux" un polynôme quelconque de degré supérieur ou égal à 5.

On peut penser que le calcul numérique de valeurs propres est un cas particulier du calcul numérique des racines d'un polynôme. La difficulté d'évaluer numériquement le polynôme $\det\,(A - \lambda I)$ fait que les algorithmes reposant sur cette idée sont en général peu performants. C'est en fait, plutôt l'inverse qui se produit à savoir que le calcul des racines de

$$\lambda^N + a_1\lambda^{N-1} + \ldots + a_{\mathrm{N}-1}\lambda + a_{\mathrm{N}}$$

peut se faire en appliquant un algorithme de recherche de valeurs propre s à la matrice-compagnon suivante dont le polynôme caractéristique est, au

facteur $(-1)^N$ près, le polynôme précédent.

$$\begin{bmatrix} -a_1 & -a_2 & \cdots & \cdots & \cdots & \cdots & -a_N \\ 1 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & 1 & 0 & & & & \vdots \\ \vdots & \ddots & \ddots & \ddots & & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & & \vdots \\ \vdots & & & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & 0 & 1 & 0 \end{bmatrix}$$

**Remarque 1** On vérifie immédiatement que les valeur s propres d'une matrice triangulaire sont les éléments diagonaux de cette matrice puisqu'on a alors

$$det(A - \lambda I) = \prod_{i=1}^{N} (a_{ii} - \lambda)$$

Plusieurs des algorithme gui vont suivre consisteront précisément a "rendre" la matrice $A$ initiale triangulaire ou même diagonale, ce en construisant une suite $A_0 = A, A_1, \ldots, A_k$ de matrices ayant les mêmes valeur s propre s et tendant vers une matrice triangulaire ou diagonale. Pour cela, il est nécessaire de rappeler que la transformation de base laissant invariant l'ensemble des valeurs propres est la similitude.

Définition 1 Deux matrices $A$ et $A'$ sont dites semblables s'il existe une

matrice inversible $P$ telle que

$$A' = P^{-1}AP$$

On dit que $A$ est diagonalisable si elle est semblable à une matrice diagonale. Deux matrices semblables ont les mêmes valeurs propres.

Proposition 1 Une matrice $A$ est diagonalisable si et seulement s'il existe une base de $R^N$ (ou $\mathrm{C}^N$ ) formée de vecteurs propres de $A$.

Démonstration : Supposons $A' = P^{-1}AP$ diagonale. Notons $e_i$ le i-ème vecteur de la base canonique de $R^N$, soit

$$e_i = (0, \ldots, 1, 0, \ldots, 0)$$

Notons $\lambda_i, i = 1, \ldots, N$ les élément s diagonaux de $A'$. Alors

$$A'e_i = \lambda_i e_i = P^{-1}APe_i \Longrightarrow \lambda_i Pe_i = APe_i$$

Donc $\{Pe_i, i = 1, \ldots, N\}$ sont des vecteurs propres de $A$. Comme $P$ est inversible, elle transforme une base en une base. On obtient ainsi une base de vecteur s propres pour $A$ qui ne sont autres que les vecteurs colonnes de $P$.

Inversement, s'il existe une base de vecteur s propres de $A$, on note $P$ la matrice dont les vecteurs colonnes sont ces vecteurs propres et on vérifie par le même cal cul que $P^{-1}AP$ est alors diagonale, avec sur la diagonale les

vecteurs propres de $A$.

**Cas particulier** : On dit que $A$ est orthogonalement (resp. unitairement) diagonalisable si la matrice $P$ peut être choisie orthogonale, i.e. $P^{-1} = {}^tP$ (resp. unitaire, i.e. $P^{-1} = {}^t\bar{P}$). Ceci est bien sur équivalent à l'existence de vecteurs propres orthogonaux pour le produit scalaire $\langle X, Y\rangle = \sum_{i=1}^{N} x_i y_i$ (resp. $\langle X, Y\rangle = \sum_{i=1}^{N} x_i \overline{y_i}$ dans le cas complexe).

**Remarque 2**

"Orthogonal" et "unitaire" coïncident lorsque les vecteurs sont réels. Si $A$ est à coefficients réels, ses valeurs et vecteurs propres ne sont pas nécessairement réels. Cependant, les valeurs et vecteurs propres complexes sont conjugués deux à deux.

Rappelons sans démonstration quelques résultats que nous serons conduits à utiliser dan la suite.

Théorème 1 i) Si $A$ a toutes ses valeurs propres distinctes, elle est diagonalisable (si A est réelle, les valeurs propres sont alors réelles).

ii) Une matrice hermitienne (i.e. ${}^tA = \bar{A}$ ) a toutes ses valeurs propres réelles et est unitairement diagonalisable.

iii) Une matrice symétrique réelle a toutes ses valeurs propres réelles et est orthogonalement diagonalisable.

## 5.2 Origine des problèmes de valeurs propres

Sans être exhaustif (loin de là), nous allons citer ici quelques exemples simples conduisant naturellement à la recherche de valeurs et vecteurs propres d'une matrice.

### 5.2.1 Vibration d'une corde

Considérons une corde tendue entre ses deux extrémités supposées fixes. Il s'agit de déterminer les mouvements vibratoires stationnaires de la corde et par là-même ses fréquences fondamentales.

On démontre que, pour de petits déplacements $u(t,x)$ de la corde, l'évolution de $u(t)$ est régie par l'équation des ondes

$$\frac{\partial^2 u}{\partial t^2}(t,x) - \omega^2 \frac{\partial^2 u}{\partial x^2}(t,x) = 0 \quad (4)$$

ce, en l'absence de forces extérieurs. Si la corde est attachée aux extrémités, il faut ajouter les condition au bord qui sont

$$u(t,0) = 0, \quad u(t,l) = 0 \quad (5)$$

si les extrémités sont définies par $x = 0$ et $x = l$

La recherche de mouvements stationnaires consiste à déterminer les solution s du système (4),(5) sous la forme

$$u(t,x) = u(x)e^{i\mu t} \quad (6)$$

où $\mu$ est la fréquence à déterminer et $u(x)$ la forme stationnaire de la corde à déterminer. On constate que les parties réelles et imaginaires de $u(t,x)$ correspondent à des vibrations de période $\frac{2\pi}{\mu}$ auxquelles la corde peut être soumise en régime libre.

Plus généralement, pour résoudre ce problème, posons à priori

$$u(t,x) = u(x)v(t)$$

alors (4) équivaut à

$$v''(t)u(x) = \omega^2 u''(x)v(t)$$

ce qui, par séparation des variables, équivaut à l'existence d'une constante $\lambda$ telle que

$$-u''(x) = \lambda u(x), -v''(t) = \lambda\omega^2 v(t)$$

Il faut ajouter à ceci les conditions au bord, à savoir

$$u(0) = u(l) = 0$$

Considérons alors le problème

$$-u''(x) = \lambda u(x) \quad (7)$$

$$u(0) = u(l) = 0 \quad (8)$$

Il est clair que $u \equiv 0$ (position d'équilibre) est toujours solution. Ce qui

nous intéresse est l'existence éventuelle de solution non triviales $u(x)$. Ceci ne se produira en fait que pour des valeurs bien particulières de $\lambda$ qui sont précisément les valeurs propres de l'opérateur $u \to u''$ avec les conditions au bord $u(0) = u(1) = 0$.

Cette situation simple peut en fait s'analyser directement. On vérifier a les résultats suivants :

* Si $\lambda < 0$, la solution générale de (7) s'écrit

$$u(x) = Ae^{\sqrt{-\lambda}x} + Be^{\sqrt{-\lambda}x}$$

où $A$ et $B$ sont des constantes à déterminer par les conditions (8). Dans tous les cas, on obtient $A = B = 0$ donc seulement la solution triviale.

* Si $\lambda = 0$, la solution générale de (7) s'écrit

$$u(x) = Ax + B$$

ce qui avec (8) conduit au même résultat négatif.

* Si $\lambda > 0$, la solution générale de (7) est

$$u(x) = A\cos\sqrt{\lambda}x + B\sin\sqrt{\lambda}x$$

On constate qu'on peut satisfaire (8) si et seulement si

$$\lambda = \lambda_k = \frac{k^2\pi^2}{l^2}, k = 1, 2, \ldots$$

les solutions correspondantes étant données par

$$u_k(x) = \sin\left(\frac{k\pi x}{l}\right)$$

Revenant à l'équation en $v$, aux $\lambda_k, u_k$, on peut associer les solution

$$v_k(t) = C_k e^{ik\omega t/l}, C_k \in C$$

D'où les solutions stationnaires cherchées:

$$u_k(t,x) = C_k \sin\left(\frac{k\pi x}{l}\right) e^{ik\pi\omega t/l}$$

Dans bien des cas, la résolution analytique explicite de tels problèmes n'est pas possible et il est nécessaire de recourir à une résolution numérique.
Le premier travail consiste a remplacer le problème continu de type (7),(8) par un problème discrétisé approché où les in connues sont en nombre fini. On utilise par exemple une discrétisation par différences finies ou éléments finis.

Ainsi, on pourra remplacer (7),(8) par le problème discrétisé

$$\begin{cases} -\frac{U_{i+1}+U_{i-1}-2U_i}{h^2} = \lambda U_i, i = 1, \ldots, N \quad (9) \\ U_0 = 0, U_{\mathrm{N}+1} = 0 \end{cases}$$

correspondant à une subdivision régulière de pas $h = \frac{l}{N+1}$ de l'intervalle [0,1] et utilisant l'approximation par différences finies à 3 points de la dérivée

seconde

$$u''(x) \approx \frac{u(x+h) + u(x-h) - 2u(x)}{h^2}$$

On est donc ramené au problème de la recherche des valeur s propres de la matrice

$$A_h = \frac{1}{h^2} \begin{bmatrix} 2 & -1 & 0 & \cdots & \cdots & 0 \\ -1 & 2 & -1 & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & \ddots & -1 \\ 0 & \cdots & \cdots & 0 & -1 & 2 \end{bmatrix}$$

Dans ce cas simple, on peut d'ailleurs calculer explicitement les valeurs propres de $A_h$ soit

$$\lambda_k^h = \frac{4}{h^2} \sin^2 \frac{k\pi}{2(N+1)} 1 \leq k \leq N$$

associées aux vecteurs propres

$$U^k = \left(U_i^k\right) \quad U_i^k = \sin \frac{k\pi i}{N+1}$$

On constate que pour $k$ fixé

$$\lim_{h \downarrow 0} \lambda_k^h = \lambda_k = k^2 \pi^2.$$

### 5.2.2 Vibration d'une membrane

La version bidimensionnelle du problème précédent consiste à déterminer les vibrations propres d'une membrane fixée à un contour rigide $\Gamma$. L'équation aux dérivées partielles s'écrit

$$\frac{\partial^2 u}{\partial t^2}(t,x,y) = \omega^2 \left[ \frac{\partial^2 u}{\partial x^2}(t,x,y) + \frac{\partial^2 u}{\partial y^2}(t,x,y) \right], (x,y) \in \Omega$$

où $\Omega$ est l'intérieur du contour $\Gamma$ et

$$u(t,x,y) = 0 \quad sur \quad \Gamma$$

Le problème de valeurs propres associé s'écrit

$$\begin{cases} -\left[ \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right] = \lambda u \quad sur \Omega \\ u = 0 \quad sur \Gamma \end{cases} \qquad (10)$$

La discrétisation de (10) conduit à la recherche des valeurs propres de la matrice de discrétisation de l'opérateur Laplacien $\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$.

### 5.2.3 Problèmes de valeurs propres généralisés

Le plus souvent la discrétisation des opérateurs aux dérivées partielles intervenant dans les problèmes de la physique conduit à des problèmes de valeurs propres généralisées du type

$$Au = \lambda Bu \quad (11)$$

où $A$ et $B$ sont des matrices. Si $B$ est diagonale, on est encore dans la situation précédente. Cependant, le plus souvent $B$ est une matrice simple mais non diagonale ; elle sera en général symétrique définie positive et creuse. En théorie, on peut se ramener au cas précédent en remplaçant (11) par le système équivalent

$$B^{-1}Au = \lambda u$$

Mais ceci détruit en général les propriétés structurelles de $A$ et $B$. Par exemple $B^{-1}A$ n'est pas en général symétrique si $A$ et $B$ le sont.

On préférera commencer par une factorisation de Cholesky de $B$, soit $B = C^tC$ où $C$ est triangulaire inférieure. Alors, on écrit (11) sou la forme

$$C^{-1}Au = \lambda {}^tCu$$

ce qui équivaut à

$$\begin{cases} C^{-1}A\left({}^tC\right)^{-1}X = \lambda X \\ \qquad {}^tCu = X \end{cases}$$

On a donc à résoudre un problème de valeur s propres à matrice symétrique $C^{-1}A\left({}^tC\right)^{-1}$ puis à résoudre un système triangulaire pour obtenir les vecteurs propres $u$ à partir de $X$.

### 5.2.4 Système mécanique

Le calcul des fréquences fondamentales de "petits" mouvements, au voisinage d'une position d'équilibre, d'un système mécanique ayant un nombre fini $N$ de degrés de liberté conduit à une équation différentielle du type

$$Mu''(t) + Su'(t) + Ru(t) = 0$$

où $u(t)$ est un vecteur dont les composantes sont les $N$ degrés de liberté du système et $M, S, R$ des matrices réelles d'ordre $N$. Ici $M$ est la matrice de 'l'énergie cinétique" ou matrice de masse, $R$ la matrice de "rappel" ou matrice de rigidité et $S$ la matrice d'amortissement.

Si on cherche des solutions de la forme $u(t) = e^{\mu t}u$ où $u$ et $\mu$ tels que

$$\left(\mu^2 M + \mu S + R\right) u = 0$$

ceci est un nouveau problème généralisé de valeur s propres. A chaque solution $\mu$ correspondra une période fondamentale $T = \frac{2\pi}{3m\mu}$. La partie réelle de $\mu$ correspond au terme d'amortissement.

On se ramène à la situation précédente en considérant les matrices d'ordre $2N$.

$$A = \begin{bmatrix} I & 0 \\ 0 & M \end{bmatrix} \quad B = \begin{bmatrix} 0 & -I \\ R & S \end{bmatrix}$$

Si$(\mu, \tilde{u})$ est solution de $\mu A\tilde{u} + B\tilde{u} = 0$ en posant $\tilde{u} = (u, v)$ on a :

$$\begin{cases} \mu u - v = 0 \\ \mu M v + Ru + Sv = 0 \end{cases}$$

soit $\mu^2 Mu + \mu Su + Ru = 0$.

### 5.2.5 Autres exemples

Citons maintenant sans détailler d'autres problèmes d'origines très diverses qui conduisent à vouloir calculer des valeurs propres:

- En analyse numérique, on a vu qu'on avait souvent besoin de déterminer le rayon spectral d'une matrice ( convergence des méthodes itératives, calcul du conditionnement).

- L'étude du comportement des suites récurrentes linéaires, qui interviennent dans de nombreux domaines à partir du moment où on effectue la modélisation d'un phénomène discret, nécessite la détermination de la plus grande valeur propre d'une matrice (exemples: matrice de transition en biologie, chaines de Markov).

- Dans le même ordre d'idées, l'étude de la stabilité des systèmes différentiels conduit à rechercher les valeurs propres de la matrice du système ou de son linéarisé.

- Les techniques statistiques d'analyse de données, l'analyse en composante

principale par exemple, conduisent à rechercher les plus grandes valeurs propres de la matrice des données. Ces techniques sont employées en économie, géographie, sciences humaines,...

On peut trouver encore d'autres exemples. L'un des faits frappants dans l'inventaire ci-dessus est que, suivant le type de questions, on ne cherchera qu'une valeur propre, ou quelques valeurs propres, ou toutes les valeurs propres. De même, on a parfois besoin des vecteurs propres correspondants, quelquefois non. Le choix de la méthode employée, parmi celles qu'on va décrire dans cette section, sera donc très dépendant de ce qu'on veut.

## 5.3 Méthode de la puissance itérée et de la puissance inverse

Commençons par un algorithme simple très souvent utilisé quand il s'agit seulement de trouver quelques valeurs et vecteurs propres d'une matrice $A$.

### 5.3.1 Méthode de la puissance itérée

Considérons d'abord la suite définie par

$$X_{n+1} = AX_n$$

Supposons que $A$ soit diagonalisable et soit $e_1, e_2, \ldots e_{\mathrm{N}}$ une base de vecteurs propres.

Notons $\lambda_1, \ldots, \lambda_{\mathrm{N}}$ les valeurs propres associées dans le même ordre. Ecrivons enfin

$$X_0 = \alpha_1 e_1 + \alpha_2 e_2 + \ldots + \alpha_{\mathrm{N}} e_{\mathrm{N}}$$

la décomposition de $X_0$ suivant la base $(e_i)$. Alors

$$X_n = \alpha_1 \lambda_1^n e_1 + \alpha_2 \lambda_2^n e_2 + \ldots + \alpha_{\mathrm{N}} \lambda_{\mathrm{N}}^n e_{\mathrm{N}}$$

Supposons maintenant qu'on ait rangé les valeurs propres $\lambda_i$ de telles façon que

$$|\lambda_{\mathrm{N}}| > |\lambda_{\mathrm{N}-1}| \geq |\lambda_{\mathrm{N}-2}| \ldots \geq |\lambda_1|$$

alors

$$X_n = \lambda_{\mathrm{N}}^n \left[ \alpha_{\mathrm{N}} e_{\mathrm{N}} + \left( \frac{\lambda_{\mathrm{N}-1}}{\lambda_{\mathrm{N}}} \right)^n e_{\mathrm{N}-1} + \ldots + \left( \frac{\lambda_1}{\lambda_{\mathrm{N}}} \right)^n e_1 \right] \quad (12)$$

Puisque, pour $i < N, \lim_{n\to\infty} \left( \frac{\lambda_i}{\lambda_{\mathrm{N}}} \right)^n = 0$, on voit que pour $n$ grand

$$X_n \sim \lambda_{\mathrm{N}}^n \alpha_{\mathrm{N}} e_{\mathrm{N}}$$

Donc

- pour $n$ grand, $X_n$ donne la direction du vecteur propre $e_N$

- si la $j$ -ième composante de $e_{\mathrm{N}}$ est non nulle, on a

$$\lambda_{\mathrm{N}} \simeq \lim_{n\to\infty} \frac{(X_{n+1})_j}{(X_n)_j}$$

Nous venons de décrire à peu de chose près la méthode de la puissance itérée permettant de calculer la plus grande valeur propre de $A$ et le vecteur propre associé. La seule modification à apporter vient du fait que comme $\|X_n\|$ peut tendre vers l'infini, on ne peut numériquement l'utiliser tel quel. Il faut donc procéder à une normalisation à chaque étape.

**Algorithme de la puissance itérée :**

$$\begin{cases} X_0 \quad choisi \\ Y_{n+1} = AX_n \\ X_{n+1} = Y_{n+1} / \|Y_{n+1}\|_2 \\ \sigma_{n+1} = \bar{X}_n Y_{n+1} \end{cases}$$

Proposition 2 On suppose $A$ diagonalisable, ses valeurs propres vérifiant

$$|\lambda_1| \leq |\lambda_2| \leq \ldots \leq |\lambda_{N-1}| < |\lambda_N|$$

On note $X_0 = \alpha_1 e_1 + \ldots + \alpha_N e_N$ où $e_1, \ldots, e_N$ sont des vecteurs propres respectivement associés à $\lambda_1, \ldots, \lambda_N$ et on suppose $\alpha_N \neq 0$. Alors

$$\begin{cases} \lim_{n\to\infty} \sigma_{n+1} = \lambda_N \\ \lim_{n\to\infty} X_{n+1} - \beta^n \varepsilon_N = 0 \end{cases} \tag{13}$$

où $\varepsilon_N$ est un vecteur colinéaire à $e_N$ et $\beta = \frac{\lambda_N}{|\lambda_N|}$. De plus la convergence est au moins linéaire de rapport $\left|\frac{\lambda_{N-1}}{\lambda_N}\right|$

Démonstration : Posons $Z_n = A^n X_0$. On vérifie par récurrence que

$X_n = \frac{Z_n}{\|Z_n\|_2}$. D'après (12), on a :

$$Z_n = \lambda_{\mathrm{N}}^n \left( \alpha_{\mathrm{N}} e_{\mathrm{N}} + \sum_{i=1}^{N-1} \left( \frac{\lambda_i}{\lambda_{\mathrm{N}}} \right)^n \alpha_i e_i \right)$$

soit

$$Z_n = \lambda_{\mathrm{N}}^n \left( \alpha_{\mathrm{N}} e_{\mathrm{N}} + \left( \frac{\lambda_{\mathrm{N}-1}}{\lambda_{\mathrm{v}}} \right)^n \theta_n \right)$$

où $\theta_n$ est un vecteur borné quand $n \to \infty$. On en déduit

$$\|Z_n\|_2 = |\lambda_{\mathrm{N}}|^n \left( |\alpha_{\mathrm{N}}| \, \|e_{\mathrm{N}}\|_2 + \left| \frac{\lambda_{\mathrm{N}-1}}{\lambda_{\mathrm{N}}} \right|^n \rho_n \right)$$

où $\rho_n$ est une suite bornée de réels. Ainsi

$$X_n = \left( \frac{\lambda_{\mathrm{N}}}{|\lambda_{\mathrm{N}}|} \right)^n \left[ \frac{\alpha_{\mathrm{N}} e_{\mathrm{N}}}{|\alpha_{\mathrm{N}}| \, \|e_{\mathrm{N}}\|_2} + \left| \frac{\lambda_{\mathrm{N}-1}}{\lambda_{\mathrm{N}}} \right|^n \theta_n \right]$$

où $\theta_n$ est une suite bornée de vecteurs. Ceci montre le deuxième point de (13) avec

$$\sigma_{\mathrm{N}+1} = \left\langle \beta^n \left( \varepsilon^N + \left( \frac{\lambda_{\mathrm{N}-1}}{\lambda_{\mathrm{N}}} \right)^n \theta_n \right), \beta^n \left( \lambda_{\mathrm{N}} \varepsilon_{\mathrm{N}} + \left( \frac{\lambda_{N-1}}{\lambda_{\mathrm{v}}} \right)^n A \theta_n \right) \right\rangle$$

Puisque $\langle \varepsilon_{\mathrm{N}}, \varepsilon_{\mathrm{N}} \rangle = 1$, on en déduit

$$\sigma_{\mathrm{N}+1} = \lambda_{\mathrm{N}} + \left( \frac{\lambda_{\mathrm{N}-1}}{\lambda_{\mathrm{N}}} \right)^n \rho_n$$

où $\rho_n$ est une suite bornée de réels. Ceci montre le deuxième point de (13). La vitesse de convergence résulte des estimations ci-dessus.

**Remarque 3** - La convergence est montrée sous l'hypothèse que la composante du vecteur initial selon le vecteur propre $e_{\mathrm{N}}$ soit non nulle. C'est

évidemment invérifiable en pratique. Cependant, on constate que les erreurs d'arrondis sont en général suffisantes pour créer une composante non nulle selon $e_{\mathrm{N}}$ qui est en suite amplifiée. Ceci fait que la méthode converge pratiquement toujours.

- Si la matrice $A$ est réelle et $X_0$ réel, les vecteurs $X_n$ et $\sigma_n$ restent réels. Ceci est compatible avec l'hypothèse faite : en effet, comme $\lambda_{\mathrm{N}}$ est de module supérieur à celui de toutes les autres valeurs propres, elle est nécessairement réelle.

- On peut se demander ce qui se passe dans cet algorithme lorsque

$$|\lambda_N| = |\lambda_{\mathrm{N}-1}| > |\lambda_{\mathrm{N}-2}| \geq \ldots \geq |\lambda_1|$$

On a alors

$$Z_n = \lambda_{\mathrm{N}}^n \left( \alpha_{\mathrm{N}} e_{\mathrm{N}} + \left( \frac{\lambda_{\mathrm{N}-1}}{\lambda_{\mathrm{N}}} \right)^n \alpha_{N-1} e_{N-1} + O\left[ \left| \frac{\lambda_{\mathrm{N}-2}}{\lambda_{\mathrm{N}}} \right|^n \right] \right)$$

L'analyse de ce cas est renvoyée en exercice.

### 5.3.2 Méthode de la puissance inverse

Nous pouvons appliquer la méthode précédente à la matrice $A^{-1}$; comme les valeurs propres de $A^{-1}$ sont les inverses de celles de $A$, on a ainsi un procédé pour obtenir la valeur propre de $A$ de plus petit module et le vecteur propre

correspondant sous l'hypothèse que $A$ est diagonalisable et

$$|\lambda_1| < |\lambda_2| \leq \ldots \leq |\lambda_N|$$

Plutôt que calculer explicitement $A^{-1}$, on préfère effectuer une factorisation de la matrice $A$ (par exemple de type $LU$ ou de type Choleski si elle est symétrique). Les itérés successifs s'obtiennent alors par la résolution du système $AY_{n+1} = X_n$. Plus généralement, ce procédé nous permet d'aborder le calcul de n'importe quelle valeur propre $\lambda$ de $A$ dont on connaît une valeur suffisamment approchée $\lambda$. Il suffit d'appliquer la méthode de la puissance inverse à la matrice translatée $A - \widetilde{\lambda} I$.

**Algorithme de la puissance inverse avec translation**

$$\begin{cases} X_0 \quad donné \\ AY_{n+1} - \tilde{\lambda} Y_{n+1} = X_n \\ X_{n+1} = Y_{n+1} / \|Y_{n+1}\|_2 \\ \sigma_{n+1} = \tilde{\lambda} + 1/\overline{X_n} Y_{n+1} \end{cases}$$

Si $A$ est diagonalisable et si $\left|\tilde{\lambda} - \lambda_j\right| < \left|\tilde{\lambda} - \lambda_i\right| \quad \forall i \neq j$ d'après la proposition 7.2 , $\frac{1}{\sigma_{n+1} - \widetilde{\lambda}}$ tend ver la valeur propre de plus grand module de $(A - \tilde{\lambda} I)^{-1}$ qui n'est autre que $\frac{1}{\lambda_i - \lambda}$ et donc $\sigma_n$ tend vers $\lambda_i$. On vérifie que $X_n$ tend vers un vecteur propre correspondant.

**Remarque 4** - La méthode ci-dessus permet en fait de trouver toutes les valeurs propres simples (et même multiples, cf. l'exercice 1 ) d'une matrice

diagonalisable.

- La convergence dans l'algorithme précédent est linéaire de rapport

$$\max_{j\neq i} \frac{\left|\tilde{\lambda}-\lambda_i\right|}{\left|\tilde{\lambda}-\lambda_j\right|}$$

On peut accélérer la vitesse de convergence en faisant varier $\tilde{\lambda}$. Prenant par exemple $\overline{\lambda_n} = \sigma_n$ à chaque étape, on accélère très sérieusement la convergence.

- Cette technique peut être aisément étendue au problème de valeurs propres généralisé :

$$AX = \lambda BX$$

**Algorithme de la puissance inverse généralisé**

$$\begin{cases} X_0 \text{ choisi } \tilde{\lambda} \text{ valeur approchée de } \lambda \\ AY_{n+1} - \tilde{\lambda}BY_{n+1} = BX_n \\ X_{n+1} = Y_{n+1}/\left\|Y_{n+1}\right\|_2 \\ \sigma_{n+1} = \tilde{\lambda} + 1/\overline{X_n}Y_{n+1} \end{cases}$$

**Remarque 5** Comme il a déjà été signalé plus haut, il est préférable de travailler avec la matrice $A-\lambda B$ plutôt que $B^{-1}A$ pour préserver les structures particulières de $A$ et $B$.

## 5.4 Méthode de Jacobi

Cette méthode s'applique aux matrices symétriques réelles. Elle permet d'en trouver simultanément toutes les valeurs propres et tous les vecteurs propres. Elle s'applique bien aux matrices pleines.

Principe : On construit des matrices orthogonales $\Omega_k$ telles que les matrices $A_k$ définies par

$$\begin{cases} A_0 = A \\ A_{k+1} = {}^t\Omega_k A_k \Omega_k \end{cases}$$

convergent vers une matrice diagonale. On "lit" alors les valeurs propres de $A$ sur la diagonale de $A_k$ pour $k$ assez grand. Les vecteurs propres de $A$ sont les vecteurs colonnes de

$$O_k = \Omega_1 \Omega_2 \dots \Omega_k$$

Les matrices $\Omega_k$ seront des matrices de rotation, soit du type

$$\Omega = \begin{bmatrix}
1 & & 0 & \vdots & & & & \vdots & & & & \\
 & \ddots & & \vdots & & 0 & & \vdots & & 0 & & \\
0 & & 1 & \vdots & & & & \vdots & & & & \\
\cdots & \cdots & \cdots & \cos\theta & \cdots & \cdots & \cdots & \sin\theta & \cdots & \cdots & \cdots & p\text{-ième ligne} \\
 & & & \vdots & 1 & & 0 & \vdots & & & & \\
 & 0 & & \vdots & & \ddots & & \vdots & & 0 & & \\
 & & & \vdots & 0 & & 1 & \vdots & & & & \\
\cdots & \cdots & \cdots & -\sin\theta & \cdots & \cdots & \cdots & \cos\theta & \cdots & \cdots & \cdots & q\text{-ième ligne} \\
 & & & \vdots & & & & \vdots & 1 & & 0 & \\
 & 0 & & \vdots & & 0 & & \vdots & & \ddots & & \\
 & & & \vdots & & & & \vdots & 0 & & 1 &
\end{bmatrix}$$

$p$-ième colonne $\qquad$ $q$-ième colonne

Si A est symétrique, examinons B= ${}^t\Omega A\Omega$. On a

$b_{ij} = \sum_{\alpha=1}^{N} \omega_{\alpha i} \sum_{k=1}^{N} a_{\alpha k}\omega_{kj} = \sum_{k=1}^{N} \sum_{\alpha=1}^{N} \omega_{\alpha i}\omega_{kj}a_{\alpha k}$

*Ainsi*

$$\begin{cases}
si \quad i \neq p,q \quad j \neq p,q \quad b_{ij} = a_{ij} \\
si \quad i = p \quad j \neq p,q \quad b_{pj} = \cos\theta a_{pj} - \sin\theta a_{pj} \\
si \quad i = q \quad j \neq p \quad b_{qj} = \sin\theta a_{pj} + \cos\theta a_{qj} \\
pour \quad i = j = p \quad b_{pp} = \cos^2\theta a_{pp} + \sin^2\theta a_{qq} - \sin 2\theta a_{pq} \\
pour \quad i = j = q \quad b_{qq} = \sin^2\theta a_{pp} + \cos^2\theta a_{qq} + \sin 2\theta a_{pq} \\
pour \quad i = p, j = q \quad b_{pq} = \cos 2\theta a_{pq} + \frac{\sin 2\theta}{2}(a_{pp} - a_{qq}) \\
\text{le reste est obtenu par symétrie.}
\end{cases}$$

Au vu de l'expression de $b_{pq}$, il apparaît que si $a_{pq} \neq 0$, on peut choisir $\theta$ pour que $b_{pq} = 0$, il suffit de prendre $\theta$ tel que

$$\cot 2\theta = \frac{a_{qq} - a_{pp}}{2a_{pq}}, -\frac{\pi}{4} \leq \theta \leq \frac{\pi}{4} \quad (14)$$

**Lemme 1** Si $\theta$ est choisi selon (14), on $a$

$$\sum_{i,j} b_{ij}^2 = \sum_{i,j} a_{ij}^2 \quad (15)$$

$$\sum_i b_{ii}^2 = \sum_i a_{ii}^2 + 2a_{pq}^2 \quad (16)$$

**Remarque 6** Au vu de ce lemme, on voit que si $a_{pq} \neq 0$, la diagonale de $B$ est globalement plus dominante que celle de $A$. Notons que par différence

$$\sum_{i \neq j} b_{ij}^2 = \sum_{i \neq j} a_{ij}^2 - 2a_{pq}^2$$

La répétition de ce type d'opérations devrait "creuser" de plus en plus la partie hors diagonale des matrices successives.

**Démonstration du lemme 1:** Pour (15) on utilise

$$trace\left({}^tBB\right) = \sum_{i,j} b_{ij}^2, \quad trace\left({}^tAA\right) = \sum_{i,j} a_{ij}^2$$

Mais

$$B = {}^t\Omega A\Omega \Longrightarrow {}^tBB = {}^t\Omega {}^tAA\Omega$$

Donc

$$trace\left({}^tBB\right) = trace\left({}^t\Omega {}^tAA\Omega\right) = trace\left({}^tAA\right)$$

Pour (16), on remarque que

$$\begin{bmatrix} b_{pp} & b_{pq} \\ b_{qp} & b_{qq} \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} a_{pp} & a_{pq} \\ a_{qp} & a_{qq} \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$$

par le même raisonnement que ci-dessus, on a

$$b_{pp}^2 + b_{qq}^2 + 2b_{pq}^2 = a_{pp}^2 + a_{qq}^2 + 2a_{pq^*}^2$$

Si on choisit $\theta$ pour que $b_{pq} = 0$, comme les autres éléments diagonaux de $B$ sont identiques à ceux de $A$, on en déduit (16).

**Remarque 7** Dans la pratique, on évite de faire appel aux fonctions transcendantes pour le calcul de $B$ en fonction de $A$. On utilise un calcul reposant sur les formules trigonométriques classiques: si $t = \tan\theta$

$$\cot 2\theta = \frac{1-t^2}{2t}, \cos^2\theta = \frac{1}{1+t^2}, \sin^2\theta = \frac{t^2}{1+t^2}$$

Ainsi, puisqu'on peut supposer $|\theta| \leq \frac{\pi}{4}$, posant

$$\lambda = \frac{a_{qq} - a_{pq}}{2a_{pq}} \quad (17)$$

la valeur $t$ est calculée comme la racine de plus petit module de la valeur $t$ est calculée comme la racine de plus petit module de

$$t^2 + 2\lambda t - 1 = 0 \quad (18)$$

Puis on calcule

$$c = \frac{1}{\sqrt{1+t^2}}, s = tc \quad (19)$$

$$\begin{aligned} b_{pi} &= ca_{pi} - sa_{qi} \quad i \neq p, q \\ b_{qi} &= ca_{qi} - sa_{pi} \quad i \neq p, q \\ b_{pp} &= a_{pp} - ta_{pq} \\ b_{qq} &= a_{qq} + ta_{pq} \end{aligned} \tag{20}$$

**Algorithme de Jacobi classique**

- On pose $A_0 = A$
- On détermine l'élément $a_{pq}^k$ *de* $\mathrm{A}_k$ réalisant le plus grand module parmi les éléments hors diagonaux.
- On calcule $A_{k+1}$ selon les formules $(17) - (20)$ où $A_k$ joue le rôle de $A$ et $A_{k+1}$ celui de $B$.

**Proposition 3** Les éléments hors diagonaux de $A_k$ tendent vers 0 lorsque $k$ tend vers l'infini.

**Démonstration :** Notons $m_k = \max_{i \neq j} \left| a_{ij}^k \right|$. Le couple $(p, q)$ est choisi pour que $\left| a_{pq}^k \right| = m_k$. D'après (15) et (16), on a

$$\sum_{i \neq j} \left( a_{ij}^{k+1} \right)^2 = \sum_{i \neq j} \left( a_{ij}^k \right)^2 - 2m_k^2$$

Puisque

$$\sum_{i \neq j} \left( a_{ij}^k \right)^2 \leq \left( N^2 - N \right) m_k^2$$

on en déduit

$$\sum_{i\neq j}\left(a_{ij}^{k+1}\right)^2 \leq \left(1-\frac{2}{N(N-1)}\right)\sum_{i\neq j}\left(a_{ij}^{k}\right)^2$$

Donc la suite $\sum_{i\neq j}\left(a_{ij}^{k+1}\right)^2$ converge de façon géométrique vers 0.

**Proposition 4** La diagonale $D_k$ de $A_k$ converge vers une matrice $D = diag\left(\lambda_{\sigma(i)}\right)$ où $\sigma$ est une permutation sur $\{1, 2, \ldots, N\}$.

**Démonstration :** On a, puisque $A$ et $A_k$ sont semblables et par définition de $D$

$$det(A-\lambda I) = det\left(A_k-\lambda I\right) = det(D-\lambda I)$$

et d'après la proposition 3

$$\lim_{k\to\infty} det\left(D_k-\lambda I\right) = det(D-\lambda I)$$

En particulier les racines du premier polynôme convergent vers celles du second. Reste à vérifier qu'il n'y a pas échange entre deux racines d'une itération à l'autre. Or, on vérifie directement à l'aide des formules (20) que $D_k$ converge vers $D$. En effet

$a_{ii}^{k+1} - a_{ii}^{k} = 0 \quad si \quad i = p, q$

$\left|a_{pp}^{k+1} - a_{pp}^{k}\right| \leq t\left|a_{pq}^{k}\right| \leq \left|a_{pq}^{k}\right| \to_{k\to\infty} 0$

$\left|a_{qq}^{k+1} - a_{qq}^{k}\right| \leq t\left|a_{pq}^{k}\right| \leq \left|a_{pq}^{k}\right| \to_{k\to\infty} 0$

**Proposition 5** On suppose que toutes les valeurs propres de $A$ sont

distinctes. Alors la matrice

$$O_k = \Omega_1 \Omega_2 \dots \Omega_k$$

converge vers une matrice orthogonale dont les vecteurs colonnes sont vecteurs propres de $A$.

La démonstration de ce dernier point est laissé au lecteur.

**Remarque 8** La détermination de $a_{pq}^k$ comme indiquée dans la méthode classique est relativement coûteuse si la taille de la matrice est importante. On lui préfère souvent les choix suivants :

* Méthode de Jacobi cyclique : on annule successivement tous les éléments hors diagonaux par un balayage cyclique par exemple ligne par ligne de la gauche jusqu'à la diagonale. Bien sûr, si un élément est nul, on passe au suivant.

* Méthode de Jacobi avec seuil : on procède au même balayage, mais on omet d'annuler les éléments inférieurs à un certain seuil qu'on diminue à chaque balayage.

On démontre que ces procédés donnent lieu aux mêmes résultats de convergence que précédemment.

## 5.5 Méthode de Givens-Householder

Elle est particulièrement bien adaptée à la recherche des valeurs propres d'une matrice symétrique réelle qui sont situées dans un intervalle présélectionné ou de rang donné. Par contre, elle ne fournit pas les vecteurs propres correspondants.

### 5.5.1 Principe

Il y a deux étapes :

1) Étant donnée $A$ symétrique, on détermine une matrice orthogonale $\Omega$ tel que ${}^t\Omega A\Omega$ soit tridiagonale : ceci se fait en un nombre fini d'opérations à l' aide de matrices dites de Householder.

2) On est alors ramené au calcul des valeurs propres d'une matrice tridiagonale symétrique : pour cela, on utilise une méthode de bissection, dite de Givens.

### 5.5.2 Description de la méthode de Givens

Soit

$$B_{\mathrm{N}} = \begin{pmatrix} b_1 & c_1 & 0 & & 0 \\ c_1 & \ddots & \ddots & & \\ 0 & \ddots & \ddots & \ddots & 0 \\ & & \ddots & \ddots & c_{\mathrm{N}-1} \\ 0 & & 0 & c_{\mathrm{N}-1} & b_{\mathrm{s}} \end{pmatrix}$$

On note $P_{\mathrm{x}}(\lambda) = det\,(B_{\mathrm{N}} - \lambda I)$ le polynôme caractéristique de $B_{\mathrm{s}}$. Ces polynômes ont la propriété remarquable de satisfaire à une formule simple de récurrence à 3 termes et de constituer ce que l'on appelle une suite de Sturm. Cette propriété a pour conséquence une disposition particulière des zéros de $P_{\mathrm{N}}$ par rapport à ceux de $P_{\mathrm{N}-1}$.

**Proposition 6**

$$P_0(\lambda) \quad = \quad 1$$

$$\mathrm{P}_1(\lambda) = \quad b_1 - \lambda$$

$$\forall N \geq 2, P_N(\lambda) = (b_N - \lambda)\, P_{N-1}(\lambda) - c_{n-1}^2 P_{N-2}(\lambda) \quad (21)$$

Ceci se démontre aisément en développant det $(B_{\mathrm{N}} - \lambda I)$ suivant la dernière ligne.

**Proposition 7** Les racines de $P_N(\lambda)$ sont toutes réelles, distinctes et séparées

par celles de $P_{N-1}(\lambda)$. De plus, $\lim_{\lambda\to-\infty} P_N(\lambda) = +\infty, \forall N \geq 1$

Ainsi les racines de $P_1, P_2, \ldots,$ sont disposées suivant le shéma suivant :

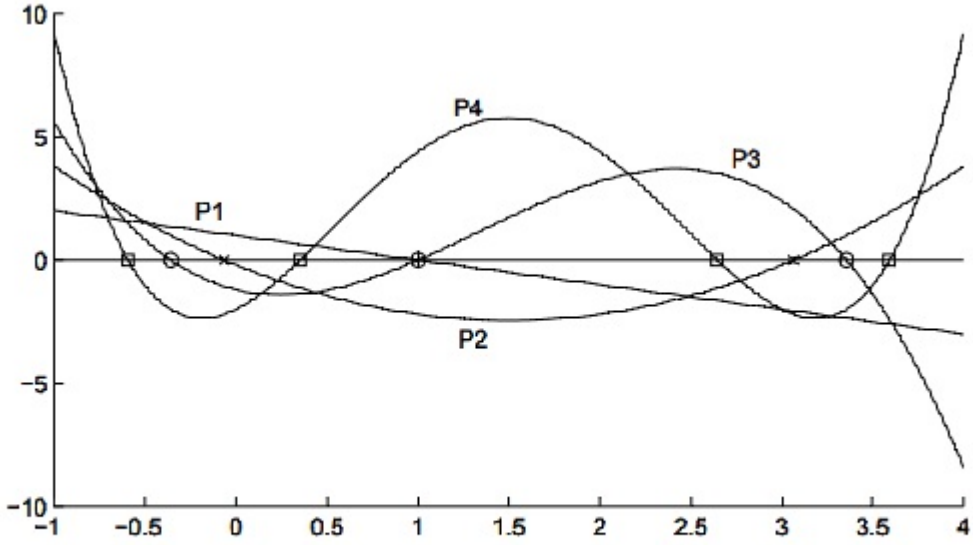



FIGURE 7.1 – Les racines des polynômes $P_1, P_2, P_3, P_4$ sont entrelacées

Ceci se démontre aisément par récurrence à l'aide de (21). Les détails sont laissés au lecteur. On en déduit la conséquence suivante sur le nombre de racines de $P_i(\lambda)$ inférieures à un nombre réel $\mu$.

Notation : On note $\mathcal{N}(i, \mu)$ le nombre de changements de signes dans la suite $\{+, P_1(\mu), P_2(\mu), \ldots, P_i(\mu)\}$ avec la convention qu'il n'y a pas de changement de signe lorsqu'un élément est nul.

**Proposition 8** Pour $i \geq 1$ et $\mu \in R, \mathcal{N}(i, \mu)$ est égal au nombre de racines de $P_i(\mu)$ qui sont strictement inférieures à $\mu$.

Ceci se démontre par récurrence. On peut se convaincre du bien-fondé de

ce résultat à l'aide du dessin ci-dessus. On en déduit:

**Algorithme de Givens :** Soit $\lambda_1 \leq \ldots \leq \lambda_{\mathrm{N}}$ les valeurs propres de $B_{\mathrm{N}}$. Supposons qu'on veuille calculer $\lambda_k$ pour un certain $k \in \{1, \ldots, N\}$.

(i) On détermine un intervalle $[a_0, b_0]$ dans lequel on est sûr de trouver $\lambda_k$.

(ii) On prend $c_0 := \frac{a_0+b_0}{2}$ et on calcule $\mathcal{N}(N, c_0)$.

– si $\mathcal{N}(N, c_0) \geq k$, alors $\lambda_k \in [a_0, c_0]$ et on pose $a_1 := a_0, b_1 := c_0$ – si $\mathcal{N}(N, c_0) < k$, alors $\lambda_k \in ]c_0, b_0]$ et on pose $a_1 := c_0, b_1 := b_0$

(iii) On recommence alors la même opération avec $c_1 = \frac{a_1+b_1}{2}$, puis avec $a_2, b_2, c_2$, et c.

On construit ainsi une suite de segments emboités $[a_i, b_i]$ dont l'intersection est précisément la valeur cherchée $\lambda_k$. Aux erreurs d'arrondi près, on peut donc obtenir $\lambda_k$ avec une précision arbitraire.

**Remarque 9** les évaluations successives des $P_j$ se font bien sûr à l'aide de la formule de récurrence (21).

Reste maintenant à décrire une technique de tridiagonalisation des matrices symétriques. Elle est en fait un cas particulier de la mise sous forme

Hessenberg d'une matrice soit

$$\begin{bmatrix} \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet \\ 0 & \bullet & \bullet & \bullet & \bullet \\ \vdots & \ddots & \bullet & \bullet & \bullet \\ 0 & \cdots & 0 & \bullet & \bullet \end{bmatrix}$$

où $a_{ij} = 0$ si $j < i - 1$. En effet, on remarque qu'une matrice de Hessenberg symétrique est tridiagonale.

Nous allons ici décrire en détail cette technique car elle sera doublement utilisée au paragraphe suivant (factorisation $QR$ ), d'une part par ce qu'on met souvent une matrice sous forme Hessenberg avant de calculer sa factorisation $QR$ pour diminuer le coût de calcul, d'autre part parce que l'outil est le même, à savoir l'utilisation de matrices de Householder.

**Définition 2** On appelle matrice de Householder une matrice $H_v$ de la forme

$$H_v = I - 2\frac{vv^*}{v^*v}$$

où $v \in C^N$ et $v^* = \bar{v}$

**Remarque 10** Noter que $v^*v = \bar{v}v = \sum_{i=1}^{N} |v_i|^2$ est le carré de la norme

euclidienne de $v$. Par contre

$$vv^* = [v_1 v_2 \dots v_{\mathrm{N}}] \begin{bmatrix} \bar{v}_1 \\ \bar{v}_2 \\ \vdots \\ \bar{v}_{\mathrm{N}} \end{bmatrix} = \begin{bmatrix} |v_1|^2 & v_1\bar{v}_2 & \dots & v_1\bar{v}_{\mathrm{N}} \\ v_2\bar{v}_1 & |v_2|^2 & \dots & v_2\bar{v}_{\mathrm{N}} \\ \vdots & & & \vdots \\ v_{\mathrm{N}}\bar{v}_1 & \dots & \dots & |v_{\mathrm{N}}|^2 \end{bmatrix}$$

De plus, si on introduit le vecteur unitaire $u = \frac{v}{\|v\|_2'}$ on a $H_v = H_u = I - 2uu^*$.

- On vérifie que ces matrices sont unitaires et hermitiennes. Ces matrices ont une interprétation géométrique simple. Supposons par exemple $u$ réel et $N = 3$. Alors $H_u$ est la matrice de la symétrie orthogonale par rapport au plan orthogonal à $u$.

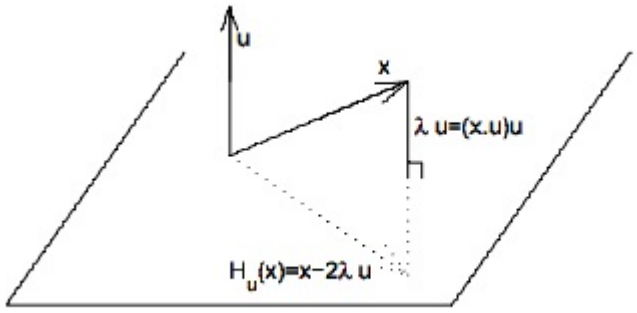



FiGURE 7.2 – $H_u(x)$ est le symétrique de $x$ par rapport au plan orthogonal à $u$.

En effet le transformé $X$ de $x$ dans la symétrie orthogonale par rapport

au plan orthogonal à $u$ est caractérisé par

$$X-x=-2\lambda u \quad o\grave{u} \quad \lambda={}^{t}uxetdoncX=x-2u\left({}^{t}ux\right)=x-2\left(u{}^{t}u\right)x=H_{u}x$$

- Diverses factorisations de matrices reposent sur l'utilisation de telles matrices de symétries et plus particulièrement de la propriété suivante.

**Théorème 2** Soit $a=(a_1,\ldots,a_N)\in R^N$ avec

$$\sum_{i=2}^{N}|a_i|\neq 0$$

Alors il existe une matrice de Householder $H_v$ telle que $H_v a$ ait ses $(N-1)$ dernières composantes nulles. Plus précisément, si $e_1,\ldots,e_n$ est la base canonique de $R^N$, on a

$$H_v a=-ke_1, \quad H_{\hat{v}}a=ke_1 \quad (22)$$

où

$$k=sgn\,(a_1)\,\|a\|_2, v=a+ke_1, \widehat{v}=a-ke_1 \quad (23)$$

**Remarque 11** L'intérêt du Théorème 2 dans les techniques de factorisation est immédiat. En effet, si $a$ est un vecteur-colonne d'une matrice $M$, on "fait apparaître des zéros" dans $M$ en multipliant $M$ par $H_v$ puisque la colonne $a$ est remplacée par la colonne $H_v a$ qui a des zéros partout sauf sur la première composante. Ceci sera utilisé de façon essentielle dans la factorisation $QR$.

Nous laissons au lecteur la vérification du Théorème 2. Nous allons plutôt interpréter géométriquement le Théorème 2. Supposons $a \in R^3$. La relation (22) exprime que $a$ n'est pas colinéaire à $e_1$. On peut alors déterminer aisément une symétrie orthogonale transformant $a$ en un vecteur colinéaire à $e_1$. On voit qu'il y a deux solutions: les symétries orthogonales par rapport aux plans bissecteurs de l'angle $(a, e_1)$. Il sont orthogonaux aux vecteurs $v = a + \|a\|_2 e_1$ et $\hat{v} = a - \|a\|_2 e_1$ et on a

$$H_v a = -\|a\|_2 e_1, H_{\hat{v}} a = \|a\|_2 e_1$$

ce qui donne bien les relations (22), (23) dans le cas réel.

**Calcul pratique :** (dans le cas réel)

$\left\{ \begin{array}{l} \cdot \text{ Calculer } \|a\|_2, v := a \pm \|a\|_2 e_1 \\ \cdot \text{ Calculer } \sigma = \frac{1}{2}(vv) = \|a\|_2 (\|a\|_2 \pm a_1) \\ \text{et choisir } + \text{ ou } - \text{ pour que } \sigma \text{ soit le plus grand possible.} \\ \text{Pour appliquer } H_v \text{ à un vecteur } b, \text{ calculer } {}^t vb \text{ et } H_v b = b - \frac{vb}{\sigma} v \end{array} \right.$

### 5.5.3 Mise sous forme Hessenberg d'une matrice

Principe : On détermine des matrices de Householder $H_1, H_2, \ldots, H_{\mathrm{N}-2}$ telles que $A_{\mathrm{s}-1} = H_{\mathrm{v}-2} \ldots H_1 A H_1 \ldots H_{\mathrm{N}-2}$ soit une matrice de Hessenberg. Puisque les matrices $H_i$ sont hermitiennes $\left(\bar{H} = H = H^{-1}\right)$, $A_{\mathrm{v}-1}$ est semblable à $A$ et a don c les mêmes valeurs propres.

Si, de plus, $A$ est symétrique réelle et que les $H_i$ sont réelles, $A_{s-1}$ est symétrique et donc tridiagonale. Comme cas particulier, nous allons donc obtenir la technique de tridiagonalisation annoncée au début du paragraphe qui constitue la première étape de la méthode de Givens-Householder.

Décrivons maintenant le calcul de $H_{k-1}$. Supposons que $A_{k-1} = H_{k-2} \dots H_1 A H_1 \dots H_{k-2}$

$$A_{k-1} = \begin{bmatrix} \bullet & \bullet & \dots & & \dots & \bullet \\ \bullet & \bullet & \dots & & \dots & \bullet \\ 0 & \bullet & \bullet & \dots & \dots & \bullet \\ \vdots & \ddots & \ddots & \bullet & & \vdots \\ & & 0 & \bullet & & \\ & & \vdots & \vdots & & \\ & & & & & \vdots \\ 0 & \dots & 0 & \bullet & \dots & \dots & \bullet \end{bmatrix} \begin{matrix} \\ \\ \\ \\ \\ \\ \\ k-\text{ième ligne} \end{matrix}$$

$$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad k-\text{ième colonne}$$

soit pour $k \geq 2$ de la forme :

Considérons le vecteur $a = \left(a_{k,k-1}^{k-1} a_{k+1,k-1}^{k-1}, \dots, a_{s,k-1}^{k-1}\right)$ constitué par la queue de la $(k-1)$ -ième colonne de $A_{k-1}$. Soit $H_v$ l'une des matrices de Householder de dimension $N-k+1$ associées au vecteur $a$ de $R^{N-k+1}$ par le théorème 7.2 et don c telle que $H_v a$ soit colinéaire à $e_k$. Posons alors

$$H_{k-1} = \begin{bmatrix} I_{k-1} & 0 \\ 0 & H_v \end{bmatrix}$$

où $I_{k-1}$ est la matrice unité de dimension $(k-1)$. Si on note

$$A_{k-1} = \begin{bmatrix} A_1^{k-1} & A_2^{k-1} \\ A_3^{k-1} & A_4^{k-1} \end{bmatrix}$$

une multiplication effectuée par blocs montre que :

$$H_{k-1}A_{k-1} = \begin{bmatrix} I_{k-1} & 0 \\ 0 & H_v \end{bmatrix} \begin{bmatrix} A_1^{k-1} & A_2^{k-1} \\ A_3^{k-1} & A_4^{k-1} \end{bmatrix} = \begin{bmatrix} A_1^{k-1} & A_2^{k-1} \\ H_v A_3^{k-1} & H_v A_4^{k-1} \end{bmatrix}$$

$$A_k = H_{k-1}A_{k-1}H_{k-1} = \begin{bmatrix} A_1^{k-1} & A_2^{k-1} \\ H_v A_3^{k-1} & H_v A_4^{k-1} \end{bmatrix} \begin{bmatrix} I_{k-1} & 0 \\ 0 & H_v \end{bmatrix} = \begin{bmatrix} A_1^{k-1} & A_2^{k-1}H_v \\ H_v A_3^{k-1} & H_v A_4^{k-1}H_v \end{bmatrix}$$

En particulier $H_v A_3^{k-1}$ est de la forme

$$H_v A_3^{k-1} = \begin{bmatrix} 0 & \cdots & 0 & \bullet \\ 0 & \cdots & \cdots & 0 \\ \vdots & & & \vdots \\ 0 & \cdots & \cdots & 0 \end{bmatrix}$$

et donc $A_k$ a la même structure que $A_{k-1}, k$ étant remplacé par $k+1$. Finalement, $A_{\mathrm{v}-1}$ est une matrice de Hessenberg.

**Remarque 12** L'application du théorème 2 nécessite que $a$ soit non colinéaire à $e_k$. Si ce n'est pas le cas, le "travail" est déjà fait et on passe immédiatement à l'étape suivante (on a alors $H_k = I$) symétrie.

## 5.6 La méthode $QR$

C'est probablement la méthode la plus couramment utilisée pour trouver toutes les valeurs propres d'une matrice quel con que, notamment non symétrique. L'en semble des vecteurs propres peut être également obtenu.

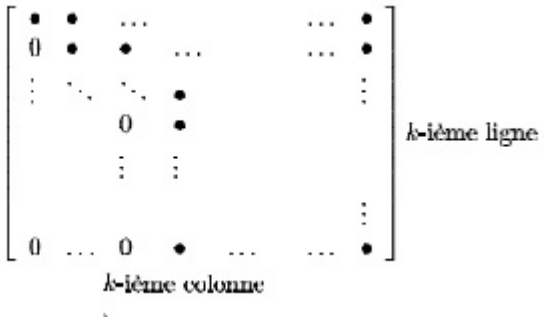
Commençons par un résultat de factorisation:

**Théorème 3** Soit $A$ une matrice à coefficients complexes (respectivement réels). Alors, il existe $Q$ unitaire (respectivement orthogonale) et $R$ triangulaire supérieure telle que $A = QR$

**Démonstration** et algorithme de construction de $Q$ et $R$.

soit triangulaire supérieure. Posant alors $Q = (H_{\mathrm{N}-1}H_{\mathrm{N}-2}\dots H_1)^{-1} = H_1H_2\dots H_{\mathrm{N}-1}$ on obtient la factorisation voulue.

La construction des matrice successives $H_k$ repose sur le théorème 2 . Supposons $A_k = H_{k-1}H_{k-2}\dots H_1A$ de la forme suivante pour $k \geq 1$ :

$A_k =$ 


Notons $a = \left(a^k_{k,k}, a^k_{k+1,k}, \dots, a^k_{\mathrm{s},k}\right)$ la queue de la $k$ -ième colonne de $A_k$

Soit $H_v$ l'une des matrices de Householder de dimension $N-k+1$ associées an vecteur $a$ selon le théorème 2 et donc telle que $H_va$ soit colinéaire à $e_k$.

Posons alors:

$$H_k = \begin{bmatrix} I_{k-1} & 0 \\ 0 & H_v \end{bmatrix}$$

où $I_{k-1}$ est la matrice unité de dimension $k-1$. Si on note

$$A_k = \begin{bmatrix} A_1^k & A_2^k \\ 0 & A_3^k \end{bmatrix}$$

la décomposition par blocs correspondants de $A_k$, une multiplication par blocs donne:

$$A_{k+1} = H_v A_k = \begin{bmatrix} A_1^k & A_2^k \\ 0 & H_v A_4^k \end{bmatrix}$$

Compte tenu de $H_v A = 0$, la m at ri ce $H_v A_4^k$ est de la forme:

$$\begin{bmatrix} \bullet & \bullet & \cdots & \cdots \\ 0 & \bullet & \cdots & \cdots \\ \vdots & \vdots & & \\ 0 & \bullet & \cdots & \cdots \end{bmatrix}$$

Donc $A_{k+1}$ a la même forme que $A_k$ ave c $k$ remplacé par $k+1$. Itérant jusqu'à $k = N-1$, on obtient une matrice triangulaire.

**Remarque 3** Si $a$ est colinéaire à $e_k$, on passe bien sur directement à l'étape suivante en posant $A_{k+1} := A_k$

Une nouvelle méthode directe de résolution d'un système linéaire :

Soit à résoudre $AX = b$. Appliquant la décomposition ci-dessus, ce système équivaut à :

$$H_{\mathrm{N}-1} \ldots H_1 AX = H_{\mathrm{N}-1 \ldots} H_1 b$$

soit

$$RX = H_{s-1} \dots H_1 b$$

où $R$ est triangulaire supérieure. On peut alors le résoudre par remontée. On obtient ainsi une nouvelle méthode de résolution directe pour les systèmes linéaires.

Pour $N$ grand, le nombre d'opération s élément aire s de la factorisation $QR$ est de l'ordre de $\frac{4}{3}N^3$ soit deux fois plus important que pour une factorisation $LU$ de Gauss. En contrepartie, il faut noter qu'aucune stratégie du pivot n'est nécessaire et surtout, comme les facteurs $H_k$ sont unitaires, le conditionnement de $R$ est égal à celui de A. (si $H$ unitaire, cond $_2(HA) = cond_2(A)$ voir le chapitre précédent). Dans certaines situations délicates, ceci peut constituer un avantage déterminant de cette méthode de ré solution sur celle de Gauss.

### 5.6.1 Algorithme $QR$ de recherche de valeurs propres

Étant donnée $A$, on forme les suites de matrices $A_i, Q_i, R_i$ de la façon suivante :

$A_0 := A$

Pour $i = 0, 1, \ldots$

(1) On effectue la décomposition $A_i = Q_i R_i$ avec $Q_i$ unitaire $R_i$ triangulaire supérieure

(2)$A_{i+1} := R_i Q_i$

On remarque que $R_i = Q_i^{-1} A_i$ et don c $A_{i+1} = Q_i^{-1} A_i Q_i$, ainsi $A_{i+1}$ est semblable à $A_i$ et, par récurrence, tous les $A_i$ sont semblables à $A$ (et ont donc les mêmes valeurs propres).

Ainsi, si $A_i$ tend vers une matrice triangulaire lorsque $i$ devient grand, on pourra lire sur la diagonale les valeurs propres de $A$. C'est ce qui se produit au moins dan s le cas fréquent suivant.

**Théorème 4** On suppose les valeurs propres de $A$ toutes de modules différents, soit

$$|\lambda_1| > |\lambda_2| > \ldots > |\lambda_N|$$

Il existe donc une matrice inversible $P$ telle que $P^{-1}AP$ soit la matrice diagonale $diag\,(\lambda_1, \lambda_2, \ldots, \lambda_N)$ ; on fait de plus l'hypothèse technique que $P$ admet une factorisation $LU$

Alors la suite de matrices $A_k$ définie par (24) vérifie

$$\forall \quad 1 \leq j < i \leq N \quad \lim_{k\to\infty} (A_k)_{ij} = 0$$

$$\forall \quad 1 \leq i \leq N \quad \lim_{k\to\infty} (A_k)_{ii} = \lambda_i$$

**Remarque 14** ( 1 ) L'hypothèse technique ci-dessus n'est pas vraiment essentielle. Si elle n'est pas satisfaite, la méthode $QR$ converge cependant, mais les $\lambda_i$ ne sont plus rangée s dans l'ordre décroissant des modules.

(2) Si les module s de s valeurs propre s ne sont pas tous différent s, $(A_i)$ tend (en général) vers une matrice triangulaire par blocs, un bloc correspondant à un module. Cette situation se présente en particulier pour des matrices réelles ayant des valeurs propres non réelles (elles sont alors conjuguées deux à deux).

(3) On montre facilement que si $A$ est de Hessenberg ou une matrice bande hermitienne, alors les $A_i$ ont la même structure. Il en résulte qu'on commence toujours par réduire une matrice sous forme Hessenberg (ou sous une forme tridiagonale si $A$ est hermitienne) avant d'appliquer l'algorithme $QR$. Le coût de calcul est alors considérablement réduit.

(4) L'algorithme $QR$ permet aussi le plus souvent l'obtention des vecteurs propres. Il est difficiles de donner des énoncés suffisamment généraux, mais on peut s'en convaincre à l'ai de du raisonnement approximatif suivant. No-

tons $\Omega_k = Q_1 Q_2 \dots Q_k$. On a alors :

$$A_k = \Omega_k^{-1} A \Omega_k$$

Supposons que pour $k$ grand, $A_k = T$ soit exactement triangulaire. Alors:

$$A\Omega_k = \Omega_k T$$

En particulier

$$A\Omega_k e_1 = \Omega_k T e_1 = \lambda_1 \Omega_k e_1$$

et $\Omega_k e_1$, c'est-à-dire le premier vecteur colonne de $\Omega_k$, est vecteur propre de $A$ pour la valeur propre de $\lambda_1$. En suite, on vérifie que pour $i \geq 2$, le vecteur $q^i = \left(q_j^i\right)_{j=1,\dots,N}$ est vecteur propre pour $\lambda_i$ s'il vérifie

$$\left\{ q_j^i = 0 \quad \forall j = i+1, \dots, N q_i^i = 1 q_j^i = -\left(t_{j,j+1} q_{j+1}^i + \dots + t_{ji} q_i^i\right) / \left(\lambda_j - \lambda_i\right) \quad \forall j = i-1, \dots 1 \right.$$

Si on ne s'intéresse qu'à quelques vecteurs propres, on peut aussi utiliser la méthode de la puissance itérée avec translation une fois déterminée la valeur propre.

(5) Si la matrice initiale $A$ est réelle, les matrices $A_i$ sont réelles et ne peuvent converger vers une matrice triangulaire que si toutes les valeurs propres sont réelles. Pour atteindre les valeurs propres complexes, on peut procéder à une translation d'argument complexe pour "séparer" les modules. Il faut alors travailler en arithmétique complexe. Certains algorithmes

couplant deux translations complexes conjuguées successives permettent de s'en passer. Nous renvoyons à la littérature spécialisée pour une description de ces méthodes plus sophistiquées.

(6) Algorithme $QR$ avec translations : déjà évoqué au point (5) ci-dessus, il permet surtout d' accélérer très sensiblement la vitesse de convergence de l'algorithme. En général, la convergence est linéaire : les éléments $A_{ij}$ tendent vers 0 comme $\left(\frac{\lambda_i}{\lambda_j}\right)^k$ sous les hypothèse s du théorème 7.4. Si la matrice $A$ est sous forme Hessenberg, la vitesse de convergence est donc régie par les rapports $\frac{\lambda_i}{\lambda_{i-1}}$. On s'arrange donc pour effectuer des translations pour diminuer ces rapports. Ain si, à chaque étape, on choisit un réel $s_k$ et $A_{k+1}$ est déterminé comme suit

$$A_k - s_k I = Q_k R_k A_{k+1} = R_k Q_k + s_k I = Q_k^{-1} A_k Q_k$$

On peut choisir $s_k$ pour que le rapport $|\lambda_{\mathrm{v}} - s_k| \,/\, |\lambda_{\mathrm{N}-1} - s_k|$ soit aussi petit que possible. Un choix fréquent consiste à choisir pour $s_k$ l'une des valeurs propres de la matrice

$$\begin{bmatrix} a^k_{\mathrm{N}-1,\mathrm{N}-1} & a^k_{\mathrm{N}-1,\mathrm{N}} \\ a^k_{\mathrm{N},\mathrm{N}-1} & a^k_{\mathrm{N},\mathrm{N}} \end{bmatrix}$$

Pour des matrices tridiagonales symétriques, ceci conduit souvent à une convergence cubique de $a^k_{\mathrm{v},\mathrm{N}-1}$ vers 0. Lorsque ce terme est jugé suffisamment petit, on peut poursuivre les calculs en laissant tomber la $N$ -ième ligne et

la $N$ -ième colonne de $A^k$.

(7) Pour terminer, insistons sur le fait que la méthode $QR$ appliquée avec les amélioration s évoquées ci-dessus (réduction préalable de s matrices, translations appropriées) est une méthode très performante. Même pour une matrice symétrique, les expériences montrent qu'elle est environ 10 fois plus rapide que la méthode de Jacobi lorsqu'on ne calcule que les valeurs propres et encore 4 fois plus rapide si on calcule à la fois valeurs propres et vecteurs propres.

# Chapitre 6

# Résolution d'équations et systèmes d'équations non linéaires

Le dernier chapitre de ce cours est consacré à la résolution d'équations et de systèmes d'équations non linéaires.

On considère tout d'abord une fonction $f : R \to R$ d'une seule variable réelle et on cherche à résoudre l'équation $f(x) = 0$ c'est-à-dire trouver une valeur approchée $\bar{x}$ d'un réel $\bar{x}$ vérifiant $f(\tilde{x}) = 0$

La mise en oeuvre pratique des méthodes que nous allons voir nécessite la donnée d'une tolérance sur la solution que l'on cherche à calculer. L'algorithme

numérique utilisé doit alors avoir un critère d'arrêt dépendant de cette tolérance et nous assurant que la solution calculée a bien la précision recherchée. En fonction de la méthode utilisée, on peut parfois savoir à l'avance combien d'étapes de l'algorithme sont nécessaires pour obtenir la précision recherchée (méthode de dichotomie) ou alors il nous faut à chaque étape vérifier une condition nous permettant d'arrêter le processus lorsque l'on est certain d'avoir obtenu la précision requise sur la solution (méthodes de points fixes). Pour comparer les différentes méthodes de résolution que l'on va considérer, on utilise les notions suivantes de vitesse de convergence d'une suite :

Définition 7.1. Soit $(x_n)_{n\in N}$ une suite convergente et soit $\tilde{x}$ sa limite.

1. On dit que la convergence de $(x_n)_{n\in\mathrm{N}}$ est linéaire de facteur $K \in ]\,0,1\,[s$ 'il existe $n_0 \in N$ tel que, pour tout $n \geq n_0, |x_{n+1} - \tilde{x}| \leq K\,|x_n - \tilde{x}|$

2. On dit que la convergence de $(x_n)_{n\in\mathrm{N}}$ est superlinéaire d'ordre $p \in N, p > 1 s'il$ existe $n_0 \in N$ et $K > 0$ tels que, pour tout $n \geq n_0, |x_{n+1} - \tilde{x}| \leq K\,|x_n - \tilde{x}|^p$. Si $p = 2$, on parle de convergence quadratique et si $p = 3$ on parle de convergence cubique.

Remarquons que $K$ n'est pas unique et qu'en pratique il peut ètre difficile de prouver la convergence d'une méthode d'autant plus qu'il faut tenir compte des erreurs d'arrondis. On utilise en général les notions de convergence plus faible suivantes :

Définition 7.2. Soit $(x_n)_{n\in N}$ une suite convergent vers une limite $\bar{x}$. On dit que la convergence de $(x_n)_{n\in N}$ est linéaire de facteur $K$ (resp. superlinéaire d'ordre $p$ ) $s$ 'il existe une suite $(y_n)_{n\in N}$ convergent vers 0, linéaire de facteur $K$ (resp. superlinéaire d'ordre p) au sens de la définition 7.1 telle que $|x_n - \tilde{x}| \leq y_n$

Soit $d_n = -\log_{10}(|x_n - \tilde{x}|)$ une mesure du nombre de décimales exactes de $x_n$. Si la convergence est d'ordre $p$, alors asymptotiquement, on a $|x_{n+1} - \tilde{x}| \sim K|x_n - \tilde{x}|^p$ d'où $-d_{n+1} \sim \log_{10}(K) - pd_n$ et donc asymptotiquement $x_{n+1}$ a $p$ fois plus de décimales exactes que $x_n$. Ainsi, l'ordre $p$ de la convergence représente asymptotiquement le facteur multiplicatif du nombre de décimales exactes que l'on gagne à chaque itération. Nous avons donc intérêt à ce qu'il soit le plus grand possible.

## 6.1 Méthode de dichotonnie

La méthode classique de dichotomie est une méthode de localisation des racines d'une équation $f(x) = 0$ basée sur le théorème des valeurs intermédiaires : si $f$ est continue sur $[a, b]$ et $f(a)f(b) < 0$, alors il existe $\tilde{x} \in ]a, b[$ tel que $f(\bar{x}) = 0$. L'idée est donc de partir d'un intervalle $[a, b]$ vérifiant la propriété $f(a)f(b) < 0$, de le scinder en deux intervalles $[a, c]$ et $[c, b]$ avec $c = \frac{a+b}{2}$, et de tester les bornes des nouveaux intervalles (on cal-

cule $f(a)f(c)$ et $f(c)f(b)$) pour en trouver un (au moins) qui vérifie encore la propriété, i.e., $f(a)f(c) < 0$ ou / et $f(c)f(b) < 0$. On itère ensuite ce procédé un certain nombre de fois dépendant de la précision que l'on recherche sur la solution (voir Théorème 7.3 ci-dessous). On obtient l'algorithme suivant :

Entrées : la fonction [1]$f$, $(a, b) \in R^2$ tels que $f$ est continue sur $[a, b]$ et $f(a)f(b) < 0$ et la précision $\epsilon$

Sortie : $x_{k+1}$ valeur approchée de $\tilde{x}$ solution de $f(\tilde{x}) = 0$ à $\epsilon$ près.

1. $x_0 \leftarrow a, y_0 \leftarrow b$

2. Pour $k$ de 0 à $E\left(\frac{\ln(b-a)-\ln(\epsilon)}{\ln(2)}\right)$ par pas de 1, faire :

• Si $f(x_k) f\left(\frac{x_k+y_k}{2}\right) > 0$, alors $x_{k+1} \leftarrow \frac{x_k+y_k}{2}, y_{k+1} \leftarrow y_k$

• Si $f(x_k) f\left(\frac{x_k+y_k}{2}\right) < 0$, alors $x_{k+1} \leftarrow x_k, y_{k+1} \leftarrow \frac{x_k+y_k}{2}$

• Sinon retourner $\frac{x_k+y_k}{2}$

[1] Il suffit en fait de connaître un moyen d'évaluer les valeurs de la fonction

3. Retourner $x_{k+1}$.

Cet algorithme construit une suite de segments emboités contenant tous la solution $\tilde{x}$.

A chaque passage dans la boucle nous devons calculer une évaluation de $f$. Remarquons qu'en pratique, avec les arrondis, $\ll > 0$ et $4 < 0$ ne veulent rien dire ! On démontre maintenant que cet algorithme est correct dans le

sens où il calcule bien une valeur approchée de la solution à $\epsilon$ près.

Théorème 7.3. Le nombre minimum d'itérations de la méthode de dichotomie nécessaire $x$.

Démonstration. A la première itération, la longueur de l'intervalle est $\frac{b-a}{2}, \ldots,$ à la nième itération, la longueur de l'intervalle est $\frac{b-a}{2^n}$. L'erreur commise a l'étape $n$ est donc majorée $n \geq \frac{\ln(b-a)-\ln(c)}{\ln(2)}$ d'où le résultat.

Proposition 7.4. La convergence de la méthode de dichotomie est linéaire de facteur $\frac{1}{2}$.

Démonstration. Prendre $y_n = \frac{b-a}{2^n}$ dans la définition 7.2 .

## 6.2 Méthode du point fixe

La méthode itérative du point fixe que nous allons décrire est aussi appelé méthode des approximations successives.

Définition 7.5. Soit $g : R \to R$. On dit que $x \in R$ est un point fixe de $g$ si $g(x) = x$

Le principe de la méthode du point fixe est d'associer a l'équation $f(x) = 0$ une équation de point fixe $g(x) = x$ de sorte que trouver une solution de $f(x) = 0$ équivaut is trouver un point fixe de $g$. La technique pour approximer le point fixe de $g$ est alors basée sur le résultat suivant :

Lemme 7.6. Soit $(x_n)_{n\in\mathrm{N}}$ la suite définie par $x_0 \in R$ donné et $x_{n+1} = g(x_n)$. Si $(x_n)_{nenest}$ convergente et $g$ est continue, alors la limite de $(x_n)_{n\in N}$ est un point fixe de $g$.

Nous devons donc trouver des conditions sur $g$ pour que la suite $(x_n)_{n\in V}$ définie ci-dessus converge.

Définition 7.7 . Soit $g : \Omega \subseteq R \to R$. On dit que $g$ est lipschitzienne sur $\Omega$ de constante de lipschitz $\gamma$ (ou $\gamma$ -lipschitzienne) si pour tout $(x, y) \in \Omega^2$, on $a|g(x) - g(y)| \leq \gamma|x - y|$. On dit que $g$ est strictement contractante sur $\Omega$ si $g$ est $\gamma$ -lipschitzienne sur $\Omega$ avec $\gamma < 1$.

On remarque que $g$ lipschitzienne sur $\Omega$ implique en particulier $g$ continue sur $\Omega$.

Théorème 7.8 (Théorème du point fixe). Soit $g$ une application strictement contractante sur un intervalle $[a, b] \subset R$ de constante de Lipschitz $\gamma < 1$. Supposons que l'intervalle $[a, b]$ soit stable sous $g$, i.e., $g([a, b]) \subseteq [a, b]$ ou encore pour tout $x \in [a, b], g(x) \in [a, b]$. Alors $g$ admet un unique point fixe $x^* \in [a, b]$ et la suite définie par $x_{n+1} = g(x_n)$ converge linéairement de facteur $\gamma$ vers $x^*$ pour tout point initial $x_0 \in [a, b]$. De plus,

$$\forall n \in N, |x_n - x^*| \leq \frac{\gamma^n}{1-\gamma} |x_1 - x_0|$$

Démonstration. Admis pour ce cours.

On remarque que l'erreur est d'autant plus petite que $\gamma$ est proche de 0. De plus, on peut montrer que l'on a aussi

$$\forall n \in N, |x_n - x^*| \leq \frac{\gamma}{1-\gamma} |x_n - x_{n-1}|$$

et si $\gamma \leq \frac{1}{2}$, alors $|x_n - x^*| \leq |x_n - x_{n-1}|$. Dans ce cas, on pourra utiliser le test d'arrêt $|x_n - x_{n-1}| < \epsilon$ qui certifiera une précision $\epsilon$ sur le résultat.

La proposition suivante donne un critère pour tester le fait qu'une fonction soit contractante sur un intervalle donné.

Proposition 7.9. Soit $g$ une fonction dérivable sur l'intervalle $[a, b]$. Si sa dérivée $g'$ vérifie $\max_{x\in[a,b]} |g'(x)| = L < 1$, alors $g$ est strictement contractante sur $[a, b]$ de constante de Lipschitz $L$.

Démonstration. Utiliser le théorème des accroissements finis.

On en déduit alors la proposition suivante :

Proposition 7.10. Soit $x^* \in [a, b]$ un point fixe d'une fonction $g \in \mathcal{C}^1([a, b])$

$\bullet Si |g'(x^*)| < 1$, alors il existe un intervalle $[\alpha, \beta] \subseteq [a, b]$ contenant $x^*$ pour lequel la suite définie par $x_0 \in [\alpha, \beta]$ et $x_{n+1} = g(x_n)$ converge vers $x^*$;

$\bullet Si |g'(x^*)| > 1$, alors pour tout $x_0 \neq x^*$, la suite définie par $x_0$ et $x_{n+1} = g(x_n)$ ne converge pas vers $x^*$;

$\bullet Si |g'(x^*)| = 1$, on ne peut pas conclure.

En pratique, on estime $g'(x^*)$, *i.e.*, *on en a une valeur approchée* $\overline{g'(x^*)}$. *Si* $\left|\overline{g'(x^*)}\right| > 1$, *alors on élimine la méthode et si* $\left|\overline{g'(x^*)}\right| < 1$, on cherche un intervalle $[\alpha, \beta] \subseteq$

$[a, b]$ dans lequel $\max_{x\in[\alpha,\beta]} |g'(x)| < 1$ et $g([\alpha, \beta]) \subseteq [\alpha, \beta]$

Revenons maintenant à notre problème initial où on cherche à résoudre une équation $f(x) = 0$. Posons $g(x) = x - f(x)$ de sorte que trouver les solutions de $f(x) = 0$ soit équivalent à trouver les points fixes de $g$. D'après le théorème du point fixe (Théorème 7.8 ), une condition suffisante pour que $g$ admette un point fixe dans l'intervalle $[a, b]$ est que $[a, b]$ soit stable sous $g$ et que $g$ soit strictement contractante sur $[a, b]$ de constante de Lipschitz $\gamma < 1$. On a alors (conséquence directe de la définition d'une fonction contractante)

$$\forall x \in [a, b], \quad |g'(x)| < \gamma \Longleftrightarrow |1 - f'(x)| < \gamma$$

qui implique en particulier que $f'$ ne s'annule pas sur $[a, b]$ et que $f$ est donc monotone sur $[a, b]$.

Étant donnée $f$, le choix précédent de $g$ est restrictif et on peut choisir $g(x) = x - \lambda f(x)$ où $\lambda$ est une constante arbitraire non nulle. Comme précédemment, une condition suffisante pour que $g$ ait un point fixe dans $[a, b]$ est que $[a, b]$ soit stable sous $g$ et que $g$ soit strictement contractante sur $[a, b]$ de constante de Lipschitz $\gamma < 1$. On obtient alors

$$\forall x \in [a, b], \quad |1 - \lambda f'(x)| < \gamma < 1$$

ce qui implique en particulier que $f'$ ne change pas de signe sur $[a, b]$ et que $\lambda$

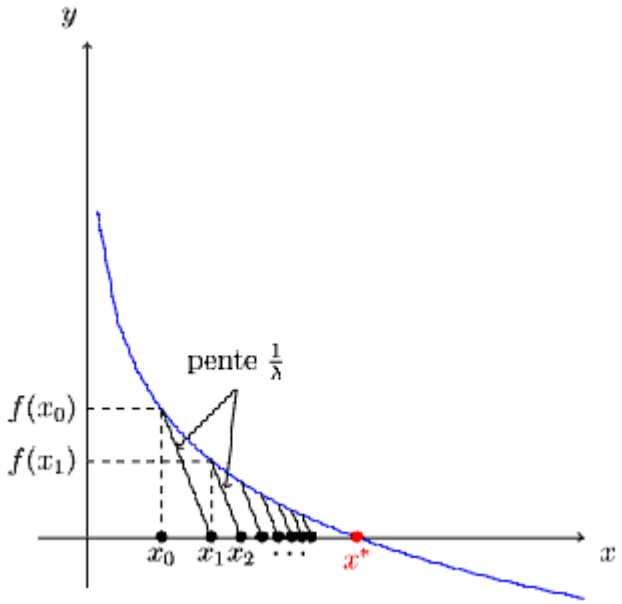



Figure 6.1:

est du même signe que $f'$. Géométriquement, on construit la suite des itérés $x_{n+1} = x_n - \lambda f(x_n)$. En remarquant que la droite de pente $\mu$ et passant par le point $(x_n, f(x_n))$ a pour équation $y = f(x_n) + \mu(x - x_n)$ et coupe l'axe des abscisses en $x = x_n - \frac{f(x_n)}{\mu}$, on voit que $x_{n+1}$ s'obtient comme point d'intersection de la droite de pente $\frac{1}{\lambda}$ passant par le point $(x_n, f(x_n))$ avec l'axe des abscisses. En itérant ce procédé, on obtient la méthode illustrée sur le graphique suivant :

La proposition suivante donne des précisions sur l'ordre de convergence d'une

méthode de point fixe.

Proposition 7.11. On considère l'équation $g(x) = x$ où $g$ est une fonction au moins $p+1$ fois dérivable avec $p \geq 1$. Supposons que les hypothèses du théorème 7.8 soient vérifiées de sorte que $g$ admette un unique point fixe $x^* \in [a, b]$. Si $g'(x^*) = g''(x^*) = \cdots = g^{(p)}(x^*) = 0$ et $g^{(p+1)}(x^*) \neq 0$, alors la convergence de la méthode $x_{n+1} = g(x_n)$ est superlinéaire d'ordre $p+1$

Démonstration. Utiliser la formule de Taylor.

## 6.3 Méthode de Newton

Si on regarde l'équation (7.1), on voit que la méthode convergera d'autant plus vite que la constante $\gamma$ est petite. Ceci motive donc le fait de remplacer la constante $\lambda$ par $\frac{1}{f'(x)}$ et de poser $g(x) = x - \frac{f(x)}{f'(x)}$

Définition 7.12. La fonction d'itération de Newton associée à l'équation $f(x) = 0$ sur $[a, b]$ est

$$\mathcal{N} : \left\{ [a, b] \to Rx \mapsto \mathcal{N}(x) = x - \frac{f(x)}{f'(x)} \right.$$

Cette fonction est définie pour $f$ dérivable sur $[a, b]$ et telle que $f'$ ne s'annule pas sur $[a, b]$.

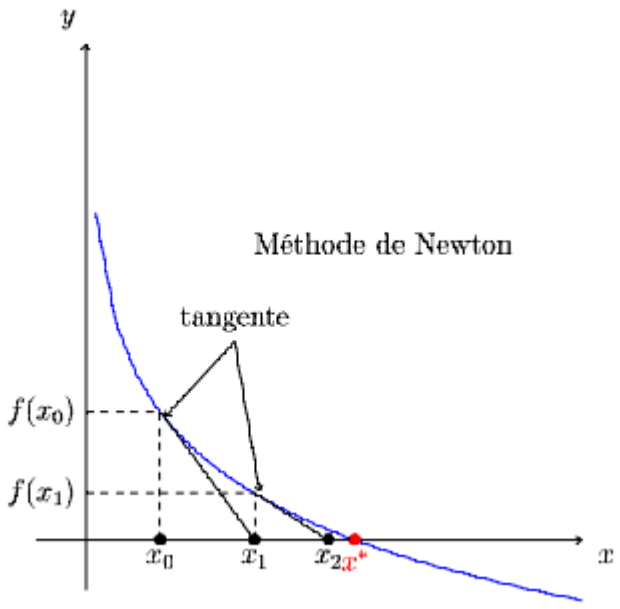



Figure 6.2:

Géométriquement, la suite des itérés de Newton se construit comme suit : étant donné $x_n \in [a, b]$, on cherche à construire $x_{n+1}$ tel que $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$. En reprenant, le raisonnement développé dans la sous-section précédente, on voit que $x_{n+1}$ s'obtient comme point d'intersection de la droite tangente au graphe de $f$ en $(x_n, f(x_n))$ avec l'axe des abscisses.

On illustre ceci sur le graphique suivant :

Théorème 7.13 (Théorème de convergence locale). Soit $f$ une fonction de classe $\mathcal{C}^2$ sur un intervalle $[a, b]$ de $R$. On suppose $qu$ 'il existe $\tilde{x} \in [a, b]$ tel

que $f(\tilde{x}) = 0$ et $f'(\tilde{x}) \neq 0$($\tilde{x}$ est un zéro simple de $f$ ). Alors il existe $\epsilon > 0$, tel que pour tout $x_0 \in [\tilde{x} - \epsilon, \bar{x} + \epsilon]$, la suite des itérés de Newton donnée par $x_{n+1} = \mathcal{N}(x_n)$ pour $n \geq 1$ est bien définie, reste dans l'intervalle $[\tilde{x} - \epsilon, \tilde{x} + \epsilon]$ et converge vers $\tilde{x}$ quand $n$ tend vers l'infini. De plus, cette convergence est ($au$ moins) quadratique.

Démonstration. Admis pour ce cours.

Exemple : Prenons l'exemple du calcul de la racine carrée d'un nombre réel $a > 0$. On cherche alors à résoudre l'équation $f(x) = 0$ avec $f(x) = x^2 - a$. On a alors

$$\mathcal{N}(x) = x - \frac{f(x)}{f'(x)} = x - \frac{x^2 - a}{2x} = \frac{1}{2}\left(x + \frac{a}{x}\right)$$

Dans le cas $a = 2$, si l'on prend comme point de départ $x_0 = 1$ on obtient les valeurs suivantes des itérés de Newton :

$$\begin{aligned}
x_0 &= 1,000000000000000 \\
x_1 &= 1,500000000000000 \\
x_2 &= 1,416666666666667 \\
x_3 &= 1,414215686274510 \\
x_4 &= 1.414213562374690 \\
x_5 &= 1,414213562373095
\end{aligned}$$

La valeur de $\sqrt{2}$ donnée par MATLAB en format long est $1,414213562373095$.

Le nombre de décimales justes double approximativement à chaque itération ce qui est bien cohérent avec le fait que la convergence de la méthode de Newton est quadratique. Notons que si on utilise la méthode de dichotomie sur l'intervalle $[1,2]$, alors nous avons besoin de 51 itérations pour avoir une valeur approchée de $\sqrt{2}$ à $10^{-15}$ près (voir Théorème 7.3 ).

Le théorème 7.13 suppose que $\tilde{x}$ est un zéro simple de $f$. Nous avons la généralisation suivante au cas d'un zéro de multiplicité quelconque.

Théorème 7.14. Avec les notations, précédentes, si $\tilde{x}$ est un zéro de multiplicité $m$ de $f$, i.e., $f(x^*) = f'(x^*) = \cdots = f^{(m-1)}(x^*) = 0$ et $f^{(m)}(x^*) \neq 0$, alors la méthode itérative définie $parx_{n+1} = \mathcal{N}_m(x_n)$ avec $\mathcal{N}_m(x_n) = x - m\frac{f(x)}{f'(x)}$ est d'ordre supérieure ou égal $\dot{a}2$

Finalement, nous avons le résultat global suivant :

Théorème 7.15 (Théorème de convergence globale). Soit $f$ une fonction de classe $\mathcal{C}^2$ sur un intervalle $[a,b]$ de $R$ vérifiant :

$\bullet f(a)f(b) < 0$

$\bullet \forall x \in [a,b], f'(x) \neq 0 (f$ strictement monotone)

$\bullet \forall x \in [a,b], f''(x) \neq 0$ (concavité dans le même sens sur $[a,b]$)

Alors, en choisissant $x_0 \in [a,b]$ tel que $f(x_0) f''(x_0) > 0$, la suite $(x_n)_{n\in\mathrm{N}}$

définie par $x_0$ et $x_{n+1} = \mathcal{N}(x_n)$ converge vers l'unique solution de $f(x) = 0$ dans $[a, b]$

Démonstration. Admis pour ce cours.

## 6.4 Méthode de la sécante

La méthode de Newton étudiée dans la sous-section précédente présente le désavantage de nécessiter le calcul de la dérivée de la fonction $f$ qui peut s'avérer difficile. Gardons en tête qu'on ne connaît pas nécessairement une expression symbolique de f. Partant de ce constat, l'idée de la méthode de la sécante est de remplacer la dérivée f de f qui apparaît dans la méthode de Newton par une différence divisée (voir la définition 5.9). La méthode de la sécante est alors donnée par l'itération suivante : $x_{n+1} = x_n - \frac{f(x_n)}{f[x_n, x_{n-1}]}$,

où $\quad f[x_n, x_{n-1}] = \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}$

Cette méthode doit alors être initialisée par deux points $x_0$ et $x_1$ à partir desquels on peut utiliser la formule précédente pour calculer $x_2$ et les itérés suivants.

Théorème 7.16. Soit $f$ une fonction de classe $\mathcal{C}^2$ sur un intervalle $[a, b]$ de $R$. On suppose $qu$ 'il existe $\tilde{x} \in [a, b]$ tel que $f(\tilde{x}) = 0$ et $f'(\tilde{x}) \neq 0$($\tilde{x}$ est un zéro

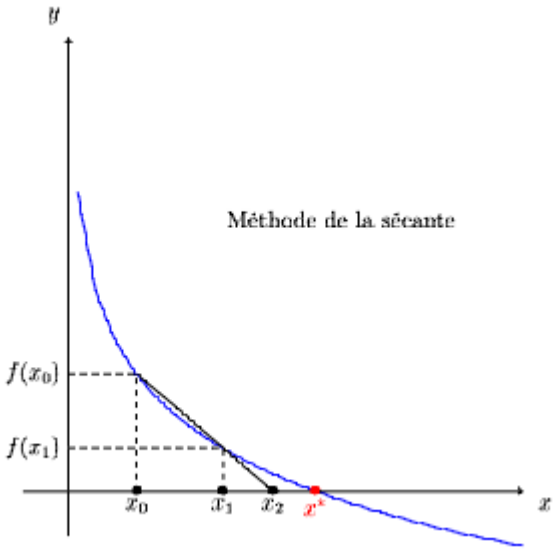



Figure 6.3:

simple de $f$). Alors il existe $\epsilon > 0$, tel que pour tout $x_0, x_1 \in [\tilde{x} - \epsilon, \tilde{x} + \epsilon]$, la suite des itérés de la méthode de la sécante donnée par (7.2) pour $n \geq 1$ est bien définie, reste dans l'intervalle $[\tilde{x} - \epsilon, \tilde{x} + \epsilon]$ et converge vers $\tilde{x}$ quand $n$ tend vers l'infini. De plus, cette convergence est d'ordre $p = \frac{1+\sqrt{5}}{2} \approx 1,618$ (nombre d'or).

Démonstration. Admis pour ce cours.

Exemple : Reprenons l'exemple du calcul de la racine carrée d'un réel $a > 0$ en résolvant $f(x) = 0$ pour $f(x) = x^2 - a$. La suite des itérés de la méthode

de la sécante est donnée par :

$$x_{n+1} = x_n - \frac{x_n^2 - a}{\frac{(x_n^2-a)-\left(x_{n-1}^2-a\right)}{x_n - x_{n-1}}} = x_n - \frac{x_n^2 - a}{x_n + x_{n-1}}$$

Dans le cas $a = 2$, si l'on prend comme points de départ $x_0 = x_1 = 1$ on obtient les valeurs suivantes des itérés de la méthode de la sécante :

$$\begin{aligned}
x_0 &= 1,000000000000000 \\
x_1 &= 1,000000000000000 \\
x_2 &= 1,500000000000000 \\
x_3 &= 1,400000000000000 \\
x_4 &= 1,413793103448276 \\
x_5 &= 1,414215686274510 \\
x_6 &= 1,414213562057320 \\
x_7 &= 1,414213562373095
\end{aligned}$$

On voit donc que l'on a besoin de plus d'itérations que pour la méthode de Newton pour avoir la même précision. Ceci vient du fait que l'ordre de cette méthode est plus petit que celui de la méthode de Newton. Par contre, on n'a pas besoin de calculer la dérivée.

## 6.5 Systèmes d'équations non linéaires

On considère maintenant un système d'équations non linéaires donné par une fonction $f : R^n \to R^n, x = (x_1 \dots x_n)^T \mapsto f(x) = (f_1(x_1, \dots, x_n), \dots, f_n(x_1, \dots, x_n))^T$.

On cherche donc un vecteur $x = (x_1 \dots x_n)^T \in R^n$ tel que

$$f(x) = 0_{\mathrm{R}^n} \iff \begin{cases} f_1(x_1, \dots, x_n) = 0 \\ \vdots \\ f_n(x_1, \dots, x_n) = 0 \end{cases}$$

La méthode décrite à la fin de la section 7.2 se généralise immédiatement à ce cadre en définissant l'itération

$$x^{(n+1)} = x^{(n)} + M^{-1} f\left(x^{(n)}\right)$$

où $M$ est une certaine matrice, et nous avons les mêmes résultats de convergence que dans le cas d'une seule équation.

Définition 7.17. La matrice jacobienne d'une fonction $f : R^n \to R^n$ notée $J_f$

est définie (lorsqu'elle existe) par :

$$\forall x = (x_1, \dots x_n)^T \in R^n, \quad J_f(x) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(x) & \frac{\partial f_1}{\partial x_2}(x) & \dots \\ \frac{\partial f_1}{\partial x_n}(x) & & \\ \frac{\partial f_2}{\partial x_1}(x) & \frac{\partial f_2}{\partial x_2}(x) & \dots \\ \frac{\partial f_2}{\partial x_n}(x) & & \\ \vdots & \vdots & \ddots \\ \vdots & & \\ \frac{\partial f_n}{\partial x_1}(x) & \frac{\partial f_n}{\partial x_2}(x) & \dots \\ \frac{\partial f_n}{\partial x_n}(x) & & \end{pmatrix}$$

La méthode de Newton se généralise naturellement au cas des systèmes d'équations non linéaires de la manière suivante : on choisit $x^{(0)} \in R^n$ et on utilise la formule d'itération

$$x^{(n+1)} = x^{(n)} - J_f\left(x^{(n)}\right)^{-1} f\left(x^{(n)}\right)$$

où $J_f\left(x^{(n)}\right)^{-1}$ désigne l'inverse de la matrice jacobienne de $f$ évaluée en $x^{(n)}$.

Théorème 7.18. Soit $f : R^n \to R^n$ une fonction de classe $\mathcal{C}^2$ sur une boule fermée $B$ de $R^n$. On suppose $qu$ 'il existe un zéro $\tilde{x}$ de $f$ dans $B$ et que $J_f(\tilde{x})$ est inversible. Alors il existe $\epsilon > 0$ tel que pour tout $x^{(0)} \in B$ tel que $\left\|x^{(0)} - \tilde{x}\right\| \leq \epsilon$, la suite des itérés de la méthode de Newton définie par (7.4)

est bien définie et converge vers $\bar{x}$ quand $n$ tend vers l'infini.

Calculer l'itéré $n+1$ à partir de l'itéré $n$ en utilisant la formule (7.4) nécessite d'inverser la matrice $J_f\left(x^{(n)}\right)$. Or, calculer l'inverse d'une matrice peut s'avérer coûteux. Par conséquent, nous ré-écrivons la formule d'itération (7.4) sous la forme

$$J_f\left(x^{(n)}\right)\left(x^{(n+1)} - x^{(n)}\right) = -f\left(x^{(n)}\right)$$

de sorte qu'à chaque itération, le calcul de l'inverse d'une matrice est remplacé par la résolution d'un système d'équations linéaires ce qui est asymptotiquement moins coûteux en nombre d'opérations à virgule flottante (voir la sous-section 2.2 .4 du chapitre 2 ).

Exemple : Considérons le système d'équations non linéaires : (S) : $\{x_1^2 + 2x_1 - x_2^2 - 2 = 0 x_1^3 + 3x_1x_2^2 -$

Avec les notations précédentes, cela correspond à $n = 2, f_1(x_1, x_2) = x_1^2 + 2x_1 - x_2^2 - 2$, et $f_2(x_1, x_2) = x_1^3 + 3x_1x_2^2 - x_2^3 - 3$. La matrice jacobienne de $f$ est alors :

$$J_f(x_1, x_2) = \begin{pmatrix} 2x_1 + 2 & -2x_2 \\ 3\left(x_1^2 + x_2^2\right) & 6x_1x_2 - 3x_2^2 \end{pmatrix}$$

Partant du point $x^{(0)} = (1 \quad -1)^T$, calculons le premier itéré de la méthode de Newton pour résoudre le système $(S)$. Pour $n = 1$, la formule d'itération

(7.5) s'écrit :

$$J_f\left(x^{(0)}\right)\left(x^{(1)}-x^{(0)}\right)=-f\left(x^{(0)}\right)$$

c'est-à-dire

$$\begin{pmatrix} 4 & 2 \\ 6 & -9 \end{pmatrix}\begin{pmatrix} x_1^{(1)}-1 \\ x_2^{(1)}+1 \end{pmatrix}=-\begin{pmatrix} 0 \\ 2 \end{pmatrix}$$

En résolvant ce système linéaire, on trouve $x_1^{(1)}-1=-\frac{1}{12}$ et $x_2^{(1)}+1=\frac{1}{6}$ de sorte que

$x^{(1)}=\left(\frac{11}{12}-\frac{5}{6}\right)^T$

La méthode de la sécante ne se généralise pas facilement au cas de plusieurs équations. En pratique, pour résoudre un système de plusieurs équations non linéaires, soit on utilise la méthode de Newton, soit on utilise une méthode type (7.3) mais en ajustant la matrice $M$ au bout d'un certain nombre d'itérations. En général, on prend pour $M$ une matrice assez proche de la jacobienne $J_f$ de $f$ pour que la convergence soit d'un ordre supérieur à 1. On obtient ainsi des méthodes de Newton généralisées qui sont surtout utilisées dans le cadre de l'optimisation.