Algèbre Matricielle Numérique

Claude Brezinski

Université des Sciences et Technologies de Lille
Claude.Brezinski@univ-lille1.fr

Chapitre 1

Arithmétique de l'ordinateur

Les algorithmes de l'analyse numérique sont destinés à être programmés sur ordinateur. Dans ce Chapitre, nous allons donc étudier l'arithmétique des ordinateurs.

1.1 Représentation des nombres

Soit a un nombre réel. On peut toujours l'écrire, en le multipliant par une puissance de 10 adéquate

$$a = \pm 0.a_1a_2a_3\ldots\ 10^q = \pm 10^q \sum_{i=1}^{\infty} a_i\ 10^{-i}$$

où $a_i \in \{0, 1, \ldots, 9\}$ avec $a_1 \neq 0$ et $q \in \mathbb{Z}$, où $\mathbb{Z}$ est l'ensemble des nombres entiers relatifs (c'est–à–dire avec un signe). C'est ce que l'on appelle la représentation du nombre a en *virgule flottante normalisée*. q s'appelle l'*exposant*, $a_1a_2a_3\ldots$ la *mantisse* et les a_i les *chiffres* (en anglais les *digits* ou, lorsque l'on travaille dans le système binaire, les *bits* pour "binary digits") de a. En général, un nombre réel possède une mantisse avec un nombre infini de chiffres (π par exemple).

Dans un ordinateur, chaque nombre est placé dans un *mot* de la mémoire. Un mot est formé d'un nombre fini de cases, chacune ne pouvant contenir qu'un seul chiffre. Dans la première, on va placer le signe de a, puis les chiffres successifs de la mantisse de a, ensuite le signe de son exposant et enfin les chiffres de son exposant. L'arithmétique de l'ordinateur est une arithmétique dérivée de l'arithmétique binaire. Il n'est pas dans notre intention d'expliquer ici cette arithmétique. Cela n'aurait, en effet, aucun intérêt pour notre propos et le compliquerait inutilement. Nous ferons donc comme si l'ordinateur travaillait en base 10. Cela ne changera rien ni aux raisonnements ni aux conclusions de notre étude, seules quelques constantes dans les inégalités que nous allons donner seront changées.

Chaque mot de la mémoire d'un ordinateur ne peut donc contenir qu'un nombre fini de chiffres. En particulier, nous appellerons t le nombre de chiffres décimaux de la mantisse des mots de l'ordinateur (en général, on aura $t = 7$ ou 8 en simple précision, 15 ou 16 en double précision). Par conséquent, seuls seront représentés de façon exacte les nombres dont la mantisse ne dépasse pas t chiffres. Le problème qui se pose à nous maintenant est simple : comment représenter un nombre réel a ayant une mantisse avec un nombre de chiffres supérieur à t à l'aide d'un mot de la mémoire qui ne peut en contenir que t ? Il y a deux possiblités

1. la troncature qui consiste à couper (à tronquer) la mantisse de a après son t-ième chiffre,

2. l'arrondi qui consiste à tenir compte du $(t + 1)$-ième chiffre. Si celui–ci est inférieur à 5, on tronque tandis que, s'il est supérieur ou égal à 5, on ajoute une unité au t-ième chiffre avant de tronquer. Il y a quatre sortes d'arrondi : supérieur, inférieur, vers zéro et au plus proche. Nous venons de décrire celui au plus proche tandis que la troncature correspond à l'arrondi vers zéro.

Dans les deux cas, le nombre réel a est donc représenté dans l'ordinateur par un nombre qui, en général, en est une approximation. Nous l'appellerons $\mathrm{fl}(a)$ que l'ordinateur travaille par troncature ou par arrondi. L'erreur ainsi commise s'appelle *erreur d'affectation* parce qu'au nombre réel a on affecte un mot contenant le nombre $fl(a)$. On a le

Théorème 1

$$|a - fl(a)| \leq K|a|\ 10^{-t}$$

où $K = 5$ si l'ordinateur travaille par arrondi et 10 s'il travaille par troncature.

Démonstration. Donnons la démonstration dans le cas de la troncature. Il est évident que, pour l'arrondi, l'erreur est deux fois plus faible. On a

$$\begin{aligned} |a - \mathrm{fl}(a)| &= 0.0\ldots 0a_{t+1}a_{t+2}\ldots\ 10^q \\ &= \frac{0.0\ldots 0a_{t+1}a_{t+2}\ldots}{0.a_1a_2\ldots\ 10^q}|a|\ 10^q \\ &= \frac{0.a_{t+1}a_{t+2}\ldots}{0.a_1a_2\ldots}|a|\ 10^{-t}. \end{aligned}$$

Donnons une borne supérieure de ce rapport en majorant son numérateur et en minorant son dénominateur. Le numérateur est majoré par $0.999\ldots = 1$ et le dénominateur est minoré par $0.100\ldots$ puisque $a_1 \neq 0$. Par conséquent

$$|a - fl(a)| \leq 10|a|\ 10^q.\blacksquare$$

Remarquons que l'on peut écrire également l'inégalité donnée dans ce Théorème sous forme d'égalité

$$fl(a) = (1 + \varepsilon\ 10^{-t})a \quad \text{avec} \quad |\varepsilon| \leq K. \tag{1.1}$$

Remarque 1
Dans chaque mot de la mémoire, un certain nombre de chiffres sont réservés pour l'exposant. Par conséquent, celui–ci ne peut, en valeur absolue, dépasser une certaine valeur. Si l'exposant comporte trop de chiffres, il se produira ce que l'on appelle un dépassement de capacité. Pour être plus précis, on pourra parler de dépassement par défaut (underflow en anglais) lorsque le signe de l'exposant est négatif et de dépassement par excès (overflow en anglais) s'il est positif. Le cas du dépassement par excès donne lieu, en général, à un diagnostic. Dans le cas du dépassement par défaut certains ordinateurs donnent un diagnostic tandis que d'autres remplacent la valeur par zéro risquant ainsi de provoquer, par la suite, une division par zéro.

1.2 Opérations arithmétiques

Voyons maintenant comment un ordinateur s'y prend pour effectuer les quatre opérations arithmétiques élémentaires $+, -, \times, /$.

Soit à calculer $A = B + C$. Cette opération, comme les trois autres d'ailleurs, ne s'effectue pas dans la mémoire centrale mais dans ce que l'on appelle l'*accumulateur*. Cet accumulateur est un ensemble de trois mots dont la particularité est d'avoir des mantisses avec $2t$ chiffres au lieu de t. C'est pour cela que l'on parle souvent d'accumulateur en double précision (même si, dans la pratique, $2t$ chiffres ne sont pas nécessaires mais seulement $t + 1$ pour l'addition et la soustraction et $t + \log t$ pour la multiplication).

Pour effectuer l'opération $A = B + C$, l'ordinateur commence par recopier sans modification dans l'accumulateur celui des deux opérandes qui est le plus grand en valeur absolue. Comme, dans l'accumulateur, la mantisse doit avoir $2t$ chiffres, il la complète à droite par des zéros. Puis il recopie l'autre opérande dans l'accumulateur en faisant en sorte que son exposant soit le même que celui du premier opérande. Pour cela, on rajoute éventuellement des zéros à gauche de la mantisse (c'est–à–dire avant le chiffre a_1). Donnons, pour fixer les idées, un exemple avec $t = 8$. Supposons que $B = 0.23487757\ 10^3$ et que $C = 0.56799442$. Dans l'accumulateur on recopie B tel quel, c'est–à–dire que l'on aura $B = 0.2348775700000000\ 10^3$. Pour que C ait le même exposant que B, il faut le multiplier par 10^3. Mais il faut, bien sûr, le diviser aussi par 10^3 pour que sa valeur ne change pas, c'est–à–dire que $C = \frac{0.56799442}{10^3}\ 10^3$ et, dans l'accumulateur, on aura $C = 0.0005679944200000\ 10^3$. Finalement, cela montre que l'ordinateur procède comme nous quand nous effectuons une addition avec un papier et un crayon : nous plaçons les uns en dessous des autres les chiffres

correspondants aux puissances identiques de 10. Maintenant, l'addition peut s'effectuer dans l'accumulateur et l'on trouve $B+C = 0.2354455644200000\ 10^3$. On voit que, dans l'accumulateur, l'addition s'est effectuée sans aucune erreur (au moins sur cet exemple ; on verra plus loin d'autres exemples où le résultat obtenu dans l'accumulateur présente une erreur). Enfin, notre résultat doit être renvoyé dans un mot de la mémoire de l'ordinateur. Or ces mots ont des mantisses qui ne possèdent que t (8 dans notre exemple) chiffres. Nous allons donc faire une erreur qui est celle que l'on commet lorsque l'on place un nombre ayant une mantisse de plus de t chiffres dans un mot qui n'en accepte que t : c'est une erreur d'affectation telle qu'elle est donnée par le Théorème 1. Il est évident que les choses se passent de la même façon pour une soustraction. Dans l'exemple précédent, on obtiendra donc $A = 0.23544556\ 10^3$ que l'ordinateur procède par arrondi ou par troncature.

Dans le cas d'une multiplication, le produit de deux mantisses de longueur t donne un résultat de longueur $2t$ et les exposants s'ajoutent. Le résultat d'une multiplication est donc exact dans l'accumulateur et la seule erreur que l'on commet est, de nouveau, une erreur d'affectation en revenant de l'accumulateur dans la mémoire.

Pour une division enfin, il est évident que, dans l'accumulateur, le résultat n'est pas toujours exact (par exemple, lorsque l'on divise 1 par 3, le résultat possède une infinité de 3). L'erreur, dans l'accumulateur, se situe au niveau du $2t$-ième chiffre. En renvoyant ce résultat de l'accumulateur dans la mémoire, on commettra une erreur sur le t-ième chiffre, c'est–à–dire une erreur supérieure. Comme le résultat du Théorème 1 fournit une borne supérieure de l'erreur, il est donc toujours valable. Nous avons donc démontré que, pour toute opération arithmétique, l'erreur est donnée par le Théorème 1 et donc nous avons le

Théorème 2

$$|(B \star C) - fl(B \star C)| \leq K|B \star C|\ 10^{-t}$$

avec $K = 5$ dans le cas de l'arrondi, $K = 10$ dans celui de la troncature et où $\star$ est l'une des opérations $+, -, \times, /$.

1.3 Conséquences

Expliquons d'abord dans quel cas, dans l'accumulateur, une somme présente une erreur. Soit à calculer, avec $t = 8$, la somme $A = B + C$ avec $B = 0.56543451\ 10^6$ et $C = 0.21554623\ 10^{-4}$. Dans l'accumulateur on aura $B = 0.5654345100000000\ 10^6$. Pour que C ait le même exposant que B, il faut le multiplier et le diviser par 10^{10}. Lorsqu'on le divise par 10^{10} on perdra, dans l'accumulateur, 2 chiffres puisque celui–ci ne peut en contenir que 16. On aura donc $C = 0.0000000000215546\ 10^{-4}$. Par conséquent, même dans l'accumulateur, la somme sera erronée. Cependant, comme dans le cas d'une division,

l'erreur sera beaucoup plus importante en revenant de l'accumulateur dans la mémoire et, par conséquent, la majoration donnée par le Théorème 2 sera toujours valable.

Poussons maintenant ce raisonnement jusqu'à sa limite. Soit à calculer $A = B + C$ avec $B = 0.10000000\ 10^1$ et $C = 0.30000000\ 10^{-7}$. Dans l'accumulateur, on aura $C = 0.0000000030000000\ 10^1$ et $B + C = 0.1000000030000000\ 10^1$. En revenant dans la mémoire, on obtiendra $A = 0.10000000\ 10^1$, c'est–à–dire que $A = B$. Donc, sur ordinateur

$$fl(\mathbf{1} + \varepsilon) = \mathbf{1} \textbf{ pour } \varepsilon \textbf{ suffisamment petit.}$$

Ce nombre ε s'appelle la *précision machine* (exercice : quelle est la valeur maximale de ε dans le cas de l'arrondi et dans celui de la troncature ?).
Le même phénomène se produit, bien sûr, dans toute somme où les deux opérandes ont des ordres de grandeur très différents l'un de l'autre puisque l'on a, si $|B| > |C|$, $A = B + C = B(1 + \varepsilon)$ avec $\varepsilon = C/B$.

Remarque 2
La précision machine eps *peut être estimée à l'aide de l'algorithme suivant*

$$\begin{aligned} a &= 4./3. \\ b &= a - 1. \\ c &= b + b + b \\ eps &= abs(c - 1.) \end{aligned}$$

Voyons maintenant une seconde conséquence. Soient à calculer les deux quantités

$$\begin{aligned} v &= y + (x - x) \\ u &= (y + x) - x. \end{aligned}$$

Un ordinateur ne peut effectuer qu'une seule opération arithmétique à la fois. Les parenthèses dans les expressions précédentes indiquent laquelle des deux opérations doit être faite en premier. Supposons que $x = 1$ et que $y = \varepsilon$ avec ε tel que $fl(1 + \varepsilon) = 1$. On trouve que $fl(v) = \varepsilon$ ce qui est la bonne réponse, mais, par contre, on a $fl(u) = 0$. Par conséquent, sur ordinateur, l'addition n'est pas associative.

Quand une expression arithmétique comporte plusieurs opérations, celles–ci sont effectuées au fur et à mesure en partant de la gauche de l'expression et en respectant certaines règles de priorité (on effectue d'abord les élévations à une puissance, puis les multiplications et les divisions et enfin les additions et les soustractions). Réécrire une expression en changeant l'ordre des opérations revient donc à les associer de façon différente. Par conséquent, sur ordinateur, l'addition n'est pas commutative dans une somme de plus de deux termes.

En résumé

sur ordinateur, l'addition n'est ni associative ni commutative.

On pourrait penser, en voyant notre exemple précédent, que de telles erreurs ne sont pas très importantes. Effectuons maintenant les calculs suivants

$$\begin{aligned} v &= \frac{y+(x-x)}{y} = 1 \\ u &= \frac{(y+x)-x}{y} = 1 \end{aligned}$$

quel que soit $y \neq 0$. Prenons, de nouveau, $x = 1$ et $y = \varepsilon$. On obtient $fl(v) = 1$ qui est la réponse correcte, mais $fl(u) = 0$. On ne peut plus dire maintenant que l'erreur est faible ! En fait le phénomène est même plus compliqué que cela. Donnons à ε des valeurs au voisinage de la précision de l'ordinateur, plus précisément, $\varepsilon = 10^{-q}$ avec $q = 16.5 + k\ 0.01$ pour $k = 0, \ldots, 300$. On obtient les résultats de la Figure 1.1. On voit que, selon la valeur de q, $fl(u)$ varie dans l'intervalle $[0, 2]$.

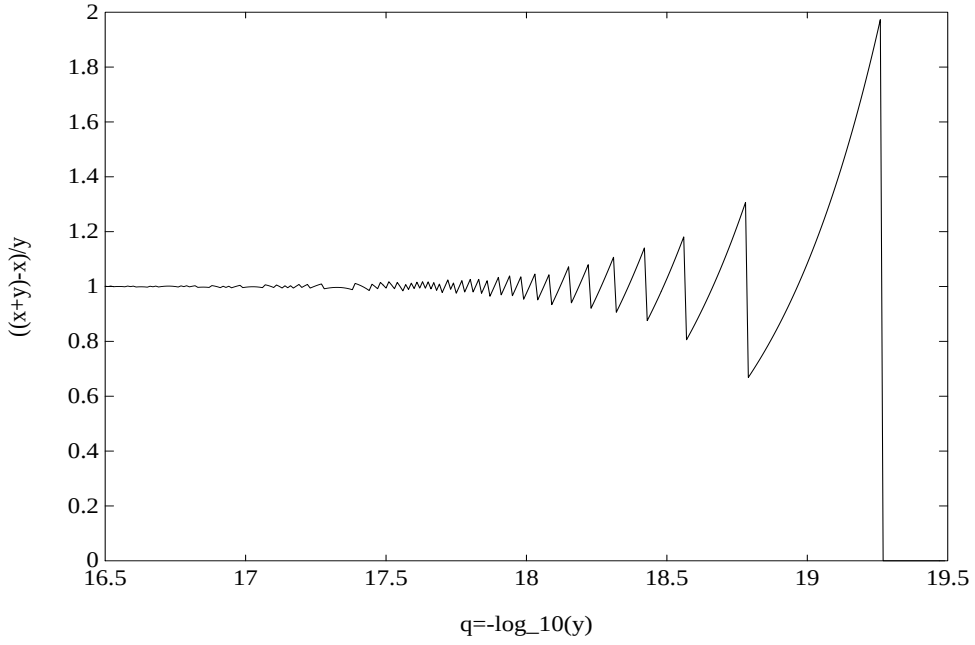


FIG. 1.1: Erreur de cancellation

Voyons maintenant la cause la plus importante d'erreurs. Soit à calculer $A = B - C$ avec $B = 0.56543287$ et $C = 0.56543286$. On trouve, dans l'accumulateur, $B - C = 0.0000000100000000$. En revenant dans la mémoire, puisque l'ordinateur n'admet que des nombres en virgule flottante normalisée, on aura $A = 0.10000000\ 10^{-7}$. Il n'y a aucune erreur, le résultat est entièrement juste ! Cependant, il faut bien comprendre que les zéros qui suivent le 1 dans le résultat n'ont absolument aucune signification. Ils proviennent des zéros par lesquels on

a complété à droite la mantisse des deux opérandes dans l'accumulateur. Ils n'ont pas plus de justification que n'importe quels autres chiffres. Si, maintenant, on utilise le résultat A dans des calculs ultérieurs, tout se passe comme si l'ordinateur ne travaillait plus qu'avec un seul chiffre décimal exact, c'est–à–dire avec $t = 1$. Ce genre d'erreur, qui se produit lorsque l'on effectue la différence de deux nombres voisins, s'appelle l'erreur de *cancellation*. Il faut donc éviter au maximum la possibilité que de telles erreurs se produisent. Par exemple, on n'utilisera pas la formule $1/x - 1/(x+1)$ mais on la remplacera par $1/x(x+1)$.

Donnons un exemple un peu plus compliqué. Soient à calculer les deux racines du polynôme $ax^2 + bx + c$. Elles sont données par les formules

$$\begin{aligned} x_1 &= \frac{-b - \sqrt{b^2 - 4ac}}{2a} \\ x_2 &= \frac{-b + \sqrt{b^2 - 4ac}}{2a}. \end{aligned}$$

Avec $a = 10^{-3}$, $b = 0.8$ et $c = -1.2\ 10^{-5}$, ces deux racines sont

$$\begin{aligned} x_1 &= -800 \\ x_2 &= 1.5\ 10^{-5}. \end{aligned}$$

La première racine, $\mathrm{fl}(x_1)$, est bien calculée tandis que, pour la seconde racine, on trouve $\mathrm{fl}(x_2) = 2.980232\ 10^{-5}$. Ce résultat est dû à la cancellation au numérateur de x_2. En effet, le discriminant est très voisin de b. La première racine est bien calculée car le signe qui est devant la racine carrée est le même que celui de $-b$ et il n'y a donc pas de cancellation. Celle-ci est, par contre, importante pour x_2. Par conséquent l'une des deux racines est toujours bien calculée et l'on peut utiliser pour cela la formule

$$x_1 = \frac{-b + s\sqrt{b^2 - 4ac}}{2a}$$

avec $s = -1$ si $b > 0$ et $s = 1$ si $b < 0$. Pour calculer l'autre racine, il faut se souvenir que leur produit est égal à c/a. Par conséquent, on utilisera la formule

$$x_2 = \frac{c}{ax_1}.$$

On a ainsi remplacé un calcul sujet à des erreurs de cancellation par de nouvelles expressions qui ne le sont pas. Donnons, dans la Figure 1.2, les résultats obtenus avec la bonne et la mauvaise formule pour calculer x_2. Les bons résultats sont représentés en tirets et les mauvais en traits pleins. Ils correspondent à $a = 10^{-3}$, $b = 0.8$ et $c = -(1 + k\ 0.05)\ 10^{-5}$ pour $k = 0, \ldots, 100$.

1.4 Conditionnement et stabilité

Dans cette Section, nous allons étudier deux notions fondamentales : le conditionnement d'un problème mathématique et la stabilité numérique d'un algorithme pour effectuer un certain calcul.

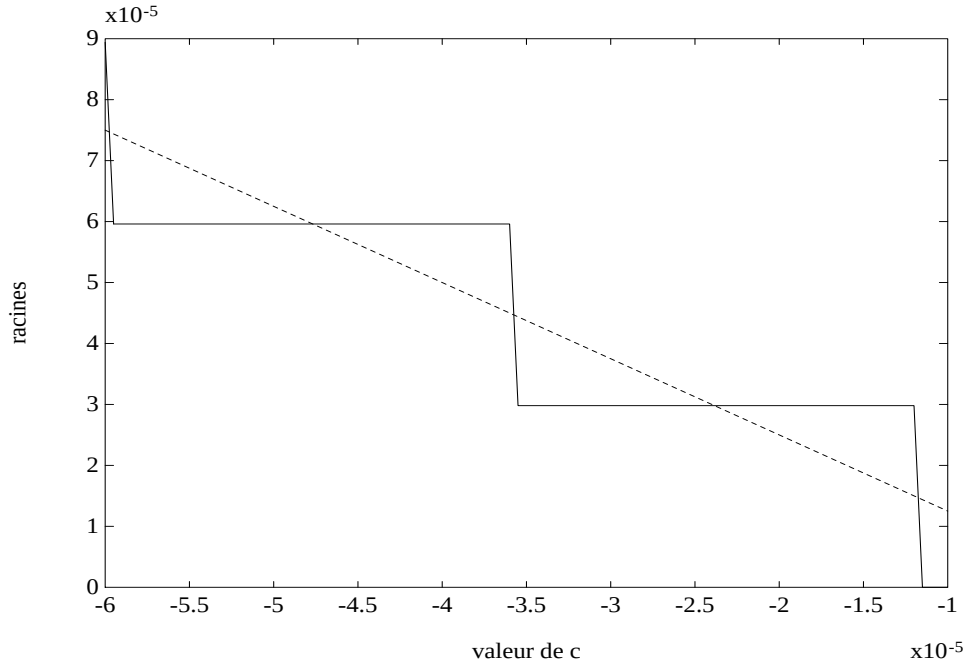


FIG. 1.2: Calcul des racines de $ax^2 + bx + c$

Lorsque l'on veut effectuer des calculs, la première opération consiste à introduire les données dans l'ordinateur. Or, d'après ce que nous avons dit plus haut, certaines données peuvent être entachées d'une erreur d'affectation (par exemple, 1/3 n'est pas représenté exactement en machine). Cela signifie que le problème que l'on va résoudre dans l'ordinateur n'est pas exactement celui que l'on voudrait traiter. Or, il se peut que la solution exacte de ce problème perturbé soit très éloignée de la solution exacte du problème non perturbé. On voit que cette notion est afférente au problème lui–même et est indépendante de l'algorithme que l'on va ensuite utiliser pour le résoudre ainsi que de la propagation, dans cet algorithme, des erreurs dues à l'arithmétique de l'ordinateur. Le problème mathématique peut être plus ou moins sensible à de petites variations sur les données. Ce phénomène s'appelle le *conditionnement* du problème. On dit qu'un problème est *bien* conditionné si une *petite* variation des données n'entraîne qu'une *petite* variation des résultats. Inversement, on dit qu'un problème est *mal* conditionné si une *petite* variation des données *peut* entraîner une *grande* variation des résultats. Les mots importants ont été écrits en italique. Les quantificateurs *petits* et *grands* dépendent bien sur de la précision de l'ordinateur. Si l'ordinateur travaille avec 7 chiffres, une variation relative de 10^{-4} sera considérée comme *grande* alors qu'elle sera *petite* si l'ordinateur travaille avec 18 chiffres. Le verbe *peut* signifie que pour certaines valeurs des données (mais pas nécessairement pour toutes) il y a une grande variation des résultats. Soit $\mathcal{P}$ un problème dont les données sont d et le résultat r. Supposons qu'une variation Δd des données entraîne une variation Δr du résultat. Le *nombre de condition* (aussi appelé plus simplement le *conditionnement*) $\mathrm{cond}(\mathcal{P})$

du problème $\mathcal{P}$ est une borne supérieure du facteur d'amplification des erreurs relatives, c'est–à–dire que

$$\frac{\|\Delta r\|}{\|r\|} \leq \mathrm{cond}(\mathcal{P}) \frac{\|\Delta d\|}{\|d\|}.$$

Par exemple, on peut définir le conditionnement relatif du calcul des racines de $ax^2 + bx + c$ par

$$\frac{|b| \cdot |x_i| + |c|}{|x_i| \cdot |2ax_i + b|}, \quad i = 1, 2.$$

Pour le polynôme étudié auparavant, on trouve un conditionnement voisin de 1 pour la racine x_1 et de 2 pour la racine x_2. On verra, dans le Chapitre 3, comment l'on peut définir le conditionnement pour la résolution d'un système d'équations linéaires.

Voyons maintenant la notion de stabilité numérique d'un algorithme. On utilise, pour effectuer les calculs, un certain algorithme. Les erreurs dues à l'arithmétique de l'ordinateur peuvent se propager plus ou moins dans cet algorithme. On voit que cette notion est attachée à l'algorithme utilisé et non pas au problème mathématique traité. Cette notion s'appelle *stabilité numérique* de l'algorithme. Par exemple, le premier des algorithmes précédents pour calculer les racines de ax^2+bx+c est numériquement instable (à cause de la cancellation qui peut s'y produire) tandis que le second est numériquement stable (car on a éliminé les soustractions qui pouvaient conduire à des erreurs de cancellation).

Naturellement, dans la pratique, ces deux phénomènes sont simultanément présents dans tout calcul. De plus, quand on ne sait pas résoudre exactement un problème, on en cherche une solution approchée par une méthode d'analyse numérique. Une *erreur de méthode* vient donc s'ajouter aux deux sources d'erreurs précédentes.

1.5 Remèdes

L'arithmétique de l'ordinateur étant ce qu'elle est, voyons comment s'en servir au mieux. Il faut d'abord éviter, autant que faire se peut, l'écueil de la cancellation. Ensuite, il est possible d'estimer ces erreurs par une méthode statistique (nous n'en parlerons pas ici) et d'en corriger certaines.

Voyons comment corriger une somme de n termes. Habituellement, une telle somme $S = a_1 + \cdots + a_n$ est calculée par

Chaque somme $S = S + a_i$ est entachée d'une erreur e_i que l'on peut calculer en refaisant les opérations dans le sens inverse. Naturellement, il faut éviter les erreurs dans ce calcul inverse. On somme ensuite les erreurs. Cette procédure est la suivante

$S = a_1$
for $i = 2, \ldots, n$ **calculer**
    $S = S + a_i$
end for i

$S = a_1$
$e = 0$
for $i = 2, \ldots, n$ **calculer**
    $V = S + a_i$
    if $|S| > |a_i|$ **alors** $g = -((V - S) - a_i)$
    sinon $g = -((V - a_i) - S)$
    $S = V$
    $e = e + g$
end for i
$S = S + e$

Soit, par exemple, à calculer

$$\sum_{i=1}^{n} \frac{i}{3} = \frac{n(n+1)}{6}.$$

Pour $n = 10^6$, cette somme vaut 1.6666683 10^{11}. Sans correction, on obtient 1.6670987 10^{11} tandis que, dans la somme corrigée, tous les chiffres sont exacts.

Il faut toutefois signaler que cette procédure ne permet pas de corriger l'erreur sur tous les ordinateurs, mais seulement sur ceux dont l'arithmétique vérifie certaines propriétés.

1.6 Un exemple

Pour terminer ce Chapitre, donnons un exemple complet depuis le problème mathématique lui–même jusqu'à l'interprétation des résultats numériques obtenus. Il a l'avantage de montrer toutes les étapes par lesquelles passe le travail d'un analyste numéricien.

Le problème mathématique que nous allons chercher à résoudre est celui du calcul de $\exp(x)$ pour une valeur négative de x avec une précision absolue de 10^{-8}.

Pour résoudre ce problème, la première idée qui vient à l'esprit est d'utiliser le développement en série de $\exp(x)$ qui est convergent pour tout x

$$\exp(x) = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \cdots$$

L'utilisation de ce développement n'est, pour l'instant, qu'une méthode théorique car, bien sûr, dans la pratique, on ne va pas sommer un nombre infini de termes. Mais, puisque x est négatif, nous avons affaire à une série alternée et l'on sait que, dès qu'un terme est plus petit que le précédent en valeur absolue, alors l'erreur est de l'ordre de grandeur du premier terme négligé. Nous avons donc là un test complètement fiable pour arrêter la sommation.

Cela donne lieu au pseudo–code suivant

$S = 1$
$t = 1$
$n = 1$
tant que $|t| > 10^{-8}$ **calculer**
    $t = t * x/n$
    $S = S + t$
    $n = n + 1$

Voici les résultats obtenus pour différentes valeurs de x. N est l'indice auquel la sommation s'est arrêtée.

x	N	S_N	$\exp(x)$
-5	26	6.738546E-03	6.737947E-03
-10	41	9.518019E-05	4.539993E-05
-15	55	4.336723E-03	3.059023E-07
-20	69	3.030620	2.061154E-09
-25	82	-247.347800	1.388794E-11
-30	96	-10193.520000	9.357623E-14
-35	110	4335985.000000	6.305117E-16
-40	123	-7.760598E+07	4.248354E-18
-45	137	5.556033E+10	2.862519E-20
-50	151	3.733105E+12	1.928750E-22
-55	164	-2.397606E+15	1.299581E-24
-60	178	-1.020143E+17	8.756511E-27
-65	191	2.133559E+19	5.900091E-29
-70	205	-2.068685E+19	3.975450E-31
-75	219	-4.898870E+23	2.678637E-33
-80	232	4.161912E+25	1.804851E-35
-85	246	-1.087440E+28	1.216099E-37
-90	259	-1.276238E+30	8.194009E-40

On voit que l'on est bien loin d'avoir obtenu la précision désirée !

Essayons d'analyser les résultats et de comprendre pourquoi il en est ainsi. Regardons, par exemple, ce qu'il se passe pour $x = -30$. Le premier terme de la

série vaut 1, le second -30, le troisième $(-30)^2/2! = 450$. En valeur absolue, les termes croissent ainsi jusqu'à $n = 30$ puis décroissent (pourquoi ?). Le terme le plus grand est $|x^{30}/30!| = 7.76\ 10^{11}$. Ainsi, pour obtenir un résultat qui vaut $9.35\ 10^{-14}$, nous sommes en train de faire une somme alternée de termes dont les plus grands valent 10^{11}. Pour obtenir un seul chiffre exact dans le résultat il faudrait donc travailler sur un ordinateur dont les mantisses auraient $11 + 14 = 25$ chiffres ! Le résultat obtenu est, on le voit, voisin de l'erreur absolue sur le terme le plus grand puisque $7.76\ 10^{11} \times\ 10^{-7}$ est du même ordre de grandeur que le résultat obtenu 10193. Autant dire que ce n'est pas de cette façon là que l'ordinateur calcule $\exp(x)$ pour des valeurs élevées de x ! Voyons comment l'on procède effectivement.

Le développement en série de l'exponentielle converge rapidement et sans trop d'erreurs d'arithmétique quand l'argument est voisin de 0 ou de 1. Lorsque ce n'est pas le cas, on effectue d'abord un changement de variable qui s'appelle une *réduction d'argument*. On peut utiliser, pour cela, la formule

$$e^x - 1 = (e^{x/2} + 1)(e^{x/2} - 1)$$

ou la formule

$$e^x = (e^{x/2^k})^{2^k}.$$

Pour cette seconde formule, on commence par calculer $x/2^k$, opération qui se ramène à des décalages en binaire. On calcule ensuite l'exponentielle par son développement en série et on l'élève ensuite k fois au carré.

Signalons que $\ln a$ peut se calculer à partir de l'exponentielle par des itérations de la méthode de Newton appliquée à l'équation $a - \exp(x) = 0$ dont la solution est $x = \ln a$

$$x_{k+1} = x_k + \frac{a - \exp(x_k)}{\exp(x_k)}.$$

On pourrait calculer $\ln a$ à l'aide du développement en série de $\ln(1 + x)$, mais la convergence est trop lente.

1.7 Exemples divers

Dans cette Section, nous allons donner d'autres exemples qui illustrent les problèmes posés par la précision finie de l'arithmétique des ordinateurs.

Considérons les itérations

$$\begin{gathered} x_0 \ \text{ donné} \\ y_0 = \sqrt{1 - x_0^2} \quad , \quad t_0 = x_0^2 + y_0^2 = 1 \\ \left.\begin{array}{l} x_{n+1} = 2x_n y_n \\ y_{n+1} = x_n^2 - y_n^2 \\ t_{n+1} = x_{n+1}^2 + y_{n+1}^2 \end{array}\right\} n = 0, 1, \ldots \end{gathered}$$

On a $t_{n+1} = (x_n^2 + y_n^2)^2 = t_n^2 = 1$ puisque $t_0 = 1$.

En programmant cet exemple, si l'on part de $x_0 = 0.6$, la suite (t_n) obtenue tend vers l'infini, alors qu'elle converge vers 0 pour $x_0 = 0.8$.

On pense souvent que, si un résultat calculé avec des précisions différentes reste invariant, alors il est obligatoirement bien calculé. Il n'en est malheureusement rien comme le montre l'exemple suivant. Soit à calculer la valeur de

$$333.75b^6 + a^2(11a^2b^2 - b^6 - 121b^4 - 2) + 5.5b^8 + \frac{a}{2b}.$$

Lorsque $a^2 - 1 = 11b^2/2$ cette expression vaut $-2 + a/2b$. Ainsi, pour $a = 77617.0$ et $b = 33096.0$, on obtient

- en simple précision 1.172603
- en double précision 1.1726039400531
- en précision étendue 1.172603940053178.

Cependant la valeur exacte est $-0.8273960599\ldots$! Les résultats dépendent aussi très fortement de l'ordinateur sur lequel ils ont été obtenus (et de la façon de programmer). Ainsi, sur un autre ordinateur, on a obtenu $-9.87\ 10^{29}$ en simple précision et $1.18\ 10^{21}$ en double précision ! Le résultat obtenu dépend donc souvent très fortement de l'ordinateur utilisé et, en particulier, de sa base de numération. C'est ainsi que le programme suivant permet d'obtenir la base de numération dans laquelle travaille l'ordinateur pour effectuer ses calculs internes

$a = 1.0$
$b = 1.0$
tant que $((a + 1.0) - a) - 1.0 = 0.0$, **calculer** $a = 2 * a$
tant que $((a + b) - a) - b \neq 0.0$, **calculer** $b = b + 1.0$
imprimer b.

Chapitre 2

Généralités

Ce Chapitre présente les outils mathématiques qui seront utilisés par la suite. Il peut être abordé sans aucune connaissance préalable sur les vecteurs et sur les matrices.

2.1 Définitions de base

Une *matrice* est un tableau, carré ou rectangulaire, de nombres. On place ce tableau entre deux parenthèses et on le désigne, en général, par une lettre majuscule. Par exemple

$$A = \begin{pmatrix} 3 & 1 & -\pi \\ 0 & \sqrt{2} & 125 \end{pmatrix}.$$

Une matrice est donc composée d'un certain nombre de *lignes* et de *colonnes*. Dans notre exemple, A a deux lignes et trois colonnes et l'on dit que cette matrice est de *dimension* 2×3. Les nombres qui la composent sont appelés *éléments* de la matrice.

Une matrice qui ne possède qu'une seule colonne s'appelle un *vecteur*. Un vecteur est, en général, désigné par une lettre minuscule. Par exemple

$$b = \begin{pmatrix} 3 \\ 0 \\ -\sin \pi/4 \end{pmatrix}.$$

On dit que ce vecteur est de dimension 3 car il possède trois lignes. Les nombres qui le forment sont appelés *composantes* du vecteur.

Pour plus de généralité, les composantes d'un vecteur sont désignées par des lettres possédant un *indice*. De même, les éléments d'une matrice sont des lettres avec deux indices : le premier indice correspond au numéro de la ligne

et le second à celui de la colonne

$$x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}, \quad A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{pmatrix}.$$

On écrit souvent $x = (x_i)$ et $A = (a_{ij})$. Pour indiquer la dimension d'un vecteur ou d'une matrice et le fait que ses éléments a_{ij} sont des nombres réels, on écrira, pour notre exemple, $x \in \mathbb{R}^3$ et $A \in \mathbb{R}^{2\times 3}$. Si les éléments a_{ij} sont des nombres complexes, on écrira $x \in \mathbb{C}^3$ et $A \in \mathbb{C}^{2\times 3}$. De façon plus générale, si les a_{ij} appartiennent à un *corps* $\mathbb{K}$, et si la matrice possède n lignes et m colonnes, on écrira $A \in \mathbb{K}^{n\times m}$. Une matrice dont le nombre de lignes n est égal au nombre de colonnes m s'appelle une matrice *carrée*. Dans ce cas là, on parle de matrice de dimension n. Si $n \neq m$, on parle de matrice *rectangulaire*. Dans une matrice carrée A de dimension n, les éléments $a_{11}, a_{22}, \ldots, a_{nn}$ forment la *diagonale* de A.

On appelle matrice *transposée* de $A = (a_{ij})$ la matrice, dénotée A^T, définie par $A^T = (a_{ji})$. On voit donc que chaque ligne i de A devient la colonne i de A^T et réciproquement. Par conséquent, si A est de dimension $n \times m$, A^T sera de dimension $m \times n$. Puisqu'un vecteur est une colonne, sa transposée est donc une ligne. On a, dans tous les cas, $(A^T)^T = A$.

Dans la suite, par souci de simplification et quand cette hypothèse ne restreindra pas la généralité, nous supposerons que les éléments des matrices et les composantes des vecteurs sont des nombres réels.

2.2 Addition et multiplication

Il est possible d'effectuer sur les matrices et sur les vecteurs des opérations algébriques similaires à celles que nous avons l'habitude d'effectuer sur des nombres. Il faut cependant se conformer à un certain nombre de règles car des différences existent. Nous allons commencer par apprendre à additionner et multiplier vecteurs et matrices.

Pour l'addition, c'est facile. D'abord on ne peut sommer que des objets (matrices ou vecteurs) dont les dimensions sont les mêmes. On additionne deux (ou plusieurs) matrices en effectuant l'addition des éléments situés aux mêmes emplacements dans les matrices. Ainsi

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{pmatrix} + \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \end{pmatrix} = \begin{pmatrix} a_{11}+b_{11} & a_{12}+b_{12} & a_{13}+b_{13} \\ a_{21}+b_{21} & a_{22}+b_{22} & a_{23}+b_{23} \end{pmatrix}.$$

Pour les vecteurs, c'est la même chose. Le vecteur $c = a + b$ est donné par

$$c = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} a_1+b_1 \\ a_2+b_2 \\ a_3+b_3 \end{pmatrix}.$$

On a donc la formule générale

$$c_i = a_i + b_i, \quad i = 1, \ldots, n.$$

L'élément neutre pour l'addition des matrices, c'est–à–dire la matrice qui additionnée à une autre ne la modifie pas est la matrice nulle dont tous les éléments sont égaux à zéro. Il en est de même pour les vecteurs (qui sont des matrices particulières).

L'addition de matrices et de vecteurs est commutative et associative, c'est–à–dire que

$$A + B = B + A \quad \text{et} \quad (A + B) + C = A + (B + C).$$

Étudions maintenant comment s'effectue le produit d'une matrice par un vecteur. Pour qu'un tel produit soit possible, il faut que le nombre de colonnes de la matrice soit égal à la dimension du vecteur, c'est–à–dire que le produit Ab n'est possible que si $A \in \mathbb{R}^{n\times m}$ et $b \in \mathbb{R}^m$. Le résultat c du produit Ab est un vecteur de $\mathbb{R}^n$. Pour obtenir la première composante c_1 du vecteur c, on ne considère que la première ligne de la matrice A. Pour calculer c_1, on multiplie le premier élément de la première ligne de A (c'est–à–dire a_{11}) par la première composante de b (qui est b_1). À ce produit, on ajoute le produit du second élément de la première ligne de A (c'est–à–dire a_{12}) par la seconde composante de b (qui est b_2). Et ainsi de suite jusqu'à la fin de le première ligne de A (c'est–à–dire jusqu'à m). Pour calculer la seconde composante c_2 du produit Ab, on procède exactement de la même manière mais en effectuant les produits avec la seconde ligne de A. Et ainsi de suite jusqu'à c_n. On a donc finalement

$$c = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix} = Ab = \begin{pmatrix} a_{11}b_1 + a_{12}b_2 + \cdots + a_{1m}b_m \\ a_{21}b_1 + a_{22}b_2 + \cdots + a_{2m}b_m \\ \vdots \\ a_{n1}b_1 + a_{n2}b_2 + \cdots + a_{nm}b_m \end{pmatrix}.$$

De façon générale, on peut donc écrire

$$c_i = \sum_{j=1}^{m} a_{ij}b_j, \quad i = 1, \ldots, n. \tag{2.1}$$

Donnons un exemple

$$\begin{pmatrix} 3 & 1 & -2 \\ 0 & 3 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ -1 \\ 4 \end{pmatrix} = \begin{pmatrix} 3 \times 1 + 1 \times (-1) + (-2) \times 4 \\ 0 \times 1 + 3 \times (-1) + 1 \times 4 \end{pmatrix} = \begin{pmatrix} -6 \\ 1 \end{pmatrix}.$$

Le produit d'une matrice par un vecteur est distributif par rapport à l'addition, ce qui signifie que $A(b + c) = Ab + Ac$ où A est une matrice, b et c des vecteurs. De même, si B est aussi un matrice, on a $(A + B)b = Ab + Bb$.

Le produit de deux matrices $C = AB$ n'est rien d'autre qu'une suite de produits matrice–vecteur en considérant successivement toutes les colonnes de la seconde matrice B. D'après ce qui a été dit plus haut concernant les dimensions pour un produit matrice–vecteur, on voit que, pour pouvoir effectuer ce produit, il faut que le nombre de colonnes de la matrice A soit égal au nombre de lignes de B. Le résultat C est une matrice ayant le même nombre de lignes que A et le même nombre de colonnes que B. On a donc $A \in \mathbb{R}^{n\times m}, B \in \mathbb{R}^{m\times q}$ et $C \in \mathbb{R}^{n\times q}$. D'après ce que nous venons d'expliquer et en utilisant (2.1), on voit que $C = (c_{ij})$ est donnée par les formules

$$c_{ij} = \sum_{k=1}^{m} a_{ik} b_{kj}, \quad i = 1, \ldots, n; \quad j = 1, \ldots, q.$$

Pour faciliter le calcul, on peut le disposer de la façon suivante

$$\begin{array}{cccc||cccc} & & & & b_{11} & b_{12} & \cdots & b_{1q} \\ & & & & \vdots & \vdots & & \vdots \\ & & & & b_{m1} & b_{m2} & \cdots & b_{mq} \\ \hline\hline a_{11} & a_{12} & \cdots & a_{1m} & c_{11} & c_{12} & \cdots & c_{1q} \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} & c_{n1} & c_{n2} & \cdots & c_{nq} \end{array}$$

Dans ce tableau, chaque élément c_{ij} est ainsi le produit scalaire (voir la définition plus loin) du vecteur formé par la ligne i de A situé à sa gauche par celui formé par la colonne j de B qui se trouve au dessus de lui.

Si $q = n$, le produit BA existe et c'est une matrice $m \times m$ alors que le produit AB est une matrice $n \times n$. On voit donc que, si $n \neq m$, le produit de deux matrices ne peut certainement pas être commutatif. Mais, même si A et B sont deux matrices carrées $n \times n$, en général, $AB \neq BA$. Si $AB = BA$ on dit que les deux matrices *commutent*.

Donnons un exemple

$$\begin{pmatrix} 1 & 2 \\ 2 & -1 \end{pmatrix} \begin{pmatrix} 4 & 0 \\ 1 & 3 \end{pmatrix} = \begin{pmatrix} 6 & 6 \\ 7 & -3 \end{pmatrix}, \quad \begin{pmatrix} 4 & 0 \\ 1 & 3 \end{pmatrix} \begin{pmatrix} 1 & 2 \\ 2 & -1 \end{pmatrix} = \begin{pmatrix} 4 & 8 \\ 7 & -1 \end{pmatrix}.$$

On pourra vérifier facilement que $(AB)^T = B^T A^T$.

Lorsque l'on ne considère que des matrices carrées (c'est–à–dire dont le nombre de lignes est égal au nombre de colonnes), l'élément neutre pour la multiplication est la matrice qui ne comporte que des 1 sur sa diagonale et des 0 ailleurs. On l'appelle la *matrice identité* et on la désigne par I. On a donc $AI = IA = A$ pour toute matrice carrée A.

Le produit de matrices est associatif et distributif par rapport à l'addition, c'est–à–dire que $(AB)C = A(BC)$ et que $A(B + C) = AB + AC$ où A, B et C sont des matrices telles que les produits aient un sens.

Soient $u = (u_1, \ldots, u_n)^T$ et $v = (v_1, \ldots, v_n)$ deux vecteurs de $\mathbb{C}^n$. Le produit *produit scalaire hermitien* des vecteurs u et v est noté par (u, v) et il est défini par

$$(u, v) = u^T \bar{v} = \sum_{i=1}^{n} u_i \bar{v}_i.$$

C'est un nombre qui vérifie les propriétés suivantes

1. $(u, v) = \overline{(v, u)}$,
2. $(au + bw, v) = a(u, v) + b(w, v), \quad w \in \mathbb{C}^n, a, b \in \mathbb{C}$,
3. $(u, av + bw) = \bar{a}(u, v) + \bar{b}(u, w), \quad w \in \mathbb{C}^n, a, b \in \mathbb{C}$,
4. $(u, u) \geq 0$,
5. $(u, u) = 0$ si et seulement si $u = 0$.
6. $(u, Av) = (A^* u, v)$ où A^* est la matrice adjointe de A, c'est–à–dire sa transposée conjuguée.

Si u et v sont deux vecteurs réels de dimension n, le produit $u^T v$ est un nombre réel et l'on parle alors de *produit scalaire*. On a $(u, Av) = (A^T u, v)$ et, de façon plus générale, $(Au, Bv) = (B^T Au, v) = (u, A^T Bv)$.

Remarque 3
Il faut faire attention que $u^T v$ est un nombre, alors que uv^T est une matrice.

2.3 Inversion

Soit a un nombre réel. L'inverse de a est le nombre a^{-1} tel que $aa^{-1} = a^{-1}a = 1$ et il est donné par $a^{-1} = 1/a$. La division d'un nombre par un autre est équivalent au produit du premier nombre par l'inverse du second, $a/b = ab^{-1} = b^{-1}a$. Par conséquent, la division de deux nombres est commutative et elle provient de la définition de l'inverse d'un nombre. Pour les matrices, on ne parle pas de division, mais seulement d'inverse et de produit par l'inverse. Nous savons déjà que le produit de deux matrices n'est pas commutatif. Nous allons donc maintenant définir l'inverse d'une matrice carrée.

Soit A une matrice carrée. On appelle, s'il existe, *inverse* de A la matrice notée A^{-1} telle que

$$AA^{-1} = A^{-1}A = I.$$

Une matrice qui possède un inverse s'appelle *inversible* ou *régulière*. Une matrice qui n'a pas d'inverse se dénomme *singulière*. L'inverse n'est défini que pour les matrices carrées ; pour les matrices rectangulaires, ou pour les matrices carrées singulières, on parle de *pseudo–inverse*.

Pour savoir si une matrice est régulière, il est d'abord nécessaire de définir la notion de *déterminant* d'une matrice. Soit A une matrice carrée de dimension 2

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}.$$

Son déterminant est le nombre

$$\det A = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{12}a_{21}.$$

On voit que pour dénoter un déterminant, on remplace les parenthèses par des barres verticales ou l'on utilise la notation det.

On va maintenant définir récursivement le déterminant des matrices carrées de dimensions supérieures. On considère le déterminant d'une matrice de dimension 3

$$\det A = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}.$$

Pour calculer sa valeur, on va le *développer* par rapport à sa première ligne (on pourrait aussi le développer par rapport à sa première colonne). Le déterminant de A est calculé en multipliant a_{11} par le déterminant de la matrice 2×2 obtenue en supprimant la première ligne et la première colonne de A, puis en soustrayant à cette valeur le produit de a_{12} par le déterminant de la matrice 2×2 obtenue en supprimant la première ligne et la seconde colonne de A et, enfin, en ajoutant à ce résultat le produit de a_{13} par le déterminant de la matrice 2×2 obtenue en supprimant le première ligne et la troisième colonne de A. On a donc

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11} \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} - a_{12} \begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + a_{13} \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix}.$$

Nous savons donc maintenant calculer le déterminant d'une matrice de dimension 3. Le déterminant d'une matrice A de dimension n se calcule de la même manière à partir de déterminants de dimension $n - 1$.

Définition 1
Les déterminants des matrices carrées obtenues en supprimant certaines lignes et certaines colonnes de A s'appellent les mineurs *de A. Les mineurs obtenus en ne gardant que les k premières lignes et colonnes d'une matrice s'appellent les* mineurs fondamentaux.

Nous noterons m_{ij}, le mineur obtenu en supprimant la ligne i et la colonne j de A. On a

$$\det A = \sum_{j=1}^{n} (-1)^{j-1} a_{1j} m_{1j}$$

et, de façon générale,

$$\begin{aligned}\det A &= \sum_{j=1}^{n}(-1)^{i+j-2}a_i j m_{ij}, \quad i=1,\ldots,n,\\ \det A &= \sum_{i=1}^{n}(-1)^{i+j-2}a_i j m_{ij}, \quad j=1,\ldots,n.\end{aligned}$$

Disons tout de suite que le calcul numérique d'un déterminant n'est pas, en général, réalisable en pratique. Il nécessite en effet de l'ordre de $n \cdot n!$ multiplications, nombre qui devient très rapidement extrêmement grand avec n comme nous le verrons dans le Chapitre 3.

Soit, par exemple

$$A = \begin{pmatrix} 2 & 1 & 3 \\ 2 & 1 & 5 \\ 3 & 1 & a \end{pmatrix}.$$

On a

$$m_{11} = \begin{vmatrix} 1 & 5 \\ 1 & a \end{vmatrix} = a-5, \quad m_{12} = \begin{vmatrix} 2 & 5 \\ 3 & a \end{vmatrix} = 2a-15, \quad m_{13} = \begin{vmatrix} 2 & 1 \\ 3 & 1 \end{vmatrix} = -1,$$

$$m_{21} = \begin{vmatrix} 1 & 3 \\ 1 & a \end{vmatrix} = a-3, \quad m_{22} = \begin{vmatrix} 2 & 3 \\ 3 & a \end{vmatrix} = 2a-9, \quad m_{23} = \begin{vmatrix} 2 & 1 \\ 3 & 1 \end{vmatrix} = -1,$$

$$m_{31} = \begin{vmatrix} 1 & 3 \\ 1 & 5 \end{vmatrix} = 2, \qquad m_{32} = \begin{vmatrix} 2 & 3 \\ 2 & 5 \end{vmatrix} = 4, \qquad m_{33} = \begin{vmatrix} 2 & 1 \\ 2 & 1 \end{vmatrix} = 0.$$

Développons le déterminant de A par rapport à sa seconde ligne. Nous obtenons

$$\begin{aligned}\det A &= -a_{21}m_{21} + a_{22}m_{22} - a_{23}m_{23}\\ &= -2(a-3) + 1(2a-9) - 5(-1)\\ &= 2.\end{aligned}$$

Notons a_i le vecteur formé par la colonne i de la matrice A. Avec cette notation celle–ci sera notée $A = [a_1, \ldots, a_n]$. Le déterminant vérifie les propriétés suivantes (λ est un nombre réel ou complexe)

$$\begin{aligned}&\det A = \det A^T\\ &\det AB = \det BA = \det A \cdot \det B\\ &\det A \cdot \det A^{-1} = 1\\ &\det B^{-1}AB = \det A\\ &\det \lambda A = \lambda^n \det A\\ &\det[\lambda a_1, \ldots, a_n] = \lambda \det[a_1, \ldots, a_n]\\ &\det[a_1, a_2 + \lambda a_1, \ldots, a_n] = \det[a_1, \ldots, a_n]\\ &\det[a_2, a_1, \ldots, a_n] = -\det[a_1, \ldots, a_n].\end{aligned}$$

Pour le produit d'une colonne par un nombre et la combinaison avec une autre colonne, la propriété reste vraie quelques soient les colonnes impliquées. On peut également effectuer ces opérations sur les lignes au lieu des colonnes puisque le déterminant d'une matrice est égal au déterminant de sa transposée.

Définition 2
On appelle rang d'une matrice la dimension de son plus grand mineur non nul.

Au sujet de l'inversion d'une matrice, on a le résultat fondamental suivant

Théorème 3
Une condition nécessaire et suffisante pour qu'une matrice soit régulière est que son déterminant ne soit pas nul, c'est–à–dire que son rang soit égal à sa dimension.

Donc une matrice est singulière si et seulement si son déterminant est nul.

Voyons maintenant comment calculer l'inverse d'une matrice A. Ce calcul fait intervenir les mineurs de A. On a

$$A^{-1} = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{pmatrix}^T \Big/ \det A$$

avec $c_{ij} = (-1)^{i+j} m_{ij}$. Ces c_{ij} s'appellent des *cofacteurs*. Le signe par lequel on doit multiplier le mineur pour obtenir le cofacteur est donné par le tableau suivant

$$\begin{array}{ccccc} + & - & + & - & \cdots \\ - & + & - & + & \cdots \\ + & - & + & - & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{array}$$

Si nous reprenons notre exemple précédent, nous obtenons

$$A^{-1} = \begin{pmatrix} (a-5)/2 & -(a-3)/2 & 1 \\ -(2a-15)/2 & (2a-9)/2 & -2 \\ -1/2 & 1/2 & 0 \end{pmatrix}.$$

L'inverse d'une matrice possède les propriétés suivantes

$$\begin{aligned} (A^{-1})^{-1} &= A \\ (AB)^{-1} &= B^{-1}A^{-1} \\ (A^{-1})^T &= (A^T)^{-1}. \end{aligned}$$

2.4 Matrices particulières

Dans cette Section, nous allons passer en revue certaines matrices particulières qui interviennent en algèbre matricielle.

Matrices diagonales et triangulaires

Une matrice A telle que $a_{ij} = 0$ si $i \neq j$ s'appelle une matrice *diagonale*. Son déterminant vaut $a_{11}a_{22}\cdots a_{nn}$. Son inverse est la matrice diagonale dont les éléments diagonaux sont $1/a_{ii}$. A^{-1} n'existe donc que si $a_{ii} \neq 0$ pour $i = 1, \ldots, n$.

Une matrice A telle que $a_{ij} = 0$ pour $i > j$ s'appelle une matrice *triangulaire supérieure*. Son déterminant est égal à $a_{11}a_{22}\cdots a_{nn}$. Son inverse est une matrice triangulaire supérieure. Le produit de deux matrices triangulaires supérieures est une matrice triangulaire supérieure.

Une matrice A telle que $a_{ij} = 0$ pour $i < j$ s'appelle une matrice *triangulaire inférieure*. Son déterminant est égal à $a_{11}a_{22}\cdots a_{nn}$. Son inverse est une matrice triangulaire inférieure. Le produit de deux matrices triangulaires inférieures est une matrice triangulaire inférieure.

Matrice orthogonale

On dit que A est une matrice *orthogonale* si $A^{-1} = A^T$.

Matrice symétrique

Une matrice est dite *symétrique* si $A^T = A$. Donc

$$a_{ij} = a_{ji}, \quad \forall i \text{ et } \forall j.$$

Matrice conjuguée

Soit A une matrice dont les éléments appartiennent à $\mathbb{C}$. B est la *conjuguée* de A si $b_{ij} = \bar{a}_{ij}, \forall i$ et $\forall j$. On notera $B = \bar{A}$. On a

$$\overline{(\bar{A})} = A, \qquad \overline{(BA)} = \bar{B}\bar{A}, \qquad (\bar{A})^{-1} = \overline{(A^{-1})}.$$

Si A est une matrice réelle alors $A = \bar{A}$.

Matrice adjointe

Soit A une matrice dont les éléments appartiennent à $\mathbb{C}$. B est l'*adjointe* de A si

$$b_{ij} = \bar{a}_{ji}, \quad \forall i \text{ et } \forall j.$$

B est donc la transposée conjuguée de A. On notera $B = A^* = (\bar{A})^T = \overline{(A^T)}$. On a

$$(A^*)^* = A, \qquad (AB)^* = B^* A^*.$$

Matrice hermitienne

Une matrice A est dite *hermitienne* si $A = A^*$. On a $a_{ij} = \bar{a}_{ji}$. Ses éléments diagonaux sont des nombres réels.

Matrice unitaire

Une matrice A est dite *unitaire* si $A^* = A^{-1}$.

Matrice idempotente

Une matrice A est dite *idempotente* si $A^2 = A$.

Matrice définie positive

Une matrice A est *définie positive* si $x^T A x > 0, \forall x \neq 0$.

Cette notion étant un peu délicate à appréhender, donnons un exemple. Soit la matrice

$$A = \begin{pmatrix} 1 & 1 \\ 1 & 5 \end{pmatrix}.$$

Soit $x = (x_1, x_2)^T$ un vecteur quelconque non nul. On a

$$Ax = \begin{pmatrix} x_1 + x_2 \\ x_1 + 5x_2 \end{pmatrix}$$

et $(x, Ax) = x_1(x_1 + x_2) + x_2(x_1 + 5x_2) = (x_1 + x_2)^2 + 4x_2^2$ qui est toujours strictement positif quels que soient x_1 et x_2 non nuls simultanément.

Matrice de permutation

Une matrice de permutation est une matrice dont tous les éléments sont nuls sauf un et un seul égal à 1 dans chaque ligne et dans chaque colonne. Une telle matrice s'obtient en permutant lignes et colonnes de la matrice identité.

Quand on multiplie une matrice A à gauche par une matrice de permutation P, on permute ses lignes. Si $p_{ij} = 1$, on obtient PA en permutant les lignes i et j de A.

Quand la matrice de permutation P est placée à droite, on permute les colonnes. Si $p_{ij} = 1$, on obtient AP en permutant les colonnes i et j de A.

Soit

$$P = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}, \quad A = \begin{pmatrix} 0 & 1 & 2 \\ 3 & 4 & 5 \\ 6 & 7 & 8 \end{pmatrix}.$$

On obtient

$$PA = \begin{pmatrix} 3 & 4 & 5 \\ 6 & 7 & 8 \\ 0 & 1 & 2 \end{pmatrix}, \quad AP = \begin{pmatrix} 2 & 0 & 1 \\ 5 & 3 & 4 \\ 8 & 6 & 7 \end{pmatrix}.$$

2.5 Un peu d'algèbre linéaire

Dans cette Section, nous allons interpréter les résultats précédents en terme d'algèbre linéaire. Cette Section peut être laissée de côté en première lecture si l'on ne s'intéresse pas à la théorie. Comme nous l'avons vu, il est possible de manipuler les matrices sans savoir ce qu'elles représentent ni connaître les notions théoriques qui s'y rattachent.

Soient x et y deux vecteurs de $\mathbb{C}^n$, α et β deux nombres complexes et A une matrice de dimension $m \times n$. Le vecteur Ax est un vecteur de $\mathbb{C}^m$. La multiplication par A a donc transformé un vecteur de $\mathbb{C}^n$ en un vecteur de $\mathbb{C}^m$. On dit que A représente une *application* de $\mathbb{C}^n$ dans $\mathbb{C}^m$.

D'après les règles de la multiplication que nous avons étudiées plus haut nous avons

$$A(\alpha x + \beta y) = \alpha Ax + \beta Ay$$

ce qui montre que l'application représentée par la matrice A est une application *linéaire*.

Nous allons maintenant étudier le côté théorique cette notion d'application linéaire.

2.5.1 Les vecteurs

Soient $x_1, x_2, \ldots, x_n$, des nombres réels (on peut généraliser facilement ce qui suit au cas de nombres complexes). L'ensemble $\mathbb{R}^n$ formé par les éléments

$$x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$$

est noté $\mathbb{R}^n$. Un tel élément x s'appelle un *vecteur* et les nombres $x_1, \ldots, x_n$ sont ses *composantes*. Si ces nombres sont complexes, l'ensemble de ces vecteurs est désigné par $\mathbb{C}^n$. On remarquera qu'un vecteur est écrit sous forme d'une colonne de nombres.

On munit $\mathbb{R}^n$ d'une loi interne d'addition (si x et $y \in \mathbb{R}^n$, les composantes du vecteur $x+y$ sont $x_i + y_i$) et d'une loi externe de multiplication par un scalaire (si $x \in \mathbb{R}^n$ et $\lambda \in \mathbb{R}$, les composantes du vecteur λx sont λx_i). Cette addition est associative et commutative, elle possède un élément neutre (le vecteur 0 dont toutes les composantes sont nulles) et tout vecteur x admet un symétrique, noté $-x$, tel que $x + (-x) = 0 \in \mathbb{R}^n$. La loi de multiplication est distributive par rapport à l'addition, elle est associative et il existe un élément unité, le scalaire 1, tel que $1 \cdot x = x$.

Alors, avec ces deux lois, $\mathbb{R}^n$ est un *espace vectoriel* sur $\mathbb{R}$. Soient $x_1, \ldots, x_n$ des vecteurs de $\mathbb{R}^n$ (à ne pas confondre avec les composantes du vecteur x qui étaient désignées auparavant par les mêmes lettres). On dit que ces vecteurs sont *linéairement indépendants* s'il n'existe pas de nombres réels $a_1, \ldots, a_n$ non tous nuls tels que

$$a_1 x_1 + \cdots + a_n x_n = 0.$$

En d'autres termes si le fait que cette somme soit nulle implique que, nécessairement, les scalaires $a_1, \ldots, a_n$ soient tous nuls, alors on dit que les vecteurs $x_1, \ldots, x_n$ sont linéairement indépendants. Dans le cas contraire, ces vecteurs sont dits *linéairement dépendants* et il existe alors des scalaires non tous nuls tels que $a_1 x_1 + \cdots + a_n x_n = 0$. Dans un espace de dimension n, il ne peut y avoir plus de n vecteurs linéairement indépendants.

Soient $x_1, \ldots, x_n$ des vecteurs linéairement indépendants dans un espace vectoriel E. Si, pour tout $x \in E$, les vecteurs $x, x_1, \ldots, x_n$ sont linéairement dépendants alors on dit que $x_1, \ldots, x_n$ forment une *base* de E et l'entier n s'appelle la dimension de E. De plus, tout $x \in E$ peut s'écrire de façon unique sous forme d'une combinaison linéaire

$$x = a_1 x_1 + \cdots + a_n x_n$$

où les a_i sont des scalaires. $\mathbb{R}^n$ est un espace de dimension n.

2.5.2 Les applications linéaires

On dit que l'on a défini une *application* f de $\mathbb{R}^n$ dans $\mathbb{R}^m$ si, à tout élément $x \in \mathbb{R}^n$, on fait correspondre un et un seul élément de $\mathbb{R}^m$, noté $f(x)$. On dit que $f(x)$ est l'*image* de x par f.

L'application f est *linéaire* si elle satisfait les conditions

$$\begin{aligned} f(x+y) &= f(x)+f(y), \quad \forall x, y \in \mathbb{R}^n \\ f(\lambda x) &= \lambda f(x), \quad \forall \lambda \in \mathbb{R}, \forall x \in \mathbb{R}^n. \end{aligned}$$

2.5.3 Les matrices

Considérons les n vecteurs de $\mathbb{R}^n$, e_i, pour $i = 1, \ldots, n$, dont toutes les composantes valent 0 sauf la i-ème qui est égale à 1. Tout vecteur $x \in \mathbb{R}^n$, de composantes $x_1, \ldots, x_n$, peut donc s'écrire

$$x = x_1 e_1 + \cdots + x_n e_n.$$

On dit que les vecteurs e_i forment une *base* de $\mathbb{R}^n$; c'est la *base canonique*.

Si f est une application linéaire de $\mathbb{R}^n$ dans $\mathbb{R}^m$ alors

$$f(x) = x_1 f(e_1) + \cdots + x_n f(e_n).$$

Par conséquent, la connaissance de $f(e_1), \ldots, f(e_n)$ détermine parfaitement l'application linéaire f. En effet, si les vecteurs $v_i = f(e_i) \in \mathbb{R}^m, i = 1, \ldots, n$, sont connus, alors $f(x) = x_1 v_1 + \cdots + x_n v_n$. Inversement, si l'on se donne n vecteurs $v_i \in \mathbb{R}^m$, on vérifie que l'application f définie par

$$f : x \in \mathbb{R}^n \longmapsto x_1 v_1 + \cdots + x_n v_n \in \mathbb{R}^m$$

est bien une application linéaire de $\mathbb{R}^n$ dans $\mathbb{R}^m$. Toute application linéaire est donc entièrement déterminée par la donnée des images des vecteurs $e_1, \ldots, e_n$.

Soit a_{ij} la $i$ème composante du vecteur $f(e_j)$. Nous allons placer ces nombres dans un tableau à double entrée, noté A, où le premier indice désigne la ligne et le second la colonne

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}.$$

Un tel tableau s'appelle une *matrice* et les nombres a_{ij} ses *éléments*. La première colonne de A est le vecteur $f(e_1)$, la seconde colonne contient le vecteur $f(e_2)$ et ainsi de suite jusqu'à la dernière colonne qui n'est rien d'autre que $f(e_n)$. Si $m \neq n$, on parle de matrice *rectangulaire* et, si $m = n$, de matrice *carrée*.

Comme il a été dit plus haut, cette matrice détermine complètement l'application linéaire f qu'elle représente. Dans la suite, nous confondrons la matrice A avec l'application linéaire f qu'elle détermine. Lorsque l'on aura besoin de montrer que les éléments de A sont notés a_{ij}, on écrira

$$A = (a_{ij}).$$

Le vecteur $y = f(x)$, dont les composantes sont $y_1, \ldots, y_m$, s'exprime en fonction des composantes $x_1, \ldots, x_n$ du vecteur x à l'aide des relations

$$\begin{aligned} y_1 &= a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \\ y_2 &= a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \\ &\vdots \\ y_m &= a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \end{aligned}$$

ce qui peut s'écrire

$$y = Ax$$

en définissant la i-ème composante $(Ax)_i$ du vecteur Ax par

$$(Ax)_i = a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n.$$

L'ensemble de ces m relations définit donc le produit d'une matrice à m lignes et n colonnes (on dit une matrice $m \times n$) par un vecteur de $\mathbb{R}^n$. Il y a donc équivalence entre l'écriture $y = f(x)$ et l'écriture $y = Ax$ où A est la matrice qui représente l'application linéaire f.

Le rang d'une matrice est égal au nombre maximum de ses lignes (ou de ses colonnes) qui forment des vecteurs linéairement indépendants.

2.5.4 Opérations sur les matrices

Étudions maintenant les opérations sur les matrices. Elles se définissent à partir des opérations correspondantes sur les applications linéaires qu'elles représentent.

Soient f et g deux applications linéaires et soit h l'application définie par

$$h(x) = f(x) + g(x).$$

À partir de la linéarité de f et g, il est facile de voir que $h(x+y) = h(x)+h(y)$ et que $h(\lambda x) = \lambda h(x)$. L'application h est donc linéaire. Si f et g sont représentées respectivement par les matrices $A = (a_{ij})$ et $B = (B_{ij})$, alors h se représente par une matrice $C = (c_{ij})$ donnée par

$$h(e_i) = f(e_i) + g(e_i), \quad i = 1, \ldots, n$$

c'est–à–dire que

$$c_{ij} = a_{ij} + b_{ij}, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n.$$

Nous avons donc défini l'addition de deux matrices et nous écrirons

$$C = A + B.$$

Cette addition est associative et commutative. Il existe un élément neutre pour l'addition qui est la matrice 0 dont tous les éléments sont nuls. Toute matrice A possède une matrice symétrique, notée $-A$, telle que $A + (-A) = 0$. Par conséquent, l'addition définit sur l'ensemble des matrices $m \times n$ une structure de *groupe abélien.*

Soit λ un scalaire. L'application $h(x) = \lambda f(x)$ est, de façon évidente, linéaire et la matrice C correspondant à h est donnée par

$$c_{ij} = \lambda a_{ij}, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n$$

c'est–à–dire que

$$C = \lambda A.$$

Cette multiplication par un scalaire est distributive par rapport à l'addition, elle est associative et il existe un élément unité, le scalaire $\lambda = 1$, tel que $1 \cdot A = A$.

Par conséquent, l'ensemble des matrices $m \times n$ muni de l'addition et de la multiplication par un scalaire est un espace vectoriel sur $\mathbb{R}$. Il est habituellement noté $\mathbb{R}^{m \times n}$.

Nous allons maintenant définir le produit de deux matrices à partir de la composition de deux applications linéaires. Soient f une application de $\mathbb{R}^n$ dans $\mathbb{R}^m$ (représentée par la matrice $A \in \mathbb{R}^{m \times n}$) et g un application de $\mathbb{R}^m$ dans $\mathbb{R}^p$ (représentée par la matrice $B \in \mathbb{R}^{p \times m}$). Définissons l'application h par

$$h(x) = g(f(x))$$

que l'on notera $h = g \circ f$.

Il est facile de voir que h est une application linéaire de de $\mathbb{R}^n$ dans $\mathbb{R}^p$. Cette application correspond à la matrice $C = (c_{ij}) \in \mathbb{R}^{p \times n}$. Nous avons

$$\begin{aligned} h(e_j) &= c_{1j}e''_1 + \cdots + c_{pj}e''_p \\ f(e_j) &= a_{1j}e'_1 + \cdots + a_{mj}e'_m \\ g(e'_j) &= b_{1j}e''_1 + \cdots + b_{pj}e''_p \end{aligned}$$

où $e'_1, \ldots, e'_m$ est la base canonique de $\mathbb{R}^m$ et $e''_1, \ldots, e''_p$ celle de $\mathbb{R}^p$. D'où

$$\begin{aligned} h(e_j) &= g(a_{1j}e'_1 + \cdots + a_{mj}e'_m) \\ &= a_{1j}g(e'_1) + \cdots + a_{mj}g(e'_m) \\ &= a_{1j}(b_{11}e''_1 + \ldots + b_{p1}e''_p) + \cdots + a_{mj}(b_{1m}e''_1 + \ldots + b_{pm}e''_p) \\ &= (b_{11}a_{1j} + \cdots + b_{1m}a_{mj})e''_1 + \cdots + (b_{p1}a_{1j} + \cdots + b_{pm}a_{mj})e''_p. \end{aligned}$$

Par conséquent, on a finalement

$$c_{ij} = \sum_{k=1}^{m} b_{ik}a_{kj}, \quad i = 1, \ldots, p, \quad j = 1, \ldots, n.$$

Nous venons donc de définir le produit de deux matrices et l'on écrira

$$C = BA.$$

On notera que le produit de deux matrices n'est possible que si le nombre de colonnes de la première matrice est égal au nombre de lignes de la seconde. Ce produit est associatif mais, en général, il n'est pas commutatif (même dans le cas de matrices carrées). Le produit est distributif, à droite et à gauche, par rapport à l'addition.

Soit I la matrice carrée $I = (\delta_{ij})$ avec $\delta_{ij} = 0$ si $i \neq j$ et $\delta_{ij} = 1$ si $i = j$ (symbole de Kronecker). Cette matrice s'appelle la *matrice identité* et l'on a

$$IA = AI = A.$$

L'ensemble $\mathbb{R}^{n\times n}$ des matrices carrées forme donc une *algèbre unitaire* puisque c'est un espace vectoriel, un anneau et qu'il existe un élément unité pour la multiplication.

Soit A une matrice $n \times n$ dont les colonnes sont linéairement indépendantes. Alors, il existe une matrice, notée A^{-1}, telle que

$$AA^{-1} = A^{-1}A = I.$$

La matrice A^{-1} s'appelle l'*inverse* de la matrice A. On dira dans ce cas que A est *inversible* ou *régulière*. Une matrice qui n'est pas inversible est appeléee *non inversible* ou *singulière*.

2.5.5 Changement de base

Soit f une application linéaire de $\mathbb{R}^n$ dans lui–même. La matrice A qui lui correspond dépend de la base choisie. La question à laquelle nous allons maintenant nous intéresser est de savoir comment la matrice A se transforme lorsque l'on change de base.

Soit $u_1, \ldots, u_n$ l'ancienne base et $v_1, \ldots, v_n$ la nouvelle. La nouvelle base peut s'écrire en fonction de l'ancienne

$$v_j = t_{1j}u_1 + \cdots + t_{nj}u_n, \quad j = 1, \ldots, n.$$

Puisque les vecteurs v_j forment une base, les lignes de la matrice $T = (t_{ij})$ sont linéairement indépendantes et la matrice T est donc inversible.

Soit $x \in \mathbb{R}^n$. On a

$$\begin{aligned} x &= x_1u_1 + \cdots + x_nu_n \\ &= y_1v_1 + \cdots + y_nv_n \\ &= y_1(t_{11}u_1 + \cdots + t_{n1}u_n) + \cdots + y_n(t_{1n}u_1 + \cdots + t_{nn}u_n) \\ &= (t_{11}y_1 + \cdots + t_{1n}y_n)u_1 + \cdots + (t_{n1}y_1 + \cdots + t_{nn}y_n)u_n. \end{aligned}$$

Ce qui donne finalement

$$x_i = t_{i1}y_1 + \cdots + t_{in}y_n, \quad i = 1, \ldots, n$$

ce qui peut s'écrire

$$x = Ty$$

où x et y sont les vecteurs de composantes $x_1, \ldots, x_n$ et $y_1, \ldots, y_n$ respectivement. Posons $z = f(x)$ ou, ce qui revient au même comme nous l'avons vu, $z = Ax$. Le vecteur z est exprimé dans l'ancienne base. Soit z' son expression dans la nouvelle base. Nous avons

$$z' = T^{-1}z = T^{-1}Ax.$$

Dans la nouvelle base, x devient x' ce qui donne

$$z' = T^{-1}z = T^{-1}ATx'.$$

L'application linéaire f' qui fait passer de x' à z' est maintenant exprimée complètement dans la nouvelle base et l'on voit qu'elle correspond à la matrice

$$B = T^{-1}AT.$$

La matrice T s'appelle la *matrice de changement de base*. On dit également que les matrices A et B sont semblables.

2.6 Vecteurs propres et valeurs propres

Nous allons maintenant faire un rappel des principaux résultats reliés aux éléments propres d'une matrice. Nous n'en donnerons pas les démonstrations.

Soit A une matrice carrée $n \times n$ dont les éléments a_{ij} sont des nombres complexes.

Définition 3
Le nombre complexe λ est dit valeur propre *de A et x* vecteur propre *de A associé à la valeur propre λ si*

$$Ax = \lambda x, \quad x \neq 0.$$

On voit que les vecteurs propres ne sont déterminés qu'à une constante multiplicative près.

D'après la définition, on a donc $(A - \lambda I)x = 0$. Si $\det(A - \lambda I) \neq 0$, alors $x = 0$. Puisque nous avons imposé, dans la définition d'un vecteur propre, que x soit non nul, ce déterminant est nécessairement nul. Les valeurs propres de A sont solution de l'équation polynômiale

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

En utilisant les règles qui permettent de calculer la valeur d'un déterminant on voit que ce déterminant est un polynôme de degré n en λ. Il s'appelle le *polynôme caractéristique* de la matrice A. On le désignera par P_n et l'on a

$$P_n(\lambda) = \det(A - \lambda I) = (-1)^n \lambda^n + \cdots + \det(A).$$

Les valeurs propres de A sont donc les racines de ce polynôme. Une valeur propre λ est dite valeur propre simple (ou de multiplicité égale à 1) de A si λ est racine simple de du polynôme caractéristique $P_n(\lambda) = \det(A - \lambda I) = 0$. Dans le cas contraire elle est dite de multiplicité k. La somme de toutes les multiplicités est égale à n.

On rappelle les propriétés suivantes

Propriété 1

1. *Si λ est valeur propre de A, alors $\bar{\lambda}$ est valeur propre de A^*,*
2. *une matrice et sa transposée ont mêmes valeurs propres,*
3. *si toutes les valeurs propres sont distinctes, les vecteurs propres forment une base.*

On a le résultat suivant qui s'appelle le Théorème de Cayley–Hamilton

Théorème 4

$$P_n(A) = 0.$$

Bien évidemment, $P_n(A)$ est une matrice et 0 est la matrice nulle.

Supposons que 0 soit valeur propre de A. Alors d'après la définition d'une valeur propre on doit avoir $Ax = 0$ avec $x \neq 0$. Par conséquent, $\lambda = 0$ est valeur propre de A si et seulement si A est singulière.

D'autre part, d'après les relations de Newton qui relient coefficients et racines d'un polynôme,

$$\det A = \lambda_1 \cdots \lambda_n.$$

Nous avons les deux résultats suivants.

Théorème 5
Soient respectivement x_i et y_i les vecteurs propres de A et de A^*.

$$(x_i, y_j) = 0 \quad si \quad \lambda_i \neq \lambda_j.$$

Démonstration.

Nous avons

$$\begin{aligned} Ax_i &= \lambda_i x_i \\ A^* y_j &= \bar{\lambda}_j y_j. \end{aligned}$$

D'où

$$\begin{aligned} (y_j, Ax_i) &= \bar{\lambda}_i (y_j, x_i) \\ (A^* y_j, x_i) &= \bar{\lambda}_j (y_j, x_i) = (u_j, Ax_i). \end{aligned}$$

D'où, en soustrayant

$$0 = (\bar{\lambda}_i - \bar{\lambda}_j)(y_j, x_i)$$

ce qui démontre le résultat si $\lambda_i \neq \lambda_j$. ∎

Théorème 6
Les valeurs propres d'une matrice hermitienne sont réelles.

Démonstration.

Soit λ une valeur propre de A et x le vecteur propre correspondant. On a

$$\begin{aligned} (Ax, x) &= \lambda (x, x) \\ (x, Ax) &= \bar{\lambda} (x, x) \\ &= (A^* x, x) = (Ax, x) \end{aligned}$$

puisque $A = A^*$. Donc, en soustrayant, $(\lambda - \bar{\lambda})(x, x) = 0$. Or $(x, x) \neq 0$ et, par conséquent, $\lambda = \bar{\lambda}$. ∎

Si la matrice est symétrique définie positive, non seulement ses valeurs propres sont réelles, mais elles sont strictement positives.

Propriété 2
Soit P une matrice inversible quelconque. $B = P^{-1}AP$ et A ont mêmes valeurs propres. On dit que les matrices A et B sont semblables*. Les vecteurs propres de A sont égaux à P fois ceux de B.*

Définition 4
On dit que la matrice A est diagonalisable *s'il existe une matrice non singulière P telle que $B = P^{-1}AP$ soit une matrice diagonale. Dans ce cas, sur la diagonale de B, se trouvent les valeurs propres de A et les colonnes de P sont les vecteurs propres de A.*

Propriété 3
Quelque soit la matrice A, il existe une matrice inversible P telle que $B = P^{-1}AP$ soit de la forme de Jordan

$$B = \begin{pmatrix} J_1 & & \\ & \ddots & \\ & & J_k \end{pmatrix}$$

où J_i est de la forme

$$J_i = \begin{pmatrix} \lambda_i & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_i \end{pmatrix}$$

et de dimension n_i. λ_i est valeur propre de A et $\sum_{i=1}^{k} n_i = n$.

Définition 5
La quantité $tr(A) = \sum_{i=1}^{n} a_{ii}$ est appelée trace *de A.*

Propriété 4
$tr(A) = \sum_{i=1}^{n} \lambda_i$ *et* $tr(A^k) = \sum_{i=1}^{n} \lambda_i^k$.

Enfin, on a le résultat suivant

Théorème 7

$$\det A = \lambda_1 \cdots \lambda_n.$$

Par conséquent, une matrice est singulière si et seulement si au moins l'une de ses valeurs propres est nulle.

2.7 La notion de norme

Rappelons d'abord ce qu'est la notion de valeur absolue d'un nombre réel. Soit x un nombre réel. On lui associe sa *valeur absolue* qui est un nombre réel positif ou nul, noté $|x|$, vérifiant les trois propriétés

$$\begin{aligned} &|x| \geq 0 \text{ et } |x| = 0 \text{ si et seulement si } x = 0, \\ &|\alpha x| = |\alpha| \cdot |x|, \quad \alpha \in \mathbb{R}, \\ &|x + y| \leq |x| + |y|. \end{aligned}$$

Le nombre $|x|$ peut être défini par $|x| = sx$ où s est le signe de x tel que $s = +1$ si $x \geq 0$ et $s = -1$ si $x \leq 0$.

La valeur absolue d'un nombre sert à mesurer sa proximité de zéro. La quantité $|x - y|$ représente ainsi la distance entre les nombres x et y. Si $x = a + ib$ est un nombre complexe, alors $|x| = (|a| + |b|)^{1/2}$ est son module. Les trois propriétés précédents sont vérifiées en définissant $|x|$ de cette manière, mais elles le seraient encore en prenant $|x| = |a| + |b|$. Ceci montre que la définition n'est pas unique.

De façon similaire, la notion de norme va associer à un vecteur ou à une matrice un nombre positif ou nul vérifiant les mêmes propriétés.

2.7.1 Normes de vecteurs

Soit x (et y) un vecteur. On lui associe un nombre réel positif ou nul, noté $\|x\|$, vérifiant les propriétés suivantes

$$\begin{array}{l} \|x\| \geq 0 \text{ et } \|x\| = 0 \text{ si et seulement si } x = 0, \\ \|\alpha x\| = |\alpha| \cdot \|x\|, \quad \alpha \in \mathbb{C}, \\ \|x + y\| \leq \|x\| + \|y\|. \end{array}$$

Tout nombre $\|x\|$ vérifiant ces trois propriétés s'appelle une *norme* du vecteur x. La norme d'un vecteur n'est pas définie de manière unique et, étant donné un vecteur, on peut lui associer plusieurs normes. Les normes les plus utilisées sont les *normes de Hölder*. Pour les distinguer les unes des autres, nous leur mettrons un indice. Elles sont définies par

$$\begin{aligned} \|x\|_k &= \left(|x_1|^k + \cdots + |x_n|^k\right)^{1/k}, \quad k = 1, 2, \ldots, \\ \|x\|_\infty &= \max_{1 \leq i \leq n} |x_i|, \end{aligned}$$

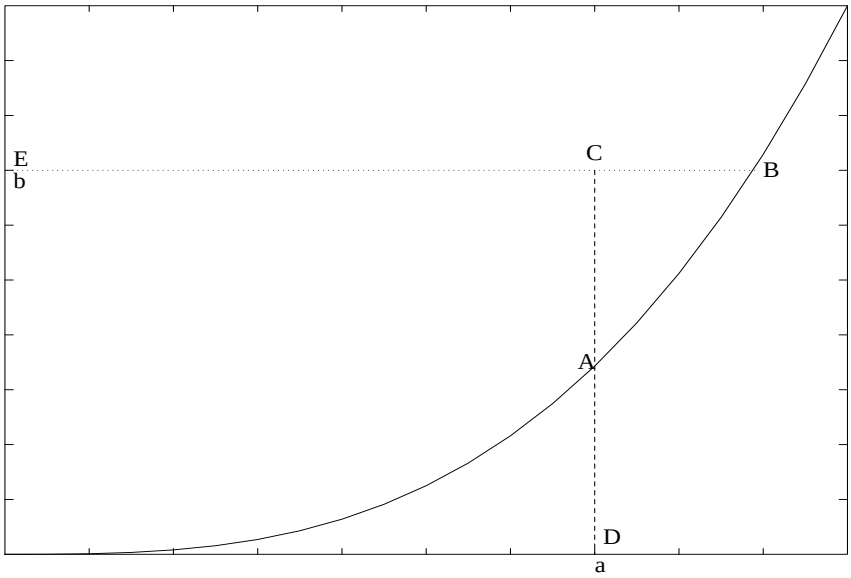
où les x_i sont les composantes du vecteur x.

Parmi ces normes, les plus utilisées sont $\|x\|_1$, $\|x\|_\infty$ et $\|x\|_2$ qui représente la longueur du vecteur x au sens euclidien du terme (c'est le théorème de Pythagore).

Pour les normes de Hölder on peut démontrer les inégalités suivantes

Théorème 8

$$\begin{array}{l} \left|\displaystyle\sum_{i=1}^{k} x_i y_i\right| \leq \|x\|_p \|y\|_q, \quad p > 1, 1/p + 1/q = 1, \\ n^{-1/p}\|x\|_p \leq n^{-1/q}\|x\|_q, \quad q > p, \\ \|x\|_q \leq \|x\|_p, \quad q > p. \end{array}$$



Démonstration.

Considérons la fonction u définie par

$$u(t) = t^{p-1}, \quad t \geq 0 \quad \text{et} \quad p > 1.$$

Elle est représentée sur la Figure.

On a aire$(OAD) = a^p/p$ et aire$(OBE) = b^q/q$ avec $q = p/(p-1)$ ou encore $1/p + 1/q = 1$ puisque $t = u^{q-1}$. On a aire$(ODCE) = ab$ et par conséquent

$$ab \leq \frac{a^p}{p} + \frac{b^q}{q}.$$

On pose

$$a = \frac{|x_i|}{\|x\|_p} \quad \text{et} \quad b = \frac{|y_i|}{\|y\|_q}$$

ce qui donne

$$\frac{|x_i y_i|}{\|x\|_p \|y\|_q} \leq \frac{|x_i|^p}{p\sum_{i=1}^{k} |x_i|^p} + \frac{|y_i|^q}{q\sum_{i=1}^{k} |y_i|^q}.$$

Ajoutons ces inégalités pour $i = 1, \ldots, k$; il vient

$$\frac{\sum_{i=1}^{k} |x_i y_i|}{\|x\|_p \|y\|_q} \leq \frac{1}{p} + \frac{1}{q} = 1$$

ce qui démontre la première inégalité du Théorème puisque la valeur absolue (ou le module) d'une somme est inférieure ou égale à la somme des valeurs absolues (ou des modules).

Prenons maintenant $q > p$ et posons $p = q/k$ avec $k > 1$. On a

$$\frac{1}{n}|x_i|^p = \frac{1}{n}|x_i|^{q/k} = \left(\frac{1}{n}|x_i|^q\right)^{1/k}\left(\frac{1}{n}\right)^{1-1/k}.$$

Posons $1/k' = 1 - 1/k$, $u_i = |x_i|^q/n$ et $v_i = 1/n$. Nous obtenons

$$\frac{1}{n}|x_i|^p = u_i^{1/k} \cdot v_i^{1/k'}.$$

L'inégalité précédente devient

$$\begin{aligned}\sum_{i=1}^n u_i^{1/k} v_i^{1/k'} &\leq \left(\sum_{i=1}^n u_i^{k/k}\right)^{1/k}\left(\sum_{i=1}^n v_i^{k'/k'}\right)^{1/k'} \\ &= \left(\sum_{i=1}^n u_i\right)^{1/k}\left(\sum_{i=1}^n \frac{1}{n}\right)^{1/k'} \\ &= \left(\sum_{i=1}^n u_i\right)^{1/k} = \left(\frac{1}{n}\sum_{i=1}^n |x_i|^q\right)^{p/q}.\end{aligned}$$

D'où l'inégalité

$$\sum_{i=1}^n |x_i|^p n^{-1/k} n^{-1/k'} \leq n^{-p/q}\left(\sum_{i=1}^n |x_i|^q\right)^{p/q},$$

soit encore

$$n^{-1}\sum_{i=1}^n |x_i|^p \leq n^{-p/q}\left(\sum_{i=1}^n |x_i|^q\right)^{p/q}.$$

D'où la seconde inégalité du Théorème en prenant la racine p-ième des deux membres de cette inégalité.

Prenons maintenant $y_i = |x_i|/\|x\|_p$. On a $y_i \geq 0$ et $\sum_{i=1}^n y_i^p = 1$. Par conséquent $y_i \leq 1$ pour tout i. Si $q > p$ cela implique donc que $y_i^q \leq y_i^p$; d'où

$$\sum_{i=1}^n y_i^q \leq \sum_{i=1}^n y_i^p = 1$$

$$\sum_{i=1}^n y_i^q = \frac{\sum_{i=1}^n |x_i|^q}{\|x\|_p^q} \leq 1$$

ce qui termine la démonstration. ∎

On a également

$$\begin{aligned}
&\|x\|_\infty \leq \|x\|_2 \leq \|x\|_1 \\
&\|x\|_2 \leq \|x\|_1 \leq \sqrt{n}\|x\|_2 \\
&\|x\|_\infty \leq \|x\|_2 \leq \sqrt{n}\|x\|_\infty \\
&\|x\|_\infty \leq \|x\|_1 \leq n\|x\|_\infty.
\end{aligned}$$

2.7.2 Normes de matrices

Les normes pour les matrices peuvent être définies à partir des normes de Hölder pour les vecteurs (mais ça n'est pas obligatoire). Soit A une matrice. La quantité

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

s'appelle *norme* (de Hölder) de la matrice A. C'est la norme de l'opérateur représenté par la matrice A. Puisque l'on peut, dans la définition précédente, remplacer x par αx on a donc également

$$\|A\|_k = \sup_{\|x\|=1} \|Ax\|_k.$$

Ces normes de Hölder vérifient les trois propriétés des normes

$$\begin{aligned}
&\|A\|_k \geq 0 \text{ et } \|A\|_k = 0 \text{ si et seulement si } A = 0, \\
&\|\alpha A\|_k = |\alpha| \cdot \|A\|_k, \quad \alpha \in \mathbb{R}, \\
&\|A + B\|_k \leq \|A\|_k + \|B\|_k.
\end{aligned}$$

Ces normes de matrices vérifient, en plus, deux autres propriétés qui nous serons très utiles par la suite. D'après sa définition, $\|A\|_k$ est une borne supérieure. Par conséquent, pour tout vecteur x, on a

$$\|Ax\|_k \leq \|A\|_k \cdot \|x\|_k.$$

D'autre part, on a, en posant $y = Bx$,

$$\|AB\|_k = \sup_{x \neq 0} \frac{\|A(Bx)\|_k}{\|Bx\|_k} \frac{\|Bx\|_k}{\|x\|_k} \leq \sup_{y \neq 0} \frac{\|Ay\|_k}{\|y\|_k} \cdot \sup_{x \neq 0} \frac{\|Bx\|_k}{\|x\|_k}.$$

Par conséquent

$$\|AB\|_k \leq \|A\|_k \cdot \|B\|_k.$$

On appelle *multiplicative* toute norme vérifiant cette inégalité.

Les normes précédentes semblent être difficiles à calculer en pratique puisqu'elles font intervenir une borne supérieure. Cependant, on connaît leurs expressions exactes dans trois cas (A est une matrice $n \times m$)

$$\|A\|_1 \quad = \quad \max_{1 \leq j \leq m} \sum_{i=1}^{n} |a_{ij}|,$$

$$\begin{aligned} \|A\|_\infty &= \max_{1\le i\le n} \sum_{j=1}^{m} |a_{ij}|, \\ \|A\|_2 &= \sqrt{\rho(AA^T)} \end{aligned}$$

où $\rho(AA^T)$ désigne le rayon spectral de la matrice AA^T, c'est–à–dire sa plus grande valeur propre puisque celles–ci sont réelles et positives. On démontre que, quelque soit la norme

$$\varrho(A) \le \|A\|.$$

Remarque 4
Si A est symétrique, $\|A\|_2 = \varrho(A)$.

On utilise aussi parfois la *norme de Frobenius* car elle est facile à calculer. Elle est définie par

$$\|A\|_F = \left(\sum_{i,j=1}^{n} |a_{ij}|^2 \right)^{1/2}.$$

Pour cette norme, on a également

$$\|AB\|_F \le \|A\|_F \cdot \|B\|_F.$$

Parmi toutes ces normes, laquelle faut-il utiliser ? Pour les vecteurs, la norme $\|x\|_2$ est certainement la plus concrète puisqu'elle représente une longueur au sens euclidien, géométrique, du terme. Mais la norme matricielle $\|A\|_2$ qui lui est associée est difficile à calculer en pratique puisqu'elle ne s'exprime pas directement à l'aide des termes de la matrice. En fait, la norme utilisée n'a que peu d'importance car on démontre qu'elles sont toutes *équivalentes*, c'est–à–dire qu'elles sont reliées entre elles par des inégalités. En particulier, on a

$$\begin{aligned} &\|A\|_2 \le \|A\|_F \le \sqrt{n}\|A\|_2, \\ &\|A\|_1/\sqrt{n} \le \|A\|_2 \le \sqrt{n}\|A\|_1 \\ &\|A\|_\infty/\sqrt{n} \le \|A\|_2 \le \sqrt{n}\|A\|_\infty \\ &\|A\|_2^2 \le \|A\|_1 \cdot \|A\|_\infty. \end{aligned}$$

Pour simplifier les notations, nous supprimerons, à partir de maintenant, l'indice dans les normes de Hölder avec la convention que, dans toute égalité ou inégalité, c'est partout le même indice qui est sous–entendu.

2.7.3 Le conditionnement

Pour toute norme de Hölder, on a, en prenant $B = A$, $\|AA^{-1}\| = \|I\| \le \|A\|\cdot\|A^{-1}\|$. Or, la norme de la matrice identité est égale à 1 d'après la définition. Il s'en suit que l'on a l'inégalité (l'indice k est sous–entendu)

$$\kappa(A) = \|A\| \cdot \|A^{-1}\| \ge 1.$$

Ce nombre $\kappa(A)$ s'appelle le *conditionnement* de A. Son importance est fondamentale. Nous l'avions déjà évoqué dans la Section 1.4 et nous le reverrons par la suite. On dit que la matrice A est *bien conditionnée* si $\kappa(A)$ est "voisin" de 1. Si $\kappa(A)$ est "grand" par rapport à 1, on dit que A est *mal conditionnée*. Naturellement les adjectifs "voisin" et "grand" sont subjectifs. Si la précision de l'ordinateur avec lequel on travaille est de 10^{-7} un conditionnement de 10^5 sera considéré comme "grand". Par contre, si la précision est de 10^{-16}, un tel conditionnement sera "petit". Ces considérations s'éclaireront quand on verra, dans la Section 3.2, que le conditionnement est, en fait, le facteur d'amplification des erreurs sur les données.

Remarque 5
On fait souvent l'erreur de croire qu'une matrice dont le déterminant est très voisin de zéro est mal conditionnée et cela parce qu'elle est proche d'une matrice singulière. Il n'en est rien. En effet, considérons la matrice diagonale dont tous les termes sont égaux à ε. Son déterminant vaut ε^n alors que son conditionnement est égal à 1.

2.8 La décomposition en valeurs singulières

Un outil particulièrement utile pour l'analyse matricielle est la *décomposition en valeurs singulières*. Nous commencerons par la présenter en détail dans le cas des matrices carrées régulières. Puis nous donnerons les résultas principaux du cas rectangulaire et nous les utiliserons pour la résolution des sytèmes d'équations linéaires au sens des moindres carrés.

2.8.1 Matrices carrées

Le résultat principal est le

Théorème 9
Soit A une matrice régulière $p \times p$. Il existe des nombres réels $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p > 0$ et deux bases orthonormales $u_1, \ldots, u_p$ et $v_1, \ldots, v_p$ de $\mathbb{R}^p$ tels que, pour $i = 1, \ldots, p$

$$Av_i = \sigma_i u_i \quad et \quad A^T u_i = \sigma_i v_i.$$

Démonstration.

On a $A^T A v_i = \sigma_i A^T u_i = \sigma_i^2 v_i$ et $AA^T u_i = \sigma_i A v_i = \sigma_i^2 u_i$ ce qui montre que les nombres σ_i^2 sont les valeurs propres (qui doivent être positives) de $A^T A$ et, bien sur, également de AA^T, que les u_i sont les vecteurs propres correspondants de AA^T et que les v_i sont ceux de $A^T A$. On sait également que ces deux ensembles de vecteurs propres sont orthonormaux (c'est–à–dire $(u_i, u_j) = (v_i, v_j) = \delta_{ij}$,

symbole de Kronecker). Enfin $(Av_i, Av_i) = \sigma_i^2 > 0$ et l'on peut choisir le signe positif pour σ_i. ∎

Soit $U = [u_1, \ldots, u_p]$ la matrice dont les colonnes sont $u_1, \ldots, u_p$, $V = [v_1, \ldots, v_p]$ et $\Sigma = \text{diag}(\sigma_1, \ldots, \sigma_p)$. On voit facilement que les relations du Théorème 9 peuvent s'écrire

$$A = U\Sigma V^T.$$

C'est ce que l'on appelle la *décomposition en valeurs singulières*. Nous la désignerons par SVD qui provient de l'anglais *singular value decomposition*. Les nombres σ_i sont appelés *valeurs singulières* de la matrice A. La SVD d'une matrice est unique.

On voit que

$$\sigma_i = (u_i, Av_i) = (Av_i, Av_i)^{1/2} = (A^T u_i, A^T u_i)^{1/2}.$$

On a également

$$(u_i, Av_j) = 0 \quad \text{pour} \quad i \neq j.$$

La SVD permet d'obtenir des expressions de A et de A^{-1}. Ainsi, nous avons le

Théorème 10

$$A = \sum_{i=1}^{p} \sigma_i u_i v_i^T \quad et \quad A^{-1} = \sum_{i=1}^{p} \frac{1}{\sigma_i} v_i u_i^T.$$

Démonstration.

Posons $B = \sum_{i=1}^{p} \sigma_i u_i v_i^T$. Pour montrer que $B = A$ il suffit de montrer que $Bv_i = Av_i$ pour $i = 1, \ldots, p$ puisque les v_i forment une base. On a

$$Bv_i = \sum_{j=1}^{p} \sigma_j u_j (v_j, v_i) = \sigma_i u_i = Av_i.$$

La SVD de A^{-1} est donnée par

$$A^{-1} = V^{-T}\Sigma^{-1}U^{-1} = V\Sigma^{-1}U^T$$

compte tenu de l'orthogonalité des matrices U et V. L'expression de A^{-1} se déduit donc immédiatement de la première partie de la démonstration. ∎

Les valeurs singulières sont reliées à certaines normes de la matrice A. En effet nous avons le

Théorème 11

$$\|A\|_2 = \sigma_1 \quad et \quad \|A^{-1}\|_2 = 1/\sigma_p.$$

Démonstration.

On doit démontrer que

$$\sigma_1 = \|A\|_2 = \max_{x \neq 0} \frac{\|Ax\|_2}{\|x\|_2}.$$

Mais, puisque

$$\frac{\|Av_1\|_2}{\|v_1\|_2} = \sigma_1 \frac{\|u_1\|_2}{\|v_1\|_2} = \sigma_1$$

on a donc $\|A\|_2 \geq \sigma_1$.

Nous allons démontrer que cette inégalité a lieu aussi dans l'autre sens ce qui démontrera l'égalité. Soit x un vecteur quelconque de $\mathbb{R}^p$. Il peut s'écrire sur la base des v_i, c'est–à–dire $x = a_1 v_1 + \cdots + a_p v_p$. A cause de l'orthonormalité des v_i, on a donc $(x, x) = a_1^2 + \cdots + a_p^2$. Mais $Ax = a_1 Av_1 + \cdots + a_p Av_p = a_1\sigma_1 u_1 + \cdots + a_p \sigma_p u_p$ et, puisque les u_i sont orthonormaux, on a $(Ax, Ax) = a_1^2\sigma_1^2 + \cdots + a_p^2\sigma_p^2$. Mais, $\forall i, \sigma_i \leq \sigma_1$ et donc $(Ax, Ax) \leq \sigma_1^2(a_1^2 + \cdots + a_p^2) = \sigma_1^2(x, x)$. D'où finalement $(Ax, Ax)/(x, x) \leq \sigma_1$.

Démontrons maintenant la seconde égalité. On a vu, dans la démonstration du Théorème 10, que $A^{-1} = V^{-T}\Sigma^{-1}U^{-1} = V\Sigma^{-1}U^T$ d'après l'orthogonalité des matrices U et V. Les valeurs singulières de A^{-1} sont donc les σ_i^{-1} avec $\sigma_p^{-1} \geq \cdots \geq \sigma_1^{-1} > 0$. En appliquant la première égalité du Théorème à A^{-1} on obtient le résultat. ∎

D'après ce Théorème, on a donc $\kappa_2(A) = \sigma_1/\sigma_p$ et, $\forall x$,

$$\sigma_p \leq \frac{\|Ax\|_2}{\|x\|_2} \leq \sigma_1.$$

σ_p représente également la distance euclidienne de A à l'ensemble des matrices singulières. En effet, soit A_S la matrice singulière la plus proche de A, c'est–à–dire telle que la norme $\|A - A_S\|_2$ soit la plus petite possible. Alors

$$\|A - A_S\|_2 = \sigma_p \quad \text{et} \quad \frac{\|A - A_S\|_2}{\|A\|_2} = \frac{1}{\kappa_2(A)}.$$

La norme de Frobenius d'un matrice A est définie par

$$\|A\|_F^2 = \sum_{i,j=1}^{p} a_{ij}^2.$$

On pourra montrer que (exercice)

$$\|A\|_F^2 = \sigma_1^2 + \cdots + \sigma_p^2.$$

On a également (à démontrer)

$$\det(A) = \prod_{i=1}^{p} \sigma_i.$$

Considérons le système linéaire $Ax = b$. Nous avons vu, dans la démonstration du Théorème 10, que $A^{-1} = V\Sigma^{-1}U^T$. Donc, x s'exprime dans la base des vecteurs v_i comme

$$x = V\Sigma^{-1}U^T b = \sum_{i=1}^{p} \frac{(u_i, b)}{\sigma_i}\ v_i.$$

2.8.2 Matrices rectangulaires

Considérons maintenant le cas où $A \in \mathbb{R}^{n\times m}$ et est de rang r. On montre, comme dans le cas des matrices carrées, qu'il existe des nombres réels $\sigma_1 \geq \cdots \geq \sigma_r > 0$, une matrice orthogonale $V = [v_1, \ldots, v_m] \in \mathbb{R}^{m\times m}$ et une matrice orthogonale $U = [u_1, \ldots, u_n] \in \mathbb{R}^{n\times n}$ tels que

$$\begin{array}{llll} Av_i = \sigma_i u_i, & i = 1, \ldots, r & A^T u_i = \sigma_i v_i, & i = 1, \ldots, r \\ Av_i = 0, & i = r+1, \ldots, m & A^T u_i = 0, & i = r+1, \ldots, n. \end{array}$$

Les vecteurs $v_1, \ldots, v_m$ sont donc les vecteurs propres de $A^T A$ et $u_1, \ldots, u_n$ ceux de AA^T correspondants aux valeurs propres non nulles $\sigma_1^2, \ldots, \sigma_r^2$ de $A^T A$ et de AA^T.

Soit Σ la matrice $n \times m$

$$\Sigma = \begin{pmatrix} \widehat{\Sigma} & 0 \\ 0 & 0 \end{pmatrix}$$

avec $\widehat{\Sigma}$ =diag$(\sigma_1, \ldots, \sigma_r)$. Alors on a

$$A = U\Sigma V^T$$

et, par conséquent

$$A = \sum_{i=1}^{r} \sigma_i u_i v_i^T.$$

Ces résultats restent valables dans le cas d'une matrice carrée de dimension n et de rang $r < n$.

2.9 Quelques méthodes et formules utiles

Dans cette Section nous allons fournir des formules et des méthodes qui peuvent être utiles en algèbre matricielle.

2.9.1 Méthode de bordage

Dans un certain nombre d'applications, on doit résoudre une suite de systèmes linéaires de dimensions croissantes où chaque matrice est obtenue à partir de

la matrice précédente en la bordant par une nouvelle ligne et une nouvelle colonne et en ajoutant un nouvel élément au second membre. On peut résoudre récirsivement ces systèmes en utilisant la méthode de bordage.

Soit A_n une matrice de dimension n et soit x_n la solution du système

$$A_n x_n = b_n.$$

On considère la matrice bordée

$$A_{n+1} = \begin{pmatrix} A_n & u_n \\ v_n & a_n \end{pmatrix}$$

où u_n et v_n^T sont des vecteurs de $\mathbb{R}^n$ et a_n un scalaire. On a (faire la démonstration en exercice)

$$A_{n+1}^{-1} = \begin{pmatrix} A_n^{-1} + A_n^{-1} u_n \beta_n^{-1} v_n A_n^{-1} & -A_n^{-1} u_n \beta_n^{-1} \\ -\beta_n^{-1} v_n A_n^{-1} & \beta_n^{-1} \end{pmatrix}$$

avec $\beta_n = a_n - v_n A_n^{-1} u_n$.

Soit x_{n+1} la solution du système

$$A_{n+1} x_{n+1} = b_{n+1} = \begin{pmatrix} b_n \\ d_n \end{pmatrix}$$

où d_n est un scalaire. On a

$$x_{n+1} = \begin{pmatrix} x_n \\ 0 \end{pmatrix} + \begin{pmatrix} -A_n^{-1} u_n \\ 1 \end{pmatrix} \beta_n^{-1} (d_n - v_n x_n).$$

Les vecteurs $q_n = -A_n^{-1} u_n$ qui interviennent dans ces formules peuvent également être calculés récursivement par la même méthode de bordage.

Toutes les formules précédentes restent valables si l'on borde simultanément par plusieurs lignes et plusieurs colonnes. Dans ce cas u_n et v_n sont des matrices rectangulaires et d_n un vecteur.

Comme nous allons maintenant le voir, β_n est un complément de Schur.

2.9.2 Complément de Schur

Considérons une matrice partitionnée en quatre blocs

$$M = \begin{pmatrix} A & B \\ C & D \end{pmatrix},$$

où la sous–matrice A est carrée et inversible. Le complément de Schur de A dans M, dénoté (M/A), est la matrice

$$(M/A) = D - CA^{-1}B.$$

Quand la matrice M est carrée, la notion de complément de Schur est reliée à l'élimination de Gauss par blocs (voir Chapitre 4)

$$M = \begin{pmatrix} A & B \\ C & D \end{pmatrix} = \begin{pmatrix} I & 0 \\ CA^{-1} & I \end{pmatrix} \begin{pmatrix} A & B \\ 0 & (M/A) \end{pmatrix}.$$

De plus, on a

$$\det M = \det A \cdot \det(M/A).$$

2.9.3 Formule de Sherman–Morrison

Soit A une matrice $p \times p$ et U et V deux matrices $p \times k$ avec $k \leq p$. La formule de Sherman–Morrison est

$$(A + UV^T)^{-1} = A^{-1} - A^{-1}U(I + V^T A^{-1}U)^{-1}V^T A^{-1}.$$

Il faut naturellement supposer que ni A ni $I + V^T A^{-1}U$ n'est singulière. Cette formule montre qu'une correction de rang k d'une matrice induit une correction de rang k de son inverse.

Cette formule est reliée à la méthode de bordage.

2.9.4 Identité de Sylvester

Soit la matrice

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix}.$$

Notons $D(i, j; k, m)$ le déterminant de la matrice formée par les lignes i à j et les colonnes k à m de la matrice A avec $i \leq j$ et $k \leq m$.

L'identité de Sylvester est

$$\begin{aligned} D(1, n; 1, n)D(2, n-1; 2, n-1) = \quad & D(1, n-1; 1, n-1)D(2, n; 2, n) \\ & - D(1, n-1; 2, n)D(2, n; 1, n-1). \end{aligned}$$

2.10 Complexité des calculs matriciels

Un domaine important, qui se situe à la frontière entre analyse numérique et informatique, est celui de la complexité des algorithmes numériques. Il consiste

dans l'étude théorique et pratique des algorithmes qui demandent le moins possible d'opérations arithmétiques pour effectuer un certain calcul. Par étude théorique, on entend la démonstration de l'existence d'une borne inférieure pour le nombre d'opérations arithmétiques nécessaires à la réalisation d'un calcul puis la recherche de la valeur de cette borne. L'étude pratique consiste ensuite à trouver un algorithme dont le nombre d'opérations arithmétiques se rapproche le plus possible de cette borne inférieure.

Nous allons illustrer ceci par l'exemple du calcul du produit de deux matrices. Soient $A = (a_{ij})$ et $B = (b_{ij})$ deux matrices de dimensions respectives $m \times n$ et $n \times p$ et soit $C = (c_{ij}) = AB$. On calcule habituellement C à l'aide de

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}, \quad i = 1, \ldots, m, \quad j = 1, \ldots, p.$$

Le calcul de chaque élément de C demande n multiplications et $n-1$ additions soit, au total, nmp multiplications et $(n-1)mp$ additions.

On a pensé, il y a quelques années seulement, que ce produit pouvait s'effectuer en un moins grand nombre d'opérations. On ne connaît pas actuellement la borne inférieure du nombre d'opérations mais l'on sait uniquement que, pour des matrices carrées de dimension n, elle est comprise entre $K_1 n^2$ et $K_2 n^{2.496}$ oùK_1 et K_2 sont des constantes indépendantes de n.

Le premier algorithme que nous allons décrire est du à S. Winograd et il date de 1970. Il est basé sur l'identité

$$x_1 y_1 + x_2 y_2 = (x_1 + y_2)(x_2 + y_1) - x_1 x_2 - y_1 y_2 \tag{2.2}$$

et son extension au calcul du produit scalaire de deux vecteurs de dimension $n = 2k$

$$\sum_{i=1}^{2k} x_i y_i = \sum_{i=1}^{k} (x_{2i-1} + y_{2i})(x_{2i} + y_{2i-1}) - \sum_{i=1}^{k} x_{2i-1} x_{2i} - \sum_{i=1}^{k} y_{2i-1} y_{2i}.$$

En utilisant la formule précédente pour chaque c_{ij}, le calcul du produit de matrices $C = AB$ s'effectue de la façon suivante (en supposant que $n = 2k$)

$$\begin{aligned}
f_i &= \sum_{r=1}^{k} a_{i,2r-1} a_{i,2r}, \quad i = 1, \ldots, m \\
g_j &= \sum_{r=1}^{k} b_{2r-1,j} b_{2r,j}, \quad j = 1, \ldots, p \\
c_{ij} &= \sum_{r=1}^{k} (a_{i,2r-1} + b_{2r,j})(a_{i,2r} + b_{2r-1,j}) - f_i - g_j, \quad i = 1, \ldots, m, \quad j = 1, \ldots, p.
\end{aligned}$$

On vérifiera que cet algorithme nécessite

$$\begin{aligned}
&\frac{nmp}{2} + \frac{n}{2}(m+p) \quad \text{multiplications} \\
&\frac{3}{2}nmp + mp + \left(\frac{n}{2} - 1\right)(m+p) \quad \text{additions.}
\end{aligned}$$

On voit, que par rapport à la formule classique rappelée plus haut, cet algorithme demande deux fois moins de multiplications mais que le nombre d'additions est augmenté. Il y a cependant un gain, surtout parce que, sur ordinateur, une addition dure bien moins de temps qu'une multiplication.

On peut naturellement se demander s'il est possible de mieux faire en utilisant une identité avec trois termes à la place de (2.2). La réponse est négative et donc l'algorithme de Winograd est optimal en ce sens. On peut utiliser cet algorithme pour l'inversion de matrices ou la résolution des systèmes linéaires.

Un second algorithme, pour effectuer le produit de deux matrices carrées de dimension $n = 2k$ a été obtenu par V. Strassen en 1969. Il nécessite moins de $4.7\ n^{\log_2 7}$ opérations. Il peut également être utilisé pour le calcul de l'inverse et du déterminant de A ainsi que pour la résolution d'un système linéaire, le tout avec un nombre d'opérations du même ordre de grandeur.

Dans cet algorithme, puisque $n = 2k$ est pair, on commence par partitionner les trois matrices en quatre blocs de dimension k

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}, B = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}, C = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}.$$

Puis on calcule les matrices suivantes

$$\begin{aligned} Q_1 &= (A_{11} + A_{22})(B_{11} + B_{22}) \\ Q_2 &= (A_{21} + A_{22})B_{11} \\ Q_3 &= A_{11}(B_{12} - B_{22}) \\ Q_4 &= A_{22}(B_{21} - B_{11}) \\ Q_5 &= (A_{11} + A_{12})B_{22} \\ Q_6 &= (A_{21} - A_{11})(B_{11} + B_{12}) \\ Q_7 &= (A_{12} - A_{22})(B_{21} + B_{22}) \end{aligned}$$

et l'on a

$$\begin{aligned} C_{11} &= Q_1 + Q_4 - Q_5 + Q_7 \\ C_{12} &= Q_3 + Q_5 \\ C_{21} &= Q_2 + Q_4 \\ C_{22} &= Q_1 + Q_3 - Q_2 + Q_6. \end{aligned}$$

Cet algorithme nécessite 7 produits et 18 additions de matrices $k \times k$. Plus n est grand et plus il devient avantageux. Cela tient au fait qu'il ne demande que 7 produits de matrices au lieu de 8 pour l'algorithme classique. Puisque le produit de deux matrices $k \times k$ a besoin habituellement de k^3 multiplications et de $(k-1)k^2$ additions et que la somme de deux matrices $k \times k$ a besoin de k^2 additions, l'algorithme de Strassen nécessite au total

$$\begin{aligned} &7k^3 = (7/8)n^3 \quad \text{multiplications} \\ &7(k^3 - k^2) + 18k^2 7k^3 + 11k^2 = (7/8)n^3 + (11/4)n^2 \quad \text{additions.} \end{aligned}$$

Dès que $n > 30$, l'algorithme de Strassen devient plus avantageux. Bien sur, si k est lui–même pair, on peut recommençer la procédure. Ainsi, lorsque $n = 2^p$, on trouve que le nombre $M(p)$ de multiplications et le nombre $A(p)$ d'additions vérifient

$$\begin{aligned} M(p+1) &= 7M(p) \\ A(p+1) &= 7A(p) + 18(4^p) \end{aligned}$$

avec $M(0) = 1$ et $A(0) = 0$. D'où finalement

$$\begin{aligned} M(p) &= 7^p = (2^p)^{\log_2 7} = n^{\log_2 7} = n^{2.807} \\ A(p) &= 6(7^p - 4^p) = 6[(2^p)^{\log_2 7} - (2^p)]^2 = 6(n^{\log_2 7} - n^2). \end{aligned}$$

Si n n'est pas une puissance de 2, on borde la matrice par des zéros pour y arriver. On a alors un nombre de multiplications inférieur à $7n^{\log_2 7} - 42n^2$ et un résultat similaire pour le nombre d'additions. Ainsi, cet algorithme ne nécessite que $4.7\ n^{2.807}$ opérations.

Chapitre 3

Les systèmes linéaires

Le but de ce chapitre est, en premier lieu, de poser le problème mathématique de la résolution des systèmes d'équations linéaires, de voir comment il peut, en théorie, se résoudre et de fournir certains outils qui seront utiles par la suite.

3.1 Généralités sur les systèmes linéaires

On suppose connue une matrice carrée complexe A de dimension n ainsi qu'un vecteur b de $\mathbb{C}^n$. Le problème que nous allons chercher à résoudre consiste à trouver le vecteur $x \in \mathbb{C}^n$ qui vérifie

$$Ax = b.$$

C'est ce que l'on appelle un *système d'équations linéaires* ou, plus simplement, un *système linéaire*. Si nous l'explicitons complètement, il s'écrit

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

ou encore

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

Comme nous l'avons vu dans le Chapitre 2, ce système a une solution unique si et seulement si $\det A \neq 0$. Si ce déterminant est nul alors le système peut

avoir une infinité de solutions ou n'en avoir aucune. Prenons un exemple

$$\begin{aligned} x_1 + 2x_2 &= 1 \\ 2x_1 + 4x_2 &= 2. \end{aligned}$$

La seconde équation est le double de la première et le déterminant de la matrice est nul. Si l'on donne à x_1 une valeur quelconque, alors $x_2 = (1 - x_1)/2$. Si l'on donne à x_2 une valeur quelconque, alors $x_1 = 1 - 2x_2$. Ce système à donc une infinité de solutions. Par contre, si nous remplaçons la seconde équation par $2x_1 + 4x_2 = a \neq 2$, alors on ne peut pas avoir simultanément $2(x_1 + 2x_2) = 2(1) = a \neq 2$ et le système n'a donc aucune solution. Dans le premier cas, le second membre b appartient à l'image de A, c'est–à–dire l'ensemble des vecteurs de la forme Ax avec x quelconque, alors que, dans le second cas, b n'appartient pas à cette image.

On sait que, si $\det A \neq 0$, les composantes x_i de la solution du système $Ax = b$ sont données par les formules suivantes pour $i = 1, \ldots, n$

$$x_i = \begin{vmatrix} a_{11} & \cdots & a_{1,i-1} & b_1 & a_{1,i+1} & \cdots & a_{1n} \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ a_{n1} & \cdots & a_{n,i-1} & b_n & a_{n,i+1} & \cdots & a_{nn} \end{vmatrix} \Big/ \begin{vmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \cdots & a_{nn} \end{vmatrix}.$$

Le calcul d'un déterminant de dimension n par les règles données dans la Chapitre 2 nécessite $n!(n-1)$ multiplications et $n! - 1$ additions (à démontrer). Par conséquent, la résolution du système $Ax = b$ demande, au total, $n!n(n+1) - 1$ opérations arithmétiques élémentaires. C'est un nombre qui devient rapidement extrêmement grand. Pour s'en convaincre, supposons que ce calcul soit réalisé sur un ordinateur effectuant 10^7 opérations par seconde. Le Tableau 3.1 donne le temps de calcul nécessaire à la résolution du système.

Rappelons que l'âge de l'univers est, sans doute, situé entre 10 et 15 10^9 ans ! ! ! On voit que, même en multipliant la vitesse des ordinateurs par un facteur important, cela ne changerait pas grand' chose.

Il est donc nécessaire de disposer d'autres méthodes pour résoudre un système linéaire et cela d'autant plus que, à l'heure actuelle, on doit résoudre des systèmes de plusieurs centaines de milliers, voire plusieurs millions d'inconnues. De tels systèmes s'obtiennent couramment lors de la discrétisation d'équations aux dérivées partielles pour différences finies ou par éléments finis. Nous en donnerons un exemple dans le Chapitre 7.

Les méthodes de résolution des systèmes linéaires se divisent en deux classes

- les méthodes *directes* qui fournissent la solution exacte x après un nombre fini d'opérations arithmétiques,
- les méthodes *itératives* où l'on construit une suite de vecteurs qui converge vers la solution x sous certaines conditions.

n	temps de calcul
9	2.9 sec.
10	36.3 sec.
11	8.0 min.
12	2 h.
13	1.2 jour
14	19.8 jours
15	340.5 jours
16	17 ans
17	326 ans
18	6577 ans
19	139250 ans
20	$3{\cdot}10^6$ ans
21	$71{\cdot}10^6$ ans
22	$2{\cdot}10^9$ ans
23	$40{\cdot}10^9$ ans

TAB. 3.1: Temps de résolution d'un système linéaire

Dans ce livre, nous n'étudierons que les méthodes directes. Elles ne peuvent être utilisées que pour des dimensions relativement faibles (mais que l'on rencontre quand même dans de nombreuses applications). Pour les très grands systèmes, il faut se tourner vers les méthodes itératives, en particulier les méthodes de projection, ou même combiner méthodes directes et itératives.

Le principe de base des méthodes directes est le suivant : on transforme un système que l'on ne sait pas résoudre en un système que l'on sait résoudre. Il existe essentiellement trois types de systèmes que l'on sait facilement résoudre

- les systèmes avec une matrice diagonale,
- les systèmes triangulaires, c'est–à–dire ceux où la matrice présente l'une des deux formes

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ & \ddots & \vdots \\ & & a_{nn} \end{pmatrix}, \qquad A = \begin{pmatrix} a_{11} & & \\ \vdots & \ddots & \\ a_{n1} & \cdots & a_{nn} \end{pmatrix}.$$

  Pour la première forme, que l'on appelle *triangulaire supérieure*, on a immédiatement

$$x_i = \frac{b_i - a_{i,i+1}x_{i+1} - \cdots - a_{in}x_n}{a_{ii}}, \quad i = n, n-1, \ldots, 1.$$

  La seconde forme de matrice s'appelle *triangulaire inférieure* et l'on résoud le système de façon similaire,

– les systèmes avec matrice orthogonale, c'est–à–dire telle que $A^{-1} = A^T$. On a donc $x = A^T b$.

Le passage à des systèmes triangulaires conduit aux méthodes du type de Gauss, celui à des systèmes orthogonaux conduit à la méthode de Householder. Le passage à un système diagonal est plus coûteux et n'est que rarement utilisé. Il correspond à la méthode de Gauss–Jordan.

Mais, auparavant, un certain nombre de notions importantes se doivent d'être étudiées.

Remarque 6

Dans ce qui suit, nous ne traiterons que le cas des systèmes linéaires réels car un système complexe peut se ramener à un système réel de dimension double. En effet

$$(A + iB)(x + iy) = b + ic$$

peut s'écrire

$$\begin{pmatrix} A & -B \\ B & A \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} b \\ c \end{pmatrix}.$$

3.2 Les erreurs numériques

Dans cette Section, nous allons d'abord étudier l'erreur a priori. Avant de résoudre un système linéaire, il faut commencer par introduire les données (éléments de la matrice et composantes du second membre) dans l'ordinateur. A cause de l'arithmétique de l'ordinateur, ces nombres peuvent ne pas être représentés exactement en mémoire. Par conséquent, le système que l'on va résoudre en pratique est quelque peu différent du véritable système et, dans certains cas, sa solution exacte peut être très éloignée de la solution exacte du système non perturbé. Le même phénomène se produit, bien sûr, si les données soient entachées d'une erreur. C'est ce genre d'erreur qui s'appelle erreur *a priori* car elle est présente a priori, c'est–à–dire avant le commencement de tout calcul. La notion qui permet de savoir si la solution d'un système est plus ou moins sensible aux perturbations sur les données est la notion de *conditionnement* définie dans la Section 2.7.3. Bien entendu, on voit que cette notion est complètement indépendante de l'algorithme qui sera utilisé par la suite pour résoudre le système.

Une fois les calculs effectués, la solution obtenue sur ordinateur est souvent différente de la solution exacte à cause des erreurs sur les données et de la propagation des erreurs dues à l'arithmétique de l'ordinateur dans l'algorithme. C'est ce que l'on appelle l'erreur *a posteriori*.

Pour ces deux types d'erreurs, nous obtiendrons des bornes supérieures. Cependant, bien souvent en pratique, ces bornes présentent l'inconvénient usuel des bornes supérieures, c'est–à–dire qu'elles surestiment largement la véritable erreur. De plus, le calcul de ces bornes fait intervenir des quantités inconnues, comme le conditionnement, ou difficiles à calculer. C'est pour cela que, dans la dernière Section, nous donnerons d'autres formules pour estimer l'erreur.

3.2.1 Étude a priori

Nous voulons étudier l'effet sur la solution d'une perturbation à la fois sur la matrice et sur le second membre. Afin de rendre cette étude plus facile, nous analyserons d'abord séparément chacune de ces perturbations.

Soit δb une perturbation sur le second membre et soit δx la perturbation induite sur la solution, c'est–à–dire

$$A(x + \delta x) = b + \delta b$$

où x est la solution exacte du système $Ax = b$. On a le résultat suivant

Théorème 12
Si $x \neq 0$, alors

$$\frac{\|\delta x\|}{\|x\|} \leq \kappa(A) \frac{\|\delta b\|}{\|b\|}.$$

Démonstration.

On a, d'après la définition d'une norme matricielle,

$$\frac{\|b\|}{\|x\|} = \frac{\|Ax\|}{\|x\|} \leq \|A\|.$$

De même, puisque $\delta x = A^{-1}\delta b$, on a

$$\frac{\|\delta x\|}{\|\delta b\|} = \frac{\|A^{-1}\delta b\|}{\|\delta b\|} \leq \|A^{-1}\|.$$

En multipliant entre elles ces inégalités, on obtient le résultat. ∎

Ce résultat nous montre donc que le conditionnement $\kappa(A)$ représente le facteur par lequel l'erreur relative sur le second membre *peut* être multipliée. En d'autre termes, $\kappa(A)$ apparaît comme étant le facteur *possible* d'amplification de l'erreur relative sur le second membre. Les mots *peut* et *possible* expriment le fait que le résultat du Théorème nous donne une borne supérieure qui, peut–être, n'est pas atteinte. Donc, si la matrice est bien conditionnée (c'est–à–dire si $\kappa(A)$ est voisin de 1) alors une petite perturbation du second membre n'entraînera qu'une petite perturbation de la solution tandis que si la matrice est mal conditionnée, une petite perturbation du second membre pourra entraîner une grande perturbation du résulat.

Considérons maintenant le cas d'une perturbation δA sur la matrice. Elle induit sur la solution une perturbation que nous continuerons d'appeler δx bien qu'il ne soit pas le même vecteur que précédemment. On a donc

$$(A + \delta A)(x + \delta x) = b.$$

Nous avons le

Théorème 13
Si $x \neq 0$ *et si* $\|A^{-1}\| \cdot \|\delta A\| < 1$, *alors*

$$\frac{\|\delta x\|}{\|x\|} \leq \frac{\kappa(A)\dfrac{\|\delta A\|}{\|A\|}}{1 - \kappa(A)\dfrac{\|\delta A\|}{\|A\|}}.$$

Démonstration.

Il faut commencer par démontrer que la matrice perturbée $A + \delta A$ est inversible. On a

$$A + \delta A = A(I + A^{-1}\delta A).$$

Si $\|A^{-1}\| \cdot \|\delta A\| < 1$, alors $\|A^{-1}\delta A\| < 1$ et donc le rayon spectral de $A^{-1}\delta A$ est strictement inférieur à 1. La matrice $I + A^{-1}\delta A$ ne peut, par conséquent, pas avoir de valeur propre nulle et elle est donc inversible.

Démontrons maintenant l'inégalité du Théorème. On a

$$(A + \delta A)\delta x = -\delta A x$$

d'où

$$\delta x = -(I + A^{-1}\delta A)^{-1}A^{-1}\delta A x$$

et

$$\|\delta x\| \leq \|(I + A^{-1}\delta A)^{-1}A^{-1}\delta A\| \cdot \|x\|$$

soit encore

$$\begin{aligned} \|\delta x\| &\leq \|(I + A^{-1}\delta A)^{-1}\| \cdot \|A^{-1}\delta A\| \cdot \|x\| \\ \|\delta x\| &\leq \|(I + A^{-1}\delta A)^{-1}\| \cdot \|A^{-1}\| \cdot \|\delta A\| \cdot \|x\|. \end{aligned}$$

Si B est une matrice telle que $\|B\| < 1$, alors elle ne peut avoir 1 comme valeur propre puisque $\varrho(B) \leq \|B\|$. Donc $I - B$ n'a pas de valeur propre nulle, elle est inversible et l'on peut écrire

$$(I - B)^{-1} = I + B(I - B)^{-1}$$

d'où

$$\|(I - B)^{-1}\| \leq 1 + \|B\| \cdot \|(I - B)^{-1}\|.$$

De plus, en itérant la formule précédente,

$$\begin{aligned}(I-B)(I+B+\cdots+B^k) &= I-B^{k+1}\\ (I+B+\cdots+B^k)-(I-B)^{-1} &= -(I-B)^{-1}B^{k+1}.\end{aligned}$$

Puisque $\|B\|<1$, lorsque k tend vers l'infini B^k tend vers la matrice nulle. Par conséquent $(I-B)^{-1}=I+B+B^2+B^3+\cdots$ puisque cette série est convergente. On a donc

$$\|(I-B)^{-1}\| \leq 1+\|B\|(1+\|B\|+\|B\|^2+\cdots) \leq \frac{1}{1-\|B\|}.$$

En appliquant ce résultat à la matrice $-A^{-1}\delta A$, on obtient donc

$$\|(I+A^{-1}\delta A)^{-1}\| \leq \frac{1}{1-\|A^{-1}\delta A\|} \leq \frac{1}{1-\|A^{-1}\|\cdot\|\delta A\|}.$$

D'où finalement

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

ce qui termine la démonstration en utilisant la définition de $\kappa(A)$. ∎

Les conclusions sont les mêmes que celles du Théorème précédent puisque, en géneral, $\kappa(A)\|\delta A\|/\|A\|$ est petit par rapport à 1. $\kappa(A)$ apparaît donc encore comme le facteur éventuel d'amplification de l'erreur relative.

Nous pouvons maintenant aborder le cas complet d'une perturbation sur la matrice et sur le second membre. De nouveau nous écrirons

$$(A+\delta A)(x+\delta x)=b+\delta b.$$

On a le

Théorème 14
Si $x\neq 0$ et si $\|A^{-1}\|\cdot\|\delta A\|<1$, alors

$$\frac{\|\delta x\|}{\|x\|} \leq \frac{\kappa(A)}{1-\kappa(A)\dfrac{\|\delta A\|}{\|A\|}}\left(\frac{\|\delta b\|}{\|b\|}+\frac{\|\delta A\|}{\|A\|}\right).$$

Démonstration.

Bien évidemment, la condition du Théorème est là pour assurer la régularité de la matrice $A+\delta A$. On a

$$\delta x=(I+A^{-1}\delta A)^{-1}A^{-1}(\delta b-\delta A x)$$

d'où

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

On obtient le résultat en utilisant ensuite $\|b\|\leq\|A\|\cdot\|x\|$. ∎

On voit que, si $\delta b=0$ ou si $\delta A=0$, on retrouve respectivement les inégalités des deux Théorèmes précédents.

3.2.2 Étude a posteriori

Supposons maintenant avoir obtenu, sur ordinateur, un vecteur y qui est une approximation (à cause des erreurs dues à l'arithmétique de l'ordinateur ou parce que l'on a utilisé une méthode ne fournissant pas la solution exacte mais seulement une solution approchée comme c'est le cas avec une méthode itérative) de la solution exacte x du système. Nous allons maintenant tester la qualité de cette approximation, c'est–à–dire étudier l'erreur a posteriori.

Soit $r = b - Ay$ le vecteur *résidu* correspondant à y. On a la

Théorème 15

$$\|y - x\| \leq \kappa(A) \frac{\|r\|}{\|b\|} \|x\|.$$

Démonstration.

On a $y = A^{-1}(b - r)$. Soustrayons $x = A^{-1}b$. On obtient $y - x = -A^{-1}r$ et

$$\|y - x\| \leq \|A^{-1}\| \cdot \|r\|.$$

Or $\|b\| \leq \|A\| \cdot \|x\|$, ce qui termine la démonstration en multipliant entre elles ces deux inégalités. ∎

L'inégalité du Théorème précédent n'est pas facile à utiliser dans la pratique car elle nécessite la connaissance de $\kappa(A)$. Supposons que l'on connaisse une approximation C de A^{-1} (ce que nous appelerons un *préconditionneur* dans la Section 3.3). Posons $R = I - AC$. On a le

Théorème 16 *Si* $\|R\| < 1$ *alors*

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

Démonstration.

D'après le Théorème précédent

$$\|y - x\| \leq \|A^{-1}\| \cdot \|r\|.$$

De plus, on a $A^{-1} = C(I - R)^{-1}$, d'où

$$\|A^{-1}\| \leq \|C\| \cdot \|(I - R)^{-1}\|.$$

Donc, puisque $\|R\| < 1$, on a, en utilisant une inégalité de la démonstration du Théorème 13,

$$\|(I - R)^{-1}\| \leq 1/(1 - \|R\|)$$

et l'on obtient l'inégalité à démontrer. ∎

La condition $\|R\| < 1$ n'est pas trop restrictive car, s'il n'en était pas ainsi, C serait une bien mauvaise approximation de A^{-1}.

3.2.3 Exemples

Donnons un exemple numérique qui montre bien l'influence d'un mauvais conditionnement. Si x est la solution du système $Ax = b$ alors le vecteur résidu $r = b - Ax$ doit être nul.

Considérons les trois systèmes suivants qui sont mal conditionnés

système 1	solution
$3x_1 + 4x_2 - 7 = 0$ $3x_1 + 4.00001x_2 - 7.00001 = 0$	$x_1 = 1$ $x_2 = 1$

système 2	solution
$3y_1 + 4y_2 - 7 = 0$ $3y_1 + 3.99999y_2 - 7.00004 = 0$	$y_1 = 7 + 2/3$ $y_2 = -4$

système 3	solution
$3z_1 + 4z_2 - 7 = 0$ $3z_1 + 3.999992z_2 - 7.000042 = 0$	$z_1 = 9 + 1/3$ $z_2 = -5 - 1/4$

Si, dans chaque système, l'on reporte les trois solutions on trouve des résidus dont la première composante est toujours nulle et dont la seconde est donnée dans le tableau suivant

	système 1	système 2	système 3
x_1, x_2	0	-0.00005	-0.00005
y_1, y_2	-0.00005	0	-0.00001
z_1, z_2	-0.0000625	-0.0000105	0

Si l'erreur est petite alors le résidu sera petit. Mais un point capital à tenir en compte est que la réciproque n'est pas toujours vérifiée : le résidu peut être petit et, cependant, l'erreur peut être grande. C'est ce que montre cet exemple.

Voici un exemple encore plus caractéristique. Considérons le système

$$\begin{pmatrix} 1.2969 & 0.8648 \\ 0.2161 & 0.1441 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0.8642 \\ 0.1440 \end{pmatrix}.$$

Sa solution exacte est $x = (2, -2)^T$.

Avec la solution approchée $y = (0.9911, -0.4870)^T$, on obtient un résidu $r = b - Ay = (-10^{-8}, 10^{-8})^T$.

Pour cette matrice $\kappa_2 = 2.4973 \cdot 10^8$.

3.2.4 Estimations de l'erreur

Soit x la solution exacte du système $Ax = b$ et y une approximation de x (obtenue par une méthode directe ou comme itéré d'une méthode itérative). L'erreur $e = x - y$ et le résidu $r = b - Ay$ sont reliés par $r = Ae$ et, donc, il n'est pas possible de calculer l'erreur à partir du résidu.

Cependant, on a

$$\begin{aligned} 1 \leq & \quad \frac{\|A\| \cdot \|e\|}{\|r\|} \quad \leq \kappa \\ \frac{1}{\kappa} \leq & \quad \frac{\|e\|}{\|A^{-1}\| \cdot \|r\|} \quad \leq 1 \end{aligned}$$

où $\kappa = \|A\| \cdot \|A^{-1}\|$, les normes étant les normes euclidiennes. Par conséquent, si l'on connaît $\|A\|$ ou $\|A^{-1}\|$, alors les quantités $\|r\|/\|A\|$ et $\|A^{-1}\| \cdot \|r\|$ peuvent être considérées comme des estimations de $\|e\|$ et l'on a les bornes

$$\frac{\|r\|}{\|A\|} \leq \|e\| \leq \|A^{-1}\| \cdot \|r\|. \tag{3.1}$$

Ces estimations demandent la connaissance de la norme euclidienne de A ou de son inverse et, de plus, dans certains cas, ces bornes peuvent être beaucoup trop larges.

Voyons donc maintenant comment estimer l'erreur.

Nous allons étudier la quantité

$$\rho = \frac{\|r\|^2}{\|Ar\|}$$

où les normes sont quelconques.

Nous avons

$$\frac{\|e\|}{\rho} = \frac{\|e\|}{\|r\|^2}\|Ar\| = \frac{\|A^{-1}r\| \cdot \|Ar\|}{\|r\|^2} \leq \|A^{-1}\| \cdot \|A\| = \kappa.$$

D'un autre côté, $r = A^{-1}Ar$ et donc $\|r\| \leq \|A^{-1}\| \cdot \|Ar\|$. Nous avons également $\|r\| \leq \|A\| \cdot \|e\|$, d'où, en multipliant ce deux inégalités entre elles,

$$\frac{1}{\kappa} \leq \frac{\|e\|}{\rho} \leq \kappa.$$

Ces inégalités montrent que, si le conditionnement de A est voisin de 1, alors on est certain que ρ est une bonne estimation de $\|e\|$. Cependant, même si le conditionnement de A est grand, cette estimation peut être satisfaisante.

Il est possible d'obtenir des bornes un peu plus fines

$$\frac{1}{\kappa} \leq m(A) \leq \frac{\|e\|}{\rho} \leq M(A) \leq \kappa$$

avec

$$\begin{aligned} m(A) &= \min_{u\neq 0} \frac{\|Au\| \cdot \|A^{-1}u\|}{\|u\|^2} \\ M(A) &= \max_{u\neq 0} \frac{\|Au\| \cdot \|A^{-1}u\|}{\|u\|^2}. \end{aligned}$$

Des démonstrations similaires sont valables pour la quantité $\rho' = \|r\|^2/\|A^T r\|$ qui est également une estimation de l'erreur. Cette estimation est, dans le cas de la norme euclidienne, une borne inférieure de la norme de l'erreur. En effet

$$\|r\|_2^2 = (r,r) = (r,Ae) = (A^T r, e) \leq \|A^T r\|_2 \cdot \|e\|_2.$$

3.3 Le préconditionnement

Si la matrice du système $Ax = b$ est mal conditionnée, une petite variation des données pourra entraîner une grande variation sur la solution. D'autre part, si la matrice est mal conditionnée un certain nombre de méthodes itératives convergeront lentement car plus le conditionnement est grand et plus leur vitesse de convergence est faible. C'est le cas, par exemple, de la méthode de la plus profonde descente et de la méthode du gradient conjugué qui ne seront pas étudiées ici.

C'est pour ces deux raisons que l'on remplace souvent le système $Ax = b$ par le système $CAx = Cb$. La matrice C est choisie de sorte que le conditionnement de CA soit plus petit que celui de A. Une telle stratégie s'appelle *préconditionnement* (à gauche) et la matrice C s'appelle *préconditionneur* (à gauche). Plus $\kappa(CA)$ sera voisin de 1, meilleur sera le préconditionnement. Il est évident que le meilleur préconditionneur possible est $C = A^{-1}$, un choix impossible en pratique. On prendra donc pour C une approximation de A^{-1}. Il est possible de construire de telles approximations en théorie. Cependant, leur utilisation pratique est souvent limitée aux systèmes de dimension réduite tels que C puisse être gardée en mémoire de l'ordinateur. Pour les grands systèmes, il n'existe pas de préconditionneurs valables pour toute matrice et chaque cas particulier doit être étudié individuellement.

On peut également définir un préconditionnement à droite, c'est–à–dire où l'on considère le système $ACy = b$ avec $x = Cy$ ainsi qu'un préconditionnement bilatéral $CAC'y = Cb$ avec $x = C'y$.

Au lieu de chercher une matrice C qui soit une bonne approximation de A^{-1}, on peut aussi chercher une matrice M qui soit une approximation de A puis prendre $C = M^{-1}$. Il est évident que, dans ces stratégies, on ne calcule ni M^{-1} ni le produit CA. En effet, dans les méthodes itératives, il est n'est nécessaire que de savoir calculer des produits CAv où v est un vecteur quelconque. Si C est donnée, on calcule d'abord le vecteur Av puis on le multiplie par la matrice

C. Pour calculer le vecteur $u = M^{-1}Av$, on résoud le système $Mu = Av$. Il faut, bien sûr, que ce système soit plus facile à résoudre que le système initial $Ax = b$. En général, on prendra pour M une matrice ayant plus d'éléments nuls que la matrice A. Par exemple, si M est la matrice diagonale formée par la diagonale de A, M^{-1} sera une bonne approximation de A^{-1} si A est à diagonale dominante, c'est–à–dire si $\forall i, |a_{ii}| > \sum_{j \neq i} |a_{ij}|$.

De bons préconditionneurs peuvent être construits par décomposition orthogonale (voir Chapitre 6). Si $A = QU$ où Q est une matrice orthogonale (i.e. $Q^T = Q^{-1}$) et U une matrice triangulaire supérieure, alors il est possible de prendre $C = U^{-1}$. En effet, dans ce cas, $\kappa_2(AC) = \kappa_2(Q) = 1$. De même, si $A^T = QU$ et si l'on prend $C = R^{-T}$, alors $\kappa_2(CA) = \kappa_2(Q^T) = 1$. Des approximations de U^{-1} et de U^{-T} peuvent s'obtenir par décomposition orthogonale incomplète. Dans la pratique, il est nécessaire de recourir à de tels procédés incomplets pour la raison évoquée plus haut. En effet, même si les matrices A et U sont creuses (c'est–à–dire qu'une grande majorité de leurs éléments sont nuls), il n'y a aucune raison pour qu'il en soit de même pour U^{-1}. On se livre donc seulement à une décomposition incomplète dans laquelle on impose à certains éléments de U^{-1} d'être nuls (même s'ils ne devaient pas l'être).

On peut également se livrer à une décomposition LU incomplète de la matrice A, une procédure connue sous le nom de ILU(0) où la première lettre signifie *incomplète*. Dans ce cas, on cherche une matrice L triangulaire inférieure à diagonale unité et une matrice U triangulaire supérieure, en imposant en plus à certains éléments de ces deux matrices d'être nuls, telles que $A = LU + R$. On prendra alors $M = LU$ comme approximation de A. On peut recommencer la décomposition LU de cette matrice M, une procédure qui s'appelle ILU(1) et qui peut être poursuivie pour obtenir des décompositions incomplètes successives ILU(k). Ces décompositions incomplètes peuvent servir également dans le préconditionnement bilatéral. En effet, si $A = LU$ (décomposition qui, comme on le verra plus loin, est obtenue par la méthode de Gauss), alors on pourra considérer le système $CAC'y = Cb$ avec $x = C'y$ et où C est une approximation de L^{-1} et C' une approximation de U^{-1}.

Lorsque, dans le préconditionnement bilatéral, les matrices C et C' sont diagonales, on parle d'*équilibrage* de la matrice A. On voit souvent écrit qu'il faut choisir C et C' de sorte que les éléments les plus grands en valeur absolue dans chaque ligne et dans chaque colonne soient à peu près égaux. C'est là une idée fausse. La bonne stratégie consiste en un choix tel que les sommes des éléments de chaque ligne et de chaque colonne de la matrice $|CAC'|$ soient à peu près égales.

Le préconditionnement d'un système linéaire est une opération fondamentale, souvent même plus importante que la méthode de résolution utilisée par la suite. Malheureusement, il n'existe pas de préconditionneur universel et il est nécessaire d'en bâtir un adapté à chaque problème (ou classe de problèmes) considéré. Quand on utilise une méthode itérative pour résoudre le système $Ax = b$ on peut envisager d'utiliser une suite de préconditionneurs C_k ou M_k,

mais c'est une technique plus coûteuse et difficile à mettre en œuvre.

3.4 Itération sur le résidu

Soit à résoudre le système $Ax = b$. On a obtenu une solution y qui, à cause des erreurs d'arrondis et du conditionnement de la matrice, n'est pas rigoureusement égale à x. Nous allons chercher à améliorer la précision de cette solution approchée.

On pose

$$\begin{aligned} e &= x - y \\ r &= b - Ay. \end{aligned}$$

On a $Ae = r$. Si l'on résoud ce nouveau système, on obtiendra l'erreur e. Mais ce système ne sera lui–même résolu que de façon approchée. On pourra alors itérer ce processus de correction, connu sous le nom d'*itération sur le résidu*. En général une seule itération suffit en fait pour améliorer le résultat, les autres itérations n'apportant pas de gain supplémentaire de précision et pouvant même ne pas converger. Cette procédure s'appelle le *raffinement itératif*.

Considérons, par exemple, le système

$$\begin{pmatrix} 3 & 2 & 1 \\ 2 & 2\ 10^{-6} & 2\ 10^{-6} \\ 1 & 2\ 10^{-6} & -10^{-6} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 3 + 3\ 10^{-6} \\ 6\ 10^{-6} \\ 2\ 10^{-6} \end{pmatrix}$$

dont la solution est $x = (10^{-6}, 1, 1)^T$. Sur un ordinateur dont la précision est de 10^{-9}, la solution obtenue

$$(0.999894122\ 10^{-6}, 0.999983255, 1.000033489)^T$$

n'a que 4 chiffres exacts. Après raffinement itératif, on obtient

$$(0.999999997\ 10^{-6}, 1.000000000, 1.000000000)^T.$$

Une technique similaire s'applique à l'inverse d'une matrice. Soit C_0 une approximation de A^{-1}. On peut améliorer cet inverse approché par une procédure qui s'apparente à l'itération sur le résidu.

On pose $R_0 = I - AC_0$ et l'on suppose que $\|R_0\| \leq K < 1$. Si ce n'est pas le cas, R_0 est vraiment une mauvaise approximation de A^{-1} et il ne sera pas possible de l'améliorer. On considère les itérations

$$\begin{array}{rclcrcl} C_1 &=& C_0(I + R_0) & & R_1 &=& I - AC_1 \\ C_2 &=& C_1(I + R_1) & & R_2 &=& I - AC_2 \\ & \vdots & & & & \vdots & \\ C_{k+1} &=& C_k(I + R_k) & & R_{k+1} &=& I - AC_{k+1} \\ & \vdots & & & & \vdots & \end{array}$$

On a

$$\begin{aligned} R_{k+1} &= I - AC_{k+1} \\ &= I - AC_k(I + R_k) \\ &= I - (I - R_k)(I + R_k) \\ &= R_k^2. \end{aligned}$$

Par conséquent, par récurrence, on a $R_k = R_0^{2^k}$ et $C_k = A^{-1}(I - R_0^{2^k})$. Il vient

$$\begin{aligned} C_k - A^{-1} &= -A^{-1}R_0^{2^k} \\ &= -C_0(I - R_0)^{-1}R_0^{2^k}. \end{aligned}$$

D'où

$$\begin{aligned} \|C_k - A^{-1}\| &\leq \|C_0\| \cdot \|(I - R_0)^{-1}\| \cdot \|R_0^{2^k}\| \\ \|R_0^{2^k}\| &\leq \|R_0\|^{2^k} \leq K^{2^k}. \end{aligned}$$

Et finalement, on obtient

$$\begin{aligned} \|C_k - A^{-1}\| &\leq \|C_0\| \cdot \|R_0^{2^k}\| \cdot (1 + \|R_0\| + \|R_0\|^2 + \cdots) \\ &\leq \|C_0\| \cdot \frac{K^{2^k}}{1 - K} \end{aligned}$$

ce qui montre que la convergence est très rapide (quadratique). Malheureusement cette procédure est onéreuse.

3.5 Moindres carrés

Considérons le système $Ax = b$ avec $A \in \mathbb{R}^{n \times m}$, $b \in \mathbb{R}^n$ et $x \in \mathbb{R}^m$.

Si $n > m$, il y a plus d'équations que d'inconnues et l'on dit alors que le système est *sur–déterminé*. On ne peut pas espérer en trouver une solution exacte et l'on va donc chercher x qui minimise $\|Ax - b\|_2$. C'est ce que l'on appelle résoudre le système au sens des *moindres carrés*. Si $n \geq m$ et si le rang r de A est égal à m, alors ce problème de moindres carrés admet une solution unique. Si $r < m$, la solution n'est pas unique et il existe plusieurs vecteurs x qui rendent $\|Ax - b\|_2$ minimum. Si $n < m$, il y a plus d'inconnues que d'équations et l'on parle alors d'un système *sous–déterminé*. Dans ce cas, il se peut quand même que le système n'admette pas de solution. Nous allons inclure tous ces cas dans ce qui suit et nous ne ferons donc plus d'hypothèses sur les valeurs respectives de n et de m.

Puisque la solution du problème aux moindres carrés n'est pas toujours unique, on va rechercher, parmi ses solutions, celle qui minimise $\|x\|_2$. Ce problème a toujours une solution unique comme nous le verrons.

Considérons la décomposition de A en valeurs singulières $A = U\Sigma V^T$ avec $U \in \mathbb{R}^{n\times n}$, $V \in \mathbb{R}^{m\times m}$ orthogonales,

$$\Sigma = \begin{pmatrix} \widehat{\Sigma} & 0 \\ 0 & 0 \end{pmatrix}$$

et $\widehat{\Sigma}$ =diag$(\sigma_1, \ldots, \sigma_r)$, $\sigma_1 \geq \cdots \geq \sigma_r > 0$.

Puisque U est orthogonale,

$$\|Ax - b\|_2 = \|U^T(Ax - b)\|_2 = \|\Sigma V^T x - U^T b\|_2.$$

Posons $c = U^T b$ et $y = V^T x$. On a

$$\|Ax - b\|_2^2 = \|\Sigma y - c\|_2^2 = \sum_{i=1}^{r} |\sigma_i y_i - c_i|^2 + \sum_{i=r+1}^{m} |c_i|^2. \tag{3.2}$$

Cette expression est minimum si et seulement si $y_i = c_i/\sigma_i$ pour $i = 1, \ldots, r$. Si $r < m$, alors $y_{r+1}, \ldots, y_m$ n'apparaissent pas dans l'expression précédente, ils n'ont aucun effet sur le résidu et peuvent donc être choisis arbitrairement. $\|y\|_2$ est minimum dans l'ensemble des solutions lorsque ces dernières composantes sont nulles. Puisque $x = Vy$ et que V est orthogonale, on a donc $\|x\|_2 = \|y\|_2$ ce qui montre que $\|x\|_2$ est minimum si et seulement si $\|y\|_2$ est minimum. Par conséquent, le problème aux moindres carrés admet une et une seule solution.

Répétons ce raisonnement en termes matriciels. On pose

$$c = \begin{pmatrix} \widehat{c} \\ d \end{pmatrix}, \quad y = \begin{pmatrix} \widehat{y} \\ z \end{pmatrix}$$

avec $\widehat{c}, \widehat{y} \in \mathbb{R}^r$. La relation (3.2) s'écrit

$$\|Ax - b\|_2^2 = \left\| \begin{pmatrix} \widehat{\Sigma} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \widehat{y} \\ z \end{pmatrix} - \begin{pmatrix} \widehat{c} \\ d \end{pmatrix} \right\|_2^2 = \left\| \begin{pmatrix} \widehat{\Sigma}\widehat{y} \\ d \end{pmatrix} \right\|_2^2 = \|\widehat{\Sigma}\widehat{y} - \widehat{c}\|_2^2 + \|d\|_2^2.$$

Cette expression est minimum si et seulement si $\widehat{y} = \widehat{\Sigma}^{-1}\widehat{c}$, c'est–à–dire $y_i = c_i/\sigma_i$ pour $i = 1, \ldots, r$. Le vecteur z peut être choisi de façon arbitraire mais on obtient la solution de norme minimale lorsque $z = 0$. La norme du résidu correspondant est alors égale à $\|d\|_2$.

La procédure à suivre pour résoudre le système $Ax = b$ au sens des moindres carrés peut donc se résumer ainsi

1. calculer $\begin{pmatrix} \widehat{c} \\ d \end{pmatrix} = c = U^T b$,
2. poser $\widehat{y} = \widehat{\Sigma}^{-1}\widehat{c}$,
3. poser $y = \begin{pmatrix} \widehat{y} \\ z \end{pmatrix} \in \mathbb{R}^n$, où z est arbitraire. la solution de norme minimale est obtenue pour $z = 0$,
4. on a $x = Vy$.

3.6 Pseudo–inverses

Il n'est possible de définir l'inverse A^{-1} d'une matrice A que si celle–ci est carrée et régulière. Si ce n'est pas le cas, nous allons essayer de définir une matrice $A^\dagger$, appelée *pseudo–inverse* de A. L'idée directrice dans la recherche de cette définition est que ce pseudo–inverse doit possèder les propriétés les plus semblables possibles à celles de l'inverse et que, bien sur, lorsque la matrice est carrée et régulière, on doit retrouver l'inverse habituel.

Commençons par le cas le plus simple. Soit A une matrice $n \times m$ de rang maximum, c'est–à–dire de rang $r = \min(n, m)$.

On appelle *pseudo–inverse* de A, la matrice $A^\dagger$ satisfaisant les quatre propriétes suivantes

1. $A^\dagger A A^\dagger = A^\dagger$,
2. $A A^\dagger A = A$,
3. $(A A^\dagger)^T = A A^\dagger$,
4. $(A^\dagger A)^T = A^\dagger A$.

Si $r = m \leq n$, alors $A^\dagger = (A^T A)^{-1} A^T$.

Si $r = n \leq m$, alors $A^\dagger = A^T (A A^T)^{-1}$.

On vérifiera facilement que ces matrices satisfont les quatre propriétés précédentes.

Considérons maintenant le cas où A n'est pas de rang maximum, c'est–à–dire que $r < \min(n, m)$. On peut alors définir le pseudo–inverse de A à l'aide de sa décomposition en valeurs singulières

$$A = U \Sigma V^T.$$

Dans ce cas, $\Sigma \in \mathbb{R}^{n \times m}$ et l'on a

$$\Sigma = \left(\begin{array}{ccc|c} \sigma_1 & & 0 & \\ & \ddots & & 0 \\ 0 & & \sigma_r & \\ \hline & 0 & & 0 \end{array} \right).$$

De même, $\Sigma^\dagger \in \mathbb{R}^{m \times n}$ et (à démontrer en exercice)

$$\Sigma^\dagger = \left(\begin{array}{ccc|c} \sigma_1^{-1} & & 0 & \\ & \ddots & & 0 \\ 0 & & \sigma_r^{-1} & \\ \hline & 0 & & 0 \end{array} \right).$$

Il s'en suit que

$$A^\dagger = V\Sigma^\dagger U^T.$$

On montrera en exercice que si $A = \sum_{i=1}^r \sigma_i u_i v_i^T$, alors $A^\dagger = \sum_{i=1}^r \sigma_i^{-1} v_i u_i^T$.

Le pseudo–inverse est relié à la solution d'un système linéaire au sens des moindres carrés par le Théorème suivant

Théorème 17
Soit $A \in \mathbb{R}^{n\times m}$ et $b \in \mathbb{R}^n$ et soit $x \in \mathbb{R}^m$ la solution au sens des moindres carrés du système $Ax = b$, c'est–à–dire que

$$\|Ax - b\|_2 = \min_{y\in\mathbb{R}^m} \|Ay - b\|_2.$$

Alors $x = A^\dagger b$.

Démonstration.

D'après ce qui a été vu dans la Section précédente

$$\begin{aligned} x &= Vy = V\begin{pmatrix} \widehat{y} \\ 0 \end{pmatrix} = V\begin{pmatrix} \widehat{\Sigma}^{-1}\widehat{c} \\ 0 \end{pmatrix} \\ &= V\begin{pmatrix} \widehat{\Sigma}^{-1} & 0 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} \widehat{c} \\ d \end{pmatrix} \\ &= V\Sigma^\dagger c = V\Sigma^\dagger U^T b = A^\dagger b. \ \blacksquare \end{aligned}$$

3.7 Matrices test

Afin de tester les diverses méthodes d'algèbre linéaires, il est nécessaire d'avoir à sa disposition un certain nombre de matrices test. Si l'on programme en MATLAB qui est un langage particulièrement adapté à l'algèbre matricielle on pourra utiliser les nombreuses matrices qui sont données dans la matrix toolbox de MATLAB. Autrement, on pourra considérer la collection Harwell–Boeing de matrices que l'on trouvera sur le web.

Voici une matrice intéressante

$$A_n = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 & 1 \\ a_1 & 1 & 1 & \cdots & 1 & 1 \\ a_1 & a_2 & 1 & \cdots & 1 & 1 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ a_1 & a_2 & a_3 & \cdots & a_{n-1} & 1 \end{pmatrix}.$$

Posons $B = (b_{ij}) = A^{-1}$; on a, pour $n \geq 2$,

$$\begin{aligned} b_{ij} &= 0, \quad j > i+1, \quad i < n-1 \\ &= 0, \quad j < i, \quad i \neq n \\ &= 1/(1-a_i), \quad j = i, \quad i \neq n \\ &= -1/(1-a_i), \quad j = i+1, \quad i \neq n \\ &= (a_{j-1} - a_j)/(1-a_j)(1-a_{j-1}), \quad i = n, \quad j \neq 1, n \\ &= -a_1/(1-a_1), \quad j = 1, \quad i = n \\ &= 1/(1-a_{n-1}), \quad j = n, \quad i = n. \end{aligned}$$

Le déterminant est égal à $\det A_n = (1-a_1)\cdots(1-a_{n-1})$. L'on peut donc fabriquer des matrices aussi voisines de la singularité que l'on désire.

Supposons que $\forall i, a_i \neq 0$. La matrice est donc régulière. Soit $x = (1, x_2, \ldots, x_n)^T$ le vecteur propre de A_n associé à la valeur propre λ. On a

$$\begin{aligned} 1 + x_2 + x_3 + \cdots + x_n &= \lambda \\ a_1 + x_2 + x_3 + \cdots + x_n &= \lambda x_2 \\ a_1 + a_2 x_2 + x_3 + \cdots + x_n &= \lambda x_3 \\ \vdots &\quad \vdots \\ a_1 + a_2 x_2 + a_3 x_3 + \cdots + x_n &= \lambda x_n. \end{aligned}$$

En soustrayant la première équation de la seconde, on obtient

$$x_2 = (\lambda - 1 + a_1)/\lambda.$$

En soustrayant la seconde équation de la troisième, on trouve

$$x_3 = (\lambda - 1 + a_2)x_2/\lambda$$

et ainsi de suite

$$x_i = (\lambda - 1 + a_{i-1})x_{i-1}/\lambda.$$

Soit $P_i(\lambda)$ le polynôme dont les racines sont $(1-a_1), \ldots, (1-a_i)$. Le polynôme caractéristique de A_n s'écrit

$$P(\lambda) = \lambda^n - \lambda^{n-1} - \lambda^{n-2}P_1(\lambda) - \cdots - \lambda P_{n-2}(\lambda) - P_{n-1}(\lambda).$$

Si l'un des a_i est nul, la colonne i est identique à la dernière et les lignes i et $i-1$ sont identiques. On supprime alors ces lignes et ces colonnes identiques et, pour trouver les valeurs propres et les vecteurs propres, on raisonne sur la matrice ainsi obtenue.

Une matrice tridiagonale intéressante est

$$\begin{pmatrix} \varepsilon & 1 & 0 & \cdots & 0 \\ -1 & \varepsilon & 1 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 1 \\ 0 & \cdots & \cdots & -1 & \varepsilon \end{pmatrix}.$$

Selon la valeur de ε, on peut obtenir des résultats qui varient du meilleur au pire.

Les matrices de Hilbert sont des prototypes de matrices mal conditionnées, puisque leur conditionnement κ_2 est de l'ordre de $\exp(3.5n)$. Elles sont symétriques définies positives et données par $a_{ij} = 1/(i+j-1)$ pour $i, j = 1, \ldots, n$. Les éléments d'une matrice de Hilbert sont définis par $a_{ij} = 1/(i+j-1)$. Les éléments b_{ij} de l'inverse d'une matrice de Hilbert de dimension n sont donnés par

$$b_{ij} = \frac{(-1)^{i+j}(n+i-1)!(n+j-1)!}{(i+j-1)[(i-1)!(j-1)!]^2(n-i)!(n-j)!}.$$

La matrice de Pascal est donnée par

$$\begin{aligned} a_{1j} &= a_{j1} = 1/k, \quad j = 1, \ldots, n \\ a_{ij} &= a_{i-1,j} + a_{i,j-1}, \quad i, j = 2, \ldots, n \end{aligned}$$

où k est un entier positif. De manière équivalente, on a

$$a_{ij} = \frac{1}{k} \frac{(i+j-2)!}{(i-1)!(j-1)!}.$$

Les éléments de A^{-1} sont des entiers et le déterminant de A vaut k^n. C'est une matrice symétrique définie positive. Le conditionnement est de l'ordre de $16^n/n\pi$.

Chapitre 4

La méthode de Gauss

Pour passer d'un système quelconque à un système équivalent (c'est–à–dire ayant la même solution) ayant une matrice triangulaire supérieure, on voit qu'il est nécessaire d'éliminer x_1 de toutes les équations à partir de la seconde, puis d'éliminer x_2 de toutes les équations à partir de la troisième et ainsi de suite. Pour cela, on peut tirer x_1 de la première équation (ce qui signifie exprimer x_1 en fonction des autres inconnues à l'aide de la première équation du système) puis remplaçer x_1 par son expression dans toutes les autres équations. Ensuite, on tirera x_2 de la seconde équation (ce qui signifie que, à l'aide de la seconde équation, on exprimera x_2 en fonction de $x_3, \ldots, x_n$) et l'on reportera cette expression dans toutes les équations à partir de la troisième et ainsi de suite. Mais si l'on réfléchit bien, on voit que ces substitutions sont difficiles à systématiser lors de l'écriture d'un algorithme.

Il est une façon plus commode de réaliser ces éliminations : c'est en faisant des combinaisons linéaires entre les équations du système et c'est exactement ainsi que la méthode de Karl Friedrich Gauss (1777–1855) va procéder. Quand cette phase de triangularisation est terminée, on est en présence d'un système avec une matrice triangulaire supérieure. La méthode de Gauss comporte donc une seconde phase qui est la résolution de ce système par les formules données au début de ce chapitre.

4.1 L'algorithme

Posons $A = (a_{ij}) = A^{(1)} = (a_{ij}^{(1)})$ et $b = (b_i) = b^{(1)} = (b_i^{(1)})$. Notre système $Ax = b$ s'écrit

$$\begin{aligned} a_{11}^{(1)}x_1 + a_{12}^{(1)}x_2 + \cdots + a_{1n}^{(1)}x_n &= b_1^{(1)} \\ a_{21}^{(1)}x_1 + a_{22}^{(1)}x_2 + \cdots + a_{2n}^{(1)}x_n &= b_2^{(1)} \\ a_{31}^{(1)}x_1 + a_{32}^{(1)}x_2 + \cdots + a_{3n}^{(1)}x_n &= b_3^{(1)} \end{aligned}$$

$$\begin{array}{rcl} & \vdots & \\ a_{n1}^{(1)}x_1 + a_{n2}^{(1)}x_2 + \cdots + a_{nn}^{(1)}x_n & = & b_n^{(1)}. \end{array}$$

Supposons que $a_{11}^{(1)}$ ne soit pas nul (on verra plus loin ce qu'il y a lieu de faire si $a_{11}^{(1)} = 0$). Divisons la première équation du système par $a_{11}^{(1)}$. Le coefficient de x_1 devient alors égal à 1 dans cette première équation. Multiplions la ensuite par $a_{21}^{(1)}$. Le coefficient de x_1 dans la première équation est maintenant égal à $a_{21}^{(1)}$. Soustrayons donc cette équation de la seconde équation du système, ce qui ne modifie pas sa solution. On voit que le coefficient de x_1 devient nul et donc que x_1 a disparu de la seconde équation. Reprenons maintenant la première équation (toujours divisée par $a_{11}^{(1)}$), multiplions la par $a_{13}^{(1)}$ et soustrayons la de la troisème équation. On voit que x_1 disparaît ainsi de la troisième équation. On continue de la même manière jusqu'à la dernière équation pour en faire disparaître x_1.

Résumons ces opérations. On considére d'abord la première équation divisée par $a_{11}^{(1)}$. Pour faire disparaître x_1 de la i-ème équation, pour $i = 2, \ldots, n$, on multiplie cette première équation modifiée par $a_{i1}^{(1)}$. Elle devient donc

$$\frac{a_{i1}^{(1)}}{a_{11}^{(1)}}a_{11}^{(1)}x_1 + \frac{a_{i1}^{(1)}}{a_{11}^{(1)}}a_{12}^{(1)}x_2 + \cdots + \frac{a_{i1}^{(1)}}{a_{11}^{(1)}}a_{1n}^{(1)}x_n = \frac{a_{i1}^{(1)}}{a_{11}^{(1)}}b_1^{(1)}.$$

On soustrait cette équation de l'équation i du système

$$a_{i1}^{(1)}x_1 + a_{i2}^{(1)}x_2 + \cdots + a_{in}^{(1)}x_n = b_1^{(1)}.$$

On obtient comme nouvelle équation i

$$\left(a_{i1}^{(1)} - \frac{a_{i1}^{(1)}}{a_{11}^{(1)}}a_{11}^{(1)}\right)x_1 + \left(a_{i2}^{(1)} - \frac{a_{i1}^{(1)}}{a_{11}^{(1)}}a_{12}^{(1)}\right)x_2 + \cdots + \left(a_{in}^{(1)} - \frac{a_{i1}^{(1)}}{a_{11}^{(1)}}a_{1n}^{(1)}\right)x_n = b_i^{(1)} - \frac{a_{i1}^{(1)}}{a_{11}^{(1)}}b_1^{(1)}.$$

On pose, pour $i = 2, \ldots, n$,

$$\begin{array}{rcl} a_{ij}^{(2)} & = & a_{ij}^{(1)} - \dfrac{a_{i1}^{(1)}}{a_{11}^{(1)}}a_{1j}^{(1)}, \quad j = 1, \ldots, n \\ b_i^{(2)} & = & b_i^{(2)} - \dfrac{a_{i1}^{(1)}}{a_{11}^{(1)}}b_i^{(1)}. \end{array}$$

On voit que $a_{11}^{(2)} = 0$. On a ainsi obtenu le nouveau système, équivalent au système initial

$$\begin{array}{rcl} a_{11}^{(2)}x_1 + a_{12}^{(2)}x_2 + \cdots + a_{1n}^{(2)}x_n & = & b_1^{(2)} \\ 0x_1 + a_{22}^{(2)}x_2 + \cdots + a_{2n}^{(2)}x_n & = & b_2^{(2)} \\ 0x_1 + a_{32}^{(2)}x_2 + \cdots + a_{3n}^{(2)}x_n & = & b_3^{(2)} \\ & \vdots & \\ 0x_1 + a_{n2}^{(2)}x_2 + \cdots + a_{nn}^{(2)}x_n & = & b_n^{(2)} \end{array}$$

avec $a_{1j}^{(2)} = a_{1j}^{(1)}$ pour $j = 1, \ldots, n$ et $b_1^{(2)} = b_1^{(1)}$.

Nous allons maintenant, dans ce nouveau système, effectuer l'élimination de x_2 de toutes les équations à partir de la troisième. Pour cela, divisons la seconde équation par $a_{22}^{(2)}$ (en le supposant non nul). Puis multiplions cette seconde équation par $a_{32}^{(2)}$ et soustrayons la de la troisième équation : nous avons éliminé x_2 de cette troisième équation. Continuons le procédé jusqu'à la dernière équation dont on élimine x_2 en en soustrayant la seconde équation (toujours divisée par $a_{22}^{(2)}$) multipliée par $a_{n2}^{(2)}$. On obtient le système équivalent

$$\begin{aligned} a_{11}^{(3)}x_1 + a_{12}^{(3)}x_2 + a_{13}^{(3)}x_3 + \cdots + a_{1n}^{(3)}x_n &= b_1^{(3)} \\ 0x_1 + a_{22}^{(3)}x_2 + a_{23}^{(3)}x_3 + \cdots + a_{2n}^{(3)}x_n &= b_2^{(3)} \\ 0x_1 + 0x_2 + a_{33}^{(3)}x_3 + \cdots + a_{3n}^{(3)}x_n &= b_3^{(3)} \\ &\vdots \\ 0x_1 + 0x_2 + a_{n3}^{(3)}x_3 + \cdots + a_{nn}^{(3)}x_n &= b_n^{(3)} \end{aligned}$$

avec $a_{1j}^{(3)} = a_{1j}^{(2)}$ pour $j = 1, \ldots, n$, $b_1^{(3)} = b_1^{(2)}$, $a_{2j}^{(3)} = a_{2j}^{(2)}$ pour $j = 1, \ldots, n$ et $b_2^{(2)} = b_2^{(2)}$.

On voit, qu'à la k-ème étape de ce procédé on obtient le système $A^{(k)}x = b^{(k)}$ donné par

$$A^{(k)} = \begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1k}^{(1)} & \cdots & a_{1n}^{(1)} \\ & a_{22}^{(2)} & \cdots & a_{2k}^{(2)} & \cdots & a_{2n}^{(2)} \\ & & \ddots & \vdots & & \vdots \\ & & & a_{kk}^{(k)} & \cdots & a_{kn}^{(k)} \\ & & & a_{k+1,k}^{(k)} & \cdots & a_{k+1,n}^{(k)} \\ & & & \vdots & & \vdots \\ & & & a_{n,k}^{(k)} & \cdots & a_{n,n}^{(k)} \end{pmatrix} \quad b^{(k)} = \begin{pmatrix} b_1^{(1)} \\ b_2^{(2)} \\ \vdots \\ b_k^{(k)} \\ b_{k+1}^{(k)} \\ \vdots \\ b_n^{(k)} \end{pmatrix}. \quad (4.1)$$

Dans ce système, tous les éléments en dessous de la diagonale principale dans les k premières colonnes sont nuls.

Pour passer du système k au système $k+1$, il faut remplacer les éléments $a_{ik}^{(k)}$ par des zéros pour $i = k+1, \ldots, n$. Pour cela, on commence par diviser l'équation k par $a_{kk}^{(k)}$ qui, bien sûr, ne doit pas être nul (on verra plus loin ce qu'il y a alors lieu de faire). Puis on multiplie successivement cette équation par $a_{ik}^{(k)}$ pour $i = k+1, \ldots, n$ et on la soustrait de l'ancienne équation i. Le nouvel élément $a_{ik}^{(k+1)}$ devient alors nul. Par conséquent, on passe du système k au système $k+1$ par les règles suivantes, pour $k = 1, \ldots, n-1$

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - \frac{a_{ik}^{(k)} a_{kj}^{(k)}}{a_{kk}^{(k)}}, \quad i, j = k+1, \ldots, n \quad (4.2)$$

$$b_i^{(k+1)} = b_i^{(k)} - \frac{a_{ik}^{(k)} b_k^{(k)}}{a_{kk}^{(k)}}, \quad i = k+1, \ldots, n \tag{4.3}$$

les autres termes restant inchangés.

On effectue les transformations (4.2) et (4.3) de $k = 1$ à $k = n-1$. La matrice $A^{(n)}$ du système $A^{(n)}x = b^{(n)}$ est triangulaire supérieure. Cette phase de triangularisation se résume donc à

for $k = 1, \ldots, n-1$
  Passage de $A^{(k)}$ *à* $A^{(k+1)}$
  for $i = 1, \ldots, k$
    Équations 1 *à* k *inchangées*
  end
  for $i = k+1, \ldots, n$
    Traitement des équations $k+1$ *à* n
  end
end

On résoud ensuite directement le système triangulaire supérieur ainsi obtenu.

La méthode de Gauss se décompose donc en deux phases : une phase de triangularisation du système suivie d'une phase de résolution du système triangulaire.

4.2 Mise en œuvre

On voit que les formules qui permettent de passer de $A^{(k)}$ à $A^{(k+1)}$ et celles qui permettent de passer de $b^{(k)}$ à $b^{(k+1)}$ se ressemblent fortement. On peut les réunir en ajoutant le vecteur $b^{(k)}$ comme $(n+1)$-ème colonne de la matrice $A^{(k)}$ et en faisant aller, dans les formules (4.2), l'indice j de $k+1$ à $n+1$ (au lieu de n).

D'autre part, pour programmer cet algorithme, il n'est pas nécessaire de disposer d'un tableau à 3 indices. En effet, dès que $a_{ij}^{(k+1)}$ à été calculé, on ne se sert plus de $a_{ij}^{(k)}$. On peut donc tout simplement remplacer $a_{ij}^{(k)}$ par le nouvel élément $a_{ij}^{(k+1)}$. Cette stratégie à l'avantage supplémentaire que les éléments des k premières équations sont automatiques conservés. Enfin, puisque ces éléments ne serviront pas dans la résolution du système triangulaire, il n'est pas nécessaire de mettre à zéro les éléments en dessous de la diagonale principale. Du point de vue programmation, la phase de triangularisation de l'algorithme de Gauss se réduit donc à

for $k = 1, \ldots, n-1$

```
    for i = k + 1, ..., n
        r = a_ik^(k) / a_kk^(k)
        for j = k + 1, ..., n + 1
            a_ij^(k+1) = a_ij^(k) − r a_kj^(k)
        end
    end
end
```

La résolution du système triangulaire s'effectue de la façon suivante (l'indice supérieur à été supprimé)

```
x_n = a_{n,n+1} / a_nn
for i = n − 1, n − 2, ..., 1
    s = a_{i,n+1}
    for j = i + 1, ..., n
        s = s − a_ij x_j
    end
x_i = s / a_ii
end
```

Les nombres $a_{kk}^{(k)}$, pour $k = 1, \ldots, n$, sont appelés les *pivots*. On voit que l'on doit les supposer tous différents de zéro. On rencontre les pivots $a_{11}^{(1)}, \ldots, a_{n-1,n-1}^{(n-1)}$ dans la phase de triangularisation, mais on ne rencontre le pivot $a_{nn}^{(n)}$ que dans la résolution du système triangulaire. Si la matrice A est singulière, alors il existera toujours un indice $k \in \{1, \ldots, n\}$ tel que $a_{kk}^{(k)} = 0$ (ce qui est normal, puisqu'il doit se produire, dans l'algorithme, quelque chose qui nous empêche de résoudre le système). Par contre, la réciproque n'est pas vraie. Il peut exister un indice k tel que le pivot $a_{kk}^{(k)}$ soit nul sans pour autant que la matrice A soit singulière. Quand on rencontre un pivot nul dans l'algorithme de Gauss il faut donc être capable de reconnaître si cela signifie que la matrice est singulière ou non. Si la matrice n'est pas singulière, il faut être capable de continuer la résolution du système. C'est la technique du *pivotage* qui permet de répondre à ces questions.

Si l'un des pivots est nul, on pourra effectuer une permutation entre l'équation ligne k et l'une des équations p du système $A^{(k)}x = b^{(k)}$ (ne pas oublier de permuter aussi les seconds membres correspondants). Pour ne pas détruire le début de triangularisation déjà effectué, on prendra $p > k$. On recherche donc $p \geq k$ tel que $a_{pk}^{(k)}$ soit différent de zéro puis l'on permute les équations p et k du système. Si, pour $p = k, \ldots, n$, tous les nombres $a_{pk}^{(k)}$ sont nuls, alors cela signifie que la matrice A est singulière. On arrête donc les calculs, le système ne pouvant être résolu.

Si le pivot $a_{kk}^{(k)}$ n'est pas nul mais est proche de zéro, alors l'algorithme pourra devenir numériquement instable. Nous avons vu, en effet, qu'il fallait se méfier

des quantités voisines de zéro car elles pouvaient provenir de la différence de deux nombres voisins et donc être entachées d'une importante erreur de cancellation. Le choix des pivots est donc essentiel en ce qui concerne la stabilité de l'algorithme de Gauss. Il ne faut pas que les quantités $|a_{kk}^{(k)}|$ soient trop petites pour que l'algorithme soit stable. On effectuera donc un pivotage même si $|a_{kk}^{(k)}|$ n'est pas nul.

Donnons un exemple. Nous considérons le système

$$\begin{aligned} \varepsilon x_1 + x_2 &= 1 \\ x_1 + x_2 &= 0. \end{aligned}$$

Dans la méthode de Gauss, on divise la première équation par ε puis on la soustrait de la seconde. On obtient

$$(1 - \varepsilon^{-1})x_2 = -\varepsilon^{-1}$$

et donc $x_2 = 1/(1-\varepsilon)$. En reportant dans la première équation on trouve $x_1 = -1/(1-\varepsilon)$. Si ε est tel que $fl(1-\varepsilon) = 1$, alors on voit que $x_2 = 1$, résultat correct à la précision machine près. Mais, en reportant dans la première équation, on obtient $fl(x_1) = 0$, ce qui est faux puisque x_1 est voisin de -1.

Par contre, si l'on effectue un pivotage des équations, c'est–à–dire si l'on considère le système

$$\begin{aligned} x_1 + x_2 &= 0 \\ \varepsilon x_1 + x_2 &= 1 \end{aligned}$$

alors on obtient comme seconde équation $(1-\varepsilon)x_2 = 1$ ce qui donne $fl(x_2) = 1$, résultat exact à la précision machine près. En reportant dans la première équation, on a $fl(x_1) = -fl(x_2) = -1$.

Pour le pivotage, il y a, en fait, deux façons de procéder

- **pivotage partiel**

  Soit $a_{pk}^{(k)}$ le nombre tel que

  $$\left|a_{pk}^{(k)}\right| = \max_{k \le i \le n} \left|a_{ik}^{(k)}\right|.$$

  On permute alors les équations k et p du k-ème système,

- **pivotage total**

  Soit $a_{pq}^{(k)}$ le nombre tel que

  $$\left|a_{pq}^{(k)}\right| = \max_{k \le i,j \le n} \left|a_{ij}^{(k)}\right|.$$

  On permute alors les équations p et k et les colonnes q et k du k-ème système. On voit que la permutation des colonnes conduit à un changement de numérotation des composantes du vecteur x. Il sera donc

nécessaire de remettre dans le bon ordre les composantes après résolution du système triangulaire supérieur. Cela s'effectue grâce à un vecteur d'indices γ (adressage indirect). Au début de l'algorithme, il n'y a aucune permutation et l'on pose donc $\gamma_i = i$ pour $i = 1, \ldots, n$. Si, à l'étape k, on intervertit les colonnes k et q alors on intervertira les composantes γ_k et γ_q dans le vecteur γ. Soit y le vecteur obtenu par résolution du système triangulaire supérieur à la fin de l'algorithme de Gauss. On aura $y_i = x_{\gamma_i}$ pour $i = 1, \ldots, n$.

On voit, dans ce qui précède, que l'on a effectué le pivotage même si le pivot $a_{kk}^{(k)}$ n'est pas nul. Cela permet en effet d'éviter de diviser, le cas échéant, par un pivot voisin de zéro et pouvant être entaché d'une importante erreur de cancellation. On évite ainsi, dans une certaine mesure, une propagation des erreurs dues à l'arithmétique de l'ordinateur. De plus, si, avec le pivotage partiel ou total, le pivot trouvé est nul alors la matrice est singulière alors que, sans pivotage, un pivot nul ne signifie pas forcément que la matrice soit singulière.

Il ne faudra pas oublier, dans tous les cas, de tester si $a_{nn}^{(n)}$ est nul ou non.

4.3 Nombre d'opérations

Calculons maintenant le nombre d'opérations arithmétiques nécessaires pour résoudre un système linéaire de dimension n par la méthode de Gauss.

Pour passer de $A^{(1)}$ à $A^{(2)}$, on effectue $n-1$ divisions. Pour passer de $A^{(2)}$ à $A^{(3)}$, il faut faire $n-2$ divisions, et ainsi de suite. Le nombre total de divisions est donc

$$(n-1) + (n-2) + \cdots + 1 = n(n-1)/2.$$

D'après les règles de l'algorithme, on voit que le nombre de multiplications est le même que le nombre d'additions. Pour passer de $A^{(k)}$ à $A^{(k+1)}$, l'indice i varie de $k+1$ à n et l'indice j de $k+1$ à $n+1$ (en considérant le second membre comme une colonne supplémentaire de la matrice). Le passage du système k au système $k+1$ nécessite donc $(n-k)(n-k+1)$ multiplications et autant d'additions. Au total on a (on rappelle que $\sum_{k=1}^{n-1} k^2 = n(n-1)(2n-1)/6$)

$$\begin{aligned}
\sum_{k=1}^{n-1}(n-k)(n-k+1) &= \sum_{k=1}^{n-1}(n^2+n-2kn+k^2-k) \\
&= n^2(n-1) + n(n-1) - 2n\sum_{k=1}^{n-1} k + \sum_{k=1}^{n-1}(k^2-k) \\
&= n(n-1) + n(n-1)(2n-1)/6 - n(n-1)/2 \\
&= n(n^2-1)/3.
\end{aligned}$$

Pour la phase de résolution du système triangulaire, on doit effectuer n divisions, $n(n-1)/2$ multiplications et autant d'additions.

Au total, la méthode de Gauss nécessite donc $n(n+1)/2$ divisions, $n(n-1)(2n+5)/6$ multiplications et autant d'additions. Soit au total de l'ordre de $2n^3/3$ opérations (pour $n = 10$ cela fait 700 au lieu de 3.10^9 avec les déterminants).

L'intérêt de ce résultat est finalement de montrer que le temps de calcul est proportionnel à $2n^3/3$. Cela signifie que si l'on multiplie la dimension d'un système par 2, le temps de calcul sera multiplié par $2^3 = 8$.

Pour de grands systèmes linéaires, le temps de calcul est toujours prohibitif. Ainsi, pour un ordinateur effectuant 10^7 opérations par seconde, la résolution d'un système linéaire de dimension n par la méthode de Gauss nécessite les temps suivants

n	temps
20000	6 jours
50000	96 jours
100000	771 jours
500000	264 ans
1000000	2114 ans

Dans certaines applications, on doit résoudre un système linéaire avec toujours la même matrice mais plusieurs seconds membres différents. Il n'est pas nécessaire de recommencer tous les calculs. La phase de triangularisation reste inchangée. Il suffit, au lieu d'ajouter à la matrice une colonne supplémentaire qui contient le second membre, d'ajouter tous les seconds membres. Bien entendu, après la phase de triangularisation, on résoudra séparément chacun des systèmes par remontée.

En particulier, le calcul de l'inverse de A s'effectue en résolvant les n systèmes linéaires

$$Ax_i = e_i, \quad i = 1, \ldots, n$$

où les e_i sont les vecteurs de base canonique de $\mathbb{R}^n$. Les vecteurs x_i solutions de ces systèmes sont les colonnes de A^{-1}. Le calcul de A^{-1} nécessite de l'ordre de $4n^3/3$ opérations arithmétiques.

4.4 Interprétation matricielle

Nous allons maintenant interpréter la méthode de Gauss en termes matriciels. Considérons les matrices

$$L^{(k)} = \begin{pmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & -l_{k+1,k} & 1 & & \\ & & \vdots & & \ddots & \\ & & -l_{n,k} & \cdots & \cdots & 1 \end{pmatrix}$$

avec une diagonale unité, $l_{i,k} = a_{ik}^{(k)}/a_{kk}^{(k)}$ et tous les autres éléments nuls. Il est facile de voir (c'est un produit matriciel) que l'on passe du système k au système $k+1$ par

$$A^{(k+1)} = L^{(k)}A^{(k)}, \qquad b^{(k+1)} = L^{(k)}b^{(k)}, \qquad k = 1, \ldots, n-1.$$

En effet, considérons le vecteur suivant dont les k premières composantes sont nulles

$$l_k = (0, \ldots, 0, l_{k+1,k}, \ldots, l_{n,k})^T.$$

On a $L^{(k)} = I - l_k e_k^T$ où e_k est le k-ème vecteur de la base canonique de $\mathbb{R}^n$. D'où $A^{(k+1)} = A^{(k)} - l_k e_k^T A^{(k)}$. Or le produit de e_k^T par la j-ème colonne de $A^{(k)}$ est égal à $a_{kj}^{(k)}$ et le produit de e_k^T par $b^{(k)}$ donne $b_k^{(k)}$. Quand on passe du système k au système $k+1$, le i-ème élément de la colonne j est remplacé par $a_{ij}^{(k)} - l_{ik}a_{kj}^{(k)}$. On voit donc que les $k-1$ premiers éléments de cette colonne ne sont pas modifiés, que le k-ème devient nul puisqu'il est égal à $a_{kj}^{(k)} - a_{kj}^{(k)}$ et que les autres sont bien données par la formule $a_{ij}^{(k)} - a_{ik}^{(k)}a_{kj}^{(k)}/a_{kk}^{(k)}$. On passe de façon similaire de $b^{(k)}$ à $b^{(k+1)}$.

On a donc, par induction,

$$A^{(n)} = L^{(n-1)} \cdots L^{(1)}A, \quad b^{(n)} = L^{(n-1)} \cdots L^{(1)}b. \tag{4.4}$$

Puisque les matrices $L^{(k)}$ n'ont que des 1 sur leurs diagonales, elles sont inversibles et l'on a

$$(L^{(k)})^{-1} = \begin{pmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & l_{k+1,k} & 1 & & \\ & & \vdots & & \ddots & \\ & & l_{n,k} & \cdots & \cdots & 1 \end{pmatrix}.$$

On peut soit vérifier ce résultat directement soit le démontrer. Montrons que $(L^{(k)})^{-1} = I + l_k e_k^T$. En effet

$$\begin{aligned} (L^{(k)})^{-1}L^{(k)} &= (I + l_k e_k^T)(I - l_k e_k^T) \\ &= I - l_k e_k^T l_k e_k^T. \end{aligned}$$

Mais $e_k^T l_k = 0$, puisque la k-ème composant du vecteur l_k est nulle, ce qui démontre le résultat.

Posons $L = (L^{(n-1)} \cdots L^{(1)})^{-1} = (L^{(1)})^{-1} \cdots (L^{(n-1)})^{-1}$. Il est facile de voir que L est triangulaire unité à diagonale unité, la lettre L qui la désigne est l'initiale de l'anglais "lower" (inférieure)

$$L = \begin{pmatrix} 1 & & & & \\ l_{21} & 1 & & & \\ \vdots & \vdots & \ddots & & \\ \vdots & \vdots & & \ddots & \\ l_{n1} & l_{n2} & \cdots & l_{n,n-1} & 1 \end{pmatrix}.$$

En effet $(L^{(k)})^{-1}(L^{(k+1)})^{-1} = I + l_k e_k^T + l_{k+1} e_{k+1}^T$ et on a donc, de nouveau par induction,

$$L = (L^{(1)})^{-1} \cdots (L^{(n-1)})^{-1} = I + l_1 e_1^T + \cdots + l_{n-1} e_{n-1}^T.$$

Puisque la matrice $A^{(n)}$ joue un rôle particulier dans la méthode de Gauss, donnons lui un nom particulier. Nous la noterons U qui est l'initiale de l'anglais "upper" (supérieure). De même, nous poserons $y = b^{(n)}$. On voit que les relations (4.4) s'écrivent donc

$$A = LU, \qquad b = Ly.$$

Notre système linéaire $Ax = b$ s'écrit donc $LUx = b$. Si nous posons $y = Ux$, il devient $Ly = b$. Par conséquent, la méthode de Gauss consiste à mettre la matrice A sous forme d'un produit de deux matrices $A = LU$ où L est une matrice triangulaire inférieure avec des 1 sur la diagonale (on dira *à diagonale unité*) et U une matrice triangulaire supérieure. On parle de *factorisation* ou de *décomposition* de Gauss. On résoud d'abord $Ly = b$ puis $Ux = y$. La phase de triangularisation de la méthode de Gauss consiste donc à calculer simultanément la matrice U et le vecteur y. Puis on résoud le système triangulaire $Ux = y$. En fait, la matrice L n'est jamais calculée explicitement.

Donnons maintenant un résultat théorique

Théorème 18
Supposons que A soit régulière. Alors, si elle existe, la décomposition $A = LU$ est unique.

Démonstration.

Supposons que deux décompositions existent

$$A = L_1 U_1 = L_2 U_2.$$

Les matrices L_1 et L_2 sont inversibles puisque leurs déterminants valent 1. Puisque A est inversible alors U_1 et U_2 le sont également. On a

$$L_2^{-1}L_1 = U_2U_1^{-1}.$$

L'inverse d'un matrice triangulaire est une matrice triangulaire de même nature. L'inverse d'une matrice triangulaire à diagonale unité est aussi à diagonale unité. Le produit de deux matrices triangulaires de mêmes natures est une matrice triangulaire de même nature. Donc, $L_2^{-1}L_1$ est triangulaire inférieure avec des 1 sur la diagonale et $U_2U_1^{-1}$ est triangulaire supérieure. Ces deux matrices ne peuvent donc être égales qu'à I d'où

$$L_2^{-1}L_1 = I = U_2U_1^{-1}$$

ce qui démontre que $L_1 = L_2$ et $U_1 = U_2$. ∎

Une conséquence le la décomposition $A = LU$ est le

Corollaire 1

$$\det A = \det L \cdot \det U = \prod_{k=1}^{n} a_{kk}^{(k)}.$$

Étudions maintenant l'existence d'une telle décomposition. On a le

Théorème 19
Supposons que A soit régulière. Alors une condition nécessaire et suffisante pour que la décomposition $A = LU$ existe est que tous les mineurs fondamentaux de A soient non nuls.

Démonstration.

Démontrons d'abord que la condition est nécessaire. Nous noterons A_k la matrice formée par les k premières lignes et les k premières colonnes de A. Une notation analogue sera utilisée pour toutes les autres matrices.

On peut vérifier facilement que $A_k = L_kU_k$. Puisque A est régulière, alors, d'après le Corollaire précédent, $a_{kk}^{(k)} \neq 0$ pour $k = 1, \ldots, n$. D'où

$$\det A_k = \det L_k \cdot \det U_k = \prod_{i=1}^{k} a_{ii}^{(i)}$$

ce qui démontre que tous les mineurs fondamentaux de A sont non nuls.

Montrons maintenant que la condition est suffisante. On suppose que tous les mineurs fondamentaux de A sont différents de zéro et l'on va montrer que l'on peut mener à bien la décomposition $A = LU$. On a $a_{11} = a_{11}^{(1)} = \det A_1 \neq 0$. On peut donc effectuer la première étape de la méthode de Gauss et calculer

le système $A^{(2)}x = b^{(2)}$. Faisons maintenant un raisonnement par récurrence et supposons que nous ayons pu obtenir le système $A^{(k)}x = b^{(k)}$. On a

$$A^{(k)} = L^{(k-1)} \cdots L^{(1)} A$$

et donc

$$\det A_k^{(k)} = \det L_k^{(k-1)} \cdots \det L_k^{(1)} \cdot \det A_k = \prod_{i=1}^{k} a_{ii}^{(i)} \neq 0.$$

Par conséquent $a_{kk}^{(k)} \neq 0$ et l'on peut donc construire le système $A^{(k+1)}x = b^{(k+1)}$. ∎

La méthode de Gauss avec pivotage partiel ou total peut s'interpréter matriciellement de manière analogue. On montre que le pivotage partiel consiste à considérer le système $PAx = Pb$, où P est une matrice de permutation, puis à appliquer la décomposition à la matrice PA. On a donc $LUx = Pb$. On pose $Ly = Pb$ et le système devient $Ux = y$. Dans la méthode de Gauss avec pivotage total, on considère le système $PAQ^{-1}z = Pb$, où P et Q sont des matrices de permutation. On a $PAQ^{-1}z = Pb$ avec $z = Qx$ et l'on effectue la décomposition $PAQ^{-1} = LU$. Le système s'écrit donc $LUz = Pb$. On pose $Uz = y$ et l'on a $Ly = Pb$. On résoud donc ce dernier système pour obtenir y, puis on calcule z par remontée dans le système triangulaire $Uz = y$ et l'on obtient finalement x par $x = Q^{-1}z$. Des détails sont donnés dans le livre de Brezinski.

4.5 Le problème du remplissage

Ça n'est pas parce que la matrice A est creuse que la matrice U le sera également. Prenons, par exemple, la matrice creuse

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

Son inverse est une matrice pleine

$$A^{-1} = \begin{pmatrix} 5 & 4 & 3 & 2 & 1 \\ 4 & 4 & 3 & 2 & 1 \\ 3 & 3 & 3 & 2 & 1 \\ 2 & 2 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

Un autre exemple intéressant est le suivant. Soit la matrice

$$A = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 7 & 1 & 0 & 0 & 0 & 0 & 0 \\ 6 & 0 & 1 & 0 & 0 & 0 & 0 \\ 5 & 0 & 0 & 1 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 & 1 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 1 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Avec la méthode de Gauss, on obtient

$$U = \begin{pmatrix} 7.0000 & 1.0000 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.8571 & 3.0000 & 4.0000 & 5.0000 & 6.0000 & 7.0000 \\ 0 & 0 & 2.3846 & 1.8462 & 2.3077 & 2.7692 & 3.2308 \\ 0 & 0 & 0 & 1.6452 & 0.8065 & 0.9677 & 1.1290 \\ 0 & 0 & 0 & 0 & 1.3922 & 0.4706 & 0.5490 \\ 0 & 0 & 0 & 0 & 0 & 1.2535 & 0.2958 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.1573 \end{pmatrix}.$$

Par contre si, avant d'effectuer la décomposition de Gauss de A, nous intervertissons la numérotation des équations (la première devenant la dernière, la seconde prenant la place de l'avant–dernière, et ainsi de suite) et si nous intervertissons de même la numérotation des colonnes (ce qui entraîne celle des inconnues), alors nous obtenons

$$U = \begin{pmatrix} 7.0000 & 6.0000 & 5.0000 & 4.0000 & 3.0000 & 2.0000 & 1.0000 \\ 0 & 1.0000 & 0 & 0 & 0 & 0 & 3.0000 \\ 0 & 0 & 1.0000 & 0 & 0 & 0 & 4.0000 \\ 0 & 0 & 0 & 1.0000 & 0 & 0 & 5.0000 \\ 0 & 0 & 0 & 0 & 1.0000 & 0 & 6.0000 \\ 0 & 0 & 0 & 0 & 0 & 1.0000 & 7.0000 \\ 0 & 0 & 0 & 0 & 0 & 0 & 14.7143 \end{pmatrix}.$$

On voit que la structure de cette nouvelle matrice U est beaucoup plus creuse que celle de la première. L'ordonnancement des équations et des inconnues afin d'obtenir une structure de U la plus creuse possible est un problème d'une grande importance mais difficile à résoudre (on démontre qu'il ne peut être résolu qu'avec des algorithmes dont le nombre d'opérations croît plus vite que n'importe quelle puissance de la dimension). Sa solution fait appel à la théorie des graphes et dépasse le cadre de cet ouvrage.

4.6 Variantes de la méthode de Gauss

Nous allons maintenant proposer des variantes qui permettent d'améliorer la méthode de Gauss dans des cas particuliers.

4.6.1 Systèmes tridiagonaux

Étudions le cas particulier d'une matrice tridiagonale de la forme

$$A = \begin{pmatrix} a_1 & b_1 & & & \\ c_2 & a_2 & b_2 & & \\ & \ddots & \ddots & \ddots & \\ & & c_{n-1} & a_{n-1} & b_{n-1} \\ & & & c_n & a_n \end{pmatrix}.$$

Sa factorisation LU est donnée par

$$L = \begin{pmatrix} 1 & & & & \\ l_2 & 1 & & & \\ & \ddots & \ddots & & \\ & & l_{n-1} & 1 & \\ & & & l_n & 1 \end{pmatrix}, \quad U = \begin{pmatrix} u_1 & b_1 & & & \\ & u_2 & b_2 & & \\ & & \ddots & \ddots & \\ & & & u_{n-1} & b_{n-1} \\ & & & & u_n \end{pmatrix}$$

avec $u_1 = a_1$ et, pour $i = 2, \ldots, n$,

$$\begin{aligned} l_i &= c_i/u_{i-1} \\ u_i &= a_i - l_i b_{i-1}. \end{aligned}$$

4.6.2 La méthode de Gauss–Jordan

La méthode de Gauss–Jordan est une variante de la méthode de Gauss. Au lieu, à l'étape k, d'éliminer l'inconnue x_k seulement des équations $k+1$ à n, on l'élimine de toutes les équations sauf de la $k$ième. Dans cette variante, les matrices $A^{(k)}$ sont donc de la forme

$$A^{(k)} = \begin{pmatrix} a_{11}^{(k)} & & & a_{1k}^{(k)} & \cdots & a_{1n}^{(k)} & a_{1,n+1}^{(k)} \\ & \ddots & & \vdots & & \vdots & \vdots \\ & & \ddots & a_{k-1,k}^{(k)} & \cdots & a_{k-1,n}^{(k)} & a_{k-1,n+1}^{(k)} \\ & & & a_{kk}^{(k)} & \cdots & a_{kn}^{(k)} & a_{k,n+1}^{(k)} \\ & & & \vdots & & \vdots & \vdots \\ & & & a_{nk}^{(k)} & \cdots & a_{nn}^{(k)} & a_{n,n+1}^{(k)} \end{pmatrix}.$$

Les éléments hors de la diagonale principale dans les $k-1$ premières colonnes sont tous nuls.

On passe du système k au système $k+1$ à l'aide des relations

$$\begin{aligned} a_{ij}^{(k+1)} &= a_{ij}^{(k)}, \quad i = 1, \ldots, k; \quad j = 1, \ldots, k, \\ a_{ij}^{(k+1)} &= 0, \quad i = k+1, \ldots, n; \quad j = 1, \ldots, k, \end{aligned}$$

$$\begin{aligned} a_{ik}^{(k+1)} &= 0, \quad i = 1, \ldots, k-1, \\ a_{ij}^{(k+1)} &= a_{ij}^{(k)} - \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} \, a_{kj}^{(k)}, \quad i \neq k; \quad j = k+1, \ldots, j = n+1. \end{aligned}$$

La stratégie du pivotage partiel sera également utilisée mais en recherchant l'indice p tel que $|a_{pk}^{(k)}| \geq |a_{ik}^{(k)}|$ pour $i = 1, \ldots, n$.

Si l'un des pivots est nul, alors le système est singulier.

On démontre que l'on passe du système k au système $k+1$ par

$$A^{(k+1)} = L^{(k)} A^{(k)}$$

avec

$$L^{(k)} = \begin{pmatrix} 1 & & & -l_{1k} & & \\ & \ddots & & \vdots & & \\ & & 1 & -l_{k-1,k} & & \\ & & & 1 & & \\ & & & -l_{k+1,k} & 1 & \\ & & & \vdots & & \ddots \\ & & & -l_{nk} & & 1 \end{pmatrix},$$

$l_{ik} = a_{ik}^{(k)} / a_{kk}^{(k)}$ et tous les autres éléments nuls.

Si l'on veut rendre unitaires les diagonales des matrices $A^{(k)}$, il faut rajouter la relation

$$a_{kj}^{(k+1)} = a_{kj}^{(k)} / a_{kk}^{(k)}, \quad j = k, \ldots, n+1.$$

Cela revient à prendre, pour le passage de $A^{(k)}$ à $A^{(k+1)}$,

$$L^{(k)} = \begin{pmatrix} 1 & & & -l_{1k} & & \\ & \ddots & & \vdots & & \\ & & 1 & -l_{k-1,k} & & \\ & & & 1/a_{kk}^{(k)} & & \\ & & & -l_{k+1,k} & 1 & \\ & & & \vdots & & \ddots \\ & & & -l_{nk} & & 1 \end{pmatrix}.$$

C'est cette variante de la méthode de Gauss–Jordan qui est utilisée en programmation linéaire dans l'algorithme du simplexe.

Il suffit ensuite de résoudre le système diagonal ainsi obtenu (l'indice supérieur n a été supprimé)

$$x_i = a_{i,n+1} / a_{ii}, \quad i = 1, \ldots, n.$$

4.6.3 La méthode de Gauss symétrique

Commençons par étudier la méthode de Gauss quand on l'applique à une matrice symétrique. Comme nous l'avons vu, cette méthode revient à décomposer A sous la forme $A = LU$ avec L triangulaire inférieure à diagonale unité et U triangulaire supérieure. Nous avons donc $A^T = (LU)^T = U^T L^T$. Puisque $A = A^T$ et bien que U^T soit triangulaire inférieure et que L^T soit triangulaire supérieure, on voit que $U^T L^T$ ne peut pas être la décomposition de Gauss de A^T parce que l'on a imposé à L d'avoir une diagonale unité. Donc U^T ne peut être identique à L. Cela brise la symétrie de la décomposition, la méthode de Gauss ne profite pas de la symétrie de A et elle est donc plus coûteuse que nécessaire. La méthode de Gauss, symétrique, une variante de la méthode de Gauss, consiste à décomposer A sous la forme

$$A = LDL^T$$

où L est triangulaire inférieure à diagonale unité et D est diagonale (d'éléments $d_1, \ldots, d_n$). En identifiant les éléments correspondants de A et de LDL^T, on trouve

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

En prenant $i = j$ dans cette relation, on obtient, puisque $\forall j, l_{jj} = 1$,

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

et pour $i > j$

$$l_{ij} = \left(a_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk} d_k \right) / d_j.$$

Cette variante nécessite deux fois moins d'opérations arithmétiques que la méthode de Gauss. Après avoir décomposé A de cette façon, on résoud successivement les systèmes triangulaires $Ly = b$ et $L^T x = D^{-1} y$. On peut aussi résoudre $Lz = b$, puis calculer $y = D^{-1} z$ et enfin résoudre $L^T x = y$.

On a $\det A = \prod_{k=1}^{n} d_k$.

Considérons le cas particulier d'une matrice tridiagonale symétrique

$$A = \begin{pmatrix} a_1 & -b_2 & & & \\ -b_2 & a_2 & -b_3 & & \\ & \ddots & \ddots & \ddots & \\ & & -b_{n-1} & a_{n-1} & -b_n \\ & & & -b_n & a_n \end{pmatrix}.$$

On suppose que $\forall i, b_i \neq 0$. On a $A = L\Delta^{-1}L^T$ avec

$$L = \begin{pmatrix} \delta_1 & & & & \\ -b_2 & \delta_2 & & & \\ & \ddots & \ddots & & \\ & & -b_{n-1} & \delta_{n-1} & \\ & & & -b_n & \delta_n \end{pmatrix} \quad \text{et} \quad \Delta = \mathrm{diag}(\delta_1, \ldots, \delta_n).$$

Les éléments de Δ sont donnés par

$$\begin{aligned} \delta_1 &= a_1, \\ \delta_i &= a_i - b_i^2/\delta_{i-1}, \quad i = 2, \ldots, n. \end{aligned}$$

Il est également possible de décomposer la matrice A en un produit $A = UD^{-1}U^T$ avec $D = \mathrm{diag}(d_1, \ldots, d_n)$ et

$$U = \begin{pmatrix} d_1 & -b_1 & & & \\ & d_2 & -b_3 & & \\ & & \ddots & \ddots & \\ & & & d_{n-1} & -b_n \\ & & & & d_n \end{pmatrix}.$$

On a

$$\begin{aligned} d_n &= a_n, \\ d_i &= a_i - b_{i+1}^2/d_{i+1}, \quad i = n-1, \ldots, 1. \end{aligned}$$

L'inverse de la matrice A est de la forme

$$A^{-1} = \begin{pmatrix} u_1v_1 & u_1v_2 & u_1v_3 & \cdots & u_1v_n \\ u_1v_2 & u_2v_2 & u_2v_3 & \cdots & u_2v_n \\ u_1v_3 & u_2v_3 & u_3v_3 & \cdots & u_3v_n \\ \vdots & \vdots & \vdots & & \vdots \\ u_1v_n & u_2v_n & u_3v_n & \cdots & u_nv_n \end{pmatrix}.$$

Dans cette expression, u_1 est arbitraire et l'on prendra $u_1 = 1$.

On peut montrer que $v = (v_1, \ldots, v_n)^T$ est solution du système $Av = e_1$, où e_1 est le premier vecteur de la base canonique de $\mathbb{R}^n$. À partir des questions précédentes, il vient

$$v_1 = 1/d_1, \qquad v_i = (b_2 \cdots b_i)/(d_1 \cdots d_i), \quad i = 2, \ldots, n.$$

Si l'on pose $u = (u_1, \ldots, u_n)^T$, alors $v_n Au = e_n$, où e_n est le dernier vecteur de la base canonique de $\mathbb{R}^n$ et

$$u_n = 1/(\delta_n v_n), \qquad u_{n-i} = (b_{n-i+1} \cdots b_n)/(\delta_{n-i} \cdots \delta_n v_n), \quad i = 1, \ldots, n-1.$$

Finalement $u_1 = (b_2 \cdots b_n)/(\delta_1 \cdots \delta_n v_n) = (d_1 \cdots d_n)/(\delta_1 \cdots \delta_n)$ et $d_1 \cdots d_n = \delta_1 \cdots \delta_n = \det A$. Par conséquent, on retrouve bien que $u_1 = 1$.

4.6.4 La méthode de Gauss par blocs

Considérons le système par blocs

$$\begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1m} \\ \vdots & \vdots & & \vdots \\ A_{m1} & A_{m2} & \cdots & A_{mm} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}$$

où $A_{ij} \in \mathbb{R}^{m_i \times m_j}$, $x_i \in \mathbb{R}^{m_i}$ et $b_i \in \mathbb{R}^{m_i}$. On peut appliquer une méthode de Gauss par blocs pour résoudre ce système. L'idée est la même que celle de la méthode de Gauss. Pour éliminer x_1, on multiplie la première équation de blocs à droite par A_{11}^{-1}, puis par A_{i1} et on la soustrait de la i-ème équation. Puis ensuite, on élimine x_2 par une stratégie analogue en partant de la seconde équation de ce nouveau système et ainsi de suite. Les systèmes intermédiaires ainsi obtenus ont une structure par blocs similaire à (4.1) et que l'on passe du système k au système $k+1$ par les relations

$$\begin{aligned} A_{ij}^{(k+1)} &= A_{ij}^{(k)} - A_{ik}^{(k)}(A_{kk}^{(k)})^{-1}A_{kj}^{(k)}, \quad i,j = k+1,\ldots,m \\ b_i^{(k+1)} &= b_i^{(k)} - A_{ik}^{(k)}(A_{kk}^{(k)})^{-1}b_k^{(k)}, \quad i = k+1,\ldots,m \end{aligned}$$

les autres blocs restant inchangés. Dans ces formules, on reconnaîtra le complément de Schur étudié dans la Section 2.9.2.

On aura ensuite à résoudre un système triangulaire supérieur par blocs de la forme

$$\begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1m} \\ 0 & A_{22} & \cdots & A_{2m} \\ & & \ddots & \vdots \\ & & & A_{mm} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}.$$

On a donc $x_m = A_{mm}^{-1}b_m$ puis, pour $i = m-1,\ldots,1$,

$$x_i = A_{ii}^{-1}\left(b_i - \sum_{j=i+1}^{n} A_{ij}x_j\right).$$

4.7 Pseudo–codes

Dans cette Section, nous allons donner les pseudo–codes des algorithmes. Nous noterons $(A \mid b)$ la matrice $n \times (n+1)$ obtenue en adjoignant à A le second membre b comme $(n+1)$-ème colonne.

4.7.1 Système triangulaire supérieur

On veut résoudre $Ux = b$ où U est une matrice triangulaire supérieure. On suppose, bien entendu, que les éléments diagonaux u_{ii} de U sont tous non nuls.

$$[\mathbf{x}] = \textbf{Tri\_sup}\ (U, \mathbf{b}, n)$$

for $i = n, \ldots, 1$ **step** -1
  $x_i = b_i$
  for $j = i + 1 \ldots, n$
    $x_i = x_i - u_{ij}\, x_j$
  end for j
  $x_i = x_i / u_{ii}$
end for i

4.7.2 Système triangulaire inférieur

On veut résoudre $Lx = b$ où L est une matrice triangulaire inférieure. On suppose, bien entendu, que les éléments diagonaux l_{ii} de L sont tous non nuls.

$$[\mathbf{x}] = \textbf{Tri\_inf}\ (L, \mathbf{b}, n)$$

for $i = 1, \ldots, n$
  $x_i = b_i$
  for $j = 1 \ldots, i - 1$
    $x_i = x_i - l_{ij}\, x_j$
  end for j
  $x_i = x_i / l_{ii}$
end for i

4.7.3 Méthode de Gauss

Pour simplifier l'algorithme, on commence par placer le second membre b comme $(n+1)$-ème colonne de la matrice A, c'est–à–dire que l'on pose $a_{i,n+1} = b_i$ pour $i = 1, \ldots, n$.

$$[L, (U|\mathbf{y})] = \textbf{Gauss}\ ((A|\mathbf{b}), n)$$

Initialisation de L
for $i = 1, \ldots, n$
  $l_{ii} = 1$
  for $j = i + 1, \ldots, n$
    $l_{ij} = 0$
  end for j
end for i
Algorithme de Gauss
for $k = 1, \ldots, n - 1$
  for $i = k + 1, \ldots, n$

$l_{ik} = a_{ik}/a_{kk}$
for $j = k, \ldots, n+1$
$a_{ij} = a_{ij} - l_{ik}\, a_{kj}$
end for j
end for i
end for k

4.7.4 Méthode de Gauss avec pivotage partiel

Dans ce pseudo–code, nous n'avons pas effectué les tests sur la nullité des pivots qui permettent de décider si la matrice est singulière.

$$[L, (U|\mathbf{y})] = \textbf{Gauss\_pivot}\ ((A|\mathbf{b}), n)$$

for $k = 1, \ldots, n-1$
Calculer r tant que $|a_{rk}| = \max\limits_{k \le i \le n} |a_{ik}|$
if $r \neq k$ **then**
Intervertir la ligne r de $(A|\mathbf{b})$ avec la ligne k
endif
for $i = k+1, \ldots, n$
$a_{ik} = a_{ik}/a_{kk}$
for $j = k+1, \ldots, n+1$
$a_{ij} = a_{ij} - a_{ik}\, a_{kj}$
end for j
end for i
end for k
for $i = 1, \ldots, n$
$l_{ii} = 1$
for $j = i+1, \ldots, n$
$l_{ij} = 0$
$l_{ji} = a_{ji}$
$a_{ji} = 0$
end for j
end for i

4.7.5 Factorisation de Gauss–Crout

$$[L, U] = \textbf{LU\_Crout}\ (A, n)$$

for $i = 1, \ldots, n$
$l_{i1} = a_{i1}$
$u_{1i} = a_{1i}/l_{11}$
$u_{ii} = 1$

end for i
for $j = 2, \ldots, n$
  $l_{1j} = 0$
  $u_{j1} = 0$
  for $i = j, \ldots, n$
    $l_{ji} = 0$
    $l_{ij} = a_{ij}$
    for $k = 1, \ldots, j-1$
      $l_{ij} = l_{ij} - l_{ik}\, u_{kj}$
    end for k
  end for i
  for $i = j+1, \ldots, n$
    $u_{ij} = 0$
    $u_{ji} = a_{ji}$
    for $k = 1, \ldots, j-1$
      $u_{ji} = u_{ji} - l_{jk}\, u_{ki}$
    end for k
    $u_{ji} = u_{ji}/l_{jj}$
  end for i
end for j

4.8 Expériences numériques

Donnons maintenant les résultats de quelques expériences numériques caractéristiques.

Considérons le système

$$\begin{pmatrix} 21 & 130 & 0 & 2.1 \\ 13 & 80 & 4.74\ 10^{8} & 752 \\ 0 & -0.4 & 3.9816\ 10^{8} & 4.2 \\ 0 & 0 & 1.7 & 9\ 10^{-9} \end{pmatrix} x = \begin{pmatrix} 153.1 \\ 849.74 \\ 7.7816 \\ 2.6\ 10^{-8} \end{pmatrix}$$

dont la solution exacte est $x = (1, 1, 10^{-8}, 1)^T$. La méthode de Gauss avec pivotage partiel donne, en arithmétique simple précision,

$$\begin{pmatrix} 0.6261987\ 10^{2} \\ -0.8953979\ 10^{1} \\ 0.0000000 \\ 0.9999999 \end{pmatrix}.$$

Ce résultat n'est pas surprenant dans la mesure où $\kappa = 2.2729 \cdot 10^{18}$.

Donnons un autre exemple qui, bien que peu réaliste puisque nous ne ferons les calculs qu'avec 4 chiffres significatifs, montre bien comment les erreurs dues à l'arithmétique de l'ordinateur se propagent dans les calculs. On considère le

système

$$\begin{pmatrix} 0.001 & 1.200 & 2.000 \\ 1.207 & 2.051 & 1.963 \\ 1.006 & 2.002 & 3.000 \end{pmatrix} x = \begin{pmatrix} 3.201 \\ 5.221 \\ 6.008 \end{pmatrix}$$

dont la solution exacte est $x = (1, 1, 1)^T$. La méthode de Gauss donne, en faisant les calculs avec 4 chiffres décimaux

$$a_{21}/a_{11} = 1207, \quad a_{31}/a_{11} = 1006$$

et l'on obtient

$$\begin{pmatrix} 0.001 & 1.200 & 2.000 \\ 0 & -1446 & -2412 \\ 0 & -1205 & -2009 \end{pmatrix} x = \begin{pmatrix} 3.201 \\ -3859 \\ -3214 \end{pmatrix}.$$

Puis, on obtient

$$\frac{a_{32}^{(2)}}{a_{22}^{(2)}} = \frac{-1205}{-1446} = 0.8333, \quad a_{33}^{(3)} = 1, \quad b_3^{(3)} = 2$$

c'est–à–dire

$$\begin{pmatrix} 0.001 & 1.200 & 2.000 \\ 0 & -1446 & -2412 \\ 0 & 0 & 1 \end{pmatrix} x = \begin{pmatrix} 3.201 \\ -3859 \\ 2 \end{pmatrix}.$$

Le résolution de ce système triangulaire supérieur conduit à

$$x = (1.900 \ ; \ -0.6674 \ ; \ 2.000)^T.$$

Prenons maintenant le même système mais en intervertissant la première et la seconde équation (ce qui est équivalent à du pivotage partiel). On obtient

$$a_{21}/a_{11} = 0.8285 \ 10^{-3}, \quad a_{31}/a_{11} = 0.8335$$

et

$$\begin{pmatrix} 1.207 & 2.051 & 1.963 \\ 0 & 1.198 & 1.998 \\ 0 & 0.292 & 1.364 \end{pmatrix} x = \begin{pmatrix} 5.221 \\ 3.197 \\ 1.656 \end{pmatrix}.$$

Puis, on a

$$\frac{a_{32}^{(2)}}{a_{22}^{(2)}} = 0.2437$$

et

$$\begin{pmatrix} 1.207 & 2.051 & 1.963 \\ 0 & 1.198 & 1.998 \\ 0 & 0 & 0.8771 \end{pmatrix} x = \begin{pmatrix} 5.221 \\ 3.197 \\ 0.8769 \end{pmatrix}$$

et l'on trouve finalement comme solution

$$x = (1.002 \ ; \ 1.001 \ ; \ 0.9998)^T.$$

Il est cependant facile de construire des exemples où le pivotage partiel ne conduit pas à de bons résultats. Considérons le système

$$\begin{pmatrix} 2.000 & 2400 & 4000 \\ 1.207 & 2.051 & 1.963 \\ 1.006 & 2.002 & 3.000 \end{pmatrix} x = \begin{pmatrix} 6402 \\ 5.221 \\ 6.008 \end{pmatrix}$$

dont la solution exacte est $x = (1, 1, 1)^T$. La méthode de Gauss donne, en faisant les calculs avec 4 chiffres décimaux

$$a_{21}/a_{11} = 0.6035, \quad a_{31}/a_{11} = 0.5030$$

puisqu'il n'y a pas de pivotage à la première étape. On obtient donc

$$\begin{pmatrix} 2.000 & 2400 & 4000 \\ 0 & -1446 & -2412 \\ 0 & -1205 & -2009 \end{pmatrix} x = \begin{pmatrix} 6402 \\ -3859 \\ -3214 \end{pmatrix}.$$

Il n'y a pas non plus de pivotage à la seconde étape et l'on a

$$\frac{a_{32}^{(2)}}{a_{22}^{(2)}} = 0.8333$$

et

$$\begin{pmatrix} 2.000 & 2400 & 4000 \\ 0 & -1446 & -2412 \\ 0 & 0 & 1 \end{pmatrix} x = \begin{pmatrix} 6402 \\ -3859 \\ 2 \end{pmatrix}.$$

Le résolution de ce système triangulaire supérieur conduit à

$$x = (1.900 \ ; \ -0.6674 \ ; \ 2.000)^T.$$

Cet exemple montre qu'il peut être utile d'équilibrer la matrice auparavant. Ce problème de l'équilibrage n'est pas encore résolu de façon satisfaisante (car, pour cela, il faudrait connaître la solution exacte du système). En pratique, on remplace donc le système $Ax = b$ par le système $D_1AD_2y = D_1b$ avec $x = D_2y$ et D_1 et D_2 des matrices diagonales. On voit que l'équilibrage revient à faire un préconditionnement bilatéral du système.

On peut choisir ces matrices de sorte que l'élément le plus grand en valeur absolue de chaque ligne et de chaque colonne soit égal à 1, ou que la somme des valeurs absolues des éléments de chaque ligne (ou de chaque colonne) soit égale à 1, ou que les éléments de la matrice D_1AD_2 soient tous de même ordre de grandeur. Cependant, aucune de ces solutions ne semble satisfaisante dans tous les cas.

Si on n'utilise qu'un équlibrage par lignes (c'est–à–dire si $D_2 = I$ et donc $x = y$), cet équilibrage n'affecte la solution obtenue par la méthode de Gauss avec pivotage partiel que s'il change la sélection des pivots (puisque chaque

équation du système est multipliée par l'éléments diagonal correspondant de D_1). Le choix de D_1 est donc important dans la méthode de Gauss.

Reprenons l'exemple donné dans la Section 3.2.3.

$$\begin{pmatrix} 1.2969 & 0.8648 \\ 0.2161 & 0.1441 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0.8642 \\ 0.1440 \end{pmatrix}$$

dont la solution exacte est $x = (2, -2)^T$. En appliquant la méthode de Gauss (le pivotage partiel ou total ne modifie ni l'ordre des équations ni celui des colonnes) on obtient $x_2 = 0$ et $x_1 = 0.6663582$.

Chapitre 5

La méthode de Cholesky

La méthode de André Louis Cholesky (1875–1918) est une variante de la méthode de Gauss dans le cas où la matrice A est symétrique définie positive (cas que l'on rencontre dans la résolution au sens des moindres carrés d'un système linéaire avec plus d'équations que d'inconnues). Dans la méthode de Gauss, la symétrie de la matrice n'est pas exploitée et, par conséquent et elle est donc plus coûteuse que nécessaire. En effet, pour une matrice symétrique, nous aurons $A = LU$ et $A^T = U^T L^T$. Le produit $U^T L^T$ n'est pas la décomposition de Gauss de la matrice A^T car, bien que U^T soit une matrice triangulaire inférieure, elle n'a pas obligatoirement une diagonale unité.

Dans la méthode de Cholesky, on va abandonner la condition que L soit à diagonale unité et ainsi les matrices L et U pourront être transposées l'une de l'autre, c'est–à–dire que l'on va effectuer une décomposition de la forme

$$A = LL^T$$

avec L est triangulaire inférieure (sans lui imposer d'avoir une diagonale unité) et inversible. On démontre qu'une condition nécessaire et suffisante pour que cette décomposition existe est que A soit symétrique définie positive.

En pratique, on calcule directement les éléments de L par identification des éléments de LL^T avec les éléments correspondants de A.

On commence par calculer

$$l_{11} = \sqrt{a_{11}},$$

puis les éléments de la première colonne de L par

$$l_{i1} = a_{i1}/l_{11}, \quad i = 2, \ldots, n.$$

Ensuite, pour $j = 2, \ldots, n$, l'élément diagonal de la $j$ème colonne est donné par

$$l_{jj} = \left(a_{jj} - \sum_{k=1}^{j-1} l_{jk}^2 \right)^{1/2}.$$

Les autres termes de la colonne j sont ensuite calculés par

$$l_{ij} = \left(a_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk} \right) / l_{jj}, \quad i = j+1, \ldots, n.$$

Enfin on résoud successivement les deux systèmes triangulaires

$$\begin{aligned} Ly &= b \\ L^T x &= y. \end{aligned}$$

On a $\det A = \prod_{j=1}^{n} l_{jj}^2$.

Si la matrice A est symétrique mais non définie positive alors il existe au moins un indice j pour lequel $l_{jj}^2 < 0$. En effet $l_{jj}^2 = (L^T e_j, L^T e_j) = (e_j, A e_j)$ est strictement positif si A est définie positive. On a le

Théorème 20
Soit A une matrice symétrique régulière. Une condition nécessaire et suffisante pour que la décomposition de Cholesky $A = LL^T$ existe est que A soit définie positive.

Démonstration.

Supposons l'existence de la décomposition de Cholesky $A = LL^T$ avec L triangulaire inférieure. Soit λ l'une de ses valeurs propres et x le vecteur propre correspondant. On a $(x, LL^T x) = (Lx, Lx) = \lambda(x, x)$. Donc $\lambda > 0$, ce qui démontre que la matrice A est définie positive.

Réciproquement, considérons la décomposition de Gauss symétrique $A = LDL^T$ où la matrice L est triangulaire inférieure à diagonale unité. On a $(L^T x, DL^T x) = \lambda(x, x)$. Si A est définie positive, ses valeurs propres sont strictement positives. Donc $(L^T x, DL^T x) > 0$, ce qui montre que D est définie positive, c'est–à–dire que ses éléments diagonaux sont strictement positifs. Sa racine carrée $D^{1/2}$ existe donc. D'où $A = (LD^{1/2})(LD^{1/2})^T$ qui n'est autre que la décomposition de Cholesky de A. ∎

La détermination de L nécessite n extractions de racines carrées, $n(n-1)/2$ divisions, $n(n^2-1)/6$ multiplications et autant d'additions. La résolution des deux systèmes triangulaires nécessite $2n$ divisions, $n(n-1)$ multiplications et autant d'additions. Au total, il faut donc n calculs de racines carrées et $n^3/3$ autres opérations arithmétiques soit deux fois moins d'opérations qu'avec la méthode de Gauss.

La décomposition de Cholesky peut s'obtenir à partir de la décomposition de Gauss symétrique. En effet, si A est symétrique définie positive, les éléments de

la matrice diagonale D sont strictement positifs et l'on peut calculer la racine carrée $D^{1/2}$ de D. On a donc, dans la décomposition de Gauss symétrique, $A = (LD^{1/2})(D^{1/2}L^T) = (LD^{1/2})(LD^{1/2})^T$, ce qui n'est autre que la décomposition de Cholesky.

Il faut remarquer que la décomposition $A = LL^T$, avec L triangulaire inférieure, n'est pas unique. Elle depend, en effet, du signe que l'on met devant les racines carrées. Dans la méthode de Cholesky, ce signe est toujours positif. La décomposition est alors unique. Soit, par exemple,

$$A = \begin{pmatrix} 1 & 2 & -1 \\ 2 & 5 & -3 \\ -1 & -3 & 6 \end{pmatrix}.$$

On a $A = LL^T = \widetilde{L}\widetilde{L}^T$ avec

$$L = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & -1 & 2 \end{pmatrix}, \quad \widetilde{L} = \begin{pmatrix} 1 & 0 & 0 \\ 2 & -1 & 0 \\ -1 & 1 & 2 \end{pmatrix}.$$

Chapitre 6

La méthode de Householder

La méthode de Alston S. Householder (1904–1993) consiste à mettre la matrice A sous la forme d'un produit $A = QU$ où Q est une matrice orthogonale (c'est–à–dire telle que $Q^T = Q^{-1}$) et U est une matrice triangulaire supérieure. La matrice Q est construite à partir de *matrices orthogonales élémentaires* H définies de la façon suivante

Lemme 1
Soit $u \in \mathbb{R}^n$ tel que $(u, u) = 1$. Alors la matrice $H = I - 2uu^T$ est symétrique et orthogonale.

Démonstration.

Il est évident que H est symétrique. Montrons que $HH^T = H^2 = I$. On a

$$H^2 = (I - 2uu^T)^2 = I - 4uu^T + 4uu^T uu^T$$

or $u^T u = 1$ d'où $I = H^2$. ∎

Donnons une interprétation géométrique d'une telle matrice H. Pour cela, considérons d'abord la matrice $P = I - uu^T$. Elle représente une projection sur $u^\perp$, le complémentaire orthogonal à u. En effet, soit v un vecteur quelconque. On a $v - Pv = (u, v)u$. Donc $(y, v - Pv) = 0$ pour tout vecteur y orthogonal à u, c'est–à–dire que $v - Pv$ est orthogonal à $u^\perp$, soit encore parallèle à u. L'interprétation géométrique de la matrice $H = I - 2uu^T$ s'en déduit maintenant. En effet, Hv est le symétrique (à cause du coefficient 2) de v par rapport a $u^\perp$ et $v - Hv$ est, lui aussi, orthogonal à $u^\perp$. D'où la

Propriété 5
H admet -1 comme valeur propre simple avec u comme vecteur propre et 1 comme valeur propre de multiplicité $n-1$ correspondant au sous–espace propre formé de l'hyperplan orthogonal à u. H représente donc une symétrie par rapport à cet hyperplan. De plus $\det(H) = -1$.

Revenons maintenant à la méthode de Householder. Elle repose sur le résultat fondamental suivant

Théorème 21
Soit $a \in \mathbb{R}^n$, $a \neq 0$. Alors il existe une matrice orthogonale élémentaire H et un nombre $\alpha \in \mathbb{R}$, $\alpha \neq 0$, tels que

$$Ha = \alpha e_1$$

où e_1 est le premier vecteur de la base canonique de $\mathbb{R}^n$.

Démonstration.

Soit $H = I - 2uu^T$ une matrice répondant à la question. Il nous faut détemriner u et α. On a

$$\|Ha\|^2 = (Ha, Ha) = (a, H^T Ha) = (a, a) = \|a\|^2.$$

Donc

$$\|Ha\|^2 = (\alpha e_1, \alpha e_1) = |\alpha|^2 = \|a\|^2$$

ce qui donne deux valeurs possible pour α, $\alpha = \pm\|a\|$.

Posons

$$\mu = 2(u, a).$$

On a

$$Ha = (I - 2uu^T)a = a - \mu u = \alpha e_1$$

d'où

$$\mu u = a - \alpha e_1.$$

Multiplions scalairement à gauche par $2a$, il vient

$$2\mu(a, u) = 2(a, a) - 2\alpha(a, e_1)$$

ce qui donne

$$\mu^2 = 2\alpha(\alpha - a_1)$$

où a_1 est la première composante du vecteur a. On peut donc trouver u, α et μ vérifiant

$$\begin{aligned} |\alpha| &= \|a\| \\ \mu^2 &= 2\alpha(\alpha - a_1) \\ \mu u &= a - \alpha e_1. \blacksquare \end{aligned}$$

Les formules précédentes peuvent être simplifiées. Posons $\nu = \mu^2/2$ et $v = \mu u$; on a

$$H = I - vv^T/\nu$$

avec

$$\begin{aligned} \alpha &= -(\text{signe de } a_1)\|a\| \\ \nu &= \alpha(\alpha - a_1) \\ v &= a - \alpha e_1. \end{aligned}$$

On a choisi comme valeur de α celle qui rend ν maximum pour des raisons évidentes de stabilité numérique puisque ν intervient au dénominateur dans la détermination de H. Le remplacement de μ par ν permet d'éviter le calcul d'une racine carrée.

Exposons maintenant la méthode de Householder proprement dite.

On pose $A^{(1)} = A$ et $b^{(1)} = b$ et l'on génère les $n-1$ systèmes équivalents

$$A^{(k)}x = b^{(k)}, \quad k = 2, \ldots, n$$

avec

$$A^{(k)} = \begin{pmatrix} a_{11}^{(2)} & a_{12}^{(2)} & \cdots & a_{1,k-1}^{(2)} & a_{1k}^{(2)} & \cdots & a_{1n}^{(2)} \\ & a_{22}^{(3)} & \cdots & a_{2,k-1}^{(3)} & a_{2,k-1}^{(3)} & \cdots & a_{2n}^{(3)} \\ & & \ddots & \vdots & \vdots & & \vdots \\ & & & a_{k-1,k-1}^{(k)} & a_{k-1,k}^{(k)} & \cdots & a_{k-1,n}^{(k)} \\ & & & & a_{kk}^{(k)} & \cdots & a_{kn}^{(k)} \\ & & & & \vdots & & \vdots \\ & & & & a_{nk}^{(k)} & \cdots & a_{nn}^{(k)} \end{pmatrix}, \quad b^{(k)} = \begin{pmatrix} b_1^{(2)} \\ b_2^{(3)} \\ \vdots \\ b_{k-1}^{(k)} \\ b_k^{(k)} \\ \vdots \\ b_n^{(k)} \end{pmatrix}.$$

Nous écrirons ce système sous une forme partitionnée en 4 blocs

$$A^{(k)} = \begin{pmatrix} A_{11}^{(k)} & A_{12}^{(k)} \\ 0 & A_{22}^{(k)} \end{pmatrix}, \quad b^{(k)} = \begin{pmatrix} c^{(k)} \\ d^{(k)} \end{pmatrix}.$$

On passe de $A^{(k)}$, $b^{(k)}$ à $A^{(k+1)}$, $b^{(k+1)}$ à l'aide d'une matrice orthogonale élémentaire $H^{(k)} = I - 2u^{(k)}u^{(k)^T}$, avec $u_i^{(k)} = 0$ pour $i = 1, \ldots, k-1$, par les relations

$$A^{(k+1)} = H^{(k)}A^{(k)}, \qquad b^{(k+1)} = H^{(k)}b^{(k)}.$$

Désignons par $\tilde{u}^{(k)}$ le vecteur de $\mathbb{R}^{n-k+1}$ de composantes $u_k^{(k)}, \ldots, u_n^{(k)}$ et notons $\tilde{H}^{(k)} = I - 2\tilde{u}^{(k)}\tilde{u}^{(k)T}$. On a

$$\begin{aligned} H^{(k)} &= \begin{pmatrix} I & 0 \\ 0 & \tilde{H}^{(k)} \end{pmatrix} \\ H^{(k+1)} &= \begin{pmatrix} I & 0 \\ 0 & \tilde{H}^{(k)} \end{pmatrix} \begin{pmatrix} A_{11}^{(k)} & A_{12}^{(k)} \\ 0 & A_{22}^{(k)} \end{pmatrix} = \begin{pmatrix} A_{11}^{(k)} & A_{12}^{(k)} \\ 0 & \tilde{H}^{(k)}A_{22}^{(k)} \end{pmatrix} \\ b^{(k+1)} &= \begin{pmatrix} I & 0 \\ 0 & \tilde{H}^{(k)} \end{pmatrix} \begin{pmatrix} c^{(k)} \\ d^{(k)} \end{pmatrix} = \begin{pmatrix} c^{(k)} \\ \tilde{H}^{(k)}d^{(k)} \end{pmatrix}. \end{aligned}$$

Les éléments de la première colonne de $A^{(k+1)}$ ne sont pas tous nuls. $\tilde{H}^{(k)}$ sera donc choisi en accord avec les résultats du Lemme précédent de façon à annuler tous les éléments de la première colonne de $A_{22}^{(k+1)}$ sauf un seul, le premier. Du point de vue algorithmique la méthode de Householder se résume donc aux règles suivantes

On calcule, pour $k = 1, \ldots, n-1$

$$\begin{aligned} a_{kk}^{(k+1)} &= -(\text{signe de } a_{kk}^{(k)}) \left(\sum_{i=k}^{n} |a_{ik}^{(k)}|^2 \right)^{1/2} \\ \nu^{(k)} &= a_{kk}^{(k+1)} (a_{kk}^{(k+1)} - a_{kk}^{(k)}) \\ v_k^{(k)} &= a_{kk}^{(k)} - a_{kk}^{(k+1)} \\ v_i^{(k)} &= a_{ik}^{(k)}, \quad i = k+1, \ldots, n \end{aligned}$$

puis

$$\begin{aligned} \beta_j^{(k)} &= \sum_{i=k}^{n} v_i^{(k)} a_{ij}^{(k)}, \quad j = k+1, \ldots n \\ \gamma_j^{(k)} &= \beta_j^{(k)} / \nu^{(k)}, \quad j = k+1, \ldots, n \\ a_{ij}^{(k+1)} &= a_{ij}^{(k)} - \gamma_j^{(k)} v_i^{(k)}, \quad i = k, \ldots, n; \quad j = k+1, \ldots, n. \end{aligned}$$

et de même

$$\begin{aligned} \beta_{n+1}^{(k)} &= \sum_{i=k}^{n} v_i^{(k)} b_i^{(k)} \\ \gamma_{n+1}^{(k)} &= \beta_{n+1}^{(k)} / \nu^{(k)} \\ b_i^{(k+1)} &= b_i^{(k)} - \gamma_{n+1}^{(k)} v_i^{(k)}, \quad i = k, \ldots, n. \end{aligned}$$

Le système $A^{(n)}x = b^{(n)}$ ainsi obtenu est triangulaire supérieur.

Le scalaire $\nu^{(k)}$ qui intervient comme dénominateur dans les règles précédentes est nul si et seulement si $a_{ik}^{(k)} = 0$ pour $i = k, \ldots, n$. Dans ce cas, $\det A_{22}^{(k)} = 0$ et il s'en suit que A est singulière.

Remarque 7
On voit que, comme dans la méthode de Gauss, les relations qui permettent d'obtenir $b^{(k+1)}$ sont tout à fait semblables à celles qui donnent $A^{(k+1)}$. On peut donc traiter les seconds membres en les rajoutant, comme dans la méthode de Gauss, comme $(n+1)$-ème colonne des matrices.

Il ne faut pas, comme on le faisait dans la méthode de Gauss, remplacer immédiatement $a_{kk}^{(k)}$ par la nouvelle valeur $a_{kk}^{(k+1)}$ puisque $a_{kk}^{(k)}$ est utilisé dans plusieurs relations.

Par récurrence, on voit que

$$A^{(n)} = H^{(k-1)} \cdots H^{(1)} A, \quad b^{(n)} = H^{(k-1)} \cdots H^{(1)} b.$$

Posons

$$Q^T = Q^{-1} = H^{(k-1)} \cdots H^{(1)}.$$

On a $A = QU$ et $Qy = b$, avec $U = A^{(n)}$ et $y = b^{(n)}$, ainsi que (à cause du signe négatif dans la formule pour $a_{kk}^{(k+1)}$) $\det A = (-1)^{n-1} \prod_{i=1}^{n} a_{ii}^{(n)}$.

Avec la résolution du système triangulaire ainsi obtenu, la méthode de Householder nécessite en tout

$n-1$ extractions de racines carrées

$\dfrac{4n^3 + 9n^2 - n - 12}{6}$ additions

$\dfrac{2n^3 + 6n^2 + 4n - 12}{3}$ multiplications

$\dfrac{n^2 + 3n - 2}{2}$ divisions

soit, au total, à peu près $2n^3/3$ additions et autant de multiplications. Cette méthode requiert donc environ deux fois plus d'opérations que la méthode de Gauss mais elle a l'avantage d'être numériquement plus stable.

La méthode de Householder montre donc qu'une matrice A non singulière peut se mettre sous la forme $A = QU$ avec $Q = H^{(1)} H^{(2)} \cdots H^{(n-1)}$. Quand A est singulière, on trouve obligatoirement une matrice $A^{(r)}$ dont les $n - r + 1$ derniers éléments de la n–ième colonne sont nuls. On pose alors $H^{(r)} = I$ et l'on continue la triangularisation. Si A est inversible, on démontre que la décomposition QU est unique.

Les décompositions LU et QU conduisent à des méthodes itératives de calcul des valeurs propres : les algorithmes LR et QR. Ils ne seront pas étudiés dans ce livre.

Chapitre 7

Matrices creuses

Dans de très nombreux cas, on doit résoudre des systèmes d'équations linéaires de très grande dimension n. On a couramment à traiter le cas $n = 100000$ mais, à l'heure actuelle, on peut aller jusqu'à 10^6 ou plus. Naturellement, le premier problème qui se pose est de pouvoir stocker de telles matrices dans l'ordinateur. On va, bien sûr, essayer de tirer profit des caractéristiques de la matrice pour réduire l'encombrement mémoire. C'est ainsi, que, si elle est symétrique, on n'en stockera que la moitié. En examinant, comme nous le ferons dans la Section 7.1, la provenance des très grands sytèmes, on s'aperçoit que, dans beaucoup de cas qui se présentent dans la réalité, un très grand nombre d'éléments de la matrice sont nuls. Il faut en tirer avantage et, bien entendu, ne pas les stocker. Cette question sera examinée dans la Section 7.2. Il faudra également essayer de profiter de la structure de ces matrices pour réduire le nombre d'opérations arithmétiques dans les calculs matriciels classiques comme les produits matrice–vecteur. C'est ce qui sera étudié dans la Section 7.3.

7.1 Origine des grands systèmes creux

On dit qu'une matrice $n \times n$ est *creuse* (*sparse* en anglais) si elle possède un très grand nombre d'éléments nuls ou, en d'autres termes, si le nombre d'éléments non nuls est très petit par rapport à n^2, nombre total de ses éléments. Un système d'équations linéaires ayant une telle matrice s'appelle, naturellement, un système *creux*. Les systèmes creux proviennent, bien souvent, de la discrétisation par différences finies ou par éléments finis d'équations différentielles ou aux dérivées partielles. Ainsi, pour un problème en dimension 3, si l'on prend 100 points de discrétisation dans chaque direction, on arrive à un système de 10^6 équations. Nous allons maintenant voir comment de tels systèmes sont obtenus et pourquoi ils sont creux.

Commençons par un exemple très simple mais qui montre cependant pourquoi les systèmes sont creux. Considérons l'équation différentielle d'ordre 2 avec

conditions aux limites

$$\left.\begin{array}{ll} u''(x) = -f(x), & x \in [0,1] \\ u(0) = u(1) = 0 & \end{array}\right\} \tag{7.1}$$

où f est une fonction connue. On pose $h = 1/(n+1)$ et l'on considère les $n+2$ points équidistants de $[0,1]$

$$x_i = ih, \quad i = 0, \ldots, n+1.$$

Les conditions aux limites s'écrivent $u(x_0) = u(0) = 0$ et $u(x_{n+1}) = u(1) = 0$. Les dérivées secondes $u''(x_i)$ vont être approchées par

$$u''(x_i) \simeq (u_{i+1} - 2u_i + u_{i-1})/h^2$$

où u_i est une approximation de $u(x_i)$, solution exacte en x_i. En posant $f_i = f(x_i)$, on arrive donc, en chaque point x_i ou, ce qui revient au même, pour chaque indice $i = 1, \ldots, n$ à

$$-u_{i-1} + 2u_i - u_{i+1} = h^2 f_i.$$

Lorsque $i = 1$ et $i = n$, on a respectivement, d'après les conditions aux limites, $u_0 = u_{n+1} = 0$. Par conséquent, en posant $u = (u_1, \ldots, u_n)^T$ et $f = (f_1, \ldots, f_n)^T$, on doit résoudre le système linéaire $n \times n$, $Au = f$ avec

$$A = \frac{1}{h^2} \begin{pmatrix} 2 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & -1 & 2 & -1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & -1 & 2 & -1 \\ & & & & -1 & 2 \end{pmatrix}.$$

Ce système est creux mais il n'est pas de très grande taille car il provient de la discrétisation d'une seule équation différentielle. Cependant signalons que la matrice inverse est pleine et que le conditionnement est de l'ordre de $4n^2/\pi^2$.

Naturellement, si l'on a affaire à un système de p équations différentielles, le système linéaire obtenu après discrétisation sera de dimension np. Mais les très grands systèmes proviennent de la discrétisation d'équations (ou de systèmes d'équations) aux dérivées partielles. En effet, si l'on utilise n points de discrétisation pour chacune des q variables et si l'on a p équations, on obtient un système de dimension pn^q.

Considérons, par exemple, le cas du Laplacien dans un domaine carré

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = -f(x,y), \quad (x,y) \in [0,1] \times [0,1]$$

où f est une fonction connue en tout point et où l'on suppose la solution u connue sur la frontière du carré. On va effectuer une discrétisation de cette

équation aux dérivées partielles par différences finies de la même façon que dans le problème aux limites précédent. On pose de nouveau $h = 1/(n+1)$ ainsi que

$$x_i = ih, \quad y_j = jh, \quad i, j = 0, \ldots, n+1.$$

Les dérivées partielles de u en (x_i, y_j) sont approchées par

$$\begin{aligned} \frac{\partial^2 u(x_i, y_j)}{\partial x^2} &\simeq (u_{i+1,j} - 2u_{ij} + u_{i-1,j})/h^2 \\ \frac{\partial^2 u(x_i, y_j)}{\partial y^2} &\simeq (u_{i,j+1} - 2u_{ij} + u_{i,j-1})/h^2 \end{aligned}$$

où u_{ij} est une approximation de $u(x_i, y_j)$, solution exacte en (x_i, y_j). En chaque point (i, j) pour $i, j = 1, \ldots, n$, on a donc

$$u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{ij} = h^2 f_{ij}$$

où $f_{ij} = f(x_i, y_j)$. Si l'on suppose que u est nulle sur le bord du carré, alors $u_{i0} = u_{0j} = u_{0,n+1} = u_{n+1,0} = 0$.

Dans l'exemple (7.1), le problème de la numérotation des points x_i de la discrétisation ne se posait pas. Ils étaient numérotés dans l'ordre naturel. Maintenant, il est nécessaire d'ordonner les points (x_i, y_j). La structure de la matrice et sa simplicité dépendront beaucoup de cet ordre. Si les points sont numérotés dans l'ordre naturel, c'est–à–dire en commençant par le haut du carré $[0, 1] \times [0, 1]$ et ligne après ligne, on obtient une matrice A de la forme

$$A = -\frac{1}{h^2} \begin{pmatrix} B & -I & & & \\ -I & B & -I & & \\ & \ddots & \ddots & \ddots & \\ & & -I & B & -I \\ & & & -I & B \end{pmatrix} \quad \text{avec} \quad B = \begin{pmatrix} 4 & -1 & & & \\ -1 & 4 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 4 & -1 \\ & & & -1 & 4 \end{pmatrix}.$$

Chaque bloc B est de dimension $n \times n$ et A est aussi de dimension $n \times n$ mais par blocs, c'est–à–dire qu'elle est de dimension $n^2 \times n^2$.

7.2 Stockage des matrices creuses

Afin de tirer avantage de la structure creuse des matrices, on utilise, bien entendu, des schémas spéciaux pour les stocker en mémoire. Le but principal est de ne stocker que les éléments non nuls et leur position dans la matrice et d'être ensuite capable d'effectuer facilement les opérations matricielles habituelles. Nous n'allons voir ici que les schémas de stockage les plus courants.

Dans la suite n désignera la dimension de la matrice et n_z le nombre de ses éléments non nuls. En général n_z est beaucoup plus petit que n^2.

Le schéma de stockage le plus simple est le format par coordonnées (appelé en anglais *coordinate format*). Il consiste en trois tableaux de dimension n_z

1. un tableau contenant, dans un ordre quelconque, les valeurs des éléments non nuls de la matrice,
2. un tableau d'entiers contenant les numéros correspondants des lignes de ces éléments,
3. un tableau d'entiers contenant les numéros correspondants de leurs colonnes.

Considérons par exemple la matrice

$$A = \begin{pmatrix} 1 & 0 & 0 & 2 \\ 3 & 4 & 0 & 0 \\ 0 & 0 & 5 & 6 \\ 7 & 0 & 8 & 0 \end{pmatrix}.$$

On a $n_z = 8$ et, par exemple,

a_{ij}	2	6	4	8	3	5	1	7
lignes	1	3	2	4	2	3	1	4
colonnes	4	4	2	3	1	3	1	1

Dans cet exemple, nous avons rangé volontairement les éléments dans un ordre arbitraire. Cependant, ils sont habituellement listés par ligne ou par colonne. C'est ainsi que, pour notre exemple, en rangeant les éléments par ligne, on aurait

a_{ij}	1	2	3	4	5	6	7	8
lignes	1	1	2	2	3	3	4	4
colonnes	1	4	1	2	3	4	1	3

On voit que, avec ce stockage, le tableau des numéros de lignes contient des informations redondantes. On peut les éliminer grâce à un tableau donnant, à leur place, l'endroit où commencent les éléments non nuls de chaque ligne. Ainsi, cette nouvelle structure de données a trois tableaux dont les fonctionnalités sont les suivantes

1. un tableau contenant les valeurs des éléments non nuls de la matrice, donnés ligne par ligne. Nous appellerons ce tableau AA ; sa longueur est n_z,
2. un tableau d'entiers contenant les indices des colonnes correspondants aux éléments du tableau AA. Nous appellerons ce tableau JA ; sa longueur est n_z,
3. un tableau d'entiers contenant un pointeur sur la position où commence chacune des lignes dans les tableaux AA et JA. Nous appellerons ce tableau IA. Son i–ème élément, IA(i), contiendra donc, pour $i = 1, \ldots, n$, la position où commence la i–ème ligne dans les tableaux AA et JA. Sa longueur sera $n+1$ avec, par convention, IA$(n+1) = n_z + 1$, c'est–à–dire l'adresse de début d'une $(n+1)$-ème ligne fictive dans les tableaux AA et JA.

Ainsi, pour notre exemple, on aura

AA	1	2	3	4	5	6	7	8
JA	1	4	1	2	3	4	1	3
IA	1	3	5	7	9			

Ce schéma de stockage est sans doute le plus utilisé pour les matrices creuses générales et il s'appelle *Compressed Sparse Row* (CSR) ou, des fois, stockage *Morse*. Avec ce genre de stockage, au lieu de n^2 mots de mémoire, on en utilise seulement $2n_z + n + 1$.

Soit A une matrice $n \times n$ déjà stockée dans un tableau à double entrée. On peut la stocker en format CSR de la façon suivante

$k = 1$
for $i = 1 : n$
  $IA(i) = k$
  for $j = 1 : n$
    if $A(i, j) \neq 0$ **then**
      $AA(k) = A(i, j)$
      $JA(k) = j$
      $k = k + 1$
    end
  end
end
$IA(n + 1) = k$

Naturellement, au lieu d'explorer la matrice ligne par ligne, on peut aussi l'explorer colonne par colonne. On aboutit alors à un schéma dit *Compressed Sparse Column* (CSC).

Ces deux types de schémas sont, bien sûr, préférables au format des coordonnées, d'abord parce qu'ils sont plus compacts et ensuite parce qu'ils sont souvent plus simples pour effectuer des opérations matricielles. Cependant, d'un autre côté, le schéma par coordonnées est avantageux par sa simplicité et sa flexibilité. Le format par coordonnées et le format CSR sont les deux formats les plus utilisés.

Dans de nombreuses matrices, les éléments diagonaux sont non nuls et/ou sont utilisés plus fréquemment que les autres éléments. On les stocke donc séparément. Le format *Modified Sparse Row* (MSR) ne nécessite que deux tableaux : AA pour les éléments de la matrice et un tableau d'entiers JA. On commence par introduire la diagonale principale de la matrice dans AA, c'est–à–dire que $\mathrm{AA}(i) = a_{ii}$ pour $i = 1, \ldots, n$. $\mathrm{AA}(n + 1)$ n'est pas utilisé mais on y place, parfois, une autre information sur la matrice. Ici, nous y mettrons $*$. Les éléments non nuls de A situés hors de la diagonale principale sont ensuite stockés ligne par ligne dans AA à partir de l'indice $n + 2$.

Pour $i = 1, \ldots, n$, JA(i) contient un pointeur indiquant le début de la ligne i de la matrice A dans le tableau AA. JA(n+1) sera égal à la taille du tableau plus 1, c'est–à–dire la valeur qu'aurait le pointeur indiquant le début d'une $(n+1)$–ième ligne fictive. Il faut remarquer que si, une fois enlevé l'élément diagonal, la dernière ligne ne contient que des zéros, alors on aura JA(n) =JA($n+1$). Ensuite, pour $i \geq n+2$, JA(i) contiendra le numéro de la colonne de A où se situe l'élément hors diagonale dont la valeur a été mise dans AA(i).

Ainsi, pour notre exemple, nous aurons

	1	2	3	4	5	6	7	8	9	10
AA	1	4	5	0	$*$	2	3	6	7	8
JA	6	7	8	9	11	4	1	4	1	3

Répétons que si la dernière ligne de la matrice ne contient que des éléments nuls à part a_{nn}, alors on aura JA(n) =JA($n+1$).

Dans de nombreux cas, les éléments non nuls d'une matrice se situent uniquement sur la diagonale principale (mais celle–ci peut être nulle aussi) et sur un certain nombre d'autres diagonales. Il est alors possible de ne stocker que ces diagonales. On utilise pour cela un tableau rectangulaire DIAG($1:n, 1:nd$) où nd est le nombre de diagonales dont tous les éléments ne sont pas nuls. L'éloignement de chacune des diagonales par rapport à la diagonale principale doit être connu. Cette information sera placée dans un tableau IOFF($1:nd$). Ainsi DIAG$(i,j) = a_{i,i+\text{ioff}(j)}$. L'ordre dans lequel les diagonales sont stockées est sans importance. Cependant, si l'on doit utiliser la diagonale principale plus souvent que les autres, il peut être avantageux de la stocker dans la première colonne de DIAG. Il ne faut pas oublier que les diagonales autres que la principale contiennent moins de n éléments, c'est–à–dire que les sous–diagonales (donc correspondant à une valeur négative dans IOFF) ne commençent pas à la première ligne mais plus loin et que les sur–diagonales (donc correspondant à une valeur positive dans IOFF) se terminent avant la dernière ligne. Il y a, par conséquent, des positions dans le tableau DIAG qui ne sont pas utilisées. Nous mettrons de nouveau des $*$ pour indiquer les éléments qui devraient normalement s'y trouver. Soit, par exemple, la matrice suivante pour laquelle $nd = 3$

$$A = \begin{pmatrix} 1 & 0 & 2 & 0 \\ 3 & 4 & 0 & 5 \\ 0 & 6 & 7 & 0 \\ 0 & 0 & 8 & 9 \end{pmatrix}.$$

On aura

$$\text{DIAG} = \begin{array}{|c|c|c|} \hline * & 1 & 2 \\ \hline 3 & 4 & 5 \\ \hline 6 & 7 & * \\ \hline 8 & 9 & * \\ \hline \end{array} \quad \text{IOFF} = \begin{array}{|c|c|c|} \hline -1 & 0 & 2 \\ \hline \end{array}$$

Ce schéma s'appelle *stockage diagonal*.

7.3 Opérations sur les matrices creuses

Nous allons maintenant étudier comment effectuer les opérations matricielles habituelles sur des matrices creuses stockées selon les schémas décrits dans la Section précédente. Nous ne parlerons que des produits matrice–vecteur puisque le produit de deux matrices se ramène à une succession de produits matrice–vecteur. Soit donc A une matrice et x un vecteur.

Supposons A stockée dans AA avec le format CSR. Le calcul de $y = Ax$ s'effectue ainsi

for $i = 1 : n$
  $k1 = IA(i)$
  $k2 = IA(i+1) - 1$
  $y(i) =$ produit scalaire de $AA(k1 : k2)$ et de $x(JA(k1 : k2)$
end

Chaque itération de la boucle calcule une composante du vecteur y. Ceci est avantageux parce que chacune de ces composantes peut être calculée indépendamment des autres ce qui permet aussi de paralléliser le calcul.

Si la matrice est stockée par colonne, c'est–à–dire en format CSC, alors il faut faire

for $j = 1 : n$
  $k1 = IA(j)$
  $k2 = IA(j+1) - 1$
  $y(JA(k1 : k2)) = y(JA(k1 : k2)) + x(j) * AA(k1 : k2)$
end

Ici, à chaque itération de la boucle, on ajoute un multiple de la j–ème colonne (que l'on suppose avoir été initialisée à zéro avant la boucle). Par conséquent, ce calcul est moins facilement parallélisable. Pour le faire, il faut décomposer les opérations de la boucle interne (celle de $k1$ à $k2$). Comme cette boucle ne nécessite, en général, que peu d'opérations le gain n'est pas appréciable. On voit donc qu'il est parfois nécessaire de changer la structure des données afin d'améliorer les performances d'un calcul sur certains ordinateurs.

Voyons maintenant comme effectuer le produit matrice–vecteur avec un stockage diagonal.

$y(1 : n) = 0$
for $j = 1 : nd$
  $k = IOFF(j)$
  for $i = 1 : n$

$y(i) = y(i) + DIAG(i,j) * x(k+i)$
end
end

Dans cette façon de procéder, chacune des diagonales est multipliée par le vecteur x et le résultat est ajouté au vecteur y (initialisé à zéro avant la boucle en j). Pour le calcul, les $*$ du tableau DIAG sont remplaçées par 0. Cette façon de procéder est sans doute la meilleure pour la parallélisation ou la vectorisation. D'un autre côté, il n'est pas assez général. Il est possible d'éviter les multiplications par les éléments nuls de DIAG (qui remplacent des zéros) en effectuant la boucle en i de $n_1 = \max(1, 1-k)$ à $n_2 = \min(n, n-k)$.

Un autre type fréquent de calcul que l'on doit être capable de réaliser est la résolution des systèmes linéaires avec une matrice triangulaire. Considérons donc la résolution de $Lx = y$ où L est triangulaire inférieure à diagonale unité. Supposons la matrice L stockée dans AA avec un format CSR ; on a

$x(1) = y(1)$
for $i = 2 : n$
$k1 = IA(i)$
$k2 = IA(i+1) - 1$
$x(i) = y(i)-$[produit scalaire de $AA(k1 : k2)$ et de $x(JA(k1 : k2))$]
end

Chapitre 8

Calcul du polynôme caractéristique

Nous avons vu que les valeurs propres d'une matrice de dimension n étaient les n racines de son polynôme caractéristique. Or, lorsque $n \geq 5$, il n'existe pas de formule qui donne ces racines dans tous les cas. C'est le célèbre résultat démontré par Évariste Galois (1811–1832) dans la nuit précédant le duel où il fut tué (on pourrait dire exécuté). Pour calculer les racines d'un polynôme, il est donc impératif d'utiliser des méthodes itératives. Dans certains domaines des mathématiques appliquées, comme on contrôle linéaire, on peut avoir besoin de calculer ce polynôme caractéristique (c'est–à–dire ses coefficients ou ses valeurs en certains points) mais ne pas avoir besoin de calculer ses racines.

Dans ce Chapitre, nous allons étudier les méthodes numériques, qui sont des méthodes directes, qui permettent de calculer le polynôme caractéristique d'une matrice.

8.1 La méthode de Souriau

L'idée théoriquement la plus simple pour calculer toutes les valeurs propres d'une matrice consiste à calculer les coefficients de son polynôme caractéristique.

En pratique cependant ces méthodes se révèlent délicates à utiliser parce qu'elles sont numériquement instables.

Nous allons cependant donner l'une de ces méthodes à titre d'exemple et parce qu'elle présente une utilité en calcul formel.

La méthode de Souriau est une modification d'une méthode due à l'astronome Urbain Le Verrier (1811–1877) qui découvrit la planète Neptune en cherchant

à expliquer les perturbations observées dans la mouvement d'Uranus. On pose

$$\begin{array}{lll} A_1 = A, & a_1 = \mathrm{tr}A_1, & B_1 = A_1 - a_1 I \\ A_2 = B_1 A, & a_2 = \frac{1}{2}\mathrm{tr}A_2, & B_2 = A_2 - a_2 I \\ A_3 = B_2 A, & a_3 = \frac{1}{3}\mathrm{tr}A_3, & B_3 = A_3 - a_3 I \\ \vdots & \vdots & \vdots \\ A_n = B_{n-1} A, & a_n = \frac{1}{n}\mathrm{tr}A_n, & B_n = A_n - a_n I. \end{array}$$

On a les résulats suivants

Théorème 22

$$\begin{aligned} B_n &= 0 \\ P_n(\lambda) &= (-1)^n[\lambda^n - a_1\lambda^{n-1} - \cdots - a_{n-1}\lambda - a_n] \\ A^{-1} &= \frac{1}{a_n} B_{n-1}. \end{aligned}$$

Démonstration.

Supposons que le polynôme caractéristique de A soit

$$P_n(\lambda) = \det(A - \lambda I) = (-1)^n[\lambda^n - c_1\lambda^{n-1} - \cdots - c_n].$$

On a

$$c_1 = \sum_{i=1}^{n} \lambda_i = \mathrm{tr}A = \mathrm{tr}A_1 = a_1.$$

Nous allons démontrer par récurrence que $c_k = a_k$. Pour cela supposons avoir démontré que $c_1 = a_1, \ldots, c_{k-1} = a_{k-1}$. On a

$$A_k = B_{k-1}A = (A_{k-1} - a_{k-1}I)A = A_{k-1}A - a_{k-1}A.$$

Or $A_{k-1} = A_{k-2}A - a_{k-2}A$ et l'on obtient

$$\begin{aligned} A_k &= (A_{k-2}A - a_{k-2}A)A - a_{k-1}A \\ &= A_{k-2}A^2 - a_{k-2}A^2 - a_{k-1}A \end{aligned}$$

et ainsi de suite, d'où

$$A_k = A^k - a_1 A^{k-1} - a_2 A^{k-2} - \cdots - a_{k-1}A.$$

Or $a_i = c_i$ pour $i = 1, \ldots, k-1$, ce qui donne

$$A_k = A^k - c_1 A^{k-1} - c - 2A^{k-2} - \cdots - c_{k-1}A$$

et

$$\mathrm{tr}A_k = \mathrm{tr}A^k - c_1\mathrm{tr}A^{k-1} - c_2\mathrm{tr}A^{k-2} - \cdots - c_{k-1}\mathrm{tr}A = kc_k$$

en utilisant les relations de Newton entre racines et coefficients d'un polynôme. D'où

$$\frac{1}{k}\mathrm{tr}A_k = a_k = c_k.$$

Démontrons maintenant que $B_n = 0$. On a

$$\begin{aligned} B_n &= A_n - a_nI = B_{n-1}A - a_nI = (A_{n-1} - a_{n-1}I)A - a_nI \\ &= A^n - a_1A^{n-1} - \cdots - a_nI = 0 \end{aligned}$$

d'après le théorème de Cayley–Hamilton. On a donc

$$A_n = a_nI \quad \text{et} \quad B_{n-1}A = a_nI$$

ce qui donne $B_{n-1}AA^{-1} = a_nA^{-1}$ et donc

$$A^{-1} = \frac{1}{a_n}B_{n-1}.$$

Ce procédé fournit par conséquent l'inverse de A. ∎

Malheureusement cette méthode n'est pas utilisable en pratique car elle est numériquement instable. Elle rend par contre de grands services pour le calcul formel de l'inverse d'une matrice.

8.2 Les méthodes de réduction à la forme tridiagonale

La réduction d'une matrice à une matrice tridiagonale semblable est particulièrement intéressante parce que l'on peut calculer alors facilement le polynôme caractéristique.

Supposons que la matrice A soit de la forme

$$A = \begin{pmatrix} a_1 & b_1 & & & \\ c_1 & a_2 & b_2 & & \\ & \ddots & \ddots & \ddots & \\ & & c_{n-2} & a_{n-1} & b_{n-1} \\ & & & c_{n-1} & a_n \end{pmatrix}.$$

Appelons $P_k(\lambda)$ le polynôme caractéristique de la matrice formée par les k premières lignes et les k premières colonnes de A. On a

$$P_k(\lambda) = \det(A_k - \lambda I) = \det \begin{pmatrix} a_1 - \lambda & b_1 & & & \\ c_1 & a_2 - \lambda & b_2 & & \\ & \ddots & \ddots & \ddots & \\ & & c_{k-2} & a_{k-1} - \lambda & b_{k-1} \\ & & & c_{k-1} & a_k - \lambda \end{pmatrix}.$$

En développant ce déterminant par rapport à sa dernière ligne (ou sa dernière colonne) on obtient

$$P_k(\lambda) = (a_k - \lambda)P_{k-1}(\lambda) - b_{k-1}c_{k-1}P_{k-2}(\lambda).$$

Pour pouvoir utiliser cette relation de récurrence il faut évidemment donner $P_0(\lambda)$ et $P_1(\lambda)$. On a

$$P_1(\lambda) = a_1 - \lambda$$

et l'on voit qu'en prenant $P_0(\lambda) = 1$ et $n = 2$ on obtient

$$P_2(\lambda) = (a_2 - \lambda)(a_1 - \lambda) - b_1c_1$$

qui est bien le polynôme caractéristique de la matrice

$$\begin{pmatrix} a_1 & b_1 \\ c_1 & a_2 \end{pmatrix}.$$

Les valeurs initiales seront donc $P_0(\lambda) = 1$ et $P_1(\lambda) = a_1 - \lambda$ (on aurait pu prendre aussi $P_0(\lambda) = 1$ et $P_{-1}(\lambda) = 0$). On peut donc, à l'aide de la relation de récurrence précédente, calculer $P_n(\lambda)$ pour toute valeur de λ.

Étudions quelques propriétés des racines de ces polynômes.

Propriété 6
Si, $\forall i$, $b_ic_i \neq 0$, alors $\forall i$, P_i et P_{i-1} n'ont pas de racine commune.

Démonstration.

Elle se fait par l'absurde. Si P_i et P_{i-1} admettent a comme racine commune alors

$$P_i(a) = (a_i - a)P_{i-1}(a) - b_{i-1}c_{i-1}P_{i-2}(a) = 0.$$

Donc, si b_{i-1} et c_{i-1} sont différents de zéro, $P_{i-2}(a) = 0$ et ainsi de suite ; ce qui conduit à $P_0(a) = 0$ qui est impossible. ∎

Propriété 7
Si $\forall i, b_ic_i > 0$, alors, $\forall i$, les racines de P_{i+1} sont réelles, distinctes et séparées par celles de P_i.

Démonstration.

Elle se fait par récurrence. P_1 admet une racine réelle unique a_1. Étudions les racines de P_2

$$\begin{aligned} P_2(\lambda) &= (a_2 - \lambda)(a_1 - \lambda) - b_1c_1 \\ &= \lambda^2 - (a_1 + a_2)\lambda + a_1a_2 - b_1c_1. \end{aligned}$$

Le discriminant est

$$\Delta = (a_1 + a_2)^2 - 4(a_1 a_2 - b_1 c_1) = (a_1 - a_2)^2 + 4 b_1 c_1.$$

Si $b_1 c_1 > 0$ alors $\Delta > 0$. P_2 possède donc deux racines réelles distinctes. On a $P_2(a_1) = -b_1 c_1 < 0$, donc a_1 est situé entre les racines de P_2 puisque le coefficient de λ^2 dans P_2 est positif.

Supposons la propriété vraie jusqu'à i et démontrons qu'elle l'est encore pour $i+1$.

Appelons

$$\begin{aligned} &t_1, \ldots, t_{i+1} \text{ les racines de } P_{i+1} \\ &s_1, \ldots, s_i \text{ les racines de } P_i \\ &r_1, \ldots, r_{i-1} \text{ les racines de } P_{i-1}. \end{aligned}$$

On suppose donc la propriété vraie pour i, c'est–à–dire que

$$s_1 < r_1 < s_2 < r_2 < \ldots < s_{i-1} < r_{i-1} < s_i.$$

Or $P_{i-1}(\lambda) = (r_1 - \lambda) \cdots (r_{i-1} - \lambda)$ et $P_i(\lambda) = (s_1 - \lambda) \cdots (s_i - \lambda)$. D'où, en utilisant la relation de récurrence

$$P_{i+1}(\lambda) = (a_{i+1} - \lambda)(s_1 - \lambda) \cdots (s_i - \lambda) - b_i c_i (r_1 - \lambda) \cdots (r_{i-1} - \lambda).$$

En particulier on a

$$P_{i+1}(s_j) = 0 - b_i c_i \underbrace{(r_1 - s_j) \cdots (r_{j-1} - s_j)}_{\text{du signe de } (-1)^{j-1}} \underbrace{(r_j - s_j) \cdots (r_{i-1} - s_j)}_{>0}$$

et, par conséquent, $P_{i+1}(s_j)$ est du signe de $(-1)^j$ pour $j = 1, \ldots, i$ car $b_i c_i > 0$. Il y a donc une racine de P_{i+1} entre chaque racine de P_i, ce qui donne $i-1$ racines ($s_j < t_{j+1} < s_{j+1}, j = 1, \ldots, i-1$). De plus, le terme de plus haut degré de P_{i+1} est $(-1)^{i+1} \lambda^{i+1}$. Donc $P_{i+1}(\lambda)$ tend vers $+\infty$ lorsque λ tend vers $-\infty$. Comme, d'autre part, $P_{i+1}(s_1) < 0$, il y a une racine t_1 de P_{i+1} inférieure à $s_1, (t_1 < s_1)$. Enfin, $P_{i+1}(s_i)$ est du signe de $(-1)^i$ et $P_{i+1}(+\infty)$ est du signe de $(-1)^{i+1}$. Il existe donc une racine t_{i+1} de P_{i+1} supérieure à $s_i, (s_i < t_{i+1})$ ce qui termine la démonstration. ∎

Propriété 8
Soit a tel que $P_n(a) \neq 0$ et $V(a)$ le nombre de racines supérieures à a ; si $\forall j, b_j c_j > 0$, alors $V(a)$ est égal au nombre de changements de signes dans la suite

$$1, -P_1(a), P_2(a), -P_3(a), \ldots, (-1)^n P_n(a).$$

Démonstration.

Elle tient simplement au fait que l'on a une suite de Sturm. Le nombre de racines appartenant à $]a, b[$ est donc égal à $V(a) - V(b)$. ∎

Remarque 8

1. *Si la matrice A est symétrique et si $\forall j, b_j \neq 0$, alors les trois propriétés précédentes sont vraies puisque $c_j = b_j$ et $c_j b_j = b_j^2 > 0$.*
2. *On rappelle qu'une suite de polynômes est une suite de Sturm si deux polynômes successifs ne s'annulent pas simultanément et si, quand un polynôme s'annule, les deux polynômes adjacents sont de signes contraires.*

Si, pour calculer $P_n(\lambda)$, on utilise la relation de récurrence précédente, on est obligé de recommencer les calculs pour chaque valeur de λ. On peut éviter cela en calculant une fois pour tous les coefficients de P_n à l'aide de la relation de récurrence. Posons

$$P_i(\lambda) = A_0^{(i)}\lambda^i + A_1^{(i)}\lambda^{i-1} + \cdots + A_{i-1}^{(i)}\lambda + A_i^{(i)}$$

alors, en utilisant la relation de récurrence précédente et en identifiant de part et d'autre du signe égal les coefficients des termes de même degré en λ, on obtient

$$A_p^{(i+1)} = A_p^{(i)} - a_{i+1}A_{p-1}^{(i)} - b_i c_i A_{p-2}^{(i-1)}, \quad p = 0, \ldots, i+1$$

avec $A_p^{(q)} = 0$ si $p > q$, $A_0^{(i)} = 1$ et $A_1^{(i)} = -\sum_{j=1}^{i} a_j$.

Remarque 9
Si l'un des b_i ou l'un des c_i est nul, le polynôme caractéristique se factorise en un produit de deux polynômes que l'on peut calculer séparément. Si plusieurs b_i ou c_i sont nuls, on a un produit de plusieurs polynômes.

Le calcul des vecteurs propres d'une matrice tridiagonale est particulièrement simple. Soit λ une valeur propre et $x_1, \ldots, x_n$ les composantes du vecteur propre x associé à λ. Le système $Ax = \lambda x$ s'écrit alors

$$\begin{aligned}
&(a_1 - \lambda)x_1 + b_1 x_2 = 0\\
&c_1 x_1 + (a_2 - \lambda)x_2 + b_2 x_3 = 0\\
&c_2 x_2 + (a_3 - \lambda)x_3 + b_3 x_4 = 0\\
&\cdots\cdots\cdots\cdots\cdots\\
&c_{n-2}x_{n-2} + (a_{n-1} - \lambda)x_{n-1} + b_{n-1}x_n = 0\\
&c_{n-1}x_{n-1} + (a_n - \lambda)x_n = 0.
\end{aligned}$$

On choisit $x_1 = 1$ (ou $x_n = 1$) et le système se résoud immédiatement. Il faut cependant prendre certaines précautions pour éviter les erreurs d'arrondis.

8.2.1 La méthode de Lanczos

Elle a été donnée par Lanczos en 1950. Soient x et y deux vecteurs quelconques non nuls. On pose

$$\begin{aligned} x_0 = 0, \quad y_0 = 0 \\ x_1 = x, \quad y_1 = y \end{aligned}$$

puis l'on calcule

$$\left.\begin{array}{rcl} x_{k+1} & = & Ax_k - a_k x_k - b_{k-1} x_{k-1} \\ y_{k+1} & = & A^T y_k - a_k y_k - b_{k-1} y_{k-1} \end{array}\right\} k = 1, 2, \ldots,$$

avec $b_0 = 0$ et

$$a_k = \frac{(Ax_k, y_k)}{(x_k, y_k)}, \quad b_{k-1} = \frac{(Ax_k, y_{k-1})}{(x_{k-1}, y_{k-1})}.$$

Nous supposerons que ces vecteurs peuvent effectivement être construits, c'est–à–dire que, pour tout k, $(x_k, y_k) \neq 0$.

Théorème 23
$(x_i, y_j) = 0, \quad i \neq j.$

Démonstration.

On a $(x_0, y_1) = (x_1, y_0) = 0$. Puis

$$\begin{aligned} (x_2, y_1) &= (Ax_1 - a_1 x_1, y_1) = (Ax_1, y_1) - \frac{(Ax_1, y_1)}{(x_1, y_1)}(x_1, y_1) = 0 \\ (x_1, y_2) &= (x_1, A^T y_1 - a_1 y_1) = (x_1, A^T y_1) - \frac{(Ax_1, y_1)}{(x_1, y_1)}(x_1, y_1) = 0. \end{aligned}$$

Supposons avoir démontré que $(x_{p+1}, y_p) = (x_p, y_{p+1}) = 0$ pour $p = 0, \ldots, i-1$. On a

$$\begin{aligned} (x_{i+1}, y_i) &= (Ax_i, y_i) - a_i(x_i, y_i) - b_{i-1}(x_{i-1}, y_i) \\ &= (Ax_i, y_i) - \frac{(Ax_i, y_i)}{(x_i, y_i)}(x_i, y_i) - 0 = 0 \end{aligned}$$

et

$$\begin{aligned} (x_i, y_{i+1}) &= (x_i, A^T y_i) - a_i(x_i, y_i) - b_{i-1}(x_i, y_{i-1}) \\ &= (x_i, A^T y_i) - \frac{(Ax_i, y_i)}{(x_i, y_i)}(x_i, y_i) - 0 = 0. \end{aligned}$$

Étudions maintenant ce qu'il se passe lorsqu'il y a un écart de 2 entre les indices. On a $(x_2, y_0) = (x_0, y_2) = 0$ et

$$\begin{aligned}
(x_3, y_1) &= (Ax_2 - a_2x_2 - b_1x_1, y_1) \\
&= (Ax_2, y_1) - a_2(x_2, y_1) - \frac{(Ax_2, y_1)}{(x_1, y_1)}(x_1, y_1) = 0 \\
(x_1, y_3) &= (x_1, A^T y_2 - a_2 y_2 - b_1 y_1) \\
&= (x_1, A^T y_2) - a_2(x_1, y_2) - \frac{(Ax_2, y_1)}{(x_1, y_1)}(x_1, y_1) \\
&= (Ax_1, y_2) - (Ax_2, y_1) = (Ax_1, A^T y_1 - a_1 y_1) - (A^2 x_1 - a_1 Ax_1, y_1) \\
&= (A^2 x_1, y_1) - a_1(Ax_1, y_1) - (A^2 x_1, y_1) + a_1(Ax_1, y_1) = 0.
\end{aligned}$$

Supposons avoir démontré que

$$(x_{p+1}, y_{p-1}) = (x_{p-1}, y_{p+1}) = 0, \quad p = 1, \ldots, i-1.$$

On a

$$\begin{aligned}
(x_{i+1}, y_{i-1}) &= (Ax_i, y_{i-1}) - a_i(x_i, y_{i-1}) - b_{i-1}(x_{i-1}, y_{i-1}) \\
&= (Ax_i, y_{i-1}) - 0 - \frac{(Ax_i, y_{i-1})}{(x_{i-1}, y_{i-1})}(x_{i-1}, y_{i-1}) = 0 \\
(x_{i-1}, y_{i+1}) &= (x_{i-1}, A^T y_i) - a_i(x_{i-1}, y_i) - b_{i-1}(x_{i-1}, y_{i-1}) \\
&= (Ax_{i-1}, y_i) - 0 - (Ax_i, y_{i-1}) \\
&= (Ax_{i-1}, A^T y_{i-1} - a_{i-1} y_{i-1} - b_{i-2} y_{i-2}) \\
&\quad -(A^2 x_{i-1} - a_{i-1} Ax_{i-1} - b_{i-2} Ax_{i-2}, y_{i-1}) \\
&= (Ax_{i-1}, A^T y_{i-1}) - a_{i-1}(x_{i-1}, y_{i-1}) - \\
&\quad b_{i-2}(Ax_{i-1}, y_{i-2}) - (A^2 x_{i-1}, y_{i-1}) \\
&\quad +a_{i-1}(Ax_{i-1}, y_{i-1}) + b_{i-2}(Ax_{i-2}, y_{i-1}) \\
&= b_{i-2}(Ax_{i-2}, y_{i-1}) - b_{i-2}(Ax_{i-1}, y_{i-2}) \\
&= \cdots = b_{i-2} b_{i-3} \cdots b_0[(Ax_0, y_1) - (Ax_1, y_0)] = 0.
\end{aligned}$$

On a $x_{i+1} \perp y_{i-1} \perp x_i \perp y_{i-2}$. Donc $(x_{i+1}, y_{i-2}) = 0$ et de même $(x_{i-2}, y_{i+1}) = 0$ et ainsi de proche en proche d'où le résultat du Théorème. ∎

Remarque 10
On appelle biorthogonales *deux suites de vecteurs qui vérifient la propriété du Théorème précédent.*

Corollaire 2

$$x_{n+1} = y_{n+1} = 0.$$

Démonstration.

Dans un espace de dimension n, il ne peut exister plus de n vecteurs orthogonaux. ∎

Les relations de récurrence précédentes peuvent s'écrire

$$\begin{aligned} Ax_k &= b_{k-1}x_{k-1} + a_k x_k + x_{k+1} \\ A^T y_k &= b_{k-1}y_{k-1} + a_k y_k + y_{k+1}. \end{aligned}$$

Introduisons les matrices

$$B = \begin{pmatrix} a_1 & b_1 & & & \\ 1 & a_2 & b_2 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & a_{n-1} & b_{n-1} \\ & & & 1 & a_n \end{pmatrix}$$

X : matrice dont les colonnes sont $x_1, \ldots, x_n$,

Y : matrice dont les lignes sont $y_1, \ldots, y_n$,

Il est facile de voir que les relations précédentes s'écrivent

$$\begin{aligned} AX &= XB \quad \text{d'où } B = X^{-1}AX \\ YA &= B^T Y \quad \text{d'où } B^T = YAY^{-1}. \end{aligned}$$

Par conséquent les matrices A et B sont semblables et B est tridiagonale. Posons $P_0(\lambda) = 1$ et $P_1(\lambda) = a_1 - \lambda$ et construisons la suite

$$P_{k+1}(\lambda) = (a_{k+1} - \lambda)P_k(\lambda) - b_k P_{k-1}(\lambda), \quad k = 1, \ldots, n-1.$$

On voit que P_n est le polynôme caractéristique de la matrice A.

Du point de vue numérique cette méthode est assez rapide mais sa précision est souvent médiocre pour des matrices d'ordre élevé : cela tient au fait que la biorthogonalité des suites (x_i) et (y_i) est numériquement médiocre surtout quand ces vecteurs ont des composantes petites en module. On peut diminuer cette perte de précision en corrigeant les vecteurs x_i à chaque itération. Notons $\widetilde{x}_i$ le vecteur corrigé

$$\widetilde{x}_i = x_i - \sum_{j=1}^{i-1} \gamma_{ij} x_j$$

avec

$$\gamma_{ij} = \frac{(x_i, y_j)}{(x_j, y_j)}.$$

On peut vérifier que l'on a alors exactement

$$(\widetilde{x}_i, y_j) = 0, \quad j \neq i.$$

On vu précédemment qu'il y avait intérêt à ce que la matrice triangulaire soit symétrique. Durand a proposé une variante de la méthode de Lanczos qui aboutit à une matrice B symétrique. On pose

$$\begin{aligned} x_0 &= 0, \qquad y_0 = 0 \\ x_1 &= e_1, \qquad y_1 = e_1 \\ b_k x_{k+1} &= Ax_k - a_k x_k - b_{k-1} x_{k-1} \\ b_k y_{k+1} &= A^T y_k - a_k y_k - b_{k-1} y_{k-1}. \end{aligned}$$

On choisit les a_k et les b_k de sorte que $(x_i, y_j) = \delta_{ij}$ d'où $a_k = (Ax_k, y_k)$ et $b_k = (Ax_k, y_{k+1}) = (Ax_{k+1}, y_k)$ ce qui donne $b_k = [(Ax_k, Ay_k) - a_k^2 - b_{k-1}^2]^{1/2}$ avec $b_0 = 0$. On obtient alors

$$B = \begin{pmatrix} a_1 & b_1 & & & \\ b_1 & a_2 & b_2 & & \\ & \ddots & \ddots & \ddots & \\ & & b_{n-2} & a_{n-1} & b_{n-1} \\ & & & b_{n-1} & a_n \end{pmatrix}$$

et $B = X^{-1}AX$.

Cette méthode présente l'inconvénient que le calcul des b_k nécessite l'utilisation de l'arithmétique complexe. D'autre part, si l'un des b_k est nul, alors on ne peut pas calculer x_{k+1} et y_{k+1}. Cependant, dans ce cas, $\det(B - \lambda I)$ se décompose en un produit de deux déterminants. Numériquement les défauts de la méthode de Lanczos sont encore aggravés car il est en plus difficile d'avoir $(x_i, y_i) = 1$.

8.3 Les méthodes de Givens et Householder

Les méthodes de Givens et de Householder permettent de transformer toute matrice A en une matrice de Hessenberg semblable. Si la matrice A est symétrique, sa matrice de Hessenberg sera tridiagonale et l'on pourra donc calculer son polynôme caractéristique par les procédures décrites auparavant.

8.3.1 La forme de Hessenberg

Définition 6
On dit qu'une matrice $A = (a_{ij})$ est de la forme de Hessenberg supérieure *si*

$$a_{ij} = 0, \quad i > j + 1, \quad j = 1, \ldots, n - 1$$

c'est à dire que A est de la forme

$$A = \begin{pmatrix} a_{11} & \cdots & \cdots & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & \cdots & a_{2n} \\ 0 & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{n,n-1} & a_{nn} \end{pmatrix}.$$

Les matrices de Hessenberg jouent un rôle important dans de nombreuses méthodes de calcul de valeurs propres. Pour certaines méthodes itératives, on ne sait même démontrer leur convergence que quand la matrice dont on cherche les valeurs propres est de la forme de Hessenberg. Cependant c'est le théorème suivant qui donne toute leur importance aux matrices de Hessenberg

Théorème 24
Soit A une matrice quelconque. Il existe une matrice orthogonale P telle que $H = P^T AP$ soit une matrice de Hessenberg supérieure.

Ce théorème sera démontré de façon constructive, c'est–à–dire en décrivant une méthode permettant de passer de A à H. Il existe essentiellement deux méthodes de réduction d'une matrice quelconque en une matrice de Hessenberg supérieure. Dans ces deux méthodes on effectue une succession de transformations orthogonales. On pose $A_0 = A$ puis

$$A_k = P_k^T A_{k-1} P_k, \quad k = 1, \ldots, n-2$$

où A_{k-1} est de la forme (pour $n = 6$ et $k = 3$)

$$A_{k-1} = \left(\begin{array}{ccc|ccc} * & * & * & * & * & * \\ * & * & * & * & * & * \\ 0 & * & * & * & * & * \\ \hline 0 & 0 & * & * & * & * \\ 0 & 0 & * & * & * & * \\ 0 & 0 & * & * & * & * \end{array}\right)$$

où $*$ désigne un élément qui peut ne pas être nul. Le bloc en haut et à gauche est de dimension $k \times k$ et celui en bas à droite est de dimension $(n-k) \times (n-k)$. La dimension des deux autres blocs s'en déduit facilement.

La matrice A_{n-2} est semblable à A et elle est de la forme de Hessenberg supérieure. La première méthode, donnée par Wallace Givens en 1954, utilise des matrices de rotation plane. La seconde méthode, introduite en 1958 par Householder, utilise des matrices orthogonales élémentaires.

Si la matrice A est symétrique, alors sa forme de Hessenberg est une matrice tridiagonale. Nous allons donc voir maintenant comment réduire une matrice en un matrice de Hesseberg semblable.

8.3.2 La méthode de Givens

Soit $B = (b_{ij})$ une matrice quelconque et Q_{pq} la matrice

$$Q_{pq} = \begin{pmatrix} 1 & & & & & & & & & \\ & \ddots & & & & & & & & \\ & & 1 & & & & & & & \\ & & & \cos\theta & \cdots & \cdots & \cdots & -\sin\theta & & \\ & & & \vdots & 1 & & & \vdots & & \\ & & & \vdots & & \ddots & & \vdots & & \\ & & & \vdots & & & 1 & \vdots & & \\ & & & \sin\theta & \cdots & \cdots & \cdots & \cos\theta & & \\ & & & & & & & & 1 & \\ & & & & & & & & & \ddots \\ & & & & & & & & & & 1 \end{pmatrix}$$

où les $\sin\theta$ and $\cos\theta$ sont aux intersections des lignes et des colonnes p et q et où tous les autres éléments en dehors de la diagonale principale sont nuls.

Posons $C = (c_{ij}) = Q_{pq}^T B Q_{pq}$. Quand on passe ainsi de B à C, seuls sont modifiés les éléments des lignes et des colonnes p et q et l'on a

$$\begin{aligned}
c_{ij} &= b_{ij}, \quad i,j \neq p,q \\
c_{ip} &= b_{ip}\cos\theta + b_{iq}\sin\theta, \quad i \neq p,q \\
c_{pj} &= b_{pj}\cos\theta + b_{qj}\sin\theta, \quad j \neq p,q \\
c_{iq} &= -b_{ip}\sin\theta + b_{iq}\cos\theta, \quad i \neq p,q \\
c_{qj} &= -b_{pj}\sin\theta + b_{qj}\cos\theta, \quad j \neq p,q \\
c_{pp} &= b_{pp}\cos^2\theta + (b_{pq}+b_{qp})\sin\theta\cos\theta + b_{qq}\sin^2\theta \\
c_{qq} &= b_{pp}\sin^2\theta - (b_{pq}+b_{qp})\sin\theta\cos\theta + b_{qq}\cos^2\theta \\
c_{pq} &= b_{pq}\cos^2\theta + (b_{qq}-b_{pp})\sin\theta\cos\theta - b_{qp}\sin^2\theta \\
c_{qp} &= b_{qp}\sin^2\theta + (b_{qq}-b_{pp})\sin\theta\cos\theta - b_{pq}\sin^2\theta.
\end{aligned}$$

On choisit θ afin que $c_{q,p-1} = 0$ d'où

$$-b_{p,p-1}\sin\theta + b_{q,p-1}\cos\theta = 0.$$

Si $b_{p,p-1} = 0$ on prendra $\theta = 0$ d'où $Q_{pq} = I$.

p étant fixé, si l'on fait varier q de $p+1$ à n on voit qu'un élément de la $(p-1)$-ème colonne qui était nul ou qui avait été annulé auparavant n'est pas modifié par les transformations suivantes. Donc en prenant

$$P_k = \prod_{q=k+2}^{n} Q_{k+1,q}$$

on annule successivement $a_{i,k+1}$ pour $i = k+2, \ldots, n$ et par conséquent A_{n-2} est de la forme de Hessenberg supérieure.

Du point de vue algorithmique les règles se résument à

pour $k = 1, \ldots, n-2$ et pour $i = k+2, \ldots, n$, faire

1. calculer $\alpha = [(a_{k+1,k}^{(k-1)})^2 + (a_{i,k}^{(k-1)})^2]^{1/2}$,
2. calculer $\cos\theta = a_{k+1,k}^{(k-1)}/\alpha$ et $\sin\theta = a_{ik}^{(k-1)}/\alpha$ (si $\alpha = 0$, prendre $\theta = 0$) et $a_{k+1,k}^{(k)} = \alpha, \quad a_{ik}^{(k)} = 0$,
3. pour $j = k+1, \ldots, n$, calculer
$$\begin{aligned} a_{k+1,j}^{(k)} &= a_{k+1,j}^{(k-1)} \cos\theta + a_{ij}^{(k-1)} \sin\theta \\ a_{ij}^{(k)} &= -a_{k+1,j}^{(k-1)} \sin\theta + a_{ij}^{(k-1)} \cos\theta, \end{aligned}$$
4. pour $j = 1, \ldots, n$, calculer
$$\begin{aligned} a_{j,k+1}^{(k)} &= a_{j,k+1}^{(k-1)} \cos\theta + a_{ji}^{(k-1)} \sin\theta \\ a_{ji}^{(k)} &= -a_{j,k+1}^{(k-1)} \sin\theta + a_{ji}^{(k-1)} \cos\theta. \end{aligned}$$

L'étape 3 correspond au produit $P_k^T A_{k-1}$ et l'étape 4 représente $(P_k^T A_{k-1})P_k$.

Le passage de A_0 à A_{n-2} nécessite environ $10n^3/3$ multiplications.

Remarque 11
Si la matrice A est symétrique, alors les matrices A_k successives sont également symétriques. Par conséquent la forme de Hessenberg supérieure devient tridiagonale. La méthode de Givens est donc une méthode de réduction d'une matrice symétrique à une matrice tridiagonale semblable.

8.3.3 La méthode de Householder

On pose
$$A_{k-1} = \left(\begin{array}{cc|c} & & C_{k-1} \\ & H_{k-1} & \\ 0 & b_{k-1} & B_{k-1} \end{array}\right).$$

La matrice H_{k-1} est de la forme de Hessenberg supérieure. On exprime la matrice P_k sous la forme
$$P_k = \left(\begin{array}{c|c} I & 0 \\ \hline 0 & Q_k \end{array}\right)$$
avec $Q_k = I - 2v_k v_k^T$ où $v_k \in \mathbb{R}^{n-k}$ et $(v_k, v_k) = 1$. La matrice identité en haut et à gauche est de dimension $k \times k$ et la matrice Q_k est de dimension

$(n-k) \times (n-k)$. Les dimensions des autres blocs s'en déduisent facilement. Les matrices P_k et Q_k sont symétriques.

On obtient donc

$$A_k = P_k A_{k-1} P_k = \left(\begin{array}{c|c} H_{k-1} & C_{k-1}Q_k \\ \hline 0 \quad Q_k b_{k-1} & Q_k B_{k-1} Q_k \end{array} \right).$$

Les dimensions des blocs sont les mêmes que celles des blocs correspondants de la matrice P_k.

Si l'on choisit Q_k de sorte que toutes les composantes du vecteur $Q_k b_{k-1}$ soient nulles sauf la première (et l'on sait que c'est possible grâce au résultat du Théorème 21), alors la sous matrice principale de dimension $k+1$ est de la forme de Hessenberg supérieure. On peut simplifier les formules précédentes en introduisant le vecteur $w_k \in \mathbb{R}^n$ dont les k premières composantes sont nulles. On a alors

$$P_k = I - 2w_k w_k^T = I - u_k u_k^T / 2\nu_k^2$$

avec

$$\begin{aligned} (u_k)_i &= 0, \quad i = 1, \ldots, k \quad (i\text{ème composante de } u_k) \\ (u_k)_{k+1} &= a_{k+1,k}^{(k-1)} \mp S_k \\ (u_k)_i &= a_{ik}^{(k-1)}, \quad i = k+2, \ldots, n \\ S_k &= \left[\sum_{i=k+1}^{n} (a_{ik}^{(k-1)})^2 \right]^{1/2} \\ 2\nu_k^2 &= S_k^2 \mp a_{k+1,k}^{(k-1)} S_k \end{aligned}$$

et l'on a $a_{k+1,k}^{(k)} = \pm S_k$. Le signe $\pm$ devant S_k est celui de $a_{k+1,k}^{(k-1)}$ de sorte que $(u_k)_{k+1}$ ne puisse pas être nul.

Le passage de A_0 à A_{n-2} nécessite de l'ordre de $5n^3/3$ multiplications, soit environ deux fois moins que dans la méthode de Givens.

Remarque 12

1. *Si la matrice A est symétrique alors toutes les matrices A_k sont symétriques et par conséquent A_{n-2} est une matrice tridiagonale.*
2. *Les méthodes de Givens et Householder ne sont pas totalement indépendantes. On montre que les matrices des deux transformations sont identiques à des facteurs multiplicatifs ± 1 près pour chaque colonne.*

Nous avons vu que, lorsque la matrice A est symétrique, alors on peut utiliser les méthodes de Givens et de Householder pour réduire A à la forme tridiagonale. En effectuant ces réductions en tenant compte des symétries des matrices

il faut de l'ordre de $4n^2/3$ multiplications pour la méthode de Givens et $2n^3/3$ pour celle de Householder. En pratique on utilisera donc toujours la méthode de Householder.

Chapitre 9

Quelques méthodes itératives

Soit à résoudre le système d'équations linéaires

$$Ax = b.$$

Une méthode itérative consiste à construire une suite de vecteurs qui, sous certaines conditions, converge vers la solution du système.

Dans ce Chapitre, après un certain nombre de résultats généraux, on étudiera les méthodes de relaxation puis la méthode des directions alternées. Les Chapitres suivants seront consacrés aux méthodes de projection qui, bien qu'étant des méthodes itératives, sont de nature complètement différente.

Pour plus de résultats, on pourra consulter l'ouvrage fondamental de Varga [**?**], ou les livres de Ciarlet [**?**], de Meurant [**?**] et de Stewart [**?**].

9.1 Généralités

Commençons par des définitions et des résultats généraux.

Définition 7
Une matrice A est dite non négative si

$$a_{ij} \geq 0, \quad \forall i, j.$$

On écrira $A \geq 0$, 0 étant la matrice nulle. On définirait de même une matrice positive par l'inégalité stricte.

Définition 8
On dit que la matrice A est monotone si et seulement si elle est régulière et $A^{-1} \geq 0$.

Cette définition est la version en dimension finie du principe du maximum. Par conséquent, si $b \geq 0$ alors $x = A^{-1}b \geq 0$.

Définition 9
A est réductible s'il existe une matrice de permutation P telle que

$$PAP^T = \begin{pmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{pmatrix}$$

les sous–matrices A_{11} et A_{22} étant carrées. Une matrice qui ne vérifie pas cette propriété est appelée irréductible.

Une définition fondamentale est

Définition 10
Décomposons la matrice A en deux matrices M et N de façon que

$$A = M - N.$$

On dit que cette décomposition est régulière si M est régulière, si $M^{-1} \geq 0$ et si $N \geq 0$.
On dit que la décomposition est faiblement régulière si M est régulière, si $M^{-1} \geq 0$ et si $M^{-1}N \geq 0$.

Donnons maintenant le Théorème de Perron–Frobenius. On rappelle que $\rho(A)$ désigne le rayon spectral de A, c'est–à–dire que $\rho(A) = \max_i |\lambda_i|$, où les λ_i sont les valeurs propres de A.

Théorème 25
Si $A \geq 0$ et est irréductible, alors

i) A a une valeur propre positive ou nulle égale à $\rho(A)$,

ii) Le vecteur propre correspondant à $\rho(A)$ est positif,

iii) $\rho(A)$ croît quand un élément quelconque de A croît,

iv) $\rho(A)$ est une valeur propre simple.

On a également le

Théorème 26
Si $a_{ij} \leq 0, \forall i \neq j$, alors les deux propositions suivantes sont équivalentes

i) A est régulière et $A^{-1} \geq 0$,

ii) $\forall i, a_{ii} > 0$ *et* $B = I - D^{-1}A \geq 0$ *est irréductible et* $\rho(B) < 1$, D *étant la matrice diagonale des éléments diagonaux de* A.

Sur les matrices non négatives et le Théorème de Perron–Frobenius, voir [**?**].

Donnons des définitions qui seront utilisées dans la suite

Définition 11
Une matrice régulière A *telle que* $a_{ij} \leq 0$, $\forall i \neq j$, *et* $A^{-1} \geq 0$ *s'appelle une* M*–matrice.*
Soit $M(A)$ *la matrice dont les éléments* m_{ij} *sont donnés par* $m_{ii} = |a_{ii}|$ *et* $m_{ij} = -|a_{ij}|$ *pour* $i \neq j$. *On dit que* A *est une* H*–matrice si et seulement si* $M(A)$ *est une* M*–matrice.*
Une matrice A *symétrique définie positive et telle que* $a_{ij} \leq 0$, $\forall i \neq j$, *s'appelle une matrice de Stieltjes.*

On voit qu'une M–matrice est une H–matrice.

Une conséquence du Théorème précédent est le

Corollaire 3
Si A *est une matrice de Stieltjes, alors c'est une* M*–matrice. De plus,* A *est irréductible si et seulement si* $A^{-1} > 0$.

Un résultat intéressant sur la construction des M–matrices est fourni par le

Théorème 27
Soit A *une* M*–matrice et soit* C *une matrice obtenue en remplaçant certains éléments hors diagonaux de* A *par 0. Alors* C *est une* M*–matrice.*

On démontre aussi le résultat suivant (voir la Définition 14)

Théorème 28
A *est une* H*–matrice non singulière si et seulement si* A *est à dominance diagonale stricte généralisée.*

Étudions maintenant ce que l'on appelle matrices cycliques et primitives.

Soit $A \geq 0$ et irréductible et soit k le nombre de valeurs propres de A de module égal à $\rho(A)$.

Définition 12
Si $k = 1$ *on dit que* A *est primitive et si* $k > 1$ *on dit qu'elle est cyclique d'ordre* k.

Dans le cas d'une matrice cyclique d'ordre k, les k valeurs propres de module égal à $\rho(A)$ sont

$$\lambda_j = \rho(A)e^{i\varphi_j}, \quad j = 0, \ldots, k-1$$

avec $\varphi_j = 2j\pi/k$.

Définition 13
Une matrice A (non nécessairement non négative ou irréductible) est faiblement cyclique d'ordre $k > 1$ s'il existe une matrice de permutation P telle que PAP^T soit de la forme

$$PAP^T = \begin{pmatrix} 0 & \cdots & \cdots & \cdots & A_{1k} \\ A_{21} & 0 & & & \vdots \\ 0 & A_{32} & \ddots & & \vdots \\ & & \ddots & \ddots & \vdots \\ & & & A_{k,k-1} & 0 \end{pmatrix}$$

où les sous–matrices nulles sont carrées.

On a les propriétés suivantes

Théorème 29

1. *Si $A > 0$, alors A est primitive.*
2. *Si A est primitive, alors A^m est primitive $\forall m > 0$.*
3. *$A \geq 0$ et $\forall m > 0, A^m > 0$ si et seulement si A est primitive.*

Une notion importante est donnée dans la

Définition 14
On dit que A est à dominance diagonale stricte si

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

S'il existe une matrice diagonale $D > 0$ telle que $D^{-1}AD$ soit à dominance diagonale (stricte), on dit que A est à dominance diagonale (stricte) généralisée.

9.2 Convergence de matrices

Soit $(A^{(k)})$ une suite de matrices. On dit que $(A^{(k)})$ converge vers A si $\lim_{k\to\infty} \|A^{(k)} - A\| = 0$.

Cherchons la condition nécessaire et suffisante sur A pour que la suite (A^m) converge vers 0 lorsque m tend vers l'infini. Dans ce cas, on dira que la matrice A est convergente.

Il existe une matrice S régulière telle que $SAS^{-1} = \tilde{A}$ où $\tilde{A}$ est de la forme de Jordan

$$\tilde{A} = \begin{pmatrix} J_1 & & & \\ & J_2 & & \\ & & \ddots & \\ & & & J_r \end{pmatrix},$$

J_k étant une matrice $n_k \times n_k$ de la forme

$$J_k = \begin{pmatrix} \lambda_k & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_k \end{pmatrix},$$

et λ_k étant une valeur propre de A de multiplicité n_k avec $n_1 + \cdots + n_r = n$.

On a $\tilde{A}^m = SA^mS^{-1}$, c'est–à–dire $A^m = S^{-1}\tilde{A}^mS$. Il est donc équivalent de chercher la condition nécessaire et suffisante pour que $\tilde{A}^m$ tende vers 0 quand m tend vers l'infini. On a

$$\tilde{A}^m = \begin{pmatrix} J_1^m & & & \\ & J_2^m & & \\ & & \ddots & \\ & & & J_r^m \end{pmatrix}.$$

Donc A^m tend vers 0 si et seulement si J_k^m tend vers 0 quand m tend vers l'infini. Calculons J_k^m. On a

$$J_k^2 = \begin{pmatrix} \lambda_k^2 & 2\lambda_k & 1 & & & \\ & \lambda_k^2 & 2\lambda_k & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & \ddots & 1 \\ & & & & \ddots & 2\lambda_k \\ & & & & & \lambda_k^2 \end{pmatrix}$$

et par récurrence, en notant par $d_{i,j;k}^m$ les éléments de J_k^m,

$$d_{i,j;k}^m = \begin{cases} 0 & \text{si } j < i \\ (j^m - 1)\lambda_k^{m-j+i} & \text{si } i \leq j < \max(n_k, m+i) \\ 0 & m+i \leq j < n_k. \end{cases}$$

Par conséquent J_k^m tend vers 0 si $|\lambda_k| < 1, \forall k$. On a donc démontré le

Théorème 30
Une condition nécessaire et suffisante pour que $\lim_{m\to\infty} A^m = 0$ *est que* $\rho(A) < 1$.

Nous avons

$$\|J^m\| \sim \begin{pmatrix} m \\ p-1 \end{pmatrix} [\rho(J)]^{m-(p-1)} \quad (m \to \infty)$$

p étant la dimension de la matrice J.

Une conséquence du Théorème précédent est le

Théorème 31
La série $I + A + A^2 + \cdots$ *converge vers* $(I - A)^{-1}$ *si et seulement si* $\rho(A) < 1$.

Démonstration.
Supposons que $\rho(A) < 1$. Soit λ une valeur propre de A. Alors $1 - \lambda$ est valeur propre de $I - A$. Puisque $\rho(A) < 1$, on a $|\lambda| < 1$ et donc $1 - \lambda \neq 0$. Par conséquent, la matrice $I - A$ est régulière.

Posons $S_k = I + A + \cdots + A^k$. Alors $(I - A)S_k = I - A^{k+1}$, c'est–à–dire $S_k = (I - A)^{-1}(I - A^{k+1})$. On a donc $S_k - (I - A)^{-1} = -(I - A)^{-1}A^{k+1}$ ce qui, en passant aux normes, conduit à

$$\|S_k - (I - A)^{-1}\| \leq \|(I - A)^{-1}\| \cdot \|A^{k+1}\|.$$

D'après le Théorème précédent, la suite (A^k) converge vers zéro ce qui démontre que la suite (S_k) converge vers $(I - A)^{-1}$.

Réciproquement, si la suite (S_k) admet une limite, alors la suite (A^k) converge vers zéro. Donc, d'après le Théorème précédent, $\rho(A) < 1$. ∎

9.3 Méthodes de relaxation

Une méthode de relaxation pour résoudre le système de n équations $Ax = b$ consiste à effectuer une décomposition de la matrice A sous la forme

$$A = M - N$$

où M est une matrice régulière. Le système peut alors s'écrire

$$Mx = Nx + b.$$

Construisons la suite de vecteurs x_0, x_1, ... de la façon suivante

– x_0 arbitraire,

– $Mx_{k+1} = Nx_k + b$, soit encore $x_{k+1} = M^{-1}Nx_k + M^{-1}b$.

Nous avons ainsi défini une méthode itérative dite *de relaxation*.

Il est possible de mettre les itérations précédentes sous une autre forme. On a $M^{-1}N = I - M^{-1}A$, d'où

$$x_{k+1} = x_k + M^{-1}(b - Ax_k) = x_k + M^{-1}r_k$$

avec $r_k = b - Ax_k$. La matrice M^{-1} apparaît donc comme devant être une approximation de A^{-1}, c'est–à–dire comme un préconditionneur. En d'autres termes, M doit être une approximation de A mais, bien sûr, le système $M(x_{k+1} - x_k) = r_k$ doit être beaucoup plus facile à résoudre que le système $Ay_k = r_k$ dont la solution est $y_k = x - x_k$. En effet, sauf dans le cas où le calcul de M^{-1} est très simple, x_{k+1} se calcule en résolvant le système linéaire $Mx_{k+1} = Nx_k + b$.

On a

$$M^{-1}N = (A + N)^{-1}N = (I + G)^{-1}G \quad \text{avec} \quad G = A^{-1}N.$$

Soit u un vecteur propre de G correspondant à la valeur propre τ. Alors $1+\tau \neq 0$ et l'on a

$$Gu = \tau u \quad \text{et} \quad (I + G)^{-1}Gu = \frac{\tau}{1 + \tau}u.$$

Par conséquent u est aussi vecteur propre de $M^{-1}N$ et il correspond à la valeur propre $\mu = \tau/(1 + \tau)$.
Réciproquement, si μ est valeur propre de $(I + G)^{-1}G$ et si v est le vecteur propre correspondant, alors

$$Gv = \mu(I + G)v.$$

D'après cette relation, μ ne peut pas être égal à 1 et donc

$$Gv = \frac{\mu}{1 - \mu}v = \tau v.$$

Puisque $\mu = \tau/(1 + \tau)$ est une fonction strictement croissante de τ, il découle

$$\rho(M^{-1}N) = \frac{\rho(A^{-1}N)}{1 + \rho(A^{-1}N)}. \tag{9.1}$$

Cette relation sera utile par la suite.

Il est possible de considérer, de façon plus générale, une méthode itérative de la forme

$$x_{k+1} = B_k x_k + c_k.$$

Alors, afin d'avoir $x_{k+1} = x$ dans le cas où $x_k = x$, on doit avoir $c_k = H_k b$ et $I - B_k = H_k A$ où H_k est une matrice inversible. On obtient

$$\begin{aligned} x_{k+1} &= x_k + (B_k - I)x_k + c_k \\ &= x_k - H_k Ax_k + H_k b \\ &= x_k + H_k r_k. \end{aligned}$$

H_k est donc également un préconditionneur.

Remarque 13
À l'heure actuelle, les méthodes de relaxation ne sont plus tellement utilisées comme méthodes itératives car il existe des méthodes beaucoup plus efficaces. Cependant, les matrices M^{-1} sont utilisées comme précondionneurs.

9.3.1 Généralités sur la convergence

Lorsque k tend vers l'infini nous voulons que x_k, obtenu par une méthode de relaxation, converge vers x, solution unique du système. De ce qui précède, nous déduisons

$$M(x_{k+1} - x) = N(x_k - x).$$

Posons $e_k = x - x_k$. On a

$$e_k = M^{-1}Ne_{k-1} = (M^{-1}N)^k e_0.$$

Par conséquent

$$\|e_k\| \leq \|M^{-1}N\|^k \cdot \|e_0\|$$

d'où immédiatement le

Théorème 32
Si $\|M^{-1}N\| < 1$, alors (x_k) converge vers x quel que soit x_0.

La condition suffisante du ce Théorème est trop forte. En effet, nous avons vu, dans [**?**], que $\rho(M^{-1}N) \leq \|M^{-1}N\|$ et le résultat précédent ne permet pas de conclure si 1 est compris entre ces deux quantités.

Le résultat fondamental est le suivant

Théorème 33
Une condition nécessaire et suffisante pour que la suite (x_k) converge vers x quel que soit x_0 est que $\rho(M^{-1}N) < 1$.

Démonstration.

Puisque $e_k = (M^{-1}N)^k e_0$, une condition nécessaire et suffisante pour que la suite (e_k) tende vers 0 est que la matrice $(M^{-1}N)^k$ tende vers la matrice nulle quand k tends vers l'infini. D'après ce qui a été vu dans [**?**], il faut et il suffit que toutes les valeurs propres de $M^{-1}N$ soient de module strictement plus petit que 1. ∎

Remarque 14
D'après le Théorème précédent, on voit que plus $\rho(M^{-1}N) < 1$ est petit et plus la convergence de la méthode est rapide.

Théorème 34
Si A n'est pas singulière, une condition nécessaire et suffisante pour que la suite (x_k) tende vers x lorsque k tend vers l'infini est que la suite $(r_k = b - Ax_k)$ tende vers 0.

Démonstration.

Nous avons

$$\begin{aligned} r_k &= Ae_k \\ e_k &= A^{-1}r_k. \end{aligned}$$

D'après les inégalités sur les normes, on a

$$\begin{aligned} \|r_k\| &\leq \|A\| \cdot \|e_k\| \\ \|e_k\| &\leq \|A^{-1}\| \cdot \|r_k\|. \end{aligned}$$

On voit que la première inégalité implique que si x_k tend vers x , $\|e_k\|$ tend vers 0. Donc $\|r_k\|$ tend vers 0, c'est–à–dire que r_k tend vers 0.

La seconde inégalité implique que si r_k tend vers 0, $\|r_k\|$ tend vers 0. Donc $\|e_k\|$ tend vers 0, c'est–à– dire que (x_k) converge vers x. ∎

L'inégalité

$$\|e_k\| \leq \|A^{-1}\| \cdot \|r_k\|$$

indique une majoration absolue de la norme de l'erreur. Cependant on préfère à cela une majoration basée sur l'affaiblissement relatif des résidus $\|r_k\|/\|x_k\|$ et sur l'erreur relative $\|e_k\|/\|x_k\|$.

Théorème 35
Pour un affaiblissement relatif donné $\eta_k = \|r_k\|/\|x_k\|$, le rapport des erreurs relatives extrêmales est égal à l'inverse du conditionnement $\kappa(A)$ de la matrice A.

Démonstration.

$$\frac{\|r_k\|}{\|x_k\|} = \frac{\|Ae_k\|}{\|e_k\|} \cdot \frac{\|e_k\|}{\|x_k\|}$$

d'où

$$\frac{\|e_k\|}{\|x_k\|} = \eta_k \Big/ \frac{\|Ae_k\|}{\|e_k\|}.$$

L'erreur relative minimale est égale à $\eta_k/\|A\|$, tandis que l'erreur relative maximale est égale à $\eta_k \cdot \|A^{-1}\|$. Puisque

$$\frac{1}{\|A^{-1}\|} \leq \frac{\|Ax\|}{\|x\|} \leq \|A\|$$

on a

$$\frac{\text{erreur relative minimale}}{\text{erreur relative maximale}} = \frac{1}{\|A^{-1}\| \cdot \|A\|} = \frac{1}{\kappa(A)}. \quad ∎$$

Remarque 15
On a vu que $e_k = M^{-1}Ne_{k-1}$ *et que* $r_k = Ae_k$. *Donc*

$$\begin{aligned} r_k &= (M-N)M^{-1}Ne_{k-1} = N(I-M^{-1}N)e_{k-1} \\ &= NM^{-1}(M-N)e_{k-1} \\ &= NM^{-1}r_{k-1}. \end{aligned}$$

On rappelle que les matrices $M^{-1}N$ *et* NM^{-1} *sont semblables et que si* u *est vecteur propre de la première matrice, alors* $v = Mu$ *est le vecteur propre de la seconde associé à la même valeur propre.*

Nous avons les résultats de convergence suivants, dans des cas particuliers.

Théorème 36
Soit $A = M - N$ *une décomposition régulière de* A. *Alors* A *est régulière avec* $A^{-1} \geq 0$ *si et seulement si*

$$\rho(M^{-1}N) = \frac{\rho(A^{-1}N)}{1+\rho(A^{-1}N)} < 1.$$

Donc, si A *est régulière avec* $A^{-1} \geq 0$ *alors* (x_k) *converge vers* x *quel que soit* x_0.

Si une matrice est symétrique, son rayon spectral est égal à sa norme. Dans ce cas, $\rho(A^{-1}N) \leq \|A^{-1}N\| \leq \|A^{-1}\| \cdot \|N\| = \rho(A^{-1})\rho(N)$. Donc, d'après le Théorème précédent, on a

Théorème 37
Si A *est une matrice de Stieltjes et si* $A = M - N$ *est une décomposition régulière de* A *avec* N *symétrique, alors*

$$\rho(M^{-1}N) \leq \frac{\rho(N)\rho(A^{-1})}{1+\rho(N)\rho(A^{-1})} < 1.$$

Les démonstrations de ces résultats sont basées sur la relation (9.1).

Théorème 38
Si A *est une* M*–matrice et si* M *est obtenue en remplaçant certains éléments hors diagonaux de* A *par 0, alors* $A = M - N$ *est une décomposition régulière de* A *et* $\rho(M^{-1}N) < 1$.

La démonstration est basée sur le Théorème 27, d'après lequel M est également une M–matrice.

Théorème 39
Soit $A = M - N$ une décomposition faiblement régulière de A. Alors, A est régulière et $A^{-1} \geq 0$ si et seulement si $\rho(M^{-1}N) < 1$.

Certains résultats permettent de comparer entre elles différentes décompositions de la matrice A.

Théorème 40
Soient $A = M_1 - N_1 = M_2 - N_2$ deux décompositions régulières de A. Si $A^{-1} \geq 0$ et si $N_2 \geq N_1 \geq 0$, alors

$$0 \leq \rho(M_1^{-1}N_1) \leq \rho(M_2^{-1}N_2) < 1.$$

De plus, si $A^{-1} > 0$, si $N_2 \geq N_1 \geq 0$ et si $N_2 \neq N_1 \neq 0$, alors les inégalités précédentes sont strictes.

Théorème 41
Soient $A = M_1 - N_1 = M_2 - N_2$ deux décompositions régulières de A. Si $A^{-1} \geq 0$ et si $M_1^{-1} \geq M_2^{-1}$, alors

$$0 \leq \rho(M_1^{-1}N_1) \leq \rho(M_2^{-1}N_2) < 1.$$

Si $A^{-1} > 0$ et si $M_1^{-1} > M_2^{-1}$, alors les inégalités précédentes sont strictes.

D'après ce qui précède, on voit que le choix de la décomposition de la matrice A du système est guidé par les considérations suivantes

1. il faut que $\rho(M^{-1}N) < 1$,
2. la résolution de $My = c$, où c est un vecteur quelconque, doit être simple et exiger le moins d'opérations possibles,
3. la décomposition retenue doit être la meilleure possible, c'est–à–dire que $\rho(M^{-1}N)$ doit être le plus petit possible.

9.3.2 Les méthodes de Jacobi et Gauss–Seidel

Nous allons maintenant étudier un certain nombre de décompositions de la matrice A.

Soit D la matrice diagonale des éléments diagonaux de A

$$D = \begin{pmatrix} a_{11} & & \\ & \ddots & \\ & & a_{nn} \end{pmatrix},$$

soit $-E$ la matrice triangulaire strictement inférieure de A

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

et soit $-F$ la matrice triangulaire strictement supérieure de A

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

On a donc

$$A = \begin{pmatrix} & & -F \\ & D & \\ -E & & \end{pmatrix},$$

c'est–à–dire $A = D - E - F$.

On supposera que la matrice D n'est pas singulière, c'est–à–dire que $\forall i, a_{ii} \neq 0$.

Afin de construire une méthode de relaxation, nous allons maintenant regrouper deux de ces matrices afin de mettre A sous la forme $A = M - N$.

On peut également définir des méthodes de relaxation par blocs dans lesquelles, maintenant, les a_{ij} sont des sous–matrices de la matrice A.

Méthode de Jacobi

Cette méthode est due à Jacobi (1845). Elle consiste à prendre $M = D$ et $N = E + F$. D'où

$$M^{-1}N = D^{-1}(E + F)$$

ce qui nous donne l'itération suivante

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

qui correspond, pour les composantes $x_{k,i}$ des vecteurs, à

$$x_{k+1,i} = \frac{1}{a_{ii}} \left(b_i - \sum_{\substack{j=1 \\ j \neq i}}^{n} a_{ij} x_{k,j} \right), \quad i = 1, \ldots, n.$$

Or nous avons

$$\begin{aligned} r_{k,i} &= b_i - \sum_{j=1}^{n} a_{ij} x_{k,j} \\ &= b_i - \sum_{\substack{j=1 \\ j \neq i}}^{n} a_{ij} x_{k,j} - a_{ii} x_{k,i} \end{aligned}$$

d'où

$$\begin{aligned} x_{k+1,i} &= x_{k,i} + \frac{r_{k,i}}{a_{ii}} \\ r_{k+1,i} &= -\sum_{\substack{j=1 \\ j \neq i}}^{n} \frac{a_{ij}}{a_{jj}} r_{k,j}. \end{aligned}$$

La méthode de Jacobi est vraiment la plus simple à laquelle on puisse penser. Cependant, sa convergence peut être très lente. Si on l'applique, par exemple, à une matrice tridiagonale de dimension 100 avec des 2 sur la diagonale, des -1 sur les deux autres diagonales et si le second membre est nul, alors, en partant d'un vecteur x_0 aléatoire entre 0 et 1, il faut de l'ordre de 28000 itérations pour obtenir une erreur de 10^{-6}.

On a $M^{-1}N = I - D^{-1}A$. Il est possible de définir une méthode de Jacobi par blocs en partitionnant la matrice A en blocs. On a alors $D = \text{diag}(A_{11}, \ldots, A_{mm})$ où les blocs diagonaux A_{ii} sont carrés et réguliers.

Méthode de Gauss–Seidel

Cette méthode est due à Gauss (1826) et à Seidel (1874). Elle consiste à prendre $M = D - E$ et $N = F$. D'où

$$B = M^{-1}N = (D - E)^{-1}F$$

ce qui nous donne l'itération suivante

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

qui correspond, pour les composantes, à

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

On voit que la mise en œuvre de la méthode de Jacobi demande de conserver en mémoire deux vecteurs alors que la méthode de Gauss–Seidel n'en demande qu'un seul.

Il est possible de définir une méthode de Gauss–Seidel par blocs.

9.3.3 Convergence des méthodes de Jacobi et de Gauss–Seidel

La matrice $B = M^{-1}N = D^{-1}(E + F)$ de la méthode de Jacobi est une matrice à diagonale nulle. Nous la décomposons en une somme $B = L+U$ où la matrice L est strictement triangulaire inférieure (c'est–à–dire que sa diagonale principale ne contient que des termes nuls) et où U est strictement triangulaire supérieure. On a $L = D^{-1}E$ et $U = D^{-1}F$. Soit $\mathcal{L}_1$ la matrice $\mathcal{L}_1 = (I-L)^{-1}U$. Cette matrice est la matrice $M^{-1}N = (D - E)^{-1}F$ de la méthode de Gauss–Seidel.

Le résultat suivant est connu sous le nom de théorème de Stein–Rosenberg (1948)

Théorème 42
Supposons que $\forall i \neq j, b_{ij} > 0$. Alors une et seulement une des relations suivantes (qui s'excluent donc mutuellement) est vérifiée

1. $\rho(B) = \rho(\mathcal{L}_1) = 0$,
2. $0 < \rho(\mathcal{L}_1) < \rho(B) < 1$,
3. $\rho(B) = \rho(\mathcal{L}_1) = 1$,
4. $1 < \rho(B) < \rho(\mathcal{L}_1)$.

Démonstration.

Posons $m(\sigma) = \rho(\sigma L + U)$ et $n(\sigma) = \rho(L + \sigma U)$ avec $\sigma \geq 0$. On voit que

$$\begin{aligned} m(0) &= n(0) = 0 \\ m(1) &= n(1) = \rho(L + U) = \rho(B) \\ n(\sigma) &= \sigma m(1/\sigma), \quad \sigma > 0. \end{aligned}$$

Le Théorème de Perron–Frobenius nous montre que $m(\sigma)$ et $n(\sigma)$ sont des fonctions strictement croissantes de σ et que si $\rho(B) = 0$ alors $m(\sigma) = n(\sigma) = 0$.

Supposons $B \geq 0$ et irréductible. Puisque L est strictement triangulaire inférieure, on a $\forall m \geq n, L^m = 0$ et il s'en suit

$$(I - L)^{-1} = I + L + L^2 + \ldots + L^{n-1}.$$

Par conséquent, $\mathcal{L}_1 \geq 0$. Il existe donc un vecteur $x \geq 0$ tel que

$$(I - L)^{-1}Ux = \lambda x \quad \text{avec } \lambda = \rho(\mathcal{L}_1) \geq 0$$

d'où

$$(\lambda L + U)x = \lambda x \quad \text{et} \quad \left(L + \frac{1}{\lambda}U\right)x = x.$$

On en déduit, puisque ces matrices sont non négatives et irréductibles, que $m(\lambda) = \lambda$ et $n(1/\lambda) = 1$.

Si $\rho(B) = 1$ alors $n(1) = 1$ et donc $\lambda = 1$. Si $0 < \rho(B) < 1$ alors $n(1) = \rho(B) < 1$ et $n(1/\lambda) = 1$ ce qui entraîne $1/\lambda > 1$ ou $0 < \lambda < 1$ puisque la fonction n est croissante. Or puisque m est également croissante et que $m(1) = \rho(B)$, alors $0 < \lambda < \rho(B) < 1$. Finalement si $\rho(B) > 1$ alors $\lambda = \rho(\mathcal{L}_1) > \rho(B) > 1$. ∎

Ainsi les matrices B et $\mathcal{L}_1$ sont simultanément convergentes ou non. Dans le cas de la convergence, la méthode de Gauss–Seidel converge plus vite que celle de Jacobi. Lorsqu'il y a divergence, la méthode de Gauss–Seidel diverge plus vite que celle de Jacobi.

Corollaire 4
Si la matrice non négative B est telle que $0 < \rho(B) < 1$, alors $0 < \rho(\mathcal{L}_1) < \rho(B) < 1$.

On a le

Théorème 43
Si la matrice A est non singulière et à dominance diagonale stricte, alors les méthodes de Jacobi et de Gauss–Seidel sont convergentes.

Démonstration.

La matrice B de la méthode de Jacobi est telle que $b_{ij} = -a_{ij}/a_{ii}$ si $i \neq j$ et $b_{ii} = 0$ d'où, $\forall i$

$$\sum_{j=1}^{n} |b_{ij}| < 1.$$

Soit $|B|$ la matrice dont les termes sont $|b_{ij}|$. Selon un théorème d'Hadamard et ses corollaires (que nous ne donnerons pas ici), on a

$$\rho(|B|) < 1.$$

Une conséquence du Théorème de Perron–Frobenius est

$$\rho(B) < \rho(|B|) < 1.$$

Donc la méthode de Jacobi est convergente. Posons $B = L + U$ avec L matrice triangulaire strictement inférieure, U matrice triangulaire supérieure et $\mathcal{L}_1 = (I - L)^{-1}U$. Alors

$$\rho(\mathcal{L}_1) \leq \rho((I - |L|)^{-1}|U|).$$

D'après le Théorème de Stein–Rosenberg nous avons

$$\rho((I - |L|)^{-1}|U|) \leq \rho(|B|).$$

Or $\rho(|B|) < 1$, ce qui montre que la méthode de Gauss–Seidel est convergente. ∎

On a également les résultats de convergence suivants

Théorème 44
Si A est une M–matrice ou une H–matrice, alors les méthodes de Jacobi et de Gauss–Seidel sont convergentes.

Démonstration.

Notons $B(A) = M^{-1}N$ la matrice d'itération correspondant à la matrice A pour la méthode de Jacobi. Si A est une H–matrice, il existe, d'après le Théorème 28, une matrice diagonale T à éléments diagonaux strictement positifs telle que $T^{-1}AT$ soit à dominance diagonale stricte. Par conséquent

$$B(A) = M^{-1}N = -D^{-1}(L + U) = -D^{-1}(A - D) = I - D^{-1}A.$$

Les diagonales de A et $T^{-1}AT$ sont identiques et donc, puisque $T^{-1}D^{-1} = D^{-1}T^{-1}$,

$$B(T^{-1}AT) = I - D^{-1}(T^{-1}AT) = T^{-1}(I - D^{-1}A)T = T^{-1}B(A)T.$$

Donc $B(A)$ et $B(T^{-1}AT)$ sont semblables et elles ont donc les mêmes valeurs propres. Puisque $T^{-1}AT$ est à dominance diagonale stricte, on a $\rho(B(A)) = \rho(B(T^{-1}AT)) < 1$.

Puisqu'une M–matrice est une H–matrice, le résultat est également vrai pour les M–matrices.

La démonstration est similaire pour la méthode de Gauss–Seidel. ∎

9.4 Méthodes de Richardson

Sous ce nom, on regroupe plusieurs méthodes de la forme

$$x_{k+1} = x_k + \lambda_k r_k,$$

avec x_0 donné. Le vecteur résidu r_k peut être calculé soit directement par la formule $r_k = b - Ax_k$ soit de façon itérative par

$$r_{k+1} = r_k - \lambda_k A r_k.$$

Ces méthodes se distinguent par le choix du paramètre λ_k. Le choix le plus simple consiste à prendre $\lambda_k = \lambda$. On parle alors de méthode de Richardson *stationnaire*. Puisque

$$\frac{1}{\lambda} x_{k+1} = \left(\frac{1}{\lambda} I - A\right) x_k + b,$$

il correspond à la décomposition $M = I/\lambda$ et $N = I/\lambda - A$. Cette méthode converge donc si et seulement si $\rho(I - \lambda A) < 1$. D'où le

Théorème 45
Si A est symétrique définie positive, alors la méthode de Richardson stationnaire converge si et seulement si $\lambda < 2/\rho(A)$.

Il existe, dans ce cas, une valeur optimale de λ. Elle est donnée par le

Théorème 46
Si A est symétrique définie positive, alors la valeur optimale de λ pour la méthode de Richardson stationnaire est $\lambda_{\text{opt}} = 2/(\mu_1 + \mu_n)$, où μ_1 et $\mu_n = \rho(A)$ sont respectivement la plus petite et la plus grande des valeurs propres de A.

Démonstration.

La valeur optimale de λ est déterminée par la condition $1 - \lambda\mu_1 = -(1 - \lambda\mu_n)$, ce qui donne $\lambda_{\text{opt}} = 2/(\mu_1 + \mu_n)$. De plus

$$\rho(I - \lambda_{\text{opt}} A) = 1 - \frac{2}{\mu_1 + \mu_n}\mu_1 = \frac{\mu_n - \mu_1}{\mu_n + \mu_1} = \frac{\kappa(A) - 1}{\kappa(A) + 1}$$

avec $\kappa(A) = \|A\|_2 \cdot \|A^{-1}\|_2 = \mu_n/\mu_1$. ∎

Ce Théorème montre que plus $\kappa(A)$ est voisin de 1 plus la convergence est rapide.

Naturellement, un tel choix est impossible en pratique puisque les valeurs propres de A ne sont pas connues. Pour cette raison, revenons à un paramètre λ_k variable et prenons le tel que $\|r_{k+1}\|_2$ soit minimum. On a

$$(r_{k+1}, r_{k+1}) = (r_k, r_k) - 2\lambda_k (r_k, Ar_k) + \lambda_k^2 (Ar_k, Ar_k)$$

qui est minimum pour

$$\lambda_k = (r_k, Ar_k)/(Ar_k, Ar_k).$$

On a, pour ce choix,

$$(r_{k+1}, r_{k+1}) = (r_k, r_k) - \frac{(r_k, Ar_k)^2}{(Ar_k, Ar_k)} \leq (r_k, r_k).$$

Cette inégalité est stricte si et seulement si les vecteurs r_k et Ar_k ne sont pas orthogonaux. La suite $(\|r_k\|)$ est donc décroissante bornée inférieurement. Elle est donc convergente, mais rien que garantit que sa limite soit nulle.

Lorsque la matrice A est symétrique définie positive, on peut choisir λ_k qui minimise $(r_{k+1}, A^{-1}r_{k+1})$. Cette quantité est bien une norme et l'on a

$$\begin{aligned}(r_{k+1}, A^{-1}r_{k+1}) &= (r_k, A^{-1}r_k) - \lambda_k(r_k, r_k) - \lambda_k(Ar_k, A^{-1}r_k) + \lambda_k^2(r_k, Ar_k) \\ &= (r_k, A^{-1}r_k) - 2\lambda_k(r_k, r_k) + \lambda_k^2(r_k, Ar_k).\end{aligned}$$

La valeur optimale de λ_k est donc

$$\lambda_k = (r_k, r_k)/(Ar_k, r_k).$$

Cette méthode s'appelle la *méthode de la plus profonde descente*. Elle est convergente. Elle sera étudiée dans le Chapitre **??**.

De plus amples détails sur les méthodes de Richardson sont donnés dans [**?**].

Bibliographie

Dans presque tous les ouvrages généraux d'analyse numérique on trouve un chapitre ou plusieurs sur les méthodes numériques en algèbre matricielle. Il ne nous est pas possible de les citer tous, ce qui ne signifie, en aucun cas, que leur niveau ou leur valeur est moindre que ceux donnés dans la liste qui suit.

La littérature en anglais est très abondante et contient bon nombre d'ouvrages de référence, mais nous ne donnerons ici que les ouvrages en français.

1. A. Barraud et al., *Outils d'Analyse Numérique pour l'Automatique*, Hermès, Paris, 2002.

   Contient, entre autres, un chapitre sur la résolution des systèmes linéaires ainsi qu'un autre sur la méthode des moindres carrés. L'arithmétique de l'ordinateur est également traitée.

2. J. Baranger et al., *Analyse Numérique*, Hermann, Paris, 1991.

   Contient un chapitre sur la résolution des grands systèmes linéaires creux et un chapitre sur le calcul des valeurs propres.

3. C. Brezinski, *Algorithmique Numérique*, Ellipses, Paris, 1988.

   Ce livre contient un chapitre sur la méthode de Gauss ainsi que des développements concernant la propagation des erreurs dues à l'arithmétique de l'ordinateur.

4. C. Brezinski, *Initiation à la programmation linéaire et à l'algorithme du simplexe*, Ellipses, Paris, 2002.

   La méthode de Gauss est à la base de la méthode du simplexe pour le traitement des problèmes de programmation linéaire. Ce livre est une introduction à ce domaine de l'analyse numérique qui a de très nombreuses applications industrielles. Il peut être abordé sans aucune connaissance préalable.

5. F. Chatelin, *Valeurs Propres de Matrices*, Masson, Paris, 1988.

   C'est un livre de référence sur les méthodes numériques pour le calcul des valeurs propres d'une matrice. Il est écrit par une spécialiste reconnue de ce domaine.

6. P.G. Ciarlet, *Introduction à l'Analyse Numérique Matricielle et à l'Optimisation*, Masson, Paris, 1982.

   Livre extrêmement complet et fort bien écrit sur l'analyse numérique matricielle. Il requiert un bon niveau de maitrîse de mathématiques.

7. P.G. Ciarlet, J.M. Thomas, *Exercices d'Analyse Numérique Matricielle et d'Optimisation*, Masson, Paris, 1982.

   C'est le livre d'exercices qui accompagne le livre précédent.

8. M. Daumas, J.–M. Muller, *Qualité des Calculs sur Ordinateur*, Masson, Paris, 1997.

   Ce livre est une présentation de l'arithmétique de l'ordinateur, des techniques statistiques d'estimation des erreurs er de diverses arithmétiques.

9. É. Durand, *Solutions Numériques des Équations Algébriques. Tome II : Systèmes de Plusieurs Équations, Valeurs Propres de Matrices*, Masson, Paris, 1972.

   Livre assez ancien mais contenant de très nombreux algorithmes et exemples numériques.

10. N. Gastinel, *Analyse Numérique Linéaire*, Hermann, Paris, 1966.

    Le premier livre en français sur l'analyse numérique matricielle. Il contient des résultats et des méthodes qu'on ne trouve pas dans d'autres ouvrages. C'est un classique.

11. P. Lascaux, R. Théodor, *Analyse Numérique Matricielle Appliquée à l'Art de l'Ingénieur*, 2 vols., Masson, Paris, 1986.

    Écrit pour des étudiants des écoles d'ingénieurs, il ne néglige pas pour autant la théorie. C'est un très bon ouvrage, très complet, avec de nombreux exemples.

12. A. Quarteroni, R. Sacco, F. Saleri, *Méthodes Numériques pour le Calcul Scientifique. Programmes en MATLAB*, Springer–Verlag France, Paris, 2000.

    Très bon cours général d'analyse numérique avec un chapitre sur l'algèbre matricielle, un chapitre sur les méthodes directes de résolution des systèmes linéaires et un sur le calcul des valeurs propres. La précision des calculs est étudiée et les codes MATLAB sont fournis.

13. H.S. Wilf, *Algorithmes et Complexité*, Masson, Paris, 1989.

    Intéressant pour la complexité des algorithmes utilisés en algèbre matricielle.

Table des matières

Préface **1**

1 Arithmétique de l'ordinateur **3**
1.1 Représentation des nombres . . . 3
1.2 Opérations arithmétiques . . . 5
1.3 Conséquences . . . 6
1.4 Conditionnement et stabilité . . . 9
1.5 Remèdes . . . 11
1.6 Un exemple . . . 12
1.7 Exemples divers . . . 14

2 Généralités **17**
2.1 Définitions de base . . . 17
2.2 Addition et multiplication . . . 18
2.3 Inversion . . . 21
2.4 Matrices particulières . . . 25
2.5 Un peu d'algèbre linéaire . . . 27
2.5.1 Les vecteurs . . . 28
2.5.2 Les applications linéaires . . . 29
2.5.3 Les matrices . . . 29
2.5.4 Opérations sur les matrices . . . 30
2.5.5 Changement de base . . . 32
2.6 Vecteurs propres et valeurs propres . . . 33
2.7 La notion de norme . . . 36
2.7.1 Normes de vecteurs . . . 37
2.7.2 Normes de matrices . . . 40
2.7.3 Le conditionnement . . . 41
2.8 La décomposition en valeurs singulières . . . 42
2.8.1 Matrices carrées . . . 42
2.8.2 Matrices rectangulaires . . . 45
2.9 Quelques méthodes et formules utiles . . . 45
2.9.1 Méthode de bordage . . . 45
2.9.2 Complément de Schur . . . 46
2.9.3 Formule de Sherman–Morrison . . . 47
2.9.4 Identité de Sylvester . . . 47
2.10 Complexité des calculs matriciels . . . 47

3 Les systèmes linéaires **51**
3.1 Généralités sur les systèmes linéaires . . . 51
3.2 Les erreurs numériques . . . 54
3.2.1 Étude a priori . . . 55
3.2.2 Étude a posteriori . . . 58
3.2.3 Exemples . . . 59
3.2.4 Estimations de l'erreur . . . 60
3.3 Le préconditionnement . . . 61
3.4 Itération sur le résidu . . . 63
3.5 Moindres carrés . . . 64
3.6 Pseudo–inverses . . . 66
3.7 Matrices test . . . 67

4 La méthode de Gauss **71**
4.1 L'algorithme . . . 71
4.2 Mise en œuvre . . . 74
4.3 Nombre d'opérations . . . 77
4.4 Interprétation matricielle . . . 79
4.5 Le problème du remplissage . . . 82
4.6 Variantes de la méthode de Gauss . . . 83
4.6.1 Systèmes tridiagonaux . . . 84
4.6.2 La méthode de Gauss–Jordan . . . 84
4.6.3 La méthode de Gauss symétrique . . . 86
4.6.4 La méthode de Gauss par blocs . . . 88
4.7 Pseudo–codes . . . 88
4.7.1 Système triangulaire supérieur . . . 88
4.7.2 Système triangulaire inférieur . . . 89
4.7.3 Méthode de Gauss . . . 89
4.7.4 Méthode de Gauss avec pivotage partiel . . . 90
4.7.5 Factorisation de Gauss–Crout . . . 90
4.8 Expériences numériques . . . 91

5 La méthode de Cholesky **95**

6 La méthode de Householder **99**

7 Matrices creuses **105**
7.1 Origine des grands systèmes creux . . . 105
7.2 Stockage des matrices creuses . . . 107
7.3 Opérations sur les matrices creuses . . . 111

8 Calcul du polynôme caractéristique **113**
8.1 La méthode de Souriau . . . 113
8.2 Les méthodes de réduction à la forme tridiagonale . . . 115
8.2.1 La méthode de Lanczos . . . 119
8.3 Les méthodes de Givens et Householder . . . 122
8.3.1 La forme de Hessenberg . . . 122

8.3.2 La méthode de Givens 124
8.3.3 La méthode de Householder 125

9 Quelques méthodes itératives **129**
9.1 Généralités 129
9.2 Convergence de matrices 132
9.3 Méthodes de relaxation 134
9.3.1 Généralités sur la convergence 136
9.3.2 Les méthodes de Jacobi et Gauss–Seidel 139
9.3.3 Convergence des méthodes de Jacobi et de Gauss–Seidel . 142
9.4 Méthodes de Richardson 144

Bibliographie **147**